**San Jose State University**

**Computer Engineering Department**



CMPE 287 - Software Quality Assurance

**Regression Test Design Specification**

**Project – Elevator System**

**Instructor:**

**Dr. Jerry Zeyu Gao**

**Submission Date:**

23 November, 2011

**Submitted By**

Sonalika Gupta (sonalika.gupta@gmail.com) SJSU ID: 007490509

Anshu Basia (anshu.basia@gmail.com) SJSU ID: 006447987

Venkata Nagasri Lavanya Davuluri SJSU ID: 008091083

(lavanya\_davuluru@yahoo.com)

Shubhada Narkar (shubhada\_narkar@yahoo.com) SJSU ID: 008137337

Table of Contents

[List of Figures 3](#_Toc309658410)

[1. Introduction 4](#_Toc309658411)

[1.1 Regression Test Project Overview 4](#_Toc309658412)

[1.2 Structure and scope of the Document 5](#_Toc309658413)

[1.3 Test Design Document work division 6](#_Toc309658414)

[1.4 Detail Schedule 7](#_Toc309658415)

[2. Project Overview 8](#_Toc309658416)

[2.1 Overview of Updated Project 8](#_Toc309658417)

[2.2 Component Change Details 9](#_Toc309658418)

[2.3 Screenshots of updated version of Elevator system 10](#_Toc309658419)

[3. Detailed Test Strategy 13](#_Toc309658420)

[3.1 Testing Methods/Coverage 13](#_Toc309658421)

[3.1.1 Test Case ID Template 14](#_Toc309658422)

[3.1.2 Color Scheme for Test cases 14](#_Toc309658423)

[3.2 Component Testing 15](#_Toc309658424)

[3.2.1 White Box Testing 15](#_Toc309658425)

[State Based Testing for user panel component 15](#_Toc309658426)

[State based Testing for Floor Panel with AlarmIndicator 25](#_Toc309658427)

[Branch Based Testing for New Algorithm (SCAN) 31](#_Toc309658428)

[3.2.2 Black box testing 41](#_Toc309658429)

[User panel Decision Table 41](#_Toc309658430)

[User Panel Equivalence partition 51](#_Toc309658431)

[Floor Panel Decision table 61](#_Toc309658432)

[Floor Panel Equivalence partition 65](#_Toc309658433)

[3.3 System Testing 70](#_Toc309658434)

[3.3.1 Decision Table Method 70](#_Toc309658435)

[3.3.2 Equivalence Partition Method 96](#_Toc309658436)

[3.4 Configuration Testing 126](#_Toc309658437)

[4.0 SUMMARY 136](#_Toc309658438)

[References 138](#_Toc309658439)

[Appendix A: Previous Version 139](#_Toc309658440)

[Appendix B: Updated Version 139](#_Toc309658441)

[Appendix C: Component Change Records in the system 140](#_Toc309658442)

[Appendix D: System Change Records 140](#_Toc309658443)

[Appendix E: Component Change Firewall Records for updated version 141](#_Toc309658444)

[Appendix F: System Change Firewall Records for updated version 143](#_Toc309658445)

[Appendix G: Component Test Change Firewall Records of updated version 144](#_Toc309658446)

[Appendix H: System Test Change Firewall Records of updated version 153](#_Toc309658447)

# 

# List of Figures

[Figure 1: Component Diagram for the enhanced Elevator System – Version 3 4](#_Toc309658165)

[Figure 2 : UI of existing Elevator System \_v\_2 7](#_Toc309658166)

[Figure 3 : State diagram for User Panel Component 15](#_Toc309658167)

[Figure 4 : State Tree for user panel component 16](#_Toc309658168)

[Figure 5 : State Diagram for Floor Panel component 24](#_Toc309658169)

[Figure 6 : State tree for Floor Panel Component 25](#_Toc309658170)

[Figure 7: Program control flow graph for findBestCar() in SCAN Algorithm 31](#_Toc309658171)

[Figure 8: Program Control Flow Graph for partitionCars() 32](#_Toc309658172)

[Figure 9: Semantic diagram for Admin Console 126](#_Toc309658173)

[Figure 10: Spanning Tree for Admin console 129](#_Toc309658174)

[Figure 11 : Component Firewall Changed Diagram- Class Level 141](#_Toc309658175)

[Figure 12 : System Firewall Changed Diagram 143](#_Toc309658176)

[Figure 13 : White Box – State Based Testing Graph for User Panel 144](#_Toc309658177)

[Figure 14 : White Box – State Based Testing Graph for Floor Panel 144](#_Toc309658178)

[Figure 15 : White Box – Branch Testing Graph for SCAN Algorithm 145](#_Toc309658179)

[Figure 16: Summary Graph for White box Testing 146](#_Toc309658180)

[Figure 17: Black Box Decision Table test case Graph for User panel 147](#_Toc309658181)

[Figure 18 : Black Box Equivalence Partition test case graph for User Panel 148](#_Toc309658182)

[Figure 19 : Black Box Decision Table test case graph for Floor Panel 149](#_Toc309658183)

[Figure 20 : Black Box Equivalence Partition Test case graph for Floor Panel 150](#_Toc309658184)

[Figure 21 : Summary Graph for Component Black Box testing 151](#_Toc309658185)

[Figure 22 : System Testing Graph for Elevator System 152](#_Toc309658186)

# 1. Introduction

## 1.1 Regression Test Project Overview

The major objective of software testing includes uncovering as many errors as possible and to make sure that the behavior of the system matches with the requirements. Also, as part of regression testing, we need to ensure that changes made to the component do not affect the existing functionality. In this document, we provide the test documentation of the two selected components, after making changes to the existing elevator system according to the change requirements given to us. Different Black-Box and White-Box testing methodologies are used for testing the components. In Black Box unit testing, we are using the Decision Table and Equivalence partitioning methods. In White Box unit testing, we are using the State Based testing and Basis Path testing methods. We also talk about system testing and configuration testing in this document. For the system level testing, we are using the Decision Table and Equivalence Partitioning methods. The components we are testing are User Panel and Floor Panel for the newly added Alarm component and a new elevator algorithm implementation. In this document, we give all the test cases we formulated for testing each of the chosen components using all the various testing methodologies we selected. Every test case gives details about who will perform the test, which component is being tested, the pre-conditions and post-conditions, expected output, and the operational procedure.

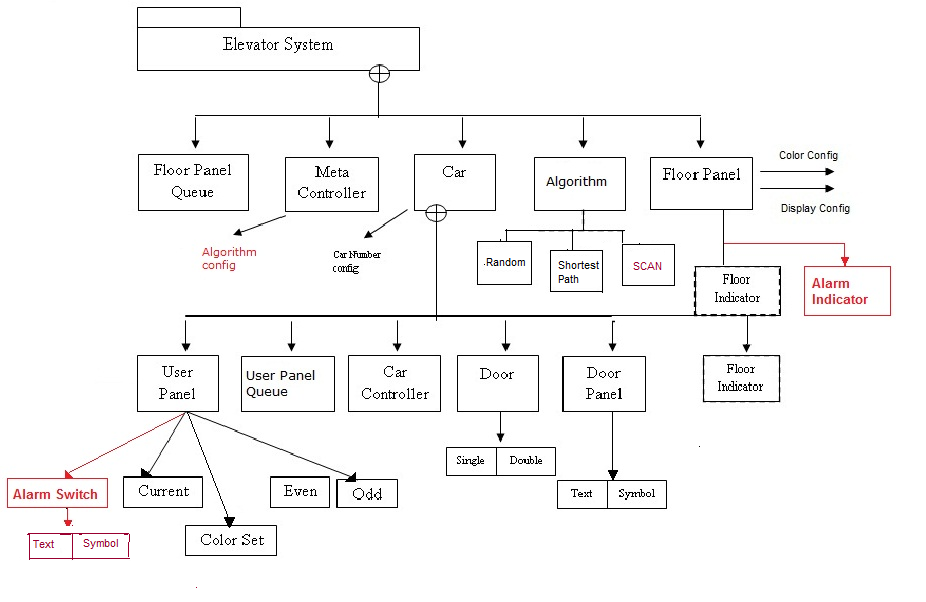


Figure 1: Component Diagram for the enhanced Elevator System – Version 3

(Changes have been marked in red)

## 1.2 Structure and scope of the Document

The scope of the document is to provide details about how the components are tested. This document starts with the detailed schedule followed by the Project overview, where we provide details of the change requirements and the updated project overview. Then there is the detailed Test Strategy and the testing methodologies are applied to come up with the test cases for the components. We also define the test coverage for each method. Finally, the document covers details about System testing and Configuration testing.

## 

## 1.3 Test Design Document work division

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **Name** | **Work done in Test Design Document and Change Requirement** |
| 1. | Anshu Basia | Source code change (Algorithm addition)  Algorithm Testing (White box)  System testing (Equivalence Partition)  Summary and References  Appendix E |
| 2. | Venkata Davuluri | Source code change (Alarm Component)  Introduction overview  Project overview – with change requirement  User Panel Testing (White box) for Alarm Component  Floor Panel Testing (White box) for Alarm Component  Appendix F  Appendix G – White Box |
| 3. | Sonalika Gupta | Detailed schedule  System Testing (Decision Table)  Appendix C – Component Change Records in the System  Appendix D – System Change Records  Appendix G – White Box  Appendix H |
| 4. | Shubhada Narkar | User Panel Testing (Black box) for Alarm Component  Floor Panel Testing (Black box) for Alarm Component  Configuration Testing  Appendix A  Appendix B  Appendix G – Black Box |

## 1.4 Detail Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Task no. | Task Performed | Period/Duration | Responsibility |
| 1 | Identifying changes and updating the code according to change requirements. | 9/20/2011- 9/24/2011 | Lavanya and Anshu |
| 2 | Integrating the code changes and setting up the test environment. | 9/25/2011 | Lavanya and Anshu |
| 3 | Finalizing test methods that shall be used for Black box and White box Testing. | 9/25/2011 | Team |
|  | Unit testing Activities |  |  |
| 4 | Design test cases of User Panel for Alarm Component for White Box testing using State based testing methodology. | 9/26/2011- 9/28/2011 | Lavanya |
| 5 | Design test cases of Floor Panel for Alarm Component for White Box testing using State based testing methodology. | 9/29/2011- 9/01/2011 | Lavanya |
| 6 | Design test cases for Algorithm for White Box testing using Branch based testing | 9/26/2011 - 10/2/2011 | Anshu |
| 7 | Design test cases of User Panel for Alarm Component for Black Box testing Using Decision Table and Equivalence Partitioning | 9/28/2011- 10/2/2011 | Shubhada |
| 8 | Design test cases of Floor Panel for Alarm Component for Black Box testing Using Decision Table and Equivalence Partitioning | 10/03/2011- 10/5/2011 | Shubhada |
|  | System Level Testing Activities |  |  |
| 9 | Design test cases for System Testing Using Decision Table | 10/06/2011-10/14/2011 | Sonalika |
| 10 | Design test cases for System Testing Using Equivalence Partitioning | 10/06/2011-10/14/2011 | Anshu |
|  | Configuration Testing Activity |  |  |
| 11 | Design the test cases for system level Configuration testing using Semantic tree method. | 10/06/2011-10/12/2011 | Shubhada |
|  | Other activities |  |  |
| 12 | Writing and managing test cases for all testing activity in Elementool | 10/19/2011-10/22/2011 | Team |
| 13 | Implementing all aforementioned activities and updating Regression Test Design Specification Document. | 10/22/2011- 10/23/2011 | Team |
| 14 | Updating System Information in Appendices | 10/24/2011 - 10/25/2011 | Sonalika and Shubhada |
| 15 | Firewall Change Records | 11/16/2011 – 11/22/2011 | Team |

# 2. Project Overview

## 2.1 Overview of Updated Project

Figure 2 shows the UI of the existing Elevator System (version2).

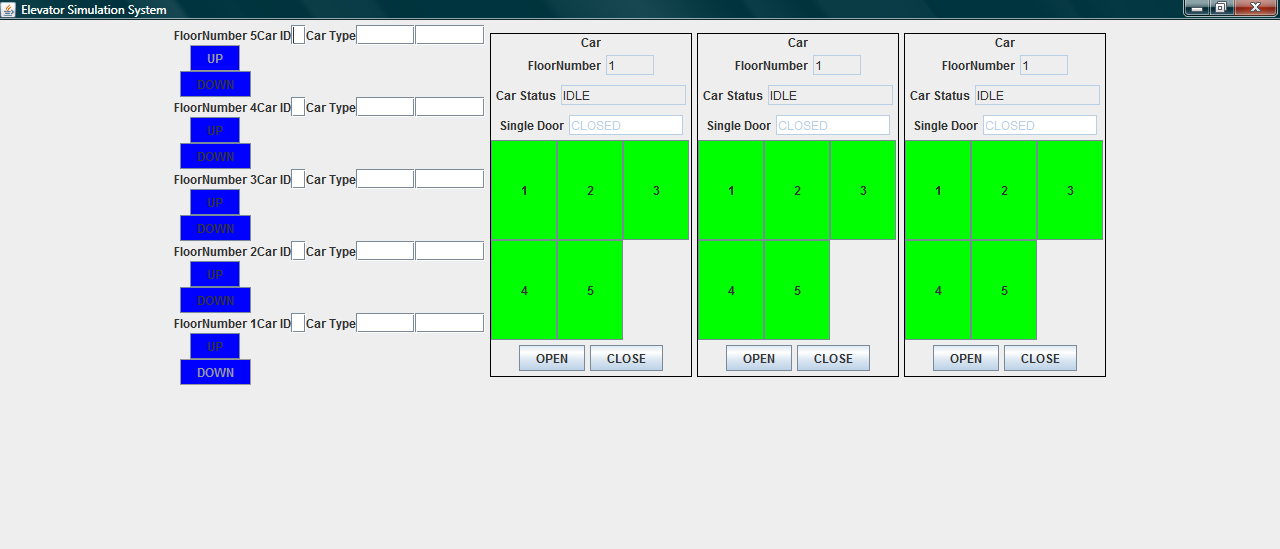


Figure 2 : UI of existing Elevator System \_v\_2

Change Requirement and Implementation

* Add an internal Alarm Component with a switch function in the user panel of each elevator Car. Provide two different models for your Alarm component to allow your customers to select
* Add an external Alarm component with an indication function on each floor for the elevator system.
* Update your floor panel component to accommodate the external Alarm component.
* Add one more elevator protocol (or algorithm) to control the movement of your Elevator car and update your Elevator Controller to support this as one of the options.

Enhance the existing configuration user interface to support the selection and configuration of this Alarm function and features.

## 2.2 Component Change Details

User Panel\_v3

We added an Alarm switch in the user panel with on/off functionality. We added a new parameter “Alarm type” to enable the user to select the type of Alarm (text/symbol) on the user panel in each Elevator car.

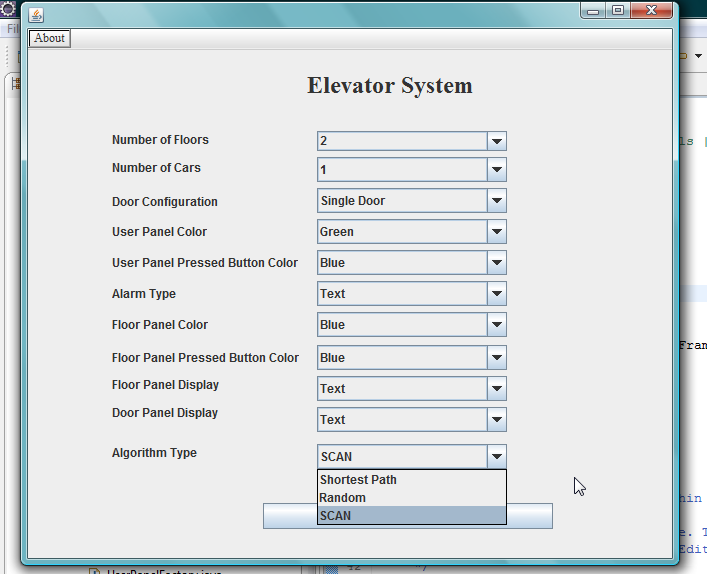
Floor Panel\_v3

We added a new Alarm indicator which corresponds to the alarm in every elevator car on the floor panel of every floor. When a user presses an alarm in any car, the alarm indicator for that car on all floors is activated.

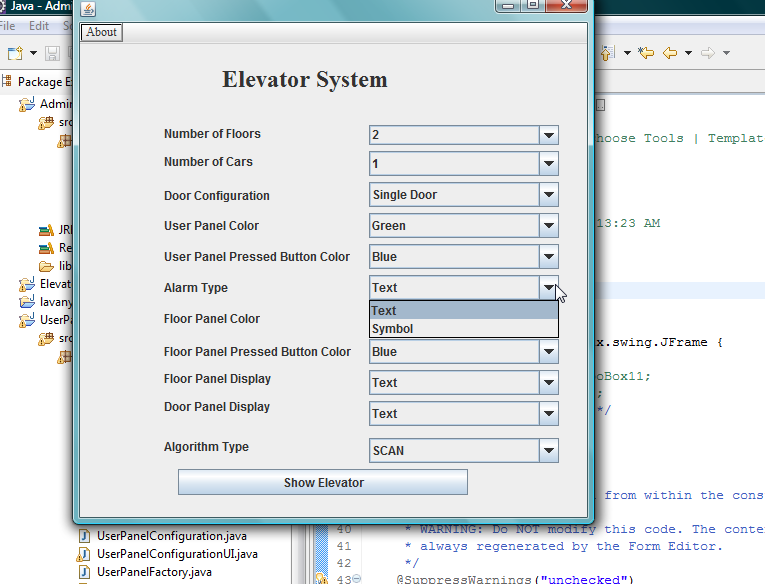
Algorithm\_v3

We added a new Elevator protocol (SCAN) to control the movement of the elevator cars. The UI is updated with this as another choice in the ‘Algorithm Type’ selection.

## 2.3 Screenshots of updated version of Elevator system

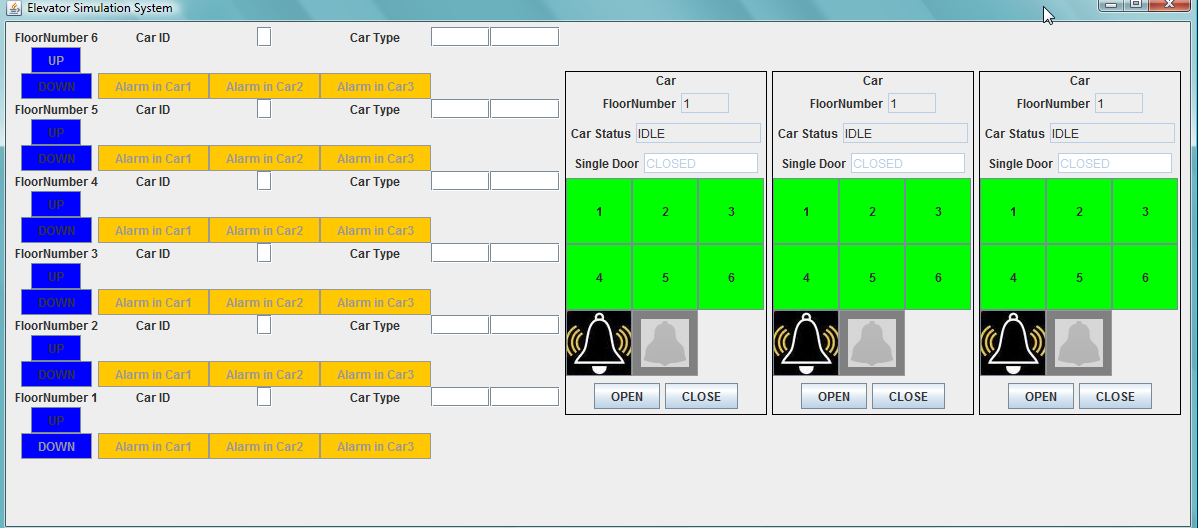
Providing choice of new Elevator Algorithm - SCAN

UserPanel Configuration – Two types of Alarm Switch

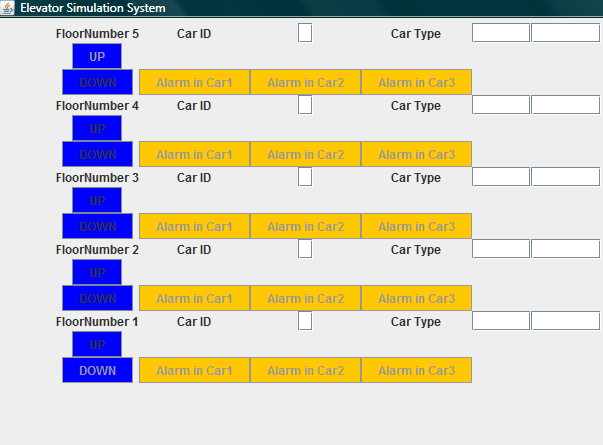


UserPanel – AlarmOn and AlarmOff - Text and Symbol

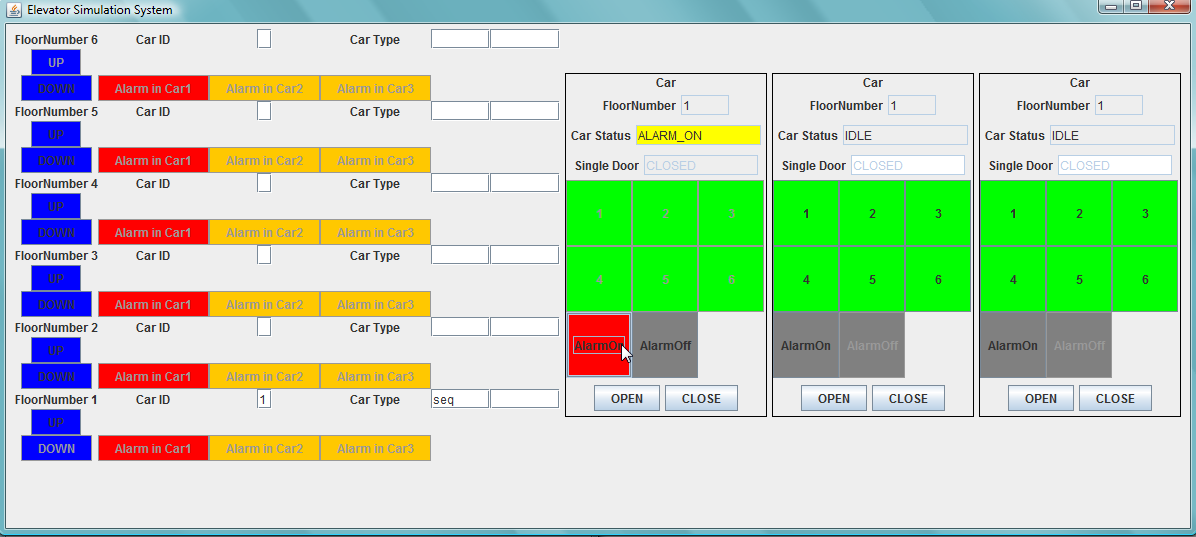


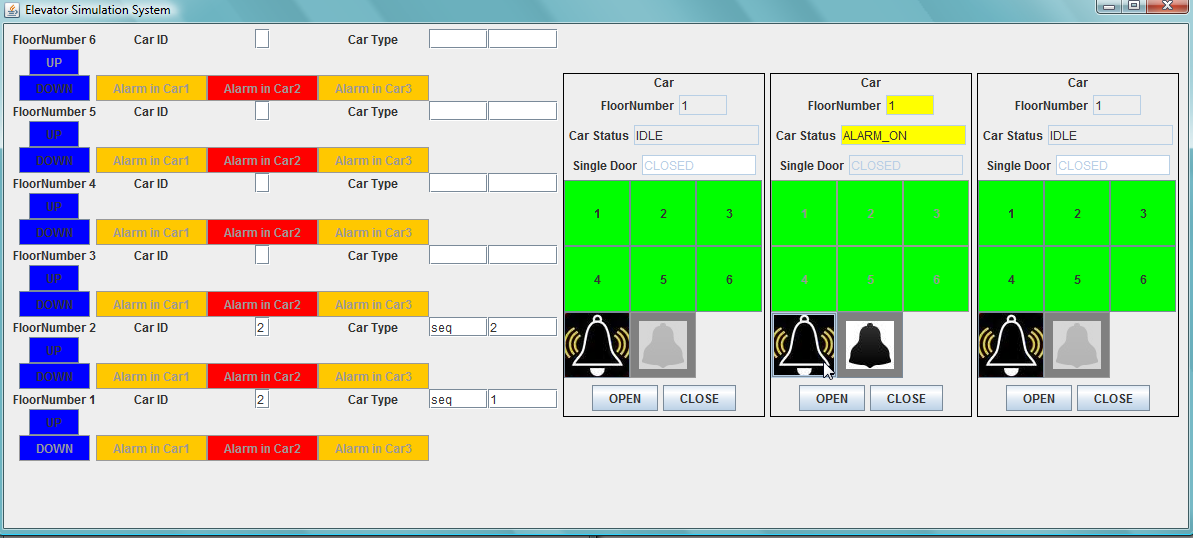


FloorPanel – Alarm Indicator for each car on all floors



UserPanel and FloorPanel – when Alarm Pressed





# 3. Detailed Test Strategy

## 3.1 Testing Methods/Coverage

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Testing Activity | Method Name | Coverage |
| 1. | Unit Testing - Black Box Testing | Decision Table based Testing | Decision Table coverage |
| Equivalence Testing | All Equivalence classes coverage |
| 2. | Unit Testing - White Box Testing | State based Testing | All States & Transition Link Coverage |
| 3. | System Level Functional testing | Decision Table Testing | Conditions and Actions (In Decision Table) coverage |
| Equivalence Partition Testing | All Equivalence classes coverage |
| 4. | System Level Configuration Testing | Semantic Tree Method | Semantic Tree Coverage |

### 3.1.1 Test Case ID Template

|  |  |  |  |
| --- | --- | --- | --- |
| Test case method | | | Test Case Id |
| Unit Testing | White Box | State-based | WB-SB-(number) |
| Black Box | Decision Table | BB-DT-(number) |
| State-based | BB-SB-(number) |
|  |  | Equivalence Partition | BB-EP-(number) |
| System Testing | Black Box | Decision Table | ST-DB-(number) |
| Equivalence Partition | ST-EP-(number) |

### 3.1.2 Color Scheme for Test cases

|  |  |
| --- | --- |
| New test Cases | Blue |
| Reused test Cases | Red |
| Modified /Changed test cases | Green |

## 

## 3.2 Component Testing

### 3.2.1 White Box Testing

#### State Based Testing for user panel component

State Based Method:



Figure 3 : State diagram for User Panel Component

State Tree:



Figure 4 : State Tree for user panel component

Yellow – Updated states

Green – New States

Red – New Transitions

New States :

S2a, S3a, S4a, S2b, S3b, S7b.

Updated States

S1 – Added AlarmConfiguration to the existing UserPanel Configuration.

S7a – Added new transition paths from Car STOP status.

List of all States :

S0 : Initial State

S1 : Configuration State

S2 : User Panel Idle State

S2a : AlarmConfig set – AlarmOn enabled

S2b : AlarmOff State

S3 : Dequeue Floor Number

S3a : AlarmOn -UserPanel Deactivate

S3b: AlarmOff - UserPanel Activate

S4 : User Panel queue empty

S4a: FlushUserPanelQueue

S5 : Car is active

S6 : Enqueue Floor Number

S7 : Car is Idle

S7a: Car stopped

S7b: Car status AlarmOn

S8 : Deactivate user panel button color

List of all Transitions:

T01 - Configuring User Panel.

T12 - User Panel activated

T23 - Request is served, destination floor = request floor.

T34 – Remove from queue

T42 – User panel is set to idle

T47 – Car is reached destination and is idle

T38 – User panel is deactivate/disabled after dequeue.

T81 – User panel button color deactivated.

T35, T56 - Enqueue and Dequeue due to more request.

T26 - Request for floor number.

T65 - Server request from queue, car is active.

T56 – Enqueue to server more request

T12a – AlarmConfig set

T2a3a – AlarmPressed - UserPanelDeactivate

T3a4a – Remove all requests from queue

T4a6 – Enqueue floor number 1

T67a – Car stopped at floor#1

T7a7b – AlarmStatus On – set CarStatus

T7a7 – AlarmStatus Off – set CarStatus Idle

T3a2b – AlarmOff pressed

T2b3b - UserPanel Activated

T3b7 – Set CarStatus Idle

Test Cases for UserPanel AlarmComponent :

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-01 | | **Documented Date:** 09/26/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** User panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate Alarm Configuration in User Panel | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0 – T01 – S1 – T12a – S2a | | | |
| **Pre-conditions:**  Pointer should be at initial node.  Node should have a branch to make transition  UserPanel in idle state and AlarmType is text. | | **Post-conditions:**  Transition should be successful. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | The component is executed. Values are set  No. of floor=5  Default button color=red  Activate button color=green  car type=default  User panel queue=default  User Panel Type=Sequential  UserPanel AlarmType = Text  OK button event is triggered | The configuration is set and all the configured values are set i.e  numberFloors=5  activeButtonColor=red  carType=default  queueType=default  User Panel Type=Sequential  User Panel AlarmType = text |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-02 | | **Documented Date:** 09/26/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** User panel | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate Alarm Configuration in User Panel | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0 – T01 – S1 – T12a – S2a | | | |
| **Pre-conditions:**  Pointer should be at initial node.  Node should have a branch to make transition  UserPanel in idle state and AlarmType is symbol. | | **Post-conditions:**  Transition should be successful. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | The component is executed. Values are set  No. of floor=5  Default button color=red  Activate button color=green  car type=default  User panel queue=default  User Panel Type=Sequential  UserPanel AlarmType = Symbol  OK button event is triggered | The configuration is set and all the configured values are set i.e  numberFloors=5  activeButtonColor=red  carType=default  queueType=default  User Panel Type=Sequential  User Panel AlarmType = symbol |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-03 | | **Documented Date:** 09/26/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** User panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate AlarmOn pressed in UserPanel | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0 – T01 – S1 – T12a – S2a – T2a3a – S3a | | | |
| **Pre-conditions:**  Pointer should be at initial node.  Node should have a branch to make transition  UserPanel in idle state and AlarmConfig is set | | **Post-conditions:**  Transition should be successful. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | AlarmOn button is pressed | AlarmOn button is activated  UserPanel deactivated  AlarmOff button is enabled. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-04 | | **Documented Date:** 09/26/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** User panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that UserPanelQueue is emptied when AlarmOn is pressed | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0 – T01 – S1 – T12a – S2a – T2a3a – S3a – T3a4a – S4a | | | |
| **Pre-conditions:**  Pointer should be at initial node.  Node should have a branch to make transition  UserPanel in idle state and AlarmConfig is set | | **Post-conditions:**  Transition should be successful. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Car at floor1  Floor number 2 pressed  AlarmOn pressed  Car at floor 4  Floor number 3 pressed  AlarmOn pressed | Floor request 2 from UserPanelQueueUp is removed  Floor request 3 from UserPanelQueueDown is removed. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-05 | | **Documented Date:** 09/26/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** User panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that floor #1 is enqueued when alarmOn is pressed. | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0 – T01 – S1 – T12a – S2a – T2a3a – S3a – T3a4a – S4a – T4a6 – S6 | | | |
| **Pre-conditions:**  Pointer should be at initial node.  Node should have a branch to make transition  UserPanel in idle state and AlarmConfig is set. | | **Post-conditions:**  Transition should be successful. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | AlarmOn is pressed. UserPanelQueue is emptied. | Floor request #1 is enqueued. Car moves down. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-06 | | **Documented Date:** 09/26/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** User panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that UserPanel is activated when AlarmOff is pressed | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0 – T01 – S1 – T12a – S2a – T2a3a – S3a – T3a2b – S2b – T2b3b – S3b | | | |
| **Pre-conditions:**  Pointer should be at initial node.  Node should have a branch to make transition  UserPanel in idle state and AlarmOn is pressed. | | **Post-conditions:**  Transition should be successful. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | AlarmOff pressed | Activate AlarmOff button  Deactivate AlarmOn button  Enable all UserPanel buttons. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-07 | | **Documented Date:** 09/26/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** User panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that Car reaches floor 1 and stops when AlarmOn is pressed. | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0 – T01 – S1 – T12a – S2a – T2a3a – S3a – T3a4a – S4a – T4a6 – S6 – T67a – S7a | | | |
| **Pre-conditions:**  Pointer should be at initial node.  Node should have a branch to make transition  UserPanel in idle state and AlarmConfig is set. | | **Post-conditions:**  Transition should be successful. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | AlarmOn is pressed. UserPanelQueue is emptied.  Floor request # 1 is enqueued | Car moves to floor 1 and stops. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-08 | | **Documented Date:** 09/27/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** User panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that Car status set to ALARMON when AlarmOn is pressed.. | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0 – T01 – S1 – T12a – S2a – T2a3a – S3a – T3a4a – S4a – T4a6 – S6 – T67a – S7a – T7a7b – S7b | | | |
| **Pre-conditions:**  Pointer should be at initial node.  Node should have a branch to make transition  UserPanel in idle state and AlarmConfig is set. | | **Post-conditions:**  Transition should be successful. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | AlarmOn is pressed. UserPanelQueue is emptied.  Car stopped at floor #1 | Car status is set to “ALARM ON” |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-09 | | **Documented Date:** 09/27/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** User panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that Car status is reset to “IDLE” when AlarmOff is pressed. | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0 – T01 – S1 – T12a – S2a – T2a3a – S3a – T3a2b – S2b – T2b3b – S3b – T3b7 – S7 | | | |
| **Pre-conditions:**  Pointer should be at initial node.  Node should have a branch to make transition  UserPanel in idle state and AlarmOn is pressed. | | **Post-conditions:**  Transition should be successful. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | AlarmOff pressed .  UserPanel activated | Car Status shows “IDLE” |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

#### State based Testing for Floor Panel with AlarmIndicator

State Diagram for Floor Panel Component:



Figure 5 : State Diagram for Floor Panel component

FloorPanel State Tree :



Figure 6 : State tree for Floor Panel Component

Yellow - Updated State

Green - New State

Red - New Transition

New States:

S1a, S2a, S2b

Updated States:

S0 – Added AlarmIndicator in FloorPanel UI

List of all States:

S0a - Initial state of the tree diagram from where the execution starts

S0 – FloorPanelUI set

S1 - Down button is pressed in floor panel

S1a – ProcessAlarmRequest

S2 - FP\_Active (floor panel is in active state)

S2a – AlarmIndicatorOn

S2b - AlarmIndicatorOff

S3 - activeButtonColor configurable

S3a- Floor Indicator configurable

S4 - Command is sent to the queue in floor panel queue

S4a- Update floor indicator status

S5 - FP\_Idle (floor panel is in idle state)

S6 - activeButtonColor null

S7 - Up button is pressed in floor panel

S8 - Enqueued

S9 - Dequeued

S10 - Disable up button

S11 - Disable down button

S12 - Disable floor panel UI

**List of all Transitions:**

The following are the transitions in the above tree diagram

T0a0 – Set FloorPanelConfig parameters

T01 - Press the down button in the floor panel

T01a – Alarm pressed in UserPanel – processAlarmRequest called from CarController

T1a2a – AlarmOn

T1a2b - AlarmOff

T12, T72 - The floor panel is changed to active state

T13, T73 - The button color is activated i.e configured**,**

T13a- floor indicator is activated for down button.

T14, T74, T94 - Command is sent to the queue awaiting to get processed,req.

T14a - process and update the floor indicator status.

T413 - The down button of the floor panel is disabled

T105, T115 - The floor panel is changed to idle state

T106, T116 - The button color is deactivated i.e set to null

T07 - Press the up button in the floor panel

T414 - The up button of the floor panel is disabled

T08 - The request of the floor panel is enqueued

T89 - The request is dequeued

T410 - disable the up button in the floor panel

T411 - disable the down button in the floor panel

T1012, T1112 - disable the floor panel.

Test Cases for Floor Panel Component for Alarm Indicator:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-01 | | **Documented Date:** 09/27/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** FloorPanel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that AlarmIndicator is shown in FloorPanel for each car | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0a – T0a0 – S0 | | | |
| **Pre-conditions:**  FloorPanelConfig is set pressed. | | **Post-conditions:**  FloorPanel should contain AlarmIndicators | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Number Of Cars = 4 | There should be 4 AlarmIndicators in the floorpanel |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-02 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** FloorPanel | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that AlarmIndicator is shown in FloorPanel for each car on all floors | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0a – T0a0 – S0 | | | |
| **Pre-conditions:**  FloorPanelConfig is set. | | **Post-conditions:**  FloorPanel should contain AlarmIndicators | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Number Of Cars = 4  Number Of Floors = 3 | There should be 4 AlarmIndicators in the floorpanel on all 3 floors |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-02 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** FloorPanel | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that AlarmIndicator is shown in FloorPanel for each car on all floors | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0a – T0a0 – S0 | | | |
| **Pre-conditions:**  FloorPanelConfig is set. | | **Post-conditions:**  FloorPanel should contain AlarmIndicators | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Number Of Cars = 4  Number Of Floors = 3 | There should be 4 AlarmIndicators in the floorpanel on all 3 floors |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-03 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** FloorPanel | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that AlarmIndicator is shown in FloorPanel for each car on all floors | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0a – T0a0 – S0 – T01a – S1a | | | |
| **Pre-conditions:**  FloorPanel is active. | | **Post-conditions:**  Transition should be successful | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Alarm button pressed in UserPanel | processAlarmRequest(alarmStatus,carId) of FloorPanel should be invoked from CarController. |  |
| **Required test scripts: none** | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-04 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** FloorPanel | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that when AlarmOn is pressed in any car, the respective AlarmIndicator should be On in the floorpanel. | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0a – T0a0 – S0 – T01a – S1a – T1a2a – S2a | | | |
| **Pre-conditions:**  FloorPanel is active. | | **Post-conditions:**  Transition should be successful | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | AlarmOn pressed in UserPanel of Car2  AlarmOn is pressed in UserPanel of Car5 | AlarmIndicator of Car2 should be on in the floorpanel of all floors.  AlarmIndicator of Car5 should be on in the floorpanel of all floors. |  |
| **Required test scripts: none** | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-SB-05 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Venkata Davuluri | | **Test Item:** FloorPanel | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** State based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate that when AlarmOn is pressed in any car, the respective AlarmIndicator should be On in the floorpanel. | | | |
| **Operation procedure:**  Select a Level 0 node  Make transition to its next node  S0a – T0a0 – S0 – T01a – S1a – T1a2b – S2b | | | |
| **Pre-conditions:**  FloorPanel is active.  AlarmIndicator is On | | **Post-conditions:**  Transition should be successful | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Alarm Off pressed in User Panel of Car2  AlarmOff is pressed in UserPanel of Car5 | AlarmIndicator of Car2 should be off in the floorpanel of all floors.  AlarmIndicator of Car5 should be off in the floorpanel of all floors. |  |
| **Required test scripts: none** | | | |
| **Comments:** | | | |

#### Branch Based Testing for New Algorithm (SCAN)

As per the change requirement specification, a new algorithm was needed to control the movement of elevator car. As a result, an entirely new algorithm has been developed, which will be called from MetaController component of the system in order to find best suitable car to fulfill user request according to disk scheduling algorithm. This algorithm also supports previous version odd and even algorithm and the functionality is implemented in the same way by partitioning the car. Hence, partition Cars() has been copied directly into the algorithm source code file.

Note: New test cases are represented in Blue.

Reused test cases are represented in Red

Modified test cases are represented in Green



Figure 7: Program control flow graph for findBestCar() in SCAN Algorithm



Figure 8: Program Control Flow Graph for partitionCars()

Branch-based Table

1) findBestCar(List<ICar> lstCars, Direction direction, int destinationFloorNumber)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Predicate Node | T/F | Test case | | | | | | | |
| T1 | T2 | T3 | T4 | T5 | T6 | T7 |
| lstCars.size() ==1 | T | x |  |  |  |  |  |  |
| F |  | x1 | x1 | x1 | x1 | x1 | x1 |
| i<lstCars.size() | T |  | x2 | x2 | x2 | x2 |  |  |
| F |  |  |  |  |  | x2 | x2 |
| (direction==Direction.UP) & (car.getStatus()==CarStatus.MOVING\_UP) || ((car.getStatus()==CarStatus.STOPPED) & (car.getUserPanelQueue().isRequestUp()==1))& (car.getCurrentFloorNumber()<=destinationFloorNumber) | T |  | x3 |  |  |  |  |  |
| F |  |  | x3 | x3 | x3 |  |  |
| (direction == Direction.DOWN) & ((car.getStatus() == CarStatus.MOVING\_DOWN) || ((car.getStatus() == CarStatus.STOPPED) & (car.getUserPanelQueue().isRequestDown() == 1) )) & (car.getCurrentFloorNumber() >= destinationFloorNumber ) | T |  |  | x4 |  |  |  |  |
| F |  |  |  | x4 | x4 |  |  |
| car.getStatus() == CarStatus.IDLE | T |  |  |  | x5 |  |  |  |
| F |  |  |  |  | x5 |  |  |
| bestCar == null | T |  |  |  |  |  | x3 |  |
| F |  | x4 | x5 |  |  |  | x3 |

Table : Branch table for findBestCar()

2) partitionCars(List<ICar>,lstCars, intdestinationFloorNumber) – for odd/even

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Predicate Node | T/F | Test case | | | | |
| T1 | T2 | T3 | T4 | T5 |
| i<lstCars.size() | T | x1 | x1 | x1 |  |  |
| F |  |  |  | x1 | x1 |
| us.getSelection() == 1 (userpanel is odd) | T | x2 |  |  |  |  |
| F |  | x2 | x2 |  |  |
| us.getSelection() == 2 (userpanel is even) | T |  | x3 |  |  |  |
| F | x3 |  | x3 |  |  |
| desitnationFloorNumber % 2 ==0 (check floor number even/odd) | T |  |  |  | x2 |  |
| F |  |  |  |  | x2 |

Table : Branch table for partitionCars()

Test Cases for branch based method

1) findBestCar(List<ICar> lstCars, Direction direction, int destinationFloorNumber)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-01 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** SCAN Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size() == 1 | | | |
| **Operation procedure:**  1. Pass list of car to sorting algorithm. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN  Number of Cars : 1 | | **Post-conditions:**  One car is present in the system hence, it is returned as Best Car to serve the floor request. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars, direction and destination floor number | Car object |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-02 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** SCAN Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size()>1 and predicate node for direction.UP, carStatus and carCurrentFloor | | | |
| **Operation procedure:**   1. Pass list of cars to sorting algorithm. 2. Direction of floor request = UP 3. Status of first car is MOVING\_UP 4. Destination floor number > Car’s current floor number. 5. First car is returned as best car | | | |
| **Pre-conditions:**  Algorithm Type: SCAN  Number of Cars : > 1 | | **Post-conditions:**  Best car to serve the request is returned. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars, direction and destination floor number | Car object |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-03 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** SCAN Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size()>1 and predicate node for direction.DOWN, carStatus and carCurrentFloor | | | |
| **Operation procedure:**   1. Pass list of cars to sorting algorithm. 2. Direction of floor request = DOWN 3. Status of first car is MOVING\_DOWN 4. Destination floor number < Car’s current floor number. 5. First car is returned as best car | | | |
| **Pre-conditions:**  Algorithm Type: SCAN  Number of Cars : > 1 | | **Post-conditions:**  Best car to serve the request is returned. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars, direction and destination floor number | Car object |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-04 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** SCAN Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size()>1 and predicate node for car.Status = IDLE | | | |
| **Operation procedure:**   1. Pass list of cars to sorting algorithm. 2. Direction = UP or DOWN 3. Status of first car is IDLE 4. First car is returned to bestCar object | | | |
| **Pre-conditions:**  Algorithm Type: SCAN  Number of Cars : > 1 | | **Post-conditions:**  Car to serve the request is found. (It might not be the best car) | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars, direction and destination floor number | Car object |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-05 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** SCAN Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size()>1 and predicate nodes for direction, carStatus and carCurrentFloor | | | |
| **Operation procedure:**   1. Pass list of cars to sorting algorithm. 2. No. of cars = 2 3. Direction of floor request = UP 4. Status of all cars = STOPPED or MOVING\_UP 5. First car current floor = 7 6. Second car current floor = 5 7. Destination floor number = 2 8. No car is returned as best car i.e. bestCar = null | | | |
| **Pre-conditions:**  Algorithm Type: SCAN  Number of Cars : = 2 | | **Post-conditions:**  No Best car is found (bestCar=null). | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars, direction and destination floor number | Car object |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-06 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** SCAN Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size()>1 and predicate nodes bestCar == null | | | |
| **Operation procedure:**   1. Pass list of cars to sorting algorithm. 2. No. of cars = 2 3. Direction of floor request = UP 4. Status of all cars = STOPPED or MOVING\_UP 5. First car current floor = 7 6. Second car current floor = 5 7. Destination floor number = 2 8. No car is returned as best car 9. bestCar = null 10. A car is randomly selected and returned as bestCar | | | |
| **Pre-conditions:**  Algorithm Type: SCAN  Number of Cars : = 2 | | **Post-conditions:**  A random car is returned as best car. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars, direction and destination floor number | Car object |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-07 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** SCAN Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size()>1 and predicate nodes bestCar == null | | | |
| **Operation procedure:**   1. Pass list of cars to sorting algorithm. 2. First car status = MOVING\_UP and current floor number = 4 3. Direction of floor request = UP 4. Destination floor number = 8 5. First car is returned as best car. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN  Number of Cars : > 1 | | **Post-conditions:**  Best car is found and returned | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars, direction and destination floor number | Car object |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

2) partitionCars(List<ICar>,lstCars, int destinationFloorNumber) – for odd/even

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-08 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** odd/even Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size()>i and predicate nodes us.getSelection == 1 and us.getSelection == 2 | | | |
| **Operation procedure:**   1. Pass list of cars to partitionCars().   2. UserPanel is of type odd so add to oddCarsList().   1. UserPanel is not of type even. | | | |
| **Pre-conditions:**  UserPanel type: seq or odd  Algorithm Type: SCAN – odd/even  Number of Cars : > 1 | | **Post-conditions:**  List of cars is updated | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars and destination floor number | Updated list of cars |  |
| **Required test scripts: -** None | | | |
| **Comments:** odd/even algorithm is a part of SCAN algorithm | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-09 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** odd/even Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size()>i and predicate nodes us.getSelection == 1 and us.getSelection == 2 | | | |
| **Operation procedure:**   1. Pass list of cars to partitionCars(). 2. UserPanel is of type even so add to evenCarsList(). 3. UserPanel is not of type odd. | | | |
| **Pre-conditions:**  UserPanel type: seq or even  Algorithm Type: SCAN – odd/even  Number of Cars : > 1 | | **Post-conditions:**  List of cars is updated | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars and destination floor number | Updated list of cars |  |
| **Required test scripts: -** None | | | |
| **Comments:** odd/even algorithm is a part of SCAN algorithm | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-10 | | **Documented Date:** 09/28/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** odd/even Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size()>i and predicate nodes us.getSelection == 1 and us.getSelection == 2 | | | |
| **Operation procedure:**   1. Pass list of cars to partitionCars(). 2. UserPanel is not of type odd or even. 3. UserPanel is of type seq hence, add to both oddListCars and evenListCars | | | |
| **Pre-conditions:**  UserPanel type: seq  Algorithm Type: SCAN – odd/even  Number of Cars : > 1 | | **Post-conditions:**  List of cars is updated | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars and destination floor number | Updated list of cars |  |
| **Required test scripts: -** None | | | |
| **Comments:** odd/even algorithm is a part of SCAN algorithm | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-11 | | **Documented Date:** 10/01/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** odd/even Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size()>i and desitnationFloorNumber % 2 ==0 (check floor number even/odd) | | | |
| **Operation procedure:**  1. Pass list of car to partitionCars() function.  2. Partition List of car.  3. Request for even floor number.   1. Pass even list. | | | |
| **Pre-conditions:**  UserPanel type: seq, even  Algorithm Type: SCAN – odd/even  Number of Cars : = 2  Destination Floor: = 4 | | **Post-conditions:**  List of cars is updated | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars and destination floor number | Updated list of cars returned |  |
| **Required test scripts: -** None | | | |
| **Comments:** odd/even algorithm is a part of SCAN algorithm | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** WB-BB-12 | | **Documented Date:** 10/01/2011 | |
| **Tester Name:** Anshu Basia | | **Test Item:** odd/even Algorithm | |
| **Product Name:** Elevator System | | **Version No.: 3** | |
| **Test Type:** Branch based White Box Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validating predicate node lstCars.size()>i and desitnationFloorNumber % 2 ==0 (check floor number even/odd) | | | |
| **Operation procedure:**  1. Pass list of car to partitionCars() function.  2. Partition List of car.  3. Request for odd floor number.  4. Pass odd list. | | | |
| **Pre-conditions:**  UserPanel type: seq, odd  Algorithm Type: SCAN – odd/even  Number of Cars : = 2  Destination Floor: = 5 | | **Post-conditions:**  List of cars is updated | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | List of cars and destination floor number | Updated list of cars |  |
| **Required test scripts: -** None | | | |
| **Comments:** odd/even algorithm is a part of SCAN algorithm | | | |

### 3.2.2 Black box testing

#### User panel Decision Table

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CONDITIONS |  | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 | T11 |
| C1 | No.of floors (2-20) selected | T | T | T | T | T | T | T | T | T | T | T |
| C2 | Floor button color selected from drop down menu | T | T | T | T | T | T | T | T | T | T | T |
| C3 | Floor Button pressed color selected same as in C2 | T | T | F | T | F | F | F | F | F | F | F |
| C4 | Floor Button pressed color selected different than selected in C2 | F | F | T | F | T | T | T | T | T | T | T |
| C5 | Car type selected ( test car) | T | T | T | T | T | T | T | T | T | T | T |
| C6 | User panel queue type selected( test user panel queue) | T | T | T | T | T | T | T | T | T | T | T |
| C7 | User panel number type selected(current) | T | F | F | T | T | T | T | T | T | T | T |
| C7.1 | Even | F | F | T | F | F | F | F | F | F | F | F |
| C7.2 | Odd | F | T | F | F | F | F | F | F | F | F | F |
| C8 | Alarm Type selected(text) | T | T | F | F | F | F | F | F | F | F | F |
| C8.1 | Alarm type selected (symbol) | F | F | T | T | T | T | T | T | T | T | T |
| C9 | Ok button pressed on user panel configuration UI | F | T | T | T | T | T | T | T | T | T | T |
| C10 | After another interface appears |  |  |  |  |  |  |  |  |  |  |  |
| C10.1 | Floor no. button selected from available no. of floor buttons | - | F | F | T | T | F | F | T | T | F | T |
| C10.2 | Alarm switch OFF pressed when it was in disabled state. | - | F | F | F | F | T | F | F | F | F | F |
| C10.3 | Alarm switch ON pressed when car is in idle state | - | F | F | F | F | F | T | T | F | F | F |
| C10.4 | Alarm On switch is pressed when car is in moving state |  | F | F | F | F | F | F | F | T | F | F |
| C10.5 | Alarm switch OFF pressed when it was in enabled state.(That is when alarm switch was in On state before) | - | F | F | - | - | F | - | F | F | T | T |
| ACTIONS |  |  |  |  |  |  |  |  |  |  |  |  |
| A1 | User panel configuration UI will remain Idle | X |  |  |  |  |  |  |  |  |  |  |
| A2 | Another user interface dispalyed with all the selected floor, button color options. |  | X | X | X | X | X | X | X | X | X | X |
| A2.1 | Panel type Current |  |  |  | X | X | X | X | X | X | X | X |
| A2.2 | Panel type even |  |  | X |  |  |  |  |  |  |  |  |
| A2.3 | Panel type odd |  | X |  |  |  |  |  |  |  |  |  |
| A2.4 | Alarm On switch is in enabled state and alarm Off switch is in disabled state with ‘text’ format. |  | X |  |  |  |  |  |  |  |  |  |
| A2.5 | Alarm On switch is in enabled state and alarm Off switch is in disabled state with ‘Symbol’ format |  |  | X | X | X |  |  |  |  |  |  |
| A3 | Floor Button color does not change to one selected in C3 after being pressed. |  |  |  | X |  |  |  |  |  |  |  |
| A4 | Floor button color changes to one selected in C3 after being pressed |  |  |  |  | X |  |  |  |  |  | X |
| A5 | Alarm switch ON button color changes to Red |  |  |  |  |  |  | X | X | X |  |  |
| A6 | Alarm switch OFF button is enabled. |  |  |  |  |  |  | X | X | X |  |  |
| A7 | Floor buttons color remains same as in C2. |  |  |  |  |  |  |  | X |  |  |  |
| A8 | Floor button color changes to color selected in C2 |  |  |  |  |  |  |  |  | X |  |  |
| A9 | Alarm Switch ON color changes back to enable state color |  |  |  |  |  |  |  |  |  | X |  |
| A10 | Alarm switch On is in enable state. |  |  |  |  |  |  |  |  |  | X |  |
| A11 | Alarm switch Off is put back in disable state. |  |  |  |  |  | X |  |  |  | X | X |
|  |  | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 | T11 |

New test cases: Blue Reused test cases: red Modified test cases : Green

We will now generate one test case with respect to each condition according to the Decision based method.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-01 | | **Documented Date:** 9/29/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: Check the output for User Panel component inputs on User Panel | | | |
| **Operation procedure:** The Main.java file is run and a new frame opens which takes input parameters from the user. | | | |
| **Pre-conditions:**  From User Panel component, run Main.java. | | **Post-conditions:**  User Configuration UI should remain idle and wait for user input. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Ok button is not pressed after filling all input parameters on User Panel | User Panel configuration UI will remain idle. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-02 | | **Documented Date:** 9/29/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Check the output for User Panel component inputs on User Panel | | | |
| **Operation procedure:** The Main.java file is run and a new frame opens which takes input from the user and press OK button. | | | |
| **Pre-conditions:**  From User Panel component, run Main.java. | | **Post-conditions:**  New User Configuration UI should remain idle and wait for user input. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Ok button is pressed after filling all input parameters on User Panel with Alarm type as ‘Text’ . | Another interface with Alarm On switch in enabled state and OFF switch in disabled state with ‘Text type’ will appear along with all other selections in User Panel. |  |
| 2 | No input given for next UI. | Next UI will remain idle due to no further input. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-03 | | **Documented Date:** 9/29/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Check the output for User Panel component inputs on User Panel | | | |
| **Operation procedure:** The Main.java file is run and a new frame opens which takes input from the user and press OK button. | | | |
| **Pre-conditions:**  From User Panel component, run Main.java. | | **Post-conditions:**  New User Configuration UI should remain idle and wait for user input. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Ok button is pressed after filling all input parameters on User Panel with Alarm type as ‘Symbol’ . | Another interface Alarm On and OFF switch in ‘Symbol’ format will appear along with all other selections in the user panel. |  |
| 2. | No input is given for next UI. | Next UI will remain idle due to no further input. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-04 | | **Documented Date:** 9/29/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Check the output for User Panel component inputs on User Panel | | | |
| **Operation procedure:** The Main.java file is run and a new frame opens which takes input from the user and press OK button. | | | |
| **Pre-conditions:**  From User Panel component, run Main.java. Click Ok to get another user panel interface. | | **Post-conditions:**  Another UI of numbered floors should appear and would not show any change in color. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Ok button is pressed after filling all input parameters including same color input for buttons and pressed buttons on User Panel | New interface appears |  |
| 2 | On the new interface ,desired floor number button is pressed . | No change in color of floor buttons even after pressing. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-05 | | **Documented Date:** 9/29/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Check the output for User Panel component inputs on User Panel | | | |
| **Operation procedure:** The Main.java file is run and a new frame opens which takes input from the user and press OK button. | | | |
| **Pre-conditions:**  From User Panel component, run Main.java. Click Ok to get another user panel interface. | | **Post-conditions:**  Another UI of numbered floors should appear and would show change on color as per the input on previous user interface. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Ok button is pressed after filling all input parameters including different color input for buttons and pressed buttons on User Panel | New interface appears. |  |
| 2 | Desired floor number button is pressed. | floors’ button color will change to one selected in user panel after being pressed. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-06 | | **Documented Date:** 9/29/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Check the output for User Panel component inputs on User Panel | | | |
| **Operation procedure:** The Main.java file is run and a new frame opens which takes input from the user and press OK button. | | | |
| **Pre-conditions:**  From User Panel component, run Main.java. Click Ok to get another user panel interface. | | **Post-conditions:**  Another UI appears with Alarm Off switch in non editable state. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Ok button is pressed after filling all input parameters including the alarm type. | New interface appears with alarm OFF switch in disabled state. |  |
| 2 | On the new interface , press alarm off switch. | The switch is not editable and no action takes place. The switch remains in disable state. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-07 | | **Documented Date:** 9/29/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Check the output for User Panel component inputs on User Panel | | | |
| **Operation procedure:** The Main.java file is run and a new frame opens which takes input from the user and press OK button. | | | |
| **Pre-conditions:**  From User Panel component, run Main.java. Click Ok to get another user panel interface. | | **Post-conditions:**  Another UI appears with Alarm On switch in editable state. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Ok button is pressed after filling all input parameters including the alarm type. | New interface appears with Alarm On switch in editable state. |  |
| 2 | On the new interface , press alarm On switch. | Alarm On switch color changes to Red and Alarm off switch changes to editable (enabled) state. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-08 | | **Documented Date:** 9/29/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Check the output for User Panel component inputs on User Panel | | | |
| **Operation procedure:** The Main.java file is run and a new frame opens which takes input from the user and press OK button. | | | |
| **Pre-conditions:**  From User Panel component, run Main.java. Click Ok to get another user panel interface. | | **Post-conditions:**  Another UI appears with Alarm On switch in editable state and floor button disabled. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Ok button is pressed after filling all input parameters including the alarm type. | New interface appears with Alarm On switch in editable state. |  |
| 2 | On the new interface , press alarm On switch . | Alarm On switch color changes to Red and Alarm off switch changes to editable (enabled) state. |  |
| 3 | Press floor button | Floor button color remains same and dose not change to floor button pressed color indicating that the floor buttons are disabled due to alarm On pressed. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-9 | | **Documented Date:** 9/29/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Check the output for User Panel component inputs on User Panel | | | |
| **Operation procedure:** The Main.java file is run and a new frame opens which takes input from the user and press OK button. | | | |
| **Pre-conditions:**  From User Panel component, run Main.java. Click Ok to get another user panel interface. | | **Post-conditions:**  Another UI appears with Alarm On switch in editable state and floor button disabled. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Ok button is pressed after filling all input parameters including the alarm type. | New interface appears with Alarm On switch in editable state. |  |
| 2 | Press floor button | Floor button color changes to pressed floor button color. |  |
| 2 | Now , press alarm On switch . | Alarm On switch color changes to Red and Alarm off switch changes to editable (enabled) state and the floor pressed button color resets back to floor button color. |  |
| 3 | Press floor button | Floor button color remains same and dose not change to floor button pressed color indicating that the floor buttons are disabled due to Alarm On pressed. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-10 | | **Documented Date:** 9/29/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Check the output for User Panel component inputs on User Panel | | | |
| **Operation procedure:** The Main.java file is run and a new frame opens which takes input from the user and press OK button. | | | |
| **Pre-conditions:**  From User Panel component, run Main.java. Click Ok to get another user panel interface. | | **Post-conditions:**  Another UI appears with alarm switch off resetting the Alarm On switch. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Ok button is pressed after filling all input parameters including the alarm type. | New interface appears with Alarm On switch in editable state. |  |
| 2 | Now , press alarm On switch . | Alarm On switch color changes to Red and Alarm off switch changes to editable (enabled) state |  |
| 3 | Press Alarm Off switch | Alarm On switch color changes to Grey. Alarm switch On becomes editable and Alarm switch off goes back to disable state. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-11 | | **Documented Date:** 9/29/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Check the output for User Panel component inputs on User Panel | | | |
| **Operation procedure:** The Main.java file is run and a new frame opens which takes input from the user and press OK button. | | | |
| **Pre-conditions:**  From User Panel component, run Main.java. Click Ok to get another user panel interface. | | **Post-conditions:**  Another UI appears with alarm switch off resetting the Alarm On switch. And also enabling the floor buttons | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Ok button is pressed after filling all input parameters including the alarm type. | New interface appears with Alarm On switch in editable state. |  |
| 2 | Now , press alarm On switch . | Alarm On switch color changes to Red and Alarm off switch changes to editable (enabled) state |  |
| 3 | Press Alarm Off switch | Alarm On switch color changes to Grey. Alarm switch On becomes editable and Alarm switch off goes back to disable state. |  |
| 4 | Press any floor button | Floor button color changes to floor button pressed color selected in UI panel indicating floor buttons are enabled. |  |

#### User Panel Equivalence partition

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Input Space | | Valid Equivalence Partition | | Invalid Equivalence Partition | | |
| Select the number of floors | | Number of Floors between  [2, 20] or no floors selected | | Number of Floors  Less than 2 or  More than 20 | | |
| Select Floor Button color | | Color from Red, Blue, Green, Yellow | | More than one color selected, | | |
| no color selected | | Color not from selection | | |
| Select Floor Button pressed color | | Color from Red, Blue, Green, Yellow | | More than one color selected | | |
| No color selected | | Color not from selection | | |
| OK button | | Pressed, Not Pressed | | Ok button disabled | | |
| User Panel Floor button pressed | | NO, One or more floors are pressed | |  | | |
| Select User Panel type | | Even, Odd, or sequential is selected. | | More than one type is selected | | |
| no selection | |
| Select Alarm type | | Text or symbol | | More than one type selected | | |
| Alarm On Switch | | Switch is in editable state in the beginning | | Switch is not in editable state at the beginning | | |
| Alarm Off Switch | | Switch is in disabled (non editable) state at the beginning | | Switch is in editable state in the beginning | | |
| Alarm On switch pressed | | Alarm On switch pressed once | | Alarm On switch pressed more than once conseqetively. | | |
| Alarm OFF switch pressed | | Alarm OFF switch pressed when it is in editable state | | Alarm Off switch pressed when it is in disabled state. | | |
| **Test Case ID:** BB-EP-01 | | | **Documented Date:10/01/2011** | | |
| **Tester Name:** Shubhada Narkar | | | **Test Item:** User Panel | | |
| **Product Name:** Elevator System | | | **Version No.: 3.0** | | |
| **Test Type:** Black Box testing | | | **Test Suite #:** 1.0 | | |
| Test case description: To check the number of floors in Configuration UI | | | | | |
| **Operation procedure:**  Select the number of floors in configuration UI for the elevator and press the OK button | | | | | |
| **Pre-conditions:**  Open Configuration UI | | | **Post-conditions:**  Display User Panel UI | | |
| **Steps** | **Inputs data and/or events:** | | **Expected output data and/or events:** | | Pass/Fail |
| 1 | 2 or more floors are selected up to 20 | | User Panel UI shows respective floors as selected | |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | | | |
| **Comments:** | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-02 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the number of floors in Configuration UI | | | |
| **Operation procedure:**  Select the number of floors in configuration UI for the elevator and press the OK button | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Select floor number <2 or >20 | N/A |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-03 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the number of floors in Configuration UI | | | |
| **Operation procedure:** Select the number of floors in configuration UI for the elevator and press the OK button. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Floor no. not selected | The user Panel shows two floors |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-04 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the Button Color in Configuration UI. | | | |
| **Operation procedure:**  Select the Floor button color in Configuration UI for the panel and click OK button. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | A color is selected from the drop down box | User Panel UI shows respective color as selected. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-05 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the Button Color in Configuration UI. | | | |
| **Operation procedure:**  Select the Floor button color in Configuration UI for the panel and click OK button. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | No color is selected from the drop down box | User Panel UI shows Red color as default color. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-06 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the Button Color in Configuration UI. | | | |
| **Operation procedure:**  Select the Floor button color in Configuration UI for the panel and click OK button. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | More than one color are selected from the menu | N/A |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-07 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the Button Color in Configuration UI. | | | |
| **Operation procedure:**  Select the Floor button color in Configuration UI for the panel and click OK button. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Select Color other than in drop down menu | N/A |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-08 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the OK Button in Configuration UI. | | | |
| **Operation procedure:**  Select menus in Configuration UI for the panel and click OK button.. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | After selection of menus OK button is pressed | User Panel UI is displayed with made selections. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-09 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the OK Button in Configuration UI. | | | |
| **Operation procedure:**  Select menus in Configuration UI for the panel and do not click OK button.. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | After selection of menus OK button is not pressed | User Panel UI is not displayed. Configuration UI remains idle. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-10 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the Floor button pressed color in User Panel UI | | | |
| **Operation procedure:**  Select the floor button pressed color in Configuration UI for the panel and click OK button. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | One or more floor buttons are pressed on User Panel UI | Floor button colors are changed to selected pressed floor button color. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-11 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the User panel type selection in User Panel UI. | | | |
| **Operation procedure:**  Select the panel type in Configuration UI and click OK button. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | More than one type selected. | N/A |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-12 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the User panel type selection in User Panel UI. | | | |
| **Operation procedure:**  Select the panel type (even)in Configuration UI and click OK button. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Even type selected. | Even no. of floors displayed in the User panel UI. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-13 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the User panel type selection in User Panel UI. | | | |
| **Operation procedure:**  Select the panel type (odd) in Configuration UI and click OK button. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Odd type selected. | Odd no. of floors displayed in the User panel UI. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-14 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the User panel type selection in User Panel UI. | | | |
| **Operation procedure:**  Do not Select the panel type in Configuration UI and click OK button. | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Do not select the user panel type but select the remaining configuration | The sequential type is selected by default with the remaining parameters. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

New Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-15 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the alarm type selection in Configuration UI. | | | |
| **Operation procedure:**  **Alarm type selected on configuration UI and then press OK button** | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Select alarm Type ‘text’ and press OK | Alram On and Off switch appear in text format along with the remaining parameters as selected. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-16 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the alarm type selection in Configuration UI. | | | |
| **Operation procedure:**  **Alarm type selected in configuration UI and then press OK button** | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Select alarm Type ‘symbol’ and press ok | Alarm On and Off switch appear in ‘symbol’ format along with the remaining parameters as selected. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-17 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the alarm ON button status in User panel UI. | | | |
| **Operation procedure:**  **Select all parameters in configuration UI and then press OK button and check the Alarm On switch on user panel UI** | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Select all parameters in configuration UI and press OK | Alarm ON switch appears in editable state in User panel UI. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-18 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check the alarm OFF button status in User panel UI. | | | |
| **Operation procedure:**  **Select all parameters in configuration UI and then press OK button and check the Alarm OFF switch on user panel UI** | | | |
| **Pre-conditions:**  Open Configuration UI | | **Post-conditions:**  Display User Panel UI | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Select all parameters in configuration UI and press OK | Alarm OFF switch appears in disable state in User panel UI. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-19 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check status of user panel after alarm ON button is pressed once in user panel UI. | | | |
| **Operation procedure:**  **Press alarm ON button in user panel UI** | | | |
| **Pre-conditions:**  Open user panel UI. | | **Post-conditions:**  **Alarm On button color change and floor button disabled.** | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm ON switch on User panel UI once. | Alarm On switch button color changes to Red , user panel floor buttons disabled and alarm Off switch enabled. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-20 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check status of user panel after alarm ON button is pressed more than once in user panel UI. | | | |
| **Operation procedure:**  **Press alarm ON button in user panel UI more than once consequitively.** | | | |
| **Pre-conditions:**  Open user panel UI. | | **Post-conditions:**  **Alarm On button color change and floor button disabled, alarm off button enabled.** | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm ON switch on User panel UI more than once in sequence . | Alarm On switch button color changes to Red , user panel floor buttons disabled and alarm Off switch enabled. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-21 | | **Documented Date:10/01/2011** | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** User Panel | |
| **Product Name:** Elevator System | | **Version No.: 3.0** | |
| **Test Type:** Black Box testing | | **Test Suite #:** 1.0 | |
| Test case description: To check status of user panel after alarm OFF button is pressed to reset the alarm. | | | |
| **Operation procedure:**  **Press alarm OFF button in user panel UI to reset the alarm.** | | | |
| **Pre-conditions:**  Open user panel UI and alarm On button was pressed. | | **Post-conditions:**  **Alarm OFF button is disabled, alarm ON button is enabled with its color change and user panel floor buttons enabled.** | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm OFF button is pressed to reset the alarm. | Alarm On switch button color changes to enabled state , user panel floor buttons enabled and alarm Off switch disabled. |  |
| **Required test scripts:** Test Specification Document, User Interface Specification | | | |
| **Comments:** | | | |

#### Floor Panel Decision table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Conditions |  | T1 | T2 | T3 | T4 | T5 | T6 | T7 |
| C1:Up Button Pressed | T | F | F | F | F | F | T |
| C2:DownButton Pressed | F | T | F | F | F | F | F |
| C3: Up button pressed when elevator is moving. | - | - | T | F | T | F | - |
| C4: Down button pressed when elevator is moving. | - | - | F | T | T | F | F |
| C5: No button pressed. | F | F | F | F | F | T | F |
| Actions | A1: Button color activated. | X | X | X | X | X |  |  |
| A2: Add to queue. | X | X | X | X | X |  |  |
| A3: Idle |  |  |  |  |  | X |  |
| A4: The Floor panel indicator activates. |  |  | X | X | X |  | X |
| A5: The Floor panel indicator Editable. | X | X | X | X | X | X | X |
| Alarm Idicator ( for car) turns Red | X |  |  |  |  |  |  |
| Alarm Indicator ( for car) resets |  | X |  |  |  |  |  |

Green – Modified, Blue – New, Red - reused

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-01 | | **Documented Date:**10/03/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test “UP” button on the floor panel and Alarm On indicator for car 1 | | | |
| **Operation procedure:** Press the “Up” button of the floor panel and press alarm ON button on user panel car1 . | | | |
| **Pre-conditions:** .Alarm indicator off. | | **Post-conditions:** Alarm indicator turns on for car1 on the floor panel. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press the “UP” button on the floor panel and alarm ON button on user panel of car 1 . | Button color activated and the up request is added to the request queue. Alarm indicator turns red for car 1. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-02 | | **Documented Date:**10/03/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test “DOWN” button on the floor panel | | | |
| **Operation procedure:** Press the “DOWN” button of the floor panel and press alarm off button on user panel of car 1. | | | |
| **Pre-conditions:** alarm indicator On for car 1on the floor panel. | | **Post-conditions:** Alarm indicator turns off for car 1 on floor panel | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press the “DOWN” button on the floor panel and alarm off button on the user panel of car 1. | Button color activated and the down request is added to the request queue.Alarm indicator turns off for car 1 on the floor panel. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-03 | | **Documented Date:**10/03/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** “UP” button is pressed when elevator is moving. | | | |
| **Operation procedure:** Press the “Up” button of the floor panel when elevator is moving. | | | |
| **Pre-conditions:** Elevator is moving. | | **Post-conditions:** Floor Indicator activates. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press the “UP” button on the floor panel. | Button color activated and the UP, DOWN request is added to the request queue one after another Floor panel activator activates by showing the corresponding (odd or even or current), shortest path car’s Car-id, car-type and floor no. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-04 | | **Documented Date:**10/03/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** “DOWN” button is pressed when elevator is moving. | | | |
| **Operation procedure:** Press the “DOWN” button of the floor panel when elevator is moving. | | | |
| **Pre-conditions:** Elevator is moving. | | **Post-conditions:** Floor Indicator activates. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press the “DOWN” button on the floor panel. | Button color activated and the UP, DOWN request is added to the request queue one after another Floor panel activator activates by showing the corresponding (odd or even or current), shortest path car’s Car-id, car-type and floor no. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-05 | | **Documented Date:**10/03/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test pressing “UP” and“DOWN” button on the floor panel one after another when elevator is moving. | | | |
| **Operation procedure:** Press the “UP” and “DOWN” button of the floor panel when elevator is moving. | | | |
| **Pre-conditions:** Elevator is moving. | | **Post-conditions:** Floor Indicator activates. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press the “UP and DOWN” button on the floor panel one after another . | Button color activated and the UP, DOWN request is added to the request queue one after another and Floor indicator activates by showing the corresponding (odd or even or current), shortest path car’s Car-id, car-type and floor no. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-06 | | **Documented Date:**10/03/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test the floor panel when there is no activity has been performed. | | | |
| **Operation procedure:** Testing the floor panel when no button is pressed. | | | |
| **Pre-conditions:** None | | **Post-conditions:** None | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | No Activity on floor panel. | Floor Panel is Idle. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-DT-07 | | **Documented Date:**10/03/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** When running the floor panel the boxes of floor indicator are editable. | | | |
| **Operation procedure:**  - Run the floor panel.  - Press any button on the floor panel.  - The floor panel indicator gets updated.  - Type something in other floor’s indicator. | | | |
| **Pre-conditions:** Floor Panel is in active state. | | **Post-conditions:**  Floor indicators are updated. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses the UP or down button in the floor panel one after another. | The floor indicator gets updated and floor indicator should not be editable. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

#### Floor Panel Equivalence partition

|  |  |  |
| --- | --- | --- |
| Input Space | Valid Equivalence Partitions | Invalid Equivalence Partitions |
| Floor- UP button | Floor-UP active on all floors except the topmost floor. | Floor-UP button active on the topmost floor. |
| Floor-DOWN button | Floor-DOWN button active on all floors except the lowest floor. | Floor-DOWN button active on the lowest floor. |
| UP/DOWN button color change | Floor UP/DOWN button color change when active | Floor UP/DOWN button color change when inactive |
| Floor indicator updates | Floor indicator get updates when car is moving | Floor indicator does not get updated when car is moving. |
| Alarm Indicator | Alarm indicator turns On ( Red) when alarm on button pressed on user panel. | Alarm indicator does not turn ON ( Red) when alarm button is pressed on the user panel. |
|  | Alarm indicator turns off when alarm off button is pressed on user panel. | Alarm indicator does not turn off when alarm off button is pressed on user panel. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-01 | | **Documented Date:**10/05/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel with Alarm Indicator | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test “UP” button of the top most floor. | | | |
| **Operation procedure:** Press the “UP” button of the top most floor | | | |
| **Pre-conditions:** Elevator car is on the top most floor. | | **Post-conditions:** Floor UP button is inactive. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Elevator car is on the top most floor |  |  |
| 2 | floorRequest is made to move upwards. | Floor UP button is not available on topmost floor. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-02 | | **Documented Date:**10/05/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel with Alarm Indicator | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test “DOWN” button of the lowest floor. | | | |
| **Operation procedure:** Press the “DOWN” button of the lowest most floor | | | |
| **Pre-conditions:** Elevator car is on the lowest floor. | | **Post-conditions:** Floor Down button is inactive. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Elevator car is on the lowest floor |  |  |
| 2 | Floor Request is made to move down. | Floor Down button is not available on the lowest floor. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-03 | | **Documented Date:**10/05/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel with Alarm Indicator | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test “UP” button of the lowest floor. | | | |
| **Operation procedure:** Press the “UP” button of the lowest floor | | | |
| **Pre-conditions:** Elevator car is on the lowest floor. | | **Post-conditions:** Floor UP button is active on lowest floor | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Elevator car is on the lowest floor |  |  |
| 2 | Floor Request is made to move UP. | Floor UP button is available on the lowest floor. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-04 | | **Documented Date:**10/05/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel with Alarm Indicator | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test “UP” and “DOWN” buttons of any middle floor on the floor panel on the floor panel | | | |
| **Operation procedure** Press the “UP” or “DOWN” buttons of any middle floor. | | | |
| Pre-conditions: Elevator car is on the any floor which is between the top and lowest floor. | | **Post-conditions:** On this Floor UP/DOWN button is active | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Elevator car is on the middle floor | On the middle floor UP/DOWN button is active |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-05 | | **Documented Date:**10/05/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel with Alarm Indicator | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test Floor UP/DOWN button color change when active. | | | |
| **Operation procedure** Press the “UP” or “DOWN” buttons of any middle floor. | | | |
| Pre-conditions: Elevator car is on the 2nd floor UP/DOWN Button Pressed | | Post-conditions: Color of the UP/DOWN button changes. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Elevator car is on the middle floor |  |  |
| 2 | UP/DOWN button pressed | Floor UP/DOWN button color changes. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-06 | | **Documented Date:**10/05/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel with Alarm Indicator | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test Floor UP/DOWN button color change when active and pressed again. | | | |
| **Operation procedure** Press the “UP” or “DOWN” buttons of any middle floor. | | | |
| Pre-conditions: Elevator car is on the 2nd floor UP/DOWN Button Pressed again when active. | | Post-conditions: Color of the UP/DOWN button changes though the request been made earlier. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Elevator car is on the middle floor |  |  |
| 2 | UP/DOWN button pressed twice | Floor UP/DOWN button color changes. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-07 | | **Documented Date:**10/05/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel with Alarm Indicator | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test Floor Indicator when car is moving. | | | |
| **Operation procedure** Press the “UP” or “DOWN” buttons on floor panel when car is moving. | | | |
| Pre-conditions: car is moving. | | **Post-conditions:** Floor indicator get updated | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press “UP” or “Down”  Button when car is moving. | Floor indicator should show the value of carid, car type and Floor no |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

New test cases:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-08 | | **Documented Date:**10/05/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel with Alarm Indicator | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test the Alarm indicator on the floor panel | | | |
| **Operation procedure** Press the alarm on button for car 1. | | | |
| Pre-conditions: Alarm on pressed for car 1 | | **Post-conditions:** Alarm indicator turns On ( Red) on the floor panel for car 1 | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm On for car 1 | Alarm indicator turns On ( red) for car 1 on the floor panel. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** BB-EP-09 | | **Documented Date:**10/05/2011 | |
| **Tester Name:** Shubhada Narkar | | **Test Item:** Floor Panel with Alarm Indicator | |
| **Product Name:** Elevator System | | **Version No.:** 3.0 | |
| **Test Type:** Black box testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Test the Alarm indicator on the floor panel | | | |
| **Operation procedure** Press the alarm Off button for car 1. | | | |
| Pre-conditions: Alarm Off pressed for car 1 | | **Post-conditions:** Alarm indicator turns Off on the floor panel for car 1 | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm Off for car 1 | Alarm indicator turns Off for car 1 on the floor panel. |  |
| **Required test scripts:** None | | | |
| **Comments:** | | | |

## 3.3 System Testing

System testing is a part of testing activity, which focuses on a complete, integrated system in order to evaluate the system’s compliance with specification document. This type of testing falls within the scope of black box testing and hence, we will be using two methods of black box testing to re-test the whole elevator system – Equivalence Partition Method and Decision Table Method. The whole behavior of the system will be taken into consideration for deriving test cases. Moreover, the system will be tested up to and beyond the bounds mentioned in specification document.

### 3.3.1 Decision Table Method

Decision tables are a precise yet compact way to model complicated logic.Each decision corresponds to a variable, relation or predicate whose possible values are listed among the condition alternatives. Each action is a procedure or operation to perform, and the entries specify whether (or in what order) the action is to be performed for the set of condition alternatives the entry corresponds to. Many decision tables include in their condition alternatives the “don’t care” symbol, a hyphen.

Following is the decision table for test cases for testing the system using Decision Table method:

“T” represents presence of a condition.

“F” represents absence of the condition.

“-“represents “Don’t care” i.e. condition has no influence on the actions to be performed

“X” represents actions as a result of specified conditions.

Newly added partitions are represented in Blue.

Reused partitions are represented in Red

Modified partitions are represented in Green

Decision Table :



Test Cases based on above Decision Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-01 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkShortest Path Algorithm when UP button of Floor Panel is pressed and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press UP button of Floor Panel on Level 5. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 1  Position of Car : 1st Floor | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing UP button on Level 5. | 1. Car should reach Level 5 and stop. 2. Doors should open at Level 5. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-02 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkRandom Path Algorithm when UP button of Floor Panel is pressed and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press UP button of Floor Panel on Level 3. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 1  Position of Car : 1st Floor | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing UP button on Level 3. | 1. Car should reach Level 3 and stop. 2. Doors should open at Level 3. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-03 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkScan First Algorithm when UP button of Floor Panel is pressed and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press UP button of Floor Panel on Level 5. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 1  Position of Car : 1st Floor | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing UP button on Level 5. | 1. Car should reach Level 5 and stop. 2. Doors should open at Level 5. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-04 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkShortest Path Algorithm when UP button of Floor Panel is pressed and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**  1. Press UP button of Floor Panel on Level 5. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 3  Position of Car 1 : 1st Floor (Moving Up)  Position of Car 2 : 4th Floor (Moving Up)  Position of Car 3 : 10th Floor (Moving Down) | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing UP button on Level 5. | 1. Car 2 should reach Level 5 and stop. 2. Doors should open at Level 5. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-05 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkRandom Path Algorithm when UP button of Floor Panel is pressed and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**  1. Press UP button of Floor Panel on Level 3. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 4  Position of Car 1 : 1st Floor (Idle)  Position of Car 2: 6th Floor (Moving Down)  Position of Car 3: 5th Floor (Idle)  Position of Car 4: 10th Floor (Moving Up)  Position of Car 5: 1st Floor (Moving Up) | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing UP button on Level 3. | 1. Any of the cars should reach Level 3 and stop. 2. Doors should open at Level 3. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-06 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkScan First Algorithm when UP button of Floor Panel is pressed and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press UP button of Floor Panel on Level 5. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 3  Position of Car 1 :1st Floor (Idle)  Position of Car 2: 4th Floor (Moving Up)  Position of Car 3: 3rd Floor (Moving Down) | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing UP button on Level 5. | 1. Car 2 should reach Level 5 and stop. 2. Doors should open at Level 5. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-07 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm component functionality with Shortest Path Algorithm when UP button of Floor Panel is pressed with one car in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel in the car for Level 5. 2. Press Alarm button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 1  Position of Car : 1st Floor | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1  2 | Press UP button on Level 5.  Press Alarm button in User Panel before lift reaches level 5. | 1. Car should stop as soon as the Alarm Switch is pressed on the next level if it was moving. 2. Alarm indicators on all Floor Panels should be activated. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-08 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm Component functionality with Random Path Algorithm when UP button of Floor Panel is pressed with one car in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel in the car for Level 5. 2. Press Alarm button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 1  Position of Car : 1st Floor | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1  2 | Pressing UP button on Level 3.  Press Alarm button in User Panel before lift reaches level 3. | 1. Car should stop as soon as the Alarm Switch is pressed on the next level if it was moving. 2. Alarm indicators on all Floor Panels should be activated. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-09 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm Component with Scan First Algorithm when UP button of Floor Panel is pressed with one car in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel on Level 5. 2. Press Alarm button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 1  Position of Car : 1st Floor | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1  2 | Pressing UP button on Level 5.  Press Alarm button in User Panel before lift reaches level 5. | 1. Car should stop as soon as the Alarm Switch is pressed on the next level if it was moving. 2. Alarm indicators on all Floor Panels should be activated. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-10 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkShortest Path Algorithm when UP button of Floor Panel is pressed and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel on Level 5. 2. Press Alarm button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 3  Position of Car 1 : 1st Floor (Moving Up)  Position of Car 2 : 4th Floor (Moving Up)  Position of Car 3 : 10th Floor (Moving Down) | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1  2 | Pressing UP button on Level 5.  Press Alarm button in User Panel of Car 2 before lift reaches level 5. | 1. Car 2 should stop as soon as the Alarm Switch is pressed on the next level if it was moving. 2. Alarm indicators on all Floor Panels should be activated. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-11 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm Component with Random Path Algorithm when UP button of Floor Panel is pressed and multiple cars in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel on Level 3. 2. Press Alarm button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 4  Position of Car 1 : 1st Floor (Idle)  Position of Car 2: 6th Floor (Moving Down)  Position of Car 3: 5th Floor (Idle)  Position of Car 4: 10th Floor (Moving Up)  Position of Car 5: 1st Floor (Moving Up) | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1  2 | Pressing UP button on Level 3.  Press Alarm button in User Panel of the Car before lift reaches level 3. | 1. Car should stop as soon as the Alarm Switch is pressed on the next level if it was moving. 2. Alarm indicators on all Floor Panels should be activated. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-12 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkScan First Algorithm when UP button of Floor Panel is pressed and one car is deployed in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel on Level 5. 2. Press Alarm button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 3  Position of Car 1 :1st Floor (Idle)  Position of Car 2: 4th Floor (Moving Up)  Position of Car 3: 3rd Floor (Moving Down) | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1  2 | Pressing UP button on Level 5.  Press Alarm button in User Panel of the Car before lift reaches level 5. | 1. Car should stop as soon as the Alarm Switch is pressed on the next level if it was moving. 2. Alarm indicators on all Floor Panels should be activated. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-13 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm component functionality deactivation with Shortest Path Algorithm when UP button of Floor Panel is pressed with one car in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel in the car for Level 5. 2. Press Alarm Off button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 1  Position of Car : Any (Stopped)  Alarm Indicator : On | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm Off button on User Panel. | 1. Alarm Indicator on User Panel of the car should be disabled. 2. Alarm indicators on all Floor Panels should be deactivated. 3. Car should start moving to its destination. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-14 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm Component functionality deactivation with Random Path Algorithm with one car in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel in the car for Level 5. 2. Press Alarm Off button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 1  Position of Car : 1st Floor  Alarm Indicator : On | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm Off button in User Panel before lift reaches level 3. | 1. Alarm Indicator on User Panel of the car should be disabled. 2. Alarm indicators on all Floor Panels should be deactivated. 3. Car should start moving to its destination. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-15 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm Component deactivation with Scan First Algorithm with one car in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel on Level 5. 2. Press Alarm Off button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 1  Position of Car : 1st Floor  Alarm Indicator : On | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm button in User Panel before lift reaches level 5. | 1. Alarm Indicator on User Panel of the car should be disabled. 2. Alarm indicators on all Floor Panels should be deactivated. 3. Car should start moving to its destination. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-16 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm Component deactivation with Shortest Path Algorithm with multiple cars in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel on Level 5. 2. Press AlarmOff button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 3  Position of Car 1 : 1st Floor (Moving Up)  Position of Car 2 : 4th Floor (Moving Up)  Position of Car 3 : 10th Floor (Moving Down)  Alarm Indicator : On | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm Off button in User Panel of Car 2 before lift reaches level 5. | 1. Alarm Indicator on User Panel of the car should be disabled. 2. Alarm indicators on all Floor Panels should be deactivated. 3. Car should start moving to its destination. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-17 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm Component deactivation with Random Path Algorithm with multiple cars in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel on Level 3. 2. Press Alarm Off button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 4  Position of Car 1 : 1st Floor (Idle)  Position of Car 2: 6th Floor (Moving Down)  Position of Car 3: 5th Floor (Idle)  Position of Car 4: 10th Floor (Moving Up)  Position of Car 5: 1st Floor (Moving Up)  Alarm Indicator : On | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1  2 | Pressing UP button on Level 3.  Press Alarm button in User Panel of the Car before lift reaches level 3. | 1. Alarm Indicator on User Panel of the car should be disabled. 2. Alarm indicators on all Floor Panels should be deactivated. 3. Car should start moving to its destination. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-18 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm Component deactivation with Scan First Algorithm with multiple cars in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel on Level 5. 2. Press Alarm Off button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 3  Position of Car 1 :1st Floor (Idle)  Position of Car 2: 4th Floor (Moving Up)  Position of Car 3: 3rd Floor (Moving Down)  Alarm Indicator : On | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1  2 | Pressing UP button on Level 5.  Press Alarm button in User Panel of the Car before lift reaches level 5. | 1. Alarm Indicator on User Panel of the car should be disabled. 2. Alarm indicators on all Floor Panels should be deactivated. 3. Car should start moving to its destination. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-19 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm Off button functionality is disabled with Shortest Path Algorithm when Alarm On button is not pressed with one car in elevator system. | | | |
| **Operation procedure:**   1. Press Alarm Off button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 1  Position of Car : 1st Floor (Moving Up)  Alarm Indicator : Off | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm off button in User Panel. | 1. Nothing should happen on User Panel and Floor Panel. 2. Car should continue functioning as normal. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-20 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** **:** To checkAlarm Off button functionality is disabled with Random Algorithm when Alarm On button is not pressed with one car in elevator system. | | | |
| **Operation procedure:**   1. Press Alarm Off button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 1  Position of Car : 5th Floor (Moving Down)  Alarm Indicator: Off | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm Off button in User Panel before lift reaches its destination. | 1. Nothing should happen on User Panel and Floor Panel. 2. Car should continue functioning as normal. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-21 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm Off button functionality is disabled with Scan First Algorithm when Alarm On button is not pressed with one car in elevator system. | | | |
| **Operation procedure:**   1. Press UP button of Floor Panel on Level 5. 2. Press Alarm button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 1  Position of Car : 1st Floor (Idle)  Alarm Indicator : Off | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm Off button in User Panel. | 1. Nothing should happen on User Panel and Floor Panel. 2. Car should continue functioning as normal. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-22 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** **:** To checkAlarm Off button functionality is disabled with Shortest Path Algorithm when Alarm On button is not pressed with multiple cars in elevator system. | | | |
| **Operation procedure:**   1. Press Alarm Off button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 3  Position of Car 1 : 1st Floor (Moving Up)  Position of Car 2 : 4th Floor (Moving Up)  Position of Car 3 : 10th Floor (Moving Down)  Alarm Indicator : Off | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm button in User Panel of Car 2 before lift reaches level 5. | 1. Nothing should happen on User Panel and Floor Panel. 2. All Cars should continue functioning as normal. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-23 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** **:** To checkAlarm Off button functionality is disabled with Random Path Algorithm when Alarm On button is not pressed with multiple cars in elevator system. | | | |
| **Operation procedure:**   * 1. Press Alarm Off button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 4  Position of Car 1 : 1st Floor (Idle)  Position of Car 2: 6th Floor (Moving Down)  Position of Car 3: 5th Floor (Idle)  Position of Car 4: 10th Floor (Moving Up)  Position of Car 5: 1st Floor (Moving Up)  Alarm Indicator : Off | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm button in User Panel of the Car before lift reaches its destination | 1. Nothing should happen on User Panel and Floor Panel. 2. All Cars should continue functioning as normal. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-24 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel, Alarm Component , User Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkAlarm Off button functionality is disabled with Scan First Algorithm when Alarm On button is not pressed with multiple cars in elevator system. | | | |
| **Operation procedure:**   * 1. Press Alarm button on User Panel of the car. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 3  Position of Car 1 :1st Floor (Idle)  Position of Car 2: 4th Floor (Moving Up)  Position of Car 3: 3rd Floor (Moving Down)  Alarm Indicator : Off | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Alarm button in User Panel of the Car before lift reaches its destination. | 1. Nothing should happen on User Panel and Floor Panel. 2. All Cars should continue functioning as normal. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-25 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To check when Floor - Down Button is pressed and Elevator is at same floor Shortest Path Algorithm and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press Down button of Floor Panel on Level 5. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 1  Position of Car : 5th Floor | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing Floor- Panel Down button on Level 5. | 1. A car is already at the same level so No movement of car should happen. 2. Doors of the car should open at Level 5. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-26 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To check when Floor - Down Button is pressed and Elevator is at same floor Random Path Algorithm and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press UP button of Floor Panel on Level 3. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 1  Position of Car : 10th Floor | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing Floor Panel Down button on Level 10. | 1. A car is already at the same level so No movement of car should happen. 2. Doors of the car should open at Level 10. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-27 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To check when Floor - Up Button is pressed and Elevator is at same floor Scan First Algorithm and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press UP button of Floor Panel on Level 5. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 1  Position of Car : 5th Floor | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing Floor Panel UP button on Level 5. | 1. A car is already at the same level so No movement of car should happen. 2. Doors of the car should open at Level 5. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-28 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To check when Floor - Up Button is pressed and Elevator is at same floor Shortest Path Algorithm and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**  1. Press UP button of Floor Panel on Level 5. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 3  Position of Car 1 : 1st Floor (Moving Up)  Position of Car 2 : 5th Floor (Moving Up)  Position of Car 3 : 10th Floor (Moving Down) | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing Floor panel UP button on Level 5. | 1. A car is already at the same level so No movement of car should happen. 2. Doors of the car should open at Level 5. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-29 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To check when Floor – Down Button is pressed and Elevator is at same floor Random Path Algorithm and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**  1. Press UP button of Floor Panel on Level 6. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 4  Position of Car 1 : 1st Floor (Idle)  Position of Car 2: 6th Floor (Moving Down)  Position of Car 3: 5th Floor (Idle)  Position of Car 4: 10th Floor (Moving Up)  Position of Car 5: 1st Floor (Moving Up) | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing Floor panel Down button on Level 6. | 1. A car is already at the same level so No movement of car should happen. 2. Doors of the car should open at Level 6. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-30 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To check when Floor – Down Button is pressed and Elevator is at same floor Scan First Algorithm and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**  1. Press Floor Panel Down button of Floor Panel on Level 5. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 3  Position of Car 1 :1st Floor (Idle)  Position of Car 2: 4th Floor (Moving Up)  Position of Car 3: 3rd Floor (Moving Down) | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing Floor Panel Down button on Level 3. | 1. A car is already at the same level so No movement of car should happen. 2. Doors of the car 3 should open at Level 3. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-31 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , User Panel , Elevator Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkdoor opens when Door Open Button is pressed when Shortest Path Algorithm and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press Door Open button of User Panel on Level 5. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 1  Position of Car : 5th Floor  Door of elevator car : Closed | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing Door Open button on Level 5. | Doors of the car should open at Level 5. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-32 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , User Panel , Elevator Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkdoor opens when Door Open Button is pressed when Random Path Algorithm and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press Door Open button of Floor Panel on Level 3. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 1  Position of Car : 3rd Floor  Door of elevator car : Closed | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing Door Open button on Level 3. | Doors of the car should open when Door Open button is pressed at Level 3. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-33 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , User Panel , Elevator Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkdoor opens when Door Open Button is pressed when Scan First Algorithm and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press Door Open button of Floor Panel on Level 10. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 1  Position of Car : 10th Floor  Door of elevator car : Closed | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing Door Open button on Level 10. | Doors should open at Level 10. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-34 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , User Panel , Elevator Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkdoor closes when Door Close Button is pressed when Shortest Path Algorithm and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press Door Close button of User Panel on Level 5. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 1  Position of Car : 5th Floor  Door of elevator car : Open | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing Door Close button on Level 5. | Doors of the car should close when Door Close button is pressed at Level 5. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-35 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , User Panel , Elevator Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description :** To checkdoor closes when Door Close Button is pressed when Random Path Algorithm and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press Door Close button of Floor Panel on Level 3. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path  Number of Cars : 1  Position of Car : 3rd Floor  Door of elevator car : Open | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Press Door Close button on Level 3. | Doors of the car should close when Door Close button is pressed at Level 3. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-36 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , User Panel , Elevator Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To checkdoor closes when Door Close Button is pressed when Random Path Algorithm and one car is deployed in elevator system. | | | |
| **Operation procedure:**  1. Press Door Close button of Floor Panel on Level 10. | | | |
| **Pre-conditions:**  Algorithm Type: SCAN First  Number of Cars : 1  Position of Car : 10th Floor  Door of elevator car : Open | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Pressing Door Open button on Level 10. | Doors of the car should close when Door Close button is pressed at Level 10. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-37 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Door Panel, Alarm Indicator, Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To Validate text option for floor panel, door panel and alarm component when Shortest Path Algorithm and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**  1. Select all option to be text. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 4 | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User selects all configuration option as text | The elevator system displayed has text UI on Door Panel, Floor Panel and Alarm |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-38 | | **Documented Date: 10/09/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Door Panel, Alarm Indicator, Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To Validate text option for floor panel, door panel and alarm component when Random Path Algorithm and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**  1. Select all option to be text. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path First  Number of Cars : 7 | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User selects all configuration option as text | The elevator system displayed has text UI on Door Panel, Floor Panel and Alarm |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-39 | | **Documented Date: 10/10/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Door Panel, Alarm Indicator, Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To Validate text option for floor panel, door panel and alarm component when Scan First Algorithm and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**  1. Select all option to be text. | | | |
| **Pre-conditions:**  Algorithm Type: Scan First  Number of Cars : 4 | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User selects all configuration option as text | The elevator system displayed has text UI on Door Panel, Floor Panel and Alarm |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-40 | | **Documented Date: 10/10/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Door Panel, Alarm Indicator, Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To Validate symbol option for floor panel, door panel and alarm component when Shortest Path Algorithm and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**  1. Select all option to be Symbol. | | | |
| **Pre-conditions:**  Algorithm Type: Shortest Path First  Number of Cars : 4 | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User selects all configuration option as Symbol. | The elevator system displayed has SYMBOL UI on Door Panel, Floor Panel and Alarm |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-41 | | **Documented Date: 10/10/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Door Panel, Alarm Indicator, Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To Validate symbol option for floor panel, door panel and alarm component when Random Path Algorithm and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**  1. Select all option to be symbol. | | | |
| **Pre-conditions:**  Algorithm Type: Random Path First  Number of Cars : 7 | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User selects all configuration option as symbol | The elevator system displayed has SYMBOL UI on Door Panel, Floor Panel and Alarm |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-DT-42 | | **Documented Date: 10/10/2011** | |
| **Tester Name:** Sonalika Gupta | | **Test Item:** Algorithm , Door Panel, Alarm Indicator, Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Decision Table Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** To Validate symbol option for floor panel, door panel and alarm component when Scan First Algorithm and multiple cars are deployed in elevator system. | | | |
| **Operation procedure:**  1. Select all option to be symbol. | | | |
| **Pre-conditions:**  Algorithm Type: Scan First  Number of Cars : 4 | | **Post-conditions:**  None. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User selects all configuration option as symbol. | The elevator system displayed has Symbol UI on Door Panel, Floor Panel and Alarm |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

### 3.3.2 Equivalence Partition Method

In this method, the input data is divided into a number of disjoint partitions such that executing one value from each partition will lead to testing of system via entire input domain. The main objective is to cover each partition at least once. This method also reduces the redundancy in test cases by dividing the whole input domain into separate partitions. Test case design for this method depends on evaluation of equivalence classes for input domain. An equivalence class represents a set of valid or invalid states for input condition.

Following are the equivalence classes for test cases for testing the system using equivalence partition method:

Newly added partitions are represented in Blue.

Reused partitions are represented in Red

Modified partitions are represented in Green

Equivalence Partition Table

|  |  |  |
| --- | --- | --- |
| Input Space | Valid Equivalence Partition | Invalid Equivalence Partition |
| Algorithm type and number of cars is one | 1) Any algorithm is selected and whenever any floor button is pressed the elevator car should fulfill the user request. | 2) When the floor button is pressed where the elevator is already present or waiting, the elevator should not move. |
| Algorithm type and number of cars > 1 | 3) Whenever any floor buttons are pressed, the elevator car should fulfill the user request as per the algorithm selected by the user | 4) When some car is already present on selected floor, other cars should not move to fulfill that request. |
| No.of Cars is one and carType | 5) Only one car (default carType is sequential) is selected by user; the destination floor number selected in the floor panel is even.  6) Only one car(default carType is sequential) is selected by user; the destination floor number selected in the floor panel is odd. |  |
| Multiple cars are selected with different carType combination | 7) User selects more than one car with carTypes as odd and even; the destination floor number selected in the floor panel is even.  8) User selects more than one car with carTypes as odd and even; the destination floor number selected in the floor panel is odd.  9) User selects more than one car with carTypes as normal, odd and even; the destination floor number selected in the floor panel is even.  10) User selects more than one car with carTypes as normal, odd and even; the destination floor number selected in the floor panel is odd. | 11) User selects more than one car and the carType for all cars are selected as even and number selected in floor panel is odd.  12) User selects more than one car and the carType for all cars are selected as odd and number selected in floor panel is even. |
| Alarm “ON” Button on user panel and number of cars | 13) User presses alarm “ON” button and number of cars = 1.  14) User presses floor button while alarm is ON and number of cars = 1.  15) User presses alarm “ON” button in multiple cars and number of cars > 1.  16) User presses floor button while alarm is “ON” and number of cars > 1. | 17) User presses any button on user panel while alarm is “ON”.  18) User presses alarm “OFF” button when alarm is not “ON”.  19) Car status does not reflect “Alarm\_On” status and car does not move to first floor when alarm “ON” button is pressed. |
| Alarm “OFF” Button on user panel and number of cars | 20) User presses alarm “OFF” when alarm is on.  21) User presses any button on floor panel and number of car = 1.  22) User presses any button on floor panel and number of cars > 1.  23) User presses any button on user panel.  24) User presses alarm “ON” button. | 25) Floor request are not delegated to the car. |
| Alarm Indicator on Floor Panel | 26) User presses alarm “ON” button of a car.  27) User presses alarm “OFF” button of the same car. |  |
| User panel | 28) User selects more than one car with carTypes as normal, even and odd | 29) User selects car as odd or even and number of cars = 1 |
| Open Elevator Door | 30) Open Elevator Door Button pressed when the Elevator is on the Floor | 31) Open Elevator Door Button pressed when the elevator is running, the door cannot be opened.  32) Open door button is pressed when the door is closing. |
| Close Elevator Door | 33) Close Elevator Door Button pressed when the Elevator is on the Floor | 34) Close Elevator Door Button pressed when the elevator is running  35) Close door button is pressed when the door is opening. |
| Floor - UP Button | 36) Floor - Up Button pressed when Elevator is running or stopped at some other floor | 37) Up Elevator Button pressed when Elevator is stopped at the same floor |
| Floor - Down Button | 38) Floor - Down Button pressed when Elevator is running or stopped at some other floor | 39) Floor – Down Button pressed when Elevator is stopped at the same floor |
| Car on 1st floor | 40) User presses a floor button other than 1st floor while car is on 1st floor. | 41) User presses Down button on 1st floor |
| Car on top floor | 42) User presses a floor button other than top floor when car is on top floor. | 43) User presses Up button on top floor. |
| Text or symbol option for floor, door and alarm | 44) User selects text option for all.  45) User selects symbol option for all. |  |

Test Cases based on above Equivalence Partition Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID: ST-EP-01** | | **Documented Date: 10/12/2011** | |
| **Tester Name: Anshu Basia** | | **Test Item: Elevator Component – Algorithm, Floor Panel** | |
| **Product Name: Elevator System** | | **Version No.: 3** | |
| **Test Type: Equivalence Partition Black Box System Testing** | | **Test Suite #: 1.0** | |
| **Test case description:** Press up or down button of different floors on floor panel and all floor requests will be handled by the same car. | | | |
| **Operation procedure:**  1. Select desired algorithm from configuration panel and number of cars = 1.  2. Select more than 5 floors  3. Press up and down request of different floors. | | | |
| **Pre-conditions:**  1. Desired algorithm should be selected with exactly 1 car and more than 5 floors.  2. Alarm button of the car should not be in “ON” state. | | **Post-conditions:**  Floor indicator should display car id and car type | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Up and Down buttons of different floors are pressed in any order.  Eg. Firstly up button of floor 6 is pressed and then down button of floor 7 is pressed and then up button of floor 3 is pressed. | As only one car is present hence, all the floor requests should be fulfilled by the same car. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-02 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Algorithm, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press up or down button of the floor where elevator is already present, the selected algorithm should return the same car. | | | |
| **Operation procedure:**  1. Select desired algorithm from configuration panel.  2. Press up and down request of floor where elevator car is present. | | | |
| **Pre-conditions:**  1. Desired algorithm should be selected with exactly 1 car and more than 5 floors.  2. Alarm button of the car should not be in “ON” state. | | **Post-conditions:**  Floor indicator should display car id and car type | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Up or Down buttons of floor having the elevator car is pressed. | Elevator car should not move as it is already present on the requested floor. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-03 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Algorithm, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press up or down button of different floors on floor panel and monitor the allocation of cars to different floor requests according to the algorithm selected. | | | |
| **Operation procedure:**  1. Select desired algorithm from configuration panel.  2. Press up and down request of different floors. | | | |
| **Pre-conditions:**  1. Desired algorithm should be selected with minimum of 3 cars and more than 5 floors.  2. Alarm button of any of the car should not be in “ON” state. | | **Post-conditions:**  Floor indicator should display car id and car type | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Up and Down buttons of different floors are pressed in any order.  Eg. Firstly up button of floor 6 is pressed and then down button of floor 7 is pressed and then up button of floor 3 is pressed. | Cars will be allocated to floor requests as per the algorithm selected by user in configuration panel |  |
| **Required test scripts: -** None | | | |
| **Comments:** Different combinations of selections are required to test the algorithms | | | |
| **Test Case ID:** ST-EP-04 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Algorithm, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press up or down button of the floor on floor panel where some car is already present and monitor the allocation of car to the floor request. | | | |
| **Operation procedure:**  1. Select desired algorithm from configuration panel.  2. Press up and down request of the floor where some car is already present. | | | |
| **Pre-conditions:**  1. Desired algorithm should be selected with minimum of 3 cars and more than 5 floors.  2. Alarm button of any of the car should not be in “ON” state. | | **Post-conditions:**  Floor indicator should display car id and car type | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Up or Down buttons of floor where car is already present is pressed. | The selected algorithm should return the car present on the floor as the best car and other cars should not move to fulfill the floor request. |  |
| **Required test scripts: -** None | | | |
| **Comments:** Different combinations of selections are required to test the algorithms | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-05 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Odd-Even Algorithm, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate odd & even algorithm, floor panel with floor indicator component by selecting default carType & destination floor number as any even number. | | | |
| **Operation procedure:**  1. Select number of car = 1, number of floors from configuration panel  2. Press up or down request of even floor say 8th floor. | | | |
| **Pre-conditions:**  1. Car type should be sequential and no of cars = 1.  2. Alarm button of the car should not be in “ON” state.  3. Up or down button of even no. of floor is selected | | **Post-conditions:**  Floor indicator should display car id and car type and car should move to that floor | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | The down direction against floor number in the floor panel with floor indicator is pressed  Eg. Down button in Floor no. 8 is pressed | Floor indicator: CarId and CarType displays “1” and “seq” respectively and car moves to that floor |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-06 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Odd-Even Algorithm, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate odd & even algorithm, floor panel with floor indicator component by selecting default carType & destination floor number as any odd number. | | | |
| **Operation procedure:**  1. Select number of car = 1, number of floors from configuration panel  2. Press up or down request of even floor say 7th floor. | | | |
| **Pre-conditions:**  1. Car type should be sequential and no of cars = 1.  2. Alarm button of the car should not be in “ON” state.  3. Up or down button of odd no. of floor is selected | | **Post-conditions:**  Floor indicator should display car id and car type and car should move to that floor | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | The down direction against floor number in the floor panel with floor indicator is pressed  Eg. Down button in Floor no. 7 is pressed | Floor indicator: CarId and CarType displays “1” and “seq” respectively and car moves to that floor |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-07 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Odd-Even Algorithm, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate odd & even algorithm, floor panel with floor indicator component by selecting carType as either odd or even & destination floor number as any even number. | | | |
| **Operation procedure:**  1. Select number of cars with car type as odd or even, number of floors from configuration panel  2. Press up or down request of even floor say 8th floor. | | | |
| **Pre-conditions:**  1. Cars type should be either odd or even and no of cars > 1.  2. Alarm button of the cars should not be in “ON” state.  3. Up or down button of even no. of floor is selected | | **Post-conditions:**  Floor indicator should display car id and car type and car should go to that floor | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | The down direction against floor number in the floor panel with floor indicator is pressed  Eg. Down button in Floor no. 8 is pressed | Floor indicator: CarId and CarType displays respective car id and “even” respectively and car moves to that floor. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-08 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Odd-Even Algorithm, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate odd & even algorithm, floor panel with floor indicator component by selecting carType as either odd or even & destination floor number as any odd number. | | | |
| **Operation procedure:**  1. Select number of cars with car type as odd or even, number of floors from configuration panel  2. Press up or down request of odd floor say 5th floor. | | | |
| **Pre-conditions:**  1. Cars type should be either odd or even and no of cars > 1.  2. Alarm button of the cars should not be in “ON” state.  3. Up or down button of odd no. of floor is selected | | **Post-conditions:**  Floor indicator should display car id and car type and car should go to that floor | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | The down direction against floor number in the floor panel with floor indicator is pressed  Eg. Down button in Floor no. 5 is pressed | Floor indicator: CarId and CarType displays respective car id and “odd” respectively and car moves to that floor. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-09 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Odd-Even Algorithm, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate odd & even algorithm, floor panel with floor indicator component by selecting carType as either odd or even or seq & destination floor number as any even number. | | | |
| **Operation procedure:**  1. Select number of cars with car type as odd or even or seq, number of floors from configuration panel  2. Press up or down request of even floor say 8th floor. | | | |
| **Pre-conditions:**  1. Cars type should be either odd or even or seq and no of cars > 1.  2. Alarm button of the cars should not be in “ON” state.  3. Up or down button of even no. of floor is selected | | **Post-conditions:**  Floor indicator should display car id and car type and car should go to that floor | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | The down direction against floor number in the floor panel with floor indicator is pressed  Eg. Down button in Floor no. 8 is pressed | Floor indicator: CarId and CarType displays respective car id and “even” or “seq” respectively and car moves to that floor. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-10 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Odd-Even Algorithm, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate odd & even algorithm, floor panel with floor indicator component by selecting carType as either odd or even or seq & destination floor number as any odd number. | | | |
| **Operation procedure:**  1. Select number of cars with car type as odd or even or seq, number of floors from configuration panel  2. Press up or down request of odd floor say 5th floor. | | | |
| **Pre-conditions:**  1. Cars type should be either odd or even or seq and no of cars > 1.  2. Alarm button of the cars should not be in “ON” state.  3. Up or down button of odd no. of floor is selected | | **Post-conditions:**  Floor indicator should display car id and car type and car should go to that floor | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | The down direction against floor number in the floor panel with floor indicator is pressed  Eg. Down button in Floor no. 5 is pressed | Floor indicator: CarId and CarType displays respective car id and “odd” or “seq” respectively and car moves to that floor. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-11 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Odd-Even Algorithm, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate odd & even algorithm, floor panel with floor indicator component by selecting carType as all even & destination floor number as any odd number. | | | |
| **Operation procedure:**  1. Select number of cars with car type as all even, number of floors from configuration panel  2. Press up or down request of odd floor say 5th floor. | | | |
| **Pre-conditions:**  1. Cars type should be all even and no of cars > 1.  2. Alarm button of the cars should not be in “ON” state.  3. Up or down button of odd no. of floor is selected | | **Post-conditions:**  Floor indicator should display car id and car type and car should go to that floor | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | The down direction against floor number in the floor panel with floor indicator is pressed  Eg. Down button in Floor no. 5 is pressed | Nothing is displayed in the CarId and CarType of the floor indicator and car does not move. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-12 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Odd-Even Algorithm, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate odd & even algorithm, floor panel with floor indicator component by selecting carType as all old & destination floor number as any even number. | | | |
| **Operation procedure:**  1. Select number of cars with car type as all odd, number of floors from configuration panel  2. Press up or down request of even floor say 6th floor. | | | |
| **Pre-conditions:**  1. Cars type should be all odd and no of cars > 1.  2. Alarm button of the cars should not be in “ON” state.  3. Up or down button of even no. of floor is selected | | **Post-conditions:**  Floor indicator should display car id and car type and car should go to that floor | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | The down direction against floor number in the floor panel with floor indicator is pressed  Eg. Down button in Floor no. 6 is pressed | Nothing is displayed in the CarId and CarType of the floor indicator and car does not move. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-13 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Alarm Switch on User Panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and car movement. | | | |
| **Operation procedure:**  1. Select only one car, number of floors from configuration panel  2. Press Alarm ON button in user panel. | | | |
| **Pre-conditions:**  1. No. of cars = 1.  2. Alarm button of the cars should not be in “ON” state. | | **Post-conditions:**  All other buttons on the user panle should disable along with alarm on button and alarm off button should be enabled and car should move to the first floor. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | ON Alarm button of the car is pressed. | 1. All the buttons in user panel of the car are disabled  2. OFF alarm button is enabled  3. User panel queue is flushed out.  4. Car moves to first floor  5. Floor indicator of first floor should display the carId and carType  6. Car status is changed to Alarm\_on |  |
| **Required test scripts: -** None | | | |
| **Comments:** When alarm ON button is pressed, car fulfills the current request it is executing and then moves to the first floor. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-14 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Alarm Switch on User Panel, Car, Floor Panel, Algorithm | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and car movement and floor panel. | | | |
| **Operation procedure:**  1. Select only one car, number of floors from configuration panel  2. Press Alarm ON button in user panel.  3. While Alarm is ON, press any floor up/down button | | | |
| **Pre-conditions:**  1. No. of cars = 1.  2. Alarm button of the car should be in “ON” state.  3. Up/down button of any floor is pressed | | **Post-conditions:**  Floor indicator for alarm should keep showing changed color of the indicator and car should not move to handle floor request. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | ON Alarm button of the car is pressed and then up/down button of some floor is pressed. | Car does not move to handle floor request |  |
| **Required test scripts: -** None | | | |
| **Comments:** As there is no more car, so floor request is never catered | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-15 | | **Documented Date: 10/12/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Alarm Switch on User Panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and car movement. | | | |
| **Operation procedure:**  1. Select number of cars, number of floors from configuration panel  2. Press Alarm ON button in user panel. | | | |
| **Pre-conditions:**  1. No. of cars > 1.  2. Alarm button of the cars should not be in “ON” state. | | **Post-conditions:**  All other buttons of user panel of that car should be disabled along with alarm on button and alarm off button should be enabled and car should move to the first floor. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | ON Alarm button of any of the car is pressed. | 1. All the buttons in user panel of the car are disabled  2. OFF alarm button is enabled  3. User panel queue is flushed out.  4. Car moves to first floor  5. Car status is changed to Alarm\_on |  |
| **Required test scripts: -** None | | | |
| **Comments:** When alarm ON button is pressed, car fulfills the current request it is executing and then moves to the first floor. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-16 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Alarm Switch on User Panel, Car, Floor Panel, Algorithm | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and car movement and floor panel. | | | |
| **Operation procedure:**  1. Select number of cars, number of floors from configuration panel  2. Press Alarm ON button in user panel of one or more car.  3. While Alarm is ON, press any floor up/down button | | | |
| **Pre-conditions:**  1. No. of cars > 1.  2. Alarm button of any car should be in “ON” state such that there’s at least one car with no alarm and one car with alarm ON  3. Up/down button of any floor is pressed | | **Post-conditions:**  Floor indicator for alarm should keep showing changed color of the indicator and that particular car should not move to handle floor request.  The floor request should be handled by some other car which does not have alarm ON | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | 1. Select 3 cars from configuration panel.  2. Press alarm ON button of 1st car  3. Press floor up request for floor no. 5 | 1) First car is not allocated by algorithm to cater floor request.  2) Best car from the other two cars is assigned as best car.  3) Floor indicator shows carId as 2 or 3 and Alarm indicator for car1 is on on all floors. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-17 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User Panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and car movement. (invalid case) | | | |
| **Operation procedure:**  1. Select number of cars, number of floors from configuration panel  2. Press Alarm ON button in user panel of one or more car.  3. While Alarm is ON, press any other button on user panel | | | |
| **Pre-conditions:**  1. Alarm button of any car should be in “ON” state such that there’s at least one car with no alarm and one car with alarm ON  3. Try to press any other or alarm on button on same car’s user panel except Alarm Off button | | **Post-conditions:**  As soon as Alarm on button is pressed, other buttons along with alarm on button on user panel should get disable and alarm off button should get enable | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | 1. Press alarm on button of some car’s user panel.  2. Press any other button on same car’s user panel or again press alarm on button but not alarm off button | User should not be able to press any button except alarm off button on user panel once alarm is on. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-18 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User Panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and car movement. (invalid case) | | | |
| **Operation procedure:**  1. Select number of cars, number of floors from configuration panel  2. Press Alarm OFF button in user panel of one or more car. | | | |
| **Pre-conditions:**  1. Alarm button of the car should not be in “ON” state. | | **Post-conditions:** | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | 1. Press alarm off button of some car’s user panel. | User should not be able to press off button as alarm is not in on state and also alarm off button should be in disable state. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-20 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User Panel, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and floor panel. | | | |
| **Operation procedure:**  1. Select number of cars, number of floors from configuration panel  2. Press Alarm ON button on user panel of one or more car.  3. Press Alarm Off button on user panel of the car, which has alarm on | | | |
| **Pre-conditions:**  1. Alarm button of the car should be in “ON” state. | | **Post-conditions:**  1. All other buttons on user panel of that car should be activated and alarm off button should be disabled | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | 1. Press alarm off button of the car whose alarm is in ON state. | 1. All other buttons on car user panel are activated and alarm off button is disabled.  2. Car’s status changes from Alarm\_on to Idle.  3. Alarm indicator on floor panel is turned off. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-21 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User Panel, Floor Panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and floor panel and car. | | | |
| **Operation procedure:**  1. Select only one car, number of floors from configuration panel  2. Press Alarm ON button on user panel of that car.  3. Press Alarm Off button on user panel of the same car  4. Press up/down button of any floor | | | |
| **Pre-conditions:**  1. Alarm button of the car should be in “ON” state. | | **Post-conditions:**  1. All other buttons on user panel of that car should be activated and alarm off button should be disabled and car should cater to floor request. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | 1. Press alarm off button of the car whose alarm is in ON state.  2. Press up/down button of any floor. | 1. All other buttons on car user panel are activated and alarm off button is disabled.  2. Car’s status changes from Alarm\_on to Idle.  3. Alarm indicator on floor panel is turned off.  4. Car caters to the floor request as it is the only car present in the system |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-22 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User Panel, Floor Panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and floor panel and car. | | | |
| **Operation procedure:**  1. Select number of cars, number of floors from configuration panel  2. Press Alarm ON button on user panel of some car.  3. Press Alarm Off button on user panel of the same car  4. Press up/down button of any floor | | | |
| **Pre-conditions:**  1. Alarm button of some car should be in “ON” state. | | **Post-conditions:**  1. All other buttons on user panel of that car should be activated and alarm off button should be disabled and car should cater to floor request. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | 1. Press alarm off button of the car whose alarm is in ON state.  2. Press up/down button of any floor. | 1. All other buttons on car user panel are activated and alarm off button is disabled.  2. Car’s status changes from Alarm\_on to Idle.  3. Alarm indicator on floor panel is turned off.  4. Car can be returned as best car and thus, caters to floor request. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-23 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User Panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button and floor button on user panel and car. | | | |
| **Operation procedure:**  1. Select number of cars, number of floors from configuration panel  2. Press Alarm ON button on user panel of some car.  3. Press Alarm Off button on user panel of the same car  4. Press any floor button on user panel | | | |
| **Pre-conditions:**  1. Alarm button of some car should be in “ON” state. | | **Post-conditions:**  1. All other buttons on user panel of that car should be activated and alarm off button should be disabled and car should cater to floor request. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | 1. Press alarm off button of the car whose alarm is in ON state.  2. Press any floor button of the same car’s user panel. | 1. All other buttons on car user panel are activated and alarm off button is disabled.  2. Car’s status changes from Alarm\_on to Idle.  3. Alarm indicator on floor panel is turned off.  4. Car moves to the desired floor as pressed in user panel |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-24 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User Panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and Car. | | | |
| **Operation procedure:**  1. Select number of cars, number of floors from configuration panel  2. Press Alarm ON button on user panel of some car.  3. Press Alarm Off button on user panel of the same car  4. Again press alarm ON button of the same car | | | |
| **Pre-conditions:**  1. Alarm button of some car should be in “ON” state. | | **Post-conditions:**  1. All other buttons on user panel of that car should be activated and alarm off button should be disabled. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | 1. Press alarm off button of the car whose alarm is in ON state.  2. Again press alarm on button of the same car | 1. All other buttons on car user panel are activated and alarm off button is disabled.  2. Car’s status changes from Alarm\_on to Idle.  3. Alarm indicator on floor panel is turned off.  4. When alarm on button is pressed again, all buttons are disabled and alarm off button is enabled and car status is changed from idle to alarm\_on. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-25 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User Panel, Car, Floor Panel, Algorithm | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and Car movement in response to floor request. | | | |
| **Operation procedure:**  1. Select number of cars, number of floors from configuration panel  2. Press Alarm ON button on user panel of some car.  3. Press Alarm Off button on user panel of the same car  4. Press any up/down button on any floor | | | |
| **Pre-conditions:**  1. Alarm button of some car should be in “ON” state. | | **Post-conditions:**  1. All other buttons on user panel of that car should be activated and alarm off button should be disabled and car should cater to the floor requests. | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | 1. Press alarm off button of the car whose alarm is in ON state.  2. Press up/down button of any floor | 1. All other buttons on car user panel are activated and alarm off button is disabled.  2. Car’s status changes from Alarm\_on to Idle.  3. Alarm indicator on floor panel is turned off.  4. Car is assigned as best car to floor request as per the algorithm selected |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-26 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Alarm Switch on User Panel, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and floor panel indicator. | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press Alarm ON button in user panel of some care. | | | |
| **Pre-conditions:**  1. Alarm button of the cars should not be in “ON” state. | | **Post-conditions:**  Alarm Floor indicator color on all floors for the specific car should change color | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | ON Alarm button of the car is pressed. | 1. All the buttons in user panel of the car are disabled  2. OFF alarm button is enabled  3. Alarm Floor indicator for that car on all floors changes color to red. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-27 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Alarm Switch on User Panel, Floor Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate alarm button on user panel and floor panel indicator. | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press Alarm ON button in user panel of some car.  3. Press Alarm Off button in user panel of the same car | | | |
| **Pre-conditions:**  1. Alarm button of the cars should be in “ON” state. | | **Post-conditions:**  Alarm Floor indicator color on all floors for the specific car should change color | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | Off Alarm button of the car is pressed, whose alarm is switched on. | 1. All the buttons in user panel of the car are enabled  2. OFF alarm button is disabled  3. Alarm Floor indicator for that car on all floors changes color to normal. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-28 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User Panel with odd, even and sequential functionality | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate user panel by selecting more than one car and car type as seq, odd or even. | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Select different type of car types from configuration. | | | |
| **Pre-conditions:**  1. The system should function properly and should not give any error. | | **Post-conditions:**  The system should display all the cars with different configuration types with out any exception | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User selects more than one car and cartype as odd, even or seq in configuration gui | The number of cars and their corresponding user panels i.e odd, even or seq should be displayed to the user |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-29 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User Panel with odd, even and sequential functionality | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate user panel by selecting only one car and car type as odd or even. | | | |
| **Operation procedure:**  1. Select only one car, number of floors from configuration panel  2. No selection of car type as odd, even or seq. should be there as no. of cars = 1 | | | |
| **Pre-conditions:**  1. The system should function properly and should not give any error. | | **Post-conditions:**  The system should display the car with seq. configuration types with out any exception | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User selects only one car and no choice of cartype is given in configuration gui | The car should be displayed as seq car type to the user |  |
| **Required test scripts: -** None | | | |
| **Comments:** When number of cars selected = 1, then default car typei.e. seq is used | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-30 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User panel, Car, Door Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press open button on elevator’s internal pad and monitor the behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press elevator car open button | | | |
| **Pre-conditions:**  1. The elevator car should be at some floor in stopped or idle position.  2. The door of the car are closed | | **Post-conditions:**  Elevator car door should open on pressing the open button | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses door open button when elevator is standing still on some floor | The door is opened on pressing open button |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-31 | | **Documented Date: 10/13/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User panel, Car, Door Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press open button on elevator’s internal pad and monitor the behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press elevator car open button when elevator is in motion | | | |
| **Pre-conditions:**  1. The elevator car should be running.  2. The elevator car status should show moving\_up or moving\_down | | **Post-conditions:** | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses door open button when elevator is in motion | Nothing happens |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-32 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User panel, Car, Door Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press open button on elevator’s internal pad and monitor the behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press elevator car open button when elevator door is closing | | | |
| **Pre-conditions:**  1. The elevator car is stopped at some floor.  2. The elevator car status should show door\_closing | | **Post-conditions:** | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses door open button when elevator doors are closing | Nothing happens |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-33 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User panel, Car, Door Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press close button on elevator’s internal pad and monitor the behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press elevator car door close button | | | |
| **Pre-conditions:**  1. The elevator car should be at some floor in stopped or idle position.  2. The door of the car are opened | | **Post-conditions:**  Elevator car door should close on pressing the close button | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses door close button when elevator is standing still on some floor | The door is closed on pressing close button |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-34 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User panel, Car, Door Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press close button on elevator’s internal pad and monitor the behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press elevator car close button when elevator is in motion | | | |
| **Pre-conditions:**  1. The elevator car should be running.  2. The elevator car status should show moving\_up or moving\_down | | **Post-conditions:** | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses door close button when elevator is in motion | Nothing happens |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-35 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – User panel, Car, Door Panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press close button on elevator’s internal pad and monitor the behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press elevator car close button when elevator door is opening | | | |
| **Pre-conditions:**  1. The elevator car is stopped at some floor.  2. The elevator car status should show door\_opening | | **Post-conditions:** | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses door close button when elevator doors are opening | Nothing happens |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-36 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Floor up button of floor panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press up button on floor panel and monitor the car movement behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press elevator floor up button of any floor when elevator is either stopped at different floor or running | | | |
| **Pre-conditions:**  1. The elevator should not be present on the same floor where the up button is pressed | | **Post-conditions:**  1. The elevator car should fulfill that request according to the algorithm selected in configuration panel | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses floor up request when no elevator car was present on that floor irrespective of status of elevator car (in motion or stopped) | The floor request is queued to floor request queue and was handled by the best car as returned by the algorithm in action. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-37 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Floor up button of floor panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press up button on floor panel and monitor the car movement behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press elevator floor up button of any floor when elevator is stopped at same floor | | | |
| **Pre-conditions:**  1. The elevator should be present on the same floor where the up button is pressed | | **Post-conditions:**  The car already present on the floor handles the request | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses floor up request when some elevator car was present on that floor | The floor request was handled by that car only and doors are opened |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-38 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Floor down button of floor panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press down button on floor panel and monitor the car movement behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press elevator floor down button of any floor when elevator is either stopped at different floor or running | | | |
| **Pre-conditions:**  1. The elevator should not be present on the same floor where the down button is pressed | | **Post-conditions:**  1. The elevator car should fulfill that request according to the algorithm selected in configuration panel | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses floor down request when no elevator car was present on that floor irrespective of status of elevator car (in motion or stopped) | The floor request is queued to floor request queue and was handled by the best car as returned by the algorithm in action. |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-39 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Floor down button of floor panel | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press down button on floor panel and monitor the car movement behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Press elevator floor down button of any floor when elevator is stopped at same floor | | | |
| **Pre-conditions:**  1. The elevator should be present on the same floor where the down button is pressed | | **Post-conditions:**  The car already present on the floor handles the request | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses floor down button when some elevator car was present on that floor | The floor request was handled by that car only and doors are opened |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-40 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press up/down button on any floor other than 1st floor and monitor the car movement behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Let the car be on 1st floor.  2. Press elevator floor up/down button of any floor other than 1st floor | | | |
| **Pre-conditions:**  1. The elevator should be present on 1st floor | | **Post-conditions:**  Best car is returned by the algorithm to cater to the floor request | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses floor up/down button of any floor other than 1st floor when some elevator car was present on 1st floor | The floor request was handled by the best car returned by the algorithm |  |
| **Required test scripts: -** None | | | |
| **Comments:** If only one car is present, then that car will handle all the floor request. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-41 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press down button on 1st floor and monitor the car movement behavior (invalid case) | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Let the car be on 1st floor.  2. Press elevator floor down button of 1st floor, if possible | | | |
| **Pre-conditions:**  1. The elevator should be present on 1st floor | | **Post-conditions:**  Best car is returned by the algorithm to cater to the floor request | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses floor down button of 1st floor | User should not be able to press that button as there is not floor below 1st floor |  |
| **Required test scripts: -** None | | | |
| **Comments:** If only one car is present, then that car will handle all the floor request. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-42 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press up/down button on any floor other than top floor and monitor the car movement behavior | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Let the car be on top floor.  2. Press elevator floor up/down button of any floor other than top floor | | | |
| **Pre-conditions:**  1. The elevator should be present on top floor | | **Post-conditions:**  Best car is returned by the algorithm to cater to the floor request | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses floor up/down button of any floor other than top floor when some elevator car was present on top floor | The floor request was handled by the best car returned by the algorithm |  |
| **Required test scripts: -** None | | | |
| **Comments:** If only one car is present, then that car will handle all the floor request. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-43 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Press up button on top floor and monitor the car movement behavior (invalid case) | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Let the car be on 1st floor.  2. Press elevator floor up button of top floor, if possible | | | |
| **Pre-conditions:**  1. The elevator should be present on top floor | | **Post-conditions:**  Best car is returned by the algorithm to cater to the floor request | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User presses floor up button of top floor | User should not be able to press that button as there is no floor above top floor |  |
| **Required test scripts: -** None | | | |
| **Comments:** If only one car is present, then that car will handle all the floor request. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-44 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Door Panel, Alarm Indicator, Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate text option for floor panel, door panel and alarm component | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Select all options to be text. | | | |
| **Pre-conditions:**  1. Select door, floor and alarm option as text in configuration panel | | **Post-conditions:** | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User selects all configuration option as text | The elevator system displayed has text UI on Door Panel, Floor Panel and Alarm |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID:** ST-EP-45 | | **Documented Date: 10/14/2011** | |
| **Tester Name:** Anshu Basia | | **Test Item:** Elevator Component – Door Panel, Alarm Indicator, Floor panel, Car | |
| **Product Name:** Elevator System | | **Version No.:** 3 | |
| **Test Type:** Equivalence Partition Black Box System Testing | | **Test Suite #:** 1.0 | |
| **Test case description:** Validate symbol option for floor panel, door panel and alarm component | | | |
| **Operation procedure:**  1. Select no. of cars, number of floors from configuration panel  2. Select all options to be symbol. | | | |
| **Pre-conditions:**  1. Select door, floor and alarm option as symbol in configuration panel | | **Post-conditions:** | |
| **Steps** | **Inputs data and/or events:** | **Expected output data and/or events:** | Pass/Fail |
| 1 | User selects all configuration option as symbol | The elevator system displayed has symbolic UI on Door Panel, Floor Panel and Alarm |  |
| **Required test scripts: -** None | | | |
| **Comments:** | | | |

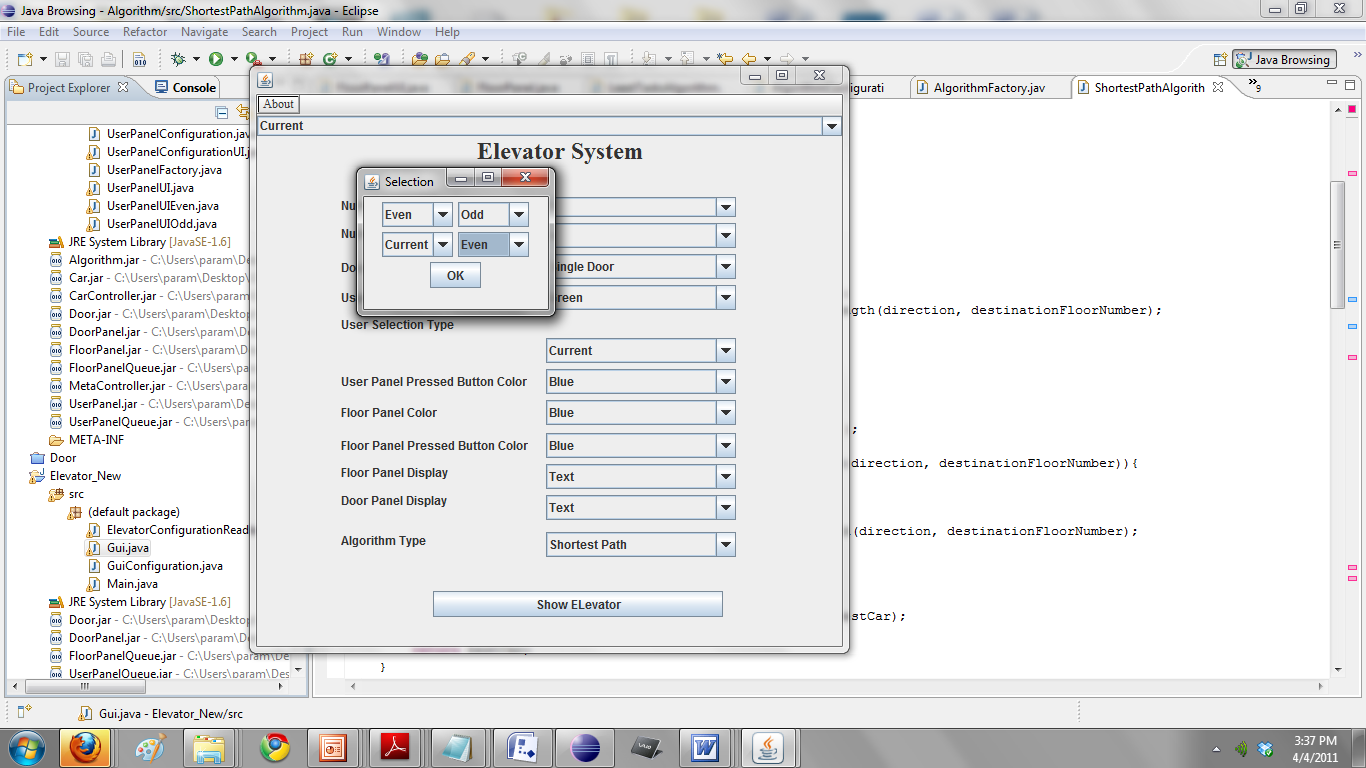
## 3.4 Configuration Testing

We are using the semantic tree model for configuration testing. The semantic tree of the elevator system is shown below.

Admin Console:

In the system, there are a number of configurable components.

Admin console is one of the ways of configuring the elevator system. Below is the snapshot of admin console.



Admin console

Below is the semantic diagram for Admin console.

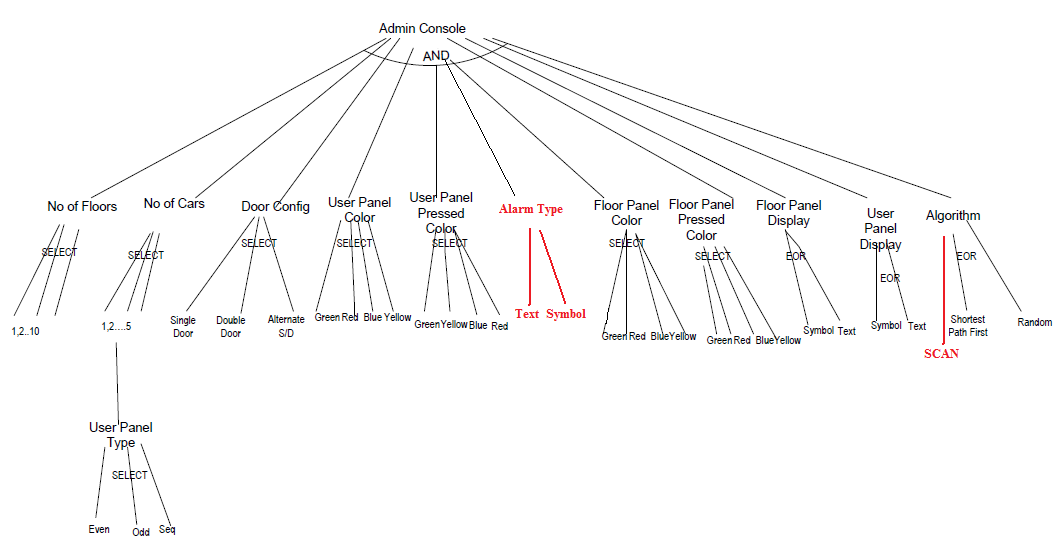


Figure 9: Semantic diagram for Admin Console

Semantic tree of Admin Console

For a configurable software (or composite component), it is useful to define adequate test criteria to validate its various configurable composition structures. Here we define Semantic tree concept based on Dr. Gao’s paper. Here, we provide test criteria for configurable composition structures.

Adequate Test Criterion for Single-Node Inclusion:

For a node Ni (where Ni in NS), this test criterion is achieved if and only if a test set TS of C has been exercised when the corresponding component of Ni is configured as a part of C.

Adequate Test Criterion for All-Node Inclusion:

For each node Nj (where Nj in NS), this test criterion is achieved if and only if a test set TS of C has been exercised when the corresponding component of Nj is configured as a part of C.

Adequate Test Criterion for Single-Parent-Node Composition:

For a parent node NP (where NP in NS), this test criterion is achieved if and only if a test set TS of C has been exercised under both of the following conditions:

a) The corresponding component of NP is configured as a part of C.

b) NP has been configured with all of its child nodes based on each of its semantic relation RP in RS.

Adequate Test Criterion for All-Parent-Node Composition:

For each parent node Np (where NSpj in NS), this test criterion is achieved if and only if a test set (TS) of C has been exercised to achieve its adequate Single-Parent-Node Composition.

EOR ( Np <C1, C2> ) : Np has two configurations with its two child nodes C1 and C2. Each child node can be exclusively configured with its parent node.

AND( Np <C1, C2,…Cn) : Np only has one configuration with its child nodes, in which all child nodes are included.

SELECT( Np <C1, C2,…Cn): Np has n different configurations. Each of them only configured with one of its selective child nodes.

For any parent node NPi of GS, its configuration test complexity can be computed based on its semantic relation with its child nodes and their configurable organization test complexity. Let Nj be a child node of NPi. and its test complexity for organization configurations can be computed below as T-complexityC(NPi) in GS.

If its semantic relation with its child nodes is EOR, then its complexity can be computed with the following.

T-complexityC(Np) =T-complexityC(NC1)+T-complexityC(NC2)

Where NC1 and NC2 are the two child nodes

If its semantic relation with its child nodes is OR, then its complexity can be computed with the following.

T-complexityC(Np) = T-complexityC(NC1)+T-complexityC(NC2) + T-complexityC(NC1) \* T-complexityC(NC2)

Where NC1 and NC2 are the two child nodes of NPi

If its semantic relation with its child nodes is AND, then its complexity can be computed with the following.

T-complexityC(Np) = Π (T-complexityC(NCj))

Where j = 1, …, n, n is the number of its child nodes, and NCj is a child node of NPi.

If its semantic relation with its child nodes is SELECT-1, then its complexity can be computed with the following.

T-complexityC(NPi) = Σ (T-complexityC(NCj))

*Where* j = 1, …, n, n is the number of its child nodes, and *NCj* is a child node of *NPi*.

These formulae have been applied to calculate the test complexity of each configurable node in the admin console. The corresponding complexity numbers a noted just near the node in the semantic tree diagram.

We can see that the total complexity of the system is:

T-complexityC(Np) = Π (T-complexityC(NCj)) Where j = 1, …, n, n is the number of its child nodes, and NCj is a child node of NPi.

Which gives us the system complexity of admin console as:

T-Complexity = 10 X 5 X 3 X 2 X 2 X 2 X 2 X 4 X 4 X4 X 4 X 3 = 1843200

Spanning tree of Admin Console:

We can build the test cases by considering the spanning tree of the admin console.

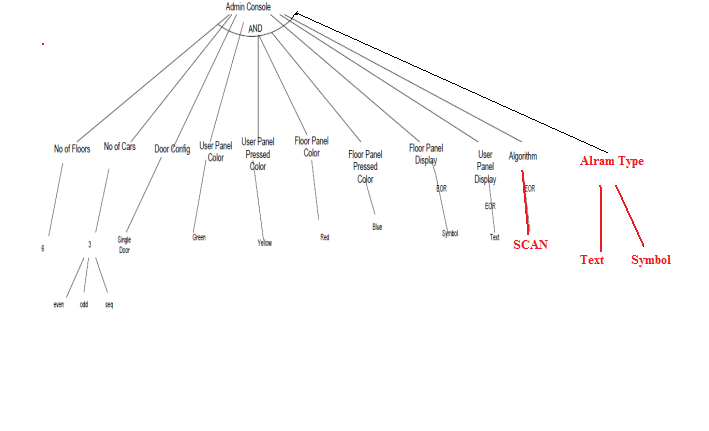


Figure 10: Spanning Tree for Admin console

Test Cases based on above configuration in Admin Console

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID: SC-ST-01 | | Documented Date: 10/10/2011 | |
| Tester Name: Shubhada Narkar | | Test Item: Admin Console | |
| Product Name: Elevator System | | Version No.: 3.0 | |
| Test Type: System-level configuration testing | | Test Suite#: 1.0 | |
| Test Case Description: To check if the number of floors entered in admin console reflects when the system runs | | | |
| Operation Procedure: Main.java file is run and Admin console opens. | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | Pass/ Fail |
| 1 | Configure No. of floors= 10 | The number of floors on the elevator simulator should be same as the one entered on admin console. No. of floors = 10 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-02 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the number of cars entered in admin console reflects when the system runs | | | | | |
| Operation Procedure: Main.java file is run and Admin console opens. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/Fail |
| 1 | Configure No. of cars= 4 | The number of cars on the elevator simulator should be same as the one entered on admin console. No. of cars =4.  New Frame with 4 combobox opens, to configure userpanel for each car. (changed test case) | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-03 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the door type selected in admin console reflects when the system runs. | | | | | |
| Operation Procedure: Main.java file is run and Admin console opens. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | Door type = single | Single door configuration is displayed | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-04 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the user panel color selected in admin console reflects when the system runs. | | | | | |
| Operation Procedure: Main.java file is run and Admin console opens. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | User panel color = Green | User panel color is green. | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-05 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the user panel pressed color selected in admin console reflects when the system runs. | | | | | |
| Operation Procedure: Main.java file is run and Admin console opens. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | User panel pressed color = yellow | User panel pressed color = yellow | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-06 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the floor panel color selected in admin console reflects when the system runs. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | Floor panel = red | Floor panel color = red | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-07 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the floor panel pressed color selected in admin console reflects when the system runs. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | Floor panel pressed color = blue | Floor panel pressed color = blue | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-08 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the floor panel display Text/Symbol selected in admin console reflects when the system runs. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | Floor panel display = Symbol | Floor panel display = Symbol | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-09 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the user panel display Text/Symbol selected in admin console reflects when the system runs. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | User panel display = Text | User panel display = Text | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-10 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the User Panel type selected in User Panel Selection reflects when the system runs. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | Userpanel Type = Even  Car 1 configured to have even user panel | UserPanel = shows even number userpanel. | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-11 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the User Panel type selected in User Panel Selection reflects when the system runs. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | Userpanel Type = Odd  Car 2 configured to have odd user panel | UserPanel = shows odd number userpanel. | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-12 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the User Panel selected in User Panel Selection reflects when the system runs. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | Userpanel Type = Even  Car 3 configured to have sequential user panel | UserPanel = shows sequential number userpanel. | | |  |

### 

New Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-13 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the Algorithm selected in admin console reflects when the system runs. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | Algorithm = SCAN | Algorithm = SCAN | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-14 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the alarm type Text/Symbol selected in admin console reflects when the system runs. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | Alarm Type = Text | Alarm type = Text | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID: SC-ST-15 | | | Documented Date: 10/10/2011 | | |
| Tester Name: Shubhada Narkar | | | Test Item: Admin Console | | |
| Test Type: System-level configuration testing | | | Version No.:3.0 | | |
| Product Name: Elevator System | | | Test Suite#: 1.0 | | |
| Test Case Description: To check if the alarm type Text/Symbol selected in admin console reflects when the system runs. | | | | | |
| Pre- Conditions: Admin console should be configured and Show Elevator button should be pressed. | | | | Post- Conditions:  Elevator simulator should be running. | |
| Step: | Input Data: | Expected  Output Data: | | | Pass/ Fail |
| 1 | Alarm Type = Symbol | Alarm type = symbol | | |  |

# 4.0 SUMMARY

In this Regression test design documented we have mentioned about the different types of Black box and White Box testing. This design document contains the following:

White-Box Regression Test Design

Component Regression Test Design

Component Re-Test Models, Methods, and Criteria

Component Re-Test Methods and Test Data

System Function Test Design

System Function Re-Test Models, Methods, and Criteria

System Function Test Case and Test Data

The methods that we used for the different types of regression testing are decision table, equivalence partition for black box testing and branch based and state based for white box testing. We have developed the test cases for the newly added components and the other affected components after changes.

# References

* Website: <http://softwaretestingfundamentals.com>
* Jerry Gao; Yumei Wu; Lee Chang; Sigurd Meldal; , "Measuring Component-Based Systems Using a Systematic Approach and Environment," *Service-Oriented System Engineering, 2006. SOSE '06. Second IEEE International Workshop* , vol., no., pp.121-129,Oct.2006.doi:10.1109/SOSE.2006.19  
  URL: [http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4027127&isnumber=4027102](http://ieeexplore.ieee.org.libaccess.sjlibrary.org/stamp/stamp.jsp?tp=&arnumber=4027127&isnumber=4027102)
* Class notes and slides

# Appendix A: Previous Version

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Component Name | Version No. | No. of Classes | No. of functions | Total of component size(KLC) | No. of Source Code Files |
| Admin Console | V\_2 | 7 | 35 | 0.96 | 4 |
| Algorithm | V\_2 | 5 | 8 | 0.25 | 6 |
| Meta controller | V\_2 | 3 | 11 | 0.097 | 5 |
| UserPanelQueue | V\_2 | 7 | 22 | 0.35 | 8 |
| Car Controller | V\_2 | 1 | 9 | 0.17 | 4 |
| Floor Panel | V\_2 | 7 | 32 | 0.44 | 7 |
| User Panel | V\_2 | 12 | 57 | 0.87 | 13 |

# 

# Appendix B: Updated Version

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Component Name | Version No. | No. of Classes | No. of functions | Total of component size (KLC) | No. of Source Code Files |
| AdminConsole1 | V\_3 | 7 | 40 | 1.026 | 4 |
| Algorithm | V\_3 | 6 | 10 | 0.417 | 7 |
| MetaController | V\_3 | 3 | 13 | 0.117 | 5 |
| UserPanelQueue | V\_3 | 7 | 26 | 0.402 | 8 |
| CarController | V\_3 | 1 | 11 | 0.178 | 4 |
| FloorPanel | V\_3 | 7 | 38 | 0.641 | 11 |
| UserPanel | V\_3 | 13 | 74 | 1.317 | 15 |

# Appendix C: Component Change Records in the system

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Component Name (Version) | Component API Function | | | Component Interaction function and data changes | | Internal Function/ Data Changes | | |
| Add API function (#) | Delete API function (#) | Change API function (#) | Call to other component (#) | Protocol/queue messages to other component (#) | Add Internal Function/Data (#) | Delete Internal Function/Data (# | Change Internal Function/Data (# |
| AdminConsole\_v3 | - | - | - | 3 | - | 2 | - | - |
| Algorithm\_v3 | 1 | - | - | - | - | 1 | - | - |
| MetaController\_v3 |  | - | - | 1 | - | 1 | - | - |
| UserPanelQueue\_v3 | - | - | - | 2 | 2 | 3 | - | - |
| CarController\_v3 | 1 | - | - | - | - | 1 | - | - |
| FloorPanel\_v3 | - | - | - | 1 | - | 5 | - | - |
| UserPanel\_v3 | 1 |  |  |  |  | 16 |  | 7 |

# 

# Appendix D: System Change Records

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Added Component | Deleted Component | Changed Component | Composition Relation Change | Configuration Relation Change |
| - | - | AdminConsole\_v3 | Sending Algorithm type to Metacontroller | Alarm and Scan First Algorithm configuration |
| - | - | Algorithm\_v3 | Scan Algorithm function added |  |
| - | - | MetaController\_v3 |  | Scan Algorithm configuration |
| - | - | UserPanelQueue\_v3 |  | Alarm Component and Scan first algorithm changes. |
| - | - | CarController\_v3 | Processing Alarm request |  |
| - | - | FloorPanel\_v3 |  | Configuring Alarm indicator on Floor Panel |
| - | - | UserPanel\_v3 |  | Alarm Component changes |

# Appendix E: Component Change Firewall Records for updated version

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Component Name  (Version No.) | Affected Component API Function Changes | Affected Component Interactions | | Affected Internal Function and Data | |
| Affected API Function | Affected call to other components | Affected protocol/queue messages to other components | Affected Internal Data | Affected Internal Function |
| AdminConsole\_v3 | 1 | 3 | - | 2 | - |
| Algorithm\_v3 | 1 | - | - | - | - |
| MetaController\_v3 | - | 1 | - | - | - |
| UserPanelQueue\_v3 | - | 2 | 2 | 1 | - |
| CarController\_v3 | 1 | - | - | - | - |
| FloorPanel\_v3 | 1 | 1 | - | - | - |
| UserPanel\_v3 | 1 | - | - | - | 7 |



Figure 11 : Component Firewall Changed Diagram- Class Level

# Appendix F: System Change Firewall Records for updated version

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Affected Component  (Version No.) | Affected System-Level Composition | | Affected System-Level Component Configuration | | Affected System-Level Architectures | |
| Added New Composit-ions | Deleted Composit-ions | Added New Configurat-ion | Deleted Configurat-ion | Added New Architect-ures | Deleted Architect-ures |
| UserPanel\_v3 | 2 | - | 2 | - | 1 | - |
| FloorPanel\_v3 | 1 | - | - | - | 1 | - |
| AdminConsole\_v3 | 2 | - | 2 | - | 3 | - |
| Algorithm\_v3 | 1 | - | - | - | 1 | - |
| Car\_v3 | 1 | - | - | - | - | - |



Figure 12 : System Firewall Changed Diagram

# Appendix G: Component Test Change Firewall Records of updated version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component Name  (Version No.) | Test Method #1 – State Based Testing (White Box) | | | |
| Added Test Cases(#) | Changed Test Cases(#) | Deleted Test Cases(#) | Reused Test Cases(#) |
| UserPanel Ver. 3 | 9 | - | - | - |

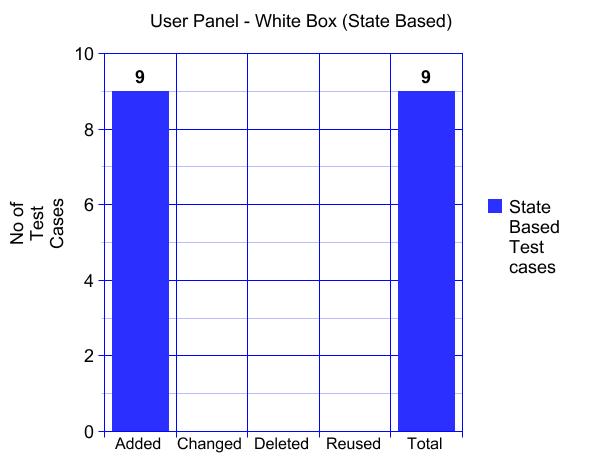


Figure 13 : White Box – State Based Testing Graph for User Panel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component Name  (Version No.) | Test Method #2 – State Based Testing (White Box) | | | |
| Added Test Cases(#) | Changed Test Cases(#) | Deleted Test Cases(#) | Reused Test Cases(#) |
| FloorPanel with Alarm Indicator Ver. 3 | 5 | - | - | - |

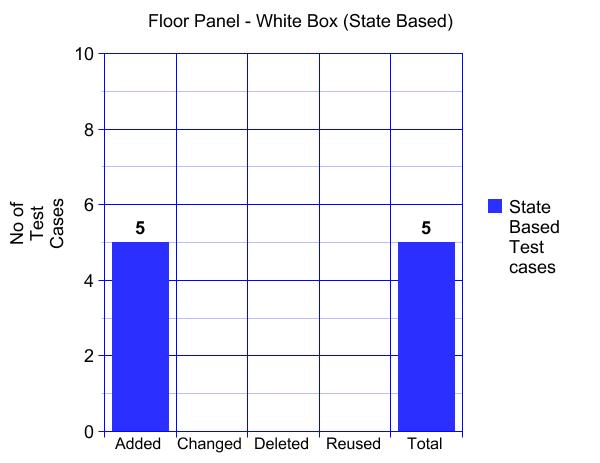


Figure 14 : White Box – State Based Testing Graph for Floor Panel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component Name  (Version No.) | Test Method #3 – Branch Based Testing (White Box) | | | |
| Added Test Cases(#) | Changed Test Cases(#) | Deleted Test Cases(#) | Reused Test Cases(#) |
| MetaController with Scan Algorithm Ver.3 | 7 | - | - | 5 |

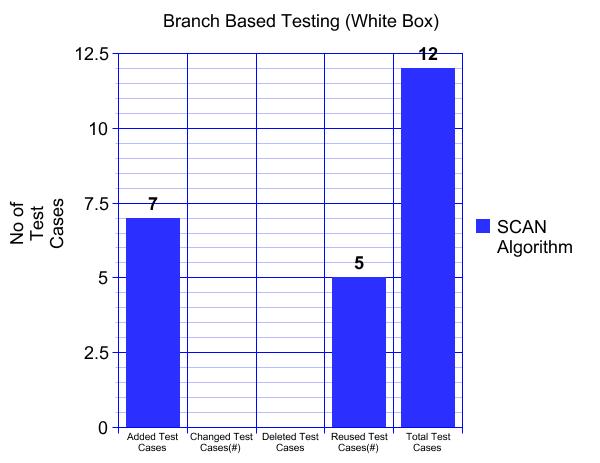


Figure 15 : White Box – Branch Testing Graph for SCAN Algorithm

Summary for White box Testing

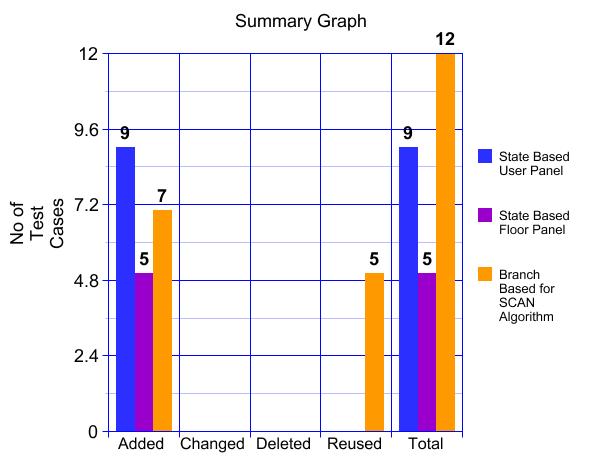


Figure 16: Summary Graph for White box Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component Name  (Version No.) | Test Method #4 – Decision table (Black Box) | | | |
| Added Test Cases(#) | Changed Test Cases(#) | Deleted Test Cases(#) | Reused Test Cases(#) |
| User Panel Ver. 3 | 3 | - | - | 8 |

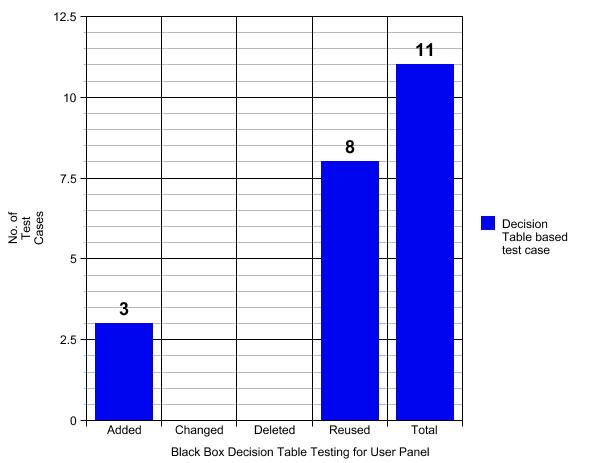


Figure 17: Black Box Decision Table test case Graph for User panel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component Name  (Version No.) | Test Method #5 – Equivalence partition ( Black Box) | | | |
| Added Test Cases(#) | Changed Test Cases(#) | Deleted Test Cases(#) | Reused Test Cases(#) |
| User panel Version 3 | 7 | - | - | 14 |

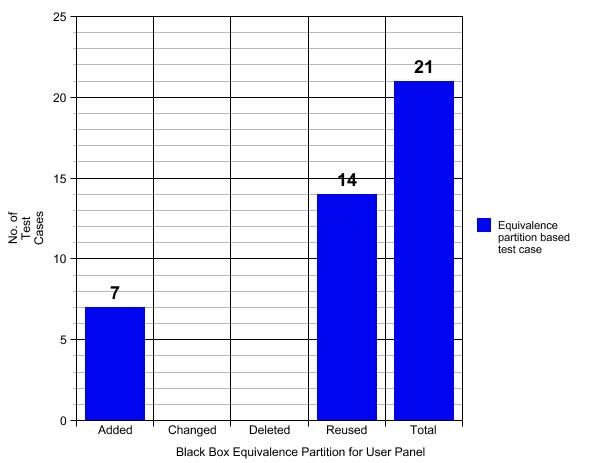


Figure 18 : Black Box Equivalence Partition test case graph for User Panel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component Name  (Version No.) | Test Method #6 – Decision Table (Black Box) | | | |
| Added Test Cases(#) | Changed Test Cases(#) | Deleted Test Cases(#) | Reused Test Cases(#) |
| Floor panel with alarm indicator Ver.3 |  | 2 | - | 5 |

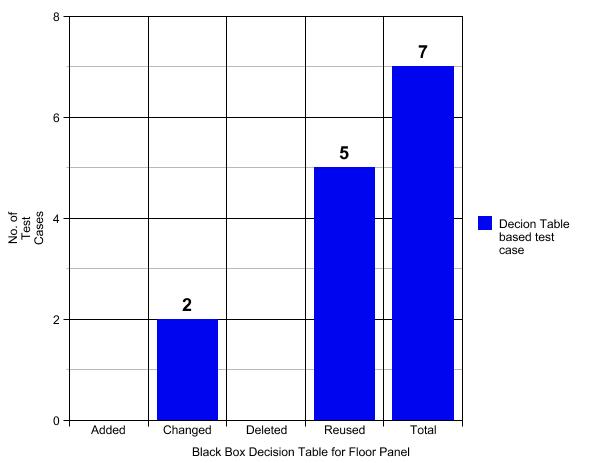


Figure 19 : Black Box Decision Table test case graph for Floor Panel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component Name  (Version No.) | Test Method #7 – Equivalence partition (Black Box) | | | |
| Added Test Cases(#) | Changed Test Cases(#) | Deleted Test Cases(#) | Reused Test Cases(#) |
| Floor panel with alarm indicator Ver.3 | 2 | - | - | 7 |

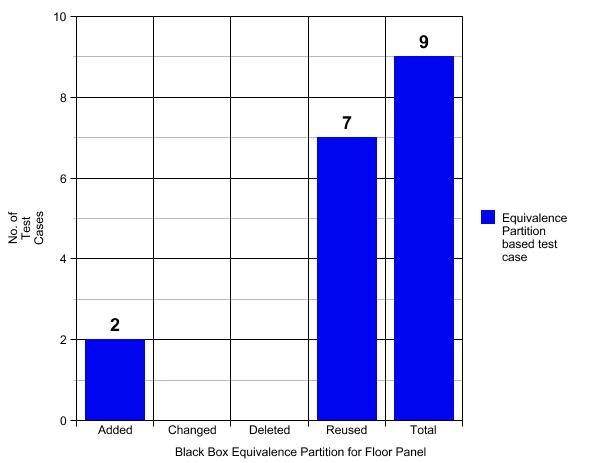


Figure 20 : Black Box Equivalence Partition Test case graph for Floor Panel

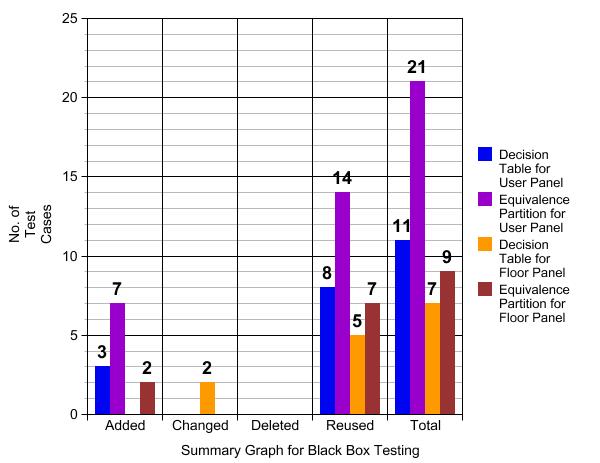


Figure 21 : Summary Graph for Component Black Box testing

# Appendix H: System Test Change Firewall Records of updated version

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Type | Test Method | Added | Changed | Deleted | Reuse |
| System Testing | Decision Table | 26 | 8 | - | 8 |
| Equivalence Partition | 18 | 14 | - | 13 |

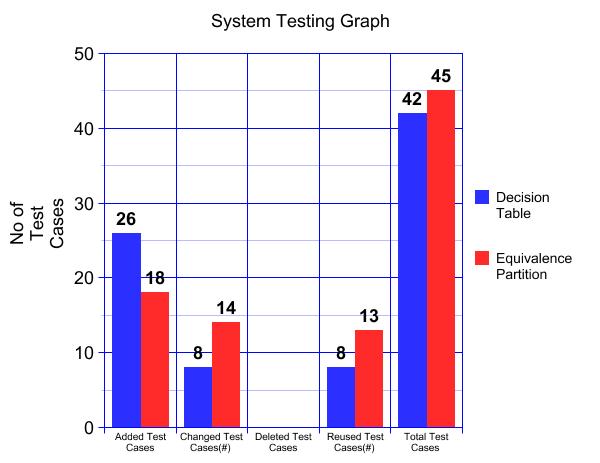


Figure 22 : System Testing Graph for Elevator System