One Percent Productions

<<This RequisitePro™ Outline is provided as a starting point to assist Requirements Writers in the process of defining the structure of a Software Requirements Specification (SRS) document. Required contents of an SRS document vary across the many segments of the software industry. This is evident in the existence of so many different standards (DOD, ANSI/IEEE, etc.). Therefore, this Requisite™ Outline is only a skeleton to jump-start the specification process. >>

Event Organization Console   
  
Software Requirements Specification

© 2008

Dan Dobbs

Josh Weatherly

George Williamson

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 02/28/08 | 1.0 | Initial Version | Dan Dobbs, Josh Weatherly, George Williamson |
| 03/02/08 | 1.1 | Non-Functional Requirements Revision | Dan Dobbs, Josh Weatherly, George Williamson |
| 3/04/08 | 1.2 | Functional Requirements Revision | Dan Dobbs, Josh Weatherly, George Williamson |
| 3/06/08 | 1.3 | Added Sequence Diagrams and Use Cases | Dan Dobbs, Josh Weatherly, George Williamson |
| 03/07/08 | 1.4 | Added Traceability Matrices | Dan Dobbs, Josh Weatherly, George Williamson |
| 03/07/08 | 1.5 | Added Screen Shots | Dan Dobbs, Josh Weatherly, George Williamson |
| 03/08/08 | 1.6 | Document Integration/Formatting | Dan Dobbs, Josh Weatherly, George Williamson |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Purpose 1

1.1. Scope 1

1.2. Definitions, Acronyms, Abbreviations 1

1.2.1. Definitions 1

1.2.2. Acronyms 2

1.3. References 2

1.4. Overview 3

2. Overall Description 4

2.1. Product Perspective 4

2.1.1. System Interfaces 4

2.1.2. User Interface 4

2.1.3. Operation 7

2.2. Product Architecture 7

2.2.1. Software Architecture 7

2.2.2. Database Architecture 8

2.3. Product Functionality/Features 9

2.3.1. Add Events 10

2.3.2. Create Task 11

2.3.3. View Assigned Tasks 11

2.3.4. Add Contacts to the Directory 12

2.3.5. Add Attractions 12

2.3.6. Add Venues 12

2.3.7. Look-Up Contacts in the Directory 13

2.3.8. View a Calendar of Events 13

2.3.9. Manage the Budget for Events 14

2.3.10. Add System Users 14

2.3.11. Automated Back-Ups 14

2.4. Use Cases 15

2.4.1. User Logs On to the System. 15

2.4.2. User Adds an Event. 16

2.4.3. User Adds an Attraction. 17

2.4.4. User Adds a Venue. 18

2.4.5. User Adds a Contact. 19

2.4.6. User Adds a Manager. 20

2.4.7. User Creates a Task. 21

2.4.8. User Sets-Up Notifications. 21

2.4.9. Admin Adds a User Account. 22

2.4.10. Admin Schedules System Back-Ups. 23

2.4.11. User Updates Task Status. 23

2.4.12. Users Adds a Comment to a Task. 24

2.4.13. User Searches the Directory by a Last Name. 24

2.4.14. User Searches the Directory by Phone Number. 25

2.4.15. User Browses the Directory by First Letter of a Contact’s Last Name. 25

2.4.16. User Browses the Venues in the Directory 26

2.4.17. User Browses the Attractions in the Directory 26

2.5. User Characteristics 26

2.5.1. Employees 26

2.5.2. Non-Employees 27

2.6. Constraints 27

2.6.1. Type of Application 27

2.6.2. Platform Independence 27

2.6.3. Integration of Existing Web-sites 27

2.6.4. Accessibility 27

2.6.5. Database 27

2.6.6. Email 27

2.6.7. Reliability 27

2.7. Assumptions and Dependencies 28

2.7.1. Hosting 28

2.7.2. Storage 28

2.7.3. Usability 28

2.7.4. Maintenance 28

2.8. Apportioning of Requirements 28

3. Specific Requirements 29

3.1. Functional Requirements 29

3.1.1. The system shall maintain user accounts 29

3.1.2. The system shall maintain customized pages based on the user’s role. 29

3.1.3. The system shall maintain an administrator page 29

3.1.4. The system shall maintain a booking agent page 31

3.1.5. The system shall maintain a business user page. 33

3.1.6. The system shall maintain a financial user page. 34

3.1.7. The system shall provide a list of tasks to be worked on by the current user. 35

3.1.8. The system shall display a page specific to each event 38

3.1.9. The system shall display a page specific to each attraction. 39

3.1.10. The system shall display a page specific to each venue. 40

3.1.11. The system shall maintain a directory page for individuals. 42

3.1.12. The system shall maintain a directory of venues 44

3.1.13. The system shall maintain a directory of attractions 45

3.1.14. The system shall maintain a directory of events. 46

3.1.15. The system shall maintain a calendar of events. 48

3.1.16. The system shall provide an automatic notifications subscription page. 48

3.2. External Interface Requirements 49

3.2.1. The system shall interface with the user at a website. 49

3.2.2. The system shall store data in a database. 50

3.3. Internal Interface Requirements 50

3.3.1. The internal interface design shall be left to the programmer. 50

3.4. Internal Data Requirements 50

3.5. Design and Implementation Constraints 52

3.5.1. Standard Compliance 52

4. Non-Functional Requirements 53

4.1. Safety Requirements 53

4.1.1. The system shall not have any requirements regarding the physical safety of the users. 53

4.2. Security and Privacy Requirements 53

4.2.1. The system shall keep all user email addresses private and secure unless the user specifically requests that it be made public through another feature. 53

4.2.2. The system shall keep user login and password information private and secure. 53

4.2.3. The system shall not allow any user to access features that are specifically denied to them 53

4.2.4. The system shall use a secure HTTP connection when not being accessed by the public. 53

4.2.5. The system shall be kept updated to protect from all known security issues 53

4.2.6. The system shall keep unconfirmed event information unpublished and secure from leaking to the public. 53

4.2.7. The system shall publish confirmed events per the event organizers requests 53

4.2.8. The system should have a mechanism for users to reset their password if they forget it. 53

4.2.9. The system shall allow users to create their own accounts through a sign-up mechanism. 53

4.2.10. The system shall require users to enter their email address, full name, and a password at sign up. 53

4.2.11. The system shall keep a log of all activities. 53

4.3. Environmental Requirements 54

4.4. Computer Resource Requirements 54

4.4.1. Computer Hardware Requirements 54

4.4.2. Computer Hardware Resource Utilization Requirements 54

4.4.3. Computer Software Requirements 54

4.4.4. Computer Communication Requirements 55

4.5. Software Quality Factors 55

4.5.1. The system shall have a 99.9% uptime. 55

4.5.2. The system shall have scheduled monthly maintenance. 55

4.5.3. The system shall accurately store all information entered by the users. 55

4.5.4. The system shall reject any input that does not pass validation routines 55

4.5.5. The system shall not email notifications more often than the user specifies. 55

4.5.6. The system shall respond in less than two seconds for 95% of all transactions. 55

4.5.7. The system shall be available for operation 24 hours a day/7 days a week. 55

4.5.8. The system shall be easy enough to use for a new user to learn basic functions in under 5 minutes. 55

4.5.9. The system shall not become inoperative because of any input given by a user 55

4.6. Packaging Requirements 55

4.6.1. The system shall not have any packaging requirements. 55

4.7. Precedence and Criticality of Requirements 56

5. Qualification Provisions 57

5.1. Functional Requirements 57

5.2. Non-Functional Requirements 61

6. Requirements Traceability 63

6.1. Upward Traceability 63

6.1.1. Functional Requirements 63

6.1.2. Non-Functional Requirements 67

6.2. Downward Traceability 68

6.2.1. Functional Requirements 68

6.2.2. Non-Functional Requirements 71

# Purpose

This document is meant to delineate the features of Event Organization Console, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

## Scope

The scope of this document is to describe The Event Organization Console. The client needs a process management system that integrates the components of scheduling, communication, finance, and promotions associated with planning and executing an event in the Omaha area. The main purpose of the system is to maintain work queues for people associated with One Percent Productions for performing the specific tasks needed to carry out the execution of an event. This includes managing contacts, scheduling venues, negotiating contracts, acquiring security, accounting activities, and marketing events.

## Definitions, Acronyms, Abbreviations

The following is a list of commonly used definitions used throughout this document:

### Definitions

**Administrator**

An administrator is an employee of One Percent Productions who is internally classified as the technology expert responsible for the operation of IT systems.

**Attraction**

An attraction is the focus of an event is being held. For example, an attraction may be a band, a comedian or an activity such as a gaming tournament.

**Band Manager**

A band manager is the person who negotiates contracts, schedules performances and other non-music related business for the band.

**Booking Agent**

A booking agent is an employee of One Percent Productions who is responsible for the scheduling and management of an event.

**Event**

An event is most often a performance by a band at a venue. It may also be performances by comedians, video game tournaments or other entertainment attractions.

**External User**

An external user is any person that accesses the system who is not an employee of One Percent Productions, a venue manager or a venue owner.

**System**

A system is the Event Organization Console.

**Task**

A task is a unit of work that must be completed by an employee of One Percent Productions as part of putting on a successful event. Tasks are usually assigned to employees by booking agents.

**Work Queue**

A work queue is a collection of tasks assigned to an employee of One Percent Productions, ordered by priority which may be classified by deadline and criticality.

**Venue**

A venue is the physical location of an event. Usually places like night clubs, bars, convention centers and reception halls.

**Venue Manager**

A venue manager is a person responsible for the operation, scheduling and management of a venue.

**Venue Owner**

A venue owner is the person(s) who own a venue.

### Acronyms

No acronyms are used in this document.

## References

* IEEE Std 830, 1998 Edition (R2002) – *IEEE Recommended Practice for Software Requirements Specifications*
* *Spring 2008 Team Project Requirements* – Dr. Mansour Zand, January 2008
* *Concert Promotion Application – Rough Draft*, Marc Leibowitz

## Overview

The rest of the SRS is organized as follows:

**Section 2: Overall Description of the Event Organization Console.** This section contains information about the how the product fits into the day-to-day operation of One Percent Productions, a high level description of the system architecture as well as descriptions of the major functionality provided by the system.

**Section 3: Specific Requirements.** This section contains a list of the functional requirements of the system, how the system will interface with external systems as well as how subsystems of the Event Organization Console will interface.

**Section 4: Non-Functional Requirements.** This section includes a list a list of all non-functional requirements for the system as well as the priority and criticality of each non-functional requirement.

**Section 5: Qualification Provisions.** This section contains a table consisting of each software requirement (functional and non-functional) and what method will be used to verify that the system adheres to the requirement. Note that specifics are not given as to how each method will be carried out. For example, if requirement A will be verified using a test, the specifics of the test are not given.

**Section 6: Requirements Traceability.** This section gives forward and backward traceability matrices relating the requirements in this document to the requirements in the System Requirements Document.

# Overall Description

## Product Perspective

The Event Organization Console is a stand-alone Internet product intended to aid in the process of scheduling, confirming and managing events. The goal of the product is to streamline the process of taking an event from preliminary stages (“we may have an event on this day, at this location”) to its completion (“the band just walked off the stage”).

The Event Organization Console will replace the “Excel-Access-Ubunto” system of scheduling, confirming and managing events.

One Percent Productions does currently use a web-site to convey public information. The Event Organization Console must also have the capability of sharing information with the public.

### System Interfaces

The Event Organization Console will use an existing email server to send automated email to users of the system.

The Even Organization Console will be served by a web-server, either hosted by One Percent Productions or a third party.

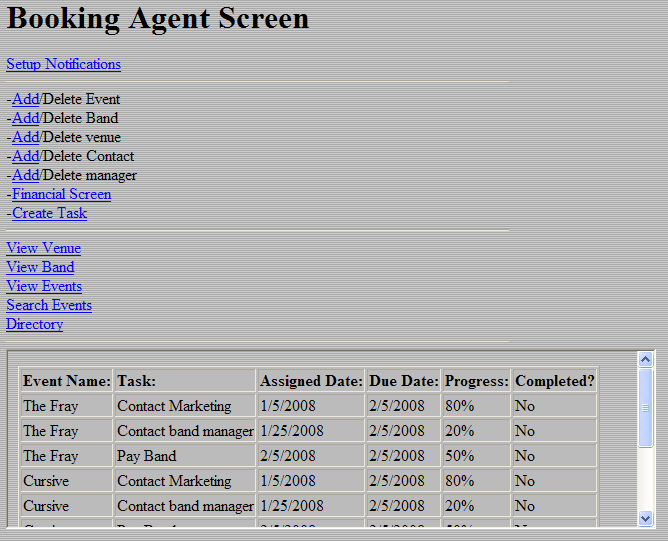
### User Interface

The users of the Event Organization Console will interface the system though a web-browser. As such, the interface should be designed in a way that is consistent across a variety of browsers including Internet Explorer and Fire Fox.

Different roles will have different main pages.

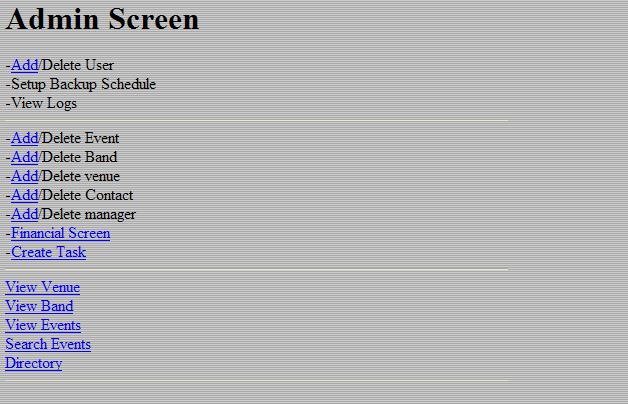
#### Booking Agent Main Page

From the booking agent’s main page, a booking agent will be able to access pages for adding/deleting events, bands, venues, contacts and managers. They will also be able to setup notifications, access financial data, create tasks, view venues, bands, events and search the directory.



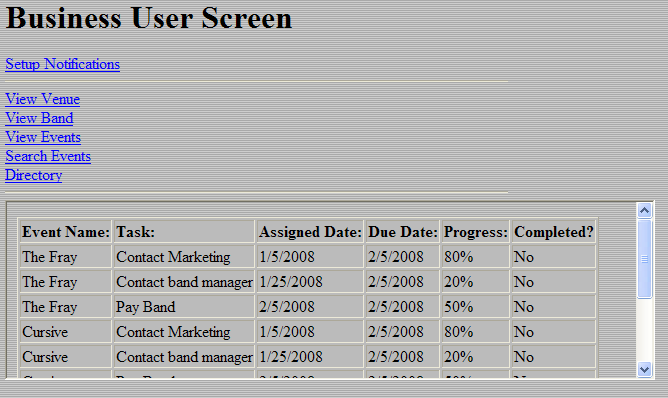
#### Administrator Main Page

From the administrator’s main page, a booking agent will be able to access pages for adding/deleting users, events, bands, venues, contacts and managers. They will also be able to schedule back-ups, view system logs, setup notifications, access financial data, create tasks, view venues, bands, events and search the directory.



#### Business User Main Page

From the business user’s main page, a business user will be able to access pages for adding/deleting users, events, bands, venues, contacts and managers. They will also be able to setup notifications, view venues, bands, events and search the directory.



#### Financial User Main Page

From the financial user’s main page, a financial user will be able to access pages for financial information related to events. They will also be able to setup notifications, view venues, bands, events and search the directory.

### Operation

Due to the nature of the music industry, The Event Organization Console may be used at all hours of the day and night. System back-ups are required and are best done between 4 and 7 a.m.

## Product Architecture

### Software Architecture

The Event Organization is a website that has components for managing events, contacts, and user task lists. Associated with an event is the ability to

1. Schedule the event
2. Record financial information associated with the event
3. Assign tasks to other system users

The calendar of confirmed and potential events should always be visible by users that are employees of One Percent Productions.

Schematically, the Event Organization Console has the following structure (all functionality may not be available to all users):

Main Screen

User Task List

Login

Add Event

Manage Event

Directory

Finance

Schedule Event

Assign Tasks

\* The User may logout from any component and return to the main screen

\* All data is written to and retrieved from a database

\* When tasks are assigned, they are emailed to the assignee as well as added to the appropriate task list

### Database Architecture

The Event Organization Console will use a database to store data and query date from. The database will have the following schema:

Event(Event\_Id, Name, Attraction\_List, Possible\_Venue\_List, Date\_Range, Scheduled\_Venue, Schedule\_Date)

Performer\_List(Attraction \_List\_Id, Attraction1\_Id,

Attraction2\_Id, Attraction3\_Id, Attraction4\_Id)

Possible\_Venue\_List(Venue\_List\_Id, Venue1\_Id,

Venue2\_Id, Venue3\_Id, Venue4\_Id)

Date\_Range(Date\_Range\_Id, Earliest\_Date, Latest\_Date)

Venue(Venue\_Id, Name, Owner, Manager, Street\_Address,

City, State, Zip, Phone1, Phone2, Fax)

Attraction(Attraction\_Id, Name, Manager)

Contact\_Directory(Contact\_Id, Name, Street\_Address,

City, State, Zip, Phone1, Phone2, Fax, Email)

Users(Username, Password, Fullname, Position,

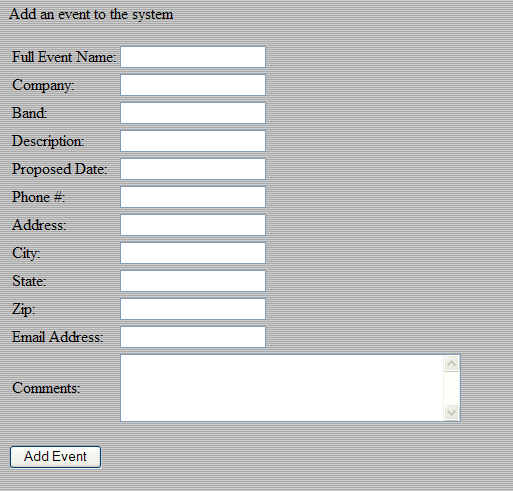
Access\_Level)

## Product Functionality/Features

The Event Organization Console will have each of the features below. Features will be available to users based on their profile as described below.

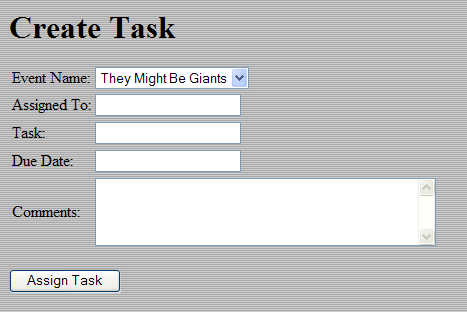
### Add Events

From the web-interface, users may add information corresponding to potential events. This information includes the name of the event, the performer(s), possible venues at which the event will be held and possible dates for the event.



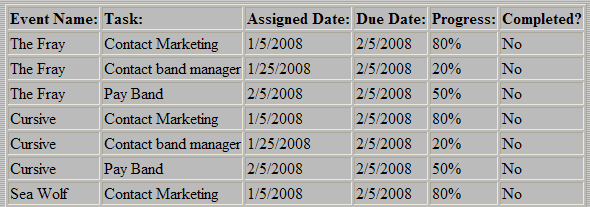
### Create Task

From the web-interface, users may assign tasks associated with events to other users of the Event Organization Console. Once a task is assigned to a user, she will be automatically notified and the task will be assigned to her task list.



### View Assigned Tasks

From the web-interface, users may view the list of tasks that have been assigned to them by other users.



### Add Contacts to the Directory

From the web-interface, users may add contacts to the directory. The contact information includes, name, title, phone number, cell-phone number, fax number, email address and mailing address.



### Add Attractions

From the web-interface, users may add attractions to the system. The attraction information includes, the name of the attraction, the name of the attractions manger/contact person, a description of the attraction and a picture of the attraction.

### Add Venues

From the web-interface, users may add venues to the system. The venue information includes, name, manager/owner name, phone number, cell-phone number, fax number, email address, mailing address, directions to the venue, a description of the venue and a picture of the venue.

### Look-Up Contacts in the Directory

From the web-interface, users may look-up contacts in the directory. Searches will be able to be done on name, location, phone number or title.

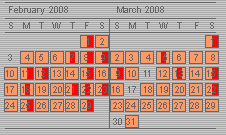


Users may also browse the directory alphabetically.



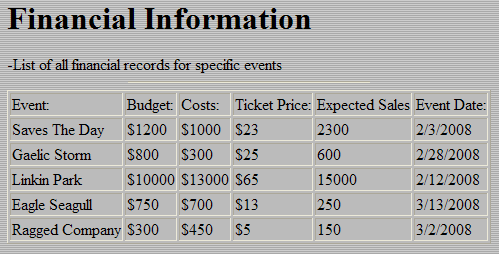
### View a Calendar of Events

From the web-interface, users may view a calendar of events. There will be two types of calendars – one for employees of One Percent Productions that contains potential and confirmed events and one for non-employees that only contains confirmed events.



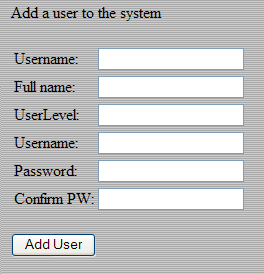
### Manage the Budget for Events

From the web-interface, users may manage the budget events. Users may allocate money to the budget items associated with events as well as record money actually spent on different budget items for the events.



### Add System Users

From the web-interface, the system admin may add users and specify their user type.



### Automated Back-Ups

The system will perform local back-ups nightly as well as off-site back-ups weekly. From the web-interface, the user may specify the time of each back-up or manually trigger a back-up to occur immediately.

## Use Cases

In the following uses cases an actor defined as “User” refers to an actor that could be any of the standard roles previously defined. “Admin” refers to an actor that is a system administrator.

### User Logs On to the System.

**Name:** UserLogon

**Actors:** Initiated by User

**Entry Conditions:** The One Percent Productions home page is displayed

**Flow of Events:**

1. User enters a username in the Username textbox.
2. User enters a password in the Password textbox.
3. User clicks Log On button.
4. The user’s main page is displayed.

**Exit Conditions:** The system’s main page associated with the current User is displayed.

**Exceptions:**

1. The system cannot find the specified username.
2. The password entered does not match the password found in the database.

### User Adds an Event.

**Name:** AddEvent

**Actors:** Initiated by User

**Entry Conditions:** The Add User page is displayed

**Flow of Events:**

1. User enters the name of the event in the Event Name textbox.
2. User enters the events attraction in the Attraction textbox.
3. User enters the earliest possible date in the Earliest Date textbox.
4. User enters the latest possible date in the Latest Date textbox.
5. User enters a venue name in the Possible Venue textbox.
6. User enters a description of the even in the Description textbox.
7. User clicks Add Event button.

**Exit Conditions:** The system’s main page associated with the current User is displayed.

**Exceptions:**

1. The system is unsuccessful in adding the event.

### User Adds an Attraction.

**Name:** AddAttraction

**Actors:** Initiated by User

**Entry Conditions:** The Add Attraction page is displayed.

**Flow of Events:**

1. User enters the name of the attraction in the Attraction Name text box.
2. User enters the name of the attractions manger in the Manager Name textbox.
3. User enters the mailing address of the attraction the Mailing Address text box.
4. User enters the city of the attraction in the City textbox.
5. User enters the state of the attraction in the State textbox.
6. User enters the zip code of the attraction in the Zip textbox.
7. User enters the phone number of the attraction in the Phone textbox.
8. User enters the email address of the attraction the Email textbox.
9. User enters a description of the attraction in the Description textbox.
10. User clicks the Add Attraction button.

**Exit Conditions:** The system’s main page associated with the current User is displayed.

**Exceptions:**

1. The system is unsuccessful in adding the event.

### User Adds a Venue.

**Name:** AddVenue

**Actors:** Initiated by User

**Entry Conditions:** The Add Venue page is displayed.

**Flow of Events:**

1. User enters the name of the venue in the Name textbox.
2. User enters the name of the venue’s manger in the Manager textbox.
3. User enters the name of the venue’s owner in the Owner textbox.
4. User enters the venue’s address in the Address textbox.
5. User enters the venue’s city in the City textbox.
6. User enters the venue’s state in the State textbox.
7. User enters a description of the venue in the description textbox.
8. User selects a picture to be uploaded using the Picture file chooser.
9. User clicks the Add Venue button.

**Exit Conditions:**  The system’s main page associated with the current User is displayed.

**Exceptions:**

1. The system is unsuccessful in adding the venue.

### User Adds a Contact.

**Name:** AddContact

**Actors:** Initiated by User

**Entry Conditions:** The Add Contact page is displayed.

**Flow of Events:**

1. User enters the name of the contact in the Name textbox.
2. User enters the contact’s address in the Address textbox.
3. User enters the contact’s city in the City textbox.
4. User enters the contact’s state in the State textbox.
5. User enters the contact’s zip code in the Zip textbox.
6. User enters the contact’s phone number in the Phone textbox.
7. User enters the contact’s cell phone number in the Cell textbox.
8. User enters the contact’s fax number in the Fax textbox.
9. User enters the contact’s email address in the Email textbox.
10. User clicks the Add Contact button.

**Exit Conditions:**  The system’s main page associated with the current User is displayed.

**Exceptions:**

1. The system is unsuccessful in adding the contact.

### User Adds a Manager.

**Name:** AddManager

**Actors:** Initiated by User

**Entry Conditions:** The Add Manager page is displayed.

**Flow of Events:**

1. User enters the manager’s name in the Name textbox.
2. User enters the name of the venue of attraction that she manages in the Client textbox.
3. User enters the manager’s address in the Address textbox.
4. User enters the manager’s city in the City textbox.
5. User enters the manager’s state in the State textbox.
6. User enters the manager’s zip code in the Zip textbox.
7. User enters the manager’s phone number in the Phone textbox.
8. User enters the manager’s cell phone number in the Cell textbox.
9. User enters the manager’s fax number in the Fax textbox.
10. User enters the manager’s email address in the Email textbox.
11. User clicks the Add Manager button.

**Exit Conditions:**  The system’s main page associated with the current User is displayed.

**Exceptions:**

1. The system is unsuccessful in adding the manager.

### User Creates a Task.

**Name:** CreateTask

**Actors:** Initiated by User

**Entry Conditions:** The Create Task page is displayed.

**Flow of Events:**

1. User selects the event the task is supporting from the Event drop down box.
2. User selects the person the task is assigned to from the Assigned To drop down box.
3. User enters the name of the task in the Task textbox.
4. User enters the date the task should be completed in the Due Date textbox.
5. User enters other information about the task in the Comments textbox.
6. User clicks the Assign Task button.

**Exit Conditions:**  The system’s main page associated with the current User is displayed.

**Exceptions:**

1. The system is unsuccessful in adding the task.

### User Sets-Up Notifications.

**Name:** SetupNotifications

**Actors:** Initiated by User

**Entry Conditions:** The Setup Notifications page is displayed.

**Flow of Events:**

1. User enters the time she would like to be notified each day what tasks have been assigned to her.
2. User enters the time she would like to be notified each day what tasks she has assigned have been completed by others.
3. User enters the time she would like to be notified each day of the status of all tasks she is associated with.
4. User clicks the Submit button.

**Exit Conditions:**  The system’s main page associated with the current User is displayed.

**Exceptions:**

1. The system is unsuccessful in adding the task.

### Admin Adds a User Account.

**Name:** AddUser

**Actors:** Initiated by Admin

**Entry Conditions:** The Add User page is displayed.

**Flow of Events:**

1. Admin enters the name of the person whose account she is adding in the Name textbox.
2. Admin enters the username of the person whose account she is adding in the Username textbox.
3. Admin enters the password of the person whose account she is adding in the Password textbox.
4. Admin selects the permission level of the person whose account she is adding from the Permission drop down box.
5. Admin clicks the Add User button.

**Exit Conditions:** The system’s main page associated with the current Admin is displayed.

**Exceptions:**

1. The entered username is already in the database.
2. The system is unsuccessful in adding the user.

### Admin Schedules System Back-Ups.

**Name:** ScheduleBackups

**Actors:** Initiated by Admin

**Entry Conditions:** The Schedule Backup page is displayed.

**Flow of Events:**

1. Admin selects the radio button corresponding day of the week that off-site back-ups are scheduled to be run.
2. Admin enters the time of day that weekly backups are scheduled to run in the Time textbox.
3. Admin selects the days of the week that local back-ups are scheduled.
4. Admin enters the time that local back-ups are scheduled.
5. Admin clicks the Schedule Backups button.

**Exit Conditions:** The system’s main page associated with the current Admin is displayed.

**Exceptions:**

1. Off-site and local back-ups are scheduled on the same day.
2. The system in unsuccessful in scheduling back-ups.

### User Updates Task Status.

**Name:** UpdateTaskStatus

**Actors:** Initiated by User

**Entry Conditions:** The page associated with a task is displayed.

**Flow of Events:**

1. User enters the level of completion in the %Done textbox.
2. User checks the Task Completed check box if the task is completed.
3. User clicks the Update Task Status button.

**Exit Conditions:** The system’s main page associated with the current User is displayed.

**Exceptions:**

1. The system is unsuccessful in updating the task status.

### Users Adds a Comment to a Task.

**Name:** AddTaskComment

**Actors:** Initiated by User

**Entry Conditions:** The page associated with a task is displayed.

**Flow of Events:**

1. User enters comments about the task in the Comment textbox.
2. User clicks Add Comment button.

**Exit Conditions:** The system’s main page associated with the current User is displayed.

**Exceptions:**

1. The system is unsuccessful in adding the comment to the task.

### User Searches the Directory by a Last Name.

**Name:** SearchDirectoryByName

**Actors:** Initiated by User

**Entry Conditions:** The Directory Search page is displayed.

**Flow of Events:**

1. User clicks the Name radio button.
2. User enters the last name of the person they are looking for in the Search textbox.
3. User clicks the Search button.

**Exit Conditions:** The system displays a page containing the results of the search.

**Exceptions:**

1. The system cannot find the entered name in the directory.

### User Searches the Directory by Phone Number.

**Name:** SearchDirectoryByPhoneNumber

**Actors:** Initiated by User

**Entry Conditions:** The Directory Search page is displayed.

**Flow of Events:**

1. User clicks the Phone radio button.
2. Users enters the phone number they are looking for in the Search textbox.
3. User clicks the Search button.

**Exit Conditions:** The system displays a page containing the row of the database in which the searched for phone number was found.

**Exceptions:**

1. The system cannot find the entered phone number in the directory.

### User Browses the Directory by First Letter of a Contact’s Last Name.

**Name:** BrowseDirectory

**Actors:** Initiated by User

**Entry Conditions:** The Browse Directory page is displayed.

**Flow of Events:**

1. User clicks on the letter of the alphabet she would like to browse.

**Exit Conditions:** System displays a page containing all of the contacts whose last name begins with the specified letter.

**Exceptions:**

1. There are no contacts in the directory whose last name begins with the specified letter.

### User Browses the Venues in the Directory

**Name:** BrowseVenues

**Actors:** Initiated by User

**Entry Conditions:** The Browse by Venue page is displayed.

**Flow of Events:**

1. User clicks on the letter of the alphabet she would like to browse.

**Exit Conditions:** System displays a page containing all of the venues whose name begins with the specified letter.

**Exceptions:**

1. There are no venues in the directory whose name begins with the specified letter.

### User Browses the Attractions in the Directory

**Name:** BrowseAttractions

**Actors:** Initiated by User

**Entry Conditions:** The Browse by Venue page is displayed.

**Flow of Events:**

1. User clicks on the letter of the alphabet she would like to browse.

**Exit Conditions:** System displays a page containing all of the attractions whose name begins with the specified letter.

**Exceptions:**

1. There are no attractions in the directory whose name begins with the specified letter.

## User Characteristics

The system will have two types of users – employees of One Percent Productions and users who are not employees of One Percent Productions.

### Employees

The employees of One Percent Productions will be the primary users of the Event Organization Console. We are assuming that employees of One Percent Productions have at least a high school education and that members of the marketing and accounting departments within One Percent Productions have bachelor’s degrees in their field.

Because One Percent Productions does a lot of work with the music industry, we are assuming that most of the employees will be ‘younger’ and are familiar with basic Internet usage and comfortable using mainstream applications and web browsers.

### Non-Employees

The Event Organization Console allows for users that are not employees of One Percent Productions. Since the Event Organization Console is a web application, any non-employee wishing to access the system should be able to navigate the Internet and use a web-browser.

## Constraints

### Type of Application

The Event Organization Console is a web application.

### Platform Independence

Users may need to access the Event Organization Console from a venue or other location using whatever computer is available, so the system shall be platform independent.

### Integration of Existing Web-sites

The Event Organization Console must integrate the websites www.onepercentprodutions.com and thewaitingroom.com.

### Accessibility

The Event Organization Console will be used by a variety of people that do not have a “need to know” some of the information of the system.

The Event Organization Console is not accessible without an Internet connection.

### Database

The Event Organization Console will include a database used to storing new data and retrieving queried data.

### Email

The Event Organization Console will email task lists and other communiqué using an existing email-server.

### Reliability

The Event Organization Console may be accessed 24 hours per day.

## Assumptions and Dependencies

### Hosting

We assume that One Percent Productions has access to a web-server sufficient for hosting the Event Organization Console.

### Storage

We assume that One Percent Productions has a computer with sufficient space to store the database required for the functionality of the Event Organization Console and that there is sufficient space for the database to grow.

### Usability

We assume that the employees of One Percent Productions are computer literate and familiar with basic Internet navigation.

### Maintenance

We assume that One Percent Productions has IT personal available to maintain the hardware necessary to use the Event Organization Console as well as perform upgrades to the Event Organization Console as they become available.

## Apportioning of Requirements

The Event Organization Console will be released with full functionality adhering to all requirements. Updates will be provided as necessary.

# Specific Requirements

## Functional Requirements

Specify required behavior and include parameters such as response times, throughput times, other timing constraints, sequencing, accuracy, capacities, priorities, etc.

### The system shall maintain user accounts

#### The system shall allow users to login to their accounts.

#### The system shall maintain permissions for what actions a user is permitted to perform.

### The system shall maintain customized pages based on the user’s role.

### The system shall maintain an administrator page

#### The system shall allow administrators to login to their accounts by supplying a valid username and password for an account that has been assigned an administrator role.

* Inputs: The user provides a username and password
* Outputs: Upon failure, the system will display a login failure indicating that an incorrect password or non-existent username has been entered. Otherwise, the main administrator page is displayed.
* System Behavior: The system searches the database for the supplied username and validates the password against the one supplied. If the passwords match, the system will check that the user is assigned to a administrator role. If the user is a administrator, the main administrator page will be displayed.



#### The system shall allow the user to add/delete events from the administrator page.

#### The system shall allow the user to add/delete attractions from the administrator page.

#### The system shall allow the user to add/delete venues from the administrator page.

#### The system shall allow the user to add/delete contacts from the administrator page.

#### The system shall allow the user to add/delete managers from the administrator page.

#### The system shall allow the user to access financial information from the administrator page.

#### The system shall allow the user to create tasks from the administrator page.

#### The system shall allow the user to access the directory from the administrator page.

#### The system shall display the tasks assigned to the current user from the administrator page.

#### The system shall allow the user to setup notifications from the administrator page.

#### The system shall allow the user to add/delete user accounts from the administrator page.

* Inputs: The user provides information about the user such as the username, password, confirmed password, full name, and role/permissions.
* Outputs: Upon failure, the system will display an appropriate, human readable error message. If successful, the data about the user will be persisted to the database and the system will return to their main page.
* System Behavior: The system first checks the permissions of the user to determine if they have the appropriate access to create users. If authorized, the “add user” interface will be displayed. The user must then populate the “add user” form and submit it for processing. The data for the new user is then written to the database and the user is returned back to their main page.



#### The system shall allow the user to setup system backup schedules from the administrator page.

#### The system shall allow the user to view logs from the administrator page.

### The system shall maintain a booking agent page

#### The system shall allow booking agents to login to their accounts by supplying a valid username and password for an account that has been assigned the booking agent role

* Inputs: The user provides a username and password
* Outputs: Upon failure, the system will display a login failure indicating that an incorrect password or non-existent username has been entered. Otherwise, the main booking agent page is displayed.
* System Behavior: The system searches the database for the supplied username and validates the password against the one supplied. If the passwords match, the system will check that the user is assigned to a booking agent role. If the user is a booking agent, the main booking agent page will be displayed.



#### The system shall allow the user to add/delete events from the booking agent page.

#### The system shall allow the user to add/delete attractions from the booking agent page.

#### The system shall allow the user to add/delete venues from the booking agent page.

#### The system shall allow the user to add/delete contacts from the booking agent page.

#### The system shall allow the user to add/delete managers from the booking agent page.

#### The system shall allow the user to access financial information from the booking agent page.

#### The system shall allow the user to create tasks from the booking agent page.

#### The system shall allow the user to access the directory from the booking agent page.

#### The system shall display the tasks assigned to the current user from the booking agent page.

#### The system shall allow the user to setup notifications from the booking agent page.

### The system shall maintain a business user page.

#### The system shall allow business users to login to their accounts by supplying a valid username and password for an account that has been assigned a business user role

* Inputs: The user provides a username and password
* Outputs: Upon failure, the system will display a login failure indicating that an incorrect password or non-existent username has been entered. Otherwise, the main business user page is displayed.
* System Behavior: The system searches the database for the supplied username and validates the password against the one supplied. If the passwords match, the system will check that the user is assigned to a business user role. If the user is a business user, the main business user page will be displayed.



#### The system shall allow the user to access the directory from the business user page.

#### The system shall display the tasks assigned to the current user from the business user page.

#### The system shall allow the user to setup notifications from the business user page.

### The system shall maintain a financial user page.

#### The system shall allow financial users to login to their accounts by supplying a valid username and password for an account that has been assigned a financial user role

* Inputs: The user provides a username and password
* Outputs: Upon failure, the system will display a login failure indicating that an incorrect password or non-existent username has been entered. Otherwise, the main financial user page is displayed.
* System Behavior: The system searches the database for the supplied username and validates the password against the one supplied. If the passwords match, the system will check that the user is assigned to a financial user role. If the user is a financial user, the main financial user page will be displayed.



#### The system shall allow the user to access the directory from the financial user page.

#### The system shall display the tasks assigned to the current user from the financial user page.

#### The system shall allow the user to access financial information from the financial user page.

#### The system shall allow the user to setup notifications from the financial user page.

### The system shall provide a list of tasks to be worked on by the current user.

#### The system shall allow the user to create tasks.

* Inputs: The user provides information about the task such as the event name, who it is being assigned to, a description of the task, a due date, and comments.
* Outputs: Upon failure, the system will display an appropriate, human readable error message. If successful, the data about the task will be persisted to the database and the system will open the task page for the newly created task.
* System Behavior: The system first checks the permissions of the user to determine if they have the appropriate access to create tasks. If authorized, the “add task” interface will be displayed. The user must then populate the “add task” form and submit it for processing. The data for the new task is then written to the database and a corresponding task page is generated to which the user is then forwarded.



#### The system shall allow the user to modify tasks.

* Inputs: The user provides the updated information about a task such as the percent complete, status, additional comments, and/or the person to which to delegate the task.
* Outputs: Upon failure, the system will display an appropriate, human readable error message. If successful, the data about the updated task will be persisted to the database and the system will open the task page with the updated data.
* System Behavior: The system first checks the permissions of the user to determine if they have the appropriate access to modify tasks. If authorized, the database is queried to load the data about the task and the editable task interface is displayed. The user must then edit the data in the page and submit it for processing. The data for the updated task is then written to the database and the user is forwarded to the updated task page.

#### 

#### The system shall allow the user to add comments to a task.

#### The system shall allow the user to assign/delegate tasks to other users.

#### The system shall order the tasks in the task list by priority/criticality.

#### The system shall display all the tasks for a specific event on the corresponding event’s page.

#### The system shall display the name of each task in the list.

#### The system shall display the assignment date of each task in the list.

#### The system shall display the due date of each task in the list.

#### The system shall display the percent of progress for each task in the list.

#### The system shall display the current status of each task in the list.

### The system shall display a page specific to each event

#### The system shall allow the user to add events.

* Inputs: The user provides information about the event such as the name, company, attraction, description, date, venue, comments.
* Outputs: Upon failure, the system will display an appropriate, human readable error message. If successful, the data about the event will be persisted to the database and the system will open the event page for the newly created event.
* System Behavior: The system first checks the permissions of the user to determine if they have the appropriate access to create events. If authorized, add event interface will be displayed. The user must then populate the add event form and submit it for processing. The data for the new event is then written to the database and a corresponding event page is generated to which the user is forwarded.



#### The system shall display the name of the attraction on the event’s page.

#### The system shall provide a description of the attraction on the event’s page.

#### The system shall provide a picture of the attraction on the event’s page.

#### The system shall display the name of the venue on the event’s page.

#### The system shall provide a link to the venue’s page from the event’s page.

#### The system shall display the name of the attraction on the event’s page.

#### The system shall display the date(s) of the event on the event’s page.

#### The system shall display the ticket price(s) for the event on the event’s page.

#### The system shall provide a link to purchase tickets on the event’s page.

#### The system shall provide a list of all tasks associated with the event on the event’s page if the user is an administrator, booking agent, business user, financial user, or the venue’s owner/manager.

### The system shall display a page specific to each attraction.

#### The system shall provide the ability to add new attractions.

* Inputs: The user provides information about the attraction such as the attraction name, a picture of the attraction, a description, the type of attraction, and manager/agent contacts.
* Outputs: Upon failure, the system will display an appropriate, human readable error message. If successful, the data about the attraction will be persisted to the database and the system will open the attraction page for the newly created attraction.
* System Behavior: The system first checks the permissions of the user to determine if they have the appropriate access to create attractions. If authorized, the “add attraction” interface will be displayed. The user must then populate the “add attraction” form and submit it for processing. The data for the new attraction is then written to the database and a corresponding attraction page is generated to which the user is then forwarded.



#### The system shall display the name of the attraction on the attraction’s page.

#### The system shall display the name of the attraction on the attraction’s page.

#### The system shall display the name of the attraction’s manager/agent on the attraction’s page.

#### The system shall display a picture of the attraction on the attraction’s page.

#### The system shall display a description of the attraction on the attraction’s page.

#### The system shall display a list of all events associated with the attraction on the attraction’s page if the user is an administrator, booking agent, business user, financial user, or the attraction’s manager/agent.

#### The system shall display a list of confirmed events associated with the attraction on the attraction’s page if the user is a general user.

### The system shall display a page specific to each venue.

#### The system shall allow the user to add new venues

* Inputs: The user provides information about the venue such as the venue name, address, map, phone number, e-mail, manager/owner, description, and picture.
* Outputs: Upon failure, the system will display an appropriate, human readable error message. If successful, the data about the venue will be persisted to the database and the system will open the venue page for the newly created venue.
* System Behavior: The system first checks the permissions of the user to determine if they have the appropriate access to create venues. If authorized, the “add venue” interface will be displayed. The user must then populate the “add venue” form and submit it for processing. The data for the new venue is then written to the database and a corresponding venue page is generated to which the user is then forwarded.



#### The system shall display the address for a venue on the venue’s page.

#### The system shall display a map to the venue on the venue’s page.

#### The system shall display the owner and manager of a venue on the venue’s page if the user is an administrator, booking agent, business user, financial user, or the venue’s owner/manager.

#### The system shall display the phone number of a venue on the venue’s page.

#### The system shall display all the events scheduled at a venue on the venue’s page if the user is an administrator, booking agent, business user, or financial user.

#### The system shall display all the confirmed events scheduled at a venue on the venue’s page if the user is a general user.

#### The system shall display the owner and manager of a venue on the venue’s page.

#### The system shall display a picture of the venue on the venue’s page.

#### The system shall display the owner and manager of a venue on the venue’s page.

#### The system shall display a description of the venue on the venue’s page.

### The system shall maintain a directory page for individuals.

#### The system shall provide the ability to search the directory for an individual.

* Inputs: The user provides information about an individual by which the search will be performed such as their name, phone number, username, or associated attraction/venue.
* Outputs: Upon failure, the system will display an appropriate, human readable error message. If successful, the data about the individual is displayed.
* System Behavior: The system first checks the permissions of the user to determine if they have the appropriate access to lookup individuals in the directory. If authorized, individual search page is displayed to the user. The user then enters search criteria. The database is then searched against the given criteria. If no data is found, an error message stating that the individual does not exist is displayed to the user. Otherwise, the list of all individuals satisfying the search criteria is displayed. The user may then select an individual from the list to view the details about him/her.



#### The system shall provide the ability to search the directory for an individual by phone number.

#### The system shall provide the ability to browse for individuals in the directory by the first character of their last name.

#### The system shall provide separate pages for individuals indexed on the first character of their last name.

#### The system shall provide an alphabetically sorted list of names on each individual directory page.

#### The system shall display in the directory the individual’s name, phone number, cell-phone number, fax number, work number, mailing address, e-mail address, role/title, employer, department, and available hours.

#### The system shall display individual directory data in a table with column headings.

#### The system shall store the name of each individual in the directory.

#### The system shall store the phone number of each individual in the directory.

#### The system shall store the cell-phone number of each individual in the directory.

#### The system shall store the fax number of each individual in the directory.

#### The system shall store the work number of each individual in the directory.

#### The system shall store the mailing address of each individual in the directory.

#### The system shall store the e-mail address of each individual in the directory.

#### The system shall store the role/title of each individual in the directory.

#### The system shall store the employer of each individual in the directory.

#### The system shall store the department of each individual in the directory.

#### The system shall store the available hours of each individual in the directory.

### The system shall maintain a directory of venues

#### The system shall provide the ability to search the directory for a venue.

* Inputs: The user provides information about a venue by which the search will be performed such as its name, phone number, location, or associated attraction.
* Outputs: Upon failure, the system will display an appropriate, human readable error message. If successful, the data about the venue is displayed.
* System Behavior: All users should have permission to search for venues, so no authentication is necessary. Upon request, the venue search page is displayed to the user. The user then enters the search criteria and submits it. The database is then searched against the given criteria. If no data is found, an error message stating that the venue does not exist is displayed to the user. Otherwise, the list of all venues satisfying the search criteria is displayed. The user may then select a venue from the list to be forwarded to the venue’s page.



#### The system shall provide the ability to browse for venues in the directory by the first character of their name.

#### The system shall provide separate pages for venues indexed on the first character of their name.

#### The system shall provide an alphabetically sorted list of names on each venue directory page.

#### The system shall provide a link to each venue page from the venue directory.

### The system shall maintain a directory of attractions

#### The system shall provide the ability to search the directory for an attraction.

* Inputs: The user provides information about an attraction by which the search will be performed such as its name, attraction type, or venue at which they will perform.
* Outputs: Upon failure, the system will display an appropriate, human readable error message. If successful, the data about the attaction is displayed.
* System Behavior: All users should have permission to search for attractions, so no authentication is necessary. Upon request, the attraction search page is displayed to the user. The user then enters the search criteria and submits it. The database is then searched against the given criteria. If no data is found, an error message stating that the attraction does not exist is displayed to the user. Otherwise, the list of all attractions satisfying the search criteria is displayed. The user may then select an attraction from the list to be forwarded to the attraction’s page.



#### The system shall provide the ability to browse for attractions in the directory by the first character of their name.

#### The system shall provide separate pages for attractions indexed on the first character of their name.

#### The system shall provide an alphabetically sorted list of names on each attraction directory page.

#### The system shall provide a link to each attraction page from the attraction directory.

### The system shall maintain a directory of events.

#### The system shall allow the user to add events.

* Inputs: The user provides information about the event such as the name, company, attraction, description, date, venue, comments.
* Outputs: Upon failure, the system will display an appropriate, human readable error message. If successful, the data about the event will be persisted to the database and the system will open the event page for the newly created event.
* System Behavior: The system first checks the permissions of the user to determine if they have the appropriate access to create events. If authorized, add event interface will be displayed. The user must then populate the add event form and submit it for processing. The data for the new event is then written to the database and a corresponding event page is generated to which the user is forwarded.



#### The system shall provide the ability to search the directory for an event by name.

#### The system shall provide the ability to search the directory for an event by date.

#### The system shall provide the ability to search the directory for an event by venue.

#### The system shall provide the ability to search the directory for an event by attraction.

#### The system shall display search results for the event directory in a table with column headings.

#### The system shall provide a link to the event page for each event returned in the directory search results.

#### The system shall display the event name, attraction, venue, date, and status in the directory search results.

### The system shall maintain a calendar of events.

#### The system shall display the current and next month calendars.

#### The system shall display all events on the calendar if the user is an administrator, booking agent, business user, or financial user.

#### The system shall display confirmed events on the calendar if the user is a general user.

#### The system shall display the calendar on all pages so that it is always visible.

#### The system shall color code the calendar based on the types of scheduled events.

#### The system shall provide the user the ability to navigate to different months in the calendar.

#### The system shall provide a link from the calendar to the corresponding event page.

### The system shall provide an automatic notifications subscription page.

#### The system shall provide the user the ability to choose when or if they receive notifications when a task is assigned to them.

* Inputs: The user provides the updated subscription configuration for event notifications such as when/if to receive notifications when a task is assigned or closed, and when/if to receive a summary notification of all tasks assigned to them.
* Outputs: Upon failure, the system will display an appropriate, human readable error message. If successful, the data about the updated notification subscription configuration will be persisted to the database and the system will return to the user’s main page.
* System Behavior: The system first checks the permissions of the user to determine if they have the appropriate access to modify notifications. If authorized, the database is queried to load the notification subscription configuration and the editable notification interface is displayed. The user must then edit the data in the page and submit it for processing. The data for the updated notification subscription configuration is then written to the database and the user is forwarded to their main page.



#### The system shall provide the user the ability to choose when or if they receive notifications when a task that they are assigned is updated.

#### The system shall provide the user the ability to choose when or if they receive a notification summarizing all their currently assign tasks.

#### The system shall deliver notifications in accordance to the user’s subscription.

## External Interface Requirements

### The system shall interface with the user at a website.

#### The system shall use the same fonts as the existing onepercentproductions.com web site.

#### The system shall use the same backgrounds as the existing onepercentproductions.com web site.

#### The system shall use the same color schemes as the existing onepercentproductions.com web site.

#### The system shall use the same style sheets as the existing onepercentproductions.com web site.

### The system shall store data in a database.

## Internal Interface Requirements

### The internal interface design shall be left to the programmer.

## Internal Data Requirements

The Event table contains information about events managed by the system. The primary key will be an auto generated number called Event\_ID. Many of the other tables are supporting tables of the Event Table.

|  |  |
| --- | --- |
| **Event Table** | |
| Event\_ID | AutoNumber |
| Name | char(50) |
| Attraction\_List\_ID | integer |
| Possible\_Venue\_List | integer |
| Date\_Range\_ID | integer |
| Scheduled\_Venue | char(50) |
| Scheduled\_Date | date |

The Attraction\_List table contains lists of attractions associated with events in the Event table. Its key is an auto generated number.

|  |  |
| --- | --- |
| **Attraction\_List Table** | |
| Attraction\_List\_ID | AutoNumber |
| Attraction1\_ID | integer |
| Attraction2\_ID | integer |
| Attraction3\_ID | integer |
| Attraction4\_ID | integer |

The Possible\_Venue\_List table contains lists of venues at which an event may be put-on. Its key is an auto generated number.

|  |  |
| --- | --- |
| **Possible\_Venue\_List Table** | |
| Possible\_Venue\_List\_ID | AutoNumber |
| PossibleVenue1\_ID | char(50) |
| PossibleVenue2\_ID | integer |
| PossibleVenue3\_ID | integer |
| PossibleVenue4\_ID | integer |

The Date\_Range table stores the range of dates that a given event may be held on. Its key is an auto generated number.

|  |  |
| --- | --- |
| **Date\_Range Table** | |
| Date\_Range\_ID | AutoNumber |
| Earliest\_Date | integer |
| Latest\_Date | integer |

The Venue table stores the information related to specific venues. It is referred to by both the Event table and the Possible\_Venue\_List table. Its key is an auto generated number.

|  |  |
| --- | --- |
| **Venue Table** | |
| Venue\_ID | AutoNumber |
| Name | char(50) |
| Owner | char(50) |
| Manger | char(50) |
| Owner\_Contact\_ID | integer |
| Manger\_Contact\_ID | integer |
| Street\_Address | char(50) |
| City | char(50) |
| State | char(2) |
| Zip | integer |
| Phone1 | phone |
| Phone2 | phone |
| Fax | phone |

The Attraction table stores information related to specific attractions. It is referred to by the Attraction\_List table and references the contacts table. Its key is an auto generated number.

|  |  |
| --- | --- |
| **Attraction Table** | |
| Attraction\_ID | AutoNumber |
| Name | char(50) |
| Manager\_Name | char(50) |
| Manager\_Contact\_ID | integer |

The Contact table stores information related to contacts the employees of One Percent Productions have made. It is referenced by the Attraction table and the venue table. Its key is an auto generated number.

|  |  |
| --- | --- |
| **Contact Table** | |
| Venue\_ID | AutoNumber |
| Name | char(50) |
| Role | char(50) |
| Street\_Address | char(50) |
| City | char(50) |
| State | char(2) |
| Zip | integer |
| Phone1 | phone |
| Phone2 | phone |
| Fax | phone |
| Email | char(50) |

The Users table contains information about users of the system. Most importantly, it stores the username and password associated with each user of the system. Its key is an auto generated number.

|  |  |
| --- | --- |
| **Users Table** | |
| User\_ID | AutoNumber |
| Name | char(50) |
| Username | char(50) |
| Password | char(50) |
| Position | char(50) |
| Permission\_Level | Permission |

## Design and Implementation Constraints

### Standard Compliance

One Percent Productions personnel will be responsible for writing organizational policy for usage of the system and for posting the policy.

# Non-Functional Requirements

## Safety Requirements

### The system shall not have any requirements regarding the physical safety of the users.

## Security and Privacy Requirements

### The system shall keep all user email addresses private and secure unless the user specifically requests that it be made public through another feature.

### The system shall keep user login and password information private and secure.

### The system shall not allow any user to access features that are specifically denied to them

### The system shall use a secure HTTP connection when not being accessed by the public.

### The system shall be kept updated to protect from all known security issues

### The system shall keep unconfirmed event information unpublished and secure from leaking to the public.

### The system shall publish confirmed events per the event organizers requests

### The system should have a mechanism for users to reset their password if they forget it.

### The system shall allow users to create their own accounts through a sign-up mechanism.

### The system shall require users to enter their email address, full name, and a password at sign up.

### The system shall keep a log of all activities.

## Environmental Requirements

The system shall be able to be hosted on a server running Windows 2003 operating system or newer. The system shall be accessed from any computer running a standard web browser which has network access to the EOC server.

## Computer Resource Requirements

### Computer Hardware Requirements

#### The system shall require a database server

#### The system shall require a web server

#### The system shall require an application server.

#### The system may use the same computer for use as a database, application and web server.

#### The system shall require a persistent internet connection.

#### The system shall have an uninterrupted power supply

### Computer Hardware Resource Utilization Requirements

#### The system shall not perform in such a way as to completely consume computing resources.

#### The system shall not over-saturate the available bandwidth.

### Computer Software Requirements

#### The system shall not be dependent on the type of operating system used on the server.

#### The system shall be accessible from all modern graphical web browsers such as Internet Explorer, Firefox, Opera, and Safari.

#### The system shall run backup jobs at intervals specified by the administrator

### Computer Communication Requirements

#### The system shall communicate with the outside world through the use of the internet

#### The database, application and web servers shall communication with each other over a Local Area Network

#### The system shall communicate notifications to the users through email.

## Software Quality Factors

### The system shall have a 99.9% uptime.

### The system shall have scheduled monthly maintenance.

### The system shall accurately store all information entered by the users.

### The system shall reject any input that does not pass validation routines

### The system shall not email notifications more often than the user specifies.

### The system shall respond in less than two seconds for 95% of all transactions.

### The system shall be available for operation 24 hours a day/7 days a week.

### The system shall be easy enough to use for a new user to learn basic functions in under 5 minutes.

### The system shall not become inoperative because of any input given by a user

## Packaging Requirements

### The system shall not have any packaging requirements.

## Precedence and Criticality of Requirements

The requirements listed above shall be prioritized according to the following table:

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority** | **Critical** |
| 4.2.1 | High | Yes |
| 4.2.2 | High | Yes |
| 4.2.3 | High | Yes |
| 4.2.4 | High | Yes |
| 4.2.5 | High | Yes |
| 4.2.6. | High | Yes |
| 4.2.7 | High | Yes |
| 4.2.11 | High | Yes |
| 4.4.1.4 | High | Yes |
| 4.4.1.5 | High | Yes |
| 4.4.1.6 | High | Yes |
| 4.4.4.1 | High | Yes |
| 4.4.4.2 | High | Yes |
| 4.4.4.3 | High | Yes |
| 4.5.1 | High | Yes |
| 4.5.3 | High | Yes |
| 4.5.4 | High | Yes |
| 4.5.6 | High | Yes |
| 4.5.7 | High | Yes |
| 4.5.8 | High | Yes |
| 4.5.9 | High | Yes |
| 4.2.8 | Medium | No |
| 4.2.9 | Medium | No |
| 4.4.1.1 | Medium | No |
| 4.4.1.2 | Medium | No |
| 4.4.2.1 | Medium | No |
| 4.4.2.2 | Medium | No |
| 4.4.3.1 | Medium | No |
| 4.4.3.2 | Medium | No |
| 4.4.3.3 | Medium | No |
| 4.5.2 | Medium | No |
| 4.5.5 | Medium | No |
| 4.1.1 | Low | No |
| 4.3.1 | Low | No |
| 4.6.1 | Low | No |

# Qualification Provisions

This section defines a set of qualification methods. For each requirement in Section 3, specify the methods to be used to ensure that the requirement has been met. A table may be used to present this information, or each requirement may be annotated with the method(s) to be used. Qualification methods may include: Demonstration, Test, Analysis, Inspection, Special.

The following are descriptions of qualification methods that may be used on a requirement.

**Test** – The system will be subjected to various tests to see if the requirement has been met or not.

**Demonstration** – The system will be demonstrated to the stake-holder to show that the requirement has been met.

**Analysis** –The system design will be analyzed to see that the requirement has been met

**Inspection** –The system’s code will be inspected to see that it is correct and meets the requirement.

**Special** – Special provisions will be taken to meet the requirement.

**None** – The requirement needs no testing.

## Functional Requirements

\*If the testing method needs clarification it will be listed in the Notes column.

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Qualification Method** | **Notes** |
| 3.1.1 | Analysis |  |
| 3.1.1.1 | Test | Create a set of users of different types and log each in. |
| 3.1.1.2 | Demonstration |  |
| 3.1.2 | Demonstration |  |
| 3.1.3 | Demonstration |  |
| 3.1.3.1 | Test |  |
| 3.1.3.2 | Test |  |
| 3.1.3.3 | Test |  |
| 3.1.3.4 | Test |  |
| 3.1.3.5 | Test |  |
| 3.1.3.6 | Demonstration |  |
| 3.1.3.7 | Test |  |
| 3.1.3.8 | Demonstration |  |
| 3.1.3.9 | Demonstration |  |
| 3.1.3.10 | Test |  |
| 3.1.3.11 | Test |  |
| 3.1.3.12 | Test |  |

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Qualification Method** | **Notes** |
| 3.1.3.13 | Demonstration |  |
| 3.1.4 | Demonstration |  |
| 3.1.4.1 | Test |  |
| 3.1.4.2 | Test |  |
| 3.1.4.3 | Test |  |
| 3.1.4.4 | Test |  |
| 3.1.4.5 | Test |  |
| 3.1.4.6 | Demonstration |  |
| 3.1.4.7 | Test |  |
| 3.1.4.8 | Demonstration |  |
| 3.1.4.9 | Demonstration |  |
| 3.1.4.10 | Test |  |
| 3.1.5 | Demonstration |  |
| 3.1.5.1 | Demonstration |  |
| 3.1.5.2 | Demonstration |  |
| 3.1.5.3 | Test |  |
| 3.1.6 | Demonstration |  |
| 3.1.6.1 | Demonstration |  |
| 3.1.6.2 | Demonstration |  |
| 3.1.6.3 | Demonstration |  |
| 3.1.6.4 | Test |  |
| 3.1.7 | Demonstration |  |
| 3.1.7.1 | Test |  |
| 3.1.7.2 | Test |  |
| 3.1.7.3 | Test |  |
| 3.1.7.4 | Test |  |
| 3.1.7.5 | Demonstration |  |
| 3.1.7.6 | Demonstration |  |
| 3.1.7.7 | Demonstration |  |
| 3.1.7.8 | Demonstration |  |
| 3.1.7.9 | Demonstration |  |
| 3.1.7.10 | Demonstration |  |
| 3.1.8 | Demonstration |  |
| 3.1.8.1 | Demonstration |  |
| 3.1.8.2 | Demonstration |  |
| 3.1.8.3 | Demonstration |  |
| 3.1.8.4 | Demonstration |  |
| 3.1.8.5 | Demonstration |  |
| 3.1.8.6 | Demonstration |  |
| 3.1.8.7 | Demonstration |  |
| 3.1.8.8 | Demonstration |  |

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Qualification Method** | **Notes** |
| 3.1.8.9 | Demonstration |  |
| 3.1.8.10 | Test |  |
| 3.1.9 | Demonstration |  |
| 3.1.9.1 | Demonstration |  |
| 3.1.9.2 | Demonstration |  |
| 3.1.9.3 | Demonstration |  |
| 3.1.9.4 | Demonstration |  |
| 3.1.9.5 | Demonstration |  |
| 3.1.9.6 | Test |  |
| 3.1.9.7 | Test |  |
| 3.1.10 | Demonstration |  |
| 3.1.10.1 | Demonstration |  |
| 3.1.10.2 | Demonstration |  |
| 3.1.10.3 | Test |  |
| 3.1.10.4 | Demonstration |  |
| 3.1.10.5 | Test |  |
| 3.1.10.6 | Test |  |
| 3.1.10.7 | Demonstration |  |
| 3.1.10.8 | Demonstration |  |
| 3.1.10.9 | Demonstration |  |
| 3.1.10.10 | Demonstration |  |
| 3.1.11 | Demonstration |  |
| 3.1.11.1 | Test |  |
| 3.1.11.2 | Test |  |
| 3.1.11.3 | Test |  |
| 3.1.11.4 | Test |  |
| 3.1.11.5 | Test |  |
| 3.1.11.6 | Demonstration |  |
| 3.1.11.7 | Demonstration |  |
| 3.1.11.8 | Analysis |  |
| 3.1.11.9 | Analysis |  |
| 3.1.11.10 | Analysis |  |
| 3.1.11.11 | Analysis |  |
| 3.1.11.11 | Analysis |  |
| 3.1.11.12 | Analysis |  |
| 3.1.11.13 | Analysis |  |
| 3.1.11.14 | Analysis |  |
| 3.1.11.15 | Analysis |  |
| 3.1.11.16 | Analysis |  |
| 3.1.11.17 | Analysis |  |
| 3.1.11.18 | Analysis |  |

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Qualification Method** | **Notes** |
| 3.1.12 | Demonstration |  |
| 3.1.12.1 | Test |  |
| 3.1.12.2 | Test |  |
| 3.1.12.3 | Test |  |
| 3.1.12.4 | Test |  |
| 3.1.12.5 | Demonstration |  |
| 3.1.13 | Demonstration |  |
| 3.1.13.1 | Test |  |
| 3.1.13.2 | Test |  |
| 3.1.13.3 | Test |  |
| 3.1.13.4 | Test |  |
| 3.1.13.5 | Demonstration |  |
| 3.1.14 | Demonstration |  |
| 3.1.14.1 | Test |  |
| 3.1.14.2 | Test |  |
| 3.1.14.3 | Test |  |
| 3.1.14.4 | Test |  |
| 3.1.14.5 | Demonstration |  |
| 3.1.14.6 | Demonstration |  |
| 3.1.14.7 | Demonstration |  |
| 3.1.15 | Demonstration |  |
| 3.1.15.1 | Demonstration |  |
| 3.1.15.2 | Test |  |
| 3.1.15.3 | Test |  |
| 3.1.15.4 | Test |  |
| 3.1.15.5 | Test |  |
| 3.1.15.6 | Demonstration |  |
| 3.1.15.7 | Demonstration |  |
| 3.1.16 | Demonstration |  |
| 3.1.16.1 | Test |  |
| 3.1.16.2 | Test |  |
| 3.1.16.3 | Test |  |
| 3.1.16.4 | Test |  |
| 3.2.1 | Demonstration |  |
| 3.2.1.1 | Analysis |  |
| 3.2.1.2 | Analysis |  |
| 3.2.1.3 | Analysis |  |
| 3.2.1.4 | Analysis |  |
| 3.2.2 | Analysis |  |
| 3.3.1 | None | No testing required. |

## Non-Functional Requirements

\*If the testing method needs clarification it will be listed in the Notes column.

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Qualification Method** | **Notes** |
| 4.1.1 | None | No testing required. |
| 4.2.1 | Test, Analysis |  |
| 4.2.2 | Inspection, Analysis |  |
| 4.2.3 | Test | Login with normal user and attempt to access administrator features, the attempt should fail. |
| 4.2.4 | Test, Analysis |  |
| 4.2.5 | Inspection |  |
| 4.2.6. | Test |  |
| 4.2.7 | Test |  |
| 4.2.8 | Test |  |
| 4.2.9 | Test |  |
| 4.2.10 | Test |  |
| 4.2.11 | Test |  |
| 4.3.1 | None | No testing required. |
| 4.4.1.1 | Analysis |  |
| 4.4.1.2 | Analysis |  |
| 4.4.1.4 | Analysis |  |
| 4.4.1.5 | Analysis |  |
| 4.4.1.6 | Analysis |  |
| 4.4.2.1 | Analysis | Analyze the design to look for CPU utilization inefficiencies. |
| 4.4.2.2 | Analysis | Analyze the design to look for bandwidth inefficiencies. |
| 4.4.4.1 | Analysis |  |
| 4.4.3.1 | Analysis |  |
| 4.4.3.2 | Test | Test usage with various web browsers. |
| 4.4.3.3 | Test, Demonstration | Test by setting up a backup job then demonstrating it works to the stake-holder. |
| 4.4.4.2 | Analysis |  |
| 4.4.4.3 | Test, Analysis |  |
| 4.5.1 | Test, Inspection | The code will be examined for any flaws that might introduce instability. |
| 4.5.2 | Analysis |  |
| 4.5.3 | Test |  |
| 4.5.4 | Test, Inspection | Test with erroneous input and look for flaws in validation routines. |
| 4.5.5 | Test |  |

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Qualification Method** | **Notes** |
| 4.5.6 | Test, Inspection | Test system response time with various data set sizes. |
| 4.5.7 | Inspection, Analysis |  |
| 4.5.8 | Demonstration, Special | The system will be tested by examining a user who has never used the system before and how they interact with it, specifically how long it takes for them to become familiar with basic functions. |
| 4.5.9 | Test, Inspection, Analysis | Test with erroneous input as well as examine for flaws in both design and code. |
| 4.6.1 | None | No testing required. |

# Requirements Traceability

## Upward Traceability

Traceability from each requirement in this specification to the system or subsystem requirements it addresses.

### Functional Requirements

| **Req** | **SysR1** | **SysR2** | **SysR3** | **SysR4** | **SysR5** | **SysR6** | **SysR7** | **SysR8** | **SysR9** | **SysR10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3.1.1 |  | X |  |  |  |  |  |  |  | X |
| 3.1.1.1 |  | X |  |  |  |  |  |  |  | X |
| 3.1.1.2 |  | X |  |  |  |  |  |  |  | X |
| 3.1.2 |  | X |  |  | X |  |  |  |  | X |
| 3.1.3 |  | X |  |  | X |  |  |  |  | X |
| 3.1.3.1 | X | X | X |  |  |  |  |  |  | X |
| 3.1.3.2 | X | X |  |  |  |  |  |  |  | X |
| 3.1.3.3 | X | X |  |  |  |  |  |  |  | X |
| 3.1.3.4 | X | X |  |  |  |  |  |  |  | X |
| 3.1.3.5 | X | X |  |  |  |  |  |  |  | X |
| 3.1.3.6 | X | X |  |  |  |  |  |  |  | X |
| 3.1.3.7 | X | X |  |  |  |  |  |  |  | X |
| 3.1.3.8 | X | X |  |  |  |  |  |  |  | X |
| 3.1.3.9 | X | X |  |  |  |  |  |  |  | X |
| 3.1.3.10 | X | X | X |  |  |  |  |  |  | X |
| 3.1.3.11 |  | X | X |  |  |  |  |  |  | X |
| 3.1.3.12 | X | X |  |  |  |  |  |  |  | X |
| 3.1.3.13 |  | X |  |  |  |  |  |  |  | X |
| 3.1.4 |  | X |  |  | X |  |  |  |  |  |
| 3.1.4.1 |  | X |  |  |  |  |  |  |  | X |
| 3.1.4.2 |  | X |  |  |  |  |  | X |  |  |
| 3.1.4.3 |  | X |  |  |  |  |  | X |  |  |
| 3.1.4.4 |  | X |  |  |  |  |  | X |  |  |
| 3.1.4.5 |  | X |  |  |  |  |  |  |  |  |
| 3.1.4.6 |  | X |  |  |  |  |  | X |  |  |
| 3.1.4.7 |  | X |  |  |  |  |  |  |  |  |
| 3.1.4.8 |  | X | X |  |  |  |  |  |  |  |
| 3.1.4.9 |  | X |  |  |  |  |  |  |  |  |
| 3.1.4.10 |  | X | X |  |  |  |  |  |  |  |
| 3.1.4.11 |  | X | X |  |  |  |  |  |  |  |
| 3.1.5 |  | X | X |  | X |  |  |  |  |  |
| 3.1.5.1 |  | X |  |  |  |  |  |  |  | X |
| 3.1.5.2 |  | X |  |  |  |  |  |  |  |  |
| 3.1.5.3 |  | X | X |  |  |  |  |  |  |  |
| 3.1.5.4 |  | X | X |  |  |  |  |  |  |  |
| **Req** | **SysR1** | **SysR2** | **SysR3** | **SysR4** | **SysR5** | **SysR6** | **SysR7** | **SysR8** | **SysR9** | **SysR10** |
| 3.1.6 |  | X |  |  | X |  |  |  |  |  |
| 3.1.6.1 |  | X |  |  |  |  |  |  |  | X |
| 3.1.6.2 |  | X |  |  |  |  |  |  |  |  |
| 3.1.6.3 |  | X | X |  |  |  |  |  |  |  |
| 3.1.6.4 |  | X |  |  |  |  |  |  |  |  |
| 3.1.6.5 |  | X | X |  |  |  |  | X |  |  |
| 3.1.7 |  |  | X |  | X |  |  | X |  |  |
| 3.1.7.1 |  |  | X |  |  |  |  | X |  |  |
| 3.1.7.2 |  |  | X |  |  |  |  | X |  |  |
| 3.1.7.3 |  |  | X |  |  |  |  |  |  |  |
| 3.1.7.4 |  |  | X |  |  |  |  |  |  |  |
| 3.1.7.5 |  |  | X |  |  |  |  |  |  |  |
| 3.1.7.6 |  |  | X |  |  |  |  |  |  |  |
| 3.1.7.7 |  |  | X |  |  |  |  |  |  |  |
| 3.1.7.8 |  |  | X |  |  |  |  |  |  |  |
| 3.1.7.9 |  |  | X |  |  |  |  |  |  |  |
| 3.1.7.10 |  |  | X |  |  |  |  |  |  |  |
| 3.1.7.11 |  |  | X |  |  |  |  |  |  |  |
| 3.1.8 |  | X |  |  | X |  |  |  |  |  |
| 3.1.8.1 |  | X |  |  |  |  |  |  |  |  |
| 3.1.8.2 |  | X |  |  |  |  |  |  |  |  |
| 3.1.8.3 |  | X |  |  |  |  |  |  |  |  |
| 3.1.8.4 | X | X |  |  |  |  |  |  |  |  |
| 3.1.8.5 |  | X |  |  |  |  |  |  |  |  |
| 3.1.8.6 |  | X |  |  |  |  |  |  |  |  |
| 3.1.8.7 |  | X |  |  |  |  |  |  |  |  |
| 3.1.8.8 |  | X |  |  |  |  |  |  |  |  |
| 3.1.8.9 |  | X |  |  |  |  |  |  |  |  |
| 3.1.8.10 |  | X |  |  |  |  |  |  |  |  |
| 3.1.8.11 |  | X | X |  |  |  |  |  |  |  |
| 3.1.9 | X | X |  |  | X |  |  |  |  |  |
| 3.1.9.1 |  | X |  |  |  |  |  | X |  |  |
| 3.1.9.2 |  | X |  |  |  |  |  |  |  |  |
| 3.1.9.3 |  | X |  |  |  |  |  |  |  |  |
| 3.1.9.4 |  | X |  |  |  |  |  |  |  |  |
| 3.1.9.5 | X | X |  |  |  |  |  |  |  |  |
| 3.1.9.6 |  | X |  |  |  |  |  |  |  |  |
| 3.1.9.7 |  | X | X |  |  |  |  |  |  |  |
| 3.1.9.8 |  | X | X |  |  |  |  |  |  |  |
| 3.1.10 | X | X |  |  | X |  |  |  |  |  |
| 3.1.10.1 |  | X |  |  |  |  |  | X |  |  |
| 3.1.10.2 |  | X |  |  |  |  |  |  |  |  |
| 3.1.10.3 |  | X |  |  |  |  |  |  |  |  |
| 3.1.10.4 |  | X |  |  |  |  |  |  |  |  |
| 3.1.10.5 |  | X |  |  |  |  |  |  |  |  |
| 3.1.10.6 |  | X | X |  |  |  |  |  |  |  |
| **Req** | **SysR1** | **SysR2** | **SysR3** | **SysR4** | **SysR5** | **SysR6** | **SysR7** | **SysR8** | **SysR9** | **SysR10** |
| 3.1.10.7 |  | X | X |  |  |  |  |  |  |  |
| 3.1.10.8 |  | X |  |  |  |  |  |  |  |  |
| 3.1.10.9 |  | X |  |  |  |  |  |  |  |  |
| 3.1.10.10 |  | X |  |  |  |  |  |  |  |  |
| 3.1.10.11 |  | X |  |  |  |  |  |  |  |  |
| 3.1.11 | X | X |  |  | X |  |  |  |  |  |
| 3.1.11.1 |  | X |  |  | X |  |  |  |  |  |
| 3.1.11.2 |  | X |  |  | X |  |  |  |  |  |
| 3.1.11.3 |  | X |  |  | X |  |  |  |  |  |
| 3.1.11.4 |  | X |  |  | X |  |  |  |  |  |
| 3.1.11.5 |  | X |  |  | X |  |  |  |  |  |
| 3.1.11.6 |  | X |  |  | X |  |  |  |  |  |
| 3.1.11.7 |  | X |  |  | X |  |  |  |  |  |
| 3.1.11.8 |  |  |  |  | X |  |  |  |  |  |
| 3.1.11.9 |  |  |  |  | X |  |  |  |  |  |
| 3.1.11.10 |  |  |  |  | X |  |  |  |  |  |
| 3.1.11.11 |  |  |  |  | X |  |  |  |  |  |
| 3.1.11.11 |  |  |  |  | X |  |  |  |  |  |
| 3.1.11.12 |  |  |  |  | X |  |  |  |  |  |
| 3.1.11.13 |  |  |  |  | X |  |  |  |  |  |
| 3.1.11.14 |  |  |  |  | X |  |  |  |  |  |
| 3.1.11.15 |  |  |  |  | X |  |  |  |  |  |
| 3.1.11.16 |  |  |  |  | X |  |  |  |  |  |
| 3.1.11.17 |  |  |  |  | X |  |  |  |  |  |
| 3.1.11.18 |  |  |  |  | X |  |  |  |  |  |
| 3.1.12 | X | X |  |  | X |  |  | X |  |  |
| 3.1.12.1 |  |  |  |  | X |  |  |  |  |  |
| 3.1.12.2 |  |  |  |  | X |  |  |  |  |  |
| 3.1.12.3 |  |  |  |  | X |  |  |  |  |  |
| 3.1.12.4 |  |  |  |  | X |  |  |  |  |  |
| 3.1.12.5 |  |  |  |  | X |  |  |  |  |  |
| 3.1.13 | X | X |  |  | X |  |  | X |  |  |
| 3.1.13.1 |  |  |  |  | X |  |  |  |  |  |
| 3.1.13.2 |  |  |  |  | X |  |  |  |  |  |
| 3.1.13.3 |  |  |  |  | X |  |  |  |  |  |
| 3.1.13.4 |  |  |  |  | X |  |  |  |  |  |
| 3.1.13.5 |  |  |  |  | X |  |  |  |  |  |
| 3.1.14 | X | X | X |  | X |  |  | X |  |  |
| 3.1.14.1 |  |  |  |  | X |  |  |  |  |  |
| 3.1.14.2 |  |  |  |  | X |  |  |  |  |  |
| 3.1.14.3 |  |  |  |  | X |  |  |  |  |  |
| 3.1.14.4 |  |  |  |  | X |  |  |  |  |  |
| 3.1.14.5 |  |  |  |  | X |  |  |  |  |  |
| 3.1.14.6 |  |  |  |  | X |  |  |  |  |  |
| 3.1.14.7 |  |  |  |  | X |  |  |  |  |  |
| 3.1.14.8 |  |  |  |  | X |  |  |  |  |  |
| **Req** | **SysR1** | **SysR2** | **SysR3** | **SysR4** | **SysR5** | **SysR6** | **SysR7** | **SysR8** | **SysR9** | **SysR10** |
| 3.1.15 | X | X | X |  |  |  |  | X |  |  |
| 3.1.15.1 |  |  | X |  |  |  |  |  |  |  |
| 3.1.15.2 |  |  | X |  |  |  |  |  |  |  |
| 3.1.15.3 |  |  | X |  |  |  |  |  |  |  |
| 3.1.15.4 |  |  | X |  |  |  |  |  |  |  |
| 3.1.15.5 |  |  | X |  |  |  |  |  |  |  |
| 3.1.15.6 |  |  | X |  |  |  |  |  |  |  |
| 3.1.15.7 |  |  | X |  |  |  |  |  |  |  |
| 3.1.16 | X | X | X |  |  |  |  | X |  |  |
| 3.1.16.1 |  |  | X |  |  |  |  |  |  |  |
| 3.1.16.2 |  |  | X |  |  |  |  |  |  |  |
| 3.1.16.3 |  |  | X |  |  |  |  |  |  |  |
| 3.1.16.4 |  |  | X |  |  |  |  |  |  |  |
| 3.2.1 | X |  |  | X |  | X |  |  |  |  |
| 3.2.1.1 |  |  |  |  | X |  |  |  |  |  |
| 3.2.1.2 |  |  |  |  | X |  |  |  |  |  |
| 3.2.1.3 |  |  |  |  | X |  |  |  |  |  |
| 3.2.1.4 |  |  |  |  | X |  |  |  |  |  |
| 3.2.2 |  |  |  |  |  |  |  | X |  |  |
| 3.3.1 |  |  |  |  |  |  | X |  |  |  |

### Non-Functional Requirements

Non-Functional Requirements Upward Traceability Matrix

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req** | **SysR1** | **SysR2** | **SysR3** | **SysR4** | **SysR5** | **SysR6** | **SysR7** | **SysR8** | **SysR9** | **SysR10** |
| 4.2.1 |  |  |  |  |  |  |  |  |  |  |
| 4.2.2 |  |  |  |  |  |  |  |  |  |  |
| 4.2.3 |  | X |  |  |  |  |  |  |  | X |
| 4.2.4 | X |  |  |  |  |  |  |  |  |  |
| 4.2.5 |  |  |  |  |  |  |  |  |  |  |
| 4.2.6. |  | X |  |  |  |  |  |  |  | X |
| 4.2.7 |  | X |  |  |  |  |  |  |  | X |
| 4.2.8 |  |  |  |  |  |  |  |  |  |  |
| 4.2.9 |  |  |  |  |  |  |  |  |  |  |
| 4.2.10 |  | X |  |  |  |  |  |  |  | X |
| 4.2.11 |  | X |  |  |  |  |  |  |  | X |
| 4.4.1.1 | X |  |  | X |  |  | X |  | X |  |
| 4.4.1.2 | X |  |  |  |  |  | X |  | X |  |
| 4.4.1.4 | X |  |  |  |  |  | X |  | X |  |
| 4.4.1.5 | X |  |  |  |  |  | X |  | X |  |
| 4.4.1.6 |  |  |  |  |  |  | X |  | X |  |
| 4.4.2.1 |  |  |  |  |  |  | X | X | X |  |
| 4.4.2.2 |  |  |  |  |  |  | X |  | X |  |
| 4.4.3.1 |  |  |  |  |  |  |  | X |  |  |
| 4.4.3.2 | X |  |  |  |  |  |  | X |  |  |
| 4.4.3.3 |  |  |  |  |  |  | X |  | X |  |
| 4.4.4.1 | X |  |  |  |  |  |  |  |  |  |
| 4.4.4.2 | X |  |  |  |  |  |  |  |  |  |
| 4.4.4.3 |  | X |  |  |  |  |  |  |  | X |
| 4.5.1 |  |  |  |  |  |  | X |  | X |  |
| 4.5.2 |  |  |  |  |  |  | X |  | X |  |
| 4.5.3 |  |  |  |  | X |  |  |  |  |  |
| 4.5.4 |  |  |  |  | X |  |  |  |  |  |
| 4.5.5 |  | X |  |  |  |  |  |  |  | X |
| 4.5.6 |  |  |  |  |  |  |  |  |  |  |
| 4.5.7 |  |  |  |  |  |  | X |  | X |  |
| 4.5.8 |  |  |  |  |  | X |  |  |  |  |
| 4.5.9 |  |  |  |  | X |  |  |  |  |  |

## 

## Downward Traceability

Traceability from each system or subsystem requirement to the requirements contained in this specification

### Functional Requirements

Functional Requirements Downward Traceability Matrix 1 of 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req** | 3.1.1 | 3.1.1.1 | 3.1.1.2 | 3.1.2 | 3.1.3 | 3.1.3.1 | 3.1.3.2 | 3.1.3.3 | 3.1.3.4 | 3.1.3.5 | 3.1.3.6 | 3.1.3.7 | 3.1.3.8 | 3.1.3.9 | 3.1.3.10 | 3.1.3.11 | 3.1.3.12 | 3.1.3.13 |
| **SysR1** |  |  |  |  |  | X | X | X | X | X | X | X | X | X | X |  | X |  |
| **SysR2** | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| **SysR3** |  |  |  |  |  | X |  |  |  |  |  |  |  |  | X | X |  |  |
| **SysR4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR5** |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR7** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR8** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR9** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR10** | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

Functional Requirements Downward Traceability Matrix 2 of 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req** | 3.1.4 | 3.1.4.1 | 3.1.4.2 | 3.1.4.3 | 3.1.4.4 | 3.1.4.5 | 3.1.4.6 | 3.1.4.7 | 3.1.4.8 | 3.1.4.9 | 3.1.4.10 | 3.1.4.11 | 3.1.5 | 3.1.5.1 | 3.1.5.2 | 3.1.5.3 | 3.1.5.4 | 3.1.6 |
| **SysR1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR2** | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| **SysR3** |  |  |  |  |  |  |  |  | X |  | X | X | X |  |  | X | X |  |
| **SysR4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR5** | X |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  | X |
| **SysR6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR7** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR8** |  |  | X | X | X |  | X |  |  |  |  |  |  |  |  |  |  |  |
| **SysR9** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR10** |  | X |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |

Functional Requirements Downward Traceability Matrix 3 of 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req** | 3.1.6.1 | 3.1.6.2 | 3.1.6.3 | 3.1.6.4 | 3.1.6.5 | 3.1.7 | 3.1.7.1 | 3.1.7.2 | 3.1.7.3 | 3.1.7.4 | 3.1.7.5 | 3.1.7.6 | 3.1.7.7 | 3.1.7.8 | 3.1.7.9 | 3.1.7.10 | 3.1.7.11 | 3.1.8 |
| **SysR1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR2** | X | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  | X |
| **SysR3** |  |  | X |  | X | X | X | X | X | X | X | X | X | X | X | X | X |  |
| **SysR4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR5** |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  | X |
| **SysR6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR7** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR8** |  |  |  |  | X | X | X | X |  |  |  |  |  |  |  |  |  |  |
| **SysR9** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR10** | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Functional Requirements Downward Traceability Matrix 4 of 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req** | 3.1.8.1 | 3.1.8.2 | 3.1.8.3 | 3.1.8.4 | 3.1.8.5 | 3.1.8.6 | 3.1.8.7 | 3.1.8.8 | 3.1.8.9 | 3.1.8.10 | 3.1.8.11 | 3.1.9 | 3.1.9.1 | 3.1.9.2 | 3.1.9.3 | 3.1.9.4 | 3.1.9.5 | 3.1.9.6 |
| **SysR1** |  |  |  | X |  |  |  |  |  |  |  | X |  |  |  |  | X |  |
| **SysR2** | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| **SysR3** |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| **SysR4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR5** |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |
| **SysR6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR7** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR8** |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |
| **SysR9** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Functional Requirements Downward Traceability Matrix 5 of 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req** | 3.1.9.7 | 3.1.9.8 | 3.1.10 | 3.1.10.1 | 3.1.10.2 | 3.1.10.3 | 3.1.10.4 | 3.1.10.5 | 3.1.10.6 | 3.1.10.7 | 3.1.10.8 | 3.1.10.9 | 3.1.10.10 | 3.1.10.11 | 3.1.11 | 3.1.11.1 | 3.1.11.2 | 3.1.11.3 |
| **SysR1** |  |  | X |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| **SysR2** | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| **SysR3** | X | X |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |
| **SysR4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR5** |  |  | X |  |  |  |  |  |  |  |  |  |  |  | X | X | X | X |
| **SysR6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR7** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR8** |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR9** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Functional Requirements Downward Traceability Matrix 6 of 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req** | 3.1.11.4 | 3.1.11.5 | 3.1.11.6 | 3.1.11.7 | 3.1.11.8 | 3.1.11.9 | 3.1.11.10 | 3.1.11.11 | 3.1.11.11 | 3.1.11.12 | 3.1.11.13 | 3.1.11.14 | 3.1.11.15 | 3.1.11.16 | 3.1.11.17 | 3.1.11.18 | 3.1.12 | 3.1.12.1 |
| **SysR1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |
| **SysR2** | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  | X |  |
| **SysR3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR5** | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| **SysR6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR7** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR8** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |
| **SysR9** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Functional Requirements Downward Traceability Matrix 7 of 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req** | 3.1.12.2 | 3.1.12.3 | 3.1.12.4 | 3.1.12.5 | 3.1.13 | 3.1.13.1 | 3.1.13.2 | 3.1.13.3 | 3.1.13.4 | 3.1.13.5 | 3.1.14 | 3.1.14.1 | 3.1.14.2 | 3.1.14.3 | 3.1.14.4 | 3.1.14.5 | 3.1.14.6 | 3.1.14.7 |
| **SysR1** |  |  |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  |
| **SysR2** |  |  |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  |
| **SysR3** |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| **SysR4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR5** | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| **SysR6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR7** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR8** |  |  |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  |
| **SysR9** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Functional Requirements Downward Traceability Matrix 8 of 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req** | 3.1.14.8 | 3.1.15 | 3.1.15.1 | 3.1.15.2 | 3.1.15.3 | 3.1.15.4 | 3.1.15.5 | 3.1.15.6 | 3.1.15.7 | 3.1.16 | 3.1.16.1 | 3.1.16.2 | 3.1.16.3 | 3.1.16.4 | 3.2.1 | 3.2.1.1 | 3.2.1.2 | 3.2.1.3 |
| **SysR1** |  | X |  |  |  |  |  |  |  | X |  |  |  |  | X |  |  |  |
| **SysR2** |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| **SysR3** |  | X | X | X | X | X | X | X | X | X | X | X | X | X |  |  |  |  |
| **SysR4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| **SysR5** | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X |
| **SysR6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| **SysR7** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR8** |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| **SysR9** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Functional Requirements Downward Traceability Matrix 9 of 9

|  |  |  |  |
| --- | --- | --- | --- |
| **Req** | 3.2.1.4 | 3.2.2 | 3.3.1 |
| **SysR1** |  |  |  |
| **SysR2** |  |  |  |
| **SysR3** |  |  |  |
| **SysR4** |  |  |  |
| **SysR5** | X |  |  |
| **SysR6** |  |  |  |
| **SysR7** |  |  | X |
| **SysR8** |  | X |  |
| **SysR9** |  |  |  |
| **SysR10** |  |  |  |

### Non-Functional Requirements

Non-Functional Requirements Downward Traceability Matrix 1 of 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req** | 4.2.1 | 4.2.2 | 4.2.3 | 4.2.4 | 4.2.5 | 4.2.6. | 4.2.7 | 4.2.8 | 4.2.9 | 4.2.10 | 4.2.11 | 4.4.1.1 | 4.4.1.2 | 4.4.1.4 | 4.4.1.5 | 4.4.1.6 | 4.4.2.1 | 4.4.2.2 |
| **SysR1** |  |  |  | X |  |  |  |  |  |  |  | X | X | X | X |  |  |  |
| **SysR2** |  |  | X |  |  | X | X |  |  | X | X |  |  |  |  |  |  |  |
| **SysR3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR4** |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |
| **SysR5** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR7** |  |  |  |  |  |  |  |  |  |  |  | X | X | X | X | X | X | X |
| **SysR8** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |
| **SysR9** |  |  |  |  |  |  |  |  |  |  |  | X | X | X | X | X | X | X |
| **SysR10** |  |  | X |  |  | X | X |  |  | X | X |  |  |  |  |  |  |  |

Non-Functional Requirements Downward Traceability Matrix 2 of 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req** | 4.4.3.1 | 4.4.3.2 | 4.4.3.3 | 4.4.4.1 | 4.4.4.2 | 4.4.4.3 | 4.5.1 | 4.5.2 | 4.5.3 | 4.5.4 | 4.5.5 | 4.5.6 | 4.5.7 | 4.5.8 | 4.5.9 |
| **SysR1** |  | X |  | X | X |  |  |  |  |  |  |  |  |  |  |
| **SysR2** |  |  |  |  |  | X |  |  |  |  | X |  |  |  |  |
| **SysR3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR5** |  |  |  |  |  |  |  |  | X | X |  |  |  |  | X |
| **SysR6** |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |
| **SysR7** |  |  | X |  |  |  | X | X |  |  |  |  | X |  |  |
| **SysR8** | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SysR9** |  |  | X |  |  |  | X | X |  |  |  |  | X |  |  |
| **SysR10** |  |  |  |  |  | X |  |  |  |  | X |  |  |  |  |