## Requirement Specification

Title Requirements Specification (RS)

Document Control

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared** | **Reviewed** | **Approved** |
| 14/10/2014 | 1 | Create | AB | X | X |
| 21/10/14 | 2 | Update | CD |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Version** |
| Bruce Wayne | Lead Developer | 1 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Related Documents

|  |  |
| --- | --- |
| **Title** | **Comments** |
| Title of Use Case Model |  |
| Title of Use Case Description |  |

**Table of Contents**

[1 Introduction 4](#_Toc239580623)

[1.1 Purpose 4](#_Toc239580624)

[1.2 Project Scope 4](#_Toc239580625)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc239580626)

[2 User requirements definition 4](#_Toc239580627)

[3 System architecture 4](#_Toc239580628)

[4 Requirements specification 5](#_Toc239580629)

[4.1 Physical environment requirements 5](#_Toc239580630)

[4.1.1 Requirement 1 <name of requirement in a few words> 5](#_Toc239580631)

[4.1.2 Description & Priority 5](#_Toc239580632)

[4.2 Interface requirements 5](#_Toc239580633)

[4.2.1 Requirement 1 <name of interface requirement in a few words> 5](#_Toc239580634)

[4.2.2 Description & Priority 5](#_Toc239580635)

[4.3 Functional requirements 5](#_Toc239580636)

[4.3.1 Requirement 1 <name of requirement in a few words> 6](#_Toc239580637)

[4.3.2 Requirement 1 <name of requirement in a few words> 6](#_Toc239580638)

[4.4 Documentation requirements 7](#_Toc239580639)

[4.4.1 Requirement 1 <name of document requirement in a few words> 7](#_Toc239580640)

[4.4.2 Description & Priority 7](#_Toc239580641)

[4.5 Data requirements 8](#_Toc239580642)

[4.5.1 Requirement 1 <name of data requirement in a few words> 8](#_Toc239580643)

[4.5.2 Description & Priority 8](#_Toc239580644)

[4.6 Non-Functional Requirements 8](#_Toc239580645)

[4.6.1 Performance/Response time requirement 8](#_Toc239580646)

[4.6.2 Availability requirement 8](#_Toc239580647)

[4.6.3 Recover requirement 8](#_Toc239580648)

[4.6.4 Robustness requirement 8](#_Toc239580649)

[4.6.5 Security requirement 8](#_Toc239580650)

[4.6.6 Reliability requirement 8](#_Toc239580651)

[4.6.7 Maintainability requirement 8](#_Toc239580652)

[4.6.8 Portability requirement 8](#_Toc239580653)

[4.6.9 Extendibility requirement 8](#_Toc239580654)

[4.6.10 Reusability requirement 8](#_Toc239580655)

[4.6.11 Resource utilization requirement 8](#_Toc239580656)

[5 System models 8](#_Toc239580657)

[6 System evolution 8](#_Toc239580658)

[7 Appendices 9](#_Toc239580659)

[7.1 Use case 1 9](#_Toc239580660)

# Introduction

## Purpose

The purpose of this document is to set out the requirements for the development of a system to make the organisation of an event, such as a child’s Birthday Party or a Wedding with the least amount of difficulty. The intended customers are regular consumers (parents, family members), Venue owners (Hotels, Bars, Clubs) and service providers (Singers, Musicians, Caterers).

## Project Scope

The scope of the project is to develop a user friendly, reliable system to facilitate the booking of events, and the advertising of venues and services. The system shall have an interface which will take all users through their respective processes with minimal difficulty.

Extensive and exhaustive research was carried out to elicit the following requirements.

All critical data captured i.e. personal and credit card details must be dealt with in a safe and secure fashion. Paypal??

The Project must be completed as a fully functional proof of concept by April 15th 2015.

This section also details any constraints that were placed upon the requirements elicitation process, such as schedules, costs, or the software engineering environment used to develop requirements.

## Definitions, Acronyms, and Abbreviations

AD Another Definition

……..

# User requirements definition

This section describes the set of objectives and requirements for the system from the customer’s perspective. It may include a "wish list" of desirable characteristics, along with more feasible solutions that are in line with the business objectives. User requirements may be described using natural language, and other techniques that are understandable to the customer.

The main purpose of the **Event-U-Ally** system is to enable individuals to organise events by way of enabling them to identify venues and service providers and co—ordinate the booking of both at a date and time and which is mutually convenient to all parties and at a cost which is acceptable to the organiser.

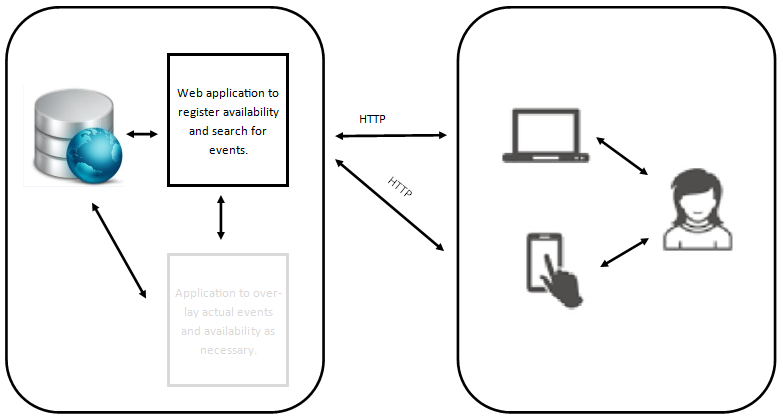
The **Event-U-Ally** will have three main actors/stakeholders as follows:

* Event Organiser
* Venue Provider
* Service Provider

All three of these must be able to register according to the category to which they belong, although it is conceivable that one actor could at times belong to all three categories.

# System architecture

This system describes the architecture view of the system.



# Requirements specification

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

The following table demonstrates how to make requirements verifiable incorporate metrics as follows:

* Register Users
* Register Services
* Organise Events
* Manage Services/Events

Organise events

The newly created profile, or a passing user should be able to enter search criteria and browse available services defined by the parameters of When, Where, What and Why. The order of choosing essentially will not matter as the criteria will return the same pool of services if all criteria are filled out.

The user must then be able to easily select services they have searched for, access available options/ information and choose to book the service or not.

**Property**

**Measure**

**Speed**

**Processed transactions/second**

**User/Event response time**

**Screen refresh time**

**Size**

**K Bytes**

**Number of RAM chips**

**Ease of use**

**Training time**

**Number of help frames**

**Reliability**

**Mean time to failure**

**Probability of unavailability**

**Rate of failure occurrence**

**Availability**

**Robustness**

**Time to restart after failure**

**Percentage of events causing failure**

**Probability of data corruption on failure**

**Portability**

**Percentage of target dependent statements**

**Number of target systems**

## Physical environment requirements

This section describes the environment where the equipment is to function and if there are any environmental restrictions, such as temperature, humidity or magnetic interference.

Event-U-Ally will be run as an online service and as such will need to be hosted on a server with adequate capacity to satisfy demand. Cloud based servers such as those provided

by Microsoft on the Azure cloud platform or by Amazon on AWS would enable the operation to get off the ground at minimal cost and yet offer the ability to scale up if and when required.

### Requirement 1 <name of requirement in a few words>

### Description & Priority

Description of requirement and priority,

## Interface requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include library routines, token streams, shared memory, data streams, and so forth. Examples of Interfaces are

* **GUI Graphical User Interfaces**
* **CLI Describes the command-line**
* API Application Platform Interface

Other considerations when identifying interface requirements:-

* Is the input coming from one or *more systems*
* Is the output going to one or more systems
* Is there a prescribed way in which the data must be formatted
* Is there a prescribed way in which the data must use

### Requirement 1 <name of interface requirement in a few words>

### Description & Priority

Description of requirement and priority eg.

Connections to Oracle, PHP, PHP > Oracle

## Functional requirements

This section lists the functional requirements in ranked order. Functional requirements describes the possible effects of a software system, in other words, what the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe how the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

Short, imperative sentence stating highest ranked functional requirement.

### Requirement 1 Organise Event

#### Description & Priority

Organising an event is the primary requirement in the Event-U-Ally system, as it’s the main reason for users to use the system, all other user activities either feed into or derive from the creating of an event.

#### Requirement Activation

A user wishing to avail of the Event-U-Ally service must have access to a web browser, ideally an up-to-date one capable of interpreting HTML 5

The first interface seen by a user navigating to the Event-U-ally site will be the option to arrange an event.

#### Technical issues

Describes any design or implementation issues involved in satisfying this requirement.

#### Risks

Describes the circumstances under which this requirement might not able to be satisfied, and what actions can be taken to reduce the probability of this occurrence.

* **Out of date browser (incompatible with HTML5)**

Use HTML if statements to handle the error.

* **User has no internet connection**

TOUGH!! What are we…. Magicians?

#### Dependencies with other requirements

Describes interactions with other requirements.

#### Use Case Diagram



#### Flow Diagram



#### Functional Requirements

**Use Case 1 ………**<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

<Create a separate use case and place it in the appendix. As an example see Appendix 7.1.>

**Use Case 1 Organise Event**

**Precondition**

The user has navigated via a web browser to the Event-U-Ally home page.

**Activation**

This user case starts when the user chooses to organise an event.

**Main Flow**

1. The user chooses to search for either available venues or services.
2. The user selects a venue or service from a list.
3. The user books the chosen venue/service.
4. The user chooses to search for an additional service/venue (See A1).
5. The user opts to confirm the booking.
6. The user signs in with credentials.
7. The user isn’t registered (See A2).
8. The user confirms the booking.
9. The user exits the system.

**Alternate Flows**

A1 : <Search additional Venues/Services>

1. The user continues to Main Flow point 1.

A2 : <User isn’t registered>

1. The user continues to Register User point 1.

**Termination**

The system presents a summary of booking with an option to print.

**Post Condition**

The system returns to home screen.

### Requirement 2 Register User

#### Description & Priority

This use case outlines the process by which a user registers to be able to avail of the Event-U-Ally system. A user can browse the system without being registered, but they must be registered in order to confirm bookings. This requirement is necessary on two levels; firstly it enables the exchange of contact details between all parties involved in an event, and secondly it ensures traceability in the event of misuse of the system.

#### Requirement Activation

How the requirement will be used by user

The user can register at any point and at their own discretion whilst browsing the system, however once they wish to make a booking the system makes it an essential requirement.

#### Technical issues

There are no technical issues concerning this requirement. ***Possibly a secure connection!!***

#### Risks

In the event that a user, for their own reasons, doesn’t want to give the necessary details, the system will not allow them to proceed.

#### Dependencies with other requirements

Describes interactions with other requirements.

#### Flow Diagram



**Use Case 2 Register**

**Precondition**

The user has navigated via a web browser to the Event-U-Ally home page and or has been browsing the available services/venues.

**Activation**

This user case starts when the user chooses to register or when they’ve reached a point in the event organisation process where it becomes necessary.

**Main Flow**

1. The user chooses to register.
2. The user fills in the details on the form presented.
3. The user clicks the submit button.
4. The user selects the return to confirm booking option (See A1).
5. The user exits the registration process.

**Alternate Flows**

A1 : <Return to confirm booking>

1. The user continues to Organise Event Main Flow point 8.

## Documentation requirements

Examples of document requirements are

* How much documentation is required
* Should it be on-line, in book format or both
* To what audience is each type of documentation addressed

### Requirement 1 <name of document requirement in a few words>

### Description & Priority

Description of requirement and priority,

## Data requirements

Data requirements should consider the following:

* What should the format of data be for input and output
* *How often will they be sent or received*
* How accurate must they be
* To what degree of precision must the calculations be made
* How much data flow through the system
* Must the data be retained for any period of time

### Requirement 1 <name of data requirement in a few words>

### Description & Priority

Description of requirement and priority,

## Non-Functional Requirements

Specifies any other particular non-functional attributes required by the system. Examples are provided below. Remove the requirement headings that are not appropriate to your project.

### Performance/Response time requirement

### Availability requirement

### Recover requirement

### Robustness requirement

### Security requirement

### Reliability requirement

### Maintainability requirement

### Portability requirement

### Extendibility requirement

### Reusability requirement

### Resource utilization requirement

# System models

This section presents a more detailed description of the system model. For example DFD, ERD, UC Model etc.

# System evolution

This section describes how the system could evolve over time

# Appendices

## Use case 1

**Use case**

Name of use case

**Scope**

The scope of this use case is to …….

**Description**

This use case describes the ………..

**Use Case Diagram**

Diagram should highlight actors and uses cases……..

**Flow Description**

**Precondition**

The system is in initialisation mode……..

**Activation**

This use case starts when an <Actor>…………

**Main flow**

1. The system identifies the ………….
2. The <Actor> …………...(See A1)
3. The system …………..(See E1)
4. The <Actor> ………….

**Alternate flow**

A1 : <title of A1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 3 of the main flow

**Exceptional flow**

E1 : <title of E1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 4 of the main flow

**Termination**

The system presents the next ……….

**Post condition**

The system goes into a wait state