

<Project Name>

Requirement Specifications

Document Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Baseline Version** | **Approved Date** | **Reviewer** | **Approver** |
| <N.M> | dd-mm-YY | <Full name> | Full name> |

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change Description** |
| <N.M> | dd-mm-YY | <Full name> | <Description> |

Table of Contents

I. Introduction 4

II. Master Use Case & Functional Diagrams 4

1. Master Use Case Diagram 4

2. Master Functional Diagram 4

III. Functional Requirements 4

1. [Function Name] - [Function ID] 4

2. [Function Name] - [Function ID] 4

3. [Function Name] - [Function ID] 4

IV. Non Functional Requirements 4

V. Risk and Communication Plan 4

1. Risks and Mitigation Plan 4

2. Communication Plan 4

VI. Glossary 4

# Introduction

*[Describe the overview of the system that Customer expects.*

*Describe business objectives of Customer]*

# Master Use Case & Functional Diagrams

## Master Use Case Diagram

*[Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, use cases and any dependencies between those use cases.]*

## Master Functional Diagram

*[Diagram indicates the functions of the principal parts of a total system and also shows the important relationships and interactions among these parts.]*

*[BA can be totally flexible to add/remove different diagrams to every function based on the project’s characteristics]*

# Functional Requirements

## [Function Name] - [Function ID]

### *Descriptions*

### *Main business rules*

### *Basic flow of Event (activity diagram)*

### *Data Schema (entity relationship diagram)*

### *Data Flow Diagram*

*[BA can be totally flexible to add/remove different diagrams to every function based on the project’s characteristics]*

### *Wireframes*

### *Related Use Cases*

### <ID><Use Case name>

### Descriptions

### Actors

### Related screens

### Constraints & Business rules

### Detailed flow of events

### Error message list

### <ID><Use Case name>

### Descriptions

### Actors

### Related screens

### Constraints & Business rules

### Detailed flow of events

### Error message list

## [Function Name] - [Function ID]

### *Descriptions*

### *Main business rules*

### *Basic flow of Event (activity diagram)*

### *Data Schema (entity relationship diagram)*

### *Data Flow Diagram*

*[BA can be totally flexible to add/remove different diagrams to every function based on the project’s characteristics]*

### *Wireframes*

### *Related Use Cases*

### <ID><Use Case name>

### Descriptions

### Actors

### Related screens

### Constraints & Business rules

### Detailed flow of events

### Error message list

### <ID><Use Case name>

### Descriptions

### Actors

### Related screens

### Constraints & Business rules

### Detailed flow of events

### Error message list

## [Function Name] - [Function ID]

# Non Functional Requirements

*[BA can be totally flexible to add/remove non functional requirements into the checklist below based on the project’s characteristics]*

|  |  |  |
| --- | --- | --- |
| **Non Functional requirement Checklist**  **(standard points for any web project, also called Quality of Service Requirement list)** | | |
| **Accessibility**  The ability is to access information and services by minimizing the barriers of distance and cost as well as the usability of the interface | | |
| **Concurrence** | How many accesses at the same time? |  |
| **Localization** | Make adaptations due to regional differences? | □ |
|  | List all regions here:  1.  2.  3. |  |
| **Portability**  One of the key concepts of [high-level programming](http://en.wikipedia.org/wiki/High-level_programming_language). Portability is the [software](http://en.wikipedia.org/wiki/Software) codebase feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another | | |
| **Strategies Configuration** | Installed program files may be transferred to another computer | □ |
| The program may be reinstalled (from the distribution files) on another computer. | □ |
| The source files may be compiled for another computer. | □ |
| **Operating System** | Window | □ |
| MAC | □ |
| Linux | □ |
| **Web Browser** | Firefox | □ |
| List version of Firefox:  1.  2.  3. |  |
| Internet Explorer | □ |
| List version of IE:  1.  2.  3. |  |
| Safari | □ |
| List version of Safari:  1.  2.  3. |  |
| Chrome | □ |
| List version of Chrome:  1.  2.  3.  4 |  |
| **Backup**  A backup or the process of backing up refers to making copies of [data](http://en.wikipedia.org/wiki/Data) so that these additional copies may be used to restore the original after a [data loss](http://en.wikipedia.org/wiki/Data_loss) event. | | |
| **Unstructured** | Using any peripheral to save minimal information about what was backed up and when | □ |
| **Full +** [**Incremental**](http://en.wikipedia.org/wiki/Incremental_backup) | > At first, a full backup (of all files) is made. After that, any number of incremental backups can be made.  > An incremental backup copies everything that has changed since the last backup (full, differential or incremental) | □ |
| **Full system backup** | Designed to allow an entire PC to be recovered to "bare metal" without any installation of operating system, application software and data | □ |
| **Configuration management**  Focuses on establishing and maintaining consistency of a system or product's performance and its functional and physical attributes with its requirements, design, and operational information throughout its life | | |
|  | Configuration identification:  >  > |  |
|  | Configuration control:  >  > |  |
|  | Configuration status accounting:  >  > |  |
|  | Configuration audits:  >  > |  |
| **Deployment** | | |
|  |  |  |
| **Capacity** | | |
|  | How many items will be saved into DB?  > |  |
| [**Performance**](http://en.wikipedia.org/wiki/Computer_performance) **/ response time (**[**performance engineering**](http://en.wikipedia.org/wiki/Performance_engineering)**)** | | |
|  | Time to load one page  >  > |  |
|  | Time to log in  >  > |  |
| **Security** | | |
|  |  |  |
|  |  |  |

# Risk and Communication Plan

## Risks and Mitigation Plan

### Identify risks

*[Requirements risks are risks that are associated directly to specific requirements.*

*Common risks related to requirement:*

* *Uncertain requirements*
* *Developing the wrong user interface*
* *Changes to the regulatory or legal environment may be coming down the pipeline causing existing requirements to become obsolete.*
* *Requirements may be overly complex leading to a risk of poor understanding by the development team.*
* *Insufficient time may be allocated to requirements gathering and definition resulting in gaps or errors in requirements]*
* *[BA can be totally flexible to define & merge requirement risks with the risks of the project which are fully managed by PM]*

### Mitigation Plan

*[BA can be totally flexible to link this section to the Risk Management Plan of PM]*

## Communication Plan

### Stakeholders and Roles

### Communication method and review schedule plan

*[BA can be totally flexible to link this section to the BA Plan]*

# Glossary

*[BA could add common error message list, Data Dictionary etc into this section]*

*--- End of document ---*