**Requirements Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **PPM #** |  | **Project Name** |  |
| **Date** |  | **Author** |  |

**A requirements specification document is a consolidation of a portfolio, program, or project’s needs. These needs include business, stakeholder, functional, nonfunctional and transition requirements. A well-documented requirements specification is a rigorous assessment of requirements before the more specific system design stages.**

**Project Overview**

Provide contextual background on the project in one to two paragraphs, including the project’s underlying goals and objectives.

[TYPE PROJECT OVERVIEW HERE]

**Scope**

List which subject areas and systems are within and beyond the project’s scope, respectively.

*In Scope*

*Out of Scope*

**Requirements**

Requirements can be classified into various categories to provide clarity and context. The following requirement types help define what information needs to be elicited and the source of that information. Expert judgment, standards or common practices associated with an organization, industry or community of knowledge may be used to determine the types of requirements and enough detail to develop the solution.

* **Business requirements.** Describe the high-level needs of the organization to address a problem or opportunity. These requirements provide the rationale for why a project or program is launched.
* **Stakeholder requirements.** Express the needs of a stakeholder. These requirements communicate stakeholders’ needs of a final product or service.
* **Solution requirements.** Describe the functions that the product or service needs to exhibit to satisfy the business and stakeholder requirements. Solution requirements may include requirements related to technology and standard compliance. All solution requirements should roll up to business and/or stakeholder requirements.
	+ *Functional requirements* denote behaviors and operations that the solution will perform. These focus on the required functionality to enable stakeholders to accomplish their objectives, which in turn fulfil the business need. All functional requirements should be written in such a way that they are testable.
	+ *Nonfunctional requirements* describe certain environmental conditions or required attributes to ensure the product or service operates effectively.
* **Transition requirements.** Describe the temporary capabilities that are essential to migrate from the current state to a future state. Requirements that fall into this category include data conversions, training, and communications.

In the table below, document all requirements pertaining to the project. Specify the requirement type as Business, Stakeholder, Functional, Nonfunctional or Transition. This will help when building out use cases and test scripts later in the project and service development life cycles.

| Req ID | Requirement | Requirement Type |
| --- | --- | --- |
| 1. |  |  |
| 1.01. |  |  |
| 2. |  |  |
| 2.01. |  |  |

**Structural Diagrams**

The structural diagrams represent the static aspects of any systems. These static parts are represented by classes, interfaces, objects, components and nodes.

**Class Diagram**

Class diagrams are a graphical, object-oriented view of the system, which is static in nature.

[INSERT DIAGRAM HERE]

**Object Diagram**

Object diagrams are an instance of a class diagram, which are closer to real-life scenarios and used to build the prototype of a system from a practical perspective.

[INSERT DIAGRAM HERE]

**Entity Relationship Diagram**

An Entity Relationship Diagram (ERD) shows the relationships of entity sets stored in a database. They illustrate the logical structure of databases.

[INSERT DIAGRAM HERE]

**Component Diagram**

Component diagrams represent a set of components and their relationships, consisting of classes, interfaces or collaborations.

[INSERT DIAGRAM HERE]

**Deployment Diagram**

Deployment diagrams are a set of nodes and their relationships, based on the components provided in the Component Diagram. These nodes are physical entities where the components are deployed.

[INSERT DIAGRAM HERE]

**Behavioral Diagrams**

Unlike structural diagrams, behavior diagrams show the dynamic behavior of objects by showing collaborations among actors/objects and changes to the internal states of objects.

**Use Case Diagram**

Use case diagrams are a set of use cases, actors and their relationships. They represent the use case view of a system and functionality of said system. Hence, use case diagrams are used to describe the relationships among the functionalities and their internal/external controllers (actors).

[INSERT DIAGRAM HERE]

## Sequence Diagram

Sequence Diagrams are interactive diagrams, which show a sequence of messages flowing amongst objects.

[INSERT DIAGRAM HERE]

**Collaboration Diagram**

Collaboration Diagrams are visualizations of the organization of objects and their interaction.

[INSERT DIAGRAM HERE]

**State Diagram**

State Diagrams represent the event driven state change of a system and the reaction of a system by internal/external factors.

[INSERT DIAGRAM HERE]

**Activity Diagram**

Activity Diagrams describe the flow of control in a system with activities and links. Flows can be sequential, concurrent or branched. Activity diagrams are used to visualize the flow of controls in a system.

[INSERT DIAGRAM HERE]

**Works Cited**

Project Management Institute. (2016). Requirements Management: A Practice Guide. In *Requirements Management: A Practice Guide* (pp. 64-74). Newtown Square, Pennsylvania: PMI Publications.

**Glossary**

All terminology in this section was pulled directly from the Project Management Institute’s Requirements Management: A Practice Guide (pp. 64-74).

**Business Requirements.** Requirements that describe the higher-level needs of the organization, such as the business issues or opportunities, and which provide the rationale for why a project is being undertaken.

**Functional Requirements.** Requirements that describe the *behaviors* of a product.

**MoSCoW.** A technique used for establishing requirement priorities. In this technique, the participants divide the requirements into four categories of must haves, should haves, could haves and won’t haves.

**Nonfunctional Requirements.** Requirements that express *properties* that the product is required to have, including interface, environment and quality attribute properties.

**Project.** A temporary endeavor undertaken to create a unique product, service or result.

**Project Management.** The application of knowledge, skills, tools and techniques to project activities to meet the project requirements.

**Requirement.** A condition or capability that is required to be present in a product, service or result to satisfy a contract or other formally imposed specification.

**Requirements Life Cycle.** The flow or life of a requirement throughout a project or program. The requirements life cycle is managed by assigning an attribute or qualifier onto the requirement to depict the requirement state at a specified point in time.

**Requirements Traceability Matrix.** A grid that links product requirements from their origin to the deliverables that satisfy them.

**Transition Requirements.** Requirements that are the temporary capabilities, such as data conversion and training requirements, needed to transition from the current as-is state to the future state.

**Use Case.** An analysis model that describes a flow of actor-system interactions and boundaries for those interactions, including trigger, initiating and participating actors, and preconditions and post conditions.

**Sign-Off**