

ROS Requirements Specification

Version 1.0

Table of Contents

1. EXECUTIVE SUMMARY	3
1.1 PROJECT OVERVIEW	3
1.2 PURPOSE AND SCOPE OF THIS SPECIFICATION	3
2. PRODUCT/SERVICE DESCRIPTION	3
2.1 PRODUCT CONTEXT	3
2.2 USER CHARACTERISTICS	3
2.3 ASSUMPTIONS	3
2.4 CONSTRAINTS	3
2.5 DEPENDENCIES	4
3. REQUIREMENTS	4
3.1 FUNCTIONAL REQUIREMENTS	5
3.2 NON-FUNCTIONAL REQUIREMENTS	5
3.2.1 <i>User Interface Requirements</i>	5
3.2.2 <i>Usability</i>	5
3.2.3 <i>Performance</i>	6
3.2.4 <i>Manageability/Maintainability</i>	6
3.2.5 <i>System Interface/Integration</i>	7
3.2.6 <i>Security</i>	8
3.2.7 <i>Data Management</i>	8
3.2.8 <i>Standards Compliance</i>	8
3.2.9 <i>Portability</i>	8
3.2.10 <i>Other Non-Functional Requirements</i>	9
3.3 DOMAIN REQUIREMENTS	9
4. USER SCENARIOS/USE CASES	9
APPENDIX	10
APPENDIX A. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	10
APPENDIX B. REFERENCES	10
APPENDIX C. REQUIREMENTS TRACEABILITY MATRIX	10
APPENDIX D. ORGANIZING THE REQUIREMENTS	12

1. Executive Summary

1.1 Project Overview

This product is meant to be a web application that services bars and restaurants in Tirana. The main purpose of it is to enhance the ordering time for the restaurant's staff as well as the transparency and menu updating in real time. The targeted audience will be bars, restaurants and other businesses that work with clients and ordering that are related in the food industry.

1.2 Purpose and Scope of this Specification

The main purpose of our product is to help in the management of a restaurant's, bar's or any other service industry business to easily manage client orders among waiters due to the order system implemented in our product. Features implemented on this system are more user friendly for both the client and the employees.

2. Product/Service Description

2.1 Product Context

This product does somehow relate to another existing product that is already applied in some bars and restaurants. Our product however contains a more complex way of managing the orders and the system of the business. It has some added features that improves the functioning of a food ordering based business. The system is meant to be used by the whole staff and also the clients of the restaurant where each of the users has a different user interface and access and limitations to different parts of the system. It is supposed to make the functioning of the business more efficient and also time saving. The users are interconnected with each other and operate simultaneously depending on each other.

2.2 User Characteristics

There will be a total of five users :

- 1- Admin
- 2- Manager
- 3- Waiter
- 4- Kitchen staff
- 5- Client

User profile : Admin

- This admin will be logged in.
- It will also be a technical support for different possible bugs.
- Can access all the other users of the software.
- Has access to the orders or even table bills.
- Can restrain users from logging in or even delete them

User profile : Manager

- Will be registered and logged in.
- Will be able to help the Waiter and Kitchen staff with the orders.
- Can access the table bills.
- Has the ability to add or remove different staff positions (waiter or kitchen).
- Can restrain waiter or kitchen staff from logging in or even remove them.
- Can add, update or delete menu items.

User profile : Waiter

- Will be registered and logged in.

[YourProject] Requirements Specification

- Will be able to access table orders and bills.
- Can access table information as in free or occupied.
- Can erase table information.
- Will be able to update table information.

User profile : Kitchen staff

- Will be registered and logged in.
- Will be able to access table orders.
- Will be able to give feedback on if the order can be prepared.

User profile : Client

- Doesn't have to be registered or logged in.
- Has access only to the menu.
- Can place orders directly from the web.
- Can ask for waiters assistance.
- Can cancel orders within 5 minutes from placing it.

2.3 Assumptions

We assume that every client should have a smart phone with an implemented web browser app.

The business using this software is assumed to have a working PC in it's work environment if not employees should have smartphones.

It is assumed that the work environment has a consistent internet connection with a working WiFi available to both the staff and clients. We assume that the system to be fully running during the business working hours where orders take place and there is interaction, which will be decided by the business itself. However we expect the manager and the admin to have full access during 24 hours.

2.4 Constraints

The system may be potentially constrained by:

- Availability of smartphones with a functioning camera and internet connection
- Ability of users to be able to operate the application
- Ability of the staff to properly and efficiently work with the software.
- Server downtimes

2.5 Dependencies

This web application is dependent on a stable internet connection and the ability of the staff / clients to work with the given software. And a lot of money to buy the software.

3. Requirements

- Describe all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements.
- Organize these requirements in a way that works best for your project. See [Appendix DAppendix D, Organizing the Requirements](#) for different ways to organize these requirements.
- Describe every input into the system, every output from the system, and every function performed by the system in response to an input or in support of an output. (Specify what functions are to be performed on what data to produce what results at what location for whom.)
- Each requirement should be numbered (or uniquely identifiable) and prioritized.
See the sample requirements in Functional Requirements, and System Interface/Integration, as well as these example priority definitions:

[YourProject] Requirements Specification

Priority Definitions

The following definitions are intended as a guideline to prioritize requirements.

- Priority 1 – The requirement is a “must have” as outlined by policy/law
- Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits
- Priority 3 – The requirement is a “nice to have” which may include new functionality

It may be helpful to phrase the requirement in terms of its priority, e.g., "The value of the employee status sent to DIS **must be** either A or I" or "It **would be nice** if the application warned the user that the expiration date was 3 business days away". Another approach would be to group requirements by priority category.

- A good requirement is:
 - Correct
 - Unambiguous (all statements have exactly one interpretation)
 - Complete (where TBDs are absolutely necessary, document why the information is unknown, who is responsible for resolution, and the deadline)
 - Consistent
 - Ranked for importance and/or stability
 - Verifiable (avoid soft descriptions like “works well”, “is user friendly”; use concrete terms and specify measurable quantities)
 - Modifiable (evolve the Requirements Specification only via a formal change process, preserving a complete audit trail of changes)
 - Does not specify any particular design
 - Traceable (cross-reference with source documents and spawned documents).

3.1 Functional Requirements

In the example below, the requirement numbering has a scheme - BR_LR_0### (BR for Business Requirement, LR for Labor Relations). For small projects simply BR-## would suffice. Keep in mind that if no prefix is used, the traceability matrix may be difficult to create (e.g., no differentiation between '02' as a business requirement vs. a test case)

The following table is an example format for requirements. Choose whatever format works best for your project.

For Example:

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
RQ_1	The system supports the idea of users with login credentials.	Business Process = “Maintenance”	3	7/13/04	Bob Dylan, Mick Jagger
RQ_2	Login credentials follow specific security guidelines.	Business Process = “Changing Dues in the System” An example of a new fee is an initiation fee.	2	7/13/04	Bob Dylan, Mick Jagger
RQ_3	User passwords must be encrypted.	Business Process = “Maintenance” Some job classes are old and are no longer used.	2	7/13/04	Bob Dylan, Mick Jagger

[YourProject] Requirements Specification

		However, they still need to be maintained for legal, contract and historical purposes.			
RQ_4	Multi user level must be present in the system.	April 2005 – New requirement. It is one of three new requirements from BR_LR_03.	2		
RQ_5	A fully functional registration form linked to the database				
RQ_6	User roles need to be predefined in the registration form . Admin, Manager, Waiter, Kitchen staff, client	April 2005 – New requirement. It is one of three new requirements from BR_LR_04. 5/11/2005 – Priority changed from 2 to 3.	2 3		
Rq_7	Admin must full control over the system.				

3.2 Non-Functional Requirements

In here try to use the Structure given at slide 13 in Requirements Engineering Lecture Slides, with main categories of:

Product Requirements

Organizational Requirements

External Requirements

And their sub categories, try to arrange them in a proper visible way.

3.2.1 User Interface Requirements

In addition to functions required, describe the characteristics of each interface between the product and its users (e.g., required screen formats/organization, report layouts, menu structures, error and other messages, or function keys).

3.2.2 Usability

Include any specific usability requirements, for example,

Learnability

- The user documentation and help should be complete
- The help should be context sensitive and explain how to achieve common tasks
- The system should be easy to learn

(See <http://www.usabilitynet.org/>)

3.2.3 Performance

Specify static and dynamic numerical requirements placed on the system or on human interaction with the system:

[YourProject] Requirements Specification

- Static numerical requirements may include the number of terminals to be supported, the number of simultaneous users to be supported, and the amount and type of information to be handled.
- Dynamic numerical requirements may include the number of transactions and tasks and the amount of data to be processed within certain time period for both normal and peak workload conditions.

All of these requirements should be stated in measurable form. For example, "95% of the transactions shall be processed in less than 1 second" rather than "an operator shall not have to wait for the transaction to complete".

3.2.3.1 Capacity

Include measurable capacity requirements (e.g., the number of simultaneous users to be supported, the maximum simultaneous user load, per-user memory requirements, expected application throughput)

3.2.3.2 Availability

Include specific and measurable requirements for:

- Hours of operation
- Level of availability required
- Coverage for geographic areas
- Impact of downtime on users and business operations
- Impact of scheduled and unscheduled maintenance on uptime and maintenance communications procedures
- reliability (e.g., acceptable mean time between failures (MTBF), or the maximum permitted number of failures per hour).

3.2.3.3 Latency

Include explicit latency requirements, e.g., the maximum acceptable time (or average time) for a service request.

3.2.4 Manageability/Maintainability

3.2.4.1 Monitoring

Include any requirements for product or service health monitoring, failure conditions, error detection, logging, and correction.

3.2.4.2 Maintenance

Specify attributes of the system that relate to ease of maintenance. These requirements may relate to modularity, complexity, or interface design. Requirements should not be placed here simply because they are thought to be good design practices.

3.2.4.3 Operations

Specify any normal and special operations required by the user, including:

- periods of interactive operations and periods of unattended operations
- data processing support functions
- backup and recovery operations
- safety considerations and requirements
- disaster recovery and business resumption

3.2.5 System Interface/Integration

Specify the use of other required products (e.g., a database or operating system), and interfaces with other systems (e.g., UWHires package interfaces with PubCookie and ODS, HEPPS system interfaces with Budget system). For each interface, define the interface in terms of message format and content. For well-documented interfaces, simply provide a reference to the documentation.

[YourProject] Requirements Specification

Outline each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

3.2.5.1 Network and Hardware Interfaces

Specify the logical characteristics of each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

3.2.5.2 Systems Interfaces

Example systems interface requirements:

A. System1-to-System2 Interface

The <external party> will create and send a fixed length text file as an email attachment to System2mail@u.washington.edu to be imported into the System2 system for payroll calculation. This file must be received on EDIT day by 4:00 PM in order to be processed in the EDIT night run. The requirements below document the file specifications, data transfer process, and specific schedule. This file is referred to as "FileName" in this document.

File Structure and Format

A1. The FileName file is a fixed length text file.

A2. The FileName file is an unformatted ASCII file (text-only).

A3. The FileName file contains a batch totals record and several detail records.

File Description: Batch Totals Record

A4. The batch totals record can be placed at the beginning, in the middle, or at the end of the file.

A5. The batch totals record contains the following:

- Record Type (value: XA)
- Process Type (value: A)
- Batch Number (3 digit number assigned by Payroll Dept)
- Origin Code (AIG)
- Total number of detail records
- Total deduction amount

File Description: Detail Records

A6. The FileName file contains a row for each record meeting xxx criteria.

A7. Each row in the FileName file contains the following fields, comma-delimited and encased in double-quotes where the data includes commas or spaces:

- Employee Id
- Record Type
- Process Date (MMDDYY)
- XYG Number
- Element Code
- Amount
- Amount Sign
- Year Flag
- Total Amount
- Total Amt Sign

3.2.6 Security

3.2.6.1 Protection

Specify the factors that will protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse. For example:

- encryption
- activity logging, historical data sets
- restrictions on intermodule communications
- data integrity checks

3.2.6.2 Authorization and Authentication

Specify the Authorization and Authentication factors. Consider using standard tools such as PubCookie.

3.2.7 Data Management

Specify the requirements for any information that is to be placed into a database, including

- types of information used by various functions
- frequency of use
- data access rules
- data entities and relationships
- integrity constraints
- data retention
- valid range, accuracy, and/or tolerance
- units of measure
- data formats
- default or initial values

3.2.8 Standards Compliance

Specify the requirements derived from existing standards, policies, regulations, or laws (e.g., report format, data naming, accounting procedures, audit tracing). For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values.

3.2.9 Portability

If portability is a requirement, specify attributes of the system that relate to the ease of porting the system to other host machines and/or operating systems. For example,

- Percentage of components with host-dependent code;
- Percentage of code that is host dependent;
- Use of a proven portable language;
- Use of a particular compiler or language subset;
- Use of a particular operating system;
- The need for environment-independence - the product must operate the same regardless of operating systems, networks, development or production environments.

3.2.10 Other Non-Functional Requirements

Please provide all necessary non-functional requirements, similar to the requirements explained in the lesson slides or in the textbook.

3.3 Domain Requirements

Everything related to the domain that might be needed in the project shall be mentioned in here. Sometimes the domain Requirements might be thought as part of either functional or non-functional requirements.

4. User Scenarios/Use Cases

Provide a summary of the major functions that the product will perform. Organize the functions to be understandable to the customer or a first time reader. Include use cases and business scenarios, or provide a link to a separate document (or documents). A business scenario:

- Describes a significant business need
- Identifies, documents, and ranks the problem that is driving the scenario
- Describes the business and technical environment that will resolve the problem
- States the desired objectives
- Shows the “Actors” and where they fit in the business model
- Is specific, and measurable, and uses clear metrics for success

APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

- Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
- Supporting or background information that can help the readers of the Requirements Specification;
- A description of the problems to be solved by the system;
- Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

Appendix A. Definitions, Acronyms, and Abbreviations

Define all terms, acronyms, and abbreviations used in this document.

Appendix B. References

List all the documents and other materials referenced in this document.

Appendix C. Requirements Traceability Matrix

The following trace matrix examples show one possible use of naming standards for deliverables (FunctionalArea-DocType-NN). The number has no other meaning than to keep the documents unique. For example, the Bargaining Unit Assignment Process Flow would be BUA-PF-01.

For example (1):

Business Requirement	Area	Deliverables	Status
BR_LR_01 The system should validate the relationship between Bargaining Unit/Location and Job Class.---Comments: Business Process = "Assigning a Bargaining Unit to an Appointment" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_09 The system should provide the capability for the Labor Relations Office to maintain the job class/union relationship.---Comments: Business Process = "Maintenance" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-02 BU Assignment Rules Maint Process Flow Diagram	ReadyForReview

For example (2):

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
BR_LR_01	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_01	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_01	1	BUA	BUA-PF-01	Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_01	1	BUA	BUA-UCD-01	BU Assign LR UseCase Diagram	ReadyForReview

[YourProject] Requirements Specification

BR_LR_01	1	BUA	BUA-UCT-001	BU Assignment by PC UseCase - Add Appointment and Derive UBU	Reviewed
BR_LR_01	1	BUA	BUA-UCT-002	BU Assignment by PC UseCase - Add Appointment (UBU Not Found)	Reviewed
BR_LR_01	1	BUA	BUA-UCT-006	BU Assignment by PC UseCase - Modify Appointment (Removed UBU)	Reviewed
BR_LR_09	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_09	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_09	1	BUA	BUA-PF-02	BU Assignment Rules Maint Process Flow Diagram	Accepted
BR_LR_09	1	BUA	BUA-UCD-03	BU Assign Rules Maint UseCase Diagram	Reviewed
BR_LR_09	1	BUA	BUA-UCT-045	BU Assignment Rules Maint: Successfully Add New Assignment Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-051	BU Assignment Rules MaintUseCase: Modify Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-053	BU Assignment Rules MaintUseCase - Review Assignment Rules	Reviewed
BR_LR_09	1	BUA	BUA-UCT-057	BU Assignment Rules MaintUseCase: Inactivate Last Rule for a BU	Reviewed
BR_LR_09	1	BUA	BUA-UI-02	BU AssignRules Maint UI Mockups	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-021	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-027	BU Assignment Rules Maint TestCase: Modify Rule - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-035	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Error Condition	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-049	BU Assignment Rules Maint TestCase: Modify Rule - Error Condition	ReadyForReview

For example (3):

BizReqID	CD01	CD02	CD03	CD04	UI01	UI02	UCT01	UCT02	UCT03	TC01	TC02	TC03	TC04
BR_LR_01			X		X		X			X		X	
BR_LR_09	X			X		X			X		X		X
BR_LR_10	X			X					X		X		
BR_LR_11		X											

Appendix D. Organizing the Requirements

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

By System Mode

Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

By User Class

Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

By Objects

Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

By Feature

A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may include validity checks on inputs, exact sequencing of operations, responses to abnormal situations, including error handling and recovery, effects of parameters, relationships of inputs to outputs, including input/output sequences and formulas for input to output.

By Stimulus

Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

By Response

Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

By Functional Hierarchy

When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

Additional Comments

Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.