ROS Requirements Specification

Version 1.0

Table of Contents

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.

- 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

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	
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	
RQ_3	User passwords must be encrypted.	Business Process = “Maintenance” Some job classes are old and are no longer used. However, they still need to be maintained for legal, contract and historical purposes.	2	7/13/04	
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

3.2.1 User Interface Requirements

Our product will be a software accessible from any device that supports a search engine, camera capable of scanning QR codes and an internet connection from the client’s side. On the client’s side it will be presented as a simple web application that displays the menu and gives the opportunity to directly order from the website. This “web application” will only be visible in the cases when the user is accessing it through the QR code provided at the respective table. At the same time it will be giving to the client the opportunity to receive further assistance from the waiter or any staff member that directly works with clients.

The software will have three types of user interfaces:
• The client interface: only visible to the client and will only display the menu and the order button if not assisted by the waiter and only the menu on the other way around.

•The web application that will require a log in depending on the role of the person or division of the business. This application will be accessed through the website's url only and not through the QR code on the client's side.

•The waiter's interface: will display the table number and the order as well as the bill. This interface will also notify the waiter if a table needs their assistance. In cases when the order is directly done through the website, the waiter's interface will simply show the orders and the bill of the respective table. Each Table will have its own number or code in order to distinct them from each other. This means that each QR code will be unique regarding the others in order to identify the tables. This interface will require an identification where each waiter will have a unique identification in the system.

•The kitchen interface: this interface will be represented by one single user identification in the system without the need to differentiate between the kitchen staff like in the waiters' case because there is no separation regarding the service in the kitchen.

•The manager interface: this interface will only be accessible if the user is the manager of the business. It will be the only user that is allowed to make changes to the menu update, add or delete different items.

3.2.2 Usability

- ☐ The software is easy to understand and use.
- ☐ In case of an error the software will give hints and will provide a specific solution depending on the case.
- ☐ There will be restricted actions based on the specific role of the active user.
- ☐ In order to be fully understood by the user and administrator the software will be available with a documentation showing different steps on how to use the software.

3.2.3 Performance

Our software consists of a web application stored on a web server. We can agree that our performance will be evaluated based on the time and space complexity created by our software.

Example :

- ☐ The number of users entering the website simultaneously.
- ☐ The time used to fetch and send data to the database.
- ☐ The internet connection on the areas that use the software (Not on our hand).
- ☐ The hardware equipment provided by the business owner.
- ☐ Population of our database based on the number of users ordering items from the menu.

3.2.3.1 Capacity

Our web application as mentioned above will be stored in a web server. Since this software only consists of a given menu, we will not need a lot of extra space to do changes on the system. So our software capacity will be relatively small (as for now), we expect that the system will work just fine with the given flux of clients that are online.

3.2.3.2 Availability

- ☐ Our softwares working hours will be 24/7.
- ☐ The application will give us the ability to have different interfaces for different users.
- ☐ In case of not being able to use the software it will give us an error message preventing further bad usage of the application.

- ☐ The software will not only be available for the physical clients on the business but also for different clients that do not have the opportunity to be there physically.
- ☐ As long as a client/administrator has a stable internet connection he will be able to access the web application.

3.2.3.3 Latency

Our softwares latency will be based on :

- ☐ The capacity of the database.
- ☐ The internet connection of the client.
- ☐ The performance of our database to fetch and send data.
- ☐ The time to search for different items of the menu.

3.2.4 Manageability/Maintainability

3.2.4.1 Monitoring

The user interface of the software is supposed to be simple and easy to understand, because clients don't have to register or log in. On the other hand the administrator, manager and waiter should create an account, after that they can access their account by entering 1) username and 2) password as an input. After these two credentials are inputted they will be validated. In case of an error we will get a "Invalid Input !" message, if not they can successfully log in to their account.

3.2.4.2 Maintenance

//Dito mreza shikoje iher kur t kesh kohe.

3.2.4.3 Operations

Operations available for the user are :

- ☐ Search for an item.
- ☐ Order an item.
- ☐ Delete an item from the order.
- ☐ Require the waiters help.
- ☐ Leave reviews.
- ☐ Clear the tables.
- ☐ Cancel an order.
- ☐ Insert new items from the menu.
- ☐ Delete old items from the menu.
- ☐ Block/unblock waiters and managers account (Admin)
- ☐ Create or delete a waiters or managers account (Admin)

3.2.5 System Interface/Integration

Our database will be read only for each user except the administrator that will be able to change different configurations (explained later). So we can say that the initial structure of the database will not be changed till a later moment.

3.2.5.1 Network and Hardware Interfaces

We are looking at a basic TCP connection with the web server, where of course our web application is directly connected to this host. We can say that this step wouldn't give us any troubles.

3.2.6 Security

3.2.6.1 Protection

Even though this software would never create protection and security problems (since clients are not creating an account) we would have to cover the administrator, manager and kitchen staff accounts. On this note no user will be able to see passwords inserted from another person.

3.2.6.2 Authorization and Authentication

- ☐ Credentials will be checked when logged in.
- ☐ Each user will only have access to their personal interface.
- ☐ We are using the PubCookie tool.
- ☐ Different users will have different authorization actions based on their roles.

3.2.7 Data Management

Data will be managed by our database and some of the sections will be:

- ☐ Clients
- ☐ Admin
- ☐ Kitchen
- ☐ Manager
- ☐ Items
- ☐ Orders
- ☐ Bills

3.2.8 Standards Compliance

The softwares policy will be on record and will follow all the regulations and laws that are required. In order to do so we will :

- ☐ Put a record on operations and actions taken from the admin and database (payrolls etc).
- ☐ 24/7 availability of information on different actions required by the authorities.
- ☐ An audit will be provided by the end of the first three months after the product is launched.

3.2.9 Portability

This application can be accessible by any device using a stable internet connection and a browser, beside the fact that the services of the business can be provided also to a client that is not there for the moment. This can be possible by the given QR code which can also be shared from you to your friends, family etc.

3.2.10 Other Non-Functional Requirements

//Ardit shiko dhe kto

3.3 Domain Requirements

The plan presented by our team is mainly related to businesses like restaurants or different bar cafes, but we can see a more spread usage of the software, for example in : dropshipping, business etc. Also this software is developed in accordance with the basic safety and performance standards.
Steps for this software requirements (client based) :

- ☐ Creation of the order.
- ☐ Selecting preferred items from the menu.
- ☐ Status checking (Processing the order / Order ready / Order canceled).
- ☐ Finalization of the order (Bill request / Bill payment).

4. User Scenarios/Use Cases

Successful menu display to the client

-The client scans the QR code
-The menu is properly displayed
-A pop up is displayed to the client asking for waiter's assistance

Clients chooses waiter's assistance

-Client asks for waiter's assistance and popup disappears
-A notification is sent to the main computer notifying the respective waiter of the table their help is needed at the respective table
-The menu is displayed without the the ordering option and the client orders in the old fashion way
-The order is placed by the waiter and then it is displayed in the kitchen interface

Client chooses to order directly from the website

-Client declines the waiter's assistance in the popup
-The popup disappears
-The menu is properly displayed with the order option beside each item
-The client orders the liked items from the menu and presses the place order button
-The successful order message is displayed
-The order is displayed in the waiter's table interface 5 mins after placing it
-The order is displayed in the kitchen interface 5 mins after placing it.

Client decides to cancel and remove items from the order after placing it

-After clients places order they click on the cancel button
-In the order section the client removes the items they don't desire to order
-The clients clicks on the place order button
-The order is displayed in the waiter's interface 5 mins after placing it
-The order is displayed in the kitchen's interface 5 mins after placing it

Client decides to cancel the order and change it

-Within the 5 min mark after placing the order the client clicks on the cancel order button
-In the order section the clients removes the items the don't desire to order
-The client returns to the menu and clicks the order button the items they desire
-The clients clicks on the place order button
-The order is displayed in the waiter's interface 5 mins after placing it
-The order is displayed in the kitchen's interface 5 mins after placing it

Client is ready to pay

-Client clicks on checkout button from the order display
-A notification pops up in the waiter's interface on the respective table number
-After the payments is done the waiter clicks on discard button on the table number

The waiter checks the tables' orders

-All tables numbers are displayed to the waiter
-The ones with an active order have a mark on their number
-Waiter clicks on a table and its order is displayed

A waiter assists a client in placing an order

-A client requires the waiter's assistance on the order placement
-A notification pops up in the waiter's interface on the respective table number
-The waiter takes the order manually in the old fashion way
-The waiter logs the order on the system manually and places it
-The order is displayed in the kitchen's interface 5 mins after placing it

The manager changes the menu

-The manager logs in with his respective credentials into the system
-Manager clicks on menu button
-The menu is displayed to the manager
-Manager clicks edit menu button
-Manager clicks on add item button
-New item interface is displayed and the manager adds a new item to the menu
-Manager can click remove item button in order to remove a certain item from the menu

Manager removes or adds a waiter or a kitchen staff member

-Manager is logged in into his account
` -Manager clicks on the staff button
-Manager decides to add or remove a staff member from the respective section

Manager closes the shift

-Manager is logged in into his account
-Manager checks the status of each table on the restaurant
-Manager checks the bills and the balances per items (inventory)
-Manager closes the system for the day

Admin edits a manager user profile

-Admin is logged in into his account
-Admin selects the managers profile
-Admin decides if he wants to delete/insert/block a manager(basically the admin can edit everything he wants related to the manager)

UC Name	*UC code and name goes here*
Summary	*UC brief summary*
Dependency	*This optional section describes whether the UC depends on other UCs.*
Actors	*This section names the actors in the use case. There is always a **primary actor** that initiates the use case. In addition, there may be secondary actors that also participate in the use case*
Preconditions	*One or more conditions that must be true at the start of use case, from the perspective of this use case.*
Description of the Main Sequence	• *Step 1: Details about step 1 here.* • *Step 2: Details about step 2 here.* • *Step 3: …* • *…*
Description of the Alternative Sequence	• *Step 1: Details about step 1 here.* • *Step 2: Details about step 2 here.* • *Step 3: …* • *…*

Non functional requirements	*Narrative description of nonfunctional requirements, such as performance and security requirements.*
Postconditions	*Condition that is always true at the end of the use case (from the perspective of this use case) if the main sequence has been followed*

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

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
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.