

# **ROS Requirements Specification**

## **Version 1.0**

# Table of Contents

<b>1. EXECUTIVE SUMMARY</b>	<b>3</b>
1.1 PROJECT OVERVIEW	3
1.2 PURPOSE AND SCOPE OF THIS SPECIFICATION	3
<b>2. PRODUCT/SERVICE DESCRIPTION</b>	<b>3</b>
2.1 PRODUCT CONTEXT	3
2.2 USER CHARACTERISTICS	3
2.3 ASSUMPTIONS	3
2.4 CONSTRAINTS	3
2.5 DEPENDENCIES	4
<b>3. REQUIREMENTS</b>	<b>4</b>
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
<b>4. USER SCENARIOS/USE CASES</b>	<b>9</b>
<b>APPENDIX</b>	<b>10</b>
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/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.

## ***[ROS] Requirements Specification***

- 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.
- Can ask for checkout at the end of his stay

### **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:

**[ROS] Requirements Specification**

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
RQ_1	The system supports the idea of users with and without login credentials.				
RQ_2	User roles need to be predefined in the registration form . Manager, Waiter, Kitchen staff, client				
RQ_3	Software should provide distinct interfaces for each level of user				
RQ_4	User accounts should be secured with passwords for the more permanent ones				
RQ_5	The system allows the users to update and change their credentials				
RQ_6	Admin has full control over the system.				
RQ_7	Manager, Waiter and Kitchen staff are able to log in to the system using their credentials.				
RQ_8	In case of log in failure due to incorrect credentials, the user will be asked to log in again.				
Rq_9	The system will provide the client with the option to decide if he needs the help of the waiter or not.				
RQ_10	If the client chooses to call the waiter , the waiter will be notified .				
RQ_11	If the client is assisted by the waiter, the order is placed by the waiter.				
RQ_12	If the client chooses to order through the device, he shall be able to select what he wants to order and proceed to confirmation.				

**[ROS] Requirements Specification**

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
RQ_13	The system shall provide client permission to cancel the order within 5 minutes.				
RQ_14	The system shall provide the authorized users : Waiter and Kitchen Staff , with the proper GUI customized for their role.				
RQ_15	The system will provide the waiter the ability to access table orders and bills.				
RQ_16	The system will provide the waiter the ability to input information about the availability of the tables.				
RQ_17	The system shall provide the kitchen staff the ability to automatically view the orders done by clients through devices.				
RQ_18	The system shall provide the kitchen staff the ability to decide if the order can be prepared or not.				
RQ_19	The system shall provide the manager/admin the ability to help the staff with the proceeding of the orders.				
RQ_20	The system shall provide the manager/admin the permission to access the table bills.				
RQ_21	The system shall provide the manager/admin the permission to add or remove(change) different staff positions.				
RQ_22	The system shall prove the manager/admin the ability to remove customers from the system				
RQ_23	The system shall provide the manager/admin with the ability to add , update , or delete menu items and close the shift.				
RQ_24					

## **[ROS] Requirements Specification**

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
RQ_25					
RQ_26					

### **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

The application admin will need to maintain the server by doing regular checks on the web server, taking feedback from the users to fix any occurring bugs in the application be it in the front or the back end. The database needs to be regularly updated to improve performance as the more the application will be used the more populated the database will become, so it needs to be normalized as best as possible and queries need to be optimized using relational algebra.

#### 3.2.4.3 Operations

Operations available for the user are :



## **[ROS] Requirements Specification**

- ☐ 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**

## **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

### **[ROS] Requirements Specification**

- 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	<i>RQ_1-User logs in</i>
Summary	<i>users that need log in credentials have a name and password</i>
Dependency	<i>none</i>
Actors	<i>Admin/Manager, New user</i>
Preconditions	<i>website is available, database is available, there is internet connection, registration must have been done by the manager</i>

Description of the Main Sequence	<ul style="list-style-type: none"><li>• <i>Each user needs to have an active job contract in order to be registered</i></li><li>• <i>Each user has a username and password</i></li><li>• <i>Each user should have an active email address</i></li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Users should have a set of attributes like:</li><li>• Name and Surname</li><li>• Identification number (passport number)</li><li>• Active email</li></ul>
Non functional requirements	<i>A validation is made for the entered credentials such as the name and surname, username, id, credible email and password</i>
Postconditions	<i>A new user is logged in</i>

UC Name	<i>RQ_2-User roles understanding</i>
Summary	<i>The system determines the user role upon registration</i>
Dependency	<i>none</i>
Actors	<i>Admin/Manager, New user</i>
Preconditions	<i>none</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>● Functionalities provided to the user are determined by the system upon authentication</li><li>● User information is stored in the domain identity store</li><li>● An anonymous user is a user that doesn't need log in credentials</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>● clients are anonymous users that do not need credential validation or log in credentials</li><li>● Clients are temporary users</li></ul>
Non functional requirements	<ul style="list-style-type: none"><li>● Credentials are validated when entered</li><li>● Each user gets access only to their respective data and system functionalities</li><li>● Sessions will be used for the currently logged user</li></ul>
Postconditions	User's role is understood

UC Name	<i>RQ_3-Distinct interfaces</i>
Summary	<i>The system provides distinct interfaces for each user whether they are logged in or not. The system doesn't provide the more permanent information and interfaces to the anonymous users. It also doesn't provide intervening among users.</i>
Dependency	<i>RQ_2, a user is logged in/client visiting the webpage</i>
Actors	<i>Admin/Manager, New user, Anonymous users</i>
Preconditions	<i>User must be registered, RQ_1</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>● Each user gets a limited amount of information based on their role</li><li>● This information is mounted on distinct interfaces</li><li>● The user will only be presented with the respective interface if they are logged in, this excludes the anonymous user</li><li>● Anonymous users get a standard interface</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>● Logged users will be able to access their profile</li><li>● They won't be able to access each other's profiles</li><li>● Only the manager/admin will be able to access other users profiles</li><li>● Any user can be anonymous and will be shown the same interface for all</li></ul>

*[ROS] Requirements Specification*

Non functional requirements	<ul style="list-style-type: none"><li>• Credentials are validated upon log in</li><li>• Depending on user type the authorization is made</li><li>• Sessions will be used for the currently logged user</li><li>• Information is fetched from the database</li><li>• Users can only access their information</li></ul>
Postconditions	Each user gets a distinct interface

UC Name	<i>RQ_4-Secure account with password</i>
Summary	<i>A staff member can only log in through his password</i>
Dependency	<i>RQ_1</i>
Actors	<i>Admin/Manager, New user</i>
Preconditions	<i>User must be registered and have an active account</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• The system allows a user to secure his account through a password</li><li>• Log in can only be done through password</li></ul>



*[ROS] Requirements Specification*

Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Incorrect password doesn't allow a user to log in</li></ul>
Non functional requirements	<ul style="list-style-type: none"><li>• <i>Users aren't able to see each other passwords</i></li><li>• <i>Passwords are encrypted in the database</i></li></ul>
Postconditions	Access account only through a password

UC Name	<i>RQ_5-Password update</i>
Summary	<i>User can change the password and update it</i>
Dependency	<i>RQ_1</i>
Actors	<i>Admin/Manager, New user</i>
Preconditions	<i>User must be registered and have an active account</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• The system provides the user to update and change the password</li><li>• update the password</li><li>• save the password</li></ul>

*[ROS] Requirements Specification*

Description of the Alternative Sequence	<ul style="list-style-type: none"><li>● Reset user password</li><li>● User uses the new password at next log in</li></ul>
Non functional requirements	<ul style="list-style-type: none"><li>● <i>Users aren't able to see each other passwords</i></li><li>● <i>They can only access their information</i></li></ul>
Postconditions	Update and save password

UC Name	<i>RQ_6-Admin has full control</i>
Summary	<i>The admin can register or ban users from the system. Admin can access other users' profiles</i>
Dependency	<i>none</i>
Actors	<i>Admin/Manager, New user</i>
Preconditions	<i>none</i>

*[ROS] Requirements Specification*

Description of the Main Sequence	<ul style="list-style-type: none"><li>• Admin has the ability to add or remove users from the system</li><li>• Admin cannot access other users' passwords</li><li>• Admin views other users' profiles</li><li>• Admin can make changes to the anonymous user interface</li><li>• Admin has access upon finances and management aspects of the system</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• There can only be one admin</li><li>• Admin can also change his credentials</li><li>• When changing admins credentials are set to default</li></ul>
Non functional requirements	<ul style="list-style-type: none"><li>• Authorization will be done based on the user type</li><li>• Admin can't view other users' passwords</li></ul>
Postconditions	Admin can easily access all the system without violating privacy

UC Name	<i>RQ_7-Staff and the manager log in using their credentials</i>
Summary	<i>Each of the users that need log in will be able to enter the system by using their credentials. These types of users are not allowed to log in if the password is incorrect or if they aren't registered. If the user is authenticated they will be presented with a specific interface, customized to their role</i>

[ROS] Requirements Specification

Dependency	<i>RQ_1,RQ_2,RQ_4</i>
Actors	<i>Admin/Manager, Waiter, Kitchen staff</i>
Preconditions	<i>User must be registered and have an active account</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>•</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Reset user password</li><li>• User uses the new password at next log in</li></ul>
Non functional requirements	<ul style="list-style-type: none"><li>• <i>Users aren't able to see each other passwords</i></li><li>• <i>They can only access their information</i></li></ul>
Postconditions	Update and save password

UC Name	<i>RQ_8 -Staff and managers log in credentials are wrong, so they have to try again</i>
Summary	<i>After the staff and manager try to log in, the web application gives a warning when the credentials are wrong. So in this case they are obligated to try again, in case they want to enter their account.</i>

[ROS] Requirements Specification

Dependency	RQ_7
Actors	<i>Admin/Manager, Waiter, Kitchen staff</i>
Preconditions	<i>User must be registered and have an active account, and the credentials imputed are wrong</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• The system provides warnings due to incorrect credentials and opportunity to retry</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Reset user password</li><li>• User uses the new password at next log in</li></ul>
Non functional requirements	<ul style="list-style-type: none"><li>• <i>Users aren't able to see each other tries to log in</i></li><li>• <i>They can only see their failure log ins</i></li></ul>
Postconditions	Create new password and save new password

UC Name	<i>RQ_9 -System gives the user the opportunity to get help from the user</i>
Summary	<i>The first moment that the user uses the web application he is given the opportunity to get help by the waiter. This</i>

[ROS] Requirements Specification

	<i>will be provided by a message box which is going to get a response from the user.</i>
Dependency	<i>RQ_2, RQ_7</i>
Actors	<i>Waiter, user</i>
Preconditions	<i>Waiter must be logged in into his account and new user should be able to open and use the web application</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• User has the ability to create a request to get the waiters help</li><li>• User cannot get the help of non active or not logged in waiters</li><li>• Waiter cannot get the request from the system if not logged in</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Incorrect use of the software will not end in successful help from the staff</li></ul>
Non functional requirements	<ul style="list-style-type: none"><li>• <i>Clients are not able to see which other client can make a request</i></li><li>• <i>Client can see their request</i></li></ul>
Postconditions	Wait for the staffs response to the request

UC Name	<i>RQ_10 -In case the client calls the waiter, he will be notified</i>
Summary	<i>In the first moment that the user uses the message box to call the waiter on the given interface, the waiter will be notified on his profile along with the details of the table</i>
Dependency	<i>RQ_9</i>
Actors	<i>Waiter, new user</i>
Preconditions	<i>Waiter must be logged in to be able to get the notification from the user on the table</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• User gives the call to the waiter</li><li>• Waiter is able to get the notification through his profile</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• In case the waiter is not logged in he cannot get the call</li></ul>
Non functional requirements	<ul style="list-style-type: none"><li>• <i>Other waiters are not able to see the call the current waiter</i></li><li>• <i>The waiter can only see his calls</i></li></ul>

Postconditions	Waiter gives a response due to his personal work condition
----------------	--

UC Name	<i>RQ_11-If the client is assisted, the waiter sets the order for the table</i>
Summary	<i>When the waiter serves the client there will be a order set by the waiter to update the information of the table</i>
Dependency	<i>RQ_10</i>
Actors	<i>Waiter, kitchen staff</i>
Preconditions	<i>Waiter should be logged in</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• Waiter sets the order to the staff</li><li>• Waiter updates the information of the table</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• In case any member of the staff is not logged in then the order cannot be placed</li></ul>
Non functional requirements	<i>The relation between the waiter and the staff is unique and cannot be updated by any other waiter</i>



Postconditions	The staff gives a feedback on the order if it is executable or not
----------------	--

UC Name	<i>RQ_12 -If the client wants to order he shall be able to select the items</i>
Summary	<i>When the client decides to order from the menu from the software, he is going to be able to select any item from the menu that is available</i>
Dependency	<i>RQ_7, RQ_9, RQ_10, RQ_11</i>
Actors	<i>Waiter, kitchen staff, user</i>
Preconditions	<i>Waiter should be logged in, user should be using the web application</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• The menu is shared the same through all the users</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Only the user and the waiter can see the clients order</li></ul>

Non functional requirements	<i>The order between the user and the waiter is unique</i>
Postconditions	The staff gives a feedback if the order can be set

UC Name	<i>RQ_13 -The system will provide the client 5 minutes to cancel the given order</i>
Summary	<i>After the client gets the help from the waiter and decides on the order it creates a list of items that he wishes to get, after that he gets a 5 minute window to cancel the order</i>
Dependency	<i>RQ_12</i>
Actors	<i>Waiter</i>
Preconditions	<i>User need to create a list of items that he wants to order</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• The waiter has the items and the time from the begging of the order</li></ul>

*[ROS] Requirements Specification*

Description of the Alternative Sequence	<ul style="list-style-type: none"><li>Only the waiter of the table can see the contents of the order and the time left</li></ul>
Non functional requirements	<i>The time that is available for the user is not resettable</i>
Postconditions	After the time has started, the order will start

UC Name	<i>RQ_14 -The system will provide each member of the staff with its own</i>
Summary	<i>After everything is set from the client side the other members of the staff can log in into their accounts with them own GUI</i>
Dependency	
Actors	<i>Waiter, kitchen staff, admin</i>
Preconditions	<i>All staff members should be logged in into their accounts</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>Staff members logged in into their personal accounts</li></ul>

*[ROS] Requirements Specification*

Description of the Alternative Sequence	<ul style="list-style-type: none"><li>Each staff member has a personal GUI representation of their account</li></ul>
Non functional requirements	<i>Each interface for each user is unique and also no one else can see the others interface</i>
Postconditions	The staff is logged in and waiting for the next orders

UC Name	<i>RQ_15 -Waiter order accessibility</i>
Summary	<i>Waiters can access the status and contents of orders and bills of each table</i>
Dependency	<i>RQ_14</i>
Actors	<i>Waiter</i>
Preconditions	<i>Waiter should be logged in</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>Waiter selects tables to view their orders and bill</li></ul>

*[ROS] Requirements Specification*

Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Orders and bills are displayed with all their info</li></ul>
Non functional requirements	<i>Orders and bills should have a status</i>
Postconditions	

UC Name	<i>RQ_16 -Table status manipulation</i>
Summary	<i>Waiters can change the the status of tables in the system to show if they are available or not</i>
Dependency	<i>RQ_14</i>
Actors	<i>Waiter</i>
Preconditions	<i>Waiter should be logged in</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• Waiter change the status of the table</li></ul>

[ROS] Requirements Specification

Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Tables are available by default</li></ul>
Non functional requirements	
Postconditions	Table status is set and can be changed

UC Name	<i>RQ_17 - Kitchen staff orders</i>
Summary	<i>The kitchen staff can see on their interface the the orders put in by their customers</i>
Dependency	<i>RQ_14</i>
Actors	<i>kitchen staff, client, waiter</i>
Preconditions	<i>Kitchen staff must be logged in and orders must be in the system</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• Customers order on their devices and kitchen staff view.</li></ul>

*[ROS] Requirements Specification*

Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Waiter puts in the order and the kitchen staff can see it</li></ul>
Non functional requirements	
Postconditions	The staff is logged in and waiting for orders to view

UC Name	<i>RQ_18 -Order preparation</i>
Summary	<i>If the kitchen staff cannot prepare a specific order they can notify si in the system</i>
Dependency	<i>RQ_14, RQ_17</i>
Actors	<i>Waiter, kitchen staff, customers</i>
Preconditions	<i>All staff members should be logged in into their accounts and orders need to be in the system</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• Kitchen staff notify the system the availability of the order</li></ul>

[ROS] Requirements Specification

Description of the Alternative Sequence	
Non functional requirements	<i>The order items are part of the menu</i>
Postconditions	The kitchen staff is logged in and waiting for orders

UC Name	<i>RQ_19 -The system will provide each member of the staff with its own</i>
Summary	<i>After everything is set from the client side the other members of the staff can log in into their accounts with them own GUI</i>
Dependency	
Actors	<i>Waiter, kitchen staff, admin</i>
Preconditions	<i>All staff members should be logged in into their accounts</i>



*[ROS] Requirements Specification*

Description of the Main Sequence	<ul style="list-style-type: none"><li>• Staff members logged in into their personal accounts</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Each staff member has a personal GUI representation of their account</li></ul>
Non functional requirements	<i>Each interface for each user is unique and also no one else can see the others interface</i>
Postconditions	The staff is logged in and waiting for the next orders

UC Name	<i>RQ_20 - Manager table bills accessibility</i>
Summary	<i>The manager/admin can access the bills of each table in the system</i>
Dependency	<i>RQ_14</i>
Actors	<i>manager/admin</i>

*[ROS] Requirements Specification*

Preconditions	<i>Manager needs to be logged in and bills should be in the system</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• Manager selects a table to view it's bill</li></ul>
Description of the Alternative Sequence	
Non functional requirements	<i>A bill should be available for the selected table</i>
Postconditions	The staff is logged in and waiting for the next orders

UC Name	<i>RQ_21 -Staff management</i>
Summary	<i>The manager/admin should be able to change the status of staff members in the system and change their work position</i>
Dependency	<i>RQ_14, RQ_2, RQ_6</i>

*[ROS] Requirements Specification*

Actors	<i>manager/admin, waiter, kitchen staff</i>
Preconditions	<i>The manager/admin is logged in and users have defined roles</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• Manager/admin selects a staff member to change their status</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Manager select a staff member to remove them</li></ul>
Non functional requirements	
Postconditions	

UC Name	<i>RQ_22 - Client management</i>
Summary	<i>Manager/admin can remove a client from the system</i>
Dependency	<i>RQ_14, RQ_2, RQ_6</i>
Actors	<i>admin, client</i>

*[ROS] Requirements Specification*

Preconditions	<i>Manager/Admin is logged in and client exists in the system</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• Manager/Admin removes client from the system</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Manager/Admin add</li></ul>
Non functional requirements	<i>Each interface for each user is unique and also no one else can see the others interface</i>
Postconditions	The staff is logged in and waiting for the next orders

UC Name	<i>RQ_23 - Menu and shift management</i>
Summary	<i>Manager/admin can manage the menu and also shifts</i>
Dependency	<i>RQ_14, RQ_2, RQ_6</i>
Actors	<i>admin, waiter, kitchen staff</i>

***[ROS] Requirements Specification***

Preconditions	<i>Manager/Admin is logged</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• Manager/Admin adds/removes item from menu</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• Manager/Admin closes a shift</li></ul>
Non functional requirements	
Postconditions	