Table of Contents

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience and Reading Suggestion 1

1.3 Product Scope 1

1.4 References 2

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 3

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3. External Interface Requirements 5

3.1 User Interfaces 5

3.2 Hardware Interfaces 5

3.3 Software Interfaces 5

3.4 Communications Interfaces 6

4. System Features 6

4.1 System Feature 1 6

4.2 System Feature 2 (and so on) 7

5. Other Nonfunctional Requirements 10

5.1 Performance Requirements 10

5.2 Safety Requirements 10

5.3 Security Requirements 11

5.4 Software Quality Attributes 11

Software Requirements Specification

For

Restaurant Management System

Deep Dutta and Kawser Morshed

Computer Science and Engineering Discipline, Khulna University

# Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

## 1.1Purpose

The purpose of this document is to give a detailed description of the requirements for the “Restaurant Management System (RMS)”. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications.

## 1.2Intended Audience and Reading Suggestions

The document is intended for different types of reader such as project manager, users, testers and documentation writers. The document will be included overall description of the system and their functionalities, user classes, product functions, operating environment, design, external interface and requirements, implementation, system features. The readers should read the documents step by step. The document also included the user interaction of that system.

## 1.3Product Scope

There are different types of Restaurant in Khulna city. But they have lack of online services and their system is not flexible. Many outsider come to the city but he/she do not know the location of different restaurant and their food items around. For this regard he/she should use our system and can easily find the location of the nearby restaurant and their food items. The system will show the facilities that provided to the guest and can also see the present status of the restaurant. The system will be user friendly and can reduce the efforts of the user. Our goal is to design a Restaurant Management System that will be help to the user to find the food list and they can order the restaurant food easily.

## 1.4References

[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

[2]<https://www.tutorialspoint.com/software_testing_dictionary/software_requirement_specification.htm>

## 2.1 Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

## 2.2 Product Perspective

Restaurant Management System is a very useful and timesaving system for those who like to eat different types of restaurant foods in home and also going to the restaurant. Another perspective of this system is ordering foods in online from anywhere in the city. User can see the price of the foods he/she will order and can easily make a menu card with their desired food.

## 2.3 Product Features

Restaurant Management System provides users the following functions/features:

* List of Restaurant under a city corporation.
* Users can order foods by online.
* User can create menu card with their desired foods.
* User can see the price of the foods that he/she will order.
* Review the food items by user.

## 2.4 User Class and Characteristics

Users of the system should be able to get information of restaurant under a city corporation with the given date/time of ordering from the database.  The system will support two types of user privileges, Customer, and Administrator. Customers will have access to customer functions, and the Administrator will have access to both customer and booking information functions. The customer should be able to do the following functions:

* Make a menu card.
* Select desire food items.
* Ordre foods.
* Flexible Date/time.

* Cancel an existing food items.
* Confirmation.

* ADMINISTRATIVE
* Add/Delete food items.
* Add a new food.
* Update price of the foods.
* Update order processing.

## 2.5 Operating Environment

Operating environment for the Restaurant management system is as listed below.

* Distributed Database.
* Client/server System.
* Operating system: Windows.
* Database: My SQL Database.
* Platform: php.net.

## 2.6 Design and Implementation Constraints

1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications
3. How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.
4. Implement the database at least using a centralized database management system.

## 2.7 Assumptions and Dependencies

Let us assume that this is a Restaurant Management System and it is used in the following application:

* A request for ordering/cancellation of food items from any source to any destination.

# External Interface Requirements

**3.1 User Interfaces**

* Front-end software: Php.net.
* Back-end software: My SQL

**3.2 Hardware Interfaces**

* Windows.
* A browser which supports PHP, HTML & JavaScript.

**3.3 Software Interfaces**

Following are the software used for Restaurant Management System.

|  |  |
| --- | --- |
| **Software used** | **Description** |
| Operating system | We have chosen Windows operating system for its best support and user-friendliness. |
| Database | To save the order, customer records we have chosen My SQL database. |
| PHP.Net | To implement the project we have chosen PHP.Net language for its more interactive support. |

**3.4 Communications Interfaces**

This project supports all types of web browsers. We are using simple electronic forms for the ordering list.

# 4.System Features

**4.1.1 Stimulus**

* Search for a restaurant for ordering foods.
* Displays a detailed list of food items and the menu card and also the cancel item from an existing menu card.

**4.1.2 Functional Requirements**

Use case determination

* Use case: Registration(User)

Actor: User.

Functionalities:

1. Entered into the system.
2. Click on registration button.
3. Fill up the text boxes named name, username, password , address, email, contact number .
4. Enter submit button.

* Use case: Login(User)

Actor: User

Functionalities:

1. Entered into the system.
2. Click on the login button.
3. Fill up the text boxes named username, password.
4. Click on login button.

* Use case: Apply for checking food list.

Actor: User

Functionalities:

1. Login into the system.
2. Click on food list button.
3. Fill up the text boxes name, address, time, date and food name .
4. Click on ordering button.

* Use case: Seeing facility details.

Actor: User.

Functionalities:

1. Login into the system.
2. Click on the about button.
3. Seeing facility details.

* Use case: Adding and Deleting food from menu card.

Actor: User.

Functionalities:

1. Login into the system.
2. Click on add button to add food.
3. Click on the delete button to delete.
4. Enter the submit button.

* Use case: Review to the service.

Actor: User.

Functionalities:

1. Login into the system.
2. Click on the review button.
3. Click on the submit button.

* Use case: Checking the order list.

Actor: Admin.

Functionalities:

1. Login into the system.
2. Click on the order button.
3. See the list.

* Use case: Confirmation the request

Actor: Admin.

Functionalities:

1. Login into the system.
2. Click on confirmation button.

* Use case: Checking the review .

Actor: Admin.

Functionalities:

1. Login into the system.
2. Click on Review status
3. See the list according to serial wise.

* Use case: Checking delivery details.

Actor: Admin.

Functionalities:

1. Login into the system.
2. Click on the delivery details button.
3. Check the details.

* Use case: Update any notice or news.

Actor: Super admin.

Functionalities:

1. Login into the system.
2. Click on the notice.
3. Add the certain file or attachment or text into the box.
4. Click on update button.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

**5.2 Safety Requirements**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

**5.3 Security Requirements**

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

**5.4 Software Quality Attributes**

* **Availability:** The restaurant should be available on the specified date and specified time as many customers are doing advance ordering.
* **Correctness:** The restaurant should strict to their rules and regulation.
* **Maintainability:** The administrators and customers should maintain correct schedules of time.
* **Use ability:** The facilities should satisfy a maximum number of customer needs.