Software Requirements Specification

for

Restaurant Automation System

Prepared by Akankshya Patra

115CS0231

National Institute of Technology Rourkela

25th January 2019

Table of Contents

1. Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS purpose, scope, definitions, acronyms, abbreviations, references and overview of SRS. A SRS is a complete description of the behaviour of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains nonfunctional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints). This is a documentation of a RAS has for automating various restaurant activities.

1.1 Purpose

The Software Requirements Specification (SRS) will provide a detailed description of the requirements for the Restaurant Automation System (RAS). This SRS will allow for a complete understanding of what is to be expected from the newly introduced system which is to be constructed. The clear understanding of the system and its' functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project. This SRS will provide the foundation for the project. From this SRS, the Restaurant Automation System can be designed, constructed, and finally tested.

The origin of most software systems is in the need of a restaurant owner, who either wants to automate the existing manual system or desires a new software system. The software system is itself created by the developer. Finally, the end user will use the completed system. Thus, there are three major features in this new system: managing prices, processing orders, maintaining ingredients, maintaining invoice data and generating receipts. Somehow the requirements for the system that will satisfy the needs of the owner and the concerns of the users have to be communicated to the developer. The problem is that the owner doesn't usually design the software or the software development process and the developer does not understand the owner's problem and the application area. This causes a communication gap between the parties involved in the development of the project.

The basic purpose of Software Requirement Specification (SRS) is to bridge this communication gap. SRS is the medium through which the owner's and the restaurant's needs are accurately specified indeed SRS forms the basis of software development.

Another important purpose of developing a SRS is to help the owner in understanding his own needs. A SRS establishes the basis for agreement between the owner and the restaurant on the working of the software product.

A SRS provides a reference for validation of the final product. A high quality SRS is a prerequisite to high quality software and it also reduces the development cost. The system that will be developed will address the following issues:

- Faster System
- Reliability
- Accuracy
- Informative
- Tracking failure issues

1.2 Product Scope

A restaurant owner wants to computerize his order processing, billing, and accounting activities. He also expects the computer to generate statistical report about sales of different items. A major goal of this computerization is to make supply ordering more accurate so that the problem of excess inventory is avoided as well as the problem of non-availability of ingredients required to satisfy orders for some popular items is minimized. The computer should maintain the prices of all the items and also support changing the prices by the manager. Whenever any item is sold, the sales clerk would enter the item code and the quantity sold. The computer should generate bills whenever food items are sold. Whenever ingredients are issued for preparation of food items, the data is to be entered into the computer. Purchase orders are generated on a daily basis, whenever the stock for any ingredient falls below a threshold value. The computer should calculate the threshold value for each item based on the average consumption of this ingredient for the past three days and assuming that a minimum of two days stock must be maintained for all ingredients. Whenever the ordered ingredients arrive, the invoice data regarding the quantity and price is entered. If sufficient cash balance is available, the computer should print cheques immediately against invoice. Monthly sales receipt and expenses data should be generated whenever the manger would request to see them. The computer should be able to print a menu card containing the menu items and their prices.

1.3 References

- Fundamentals of Software Design by Rajib Mall.

1.4 Overview of the Developer's Responsibility

This project is a prototype for the Restaurant Automation System and it is restricted within the college premises. This will be implemented under the guidance of college professor. This project is useful for the Restaurant owners of a large Restaurants and as well as to the users visiting the restaurants of the

city. The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. General description of the project is discussed in section 2 of this document. Section 3 gives the functional requirements, data requirements and constraints and assumptions made while designing the Software. It also gives the user viewpoint of the product. Section 3 also gives the specific requirements of the product and discusses the external interface requirements. Section 4 is for supporting information.

2. Overall Description

This document contains the problem statement that the current system is facing which is hampering the growth opportunities of the company. It further contains a list of the stakeholders and users of the proposed solution. It also illustrates the needs and wants of the stakeholders. It further lists and briefly describes the major features and a brief description of the proposed system. The Restaurant Automation System is designed for the large restaurants. The budget will determine the size and scope of the system.

2.1 Product Perspective

Before the automation, the traditional pen paper approach has the following drawbacks:

- Keeping track of empty tables requires either keeping a “dry erase” diagram of tables or the host constantly keeps track of the status of the tables if it is a small restaurant
- The waiter jots the order on paper and has to transfer redundant information to the terminal system. This takes some time and reduces efficiency in peak hours of patron service, also there may not be enough terminals available in the restaurant as there are number of tables which would require particular waiters to wait until the others are done entering their orders into the systems, so management of invoice data is managed poorly.
- The cooks could not notify the whether they require the specific ingredients or not and would maybe run out it during the preparations.
- Keeping billing and other statistical information was an issue of concern.

Hence the Restaurant Automation System (RAS) is proposed with the following features:

- Automatic management of prices as advised by the manager
- Automatic processing of orders and delivery of the order to the cook directly from the table.
- Maintenance of the amount of ingredients for the food beforehand in lieu of two days before and updation of ingredients everyday.
- Maintenance of the invoice data slips
- Generating receipts as soon payment is made and the table is clear.

The software application requires internet to fetch the details and information from the tables and the cook and display the results. All the information of the Restaurant Automation System is maintained on a database which is located in the web-server. Only the cooks, the owner, the Manager and the waiters would be given access to the System and it would generate invoices and menu cards withal the prices listed whenever required.

2.2 Product Functions

- Manage Prices
- Process Orders
- Maintain Ingredients
- Maintain Invoice data
- Generate Receipts

2.3 User Classes and Characteristics

Owner: The owner would be able to use the complete system, from managing prices of each item on the menu to generating receipts and handling billing details and all the payments and ingredients accounts. If sufficient cash balance is available cheques are printed for the invoice.

Manager: Would be able to print the invoice data, manage the prices of the item in the list, generating receipts and handling billing details and all the payments and ingredients accounts. Purchase orders are issued on daily basis whenever any ingredient falls below threshold value. When ordered ingredients arrive invoice data about quality and price is entered. Receipts are generated monthly or when manager wants to see them.

Sales Clerk: Enter the item code and process an order. Whenever an item is sold sales clerk would enter item code and quantity sold and generate bills. Whenever ingredients are issued for food preparation data is to be entered into the computer.

3. Functional Requirements

1. R1. Manage Prices

Description : Computer must support changes in prices made by manager and maintain the prices. The owner and the manager would be able to use the complete system, from managing prices of each item on the menu to generating receipts and handling billing details

and all the payments and ingredients accounts. If sufficient cash balance is available cheques are printed for the invoice.

Input : Enter the item code and the change in price .

Output : The price of the item is changed .

Processing: The database is updated.

2. R2. Process Orders

Description : The sales clerk would be able to process the orders by taking orders from indivual table and given the account of ingredients and other food items that will be used. Whenever an item is sold sales clerk would enter item code and quantity sold and generate bills.

Input : Enter the item code

Output :The Item is sold and bill is generated.

3. R3. Maintain Ingredients

Description : The sales clerk would be able to process the orders by taking orders from indivual table and given the account of ingredients and other food items that will be used. Whenever ingredients are issued for food preparation data is to be entered into the computer.

Input : Enter the food item

Output : Ingredients are issued and data is stored.

4. R4. Purchase Orders

Description : The owner and the manager will be able to purchase orders by checking the ingredients of the past three days and check with the threshold values. If the ingredient amount is less than the threshold value then, a purchase order will be made to make sure the stock remains for the next two days always. Purchase orders are issued on daily basis whenever any ingredient falls below threshold value .

Input : Item number of item below threshold value

Output : Purchase order is made

5. R5. Calculate threshold value

Description : The computer should calculate threshold value for each ingredient item based on average consumption of this ingredient for the past three days and assuming a two day stock is maintained for all ingredients.

Input : Enter ingredient
Output : Calculates threshold value

6. R6. Maintain Invoice data

Description : When ordered ingredients arrive invoice data about quality and price is entered
Input : Enter invoice data about quality, price.
Output : Enters Invoice data

7. R7. Print Cheques :

Description : If sufficient cash balance is available cheques are printed for the invoice
Input : Enter invoice details
Output : Prints cheque If there is sufficient cash balance

8. R8. Generate Receipts :

Description : Receipts are generated monthly or when manager wants to see them, generate receipts and handling billing details and all the payments and ingredients accounts. When ordered ingredients arrive invoice data about quality and price is entered.
Input : Enter the receipt details
Output : Generates receipt

4. External Interface Requirements

4.1 User Interfaces

4.1.1 Home Page: Home page will be common for all the users. It must contain the login options for all the users namely owner, manager, chef. It must contain the button which would lead waiters or clerks to place the order.

4.1.2 UI for Owner: Login Interface which would ask for user id and password. Owner home page which would have all the owner functionality like view all orders, generate bills and report. View purchase order: In this page all the purchase order will be displayed and against each ingredient required prior to two days.

4.1.3 UI for Manager: Login Interface which would ask for user id and password. Enter raw material interface: like view all orders, generate bills and report. View purchase order: In this page all the purchase order will be displayed and against each ingredient required prior to two days.

4.1.4 UI for Chef: Login Interface which would ask for user id and password. Place order interface wher placing an order with the correct amount would directly get the profit and the bill section interface.

4.2 Hardware Interfaces

The server is directly connected to the systems of various members in the department. Also the members has the access to the database for accessing the account details and storing the login time. The owner and manager access to the database in the server is only read only.

4.3 Software Interfaces

Software requirements at the user side

- Windows XP, 7, 8, 10
- Android
- Ios

Language Supported

- PHP
- HTML
- CSS
- Java(SIC)
- Javascript

Database

- Mysql

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Capacity: The system must be capable of supporting simultaneous connections without supporting performance loss. The system must be capable of supporting a minimum of 20 simultaneous connections.
- Response Time: The system must retrieve data and load pages within a reasonable amount of time. The load time must be within 20 seconds.

- User Satisfaction: The system is such that it stands up to the user expectations.
- Error Handling: Response to user errors and undesired situations has been taken care of to ensure that the system operates without halting.
- Safety and Robustness: The system is able to avoid or tackle disastrous action. The system safeguards against undesired events, without human intervention.
- Portable: The software should not be architecture specific. It should be easily transferable to other platforms if needed.
- User friendliness: The system is easy to learn and understand. A native user can also use the system effectively, without any difficulties.
- Information Security: When purchase orders are being done or invoice details are being updated or the price of the menu is being updated, all the details should be handled securedly
- Data Backup: The database contains not only the history of orders but also existing reports and open work orders that represent current orders and invoices and bills. Consequently, backing up the database on a regular basis (daily), is necessary.

5.2 Security Requirements

Each user must keep their password as confidential. More over the user must have individual ID for creating a login in the system. Only Administrator can control user addition and deletion in the system. Also this system could only create reports indicating the tasks of the respective users. There are a number of factors in the client's environment that may restrict the choices of a designer. Such factors include standards that must be followed, resource limits, operating environment, reliability and security requirements and policies that may have an impact on the design of the system. A SRS (Software Requirements Analysis and Specification) should identify and specify all such constraints.

- Standard Compliance: This specifies the requirements for the standards the system must follow. The standards may include the report format and accounting properties.
- Hardware Limitations: The software may have to operate on some existing or predetermined hardware, thus imposing restrictions on the design. Hardware limitations can include the types of machines to be used, operating system available on the system, languages supported and limits on primary and secondary storage.
- Reliability and Fault Tolerance: Fault tolerance requirements can place a major constraint on how the system is to be designed. Fault tolerance requirements often make the system more complex and expensive. Requirements about system behaviour in the face of certain kinds of faults are specified. Recovery requirements are often an integral part here, detailing what the system should do if some failure occurs to ensure certain properties. Reliability requirements are very important for critical applications.

6. Other Requirements

- Information Security: The system use SSL (secured socket layer) in all transactions that include any confidential customer information. The system must automatically log out all customers after a period of inactivity. The system should not leave any cookies on the customer's computer containing the user's password. The system's back-end servers shall only be accessible to authenticated management.
- Reliability: The reliability of the overall project depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Also the system will be functioning inside a container. Thus the overall stability of the system depends on the stability of container and its underlying operating system.
- Availability: The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. A customer friendly system which is in access of people around the world should work 24 hours. In case of a hardware failure or database corruption, a replacement page Software Requirements Specification for will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the Organizer. Then the service will be restarted. It means 24 x 7 availability.
- Maintainability: A commercial database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the project will be done. Also the software design is being done with modularity in mind so that maintainability can be done efficiently.
- Supportability: The code and supporting modules of the system will be well documented and easy to understand. Online User Documentation and Help System Requirements.