Software Requirements Specifications for a Gourmet Restaurant Management System

Prepared by: Asithandile Malote (566087), Milka Madahana (0304755j), Sibonelo Xulu (348740), Mfundo Ntini (496608)
Version 1.0
Date: 10 March 2016

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: This document presents a descriptive Software Requiremnts Specification (SRS) summary of the implementation of the Gourmet Restaurant Management System (GRMS). The proposed tools to be used for implementation of he front-end of the project are HTML5, CSS and Python. The back-end tools include Sqlite using Django. These are prone to changes during the course of the project.

Key words: CSS, Django, GRMS, HTML5, Python, Sqlite

1. INTRODUCTION

This document presents the SRS for an on-line reservation and meal ordering Gourmet restaurant. The system has close interaction between four parties; namely: the client, restaurant manager, software administrator and

1.1 PURPOSE

This document details the SRS of a Gourmet Restaurant Management System (GRMS). This document will be used mainly by the group members involved in the implementation of the GRMS.

1.2 SCOPE

The software product to be implemented for a client is a custom tailored restaurant software management system to run on a PC server (or on cloud computing service). Table 1 below shows the role the user.

Table 1 : Roles of users

User	Role
Client	Registration, log-in, meal selection, reservations, payments
Administrator	Log-in, stock taking, monitor payments, set meal prices, update specials
Auditor	Monitor the overall operation of GRMS

The restaurant manager is the most dominant user, in terms of control and management of the system, of the software. They are automatically given access to the system. They interact with the graphical user interface of the system to manage the following:

- Stock Inventory
- Ingredients/Menu planning
- Reservations
- Dish/Food costs
- Customer database
- Track Delivery
- Accounting/Financial administration
- Waitors/waitresses' duties, off-days, leaves, etc.
- Generation of reports (analysis)
- Special events

1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

- SRS - Software Requirement Specification
- GRMS - Gourmet Restaurant Management System

1.4 OVERVIEW

The SRS has two main sections: Firstly, the overall high level description and secondly the detailed specific requirement.

1.5 OVERALL SYSTEM

There is a main system of the Gourmet restaurant and it includes an online subsystem. The details of this hierarchy is illustrated in Figure 1 and Figure 2 below.

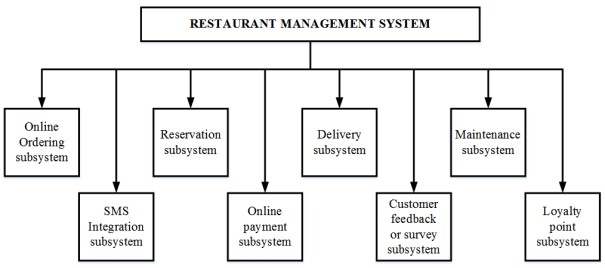

Figure 1 : Gourmet restaurant management system flow diagram

The main subsystem of the Gourmet restaurant system is an online system. This system allows clients to access interact with the restaurant system using smart

electronic devices. The online system is shown in the flow diagram below:

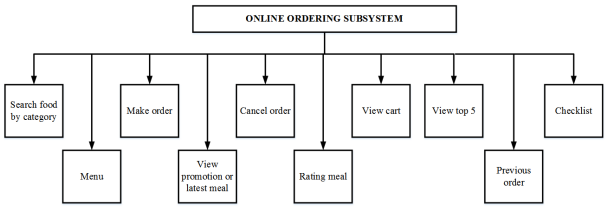

Figure 2 : Online sub-system of Gourmet restaurant

1.6 PRODUCT PERSPECTIVE

The Gourmet restaurant management system is an autonomous system that is to replace the current system which is mainly manual. In the current system, restaurant orders and reservations are done through telephonic booking or visiting the restaurant and having information manually captured. Figure 1 shows the system that is to be implemented.

1.7 HARDWARE INTERFACES

- Personal computers
- Tablets
- Smart phones

1.8 PRODUCT FUNCTION

Table reservation and meal ordering system

- Allows for customers information to be captured
- Adjustable menu prices
- Ability to modify reservations
- When there are no tables and a customer would like to make a reservation, their information is captured in the data-base. When the tables are available, the first customer on the list is notified.
- The number of unoccupied tables at the restaurant are shown
- Payments records are also shown
- Allows for customer feed-back

Tracking and selling food system

- All meals purchased can be tracked

General Management

- Reports are generated to audit sales
- Allows addition, deletion and modification of information on menu items, prices and user profiles
- Registration of users and allocation of passwords

1.9 ASSUMPTIONS AND DEPENDENCIES

In simplifying the scope of the project a few assumptions have been defined. Firstly the hierarchy of the restaurant management is assumed to be simplified in that only the restaurant manager and system administrator have full permissions into the system. This means that any random employee requiring changing something through the system must go through one of the mentioned people. This simplifies the system in that there need not be customized permissions for each employee which can be very time consuming.

It is also assumed that a specific user can log on into multiple instances of the system, thus the system does not block a user to login whilst there is another instance of that same user being logged in. This is important in the case where the user is logged on in a slow internet speed area and needs a peer to log in and submit an urgent order for them.

A single user is allowed to reserve multiple tables in the restaurant even if they are not spatially linked. This is important in the case where a user wants to host a party and might need to use different sections of the restaurant or in the case where a user wants to book for friends.

Django is the major dependency introduced to the project scope. This platform and framework is used since it combines multiple functionalities which would otherwise require different IDEs to implement. The Django framework provides different PYTHON Web Development libraries, an in house database management system and it also open-source amongst other advantages.

Notepad++, Visual Studio and XAMPP are also used in the testing process in order to compare the different frameworks capable of producing the solution as ASP.NET and PHP as the server side language were one of the contender frameworks and languages.

2. SPECIFIC REQUIREMENTS

This section covers information that will be used by designers to design and test that the system fully satisfies the specific requirements and outcome.

2.1 EXTERNAL INTERFACES

The GRMS will use the standard input/output devices for a personal computer as listed below

- Mouse
- Printer
- Keyboard
- Monitor

2.2 USER INTERFACE

The system is organized such that the system administrator, Restaurant administrator, the client and the auditor are able to log into and out of the system using a username and password. They can then view what is relevant to them as shown in the tables below:

Table 2 : GRMS user interface for the client

Action	**Description**
Register	Allows client to register
Update details	Allows client to update their personal details should there be a change in original details provided
View promotions	Allows users(regular and temporal clients) to view current promotions that the restaurant is offering. Meals on specials are also included
View previous orders	Client is able to check what they have ordered previously
View cart	Client views the meal orders with ingredients
Cancel order	Allows client to cancel order
Search	Allows client to search meal by category, these also include vegetarian or non-vegetarian
View menu	Allows client to view menu
View top five meals of the day	The best rated meals, by other clients, are displayed to attract other clients
Feedback	This is a feature used by the restaurant administrator to get feedback and comments from clients. The information is then used to improve their services. The system administrator should be able to generate daily, weekly, monthly or yearly chart reports.
Make a reservation	Clients are able to make a reservation at the restaurant. Important details for instance day and time of arrival, number of tables and the grouping whether it should include children or not. Do they wish to be inside or outside the restaurant.
Make payment	The client is given a choice to make payment cash or to use the credit system.
Order Online	The feature allows customers to order on line.

A delivery feature maybe added in future. This feature is used when customers request for delivery. The customer has to provide their full names, contact number, delivery address, date and time. After filling in the delivery details, it is submitted to the restaurant. The staff members keep track of the delivery after the meal is cooked. The staff can view the meal order, delivery detail, total amount, tax, charge and they can also print delivery information for their reference.

An added Search icon will display the meal name, description of the meal, image of the meal and price. Customers can select quantity and add to their list. The order is sent to the restaurant by selecting confirm button. Chefs and other staff members are able to check orders from the check list and update status after the meal is cooked.

Table 3 : GRMS user interface for the restaurant manager

Action	Description
View/Edit meals	The restaurant manager can edit the details of the meal to present newly prepared seasonal meals.
View/Edit quantity	Allow manager to view and edit the quantity of meals and ingredients available on stock.
Update meals	New meals introduced by the chef are captured by the system administrator including the image, description of the food, ingredients and the prices.
Access feedback report	Allows manager to view the report given by the clients.
Delete or add clients	If a client is no longer using the services rendered or is abusing the services then the manager is able to delete them.
View/Edit ingredients	The administrator is able to edit the ingredients in a meal.

Table 4 : GRMS user interface for the software administrator

Action	Description
View/Edit restaurant opening days and times	Allows administrator to edit the restaurants opening times, days, dates and whether it will be open on special occasions on not.
View feedback report	View feedback report from the clients.
Vacancies	The restaurant manager can view available vacancies at the restaurant
View reserved tables	The total number of tables reserved and their reservation times.
View updates from staff members and chefs	These updated includes the details of the stock needed.
View customers loyalty points	For the purpose of ordering new stock, the administrator has to have a report on the clients loyalty system.

2.3 SOFTWARE INTERFACES

The system shall interface with an Access database.

2.4 HARDWARE INTERFACES

The system shall run on Microsoft Windows, Mac OS X and Linux systems.

2.5 COMMUNICATION INTERFACES

They will be no need for any communication devices because the system is fully autonomous.

3. REQUIREMENTS

Listed below are the requirements of what the designed system should do.

3.1 FUNCTIONAL REQUIREMENTS

Reservation/Booking

- The system shall record reservations
- The system shall record the customer's first name and last name
- The system shall record the customer's phone number
- The system shall display whether tables are guaranteed or not
- The system shall generate a unique confirmation number for each reservation
- The system shall automatically cancel non-guaranteed reservations if the customer has not provided their credit card number by 2 hours prior to coming to the restaurant
- The system shall record the reservation date and time
- The system shall allow reservations to be modified without having to re-enter all the customer information
- The system shall display the amount owned by the customer prior to them coming for a booked meal
- To retrieve client's details details the last name or room number shall be used
- The system shall record empty tables
- The system shall record the payment
- The system shall record the payment type
- The system shall charge the customer a certain amount of money if the booking is canceled
- The system shall record customer feedback

Food

- The system shall track all meals bought at the restaurant
- The system shall record payment and payment type for meals
- The system shall accept reservations for tables and deliveries

Management

- The system shall display the restaurant profiles. These would include number of tables booked previously, currently and in future
- The system shall display food and drink revenue for a specified period of time
- The system shall allow for the addition of information, regarding menu items, prices and user profiles
- The system shall allow for the deletion of information, regarding rates, menu items, prices and user profiles
- The system shall allow for the modification of information, regarding rates, menu items, prices and user profiles
- The system shall allow the system administrator to assign user passwords

3.2 NON-FUNCTIONAL REQUIREMENTS

Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability and portability.

3.2.1 PERFORMANCE REQUIREMENTS

The following are the system acceptance performance response times:

- The load time for user interface screens shall take no longer than three seconds
- The login information shall be verified within six seconds
- Queries shall return results within six seconds

3.2.2 LOGICAL DATABASE REQUIREMENTS

The database requirements are not limited but include the following:

Booking/Reservation System

- Customer first name
- Customer last name
- Customer address
- Customer phone number
- Number of tables booked
- Assigned room
- Credit card number
- Confirmation number
- Automatic cancellation date
- Expected arrival time at the restaurant
- Customer feedback
- Payment received (yes/no)
- Payment type
- Total Bill

Food Services

- Meal
- Meal type

- Meal item
- Meal order
- Meal payment
- Ingredients

3.2.3 AVAILABILITY

The system shall be available at all times to ensure clients can access important information

3.2.4 Security

The following will be able to access the subsystems;

- The system administrator
- The restaurant administrator
- The client
- The auditors

Access to the various subsystems will be protected by a user log in screen that requires a user name and password.

3.2.5 MAINTAINABILITY

The GRMS system is being developed in Python using the Django framework. Web applications written using Django are easy to maintain.

4. CHANGE IN MANAGEMENT PROCESS

Change to this document will be done by the front end and back end managers

5. FRONT-END COMPONENT OF GRMS

This section is allocated to Milka Madahana and Sibonelo Xulu. The front-end of the project consists of the following specifications:

- Graphical User Interface (GUI). This will be achieved using HTML5 and CSS.
- Script (Dynamic Web Page). The implementation will be carried out on Python.

6. BACK-END COMPONENT OF GRMS

This section of the project will be implemented by Asithandile Malote and Mfundo Ntini.

The back-end component consists of the database model of the GRMS. In addition, the database models relationships to illustrate the user interactions with the GRMS, the relationship between the meals database and client database. The meal database will be linked to the ingredient, calories and price database. Sqlite and BerkeleyDB are suggested as tools for the implementation of the database system.

7. INPUT-OUTPUT DESCRIPTION OF GRMS

The input to the GRMS will be the client interaction with the GUI. This SQL commands will then allow the to access the tables in the database. The client is to login to the system via their user credentials and once that information has been captured the client can then navigate their way around the site. The site will allow the client to choose an order via a drop down menu. Once the base order has been chosen the client can add or remove side ingredients of their choice with little limitations. Upon the confirmation of the order the site will then output to the client the amount of money the order costs and the nutritional information of the order. The administrator can also login to the system but with different permissions as compared to the client. the administrator can change the available base orders and side ingredients currently on the database and can also update the prices if need be.

8. CONCLUSION

HTML5, CSS and Python will be used for the front-end of the project whilst Sqlite and BerkeleyDB will be used for the back-end. Careful attention is to paid in the database design and table relations such the system is efficient without repetitions and an acceptable level of information integrity