TEAM X

On-line Restaurant Order and Delivery System
Software Requirements Specification
For Online Restaurant Application

Version 1.0

Revision History

Date	Version	Description	Author
13/10/2020	1.0	Creating a login screen	Saiful Islam
18/10/2020	1.0	Specification report for phase 1	Hafsa Nadim Tanzil Baraskar Saiful Islam Daniel Lichter Dante Betancourt

grade: 90
comments: the use case diagram is good, the description should be 1-1 with each oval (rectangle in your diag) to state the interested user(s) and processing, the one you gave is too casual and cursory for a spec doc. (formatting complaint: your indexes cost too many pages: 7+ out of 20, will raise many eyebrows).

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the system requirements for an online restaurant order/delivery system. The SRS will explain what the online restaurant system will do, its use/purpose (functional and non-functional requirements), and explain in-depth about what sorts of features this system will include. The SRS mentions who will be associated/connected with the system (registered customers, VIP customers, managers/superuser, chefs, delivery people, and surfers).

1.2 Scope

Restaurant system is a GUI based application that helps customers order food online based on their preferences provided by the restaurant. This also helps the employees manage the restaurant by allowing them to use the system to deliver food, arrange time and seats for the customers, demote/promote chefs and delivery people based on their ratings, or get complaints/compliments from customers.

Customers can choose how they want their food, such as delivery, take out, or eating at the restaurant. The restaurant provides menus of food and keeps a history of prior orders, making ordering more efficient for the users.

With this system, the restaurant increases its business. Also, customers, managers, chefs, and delivery people are greater connected. Chefs put up descriptions of menus, and the ratings of food items are shown to everyone.

Customers can view their three recent orders (if they have made prior orders) or the top three popular and best-rated dishes (if they have not made previous orders), which helps them choose the dish they want. Delivery people get to chime in and deliver the food.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
SRS	Stands for Software Requirements Specification, a document that thoroughly describes all the functionalities of the software system that is to be developed
Software	Set of instructions, data, or programs used to operate computers and execute specific tasks
System Software	Consists of multiple programs that control and manage the computer operations
User	A person who will be interacting with the system
GUI	An acronym for Graphical User Interface. It is a user interface that includes graphical elements such as icons and buttons. It displays objects that convey information and represent actions that can be taken by the user.
Feature	Distinctive attribute or aspect of something. Tools you use within a system to complete a set of tasks actions.

Functionality	How the features work to provide you with the desired outcome. It is the actions, capabilities, and usefulness of something such as a software application.
Animation	A method in which figures are manipulated to appear as moving images
UML	Stands for Unified Modeling Language, and it is intended to provide a standard way to visualize the design of a system
Use-Case Diagram	A high-level diagram that is part of UML. It is a way to describe the system. The whole process of this is that we have a system boundary (represented by a rectangle). Next, we have to identify all the users that will be interacting with this system. On the left-hand side, we have the ordinary users, and on the right-hand side, we have the privileged users (people who have power). After identifying the users, we list the features and link all the features to the users interacting with them.
Restaurant	Place where people go to dine
Chef	A user who is responsible for making the food
Customer	A user who buy goods and services
Manager	A user who runs the restaurant and oversees everything
Delivery person	A user who delivers the food

Registered customer	A user who can browse, search, order, and vote on food items
VIP customer	A user who spends a great amount of money at the restaurant

1.4 References

[1] “UML Use Case Diagram Tutorial,” [Online]. Available: www.lucidchart.com/pages/uml-use-case-diagram. [Last Accessed: Oct 14, 20]

[2] Krüger, Nico, “How to Write a Software Requirements Specification (SRS Document),” Oct. 28, 2018. [Online]. Available: www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document?fbclid=IwAR15l8bhyJJ0zg2ul5VteXZpxOhKu-SfNrL2twkD9Rmk6Evt4j3l_iWDMm4. [Last Accessed: Oct 14, 20]

1.5 Overview

The remainder of this document includes an overall description of the software, specific requirements, and supporting information.

The next part, the overall description, gives an overview of the functionality of the product.

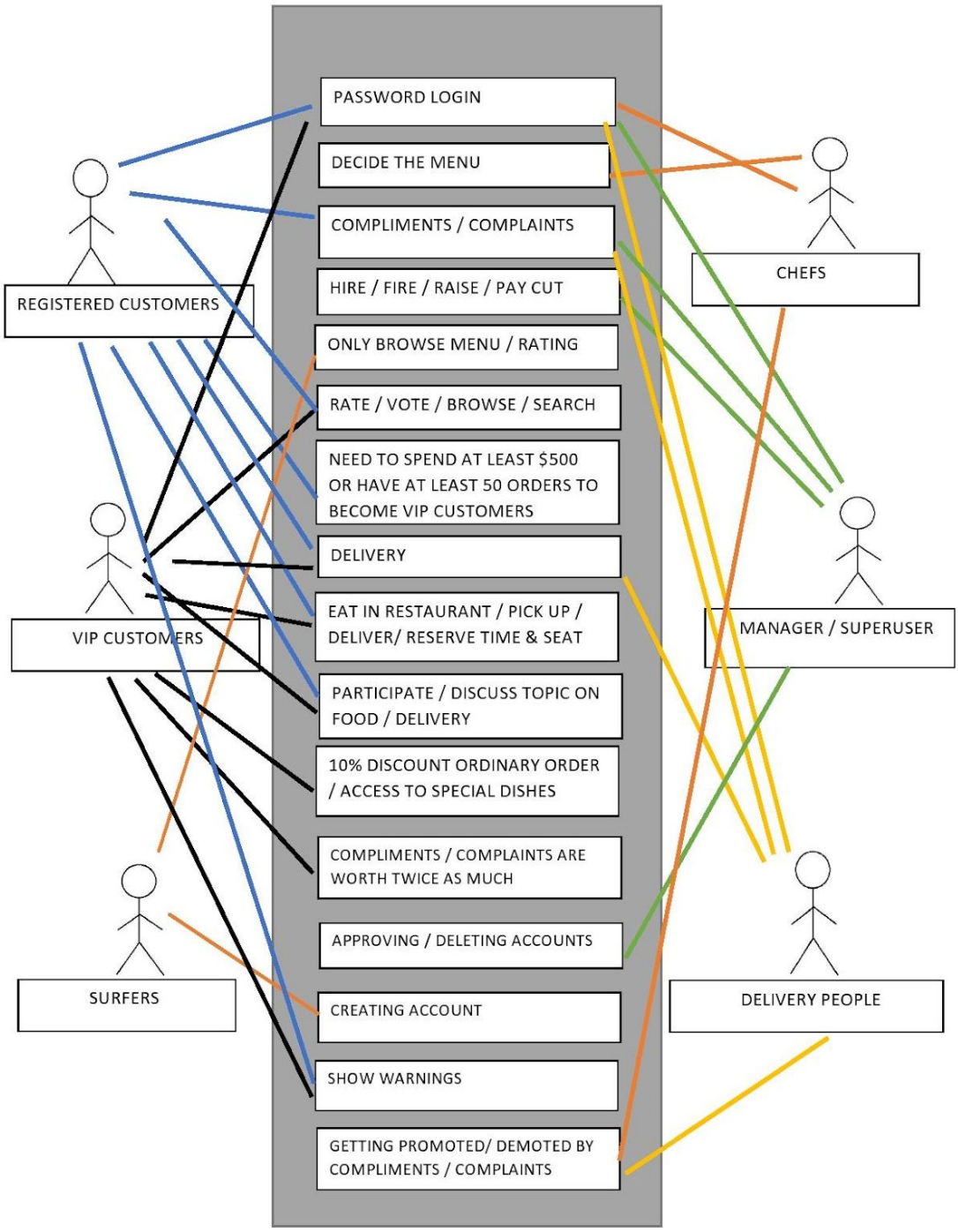
The third part, the specific requirement, contains all of the system’s functional and quality requirements. It is also a detailed description of the system and all its features. In this part, it also goes in-depth about features that are not mentioned in the case diagram. It is primarily written for the developers and describes the product’s functionality in technical terms.

The fourth and final part, supporting information, makes the SRS easier to follow because it includes the following: Table of Contents, Index, and Appendices.

2. Overall Description

2.1 Use-Case Model Survey

RESTAURANT SYSTEM:



2.2 Assumptions and Dependencies

The development of this software will focus on desktop and laptop systems. It is assumed that the online restaurant will be made available to users that are familiar with and possess an internet connection and a computer. We can also assume that the users are familiar with how a restaurant and its order/delivery work. If users are not familiar with this process, then utilizing this system will be difficult.

3. Specific Requirements

3.1 Use-Case Reports

The system needs to include all the components that will be interacting with it, such as the users and features/functional requirements. The system needs to account for both the ordinary (LHS) and privilege (RHS) users: registered customers, VIP customers, delivery people, surfers, chefs, and managers/superusers.

The functional requirements are:

1. Registered customers, VIP customers, chefs, and delivery people can access their accounts after they login using their passwords.
2. Both registered and VIP customers choose to either eat their food in the restraint, pick up the dishes themselves, or have their food served by delivery. When a registered or VIP customer chooses to eat their food in restraint, they arrange the available time and seating.
3. The chefs decide the menus. They put in the description and keywords for registered customers, VIP customers, and surfers to search and browse through.

4. The delivery system manages delivery to the users (registered customers and VIP customers)
5. Only the manager/superuser handles customer compliments and complaints (registered and VIP) and delivery people. VIP's complaints or compliments are counted twice as important as ordinary customers.
6. Only the manager/superuser can hire/fire/raise or cut pay for chefs and delivery people.
7. Managers are also responsible for handling the customers who are kicked out of the system or decide to quit the system by clearing their deposits and closing their accounts. Managers also keep a record of inappropriate words used by customers. Customers will receive a warning for using such words, and in the message, the words will be replaced with ***. A message that has more than three taboo words will automatically block the user.
8. Surfers can apply to be registered customers with a fixed amount of money in their deposit accounts (which should be checked by the manager). When a new customer/surfer signs up, the top three most popular and three highest-rated dishes will show on the page.
9. Registered customers who have three warnings are de-registered. VIP customers who have two warnings go back to as being referred to registered customers. These warnings should be displayed on the personalized page when they (customers) login.
10. A chef whose dishes received constantly low ratings, three complaints, or no order at all for three days will be demoted, meaning their salary will decrease. A chef whose dishes received a high rating or three compliments will get promoted, meaning their salary will increase. Delivery people are handled in the same way.

11. Surfers can only browse the menu and the ratings.
12. Registered and VIP customers can browse, search, order, and vote (rate) the food made by the chefs and rate the food delivered by the delivery people (1 being the worst and 5 being the best rating)
13. Registered and VIP customers to start/participate in discussion topics on the cooks, dishes, and delivery people.
14. Registered customers who spent more than $500 or placed more than 50 orders to be considered as VIP customers
15. VIP customers receive a ten percent discount on their ordinary orders and have access to specially developed dishes
16. Our team's special/creativity feature is an animation, such as a logo for the restaurant. We plan to pop up the restaurant logo as someone logs into their account (this is subject to change). This animation feature will make the system stand out and make the GUI more appealing to users.

3.2 Supplementary Requirements

To ensure that the system lasts longer and is reliable, it needs to have the following:

1. **Easy to use and maintain**

   For the online restaurant system to be applicable for the users, our design will need to have a user friendly and easy to use interface. Users from different areas of computer expertise must find it easy to interact with the system. We must develop the system in a way where we can make adjustments to the system based on user feedback and change the system to meet user's needs after the deployment of the system.

2. **Efficiency, Reliability, & Security**

We expect there will be many users using this system; therefore, it must be efficient. There must be enough servers so that the system can handle all the users and does not crash or slow down.

The users will have personal information registered in their accounts, such as address and bank account information so it must be secured and prevent any hackings.

3. **Reliability**

The system must perform the function as the user expected. It must tolerate the user's mistakes or using the software in an unexpected way. The system must perform well enough for the required case in a given time.

4. Supporting Information

4.1 INDEX

C

D

E

F

G

H

I

L

M

N

O

P

T

U

V

W

4.2 APPENDIX