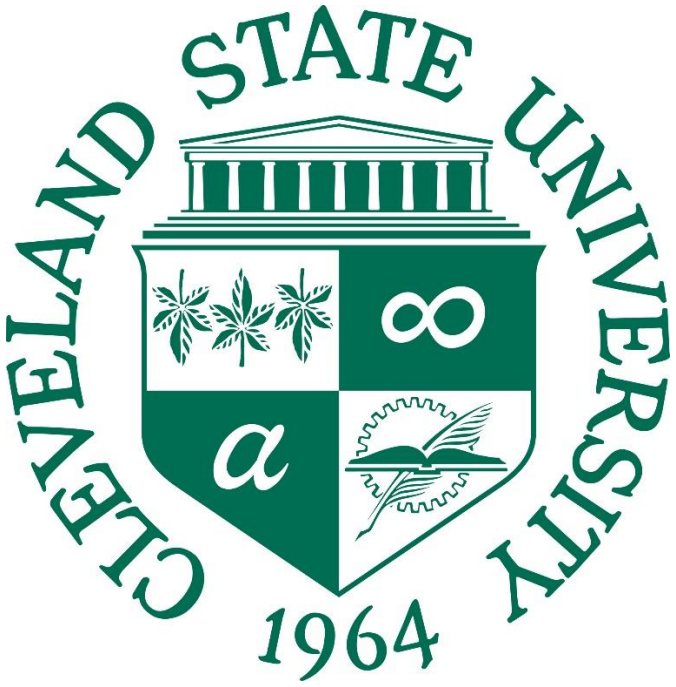
CLEVELAND STATE UNIVERSITY



CIS 694 – Object-oriented Software Engineer

SOFTWARE REQUIREMENTS SPECIFICATION

Project: Restaurant Management System

Presented By:
Group 5
Bhavana Tedlapalli (2808568)
Sravan Kumar Singupuram (2836831)
Riya Patel (2829317)
Calvin Raj Namburi (2836250)

1.0 Introduction

This document presents a detailed explanation of the objectives, features, user interface and application of Restaurant Management System in real life. It will also describe how the system will perform and under which it must operate. This document is intended for different types of readers such as restaurant owner, system designer, system developer and tester. By reading this document a reader can learn about what the project is implemented for and how it will present its basic ideas. In this document it will be also shown user interface. Both the managers and the developers of the system can benefit from this document. This document describes all data, functional and behavioral requirements for software development.

1.1 Goals and objectives

The project's goal is to offer an effective catering control system whose primary functionality, includes anticipating the need for the upcoming order. If there is a "Special Occasion," the user can order party food requesting for catering. The user can add the required items by checking the menu, which is provided by us, and add their special requirements according to the taste of their preference.

The success criteria depend on:

- Predicting the needs of the upcoming order.
- Correlating recipes to their appropriate ingredients and spice level.
- Simplicity of use for updating orders and sending orders to users.

1.2 Statement of scope

Traditional paper-based menus are commonly used in formal dining environments to convey the various food and beverage options available to customers. These menus are typically limited in their textual content and can be updated only by the restaurant owner. The related concepts are covered by the restaurant food ordering system's general scope. This system aims to replace these menus with electronic ones.

This system will aid in the systematic management and operation of the restaurant business. In restaurant management system website, we will include a management system that customers can utilize to place food orders. Through this website, customers may also provide feedback. So that the restaurant's owner can assess the entire system. In the end, this will result in the hiring of fewer waiters and present an opportunity to appoint additional chefs and improved kitchen facilities to serve food more quickly.

Customers may also use POS(Point of Sale), which will be integrated with the management software, to make payments using debit or credit cards and through cash. Customers can view the restaurant's current discount options. Additionally, customers can view the calorie chart, which will raise their awareness of their health. The system will save all the data regarding daily costs and revenue.

1.3 Software context

The system must communicate with the configurator to identify all components available to configure the product. The system needs to communicate with content managers to retrieve product specifications. With such a broad realm there comes a broad spectrum of development. As a result, we focus primarily on the fundamental tools used in outlet inventory control in our program. Although the program will first be designed with the needs and data already available in mind, it will be simple to adapt it to a larger domain, such as the entire restaurant sector.

While there is plenty of software available to track food, item sales, inventory management is lacking in our target domain. From huge corporate dining facilities to tiny, independently owned eateries, our software can be scaled. Additionally, it has a somewhat narrow scope because the database relies on recipes to produce the required ingredients. Additionally, based on the sales of certain recipes, the inventory is updated. This criterion narrows the scope of our offering to our industry and increases its appeal to anyone looking for a fix for this particular issue.

Restaurant Management System website will attempt to replace the traditional manual ordering process and is a new self-contained software system that consists of website application will be used for ordering and ordering related information about the food items like pending and complete order queues and calculating the sales regularly.

1.4 Major constraints

The system interface to the phone's GPS navigation system places restrictions on the web application. Since there are numerous GPS and system manufacturers, the interfaces for each of them will probably differ. Additionally, the navigational elements that each of them offers could vary.

Another limitation of the application is the Internet connection. It is essential that there be an Internet connection for the application to run because it retrieves data from the database over the Internet. The database's storage space will have an impact on both the online portal. Since the database is shared by website, it might be necessary to queue incoming requests, which would lengthen the time it takes to fetch data.

The Restaurant Management System helps the restaurant manager to manage the restaurant more effectively and efficiently by computerizing meal ordering, billing, and inventory control. The system processes transaction and stores the resulting data. Reports will be generated from these data which help the manager to make appropriate business decisions for the restaurant. For example, knowing the number of customers for a particular time interval, the manager can decide whether more waiters and chefs are required. Moreover, easily calculate daily expenditure and profit.

The whole management system is designed for a general Computerized Digital Restaurant. So that any restaurant owner can get it and can start automated process to his/her restaurant.

2.0 User Scenario

Here in the above picture, you can see various users apart from receptionist. There are different scenarios for each for Customer, Manager, Head Chef, Admin and Chef. Manager can see/edit the status of available/reserved tables. Customer's interface will consist of a scrollable menu listing available items with the number of calories and their price. When the customer selects some dishes and place the order, it will be stored in "pending orders" table in Firebase database. Chef's interface will be such that he is notified of the pending order, and he is able to assign it to one the available queues of chefs who are then able to see the new order in their screens or on a central display in kitchen. After each item/dish in an order is prepared, the order is marked completed through the Chef's interface, the hall manager gets notified through his interface. Customer's interface has an option for requesting the bill. Bill is printed through the Manager's interface. Admin can change and modify the Firebase database like add new menus or staff.

2.1 User Profiles

The web application will have two interfaces. Each for Restaurant Manager and Customer. Customer's interface will consist of a scrollable menu listing available items, and their price. When the customer selects some dishes and place the order, it will be stored in "Cart". Whenever there is a party order customers can add/modify/delete accordingly to their requirement. And they can add/modify/delete number of people. And they can specify their instructions and add up to their spice levels.
Manager's interface will be such that he is notified of the pending order, and he is able to assign it to one the available queues of chefs who are then able to see the new order in their screens or on a central display in kitchen. After each item/dish in an order is prepared, the order is marked completed through the Manager's interface. Customer's interface has an option for requesting the bill. Bill is printed through the Manager's interface.
Admin can change and modify the Firebase database like add/remove new items into the menu on the website. After each item/dish in an order is prepared, the order is marked completed through the admin's interface.

2.2 Use-cases

The customer use-case is for the customer signup, log in, forget password, checking food to be catered. Each customer check tables and select table that's the reservation. If customer login, they can see all food list, they can check food name with calories and budget.
The Manager use-case is for the manager login, check tables, check how many customers visited, how much earn.
The Receptionist use-case is for the Receptionist login, which table request which food and after check food, serve the food in specific table.

The admin use-case is for the admin login, check all customers, managers, and receptionists, who want change password or who want to delete.

2.3 Special usage considerations

Since this is the restaurant manage system, some special requirements may be associated with using the software. For example, the software may need to be able to interface with a customer request database. Additionally, the software may need to be able to serve food or if anyone has no money in his/her account, he can request to deposit money from his card/bank.
Another special requirement that may be necessary is to change customer, manager, receptionist, admin, and chef. This is the business so that there are some levels. In this restaurant manager system, receptionist can manage manager. Admin can change receptionist, manager, and customer. Chef can change all of them.

3.0 Data Model and Description

Customer, Manager, Receptionist, Admin and Chef roles.

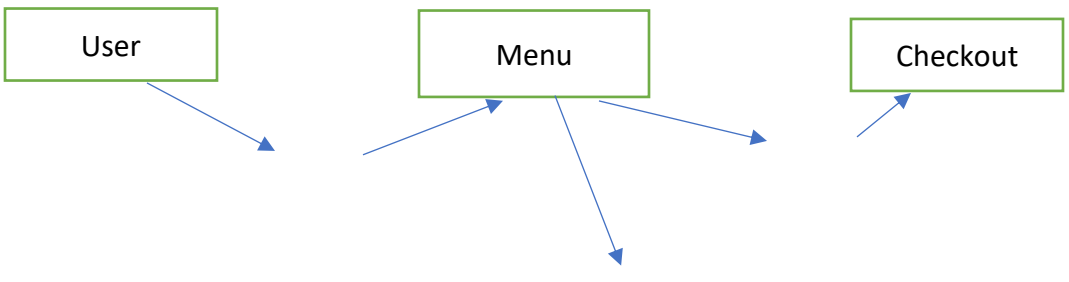
3.1 Data Description

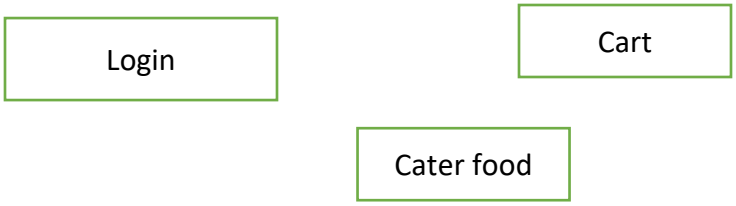
The customer, manager, receptionist Admin and Chef has the same data, but role is different. In this project, customer is role 5, manager is role 4, receptionist is role 3, admin is role 2, Chef is role 1.
In food table, there are 3 columns, food name, food price, food image URL.
In cart table, there are 2 columns, customer email/name and food name.
In checkout table, there are 2 columns, customer email/name and food name.

3.1.1 Data objects

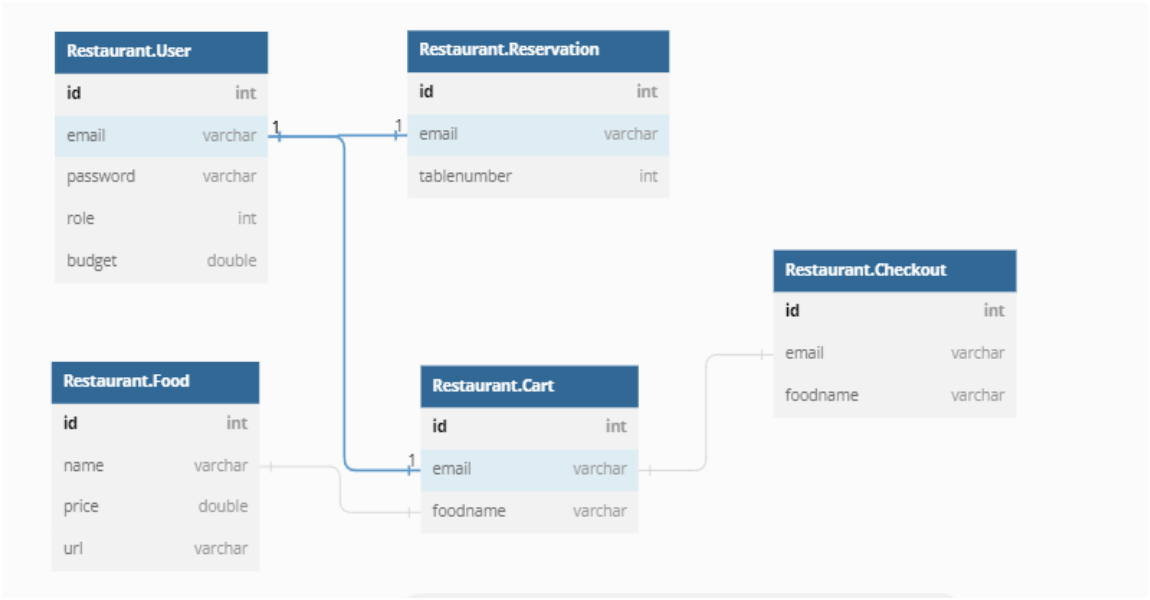
Food – name, price, URL
User – email/name, password, role, budget
Reservation – email/name, table name
Cart – email/name for customer, food name
Checkout – email/name for customer, food name

3.1.2 Relationships





3.1.3 Complete data model

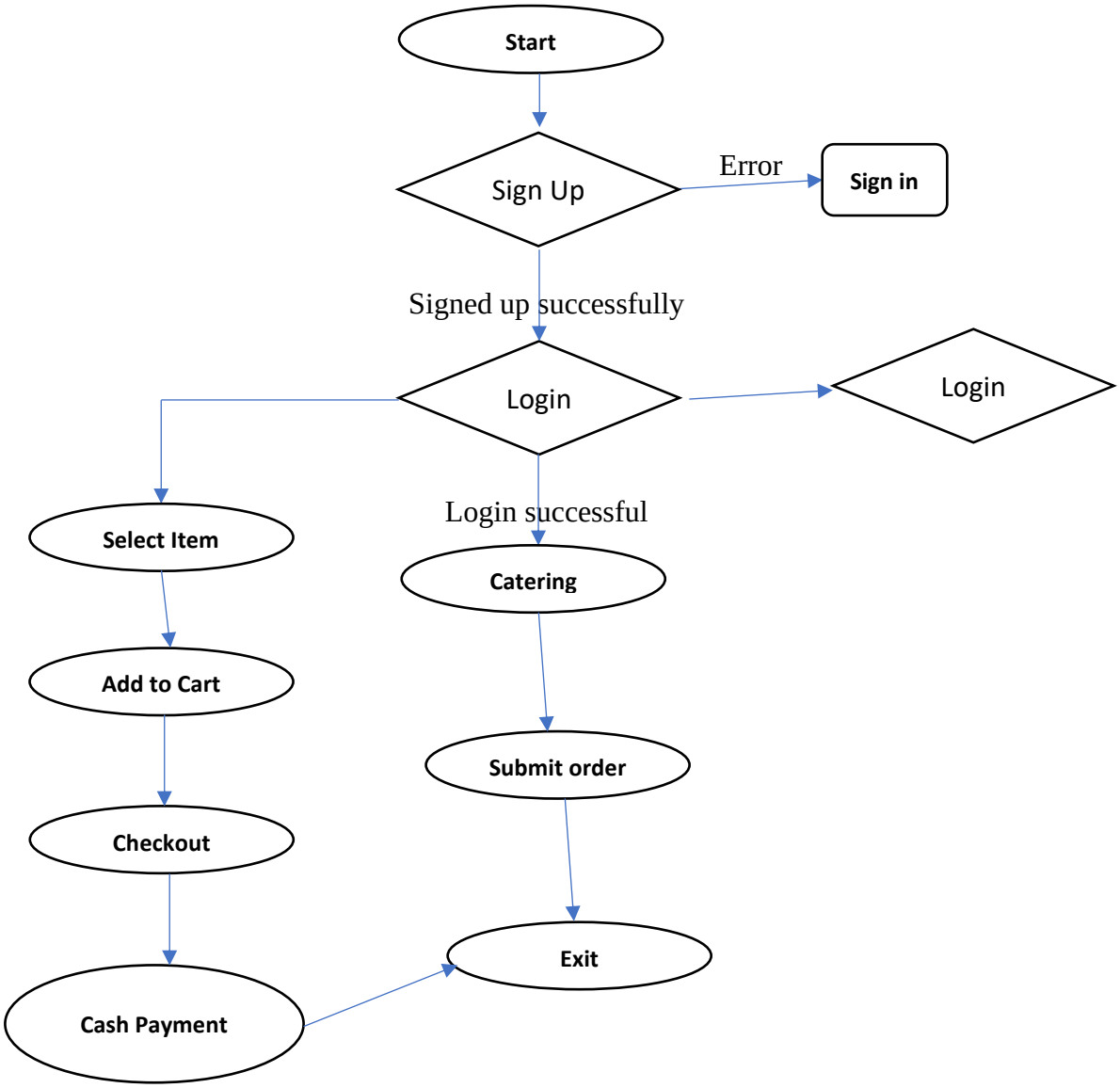


3.1.4 Data dictionary

Data dictionary is same as the Data model in 3.1.3.
All users' details will be stored in user table, and if there is no user in user table user can create new user. And user can check food price, type, calories from food table.
Other users will be the same, but when create new other user, we must check roles for manager, receptionist, admin, and chef.

4.0 Functional Model and Description

This project has been done using python, flask, html, and MySQL. The project has the 3 modules: User management, Catering management, Food request, Checkout money.



The above picture shows the data flow of the proposed system.

4.1. Description of Major Functions

- User management: Store user with name/email, password, role, and default budget.
- Catering management: Store new caterings with name/email and into that able name.
- Food request: Store a new row with name/email and food name into cart.
- Checkout money: Store a new checkout with name/email and food name into checkout and delete from cart.

4.1.1 Requirement 1

User management: The module is responsible for storing and retrieving user information.
As we mentioned above, role 1 is for Admin, role 2 is for Customer.

4.1.2 Requirement 2

Catering management: The module required the email/name from user. All users can cater food from the restaurant. Check empty table exist or not first and if there is any empty table, then store email, firstname, lastname, contact number, items, catering date, venue, special instructions into the cater table.

4.1.3 Requirement 3

Food request: The module required the email/name from user after reservation. All users can request food. If user want to 1+ food, they can repeat to request food.
All foods and user email/name will be stored in cart.

4.1.4 Requirement 4

Checkout money: The module required the email/name from user after request food. All users can checkout money. The total price of foods, username will be stored in checkout.

4.2 Software Interface Description

The software interface of the above project can be divided into three parts. The first part is the Web base interface, which is used by the user to input the user's information and view the user's request. The second part is the controller part, if the user enters their information, the web base interface create the URL for each request. The third part is the model part, that will read or write data into database, the data is from second part.

The Web base interface is developed using the python and flask template. The Flask module is a standard python module that is used to create Website. The Flask provides a good object-oriented interface.

4.2.1 External machine interfaces

The external machine interface is the flask web base application using MySQL. There are many databases, but MySQL is standard SQL library, I think mongo dB is also good no-SQL database but it's not good in this project. And Flask is also

simple and easy to understand and implement by all students or users who want to create new website.

4.2.2 External system interfaces

- Flask module is used to create Web base application.
- User Information Form is used to input or output user information.
- Catering management is used to cater food for the customer's party order..
- Cart management/Food request is used to store food data that user want to request.
- Checkout management is used to store food data and total price that user request.

4.2.3 Human interface

This is the safest and most effective management method in future technology. The system provides an interactive user-friendly interface that is easily understandable for all users. The System is available at least during the restaurant operating hours and must be recovered within an hour or less if it fails. The system responds to the requests within two seconds or less. The system provides consistent performance with easy tracking of records and updating of items. The software is easily maintainable and adding new features and making changes to the software must be as simple as possible. Only authorized users are able to access the system and view and modify the data.

5.0 Restrictions, Limitations, and Constraints

1) The first issue is related to the User Information management. When user input role, the role showing as the string as like "Admin", "Customer" but in database the data will be stored as 1, 2. So that the runtime will be much when create or read data.
2) The second issue is also related to the User Management. When user input data, user can't input initial budget for them. In this project, I added the static variable as 400, if user create new user.
3) The third issue is related to the implementation of the software. The software should be such that the users can easily use it, other types of users can manage the system, but there is no rate so that other users can't check this restaurant is good or not.