

Software Requirements Specification

for

Restaurant Management system

Version 1.0

Prepared by:

Rushabh Maru

04-03-2023

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	2
1.5 References	2
2. Overall Description	3
2.1 Product Perspective	3
2.2 Product Functions	4
2.3 User Classes and Characteristics	4
2.4 Operating Environment	5
2.5 Design and Implementation Constraints	5
2.6 User Documentation	6
2.7 Assumptions and Dependencies	6
3. External Interface Requirements	7
3.1 User Interfaces	7
3.2 Hardware Interfaces	8
3.3 Software Interfaces	8
3.4 Communications Interfaces	8
4. System Requirements	9
4.1 Place Order	9
4.1.1 Description and Priority	9
4.1.2 Stimulus/Response sequences	9
4.1.3 Functional Requirements	10
4.2 Help	10
4.2.1 Description and Priority	10
4.2.2 Stimulus/Response sequences	10
4.2.3 Functional Requirements	11
4.3 Edit Order	12
4.3.1 Description and Priority	12
4.3.2 Stimulus/Response sequences	12
4.3.3 Functional Requirements	13

Software Requirements Specification

4.4	Cancel Order	13
4.4.1	Description and Priority	13
4.4.2	Stimulus/Response sequences	13
4.4.3	Functional Requirements	13
4.5	Request Bill	15
4.5.1	Description and Priority	15
4.5.2	Stimulus/Response sequences	15
4.5.3	Functional Requirements	15
4.6	Customer Feedback	15
4.6.1	Description and Priority	15
4.6.2	Stimulus/Response sequences	16
4.6.3	Functional Requirements	16
4.7	Add/Edit/Delete Staff Members	16
4.7.1	Description and Priority	16
4.7.2	Stimulus/Response sequences	16
4.7.3	Functional Requirements	17
4.8	Add/Edit/Delete Menu Items	17
4.8.1	Description and Priority	17
4.8.2	Stimulus/response sequences	17
4.8.3	Functional Requirements	18
5.	Nonfunctional Requirements	19
5.1	Performance Requirements	19
5.2	Safety Requirements	19
5.3	Security Requirements	19
5.4	Software Quality Attributes	19
5.5	Business Rules	21
Appendix A: Glossary		22
Appendix B: Analysis Models		23
	Dish State Diagram	23
	Data Flow Diagram	24
	Use Case Diagram	25
Appendix C: To Be Determined List		26

Revision History

Name	Date	Reason for Changes	Version
RMS	02-03-2023	First version's specifications are defined	1.0

1. Introduction

1.1 Purpose

This is web application that aims to digitalize the process of various restaurant management operations including ordering and inventory management and POS. This document aims to capture the system requirements and features particularly related to ordering and inventory management to be implemented in version 1.0

1.2 Document Conventions

1.2.1 Priority Conventions

In this complete document, we will mention priority as “low” or “high” throughout the document. Secondly, priorities are only mentioned in section 4 and section 5 along with detailed description of the requirements. Any high-level requirements mentioned elsewhere are assumed to inherit priorities of their detailed counterparts in section 4,5.

1.2.2 Fonts Conventions

Throughout this document, All the user entities are written in capitalizations i.e. first letter as capital. Also, any significant term which has been described in the glossary is made bold and italic in the text. On the other hand, those terms which are significant (but not described in glossary) are bold in text.

1.3 Intended Audience and Reading Suggestions

The purpose of this document is to give a detailed description of the requirements for the software. It will illustrate the purpose, scope and complete description for the development of system. It will also explain external interface requirements and system requirements as well as non-functional requirements. This document is primarily intended to be proposed to a customer for its approval and also for further processing such as additions to be developed in later releases.

Customers can refer to section 3 and 4 for the list of requirements implemented in Version 1.0. Users are advised to refer to user documentation section for tutorials and online support information.

.

1.4 Product Scope

This web app is a restaurant management system developed with the intention of automating the day to day tasks in a restaurant like order and inventory management, bill generation and taking feedback. This release of the software would deal with these tasks only whereas more areas might be automated in the future versions of this software. The main purpose is to improve the performance of the restaurant by eradicating the daily paperwork. With this system the tasks would be performed in less amount of time and more efficiently. An additional benefit of this software is that during the rush hours the load can be balanced effectively, and restaurants would perform better than usual. In addition to this, human error that occurs when performing tasks manually is also minimized and presence of queues in the system to assign tasks to chefs can reduce congestion in the kitchen. The system would also result in reduction of labor which would result in the reduction of expenses of the restaurant. Feedback module would help the restaurant check for how well they are performing, and monthly/yearly figures can be checked by the billing module to see the trends in sales and profits. These benefits can potentially result in generation of more revenues for the restaurant.

1.5 References

1. IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2. Overall Description

2.1 Product Perspective

This web app will attempt to replace the traditional manual ordering process and is a new self contained software system. The application will be used for ordering and interacting with the inventory while the database will be used for storing the inventory and ordering related information about the food items like pending and complete order queue.

2.2 Product Functions

Given below are the major functions that can be performed using DineOut app. Moreover, a Data Flow Diagram (DFD) for better understanding of the system is also given in Appendix B.

The system will:

- Add Store
- Add Groups
- Add employees in respective stores and in groups
- Add Table with status and availability
- Add Categories of food available
- Manage as well as add order
- Manage and add menu
- Enter details about restaurant

2.3 User Classes and Characteristics

There are four types of users that interact with our system (See appendix B). Firstly, there is a Hall Manager, then Customer, Head Chef and Admin. We'll provide an interface for Chefs as well through which they are looking at the status of their order queues, but they will not interact with our system.

2.3.1 Customer Class

Customers interact with our system directly in order to place order, modify order, get bill and give feedback. We do not store any information related to customers in our system. The process of order taking starts from customers placing order and then the other series of events begin.

2.3.2 Head Chef/Kitchen Manager

Head Chef can mark a dish as prepared when a chef tells him to do so. He can approve the cancellation of an order whenever a customer edits or removes a dish from his order. He can also assign a dish to a particular chef based on the specialty of the chef.

2.3.3 Chef

Chefs don't interact with the system. They just have to look at the dishes present in their queues and prepare the dishes accordingly. Chef's name, address and specialty etc. are stored in the database.

2.3.4 Admin

Admin's job is to manage the inventory and other information related to menu and chefs in the system.

2.3.5 Hall Manager

Hall Managers will provide its input when he marks the bill as paid when customers pay for their order or get the bill printed. Moreover, he gets a notification whenever a particular order is complete, or some customer asks for help through the system. Hall manager can also see tables in the hall and their status i.e. empty or filled.

2.4 Operating Environment

It is an application running on a PC and the PCs are present in a restaurant. Firstly, manager would be present at the entrance and system in his PC would show the tables that are empty/reserved. There would be a tab present at every table for customers which they will use to give order. When an order is placed the server would notify the head chef/ kitchen manager who would be in the kitchen. Head chef would use his PC which also would have the system installed and would add the order to the appropriate queues of the chefs. The chefs would be present in the kitchen area and their interface would allow them to check for the dishes they have to prepare. So, the system is running on various devices but the operating environment and purpose of each is different for each user.

2.5 Design and Implementation Constraints

2.5.1 Operating System Constraint

The computers that have the software installed must have a strong internet connection and have a windows or a mac operating system. The main computer system must have strong security measures to ensure security of the system.

2.6 User Documentation

The software is accompanied by the following materials for further help:

- User Manual Version 1.0
- Online support at the website

2.7 Assumptions and Dependencies

One assumption about the software is that it will always be used on PCs that have enough resources to run the application. If the PC does not have enough hardware resources available for the application, there may be scenarios where the application does not work as intended or not even at all.

The application uses database for online storage of information like orders and menu items that needs to be in working state. If the interface changes the application needs to be adjusted accordingly

3. External Interface Requirements

3.1 User Interfaces

1. Table Interface

Used to manage tables and the orders placed by the customers on those tables.

2. User Interface

Used to manage users that are using the system. This includes all the users that are part of the system.

3. Admin Interface

As Admin is authorized to perform **CRUD** operations on Staff Members, Menu Items and Inventory Items. He'll be having three different screens for Staff Members, Menu Items, and Inventory.

3.2 Hardware Interfaces

Our system can interact with a hardware device directly. We have to connect our system to the bill printer for handing the hard copy of the bill to the customer. For billing module, we may have to use a credit card reader for payment, but the interaction and the results generated by that reader are just entered into our system manually by the user. Moreover, the central screen in kitchen which will be displaying the status of order queues.

3.3 Software Interfaces

- For Database services system shall use to phpmyadmin latest version
- System will run on any operating system with a high speed internet connection.

3.4 Communications Interfaces

This is a web based application and will use HTTPS protocol for communicating with the server.

4. System Requirements

4.1 Place Order

4.1.1 Description and Priority

The system will give customers the ability to place their orders using our product. It will display a list of available and unavailable dishes in the menu where unavailable dishes will be grayed out. Customer will be able to select multiple dishes and their quantity for a particular order.

Priority:
high

4.1.2 Stimulus/Response sequences

When user enters the order activity/page, initially system displays a list of available and unavailable dishes along with their prices.

1. Stimulus:

Waiter taps on an available dish.

Response:

System shows a popup having name of the dish and price per serving. Also, it contains a text box for the customer to enter the quantity, OK button and a Cancel button.

1.1. Stimulus:

Waiter taps on an unavailable dish.

Response:

Nothing happens.

2. Stimulus:

Waiter enters the quantity and press OK button.

Response:

System closes the popup, shows a small green tick mark at the side of dish. Below the tick mark it shows quantity selected and total price of that dish.

2.1. Stimulus:

If Waiter taps on cancel button after 1.

Response:

the system closes the popup, and nothing happens.

3. Stimulus:

Waiter taps on confirm order button at the bottom

Response:

System closes the order screen and displays a timer along with a “Cancel Order” button and “Edit Order” button

4.1.3 Functional Requirements

REQ-1: The system will show a list of cards (UI element) of dishes. Each card will have a picture of the dish. Below the dish it shows the price in Rupees per serving.

REQ-2: The system must show all available and unavailable dishes to the Customer.

REQ-3: Tap on any of the displayed dish will result in a popup for quantity and a green mark after quantity has been selected.

REQ-4: The popup for quantity input will not allow the user to enter letters, negative numbers or any invalid characters.

REQ-5: After completing the order the system will display a timer “Time to complete the order” and it is the total time required to serve the dish keeping in view the previously queued orders. Moreover, it also shows a cancel order button.

REQ-6: Unavailable dishes must be displayed but their operations must be disabled.

4.2 Help**4.2.1 Description and Priority**

Our system will provide help for the customer if the customer faces issues in using the tab, there will be a ‘help’ option in his interface. If he faces issues in using the tab or want some other assistance, he can notify the hall manager through the system.

Priority:

high

4.2.2 Stimulus/Response sequences

The home screen for the customer shows a help button on top right corner of the screen.

4. Stimulus:

Customer taps on the help button.

Response:

System shows a popup with two buttons, “Call a waiter to manage order”, “Call a waiter for help”

5. Stimulus:

Customer taps on any of the button.

Response:

System closes the popup and sends a notification to the Hall Manager. Notification will include the table number of the Customer.

4.2.3 Functional Requirements

REQ-2: The system must give Customer the ability to ask for help.

REQ-1: When the customer taps on “Call the manager to manage order”. the system must store that the above order was given by the waiter.

4.3 Edit Order**4.3.1 Description and Priority**

Waiter can edit the order any time before the serving. In editing mode, the customer can change the quantity of the of the food ordered, add and remove dishes from the order.

priority:

high

4.3.2 Stimulus/Response sequences

The timer screen shows two buttons “Cancel Order” and “Edit Order” button

1. Stimulus:

Waiter taps on “Edit Order” button.

Response:

System shows the previous menu screen where selected dishes are already marked with green tick.

2. Stimulus:

Customer taps on any of the selected dish.

Response:

System opens a popup with previous quantity pre-filled. This popup will also contain a button "Remove Dish".

3. Stimulus:

Waiter enters new quantity mentioned by the customer and press "OK"

Response:

System shows an error "Cannot edit <Name> dish" or System closes the popup and new quantity will be displayed on that dish in the list.

4. Stimulus:

Customer taps on "Remove Dish"

Response:

system responds with "Dish <Name> removed" or "Dish <Name> cannot be removed"

4.3.3 Functional Requirements

REQ-1:

System must allow the Customer to increase, decrease or even remove the dish from the order any time before serving.

REQ-2:

System must remove the dish or decrease quantity of the dish with the approval of head chef.

4.4 Cancel Order

4.4.1 Description and Priority

Our system will also provide an option to cancel the current order. When the waiter taps on the "Cancel Order" button. Customer can cancel the order at any time before serving.

priority:

high

4.4.2 Stimulus/Response sequences

1. Stimulus:

Waiter taps on the “Cancel Order” button

Response:

system responds with a popup “Order canceled successfully” or “Order cannot be cancelled”

4.4.3 Functional Requirements

REQ-1:

System must allow the customer to cancel order at any time before serving.

REQ-2:

In cancel order, all the dishes will be presented for approval to the head chef. Only approved dishes will be dropped.

4.5 Request Bill

4.5.1 Description and Priority

Request bill option gives the ability to the customer to ask for receipt and pay the bill.

priority:

high

4.5.2 Stimulus/Response sequences

1. Stimulus:

Customer taps on the request bill button

Response:

The system prints the bill through a printer. System will add a bill to the hall manager’s view with the button that says “paid”.

4.5.3 Functional Requirements

REQ-1: The system must notify the hall manager that a customer has request for a bill

REQ-2: The system must show Hall manager the order no, table no and total payable amount

REQ-3: The system must give ability to the hall manager to change the status of the bill to paid.

4.6 Customer Feedback

4.6.1 Description and Priority

The system will give customers the ability to give a feedback for the food or overall services. In the feedback screen there are multiple choice questions each having two options “Satisfactory” and “Unsatisfactory”. At the end there is a submit button.

priority:
high

4.6.2 Stimulus/Response sequences

1. Stimulus:

The customer taps on request bill option

Response:

the system shows a feedback screen with multiple choice questions and a submit button.

4.6.3 Functional Requirements

REQ-1: System must show the feedback screen to the user.

REQ-2: System must display multiple choice questions for feedback.

4.7 Add/Edit/Delete Staff Members

4.7.1 Description and Priority

The system gives ability to the admin to add, edit and delete staff members. Using this feature an admin can add chefs, waiters, managers.

priority:
high

4.7.2 Stimulus/Response sequences

Admin/Manage screen shows a grid of staff members. There is a button at the top of grid which says Add Member. In the grid after every entry there is a “Edit” and “Remove” button.

1. Stimulus:

Admin taps on “Add Staff” button

Response:

System opens another screen with a form

2. Stimulus:

Admin fills the information and hit submit

Response:

System responds with “<Staff Member> added successfully”

3. Stimulus:

Admin taps on edit button

Response:

System opens a screen with a form prefilled with the existing values.

4. Stimulus:

Admin edits the information and hit submit

Response:

System responds with “<Staff Member> edited successfully”

5. Stimulus:

Admin taps on remove button on a particular row

Response:

responds with a “<Staff Name> removed successfully”

4.7.3 Functional Requirements

REQ-1: Admin should be able to add all necessary information about the staff member

REQ-2: System must give admin the ability to edit information about all staff members

REQ-3: System must give admin the ability to remove staff members.

4.8 Add/Edit/Delete Menu Items

4.8.1 Description and Priority

The system gives ability to the admin to add, edit and delete staff members. Using this feature an admin can add chefs, waiters, managers.

priority:
high

4.8.2 Stimulus/response sequences

Admin screen shows all the previously added dishes. It also shows a “Add Dish” button along with “Edit” and “Remove” with all the available dishes

1. Stimulus:

Admin taps on “Add Dish” button

Response:

System opens another screen with a form

2. Stimulus:

Admin fills the information and hit submit

Response:

System responds with “<Dish> added successfully”

6. Stimulus:

Admin taps on edit button

Response:

System opens a screen with a form prefilled with the existing values.

7. Stimulus:

Admin edits the information and hit submit

Response:

System responds with “<Dish Member> edited successfully”

8. Stimulus:

Admin taps on remove button on a particular row

Response:

responds with a “<Dish> removed successfully”

4.8.3 Functional Requirements

REQ-1: Admin should be able to add all necessary information about the staff member

REQ-2: System must give admin the ability to edit information about all staff members

REQ-3: System must give admin the ability to remove staff members.

4.9 Add/Edit/Delete Tables

4.9.1 Description and Priority

The system gives ability to the admin to add, edit and delete tables. Using this feature an admin can add various new tables and manage them.

priority:

high

4.9.2 Stimulus/response sequences

Admin screen shows all the previously added dishes. It also shows a “Add Dish” button along with “Edit” and “Remove” with all the available dishes

1. Stimulus:

Admin taps on “Add Table” button

Response:

System opens another screen with a form

2. Stimulus:

Admin fills the information and hit submit

Response:

System responds with “<Table> added successfully”

9. Stimulus:

Admin taps on edit button

Response:

System opens a screen with a form prefilled with the existing values.

10. Stimulus:

Admin edits the information and hit submit

Response:

System responds with “<Table Member> edited successfully”

11. Stimulus:

Admin taps on remove button on a particular row

Response:

responds with a “<Table> removed successfully”

4.9.3 Functional Requirements

REQ-1: Admin should be able to add all necessary information about the table

REQ-2: System must give admin the ability to edit information about all tables

REQ-3: System must give admin the ability to remove tables.

5. Nonfunctional Requirements

5.1 Performance Requirements

The system must be interactive, and the delays involved must be less. So, in every action-response of the system, there are no immediate delays. In case of scrolling through the menu there should be a delay of no more than 2 second before the next page of menu items is displayed otherwise our people's dining experience is affected. The order should be placed in pending orders and be visible to the head chef/chefs in less than 1 second to start the preparation.

Cancel Order/ updates must be made with little delay to avoid delivery delay. Also, when connecting to the Firebase server the delay to make a successful connection should be less for effective real time communication.

5.2 Safety Requirements

The software is completely environmentally friendly and does not cause any safety violations. The menu will have a flexible font that can be zoomed so as to not over constrain the eyes.

5.3 Security Requirements

There is a need for a proper and encrypted login authentication for head chef and admin as employee sensitive information as well as inventory should be protected from hacking. Information transmission should be securely transmitted to database without any changes in information to avoid disturbances in orders and billing

5.4 Software Quality Attributes

5.4.1 Adaptability:

There can be a change in the menu and information stored in the database about employees and inventory.

5.4.2 Availability:

The system is up and running for most of the time and server is not down for more than a few minutes to avoid inconvenience of the customers.

5.4.3 Correctness:

The bill generated by the application must be accurate and the orders placed should exactly be the same which the user has selected.

5.4.4 Flexibility:

If need arises in the future, software can be modified to change the requirements.

5.4.5 Interoperability:

The data is transferred from the customer's end to the kitchen and then head chef assigns orders to each chef. This way data is transferred from one part of the system to another.

5.4.6 Maintainability:

Software can be easily repaired if a fault occurs.

5.4.7 Portability:

Software can be easily installed on devices and would run smoothly according to the requirement.

5.4.8 Reliability:

No matter how many orders are placed, system must give the correct results.

5.4.9 Reusability:

Current version can be used in the future versions with more functionality added.

5.4.10 Robustness:

Software must have checks to ensure that the items that are not available in the menu cannot be selected and the emails, phone numbers added are all valid.

5.4.11 Testability:

All the requirements are fulfilled, response time is low, and all functions are working perfectly.

5.4.12 Usability:

Interface of the software must be easy to use. It would not be complex since managers, chefs have a view, so interface should be simple.

5.5 Business Rules

1. Manager's interface contains the view of tables that are free, and manager can just view and doesn't provide any input to the system.
2. Once the bill is paid, manager can mark the order as paid.
3. Admin has access to perform add, delete, update operations on the database for menu, inventory, employees and no other person can modify the data in the db.
4. Customers can place order from the list of available items and can update order and pay bill.
5. Head chef assigns orders to chefs and can update the queues and has an additional functionality of load balance.
6. Chefs can only view the orders and cannot remove an order from their queue. Only head chef can interact with the queues containing orders

Appendix A: Glossary

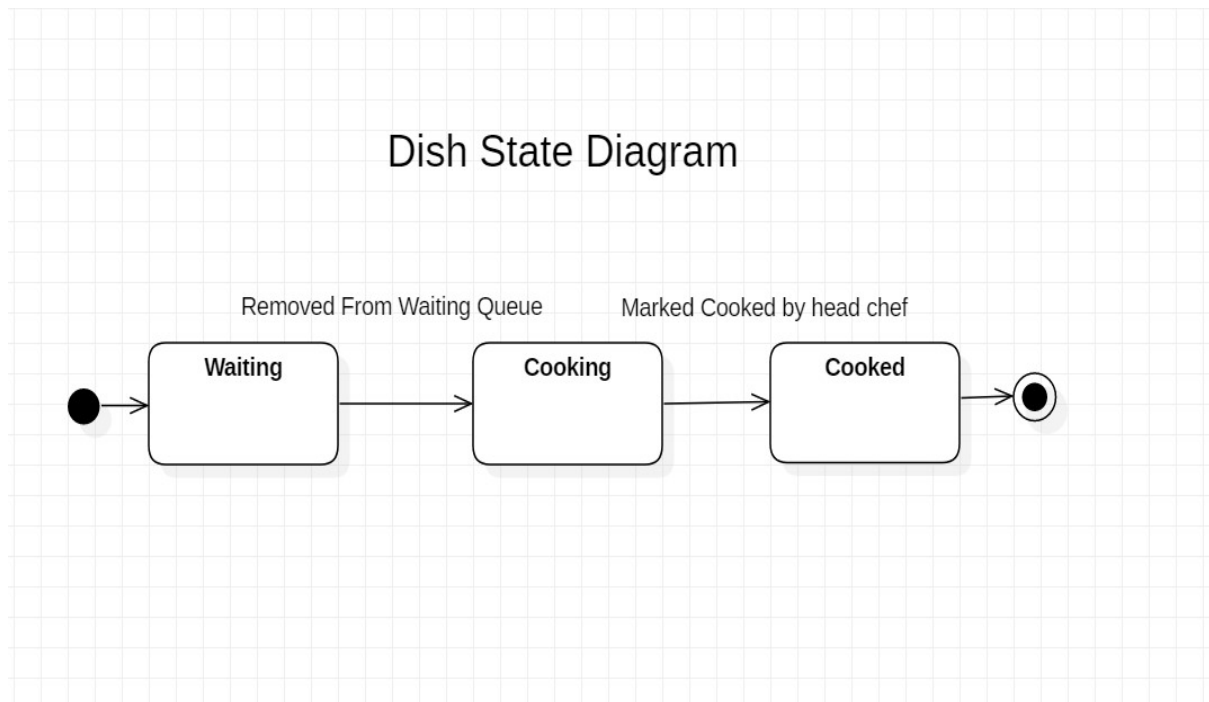
CRUD: *In computer programming, create, read, update, and delete (CRUD) are the four basic functions of persistent storage. Alternate words are sometimes used when defining the four basic functions of CRUD, such as retrieve instead of reading, modify instead of update, or destroy instead of deleting.*

Print Helper: *It is an android library that is used to connect to remote printer and send commands to that printer for printing.*

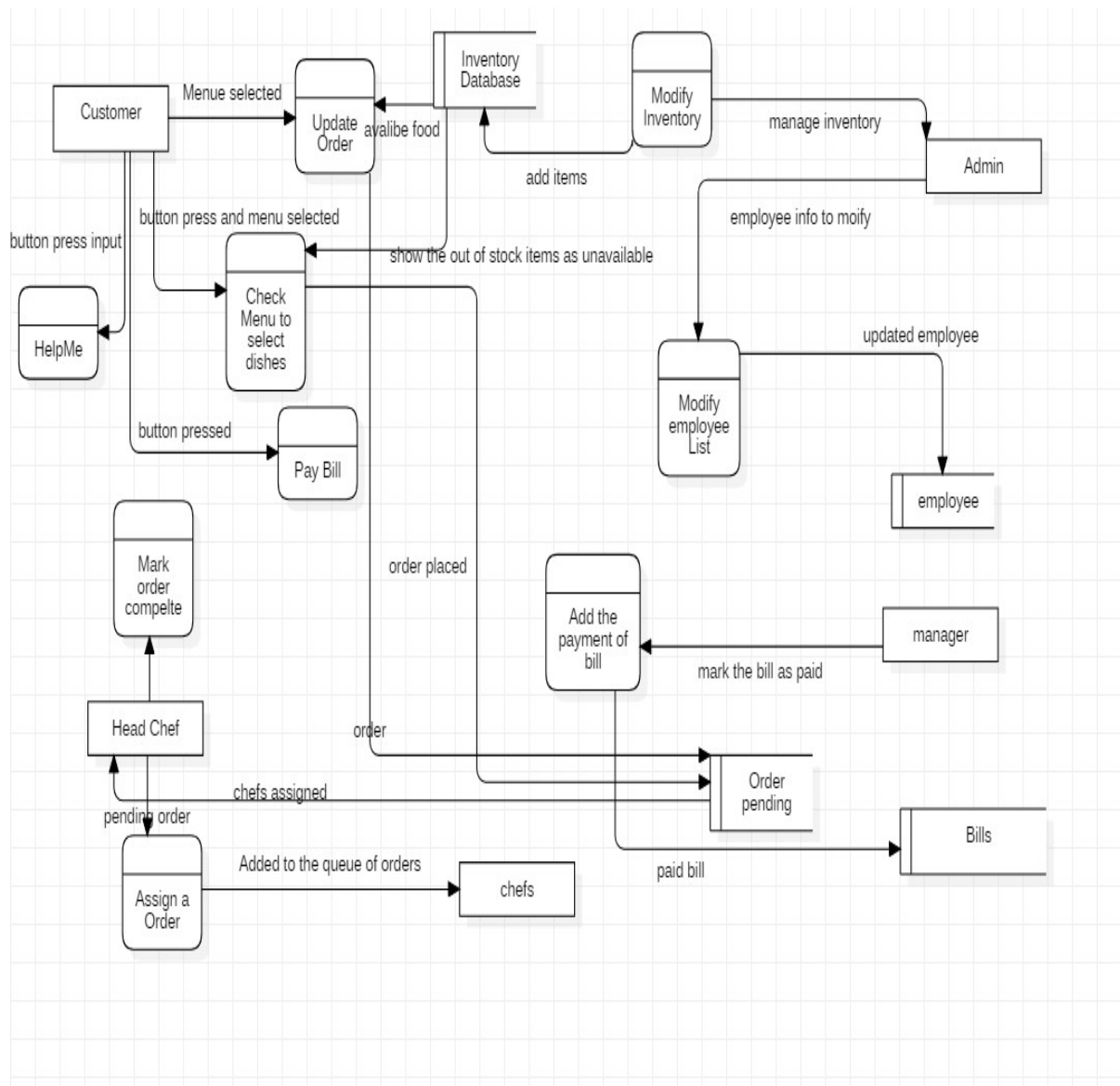
marshmallow 6.0: *Android 6.0 "Marshmallow" is the sixth major version of the Android operating system and the 13th version of Android. First released as a beta build on May 28, 2015, it was officially released on October 5, 2015, with Nexus devices being the first to receive the update.*

Appendix B: Analysis Models

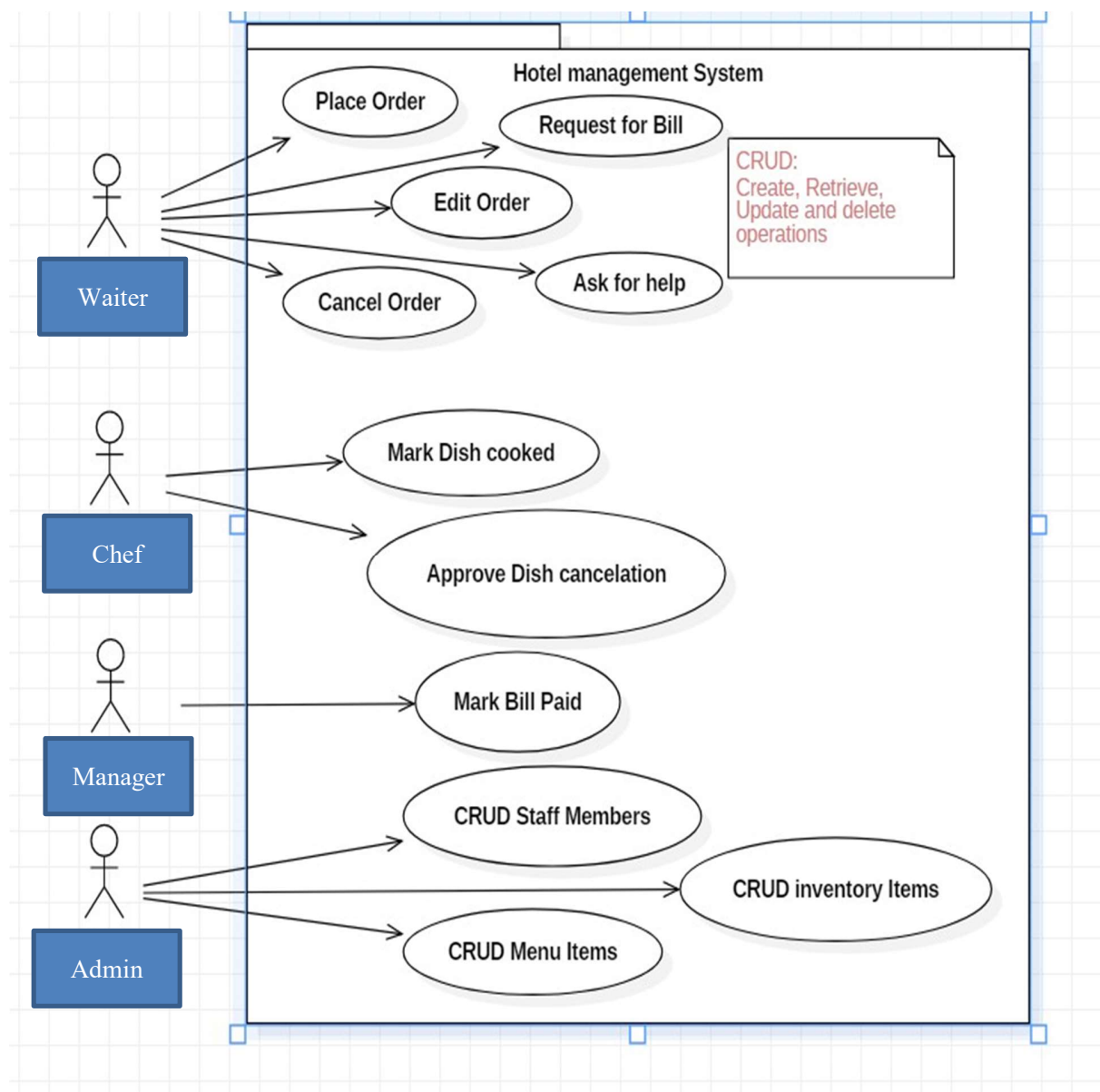
Dish State Diagram



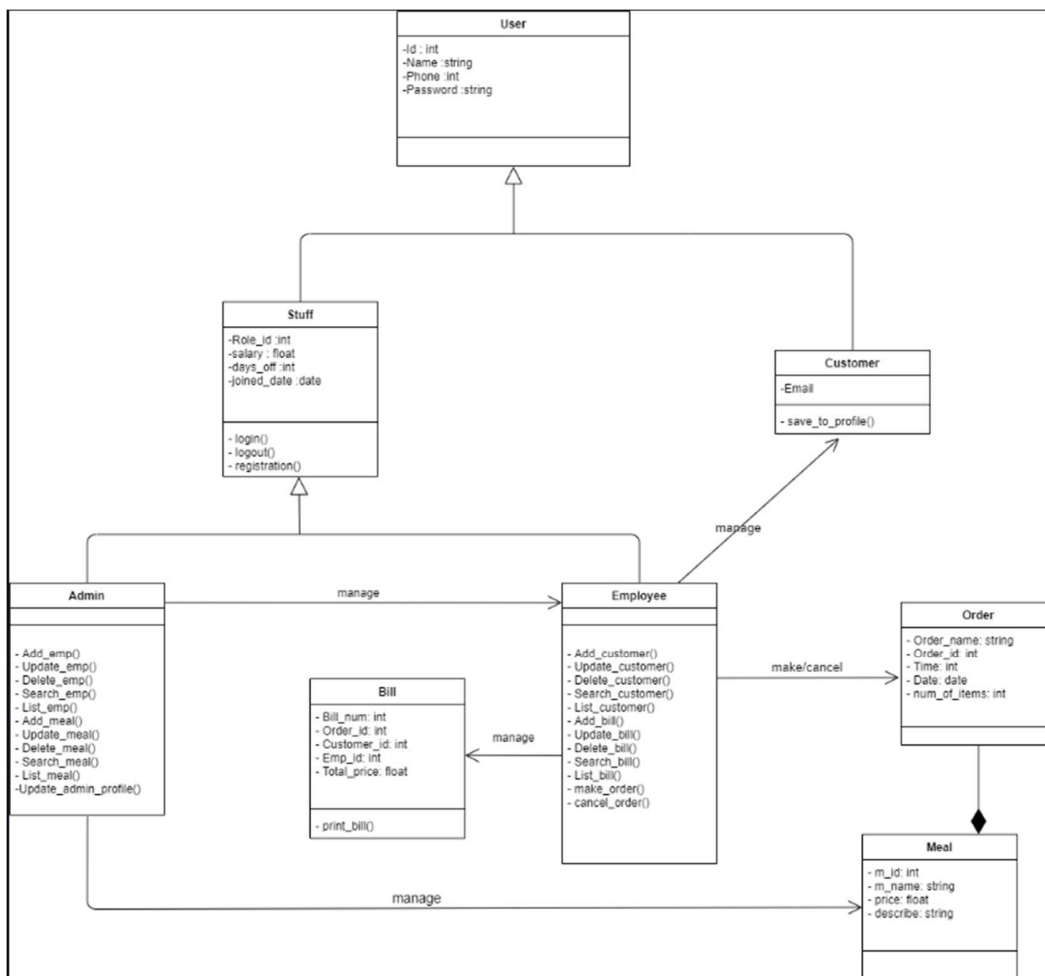
Data Flow Diagram



Use Case Diagram



Class diagram

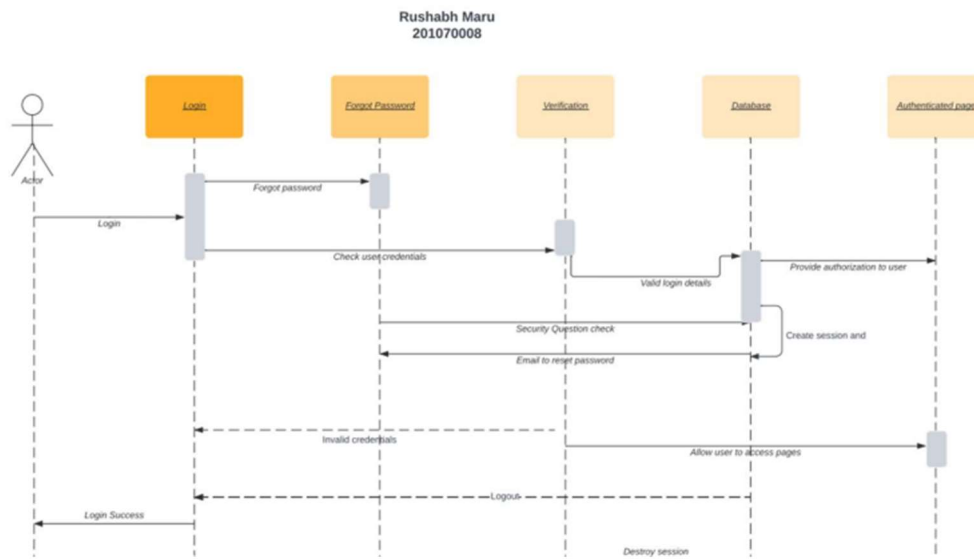


Appendix C: To Be Determined List

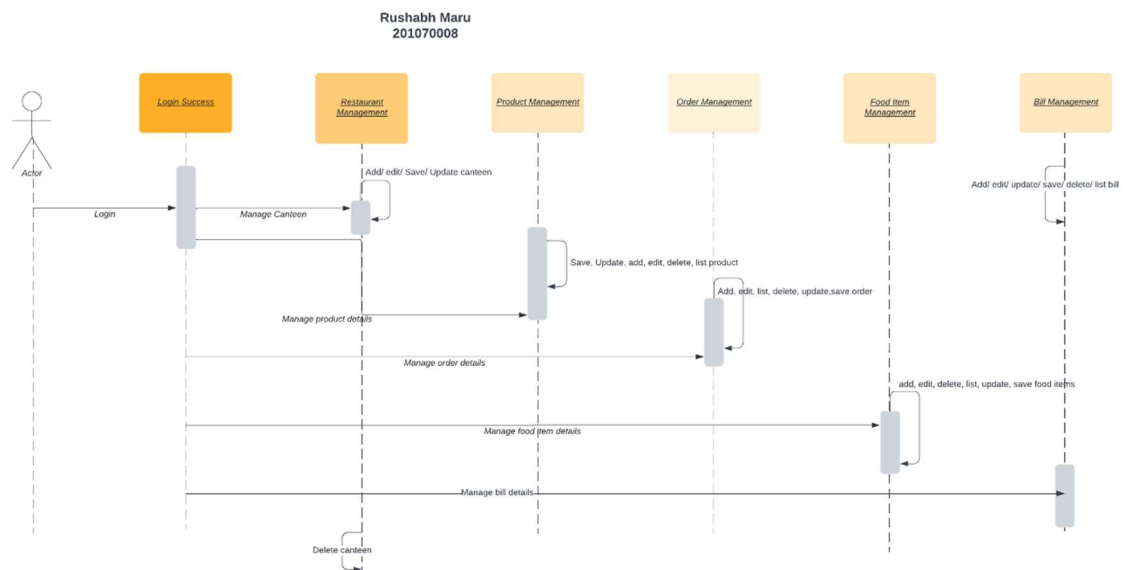
Weekly sales report and tracking most ordered dish and prioritizing its inventory stocking feature (restock the items that are most ordered often) is yet to be determined by the client and may need further meetings for elaboration.

Adding POS (point of sale) features to the application is yet to be determined as well.

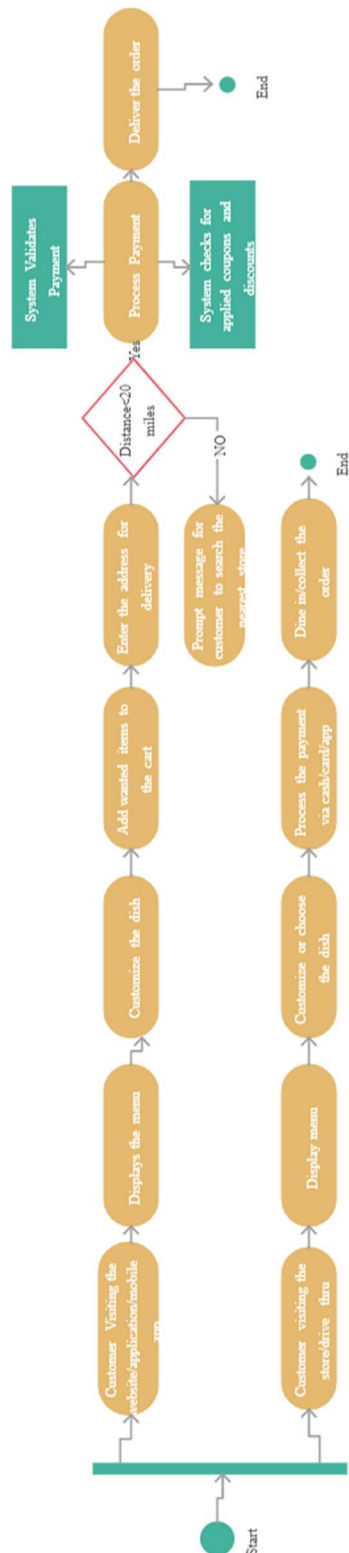
Sequence diagram



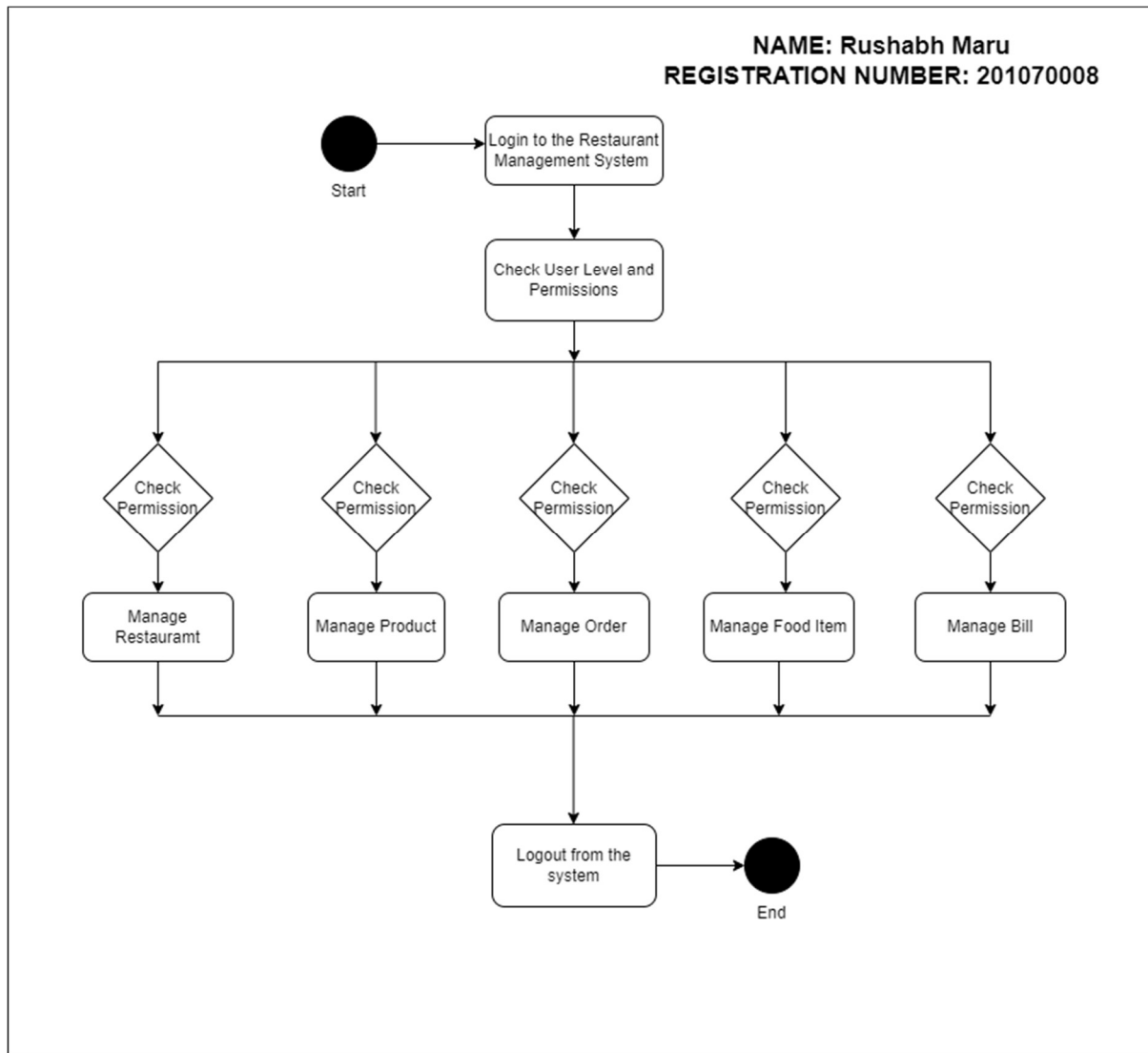
Restaurant management system Sequence diagram Part 2



State chart diagram

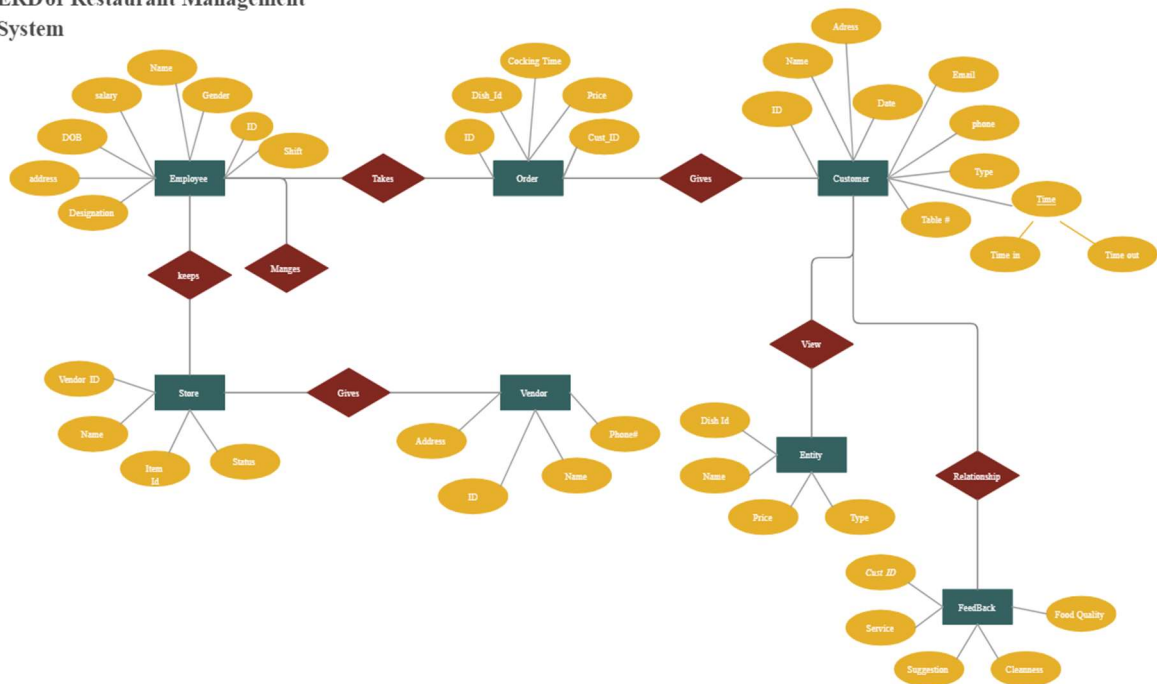


Activity diagram



ER Diagram

ERD of Restaurant Management System



Collaboration diagram

