|  |
| --- |
| Software Design Specification  **<QuickMeal.com>**  **Project Code: 01**  **Internal Advisor:**  **Sir Abid Raffique**  **External Advisor:**  **Sir Saad Razzaq**  **Team Leader:**  **Muhammad Waqas**  **Project Team:**  **BSCSF18M016 (Muhammad Waqas)**  **BSCSF18M036 (Hammad Hassan)**  **BSCSF18M039 (Shan Abbas)**  **Submission Date:**  **19/12/2021** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Project Manager’s Signature**Document** |

**Document Information**

|  |  |
| --- | --- |
| **Category** | **Information** |
| University | UOS |
| Project | Quickmeal.com |
| Document | Software Design Specification |
| Document Version | 1.0 |
| Identifier |  |
| Status | Draft |
| Author(s) | Muhammad waqas  Shan Abbas  Hammad Hassan |
| Approver(s) | PM |
| Issue Date | Sept. 15, 2003 |
| Document Location |  |
| Distribution | 1. Advisor: Sir Abid Rafique 2. PM: Sir Saad Razzaq 3. Project Office |

**Definition of Terms, Acronyms and Abbreviations**

*This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.*

|  |  |
| --- | --- |
| **Term** | **Description** |
| ASP | Active Server Pages |
| DD | Design Specification |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Table of Contents**

[**1.** **Introduction 4**](https://docs.google.com/document/d/1oZZVZIeEKKOpdCZ2Z_TgtD6m3ntVmm9D/edit#heading=h.1y810tw)

[1.1](#_heading=h.tyjcwt) Purpose of Document 4

[1.2 Project Overview 4](https://docs.google.com/document/d/1oZZVZIeEKKOpdCZ2Z_TgtD6m3ntVmm9D/edit#heading=h.4i7ojhp)

[1.3](#_heading=h.1t3h5sf) Scope 4

[**2.**](#_heading=h.4d34og8) **Design Considerations 4**

[2.1 Assumptions and Dependencies 4](https://docs.google.com/document/d/1oZZVZIeEKKOpdCZ2Z_TgtD6m3ntVmm9D/edit#heading=h.2xcytpi)

[2.2 Risks and Volatile Areas 4](https://docs.google.com/document/d/1oZZVZIeEKKOpdCZ2Z_TgtD6m3ntVmm9D/edit#heading=h.1ci93xb)

[**3.**](#_heading=h.3rdcrjn) **System Architecture 4**

[3.1](#_heading=h.26in1rg) System Level Architecture 4

[3.2](#_heading=h.lnxbz9) Sub-System / Component / Module Level Architecture 5

[3.3](#_heading=h.35nkun2) Sub-Component / Sub-Module Level Architecture (1…n) 5

[**4.** **Design Strategies 5**](https://docs.google.com/document/d/1oZZVZIeEKKOpdCZ2Z_TgtD6m3ntVmm9D/edit#heading=h.3whwml4)

[4.1 Strategy 1…n 5](https://docs.google.com/document/d/1oZZVZIeEKKOpdCZ2Z_TgtD6m3ntVmm9D/edit#heading=h.2bn6wsx)

[**5.**](#_heading=h.2jxsxqh) **Detailed System Design 5**

[**6.** **References 6**](https://docs.google.com/document/d/1oZZVZIeEKKOpdCZ2Z_TgtD6m3ntVmm9D/edit#heading=h.qsh70q)

[**7.**](#_heading=h.3j2qqm3) **Appendices 6**

**Section 1**

# Introduction

## Purpose of Document

Purpose of the document is to explain every single detail of the project to the reader. So, whenever a new person wants to know about the idea of the project that we are going to do. The reader can just read out the introductory part of the document and he will get the whole overview of the project and why and what is going to be performed in this project.This document will provide details about what our project will be about and what resources we will be needing. What are the possible constraints in our project? For example if we have a limited amount of time and or a low budget available by our client. Who is going to interact with our system?

## Project Overview

Nowadays everything we do is based on technology. The workload is now shared by many shoulders. If we particularly talk about the restaurant's traditional systems, there is a need for a waiter to come first to take the order and people have to wait. And in the present condition of Covid-19 there is a need for social distancing. Automated table to table food quickmeal.com is such a facility which minimizes the resources and makes the service better. In restaurants people have to call the waiter staff to order the food or people have to stand in queue to order the food and it consumes precious time of the people and the hotel management have to hire more staff to take orders from customers. So, we built quickmeal.com that automatically takes the order and people will wait on their seats without discussing anything with staff. Our system will minimize the staff required. Therewill be special discounts, recommendations and mystery gifts for the regular customers. Timeof order will start after placing the order. There will be different times for each order category. When 3-5 minutes will left there will be a reminder for order completion. For regular customers, the system will predict the foods most liked. It will predict Sales of the coming days by looking at previous history.

## Scope

### Main Modules

**Module I Customer**

* The customer can log in to system
* The customer can use guest system
* The customer can order their meal
* Reserve table with food
* See menu list
* Add or Delete food to favorites
* See order history
* Can place a special order with own choice
* Direct communication
* Can order food by persons
* Can get special gifts
* See offers

**Module II Manager**

* Every restaurant has a manager who will manage all the problems regarding a restaurant.
* Manager has a unique Username and password.
* The manager can add, update and delete menu items.
* The manager can manage the reserved tables.
* The manager can see the history of sales.
* The manager can see the prediction of next week's sale.
* The manager can add special gifts to the menu.

# Design Considerations

The proposed system “QuickMeal.com;” is an on-line web-based application. This application will provide different modules like ordering, manage ordering, store history of the orders and sale, announce special gifts for the customer and predict the sales for better business. Customer views have limited access. They can give orders and delete or update his/her order and check the special deals while Manager view can edit the menu, add the special offers to menu, check the available stock and see the predicted sale of next day or week items.

## 2.1 Design and Implementation Constraints:

The product is developed using the Html CSS Javascript for the front end and python Node JS, php and mysql for the backend. The product is accomplished with a login facility so that specific function is available to specific users.

## 2.2 Assumption and Dependencies:

**The product needs the following third-party products:**

* Visual studio code to develop the product
* Google chrome browser
* Xampp for apache server
* Mysql

**Assumptions:**

* Coding should be error free.
* Application should be user friendly.
* All the information should be stored in a database that is accessible by site.
* It should provide fast access to the database.
* Users may access info if he/she has internet connection.
* Users can use a login or guest account.

**Dependencies:**

* The Manager should have proper understanding of product
* Any Update Date must be entered into the system.

## 2.3 User characteristics

The product has two users:

* Customers order the food from quickmeal.com.
* The manager to view and manage the orders and data of the site.

## 2.4 Operating environment

The application will run on android devices (smartphones and tablets). For proper functioning, it will require a database.

**System constraints**

* **Software constraints**

Programming will be done in Python, Php, MySql, and JavaScript and the development environment will be an VS Code.

* **The cultural constraints**

Interface is only in English, no other language option is available.

* **User constraints**: This application is for community people, so they should have basic knowledge of using tablets.

# 3. External Interface Requirements

**Software Interfaces:**

**Programming language:** JavaScript(front end) and Python, Php (back end)

We will use Html5, JavaScript, CSS for layout, how the screen looks.The elements show on screen. To set the attributes of these elements (e.g the text color, background colour, visibility, font, width, height and much more).

We will use Python and Php for coding. This page controls all the elements of xml with time.

**Application:** VSCode (we will use VSCode to write the programs to develop the app)

**Database:** XAMPP server 2021(we will use a database to manage large amounts of data, Data security, and easy data search).

**Operating system:** Android **(**smartphone and tablets).

The operating system sets permission for All the files in the app so the user id assigned to that app can access them.

Mostly the data that we exchange between the software or applications will be the definitions of data.

**4. Functional Requirements:**

This section contains the functional requirements of the system. The possible requirements according to the relevance to the user of the system are customers, waiters, chefs and supervisors.

**General:**

* A server will host the “QuickMeal.com” and provide system data processing and storage capability
* It will provide a customer with all customer system functionality.
* A display will provide a chef with all chef system functionality.
* All system functionality will be accessible once you open the system.

**Customer will be able to;**

* Check the deals and discounts once they open the system.
* Cancel their order in the system.
* Create an empty pending order through their engaged menu.
* Add an item to a pending order by dragging the item from the engaged menu onto the order.
* Delete an item from a pending order by dragging the item off the order.
* Add a special dietary requirement to an order.
* Call for waiter assistance through their engaged menu.
* to drag a tip denomination into a bankcard payment or a cash payment.

**Waiter will be able to:**

* Log into the system using their assigned username and password.
* Log out the system.
* A waiter assigned to a table will be alerted via their wireless tablet when:
  + An order is placed from that table
  + An item ordered by that table is rejected by the kitchen
  + An item ordered by that table is ready to be served
  + The table has requested waiter assistance

**System will allow a waiter to;**

* Accept an order placed by a customer.
* Reject an order placed by a customer.
* Indicate the delivery of an item to its customer.
* Process a payment using cash.
* Process a payment using a bankcard.

**A chef will be able to:**

* Accept a customer’s order item through the system.
* To reject a customer’s order item through a display.
* To indicate that a customer’s order item is ready to be served through system

**Manager will be able to:**

* Do everything a waiter can.
* Do everything a chef can.
* Abort/purge a customer’s meal from the active system with no expectation of payment.
* Abort/purge a table's account/meals from the active system with no expectation of payment.
* Issue a refund for one or more items to a customer.

# 5. Non-functional Requirements

This subsection presents the identified non-functional requirements for the subject “QuickMeal.com”.

**5.1 Performance Requirements:**

* The server shall be capable of supporting no less than 200 concurrent connections.
* It will provide no limit on how many devices are in the system.
* The server will be capable of supporting an arbitrary number of active orders, that is, no orders be lost under any circumstances.
* The server will be capable of supporting an arbitrary number of active customer payments, that is, no payments will be lost under any circumstances.
* Any element of the system will take no longer than 10-seconds to restart.
* It would not dismiss an engaged menu unless the customer requests it.

**5.2 Safety Requirements:**

* The system will be capable of restoring itself to its previous state in the event of failure.
* The system will be able to display a menu at all times to facilitate manual order taking should the need arise.
* The system will utilize periodic 30-second keep-alive messages.

**5.3 Security Requirements:**

* A waiter password used for login must have a bit-strength of at least 64 bits.
* A waiter password used for login must be changed every three months.
* A waiter that attempts to log into a second device while already logged into another device will be rejected and notified.
* The system will provide two levels of access:
  + A supervisor level for unrestricted access to system functionality.
  + A waiter level for access to waiter functionality.
* System will require a user to log in using a username and password and it will also provide an option as a guest account.

# 6. System Architecture

*This section should provide a high-level overview of how the functionality and responsibilities of the system are partitioned and then assigned to subsystems or components. The main purpose is to gain a general understanding of how the system is decomposed, and how the individual parts work together to provide the desired functionality.*

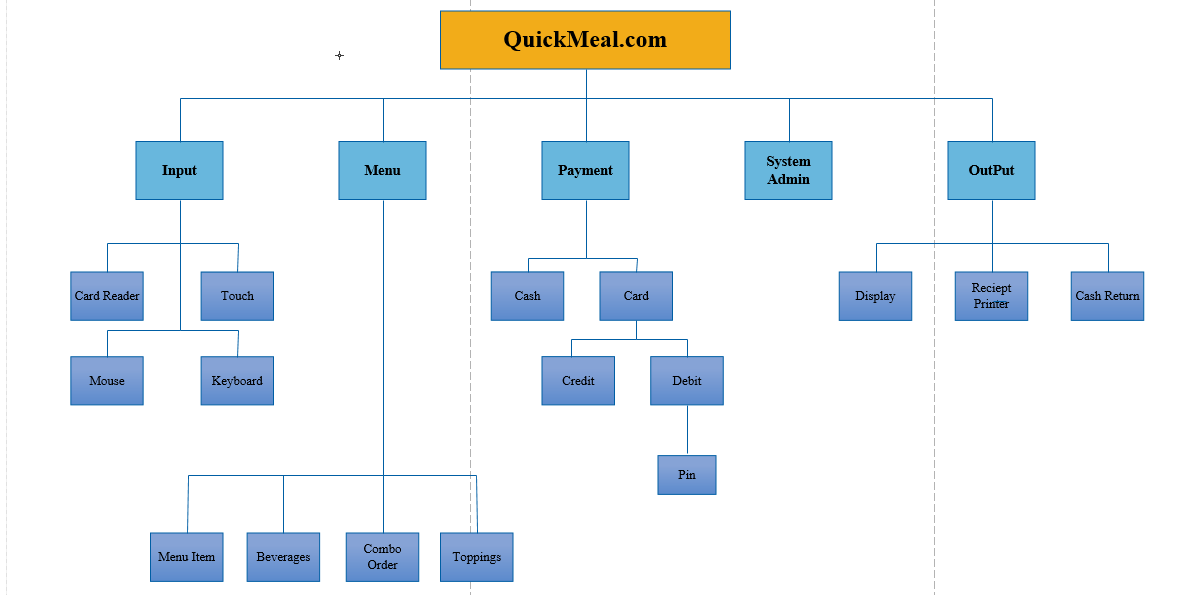
## 6.1 System Level Architecture

*The architecture should decompose the system at a top level in a way that provides a foundation for more detailed design work. The architecture discusses relationships and roles of system elements (subsystems, components, modules, etc.), but does not provide internal details. Areas for consideration are:*

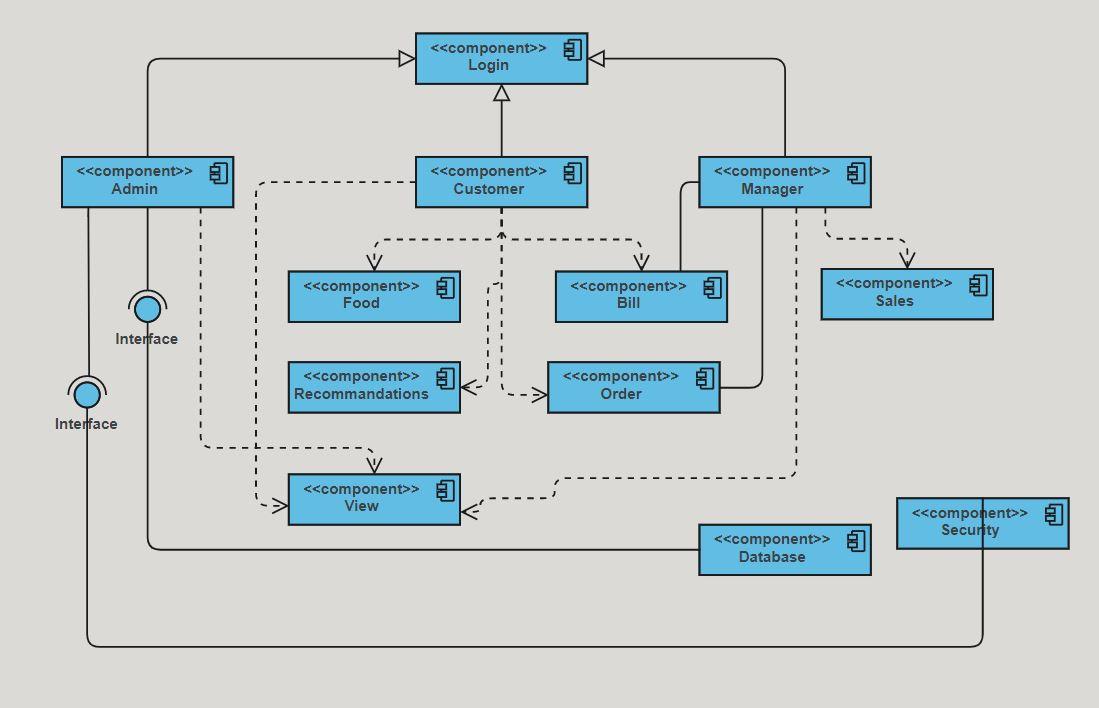
* *System decomposition into elements*
* *The relationship between the elements*
* *Interfaces to external systems*
* *Major physical design issues such as where elements will execute*
* *Global design strategies such as error handling*

**Preliminary System Architecture:**

In the preliminary implementation, the "QuickMeal.com" was partitioned into elements for input, menu, payment, system administration and output.

**

**System level architecture**

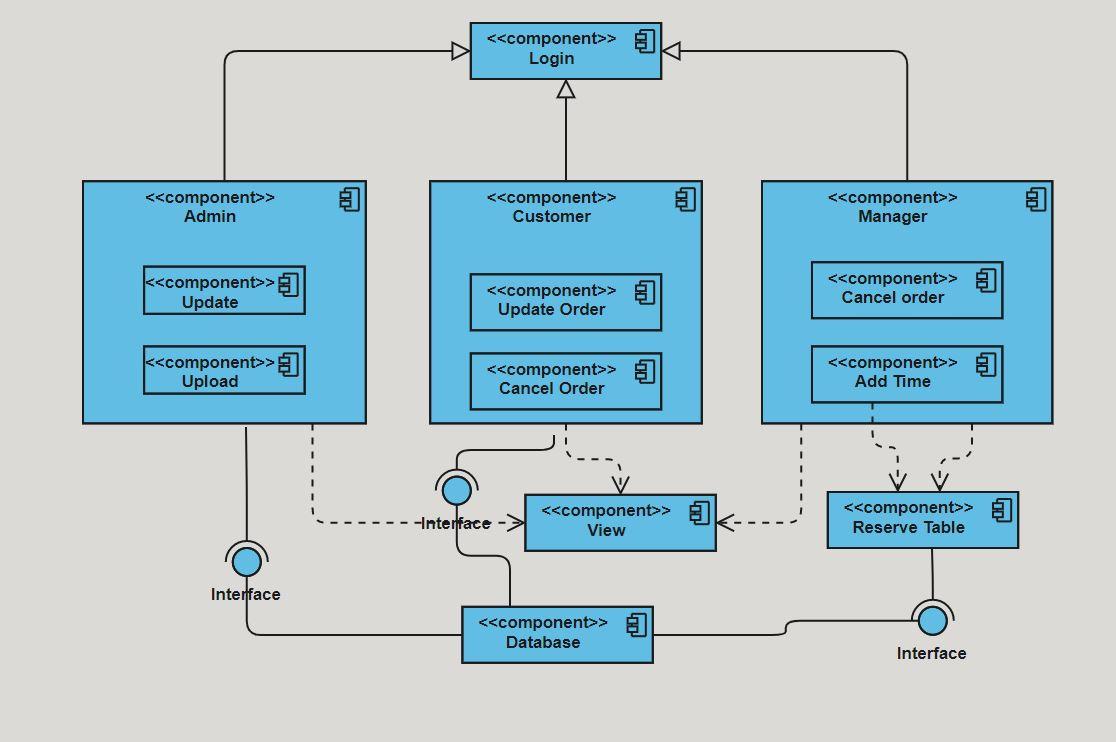
**

**System Level Architecture**

*NOTE: You may use appropriate UML diagrams (Package and Deployment diagrams) to document the overall system architecture. Similarly you can describe the top-level decomposition of the system using Component diagrams.*

## 6.2 Sub-System / Component / Module Level Architecture

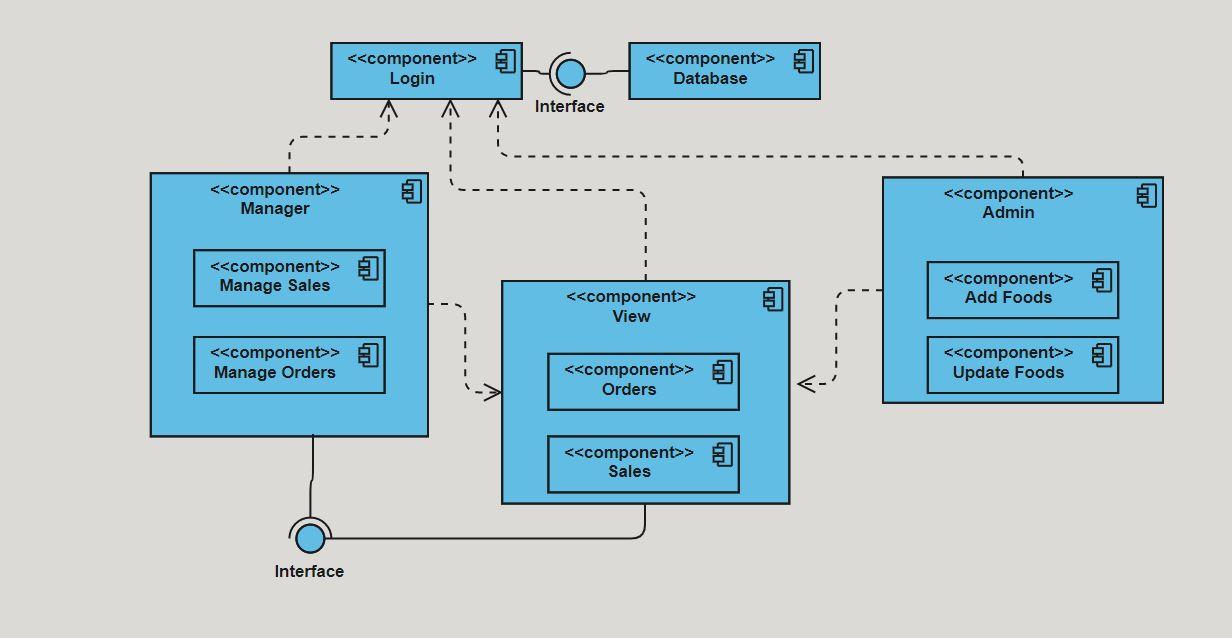
*Identify the sub-systems, component or modules of your overall software system and provide their diagrammatic view using appropriate detailed architecture diagram presenting how the system is further divided into sub systems, components and modules and the relationships and interactions between them.*

**

**Subsystem/Module level architecture**

## 6.3 Sub-Component / Sub-Module Level Architecture (1…n)

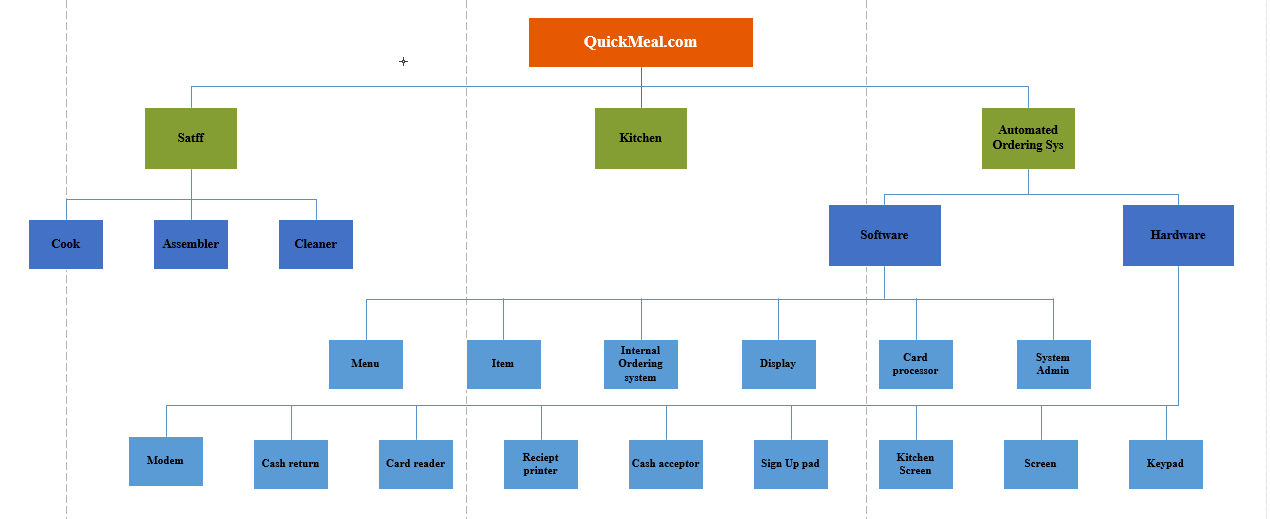
*Identify all the sub components or sub modules (if any) of the already identified modules and components. Provide their diagrammatic view using appropriate detailed architecture diagram presenting how those sub systems, modules and components are further divided into sub components and sub modules and how they interact with each other.*

**

Subcomponent/Submodule level architecture

**Revised System Architecture:**

Version two of the system structure has a larger scope. The highest-level of system structure is a composition of three systems -- staff, kitchen and automated food ordering system.

**

**Revised System Architecture**

# 7. Use Cases:

# Use Case 1. Place Order

# Primary Actor: Customer Description: Customer places an order from the available choices after indicating his language preference for the session. Pre-conditions: System is connected to a power source, display is turned on and system is configured to accept the inputs. Flow of Events:

# User selects his language preference for the session.

# User selects from the menu.

# User selects from the drinks menu

# User selects from the combo deals

# User confirms the order

# Alternative Flow of Events:

# User accidentally presses a wrong button and after realizing it he hits the backspace button.

# User enters the wrong order and wants to go back to the main menu.

# Post-condition: Order has been made that goes to the kitchen for processing. Assumption: User is familiar with how to enter values through the mouse and has a general idea why the inputs are being provided and what is expected out of the system.

**Use Case 2.** Make Payment

* **Primary Actors:** customer, Credit/Debit system, cash collector.  
  **Description:** The user is asked for the mode of payment. The payment is accepted in terms of credit/debit card or is collected by a cash collector. And the customer is given a token with their order number.  
  **Pre-condition:** The order has been confirmed and the total bill has been displayed on the screen to the customer. Customer decides to go ahead with the order.
* **Flow of Events:**
  1. User enters the mode of payment. (Credit/Debit/Cash)
  2. User makes the payment in cash
  3. Cash collector collects the money and gives back the change if required.
  4. User makes the payment by credit/debit card.
  5. User receives a token number and final bill.
* **Alternative Flow of Events:**
  1. User selects the mode of payment.
* **Post-condition:** Customer waits for the order to be processed.  
  **Assumption:** User is familiar with how the system works and what is expected out of the system.

**Use Case 3.** Update Menu.

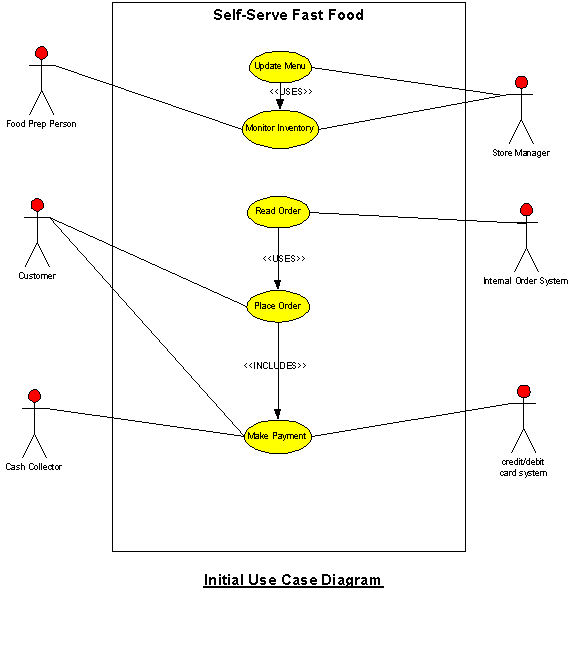
* **Primary Actor:** Store Manager.  
  **Description:** The menu might change according to the inventories or add/delete items from menu and deals. The prices of each item might change for the period of time.  
  **Pre-condition:** An order menu with their respective price already exists in the system in some particular format.  
  **Flow of Events:**
  1. The Store manager enters the system with some password.
  2. The Store manager makes the required changes.
  3. The Store manager saves the changes and logs out.
* **Alternative Flow of Events:**
  1. Some of the menu might not need any change.
  2. Users might enter invalid passwords and need to go back.
* **Post-condition:** A menu list will be displayed when the user enters the system.  
  **Assumption:** The Store manager is given the rights and privileges to enter the system and make the required changes.

**Use Case 4.** Monitor Inventory.

* **Primary Actor:** Food preparation person, Store Manager **Description:** This use case triggers when an item goes out of stock.  
  **Pre-condition:** None  
  **Flow of Events:**
  1. Food preparation person/Store manager notices an item out of stock
  2. Updates the menu accordingly.
* **Post-condition:** A new and updated menu list will be displayed.  
  **Assumption:** The Store manager is given the rights and privileges to enter the system and make the required changes.

**Use Case 5.** Read Order.

* **Primary Actor:** Food preparation person, Internal Order system.  
  **Description:** Internal order system reads the order once the customer confirms his order and then he communicates the order to the food preparation person.
* **Pre-condition:** User confirms the order.  
  **Flow of Events:**
  1. Internal order system reads the order
  2. Communicates the order to the food preparation person
* **Post-condition:** The final order is being processed in the kitchen.  
  **Assumption:** Food preparation person is available to take the order and know the sequence of processing the orders.



# Design Strategies

*Describe the design strategies or decisions that impact the overall organization of the system and its high-level structures. This information should provide the reader with insights into the key abstractions and mechanisms used in the system architecture.*

## 7.1 Strategy 1…n

*For each strategy, discuss the reasoning employed (possibly referring to previously stated design goals and principles) and any trade-offs. Areas for consideration include:*

* *Future system extension or enhancement*
* *System reuse*
* *User interface paradigms*
* *Data management (storage, distribution, persistence)*
* *Concurrency and synchronization*

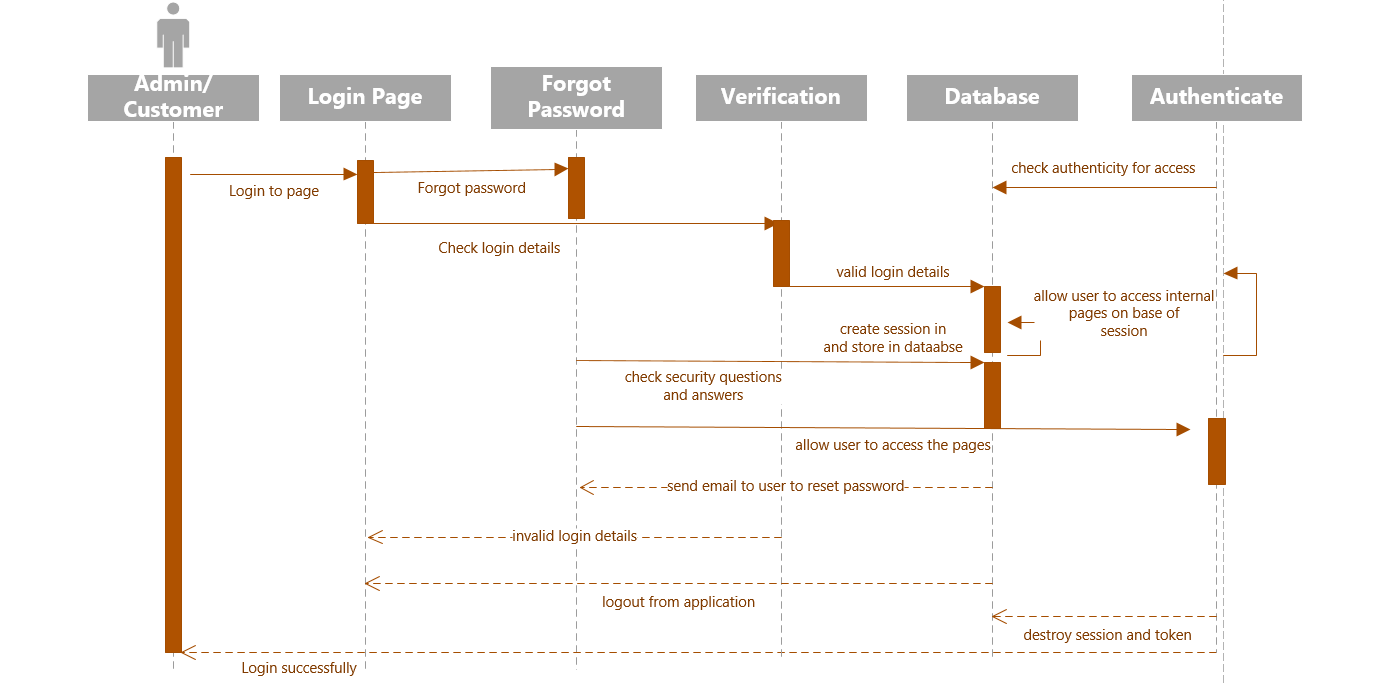
# Detailed System Design

*A detailed design should include the following:*

* *Detailed class diagram along with a detailed description of all attributes, functions or methods specifying interactions between different classes/modules.*
* *Detailed Sequence diagram with parameter list*
* *State Transaction Diagram*
* *Logical data model (E/R model)*
* *Physical data models*
* *Detailed GUIs*

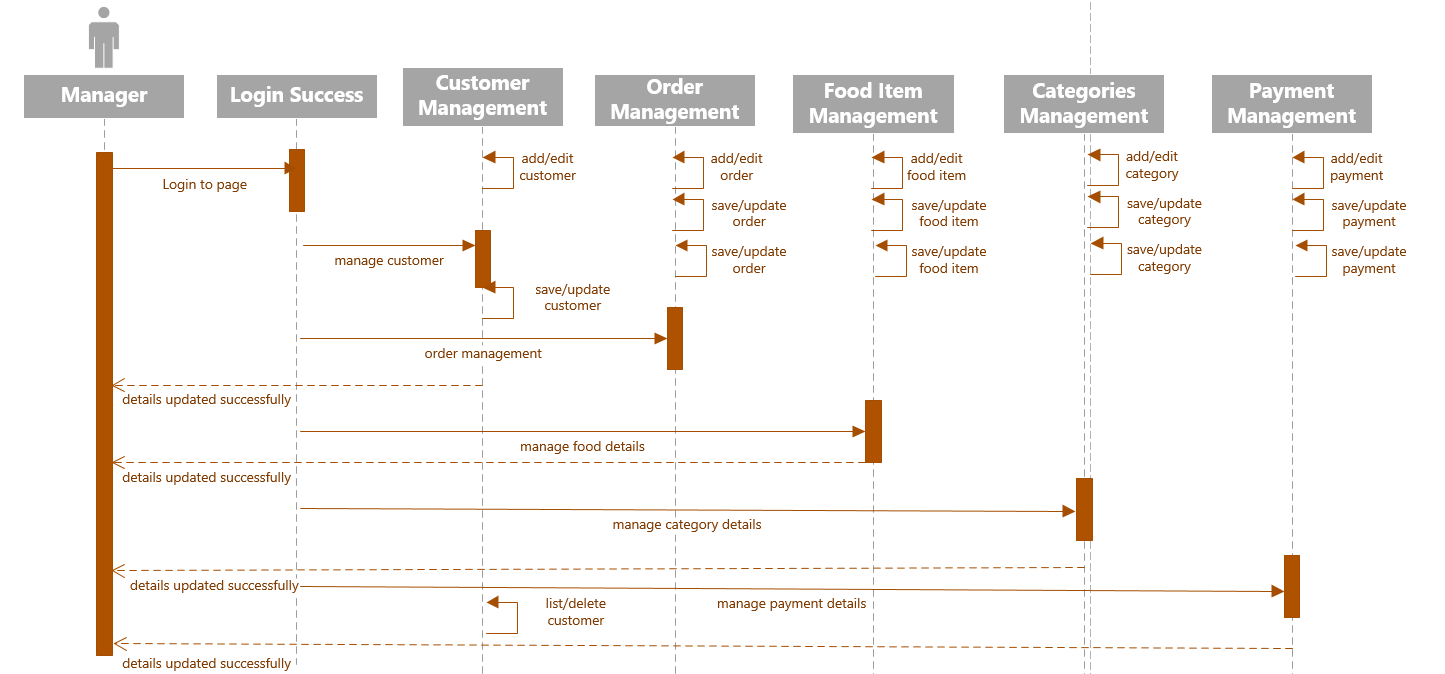
**5.2 Sequence Diagrams:**

**5.2.1 User Login:**

****

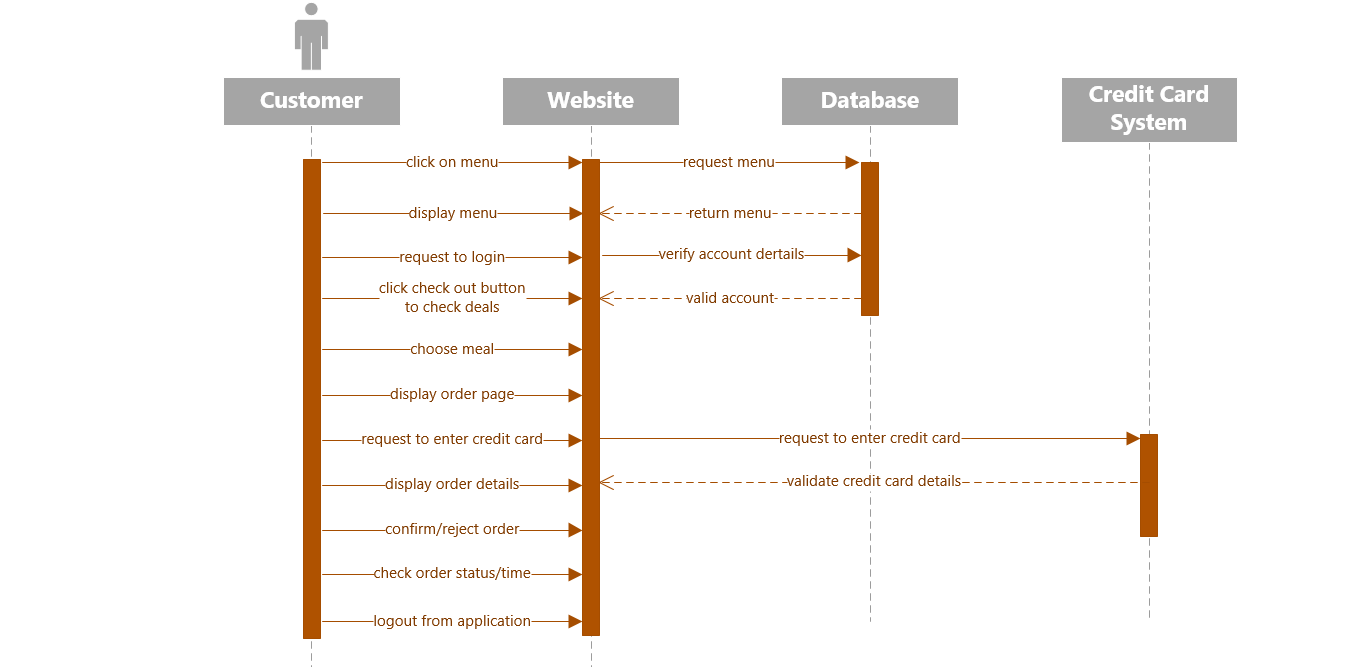
**User Login Sequence Diagram**

**5.2.2 Admin/Manager:**

****

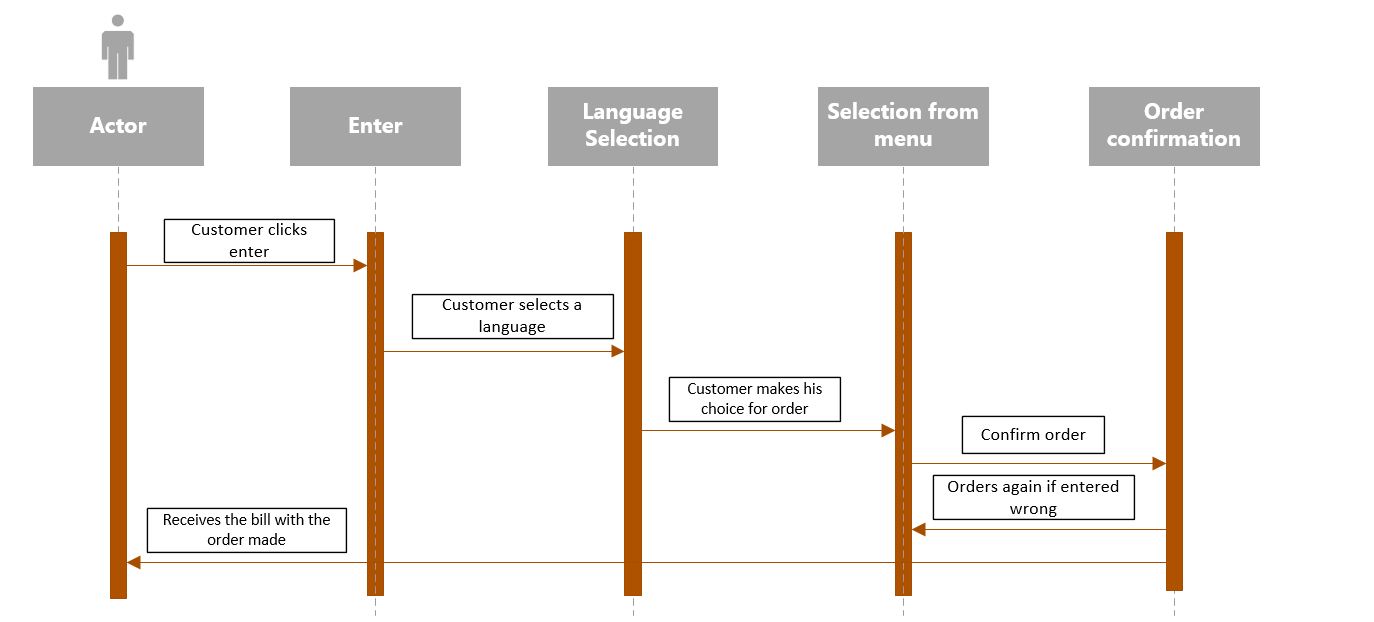
**Admin Login Sequence Diagram**

**5.2.3 Customer:**

****

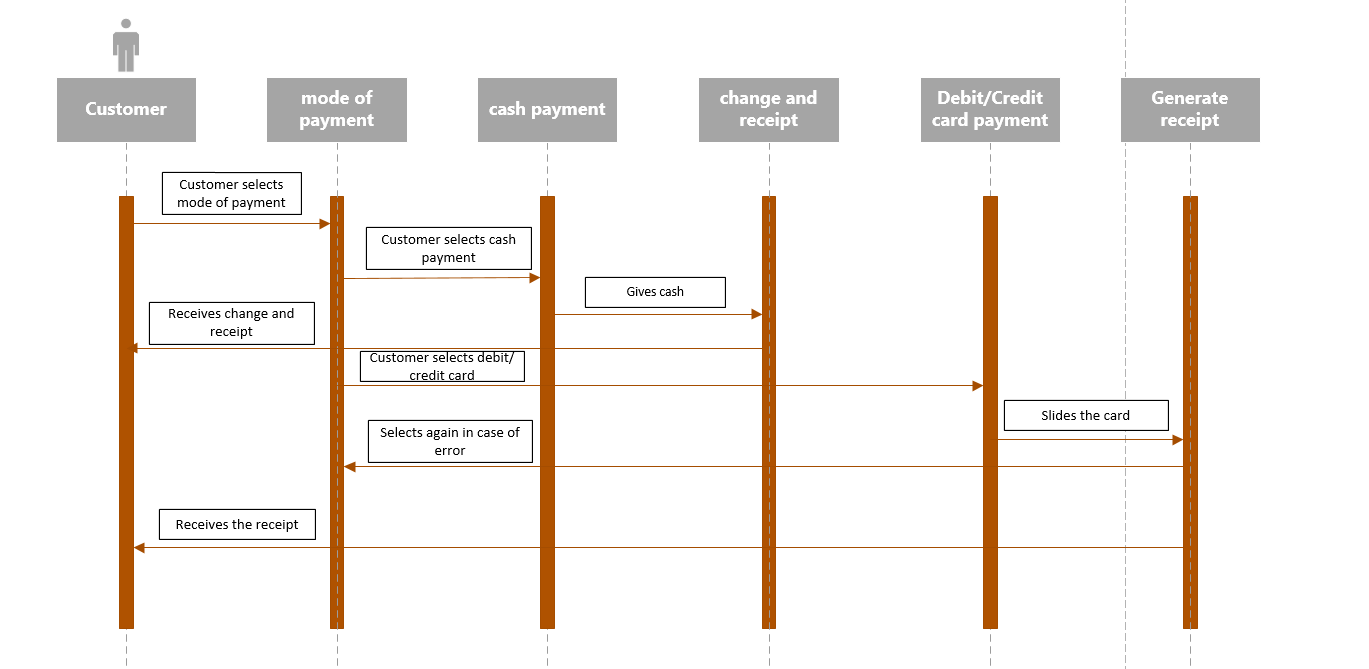
**Customer Interaction Sequence Diagram**

**5.2.4 Place Order:**

****

**Place Order Sequence Diagram**

**5.2.5 Make Payment:**

****

**Make Payment Sequence Diagram**

**Classes, Their attributes and Operations:**

**Food Class:**

***Attributes:***

* menuid
* name
* description
* price
* type // type of food
* prepared
* served

***Operations:***

* setfood() //Used to set the food in the menu
* modifyfood() //Used to update the food in the menu
* deletefood() //Used to delete the food item

**Customer Class:**

***Attributes:***

* id
* name
* mobile
* address
* regular
* favourites

***Operations:***

* checkmenu() //Customer will check the menu items
* checkdeals() //Customer will check the active deals
* confirmorder() //Customer will confirm order
* cancelorder() //Used to cancel order
* makepayment() to make payment from credit or debit card

**Deals Class:**

***Attributes:***

id

title

price

description

***Operations:***

* displaydeals() //Used to display the deals
* update() //Used to update the deals in the menu
* delete() //Used to delete the deals

**Order Class:**

***Attributes:***

* id
* customerid
* foodid

***Operations:***

* makenotification()
* setorder()
* vieworder()
* modifyorder()

**Payment Class:**

***Attributes:***

* id
* customerid
* orderid
* date
* total

***Operations:***

makepayment() //Used by customer to make payment

display() // It will display the payment details

**Category Class:**

***Attributes:***

* id
* name
* type
* status //category is active or inactive
* description
* image

***Operations:***

add() //Used to add categories

delete() // Used to delete the category

update() // Used to modify or update category

**Rating Class:**

***Attributes:***

* menuid
* customerid
* ID
* score
* remarks
* data recorded

***Operations:***

* display() //It will display the ratings.

**Waiter Class:**

***Attributes:***

* ID

***Operations:***

**Chef Class:**

***Attributes:***

* ID

***Operations:***

* markcookedfood()//Used to mark the cooked food

**Staff Class:**

***Attributes:***

* staffid
* naem
* jobtype
* description
* salary

***Operations:***

* Add() //Used to add the staff member
* modifyinfo() //Used to modify the info of staff

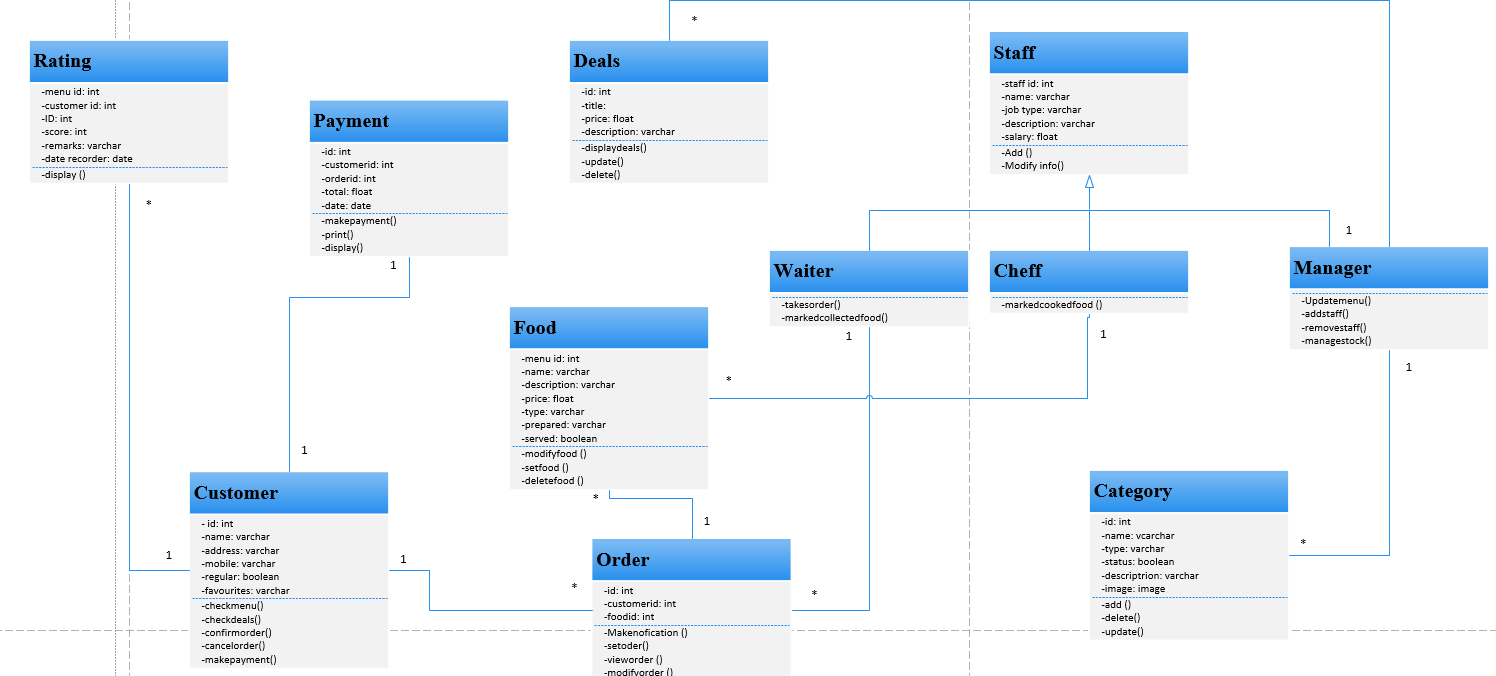
**Manager Class:**

***Attributes:***

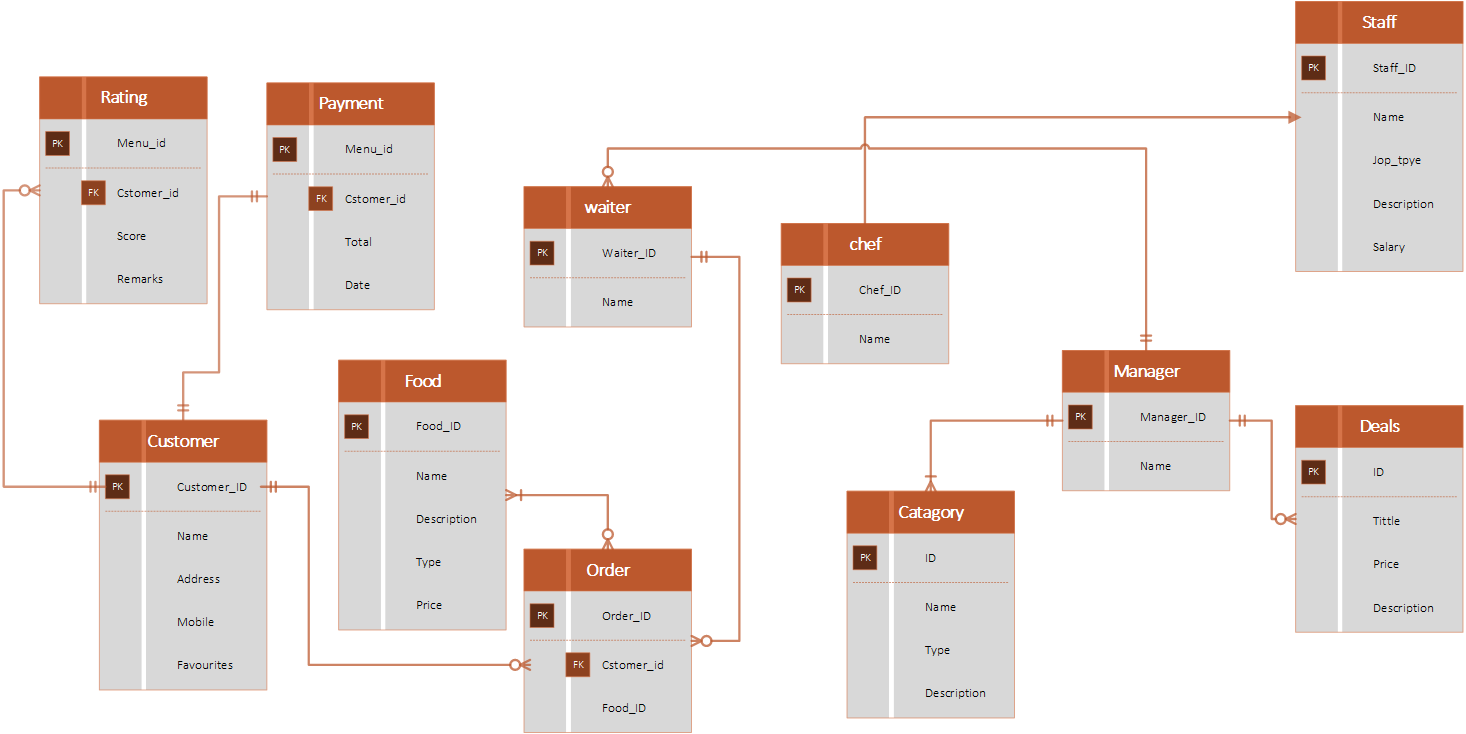
***Operations:***

* updatemenu() //Used to update the menu
* addstaff() //Used to add the staff
* removestaff() //Used to remove the staff
* managestock()//Used to manage the staff

**5.3 Class diagram:**

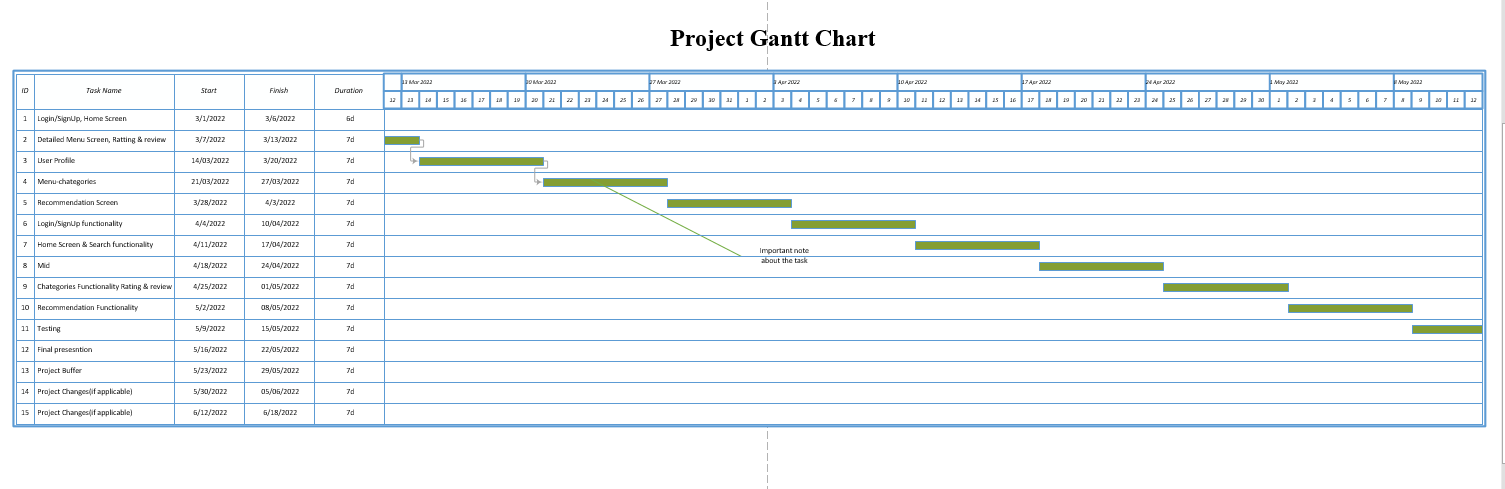


**UML Class Diagram**



**ERD Diagram**

**Implementation Plane:**

****

# References

*This section should provide a complete list of all documents referenced at specific point in time. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained (This section is like the bibliography in a published book).*

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref. No.** | **Document Title** | **Date of Release/ Publication** | **Document Source** |
| PGBH01-2003-Proposal | Project Proposal | Oct 20, 2003 | https://docs.google.com/document/d/1XL0oewuPvyWsFxAeBf2qQ91VkDHRInUa/edit?usp=sharing&ouid=110372962024288664465&rtpof=true&sd=true |
| PGBH01-2003-FS | Functional Specification | Oct 20, 2003 | https://docs.google.com/document/d/1T2et8qigHllne5Rt\_x\_OximV\_rUJGdq1/edit?usp=sharing&ouid=110372962024288664465&rtpof=true&sd=true |
| PGBH01-2003-FS | Software requirement Specification | Oct 20, 2003 | https://docs.google.com/document/d/1UerRwxlLSnkQdmlAFdjjVmWTyS62SCG\_/edit?usp=sharing&ouid=110372962024288664465&rtpof=true&sd=true |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

# Appendices

*Include supporting detail that would be too distracting to include in the main body of the document.*