**SOFTWARE REQUIREMENT SPECIFICATION**

**FOR**

**DIGITAL TABLE SERVICE**

**[26-06-2019]**

**CDAC, MUMBAI**

**Revision History:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version.** | **DATE** | **Authored** | **Reviewed** | **REASON FOR CHANGE** |
|  |  | **By** | **By** |  |
| 00 | 28/06/2019 | Team-05 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**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 ................................................................................................................................ | 3 |

|  |  |  |
| --- | --- | --- |
| **2. Overall Description .........................................................................................................................** | | **3** |
| 2.1 | Product Perspective .................................................................................................................. | 3 |
| 2.2 | Product Functions ..................................................................................................................... | 3 |
| 2.3 | User Classes and Characteristics .............................................................................................. | 3 |
| 2.4 | Operating Environment ............................................................................................................ | 3 |
| 2.5 | Design and Implementation Constraints .................................................................................. | 3 |
| 2.6 | User Documentation ................................................................................................................. | 4 |
| 2.7 | Assumptions and Dependencies ............................................................................................... | 4 |

|  |  |  |
| --- | --- | --- |
| **3. External Interface Requirements** | | **..................................................................................................4** |
| 3.1 | User interface | 4 |
| 3.2 | Hardware interface | 5 |
| 3.3 | Software interface | 5 |
| 3.4 | Communication interface | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| **4.** | **System Features ..............................................................................................................................** | | **5** |
|  | 4.1 | System Feature 1 ...................................................................................................................... | 5 |
|  | 4.2 | System Feature 2 (and so on) ................................................................................................... | 5 |
| **5.** | **Other Non-functional Requirements.............................................................................................** | | **6** |
|  | 5.1 | Safety Requirements ................................................................................................................. | 7 |
|  | 5.2 | Security Requirements ............................................................................................................. | 7 |
|  | 5.3 | Software Quality Attributes |  |
| **6.** | **Other Requirements .......................................................................................................................** | | **8** |

1. **Introduction**
   1. **Purpose**

Generally, in Restaurants waiter comes and collects order on either notepad or tab. So they are

not able to maintain their customer’s records and feedback about their food. Even customer

cannot track his/her order. So **Digital Table Service** is the web application that will provide them all these facility. Customer can track his/her order and after ordering food in mean time they can listen to music.

Also Restaurants owner can maintain database of his customers and collect feedback from them and on the basis of that feedback they can improve. Then for payment, this application has two options either customer can pay by cash or if customer wants to pay through online then they can pay through default payment gateway.

* 1. **Document**

**Convention**

**Headings: -**

Text: -Bold

Font-Size: - 14

Highlighting: - Times New Roman

**Sub Headings: -**

Text: -Bold

Font-Size: - 12

Highlighting: - Times New Roman

**Header: -**

Text: - Simple

Font-Size: -10

Highlighting: - Times New Roman

**Footer: -**

Text: - Simple

Font-Size: -10

Highlighting: - Times New Roman

* 1. **Intended Audience and Reading Suggestions**

This document is intended for developers, users, testers and project managers for the purpose of understanding the design of system in terms of different perspectives. Further, this document contains functionalities and characteristics of system along with the working

environment. It also includes other information related to system such as external interface requirements, features and other non - functional requirements.

* 1. **Product Scope**

Our project is targeted to reduce the manpower and time. It is a process of maintaining day to day records of Restaurants. This application is design in such a way that even if users keep on increasing, the data will be stored in a proper manner that can be very useful for Restaurants for their growth. In short it is easy for Restaurants owners to maintain data in future also.

* 1. **References**
* <https://u.ubereats.com/en-IN/>
* [https://www.zomato.com](https://www.zomato.com/)
* [https://www.swiggy.com/mumbai](https://www.swiggy.com/mumbai/nerul-restaurants)

1. **Overall Description**
   1. **Product Perspective**

Currently in Restaurants waiter comes and collects the order on either notepad or tab then they inform chef about those order, in that sometimes miscommunication happens and also Restaurants owner are not able to maintain their customer’s records and feedback about their food. Even the customer cannot track his/her order.

So **Digital Table Service** will provide all these facility. Customers can track his/her order and after ordering food in the mean time they can listen to music. Also, a Restaurants owner can maintain a database of his customers and collect feedback from them. Then for payment, this application has two options – either customer can pay by cash or if the customer wants to pay through online then they can pay through the default payment gateway.

* 1. **Product Functions**
* The admin will be able to view customers records, Add menu, Add Special Offer, Add

Discount.

* The customer will be able to track his/her order.
* After ordering food in mean time they can listen to music.
  1. **User Classes and Characteristics**

**Admin**:

This person will be able to access all the functions of the system. This person can view Customers record, Add Chef, Add Tables, Add/Update Menu, and Add Tab.

**Customer**:

Customer can view menu, ordered menu list and time which is required for order delivery. Customers will be able to give feedback for food.

**Chef**:

This person will receive the order and as per the availability of menu he can confirm the order.

* 1. **Operating Environment**
* **Hardware platform:**
  1. Processor – Above Intel I3 with clock speed of 2.9 GHz
  2. RAM – 8 GB and Above
  + Hard Disk – above 1TB
* **Software platform:**
  + Front-end: HTML, CSS, and Bootstrap.
  + Back-end: Java and spring.
  + Database : MySQL
* **Supported tools:**
  1. Visual Code Studio, MySQL Workbench, Eclipse
  2. **Design and Implementation Constraints**

**Constraints:**

1. The user interface is only in English. No other language option is available.

o The receptionist can log-in only with his assigned user-name and password.

o Limited to HTTP/HTTPS.

* 1. **User Documentation**

User documentation mainly comprises of Help menu of application. It will give all the minute details about the project, if any user has any query about any module or functionality, one can refer it and see how to operate the application. This report is the complete documentation of our project. It gives complete details about the project, its functionality, users, software used, hardware requirement, and environment and so on.

* 1. **Assumptions and Dependencies**
  + Assumptions:
    - There is an active internet connection with the system.
    - The system has an internet browser installed
    - Users know the English language, as the user interface will be provided in English.
  + Dependencies:
    - There is a need for constant updating of records after each order.
    - Active participation from Chef for accepting the order.

1. **External Interface Requirements**
   1. **User Interfaces**

The main element is web Pages using HTML, CSS and Bootstrap. Multiple Interfaces are there like Registration-page for customers, Login - page for Restaurants-staff, menu-page. Customer will register and place order online then it will go to both admin and kitchen staff. After finishing food, the payment option window will get open and there will be two options: cash and online pay, then customer will pay according to their convenience and at the last; they will rate the dish in a feedback form. Chef will receive order and as per the availability of food Chef will accept or reject order. And For Admin Add new Menu, Add new table, View Menu, View Order, Add Chef these interfaces are available.

* 1. **Hardware Interfaces**

In the hardware interface, the system interacts with hardware given the processor Intel i3/i5/i7 with processor speed of 2.9 GHz and 4GB to 8 GB RAM, Hard Disk above 1TB. In future enhancements, it can be made responsive to be able to work with mobile devices as well.

* 1. **Software Interfaces**

In the software interface, Spring Framework is back-end technology used along with MySQL Database. The Front-end technologies include HTML, CSS, Bootstrap. Data will be communicated between these interfaces accordingly

* 1. **Communications Interfaces**

The main communication interface with the system will be the Web Browser.

1. **System Features**

**4.1 Description**

This system will help Restaurants Billing System and also Restaurants order management. This system will help to Restaurants owner for Account maintenance. This billing software also aims at carrying out maintain history of customers. And also improve the performance of billing process. And also maintain standard.

**4.2 Functional Requirements**

**4.2.1 Forum**

1. This system will also ask for customers feedback.

o The system will play music for customers entertainment and also you can gives feedback for Food .

**4.2.2 Agenda**

1. The customer will select the available dish and enjoy the food.
2. The system will allow customer to suggest and add up some points in the feedback.
3. The system will also display the on-going / important agenda on the page (display).

**4.2.3 Customers History**

1. The system should maintain a log of the history of Customers.
2. The system will help us to automate the process of manual Billing. Thus, 1using this system, we will be able to save a lot of time and energy.

**4.2.4 Hotel menu**

1. The system will display all dishes menu based on Restaurants Admin.

**4.2.5 Customer Feedback**

This system will maintain the Customers feedback.

**4.2.6 Payment Details**

This system will maintain the Customers Payment Details by using cash or online payment system.

1. **Performance Requirements**

The system should store all the database records of each menu, customer and admin staff properly and the application should be available for use 24\*7 through the server. Also, the application should be user friendly with a proper user interface which makes it easy for the user to understand. All the options should be present in properly accessible places for user convenience.

**5.1 Safety Requirements**

All login ids and passwords of the admin staff should be protected for privacy using whatever constraints required in the database or the application. In case any admin staff access account is hacked by any intruder, user id and passwords of all the

Admin staff personnel should be changed and new passwords should be issued. Customer, Food Menu and admin records are to be backed up securely across database servers. Incase database is hacked by someone and data is deleted a backup server should be present for such purpose.

**5.2 Security Requirements**

All passwords of the administrators should be protected for privacy using whatever constraints required in the database or the application. Transactions regarding customer and menu records should be carried out properly. Only admin staff will have access rights to the menu as well as customer data, according to the need for E.g.: -To set today special menu and feedback for food and service, customer information etc. The database should be protected from attacks and unauthorized access. The interface should be protected from attacks. All passwords should be stored as a secure hash of the administrator password.

5.3 **Software Quality Attributes**

**5.3.1 Availability**

The system should run on a variety of operating systems that support the JavaScript language. The system should run on a variety of hardware.

**5.3.2 Accessibility**

The software will be accessible to admin.

**5.3.3 Compatibility**

The software will be compatible with multiple platforms.

**5.3.4 Durability**

The software will be tested for working with multiple users.

**5.3.5 Effectiveness**

The software will be made to handle operations effectively.

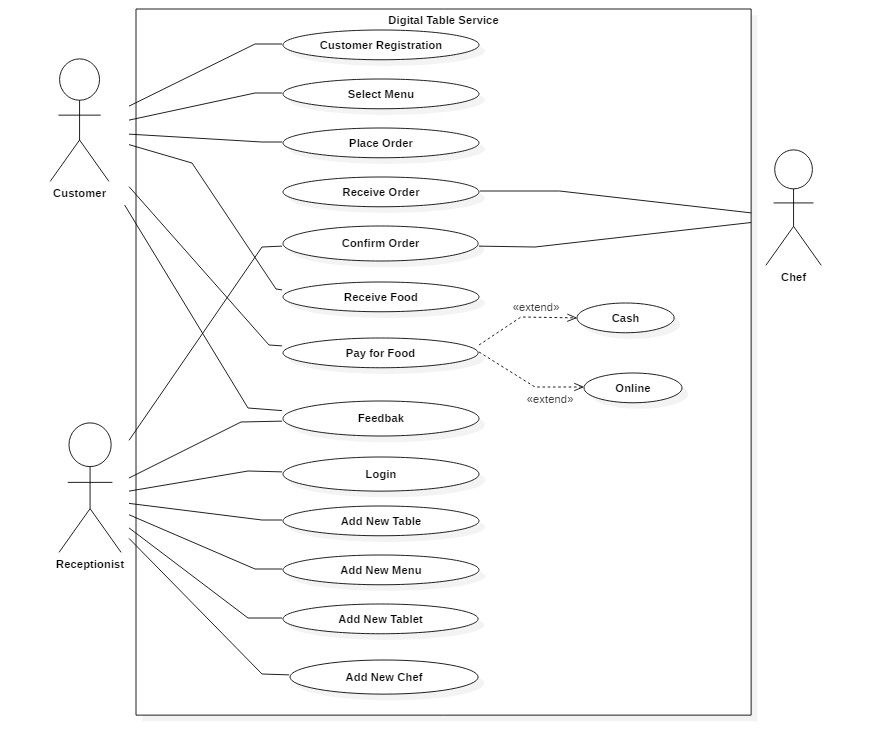
**5.3.6 Maintainability**

The system should be easy to maintain. There should be a clear separation between the interface and the business logic code. There should be a clear separation between the data access objects that map the database and the business logic code.

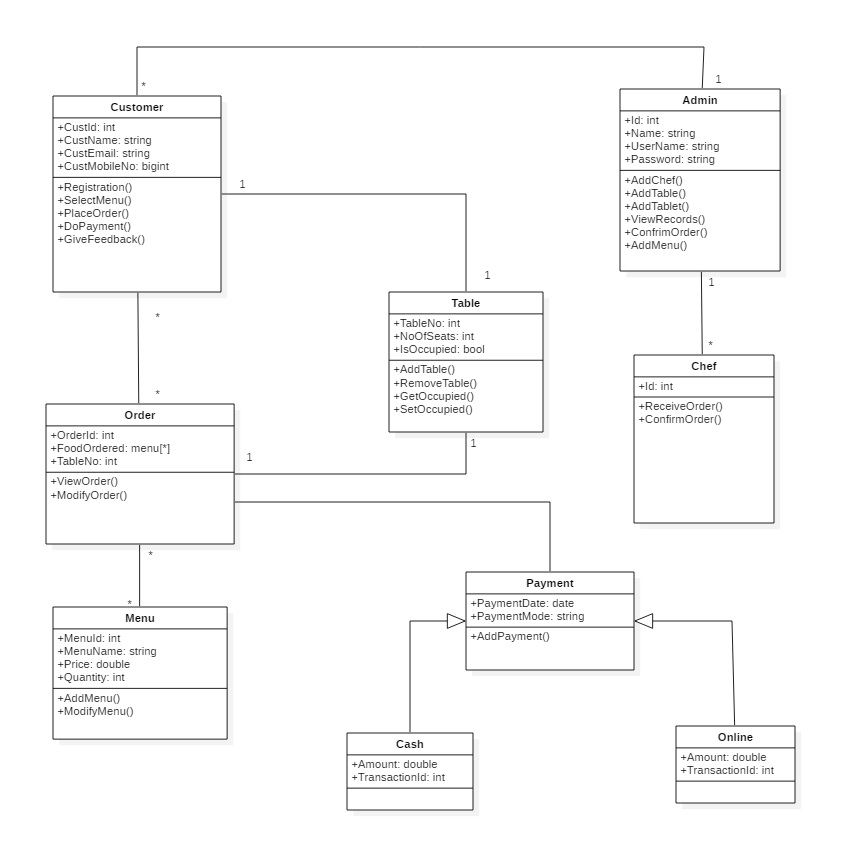
**6.Other Requirement**

**Appendix A: Analysis Models**

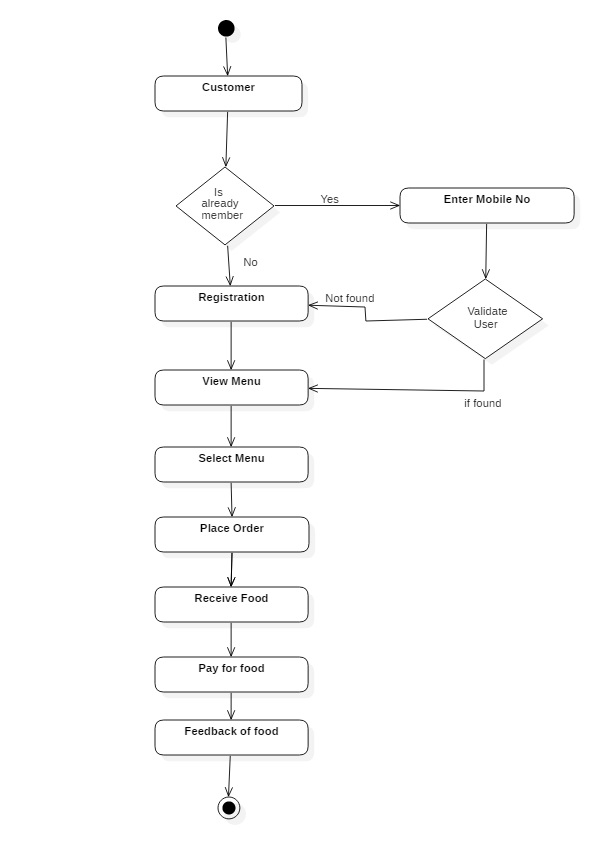
**a). Use Case Diagram**

****

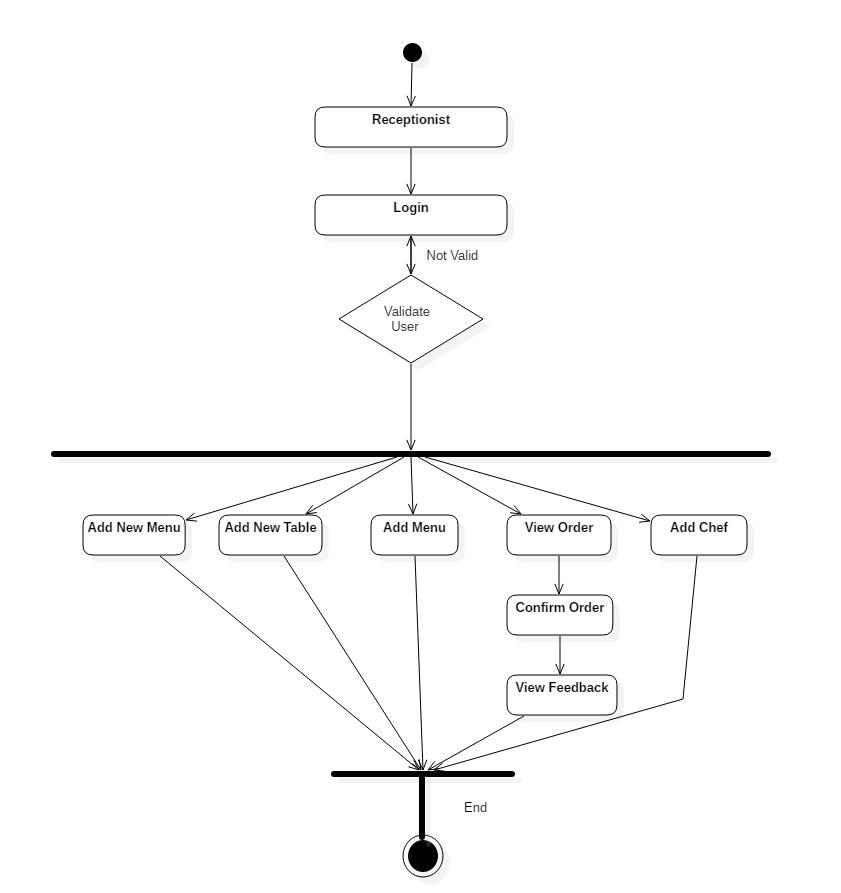
**b) Class Diagram**

****

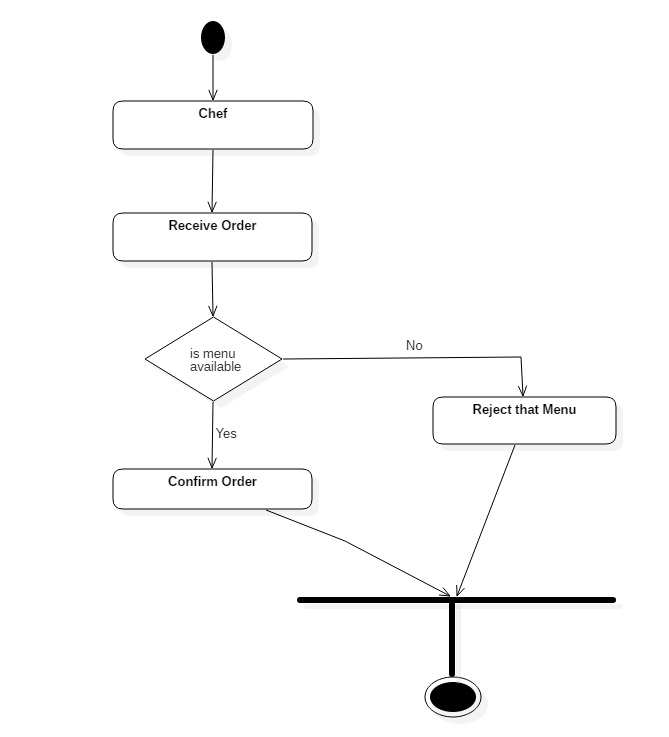
**c) Activity Diagram 1.1**

****

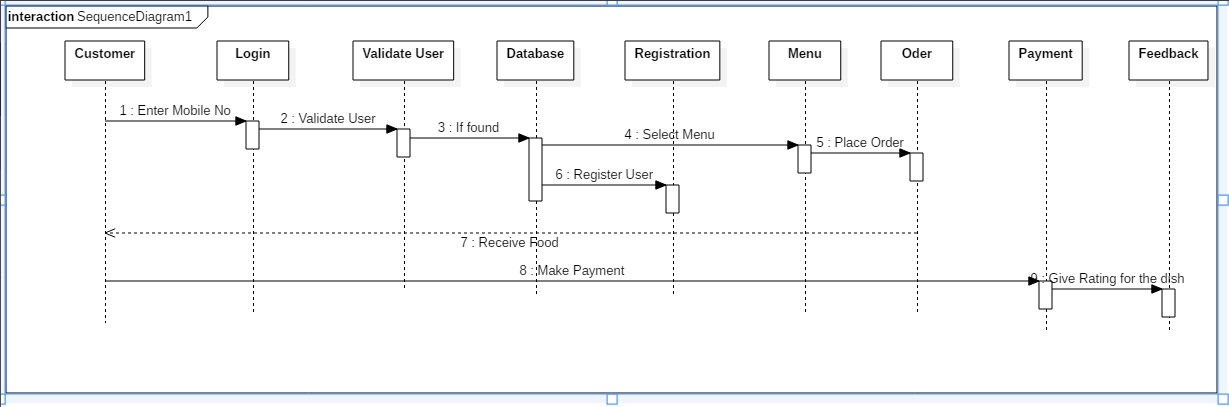
**d) Activity Diagram 1.2**

****

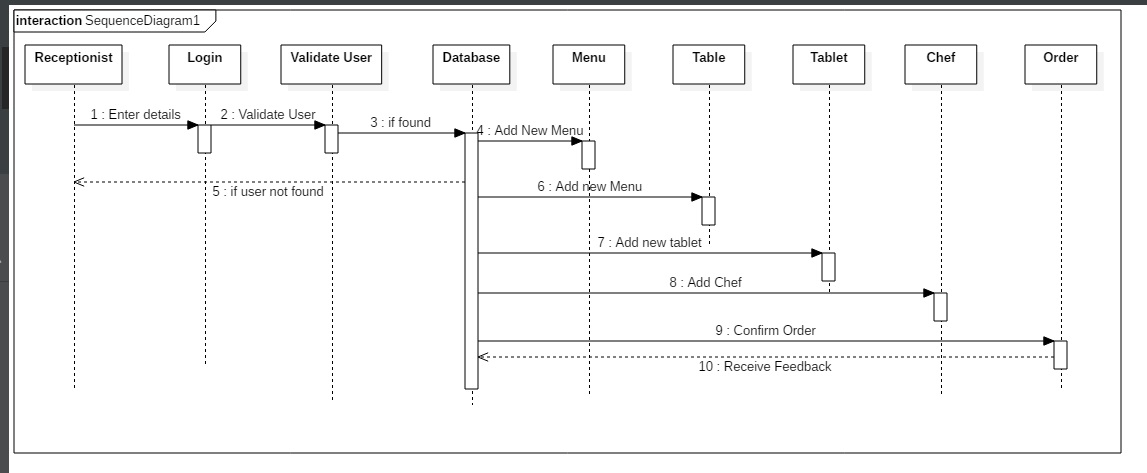
**e) Activity Diagram 1.3**

****

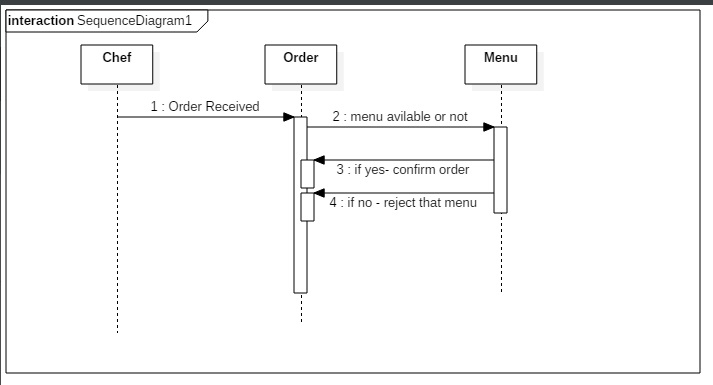
**f) Sequence Diagram 1.1**

****

**g) Sequence Diagram 1.2**

****

**g) Sequence Diagram 1.3**

****