# Table of Contents

1. Abstract
2. Introduction
3. Software Requirements Specification
4. Design
5. Implementation
6. Testing

1.ABSTRACT

SmartMenu Restaurant Billing Software has advanced features to manage all transactions of Restaurants and Hotels like restaurant billing, Corporate Catering on customized concept of KOT (Kitchen Order Token) system etc. The Software also allows making the restaurant billing without KOT system.

Restaurant billing software provides fully fledged summery report about total items and daily sales summary without any anxiety. Stock can be also managed regularly. It is a Web based application and it will work in intranet as well as internet.

Managing a restaurant business is less complex when you have the fundamental restaurant billing software. Billing is an essential operate for small/large restaurants or cafe trades, however program programs can be luxurious.

With SmartMenu you can manage your accounting at anytime from anywhere. It provides fully fledged Billing Software in Admin section of SmartMenu from where Admin can check the Sales and can generate the reports. Admin section is fully responsive provides smart layout for Managers so that they can manage the restaurant billing from any device.

SmartMenu Restaurant Billing Software provides the direct access to your Account managers or chartered accountant so that they can work on their books in real time and keep the business updated.

SmartMenu Billing Software is “all-in-one” software that provides all information on fingertips of owners and their managers.

The coolest feature of accessing the Restaurant Billing Software from any place at any time is a huge plus for business managers.

It not only saves the time, but also reduces resource allocation as a single person can manage all financial activities.

2. INTRODUCTION

2.1 Brief Introduction:

We have created Restaurant billing software for pizza store. It is applicable to all franchise. It is used for take the customer’s order and generate bill corresponding to customer’s order. It is also used for manage the all the chefs. It is also used for add item in to the list, update item etc.

2.2 Tools/Technologies Used:

Technologies:

* ASP.net WindowsForm based Application

Tools:

* Microsoft Visual Studio 2019
* SQL server

3. SOFTWARE REQUIREMENTS

SPECIFICATION

Types of Users:

* Manager
* Chef

*R.1 Manager*

R.1.1 Add item :

Input : Item name.

Output : Success message.

R.1.2 Delete item :

Input : Item name.

Output : Success message.

R.1.3 Update item :

Input : Item Details.

Output : Updated details.

R.1.4 Display all item :

Input : User selection

Output : All list of items.

R.1.5 Manage Chef :

Description : Manager can add chef, remove chef , Edit chef details as per requirement.

R.1.5.1 Add chef :

Input : Chef details.

Output : Success message.

R.1.5.2 Remove chef :

Input : Chef name.

Output : Success message.

R.1.5.3 Edit chef :

Input : Chef details.

Output : Updated details.

*R.2 Chef*

R.2.1 Select item :

Input : item as selected by customer.

Output : Add to customer’s bill.

R.2.2 Adjust Quantity :

R.2.2.1 Add Quantity :

Input : User selection.

Output : Add to customer’s bill.

R.2.2.2 Remove Quantity :

Input : User selection.

Output : Remove from customer’s bill.

R.2.3 Customize Item :

Description : Chef can customize item as customer asked for.

Input : Customize option.

Output : Add amount to customer’s bill.

R.2.4 Create order :

Input : Order initiation.

Output: Success Message.

R.2.5 Update order :

Description: Chef selects state of the order so Customers are comes to know , where my order has been reached .

e.g. confirmed, preparing, ready .

Input : Select Order State.

Output: Update Order State Message .

R.2.6 Generate customer’s bill :

Description : After taking order of customer chef generates a bill. First chef ask for mobile number of customer and if it is matched with existing customer then chef not need to fill other details like customer’s name, birthdate otherwise chef needs to add customer’s details manually. Then chef accepts payment in terms of credit card, debit card, cash or other methods. After paying cash system generate bill receipt.

R.2.6.1 Fetch customer details :

Input : Customer’s mobile number.

Output : If he/she is existing customer then all details fill automatically otherwise chef needs to add details.

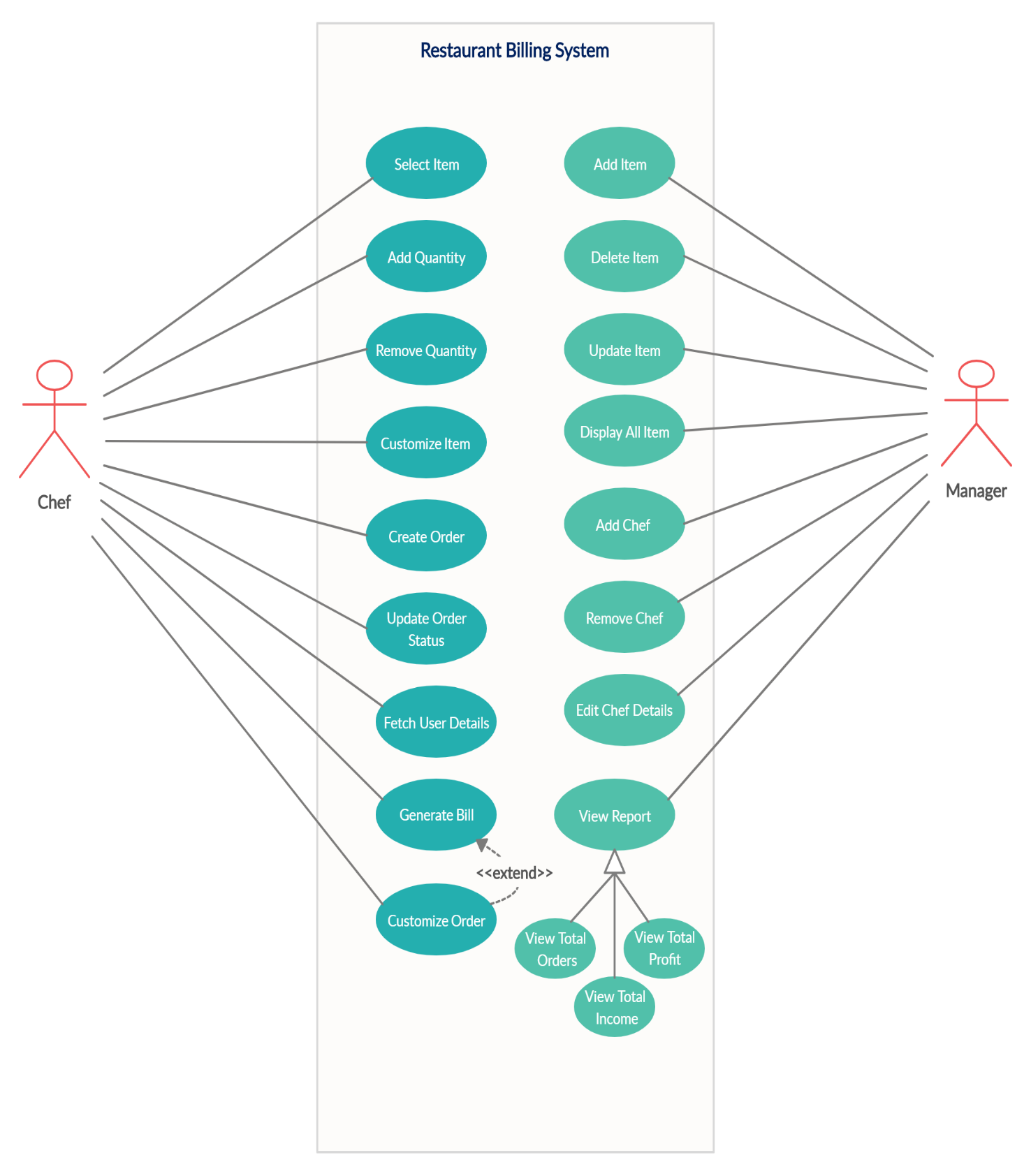
R.2.6.2 Generate bill receipt :

Input : User selection.

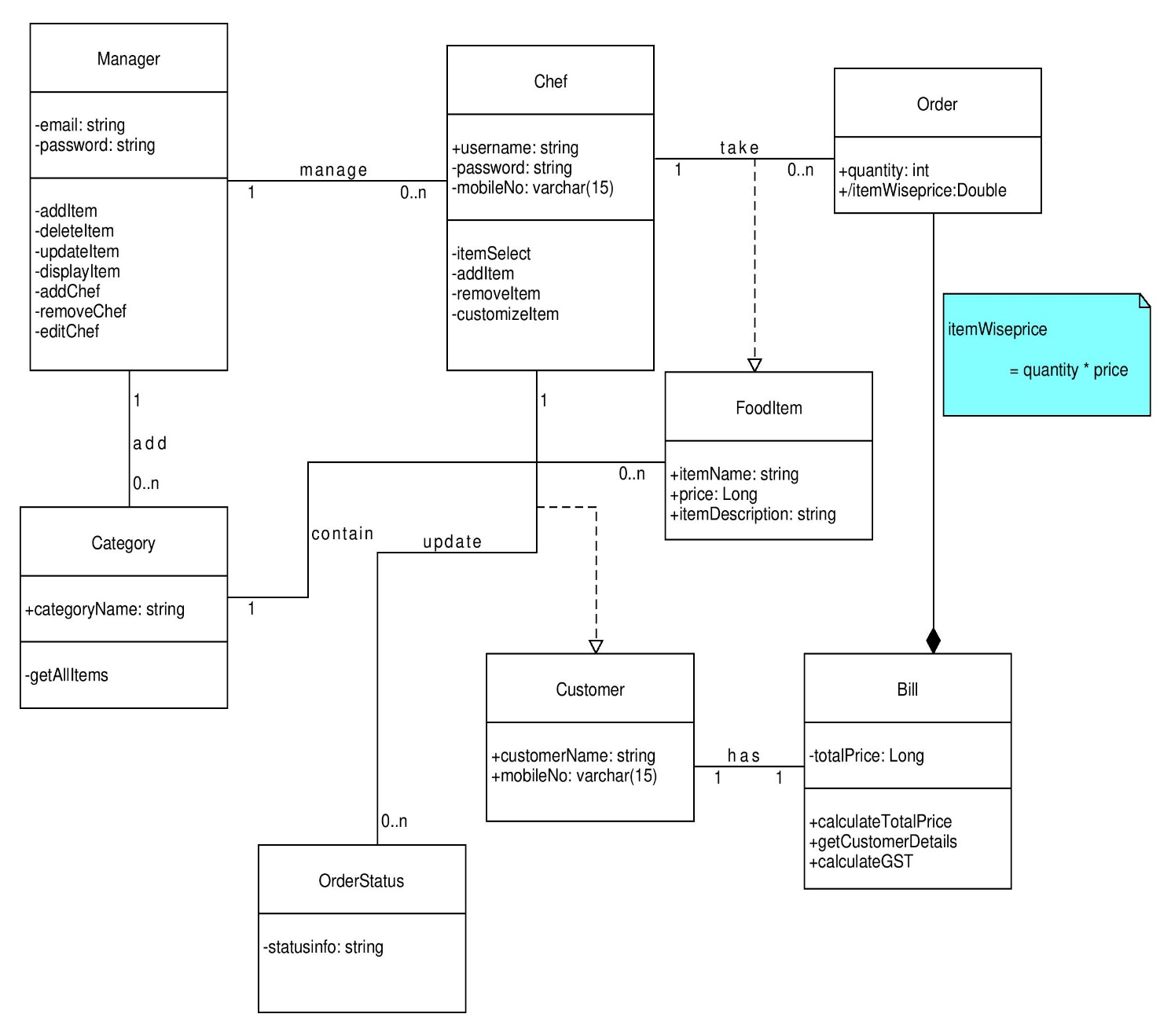
Output: Customer’s bill receipt.

4. DESIGN

**Use case Diagram :**



**Class diagram :**

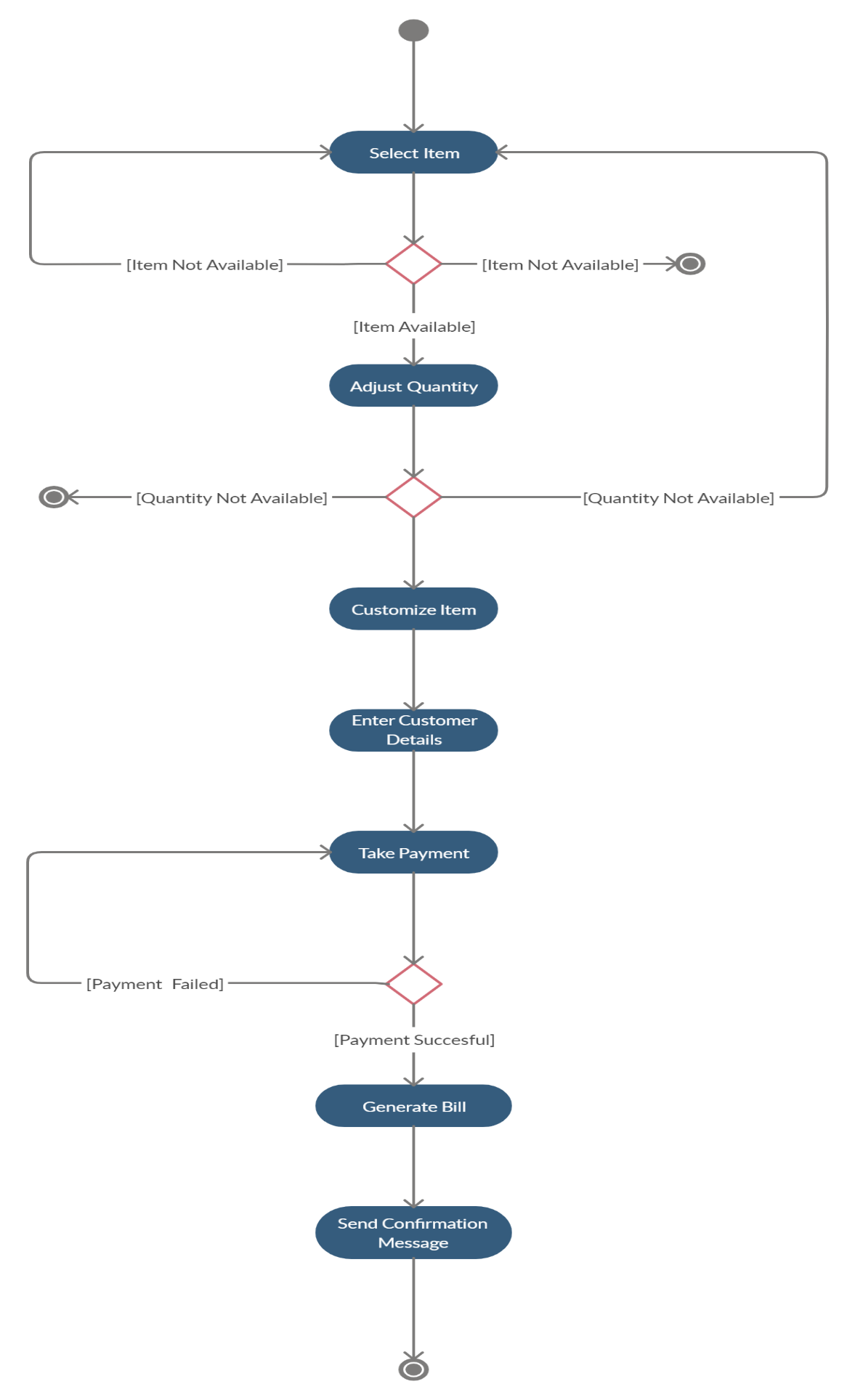


**Sequence diagrams:**

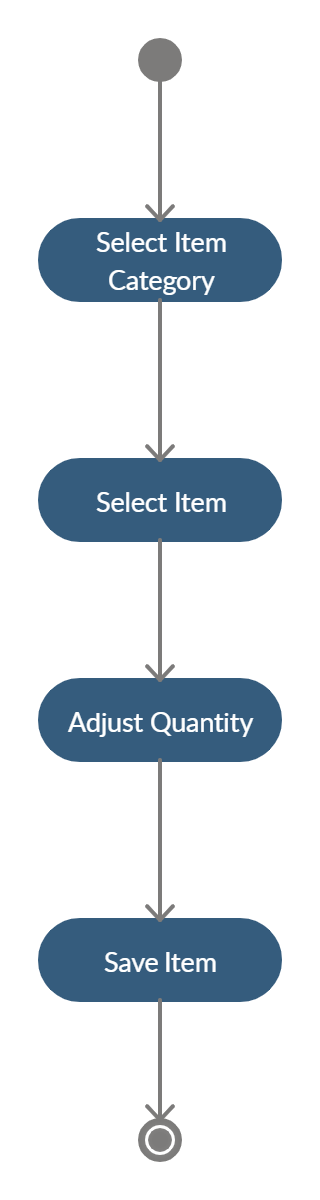


**Activity diagrams:**

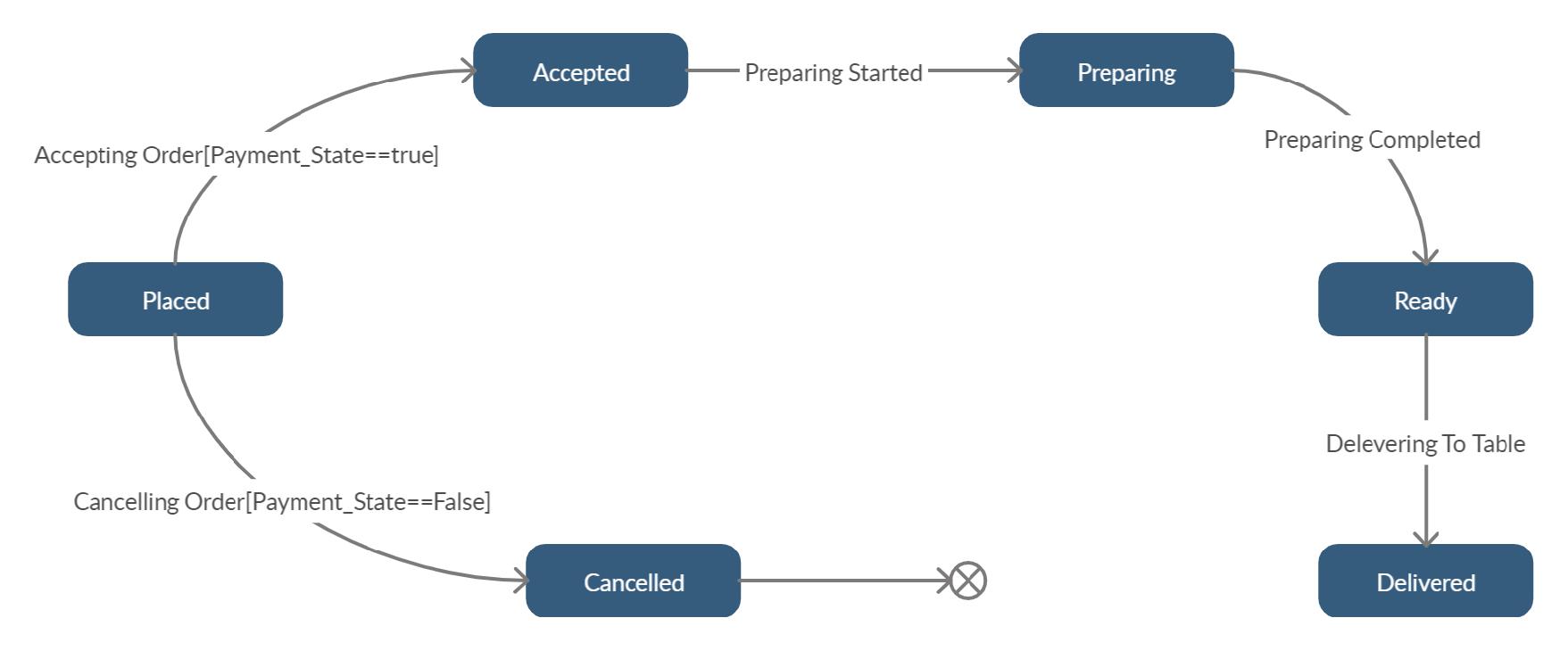
Take Order:



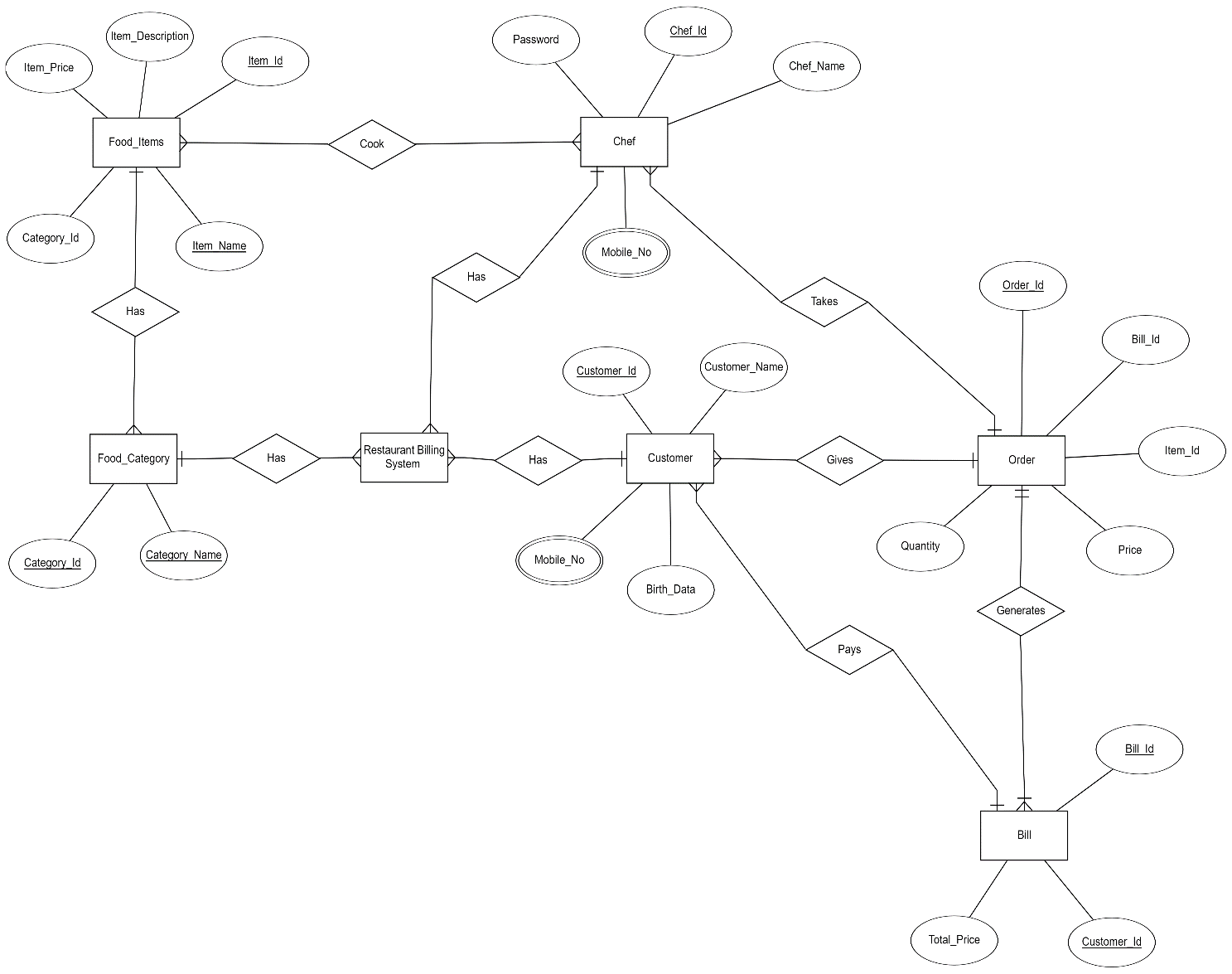
Edit Items:



**State diagrams:**



**E-R Diagram:**



**Data dictionary :**

# Data Dictionary for Restaurant Billing System

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Food Category | | | | | |
| Field | Type | Required | Unique | PK/FK | Reference Table |
| Category\_Id | Int | Yes | Yes | PK | - |
| Category\_name | Varchar(50) | Yes | Yes | - | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Items | | | | | |
| Field | Type | Required | Unique | PK/FK | Reference Table |
| Item\_Id | Int | Yes | Yes | PK | - |
| Category\_Id | Int | Yes | No | FK | Food\_Category |
| Item\_Name | Varchar (50) | Yes | Yes | - | - |
| Item\_Price | Varchar (50) | Yes | No | - | - |
| Item\_Description | Varchar(50) | Yes | - | - | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Customer | | | | | |
| Field | Type | Required | Unique | PK/FK | Reference Table |
| Customer\_Id | Int | Yes | Yes | PK | - |
| Customer \_Name | Varchar(50) | Yes | No | FK | - |
| Mobile\_No | Varchar(50) | Yes | No | - | - |
| Birth\_Date | Date | No | No | - | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bill | | | | | |
| Field | Type | Required | Unique | PK/FK | Reference Table |
| Bill\_Id | Int | Yes | Yes | PK | - |
| Customer\_Id | Int | Yes | Yes | FK | Customer |
| Total\_Price | Varchar(50) | Yes | No | - | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chef | | | | | |
| Field | Type | Required | Unique | PK/FK | Reference Table |
| Chef\_Id | Int | Yes | Yes | PK | - |
| Chef \_Name | Varchar (50) | Yes | No | FK | - |
| Mobile\_No | Varchar (50) | Yes | No | - | - |
| Password | Varchar (50) | Yes | No | - | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Order | | | | | |
| Field | Type | Required | Unique | PK/FK | Reference Table |
| Order\_Id | Int | Yes | Yes | PK | - |
| Bill\_Id | Int | Yes | No | FK | Bill |
| Item\_Id | Int | Yes | No | FK | Items |
| Price | Varchar (50) | Yes | No | - | - |
| Quantity | Int | Yes | No | - | - |

5. IMPLEMENTATION DETAIL

I). Modules

* Chef Login Module :

This module takes chef credentials and then verifies it with registered users, if user have entered incorrect credentials then alert with “Incorrect details” , otherwise chef will be redirected to the home page.

* Add Item Module:

This module use for add new food item into the list like item name, image, description, price, category by any chef.

* Display all item Module :

This module is used to display all item by categorywise, so chef can select item by according to customer’s demand into the list for take orders.

* Bill Generate Module :

This module allows chef to generate bill and print receipt. It contains shop title, shop address, cutomer name, customer’s mobile no, all the item list which has been ordered by individual customer.

* Customer Details Module :

This module is built for take customer details like customer name , customer mobile no for record. So, if shop wants to contact customer it can .

II). Function prototypes

* Chef Login :

private void button1\_Click(object sender, EventArgs e)

* Add Item :

internal static string AddNewItem(FoodItem fi)

* Display All item:

private void DisplayItems\_Load(object sender, EventArgs e)

* Add item in list :

protected void add\_item(object sender, EventArgs e)

6. TESTING

In this system we have used Black Box testing.

The main focus of black box testing is on the validation of your functional requirements.

Here are the generic steps followed to carry out any type of Black Box Testing.

* Initially, the requirements and specifications of the system are examined.
* Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
* Tester determines expected outputs for all those inputs.
* Software tester constructs test cases with the selected inputs.
* The test cases are executed.
* Software tester compares the actual outputs with the expected outputs.
* Defects if any are fixed and re-tested.