# MO FOODS

**SOFTWARE REQUIREMENT SPECIFICATIONS**

**PROPOSED BY:**

**OSAMA MALIK 4319-FBAS/BSSE/F21-A**

**MANSOOR AHMAD 42XX-FBAS/BSSE/F21-A**

**JAN BAHADUR 4298-FBAS/BSSE/F21-A**

**ADNAN HUSSAIN 4284-FBAS/BSSE/F21-A**

## Contents

**ABSTRACT** ……3

Existing System: ……3

Proposed System: ……3

**SOCIO TECHNICAL SYSTEM** ……3

**MODEL** ……4

RUP Model ( Rational Unified Process Model ) ……4

Advantages of RUP Software Development ……4

Other Models ……5

WaterFall Model ……5

Incremental Model ……5

RAD Model ……5

Prototype Model ……5

Spiral Model ……5

**FUNCTIONAL AND NON- FUNCTIONAL REQUIREMENTS** ……5

Functional Requirements ……5

Non-Functional Requirements ……6

**UML DIAGRAMS** ……7

Sequence Diagram ……7

Activity Diagram ……7

Class Diagram ……8

User-Case Diagram ……9

**TESTING** ……9

## ABSTRACT

Online Food Delivery System is a system which will help restaurant to optimized and control over their restaurants. For the waiters, it is making life easier because they don't have to go kitchen and give the orders to chef easily. For the management point of view, the manager will able to control the restaurant by having all the reports to hand and able to see the records of each employees and orders. This application helps the restaurants to do all functionalities more accurately and enhances the spend of all the tasks. Online Food Delivery System reduces manual works and improves efficiency of restaurant.

The online food delivery system set up menu online and the customers easily places the order with a simple mouse click. Also with a food menu online you can track orders, maintain customer's database and improve your food delivery service. This system allows the user to select the desired food items from the displayed menu. The user orders the food items. The payment can be made online or pay-on-delivery system. The user's details are maintained confidential because it maintains a separate account for each user. And id and password is provided for each user. Therefore, it provides a more secured ordering.

### Existing System:

In existing system for giving any order user should visit hotels to know about food items and then give order and pay advance. In this method time and manual work is required. Maintaining critical information in the files and manual is full of risks and a tedious process.

### Proposed System:

This online application enables the end user to register online, select the food from the e-menu card and order the food online. By just selecting the food that the user want to have. The result after selecting the food the e-menu card will directly appear in the screen near the Chef who is going to cook food for you. By using this system the work of the waiter is reduced and we can also say that work is nullified. The benefit of this if there us rush in the restaurant then their will be chances that waiter will have be unavailable and the user can directly order the food to the chief online by using the system. The user will be given a username and a password to login.

## SOCIO TECHNICAL SYSTEM

This system is a socio-technical system as the system has such properties:

- Involves operational processes.
- Include people who use and interact with the technical system.

So, we can conclude that the system is Social technical system because it involves operational processes (like place order, online payment) and include people.

# MODEL

## RUP Model (Rational Unified Process Model)

For the implementation of the system the RUP Model is used. Its goal is to ensure the production of the high-quality software that meets the need of its end user within periodic schedule, It is easily customized and useful for web based implementations.

RUP methodology has a highly flexible development path. It uses the some of the industry's best practices. These are known as 6 best practices of RUP methodology. The reasons why we are using this model for developing this system are also these best practices.

**Develop Iteratively:** The software requirements specifications keep on evolving throughout the development proves and loops are crated to add them without affecting the cost of development.

**Manage Requirements:** The business requirements documentation and project management requirement need to be gathered properly from the user in order to reach the targeted goal.

**Use Components:** The components of the large product which are already tested and are in the use can e conveniently used in other projects. This reuse of components reduces the production time.

**Model Visually:** Use of Unified modelling language (UML) facilitates the analysis and design of various components. Diagrams and models are used to represent various components and their interactions.

**Verify Quality:** Testing and implementations effective project quality management should be a major part of each and every phase of the project from initiation to delivery.

**Control Changes:** Synchronization of various parts of the system becomes more challenging when the parts are being developed by various teams working from different geographic locations on different platforms. Hence special care should be taken in this direction so that the changes can be controlled.

## Advantages of RUP Software Development

1. This is a complete methodology in itself with an emphasis on accurate documentation
2. It is proactively able to resolve the project risks associated with the client's evolving requirements requiring careful change request management
3. Less time is required for integration as the process of integration goes on throughout the software development life cycle.
4. The development time required is less due to reuse of components.
5. There is online training and tutorial available for this process.

## Other Models

### Waterfall model

The water fall model can be used for the development of the system but we are not using it because the requirement must be known at the start otherwise it would be a disaster as the requirements are frozen before the designing of the system. In our system we need the testing at every phase but in waterfall model there is only Big Bang Testing which is done at the end of the completion of all the phases.

### RAD model

The reason we are not using this model is that the requirement can not be changed at later stages as, if the requirements are not well understood at the start, then it would be very difficult. This process model, is no doubt for small scale projects like ours work force is needed as number of teams are working parallel and that can be risky and costly.

### Prototype model

This model is not used by us because risk factor very high in this model, If our system is not accepted by the user then it will remain in an unending loop which will cause al lot of financial drop.

### Spiral model

The spiral model is used by us in development our system as it is not suitable for low-risk systems. In this model, it is hard to define objective, verifiable milestones and doesn't work well for our type of smaller projects. Similarly, it can be a costly model to use and risks analysis requires highly specific expertise.

## Functional & Non-Functional Requirements:

### Functional Requirements:

1. **Registration:**
   If customer wants to order the food, then, he/she must be registered, unregistered users cannot go for ordering.
2. **Login:**
   The customer login too the system by entering valid user id and password for ordering.
3. **Display the menu:**
   In this system all the items are displayed with their rates.
4. **Modify menu:**
   System can make changings in menu like adding or removing food items which are not available.
5. **Select food items:**
   Items are selected customer feel free to order.

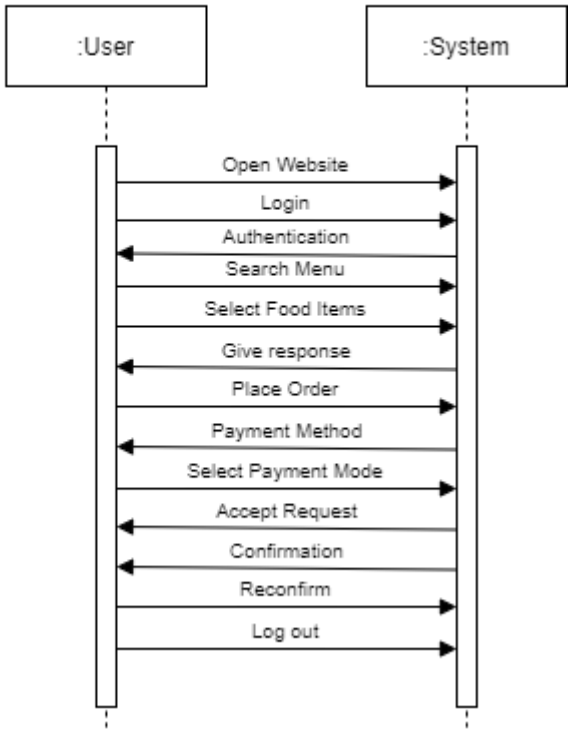
6. **Changes to order:**
   Changes to order mean the customer can make the changings in order like he /she can delete or add food item in order.
7. **Review the order before submitting:**
   Before submitting the complete order is reviewed to the customer. Customer name, phone number location (address) and place order, then finally order is submitted.
8. **Payment:**
   For customer there are many types of secure billing will be prepared as debit or credit cards.
9. **Provide delivery & payment details:**
   Here bill is generated, order no and payment is given and confirmation of delivery is done.
10. **Logout:**
    After the payment or surf the product, the customer will log out.

**Non-Functional Requirements:**

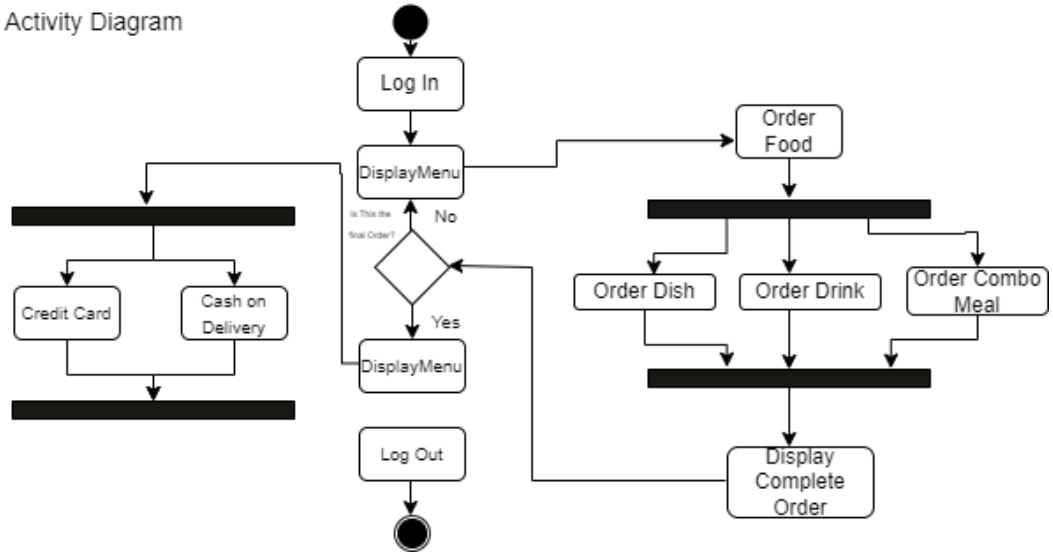
1. **Portability:**
   System running on one platform can easily be converted to run another platform
2. **Reliability:**
   The ability of the system to behave consistently in a user-acceptable manner when operating within the environment for which the system was intended.
3. **Availability:**
   The system should be available at all times, meaning the user can access it using a web browser.
4. **Maintainability:**
   A commercial database is used for maintaining the database and the application server takes care of the site.
5. **Security:**
   Secure access of confidential data.
6. **User friendly:**
   System should be easily used by the customer.
7. **Performance**:
   It should be fast.
8. **Safety:**
   Data in the database of the system should not loss or damage.
9. **Privacy:**
   Personal data of the system should not disclose to anyone

# UML DIAGRAMS

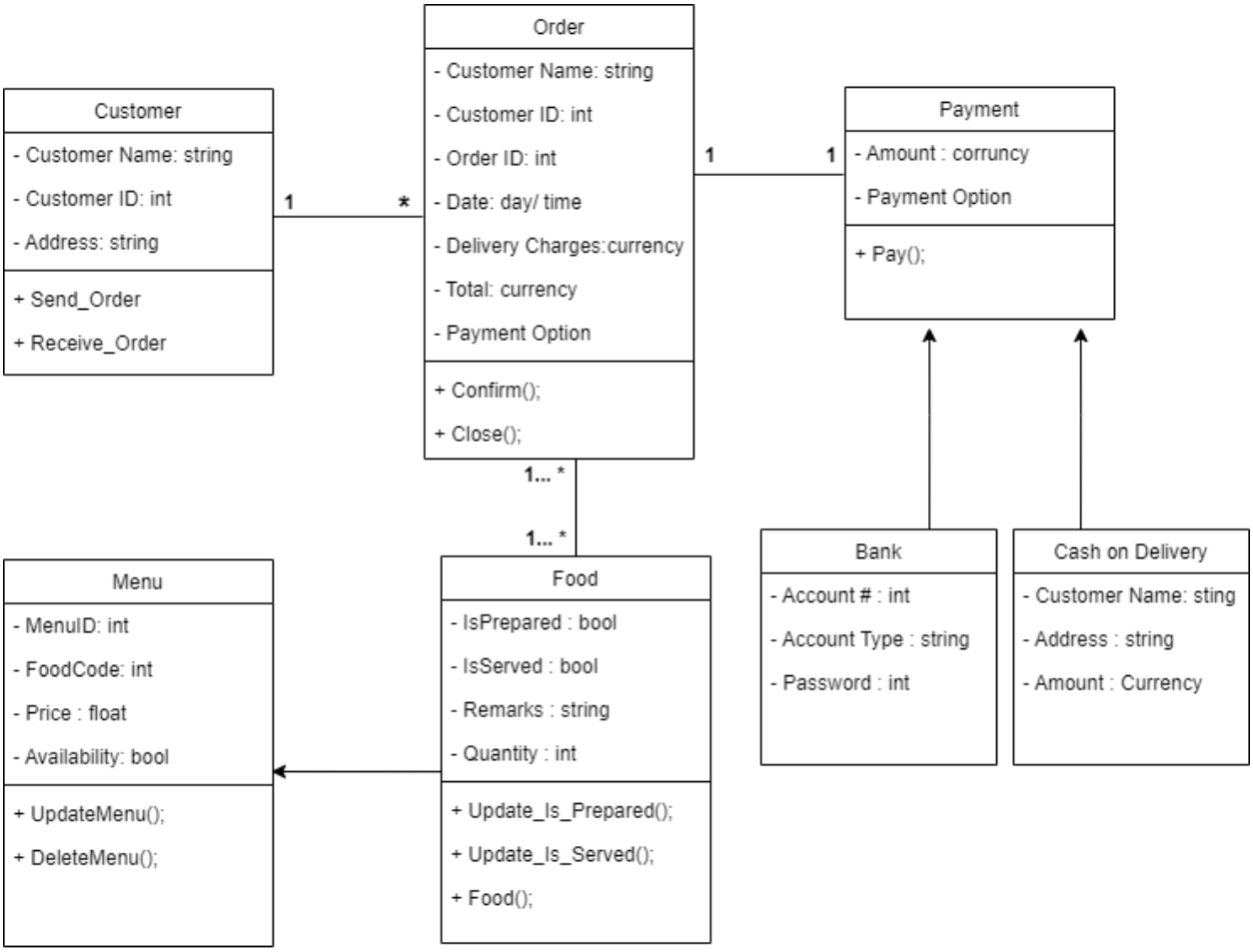
Sequence Diagram



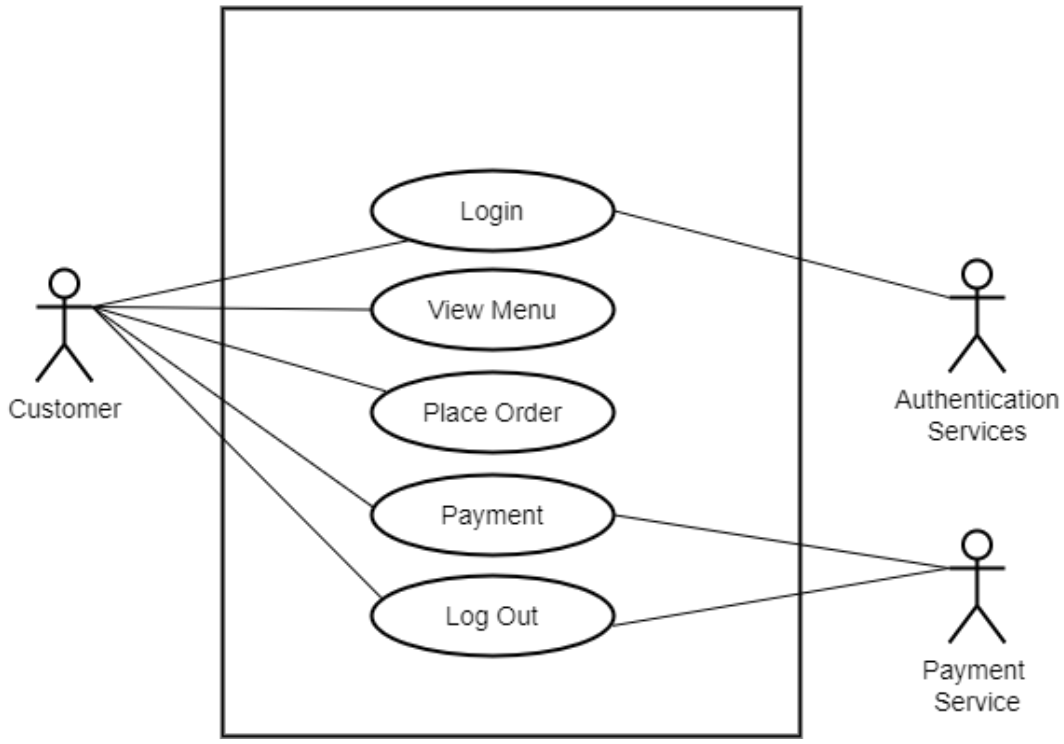

Activity Diagram



# Class Diagram



Use Case Diagram




**Testing:**

At the end software testing is done to point out the defects and errors that were made during the development phases. In our project we have used 5 types of testing these are listed below.

**Unit Testing:** Unit testing where individuals program units or object classes are tested. Here by using this testing we have focused on testing the functionality of methods.

**Module Testing:** where this is the combination of unit program is called module. Here we test the unit program (5-6 programs) is where the module programs have dependency.

**Sub-system Testing:** Then we combined some module for the Preliminary System Testing in our Project.

**System Testing:** It is the combination of two or more sub-system and then it is tested. Here we tested the entire system as per the requirements.

**Acceptance Testing:** Normally this type of testing is done to verify if system meets the customer specified requirements. After submitting the project to user, they test it and determine whether to accept the system or not. It is the system testing performed by the customer(s) to determine whether they should accept the delivery of the system.