Team Name

Name (s)  
Lab section:  
Workstation:  
Date.

sOFTWARE DESIGN DOCUMENT

Team Name

# TABLE OF CONTENTS

Contents

[**TABLE OF CONTENTS 1**](#_Toc172526729)

[**1. INTRODUCTION 1**](#_Toc172526730)

[**3. SYSTEM ARCHITECTURE 1**](#_Toc172526731)

[**4. DATA DESIGN 1**](#_Toc172526732)

[**6. HUMAN INTERFACE DESIGN 1**](#_Toc172526733)

[**1. INTRODUCTION 2**](#_Toc172526734)

[**1.1 Purpose 2**](#_Toc172526735)

[**1.2 Scope 2**](#_Toc172526736)

[**1.5 Definitions and Acronyms 2**](#_Toc172526737)

[**2. SYSTEM OVERVIEW 2**](#_Toc172526738)

[**3. SYSTEM ARCHITECTURE 3**](#_Toc172526739)

[**3.1 Architectural Design 3**](#_Toc172526740)

[**3.2 Decomposition Description 4**](#_Toc172526741)

[**3.3 Design Rationale 5**](#_Toc172526742)

[**4. DATA DESIGN 5**](#_Toc172526743)

[**4.1 Data Description 5**](#_Toc172526744)

[**4.2 Data Dictionary 6**](#_Toc172526745)

[**5. COMPONENT DESIGN 6**](#_Toc172526746)

[**5.1 Food Delivery Application Process: 6**](#_Toc172526747)

[**5.2 Placing an Order: 6**](#_Toc172526748)

[**5.3 Deliver the order: 6**](#_Toc172526749)

[**6. HUMAN INTERFACE DESIGN 6**](#_Toc172526750)

[**6.1 Overview of User Interface 6**](#_Toc172526751)

[**6.2 Screen Images 7**](#_Toc172526752)

[**7. REQUIREMENTS MATRIX 7**](#_Toc172526753)

[**8. APPENDICES 7**](#_Toc172526754)

[**Appendix A: Glossary 7**](#_Toc172526755)

[**Appendix B: Analysis Models 8**](#_Toc172526756)

[**Appendix C: To Be Determined List 8**](#_Toc172526757)

# 1. INTRODUCTION

## 1.1 Purpose

This Software Design Document (SDD) describes the architecture and system design of a food delivery application. The purpose of this SDD is to provide a detailed description of the system's architecture, design, and implementation plan to ensure a coherent and maintainable product. This document is intended for developers, project managers, testers, and stakeholders involved in the project.

## 1.2 Scope

The food delivery application aims to streamline the process of ordering and delivering food to customers within a city. It facilitates user registration, browsing restaurants and menus, placing and tracking orders, payment processing, and providing feedback. The primary goals are to enhance customer satisfaction and operational efficiency.

**1.3 Overview**

The document is organized as:

1. Introduction
2. System Overview
3. System Architecture
4. Data Design
5. Component Design
6. Human Interface Design
7. Requirements Matrix
8. Appendices

**1.4 Reference Material**

1. Reference 1
2. Reference 2

## 1.5 Definitions and Acronyms

1. SDD: Software Design Document
2. UML: Unified Modeling Language
3. ERD: Entity Relationship Diagram
4. DFD: Data Flow Diagram
5. PDL: Procedural Description Language

# 2. SYSTEM OVERVIEW

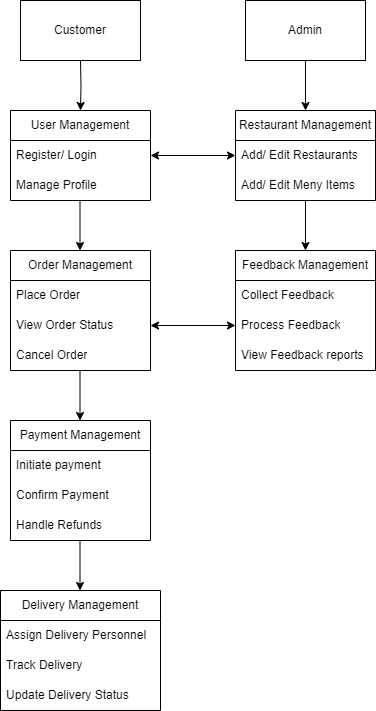
The Food Delivery App allows users to browse restaurant menus, place orders, make payments, and track deliveries in real-time. The system supports multiple user roles including customers, delivery drivers, and restaurant owners, each with specific functionalities. The app ensures secure data handling, efficient communication, and real-time updates to enhance user experience.

# 3. SYSTEM ARCHITECTURE

## 3.1 Architectural Design

The architecture of the food delivery application is based on a modular design, dividing responsibilities among various subsystems:

* **User Management Subsystem**: Handles user registration, authentication, and profile management.
* **Restaurant Management Subsystem**: Manages restaurant listings, menus, and order processing.
* **Order Management Subsystem**: Facilitates order placement, tracking, and status updates.
* **Payment Processing Subsystem**: Integrates with payment gateways for secure transactions.
* **Delivery Management Subsystem**: Assigns delivery personnel and tracks delivery status.
* **Feedback Subsystem**: Collects and processes user feedback and ratings



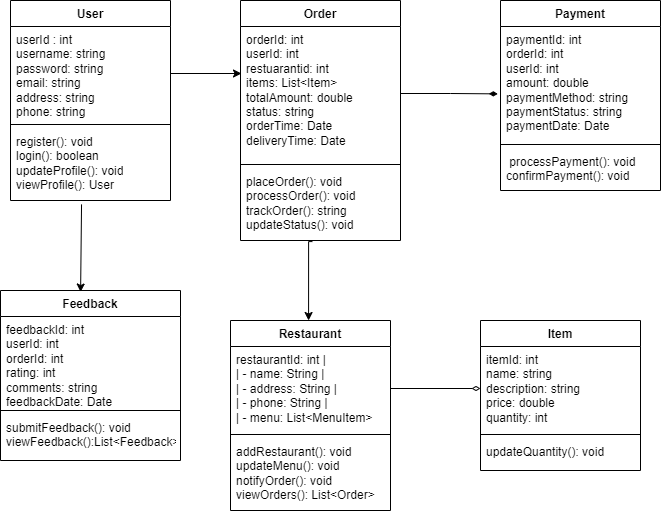
## 3.2 Decomposition Description

Provide a decomposition of the subsystems in the architectural design. Supplement with textas needed. You may choose to give a functional description or an object orienteddescription.

For a functional description, put top level data flow diagram (DFD) and structural decomposition diagrams.

For an OO description, put subsystem model, object diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here. The system is decomposed into the following modules:

* **Functional Description:**
  + **User Module**: Handles user interactions, including registration, login, and profile management.
  + **Order Module**: Manages order placement, processing, and tracking.
  + **Payment Module**: Manages payment processing and integration with payment gateways.
  + **Feedback Module**: Handles feedback and ratings from users.
  + **Restaurant Module**: Manages restaurant listings, menu updates, and order notifications to restaurants.



## 3.3 Design Rationale

The architecture was chosen to ensure scalability, maintainability, and ease of integration with external services. The modular design allows for independent development and testing of each subsystem. Alternative architectures were considered but were not selected due to higher complexity and lower flexibility.

# 4. DATA DESIGN

## 4.1 Data Description

The data structures for the system include user profiles, restaurant information, menu items, orders, payments, and feedback. These entities are stored in a relational database to ensure data integrity and efficient access.

## 4.2 Data Dictionary

1. **User**: {UserID, Name, Email, Password, PhoneNumber, Address}
2. **Restaurant**: {RestaurantID, Name, Address, MenuItems}
3. **MenuItem**: {MenuItemID, RestaurantID, Name, Description, Price}
4. **Order**: {OrderID, UserID, RestaurantID, MenuItemIDs, TotalPrice, Status, DeliveryTime}
5. **Payment**: {PaymentID, OrderID, Amount, PaymentMethod, TransactionID}
6. **Feedback**: {FeedbackID, OrderID, Rating, Comments}.

# 5. COMPONENT DESIGN

The step-by-step outline details the procedural flow of following actions:

## 5.1 Food Delivery Application Process:

1. Customer opens the application and logs in or sign ups (if not already) by entering his name, phone number and address.
2. Admin manages the application
3. Customer then selects item/s and places the order.
4. Application calculates the total price to be paid by the customer.
5. Customer then pays the bills.
6. Application chooses the delivery personnel.
7. Customer can also track the delivery ride.

## 5.2 Placing an Order:

1. Customer manages the order by first selecting item/s or deals to order by their choice by adding them to cart.
2. Then the customer chooses the amount of money needed.
3. Customer confirms the choice.
4. System validates the amount.
5. System asks central bank computer to debit the customer’s account.
6. System issues money to the user.

## 5.3 Deliver the order:

1. After customer has places the order, the application alerts the chosen restaurant about the order to be prepared.
2. Restaurant prepares the order and sends it to the delivery personnel who was assigned by application service and was most suitable.
3. Customer is also notified about the order and delivery details.

# 6. HUMAN INTERFACE DESIGN

## 6.1 Overview of User Interface

The user interface is designed for ease of use, with intuitive navigation and clear feedback. Users can browse restaurants, place orders, track delivery, and provide feedback.

## 6.2 Screen Images

Display screenshots showing the interface from the user’s perspective. These can be hand drawn or you can use an automated drawing tool. Just make them as accurate as possible.

**6.3 Screen Objects and Actions**

* **Restaurant Listing**: Displays available restaurants
* **Menu Items**: Allows users to select items for order
* **Order Summary**: Displays the items in the cart and total price
* **Payment Page**: Facilitates payment processing
* **Order Tracking**: Displays the real-time status of the order

# 7. REQUIREMENTS MATRIX

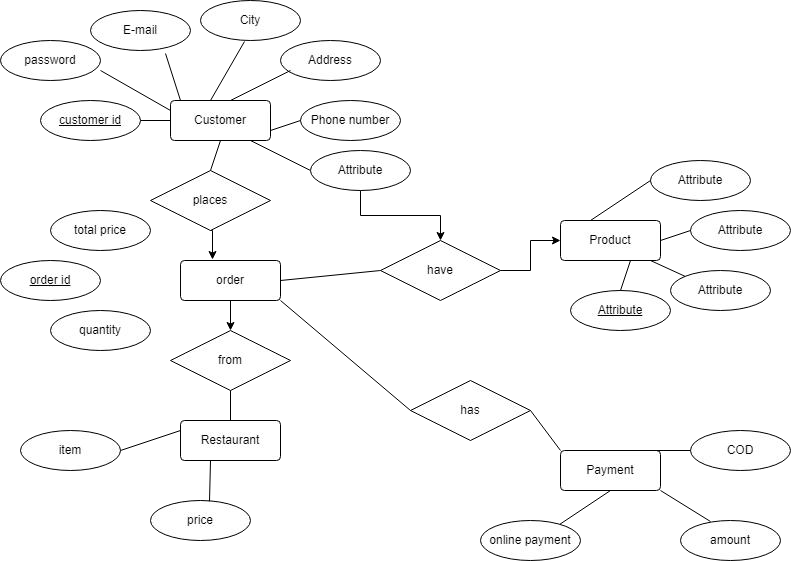
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Priority** | **Type** | **Source** | **Contained in Use Case(s)** | **Description** |
| 1 | High | F | John Smith  (Customer) | U1, U2, U3 | The system shall allow users to place food orders from available restaurants. |
| 2 | High | F | John Smith  (Customer) | U2, U3 | The system shall provide real-time tracking of order status and delivery. |
| 3 | Medium | NF | Stakeholders | N/A | The system shall maintain a response time of under 2 seconds for all user interactions. |
| 4 | High | NF | Security Team | N/A | The system shall encrypt all sensitive user data and payment information using industry-standard encryption algorithms. |
| 5 | High | F | John Smith  (Customer) | U1, U3 | The system shall allow users to search for restaurants based on location, cuisine, and other filters. |
| **ID** | **Priority** | **Type** | **Source** | **Contained in Use Case(s)** | **Description** |
| 6 | High | NF | Compliance Team | N/A | The system shall comply with GDPR and other relevant data protection regulations. |
| 7 | Medium | NF | Developers | N/A | The system shall be scalable to handle a 100% increase in concurrent users during peak hours without performance degradation. |

# 8. APPENDICES

# Appendix A: Glossary

1. Customer/User: an individual who uses the food delivery app to order food.
2. Delivery personnel: person who delivers the order/ food to the customer.
3. COD: Cash on Delivery.

# Appendix B: Analysis Models



# Appendix C: To Be Determined List

1. TBD: Localization and internationalization requirements for supporting multiple languages and regions.
2. TBD: Specific legal agreements and compliance requirements, including privacy policies and terms of use.
3. TBD: Reusable components, modules, or libraries to be integrated into the project for efficiency and cost savings.
4. TBD: Error handling mechanisms and logging practices for capturing and reporting system errors and exceptions.
5. TBD: Clarification on any additional requirements or specifications needed for the successful implementation of the SRD.