

Software Design Document

Version 1.0

September 15, 2021

Restaurant Ordering System

Harsh Saglani

hsaglani@umd.edu

Submitted in partial fulfillment
of the requirements of
ENPM809W - Project Phase 1

Table of Contents

Table of Contents	2
1. Introduction	3
1.1. Scope	3
1.2. Purpose	3
1.3. Definitions, Abbreviations and Acronyms	3
1.4. References	4
1.5. Overview	4
2. Architecture	5
2.1. Overview	5
Figure: Client-Server Architecture Design	5
2.2. Advantages of Client Server Architecture	6
2.3. Disadvantages of Client Server Architecture	6
2.4. Overview of Security Concerns & Measures	6
2.4.1. The Server System	6
2.4.2. The Database	6
2.4.3. The User and Chef's Tablets	7
2.4.4. Passwords	7
3. Database Design	8
4. Design Viewpoints	10
4.1. Introduction	10
4.2. Context Viewpoint	10
4.2.1. Block Diagram	10

1. Introduction

1.1. Scope

This project will be a web application software for ordering and reviewing food items of a restaurant. The system is designed to be a quick paced interface and help the customer decide and order the food for their meal. The ordering system will be completely automatic with almost zero interaction between the restaurant staff and the customers. The web application will help other customers in ordering food items from the restaurant by viewing previous customer ratings and reviews. In addition, all requirements documented in the SRS shall be implemented and addressed in the SDD.

1.2. Purpose

This document describes all the standards and design of the restaurant ordering system to be used by the developers to follow in the implementation phase. The SDD is an important reference not only for developers, but also for the testing programmers and editors of documentation. With the help of standards and specification described here, team members can be more efficient and successful in implementing the product

1.3. Definitions, Abbreviations and Acronyms

ER Diagram	Entity Relationship Diagram
Block Diagram	A diagram showing in schematic form the general arrangement of the parts or components of a complex system or process
Context Diagram	
Architecture Diagram	
Intranet	

1.4. References

IEEE 1016-1998 - IEEE Recommended Practice for Software Design Descriptions

1.5. Overview

This subsection briefly describes each of the remaining sections in the document. The rest of the document is organized as follows:

The next section describes in detail how the database schema of the Restaurant Ordering System will be. It highlights the main entities of the database as well as the relations between them via the Entity Relationship Diagram.

The third section describes various viewpoints from the design perspective. It includes various diagrams including the block diagram,

2. Architecture

2.1. Overview

This section describes the system architecture of the Restaurant Ordering System. The architecture is a simple client-server architecture with some additional changes. Firstly, the system will not be available or published on the internet. The system will run on a local server and will be hosted on the restaurant's intranet. Secondly, the system will be accessible by a web browser and via connecting to the local ip address of the server where the application will be hosted.

The users and chefs will access the system by using a tablet already connected to the system's intranet network. Whereas the administrator will have direct access to the server as well as the database system.

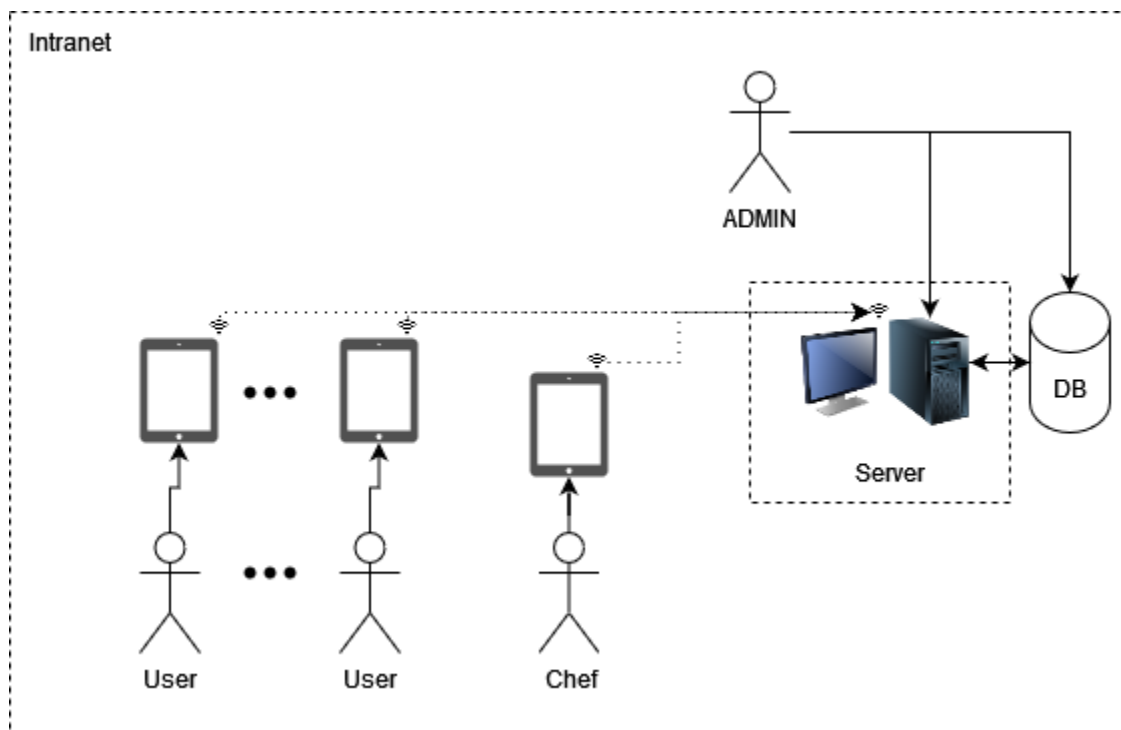


Figure: Client-Server Architecture Design

The intranet will consist of a wifi router which will make the application available to the users and chefs. The users and chefs will connect to the router via their tablets to access the application. The administrator can directly use the server system to get access to the application as well as the database. The server system can be directly connected to the router via a LAN

cable. Depending on the size of the restaurant, the restaurant must set up additional repeaters so that the network connection is strong between all the devices.

2.2. Advantages of Client Server Architecture

- 1) Can handle multiple devices and requests simultaneously
- 2) The network is centralized
- 3) There will be only one system to secure
- 4) Can be scaled easily

2.3. Disadvantages of Client Server Architecture

- 1) If the server goes down, the whole system will not be functional.
- 2) A good server can be expensive to set up.
- 3) If any part of the network fails, a lot of disruption could occur.

2.4. Overview of Security Concerns & Measures

In this subsection we will talk about all the factors that affect the overall security of the application.

2.4.1. The Server System

The entire application lives on the system at the restaurant. Therefore it is extremely essential that only the administrator has access to that system. Nobody else should be allowed to access the system. The system should be kept in a safe location away from the reach of normal users. Regular copies of the system should be maintained. Recommended duration would be monthly if not weekly. The server should at all times be connected to the local network (intranet).

2.4.2. The Database

The restaurant can choose to put the database on the same computer as the server. This means that suggestions in section 2.4.1. should be carefully followed. Even if the database is on some other system, the same measures apply. However, if the restaurant wants, they can

take daily backups of the database. Although weekly backups should suffice, unless they want to preserve each and every review of the past week in case of database failure or data loss.

2.4.3. The User and Chef's Tablets

The user's tablet should be connected to the router via wifi. The user should be able to access the restaurant system without any login, but the user must create a session by selecting the table number where the user will be ordering food to. The chefs, however, will need to login to access the orders placed by the users and to change the order statuses.

2.4.4. Passwords

All of the devices should be secured via strong passwords. Here's a list of devices or applications from higher to lower priority which should be secured via a strong password.

- 1) The server's desktop login
- 2) Database login
- 3) System login - Admin
- 4) Network login (to the router) - All tablets are connected via this.
- 5) System login - Chef
- 6) Tablet login

2.5. *Programming Language and Database choices*

The application will be a web application fully responsive for mobile, tablet and desktop devices. The backend will be coded in PHP 7 and the database will be MySQL 8.

3. Database Design

3.1. Database Overview

This section provides the translation of the informational model contained in the Software Requirements Specification (SRS) into a relational database. The relational database will be composed of tables. Each table is a series of columns which represent individual data elements. The data records in the table form the rows. Each table has a primary key. Tables are related to each other by embedding the primary key from one table into another as a foreign key to implement the relationship. Foreign keys enable the relational database management system to enforce referential integrity. Referential integrity ensures that no row in a "parent" table can be deleted if it is still referenced in a row of a "child" table.

Our database design consists of 5 tables namely Orders, Users, OrderMapping, Food and Ratings. The OrderMapping table has two foreign keys which map to the order_id in the Orders table and food_id in the Food table. Similarly, the Ratings table has a foreign key food_id which maps to food_id in the Food table. This is how food items are linked to the orders and how ratings and reviews are mapped to the food items.

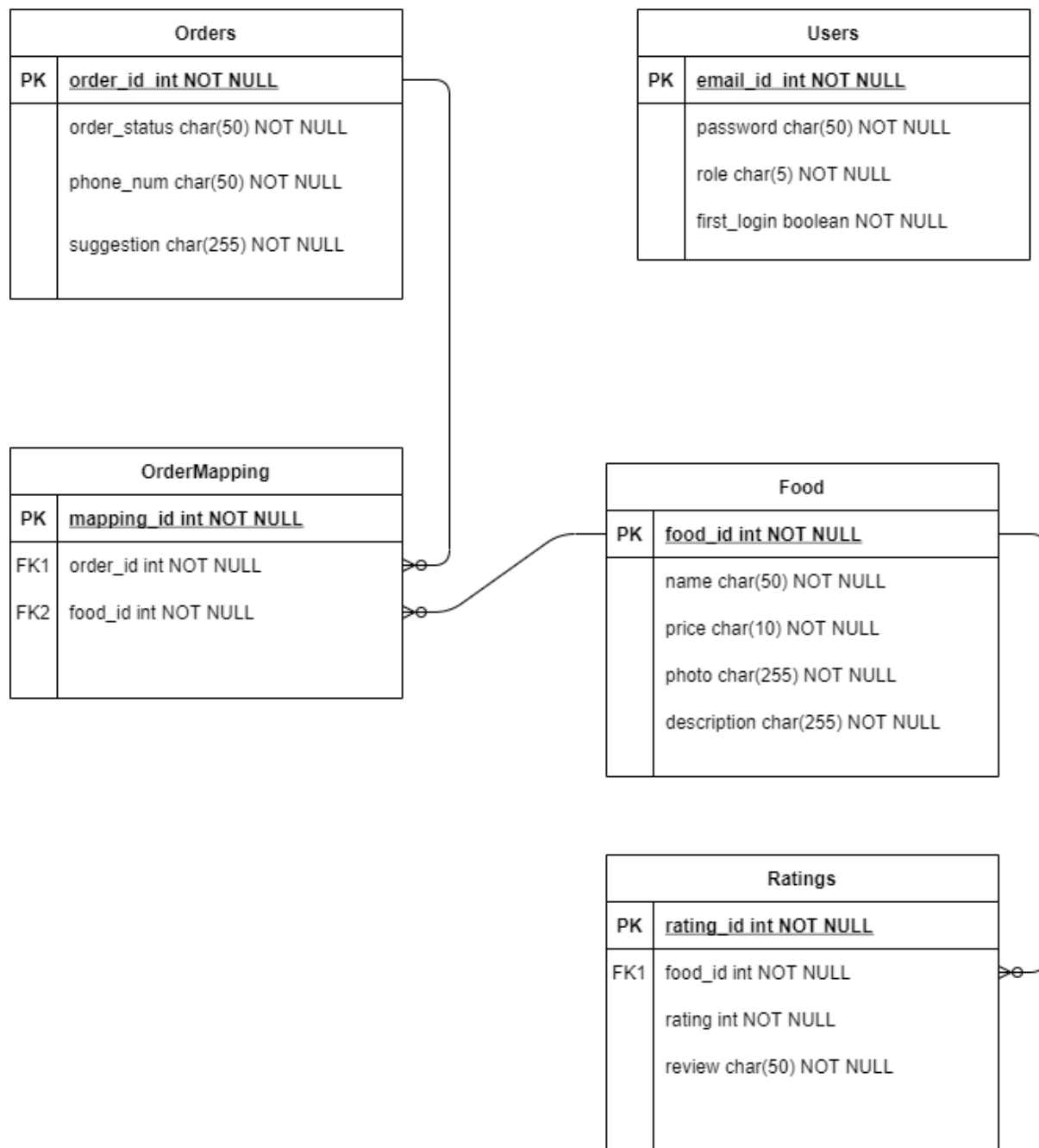


Figure 1 - ER Diagram

4. Design Viewpoints

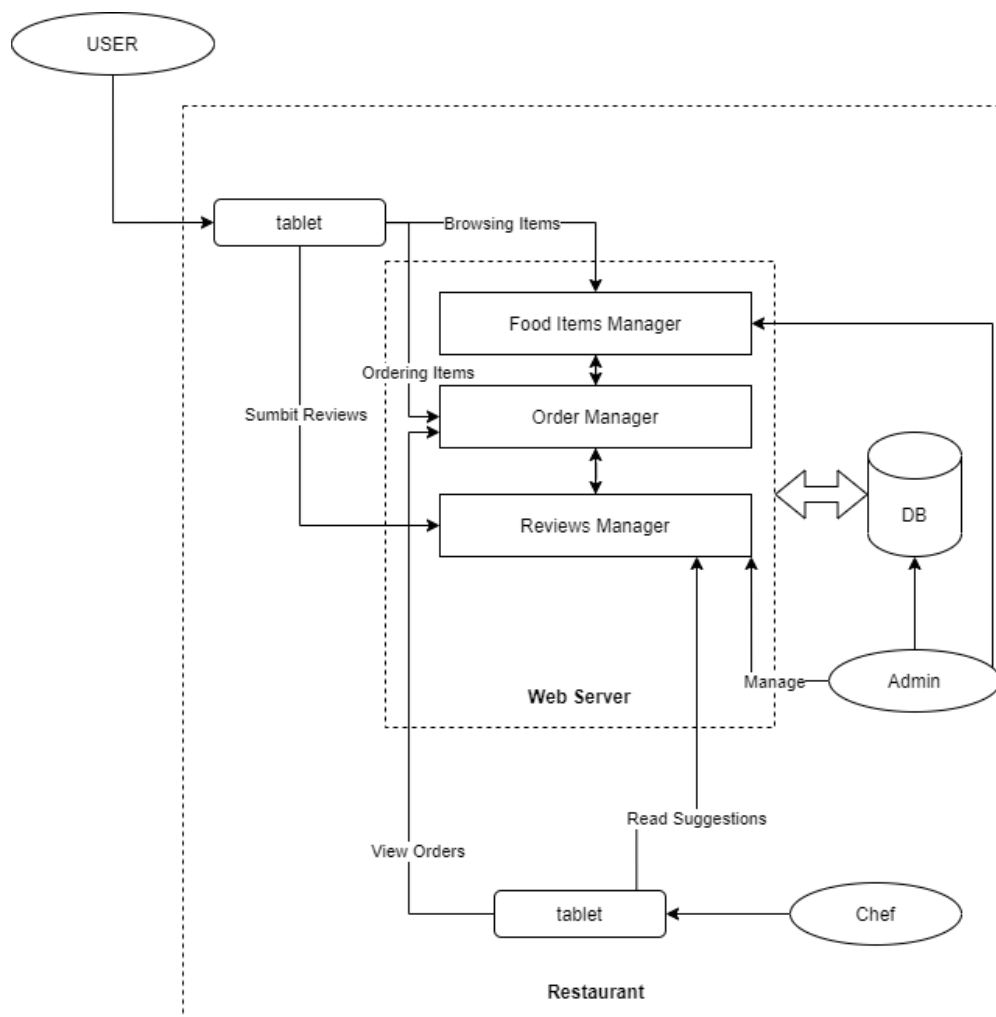
4.1. Introduction

This section describes various main design viewpoints of the Restaurant Ordering System along with some of the design concerns.

4.2. Context Viewpoint

This viewpoint of the Restaurant Ordering System illustrates the functions provided by the design subjects with reference to an explicit context. Functions, relationships, dependencies and interactions make up the services.

4.2.1. Block Diagram



There are 3 main actors for the system. The untrusted “user” will have to use the restaurant’s tablet as an interface to get access to the restaurant’s web server. The other two users are chef and admin who are part of the restaurant. The admin will have direct access to the web server as well as the database.

4.2.2. Context Diagram

