**Restaurant Ordering System**

**Software Requirements Specification**

**İstanbul Aydın University**

**Version 1.0**

Mehmet ÖRS - B2105.090156

Cihan DEMİRKAL – B2105.090137

Görkem PALALI – B2005.090056

Emir KOKUM – B2005.090085

Meriç KAYAÇAĞLAYAN – B2105.

Berkay ŞAHİN – B2005.090036

Batuhan ÖKSÜZ – B2005.090038

[1. INTRODUCTION 1](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847344)

[1.1. Purpose 1](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847345)

[1.2 Document conventions 1](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847346)

[1.3 Project scope 1](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847347)

[2. PRODUCT DESCRIPTION 1](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847348)

[2.1 Product scope 1](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847349)

[2.2 Product perspective 1](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847350)

[2.2.1 System interfaces 2](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847351)

[2.2.2. User interfaces 2](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847352)

[2.2.3 Hardware interfaces 2](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847353)

[2.2.4 Software interfaces 2](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847354)

[2.2.5 Communications interfaces 3](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847355)

[2.2.6 Memory constraints 3](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847356)

[2.2.7 Operations 3](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847357)

[2.2.8 Site adaptation requirements 3](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847358)

[2.3 Product functions 3](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847359)

[2.4 User characteristics 3](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847360)

[2.5 Limitations 4](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847361)

[2.6 Assumptions and dependencies 4](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847362)

[3. SPECIFIC REQUIREMENTS 4](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847363)

[3.1 External interfaces 4](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847364)

[3.2 Functions 5](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847365)

[3.3 Usability requirements 5](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847366)

[3.4 Performance requirements 5](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847367)

[3.5 Logical database requirements 6](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847368)

[3.6 Design constraints 6](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847369)

[3.7 Standards compliance 6](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847370)

[3.8 Software system attributes 6](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847371)

[3.9 Supporting information 7](file:///C:\Users\Mehmet\Downloads\02.%20SEN242%20SRS%20Template.docx#_Toc97847372)

**APPENDICES**

A. BUSINESS DESCRIPTION

B. USE CASE AND USER STORIES

C. ACRONYMS AND ABBREVIATIONS

D. REFERENCES

# 1. INTRODUCTION

## 1.1. Purpose

## The purpose of this SRS document is to present a detailed description and specify the requirements, rules and constraints for the development and implementation of the Restaurant Ordering System (ROS). This document is intended for the stakeholders, project managers, developers as well as the end users such as customers, staff (waitress, cook etc.).

## 1.2 Document conventions

* Diagrams are used to present the complex information in a clear way and make it more understandable for users.
* Heading and subheadings are used to distinguish the related sections.
* Some common acronyms used in the document are: ROS (Restaurant Ordering System), UI (User Interface), DBMS (Database Management System), CRM (Customer relationship management), IMS(Inventory Management System).
* Use of active voice is preferred over passive voice to improve clarity and readability.

## 1.3 Project scope

ROS is a web-based app which aims to help restaurants increase their revenue, save time and improve order accuracy. The purpose of the ROS is to streamline the ordering process in a well-organized way and enhance the customer experience by providing a user-friendly interface and provide efficient communication between staff members and also with customers.

The scope of the document is to provide a detailed set of requirements, constraints and the rules for the development and implementation of ROS. This document will serve as a reference for various stakeholders such as project managers, end users and developers.

# 

# 2. PRODUCT DESCRIPTION

## 2.1 Product scope

The two main components of the product are: a customer facing a website interface for ordering and a back-end management system for the restaurant staff.

The customer facing website will allow the users to view the menu, select the items they want to order, customize the items (amount, preference etc.) and select the appropriate payment type that suits them. Customers will be able to create an account, save their preferences for further orders. Customers will also be able to give feedback and rate the food and service, as well as to provide comments and complaints. This feedback will be used to help improve the restaurant’s offerings and service. Customers will be prompted to provide feedback after completing an order and will be able to view their order history and feedback history.

The back-end management system will provide restaurant staff (administrator) with the ability to view and manage incoming orders, updating inventory, and managing payments. The administrator will be able to log in to the system, view order details, update order status, and communicate with customers through the system. Also, administrator members can adjust the data in DBMS manually and change the information when needed.

There will be QR codes placed on the tables in the restaurants which users can scan with their phones, and they will be able to connect to the website more easily and quickly. The website will also provide information about the restaurant, such as location, hours of operation and contact information (phone number, email address).

The aim of the product is to allow customers to place orders for food and beverages and enable the staff of restaurants to manage and fulfill those orders.

This product scope statement is consistent with the higher-level system requirements specification, which outlines the overall functionality and objectives of the ROS system.

## 2.2 Product perspective

ROS is a standalone web application that interacts with customers and staff members. So, it is not an element of a large system. But it needs to interact with other systems such as DBMS for storing and retrieving data, payment gateway to confirm the payments by interacting the banking accounts.

The ROS interacts with some external systems when operating. Such systems and the purpose of interaction with them are payment gateway to process the payment, inventory management system to check the availability of items in real time and update the inventory after every order, CRM (Customer relationship management) to store and manage the customer data, order history and feedbacks, social media platforms to enable customers to share their feedback and experience on social media, marketing tools to send promotional emails or messages to customers, based on their order history and preferences.

The app interacts with customers and staff through a web-based GUI that can be accessed from any device with an internet connection and compatible with web browser. The GUI provides customers with access to an account login page, menu, payment options, feedback and rate mechanism. The GUI also integrates with payment gateway to process payments and confirm orders. For the administrator, it provides access to the system’s management tools for tracking orders, managing inventory information and customer feedback views. The administrator can update the order status and view feedback.

The hardware interfaces required are web server and database server to host the application and store the data. The device does not have any direct hardware interfaces with peripheral devices additionally.

The software interfaces required for the ROS include web servers, web frameworks, and database management systems (DBMS). The system uses HTTP and HTTPS protocols for communication between client and server, and the DBMS manages the storage and retrieval of data.

ROS operates within the constraints of a web-based application. It does not require any special memory or processing requirements, and it can run on any modern web browser.

The ROS is designed to be adaptable to different restaurant locations and settings. The system can be customized to meet the specific needs of different restaurants, such as menu items, prices, and payment options. Additionally, the system can be configured to support different languages, currencies, and time zones.

## 2.2.1 System Interfaces

The Restaurant Ordering System (ROS) interacts with several external systems to accomplish its functionality. The following system interfaces are required:

Payment Gateway: The ROS must interact with a payment gateway to process customer payments securely. The interface uses blowfish for encryption and secure data transmission.

Inventory Management System: The ROS must interact with an inventory management system to check the availability of supplies in real time and update the inventory after each order. The interface must support database connections and allow for the change of inventory data between the ROS and inventory management system.

CRM (Customer Relationship Management) System: The ROS must interact with a CRM system to store and manage customer data, order history, and feedback. The interface must support database connections and allow for the change of data between the ROS and the CRM system.

Social Media Platforms: The ROS must interact with social media platforms to enable customers to share their feedback and experience on social media.

Marketing Tools: The ROS must interact with marketing tools to send emails or messages to customers based on their order history and preferences.

Diagram

Description automatically generated

Fig 1. Ecosystem map

Diagram

Description automatically generated

Fig 2. Context Diagram

## 2.2.2 User Interfaces

a) The software product has two main user interfaces: one for customers and another for staff. For the customer interface, the software product offers a web-based GUI accessible from any device with an internet connection and a suitable web browser. The GUI includes an account login page, menu, payment options, feedback and rating mechanism.

The software product for the staff interface also provides a web-based GUI that can be accessed from any device with an internet connection and asuitable web browser. The GUI includes system management tools such as order tracking, inventory management, and customer feedback views. In addition, a notification device is placed in the kitchen to inform staff when an order is delivered.

b) The system should provide clear feedback to users to confirm actions taken or errors encountered. For example, the system should provide both long and short error messages to guide users on how to correct errors encountered.

The notification device placed in the kitchen should be designed to be easy to read and use by kitchen staff. It should provide clear and concise information about the order placed, including any special requests or notes. The notification device should be designed to ensure that administrator can easily update the order status and communicate with customers if needed.

## 2.2.3 Hardware Interfaces

As mentioned earlier, the system consists of a web server and a database server, which are hosted on cloud-based infrastructure. Therefore, there are no specific configuration characteristics, device support requirements, or protocols that need to be specified for hardware interfaces in this case.

## 2.2.4 Software Interfaces

- Web server

Name: Apache HTTP Server

Mnemonic: Apache

Specification number: N/A

Version number: 2.4.48

Source: <https://httpd.apache.org/>

- Database management system

Name: MySQL

Mnemonic: MySQL

Specification number: N/A

Version number: 8.0.26

Source: <https://www.mysql.com/>

- Payment gateway

Name: Stripe

Mnemonic: Stripe

Specification number: N/A

Version number: N/A

Source: <https://stripe.com/>

For each interface, the purpose of the interfacing software as related to this software product is as follows:

-Web server (Apache HTTP Server): The Apache HTTP Server is used to serve web pages to users accessing the software product through a web browser. It is responsible for receiving and responding to HTTP requests.

-Database management system (MySQL): MySQL is used as the backend database management system for storing and retrieving data of orders, menus, customers, and other system data.

-Payment gateway (Stripe): Stripe is used to process payments from customers using credit card or other payment methods.

The references for the software products mentioned are <https://httpd.apache.org/docs/2.4/> for apache, <https://dev.mysql.com/doc/refman/8.0/en/> for mysql and <https://stripe.com/docs/api> for stripe.

## 2.2.5 Communication Interfaces

The software product interfaces to communication protocols such as:

- Local network protocol TCP/IP: TCP/IP (Transmission Control Protocol/Internet Protocol) is a standard communication protocol used for connecting devices over a local network or the internet. The software product uses TCP/IP for communication between the customer website, the kitchen view, and the payment gateway.

- HTTP/HTTPS: HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure) are application-level protocols used for communication between web servers and web clients. The software product uses HTTP/HTTPS for communication between the customer website and the web server.

- WebSocket: WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection. The software product uses WebSocket for real-time communication between the customer website and the kitchen view.

## 2.2.6 Memory Constraints

## The software requires a minimum of 1 GB of RAM to run smoothly.

## 2.2.7 Operations

There are different types of operations that the user of the software product will need to perform. Such operations are:

* Normal operaitions: These are the regular operations the user will perform in a day-to-day basis. Such operations are creationg new orders, updating menus, managing customer data and generating reports.
* Special operations: These more complex operations that the user may need to perform, such as system maintenance tasks and data backups.
* Periods of interactive operations and periods of unattended operation: Interactive operations refer to the periods when the user interacts with the software product to perform actions, such as entering data or viewing reports. Unattended operations refer to the periods when the software product performs automated tasks, such as data processing or backups, without user intervention.
* The software product may include data processing support functions, such as data validation, data conversion, and data export/import.
* Backup and recovery operations: The software product should provide backup and recovery functionality to prevent data loss in case of system failures or disasters.

## 2.2.8 Site adaptation requirements

## The software product requires specific initialization data to be ready for use. Some initialization required are:

## Geographical location of the site

## Local currency and tax rates

## Time zone of the app

## Local language and cultural preferences

## Some network configurations (DNS, IP address, port configuration)

## The specification of the site or mission related features that should be modified to adapt the software to a particular installation are:

## Customization of user interfaces to match site-specific requirements.

## Modification of reports and documents to conform to local regulations and standards.

## Configuration of backup and recovery procedures to meet site-specific requirements.

## Integration with site-specific data sources and formats

## 

## 2.3 Product functions

The major functions that the Restaurant Ordering System (ROS) will perform are:

* User Registration: The system will allow users to create a new account by providing their personal information, such as name, email, phone number, and address.
* User Login: The system will authenticate users and allow them to access their accounts.
* Menu Management: The system will allow restaurant owners to manage their menu items, including adding, updating, and deleting items.
* Order Placement: The system will allow users to place an order by selecting items from the menu and specifying any customizations or special requests.
* Order Processing: The system will notify the restaurant staff of new orders and allow them to view order details and mark orders as in-progress or complete.
* Payment Processing: The system will securely process payments for orders, including accepting credit card payments and integrating with third-party payment gateways.
* Payment Processing: The system will securely process payments for orders, including accepting credit card payments and integrating with third-party payment gateways (stripe in our product).
* Order History: The system will keep track of user order history and allow users to view their past orders.
* Customer Support: The system will provide a customer support interface, allowing users to submit inquiries, report issues, or request refunds.

Diagram

Description automatically generatedFig 3. Data Flow Diagram

## **2.4 User characteristics**

The intended users of the software product are restaurant staff, including waitress-waiter, cooks, administrator, and also customers. The users are expected to have a range of educational levels, technical expertise, and experience with restaurant operations. Some users may have disabilities that need to be taken into consideration to ensure accessibility and usability.

Specific user characteristics that may influence usability include:

* Technical expertise: Some users may have limited experience with technology or the specific software product, while others may be more technically proficient. The software should be designed to be user-friendly and intuitive, with clear instructions and minimal technical jargon.
* Experience with restaurant operations: The software product is designed for use in a restaurant setting, and users are assumed to have some level of familiarity with restaurant operations. However, the software should be designed to be easily adaptable to different restaurant settings and workflows.
* Educational level: The users may have varying education levels. The software should be designed to be easily understandable and accessible to all users.
* Disabilities: Some users may have disabilities that rely on assistive technology such as screen readers, magnifiers, and speech-to-text software to access digital content. The software should be compatible with these tools.
* Multilingual support: The software may be used by users who speak different languages. The software should be designed to support multiple languages, with clear and accurate translations of all text and instructions.These user characteristics will be taken into consideration when specifying the specific requirements for the software product.

These user characteristics will be taken into consideration when specifying the specific requirements for the software product.

## 

## 2.5 Limitations

Here are some items that will limit the supplier's options:

* Regulatory policies: The software must comply with relevant data privacy and security regulations, as well as any regulations related to the handling and storage of credit card information.
* Hardware limitations: The software must be able to run on various devices and should not have overly demanding system requirements that would prevent it from running on older or lower-end hardware.
* Interfaces to other applications: The software must be able to integrate with third-party payment gateways, such as Stripe, and any other systems used by the restaurant, such as inventory management software.
* Parallel operation: The software should be able to handle multiple users and orders at the same time without significantly slowing down or crashing.
* Audit functions: The software must have auditing capabilities to track changes made to orders, payments, and other important data.
* Control functions: The software must provide appropriate access controls to ensure that only authorized users can perform certain functions, such as deleting orders or processing refunds.
* Quality requirements: The software must be reliable and able to operate without significant downtime, especially during peak hours.
* Criticality of the application: The software is critical to the daily operations of the restaurant and must be able to handle all orders, payments, and other functions in a timely and accurate manner.
* Safety and security considerations: The software must be designed with high security and protect against unauthorized access to sensitive data such as customer information and payment details.
* Physical/mental considerations: The software should be designed to be easy to use and navigate for users with physical or mental disabilities, and should be compatible with assistive technologies such as screen readers and speech-to-text software.

## 2.6 Assumptions and dependencies

## Listed below are the factors that may affect the requirements stated in this SRS. Any changes to these factors may impact on the requirements and the software development process.

## Availability of necessary hardware components such as servers, network equipment and kitchen display devices.

## Availability and compatibility of required third-party software, including the operating system, payment gateway and database management system.

## Compliance with relevant regulations and standards for payment processing and data protection.

## Availability of personnel with the necessary technical expertise to set up, configure, and maintain the system.

## Accessibility of the system for users with disabilities, including compatibility with assistive technologies such as screen readers and magnifiers.

# 3. SPECIFIC REQUIREMENTS

## 

## **3.1 External interfaces**

Inputs:

* Customer Info: Customer info is used to take customer username, password and order information from customer on the restaurant.(string max 50 characters)
* Inventory: This info is used to keep the stocks of ingredients in the kitchen in the database.
* Payment Info: Payment info is used to take and keep customer’s cardID, expirationDate, nameonCard, cardNumber and send to the bank.
* Feedback: Feedback is used to take comments and reviews from users.(string max 300 characters)

Outputs:

* Order Info: Order info is a string input for taking the foods that customer choose from the menu.
* Feedback: Feedback output is for the restaurant managers and staff.

User Interface

* Name of item: User Interface
* Description of purpose: Provide an easy-to-use platform for customers and staffs to interact with the software.
* Input source: Mouse, Keyboard, Touchscreen, etc. Output destination: Computer screen or mobile device screen.
* Valid range, accuracy, and/or tolerance: username and password have to comply with some rules(strenght passoword exc).
* Units of measure: For password min. 8 characters.
* The timing is very shortly after user input.
* The user interface is related to all other inputs and outputs of the system.
* The screen layout is designed to be user-friendly, with clear navigation menus and buttons to processes for the system.
* The window layout is designed to be responsive.
* Data is presented in a variety of formats, including text, images, and video.
* Commands are entered using text input or button clicks.

## The system displays end messages to confirm or successful processes or to alert users to errors.

Payment Gateway Interface

* Name of item: Payment Gateway Interface
* Description of purpose: Provide a secure and easy payment system.
* Input source: Customer payment information. Output destination: Payment system.
* Valid range, accuracy, and/or tolerance: Valid range and accuracy have to be true for card information.
* Units of measure: N/A
* The timing is dependent on the payment system.
* The payment gateway interface is closely related to the order and inventory management systems of the software.
* Screen formats/organization: N/A
* Window formats/organization: N/A
* Data formats: JSON.
* Command formats: The command format should include a field has the payment method like visa, mastercard exc. and other fields for card number, expiration date, CVV code, and other relevant card information.
* End messages: Payment system sends confirmation or error messages to the software and customer.

Database Interface

* Name of item: Database Interface
* Description of purpose: Store all data for the system.
* Input source: Software system. Output destination: Database server.
* Valid range, accuracy, and/or tolerance: Dependent on database schema and constraints.
* Units of measure: N/A
* The timing is very short for data storage operations.
* The database interface has a relationship to all inputs and outputs of the system.
* Screen formats/organization: N/A
* Window formats/organization: N/A
* Data formats: sql
* Command formats: Request and response commands for the system.
* End messages: N/A

## Kitchen Display System

## Name of item: Kitchen Display System

## Description of purpose: Displays incoming orders to the kitchen staff

## Source of input or destination of output: Inputs from the ordering system when customers give order and outputs to kitchen staff

## Valid range, accuracy, and/or tolerance: N/A

## Units of measure: N/A

## Timing: Real-time

## Relationships to other inputs/outputs: Integrates with the online ordering system

## Screen formats/organization: Display screen in the kitchen

## Window formats/organization: N/A

## Data formats: JSON

## Command formats: HTTP requests

## End messages: N/A

## **3.2 Functions**

## Validity checks on the inputs: The system should check the validity of all inputs if they are in the expected format, data type, and range. For example, the system should check that the customer's name is not empty, the order quantity is a positive integer, and the payment information is valid.

## Exact sequence of operations: Until an order being given and delivered, here are the sequence of operations that happen:

## Customer selects items from the menu and adds them to the cart.

## Customer provides the payment information.

## Payment gateway validates payment information and confirms payment.

## System adds the order to the database and updates the inventory.

## System sends a notification to the restaurant with the order details and customer information.

## Waitress-waiters are notified when the cook is ready.

## Order is delivered to the customer and order is marked as complete.

## Responses to abnormal situations:

## Overflow: Maximum number of orders must be limited to prevent overloading.

## Functional Requirements:

## FR01: Users Shall be greeted by login and signup buttons on the opening of the application.

## FR02: Users Shall be able to sign up just by using name surname a password and an email.

## FR03: Users shall be able to login using their name and password.

## FR04: Users shall be able to see all items on the menu.

## FR05: Users shall be able to add to cart or directly order items from menu.

## FR06: Users shall be able to order all items on their cart by one button.

## FR07: Users shall be able to order other items while their already existent order is getting ready.

## FR08: Users shall be able to remove items from their cart.

## FR09: Users shall be able to cancel their orders one by one or all at once.

## FR10: Users shall be able to change their table number by one button.

## FR11: Users shall be able to view their order history.

## FR12: Users shall be able to select their payment method after completing their order (cash or credit card).

## FR13: Users shall be able to hear any writing on the screen by one button.

## FR14: The system shall send notification to the kitchen display screen when an order is given.

## FR15: The display screen shall show order owners name, table number and order details.

## FR16: The system shall send notification to the waitress when an order is ready at the kitchen.

## FR17: The cooking staff shall be able to send notification to waiter-waitress when order is ready.

## FR18: Users shall be able to write a review about their order.

## FR19: Users shall be able to send a change or refund request for their order.

## FR20: Users shall be able to call a waitress to their table with one button.

## FR21: Users shall be able to rate their order.

## FR22: Users shall be able to rate their overall service by employees (waitresses etc.).

## FR23: Users shall be able to write a petition about an employee.

## FR24: Users shall be able to specify the quantity for each item they want to buy.

## FR25: Administrators shall be able to update the inventory information.

## FR26: The system shall calculate the total price of the order including tax.

## FR27: The system shall display a confirmation prompt with the total price and the items bought for customers when they give order.

## FR28: The system shall support screen reader and text to speech conversion tools to help customers who have visual impairments when giving order.

## FR29: The system shall store the order records of customers.

## FR30: The system shall allow restaurant staff to update the menus.

## FR31: The customers shall be able to connect to the website directly using the QR codes on the tables.

## **3.3 Usability requirements**

* Users should be able to finish their order and complete the payment process in two minutes on average, and a maximum of five minutes.
* The system should use simple language and avoid technical terms, so people can easily understand how to use the system.
* The user shall be able to search and find a product from the catalog in an average of 30 seconds, and in a maximum of 1 minute, 90% of the time.
* The system should make it simple for users to change their orders, with available options for adding or removing items.
* The system should show a brief error message so that the user can understand the problem easily.
* The user interface shall use maximum 2 clicks to access the most used features.

## **3.4 Performance requirements**

## Static performance requirements:

## NFR1: Users shall be able to verify their table number and start to ordering within 1 seconds.

## NFR2: The application shall let at least 300 simultaneous users.

## NFR3: Application shall be able to store at least 100000 user accounts and their payment information.

## NFR4: Application shall be able to run with 2 gb of ram.

## NFR5: Application shall be able to reach an users payment information within 1 seconds while completing an order.

## Dynamic performance requirements

## NFR6: At least 99.99 percent of orders shall reach to kitchen screen without any data lost.

## NFR7: Orders that have less than 10 items shall reach to kitchen screen within 3 seconds after customer enters their order.

## NFR8: Minimum of 200 payment transactions can be made in a second.

## NFR9: Applications Size shall not pass 50 mb.

## **3.5 Logical database requirements**

## Types of information used by various functions:

## Customer information

## Menu items

## Orders

## Frequency of use:

## Customer information is used frequently for login and order processes.

## Menu item information are accessed frequently since customers check menu to give order each time.

## Orders may be accessed frequently for updating stock according to the information of orders.

## c) Accessing capabilities:

## Customers should be able to update their information and view their order history.

## Restaurant owners should be able to manage their menu and view/order customer information.

## Administrators should have access to all data for managing the system.

## d) Data entities and their relationships:

## - Customers have many orders, and each order belongs to a single customer.

## - Customers have feedback and each feedback belongs to a single customer.

## - Orders have payments and each payment belongs to a single order.

## e) Integrity constraints:

## - Each order must have at least one menu item.

## - Menu item prices must be greater than zero.

## - Customer payment information must be valid.

## f) Data retention requirements:

## - Customer information should be retained until 1 year since the customer’s last login.

## - Order histories are retain for 3 months after it is given.

# Customers Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Description** | **Example** |
| CustomerID | int | 4 | Unique identifier for the customer | 1907 |
| FirstName | nvarchar | 50 | First name of the customer | Cihan |
| LastName | nvarchar | 50 | Last name of the customer | Demirkal |
| Email | nvarchar | 100 | Email address of the customer | [Cihan1907fb@gmail.com](mailto:Cihan1907fb@gmail.com) |
| Phone | nvarchar | 20 | Phone number of the customer | (555) 555-5555 |
| Address | nvarchar | 250 | Home address of the customer | Istanbul, Kadıkoy st 1907 |

## **Data Dictionary**

# Orders Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Description** | **Example** |
| OrderID | int | 4 bytes | Order identification number | 100 |
| CustomerID | int | 4 bytes | Customer identification number | 1907 |
| OrderDate | datetime | 8 bytes | Date and time the order was placed | 2023-04-17 |
| PaymentID | Unique identifier for the payment | 150 | 4 | PaymentID |

# Inventory Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Description** | **Example** |
| InventoryID | int | 4 bytes | Inventory identification number | 120 |
| IngredientName | Varchar | 50 | Name of ingredient | Tomato |
| Quantity | int | 4 bytes | Quantity of the ingredient | 6 |

# Payments Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Description** | **Example** |
| PaymentID | int | 4 | Unique identifier for the payment | 150 |
| PaymentMethod | nvarchar | 50 | Type of payment (credit card, cash, etc.) | Visa |
| Amount | Decimal | 10,2 | Amount paid | 25.99 |
| PaymentDate | datetime | 8 | Date and time the payment was made | 2023-04-17 14:35:00 |

## **3.6 Design constraints**

# Some of the design contraints developers of the system must obey:

# The system must compatible with popular web browsers Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

# As programming languages, python and javascript shall be used.

# For database management system, MySQL shall be used.

# The application architecture template must be based on MVC(Model View Controller).

# Backend framework should be django and frontend framework should be react.

# The Apache web server shall be used.

# Bootsrap shall be used for designing the UI.

## **3.7 Standards compliance**

Report Format: The report format ensures that the data is stored and managed correctly. Reports usually contain information such as ordering statistics, customer feedback, inventory tracking, and other business data. It is important that the reports are improved with visualization using charts, graphs, and tables.

Example: The restaurant creates a report for daily sales data. This report contains information such as total sales, average order quantity, average rating for orders and other statistics. The report can be supported with tables and graphs.

Data Naming: Data names ensure that data is correctly and uniquely identified. The system should use camelCase for data naming.

Accounting Procedures: Accounting operations ensure that financial transactions are accurately recorded and managed. The system shall allow users to view their payments by category.

Audit Tracking: Audit tracking events are an important factor for restaurants' ordering systems. Activities such as order tracking, inventory tracking and customer feedback should be provided for staff members managing the system.

## **3.8 Software system attributes**

## Reliability:

## NFR1: Application shall not stay unavailable more than 3 seconds in work hours in average.

## Availability:

## NFR2: Application shall be available 24/7 with at least 99.999 percentage availability.

## Security:

## NFR3: Customers payment information shall be kept encrypted by blowfish method.

## Maintainability:

## NFR4: A programmer experienced with the system shall be able to modify or upgrade existing code with 24 hours or less of development time.

## Portability:

## NFR5: Application shall be able to run from popular browsers google chrome, opera, internet explorer, Microsoft edge.

## **3.9 Supporting information**

## Some additional supporting information for the SRS is:

## a) In a recent survey with 1000 ROS users, 90% of the users reported that they found the platform easy to navigate and use. Additionally, 85% of the participants reported that they were satisfied with the overall functionality of ROS.

## b) Supporting or background information that can help the readers of the SRS:

## The website has a shopping cart, payment gateway integration, and order tracking to make giving orders easy for customers.

## c) Our product aims to solve the problem of time-consuming tasks when giving orders in restaurants. With this product, customers can give orders in an instant time and do not need to wait for a turn to give orders. This way, both customers’ satisfaction is improved, and restaurants’ time is saved. Also, the restaurant raises its reputation by achieving customers satisfaction.

## d) The software should be packaged in a way that makes it easy to install and configure on a system. Also it should be well explained in a way that each modules use should be described.

APPENDICES

**A. BUSINESS DESCRIPTION**

**A.1 Business objectives**

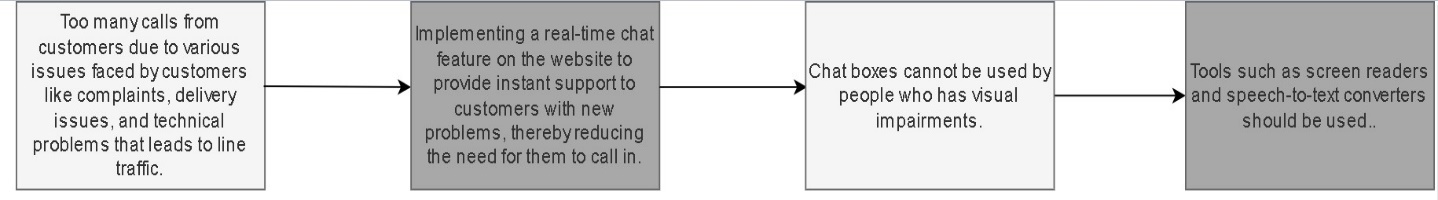
1- Increased Customer Satisfaction: The Restaurant Ordering System allows customers to order the restaurant's meals and services more easily and quickly. In this way, customer satisfaction increases, and the reputation of the restaurant rises.

2-Increasing Business Efficiency: The Restaurant Ordering System ensures that orders are taken faster and more accurately. This increases the efficiency of the restaurant and allows more customers to be served with less time and cost.

3-New Customer Acquisition: The Restaurant Ordering System helps the restaurant gain new customers by offering a modern and technological approach and with the use of social media. In this way, the market share of the restaurant increases, and its revenues are increased.

4-Building an Innovative Image: The Restaurant Ordering System helps the restaurant create a technological image. This allows the restaurant to create an innovative and modern image among its customers.

5-Data Analysis Possibility: The Restaurant Ordering System allows the restaurant to collect and analyze order data. In this way, it becomes possible for the restaurant to better understand customer preferences and determine business strategies accordingly.



**Fig 5. Business Objectives Model**

**A.2 Business scope**

a) Business Domain: Restaurant Ordering System

b) The business domain of the Restaurant Ordering System includes a range of activities that are involved in the process of receiving, managing, and fulfilling orders for food and beverages in a restaurant. This includes:

Order taking: The process that receiving orders from customers, either through a physical menu or a digital interface.

Order management: The process that organizing and tracking orders, including assigning them to specific kitchen stations and monitoring their progress.

Payment processing: The process that accepting and processing payments from customers with cash, credit cards, and mobile payments.

Inventory management: The process that monitoring and restocking restaurant supplies, such as food and cooking materials.

Reporting and analytics: The process that collecting and analyzing data about customer orders, sales, and inventory levels to make informed decisions about restaurant operations.

The scope of the Restaurant Ordering System business domain also includes interactions with external entities, such as:

Customers: People who order food and drinks at the restaurant.

Suppliers: The companies or individuals who provide the restaurant with food and cooking materials, and other supplies.

Payment processors: The companies that process and handle payments made by customers.

Delivery services: The companies that provide delivery services for orders placed through the restaurant's ordering system.

Entities outside the scope of the Restaurant Ordering System business domain include that:

Other restaurants or businesses that are not directly related to the management of orders and inventory for food and beverages.

Government or regulatory agencies that oversee restaurant operations but are not directly involved in the day-to-day operations of the Restaurant Ordering System.

c) The scope of the system being developed or changed is to provide a digital interface for customers to place orders, and to streamline the order management process for restaurant staff. The system will support the business activities of order taking, order management, payment processing, and inventory management. Assumptions include that the system will be able to integrate with existing payment processors and inventory management systems used by the restaurant. The system will also be able to provide reporting and analytics on customer orders, sales, and inventory levels to inform restaurant operations.

**A.3 Business overview**

Internal Divisions:

- Order Taking: Responsible for receiving orders from customers, either through a physical menu or a digital interface.

- Order Management: Responsible for organizing and tracking orders, including assigning them to specific kitchen stations and monitoring their progress.

- Inventory Management: Responsible for monitoring and restocking restaurant supplies, such as food and cooking materials.

- Reporting and Analytics: Responsible for collecting and analyzing data on customer orders, sales and inventory levels to make informed decisions about restaurant operations.

External Entities:

* Customers: The individuals who give orders for food and beverages in the restaurant.
* Suppliers: The companies or individuals who provide the restaurant with food ingredients, cooking utensils, and other supplies.
* Payment Gateway: The third-party service providers that enable the processing of payments made by customers, including credit cards, mobile payments, and other forms of electronic transactions.
* Delivery Services: The companies that provide delivery services for orders placed through the restaurant's ordering system

The interrelationships between these entities are as follows:

* Customers place orders through the Restaurant's Order Taking division, which are then managed by the Order Management division.
* Payment Gateway is responsible for handling the payment for the orders placed by customers.
* Inventory Management System monitors and replenishes restaurant supplies that are sourced from Suppliers.
* Reporting and Analytics collects data from all the other divisions and external entities to provide insights for informed decision-making.
* Delivery Services may be used to fulfill orders placed through the restaurant's ordering system.

Overall, the Restaurant Ordering System business domain involves multiple internal divisions and external entities that work together to ensure the efficient and effective management of orders and inventory for food and beverages in a restaurant.

**A.4 Business environment**

1-Market Trends: The restaurant industry is evolving so it's important to consider future market trends. Changes in customer preferences, new technologies and emerging competitors can affect system requirements.

2-Laws and Regulations: There are various laws and regulations that restaurants must comply with, such as food safety regulations, health and safety standards. The system should be designed to meet these requirements.

3-Social Responsibility: Restaurants are expected to implement practices such as reducing food waste and using environmentally friendly packaging. The system should facilitate the adoption of such practices.

4-Technology Base: The system should be designed so that it can be easily upgraded to include new technologies and features.

5-Economic Environment: Economic conditions can affect a restaurant's costs. The system should be designed to optimize the operating costs of the restaurant.

6-Customer Expectations: The system should be designed to meet customer expectations in terms of ease of use and comfort.

7-Employee Needs: The system should be designed to provide employees with the tools and information they need to perform their duties effectively.

Chart, box and whisker chart

Description automatically generated

**Fig 6. Business Data Diagram**

**A.5 Business Rules**

BR01

Classification: Constraint

Description: Customers must provide valid credit card information to complete payment.

BR02

Classification: Constraint

Description: A customer must be over 18 years old to buy alcoholic beverage.

BR03

Classification: Computation

Description: The total cost of an order must be calculated based on the prices of selected menu items and the taxes applicable.

BR04

Classification: Action Enabler

Description: If the customer gives an order for the first time, the payment type options must be prompted.

BR05

Classification: Action Enabler

Description: If the customer gives an order, the kitchen display system must be notified and the cooking staff must enter an average time for delivery.

By documenting these business rules, the development team can ensure that the Restaurant Ordering System is designed and implemented to meet the specific requirements and constraints of the business domain.

**B. USE CASE AND USER STORIES**

**Diagram

Description automatically generated**

**Fig 7. Use Case Diagram**

## Use Case 1:

## Identifier: UC01

## Name: Cook Food

## Description: This use case starts the order cooking process.

## Trigger: The cooking staff starts the cooking process.

## Preconditions: Order must be confirmed.

## Postconditions: The cooking process is checked as started.

## Steps:

## 1- Order is confirmed.

## 2- Available cooking staff checks the order as in process from the kitchen display system.

## 3- The cooking is in process.

## Use Case **2**:

## Identifier: UC02

## Name: Order

## Description: Customers can order food and beverage by selecting items from the menu and adding to cart.

## Trigger: When a customer gives an order after selecting the items they want.

## Preconditions: Payment must be confirmed by the payment gateway.

## Postconditions: The order is given and is in the cooking process.

## Steps:

## 1- Customer selects the food-beverage they want and add them to cart

## 2- The payment is confirmed by the payment gateway

## 3- Order is in process

## Use Case 3:

## Identifier: UC03

## Name: Order Ready

## Description: This use case states that the order is ready.

## Trigger: The waiter-waitress informs the customer that the order is ready and is in the delivery process.

## Preconditions: The cooking staff should state that the order is ready.

## Postconditions: Customers are notified that their order is ready and is being delivered.

## Steps:

## 1- Cooking staff states that the order is ready

## 2- Waitress-waiters are notified

## 3- The order is in delivery process

Use Case 4:

Identifier: UC04

Name: Feedback

Description: This use case allows customers to comment or complain about the ordered product.

Trigger: Customer clicks the feedback button on the home page.

Preconditions: User log into own account.

Postconditions: System publishes the comment or sends the complaint to staff.

Steps:

1. Customer log into own account.
2. Customer click feedback button.
3. Customer enter comment or complain.
4. System shows the comment or complain to cook and waiter.

## Use Case 5:

## Identifier: UC05

## Name: Payment

## Description: This use case allows customers to pay for the product that ordered on the website.

## Trigger: Customer click the pay button on the order page.

## Preconditions: User approve the payment of the selected product.

## Postconditions: Approved by the bank according to the payment method.

## Steps:

## Customer logs into the webpage.

## Customer login account.

## Customers click the pay button on the order page

## Payment is received.

## Use Case 6:

## Identifier: UC06

## Name: Notifications

## Description: This use case is for notifying the staff members.

## Trigger: When the order is given by a customer, the cooking staff are notified. After the order is ready for delivery, the waiter-waitress staff are notified.

## Preconditions: The customer should place an order.

## Postconditions: Cooking staff members are notified after an order is given and waiters are notified when the order is ready.

## Steps:

## 1- Customer gives an order

## 2- Cooking staff member is notified

## 3- Cooking staff states that the order is ready

## 4- Waiter-waitress is notified that order is ready for delivery.

## Use Case 8:

## Identifier: UC08

## Name: Create Account

## Description: This use case allows customers to create an account on the website.

## Trigger: Customer clicks the sign in button on the home page.

## Preconditions: User log onto webpage.

## Postconditions: User create own account with enter his/her information.

## Steps:

## Customer logs into the webpage.

## Customer click sign in button.

## Customers enter information.

## 4- Customer’s account has been created.

## 

## Use Case 9:

## Identifier: UC09

## Name: Order History

## Description: This use case allows customers to see their past orders.

## Trigger: Customer clicks my past orders button on the home page.

## Preconditions: User log into own account.

## Postconditions: System shows the list of customer’s past order.

## Steps:

## Customer log into own account.

## Customers click my past orders in button.

## System shows the list of customer’s past order.

## Use Case 10:

## Identifier: UC10

## Name: Update Information

## Description: This use case allows administrators to confirm the customer’s updating information.

## Trigger: Customer click the change information button in settings page.

## Preconditions: Customer is logged into own account.

## Postconditions: Information is updated with newly entered information.

## Steps:

## The customer is logged into own account.

## Customers click the change information button.

## Customers add new information.

## Administrators confirm the changes.

## Information are updated with newly entered information.

## User Stories:

## -As a cook I want to notify when I prepare cook.as a user I want to give feedback about my orders.

## -As a user I want to see my order history.

## -As a user I want to add products that I choose to the cart.

## -As a waiter I want to be notified when the customer gives an order.

## -As a waiter I want to be informed when the order is ready.

## -As an administrator, I want to be able to update information in databases from Inventory Management System.

**C. ACRONYMS AND ABBREVIATIONS**

## - DBMS: Database Management System

## - UI: User Interface

## - GUI: Graphical User Interface

## - IMS: Inventory Management System

## - CRM: Customer Relationship Management

## - MVC: Modal View Controller

**D. REFERENCES**

## - Software Requirements Engineering course lecture slides

## -https://www.tutorialspoint.com/software\_testing\_dictionary/software\_requirement\_specification.htm

## - https://www.lucidchart.com/pages/uml-use-case-diagram

## - https://www.oreilly.com/library/view/http-the-definitive/1565925092/ch04s01.html

## - draw.io