Software Design Document

For

Online Food Ordering And Delivery Platform

Submitted By:

- Abdelrahim Mohamed Alsadiq
- Ahmed Adel Abo Elwafa
- Ahmed Abdelrahman Shams

Sara Abdelrahim Hamed

Manar Montaser Hasan

Jena Hana Yousef

version 1.5.3| 17/5/2023

Presented To: Dr. Mohamed Ramadan

Table of Contents 1

1. System Architecture and Architecture Design 2

1.1 Logical View 3

1.2 Hardware Architecture 3

1.3 Software Architecture 4

1.4 Information Architecture 5

1.5 Internal Communications Architecture 6

2. System Design 7

2.1 Business Requirements 7

2.2 Database Design 8

2.3 Data Objects and Resultant Data Structures 9

2.4 File and Database Structures 11

3. User Interface Design 13

1. System Architecture and Architecture Design:

The system architecture of our project typically involves multiple components and subsystems working together to provide the desired functionality. Such as:

1. User Interface (UI) Component:
    a. **Responsibilities**: Handling user interactions, presenting menus, managing user profiles, and facilitating order placement.
    b. **Rationale**: The UI component is responsible for providing a user-friendly interface that allows customers to browse menus, customize orders, and place orders seamlessly. It may utilize design patterns such as Model-View-Controller (MVC) to separate concerns and ensure a modular structure.
2. Order Management Component:
    a. **Responsibilities**: Managing the order lifecycle.
    b. **Rationale**: The Order Management component is responsible for handling the end-to-end order management process, ensuring accurate updates for customers, restaurants, and delivery personnel.
3. Restaurant Management Component:
    a. **Responsibilities**: Facilitating restaurant registration, menu management, and order fulfillment.
    b. **Rationale**: The Restaurant Management component handles the registration of restaurants onto the platform, allows them to manage menus, update item availability, and process incoming orders. It may employ the Strategy pattern to support different restaurant management approaches and the Factory pattern for creating menu items.
4. Delivery Management Component:
    a. **Responsibilities**: Managing delivery personnel, assigning orders, and tracking deliveries.
    b. **Rationale**: The Delivery Management component handles the management of delivery personnel, including their availability, assigning orders to them based on location and workload, and tracking deliveries.

5. Payment Processing Component:
    a. **Responsibilities**: Facilitating secure payment transactions and integrating with payment gateways.
    b. **Rationale**: The Payment Processing component handles secure payment transactions, encrypts sensitive data, and integrates with external payment gateways.
6. External Service Integration Component:
    a. **Responsibilities**: Integrating with external services such as mapping services for delivery tracking and SMS services for notifications.
    b. **Rationale**: The External Service Integration component enables seamless integration with external services necessary for order tracking, address verification, and notifications. It may employ the Adapter pattern to provide a standardized interface for interacting with different external services.

2.1 Logical View

- **Restaurant Listings View**: This view presents a list of available restaurants or food establishments to the users. It typically includes information such as restaurant name, cuisine type, ratings, reviews, and delivery options.
- **Menu View**: This view displays the menu items offered by a specific restaurant. It includes details like item name, description, price, and customizable options if applicable.
- **Cart View**: This view represents the user's shopping cart, where they can add, remove, or modify items before placing an order. It keeps track of the selected items, quantities, and associated costs.

2.2 Hardware Architecture:

- Presentation Servers: We employ multiple presentation servers responsible for handling user interfaces and delivering web pages to our customers. These servers are distributed across different geographical locations to ensure low latency and quick response times.

- Application Servers: The application servers manage the core logic and functionality of our food ordering and delivery system. They process user requests, handle business logic, and communicate with databases and external services. We have a cluster of application servers distributed across multiple data centers to ensure high availability and scalability.
- Data Servers: Our data servers store and manage various data components such as user profiles, menu items, orders, and delivery details. We employ a distributed database system that replicates data across multiple servers for redundancy and fault tolerance. These data servers are located in secure data centers and are accessed by application servers to retrieve and store information.
- Peripheral Devices: To facilitate smooth operations and enhance user experience, we employ various peripheral devices. These include:
  a. Point-of-Sale (POS) terminals: Used by restaurant partners to receive and process orders.
  b. Mobile devices: Delivery personnel use smartphones or tablets to receive order information and navigate to the delivery locations.

2.3 Software Architecture:

- Presentation Layer:
  - User Interface (UI): The UI components handle the presentation and interaction with users. Implemented using HTML, CSS, JavaScript
  - Front-End Frameworks: These frameworks, such as React, facilitate the development of dynamic and responsive user interfaces.
  - Web Servers: Software like Apache or Nginx can be used to host and serve the web pages to users.
- Application Layer:
  - Application Logic: This layer includes the core business logic of the food ordering and delivery system. It handles tasks such as order processing, menu management, and user authentication.

- **Communication and Integration:**
    - APIs: Application Programming Interfaces are used to enable communication and integration with external services and systems. Examples include payment gateways, SMS gateways, or third-party delivery platforms.
- **Infrastructure and Utilities:**
    - Operating System: The system can be deployed on an operating system like Linux, Windows, or macOS, depending on the server infrastructure.
    - Containerization: Technologies such as Docker or Kubernetes can be used to package and deploy the system components as containers.
    - Source Control: Version control systems like Git help manage the source code and facilitate collaboration among developers.
    - Integrated Development Environments (IDEs): Tools like IntelliJ IDEA, Eclipse, or Visual Studio Code aid developers in writing, debugging, and testing the system's code.

2.4 Information Architecture:

In our food ordering and delivery website, the system will store various types of information to facilitate the ordering and delivery process. The specific data stored can include:

- User Information: This includes personally identifiable information (PII) such as names, email addresses, phone numbers, and delivery addresses. PII refers to any information that can be used to identify an individual.
- Order Details: Information related to customer orders will be stored, including items ordered, quantities, prices, special instructions, and order timestamps.
- Payment Information: We may store payment details such as credit card numbers, billing addresses, and transaction history. It's important to note that we follow strict security standards and industry best practices to ensure the safety of sensitive payment information.
- Restaurant and Menu Data: Data related to partner restaurants, their menus, menu items, descriptions, prices, and availability will be stored. This information helps in presenting accurate and up-to-date options to users.

2.5 Internal Communications Architecture:

In our food ordering and delivery website, the system relies on a robust communications network to ensure seamless connectivity and data transfer between system components. The communications network incorporates a combination of local area networks (LANs) and wide area networks (WANs) to facilitate communication at various levels.

2. System Design

2.1 Business Requirements:

- User Registration and Profiles:
  - The system should allow users to register and create personalized profiles with their contact information, delivery addresses, and preferences.
- Menu and Restaurant Management:
  - The website should provide an intuitive interface for restaurants to manage their menus, including adding, updating, and removing menu items, as well as specifying prices, descriptions, and availability.
- Ordering and Checkout Process:
  - Users should be able to browse menus, select items, customize orders, and add them to a shopping cart. The system should support secure online payments and provide confirmation notifications upon successful order placement.
- Delivery and Tracking:
  - The website should integrate with a delivery management system to assign orders to delivery personnel, track order status, and provide estimated delivery times to users.
- Reviews and Ratings:
  - Users should have the ability to rate and provide feedback on restaurants and delivery experiences, helping other users make informed decisions.
- Customer Support:
  - The system should provide channels for users to contact customer support, such as live chat, email, or phone, to address inquiries, issues, or complaints.

2.2 Database Design:

User Table:

- Data elements: UserID (integer), Name (string), Email (string), Password (string), Address (string), Phone (string)
- Maintenance: CRUD operations on user data
- Data stores: User information storage
- Outputs: User profile information

Restaurant Table:

- Data elements: RestaurantID (integer), Name (string), Address (string), Phone (string), Rating (float)
- Maintenance: CRUD operations on restaurant data
- Data stores: Restaurant information storage
- Outputs: Restaurant details and ratings

Menu Table:

- Data elements: MenuID (integer), RestaurantID (integer), ItemName (string), Description (string), Price (float), Category (string)
- Maintenance: CRUD operations on menu items
- Data stores: Menu item information storage
- Outputs: Menu details for a particular restaurant

Order Table:

- Data elements: OrderID (integer), UserID (integer), RestaurantID (integer), DeliveryAddress (string), TotalAmount (float), Status (string)
- Maintenance: CRUD operations on orders
- Data stores: Order information storage
- Outputs: Order details and status

2.3 Data Objects and Resultant Data Structures:

User Data Structure:

- Data Structure: User
  - Attributes: UserID, Name, Email, Password, Address, Phone
  - Restaurant Data Structure:
- Data Structure: Restaurant
  - Attributes: RestaurantID, Name, Address, Phone, Rating

Menu Item Data Structure:

- Data Structure: MenuItem
- Attributes: MenuID, RestaurantID, ItemName, Description, Price, Category

Order Data Structure:

- Data Structure: Order
- Attributes: OrderID, UserID, RestaurantID, DeliveryAddress, TotalAmount, Status

These data structures are the major components that store and process the core data in the system. They interact with various functions and are passed between components to perform operations. Here is a high-level description of some of the functions and parameters involved:

Function: User Registration

- Input: Name, Email, Password, Address, Phone
- Output: UserID

Function: Browse Restaurants

- Input: None
- Output: List of Restaurant data structures

Function: Browse Menu

- Input: RestaurantID
- Output: List of MenuItem data structures

Function: Add to Cart

- Input: UserID, MenuItem
- Output: None

Function: Place Order

- Input: UserID, List of MenuItem, DeliveryAddress
- Output: OrderID

Function: Track Order

- Input: OrderID
- Output: Order data structure

Function: Rate Restaurant

- Input: RestaurantID, Rating
- Output: None

2.4 File and Database Structures:

User Data Storage:

- File Structure: User table
- Location: Database system
- DBMS Considerations: The User table could be stored as a relational table with columns representing the user attributes (e.g., UserID, Name, Email, Password, Address, Phone). Appropriate indexing and data normalization can be applied based on specific requirements and performance considerations.

Restaurant Data Storage:

- File Structure: Restaurant table
- Location: Database system
- DBMS Considerations: Similar to the User table, the Restaurant table could be stored as a relational table with columns representing the restaurant attributes (e.g., RestaurantID, Name, Address, Phone, Rating). Again, indexing and normalization techniques can be applied as necessary.

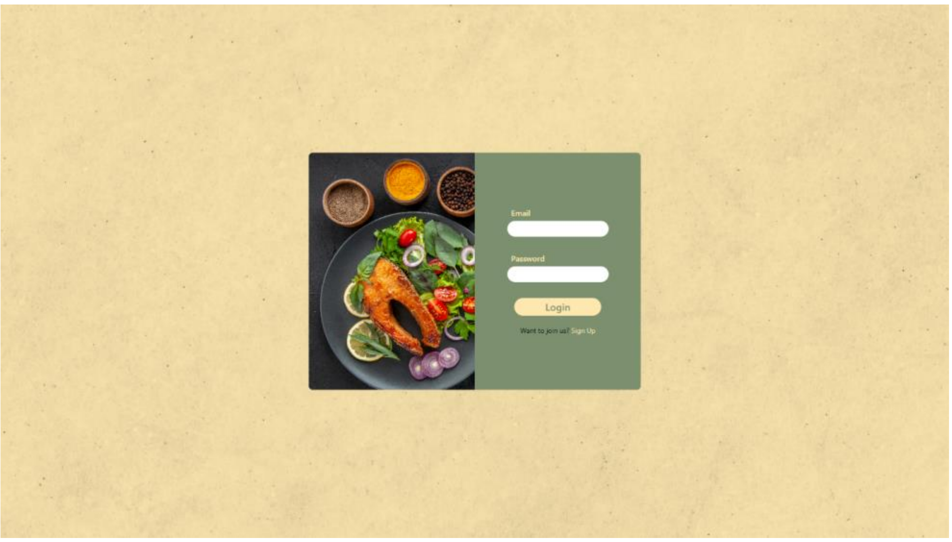
Menu Item Data Storage:

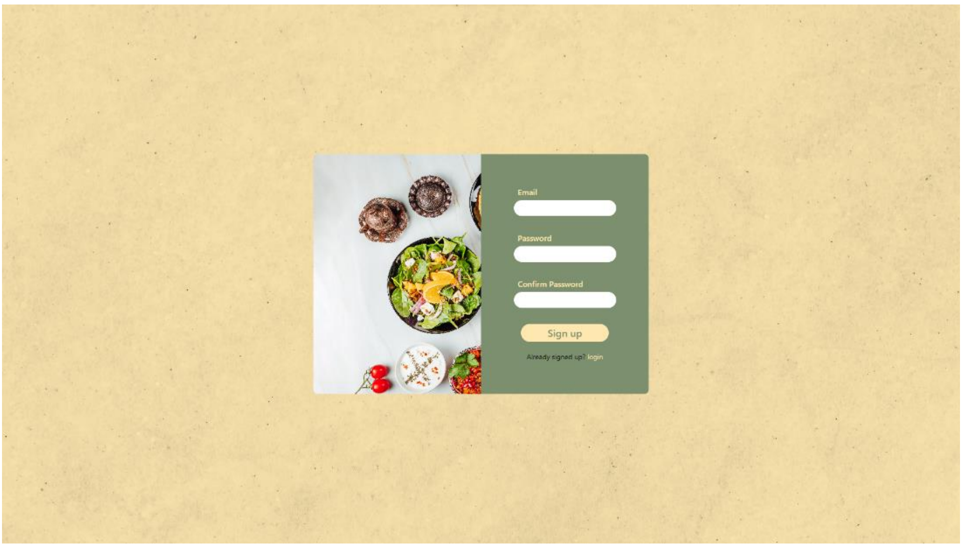
- File Structure: Menu Item table
- Location: Database system
- DBMS Considerations: The Menu Item table could be stored as a relational table with columns representing the menu item attributes (e.g., MenuID, RestaurantID, ItemName, Description, Price, Category). Depending on the requirements, additional tables or relationships might be necessary to handle menu item customization options or ingredient lists.

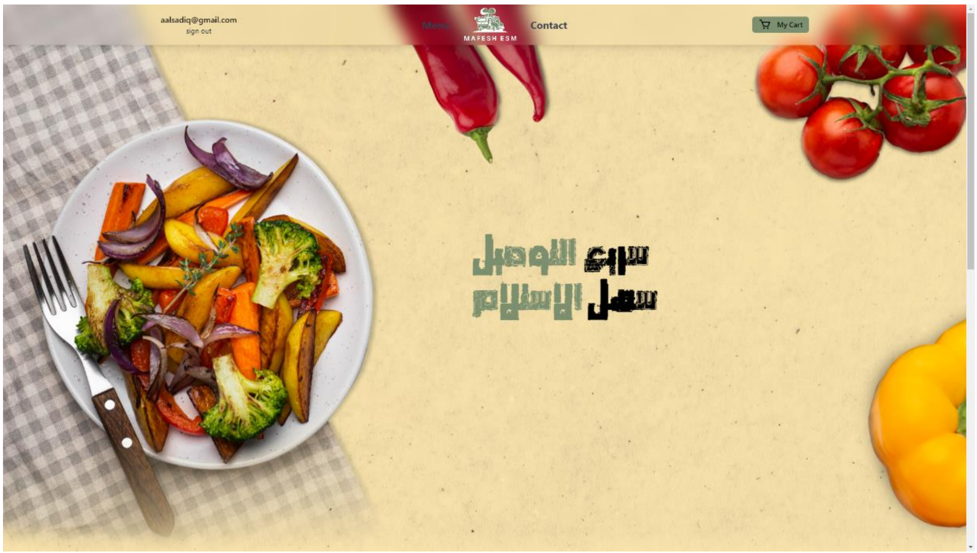
Order Data Storage:

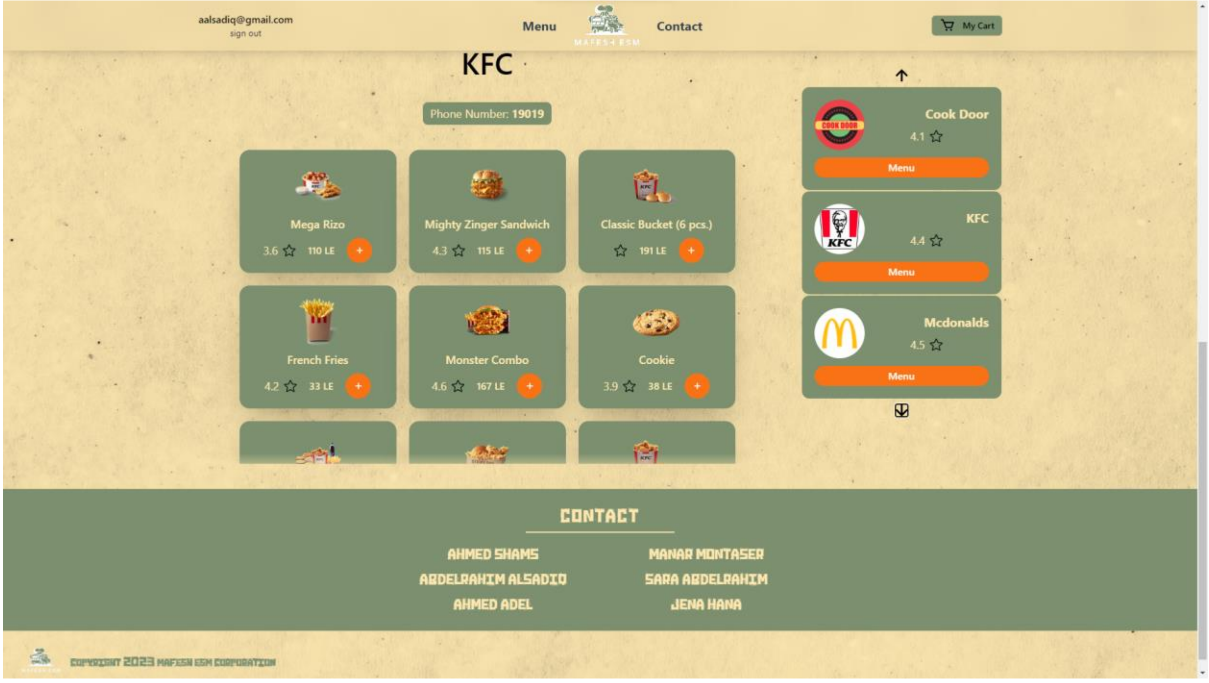
- File Structure: Order table
- Location: Database system
- DBMS Considerations: The Order table could be stored as a relational table with columns representing the order attributes (e.g., OrderID, UserID, RestaurantID, DeliveryAddress, TotalAmount, Status). Additional tables or relationships might be required to handle order details, such as item quantities or special instructions.

3. User Interface Design:









KFC

Phone Number: **19019**

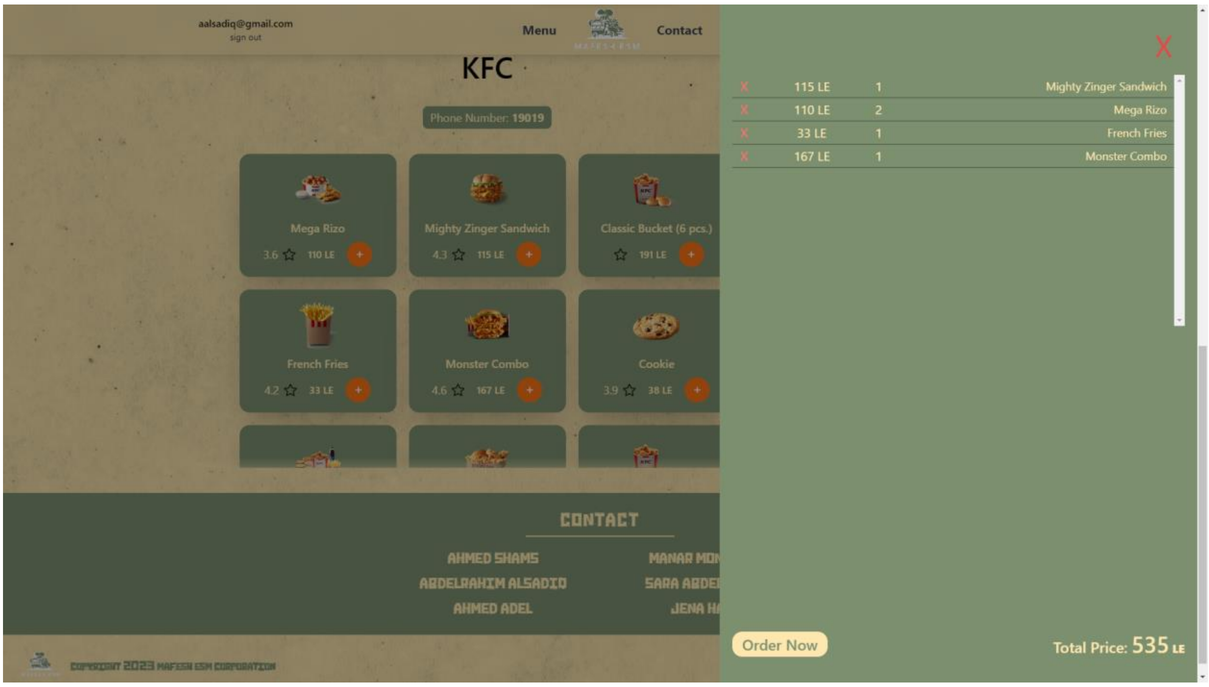
Mega Rizo 3.6 ☆ 110 LE +	Mighty Zinger Sandwich 4.3 ☆ 115 LE +	Classic Bucket (6 pcs.) ☆ 191 LE +
French Fries 4.2 ☆ 33 LE +	Monster Combo 4.6 ☆ 167 LE +	Cookie 3.9 ☆ 38 LE +

Cook Door	4.1 ☆
KFC	4.4 ☆
Mcdonalds	4.5 ☆

CONTACT

AHMED SHAMS
ABDELRAHIM ALSADIQ
AHMED ADEL

MANAR MONTASER
SARA ABDELRAHIM
JENA HANA



X	115 LE	1	Mighty Zinger Sandwich
X	110 LE	2	Mega Rizo
X	33 LE	1	French Fries
X	167 LE	1	Monster Combo

Total Price: 535 LE