|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Online Food Delivery System**  **High Level Design Document**   |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed by** | **Approved By** | | **Name** | 1.Devi Prasad Dash  2.Ashutosh Mishra  3.Kanha Sahu  4.Piyush Das | <Trainer/Project Governance Representative/Cohort Mentor to fill> | <Trainer/Project Governance Representative/Cohort Mentor to fill> | | **Role** | Programmer Analyst Trainee |  |  | | **Signature** |  |  |  | | **Date** | 05-08-2025 |  |  | |
|  |

Table of Contents

1. [Introduction](https://www.perplexity.ai/search/do-you-remeber-prvious-chat-QYwBRessSdaCoegeuWITvA?login-source=requestAccessLoginGate&login-new=false#1-introduction)
2. [Project Overview](https://www.perplexity.ai/search/do-you-remeber-prvious-chat-QYwBRessSdaCoegeuWITvA?login-source=requestAccessLoginGate&login-new=false#2-project-overview)
3. [Solution Summary](https://www.perplexity.ai/search/do-you-remeber-prvious-chat-QYwBRessSdaCoegeuWITvA?login-source=requestAccessLoginGate&login-new=false#3-solution-summary)
4. [Schematic Diagram](https://www.perplexity.ai/search/do-you-remeber-prvious-chat-QYwBRessSdaCoegeuWITvA?login-source=requestAccessLoginGate&login-new=false#1-introduction)
5. [System Architecture](https://www.perplexity.ai/search/do-you-remeber-prvious-chat-QYwBRessSdaCoegeuWITvA?login-source=requestAccessLoginGate&login-new=false#6-security-design)
6. [Security Design](https://www.perplexity.ai/search/do-you-remeber-prvious-chat-QYwBRessSdaCoegeuWITvA?login-source=requestAccessLoginGate&login-new=false#6-security-design)
7. [Database Design](https://www.perplexity.ai/search/do-you-remeber-prvious-chat-QYwBRessSdaCoegeuWITvA?login-source=requestAccessLoginGate&login-new=false#5-database-design)

# Introduction

## Purpose of this document

The Restaurant & Customer Authentication System is a modern web-based platform that enables secure authentication and role-based access control for restaurants and customers. The system facilitates seamless user management, profile handling, and secure communication between different user types through a microservices architecture.

# Project Overview

Build a secure, scalable authentication system supporting multiple user roles (Restaurant Owners, Customers, Administrators) with JWT-based authentication and comprehensive session management.

**Key Features:**

* Multi-tenant authentication system
* Role-based access control (RBAC)
* JWT token management with refresh capabilities
* Real-time session monitoring
* Secure password handling with BCrypt
* Cart preservation and order tracking
* RESTful API architecture

# Solution Summary

## Scope

* **Frontend**: Angular SPA with reactive forms and component-based architecture
* **Backend:** Spring Boot microservices with JWT authentication
* **Database:** MySQL for persistent data storage
* **Security:** JWT tokens, BCrypt password hashing, CORS configuration
* **Inter-service Communication:** OpenFeign clients with Eureka service discovery

## Assumptions

* Users access the system via web browsers only
* Scheduled database backups are maintained by database administrators
* HTTPS is enforced in production environments
* Session timeout is configurable per environment

## Dependencies

### Hardware Requirement:

* **Frontend:** Angular 19+, RxJS, TypeScript
* **Backend:** Spring Boot, JWT library
* **Database:** MySQL 8.0+
* **Infrastructure:** Eureka Server

## Risks

* Mishandled or duplicate entries of the same entity.
* Loss of data if scheduled backup is not done from time to time.
* Data privacy issues.

# Schematic Diagram

A schematic, or schematic diagram, is a representation of the elements of a [system](https://en.wikipedia.org/wiki/System) using abstract, graphic [symbols](https://en.wikipedia.org/wiki/Symbol) rather than realistic pictures. It gives an overview of overall system

A diagram of a server

AI-generated content may be incorrect.

# System Architecture

## Proposed design

**Frontend Components**

* AuthService: Handles authentication state, token management, and user sessions
* Login Components: Restaurant and Customer login forms with validation
* Guards: Route protection based on user roles
* Interceptors: Automatic JWT token attachment to HTTP requests

**Backend Services**

* Auth Service: JWT generation, validation, and refresh token management
* Restaurant Service: Restaurant profile management and internal user lookup
* Customer Service: Customer profile and order management
* Eureka Server: Service discovery and registration

# SecurityDesign

## 6.1 JWT Token Structure

{

"sub": "user@example.com",

"userId": 123,

"email": "user@example.com",

"userType": "RESTAURANT",

"roles": ["RESTAURANT"],

"iat": 1691234567,

"exp": 1691238167,

"iss": "ofds-auth-service"

}

## 6.2 Security Features

* **Password Security:** BCrypt hashing with salt
* **Token Security:** HS256 algorithm with secret key
* **Session Management:** 30-minute idle timeout
* **Role-Based Access:** Granular permissions
* **CORS Configuration:** Restricted origins

# 7.0Database Design

## 7.1Data Model

A screenshot of a computer screen

AI-generated content may be incorrect.

## 

## 7.2Tables Structure

*1. Restaurant*

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | ***Constraints*** | *Description* |
| *Id* | *BigInt* | *Primary Key, Auto Increment* | *Unique identifier for the restaurant* |
| name | VARCHAR | Not Null | Name of the restaurant |
| Location | VARCHAR | Not Null | Physical location of the restaurant |
| Email | VARCHAR | |  |  | | --- | --- | |  | Not Null,Unique | |  |  | | |  | | --- | | Email address used for login/communication | |  | |
| password | VARCHAR | Not Null | Encrypted password for authentication |

*2.Customer*

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | ***Constraints*** | ***Description*** |
| *ID* | *BigInt* | *Primary Key,Auto\_Increment* | *Unique identifier for the customer* |
| name | VARCHAR | — | Full name of the customer |
| email | VARCHAR | Not Null, Unique | Email address used for |
| password | VARCHAR | — | Encrypted password for authentication |
| phone | BIGINT | — | Contact phone number |
| address | VARCHAR | — | Physical mailing address |

*3. Order*

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| orderId | BIGINT | Primary Key, Auto Increment | Unique identifier for the order |
| userId | BIGINT | Not Null | ID of the customer placing the order |
| restaurantId | BIGINT | Not Null | ID of the restaurant fulfilling the order |
| status | ENUM | Not Null | Current status of the order (e.g., PENDING, DELIVERED) |
| totalAmount | DOUBLE | Not Null | Total monetary value of the order |
| orderTime | TIMESTAMP | Not Null | Date and time when the order was placed |
| deliveryTime | TIMESTAMP | — | Actual delivery time (if available) |
| deliveryAddress | VARCHAR | Not Null | Address where the order should be delivered |
| paymentId | BIGINT | — | Reference to associated payment (optional) |
| deliveryAgentId | BIGINT | — | ID of assigned delivery agent (optional) |
| idempotencyKey | VARCHAR | Unique | Unique key for preventing duplicate processing |
| deliveryId | BIGINT | Unique | Unique identifier for delivery record (if any) |

*4.Menu*

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
| itemId | BIGINT | Primary Key, Auto Increment | Unique identifier for each menu item |
| restaurant\_id | BIGINT | Not Null | ID of the restaurant that owns this item |
| item\_name | VARCHAR(100) | Not Null | Name of the menu item |
| description | VARCHAR(500) | Not Null | Description of the dish |
| isVegetarian | BOOLEAN | Not Null | Indicates if the item is vegetarian |
| price | DOUBLE | Not Null | Cost of the menu item |
| created\_at | TIMESTAMP | Default: LocalDateTime.now() | Timestamp when item was added (not updatable) |
| created\_by | VARCHAR | — | User or system who created the entry |
| updated\_at | TIMESTAMP | Default: LocalDateTime.now() | Last update time of the menu item |
| updated\_by | VARCHAR | — | User or system who last updated the item |

*5.Payment Module*

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
| idPayment | BIGINT | Primary Key, Auto Increment | Unique identifier for the payment |
| orderId | BIGINT | — | Reference to the associated order |
| paymentMethod | ENUM | — | Method used for payment (e.g., CARD, UPI, COD) |
| paymentAmount | DECIMAL | — | Total amount paid |
| paymentStatus | ENUM | — | Status of payment (e.g., COMPLETED, FAILED) |
| createdBy | VARCHAR | — | User/system that initiated the payment |
| createdOn | TIMESTAMP | Default: now() via @PrePersist | When the payment was recorded |
| updatedBy | VARCHAR | — | User/system who last updated the payment |
| updatedOn | TIMESTAMP | Default: now() via @PreUpdate | When the payment record was last updated |

*6.Delivery*

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
| delivery\_id | BIGINT | Primary Key, Auto Increment | Unique identifier for the delivery |
| order\_id | BIGINT | Not Null, Unique | ID of the related order |
| agent\_id | BIGINT | Foreign Key to AgentEntity | ID of the delivery agent assigned |
| status | ENUM | Not Null | Current delivery status (e.g., IN\_PROGRESS, DELIVERED) |
| estimated\_time\_of\_arrival | TIMESTAMP | Not Null | Expected time the delivery should reach the destination |
| created\_by | VARCHAR | Not Null, Not Updatable | Identifier of who created the record |
| created\_on | TIMESTAMP | Not Null, Not Updatable | Timestamp of record creation (auto-managed) |
| updated\_by | VARCHAR | — | Identifier of last modifier |
| updated\_on | TIMESTAMP | — | Timestamp of last update (auto-managed) |

# Appendices

## Glossary

|  |  |
| --- | --- |
| **Acronyms** | **Definitions** |
| Online Food Delivery System | A digital platform that enables customers to order food from restaurants via web and have it delivered. |

# Terms & Conditions

***Disclaimer: Please do not circulate or distribute this document outside of Cognizant Network,*** ***We have a Zero Tolerance Policy. Kindly adhere to 100% Compliance at all times.***

|  |
| --- |
|  |