**Project Title: Pizza Management System**

**Introduction:**

The Pizza Management System is designed to simplify and automate pizza ordering and management process. The system aims to provide a user-friendly interface for customers to place pizza orders, customize the order and for restaurant staff to efficiently manage orders.

**Technologies Used:**

* Core Java
* Spring Boot + Rest Controller + JPA with Hibernate
* Angular for UI design (Front End)
* Postgres (or any other preferred relational database management system)
* Maven (for project management)

**System Features:**

* User Registration and Authentication:
* The system allows users to register and create accounts with secure authentication mechanisms.
* Pizza Management:
* The system provides features to manage Pizza details including pizza name, pizza size, topping and associated attributes. Restaurant administrators can efficiently update and maintain pizza details.
* Customers can browse the menu, select pizza types, sizes, and toppings, customize their orders, and place them seamlessly. The system calculates the order total.
* Order Management:
* The application provides a dashboard for restaurant staff to manage and track pizza orders. Staff can view incoming orders, update order status and mark orders as complete.
* Users can view, manage their orders, can see the order history.
* Reporting Management:
* The system provides various reports related to sales, revenue, customer preferences, and other indicators.
* Admin Panel:
* Administrators have access to an admin panel for managing users, pizza, order etc.

**Testing:**

* Unit testing is performed to ensure the correctness of business logic and functionality.

**Project Directory structure:**

* The src folder contains the source code for the project.
* The main folder contains the main codebase of the application.
* The java folder is where you place your Java source code.
* The <com>.<projectname>.<packagename> is a suggested package structure.
* The controllers package contains the controllers that handle the HTTP requests.
* The entity package contains the entity classes that represent the database tables, such as Pizza, Toppings, and PizzaOrder etc.
* The repositories package contains the repository interfaces for database operations, such as PizzaRepository, PizzaOrderRepository, and CustomerRepository etc.
* The services package contains the service classes that implement the business logic, such as PizzaService, PizzaOrderService, and CustomerService etc.
* The security package contains the security configuration for authentication and authorization, such as SecurityConfig.
* The resources folder contains non-Java resources such as properties files.
* The application.properties file holds the configuration for the main application.
* The test folder contains the test code for unit testing.
* The application-test.properties file holds the configuration for testing.

Remember, this is just a suggested directory structure, and you can customize it based on your project requirements. It's important to follow proper package and class naming conventions and organize your code in a way that is maintainable and scalable.