**System Level Design Document – Distributed Order Management Service- CORBA**

**Eranda Sooriyarachchi**

**Overview:**

CORBA (Common Object Request Broker Architecture) is a middleware technology that enables distributed computing across heterogeneous systems. It is a standard developed by the Object Management Group (OMG) and defines a set of specifications for creating, accessing, and managing distributed objects.

The Distributed Order Management Service is a user-friendly and reliable client-server application that facilitates real-time placement and monitoring of food orders for consumers. It is built on the foundation of the Common Object Request Broker Architecture (CORBA), enabling seamless communication between the client and server components.

Architecture:

The architecture of the Distributed Order Management System follows the client-server model, where the client initiates requests and the server processes and responds with results. The system leverages the CORBA architecture to establish a connection between the client and server components.

The server is responsible for managing the business logic of the system, including maintaining a database of food products and their pricing, managing client orders, generating invoices, and providing real-time updates on order progress.

On the other hand, the client component provides an intuitive user interface to customers, allowing them to browse the menu, place orders, and monitor the status of their orders. The communication between the client and server takes place through remote procedure calls (RPCs), facilitated by the CORBA framework.

The Distributed Order Management Service is composed of several components, including:

1. Client: This is a user interface that allows consumers to interact with the system, browse the menu, place orders, and track order progress.

2. Server: The server component is responsible for managing the system's business logic, which includes maintaining a database of food products and pricing, managing client orders, generating invoices, and providing real-time updates on order progress. It is developed using Java and uses the CORBA framework to communicate with the client.

3. CORBA: Middleware technology that facilitates communication between the client and server components by providing APIs that enable the client to call server methods and receive real-time responses.

4. Naming Service: Created using the Name Service offered by the CORBA architecture, the naming service maps server object names to their respective object references.

The Distributed Order Management Service offers the following functionalities to its users:

For the User Role: Customer:

• View Menu: Customers can view the menu of the restaurant, which includes the prices of food items.

• Place Order: Customers can place orders for the food items they wish to purchase. They can specify the quantity of each food item they want to order.

• Check Order Status: Customers can track their orders by providing their names. The system displays the total amount to be paid along with order details.

For the User Role: Manager:

• View Current Orders: The system provides a list of all the orders that have been placed in the current session.

Efficiency Measure:

My code uses hashmaps which are O(1). Therefore it’s faster than searching through an array. There are no nested loops that would bring about O(n^2).