Case Study 1: Java-Based Configuration

Project Title: Online Food Ordering System

Configuration Type: Java-based Spring

Configuration POJO Classes: Restaurant and Customer

Scenario:

An online food ordering platform allows customers to order food from various restaurants. The system must manage customer information and restaurant offerings. The logic for selecting restaurants and placing orders is handled in a service class. Java-based configuration is used to wire beans explicitly

Components:

• Customer.java: Holds customer details like name, contact info, and preferred cuisine

. • Restaurant.java: Holds restaurant details like name, location, and available cuisines.

• FoodOrderService.java: Service that processes the food order by matching customer preferences with restaurant availability.

• AppConfig.java: A @Configuration class that defines and wires all beans manually using @Bean methods.

• MainApp.java: Initializes the Spring context using AnnotationConfigApplicationContext and executes the order flow

Why Java-Based Config?

• Useful when full control over bean creation is required.

• Suitable for projects where configuration is centralized and separated from the POJO classes (which may not be editable).

Customer.java

Code:

```java
package com.fooder;
public class Customer {
    private String name;
    private String contactInfo;
    private String preferredCuisine;
    public Customer(String name, String contactInfo, String preferredCuisine) {
        this.name = name;
        this.contactInfo = contactInfo;
        this.preferredCuisine = preferredCuisine;
    }
    public String getName() {
        return name;
    }
    public String getContactInfo() {
        return contactInfo;
    }
    public String getPreferredCuisine() {
        return preferredCuisine;
    }
}
```

Restaurant.java

Code:

```java
package com.fooder;
import java.util.List;

public class Restaurant {
    private String name;
    private String location;
    private List<String> availableCuisines;
    public Restaurant(String name, String location, List<String> availableCuisines) {
        this.name = name;
        this.location = location;
        this.availableCuisines = availableCuisines;
    }
    public String getName() {
        return name;
    }
    public String getLocation() {
        return location;
    }
    public List<String> getAvailableCuisines() {
        return availableCuisines;
    }
}
```

FoodOrderService.java

Code:

```
package com.fooder;
public class FoodOrderService {
    private Customer customer;
    private Restaurant restaurant;
    public FoodOrderService(Customer customer, Restaurant restaurant) {
        this.customer = customer;
        this.restaurant = restaurant;
    }

    public void placeOrder() {
        if (restaurant.getAvailableCuisines().contains(customer.getPreferredCuisine())) {
            System.out.println("Order placed successfully!");
            System.out.println("Customer: " + customer.getName());
            System.out.println("Restaurant: " + restaurant.getName());
            System.out.println("Cuisine: " + customer.getPreferredCuisine());
        } else {
            System.out.println("Sorry, your preferred cuisine is not available at this restaurant.");
        }
    }
}
```

AppConfig.java

Code:

```
package com.fooder;
import java.util.Arrays;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
@Configuration
public class AppConfig {
    @Bean
    public Customer customer() {
        return new Customer("Sravani", "sai@example.com", "South Indian");
    }
    @Bean
    public Restaurant restaurant() {
        return new Restaurant("Spicy House", "Hyderabad", Arrays.asList("South Indian",
"North Indian", "Chinese"));
    }

    @Bean
    public FoodOrderService foodOrderService() {
        return new FoodOrderService(customer(), restaurant());
    }
}
```

MainApp.java

Code:

```
package com.fooder;

import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class MainApp {
    public static void main(String[] args) {
        ApplicationContext context = new
AnnotationConfigApplicationContext(AppConfig.class);
        FoodOrderService service = context.getBean(FoodOrderService.class);
        service.placeOrder();
    }
}
```

Output:

```
Order placed successfully!
Customer: Sravani
Restaurant: Spicy House
Cuisine: South Indian
```