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.

Solution:

MainApp.java

```
package com.foodorder;


import com.foodorder.config.AppConfig;

import com.foodorder.model.Customer;

import com.foodorder.service.FoodOrderService;

import org.springframework.context.annotation.*;
```

```java
public class MainApp {
    public static void main(String[] args) {
        // Initialize Spring context
        AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

        // Retrieve beans
        Customer customer = context.getBean(Customer.class);
        FoodOrderService foodOrderService = context.getBean(FoodOrderService.class);

        // Execute order flow
        System.out.println("Customer Details: " + customer);
        String orderResult = foodOrderService.placeOrder(customer);
        System.out.println(orderResult);

        // Close context
        context.close();
    }
}
```

AppConfig.java

```java
package com.foodorder.config;

import com.foodorder.model.Customer;
import com.foodorder.model.Restaurant;
import com.foodorder.service.FoodOrderService;

import java.util.Arrays;
import java.util.List;
```

```
}

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class AppConfig {

    @Bean
    public Customer customer() {
        return new Customer("John Doe", "john.doe@example.com", "Italian");
    }

    @Bean
    public Restaurant restaurant1() {
        return new Restaurant("Pasta Palace", "Downtown", Arrays.asList("Italian",
"Mediterranean"));
    }

    @Bean
    public Restaurant restaurant2() {
        return new Restaurant("Spice Haven", "Uptown", Arrays.asList("Indian", "Chinese"));
    }

    @Bean
    public FoodOrderService foodOrderService() {
        List<Restaurant> restaurants = Arrays.asList(restaurant1(), restaurant2());
        return new FoodOrderService(restaurants);
    }
```

```
}
```

Customer.java

```
package com.foodorder.model;

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 void setName(String name) {
        this.name = name;
    }

    public String getContactInfo() {
        return contactInfo;
    }

    public void setContactInfo(String contactInfo) {
        this.contactInfo = contactInfo;
    }

    public String getPreferredCuisine() {
        return preferredCuisine;
    }

    public void setPreferredCuisine(String preferredCuisine) {
        this.preferredCuisine = preferredCuisine;
    }

    @Override
    public String toString() {
        return "Customer{name='" + name + "', contactInfo='" + contactInfo + "', preferredCuisine='" + preferredCuisine + "'}";
    }
}
```

Restaurant.java

```
package com.foodorder.model;

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 void setName(String name) {
    this.name = name;
}

public String getLocation() {
    return location;
}

public void setLocation(String location) {
    this.location = location;
}

public List<String> getAvailableCuisines() {
    return availableCuisines;
}

public void setAvailableCuisines(List<String> availableCuisines) {
    this.availableCuisines = availableCuisines;
}
```

```
    @Override
    public String toString() {
        return "Restaurant{name='" + name + "', location='" + location + "', availableCuisines=" + availableCuisines + "}";
    }
}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.foodorder</groupId>
    <artifactId>food-ordering-system</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>11</maven.compiler.source>
        <maven.compiler.target>11</maven.compiler.target>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>6.1.10</version>
        </dependency>
    </dependencies>
```

```
</project>
```

Case Study 2: Annotation-Based Configuration

Project Title: Smart Home Automation System

Configuration Type: Annotation-based Spring Configuration

POJO Classes: Device and User

Annotations based

AppConfig.java

```
package com.example.smarthome;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

@Configuration

@ComponentScan("com.example.smarthome")

public class AppConfig {

}
```

Device.java

```
package com.example.smarthome;

import org.springframework.stereotype.Component;

@Component

public class Device {

        private String deviceType = "AC";

          private boolean status = false;

          public void turnOn() {

             status = true;
        System.out.println(deviceType + " is turned ON.");
    }

    public void turnOff() {
        status = false;
        System.out.println(deviceType + " is turned OFF.");
    }

    public String getDeviceType() {
        return deviceType;
    }

    public boolean isStatus() {
        return status;
    }

}
```

AutomationService.java

```
package com.example.smarthome;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class AutomationService {
```

```
                @Autowired
                private User user;

                @Autowired
                private Device device;

                public void controlDevice() {
                    System.out.println("User " + user.getName() + " with Home ID: " + user.getHomeId() + " is controlling device.");

                    device.turnOn();
                    device.turnOff();
                }
            }
```

User.java

```
package com.example.smarthome;

import org.springframework.stereotype.Component;

@Component
public class User {
            private String name = "Balu";
    private String homeId = "HOME-123";

    public String getName() {
        return name;
    }


    public String getHomeId() {
        return homeId;
    }


}
```

Output:

```
User Balu with Home ID: HOME-123 is controlling device.
AC is turned ON.
AC is turned OFF.
```