Case Study 1: Java-Based Configuration

Project Title: Online Food Ordering System

Configuration Type: Java-based Spring Configuration

Code Implementation

Customer.java

```
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

```
public class Restaurant {
    private String name;
    private String location;
    private String availableCuisine;

    public Restaurant(String name, String location, String availableCuisine) {
        this.name = name;
        this.location = location;
        this.availableCuisine = availableCuisine;
    }

    public String getName() { return name; }
    public String getLocation() { return location; }
    public String getAvailableCuisine() { return availableCuisine; }
}
```

FoodOrderService.java

```
public class FoodOrderService {
    private Customer customer;
    private Restaurant restaurant;

    public FoodOrderService(Customer customer, Restaurant restaurant) {
        this.customer = customer;
        this.restaurant = restaurant;
    }

    public void processOrder() {
        if (customer.getPreferredCuisine().equalsIgnoreCase(restaurant.getAvailableCuisine())) {
            System.out.println("Order placed successfully from " + restaurant.getName() + " for " +
customer.getName());
        } else {
            System.out.println("No matching cuisine found for " + customer.getName());
        }
    }
}
```

AppConfig.java

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

@Configuration
public class AppConfig {
    @Bean
    public Customer customer() {
        return new Customer("Minakshi", "9876543210", "Indian");
    }

    @Bean
    public Restaurant restaurant() {
        return new Restaurant("Spice Villa", "Pune", "Indian");
    }

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

MainApp.java

```
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 orderService = context.getBean(FoodOrderService.class);
        orderService.processOrder();
    }
}
```

Case Study 2: Annotation-Based Configuration

Project Title: Smart Home Automation System

Configuration Type: Annotation-based Spring Configuration

Code Implementation

User.java

```
import org.springframework.stereotype.Component;

@Component
public class User {
    private String name = "Minakshi";
    private String homeId = "HM1023";

    public String getName() { return name; }
    public String getHomeId() { return homeId; }
}
```

Device.java

```
import org.springframework.stereotype.Component;

@Component
public class Device {
    private String deviceType = "Light";
    private String status = "Off";

    public String getDeviceType() { return deviceType; }
    public String getStatus() { return status; }
    public void turnOn() {
        status = "On";
        System.out.println(deviceType + " is turned ON.");
    }
}
```

AutomationService.java

```
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() + " is controlling the device.");
        device.turnOn();
    }
}
```

AppConfig.java

```
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

@Configuration
@ComponentScan(basePackages = "com.smartHome")
public class AppConfig {
}
```

MainApp.java

```
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);
        AutomationService service = context.getBean(AutomationService.class);
        service.controlDevice();
    }
}
```