Da9_springassignment

Case study1:Hospital Management System

Patient.java:

```
package com.example.hospital;


public class Patient {
    public void registerPatient() {
        System.out.println("Patient registered successfully.");
    }


    public void getPatientDetails() {
        System.out.println("Fetching patient details...");
    }
}
```

Appointment.java:

```
package com.example.hospital;


public class Appointment {
    public void bookAppointment() {
        System.out.println("Appointment booked.");
    }


    public void cancelAppointment() {
        System.out.println("Appointment cancelled.");
    }
}
```

Billling.java:

```
package com.example.hospital;
```

```
public class Billing {
    public void generateBill() {
        System.out.println("Bill generated.");
    }

    public void sendBill() {
        System.out.println("Bill sent via email.");
    }
}
```

HospitalService.java:

```
package com.example.hospital;

public class HospitalService {
    private Patient patient;
    private Appointment appointment;
    private Billing billing;

    public void setPatient(Patient patient) {
        this.patient = patient;
    }

    public void setAppointment(Appointment appointment) {
        this.appointment = appointment;
    }

    public void setBilling(Billing billing) {
        this.billing = billing;
    }

    public void runServices() {
        patient.registerPatient();

        appointment.bookAppointment();

        billing.generateBill();

        billing.sendBill();

    }

}
```

applicationContext.xml:

```
<beans xmlns="http://www.springframework.org/schema/beans"

       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

       xsi:schemaLocation="http://www.springframework.org/schema/beans

       http://www.springframework.org/schema/beans/spring-beans.xsd">


    <bean id="patient" class="com.example.hospital.Patient" />

    <bean id="appointment" class="com.example.hospital.Appointment" />

    <bean id="billing" class="com.example.hospital.Billing" />


    <bean id="hospitalService" class="com.example.hospital.HospitalService">

        <property name="patient" ref="patient"/>

        <property name="appointment" ref="appointment"/>

        <property name="billing" ref="billing"/>

    </bean>

</beans>
```

Main.java:

```
package com.example.hospital;


import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;


public class MainApp {
    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");
        HospitalService service = context.getBean(HospitalService.class);
        service.runServices();
    }
}
```

Casestudy 2:

E=-commerce order processing:

Product.java:

```
package com.example.ecommerce;


import java.util.ArrayList; import java.util.List;


public class Product {
private List<String> productList = new ArrayList<>();


public void addProduct(String productName) { productList.add(productName);
System.out.println("Product added: " + productName);
}


public List<String> getProductList() { return productList;
}


public void listProducts() { System.out.println("Available Products:"); for (String product : productList)
{
System.out.println(" - " + product);
}
}
}
```

Order.java:

```
package com.example.ecommerce;
public class Order {


private String orderedProduct; private boolean isCancelled = false;
public void createOrder(String product) { this.orderedProduct = product;
System.out.println(" Order placed for: " + product);
}


public void cancelOrder() {
if (orderedProduct != null) { isCancelled = true;
System.out.println(" Order cancelled for: " + orderedProduct);
} else {
System.out.println(" No order to cancel.");
}
}


public boolean isOrderCancelled() { return isCancelled;
}
public String getOrderedProduct() { return orderedProduct;
}
}
```

Payment.java:

```
package com.example.ecommerce; public class Payment {
private boolean paymentDone = false;

public void processPayment(String product) { paymentDone = true;
System.out.println(" Payment processed for: " + product);
}

public void refundPayment(String product) { if (paymentDone) {
System.out.println(" Payment refunded for: " + product);
} else {
System.out.println(" Payment was not made yet.");
}
}
}
```

EccommerceService:

```
package com.example.ecommerce;
public class EcommerceService {
    private Product product;
    private Order order;
    private Payment payment;

public EcommerceService(Product product, Order order, Payment payment) { this.product = product;
this.order = order; this.payment = payment;
}
public void simulateOrderFlow() {
// Add products product.addProduct("Laptop"); product.addProduct("Smartphone");
product.addProduct("Headphones");


// List products product.listProducts();


// Create Order
String selectedProduct = "Laptop"; order.createOrder(selectedProduct);
// Process payment payment.processPayment(selectedProduct);


// Cancel Order order.cancelOrder();
// Refund
if (order.isOrderCancelled()) { payment.refundPayment(selectedProduct);
}
}
}
```

AppConfig:

```
package com.example.ecommerce;


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


@Configuration
public class AppConfig {


@Bean
public Product product() { return new Product();
}

@Bean
public Order order() { return new Order();
}

@Bean
public Payment payment() { return new Payment();
}
@Bean
public EcommerceService ecommerceService() {
return new EcommerceService(product(), order(), payment());
}
}
```

MainApp:

```
package com.example.ecommerce;

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);
EcommerceService service = context.getBean(EcommerceService.class);
service.simulateOrderFlow();
}
}
```

Case Study3:

Annotation-Based Configuration:

Book.java:

```
package com.example.liberary;
import org.springframework.stereotype.Component; @Component
public class Book {
public void addBook() {
System.out.println("Maths added to the catalog.");
}
public void searchBook() {
System.out.println(" Found Maths.");
}
}
```

Member:

```
package com.example.liberary;
import org.springframework.stereotype.Component; @Component
public class Member {
public void registerMember() {
System.out.println(" New member Anu registered.");
}

public void viewMembers() {
System.out.println(" Members: Anu,Sai");
}
}
```

Loan:

```
package com.example.liberary;
```

```
import org.springframework.stereotype.Component; @Component
public class Loan {
public void issueBook() {
System.out.println("Maths book is issued to Anu.");
}

public void returnBook() {
System.out.println("Maths book was returned by Anu.");
}
}
```

LibraryService:

```
package com.example.liberary;

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

@Service
public class LibraryService {
@Autowired
private Book book;

@Autowired
private Member member;
@Autowired private Loan loan;
public void manageLibrary() {
System.out.println(" Library System Initialized...\n");
member.registerMember(); book.addBook(); book.searchBook(); loan.issueBook();
loan.returnBook(); member.viewMembers();


System.out.println("\n Library operations completed.");
}
}
```

AppConfig:

```
package com.example.liberary;


import org.springframework.context.annotation.ComponentScan; import
org.springframework.context.annotation.Configuration;
@Configuration
@ComponentScan(basePackages = "com.example.liberary") public class AppConfig {


}
```

MainApp:

```
package com.example.liberary;


import org.springframework.context.annotation.AnnotationConfigApplicationContext; public
class MainApp {
public static void main(String[] args) {
AnnotationConfigApplicationContext context = new
AnnotationConfigApplicationContext(AppConfig.class);
LibraryService libraryService = context.getBean(LibraryService.class);
libraryService.manageLibrary();
context.close();
  }
}
```