|  |
| --- |
| Implement classes named OnlinePrescriptionSevice.java and PrintablePrescriptionService.java that implements the PrescriptionService.java. These classes are responsible for generating prescription in different formats (.html and .txt) |

It seems I encountered an issue accessing the files. I'll guide you through the task of implementing the OnlinePrescriptionService and PrintablePrescriptionService based on the information you provided and the required design.

**Task Explanation**

You need to implement two classes:

1. **OnlinePrescriptionService.java**: This class will generate prescriptions in .html format.
2. **PrintablePrescriptionService.java**: This class will generate prescriptions in .txt format.

Both of these classes will implement the PrescriptionService.java interface.

**Step-by-Step Explanation:**

**1. Interface: PrescriptionService.java**

This interface defines the method for generating prescriptions. Both OnlinePrescriptionService and PrintablePrescriptionService must implement this interface.

java

Copy code

package com.algonquin.cst8288.fall24.assignment1.prescription;

import com.algonquin.cst8288.fall24.assignment1.patient.Patient;

public interface PrescriptionService {

String generatePrescription(Patient patient);

}

**2. Class: OnlinePrescriptionService.java**

This class implements the PrescriptionService interface and generates a prescription in HTML format. It will return the prescription details as a string, formatted with HTML tags.

java

Copy code

package com.algonquin.cst8288.fall24.assignment1.prescription;

import com.algonquin.cst8288.fall24.assignment1.patient.Patient;

public class OnlinePrescriptionService implements PrescriptionService {

@Override

public String generatePrescription(Patient patient) {

// Generating prescription in HTML format

return "<html><body>"

+ "<h1>Prescription</h1>"

+ "<p><strong>Patient Name:</strong> " + patient.getName() + "</p>"

+ "<p><strong>Date of Birth:</strong> " + patient.getDateOfBirth() + "</p>"

+ "<p><strong>Life Stage:</strong> " + patient.getLifeStage() + "</p>"

+ "<p><strong>Planned Treatment:</strong> " + patient.getPlannedTreatment() + "</p>"

+ "<p><strong>Prescription Details:</strong> " + patient.getPrescription().getDetails() + "</p>"

+ "</body></html>";

}

}

**3. Class: PrintablePrescriptionService.java**

This class also implements the PrescriptionService interface, but it generates a prescription in plain text format (suitable for printing).

java

Copy code

package com.algonquin.cst8288.fall24.assignment1.prescription;

import com.algonquin.cst8288.fall24.assignment1.patient.Patient;

public class PrintablePrescriptionService implements PrescriptionService {

@Override

public String generatePrescription(Patient patient) {

// Generating prescription in text format

return "Prescription\n"

+ "---------------\n"

+ "Patient Name: " + patient.getName() + "\n"

+ "Date of Birth: " + patient.getDateOfBirth() + "\n"

+ "Life Stage: " + patient.getLifeStage() + "\n"

+ "Planned Treatment: " + patient.getPlannedTreatment() + "\n"

+ "Prescription Details: " + patient.getPrescription().getDetails() + "\n";

}

}

**4. Prescription.java**

This class will hold details of the prescription. You need to make sure this class has a getDetails() method that returns the actual prescription details.

java

Copy code

package com.algonquin.cst8288.fall24.assignment1.prescription;

public class Prescription {

private String details;

// Constructor

public Prescription(String details) {

this.details = details;

}

// Getter method for details

public String getDetails() {

return details;

}

// Setter method for details

public void setDetails(String details) {

this.details = details;

}

}

**Explanation of Code:**

1. **Interface (PrescriptionService.java)**:
   * This defines the method generatePrescription(Patient patient) that both the OnlinePrescriptionService and PrintablePrescriptionService will implement.
   * It forces both classes to have a generatePrescription method that accepts a Patient object and returns the prescription details.
2. **OnlinePrescriptionService.java**:
   * This class generates the prescription in an HTML format using standard HTML tags such as <h1>, <p>, and so on.
3. **PrintablePrescriptionService.java**:
   * This class generates the prescription in a plain-text format suitable for printing. It doesn't use any HTML or special formatting.
4. **Prescription.java**:
   * This class holds the prescription details and allows them to be set or retrieved through getter and setter methods.

**How This Fits Into the Overall System:**

* In your PatientManagement class, you'll use the PrescriptionService interface to generate prescriptions. Depending on whether you're generating an online prescription (HTML) or a printable prescription (text), you can inject either OnlinePrescriptionService or PrintablePrescriptionService when instantiating the PatientManagement class.

**Example of Usage:**

java

Copy code

PrescriptionService onlinePrescription = new OnlinePrescriptionService();

String htmlPrescription = onlinePrescription.generatePrescription(patient);

System.out.println(htmlPrescription);

PrescriptionService printablePrescription = new PrintablePrescriptionService();

String textPrescription = printablePrescription.generatePrescription(patient);

System.out.println(textPrescription);

In the example above, you can easily switch between generating an HTML prescription and a plain-text prescription by changing the implementation used.

**Final Thoughts:**

This design follows the **Strategy Pattern** because you are selecting different strategies (prescription formats) at runtime based on the class you inject (OnlinePrescriptionService or PrintablePrescriptionService). This provides flexibility and scalability for future prescription formats.