**VU21CSEN0102097**

**SEERAPU LAVANYA**

**Theme:** Health

**Problem Statement: Automatic Health Monitoring System:** In present days, the patients belonging to rural and sub-urban communities do not maintain the diagnosis reports for which they frequently go for regular checkups wasting their valuable money.

Thus, an automatic report maintain system is to be developed to avoid repetitive diagnosing of the patients.

**Goal:**  Good health and wellbeing

Patient:

**package** com.healthmonitoring.model;

**public** **class** Patient {

**private** String patientId;

**private** String name;

**private** **int** age;

**private** String gender;

**private** String address;

**private** String contactNumber;

**public** Patient(String patientId, String name, **int** age, String gender, String address, String contactNumber) {

**this**.patientId = patientId;

**this**.name = name;

**this**.age = age;

**this**.gender = gender;

**this**.address = address;

**this**.contactNumber = contactNumber;

}

**public** String getPatientId() { **return** patientId; }

**public** **void** setPatientId(String patientId) { **this**.patientId = patientId; }

**public** String getName() { **return** name; }

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

**public** **int** getAge() { **return** age; }

**public** **void** setAge(**int** age) { **this**.age = age; }

**public** String getGender() { **return** gender; }

**public** **void** setGender(String gender) { **this**.gender = gender; }

**public** String getAddress() { **return** address; }

**public** **void** setAddress(String address) { **this**.address = address; }

**public** String getContactNumber() { **return** contactNumber; }

**public** **void** setContactNumber(String contactNumber) { **this**.contactNumber = contactNumber; }

}

**Report:**

**package** com.healthmonitoring.model;

**import** java.util.Date;

**public** **class** Report {

**private** String reportId;

**private** String patientId;

**private** Date date;

**private** String diagnosis;

**private** String doctorName;

**private** String reportFilePath;

**public** Report(String reportId, String patientId, Date date, String diagnosis, String doctorName, String reportFilePath) {

**this**.reportId = reportId;

**this**.patientId = patientId;

**this**.date = date;

**this**.diagnosis = diagnosis;

**this**.doctorName = doctorName;

**this**.reportFilePath = reportFilePath;

}

**public** String getReportId() { **return** reportId; }

**public** **void** setReportId(String reportId) { **this**.reportId = reportId; }

**public** String getPatientId() { **return** patientId; }

**public** **void** setPatientId(String patientId) { **this**.patientId = patientId; }

**public** Date getDate() { **return** date; }

**public** **void** setDate(Date date) { **this**.date = date; }

**public** String getDiagnosis() { **return** diagnosis; }

**public** **void** setDiagnosis(String diagnosis) { **this**.diagnosis = diagnosis; }

**public** String getDoctorName() { **return** doctorName; }

**public** **void** setDoctorName(String doctorName) { **this**.doctorName = doctorName; }

**public** String getReportFilePath() { **return** reportFilePath; }

**public** **void** setReportFilePath(String reportFilePath) { **this**.reportFilePath = reportFilePath; }

}

**Patient Repository:**

package com.healthmonitoring.repository;

import com.healthmonitoring.model.Patient;

import java.util.HashMap;

import java.util.Map;

public class PatientRepository {

private Map<String, Patient> patientDatabase = new HashMap<>();

public void addPatient(Patient patient) {

patientDatabase.put(patient.getPatientId(), patient);

}

public Patient getPatient(String patientId) {

return patientDatabase.get(patientId);

}

public boolean patientExists(String patientId) {

return patientDatabase.containsKey(patientId);

}

}

**Report repository:**

package com.healthmonitoring.repository;

import com.healthmonitoring.model.Report;

import java.util.HashMap;

import java.util.Map;

public class ReportRepository {

private Map<String, Report> reportDatabase = new HashMap<>();

public void addReport(Report report) {

reportDatabase.put(report.getReportId(), report);

}

public Report getReport(String reportId) {

return reportDatabase.get(reportId);

}

public boolean isDuplicateDiagnosis(String patientId, String diagnosis) {

return reportDatabase.values().stream()

.anyMatch(report -> report.getPatientId().equals(patientId) &&

report.getDiagnosis().equalsIgnoreCase(diagnosis));

}

}

**Health monitoring service:**

package com.healthmonitoring.service;

import com.healthmonitoring.model.Patient;

import com.healthmonitoring.model.Report;

import com.healthmonitoring.repository.PatientRepository;

import com.healthmonitoring.repository.ReportRepository;

public class HealthMonitoringService {

private PatientRepository patientRepository;

private ReportRepository reportRepository;

public HealthMonitoringService() {

this.patientRepository = new PatientRepository();

this.reportRepository = new ReportRepository();

}

public void registerPatient(Patient patient) {

patientRepository.addPatient(patient);

}

public String addReport(Report report) {

if (reportRepository.isDuplicateDiagnosis(report.getPatientId(), report.getDiagnosis())) {

return "Duplicate diagnosis found for the patient. Report not added.";

}

reportRepository.addReport(report);

return "Diagnosis record added successfully!";

}

public Patient getPatientDetails(String patientId) {

return patientRepository.getPatient(patientId);

}

public Report getReportDetails(String reportId) {

return reportRepository.getReport(reportId);

}

}

**Main class:**

package com.healthmonitoring.ui;

import com.healthmonitoring.model.Patient;

import com.healthmonitoring.model.Report;

import com.healthmonitoring.service.HealthMonitoringService;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

HealthMonitoringService service = new HealthMonitoringService();

Scanner scanner = new Scanner(System.in);

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

System.out.println("Welcome to Automatic Health Monitoring System");

while (true) {

System.out.println("1. Add Patient");

System.out.println("2. Add Diagnosis Record");

System.out.println("3. Generate Report");

System.out.println("4. Retrieve Patient Details");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

System.out.print("Enter Patient ID: ");

String patientId = scanner.nextLine();

System.out.print("Enter Name: ");

String name = scanner.nextLine();

System.out.print("Enter Age: ");

int age = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter Gender: ");

String gender = scanner.nextLine();

System.out.print("Enter Address: ");

String address = scanner.nextLine();

System.out.print("Enter Contact Number: ");

String contactNumber = scanner.nextLine();

Patient patient = new Patient(patientId, name, age, gender, address, contactNumber);

service.registerPatient(patient);

System.out.println("Patient added successfully!");

break;

case 2:

System.out.print("Enter Patient ID: ");

String rptPatientId = scanner.nextLine();

System.out.print("Enter Diagnosis Date (YYYY-MM-DD): ");

Date diagnosisDate;

try {

diagnosisDate = dateFormat.parse(scanner.nextLine());

} catch (Exception e) {

System.out.println("Invalid date format.");

break;

}

System.out.print("Enter Diagnosis Details: ");

String diagnosisDetails = scanner.nextLine();

System.out.print("Enter Doctor Name: ");

String doctorName = scanner.nextLine();

System.out.print("Enter Report File Path: ");

String reportFilePath = scanner.nextLine();

String reportId = rptPatientId + "\_" + dateFormat.format(diagnosisDate);

Report report = new Report(reportId, rptPatientId, diagnosisDate, diagnosisDetails, doctorName, reportFilePath);

String result = service.addReport(report);

System.out.println(result);

break;

case 3:

System.out.print("Enter Patient ID: ");

String reportPatientId = scanner.nextLine();

System.out.println("Generating report for Patient ID: " + reportPatientId + "...");

Patient patientDetails = service.getPatientDetails(reportPatientId);

if (patientDetails != null) {

System.out.println("Patient Name: " + patientDetails.getName());

System.out.println("Age: " + patientDetails.getAge());

System.out.println("Gender: " + patientDetails.getGender());

System.out.println("Address: " + patientDetails.getAddress());

System.out.println("Contact Number: " + patientDetails.getContactNumber());

System.out.println();

System.out.println("Diagnosis History:");

boolean hasReports = false;

if (!hasReports) {

System.out.println("No diagnosis records found.");

}

} else {

System.out.println("Patient not found.");

}

break;

case 4:

System.out.print("Enter Patient ID: ");

String patientIdToRetrieve = scanner.nextLine();

Patient patientDetail = service.getPatientDetails(patientIdToRetrieve);

if (patientDetail != null) {

System.out.println("Patient Details:");

System.out.println("Patient ID: " + patientDetail.getPatientId());

System.out.println("Name: " + patientDetail.getName());

System.out.println("Age: " + patientDetail.getAge());

System.out.println("Gender: " + patientDetail.getGender());

System.out.println("Address: " + patientDetail.getAddress());

System.out.println("Contact Number: " + patientDetail.getContactNumber());

System.out.println();

System.out.println("Diagnosis History:");

boolean hasDiagnosis = false;

if (!hasDiagnosis) {

System.out.println("No diagnosis records found.");

}

} else {

System.out.println("Patient not found.");

}

break;

case 5:

System.out.println("Exiting...");

System.exit(0);

break;

default:

System.out.println("Invalid choice. Please enter a valid option.");

}

}

}

}