```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Scanner;

class User {
    private String username;
    private String password;
    private List<Appointment> appointments;
    private List<WorkoutPlan> workoutPlans;
    private List<HealthAssessment> healthAssessments;

    public User(String username, String password) {
        this.username = username;
        this.password = password;
        this.appointments = new ArrayList<>();
        this.workoutPlans = new ArrayList<>();
        this.healthAssessments = new ArrayList<>();
    }

    public String getUsername() {
        return username;
    }

    public String getPassword() {
        return password;
    }

    public List<Appointment> getAppointments() {
        return appointments;
    }

    public List<WorkoutPlan> getWorkoutPlans() {
        return workoutPlans;
    }

    public List<HealthAssessment> getHealthAssessments() {
        return healthAssessments;
    }
}

class Appointment {
    private String date;
    private String time;
    private String description;
```

```
    public Appointment(String date, String time, String description) {
        this.date = date;
        this.time = time;
        this.description = description;
    }

    public String getDate() {
        return date;
    }

    public String getTime() {
        return time;
    }

    public String getDescription() {
        return description;
    }
}

class WorkoutPlan {
    private String name;
    private String description;

    public WorkoutPlan(String name, String description) {
        this.name = name;
        this.description = description;
    }

    public String getName() {
        return name;
    }

    public String getDescription() {
        return description;
    }
}

class HealthAssessment {
    private String date;
    private double weight;
    private double height;
    private double bmi;

    public HealthAssessment(String date, double weight, double height) {
        this.date = date;
        this.weight = weight;
        this.height = height;
        this.bmi = calculateBMI();
```

```
    }

    public String getDate() {
        return date;
    }

    public double getWeight() {
        return weight;
    }

    public double getHeight() {
        return height;
    }

    public double getBMI() {
        return bmi;
    }

    private double calculateBMI() {
        return weight / (height * height);
    }
}

public class FitnessApplication {
    private Map<String, User> users;
    private User currentUser;
    private Scanner scanner;

    public FitnessApplication() {
        this.users = new HashMap<>();
        this.scanner = new Scanner(System.in);
    }

    public void run() {
        boolean exit = false;
        while (!exit) {
            System.out.println("Welcome to Fitness Application!");
            System.out.println("1. Login");
            System.out.println("2. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline character

            switch (choice) {
                case 1:
                    login();
                    break;
                case 2:
```

```
            exit = true;
            break;
        default:
            System.out.println("Invalid choice. Please try again.");
      }
   }
}

private void login() {
   System.out.print("Enter username: ");
   String username = scanner.nextLine();
   System.out.print("Enter password: ");
   String password = scanner.nextLine();

   if (users.containsKey(username)) {
      User user = users.get(username);
      if (user.getPassword().equals(password)) {
         currentUser = user;
         System.out.println("Login successful!");
         showMainMenu();
      } else {
         System.out.println("Incorrect password. Please try again.");
      }
   } else {
      System.out.println("User not found. Please try again.");
   }
}

private void showMainMenu() {
   boolean logout = false;
   while (!logout) {
      System.out.println("\nMain Menu");
      System.out.println("1. Schedule Appointment");
      System.out.println("2. View Appointments");
      System.out.println("3. View Workout Plans");
      System.out.println("4. Track Workout Progress");
      System.out.println("5. Take Health Assessment");
      System.out.println("6. Logout");
      System.out.print("Enter your choice: ");
      int choice = scanner.nextInt();
      scanner.nextLine(); // Consume newline character

      switch (choice) {
         case 1:
            scheduleAppointment();
            break;
         case 2:
            viewAppointments();
```

```
            break;
        case 3:
            viewWorkoutPlans();
            break;
        case 4:
            trackWorkoutProgress();
            break;
        case 5:
            takeHealthAssessment();
            break;
        case 6:
            logout = true;
            currentUser = null;
            break;
        default:
            System.out.println("Invalid choice. Please try again.");
      }
    }
}

private void scheduleAppointment() {
    System.out.print("Enter appointment date: ");
    String date = scanner.nextLine();
    System.out.print("Enter appointment time: ");
    String time = scanner.nextLine();
    System.out.print("Enter appointment description: ");
    String description = scanner.nextLine();

    Appointment appointment = new Appointment(date, time, description);
    currentUser.getAppointments().add(appointment);
    System.out.println("Appointment scheduled successfully!");
}

private void viewAppointments() {
    List<Appointment> appointments = currentUser.getAppointments();
    if (appointments.isEmpty()) {
        System.out.println("No appointments scheduled.");
    } else {
        System.out.println("\nAppointments:");
        for (Appointment appointment : appointments) {
            System.out.println("Date: " + appointment.getDate());
            System.out.println("Time: " + appointment.getTime());
            System.out.println("Description: " + appointment.getDescription());
            System.out.println();
        }
    }
}
```

```
private void viewWorkoutPlans() {
    List<String> monday = new ArrayList<>();
    monday.add("Chest press");
    monday.add("Incline dumbbell press");
    monday.add("Dumbbell flyes");
    monday.add("Push-ups");

    List<String> tuesday = new ArrayList<>();
    tuesday.add("Squats");
    tuesday.add("Leg press");
    tuesday.add("Lunges");
    tuesday.add("Calf raises");

    List<String> wednesday = new ArrayList<>();
    wednesday.add("Shoulder press");
    wednesday.add("Lateral raises");
    wednesday.add("Front raises");
    wednesday.add("Upright rows");

    List<String> thursday = new ArrayList<>();
    thursday.add("Deadlifts");
    thursday.add("Bent over rows");
    thursday.add("Lat pull-downs");
    thursday.add("Seated cable rows");

    List<String> friday = new ArrayList<>();
    friday.add("Bicep curls");
    friday.add("Tricep dips");
    friday.add("Hammer curls");
    friday.add("Tricep pushdowns");

    System.out.println("Monday Workout:");
    printWorkout(monday);

    System.out.println("Tuesday Workout:");
    printWorkout(tuesday);

    System.out.println("Wednesday Workout:");
    printWorkout(wednesday);

    System.out.println("Thursday Workout:");
    printWorkout(thursday);

    System.out.println("Friday Workout:");
    printWorkout(friday);
}

public static void printWorkout(List<String> workout) {
```

```
            for (String exercise : workout) {
                System.out.println(exercise);
            }
            System.out.println();
                }
            }
        }

        private void trackWorkoutProgress() {
            // Implement workout progress tracking logic here
            System.out.println("Workout progress tracking feature is not implemented yet.");
        }

        private void takeHealthAssessment() {
            System.out.print("Enter assessment date: ");
            String date = scanner.nextLine();
            System.out.print("Enter weight (in p): ");
            double weight = scanner.nextDouble();
            System.out.print("Enter height (in meters): ");
            double height = scanner.nextDouble();

            HealthAssessment healthAssessment = new HealthAssessment(date, weight, height);
            currentUser.getHealthAssessments().add(healthAssessment);
            System.out.println("Health assessment taken successfully!");
        }

        public static void main(String[] args) {
            FitnessApplication fitnessApp = new FitnessApplication();
            // Create some sample users
            User user1 = new User("user1", "password1");
            User user2 = new User("user2", "password2");
            fitnessApp.users.put(user1.getUsername(), user1);
            fitnessApp.users.put(user2.getUsername(), user2);
            fitnessApp.run();
        }
    }
```