```
package mypackage;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Calendar;
import java.util.Date;
import java.util.ArrayList;

public class Record {
    // Variables
    private String firstName;
    private String lastName;
    private String address;
    private String city;
    private String county;
    private String state;
    private String zip;
    private String phone1;
    private String phone2;
    private String email;
    private String dateListed;
    private String unosStatus;
    private String dateOfBirth;
    private double age;
    private ArrayList<String> unosHistory = new ArrayList<>();


    //default constructor
    public Record(){}


    // Constructor for if values are known
    public Record(String firstName, String lastName, String address, String city, String county, String state, String zip, String phone1, String phone2, String email, String dateListed, String unosStatus, String dateOfBirth){
        this.firstName = firstName;
        this.lastName = lastName;
        this.address = address;
        this.city = city;
        this.county = county;
        this.state = state;
        this.zip = zip;
        this.phone1 = phone1;
        this.phone2 = phone2;
        this.email = email;
        this.dateListed = dateListed;
        this.unosStatus = unosStatus;
        this.dateOfBirth = dateOfBirth;
        this.age = calculateAge(dateOfBirth);
        this.unosHistory.add("On " + this.dateListed + ", UNOS_status changed to " + this.unosStatus);
    }



    // Get the patient's first name
    public String getFirstName(){return this.firstName;}


    // Get the patient's last name
    public String getLastName(){return this.lastName;}
```

```
// Get the patient's address
public String getAddress(){return this.address;}

// Get the patient's city
public String getCity(){return this.city;}

// Get the patient's county
public String getCounty(){return this.county;}

// Get the patient's state
public String getState(){return this.state;}

// Get the patient's zip
public String getZip(){return this.zip;}

// Get the patient's phone1
public String getPhone1(){return this.phone1;}

// Get the patient's phone2
public String getPhone2(){return this.phone2;}

// Get the patient's county
public String getEmail(){return this.email;}

// Get the patient's date listed
public String getDateListed(){return this.dateListed;}

// Get the patient's UNOS Status
```

```
public String getUnosStatus(){return this.unosStatus;}

// Get the patient's DOB
public String getDateOfBirth(){return this.dateOfBirth;}

// Get the patient's age
public double getAge(){return this.age;}

// Returns the history arrayList
public ArrayList<String> getUnosHistory(){return this.unosHistory;}

// Prints the unosHistory
public void printUnosHistory(){
   System.out.println("\nUNOS status history of patient:");
   for(int i = 0; i < this.unosHistory.size(); i++){
      System.out.println(this.unosHistory.get(i));
   }
}

// Set the UNOS Status
public void setUnosStatus(String newUnosStatus){
   this.unosStatus = newUnosStatus;
   this.unosHistory.add("On " + this.dateListed + ", UNOS_status changed to " + this.unosStatus);
}

// Set the date listed
public void setDateListed(){
   LocalDate currentDate = LocalDate.now();
   DateTimeFormatter formatter = DateTimeFormatter.ofPattern("MM/dd/yyyy");
```

```
    String formattedDate = currentDate.format(formatter);
    System.out.println("date: " + formattedDate);
    this.dateListed = formattedDate;
}

/*
 * Calculates the age of the patient
 *
 * @param dateOfBirth: the date of birth of the patient
 * @return ageInDecimal: the age of the person
 */

private double calculateAge(String dateOfBirth){
    try {
        // Parse the DOB string into a Date object
        Date dob = dateFormat.parse(dateOfBirth);

        // Get the current date
        Date currentDate = new Date();

        // Create Calendar objects for DOB and current date
        Calendar dobCalendar = Calendar.getInstance();
        dobCalendar.setTime(dob);
        Calendar currentCalendar = Calendar.getInstance();
        currentCalendar.setTime(currentDate);

        // Calculate the age in years
        int years = currentCalendar.get(Calendar.YEAR) - dobCalendar.get(Calendar.YEAR);
```

```
            // Calculate the fraction of a year in terms of months
            int currentMonth = currentCalendar.get(Calendar.MONTH);
            int dobMonth = dobCalendar.get(Calendar.MONTH);
            int months = currentMonth - dobMonth;

            if (months < 0) {
                years--;
                months = 12 + months;
            }

            double ageInDecimal = years + (months / 12.0);
            return ageInDecimal;

        } catch (ParseException e) {
            System.err.println("Invalid date format: " + dateOfBirth);
            return 0;
        }
    }
    // Define the date format for parsing
        SimpleDateFormat dateFormat = new SimpleDateFormat("MM/dd/yyyy");



}
```