NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

A MINI PROJECT REPORT

for

Mini Project in Java (19CSE48)

on

Her Cycle Tracker

Submitted by

TRIVENI A

USN: 1NH20CS230, Sem-Sec: 4-D

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Academic Year: 2021-22(EVEN SEM)

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

CERTIFICATE

This is to certify that the mini project work titled

HER CYCLE TRACKER

submitted in partial fulfillment of the degree of Bachelor of Engineering in Computer Science and Engineering by

TRIVENI A
USN:1NH20CS230

DURING

EVEN SEMESTER 2021-2022

for

Course: Mini Project in Java-19CSE48

Signature of Reviewer Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner *Signature with date*

1._____ _____

2._____ _____

ABSTRACT

Her Cycle Tracker is an application that can be used to track every woman's menstrual cycle dates. Every women tends to forget their period dates at one point of time either when we get busy with our life schedule or busy studying for our exams. It can be used for women of any age. Young girls who have just had their first period or women who are close to menopause.

This application can also be used by men to track their loved one's period dates. It is 2022 and every human deserves to know about menstruation. This can help them plan that one trip they always wanted to go to without having to compromise because of period discomfort or plan to wear that one light coloured pant which has been sitting in the corner of the cupboard.

But as this sector continues to grow, we still lack the knowledge about many women facts. Many myths have been followed through out these years which should be realized by the people that these are only myths that had been made in the past when women did not have sufficient objects to fulfil their needs such as pads or tampons.

The COVID 19 pandemic has also been a bane to many women as we just sat at our homes with no exercise which increased the rate of PCOS and PCOD. The present rate of women with PCOS or PCOD is one in every 10 women. This app will also notice their changes when the cycle gap is irregular and suggest for pelvis abdomen scan or a thyroid test.

This mini-project solves the above-mentioned problems through the development of a desktop application. This desktop application would maintain a database of the user's menstruation dates every month , the gap length between two periods,the length of the period,etc.

The entire program has been developed using Java in Eclipse .

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned my efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman, New Horizon Educational Institutions, for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha,** Principal, New Horizon College of Engineering, for his constant support and encouragement.

I would like to thank **Dr. Anandhi R J**, Professor and Dean-Academics, NHCE, for her valuable guidance.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and HOD, Department of Computer Science and Engineering, for her constant support.

I also express my gratitude to **Ms. Manjula A K** , Designation, Department of Computer Science and Engineering, my mini project reviewer, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

Student name

USN: your USN

CONTENTS

ABSTRACT **I**

ACKNOWLEDGEMENT **II**

CONTENTS **III**

LIST OF FIGURES **V**

1. INTRODUCTION

1.1. PROBLEM DEFINITION **1**

1.2. OBJECTIVES **1**

1.3. METHODOLOGY TO BE FOLLOWED **1**

1.4. EXPECTED OUTCOMES **2**

1.5. HARDWARE AND SOFTWARE REQUIREMENTS **2**

2. **FUNDAMENTALS OF JAVA PROGRAMMING**

2.1. INTRODUCTION **3**

2.2. FEATURES OF JAVA **3**

2.3. CLASSES AND OBJECTS **4**

2.4. INHERITANCE **5**

2.5. POLYMORPHISM **6**

2.6. JAVA PACKAGES **7**

2.7. EXCEPTION HANDLING **8**

2.8. THREADS **9**

2.9. I/O BASICS **10**

3. DESIGN

3.1. DESIGN GOALS **11**

3.2. ALGORITHM **11**

3.3. FLOWCHART **12**

4. IMPLEMENTATION

4.1. MODULE 1 : CONNECTION **13**

4.2. MODULE 2 : LOGIN **14**

4.3. MODULE 3 : DATES **16**

4.4. MODULE 4 : DATABASE **19**

4.5. MODULE 5 : FACTS **22**

5. RESULTS

5.1. LOGIN RESULT **24**

5.2. DATES RESULT **24**

5.3. DATABASE RESULT WHILE ADD **25**

5.4. DATABASE RESULT WHILE RESET **25**

5.5. FACTS RESULT **26**

6. CONCLUSION **27**

REFERENCES **28**

LIST OF FIGURES

Figure No	Figure Description	Page No
2.2.1	Class example	4
2.4.1	Inheritance syntax	5
2.4.2	Types of Inheritance	6
2.5.1	Types of Polymorphism	7
2.6.1	Java Packages	7
2.6.2	Package example	8
2.8	Life cycle of a thread	9
3.1	Design Flowchart	12
4.1	Connection module	13
4.2.1	Login module part 1	14
4.2.2	Login module part 2	14
4.2.3	Login module part 3	15
4.2.4	Login module Jframe	15
4.3.1	Dates module part 1	16
4.3.2	Dates module part 2	16
4.3.3	Dates module part 3	17
4.3.4	Dates module part 4	17
4.3.5	Dates module Jframe	18

Figure No	Figure Description	Page No
4.4.1	Database module part 1	19
4.4.2	Database module part 2	19
4.4.3	Database module part 3	20
4.4.4	Database module part 4	20
4.4.5	Database module part 5	21
4.4.6	Database module Jframe	21
4.5.1	Facts module part 1	22
4.5.2	Facts module part 2	22
4.5.3	Facts module part 3	23
4.5.4	Facts module Jframe	23
5.1.1	Login result	24
5.1.2	Dates result	24
5.1.3	Database result while add	25
5.1.4	Database result while reset	25
5.1.5	Facts result	26

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

Menstruation happens in every female. Inspite of having abundance of information over the internet, many still lack knowledge about various facts about their monthly cycle. Her Menstrual tracker is a menstrual tracker that not only tracks your cycle based on the cycle length, it also provides information about female health.

1.2 OBJECTIVES

It helps the user keep track of her periods in the data base provided.

It helps calculate the number of days he period lasted.

It shows facts about menstruation that are not myths as believed from generations.

1.3 METHODOLOGY TO BE FOLLOWED

Initially, a login page is setup so that the user can log in to the database.

The user can now enter the period start date and end date and calculate the number of days the period lasted.

They can now enter their details which is their name, age, period start date and period end date which will be stored in the database. Use of linked lists to store their details as mentioned above.
JFrames are used to make the front end of the project.

Use of classes and objects are used for storing the data . The facts that are displayed are by using JLabels .

1.4 EXPECTED OUTCOMES

Helps women track their menstruation cycle that happens every 25-35 days as it differs from one woman to another.

It shows facts related to menstruation.

1.5 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements:

1) A Personal Computer
2) Minimum of 2gb RAM
3) 64 bit Operating system
4) Windows 7 or above

Software requirements:

1) Eclipse
2) Netbeans

CHAPTER 2

FUNDAMENTALS OF JAVA PROGRAMMING

2.1 Introduction

Java is a programming language and a stage. Java is a significant level, vigorous, object-arranged and secure programming language. Java was created by Sun Microsystems (which is presently the auxiliary of Oracle) in the year 1995. James Gosling is known as the dad of Java. Prior to Java, its name was Oak. Since Oak was at that point an enrolled organization, so James Gosling and his group changed the Oak name to Java. Any equipment or programming climate in which a program runs, is known as stage. Since Java has a runtime climate (JRE) and API, it's anything but a stage.

2.2 Features of Java

The primary objective of Java programming language creation was to make it portable, simple and secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. The features of Java are also known as Java buzzwords.

- ✓ Simple
- ✓ Object-Oriented
- ✓ Portable
- ✓ Platform independent
- ✓ Secured
- ✓ Robust
- ✓ Architecture neutral
- ✓ Interpreted
- ✓ High Performance
- ✓ Multithreaded

- ✓ Distributed
- ✓ Dynamic

2.3 Classes and Objects

A class is a blueprint from which individual objects are created.

```
public class Dog {
   String breed;
   int age;
   String color;

   void barking() {
   }

   void hungry() {
   }

   void sleeping() {
   }
}
```

Fig 2.2.1: Class example

A class can contain any of the following variable types.

Local variables – Variables defined inside methods, constructors or blocks are called local variables. The variable will be declared and initialized within the method and the variable will be destroyed when the method has completed.

Instance variables – Instance variables are variables within a class but outside any method. These variables are initialized when the class is instantiated. Instance variables can be accessed from inside any method, constructor or blocks of that particular class.

Class variables – Class variables are variables declared within a class, outside any method, with the static keyword.

An element that has state and conduct is known as an object eg, seat, bicycle, marker, pen, table, vehicle, and so forth. It very well may be physical or intelligent. Object is an example of a class. The case of a theoretical object is the financial framework.

An object has three characteristics:

- State: represents the data(value) of an object.
- Behaviour: represents the behaviour of an object such as deposit, withdraw, etc

- Identity: An object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. However, it is used internally by the JVM to identify each object uniquely.

2.4 Inheritance

Inheritance is one of the key features of OOP that allows us to create a new class from an existing class. The new class that is created is known as subclass (child or derived class) and the existing class from where the child class is derived is known as superclass (parent or base class).

Utilization of Inheritance in java:

- For technique Overriding (to accomplish runtime polymorphism)
- For Code Reusability

Syntax of Java Inheritance:

```
class Subclass-name extends Superclass-name
{
  //methods and fields
}
```

Fig 2.4.1 - Inheritance syntax

The extends keyword indicates that you are making a new class that derives from an existing class.

Types of inheritance:

There are 5 types of inheritance on a basic level, they are as follows

- Single Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance
- Multiple Inheritance

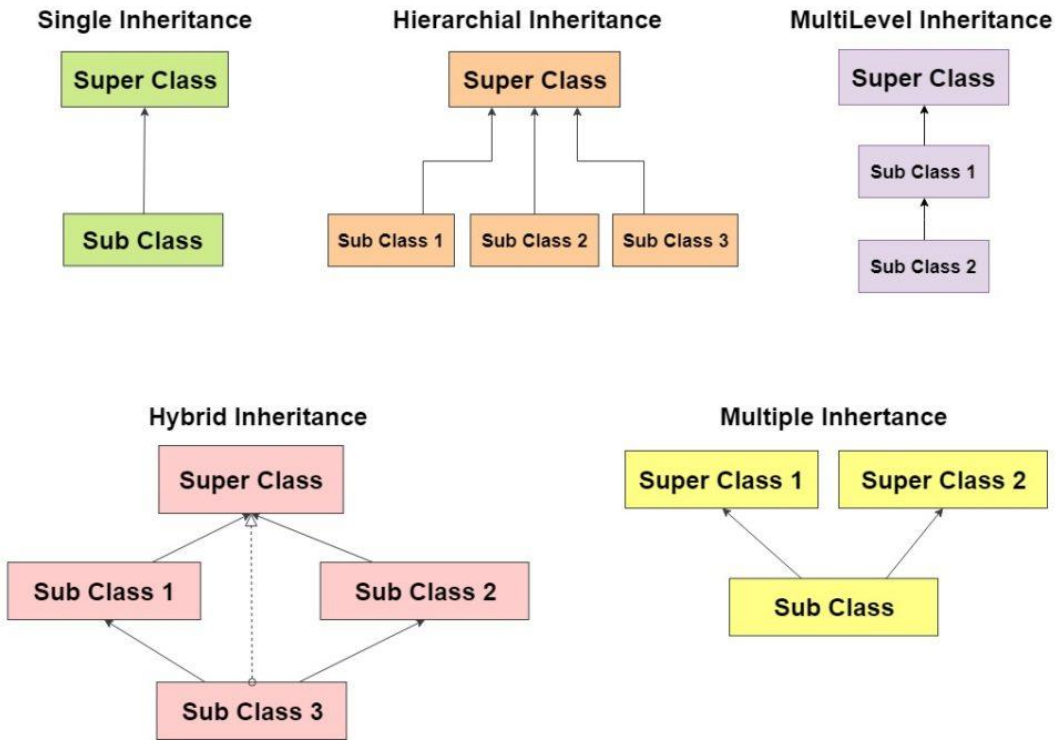


Fig 2.4.2 - Types of Inheritance

2.5 Polymorphism

Polymorphism in Java is an idea by which we can play out a solitary activity in an unexpected way. It is gotten from 2 Greek words: poly and transforms. "Poly" signifies numerous and "transforms" signifies structures.

There are two sorts of polymorphism in Java:

1. Compile-time polymorphism
2. Run-time polymorphism

Run-time polymorphism: It is an interaction where a call to a superseded strategy is settled at runtime as opposed to incorporate time.

In this cycle, an abrogated strategy is called through the reference variable of a superclass. The assurance of the strategy to be called depends on the item being alluded to by the reference variable.

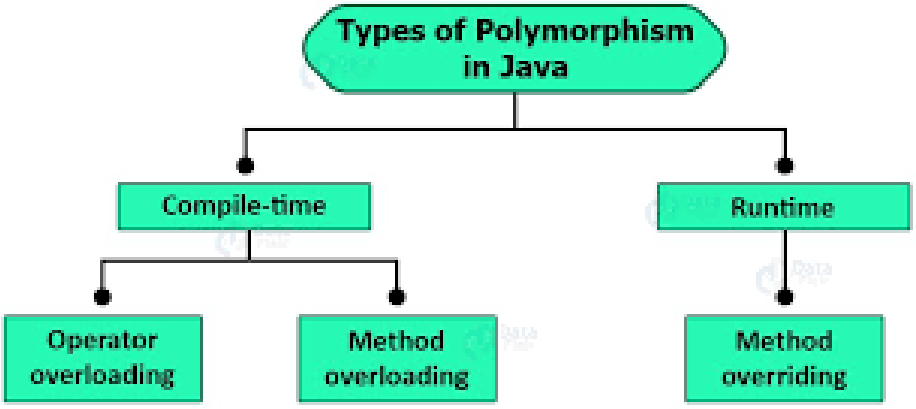


Fig 2.5.1- Types of polymorphism

2.6 Java Packages

A java package is a gathering of comparable kinds of classes, interfaces and sub-bundles.

Packages in java can be classified in two structure, built-in and user-defined packages. There are many built-in packages like java, lang, awt, javax, swing, net, io, util, sql and so on

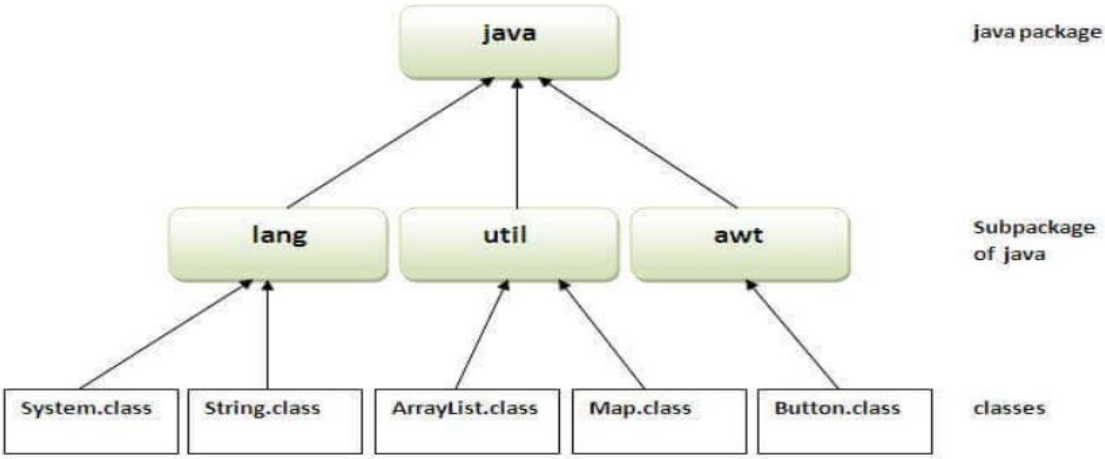


Fig 2.6.1 - Java packages

Example:

```
//save as Simple.java
package mypack;
public class Simple{
 public static void main(String args[]){
   System.out.println("Welcome to package");
  }
}
```

Fig 2.6.2- package example

2.7 Exception Handling

The exception handling in Java is one of the amazing system to deal with the runtime mistakes so ordinary progression of the application can be kept up with.

Kinds of Java Exceptions:

• Checked Exception

• Unchecked Exception

• Error

Benefit of Exception Handling:

The centre benefit of exception handling is to keep up with the typical progression of the application. A special case ordinarily upsets the typical progression of the application that is the reason we use exemption dealing with.

Handler exceptions can be handled in 3 ways:

- Try , catch block
- Finally block
- Throw and throws keywords

2.8 Threads

A thread in Java is the direction or path that is taken while a program is being executed. Generally, all the programs have at least one thread, known as the main thread, that is provided by the JVM or Java Virtual Machine at the starting of the program's execution. At this point, when the main thread is provided, the main() method is invoked by the main thread.

A thread is critical in the program because it enables multiple operations to take place within a single method. Each thread in the program often has its own program counter, stack, and local variable.

In Java, making a thread is refined by carrying out an interface and broadening a class. Each Java Thread is made and constrained by the java.lang.Thread class. A solitary strung application has just one thread and can deal with just each undertaking in turn. To deal with various undertakings in equal, multi-threading is utilized: numerous threads are made, each playing out an alternate errand.

Life cycle of a thread has 5 stages.

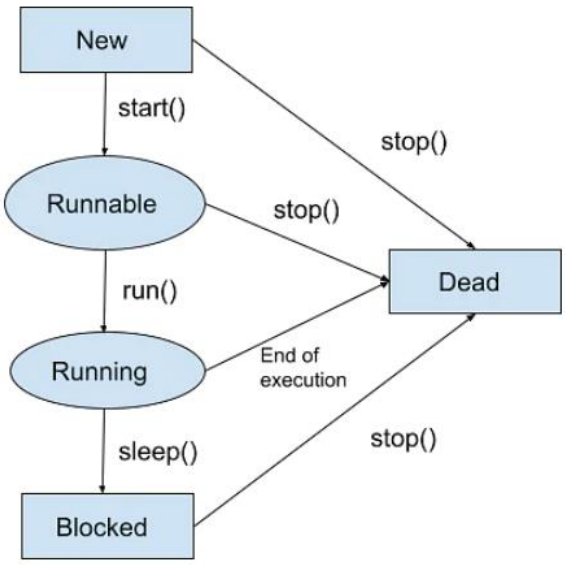


Fig 2.8 - Lifecycle of a thread

The 5 stages are namely,

1. New
2. Runnable
3. Running
4. Blocked (Non-runnable state)
5. Dead

2.9 I/O basics

Java I/O (Input and Output) is utilized to handle the info and produce the yield. Java utilizes the idea of a stream to make I/O activity quick.

Stream: A stream is a grouping of information. In Java, a stream is made out of bytes.

3 streams are created for us automatically. They are:

1. System.out: standard output stream
2. System.in: standard input stream
3. System.err: standard error stream

CHAPTER 3

DESIGN

3.1 DESIGN GOALS

Our design mechanism includes two main things:

1. Track the menstrual cycle
2. Display facts about menstruation

3.2 ALGORITHM / PSEUDOCODE

1. Login
   1.1. input username
   1.2. input password , it gets logged I if the username and password is correct.

2. Dates
   2.1. input period start date and month
   2.2. input period end date and month
   2.3. Calculate the number of days period lasted using arithematic functions

3. Database
   3.1. input name,age, period start date and end date
   3.2. It get stored in the database

4. Facts
   4.1. View the menstrual facts

5. End

3.3 FLOWCHART

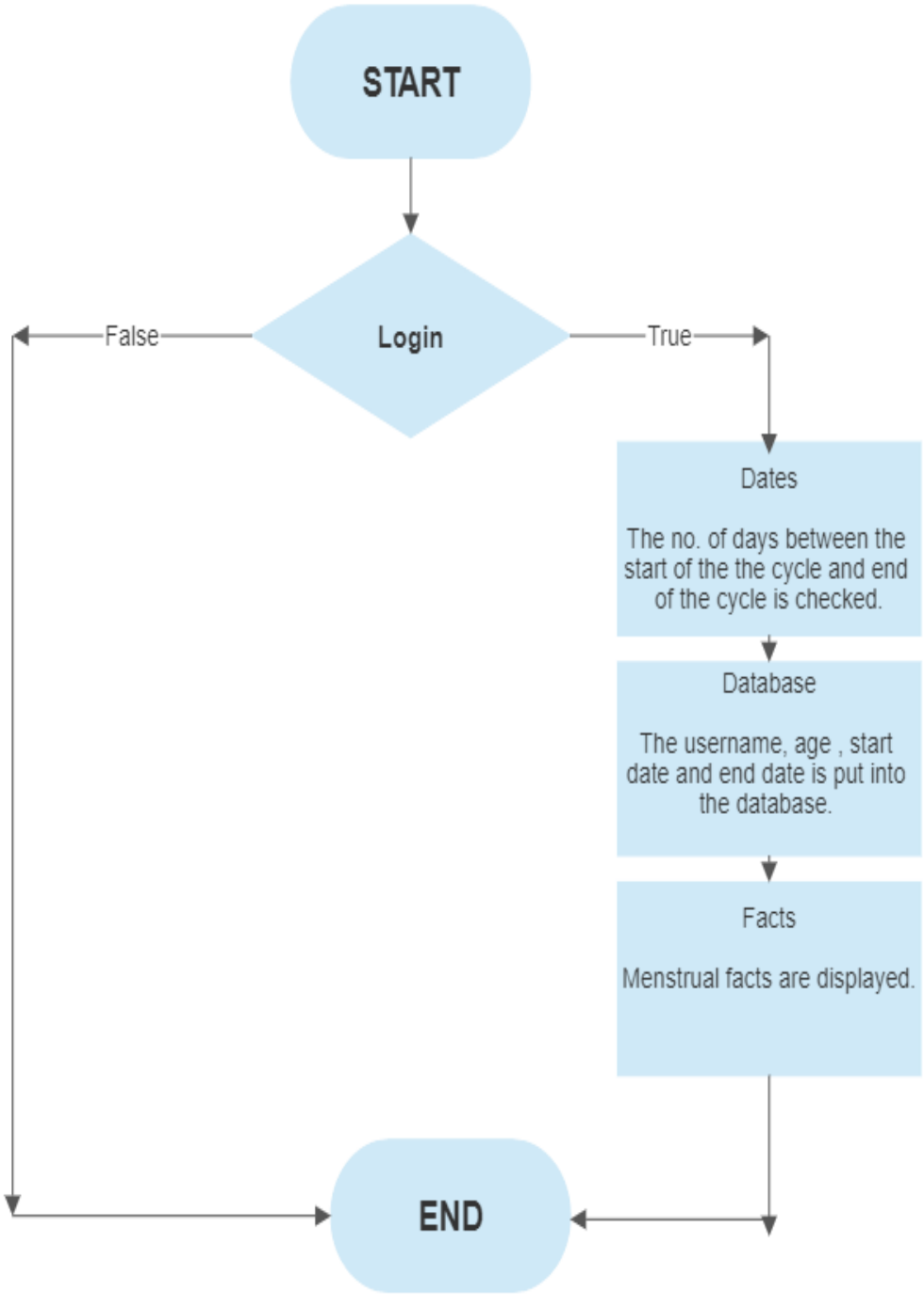
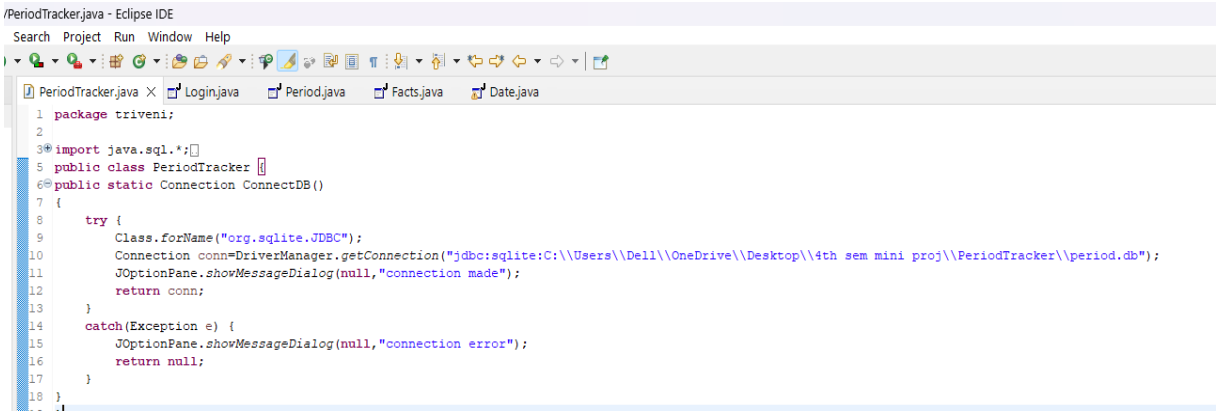


Fig 3.1: Design flowchart

CHAPTER 4

IMPLEMENTATION

4.1 MODULE 1 : CONNECTION




```
package triveni;

import java.sql.*;
public class PeriodTracker {
public static Connection ConnectDB()
{
    try {
        Class.forName("org.sqlite.JDBC");
        Connection conn=DriverManager.getConnection("jdbc:sqlite:C:\\Users\\Dell\\OneDrive\\Desktop\\4th sem mini proj\\PeriodTracker\\period.db");
        JOptionPane.showMessageDialog(null,"connection made");
        return conn;
    }
    catch(Exception e) {
        JOptionPane.showMessageDialog(null,"connection error");
        return null;
    }
}
```

Fig 4.1: Connection module

This module shows the class that helps connect the Period class which consists of the database with the sql connector .

If it isn't connected then it shows a connection error.

4.2 MODULE 2 : LOGIN

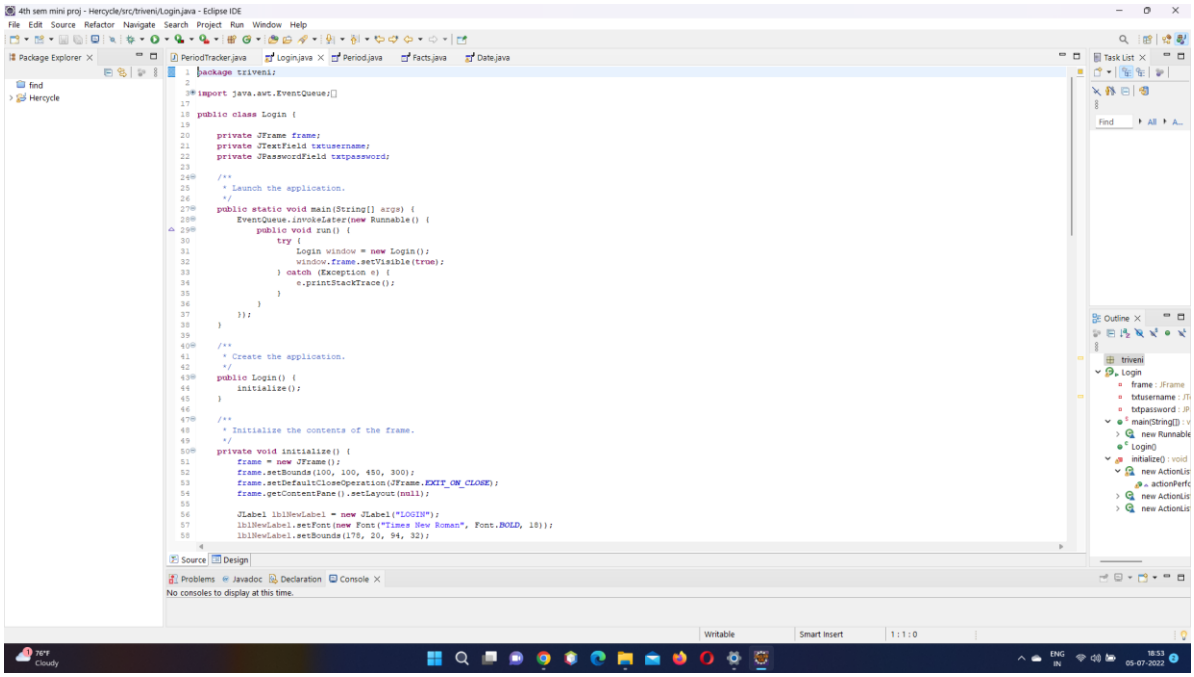


Fig 4.2.1: Login module part 1

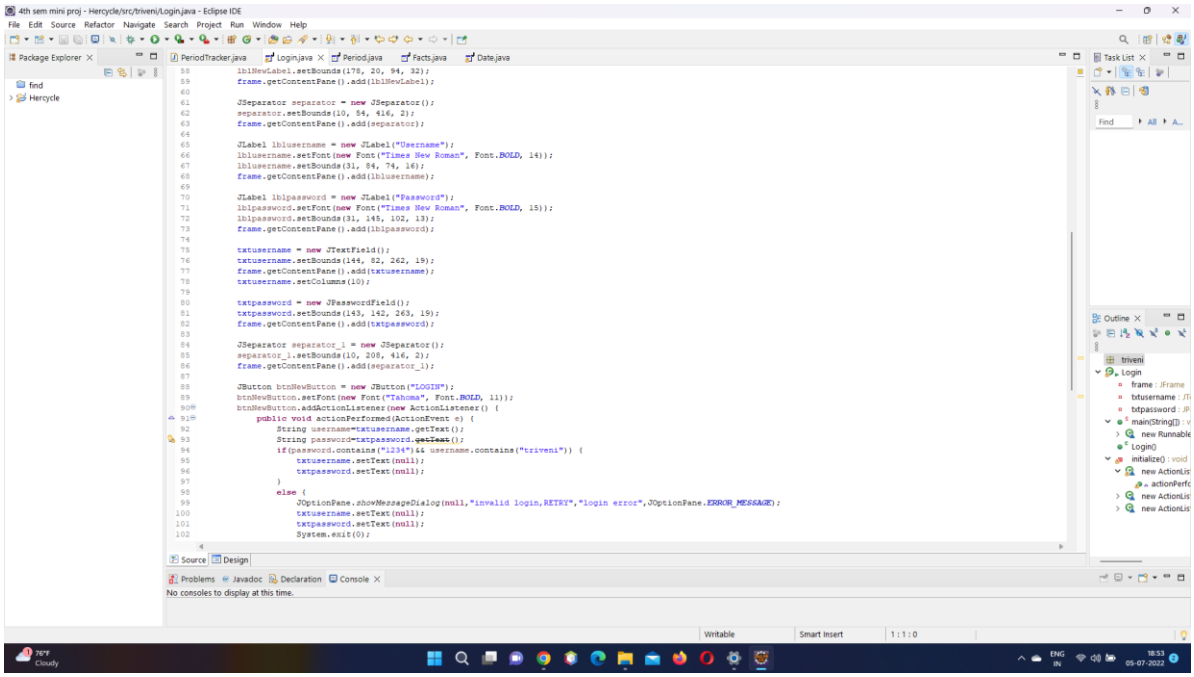


Fig 4.2.2: Login module part 2

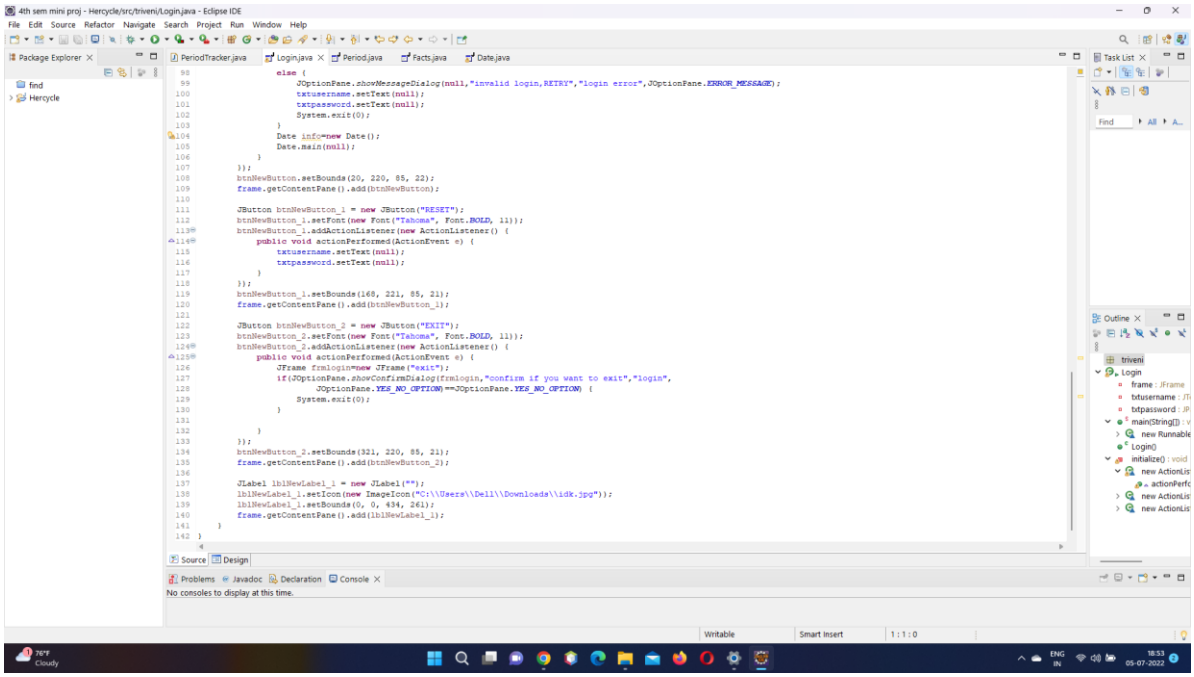


Fig 4.2.3: Login module part 3

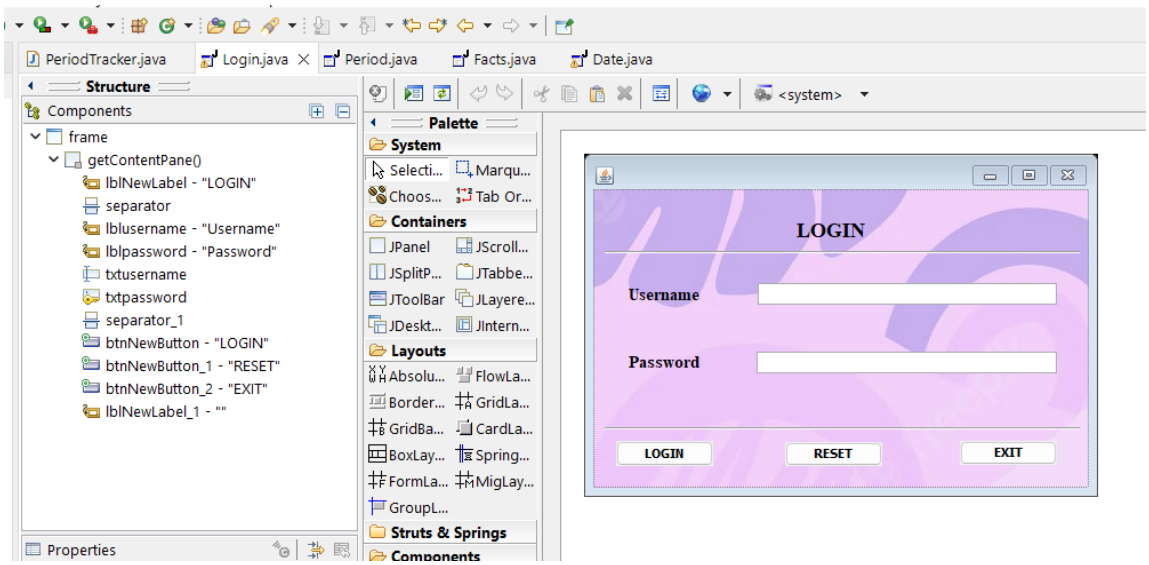
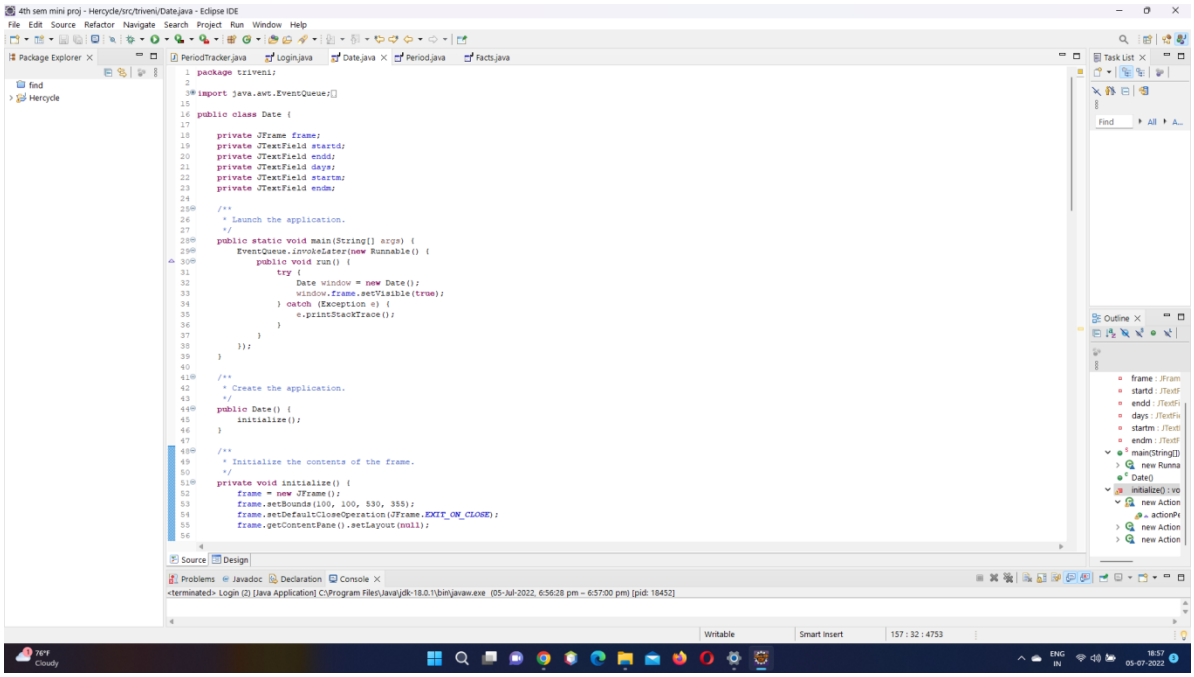


Fig 4.2.4: Login module Jframe

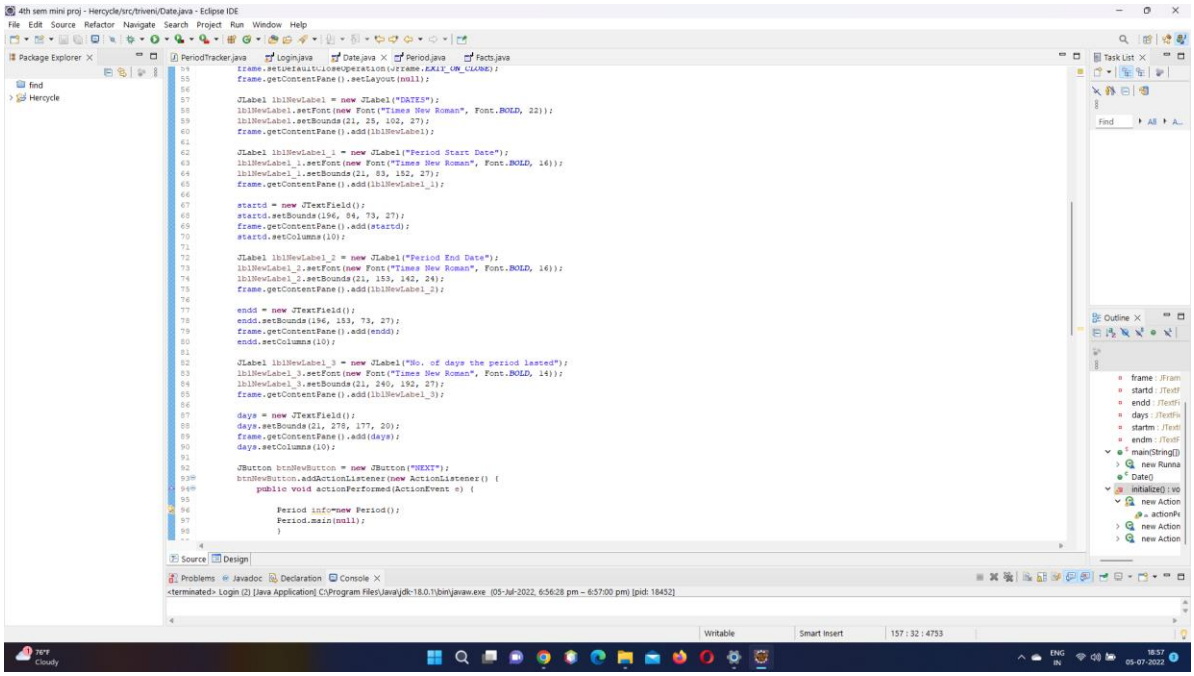
The above module shows the Login page. It takes the username and password from the user and logs in if it is correct. He username in this project is 'triveni' and the password is set as '1234'. If the password is incorrect the user can retry or exit.

4.3 MODULE 3 : DATES CALCULATION



```
package triveni;

import java.awt.EventQueue;

public class Date {

    private JFrame frame;
    private JTextField startd;
    private JTextField endd;
    private JTextField days;
    private JTextField startm;
    private JTextField endm;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Date window = new Date();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public Date() {
        initialize();
    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize() {
        frame = new JFrame();
        frame.setBounds(100, 100, 530, 355);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(null);
```

Fig 4.3.1: Dates module part 1



```
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(null);

        JLabel lblNewLabel = new JLabel("DATES");
        lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 22));
        lblNewLabel.setBounds(21, 25, 102, 27);
        frame.getContentPane().add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("Period Start Date");
        lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD, 16));
        lblNewLabel_1.setBounds(21, 83, 152, 27);
        frame.getContentPane().add(lblNewLabel_1);

        startd = new JTextField();
        startd.setBounds(196, 84, 73, 27);
        frame.getContentPane().add(startd);
        startd.setColumns(10);

        JLabel lblNewLabel_2 = new JLabel("Period End Date");
        lblNewLabel_2.setFont(new Font("Times New Roman", Font.BOLD, 16));
        lblNewLabel_2.setBounds(21, 153, 142, 24);
        frame.getContentPane().add(lblNewLabel_2);

        endd = new JTextField();
        endd.setBounds(196, 153, 73, 27);
        frame.getContentPane().add(endd);
        endd.setColumns(10);

        JLabel lblNewLabel_3 = new JLabel("No. of days the period lasted");
        lblNewLabel_3.setFont(new Font("Times New Roman", Font.BOLD, 14));
        lblNewLabel_3.setBounds(21, 240, 192, 27);
        frame.getContentPane().add(lblNewLabel_3);

        days = new JTextField();
        days.setBounds(21, 278, 177, 20);
        frame.getContentPane().add(days);
        days.setColumns(10);

        JButton btnNewButton = new JButton("NEXT");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                Period info=new Period();
                Period.main(null);
                }
```

Fig 4.3.2: Dates module part 2

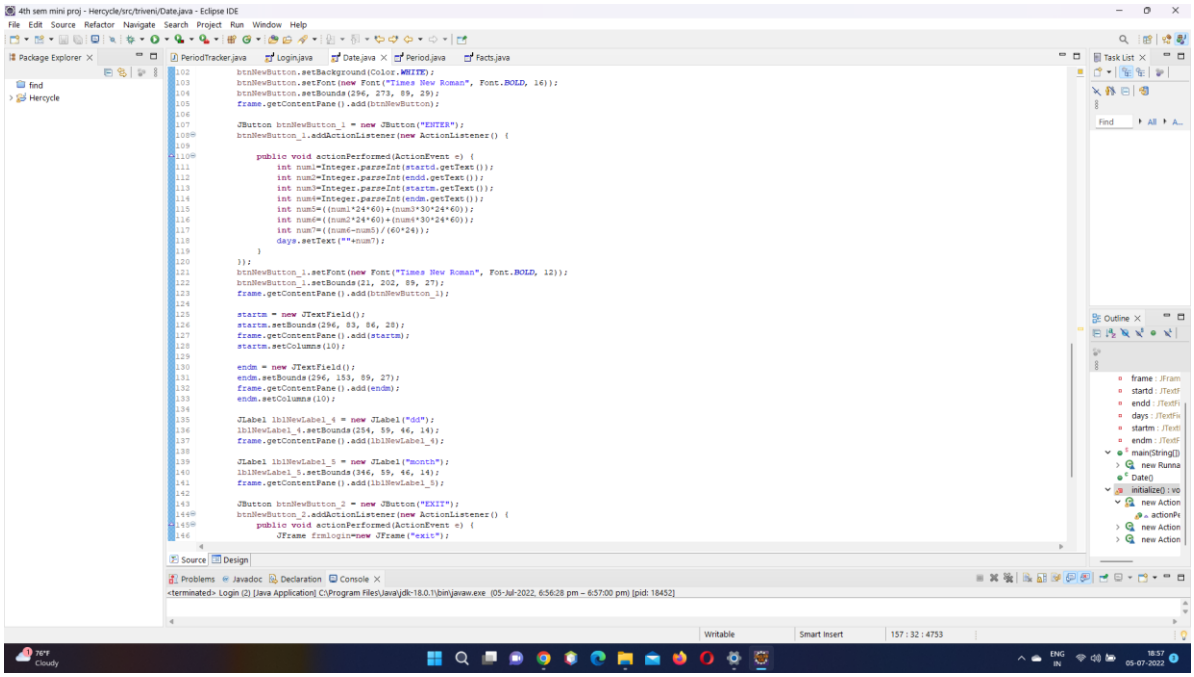


Fig 4.3.3: Dates module part 3

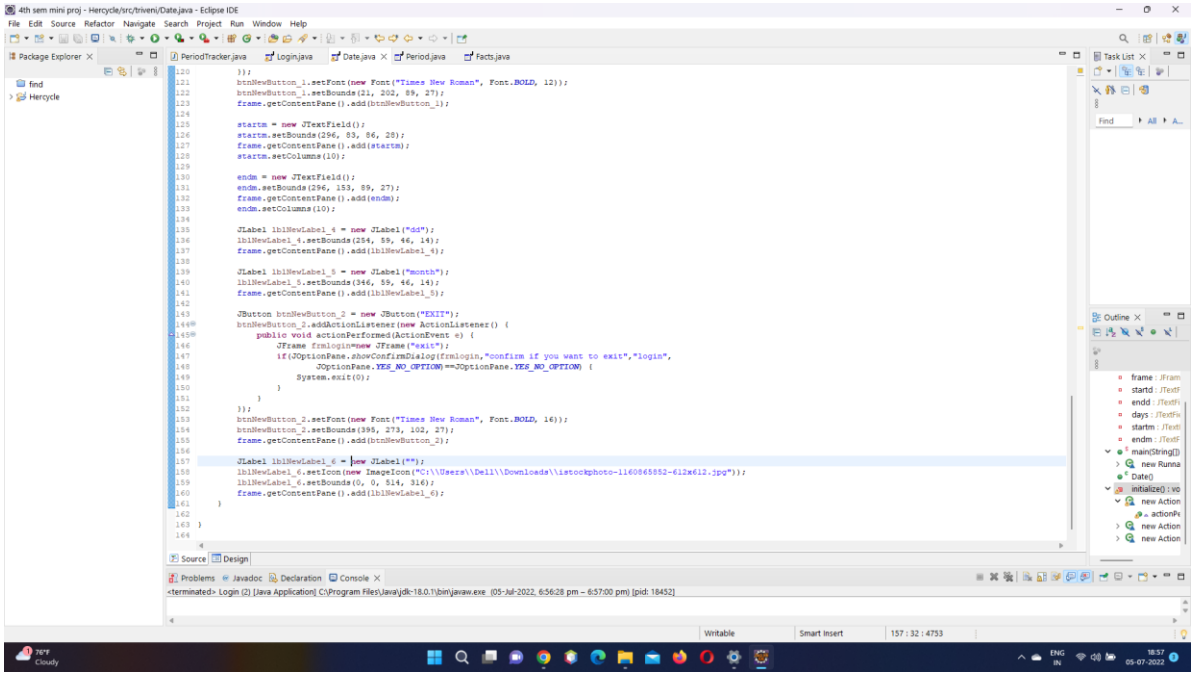


Fig 4.3.4: Dates module part 4

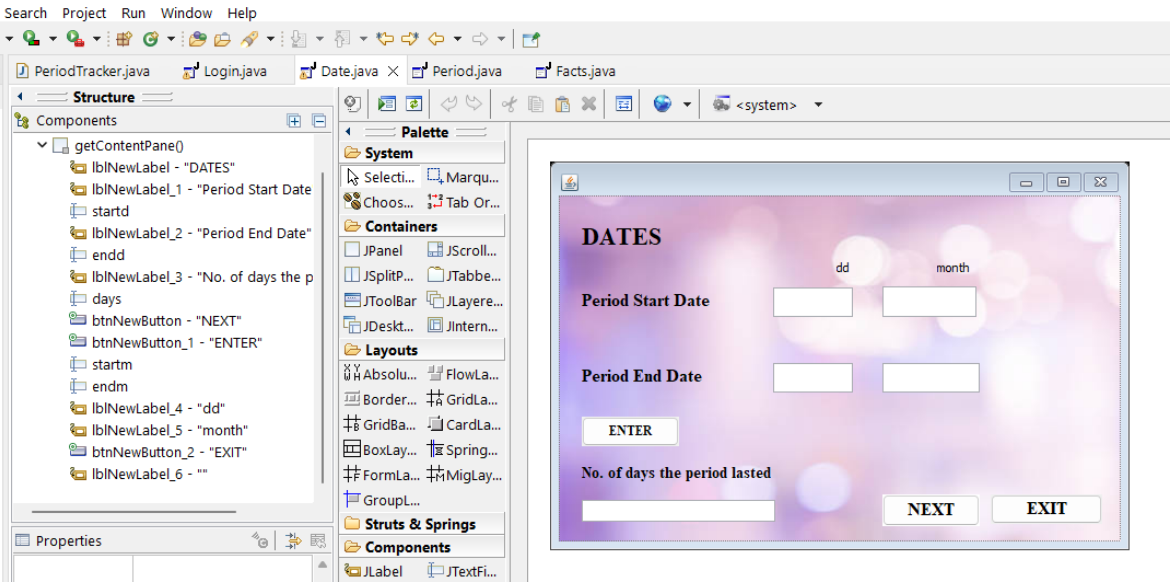
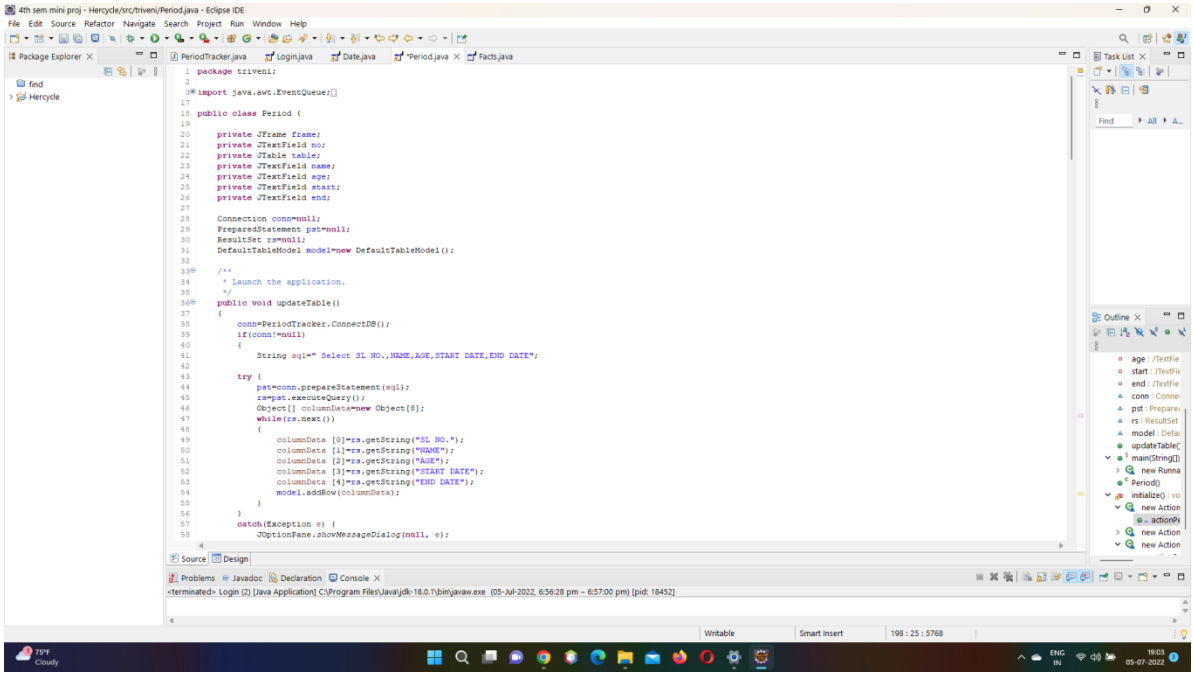


Fig 4.3.5: Dates module Jframe

In this module, the period start date and month is entered and so is the period end date and month entered. The number of days the period lasted is calculated.

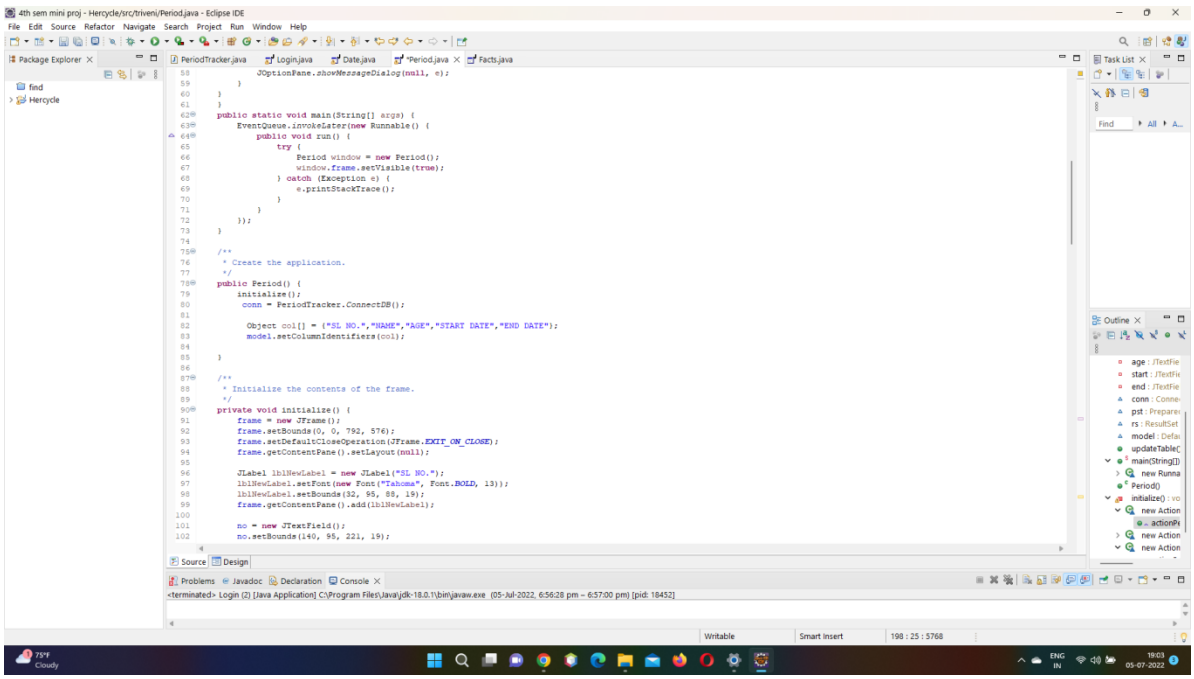
The number of days and months are converted into minutes and the number of days in a month is assumed to be 30 days . The total of minutes end date is subtracted with the start date and is converted back into days. The number of days the period lasted is hence printed.

4.4 MODULE 4 : DATABASE



```
package triveni;

import java.awt.EventQueue;

public class Period {

    private JFrame frame;
    private JTextField no;
    private JTable table;
    private JTextField name;
    private JTextField age;
    private JTextField start;
    private JTextField end;

    Connection conn=null;
    PreparedStatement pst=null;
    ResultSet rs=null;
    DefaultTableModel model=new DefaultTableModel();

    /**
     * Launch the application.
     */
    public void updateTable()
    {
        conn=PeriodTracker.ConnectDB();
        if(conn!=null)
        {
            String sql=" Select SL NO.,NAME,AGE,START DATE,END DATE";

        try {
            pst=conn.prepareStatement(sql);
            rs=pst.executeQuery();
            Object[] columnData=new Object[8];
            while(rs.next())
            {
                columnData [0]=rs.getString("SL NO.");
                columnData [1]=rs.getString("NAME");
                columnData [2]=rs.getString("AGE");
                columnData [3]=rs.getString("START DATE");
                columnData [4]=rs.getString("END DATE");
                model.addRow(columnData);
            }
        }
        catch(Exception e) {
            JOptionPane.showMessageDialog(null, e);
```

Fig 4.4.1: Database module part 1



```
            JOptionPane.showMessageDialog(null, e);
        }
    }
    }
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Period window = new Period();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public Period() {
        initialize();
         conn = PeriodTracker.ConnectDB();

         Object col[] = {"SL NO.","NAME","AGE","START DATE","END DATE"};
         model.setColumnIdentifiers(col);

    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize() {
        frame = new JFrame();
        frame.setBounds(0, 0, 792, 576);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(null);

        JLabel lblNewLabel = new JLabel("SL NO.");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 13));
        lblNewLabel.setBounds(32, 95, 88, 19);
        frame.getContentPane().add(lblNewLabel);

        no = new JTextField();
        no.setBounds(140, 95, 221, 19);
```

Fig 4.4.2: Database module part 2

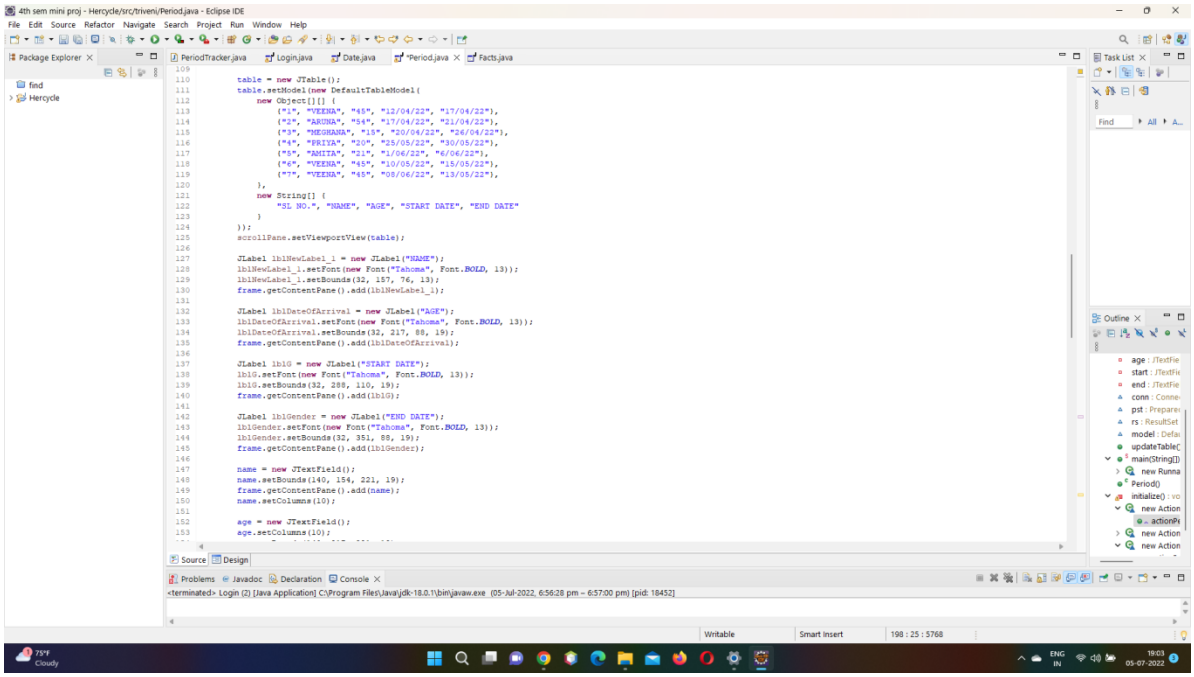


Fig 4.4.3: Database module part 3

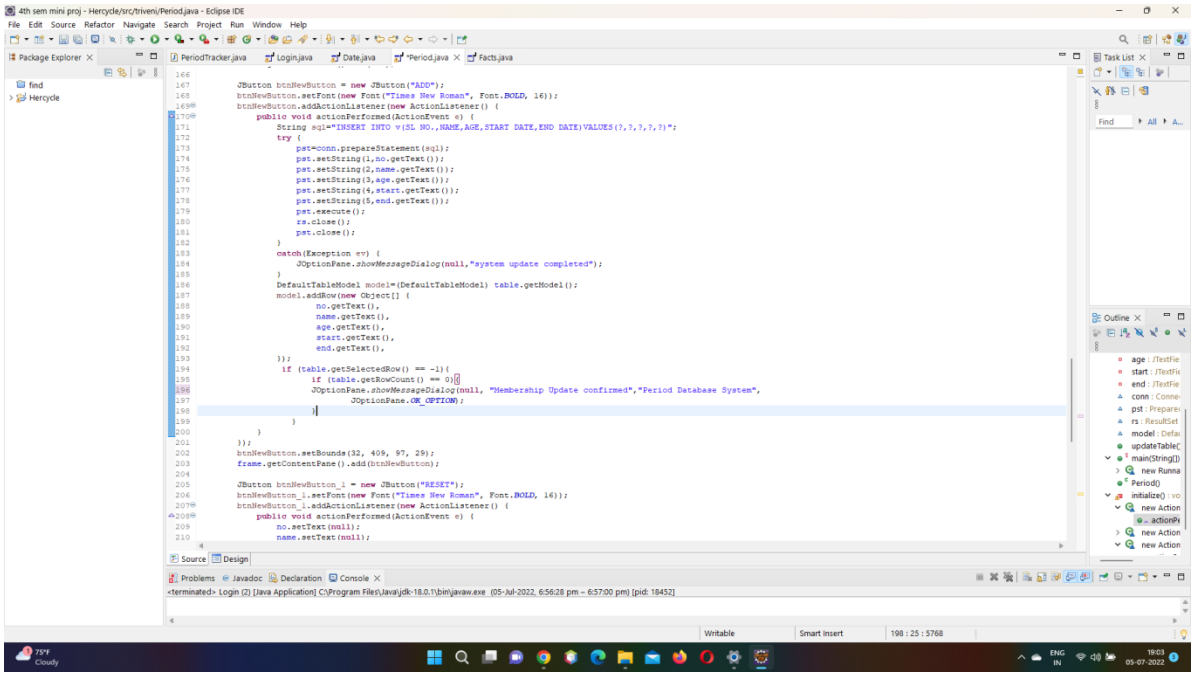


Fig 4.4.4: Database module part 4

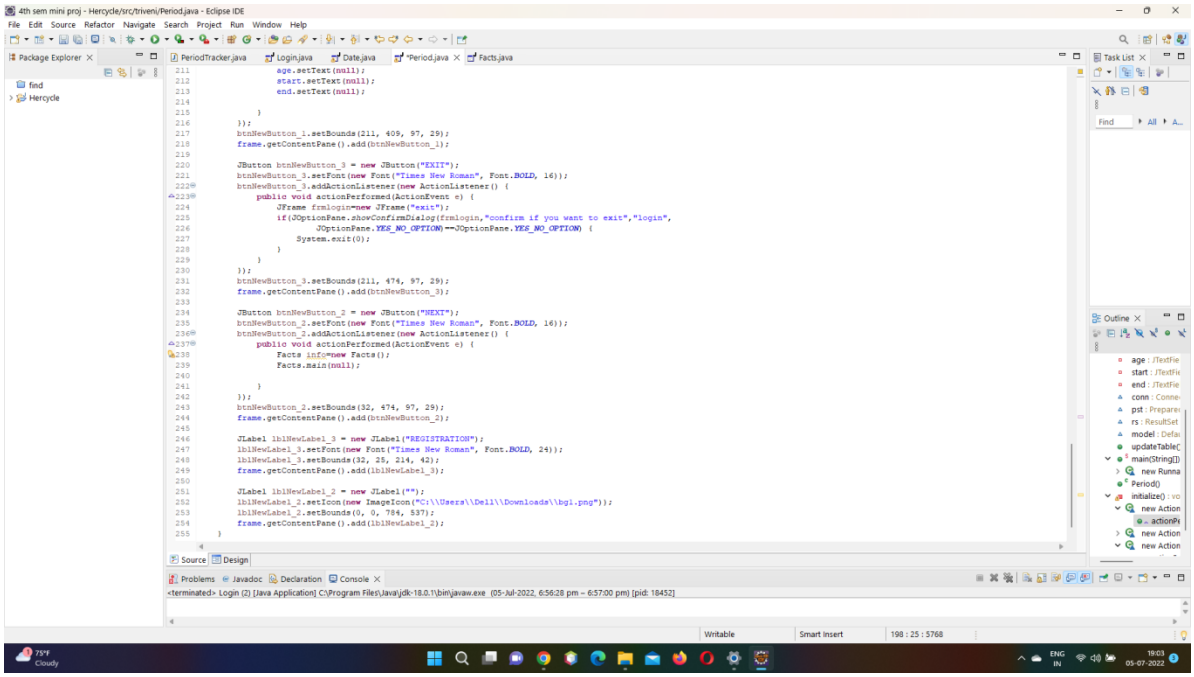


Fig 4.4.5: Database module part 5

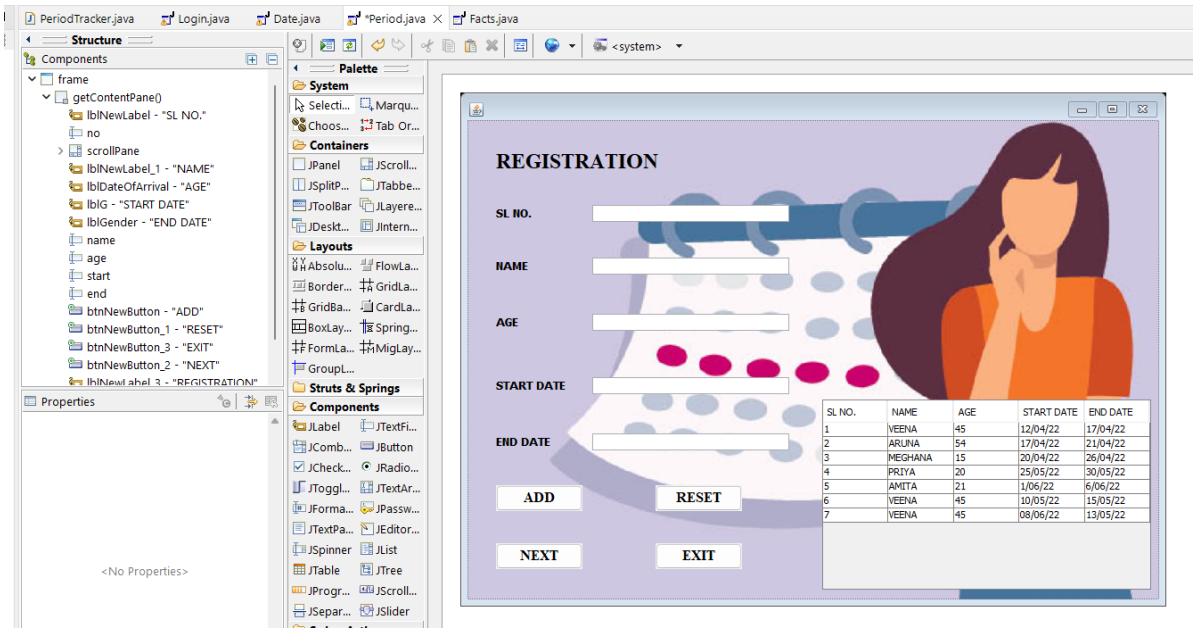
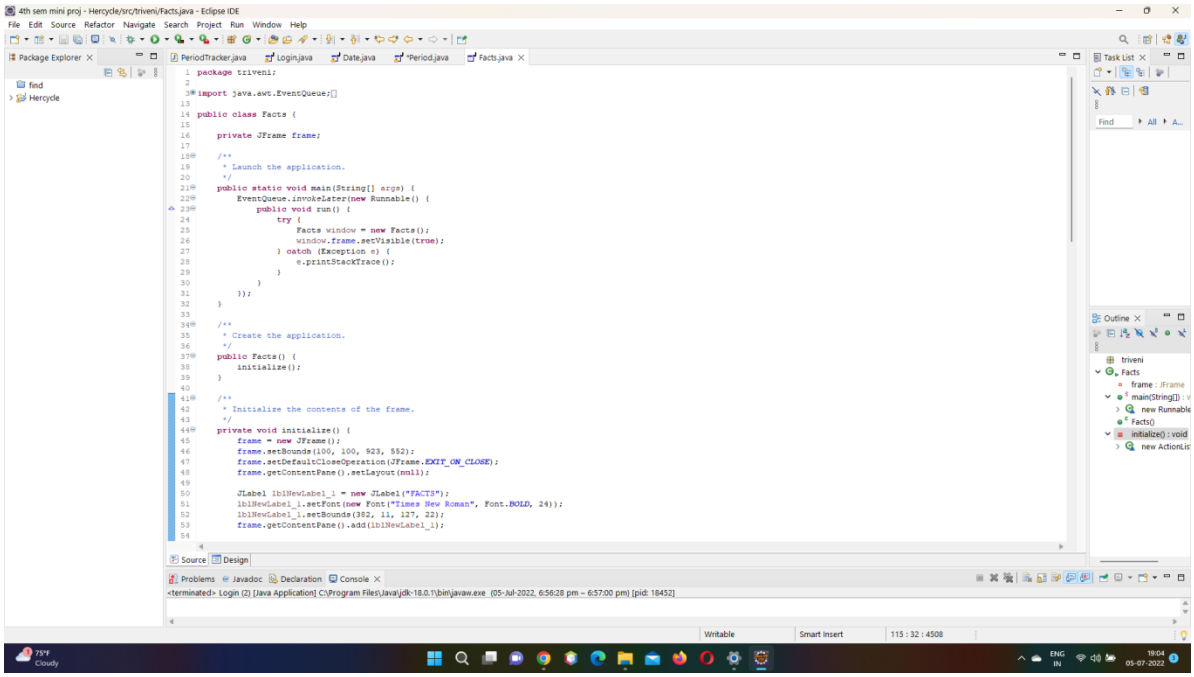


Fig 4.4.6: Database module JFrame

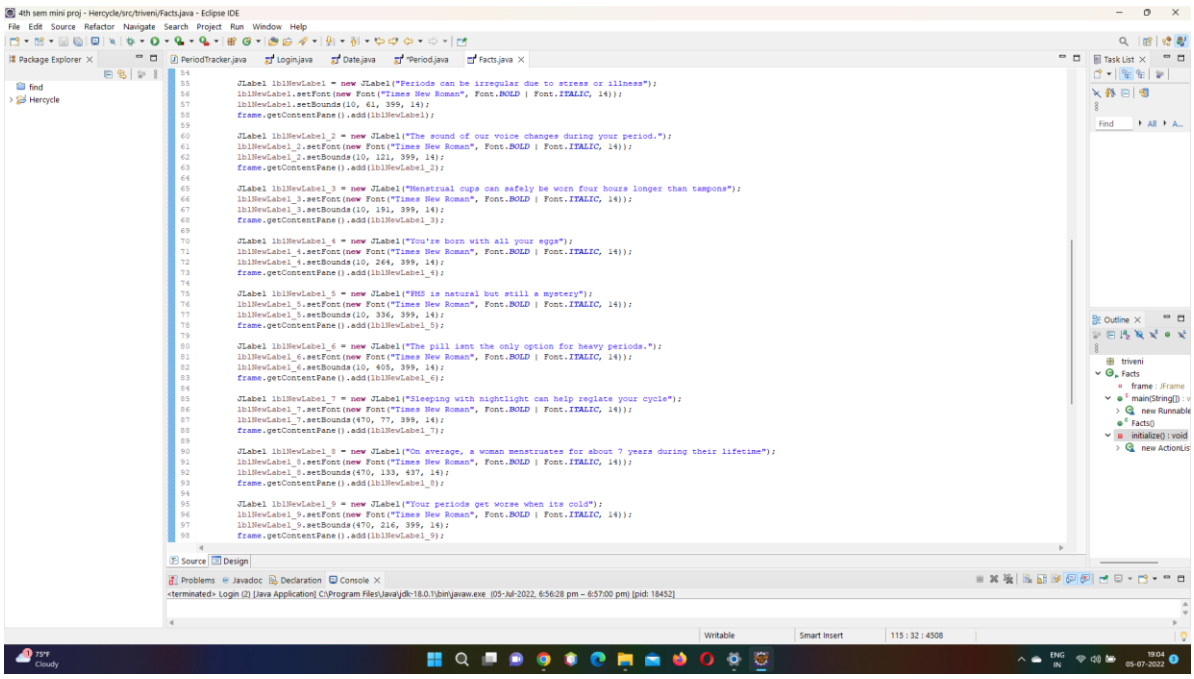
In this module, user will be asked to enter all the details which include name, age , period start date and period end date. This helps the user keep track of not only their cycle but also cycles of their close ones. By clicking on the add button the data entered gets added to the table . Another entry can be added by using the reset button. Hence, multiple entries can be stored under one logged in id.

4.5 MODULE 5 : FACTS



```
package triveni;

import java.awt.EventQueue;

public class Facts {

    private JFrame frame;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Facts window = new Facts();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public Facts() {
        initialize();
    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize() {
        frame = new JFrame();
        frame.setBounds(100, 100, 923, 552);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(null);

        JLabel lblNewLabel_1 = new JLabel("FACTS");
        lblNewLabel_1.setFont(new Font("Times New Roman", Font.BOLD, 24));
        lblNewLabel_1.setBounds(382, 11, 127, 22);
        frame.getContentPane().add(lblNewLabel_1);
```

Fig 4.5.1: Facts module part 1



```
        JLabel lblNewLabel = new JLabel("Periods can be irregular due to stress or illness");
        lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD | Font.ITALIC, 14));
        lblNewLabel.setBounds(10, 61, 399, 14);
        frame.getContentPane().add(lblNewLabel);

        JLabel lblNewLabel_2 = new JLabel("The sound of our voice changes during your period.");
        lblNewLabel_2.setFont(new Font("Times New Roman", Font.BOLD | Font.ITALIC, 14));
        lblNewLabel_2.setBounds(10, 121, 399, 14);
        frame.getContentPane().add(lblNewLabel_2);

        JLabel lblNewLabel_3 = new JLabel("Menstrual cups can safely be worn four hours longer than tampons");
        lblNewLabel_3.setFont(new Font("Times New Roman", Font.BOLD | Font.ITALIC, 14));
        lblNewLabel_3.setBounds(10, 191, 399, 14);
        frame.getContentPane().add(lblNewLabel_3);

        JLabel lblNewLabel_4 = new JLabel("You're born with all your eggs");
        lblNewLabel_4.setFont(new Font("Times New Roman", Font.BOLD | Font.ITALIC, 14));
        lblNewLabel_4.setBounds(10, 264, 399, 14);
        frame.getContentPane().add(lblNewLabel_4);

        JLabel lblNewLabel_5 = new JLabel("PMS is natural but still a mystery");
        lblNewLabel_5.setFont(new Font("Times New Roman", Font.BOLD | Font.ITALIC, 14));
        lblNewLabel_5.setBounds(10, 336, 399, 14);
        frame.getContentPane().add(lblNewLabel_5);

        JLabel lblNewLabel_6 = new JLabel("The pill isnt the only option for heavy periods.");
        lblNewLabel_6.setFont(new Font("Times New Roman", Font.BOLD | Font.ITALIC, 14));
        lblNewLabel_6.setBounds(10, 405, 399, 14);
        frame.getContentPane().add(lblNewLabel_6);

        JLabel lblNewLabel_7 = new JLabel("Sleeping with nightlight can help reglate your cycle");
        lblNewLabel_7.setFont(new Font("Times New Roman", Font.BOLD | Font.ITALIC, 14));
        lblNewLabel_7.setBounds(470, 77, 399, 14);
        frame.getContentPane().add(lblNewLabel_7);

        JLabel lblNewLabel_8 = new JLabel("On average, a woman menstruates for about 7 years during their lifetime");
        lblNewLabel_8.setFont(new Font("Times New Roman", Font.BOLD | Font.ITALIC, 14));
        lblNewLabel_8.setBounds(470, 133, 437, 14);
        frame.getContentPane().add(lblNewLabel_8);

        JLabel lblNewLabel_9 = new JLabel("Your periods get worse when its cold");
        lblNewLabel_9.setFont(new Font("Times New Roman", Font.BOLD | Font.ITALIC, 14));
        lblNewLabel_9.setBounds(470, 216, 399, 14);
        frame.getContentPane().add(lblNewLabel_9);
```

Fig 4.5.2: Facts module part 2

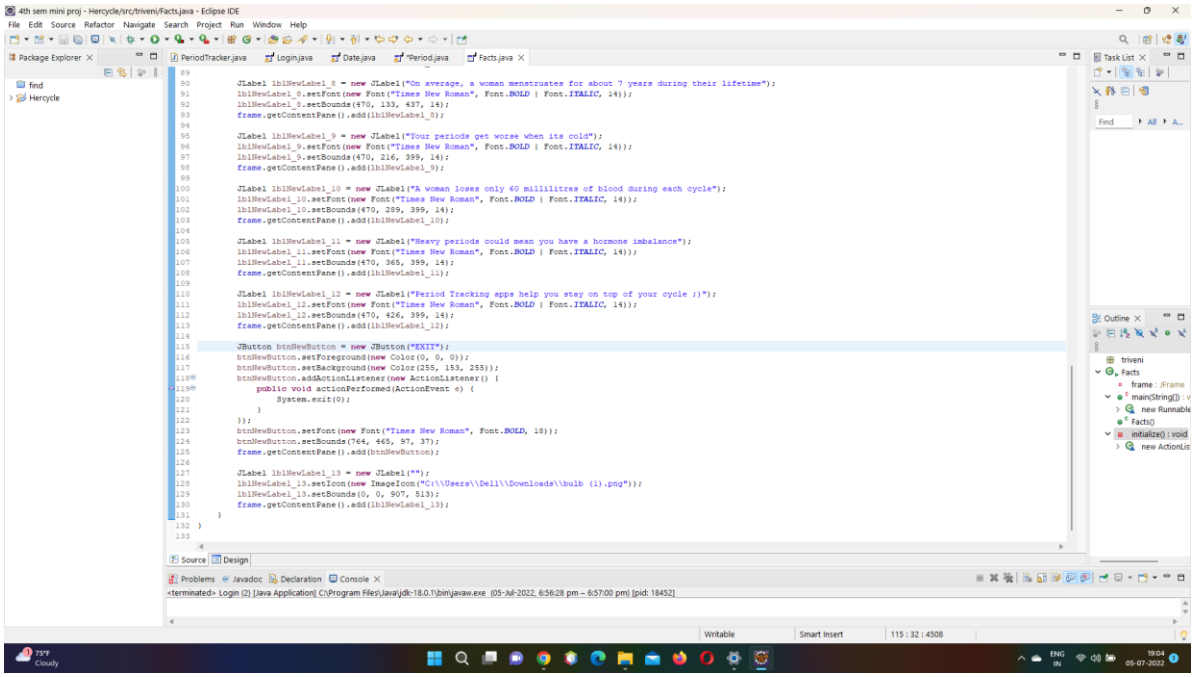


Fig 4.5.3: Facts module part 3

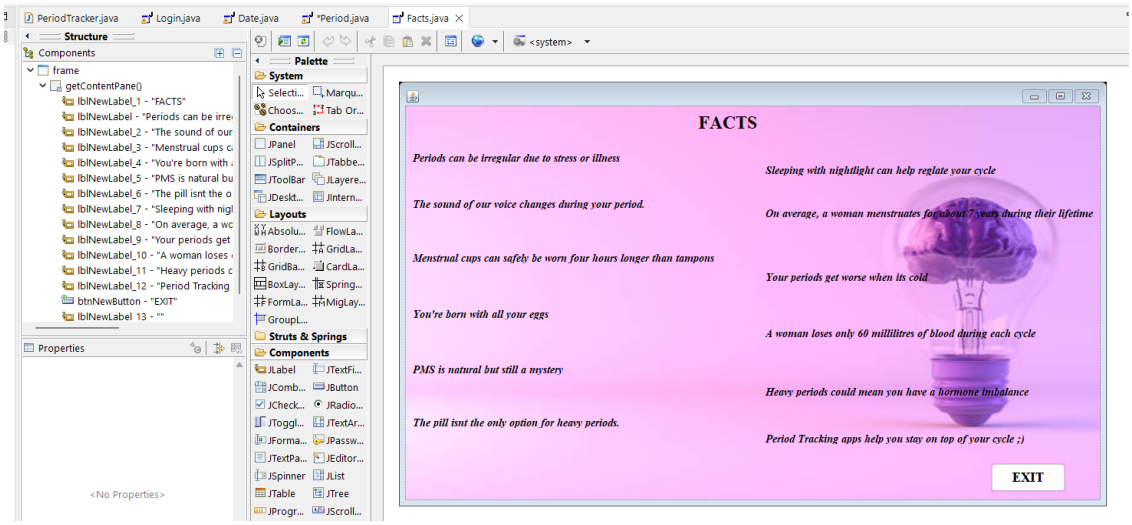


Fig 4.5.4: Facts module Jframe

This module displays interesting menstrual facts for the user to gain knowledge and perspective from.

CHAPTER 5

RESULTS

5.1 Snapshots

Fig 5.1.1: Login result

Here user account is created and he username is “triveni” and the password is “1234” , apart from that any other password will not allow the user to log in.

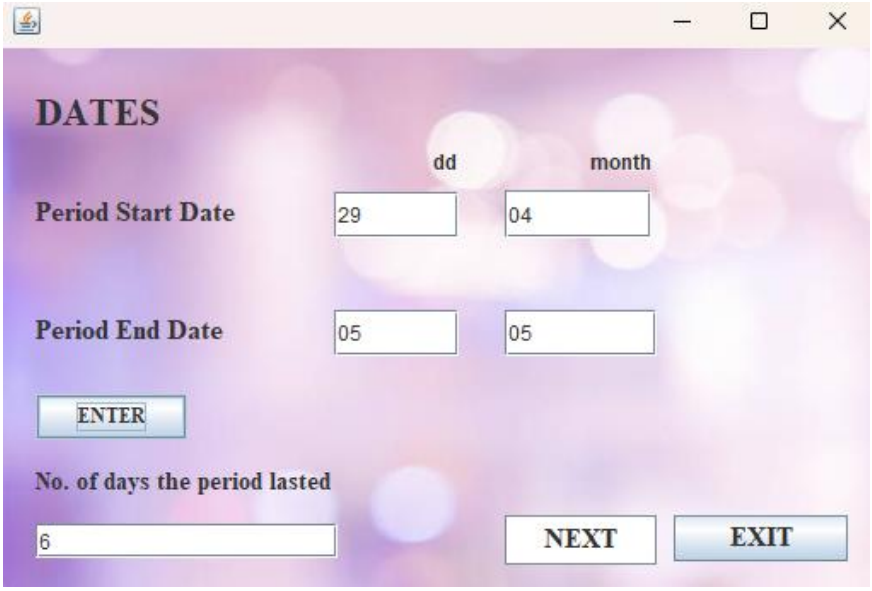


Fig 5.1.2: Dates result

Here we can enter the start date and start month in dd / mm format that takes the data as input . Simlarly the end date and end month is taken in the dd / mm format is also taken as the input. On clicking on the 'Enter' button the number of days your period lasted will be displayed.

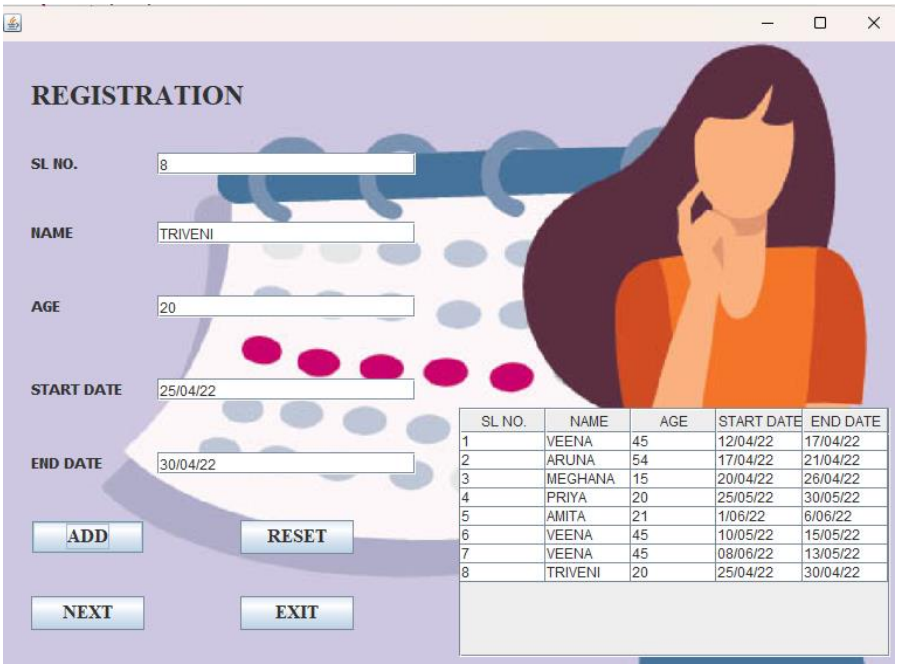


Fig 5.1.3: Database result while add

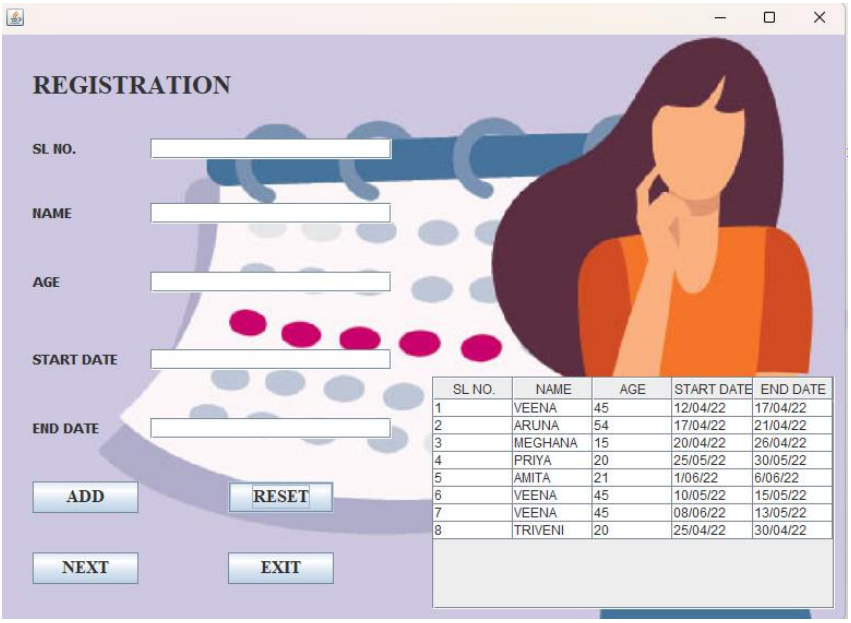


Fig 5.1.4: Database result while reset

In the registration page, the user inputs the sl no., name , age , start date and end date . This helps the user keep track of her periods . The user can also keep track of close one's periods if required.

Fig 5.1.5: Facts result

The facts page shows interesting menstrual facts.

CHAPTER 6

CONCLUSION

The mini project has successfully accomplished the goals it had set out in the objectives and design sections of this report.

This mini project helps women tack their periods such that they notice if changes happen or can plan their schedule according to the dates. It helps women with PCOS also track their cycles though the cycle gap will vary a lot.

It also provides interesting facts on menstruation for the user.

REFERENCES

[1] https://www.javatpoint.com/

[2] https://www.programiz.com/java-programming/exception-handling

[3] https://simplesnippets.tech/inheritance-in-java-types-of-inheritance/

[4]https://www.simplilearn.com/tutorials/java-tutorial/thread-in-java#:~:text=What%20is%20a%20Thread%20in,starting%20of%20the%20program's%20execution.

[5] https://www.geeksforgeeks.org/java-io-input-output-in-java-with-examples/