DBMS LAB MINI PROJECT

Feature Film Database

By

Akhil V. Rno.38
Gowtham Rno.43
Ratnakar M. Rno.47

ABSTRACT: FEATURE FILM DATABASE

Objective:

The film industry today is larger than it ever was, as a result it is more beneficial than ever to store records of information pertaining to films. Records of films must be kept as they, like any other art form are of great social and cultural significance. Hence steps must be taken to store records and a database design that is tasked with doing so must contain all the relevant information represented in an efficient manner without being burdened with irrelevant trivia.

Features: The application allows information retrieval from database.

The database stores the Film Title, Genre, Year, MCPAA Rating, synopsis and main cast and crew.

Provision for obtaining recommendations for other films with shared attributes is also present. For example, films that have been directed by the same person,etc. can be

looked up.

Values for each of these attributes can be associated with multiple films, however each film can have at most one value for each of them. Each film is associated with a film ID to uniquely identify it. .

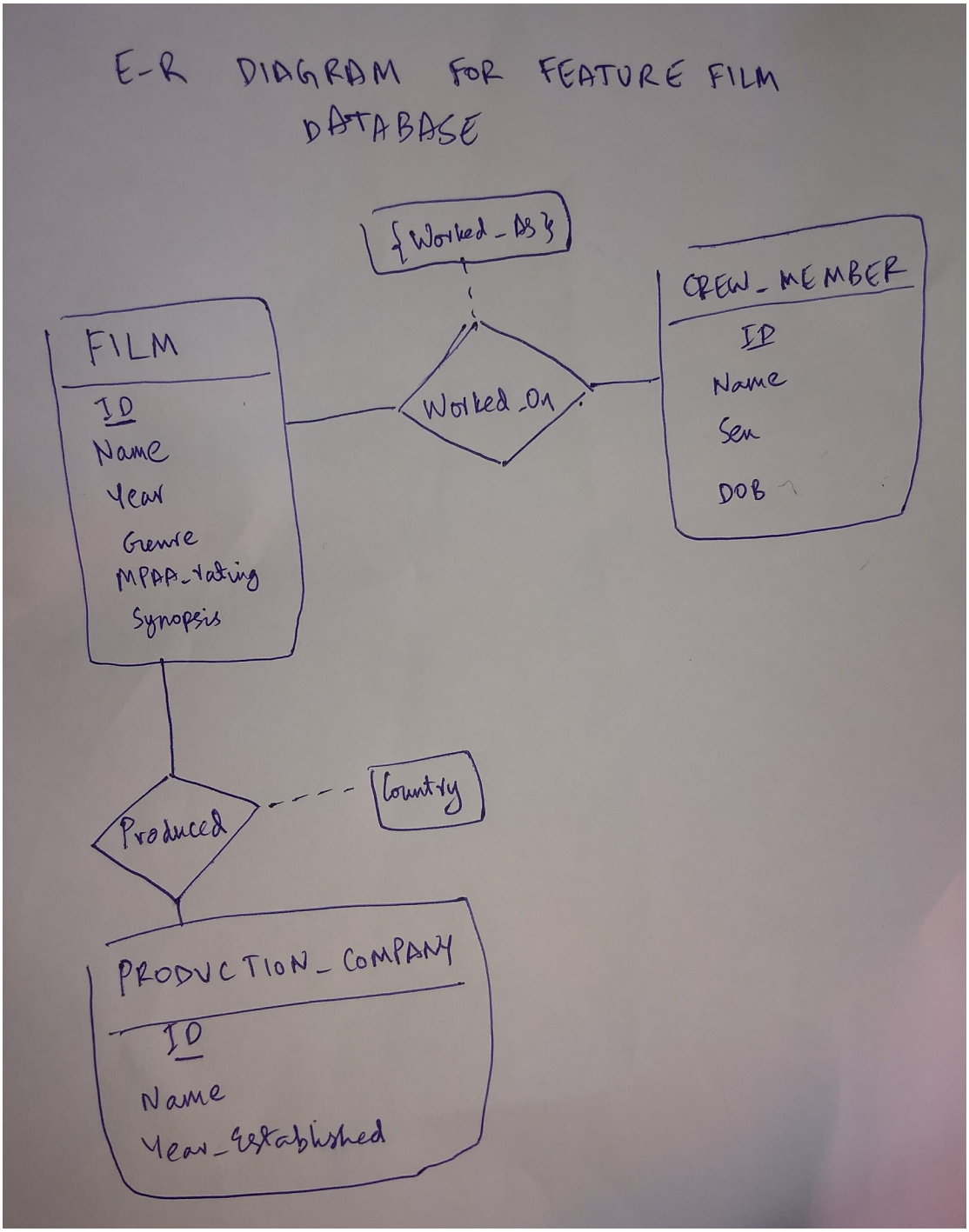
Languages used:

SQL for database creation and modification.

Java for user interface.

JDBC for Connecting with the database.

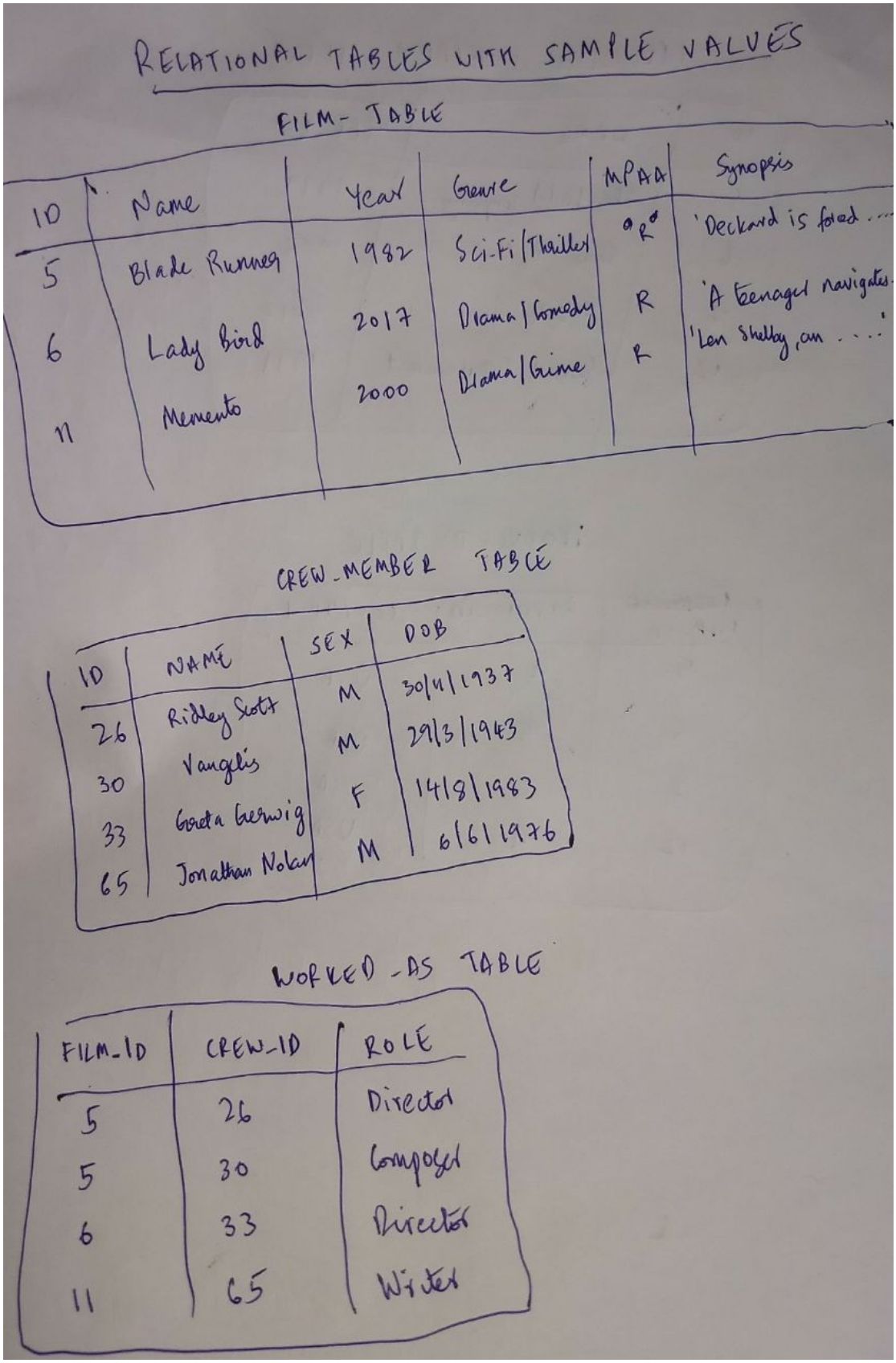
ER DIAGRAM FOR THE FEATURE FILM DATABASE:



DDL Database Creation Commands:

```
create table Film(
film_id        varchar(10)  primary key,
film_name      varchar(50)  not null,
year      numeric(4,0),
genre   varchar(15),
MPAA_rating   varchar(5) check (rating in ('G','PG','PG-13','R','NC-17')),
synopsis  varchar(2000)
);

create table Crew_Member(
crew_id        varchar(10)   primary key,
crew_name      varchar(50) not null,
sex       char(1) not null check (sex in ('M','F')),
dob      date
);

create table Worked_As(
film_id varchar(10),
crew_id varchar(10),
role                varchar(20)                check                (role                in
('Director','Writer','Composer','Editor','Cinematographer','Actor')),
foreign key(film_id) references Film(film_id),
foreign key(crew_id) references Crew_Member(crew_id),
primary key(film_id,crew_id,role)
);

create table Production_Company(
studio_id        varchar(10) primary key,
prod_name     varchar(50) not null,
year_established int
);

create table Produced(
film_id varchar(10),
studio_id varchar(10),
country varchar(20),
foreign key(film_id) references Film(film_id),
foreign key(studio_id) references Production_Company(studio_id),
primary key(film_id,studio_id)
);
```

RELATIONAL TABLES WITH SAMPLE DATA



PRODUCTION_COMPANY TABLE

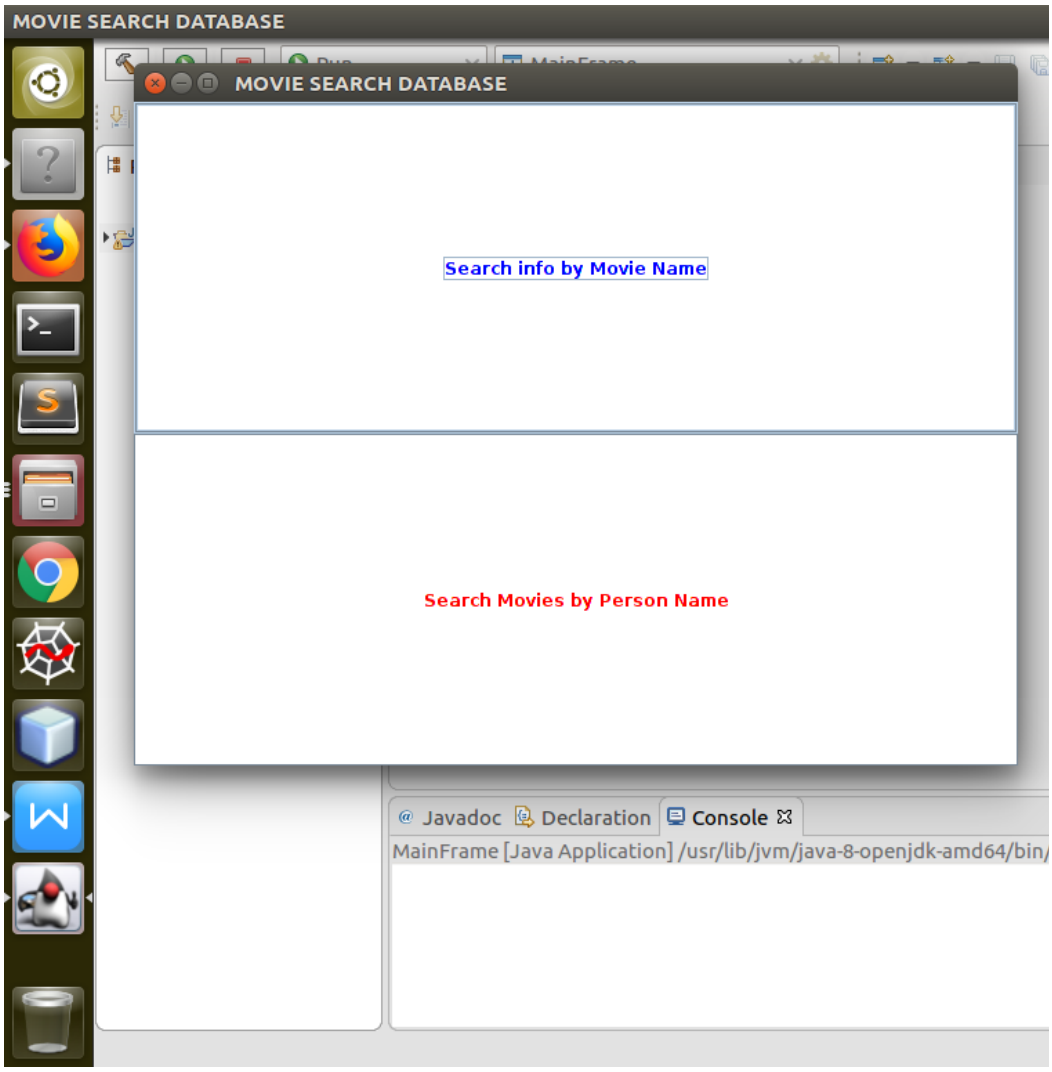
ID	NAME	YEAR_EST
5	The Ladd Company	1979
6	Shaw Brothers	1958
7	A24	2012
12	Summit Entertainment	1991

PRODUCED TABLE

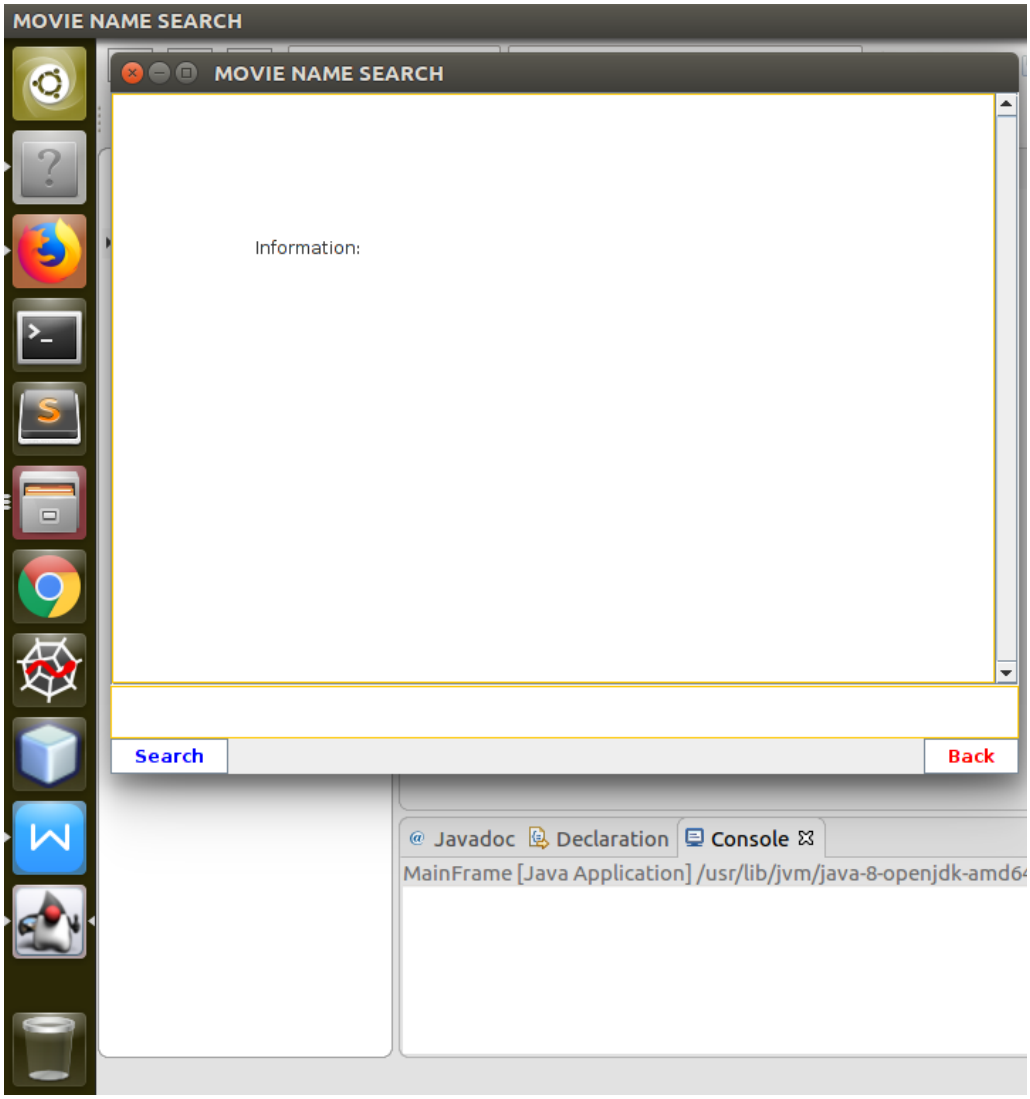
~~FILM_ID~~ FILM_ID	STUDIO_ID	COUNTRY
5	5	USA
6	5	USA
7	6	USA
12	11	USA

UI DESIGN:

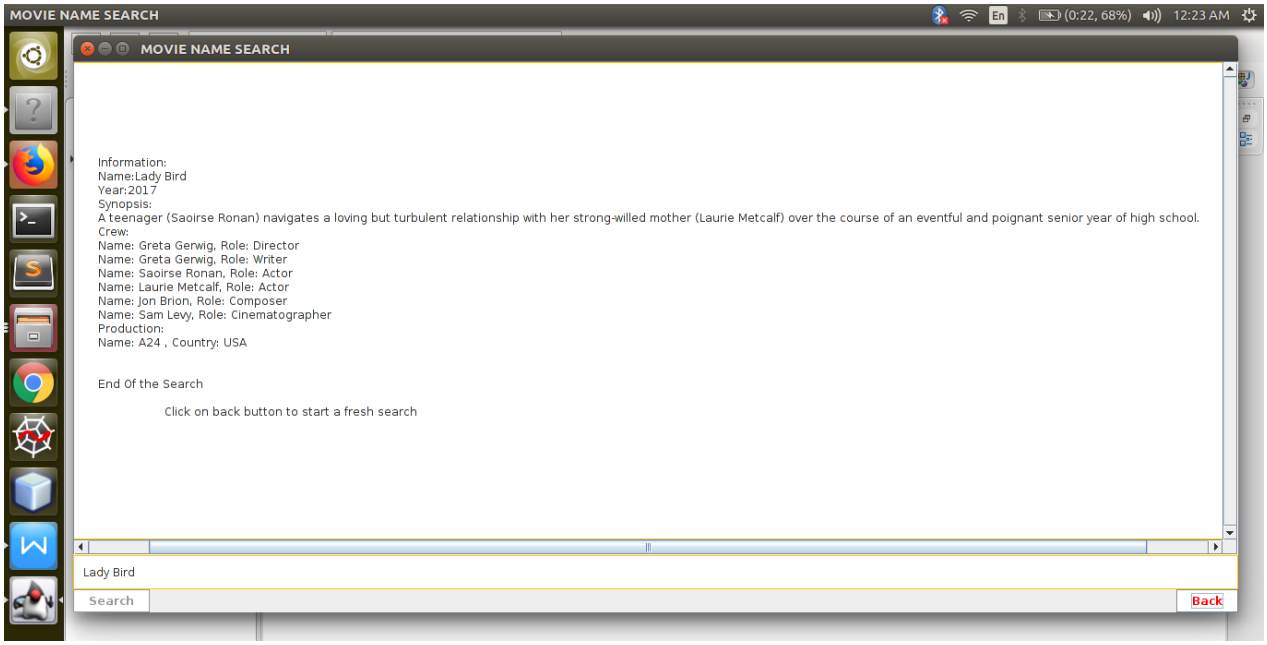
The code as shown has bee divided into many classes ,in order to achieve proper abstraction and ease while coding.
THE UI SCREENSHOTS HAVE BEEN ATTACHED BELOW



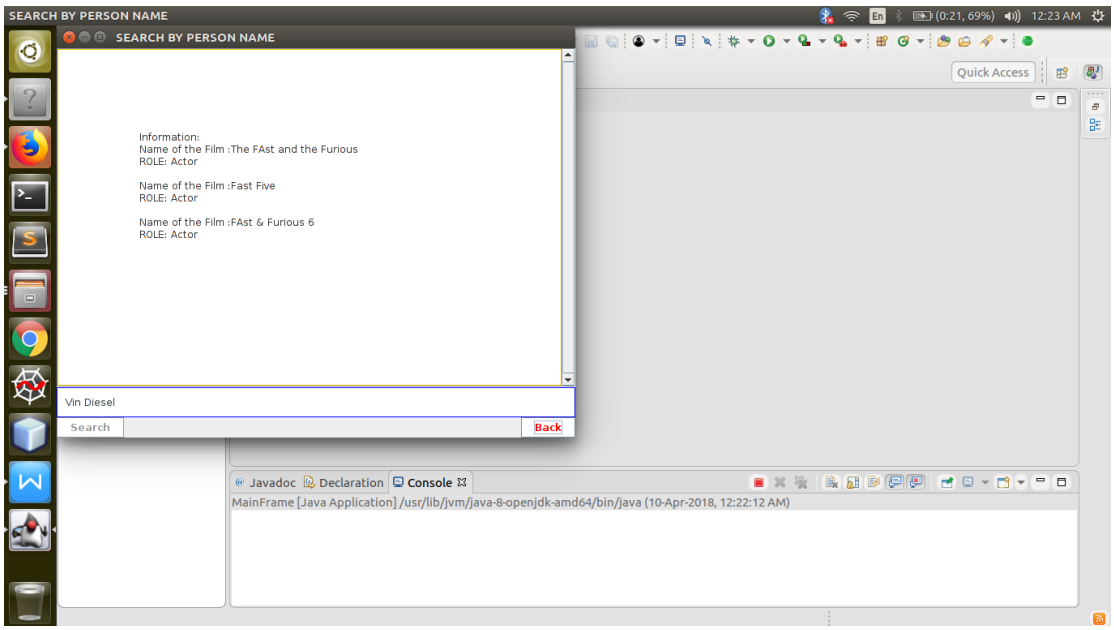
THE APPLICATION STARTS AND DISPLAYS TWO OPTIONS.
ONE, SEARCH BY MOVIE NAME
AND ANOTHER, SEARCH BY PERSON NAME
(ONLY MOUSE CLICKS ARE ALLOWED)
ON CLICKING THE SEARCH BY MOVIE NAME OPTION ,the window as Shown below,pops up



We type a movie's name in the search box,in order to get the information about that movie.



It displays the information.



Then There is search by person too,

Where on typing a technician’s name,the list of movies that he/she acted ,are shown.

QUERIES USED:

The Queries used are:

```
select film_name,year,synopsis from Film where film_name ='"+Buffer+"'".
```

Where Buffer is the Film’s name saved in a Variable named Buffer.

```
select crew_name,role from Crew_Member natural join Film natural join Worked_As where film_name='"+Buffer+"'";
```

Here,Buffer is the Film’s name.

```
select prod_name,country from Produced natural join Production_Company natural join Film where film_name ='"+Buffer+"'";
```

Where,Buffer stores film’s name once more

```
select film_name,role from Crew_Member natural join Film natural join Worked_As where crew_name='"+Buffer+"'";
```

FUNCTIONAL DESIGN CODE (IN JAVA):

```
import javax.swing.SwingUtilities;
public class MainFrame {
	public static void main(String[] args) {
		SwingUtilities.invokeLater(new Runnable()
		{
		public void run()
		{
			Opening_Interface o = new Opening_Interface();
		}
		});
	}
}
```

```
public class MSPanel extends JPanel {
      String result1 = "error";
      String Buffer;
      JButton search = new JButton("Search");
      JButton back   = new JButton("Back");
      JTextArea text = new JTextArea();
      Border border = BorderFactory.createLineBorder(Color.ORANGE);
      public MSPanel()
      {setLayout(new BorderLayout());
      text.setBorder(BorderFactory.createCompoundBorder(border,
      BorderFactory.createEmptyBorder(10, 10, 10, 10)));
                  add(text,BorderLayout.NORTH);
                  add(back,BorderLayout.EAST);
                  add(search,BorderLayout.WEST);
                  back.setForeground(Color.RED);
                  search.setForeground(Color.BLUE);
                  back.setBackground(Color.WHITE);
                  search.setBackground(Color.WHITE);
      }
}

public class TextPanel extends JPanel {
      JTextArea text1 = new JTextArea("Information:");
      public TextPanel()
      {
       setLayout(new BorderLayout());
       add(text1,BorderLayout.CENTER);
       Border border = BorderFactory.createLineBorder(Color.ORANGE);
       text1.setBorder(BorderFactory.createCompoundBorder(border,
       BorderFactory.createEmptyBorder(100, 100, 100, 100)));
       JScrollPane scroll = new JScrollPane(text1);
scroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALW
AYS);
        text1.setEditable(false);

        this.add(scroll);
      }
}
```

```
public class PSPanel extends JPanel{
	JButton search = new JButton("Search");
	JButton back   = new JButton("Back");
	JTextArea text = new JTextArea();
	String Buffer;
	Border border = BorderFactory.createLineBorder(Color.BLUE);

	public PSPanel()
	{
		setLayout(new BorderLayout());
		text.setBorder(BorderFactory.createCompoundBorder(border,
		BorderFactory.createEmptyBorder(10, 10, 10, 10)));

		add(text,BorderLayout.NORTH);
		add(back,BorderLayout.EAST);
		add(search,BorderLayout.WEST);

		back.setForeground(Color.RED);
		search.setForeground(Color.BLUE);
		back.setBackground(Color.WHITE);
		search.setBackground(Color.WHITE);
		search.addActionListener(new ActionListener() {

		public void actionPerformed(ActionEvent e) {

			Buffer = text.getText();
		}
	   });
	}
}

public class PSInterface extends JFrame {
	PSPanel ps1 = new PSPanel();
	TextPanel t2 = new TextPanel();
	String temp="";
	int count =0;
	String result1 = "\n"+"No Results Found\n\n
Click On Back Button to Start a Fresh Search";
	String result2 ="";
	String temp_name ="";
	public PSInterface()
	{
	  super("SEARCH BY PERSON NAME");
```

```
            setLayout(new BorderLayout());
            add(ps1,BorderLayout.SOUTH);
            add(t2,BorderLayout.CENTER);

            setSize(640, 480);
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            setVisible(true);
      }
      public String search_and_display(String Buffer)
      {
try {
Connection
myConn=DriverManager.getConnection("jdbc:mysql://localhost:3306/proje
ct??autoReconnect=true&useSSL=false","root","123$%^akhil..");

Statement myStmt = myConn.createStatement();
ResultSet myRs = myStmt.executeQuery("select film_name,role from
Crew_Member natural join Film natural join Worked_As where
crew_name='"+Buffer+"'");

                  while(myRs.next())
                  {
                        result1 = myRs.getString("film_name");
                         String result3 = myRs.getString("role");
temp = temp+"\nName of the Film :"+result1+"\nROLE: "+result3+"\n";
                         count++;
                        }
                  myStmt.close();
                  myConn.close();
                  }
            catch(Exception exc)
            {
                  exc.printStackTrace();
            }
if (count == 0)
      {
      return result1;
      }
else
{
return temp;
}
      }
}
```

```
public class FirstFrame extends JPanel {
      JButton button1 = new JButton("Search info by Movie Name");
      JButton button2 = new JButton("Search Movies by Person Name");
         public FirstFrame() {

            setLayout(new BorderLayout());
            add(button1,BorderLayout.NORTH);
            add(button2,BorderLayout.SOUTH);
            button2.setForeground(Color.RED);
            button1.setForeground(Color.BLUE);
            button2.setBackground(Color.WHITE);
            button1.setBackground(Color.WHITE);
            button2.setSize(new Dimension(10,10));
            button1.setSize(new Dimension(10,10));
            button1.setPreferredSize(new Dimension(640, 240));
            button2.setPreferredSize(new Dimension(640, 240));

            }

      }
public Opening_Interface()
      {
            super("MOVIE SEARCH DATABASE");
            setLayout(new BorderLayout());
            FirstFrame f1 = new FirstFrame();
            add(f1,BorderLayout.CENTER);
            f1.button1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                  setVisible(false);
                  MSInterface m = new MSInterface();
                  m.ms1.search.setEnabled(true);
                  m.ms1.search.addActionListener(new
ActionListener() {
                 public void actionPerformed(ActionEvent e) {

                       m.ms1.Buffer = m.ms1.text.getText();
                  m.ms1.Buffer= m.search_and_display(m.ms1.Buffer);

                       m.t1.text1.append(m.ms1.Buffer);
                       m.ms1.search.setEnabled(false);
                                  }
                   });
```

```
 m.ms1.back.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                         m.setVisible(false);
                         setVisible(true);

                        }
                                });
                        }
                    });



            f1.button2.addActionListener(new ActionListener() {

                    public void actionPerformed(ActionEvent e) {

                         setVisible(false);

                         PSInterface p = new PSInterface();
                         p.ps1.search.setEnabled(true);

                        p.ps1.search.addActionListener(new
ActionListener() {


                        public void actionPerformed(ActionEvent e) {

                              p.ps1.Buffer = p.ps1.text.getText();
                              p.ps1.Buffer =
p.search_and_display(p.ps1.Buffer);
                              p.t2.text1.append(p.ps1.Buffer);
                              p.ps1.search.setEnabled(false);

                        }

                    });
                        p.ps1.back.addActionListener(new
ActionListener() {


                              public void actionPerformed(ActionEvent
e) {

                              p.setVisible(false);
```

```
                                setVisible(true);

                                }

                           });

                    }

          });



               setSize(640, 480);
               setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
               setVisible(true);

        }

}


import javax.swing.SwingUtilities;

public class MainFrame {
        public static void main(String[] args) {
               SwingUtilities.invokeLater(new Runnable()
               {
               public void run()
               {
                     Opening_Interface o = new Opening_Interface();
               }
               });
        }
}
```

REFERENCES:

JAVA SWINGS- Stack Overflow
SQL-Database system Concepts,6th ED
JDBC - LAB MANUAL