TrustMe Video Player and Video Management System

Project report

Submitted by

M. N. Fathima Saliha - UWU/CST/20/011
H. G. H. Lakruwan Jayathissa - UWU/CST/20/014
A. H. M. H. Sulakshana Kumari - UWU/CST/20/045
H. N. Githma - UWU/CST/20/067
P. W. T. Dulanjani Weerasingha - UWU/CST/20/082

Under Supervision of

Mr. M.N.T. Nandasena
Lecturer of UWU

Computer Science and Technology
Department of Computer Science and informatics
Faculty of applied sciences
Uva Wellassa University February 2022

Content

Acknowledgment

The completion of this project could not have been possible without the participation and assistance of a lot of individuals contributing to this project. However, we would like to express our deep appreciation and indebtedness to our lecturer Mr.Nisal Nandasena for his endless support, kindness, and understanding during the project duration.

Many people, especially our team members itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our group project. We thank all the people for their help directly and indirectly to complete our group project.

Also, we would like to thank all our relatives, family, and friends who supported us in one way or another. Finally, we would like to thank all.

Introduction

The project we were given was to discover a real-world problem and suggest a solution in the form of a program written in the java language. Java is a simple language because of its clear, concise, and understandable syntax.

Object-oriented: Everything in Java takes the shape of an object. It indicates the device has some data and behavior. There must be at least one class and object in a program. Java contains no explicit pointers and runs programs in a virtual machine, making it a secure programming language. A security manager for Java classes is present and controls access.

Platform-Independent: Java offers the assurance that code may be written once and executed on any platform. Any machine can execute this platform-neutral byte code. Java Byte code is portable, meaning it can run on any platform. No features that depend on the implementation. The size of primitive data types, for instance, is fixed, as is everything else related to storage. Java is an interpreted language with high performance. Java uses the Just-In-Time compiler to achieve excellent performance.

This is how we develop our code step by step. This is an introduction to the main goal of our project and our report that presents some important features in our project. Here, we considered the key concepts of OOP. Finally, we should know programming is a great skill to develop, learning to think and express the scenario logically.

Our aim in this project was to create a media player. We accomplished it by complicity a list of using all the things that we have learned. When we build our project, we use IntelliJ IDE and Scenebuilder to design our code. Also, we used a flow chart. This is a desktop application which is developed in Java Platform. We develop this project not only using Java but also SQL database.

This program works like a normal PC's media player and there are some little differences in our media player considering with other media players. A user can browse any video which are in their PC and play it. This one is easy to operate and understand by the user.

This media player in which we can play a video, pause the video and restart it. We can create media player using any IDE. In our case, we used IntelliJ. We are using buttons on the front and choose to the video. First, we need to select the video that the user wants to watch. After selecting the video user can press the play button to play the song.

When you watch a video, the name of the video, the date the video was watched, the time it started, the time it ended, the start date, the end date, and everything is recorded in the database.

There are many advantages of using this media player. The information of all the videos watched by the player is stored in the database, so parents can easily check the information of all the videos watched by their children at any time.

In any educational institution where children are recommended to watch a certain video, the instructor can use this software to check whether all of them have watched the entire video or not.

And even a person who uses this can get a comprehensive understanding of the videos he has watched, the number of times he has watched them and the time duration he has watched them.

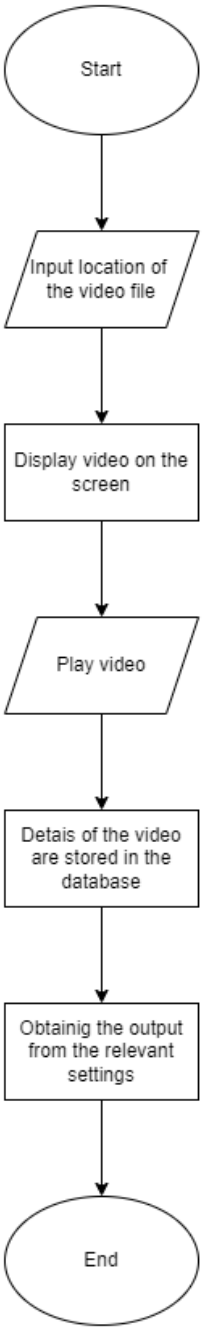
Solution steps

First, we draw the flowchart according to the scenario which we have selected. Next we created interfaces and created our code.

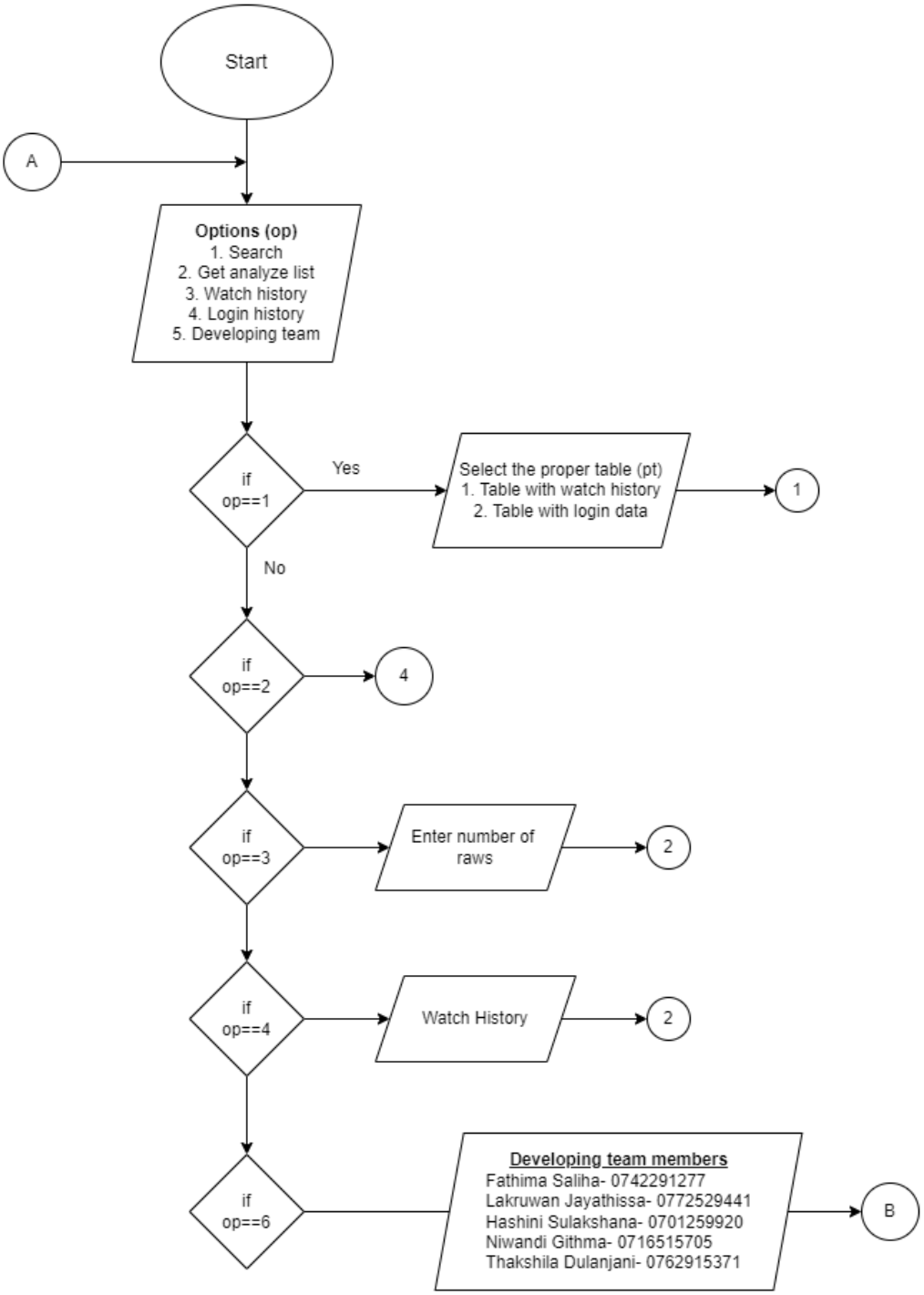
Flow chart

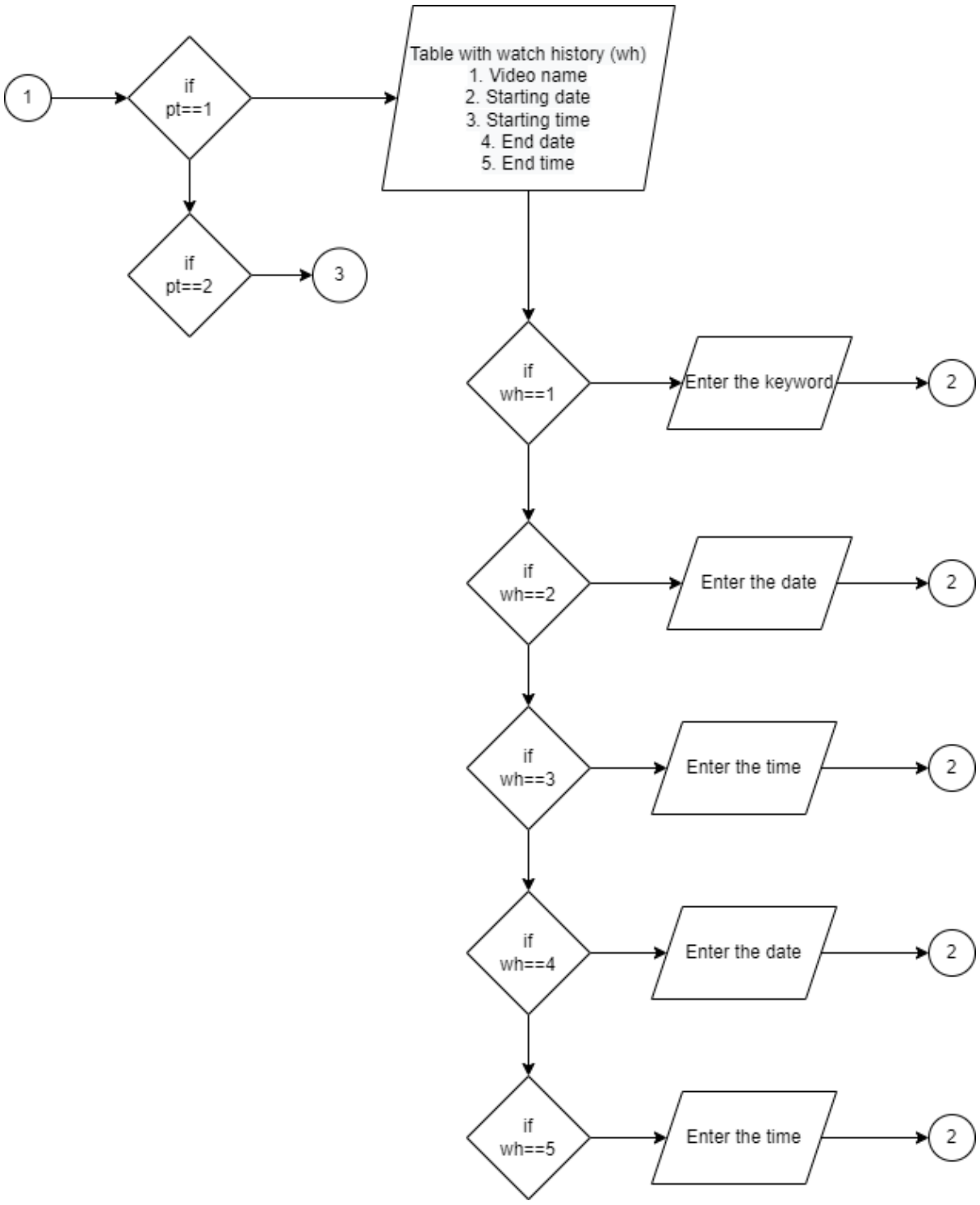
In this project we had to draw two flowcharts. One is for the player and other one is for the analyzing part.

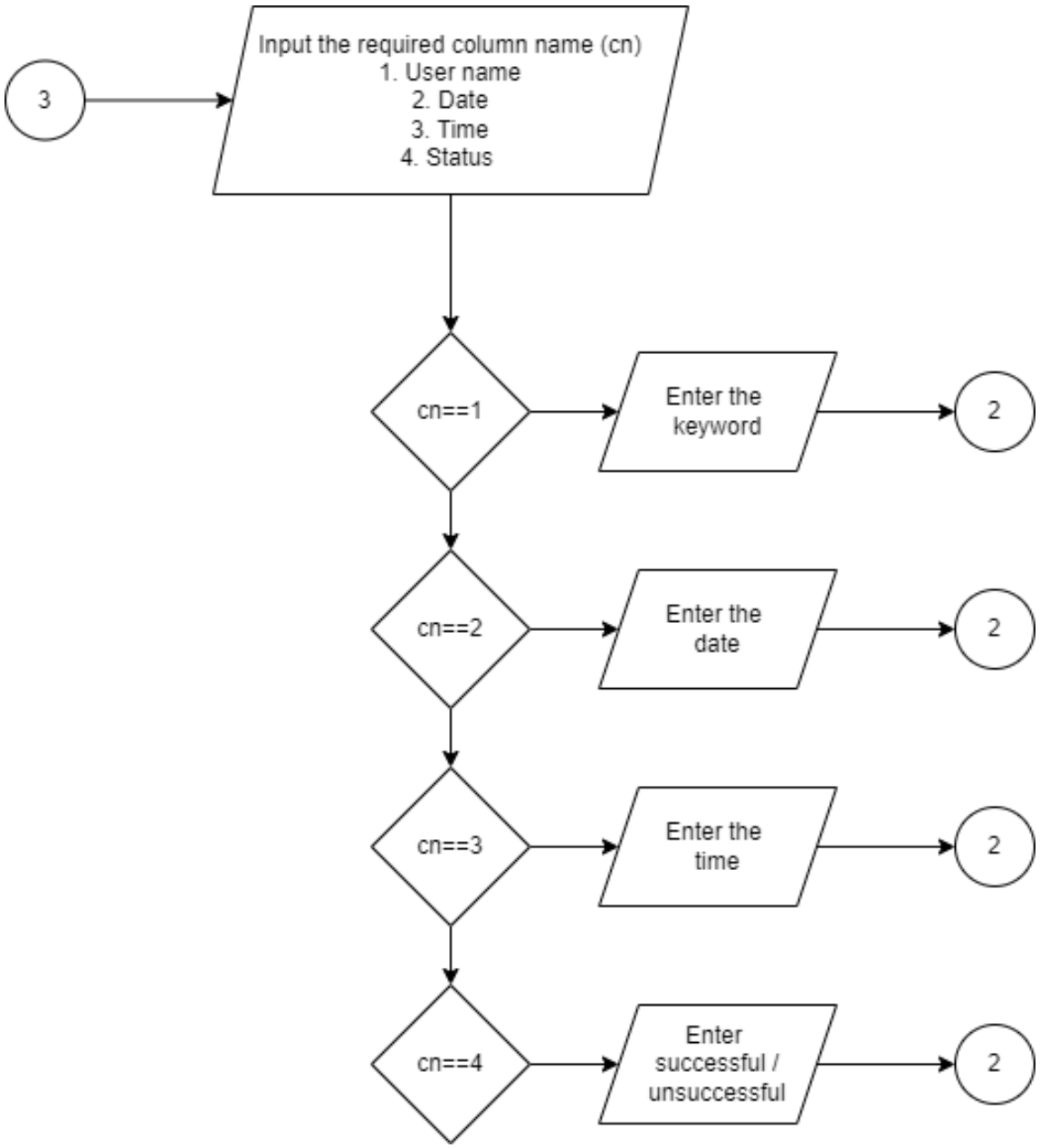
This is the flowchart related to the part where the media player plays. Can't insert a big process there. This section shows how, the data goes to the database while the video is playing when we open a video.

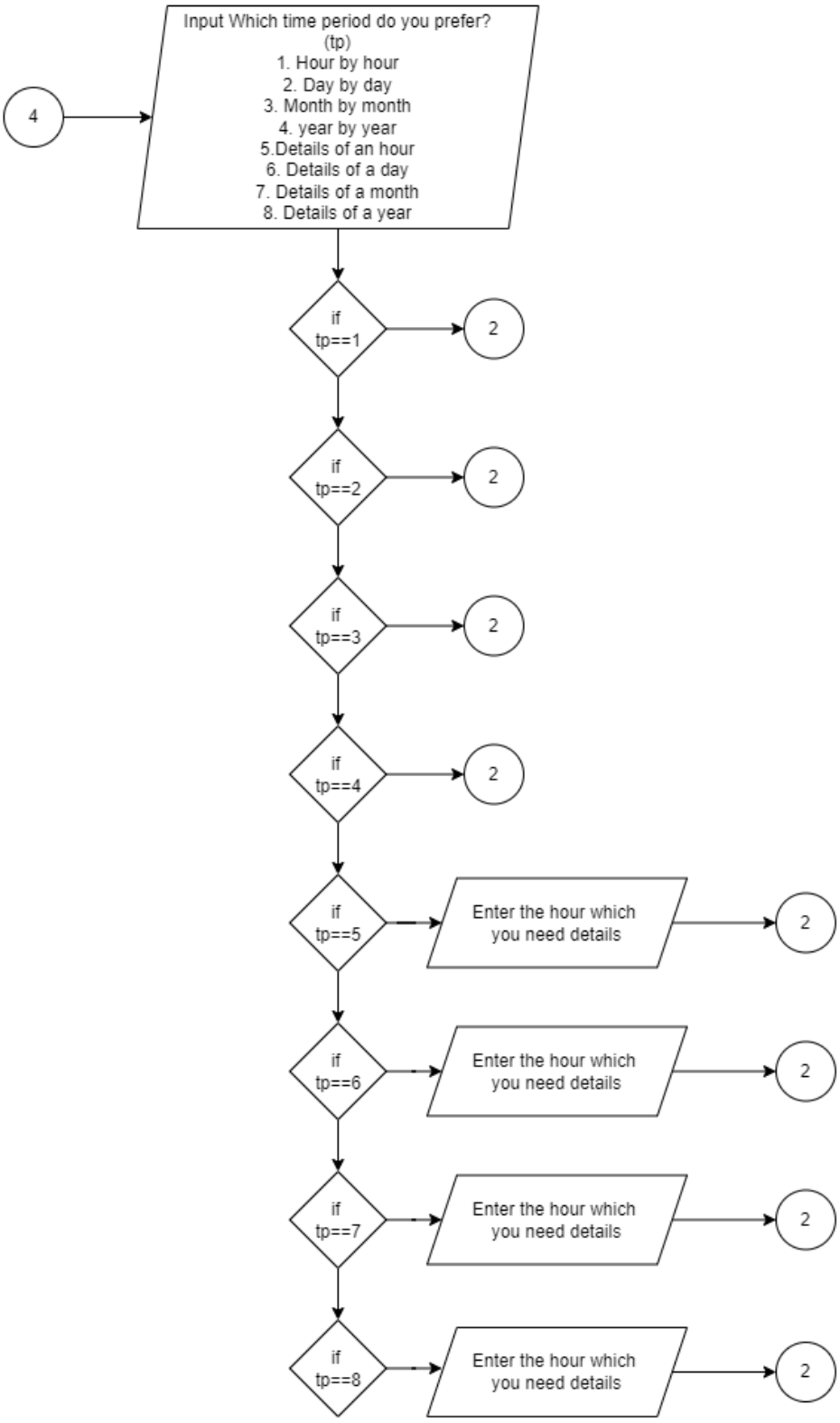


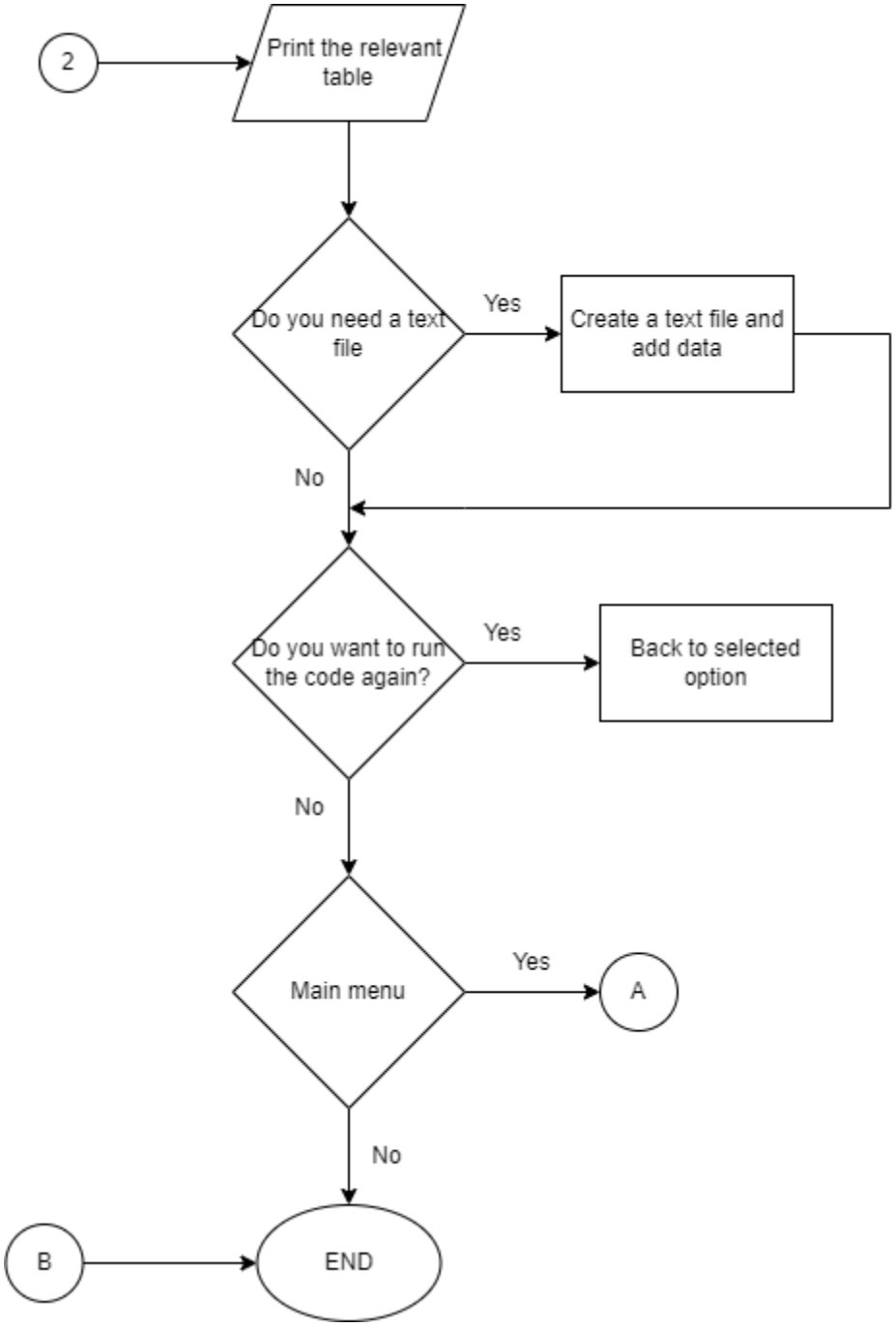
Although a big process is not visible in this flow chart, almost all the features of all other players can be seen in our player too.




1
if
pt==1
Table with watch history (wh)
1. Video name
2. Starting date
3. Starting time
4. End date
5. End time
if
pt==2
3
if
wh==1
Enter the keyword
2
if
wh==2
Enter the date
2
if
wh==3
Enter the time
2
if
wh==4
Enter the date
2
if
wh==5
Enter the time
2


3
Input the required column name (cn)
1. User name
2. Date
3. Time
4. Status
cn==1
Enter the keyword
2
cn==2
Enter the date
2
cn==3
Enter the time
2
cn==4
Enter successful / unsuccessful
2


4
Input Which time period do you prefer?
(tp)
1. Hour by hour
2. Day by day
3. Month by month
4. year by year
5.Details of an hour
6. Details of a day
7. Details of a month
8. Details of a year
if tp==1
2
if tp==2
2
if tp==3
2
if tp==4
2
if tp==5
Enter the hour which you need details
2
if tp==6
Enter the hour which you need details
2
if tp==7
Enter the hour which you need details
2
if tp==8
Enter the hour which you need details
2


2
Print the relevant table
Do you need a text file
Yes
Create a text file and add data
No
Do you want to run the code again?
Yes
Back to selected option
No
Main menu
Yes
A
No
B
END

Source code

Java files

TmPlayer.java
player.java
PlayTime.java
opennew.java
Playlist.java
Miniview.java
Login.java
DataList.java
TableData1.java
connectDB.java
Analyze.java
Analyze 2.java

fxml files

Analize.fxml
DaraList.fxml
Login.fxml
MiniView240p.fxml
MiniView320p.fxml
player.fxml
Playlist.fxml
Sec.fxml

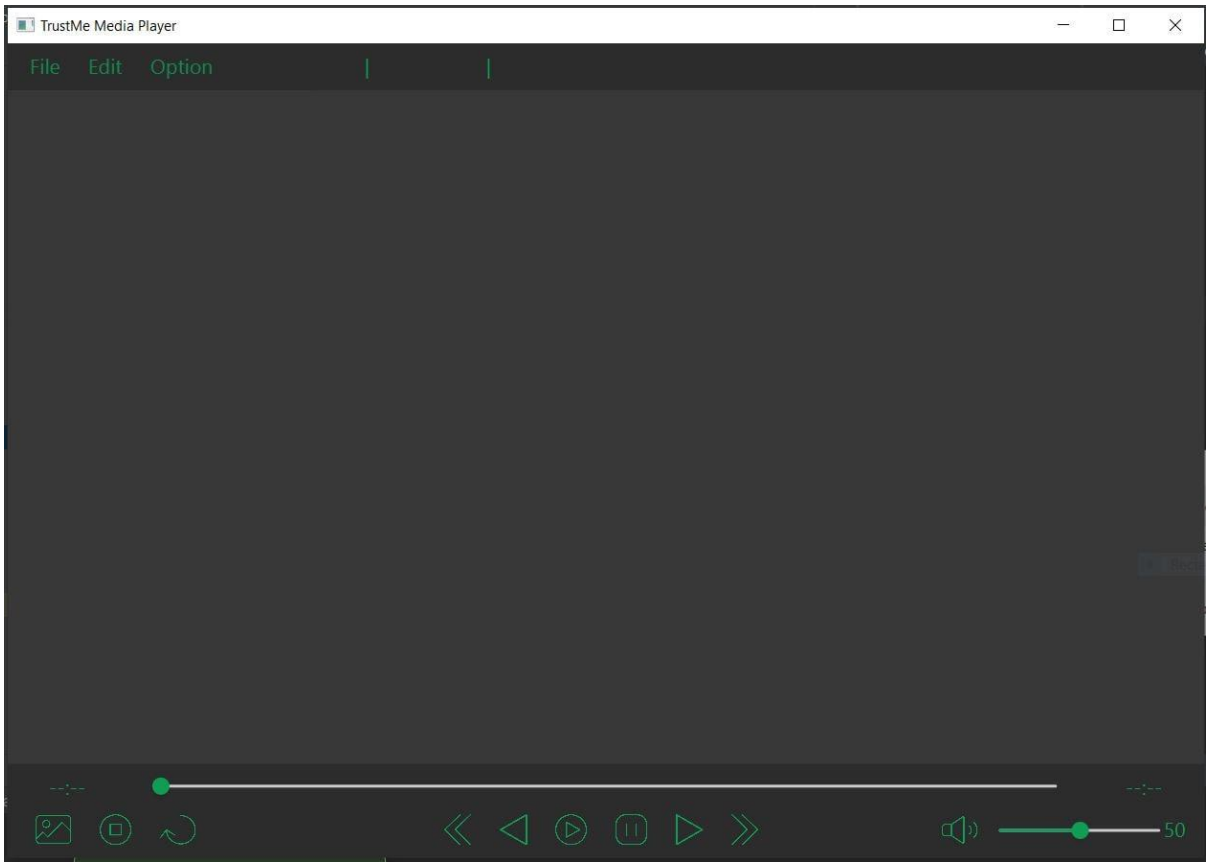
Css files

player.css

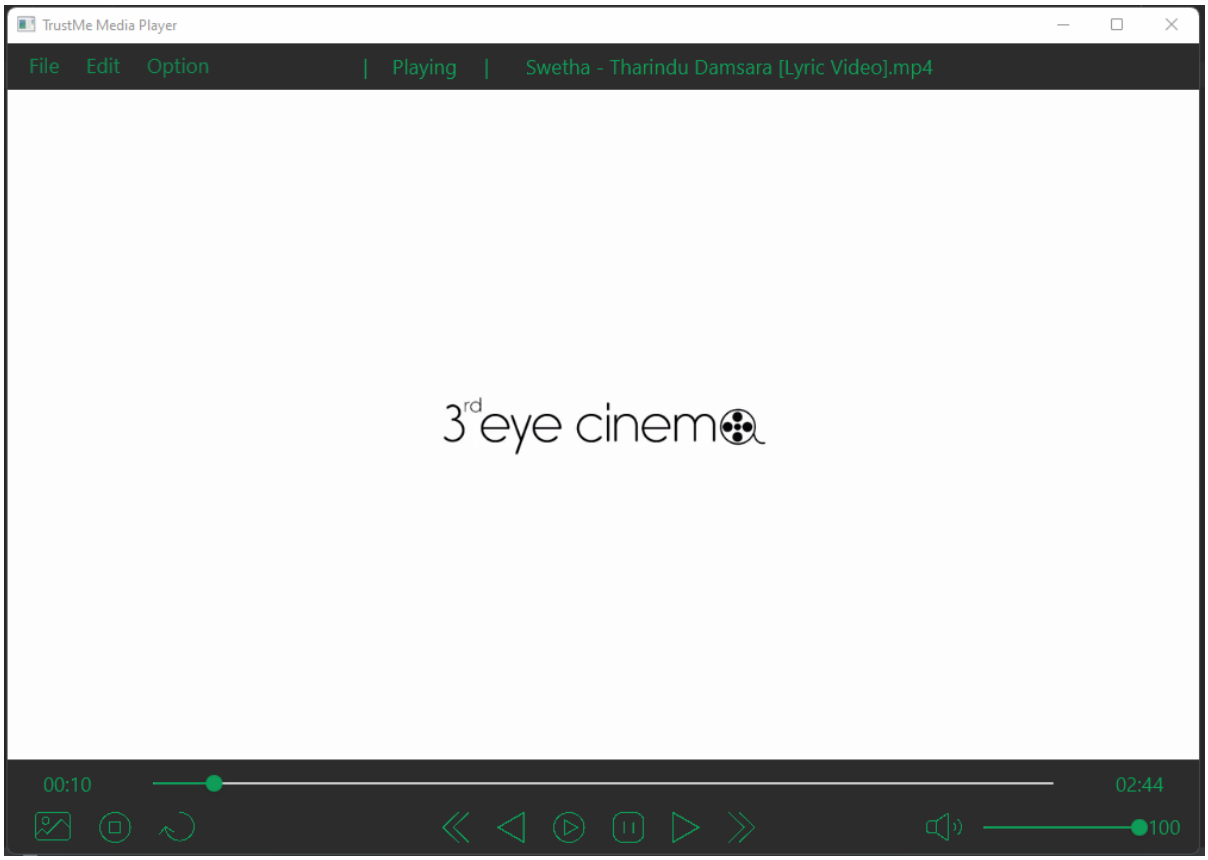
TmPlayer seems to be the name of our program, and its code is divided into 12 java file parts, seven fxml files and also one css file. A media player is a challenging part was the development of the mini player with covering all the guidelines which we given.

Here we can open a video file, stop or pause a video that is playing. Also, we can skip ten seconds, back ten seconds, skip one minute, back one minute, stop or repeat the video. You can increase or decrease the sound. You can quickly view the last five videos you watched from the recent list in the file. User can exit from the player from click on the exit option. With the speed up under edit, the speed will be doubled, with normal speed the speed will remain same as normal, with a speed down, the speed will be reduced by half. In addition to that, we can view the video by pinching it with the miniview option. Even while doing other work, you can do 2 things at the same time because of the miniview option. The playlist shows the last ten videos we watched. From there we can click on the name of the video we want and watch it quickly. When you go to the history and the admin panel, you can see the things that an admin can control.

Below is the first interface that appears when the program is run.

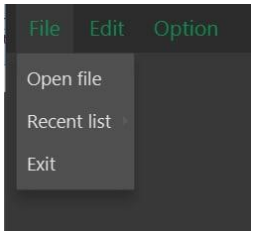


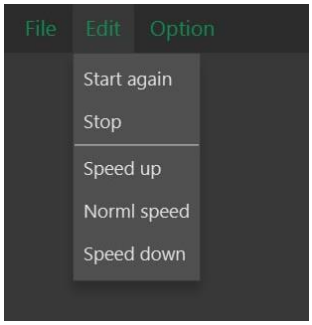
Here is what it looks like when we load a video.

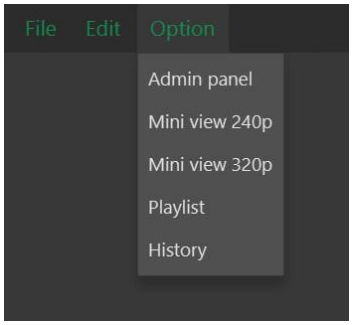


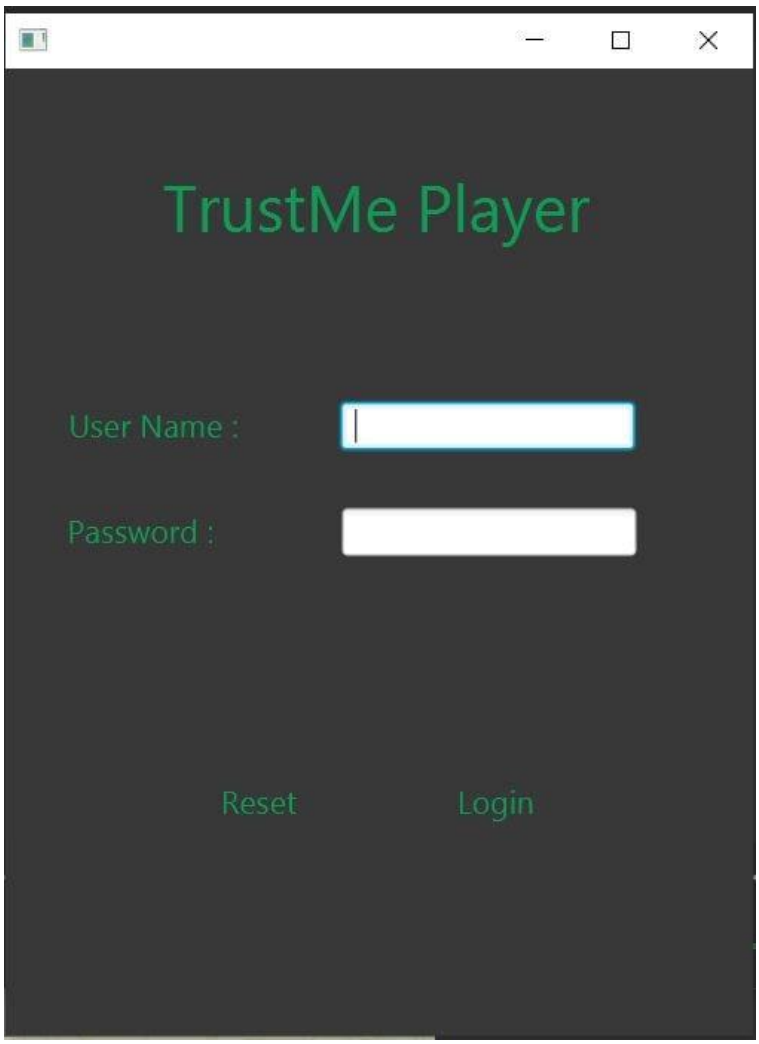
There, the name of the video is displayed above separately. You can see the total time of the video and the time the video is playing now. Apart from that, the features found in a normal video player can also be seen in this player.

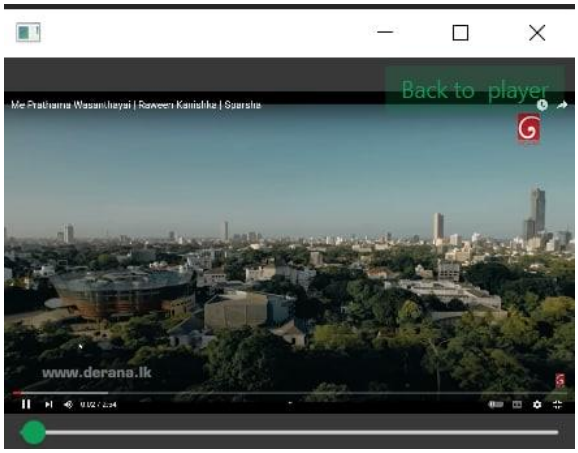
Also, some more screenshots are shown below to get a brief understanding of the options available in our player.

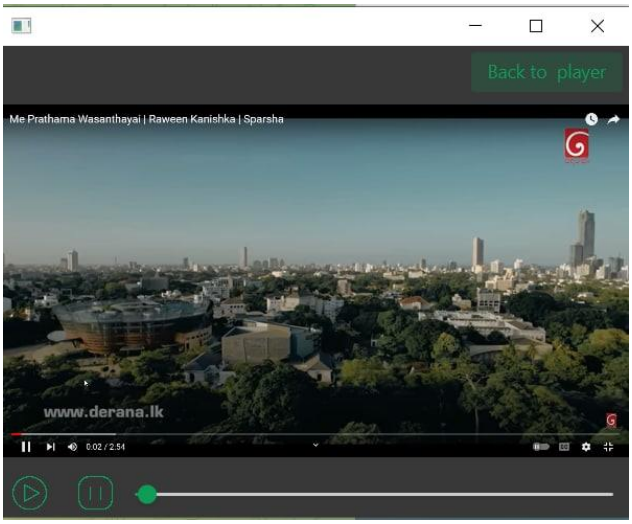


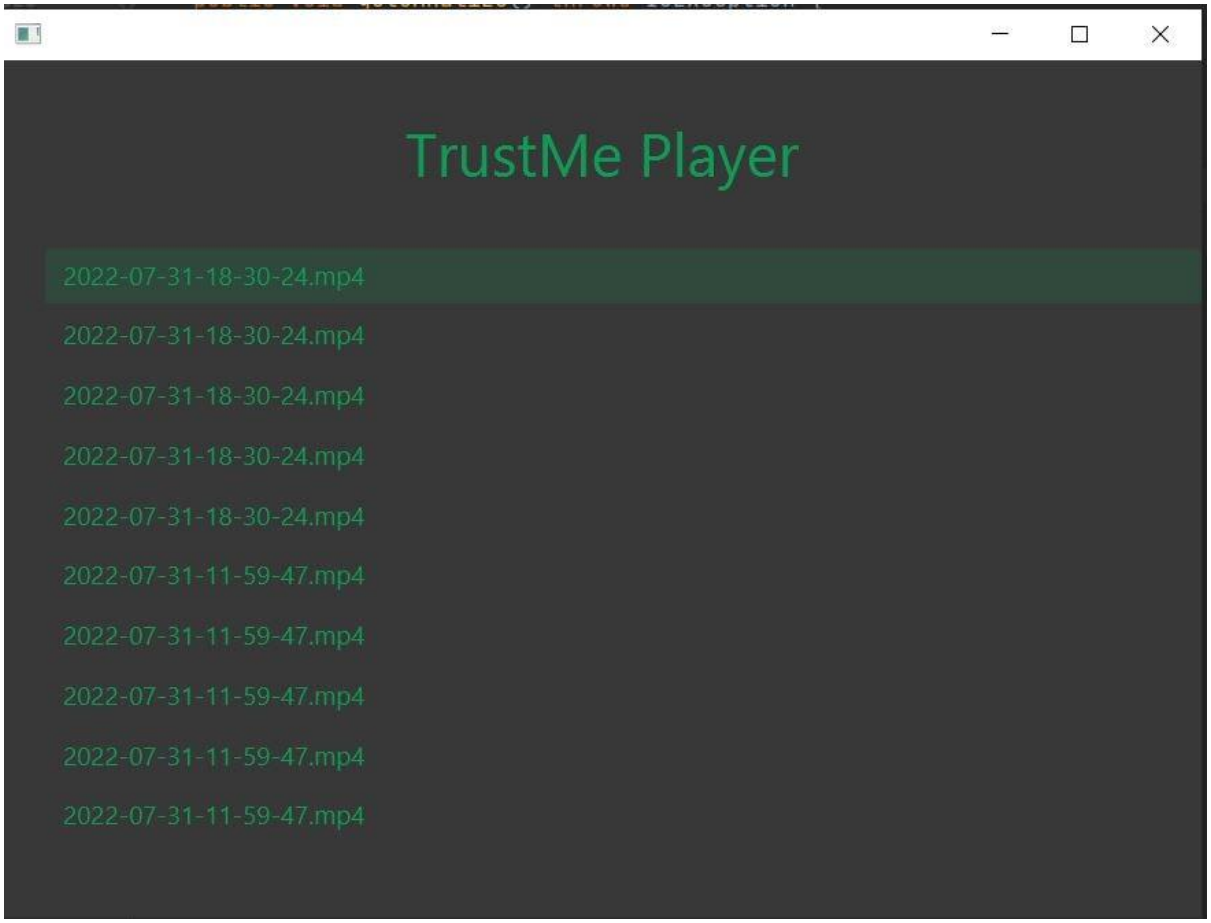


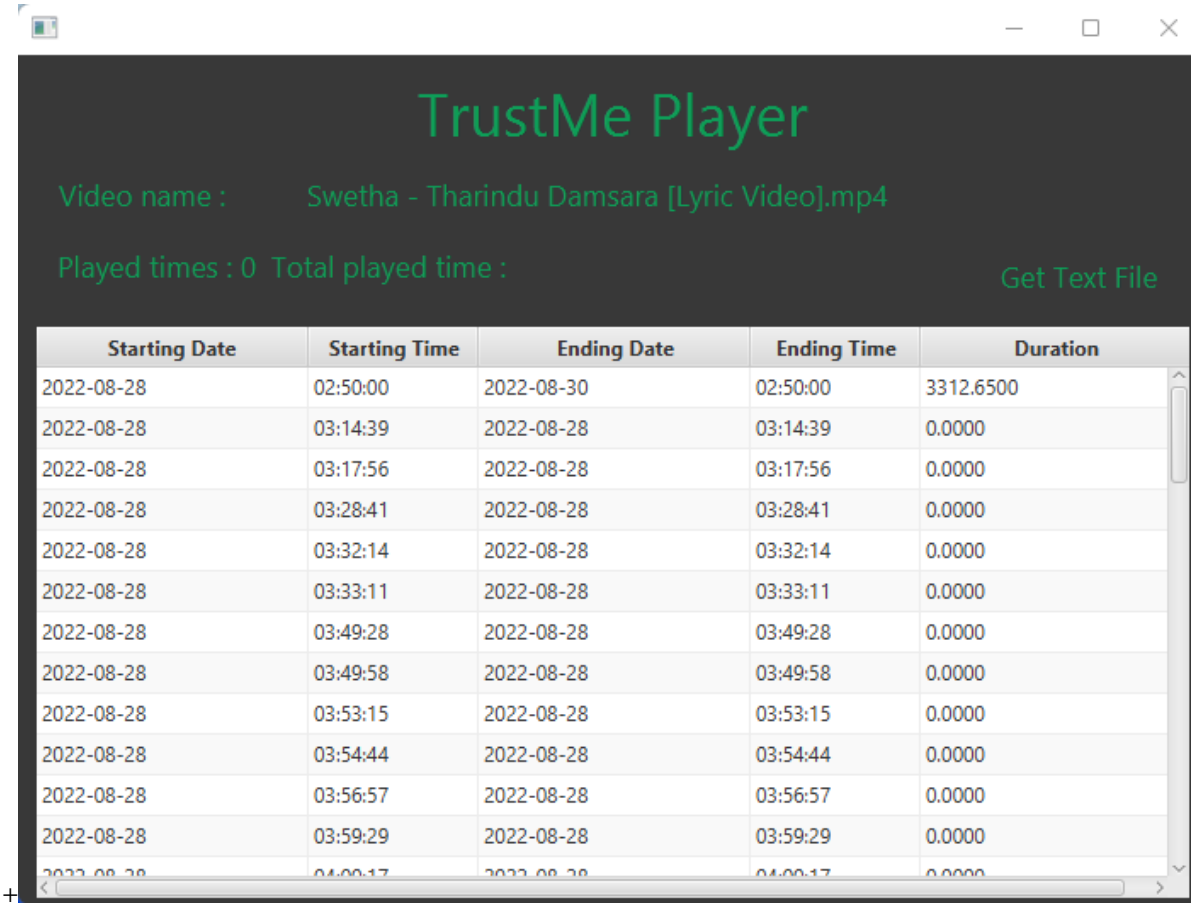





Back to player
Me Prathama Wasanthayai | Raweem Kanishka | Suarsha
www.derana.lk


Back to player
Me Prathama Wasanthayai | Raweem Kanishka | Sparsha
www.derana.lk


TrustMe Player
2022-07-31-18-30-24.mp4
2022-07-31-18-30-24.mp4
2022-07-31-18-30-24.mp4
2022-07-31-18-30-24.mp4
2022-07-31-18-30-24.mp4
2022-07-31-11-59-47.mp4
2022-07-31-11-59-47.mp4
2022-07-31-11-59-47.mp4
2022-07-31-11-59-47.mp4
2022-07-31-11-59-47.mp4



Starting Date	Starting Time	Ending Date	Ending Time	Duration
2022-08-28	02:50:00	2022-08-30	02:50:00	3312.6500
2022-08-28	03:14:39	2022-08-28	03:14:39	0.0000
2022-08-28	03:17:56	2022-08-28	03:17:56	0.0000
2022-08-28	03:28:41	2022-08-28	03:28:41	0.0000
2022-08-28	03:32:14	2022-08-28	03:32:14	0.0000
2022-08-28	03:33:11	2022-08-28	03:33:11	0.0000
2022-08-28	03:49:28	2022-08-28	03:49:28	0.0000
2022-08-28	03:49:58	2022-08-28	03:49:58	0.0000
2022-08-28	03:53:15	2022-08-28	03:53:15	0.0000
2022-08-28	03:54:44	2022-08-28	03:54:44	0.0000
2022-08-28	03:56:57	2022-08-28	03:56:57	0.0000
2022-08-28	03:59:29	2022-08-28	03:59:29	0.0000

```
+-------------------------------------------------------------------------------+
|                            >>> TrustMe Player <<<                             |
|                         *---------------------------------*                   |
|                          ~ All You Gotta Do is TrustMe ~                      |
+ - - ------------------------------------------------------------------- - - +


Options :
  Search --->                Press 1
  Get Analyze List --->      Press 2
  Watch History --->         Press 3
  Login History --->         Press 4
  Developing team --->       Press 5

Enter your choice :
```

TmPlayer.java

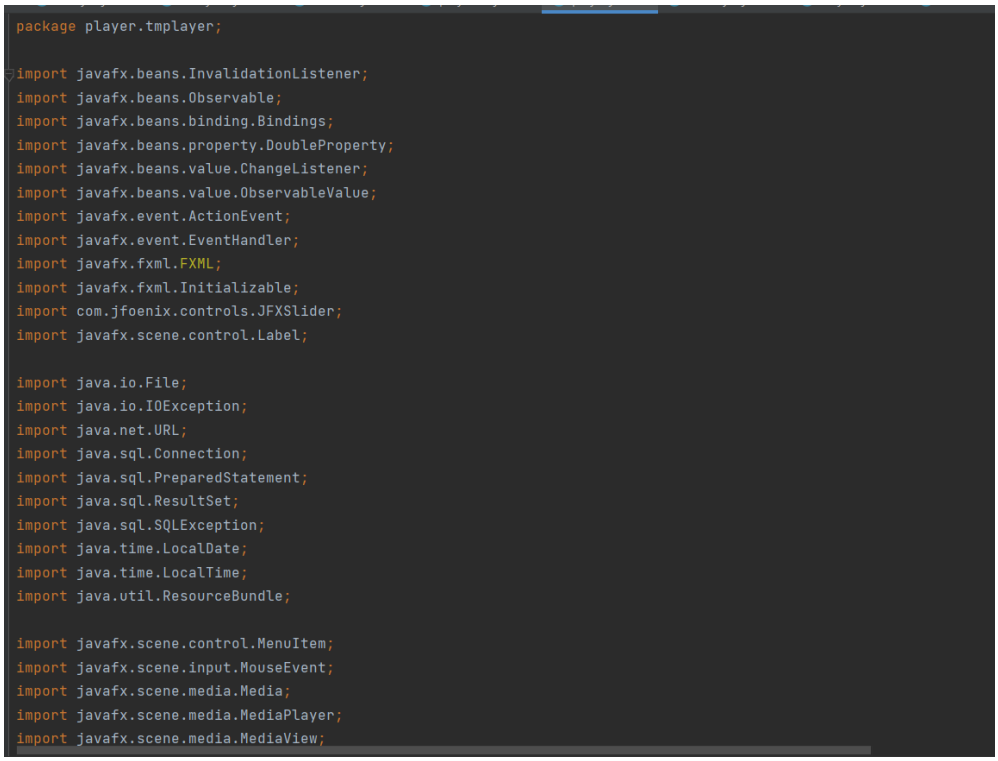
This is where we run the main method of the player. What is done in it is to run the file called player.fxml. That is the main part 1 of our program.

```
import ...


public class TmPlayer extends Application {

    @Override
    public void start(Stage stage) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(TmPlayer.class.getResource( name: "player.fxml"));
        Scene scene = new Scene(fxmlLoader.load());

        stage.setTitle("TrustMe Media Player");


        scene.setOnMouseClicked(new EventHandler<MouseEvent>() {

            @Override
            public void handle(MouseEvent event) {
                if(event.getClickCount() ==2){
                    stage.setFullScreen(true);
                }
            }
        });

        stage.setScene(scene);
        stage.show();
    }


    public static void main(String[] args) { launch(); }
}
```

Player.java

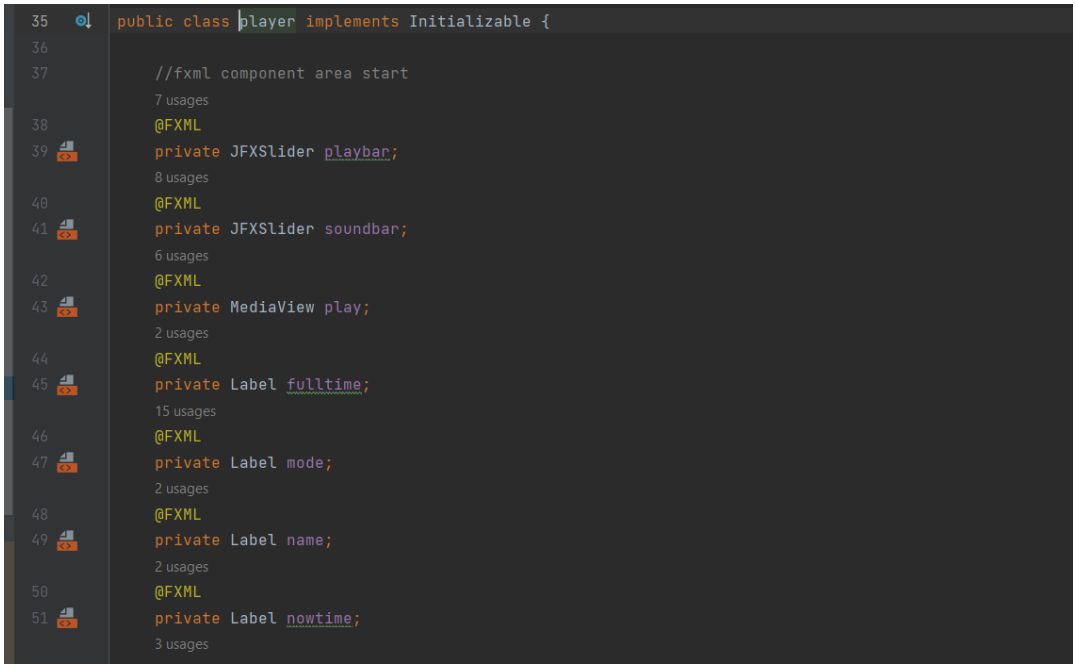
This is the major window. This one will build the interface of the player.

Import
Here we have imported we have imported required classes.



```
package player.tmplayer;

import javafx.beans.InvalidationListener;
import javafx.beans.Observable;
import javafx.beans.binding.Bindings;
import javafx.beans.property.DoubleProperty;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import com.jfoenix.controls.JFXSlider;
import javafx.scene.control.Label;

import java.io.File;
import java.io.IOException;
import java.net.URL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.time.LocalDate;
import java.time.LocalTime;
import java.util.ResourceBundle;

import javafx.scene.control.MenuItem;
import javafx.scene.input.MouseEvent;
import javafx.scene.media.Media;
import javafx.scene.media.MediaPlayer;
import javafx.scene.media.MediaView;
```

player
FXML components have been introduced to the code.



```
35  public class player implements Initializable {
36
37      //fxml component area start
        7 usages
38      @FXML
39      private JFXSlider playbar;
        8 usages
40      @FXML
41      private JFXSlider soundbar;
        6 usages
42      @FXML
43      private MediaView play;
        2 usages
44      @FXML
45      private Label fulltime;
        15 usages
46      @FXML
47      private Label mode;
        2 usages
48      @FXML
49      private Label name;
        2 usages
50      @FXML
51      private Label nowtime;
        3 usages
```

Creating variables

From this part we have created required variables.

```
25 usages
public static String path;
5 usages
public static float sound;
8 usages
public static String fname;

4 usages
public static int modeSet;
6 usages
public String[] rl = {"","","","",""};
6 usages
public String[] rlPath = {"","","","",""};
public MediaPlayer player;
2 usages
playTime Slidertime = new playTime();
```

player()

Create required necessary variables, method to connect database.

```
public player() throws SQLException, IOException {
}

//-------------------DataBase _ Start-------------------

7 usages
public static Connection con = null;
8 usages
PreparedStatement pst = null;
7 usages
ResultSet rs = null;
3 usages
public static int nTime;
7 usages
int id;

1 usage
LocalDate sDate = LocalDate.now(),eDate = LocalDate.now();
1 usage
LocalTime sTime = LocalTime.now(),eTime = LocalTime.now();
1 usage
String fomat = ".mp4",fa="x",aa="x",lid;
```

getcon()

This code part use to connect database.

```
public void getcon(){
    //connect db
    con = connectDB.connect();
}
```

execute_()

Contain the code which is use to run a sql query.

```
public void execute_(String q) throws SQLException {
    getcon();
    pst = con.prepareStatement(q);
    pst.execute();
    id = id + 1 ;
}
```

lastid()

In watch video, the last id of the currently watched video is contained in this. Here string lid, will return.

```
private String lastid() throws SQLException {
    String p = "select * from watchvideo order by ID desc limit 1";
    con = connectDB.connect();
    pst = con.prepareStatement(p);
    rs = pst.executeQuery();
    while (rs.next()){
        lid = rs.getString( columnLabel: "ID");
    }
    return lid;
}
```

getPath()

Using above part gain the path of video.

```
public void getPath(){
    //choose
    FileChooser file = new FileChooser();
    File file1 = file.showOpenDialog( ownerWindow: null);

    path = file1.toURI().toString();
    fname = file1.getName();
}
```

loadvideo()

In here load the video and the length and width are made accurately as required. Then play the video.

```
public void loadVideo() throws SQLException {

    if(path != null) {
        Media media = new Media(path);
        player = new MediaPlayer(media);
        play.setMediaPlayer(player);

        final DoubleProperty width = play.fitWidthProperty();
        final DoubleProperty height = play.fitHeightProperty();

        width.bind(Bindings.selectDouble(play.sceneProperty(),  ...steps: "width"));
        height.bind(Bindings.selectDouble(play.sceneProperty(),  ...steps: "height"));

        player.play();
        vlevel.setText("100");

        name.setText(fname);
```

Play

The play method inside the mediaplayer library is called. That will play the video. In mode.set.text, a label called mode has been created before. A text has been set for that label. Now we have been able to see what has been played in the player. When the video is played, playing is displayed instead of mode.

```
//play

public void play(ActionEvent event){
    player.play();
    mode.setText("Playing");
}
```

Pause

The pause method inside the mediaplayer library is called. That will pause the video being played. In mode.setText, a text has been set for the mode label. When the video is paused, the word "pause" is displayed instead of "mode".

```
//pause

public void pause(ActionEvent event){
    player.pause();
    mode.setText("Pause");
}
```

stop

The stop method inside the mediaplayer library is called. By that, the playing video will be completely stopped. In mode.setText, a text has been set for the mode label. When the video is stopped, the word "stopped" will be displayed instead of "mode".

```
//stop

public void stop(ActionEvent event){
    player.stop();
    mode.setText("Stoped");
}
```

Slow

The setRate method inside the mediaplayer library is called. By that, the speed of the video played is ×0.5 lower than the normal speed. In mode.set.text, a text has been set for the mode label. Here the text speed × 0.5 is displayed.

```
//slow

public void slow(ActionEvent event){
    player.setRate(0.5);
    mode.setText("Speed x0.5");
}
```

Fast

The setRate method inside the mediaplayer library is called. By that, the speed of the video played is ×2 higher than the normal speed. In mode.set.text, a text has been set for the mode label. Here the text speed × 2 is displayed.

```
//fast

public void fast(ActionEvent event){
    player.setRate(2);
    mode.setText("Speed x2");
}
```

Normal speed

The setRate method inside the mediaplayer library is called. By that, the speed of the video played will be ×1 at normal speed. In mode.set.text, a text has been set for the mode label. Here, where mode is called, playing is displayed.

```
//speed normal

public void sNomal(ActionEvent event){
    player.setRate(1);
    mode.setText("Playing");

}
```

Repeat

The load video method inside the mediaplayer library is called. By that, the video that has been played will be repeated and played again. In mode.set.text, a text has been set for the mode label. When the video is replayed, the word "playing" will be displayed instead of "mode".

```
//repeat

public void rep(ActionEvent event){
    loadVideo();
    mode.setText("Playing");
}
```

Start Again

The load video method inside the mediaplayer library is called. By that, the stopped video will be replayed. In mode.set.text, a text has been set for the mode label. When the video is restarted, playing will be displayed instead of mode.

```
//start again

public void sAgain(ActionEvent event){
    loadVideo();
    mode.setText("Playing");
}
```

Change db end time

In this method cdet means change database end time. This method does change the end time. We create Templary int variable which is called k. here 1 is subtracted from the id variable. Then write the SQL code we want. In here update E_date and E_time according present situation. Where id equal to K in this step we subtract 1 from id variable because we add 1 from above execute method for, we need to next step. After that we create connection to database. Finally, we execute our code and update.

```
//change db end time

public void cedt() throws SQLException {
    int k = id - 1;
    String q = "UPDATE watchvideo SET E_date = '" + LocalDate.now() +"', E_time = '" + LocalTime.now() +"' WHERE id = '"+ k
    getcon();
    pst = con.prepareStatement(q);
    pst.execute();
}
```

With this, the video is played when the mouse is clicked as well as the slider moves around. **currentTimeProperty().addListener()** as it is updated regularly, the end date needs to be updated every time, so it should be called in the code.

```
//slider
player.currentTimeProperty().addListener(new ChangeListener<javafx.util.Duration>() {
    @Override
    public void changed(ObservableValue<? extends Duration> observable, Duration oldValue, Duration newValue) {
        playbar.setValue(newValue.toSeconds());
        nowtime.setText(Slidertime.displayTime(newValue.toSeconds()));
        fulltime.setText(Slidertime.displayTime(player.getTotalDuration().toSeconds()));
        nTime = (int) newValue.toSeconds();
        //db
        try {
            cedt();
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }
});
playbar.setOnMousePressed(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent event) {
        player.seek(javafx.util.Duration.seconds(playbar.getValue()));
    }
});

playbar.setOnMouseDragged(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent event) {
        player.seek(javafx.util.Duration.seconds(playbar.getValue()));
    }
```

Mode.setText() will set the above things to a label above. What this does is that the information goes to a database.

```
mode.setText("Playing");

//db
String[] wv = {String.valueOf(id),String.valueOf(fname),String.valueOf(fomat), String.valueOf(path),String.valueOf(sDate), String.valueOf(sTime), String.value
String p = "INSERT INTO watchvideo VALUES ('"+ wv[0] +"', '"+ wv[1] +"', '"+ wv[2] +"', '"+ wv[3] +"', '"+ wv[4] +"', '"+ wv[5] +"', '"+ wv[6] +"',  '"+ wv[7]
execute_(p);
```

Openfile()

With Get path and load video two methods are run. It is replaced by button.

```
//open file

public void openfile() throws SQLException {
    getPath();
    loadVideo();
}
```

Load video

slider

currentTimeProperty().addListener as it is updated regularly, the end date needs to be updated every time, so it should be called in the code.

```
//slider

player.currentTimeProperty().addListener(new ChangeListener<javafx.util.Duration>() {

    @Override
    public void changed(ObservableValue<? extends Duration> observable, Duration oldValue, Duration newValue) {
        playbar.setValue(newValue.toSeconds());
        nowtime.setText(Slidertime.displayTime(newValue.toSeconds()));
        fulltime.setText(Slidertime.displayTime(player.getTotalDuration().toSeconds()));
        nTime = (int) newValue.toSeconds();
```

Db

in this step we create a string array and enter data in it as need our database.

```
//db
String[] wv = {String.valueOf(id),String.valueOf(fname),String.valueOf(fomat), String.valueOf(path),String.valueOf(sDate), String.valueOf(sTime), String.valueOf(eDate), String
String p = "INSERT INTO watchvideo VALUES ('"+ wv[0] +"', '"+ wv[1] +"', '"+ wv[2] +"', '"+ wv[3] +"', '"+ wv[4] +"', '"+ wv[5] +"', '"+ wv[6] +"',  '"+ wv[7] +"', '"+ wv[8] +
execute_(p);
```

skip and back

First, we create a skip and back method. That will give you the current position of the video. It is saying the media is played for one minute, it returns for one minute and we have to add we need time. It is denoted by the letter t. In this method, we use an int variable, which is t, as a parameter in the method we are creating. We can call this method anywhere. The parameters we provided go to the relevant place. If we want to go back 10 seconds in the video, we can call this method skip and back, and we must give t as (t:-10). If we want to go back 60 seconds in the video, we can call this the skip and back method, and we must give t as (t: -60). If we want to skip 10 seconds in the video, we can call this the skip and back method, and we must give t as (t: 10). If we want to skip 60 seconds in the video, we can call this the skip and back method, and we must give t as (t: 60).

```
//skip and back

public void skipAndBack(int t){
    player.seek(player.getCurrentTime().add(javafx.util.Duration.seconds(t)));
}
//-10

public void back10(ActionEvent event){
    skipAndBack( t -10);
}
//-60

public void back60(ActionEvent event){
    skipAndBack( t -60);
}
//+10

public void skip10(ActionEvent event){
    skipAndBack( t 10);
}
//+60

public void skip60(ActionEvent event) {
    skipAndBack( t 60);
}
```

menu item play list

This goplaylist method, If something is running now, if the path is null, if a video is running, it will stop.

mode.getScene().getWindow().hide() in this hide and in **openNew.onlyOpen** go to play list.

```
//menu item playlist

public void gotoplaylist() throws IOException {
    if (path!=null) {
        player.stop();
    }
    mode.getScene().getWindow().hide();
    openNew.onlyOpen( x: "Playlist.fxml");
}
```

menu item miniview 240

In this method, the current sound is obtained and the video is stopped.

mode.getScene().getWindow().hide(); hide it and finally goes to the miniview.

```
//menu item miniview 240

public void goto240() throws IOException {
    sound =  (float) soundbar.getValue()/100;
    player.stop();
    mode.getScene().getWindow().hide();
    openNew.onlyOpen( x: "MiniView240p.fxml");
}
```

menu item miniview 320

In this method, the current sound is obtained and the video is stopped.

mode.getScene().getWindow().hide(); hide it and finally goes to the miniview.

```
//menu item miniview 320

public void goto320() throws IOException {
    sound = (int) soundbar.getValue();
    player.stop();
    mode.getScene().getWindow().hide();
    openNew.onlyOpen( x: "MiniView320p.fxml");
}
```

menu item Analyze

The value of the variable called modeset is 1. In the next step, the login form will open and finally it will go to analyze.

```
//menu item Analize

public void gotoAnalize() throws IOException {
    modeSet = 1;
    openNew.onlyOpen( x: "Login.fxml");
}
```

menu item History

If the value of the variable called modeset is 2 and the path is not null, then you can go to the login form and then go to the datalist.

```
//menu item History

public void gotodata() throws IOException {
    modeSet = 2;
    if (path != null) {
        openNew.onlyOpen( x: "Login.fxml");
    }
}
```

menu item exit

In this method, if a video is playing, it stops and closes it.

```
//menu item exit

public void close() {
    if (path!=null) {
        player.stop();
    }
    mode.getScene().getWindow().hide();
}
```

set items to recent list

Here the resent item in the menu bar is set. Then the last few IDs of the watchvideo are limited to 5 and selected and executed. After that, the video name related to the array called rl() and the path related to the array called rlpath are included.

```
//set items to recent list

public void rl() throws SQLException {

    String p = "select * from watchvideo order by ID desc limit 5";
    con = connectDB.connect();
    pst = con.prepareStatement(p);
    rs = pst.executeQuery();

    int x = 0;
    while (rs.next()){
        rl[x] = rs.getString( columnLabel: "VideoName");
        rlPath[x] = rs.getString( columnLabel: "Path");
        x=x+1;
    }

    rl1.setText(rl[0]);
    rl2.setText(rl[1]);
    rl3.setText(rl[2]);
    rl4.setText(rl[3]);
    rl5.setText(rl[4]);
}
```

Play recent list videos

Here is the video play in the resent list. In the 1st item of the resent list, the 0 item in the array called rlpath is replaced with the variable called path and the video is loaded and the database is updated at the same time. These steps are implemented in the same 5 items.

```
//play resent list videos

public void play1() throws SQLException {
    path = rlPath[0];
    loadVideo();
}

public void play2() throws SQLException {
    path = rlPath[1];
    loadVideo();
}

public void play3() throws SQLException {
    path = rlPath[2];
    loadVideo();
}

public void play4() throws SQLException {
    path = rlPath[3];
    loadVideo();
}

public void play5() throws SQLException {
    path = rlPath[4];
    loadVideo();
}
```

When this class starts to run, the lastid method is called. After that, the value of the last id of the watch video table is taken.

```
@Override
public void initialize(URL location, ResourceBundle resources) {

    //set correct id to db
    try {
        lastid();
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
    if (lid == null) {
        id = 0;
    } else if (Integer.parseInt(lid)<0){
        id = 0;
    }else {
        id = Integer.parseInt(lid) + 1 ;
    }

    try {
        rl();
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
```

playtime.java

Using this class, we are trying to demonstrate the duration of the playing video. The time duration is being given in seconds and to set it as Standard Time we want to convert it into hours, minutes and seconds. We have created a method called display Time and added ‘t’ as a parameter. Its variable type is double.

To store time which we are going to convert, we have created 3 integer variables called hr, min and sec. the duration of a particular video is given in seconds, first the total duration must divide by 3600. Then to find the rest of minutes first we need to find the remainder when t is divided by 3600 and then the answer is divided by 60.

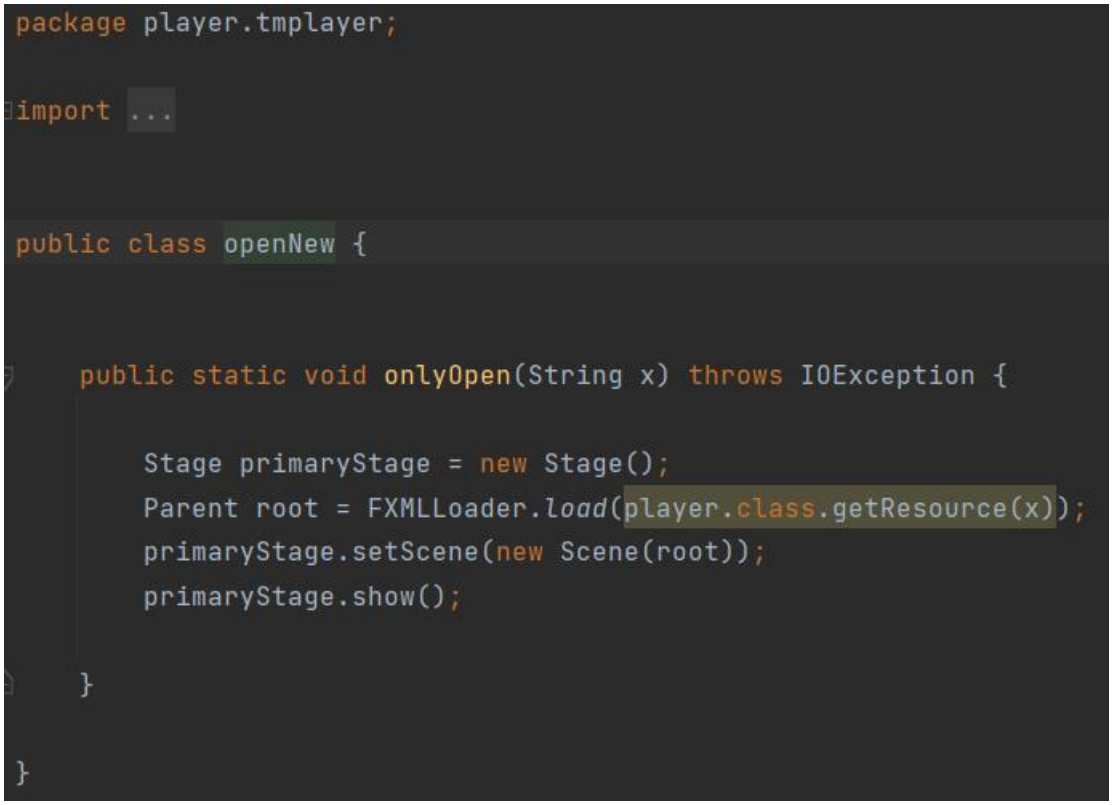
Finally, for find seconds first t is divided by 3600 and take the remainder, then again it is divided by 60 and find the remainder. All those three values must convert into integers using narrow typecasting.

Then t is running through if else statements. If t value is greater than or equal to 3600, it’ll go through another if else statement to check whether sec and is less than 10 separately. If they are true 0 will print in front of the number 0-9. If t value is less than 3600, it’ll go through the first if else statement and execute the else part like previously. For executing this part integer values of sec, min and are converted into strings to assign into ‘time’ variables and add those strings. Then the string ‘time’ will return. As the final step of this part video time will print at relevant labels.

```
public class playTime {


    public static String displayTime(double t) {
        String time = "";

        int hr = (int) (t/3600);
        int min = (int) ((t%3600) / 60);
        int sec = (int) ((t%3600) % 60);

        if (t>=3600) {
            if (sec < 10) {
                time = time + Integer.toString(hr) + ":" + Integer.toString(min) + ":" + "0" + Integer.toString(sec);
            } else if (min < 10) {
                time = Integer.toString(hr) + ":" +"0" + Integer.toString(min) + ":" + Integer.toString(sec);
            } else {
                time = Integer.toString(hr) + ":" + Integer.toString(min) + ":" + Integer.toString(sec);
            }
        } else {
            if (sec < 10) {
                time = Integer.toString(min) + ":" + "0" + Integer.toString(sec);
            } else if (min < 10) {
                time = "0" + Integer.toString(min) + ":" + Integer.toString(sec);
            } else {
                time = Integer.toString(min) + ":" + Integer.toString(sec);
            }
        }

        return time;
    }
```

openNew.java

In this case, we have created a method called only open. A string parameter is given to it. In that, the relevant file will be loaded according to the parameter we give. Here, the fxml loader is used to load an fxml file.



```
package player.tmplayer;

import ...


public class openNew {


    public static void onlyOpen(String x) throws IOException {

        Stage primaryStage = new Stage();
        Parent root = FXMLLoader.load(player.class.getResource(x));
        primaryStage.setScene(new Scene(root));
        primaryStage.show();

    }

}
```

Playlist.java

First of all, the necessary components related to fxml have been created.

```
public class Playlist extends player implements Initializable {
    @FXML
    private JFXButton v1;
    @FXML
    private JFXButton v2;
    @FXML
    private JFXButton v3;
    @FXML
    private JFXButton v4;
    @FXML
    private JFXButton v5;
    @FXML
    private JFXButton v6;
    @FXML
    private JFXButton v7;
    @FXML
    private JFXButton v8;
    @FXML
    private JFXButton v9;
    @FXML
    private JFXButton v10;
```

After that, the variables needed for the connection and the arrays needed for that have been created.

```
Connection con = null;
PreparedStatement pst = null;
ResultSet rs = null;
String[] te = {"","","","","","","","","",""};
String[] patharr = {"","","","","","","","","",""};
```

get last id in db and 10 items

From the database, we can get the details of ten videos in the playlist. The video name is put in the array called string[ ] te and the path is put in the array called string [ ] patharr. Since the limit is set to ten, this sql code should be run ten times.

```
//get names in db and 10 items
public void getnames() throws SQLException {
    String p = "select * from watchvideo order by ID desc limit 10";
    con = connectDB.connect();
    pst = con.prepareStatement(p);
    rs = pst.executeQuery();
    int x = 0;
    while (rs.next()){
        te[x] = rs.getString( columnLabel: "VideoName");
        patharr[x] = rs.getString( columnLabel: "Path");
        x=x+1;
```

In the while (rs.next ()), run 10 and run them separately. Meanwhile, when this 10 is run, the relevant value should be put, so x = x+1; We can get the relevant number by saying.

```
int x = 0;
while (rs.next()){
    te[x] = rs.getString( columnLabel: "VideoName");
    patharr[x] = rs.getString( columnLabel: "Path");
    x=x+1;
}
```

set value for button

set value for button, here the name of the button is set. For that, an array was created earlier. Here, v1, v2, v3 are the IDs given to the buttons. The name shown in the video is given as the name.

```
//set value for button
public void setname() throws SQLException {
    getnames();
    v1.setText(te[0]);
    v2.setText(te[1]);
    v3.setText(te[2]);
    v4.setText(te[3]);
    v5.setText(te[4]);
    v6.setText(te[5]);
    v7.setText(te[6]);
    v8.setText(te[7]);
    v9.setText(te[8]);
    v10.setText(te[9]);
}
```

Play video

Here, ten similar methods have been written for the ten related videos. In the 10th button, for example, in the 1st button, the path variable is given the value "0" in the path array. 2 is equal to the second value in the path array. The window we are in has been hidden. Call the class called onlyopen in the java class called opennew and give it as a parameter as fxml. open new only open(x : "player.fxml") Here what happens is that player.fxml is run and the video related to the value of path is run. In that way ten methods have been made.

```
//play video
public void b_v1() throws IOException {
    path = patharr[0];
    v9.getScene().getWindow().hide();
    openNew.onlyOpen( x: "player.fxml");
}
public void b_v2() throws IOException {
    path = patharr[1];
    v9.getScene().getWindow().hide();
    openNew.onlyOpen( x: "player.fxml");
}
public void b_v3() throws IOException {
    path = patharr[2];
    v9.getScene().getWindow().hide();
    openNew.onlyOpen( x: "player.fxml");
}
public void b_v4() throws IOException {
    path = patharr[3];
    v9.getScene().getWindow().hide();
    openNew.onlyOpen( x: "player.fxml");
}
public void b_v5() throws IOException {
    path = patharr[4];
    v9.getScene().getWindow().hide();
    openNew.onlyOpen( x: "player.fxml");
}
public void b_v6() throws IOException {
    path = patharr[5];
    v9.getScene().getWindow().hide();
    openNew.onlyOpen( x: "player.fxml");
}
```

Here the get name and set name methods have been called. get name is to get the names and paths of the videos. The value is set by the set name.

```
@Override
public void initialize(URL location, ResourceBundle resources) {
    try {
        getnames();
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
    try {
        setname();
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
```

Miniview.java

The items related to fxml are introduced in the miniview.

```
public class MiniView extends player{


    @FXML
    private MediaView play;

    @FXML
    private JFXSlider playbar;

    private MediaPlayer player;

    int x;
```

Play video

play video, if the path has a value, the video will be played. The sound before entering the mini view is set to this.

```
//play video

public void loadVideo(){
    if(path != null) {
        Media media = new Media(path);
        player = new MediaPlayer(media);
        play.setMediaPlayer(player);

        final DoubleProperty width = play.fitWidthProperty();
        final DoubleProperty height = play.fitHeightProperty();

        width.bind(Bindings.selectDouble(play.sceneProperty(), ...steps: "width"));
        height.bind(Bindings.selectDouble(play.sceneProperty(), ...steps: "height"));

        player.play();
        player.setVolume(sound);
```

Play

A method called play has been created here. For those methods, the method in the player is loaded. Same thing happens in pause.

```
//play

public void play(ActionEvent event) { player.play(); }

//pause

public void pause(ActionEvent event) { player.pause(); }
```

Back to main

This is the button that goes back to the player. First stop the video. What happens here is that the miniview that has been played is closed. Then we give player.fxml as the parameter of the open new we talked about.

```
//back

public void gotomain() throws IOException {
    player.stop();
    nTime = x;
    playbar.getScene().getWindow().hide();
    openNew.onlyOpen( x: "player.fxml");
}
```

Login.java

This is a code which is related to the Login form. First the relevant fxml items have been taken into this program.

```
package player.tmplayer;

import ...


public class Login implements Initializable {


    @FXML
    private Label e_all;

    @FXML
    private Label e_pw;

    @FXML
    private Label e_uname;

    @FXML
    private PasswordField pw;

    @FXML
    private TextField uname;
```

next other things related to the database the coming on the database area here we have created some variables. con from connection data type, pst from prepared statement, rs from resultset and make null all the all those three variables. then we have created a string array called info and inserted 5 items into it.

```
    Connection con = null;

    PreparedStatement pst = null;

    ResultSet rs = null;

    String[] info = {"","","","",""};
```

In the login form we have to check the username and password. So in this method we give two parameters called username and password. Both of them are string.

Then the throws keyword indicates what exception type may be drawn by a method. It can declare multiple exceptions. 'throws' is followed by a class and used with the method signature. so here we mention this throws SQL exception and IO exception..

So here we used SQL exception and IO exception. SQL exception is an exception that provides information on a database access error or other errors. Each SQL exception provide several kinds of information it is a string describing the error.

IO exception is related to Input and Output operations in the Java code. It happens when there is a failure during reading, writing and searching file or directory operations.

The database has been connected through the variable called 'con' taken from the connection data type. Then the SQL code that we want to run has been put in the variable called ' q'as a string. Then, to the prepare statement variable we call 'q' through 'con' variable. So in this level if we run the SQL code, the username and password in the table are checked one by one. If the given username and password in a line somewhere in the database, the array called 'info' , which is the fifth data, is set as successful and then hide the window that we were at that time.

Then we call 'onlyopen' method which is in the openNew class. There we gave analyze fxml as a parameter. Then it will ru. If the username or password is wrong, the label e_all will set text as "username or password is incorrect". Then set the e_all label visible. With this info array 4th index(5th data) will set as unsuccessful.

```
//check username and password

public void check(String userName, String passWord) throws SQLException, IOException {

    con = connectDB.connect();
    String q = "SELECT * FROM login where username=? and password=?";
    pst = con.prepareStatement(q);

    pst.setString( parameterIndex: 1, userName);
    pst.setString( parameterIndex: 2, passWord);
    rs = pst.executeQuery();

    if (rs.next()){
        info[4] = "Successful";
        uname.getScene().getWindow().hide();
        if (player.modeSet==1){
            openNew.onlyOpen( x: "Analize.fxml");
        } else if (player.modeSet==2){
            openNew.onlyOpen( x: "DataList.fxml");
        }
    }else {
        e_all.setText("Username or Password is incorrect");
        e_all.setVisible(true);
        info[4] = "Unsuccessful";
    }

}
```

Next we have to add status to the table. Here we have to put a table in the database to check who logged into the player and who try to log in. So, we created a method called 'add status'. temp string is set as parameter here. Then we have to run the SQL code to insert into use a logging values table. Adding the items which are index 0, 1, 2, 3 and 4 in the array of the term parameter is done by 'add status' method. From that database area ends.

```
//add status to table

public void addStatus(String[] temp) throws SQLException {
    con = connectDB.connect();
    String p = "INSERT INTO userlogin VALUES ('"+ temp[0] +"', '"+ temp[1] +"', '"+ temp[2] +"', '"+ temp[3] +"', '"+ temp[4]
    pst = con.prepareStatement(p);
    pst.execute();
}
```

Next we create a method called 'clear error'. There you can see three labels which were made to show the errors. They are e_pw e-uname and e_all. e_pw is to show the error in password, e_uname is to show the error in username and e_all is to show the errors in both username and password. Here by calling the 'clear error' method those three labels will be invisible by setvisible as 'false'.

```
//hide error lables

public void cleaererror(){
    e_pw.setVisible(false);
    e_uname.setVisible(false);
    e_all.setVisible(false);
}
```

Then we created method called 'login'. Here, username and password are taken as strings. There from uname.getText( ) we get the you username which is inserted by the user and from pw.getText( ) we get the password into relevant string. If there was any error in username or password, and it was displayed until now, then it will be disappear when we call the 'clear error' method.

As u can remember we created an array called 'info'. There, username in to first item info[0], password into second item info[1], local date into third item info[2] as a string value and current time in to forth item info[3].

Then we check that, if there is nothing in username in the login forum and press the login, it will set the text visible as enter username. Then we have to put the info[4] as unsuccessful. If there is nothing in the password and press the login, it will set the text visible as enter password. There also we have to put the info[4] as unuccessful.

If there is anything in the username or password we call 'check' method. We give username and password as two parameters. So here, it checked that the username and password are correct. If both of them are correct, login will be successful and open the player. If either one is wrong, then it will be display username or password is incorrect. After these steps happen, it will be call 'add status' method. Then all the four indexes in info array will display the details

usedname(used user name), usedpw(used password) , date(local date), time(logged time) and status(successful / unsuccessful) in a table.

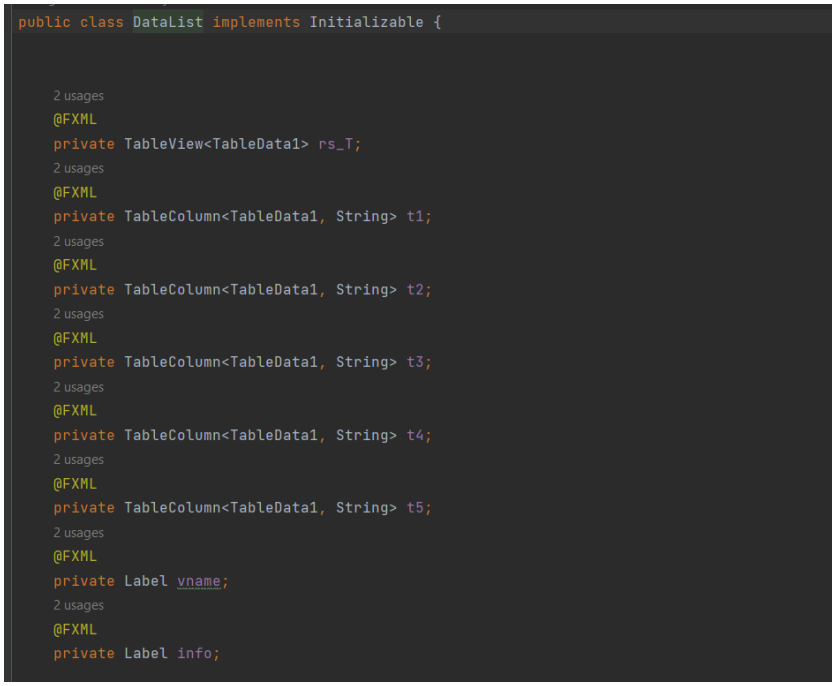
```
//for login button

public void login(ActionEvent event) throws IOException, SQLException {

    String userName = uname.getText();
    String passWord = pw.getText();

    clearerror();

    info[0] = userName;
    info[1] = passWord;
    info[2] = String.valueOf(LocalDate.now());
    info[3] = String.valueOf(LocalTime.now());

    if (userName.equals("")){
        e_all.setText("Enter Username");
        e_uname.setVisible(true);
        e_all.setVisible(true);
        info[4] = "Unsuccessful";
    }
    else if (passWord.equals("")) {
        e_all.setText("Enter Password");
        e_all.setVisible(true);
        e_pw.setVisible(true);
        info[4] = "Unsuccessful";
    }
    else {
        check(userName,passWord);
    }
    //add status to database
    addStatus(info);
```

For reset button also we call 'clear error' method. So, the errors will be invisible. Username and password will be set text. It means existing text will be removed from username and password and will look empty.

```
//for reset button

public void reset(){
    clearerror();
    uname.setText("");
    pw.setText("");
}
```

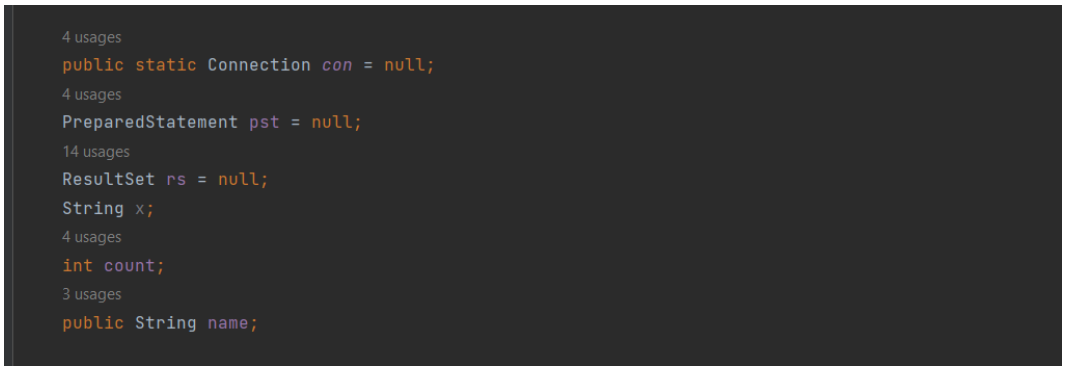
Datalist.java

The class dataList is created to take data from the getter files. They are created as privet since the data doesn't want to leak. Here tableData1 is the class which we have created to gain data to this class. To recognize those columns separately we named those as T1, T2 etc. To set value we have created two lables called vname and info.



```
public class DataList implements Initializable {


    2 usages
    @FXML
    private TableView<TableData1> rs_T;
    2 usages
    @FXML
    private TableColumn<TableData1, String> t1;
    2 usages
    @FXML
    private TableColumn<TableData1, String> t2;
    2 usages
    @FXML
    private TableColumn<TableData1, String> t3;
    2 usages
    @FXML
    private TableColumn<TableData1, String> t4;
    2 usages
    @FXML
    private TableColumn<TableData1, String> t5;
    2 usages
    @FXML
    private Label vname;
    2 usages
    @FXML
    private Label info;
```

Here are the variables which are need to connect with database.



```
    4 usages
    public static Connection con = null;
    4 usages
    PreparedStatement pst = null;
    14 usages
    ResultSet rs = null;
    String x;
    4 usages
    int count;
    3 usages
    public String name;
```

Using this method, we can load the table. Here we run the required sql query. Then the code will run line by line using a while loop.

```
public void loadTb() throws SQLException {

    //sql part
    String p = "select S_date,S_time,E_date,E_time,(E_time-S_time)/60 from watchvideo where VideoName = '"+ player.fname +"'";
    con = connectDB.connect();
    pst = con.prepareStatement(p);
    rs = pst.executeQuery();

    String[] temp = {"","","","","",""};
    ObservableList<TableData1> list =  FXCollections.observableArrayList();

    while (rs.next()) {

        temp[0] = rs.getString( columnLabel: "S_date");
        temp[1] = rs.getString( columnLabel: "S_time");
        temp[2] = rs.getString( columnLabel: "E_date");
        temp[3] = rs.getString( columnLabel: "S_time");
        temp[4] = rs.getString( columnLabel: "(E_time-S_time)/60");

        list.add(new TableData1(temp[0],temp[1],temp[2],temp[3],temp[4]));
        rs_T.setItems(list);

        count = count + 1;

    }

    t1.setCellValueFactory(new PropertyValueFactory<TableData1, String>("tr1"));
    t2.setCellValueFactory(new PropertyValueFactory<TableData1, String>("tr2"));
    t3.setCellValueFactory(new PropertyValueFactory<TableData1, String>("tr3"));
    t4.setCellValueFactory(new PropertyValueFactory<TableData1, String>("tr4"));
    t5.setCellValueFactory(new PropertyValueFactory<TableData1, String>("tr5"));
```

Set the video's name for the label called vname which is already created. And also check whether how many times wached that particular video, set playtime for label called info.

```
1 usage   Lakruwan Jayathissa
public void mainInfo() throws SQLException, IOException {
    vname.setText(player.fname);
    info.setText("     Played times : "+ count + "\tTotal played time : ");
}
```

We can add all the data to a text file as our wish. Its name format will be display as video name_localdate.txt.

```
1 usage   Lakruwan Jayathissa
public void txtfile() throws IOException, SQLException {

    //file generate
    name = player.fname + "_" + LocalDate.now() + ".txt";
    File f = new File(name);
    f.createNewFile();
```

Video's data will be added to the text file along with the heading of existing table.

```
    //file write
    fww.write( str: "Video Name : "+ player.fname + "\n");
    fww.write( str: "Played times : "+ count + "\nTotal played time : \n\n");
    fww.write( str: "+-----------------+-----------------+-----------------+-----------------+-----------+\n");
    fww.write( str: "| Starting Date" + "\t| Starting Time" + "\t| Ending Date" + "\t| Ending Time" + "\t| Duration \t|\n");
    fww.write( str: "+-----------------+-----------------+-----------------+-----------------+-----------+\n");

    //sql part
    String p = "select S_date,S_time,E_date,E_time,(E_time-S_time)/60 from watchvideo where VideoName = '" + player.fname + "'";
    con = connectDB.connect();
    pst = con.prepareStatement(p);
    rs = pst.executeQuery();

    //write the data witch are in db
    while (rs.next()) {
        fww.write(
            str: "| " + rs.getString( columnLabel: "S_date") +
            "\t| " + rs.getString( columnLabel: "S_time") +
            "\t\t| " + rs.getString( columnLabel: "E_date") +
            "\t| " + rs.getString( columnLabel: "S_time") +
            "\t\t| " + rs.getString( columnLabel: "(E_time-S_time)/60") +
            "\t|\n");
    }
    fww.write( str: "+-----------------+-----------------+-----------------+-----------------+-----------+\n");
    fww.close();
}

 Lakruwan Jayathissa
@Override
public void initialize(URL location, ResourceBundle resources){
```

ConnectDB.java

What happens with this code is that when the connectDB method is that when the connect method is called, the database is connected according to the information we have given. First, we need to create the parameters to connect the database. Then the variable should be created as a string data type such as host as host name, port, user name, password, dbname as database name. These variables can change from lap to lap, so they are used with invented commas so we can change them as needed. When connecting to the database, the URL should be created automatically. It is normally, **jdbc: mysql://host :+port+/dbname** these are stored in the string datatype. The previously entered host name, port , and database name will be added to these strings. The URL is created as a string as we want. Here, the + sign is used to concatenate the string.

```
package player.tmplayer;

import ...


public class connectDB {


    private static String host = "localhost";

    private static String port = "3306";

    private static String username = "root";

    private static String password = "";

    private static String dbname = "playerdb";


    private static String url = "jdbc:mysql://"+host +":"+port+"/"+dbname;
```

Next, the connect method should be created. Its data type is connection. A variable must be created as conn from the connection datatype. Then we can return the conn as desired. which is null. This try-catch part is used for error handling. which is used to display errors and prevent software crashes. This conn variable is connected to the database and finally returns the conn variable.

```java
public static Connection connect(){

    Connection conn = null;

    try {
        Class.forName( className: "com.mysql.cj.jdbc.Driver");
    } catch (ClassNotFoundException e) {
        throw new RuntimeException(e);
    }

    try {
        conn = DriverManager.getConnection(url,username,password);
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }

    return conn;
}
```

TableData1.java

Collecting data from the database to a table is mainly what happens here. First of all, we have created the variable as required.

```java
package player.tmplayer;

import javafx.beans.property.SimpleStringProperty;


public class TableData1 {


    private SimpleStringProperty tr1 = new SimpleStringProperty();

    private SimpleStringProperty tr2 = new SimpleStringProperty();

    private SimpleStringProperty tr3 = new SimpleStringProperty();

    private SimpleStringProperty tr4 = new SimpleStringProperty();

    private SimpleStringProperty tr5 = new SimpleStringProperty();
```

Then the constructor has been created in this.

```
public TableData1(String tr1, String tr2, String tr3, String tr4, String tr5) {
    this.tr1 = new SimpleStringProperty(tr1);
    this.tr2 = new SimpleStringProperty(tr2);
    this.tr3 = new SimpleStringProperty(tr3);
    this.tr4 = new SimpleStringProperty(tr4);
    this.tr5 = new SimpleStringProperty(tr5);
}
```

It has five parameters. Because these variables are private, they cannot be given at once from the database, so they are kept private. For that we use get Tr. Get Tr related to them has been created in these.

```
public String getTr1() { return tr1.get(); }

public SimpleStringProperty tr1Property() { return tr1; }

public String getTr2() { return tr2.get(); }

public SimpleStringProperty tr2Property() { return tr2; }

public String getTr3() { return tr3.get(); }

public SimpleStringProperty tr3Property() { return tr3; }

public String getTr4() { return tr4.get(); }

public SimpleStringProperty tr4Property() { return tr4; }

public String getTr5() { return tr5.get(); }

public SimpleStringProperty tr5Property() { return tr5; }
```

Analyze.java

First we create analyze method and it extend with analyze2. In this step we use inherit oop concept. After that we create all the variables as we need. Next we crate to print like “Trust me player” and “All You Gotta Do is TrustMe” because this is console application.

```
package player.tmplayer;

import ...

public class Analyze extends Analyze2{

    public static int sch1_;
    public static int sch2_;
    public static Scanner scan = new Scanner(System.in);

    //variable for search
    public static String cname;
    public static String sname;
    public static int again;
    public static int textfile;
    public static int limit;


    public static void main(String[] args) throws SQLException {

        //start
        System.out.println("+-------------------------------------------------------------------------------+");
        System.out.println("|                              >>> TrustMe Player <<<                           |");
        System.out.println("|                         *-------------------------------*                     |");
        System.out.println("|                          ~ All You Gotta Do is TrustMe ~                      |");
        System.out.println("+ - - ------------------------------------------------------------------ - - +\n\n");
```

Main option

First we get user’s choice and after that it assign option_ int variable then select a variable and after that go to rest of the program if option is 1 call sch() method and if option is 2 call al() if the option is 3 print the these outputs and go the input to limit. After that call this w_his(); method. If option is 5 we create to print members of our developing team else print as incorrect input and call main method.

```
        // main options
        System.out.println("Options :");
        System.out.println("  Search ---> \t\t\t\tPress 1");
        System.out.println("  Get Analyze List ---> \tPress 2");
        System.out.println("  Watch History ---> \t\tPress 3");
        System.out.println("  Login History ---> \t\tPress 4");
        System.out.println("  Developing team --->  \tPress 5");
        System.out.print("\nEnter your choice : ");
        int option_ = scan.nextInt();
        System.out.println("\n- - - ------------------------------------------------------------------------------+\n");

        if (option_ == 1){

            sch();

        }else if (option_ == 2){

            al();

        }else if (option_ == 3){
```

```
            // 3 - Watch History
            System.out.println(" *----- Watch History");
            System.out.print("\nEnter the number of rows : ");
            limit = scan.nextInt();
            System.out.println("\n- - - ---------------------------------------------------------------------------- - - +\n");
            w_his();

        }else if (option_ == 4){

            // 4 - Login History
            System.out.println(" *----- Login History");
            System.out.print("\nEnter the number of rows : ");
            limit = scan.nextInt();
            System.out.println("\n- - - ---------------------------------------------------------------------------- - - +\n");
            l_his();
        }else if (option_ == 4){

            // 4 - Login History
            System.out.println(" *----- Login History");
            System.out.print("\nEnter the number of rows : ");
            limit = scan.nextInt();
            System.out.println("\n- - - ---------------------------------------------------------------------------- - - +\n");
            l_his();
```

```
        }else if (option_ == 5){
            System.out.println("+-------------------------------------------------------------------------------------+");
            System.out.println("|                              >>> TrustMe Player <<<                                 |");
            System.out.println("|                           ~ Developing Team Members ~                               |");
            System.out.println("+-------------------------------------------------------------------------------------+");
            System.out.println("|                                                                                     |");
            System.out.println("|\t* Fathima Saliha \t\t\t0742291277                                              |");
            System.out.println("|\t* Lakruwan Jayathissa \t\t0772529441                                              |");
            System.out.println("|\t* Hashini Sulakshana \t\t0701259920                                              |");
            System.out.println("|\t* Niwandi Githma \t\t\t0716515705                                              |");
            System.out.println("|\t* Thakshila Dulanjani \t\t0762915371                                              |");
            System.out.println("|                                                                                     |");
            System.out.println("+-------------------------------------------------------------------------------------+");

        } else {
            System.out.println("Enter the valid input ");
            System.out.println("*************************************************************************************\n");
            main( args: null);
        }

    }
```

Sch( )

If user select option 1 in search run sch(); method in this method there are two ways we can search Table with watch history and Table with login data. Then get to values for sch_ int variable from above tables. After that check column of the table from user and assign it sch1_ variable. If sch1_ equal to 2 or 4 go to date format. If sch1_ equal to 3 or 5 go to time format. If sch1_ equal to 1 say to user to Enter the key word also in this step display the video name. get necessary key word for sname. After that call search1 method if select sch_2 equal to 2 check the column and assign it sch2_ variable If sch2_ equal to 2 go to date format. If sch2_ equal to 3 go to time format. If sch2_ equal to 1 say to user to Enter the key word also in this step display the video name. if sch2_ equal to 4 say to Enter the value /(Successful/Unsuccessful/). get necessary key word for sname. After that call search2 method else print Enter the valid input and after that call sch() method.

```
public static void sch() throws SQLException {

    System.out.println(" *----- Search");
    System.out.println("Select the proper table :");
    System.out.println("  Table with watch history --> \tPress 1");
    System.out.println("  Table with login data --> \tPress 2");
    System.out.print("\nEnter your choice : ");
    int sch_ = scan.nextInt();
    System.out.println("\n- - - ---------------------------------------------------------------------------+\n");

    if (sch_ == 1){

        // 1 - 1 - Table with watch history
        System.out.println(" *----- Table with watch History");
        System.out.println("Select the required column name : ");
        System.out.println("  Video name --> \tPress 1");
        System.out.println("  Starting date --> Press 2");
        System.out.println("  Starting time --> Press 3");
        System.out.println("  End date --> \t\tPress 4");
        System.out.println("  End time --> \t\tPress 5");

        System.out.print("\nEnter your choice : ");
        sch1_ = scan.nextInt();

     } else if (sch_ == 2){

         // 1 - 2 - Table with login history
         System.out.println(" *----- Table with login history");
         System.out.println("Select the required column name : ");
         System.out.println("  User name --> \tPress 1");
         System.out.println("  Date --> \t\t\tPress 2");
         System.out.println("  Time --> \t\t\tPress 3");
         System.out.println("  Status --> \t\tPress 4");

         System.out.print("\nEnter your choice : ");
         sch2_ = scan.nextInt();

         if (sch2_==2){
             System.out.println("(Date format :- YYYY-MM-DD)");
             System.out.print("Enter the Date : ");
         } else if (sch2_==3){
             System.out.println("(Time format :- HH:MM:SS)");
             System.out.print("Enter the Time : ");
         } else if (sch2_==1){
             System.out.print("Enter the key word : ");
         }else if (sch2_==4){
             System.out.print("Enter the value (Successful/Unsuccessful) : ");
         }
         sname = scan.next();
         System.out.println("\n- - - ---------------------------------------------------------------------------+\n");

         search2();

    } else{
        System.out.println("Enter the valid input ");
        System.out.println("*******************************************************************************\n");
        sch();
    }

}
```

al( )

when we go the analyze there are so many options in this method. They are hour by hour, day by day , year by year, or as we specialize option Details of a hour, Details of a day, Details of a year. Next we assign different kinds of queries for q string. first we create new object using analyze 2 class called as a2. After that run a2.an(q) this method g as a parameter. Then print analyze list according our requirement. If user need text run tsch_1() method. If user need to run again run al() method. Other wise close the programme.

```
public static void al() throws SQLException {
    // 2 - Analyze
    System.out.println(" *----- Get Analyze List");
    System.out.println("Which time period do you prefer ?");
    System.out.println("  Hour by hour --> \t\t\tPress 1");
    System.out.println("  Day by day --> \t\t\tPress 2");
    System.out.println("  Month by month --> \t\tPress 3");
    System.out.println("  Year by Year --> \t\t\tPress 4");
    System.out.println("  Details of a hour --> \tPress 5");
    System.out.println("  Details of a day --> \t\tPress 6");
    System.out.println("  Details of a month --> \tPress 7");
    System.out.println("  Details of a year --> \tPress 8");
    System.out.print("\nEnter your choice : ");
    int ana = scan.nextInt();
    System.out.println("\n- - - ------------------------------------------------------------------------------+\n");

    String q = null;

    if (ana==1){

        q = "SELECT * FROM watchvideo ORDER BY S_time DESC'";

    } else if (ana==2) {

        q = "SELECT * FROM watchvideo ORDER BY date(S_date) DESC'";

    } else if (ana==3) {

        q = "SELECT * FROM watchvideo ORDER BY month(S_date) DESC'";

    } else if (ana==4) {

        q = "SELECT * FROM watchvideo ORDER BY year(S_date) DESC'";
    } else if (ana==5) {
        // 2 - 5 - Analyze
        System.out.println(" *----- Details of a hour");
        System.out.print("\nEnter the required hour: ");
        int a = scan.nextInt();
        q = "SELECT * FROM watchvideo where hour(S_date) = " + a;
        System.out.println("\n- - - ------------------------------------------------------------------------------+\n");
    } else if (ana==6) {
        // 2 - 6 - Analyze
        System.out.println(" *----- Details of a day");
        System.out.print("\nEnter the required day: ");
        int a = scan.nextInt();
        System.out.println("(format :- 1,2,3...30,31)");
        q = "SELECT * FROM watchvideo where day(S_date) = " + a;
        System.out.println("\n- - - ------------------------------------------------------------------------------+\n");
    } else if (ana==7) {
        // 2 - 7 - Analyze
        System.out.println(" *----- Details of a month");
        System.out.print("\nEnter the required month: ");
        System.out.println("(format :- 1,2,3...23,24)");
        int a = scan.nextInt();
        q = "SELECT * FROM watchvideo where month(S_date) = " + a;
        System.out.println("\n- - - ------------------------------------------------------------------------------+\n");
```

```
    } else if (ana==8) {
        // 2 - 8 - Analyze
        System.out.println(" *----- Details of a year");
        System.out.print("\nEnter the required year: ");
        System.out.println("(format :- 2022,2023...)");
        int a = scan.nextInt();
        q = "SELECT * FROM watchvideo where year(S_date) = " + a;
        System.out.println("\n- - - ----------------------------------------------------------------------------+\n");
    } else {
        System.out.println("Enter the valid input ");
        System.out.println("*******************************************************************************\n");
        al();
    }
    }

    Analyze2 a2 = new Analyze2();
    a2.an(q);

    //word file
    int i = 0;
    while(i==0){
        System.out.print("\nDo you need text file for this (Yes >> 1 / No  >> 2) : ");
        again = scan.nextInt();
        System.out.println(again);
        if (again == 1){
            try {
                tsch_1();
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
            i = i + 1;
        } else if (again == 2){
            i = i + 1;
        }
    }
```

```
    //run again
    int j = 0;
    while(j==0){
        System.out.print("\nDo you want run this again (Yes >> 1 / No  >> 2), Go to main >> 3 : ");
        textfile = scan.nextInt();
        if (textfile == 1){
            al();
            j = j + 1;
        } else if (textfile == 2){
            j = j + 1;
        } else if (textfile == 3){
            main( args: null);
            j = j + 1;
        }
    }

}
```

Search1( )

In this method run using value of the sch_1. In here according to value of the sch_1 assign data base table name to cname variable. After that call sch_1() method. If user give incorrect input call again this method. In this method run according to watch video table.

```
public static void search1() throws SQLException {
    if (sch1_==1){
        cname = "VideoName";
        sch_1();
    } else if (sch1_==2){
        cname = "S_date";
        sch_1();
    }  else if (sch1_==3){
        cname = "S_time";
        sch_1();
    }else if (sch1_==4){
        cname = "E_date";
        sch_1();
    }else if (sch1_==5){
        cname = "E_time";
        sch_1();
    }else {
        System.out.println("Enter the valid input ");
        System.out.println("**********************************************************************************\n");
        search1();
    }
}
```

Search2( )

In this method run using value of the sch_2. In here according to value of the sch_2 assign data base table name to cname variable. After that call sch_2() method. If user give incorrect input call again this method. In this method run according to user login table.

```
public static void search2() throws SQLException {

    if (sch2_==1){
        cname = "usedName";
        sch_2();
    } else if (sch2_==2){
        cname = "Date";
        sch_2();
    } else if (sch2_==3){
        cname = "Time";
        sch_2();
    }else if (sch2_==4){
        cname = "Status";
        sch_2();
    }else {
        System.out.println("Enter the valid input ");
        System.out.println("**********************************************************************************\n");
        search2();
    }
}
```

Sch_1( )

first we create new object using analize 2 class called as a2. Then we create sql quarry using earlier values. After that this query give a2.an method and called this method. There for call the a2.an(q) method which is in analyze2 class. In this we must give string as a parameter we give above quarry for this parameter. So print the necessary output.

Word file

In this step we ask need to run the word file if user need run tsch_1(); method and create necessary text file else exit from while loop.

Run again

In this if we need run again code call sch() method else close the program

```
public static void sch_1() throws SQLException {

    Analyze2 a1 = new Analyze2();
    String q = "SELECT * FROM watchvideo where " + cname + " like \'%" + sname + "%\'";
    a1.an(q);

    //word file
    int i = 0;
    while(i==0){
        System.out.print("\nDo you need text file for this (Yes >> 1 / No  >> 2) : ");
        again = scan.nextInt();
        System.out.println(again);
        if (again == 1){
            try {
                tsch_1();
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
            i = i + 1;
        } else if (again == 2){
            i = i + 1;
        }
    }
    //run again
    int j = 0;
    while(j==0){
        System.out.print("\nDo you want run this again (Yes >> 1 / No  >> 2), Go to main >> 3 : ");
        textfile = scan.nextInt();
        if (textfile == 1){
            sch();
            j = j + 1;
        } else if (textfile == 2){
            j = j + 1;
        } else if (textfile == 3){
            main( args: null);
            j = j + 1;
        }
    }

}
```

Sch_2( )

first we create new object using analize 2 class called as an2. Then we create sql quarry using earlier values. After that this query give a2.an method and called this method.

Word file

In this step get the value for again and if text file variable value is 1 call tsch_2() method and create necessary text file else exit from while loop.

Run again

In this if we need run again code call sch() method else close the program

```
public static void sch_2() throws SQLException {

    //sql part
    Analyze2 a2 = new Analyze2();
    String q = "SELECT * FROM userlogin where " +cname+ " like '%" +sname+ "%'";
    a2.an2(q);

    //word file
    int i = 0;
    while(i==0){
        System.out.print("\nDo you need text file for this (Yes >> 1 / No  >> 2) : ");
        textfile = scan.nextInt();
        System.out.println(again);
        if (textfile == 1){
            try {
                tsch_2();
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
            i = i + 1;
        } else if (textfile == 2){
            i = i + 1;
        }
    }
```

```
    //run again
    int j = 0;
    while(j==0){
        System.out.print("\nDo you want run this again (Yes >> 1 / No  >> 2), Go to main >> 3 : ");
        again = scan.nextInt();
        if (again == 1){
            sch();
            j = j + 1;
        } else if (again == 2){
            j = j + 1;
        } else if (again == 3){
            main( args: null);
            j = j + 1;
        }
    }

}
```

w_his( )

get the value for limit. Which is public variable. first we create new object using analize 2 class called as a2. Then we create sql quarry using earlier values. If user need word file create a word file running tw_his() method. This steps are near to watch video table.

go to main

In here ask from user need go to main if yes go to main

```
public static void w_his() throws SQLException {

    String q = "SELECT * FROM watchvideo order by ID desc limit "+limit+"";
    Analyze2 a1 = new Analyze2();
    a1.an(q);

    //word file
    int i = 0;
    while(i==0){
        System.out.print("\nDo you need text file for this (Yes >> 1 / No  >> 2) : ");
        again = scan.nextInt();
        System.out.println(again);
        if (again == 1){
            try {
                tw_his();
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
            i = i + 1;
        } else if (again == 2){
            i = i + 1;
        }
    }
    //go to main
    int j = 0;
    while(j==0){
        System.out.print("\nGo to main >> 1 : ");
        textfile = scan.nextInt();
        if (textfile == 1) {
            main( args: null);
            j = j + 1;
        }else{
            j = j + 1;
        }
    }

}
```

l_his( )

first we create new object using analize 2 class called as an2. Then we create sql quarry using earlier values. If user need word file create a word file running tl_his() method. This steps are near to watch video table. This steps are near to userlogin table.

```
public static void l_his() throws SQLException {

    String q ="SELECT * FROM userlogin order by Date desc limit "+ limit +"";
    Analyze2 a2 = new Analyze2();
    a2.an2(q);

    //word file
    int i = 0;
    while(i==0){
        System.out.print("\nDo you need text file for this (Yes >> 1 / No  >> 2) : ");
        again = scan.nextInt();
        System.out.println(again);
        if (again == 1){
            try {
                tl_his();
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
            i = i + 1;
        } else if (again == 2){
            i = i + 1;
        }
    }
    //go to main
    int j = 0;
    while(j==0){
        System.out.print("\nGo to main >> 1 : ");
        textfile = scan.nextInt();
        if (textfile == 1) {
            main( args: null);
            j = j + 1;
        }else{
            j = j + 1;
        }
    }

}

// ------------ Database controlling part end ---------------


// ------------ Textfile creating part start ---------------
```

tsch_1( )

first we create new object using analize 2 class called as a3. Then we create sql quarry using earlier values. And run a3.an3(q) method.

```
public static void tsch_1() throws SQLException, IOException {

    Analyze2 a3 = new Analyze2();
    String q = "SELECT * FROM watchvideo where " +cname+ " like '%" +sname+ "%'";
    a3.an3(q);

}
```

tsch_2( )

first we create new object using analize 2 class called as a4. Then we create sql quarry using earlier values. And run a4.an4(q) method.

```
    public static void tsch_2() throws SQLException, IOException {

        Analyze2 a4 = new Analyze2();
        String q = "SELECT * FROM userlogin where " +cname+ " like '%" +sname+ "%'";
        a4.an4(q);

    }
```

tw_his( )

first we create new object using analize 2 class called as a3. Then we create sql quarry using earlier values. And run a3.an3(q) method.

```
    public static void tw_his() throws SQLException, IOException {

        Analyze2 a3 = new Analyze2();
        String q = "SELECT * FROM watchvideo where " +cname+ " like '%" +sname+ "%'";
        a3.an3(q);

    }
```

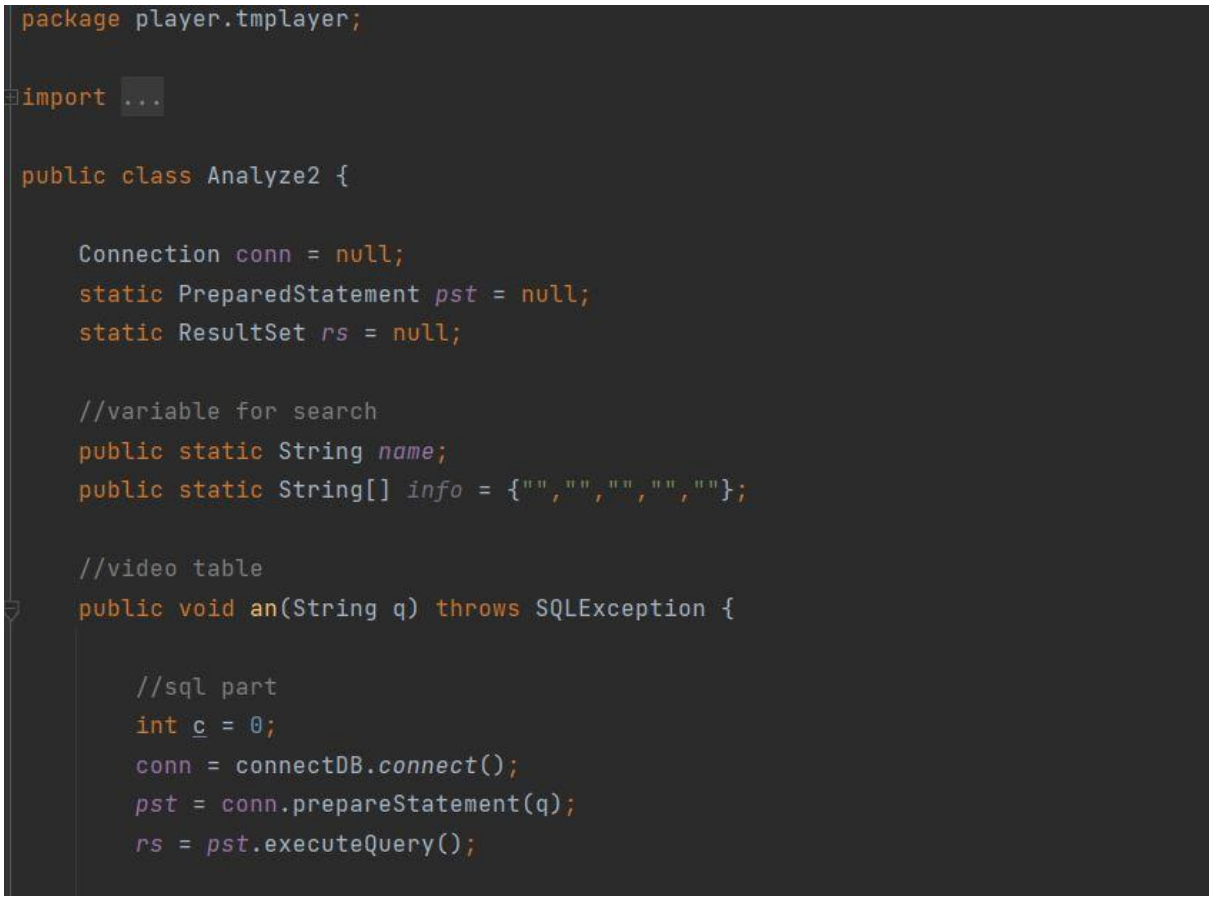
tl_his( )

first we create new object using analize 2 class called as a4. Then we create sql quarry using earlier values. And run a4.an4(q) method

```
    public static void tl_his() throws SQLException, IOException {

        Analyze2 a4 = new Analyze2();
        String q = "SELECT * FROM userlogin where " +cname+ " like '%" +sname+ "%'";
        a4.an4(q);

    }

    // ------------- Textfile creating part end ---------------

}
```

Analyze2.java

There four method first we create necessary variables. We create getcon method for build the connection. This an method use for display the data watch video's table. Here we must give sql part to the method as a parameter after that necessary code has been run and print in the control. Second method is an2 in this method print userlogin's table data as our required and we must give parameters correctly in necessary table.



```
package player.tmplayer;

import ...

public class Analyze2 {

    Connection conn = null;
    static PreparedStatement pst = null;
    static ResultSet rs = null;

    //variable for search
    public static String name;
    public static String[] info = {"","","","",""};

    //video table
    public void an(String q) throws SQLException {

        //sql part
        int c = 0;
        conn = connectDB.connect();
        pst = conn.prepareStatement(q);
        rs = pst.executeQuery();
```

```
        //printing part
        System.out.println("+---------------+---------------+---------------+---------------+---------------+-------------------------");
        System.out.println("| Starting Date"  + "\t| Starting Time" + "\t| End Date\t" + "\t| End Time\t" + "\t| Duration\t" + "\t| Video Name");
        System.out.println("+---------------+---------------+---------------+---------------+---------------+-------------------------");

        while (rs.next()) {
            System.out.println(
                "| " +
                rs.getString( columnLabel: "S_date")+"\t| "+
                rs.getString( columnLabel: "S_time")+"\t\t| "+
                rs.getString( columnLabel: "E_date")+"\t| "+
                rs.getString( columnLabel: "E_time")+"\t\t| "+
                "\t\t\t\t| "+
                rs.getString( columnLabel: "VideoName")
            );
            c = c + 1;
        }
        if (c==0){
            System.out.println("|\t\t No content found");
        }
```

```
        c = c + 1;
    }
    if (c==0){
        System.out.println("|\t\t No content found");
    }
    System.out.println("+--------------+--------------+--------------+--------------+--------------+----------------------");
}
```

```
//login data table
public void an2(String q) throws SQLException {

    int c = 0;
    conn = connectDB.connect();
    pst = conn.prepareStatement(q);
    rs = pst.executeQuery();

    //printing part
    System.out.println("+---------------+---------------+---------------+---------------");
    System.out.println("| Date" + "\t\t\t| Time\t\t" + "\t| Status" + "\t\t| User Name");
    System.out.println("+---------------+---------------+---------------+---------------");

    while (rs.next()) {
        System.out.println(
            "| " +
            rs.getString( columnLabel: "Date")+"\t| "+
            rs.getString( columnLabel: "Time")+"\t\t| "+
            rs.getString( columnLabel: "Status")+"\t| "+
            rs.getString( columnLabel: "usedName")+""
        );
        c = c + 1;
    }
    if (c==0){
        System.out.println("|\t\t No content found");
    }
    System.out.println("+---------------+---------------+---------------+---------------");
}
```

From this an 3 method data goes to a text file and write on it. We created sql q parameter and get data which is related to that parameter. Then run the code. A new file will be created called temp.txt. The header of the existing table will be printed already in that file. Data will write on that file as user's request. If there is no content to show, it'll print "No content found" and continue.

```
//video table >> text file
public void an3(String q) throws SQLException, IOException {

    int c = 0;
    conn = connectDB.connect();
    pst = conn.prepareStatement(q);
    rs = pst.executeQuery();

    //file generate
    name = "temp.txt";
    File f = new File(name);
    f.createNewFile();
    FileWriter fw = new FileWriter(name);
    BufferedWriter fww = new BufferedWriter(fw);

    fww.write( str: "+----------------+----------------+----------------+----------------+----------------+----------------------\n");
    fww.write( str: "| Starting Date"  + "\t| Starting Time" + "\t| End Date\t" + "\t| End Time\t" + "\t| Duration\t" + "\t| Video Name\n");
    fww.write( str: "+----------------+----------------+----------------+----------------+----------------+----------------------\n");
```

```
        while (rs.next()) {

            fww.write(
                 str: "| " + rs.getString( columnLabel: "S_date") +
                "\t| " + rs.getString( columnLabel: "S_time") +
                "\t\t| " + rs.getString( columnLabel: "E_date") +
                "\t| " + rs.getString( columnLabel: "S_time") +
                "\t\t\t|"+
                 "\t\t| " + rs.getString( columnLabel: "VideoName") +
                 "\n"
            );
            c = c + 1;
        }

        if (c==0){
            fww.write( str: "|\t\t No content found");
        }
        fww.write( str: "+-----------------+----------------+----------------+---------------+---------------+----------------------\n");
        fww.close();

        System.out.println("Textfile generated");
    }
```

An 4 will do the same thing as an 3. The difference is that an 3 controls the watch video table. The table called user login can be controlled by an 4.

```
    //login data table >> text file
    public void an4(String q) throws SQLException, IOException {

        int c = 0;
        conn = connectDB.connect();
        pst = conn.prepareStatement(q);
        rs = pst.executeQuery();

        //file generate
        name = "temp.txt";
        File f = new File(name);
        f.createNewFile();
        FileWriter fw = new FileWriter(name);
        BufferedWriter fww = new BufferedWriter(fw);

        fww.write( str: "+------------------+-----------------+-----------------+--------------------\n");
        fww.write( str: "| Date" + "\t\t| Time\t" + "\t| Status" + "\t\t| User Name\n");
        fww.write( str: "+------------------+-----------------+-----------------+--------------------\n");
```

```
        while (rs.next()) {
            System.out.println(
                "| " +
                rs.getString( columnLabel: "Date") + "\t| " +
                rs.getString( columnLabel: "Time") + "\t\t| " +
                rs.getString( columnLabel: "Status") + "\t| " +
                rs.getString( columnLabel: "usedName") + ""
            );
            c = c + 1;
        }
        if (c == 0) {
            fww.write( str: "|\t\t No content found");
        }
        fww.write( str: "+-----------------+----------------+----------------+---------------+---------------+----------------------\n");
        fww.close();

        System.out.println("Textfile generated");

    }

}
```

Database part

Login table

You have to log in with username and password.

username	password

It will run only if the password for this username is entered correctly. Especially in this username and password capital, simple cannot be changed. Also, if there is one more space, an error will occur and you will not be able to log in. The username and password should be exactly the same as this format.

User Login table

In this case, all the cases where a user tried to log in to this will be updated in the database. For example, try to log in by entering the username and password, we can check whether it was possible to log in or not.

Even if you click the login button without entering a user name, it will definitely be updated in this database. And the following things are updated from this database,

1. username
2. password
3. The date the user tried to log in
4. The time the user tried to log in
5. status

This means whether the user tried to log in was successful or unsuccessful.

usedName	usedPw	Date	Time	Status

Watch Video table

Because we usually watch a lot of videos in this, we have introduced an ID to analyze all of them. This ID falls in the viewing order from zero to the top. This ID is assigned to each video. Before starting the program, we will check and see what the ID is. If there is no ID, it will be assigned as "0".

Then the video name, format (as mp4 or AVI), and where the video is saved on the local computer will be added.

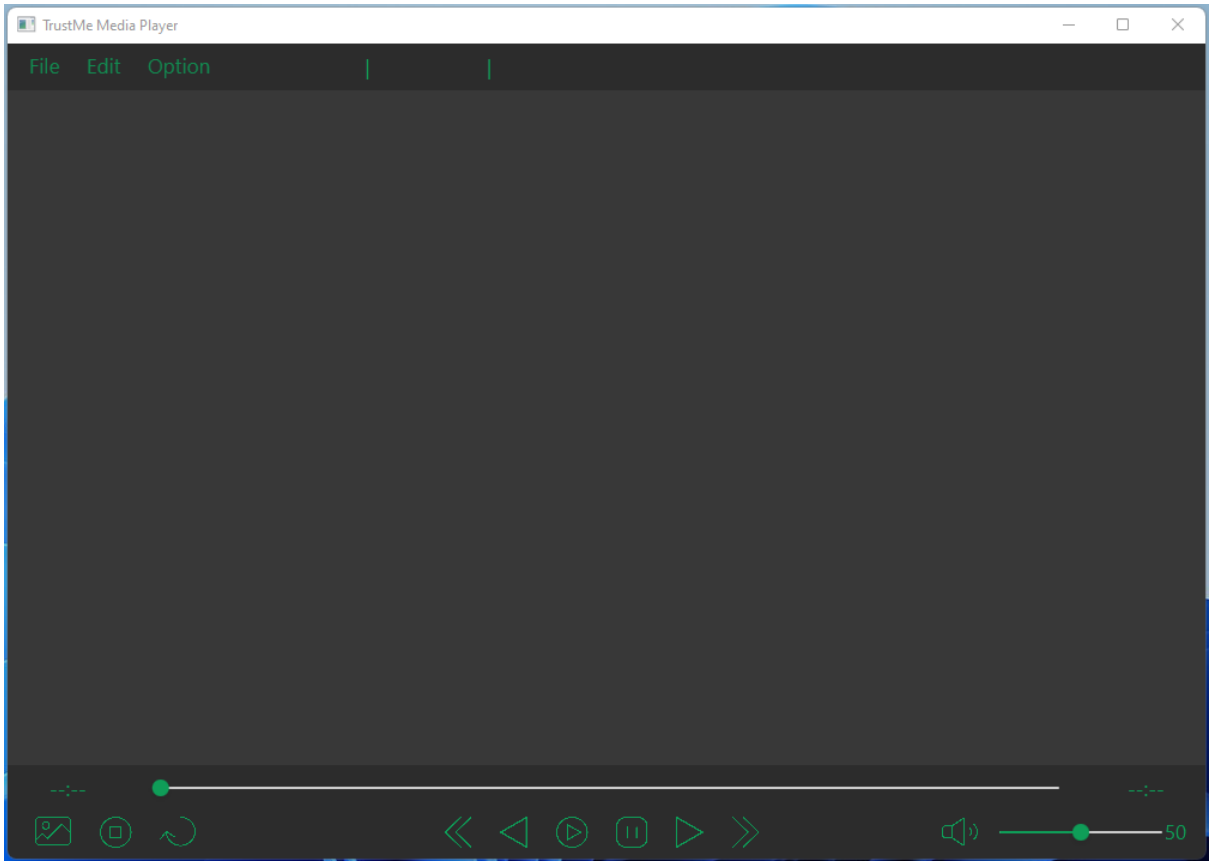
- S_date - The date the video was started.
- S_time - The time the video was started.
- E_date - The day the video ended.
- E_time - The time when the video ended.

Example :- If the video is finished from the mini view, it will also be updated in the database. Even if the video is stopped, it will be updated in the database.

ID	VideoName	Format	Path	S_date	S_time	E_date	E_time

Opening and working

When you first start and run TmPlayer.java, It is as shown in the screen shot below.



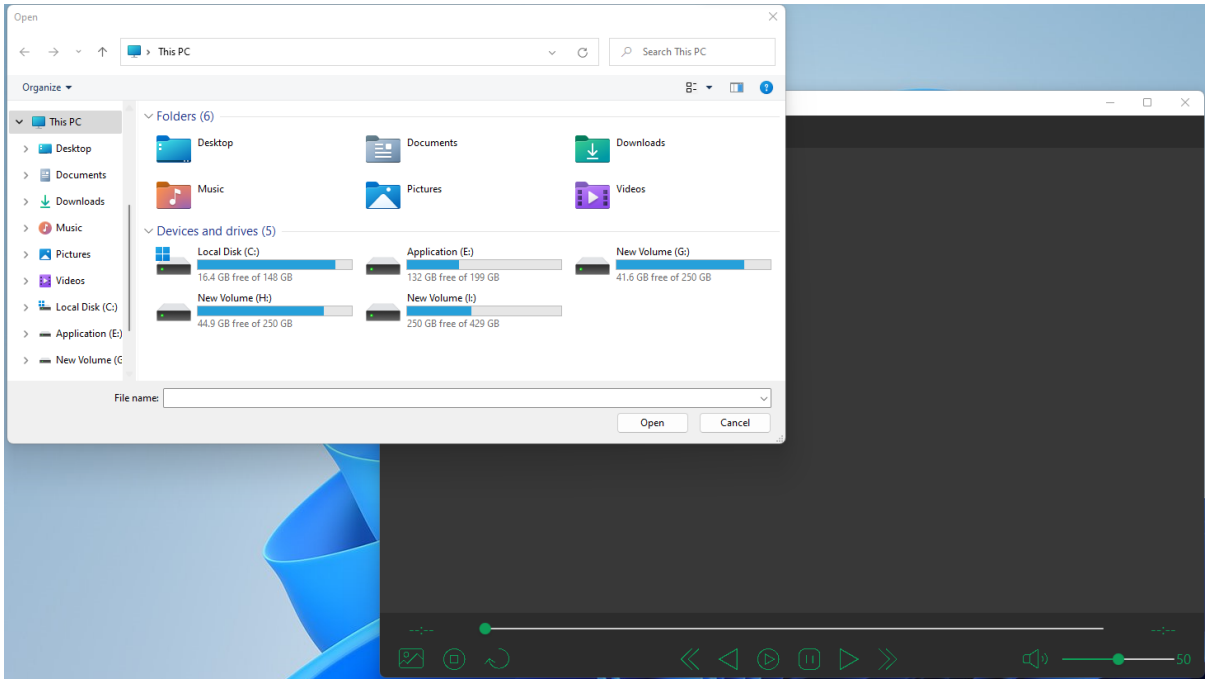
When you start and run analyze.java,you can see the screen like this.

```
+-------------------------------------------------------------------------+
|                          >>> TrustMe Player <<<                         |
|                      *-------------------------------*                  |
|                       ~ All You Gotta Do is TrustMe ~                   |
+ - - ------------------------------------------------------------------ - - +


Options :
  Search --->              Press 1
  Get Analyze List --->    Press 2
  Watch History --->       Press 3
  Login History --->       Press 4
  Developing team --->     Press 5

Enter your choice :
```

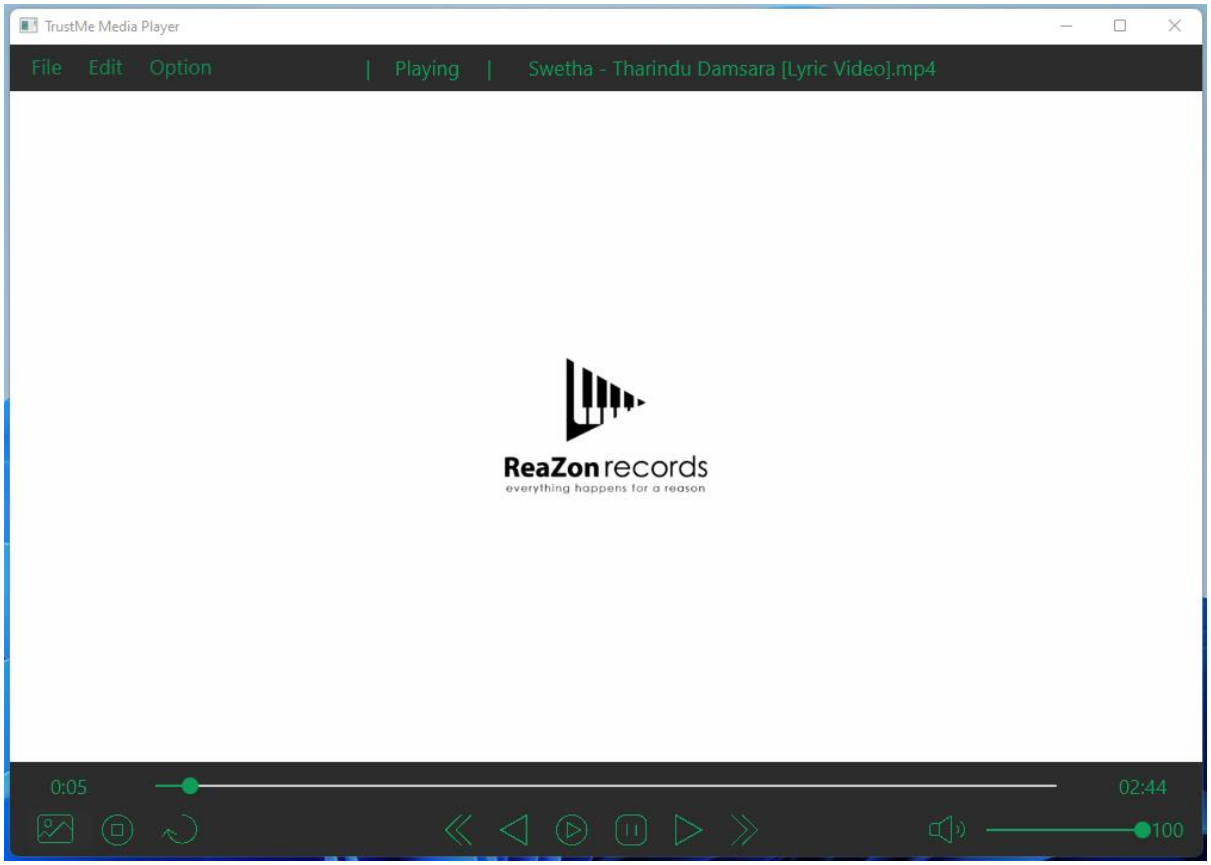
When this use by users should have user names and pass word so we should update logging table.

username	password
Lakruwan	cst20014
Hashini	cst20045
niwandi	cst20067
saliha	cst20011
thakshila	cst20082

The main purpose of our player is to play a video. It can be done by using the open button or by opening a file.



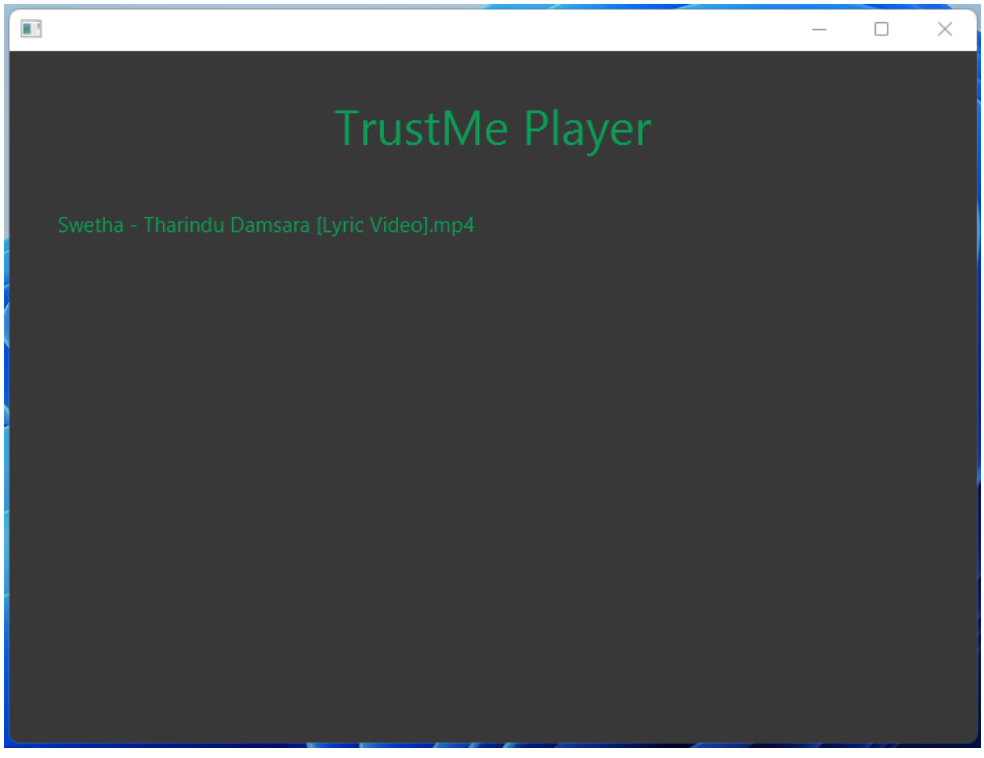
When playing the video, we can see various kinds of differences
1.play video
2.we can see playing where the video is playing
3.update the label were display video duration
4.sound bar can go to highest value so user can agent sound preference
5.when playing video update the play time slider
And more….



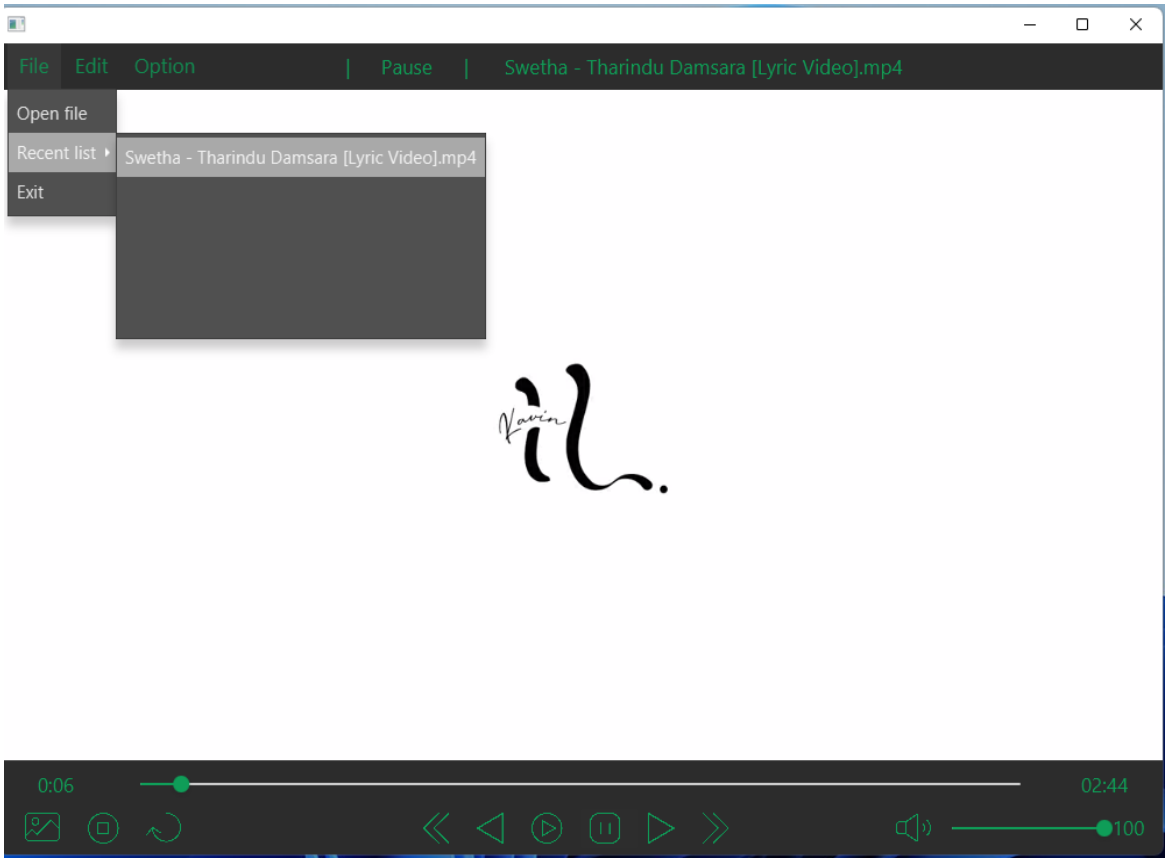
When we start watching a video, its information is added to the database.

ID	VideoName	Format	Path	S_date	S_time	E_date	E_time
0	Swetha - Tharindu Damsara [Lyric Video].mp4	.mp4	file:/C:/Users/lakru/OneDrive/Pictures/Swetha%20-%...	2022-09-03	08:03:27	2022-09-03	08:03:39

After watching a video, the play list is also updated.

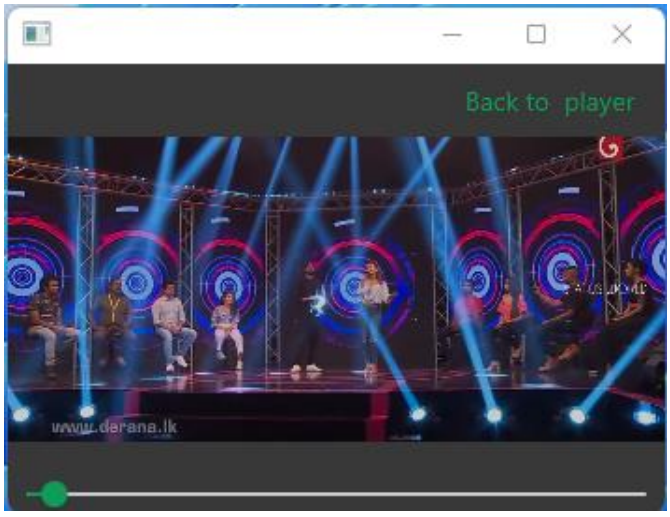


At the same time, the recent file in the file is also updated.

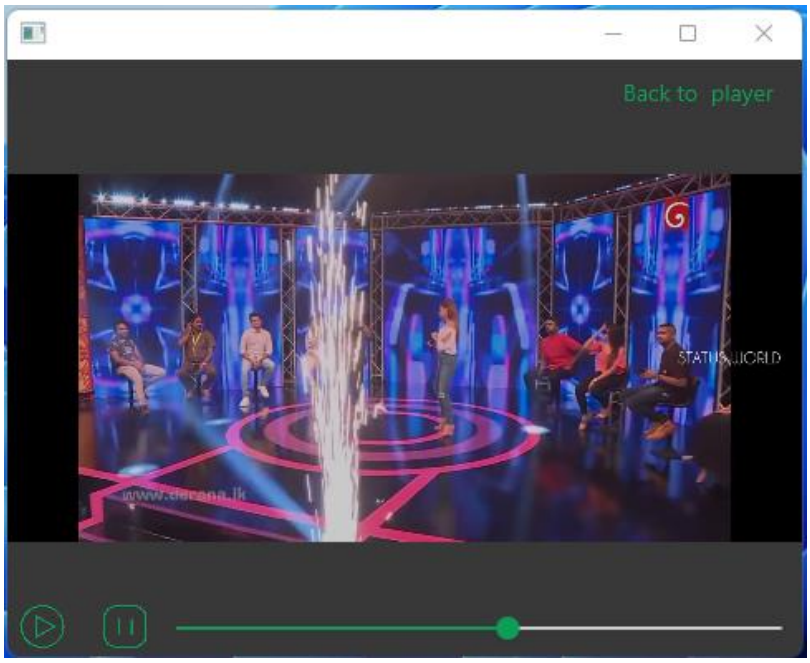


Also, what makes this different from a normal player is that it can be viewed in full screen. Similarly, 2 Mini views have been created based on the user's needs, but the features provided in the mini view are less.

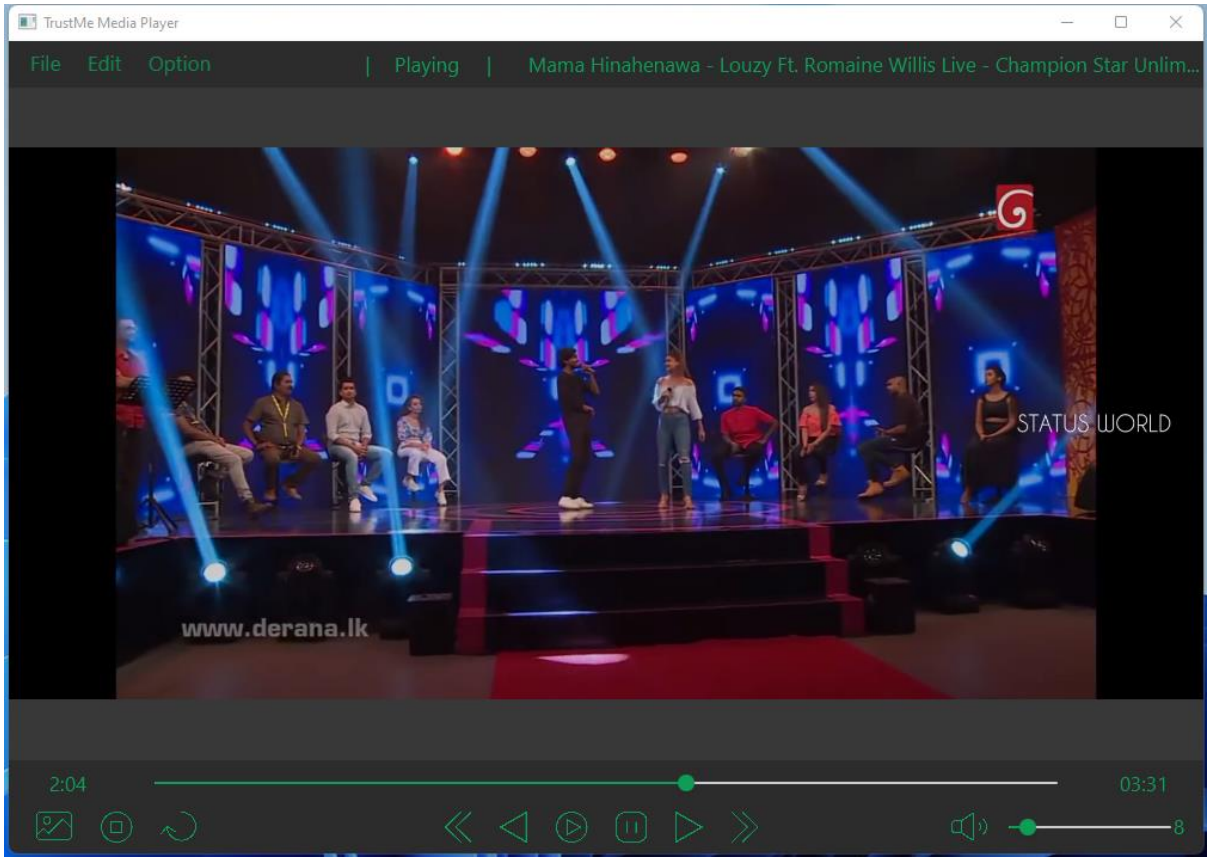
Miniview 1 - 240p



Miniview 1 - 320p

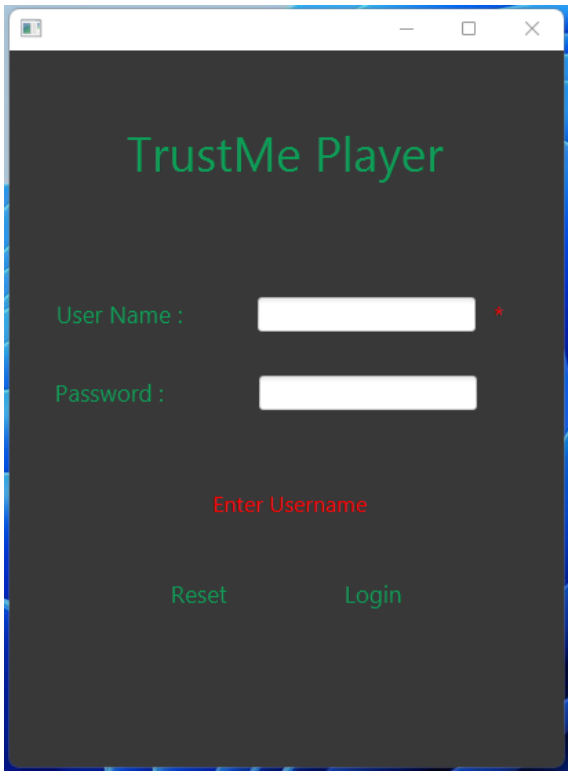


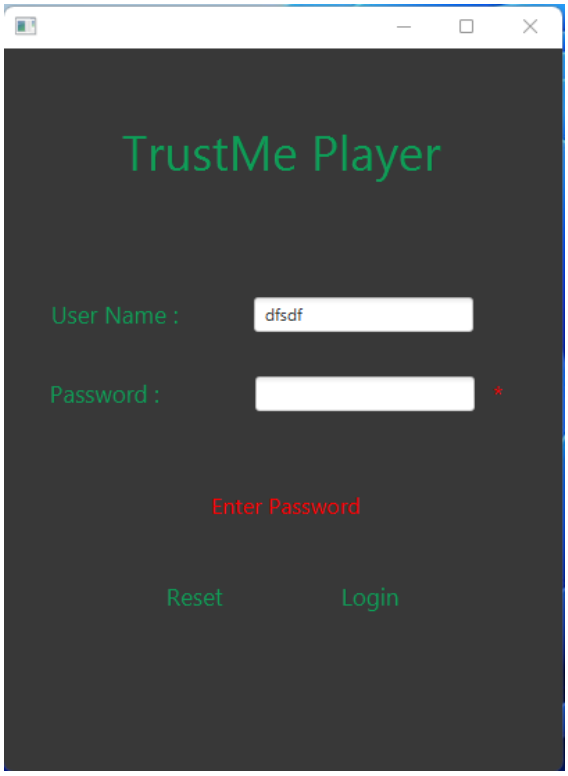
Normal size

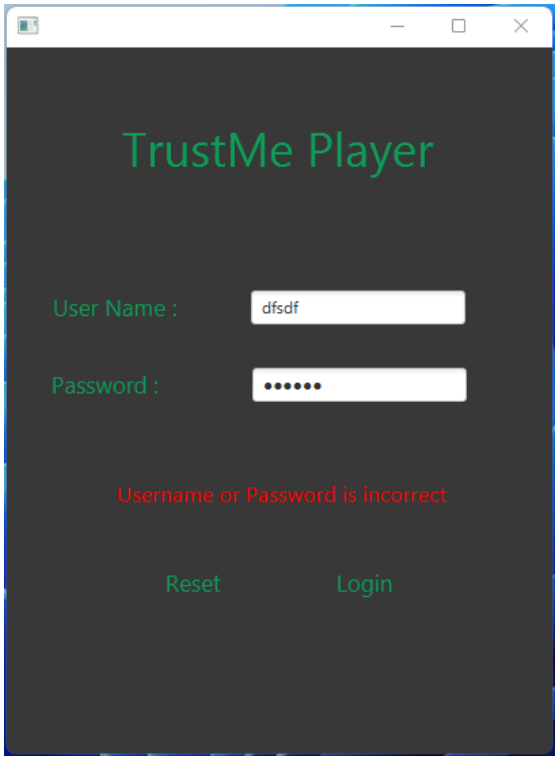


Fullscreen

In addition to this, if you want to go to the options that can be used by the admin, you need to go through the login. During the login, user name and password are checked and their correctness is also checked. If it is correct, it should be logged, and if it is wrong, an error message should be displayed.

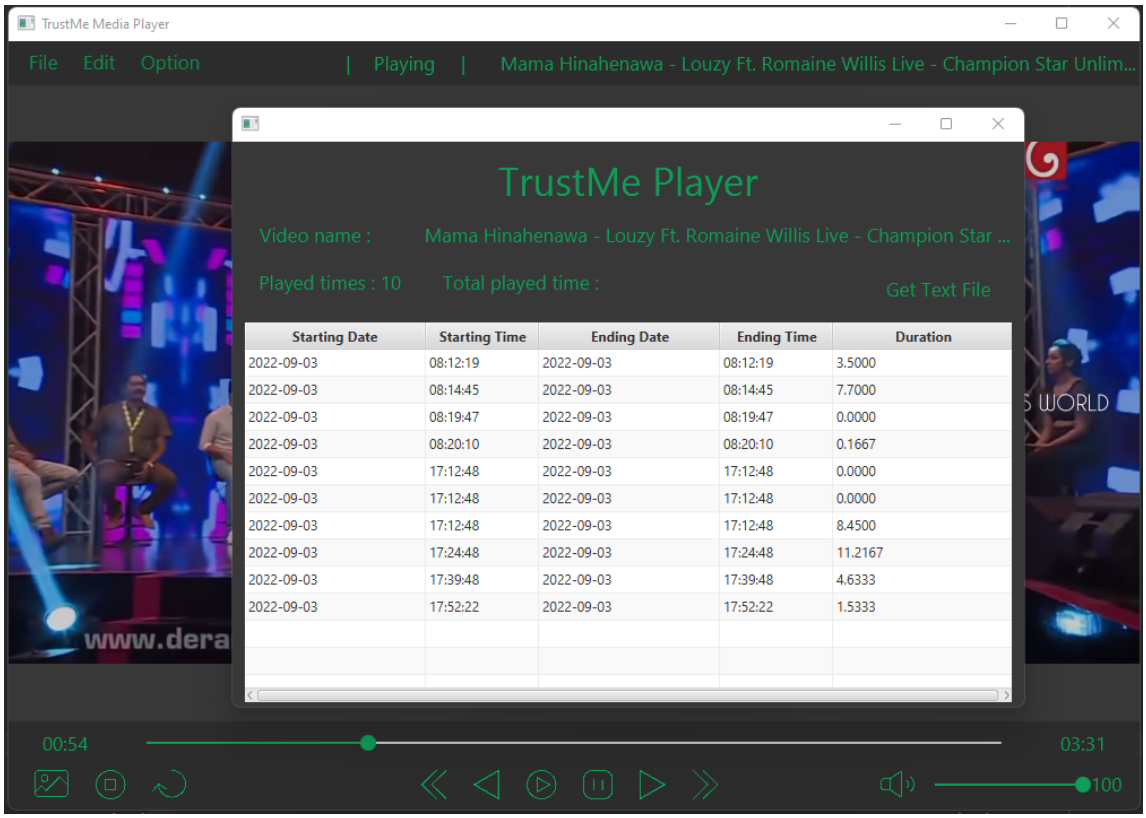






You have to select an option before giving the username and password. If the username and password you provided are correct, you can select the option you have chosen.

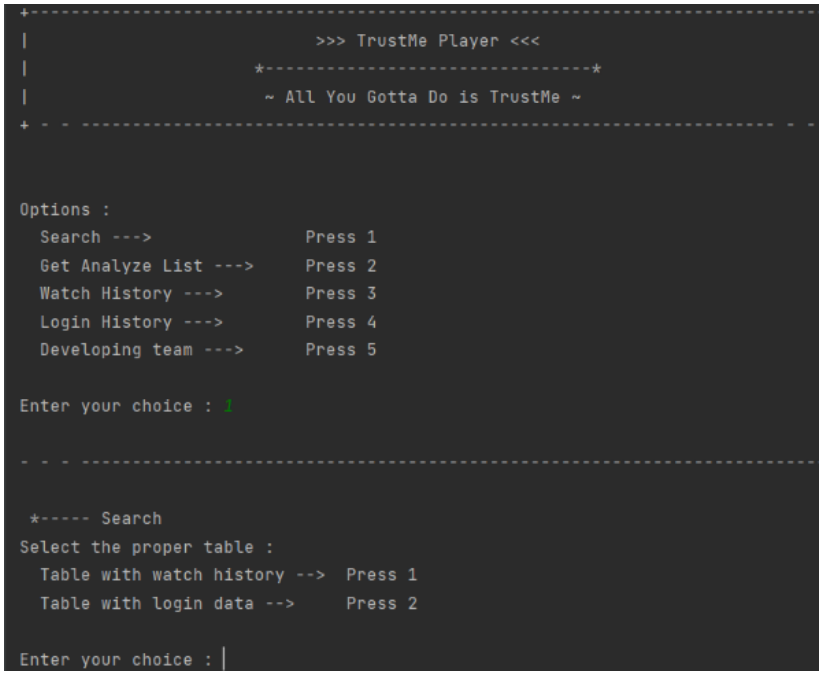
If we click on the history option there, we can see the history of the video played at that time.



When watching that video, you can get details like how long you have watched the video, how long you have watched it, how many times you have watched it and etc. But if you want to get those details, it is not enough to just have username and password and be correct. The video must be playing at that time.
In addition to showing that information in a table, the name of that video and how many times that video has been watched are shown for us to get a better idea.

Search

There are five options in this player. When we choose 1st option we can see search after that we can see two tables like this. And among this table if we select watch history table, we can see like this.



Then if we enter any key word as a video name, we can get related table to the video name as output.

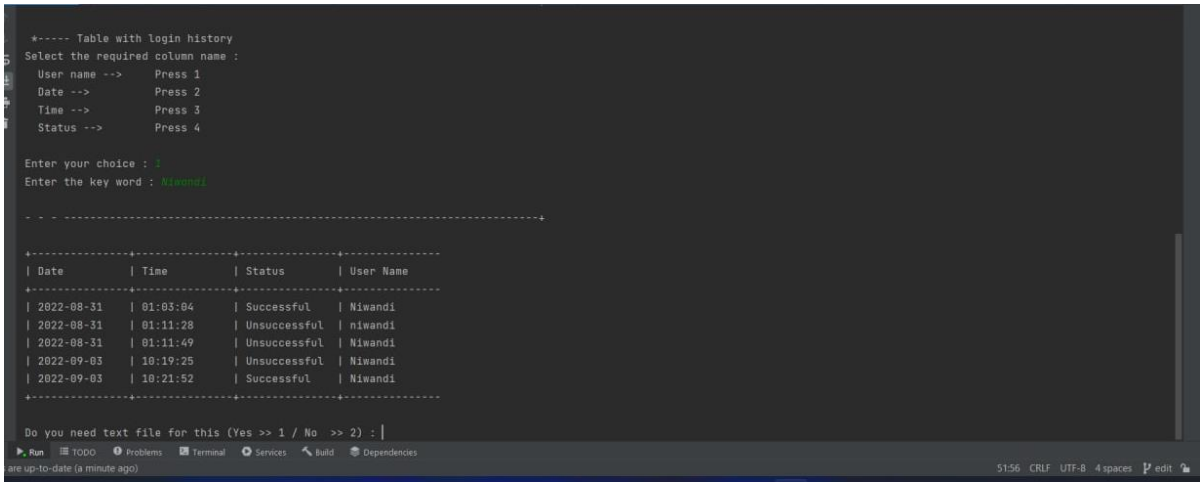
Watch history

```
Enter your choice : 1

- - - ------------------------------------------------------------------------+

 *----- Search
Select the proper table :
  Table with watch history -->  Press 1
  Table with login data -->     Press 2

Enter your choice : 1

- - - ------------------------------------------------------------------------+

 *----- Table with watch History
Select the required column name :
  Video name -->    Press 1
  Starting date --> Press 2
  Starting time --> Press 3
  End date -->      Press 4
  End time -->      Press 5

Enter your choice : 1
Enter the key word : swe
```

If we want to print text file, we should enter 1 as our choice. With or without generating a text file we can go back to main menu or terminate the program as we wish.

If you choose second option to view login history four options will display. If you choose 1st option then it will ask you to enter the user's name. If you enter correct user name you will get the table with relevant user name.

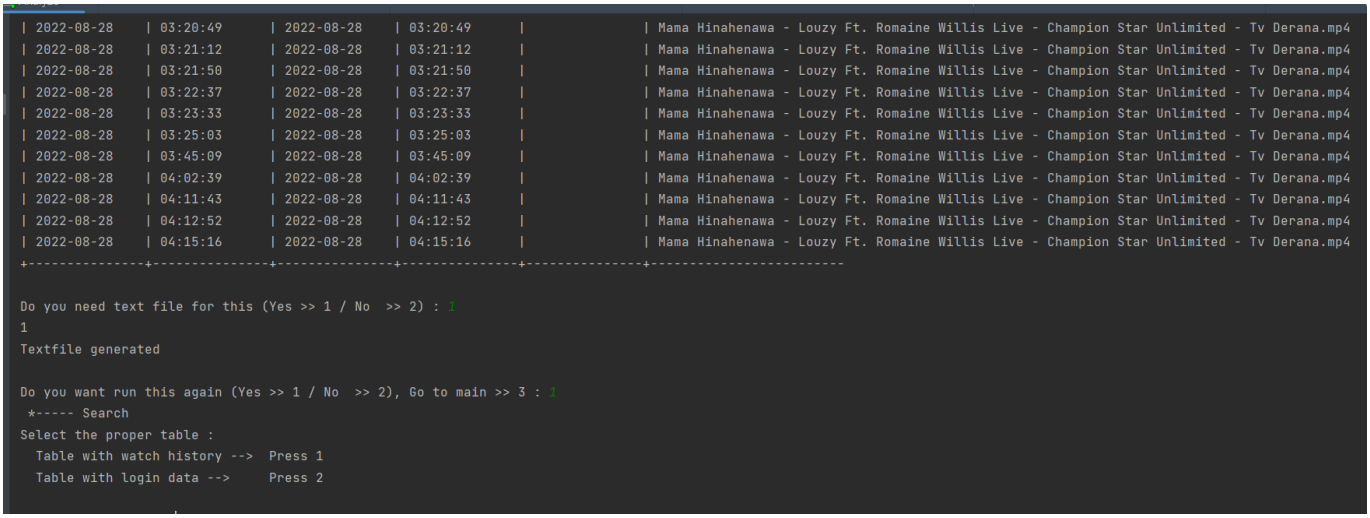
Login history



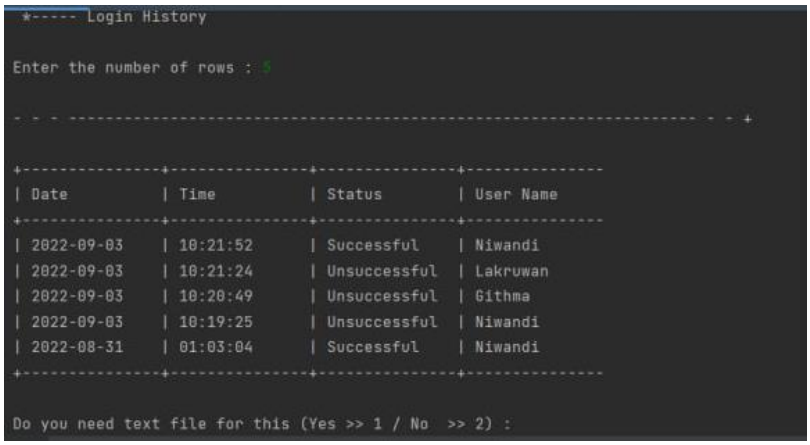
If we click on the history option there, we can see the history of the video played at that time.

When watching that video, you can get details like how long you have watched the video, how long you have watched it, how many times you have watched it and etc. But if you want to get those details, it is not enough to just have username and password and be correct. The video must be playing at that time.

In addition to showing that information in a table, the name of that video and how many times that video has been watched are shown for us to get a better idea.



```
| 2022-08-28    | 03:20:49      | 2022-08-28    | 03:20:49      |               | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28    | 03:21:12      | 2022-08-28    | 03:21:12      |               | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28    | 03:21:50      | 2022-08-28    | 03:21:50      |               | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28    | 03:22:37      | 2022-08-28    | 03:22:37      |               | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28    | 03:23:33      | 2022-08-28    | 03:23:33      |               | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28    | 03:25:03      | 2022-08-28    | 03:25:03      |               | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28    | 03:45:09      | 2022-08-28    | 03:45:09      |               | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28    | 04:02:39      | 2022-08-28    | 04:02:39      |               | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28    | 04:11:43      | 2022-08-28    | 04:11:43      |               | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28    | 04:12:52      | 2022-08-28    | 04:12:52      |               | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28    | 04:15:16      | 2022-08-28    | 04:15:16      |               | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
+---------------+---------------+---------------+---------------+---------------+------------------------

Do you need text file for this (Yes >> 1 / No  >> 2) : 1
1
Textfile generated

Do you want run this again (Yes >> 1 / No  >> 2), Go to main >> 3 : 1
 *----- Search
Select the proper table :
  Table with watch history -->  Press 1
  Table with login data -->     Press 2
```



```
 *----- Login History

Enter the number of rows : 5

- - - ------------------------------------------------------------------------ - - +

+---------------+---------------+---------------+---------------
| Date          | Time          | Status        | User Name
+---------------+---------------+---------------+---------------
| 2022-09-03    | 10:21:52      | Successful    | Niwandi
| 2022-09-03    | 10:21:24      | Unsuccessful  | Lakruwan
| 2022-09-03    | 10:20:49      | Unsuccessful  | Githma
| 2022-09-03    | 10:19:25      | Unsuccessful  | Niwandi
| 2022-08-31    | 01:03:04      | Successful    | Niwandi
+---------------+---------------+---------------+---------------

Do you need text file for this (Yes >> 1 / No  >> 2) :
```

Here you can check the history of logged in to the player, that is, when someone logs in to the player, the time he logged in, the date and whether the login was successful or not, and the username of the person who logged in is mentioned there.

Get analyze

```
Options :
  Search --->               Press 1
  Get Analyze List --->     Press 2
  Watch History --->        Press 3
  Login History --->        Press 4
  Developing team --->      Press 5

Enter your choice : 2

- - - ------------------------------------------------------------------------------+

 *----- Get Analyze List
Which time period do you prefer ?
  Hour by hour -->          Press 1
  Day by day -->            Press 2
  Month by month -->        Press 3
  Year by Year -->          Press 4
  Details of a hour -->     Press 5
  Details of a day -->      Press 6
  Details of a month -->    Press 7
  Details of a year -->     Press 8
```

Here, anyone can select one of the following periods and get the related details by using the Get Analyze list option. There you can get the details hour by hour, day by day, month by month or any selected hour, day, month or year.

```
Enter your choice : 1

- - - ------------------------------------------------------------------------+

+--------------+--------------+--------------+--------------+--------------+------------------------
| Starting Date | Starting Time | End Date     | End Time     | Duration     | Video Name
+--------------+--------------+--------------+--------------+--------------+------------------------
| 2022-08-28   | 04:15:16     | 2022-08-28   | 04:15:16     |              | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28   | 04:12:52     | 2022-08-28   | 04:12:52     |              | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28   | 04:11:43     | 2022-08-28   | 04:11:43     |              | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28   | 04:10:23     | 2022-08-28   | 04:10:23     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 04:09:02     | 2022-08-28   | 04:09:02     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 04:07:25     | 2022-08-28   | 04:07:25     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 04:03:38     | 2022-08-28   | 04:03:38     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 04:02:39     | 2022-08-28   | 04:02:39     |              | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-08-28   | 04:01:51     | 2022-08-28   | 04:01:51     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 04:00:17     | 2022-08-28   | 04:00:17     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 03:59:29     | 2022-08-28   | 03:59:29     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 03:56:57     | 2022-08-28   | 03:56:57     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 03:54:44     | 2022-08-28   | 03:54:44     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 03:53:15     | 2022-08-28   | 03:53:15     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 03:49:58     | 2022-08-28   | 03:49:58     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 03:49:28     | 2022-08-28   | 03:49:28     |              | Swetha - Tharindu Damsara [Lyric Video].mp4
| 2022-08-28   | 03:45:09     | 2022-08-28   | 03:45:09     |              | Mama Hinahenawa - Louzy Ft. Romaine Willis Live - Champion Star Unlimited - Tv Derana.mp4
| 2022-09-03   | 03:39:47     | 2022-09-03   | 03:42:48     |              | 2022-07-31-18-30-24.mp4
| 2022-08-28   | 03:38:48     | 2022-08-28   | 03:38:48     |              | ap.mp4
| 2022-09-03   | 03:38:28     | 2022-09-03   | 03:39:43     |              | 2022-07-31-18-30-24.mp4
| 2022-09-03   | 03:34:32     | 2022-09-03   | 03:38:24     |              | 2022-07-31-18-30-24.mp4
| 2022-09-03   | 03:33:50     | 2022-09-03   | 03:34:25     |              | 2022-07-31-18-30-24.mp4
```

If someone clicks on the option 'get analyze list' and clicks on 'hour by hour', this is the chart that will be displayed.

```
Enter your choice : 6

- - - ------------------------------------------------------------------------+

 *----- Details of a day

Enter the required day: 3
(format :- 1,2,3...30,31)

- - - ------------------------------------------------------------------------+

+--------------+--------------+--------------+--------------+--------------+------------------------
| Starting Date | Starting Time | End Date      | End Time      | Duration      | Video Name
+--------------+--------------+--------------+--------------+--------------+------------------------
| 2022-09-03    | 03:18:25      | 2022-09-03    | 03:31:51      |               | 2022-07-31-18-30-24.mp4
| 2022-09-03    | 03:33:50      | 2022-09-03    | 03:34:25      |               | 2022-07-31-18-30-24.mp4
| 2022-09-03    | 03:34:32      | 2022-09-03    | 03:38:24      |               | 2022-07-31-18-30-24.mp4
| 2022-09-03    | 03:38:28      | 2022-09-03    | 03:39:43      |               | 2022-07-31-18-30-24.mp4
| 2022-09-03    | 03:39:47      | 2022-09-03    | 03:42:48      |               | 2022-07-31-18-30-24.mp4
+--------------+--------------+--------------+--------------+--------------+------------------------
```

In here you choice as a choice number 6 which is details of a day. You can get Stating Date, Starting time, End Date, End Time, Video Name in this table.

Developing team

```
Options :
  Search --->               Press 1
  Get Analyze List --->     Press 2
  Watch History --->        Press 3
  Login History --->        Press 4
  Developing team --->      Press 5

Enter your choice : 5

- - - ------------------------------------------------------------------------+

+-------------------------------------------------------------------------------+
|                          >>> TrustMe Player <<<                               |
|                       ~ Developing Team Members ~                             |
+-------------------------------------------------------------------------------+
|                                                                               |
|   * Fathima Saliha            0742291277                                      |
|   * Lakruwan Jayathissa       0772529441                                      |
|   * Hashini Sulakshana        0701259920                                      |
|   * Niwandi Githma            0716515705                                      |
|   * Thakshila Dulanjani       0762915371                                      |
|                                                                               |
+-------------------------------------------------------------------------------+
```

If you choice option 5. You can see our developing team like this. You can see our member's names and contact number

Conclusion

In conclusion ,we believe that the use of TmPlayer will ease all the predicaments of the users who use media players frequently. It will provide functionalities for playing video files and also will help user to manage their playlist efficiently. This mini-project has come to a conclusion, and we have learned earlier. We have expanded our understanding of Java programming as well as the theory lectures on object-oriented programming by working on this project. We have also made sure that during this group project, the rules established by our lecturer are being followed.

Since, the player has been developed using java ,it is robust ,platform-indipendent ,simple, fast and reliable, which are the most requirements in this technical era.

Individual Contribution

When we first began working as a collective, none of us were very sure about how to proceed; we knew that for this work, everyone would need to take their role within the group, but the problem was finding what those roles should be. Not being able to meet and discuss with each other was a big challenge for us, and we overcame that challenge by organizing all the work on this mini project with the help of Zoom Meeting. Everyone's full input was needed throughout the process, as we realized early on that if this did not happen, then there would be a large proportion of the group not taking part. In larger groups, having someone not taking part would not necessarily be that important, however, with ours, we would be at a serious disadvantage. Luckily every one of group was very committed to producing the best piece that we could, and this involved not letting each other down. This work ethic allowed us to have many in-depth discussions into the possible interpretations and meanings of our work, and so we developed a very complex understanding of what we were actually trying to communicate, and, to us, gave the work much greater meaning. In this mini project, we drew and designed the flowchart and pseudo code. Creating the structure to make the code as well as designing the code as it is in the structure, extending the individual code in a user-friendly manner. In addition to that we also successfully completed this mini project by sharing the information needed to create the report and how to create it and for preparing the presentation we took images of code and output used for our power point slides. As a result, the group worked both individually and collaboratively. Our project began with brainstorming sessions on how to approach the task at hand. On this first level, we were discussing about so many information that we felt it was time that we pulled those ideas together and start working on the project. Each of us did our share in the project and later a meet up was initiated to discuss and compiled our information.

Contribution	Name and Enrollment Number
Idea About Our Project	CST/20/014 - Lakruwan
Flow Chart	CST/20/067 - Niwandi
Source Code	
TmPlayer.java	CST/20/045 - Hashini
player.java	CST/20/014 – Lakruwan, CST/20/045 - Hashini
PlayTime.java	CST/20/014 – Lakruwan
opennew.java	CST/20/082 – Thakshila
Playlist.java	CST/20/011 - Saliha
Miniview.java	CST/20/082 – Thakshila, CST/20/011 - Saliha
Login.java	CST/20/045 – Hashini, CST/20/067 - Niwandi
DataList.java	CST/20/014 – Lakruwan
TableData1.java	CST/20/014 – Lakruwan
connectDB.java	CST/20/067 - Niwandi
Analyze.java	CST/20/014 – Lakruwan, CST/20/067 - Niwandi
Analyze 2.java	CST/20/082 – Thakshila, CST/20/011 - Saliha
Report	All group members

Source code link

https://drive.google.com/drive/folders/1x5ibbfJxoN5EanIKvyqnsVtN_1ChspUV?usp=sharing

References

- Lecture notes
- Youtube
- W3 school
- draw.io (For flowchart)
- Google