“ Media Player in Java ”

Mini Project Report

Submitted in partial fulfillment of the requirements of the subject Java OOPM

by

Anirudh Bhattacharya

Mink Shethia

Prachiti Bapat

Saumya Shah

Supervisor/Guide

Prof. Kavita Bathe

Department of Computer Engineering

K.J. Somaiya Institute of Engineering and Information Technology

Ayurvihar, Sion Mumbai-400022

2020-21

K.J.Somaiya Institute Of Engineering and Information Technology,Sion(E),Mumbai-400 022

*This is to certify that the project entitled **"Media Player in Java"** is a bona fide work of Anirudh Bhattacharya, Mink Shethia, Prachiti Bapat, Saumya Shah submitted as mini project in the subject of Mini project in **"Computer Engineering".***

Prof. Kavita Bathe

(Project Guide)

Department of Computer Class-SE Sem-3 Sub-Java OOPM Academic Year-2020-21

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. we also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. we understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Anirudh Bhattacharya ____________

Mink Shethia ____________

Prachiti Bapat ____________

Saumya Shah ____________

Date:

ACKNOWLEDGEMENT

Before presenting out our seminar work entitled “**MEDIA PLAYER IN JAVA**” we would like to convey our sincere thanks to many people who guided us throughout the course for this seminar work. First, we would like to express our sincere thanks to our beloved Principal **DR. SURESH K. UKARANDE** for providing various facilities to carry out this report.

We would like to express our sincere thanks to **PROF. KAVITA BATHE** for her guidance, encouragement, co-operation and suggestions given to us at progressing stages of report.

Finally, we would like to thank our **H.O.D. Dr. SARITA AMBADEKAR** and all teaching, non-teaching staff of the college and friends for their moral support rendered during the course of the report work and for their direct and indirect involvement in the completion of our report work, which made our endeavour fruitful.

Anirudh Bhattacharya

Mink Shethia

Prachiti Bapat

Saumya Shah

ABSTRACT

Videos are a part of human life now. Be it educational or entertainment. But many PCs don’t have the best inbuilt video players. Our solution? To make a video player, complete with play, stop, pause, fast forward and volume control buttons. This video player has been coded in Java, made using the NetBeans application. This is a Java FXML Application also styled by CSS. Our video player can support MP4 files as they are the most common video file to be viewed on a desktop. The size of the player can also increase to support full screen play.

Introduction:

Video Player in Java helps the user play videos easily. They can pause, play, fast forward or even slow down the video according to their need. This video player also includes a volume slider so as to adjust the audio settings of the video they are watching.

INDEX

6. CONCLUSION

7. REFERENCES

CHAPTER 1
INTRODUCTION

1.1 Introduction:

Media Player in Java

This desktop and laptop application can play videos of all quality, length and size seamlessly. It will come with additional features, allowing users to pause, stop and even alter the viewing speed of the video or audio selected.

1.2 Problem Introduction:

- Many PCs come without a good video or music player pre installed.
- Media is necessary, be it for education or for entertainment.
- Videos must also have the feature to viewed at full screen setting with the best display setting.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 INTRODUCTION:

To be used properly, the device software needs certain hardware components or the other software resources to be present on the device. These pre-requisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

2.2 HARDWARE REQUIREMENTS:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

HARDWARE REQUIREMENTS FOR PRESENT PROJECT:

PROCESSOR : Intel Pentium dual core or above.

RAM : 1 GB

Network Interface.

2.3 SOFTWARE REQUIREMENTS:

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

SOFTWARE REQUIREMENTS FOR OUR PROJECT:

OPERATING SYSTEM : Windows XP and above, Ubuntu v12.04, Android 4 and above.

FRONT END : HTML, CSS, JS

CHAPTER 3
ANALYSIS

3.1 PROPOSED SYSTEM:

Our video player will be able to play MP4 and MP3 files seamlessly, no matter the size of the video. This player will come with options to select files from the PCs internal storage, play the video, pause it and change the time settings to 1.5x speed or 2x speed or to even slow the video down to 0.5x or 0.75x speed according to the viewer's viewing choices. There is also a volume slider included which can adjust the volume from 0% to 100% according to the user's choice. A time seeking slider is also included which will tell the user how much time has passed or how much is left.

3.2 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

3.2.1 Economic Feasibility

This video player will be free for everyone to access and has been built using free software, namely NetBeans IDE and SceneBuilder with Java 1.8 which has JavaFX enabled which allowed us to build our application.

3.2.2 Technical Feasibility

The technical feasibility assessment meets with the expected needs of the proposed system. It has evaluated that hardware and software meets the need of the proposed system. The assessment based on the project of online testing consist of an interactive interface between student and teachers reveals the following outline design of system requirements:

- JDK 1.8 (with JavaFX)
- NetBeans IDE
- Screen Builder Software
- Java FXML
- CSS

3.2.3 Operational Feasibility

This application for Desktop can be accessed from any Windows device from Windows XP to Windows 10 Professional Plus with a minimum of 2 Gigabytes of RAM. Even Ubuntu and Linux users can run the application.

3.3 SOFTWARE SPECIFICATION

JAVA:

What is Java?

Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

Java is −

- Object Oriented − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- Platform Independent − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
- Simple − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- Secure − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- Architecture-neutral − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- Portable − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- Robust − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
- Multithreaded − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
- Interpreted − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.
- High Performance − With the use of Just-In-Time compilers, Java enables high performance.
- Distributed − Java is designed for the distributed environment of the internet.
- Dynamic − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

Why Use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful
- It has a huge community support (tens of millions of developers)
- Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

JavaFX:

Rich Internet Applications are those web applications which provide similar features and experience as that of desktop applications. They offer a better visual experience when compared to the normal web applications to the users. These applications are delivered as browser plug-ins or as a virtual machine and are used to transform traditional static applications into more enhanced, fluid, animated and engaging applications.

Unlike traditional desktop applications, RIA's don't require to have any additional software to run. As an alternative, you should install software such as ActiveX, Java, Flash, depending on the Application.

In an RIA, the graphical presentation is handled on the client side, as it has a plugin that provides support for rich graphics. In a nutshell, data manipulation in an RIA is carried out on the server side, while related object manipulation is carried out on the client side.

JavaFX can be used to develop and RIA

JavaFX

JavaFX is a Java library using which you can develop Rich Internet Applications. By using Java technology, these applications have a browser penetration rate of 76%.

What is JavaFX?

JavaFX is a Java library used to build Rich Internet Applications. The applications written using this library can run consistently across multiple platforms. The applications developed using JavaFX can run on various devices such as Desktop Computers, Mobile Phones, TVs, Tablets, etc.

To develop GUI Applications using Java programming language, the programmers rely on libraries such as Advanced Windowing Toolkit and Swing. After the advent of JavaFX, these Java programmers can now develop GUI applications effectively with rich content.

Need for JavaFX

To develop Client Side Applications with rich features, the programmers used to depend on various libraries to add features such as Media, UI controls, Web, 2D and 3D, etc. JavaFX includes all these features in a single library. In addition to these, the developers can also access the existing features of a Java library such as Swing.

JavaFX provides a rich set of graphics and media API's and it leverages the modern Graphical Processing Unit through hardware accelerated graphics. JavaFX also provides interfaces using which developers can combine graphics animation and UI control.

One can use JavaFX with JVM based technologies such as Java, Groovy and JRuby. If developers opt for JavaFX, there is no need to learn additional technologies, as prior knowledge of any of the above-mentioned technologies will be good enough to develop RIA's using JavaFX.

Features of JavaFX

Following are some of the important features of JavaFX −

- Written in Java − The JavaFX library is written in Java and is available for the languages that can be executed on a JVM, which include − Java, Groovy and JRuby. These JavaFX applications are also platform independent.
- FXML − JavaFX features a language known as FXML, which is a HTML like declarative markup language. The sole purpose of this language is to define a user Interface.
- Scene Builder − JavaFX provides an application named Scene Builder. On integrating this application in IDE's such as Eclipse and NetBeans, the users can access a drag and drop design interface, which is used to develop FXML applications (just like Swing Drag & Drop and DreamWeaver Applications).
- Swing Interoperability − In a JavaFX application, you can embed Swing content using the Swing Node class. Similarly, you can update the existing Swing applications with JavaFX features like embedded web content and rich graphics media.
- Built-in UI controls − JavaFX library caters UI controls using which we can develop a full-featured application.
- CSS like Styling − JavaFX provides a CSS like styling. By using this, you can improve the design of your application with a simple knowledge of CSS.
- Canvas and Printing API − JavaFX provides Canvas, an immediate mode style of rendering API. Within the package javafx.scene.canvas it holds a set of classes for canvas, using which we can draw directly within an area of the JavaFX scene. JavaFX also provides classes for Printing purposes in the package javafx.print.
- Rich set of API's − JavaFX library provides a rich set of API's to develop GUI applications, 2D and 3D graphics, etc. This set of API's also includes capabilities of

Java platform. Therefore, using this API, you can access the features of Java languages such as Generics, Annotations, Multithreading, and Lambda Expressions. The traditional Java Collections library was enhanced and concepts like observable lists and maps were included in it. Using these, the users can observe the changes in the data models.

- Integrated Graphics library − JavaFX provides classes for 2d and 3d graphics.
- Graphics pipeline − JavaFX supports graphics based on the Hardware-accelerated graphics pipeline known as Prism. When used with a supported Graphic Card or GPU it offers smooth graphics. In case the system does not support graphic card then prism defaults to the software rendering stack.

History of JavaFX

JavaFX was originally developed by Chris Oliver, when he was working for a company named See Beyond Technology Corporation, which was later acquired by Sun Microsystems in the year 2005.

The following points give us more information of this project −

- Initially this project was named as F3 (Form Follows Functions) and it was developed with an intention to provide richer interfaces for developing GUI Applications.
- Sun Microsystems acquired the See Beyond company in June 2005, it adapted the F3 project as JavaFX.
- In the year 2007, JavaFX was announced officially at Java One, a world wide web conference which is held yearly.
- In the year 2008, Net Beans integrated with JavaFX was available. In the same year, the Java Standard Development Kit for JavaFX 1.0 was released.
- In the year 2009, Oracle Corporation acquired Sun Microsystems and in the same year the next version of JavaFX (1.2) was released as well.
- In the year 2010, JavaFX 1.3 came out and in the year 2011 JavaFX 2.0 was released.
- The latest version, JavaFX8, was released as an integral part of Java on 18th of March 2014.

CASCADING STYLE SHEETS (CSS):

What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

Why use it?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

An example of CSS:

```
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

What are it’s advantages? Why is it the best tool to use with html?

HTML was NEVER intended to contain tags for formatting a web page!

HTML was created to describe the content of a web page, like:

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

The style definitions are normally saved in external .css files.

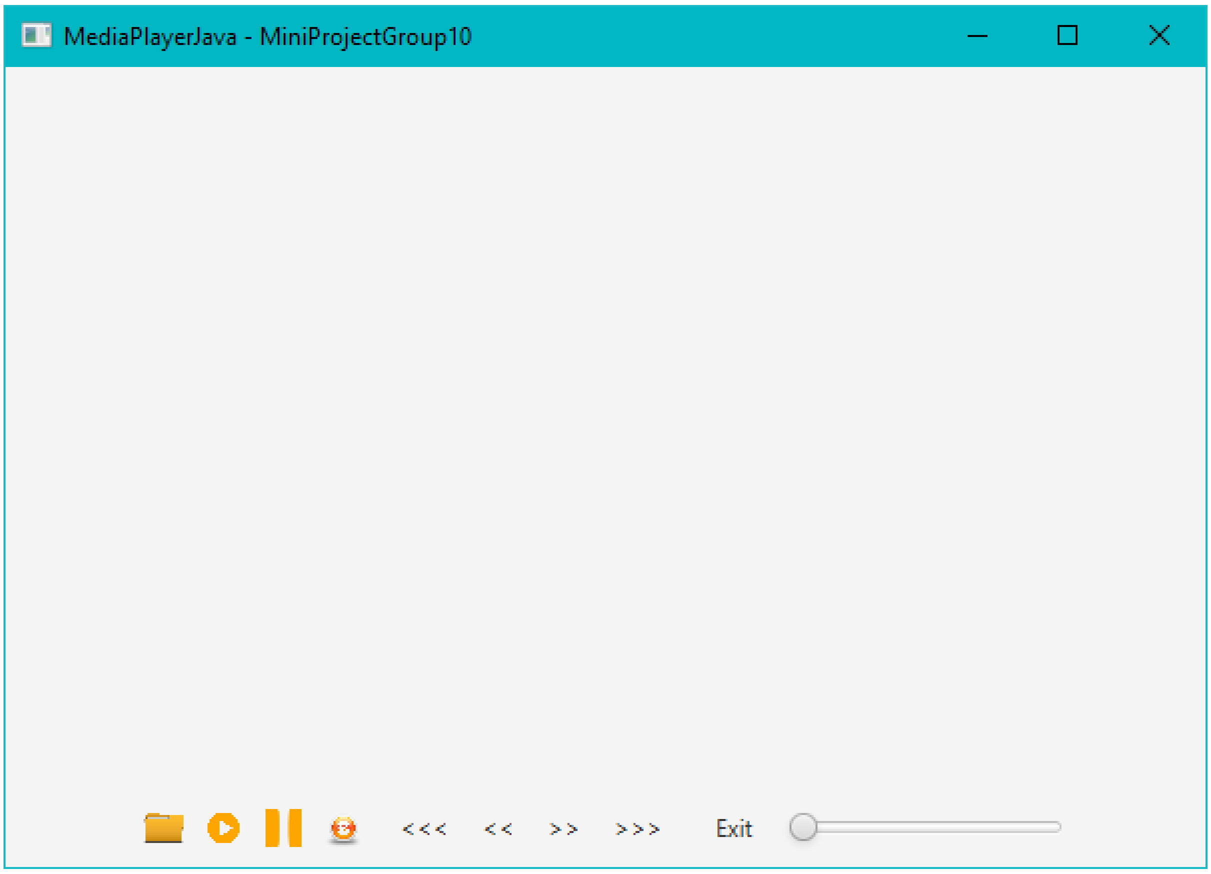
With an external stylesheet file, you can change the look of an entire website by changing just one file!

CHAPTER 4

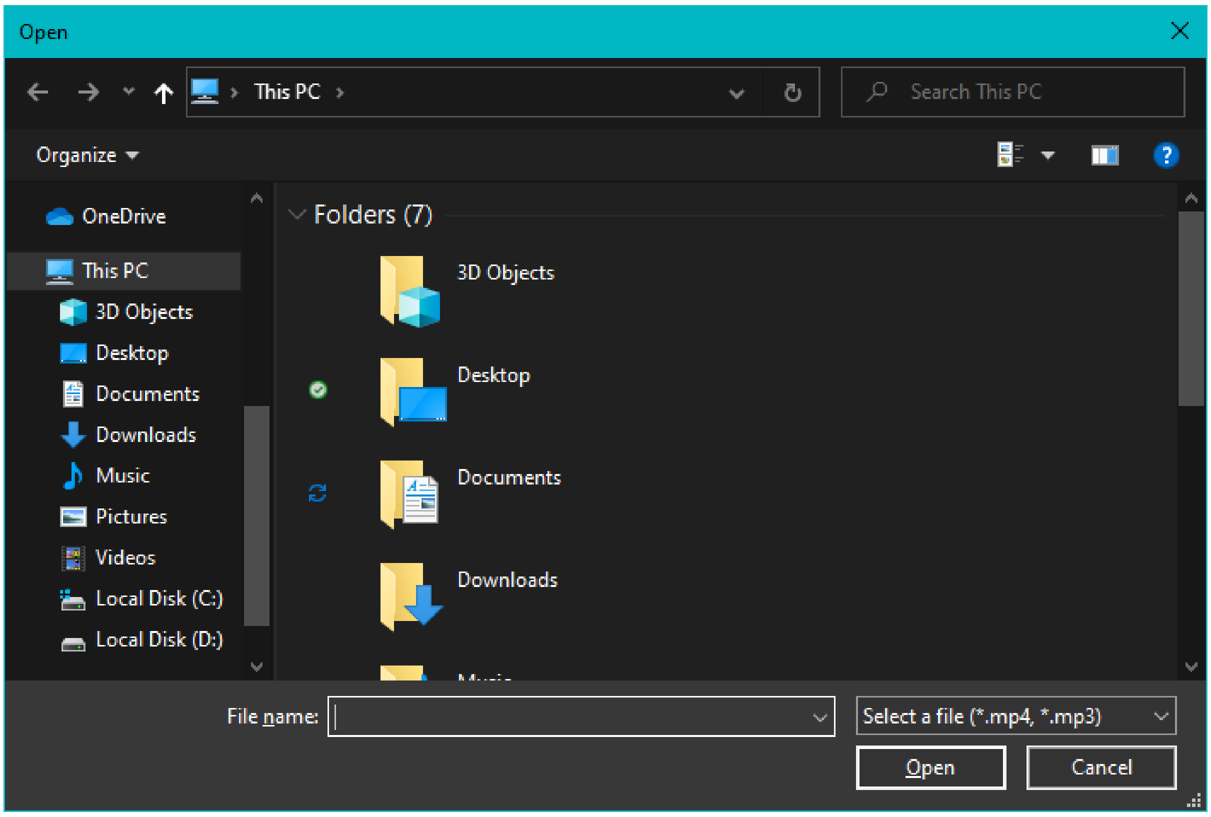
IMPLEMENTATION

Screenshots are provided below:

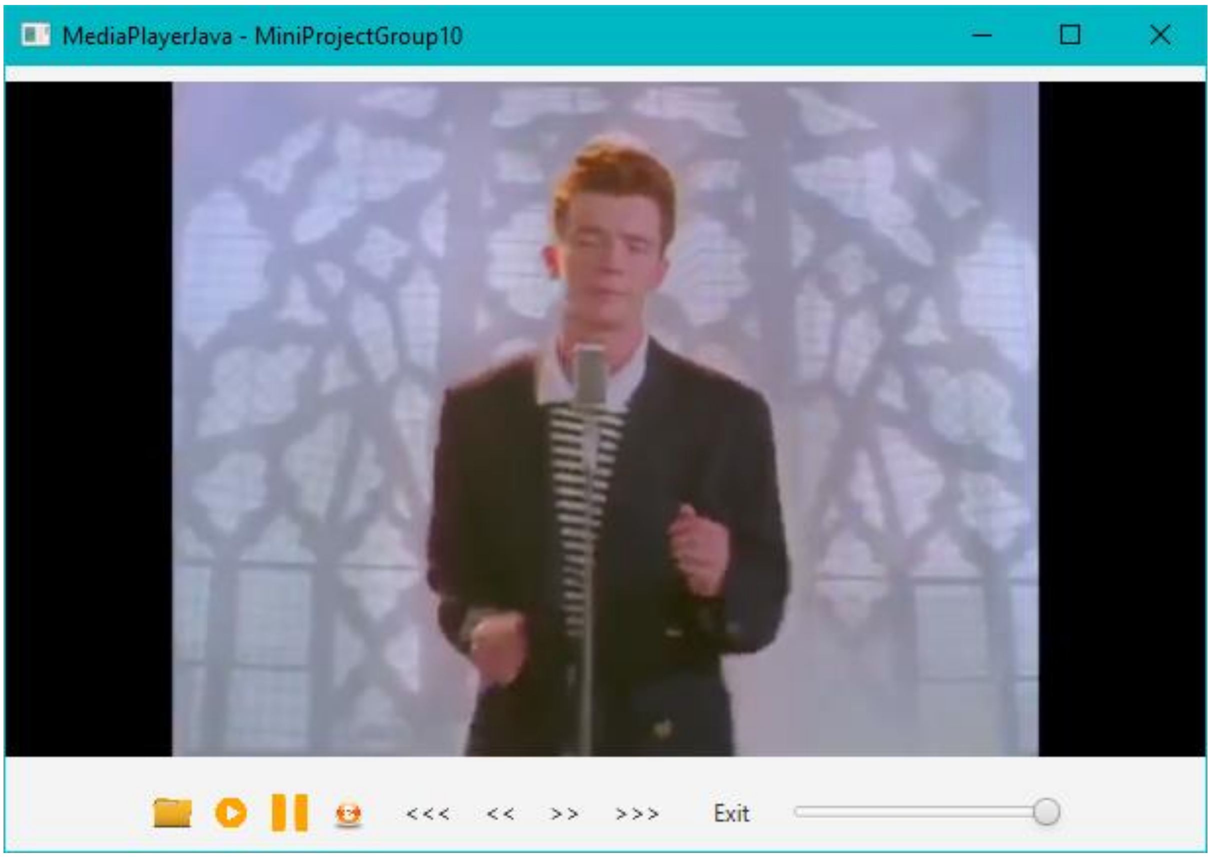
On start up:



Selection of a media file: File types are mentioned.

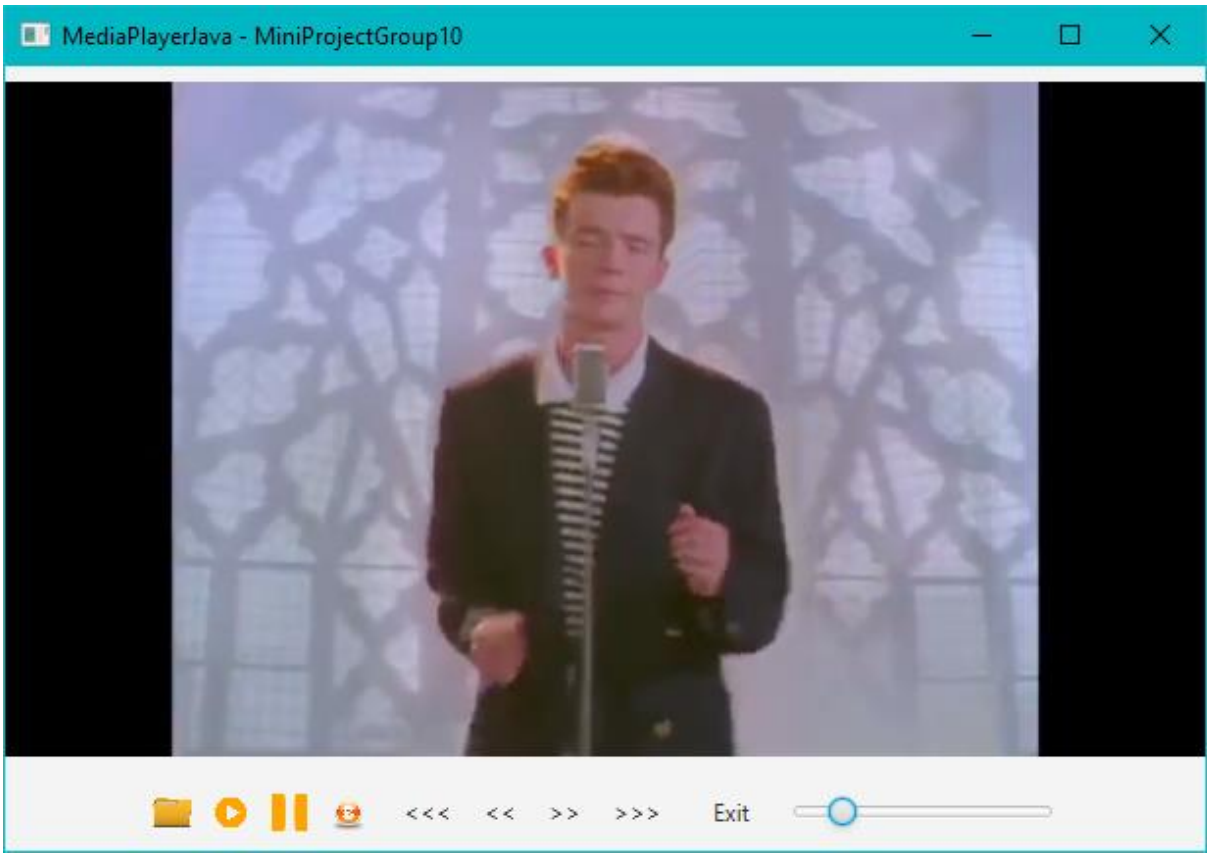


File is selected:

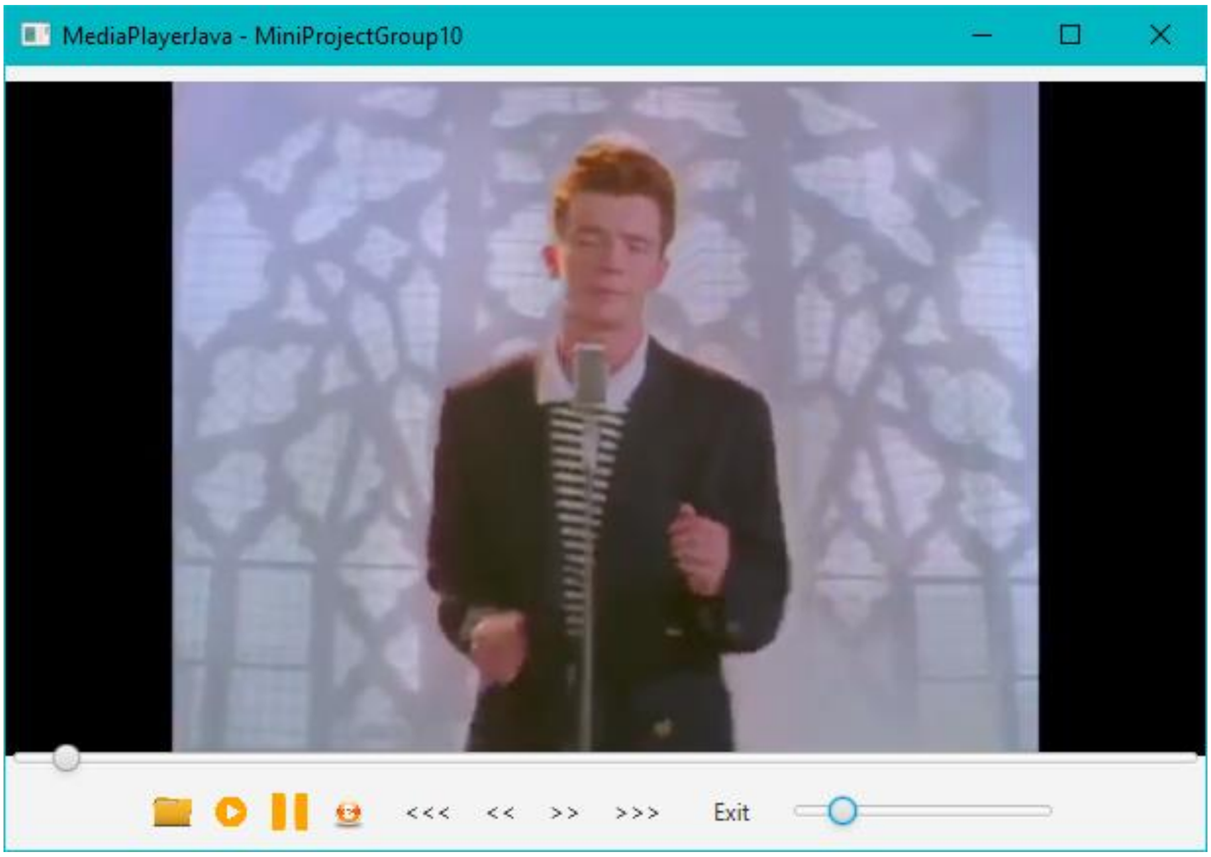


Player supports videos of all aspect ratios.

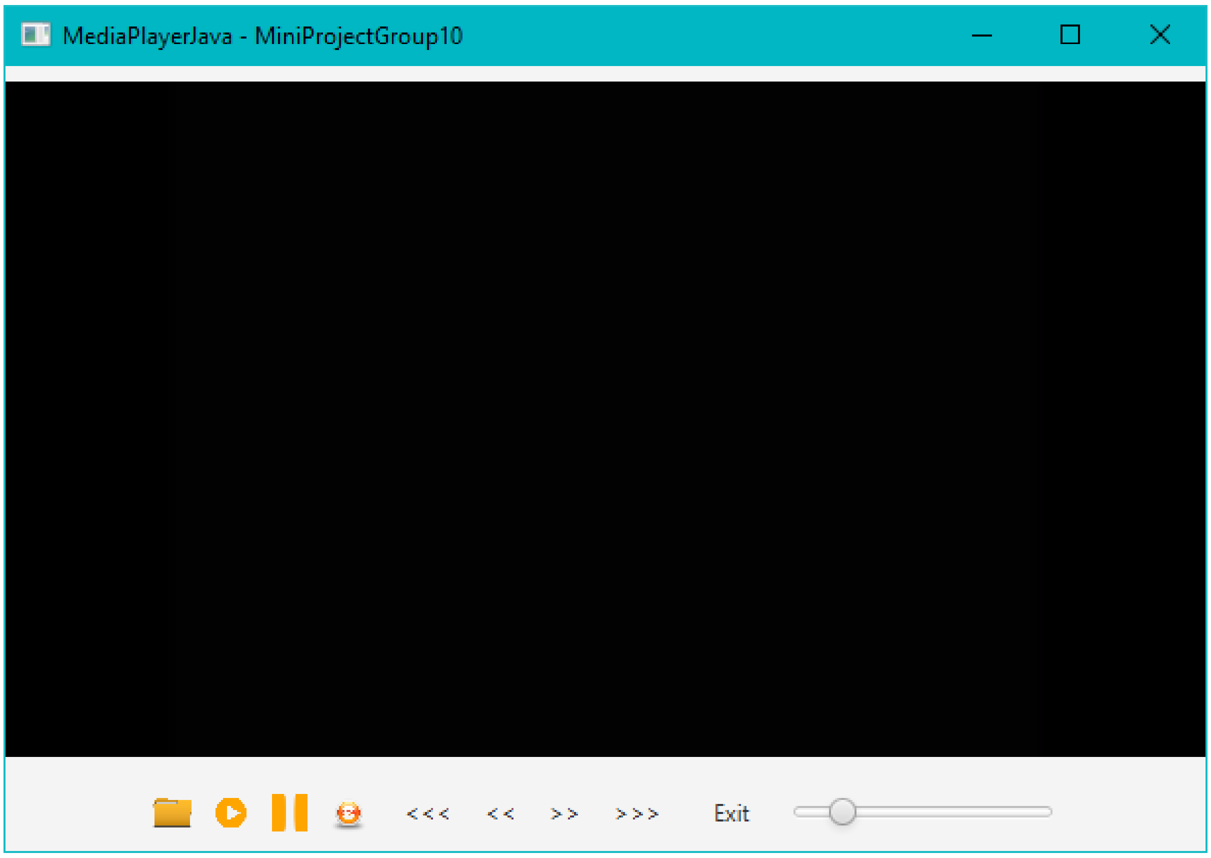
Volume slider can adjust volume from 0-100%



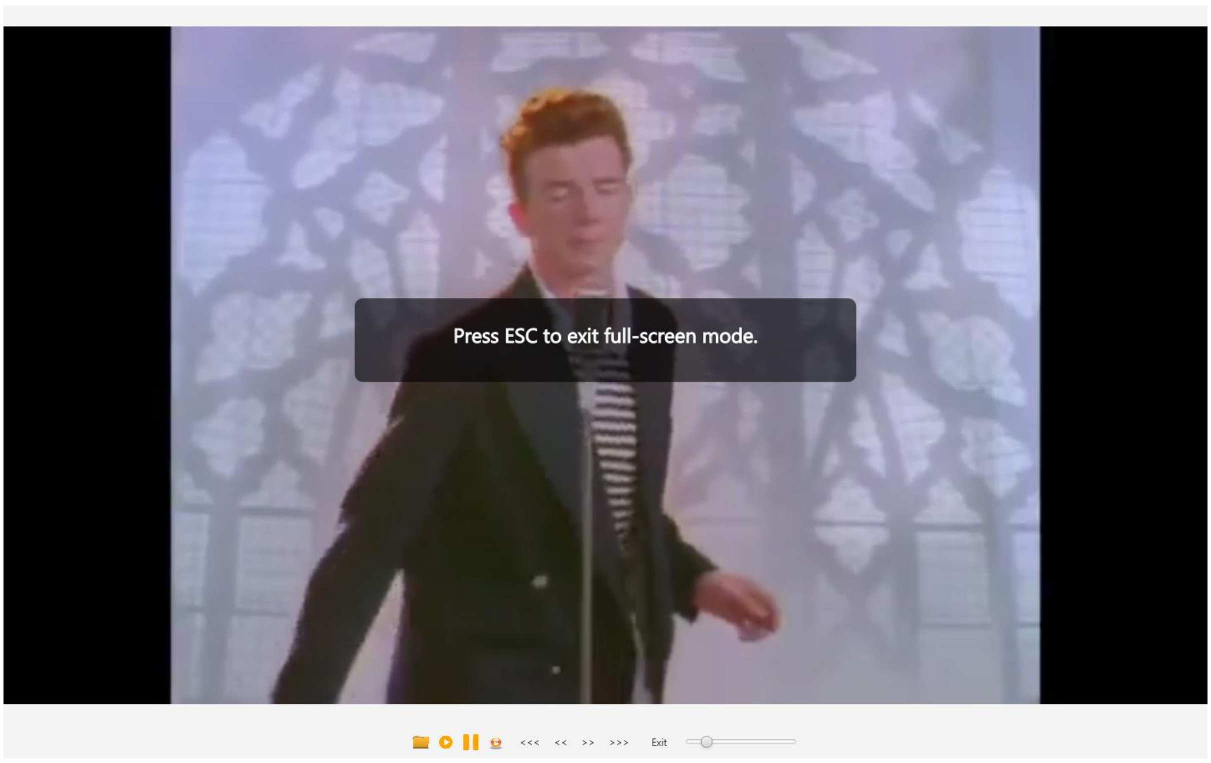
Another slider indicates time passed and left. Can also be used to seek.



Stop button ends the video but keeps the file selected such that it can be played whenever required.



Double clicking on window makes it full screen.



How to run:

Click on the VideoPlayer.exe file. The .jar and .bat files have been converted to Windows executable.

You can now use the app to the fullest.

CHAPTER 5

TESTING

5.1 INTRODUCTION TO SYSTEM TESTING:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.2 TYPES OF TESTING:

1. **Unit testing:**
   Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

2.Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

3.Functional test:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:
Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

4. System Test:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5. **White Box Testing:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6. **Black Box Testing:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7. **Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

8. **Integration Testing:**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

9. **Acceptance Testing:**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

5.3 Testing of Project

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Testing of the Buttons:

Open File:
This button opens up the computer’s storage. By the specified extensions, this button recognizes mp4 files. Once clicked, and the appropriate file selected, the player will play the selected file in whatever quality the file is previously downloaded. This button is determined to be fully functional.
Play File:
This button plays the video file when it has been paused by the user. As we have fast forward buttons and even slow down buttons, once we click the play button, the video continues in the default speed in which the video exists. This button is determined to be fully functional.
Pause File:
This button pauses the file. It can pause at any moment and when the play button is clicked after, it continues at the exact second it had been paused. It can also be paused when fast forward mode has been activated. Button has been determined to be fully functional.
Stop File:
This button stops the video playback. Tested and determined to be fully functional.
Fast forward and Slow down:
There are two buttons each for fast forwarding and slowing down the video. The buttons respectively decrease or increase the speed by 2x, 1.5x, 0.75x, 0.5x. These options can be reset using pause and play buttons. Determined to be fully functional.
Volume:
The volume slider adjusts the volume from 0% to 100% according to the user’s choice. Tested and determined to be fully functional.

CHAPTER 6

CONCLUSION

In our project we have made a media player supporting the necessary file types with different controls for volume, pausing, playing video and stopping the video from playing. This application can be supported on all Windows, Ubuntu and Linux desktops and Laptops.

REFERENCES

We have referred the following websites:

- https://www.w3schools.com/java/default.asp
- https://www.javatpoint.com/javafx-tutorial
- https://docs.oracle.com/javase/8/javase-clienttechnologies.htm
- https://openjfx.io/openjfx-docs/
- https://netbeans.org/kb/docs/java/quickstart.html
- https://docs.oracle.com/javase/8/scene-builder-2/get-started-tutorial/jfxsb-get_started.htm#JSBGS101

SOURCE CODE

https://drive.google.com/drive/folders/1LclI5N_r3uhaMPcwYdU9Rt13x5rUluVQ?usp=sharing

Above link includes project code (NetBeans project), Screen Recorded Video of the application and the actual application in a .rar file.