



Bangabandhu Sheikh Mujibur Rahman Science and Technology University

Gopalganj, Bangladesh

BSMRSTU Multimedia Player

Project Report

Supervisor:

Faruk Hossain
Assistant Professor
Department of Computer Science and Engineering
BSMRSTU, Gopalganj

Submitted By:

Md. Kazi Iqbal Hossen
18ICTCSE065
Session 2018-2019
Dept. of CSE, BSMRSTU

Rafiqur Rahman Ripon
18ICTCSE066
Session 2018-2019
Dept. of CSE, BSMRSTU

18 JULY 2023

Contents

0.1	Introduction	iii
0.2	Background	iii
0.3	Software Pattern	iv
0.4	Implementation	v
0.5	Result & Evolution	v
0.6	Main features	vi
0.7	Future Work	vi
0.8	Conclusion	vi
0.9	Interface Screenshot	vii
0.10	Code Snapshot	viii

List of Figures

1	MVC	iv
2	Interface Initial	vii
3	Interface Menus	vii

0.1 Introduction

There are so many multimedia player developed in the industry but not all the media player are enriched and 4k or UHD streaming supported. Beloved to this issue, I wished to develop a ultra modern software, not so innovative but more scalable multimedia player. Which is enriched with JavaFX and modern media technologies. It supports major kinds of media types as developer intended: MP4, MOV, MP3 and so on. Here is our media player application without any platform dependency. Made with love and JFX...

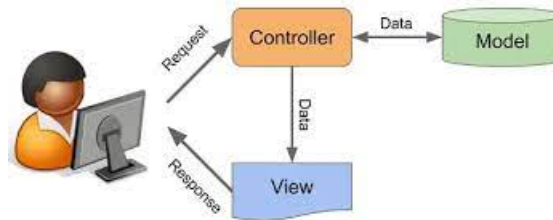
0.2 Background

This project is developed under my university project implementation. The related courses were *Java Technology* and Lab works based on it. Intended to teach us rich client application development with multithreading and concurrency.

0.3 Software Pattern

This software is developed using most common and popular software architecture named MVC: Model-View-Controller. Which is one of the most popular software architecture. Used by lots of thousands of developers and engineers in the leading tech companies. It plays a vital role on the other hand, major provision due to its scalability. In our project, we just used Controller and View template in regard that we don't need any model in a multimedia application. Only the certain standard issue we need is browsing file system.

Figure 1: Model View Controller Architecture



0.4 Implementation

Developed using rich client application development tools. As well as software grows, complexity increases in parallel. This application actually based on JavaFX or a technology which is used to develop more rich desktop apps and mobile embedded systems also. Beloved by most of the software engineers and maintained by a large community of developers alongside. Technologies used here are:

Core Java: Used to implement backend logic.

JavaFX: A modern software development technology.

FXML: An extended XML or markup language to give structure of the software.

CSS: Cascading Style Sheet is another ruling language to design the markup or structure like adding padding around the content, give some margin and color or gradients about the component. It makes it looking nice.

Media API: An api for accessing media and its related issues from operating system no matter whatever OS it is, it just creates its own runtime and fetches the files.

Scene Builder: A sophisticated GUI designer developed by Gluon software technologies Inc. Most popular in the community of Java developers.

Maven: Dependency resolver and library bundler for Java. Maintained by Apache Software Foundation. It has huge stock of repositories that enhances developer experience.

0.5 Result & Evolution

After all, it produces a nice looking media player software. More scaled and enhanced, sophisticated and usable, handy interface intended to give better experiences to its end user. Code and components structured and organized in such way that is more handy and case easy to debug and inject more feature.

0.6 Main features

- Both audio & video playing support.
- Toggling between fullscreen and minimization mode.
- Rollable playlist.
- Drag & Drop accessibility.
- UHD, 4K streaming support.
- Menubar with huge functionality of interact with the system and optimized shorthands.
- High-Level interface includes massive control.
- System default interface. No matter whatever users OS is.
- Informative media info panel.
- Material design included on playlist items.
- Don't worry ... use it as you want!

0.7 Future Work

Developers behind this have a major goal to employ it as a leading software in the industry level as well as keep it as a open source software under BSMRSTU OPS software license. Backends are opened to collaborate. Come up with a handy solution. Add live streaming support with server side programming. Robust formation and focus on your media dear client.

0.8 Conclusion

After all, successfully developed a JavaFX-based multimedia player that can play audio and video files of various formats. We have created a user-friendly and responsive interface that can display and control the media playback. Implemented also the features of opening, playing, pausing, stopping, and seeking media files, as well as adjusting the volume, muting, and toggling from full-screen mode to adjusted window. We have followed the MVC software pattern to separate the logic, interface, and interaction of this product.

0.9 Interface Screenshot

Figure 2: View-1

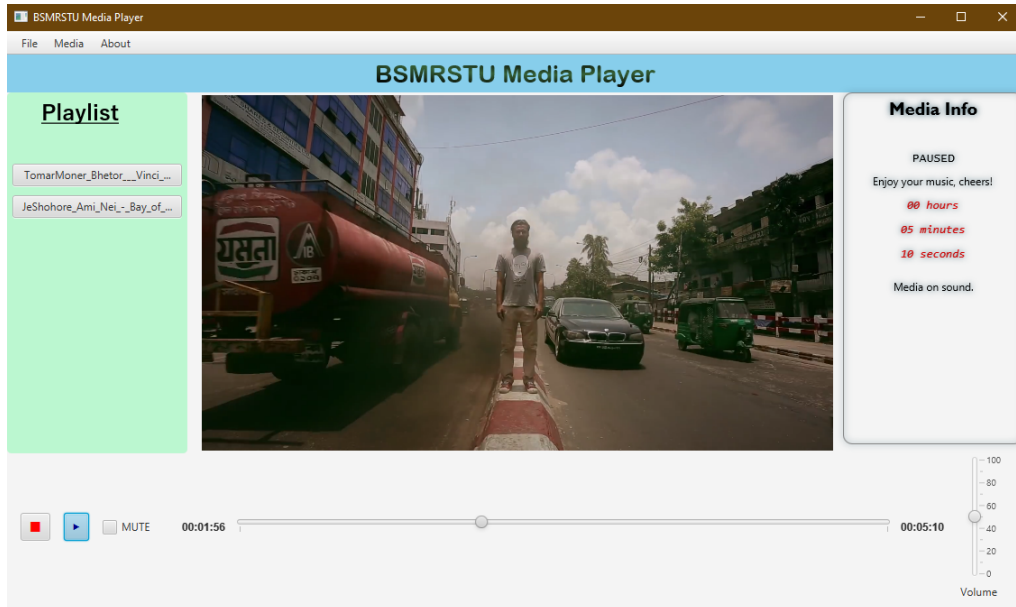
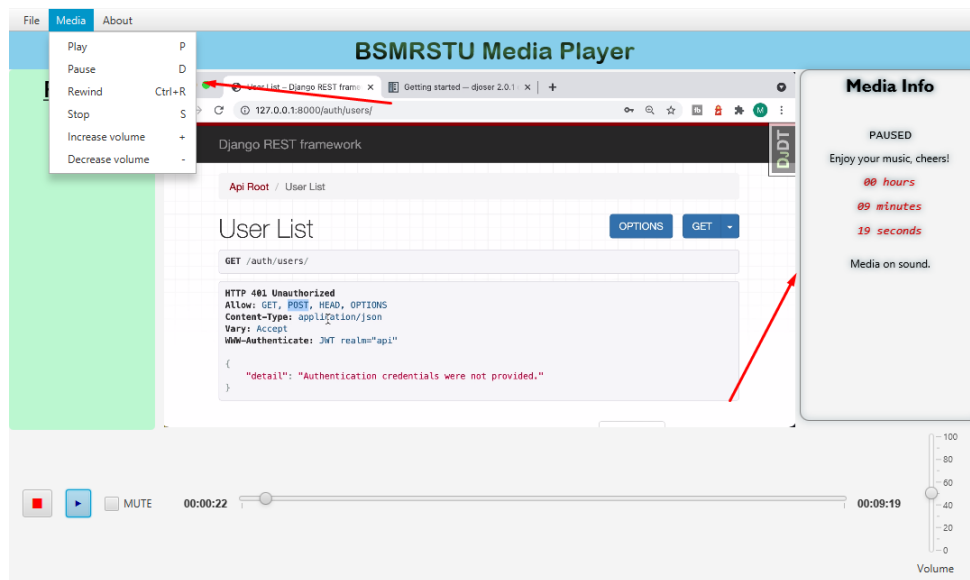


Figure 3: View-2



0.10 Code Snapshot

```
@Override
public void initialize(URL location, ResourceBundle resources) {
    System.out.println(getClass().
        getResource("video.mp4").toString());

    String media_duration = media.getDuration().toString();
    media_view.setMediaPlayer(mediaPlayer);
    total_duration.setText(mediaPlayer.
        getTotalDuration().toString());
    total_duration.setText(media_duration);

    init_media_player();
    evaluate_duration_slider();
    evaluate_volume_slider();
    evaluate_playlist();
}

@FXML
protected void stop_media(ActionEvent e) {
    if (mediaPlayer != null) {
        if (!mediaPlayer.getStatus().
            equals(MediaPlayer.Status.STOPPED)) {
            mediaPlayer.stop();
            current_status.setText("STOPPED");
            mediaPlayer.seek(new Duration(0));
            media_view.setOpacity(1);
            duration_slider.adjustValue(0);
        }
    }
    // Platform.exit();
}

@FXML
protected void play_pause_media(ActionEvent e) {
    Status status = mediaPlayer.getStatus();
    if (status == Status.UNKNOWN || status == Status.HALTED) return;
    if (status == Status.PAUSED || status == Status.READY) {
        mediaPlayer.play();
    } else if (status == Status.PLAYING) {
        mediaPlayer.pause();
    }
    alert_message.setText("Enjoy your music, cheers!");
}
```

Listing 1: Snippet from Controller class

!END...