COIS 2240H

Final Exam - Part 2

Question 1

Winter 2024

Dikshith Reddy Macherla

Student Id : 0789055

Bachelor Of Computer Science, Trent University

COIS 2240H: Software Design and Modelling

Professor : Alaadin Addas

9th April, 2024

Final Exam - Part 2
Question 1

Copy and paste your code :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;

public class MusicPlaylistManager extends JFrame {
    // Fields for the song input fields and list model
    private JTextField titleField, artistField, albumField;
    private DefaultListModel<Song> songListModel;
    private JList<Song> songList;
    private JLabel songDetails;

    public MusicPlaylistManager() {
        // Basic frame setup
        setTitle("Music Surfers");
        setSize(600, 400);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLayout(new BorderLayout(10, 10));

        // Initialize the GUI components
        initComponents();
    }

    private void initComponents() {
        // Define color palette for the light theme
    Color backgroundColor = new Color(240, 200, 200); // Soft light gray for
backgrounds
    Color textColor = new Color(50, 50, 50); // Dark gray for text, providing
contrast
    Color buttonColor = new Color(245, 245, 245); // Very light gray for buttons
    Color inputBackgroundColor = new Color(220, 210, 210); // Slightly darker gray
for input fields


        // Input Panel with GridLayout for song input fields and buttons
        JPanel inputPanel = new JPanel(new GridLayout(4, 2, 5, 5));
        inputPanel.setBackground(backgroundColor);
        titleField = new JTextField();
        artistField = new JTextField();
```

```
            albumField = new JTextField();

            // Customize input fields
            titleField.setBackground(inputBackgroundColor);
            titleField.setForeground(textColor);
            artistField.setBackground(inputBackgroundColor);
            artistField.setForeground(textColor);
            albumField.setBackground(inputBackgroundColor);
            albumField.setForeground(textColor);

            JButton addButton = new JButton("Add");
            JButton deleteButton = new JButton("Delete");
            inputPanel.add(new JLabel("Title:"));
            inputPanel.add(titleField);
            inputPanel.add(new JLabel("Artist:"));
            inputPanel.add(artistField);
            inputPanel.add(new JLabel("Album:"));
            inputPanel.add(albumField);
            inputPanel.add(addButton);
            inputPanel.add(deleteButton);

            // Playlist Panel to display the list of songs
            songListModel = new DefaultListModel<>();
            songList = new JList<>(songListModel);
            JScrollPane playlistPanel = new JScrollPane(songList);

            // View Panel to display details of the selected song
            JPanel viewPanel = new JPanel();
            songDetails = new JLabel("Song details appear here.");
            viewPanel.add(songDetails);

            // Label for displaying the developer's name and student number
            JLabel nameLabel = new JLabel("Dikshith Reddy Macherla - 0789055",
JLabel.CENTER);

            // Adding all panels to the main frame
            add(inputPanel, BorderLayout.NORTH);
            add(playlistPanel, BorderLayout.CENTER);
            add(viewPanel, BorderLayout.SOUTH);
            add(nameLabel, BorderLayout.PAGE_END);

            // Adding functionality to the "Add" button
            addButton.addActionListener(e -> addSong());

            // Adding functionality to the "Delete" button
            deleteButton.addActionListener(e -> deleteSong());

            // Listener for selection changes in the song list, to display song details
            songList.addListSelectionListener(e -> showSongDetails());
        }
```

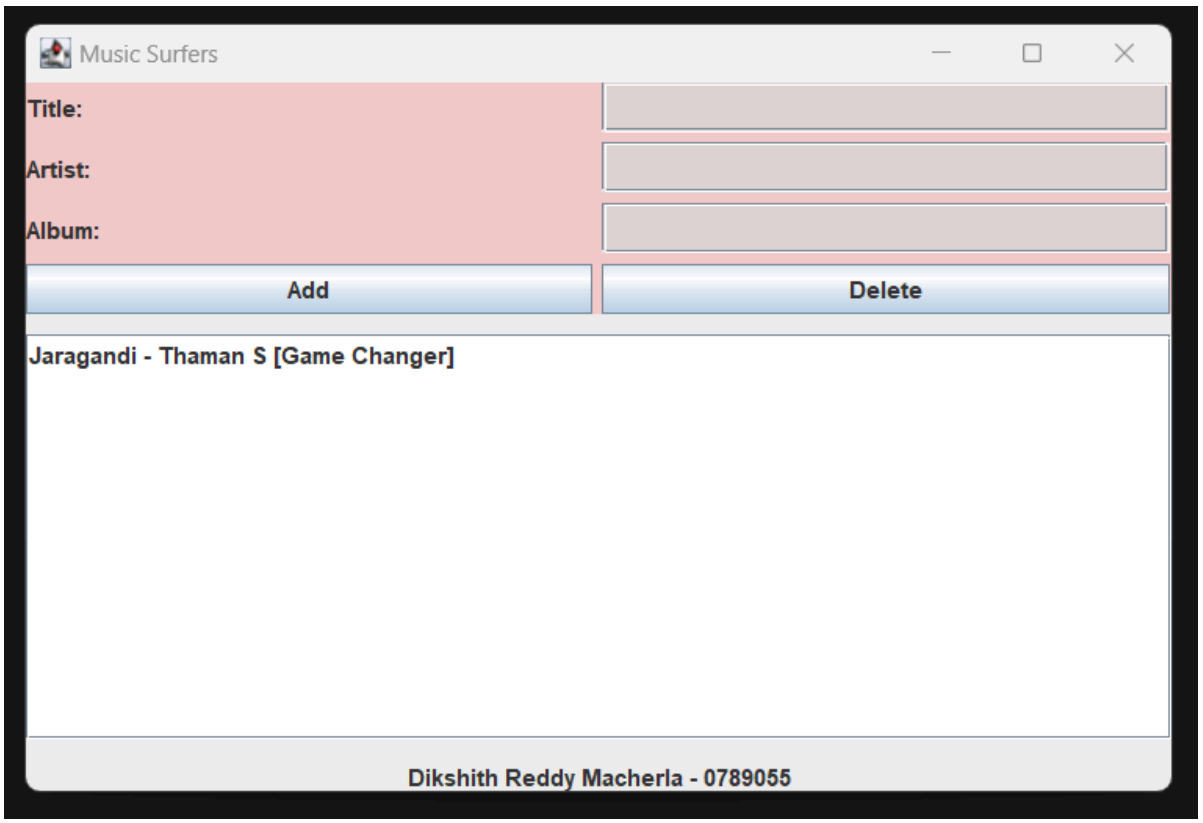
```
        private void addSong() {
            // Read input from text fields
            String title = titleField.getText().trim();
            String artist = artistField.getText().trim();
            String album = albumField.getText().trim();

            // Validate input (ensure none of the fields are empty)
            if (title.isEmpty() || artist.isEmpty() || album.isEmpty()) {
                JOptionPane.showMessageDialog(this, "All fields must be filled out",
"Error", JOptionPane.ERROR_MESSAGE);
                return;
            }

            // Add new song to the list model and clear input fields
            songListModel.addElement(new Song(title, artist, album));
            titleField.setText("");
            artistField.setText("");
            albumField.setText("");
        }

        private void deleteSong() {
            // Delete the selected song from the list
            int selectedIndex = songList.getSelectedIndex();
            if (selectedIndex != -1) {
                songListModel.remove(selectedIndex);
            }
        }

        private void showSongDetails() {
            // Display details of the selected song in the songDetails label
            Song selectedSong = songList.getSelectedValue();
            if (selectedSong != null) {
                songDetails.setText("Title: " + selectedSong.getTitle() + ", Artist: " +
selectedSong.getArtist() + ", Album: " + selectedSong.getAlbum());
            }
        }

        public static void main(String[] args) {
            // Ensure the application runs on the Event Dispatch Thread
            SwingUtilities.invokeLater(() -> new
MusicPlaylistManager().setVisible(true));
        }

        // Inner class for Song
        class Song {
            private String title;
            private String artist;
            private String album;
```

```
        public Song(String title, String artist, String album) {
            this.title = title;
            this.artist = artist;
            this.album = album;
        }

        // Getters for song properties
        public String getTitle() { return title; }
        public String getArtist() { return artist; }
        public String getAlbum() { return album; }

        // ToString method used for displaying songs in the JList
        @Override
        public String toString() {
            return title + " - " + artist + " [" + album + "]";
        }
    }
}
```

Test 1: Adding a Song

Description: Verify that a song can be added to the playlist.
Expected Outcome: The song appears in the playlist panel, and input fields are cleared.

Actual Output:



Test 2: Attempting to Add a Song with Empty Fields

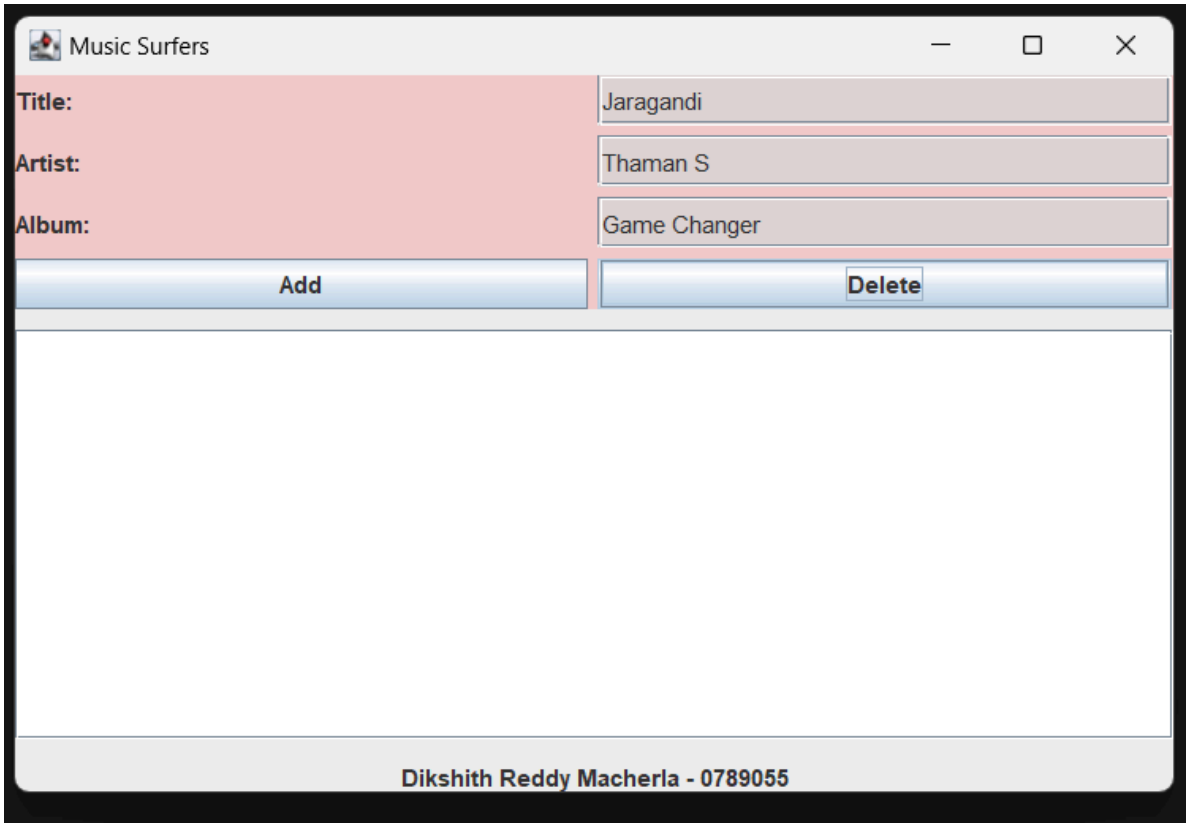
Description: Verify the application prevents adding a song when one or more fields are empty.
Expected Outcome: A dialog box appears informing the user that all fields must be filled out.

Actual Output:

Test 3: Deleting a Song

Description: Verify that a selected song can be deleted from the playlist.
Expected Outcome: The selected song is removed from the playlist.

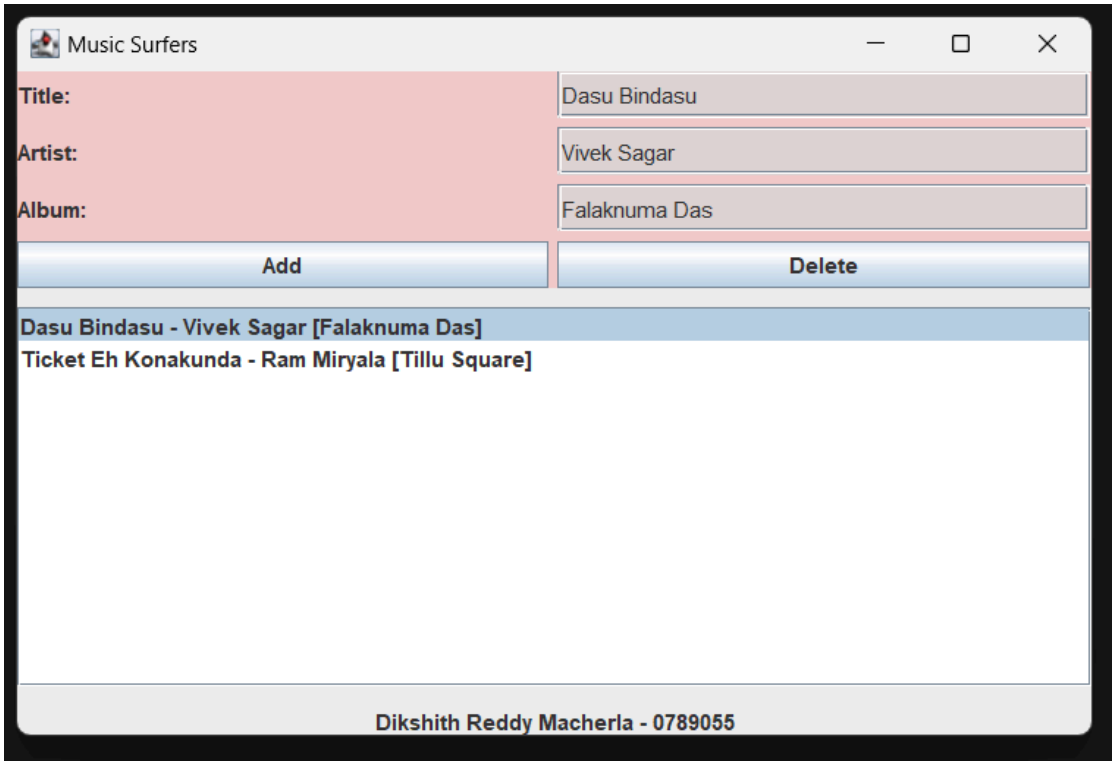
Actual Output:



Test 4: Viewing Song Details

Description: Verify that details of the selected song are displayed in the view panel.
Expected Outcome: The view panel updates to show the selected song's details.

Actual Output:



Test 5: Deleting a Song Without Selection

Description: Verify the application's behavior when attempting to delete without selecting a song.
Expected Outcome: No action is taken since there is no selection.

Actual Output: