NETB 378, Java Programming

Alexander Kolchev, F55283

2012/2013 Spring Semester

I. Assignment description

Variant 4, Music Manager. Using Java / Swing write a program that displays and edits the tags of music media files. The program opens a folder that contains music .mp3 and .flac files, displays file list to the user, and lets him display, and edit the tags contents. The program must have the following features:

- GUI to edit file tags, such as author, title, album, year, genre, etc.
- GUI for to select the folder to open.

II. Functional description

The application provides the following functionalities:

- **File menu**
  - **Import Music Files** – When selected, a filechooser window appears, allowing import of multiple files with extensions restricted to .mp3 and .flac.
  - **Save** – Saves the file tags.
  - **Export Project** – Creates a project file of the files currently being modified and saves the project file, without modifying the actual music files. This allows the user to save his progress.
  - **Import Project** – When selected, a filechooser menu appears, prompting the user to select a Music Manager (.mmproj) file for import. This feature allows the user to continue his progress from a previously saved state.
  - **Exit** – Exits the application.
- **View menu**
  - **Full Screen –** Switches the application to and out of full screen mode.
  - **Show Toolbar –** Shows/Hides the buttons toolbar.
- **Refresh music files button –** Refreshes all the files loaded in the table, loosing all unsaved changes. Used if the user wants to start over. When clicked, a confirmation popup appears.
- **Save music files button –** Saves the file tags**.**

- **Search bar –** Allows the user to filter the files displayed in the table, by searching by all visible attributes of the music files.
- **Table –** Displays the loaded music file data. Deletion of records from the table is possible, by right clicking on a given music file and selecting *Delete*.
- **Mnemonics and Accelerators –** Short keys, for easier and faster navigation through the menus and functionalities of the user interface.
    - **Mnemonics** – used over visible UI elements**:**
        - **Alt + F** – Opens the *File* menu
            - **Alt + S** – Selects the *Save* menu item
            - **Alt + I** – Selects the *Import music files* menu item
            - **Alt + E** – Selects the *Export project* menu item
            - **Alt + P** – Selects the *Import project* menu item
            - **Alt + X** – Selects the *Exit* menu item
        - **Alt + V –** Opens the *View* menu
    - **Accelerators:**
        - **Ctrl + X** – Exits the application

III. Technical description

Application environment and dependencies:

- Platform – JDK 11.0.7
- Framework – Swing
- Development environment – NetBeans 12.0
- External libraries – JAudioTagger
- Source Control – Git
- Repository - https://github.com/AKolchev/MusicManager

The application is based on the Model/View/Controller architecture and is built on the SOLID principles - Single responsibility/Separation of concerns, dependency inversion, interface segregation. The application logic is segregated into different layers, each of them with it's own responsibility. An external java library is used for music file tags manipulation – JAudioTagger. The UI has been implemented without the use of a graphic designer and follows the concepts of Swing.

a. Views layer

The views layer represents the presentation logic of the application. The user interface consists of one window, containing two main components, implemented as partial views, and one main view, wiring together all the components and events:

- ToolbarPartialView – Custom class extending JToolBar. Contains the following components implemented:
    - Load button – Imports music files data into the application
    - Save button – Saves tags to the music files on the file system
    - Reload button – Reloads all files from the file system
    - Search bar – Filters the records in the table by any visible column data
- TablePanelPartialView – Custom class extending JTable. All the presentation logic for the table visualizing music file tags is implemented here. The table's characteristics are described by the views.tableModels – rows count, column count, column names, column value types, together with logic for getting/setting values of a given table cell.
- MainFrame – Custom class, extending JFrame
    - The MainFrame view is the common view. It is responsible for setting the general layout of the application and serves as a controller for the different components. It initializes all the UI components, creates menus, file choosers, controls UI events, defines accelerators and mnemonics.

This implementation decouples the different components and they have no dependencies between each other. Every event is controlled by the MainFrameView, by injecting event listener interfaces from the MainFrameView into the partial views. When an event is being fired, the MainFrame is responsible for the interception of the event and invoking the corresponding logic in the controller.

b. Controllers layer

Segregates the presentation logic and the business logic, carries out the dataflow between the views and the file operations layers through the models layer. All the exceptions are handled here. This layer consists of only one controller – MainFrameController, used by the MainFrameView. The MainFrameController exposes all the methods for data manipulation, consumed by the MainFrameView.

c. Models layer

Describes the data, being transferred between the different application layers and the attributes of the music files.

MusicFileTagsModel – Describes the attributes of the music files (title, artist, year, etc..), and provides getters and setters.

FlacCommentKeysEnum – The music tags in .flac files are stored in the form of comments with certain keys for each value. This enumeration describes the format of the music tags for .flac files. The enumeration is used in order to avoid string hard-coding in various places in the application.

d. File Operations layer

This layer contains all the logic responsible for data manipulation – loading music files into the system, saving modified tags, removing files, project file import and export, filtering of the data collection. This layer is exposed only to the Controllers layer. The FileOperations object operates with a List collection of type MusicFileTagsModel. This collection stores all the necessary information regarding each music file. Instead of keeping the files open in the memory, the collection stores the file location on the file system, allowing the FileOperations object to open and close each file for each operation. The model contains a property called "modified", which indicates if a given file has been modified or not. This way the FileOperations object iterates only on modified files, when the user triggers the "Save files" functionality.

e. Event Listeners

Contains a set of interfaces, describing the application events

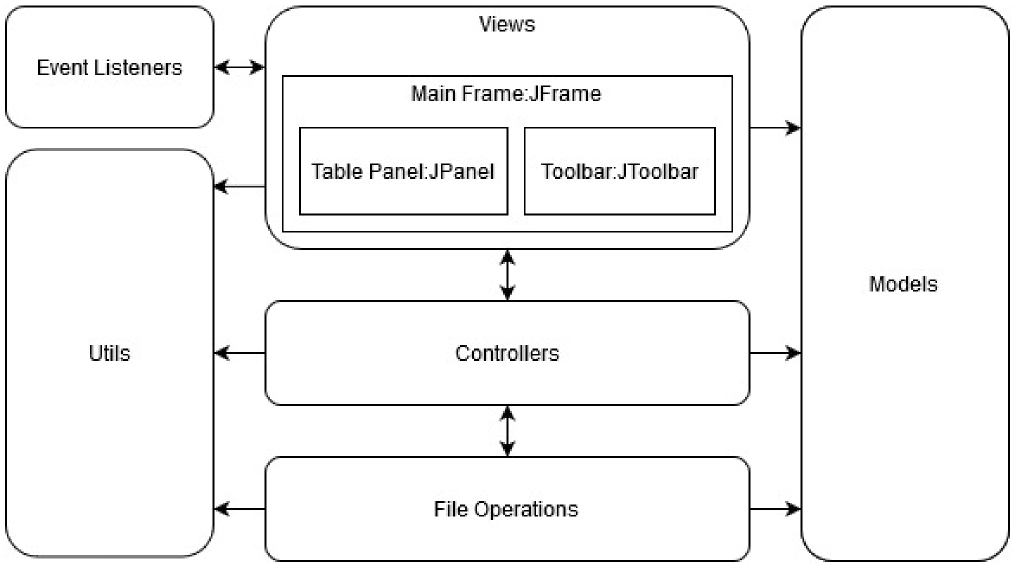
- TableFilteredEventListener – This event is being fired when the user inputs search criteria in the SearchBar. The MainFrame catches the event and notifies the controller. The controller invokes the FilesOperations object, where the data collection, loaded into the table, is being filtered.
- TableRowDeletedEventListener – An event triggered when a user removes a record from the table. The MainFrame view catches the event and notifies the controller. The controller invokes the table row deletion functionality of the FileOperations object.
- ToolbarButtonsEventListener – An event listener, responsible for the Save and Reload button actions in the toolbar. When one of the buttons is being clicked, the toolbar fires an event, handled by the MainFrame view. The MainFrameView invokes the corresponding logic in the Controller, who transfers the execution down to the FileOperations object.

f. Utils

- This folder contains helpers, common for the different layers of the project. The functionalities, implemented here, include:
- Helper - Extraction of file extensions, conversions from Array to List, normalizing values for the Genre file tag.
- ImportSongsFileFilter - Music file import restrictions by file type.
- ProjectFileFilter – Project file import restrictions by file type.

- SortTableRows – Mechanism for sorting the table data by visibility and artist name. Used to ensure that 1) Records have a meaningful order and 2) After deletion of records from the table, the order of the records in the table will stay the same with no empty rows.

Solution Diagram:


Event Listeners
Views
Main Frame:JFrame
Table Panel:JPanel
Toolbar:JToolbar
Models
Utils
Controllers
File Operations