Band Manager user's manual

Band service for managing setlist, songs and events with an easy GUI for the final user.

The program allows information to be maintained on multiple different band profiles, each with its own set of repertoire songs and scheduled events. The user can log on to the system in guest mode to access and view the data of the band of interest, or, using appropriate login credentials, can log on to the system as a band profile and edit the songs and events in the database.

Last version released on 1/03/2024.

Database informations

Before launching the service, you need to make sure that you have on the host machine any http based server database manager (ex. Apachi) and a MySQL application listening on port 3306 in localhost, from which the actual connection to the database will be made.

Once the connection to the MySQL application is made, the program will look up the name of the database ("bandmanagerdb"). If it does not exist, it will be automatically created with a set of values in order to create a demo of the program.

In order for the program to operate and interact with the database, it is essential to verify that the JDBC communication driver "mysql-connector-j-8.2.0.jar," or earlier compatible versions, is installed and configured correctly in the execution IDE on the program host computer.

Example:
In my case I am using XAMPP control panel for managing the apache http server and the activation of the the MySQL application, provided by phpMyAdmin. In order to start the service, it is necessary to initially start the apache server and then the MySQL app. The app in my case is automatically configured to go listening on port 3306 and I can easily access its contents at the link : http://localhost/phpmyadmin/

Usage

1- To run the application, you must initially compile and run the **ServerLogic.java** file. The server once started will make the connection with the database. The console will report the result of the connection, if it fails, return to the *Database Information* section.

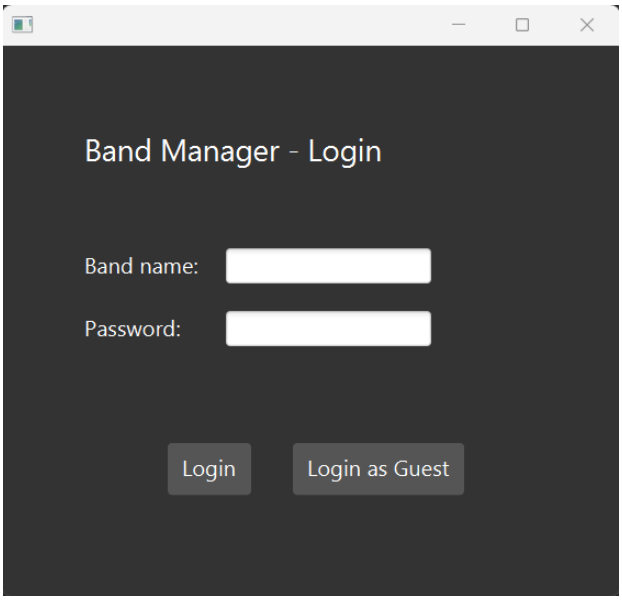
2- Once the connection with the "bandmanagerdb" is done, to establish the connection with the client, compile and run **ClientLoginGUI.java**, an interface will open where the user will be prompted to log in to the system with the right credentials or log in as guest.

To summarize, in order to run Online Service you must:
-Run the server
-Run the client

The application allow two possible access modes:

In the **Guest Mode** the user can only view the data (songs and events) of the band of interest.

In the **Band Profile Mode**, after entering the right credentials (see next paragraph for more information), the user can edit the songs and events in the database related to his band.



Login Info

To log in correctly, you must enter username and password in the fields provided in the Login window. The server allows 2 possible username-password pairs, which are shown in text file *accounts.txt,* in the project folder.

The syntax of each line of the file is: *<username> <password>,* where the username coincides with the name of the band.

The program does not provide for user registration so the only way to add a new account is to edit *the accounts.txt* file, entering the username and the password (with the right syntax) of the new account.

Main features Guest Mode

Once logged in with guest mode, a graphical interface will be offered for choosing the band whose information you want to know. It will be possible to proceed to the next screen only if a band has been chosen, by clicking on it and pressing "ok".

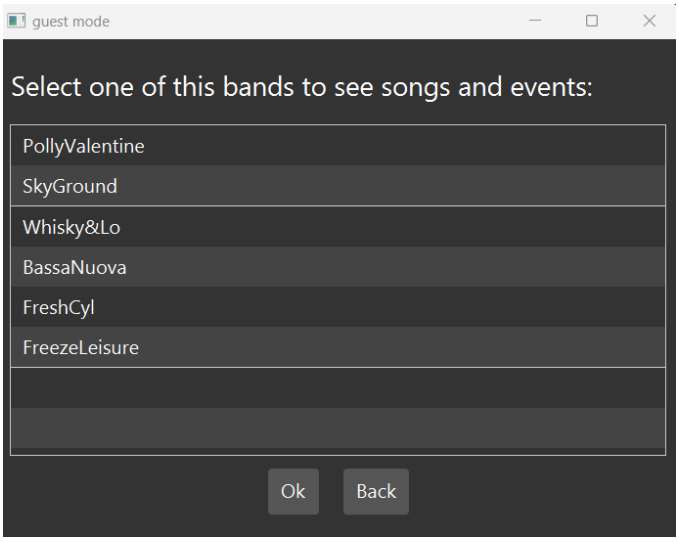
Once the band of interest has been selected, a window will open in which the user can decide what to consult about the selected band:

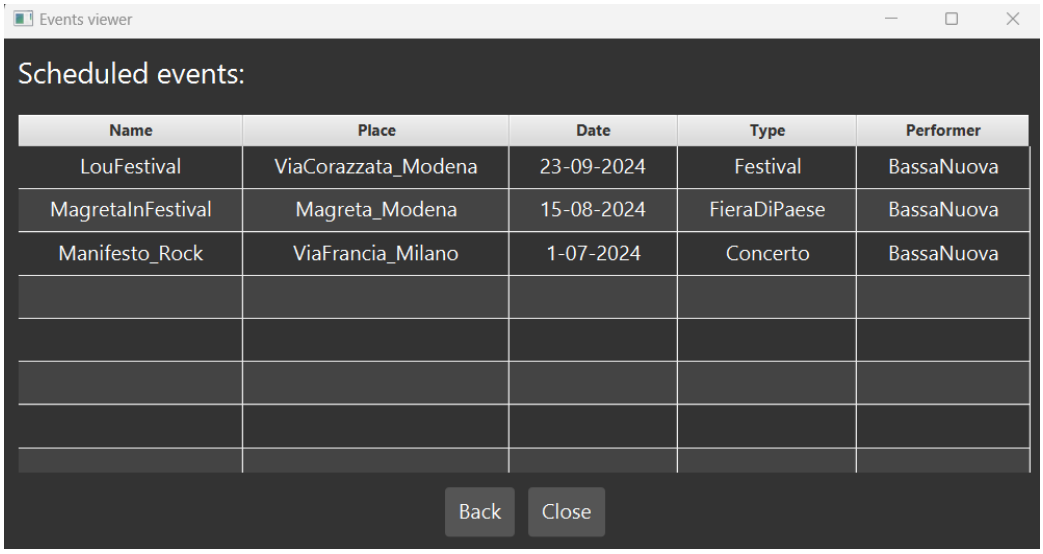
1) The "**songs**" button will allow you to see all the songs in the playlist with some of their characteristics (name, artist, length and bpm).

2) The "**events**" button will allow you to see the live events scheduled by the band of interest, also providing useful information (event name, location, type of event and date).

Each screen has a "**back**" button to return to the previous screen.

The window displaying songs or events for the selected band also has a "**close**" button to close the application.



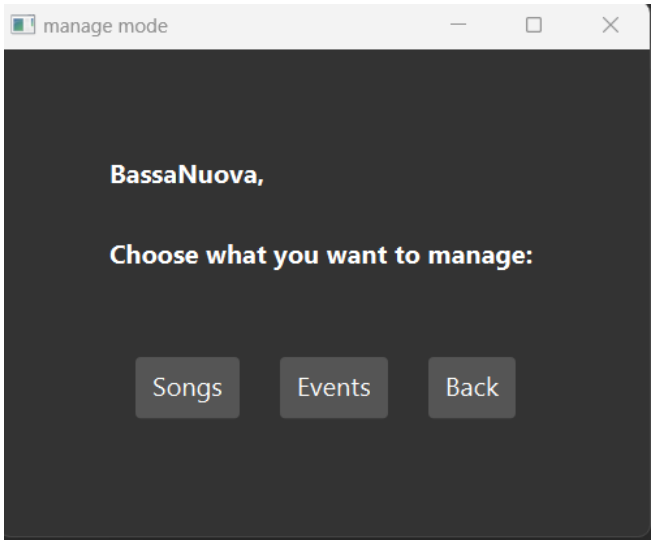


Main features Band Profile Mode

Once logged in with credentials of a specific account (band name), a window identical to that of the guest mode will open in which the user can decide what to see or modify about his data:

1) The "**songs**" button will allow you to see all the songs in the playlist with some of their characteristics (name, artist, length and bpm).

2) The "**events**" button will allow you to see the live events scheduled by the band of interest, also providing useful information (event name, location, type of event and date).



Once you have chosen whether to manage the songs or the events, a window will open in which it will be possible to consult all the songs (or events) and it will be possible to carry out some operations on them:

1) **ADD.**

Adding a new song/event to the list.

The add button will open a new interface in which, by correctly filling in the appropriate fields, it will be possible to add a new song or event. However, in adding new song it is necessary that there are no white spaces in the names (it is recommended using the underscore if the name of the song or the artist involves multiple words), that the length of the song is expressed in the format *<minutes>.<seconds>* (ex. 4.30 will be 4 minutes and 30 seconds) and that the bpm is a valid number.

Also regarding the addition of a new event, no white spaces should be inserted, but there are no further conditions to be respected, as there are no numerical fields.

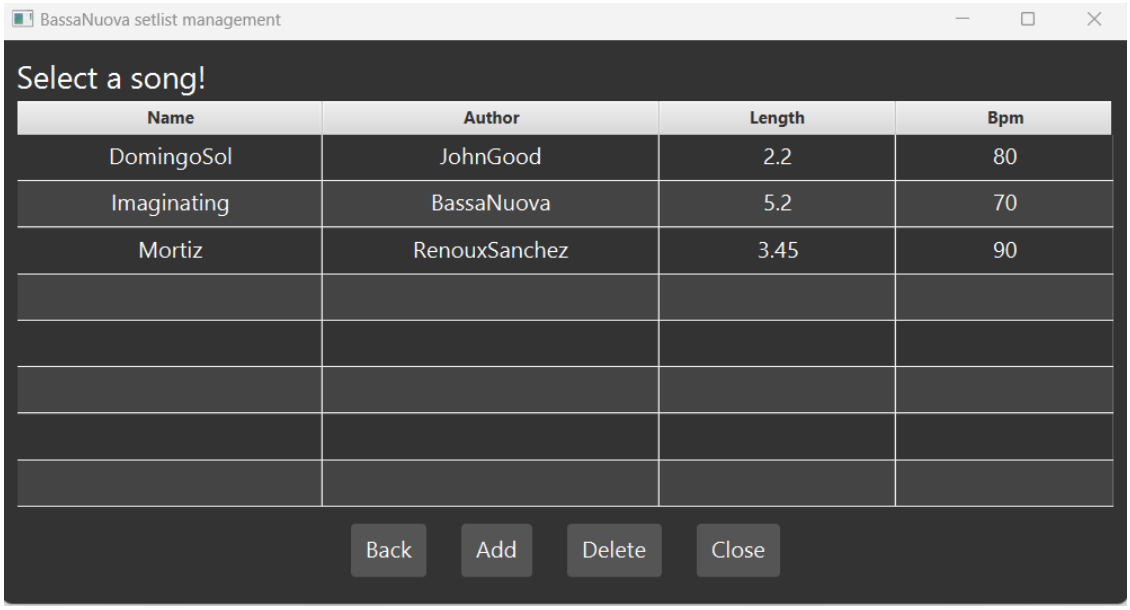
2) **DELETE**

Deleting a song/event from the list.

The delete button once pressed will delete the selected song/event. If this is pressed without anything being selected it will be reported to the user at the top in the same window.

Again, each screen has also a "**back**" button to return to the previous screen.

The window displaying songs or events for the selected band also has a "**close"** button to close the application.



Extra Note on application architecture

In the creation of the software system, the **Layered architectural** pattern was adopted in which 3 different layers can be distinguished in ascending order of level:

1) Database Layer

The *DataAccess.java* file is part of this level.

In this level there is complete management of the iterations with the database, such as data updating, maintenance and their recovery in order to show them to the high-level user who requests them. In this level, all those functions are therefore implemented that allow the database to be manipulated through queries and the SQL language, even during the database creation phase itself.

2) Server Layer

The *ServerLogic.java* is part of this level.

In this intermediate level, all the logic regarding the services that are requested by the user and which must be satisfied is managed. The methods of the class that deals with this level are called by the client and are responsible for acquiring the data correctly through communication with the underlying database layer and preparing them for display.

3) Client/GUI Layer

The *ClientLoginGUI.java, guestMainGUI.java, BandManagementGUI.java* are part of this level.

In this higher the methods of the class mainly deal with the graphics and the arrangement of the JavaFX elements in the various windows dedicated to the application GUI. The windows that are created take the data through communication with the logical level of the server that is responsible for providing them, but in general it can be said that in this level the user experience and its interaction with the application are managed.