

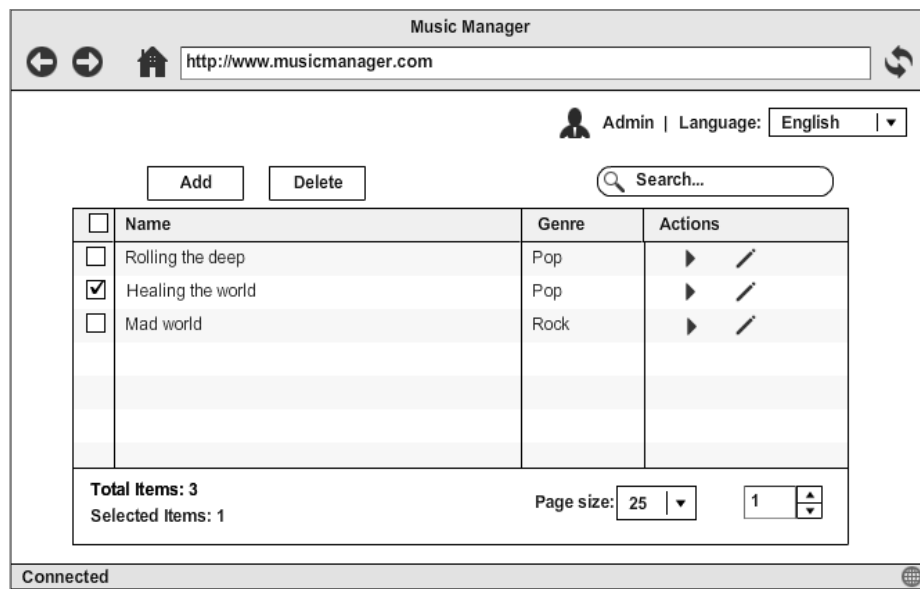
Music Manager

The goal of the exercise is to make a music manager application. A music manager application allows you to browse through a music library, add new entries and delete them (so the usual Create, Update and Delete stuffs (CRUD)).

All music data should be stored server side in a database. The client should be a web client. Communication between clients and server should go over REST.

On the server side we expect the application to be deployed in a Tomcat server. The client should always communicate to one dedicated service interface (MusicManagerService). The MusicManagerService implementation is a POJO which is using Spring to interact with the database.

How should the GUI look?

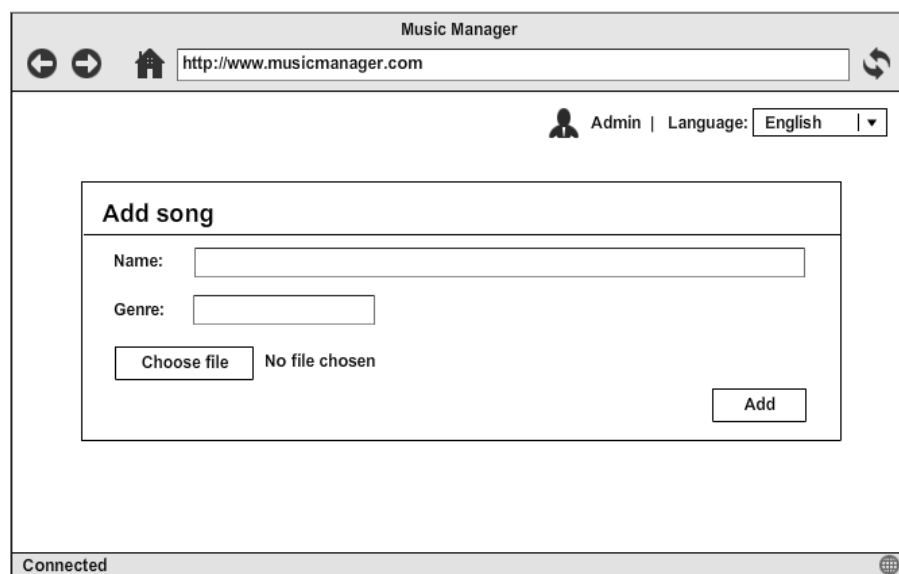


The screenshot shows a web browser window titled "Music Manager" with the URL "http://www.musicmanager.com". The page has a header with a user icon, "Admin", and a language dropdown set to "English". Below the header, there are "Add" and "Delete" buttons, a search bar with the placeholder "Search...", and a table of songs.

<input type="checkbox"/>	Name	Genre	Actions
<input type="checkbox"/>	Rolling the deep	Pop	
<input checked="" type="checkbox"/>	Healing the world	Pop	
<input type="checkbox"/>	Mad world	Rock	

At the bottom of the table area, it says "Total Items: 3" and "Selected Items: 1". To the right, there is a "Page size:" dropdown set to "25" and a pagination control showing "1" with up and down arrows. The status bar at the bottom says "Connected".

The GUI can be very basic: a table which lists all music files. There should be an action bar with 'Add' (which leads to 'Add Song' page mentioned below), 'Delete' buttons and a Search box.



The screenshot shows the "Add song" page of the Music Manager web application. The browser window title is "Music Manager" and the URL is "http://www.musicmanager.com". The header is the same as the previous screenshot, showing "Admin" and "English".

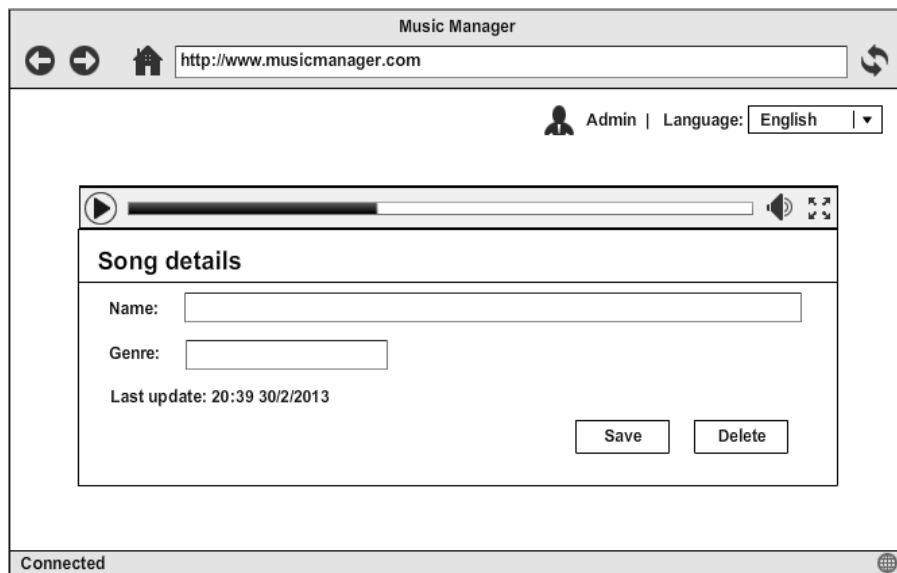
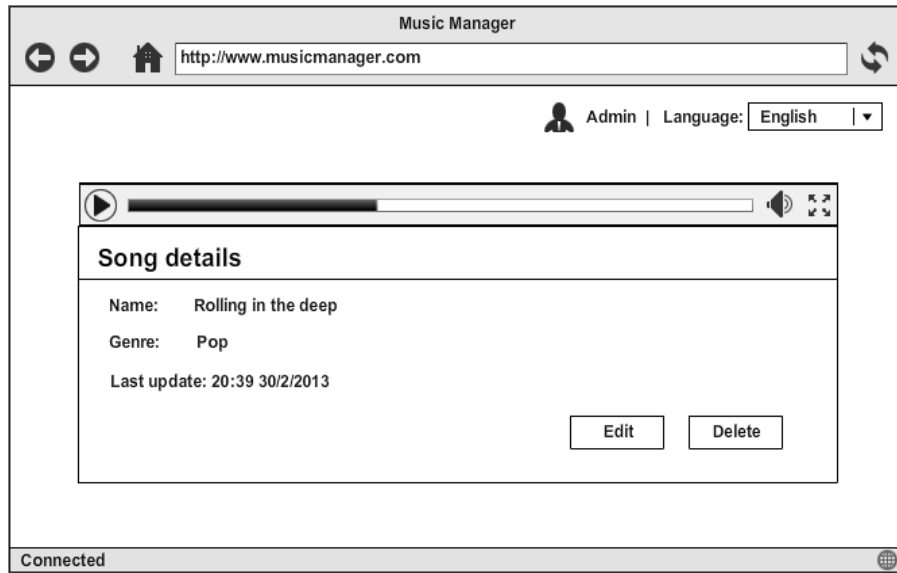
The main content area has a form titled "Add song". It contains the following fields and controls:

- "Name:" followed by a text input field.
- "Genre:" followed by a text input field.
- A "Choose file" button next to the text "No file chosen".
- An "Add" button at the bottom right of the form.

The status bar at the bottom says "Connected".

hmchung@tma.comvn

We also need to support 'Play' and 'Edit' modes for each song, as followings:



Technology requirements



Technologies need to be used:

- Spring
- Spring Data Mongo (Transaction, Mapping, Paging)
- Spring MVC
- MongoDB, Mongo Template & Tomcat
- Or Hibernate, MySQL

Nice to have:

- Spring Security (Login)
- Notifications to the clients

Technologies need to be used:

- Javascript
- Angular
- Twitter Bootstrap
- HTML5/CSS3
- JSON

Nice to have:

- Unit-testing (Jasmine, Karma)
- Localization (L10n) and Internalization (i18n)
- Multi-browser support (Firefox, IE, Safari, Chrome...) => just an improvement