Project Name: Muses Music Website

Project Introduction:

Muses is a music website, which allows user to search artist, albums, songs and listen the 30s' preview demo.

They are two type of users: common user and VIP user.
Both of them could to search artist, albums, songs and the preview demo. They could also add the artists and albums in favorite artists list and in favorite albums list.
VIP users have a playlist and they can add songs in their own playlist.
Common could press the add button of each song, but playlist in account is invisible to them until they upgrade to be a VIP user.

How to become VIP user?
Take the quiz.
Common user could find upgrade button in the menu to take the quiz. There are 10 music questions and if they could get more than 6 correct answers, they will become the VIP user and see the playlist in their account. Once they become the VIP user, they will not see upgrade label anymore.

Project data and database:

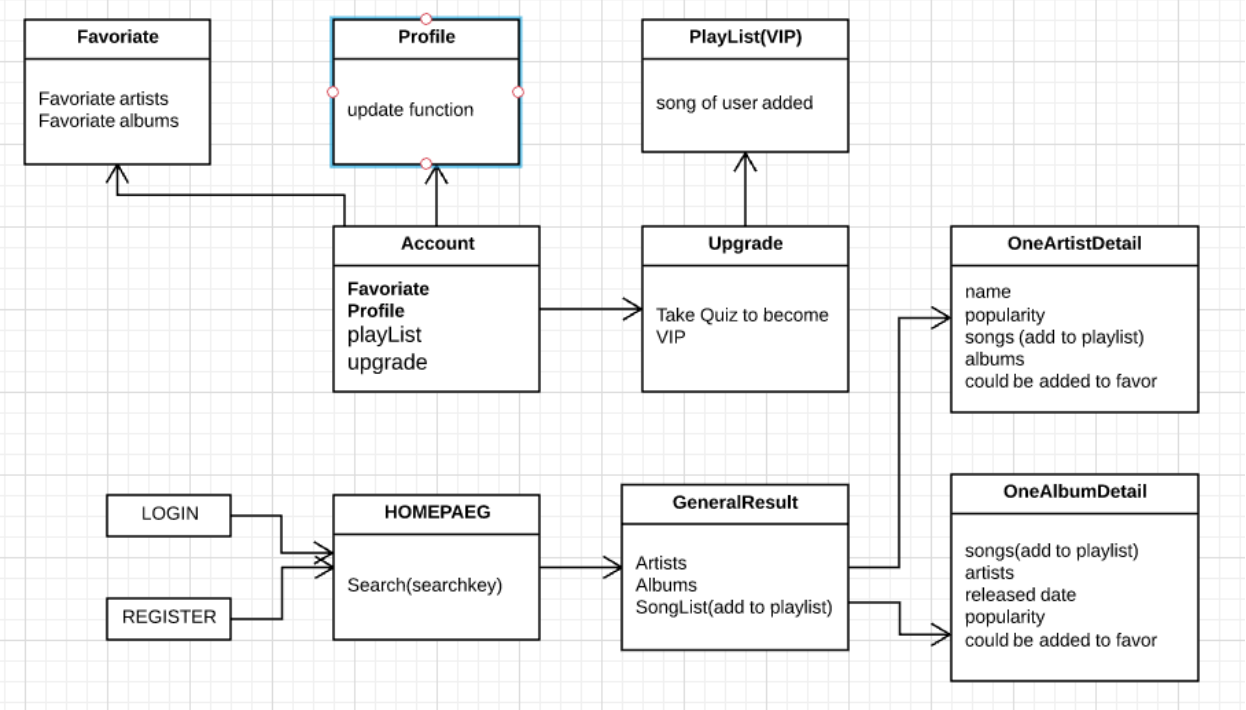
I use mongo dB to store user data. And all music data comes from Spotify API. The user schema is as following:

```
var UserSchema = mongoose.Schema({
    "username" : { type: String, required: true, unique: true},
    "password": { type: String, required: true },
    "firstName" : String,
    "lastName" : String,
    "email" : String,
    "VIP" : { type: Boolean, required: true },
    "LikeArtists": [{type: String}],
    "LikeAlbums": [{type: String}],
    "PlayList": [{type: String}],
    "dateCreated": { type: Date, default: Date.now }
}, {collection: "user"});
return UserSchema;
```

I did not create schema for artist, album and songs, instead, storing the Spotify id in LikeArtists, LikeAlbums and PlayList of user. And when I need to use the information I call API to get.

Project Implementation:

The general workflow is as fllow:



The front end is implemented by JavaScript, css, bootstrap, font-awesome and so on.
The back end

- Login: using local strategy.
- After login, each page will check logged status
- Register: using create user function
- Profile: using user update
- Search: enter the searchKey and call the find artist api, find albums api, find tracks api by parsing searchKey. Artist and albums result is shown by picture list, and tracks is listed in a table. Each song includes the artist, albums, 30s preview demo, popularity and a button allowing user to add into playlist.
  Click the picture, user could jump to detail of one artist or one album.
  In each page, user could see the detail information including popularity, released date, name, songs list and could listen the demo. User could also click like/unlike button to add/delete the artist/album from their favorite artist/album list.
- In each page, user could jump back to homepage, account, logout. In account page, user could see the sidebar including profile, favorite artist. VIP user could see PlayList. Click each they will rend detail information.
- All the find function and list item is based on the external api.