Bilkent University

CS - 353

DATABASE SYSTEMS

TERM PROJECT FINAL REPORT

NEMO

A Web-based Social Music Platform

Kerem Ayöz - 21510569 - Section 1

Ali Bulut - 21402408 - Section 2

Musab Erayman - 21401025 - Section 3

Ömer Faruk Karakaya - 21401431 - Section 3

Department of Computer Engineering, Bilkent University, Ankara, TURKEY

14.05.2018

1. Introduction

1.1. Brief Description

NEMO is a web-based social music platform with a database and efficient built-in queries. There are some other web-based music platforms such as Spotify to have a look at. Although we have been inspired and influenced by Spotify we have added some important new features. We provide a neat user interface for users to interact with the system easily. Users can easily socialize while they are listening to music. The detailed user manual is explained in Chapter 6 (User's Manual). One of the thing that we have added to our system different from Spotify is standard users and artists can communicate directly and users can see events of artists(such as concert, autograph sessions etc) in the system.

1.2. Contributions by

It is important to say that from beginning of the semester all group members have contributed to the project by high effort and equally from documenting to implementation. All group members have written reports and code. In that case, we are all grateful to each others.

- **Kerem Ayöz**

  Kerem implemented mostly the GUI part of the program. He created the page templates with HTML. He also helped for structuring the application and writing some queries. He also contributed to make the system play music.

- **Ali Bulut**

  Ali helped to provide related sql queries for database. According to needings he added data to db and tested for his sql part. Also contributed to Market Page and Login page of GUI. He helped about Login authentication and song and album rate.

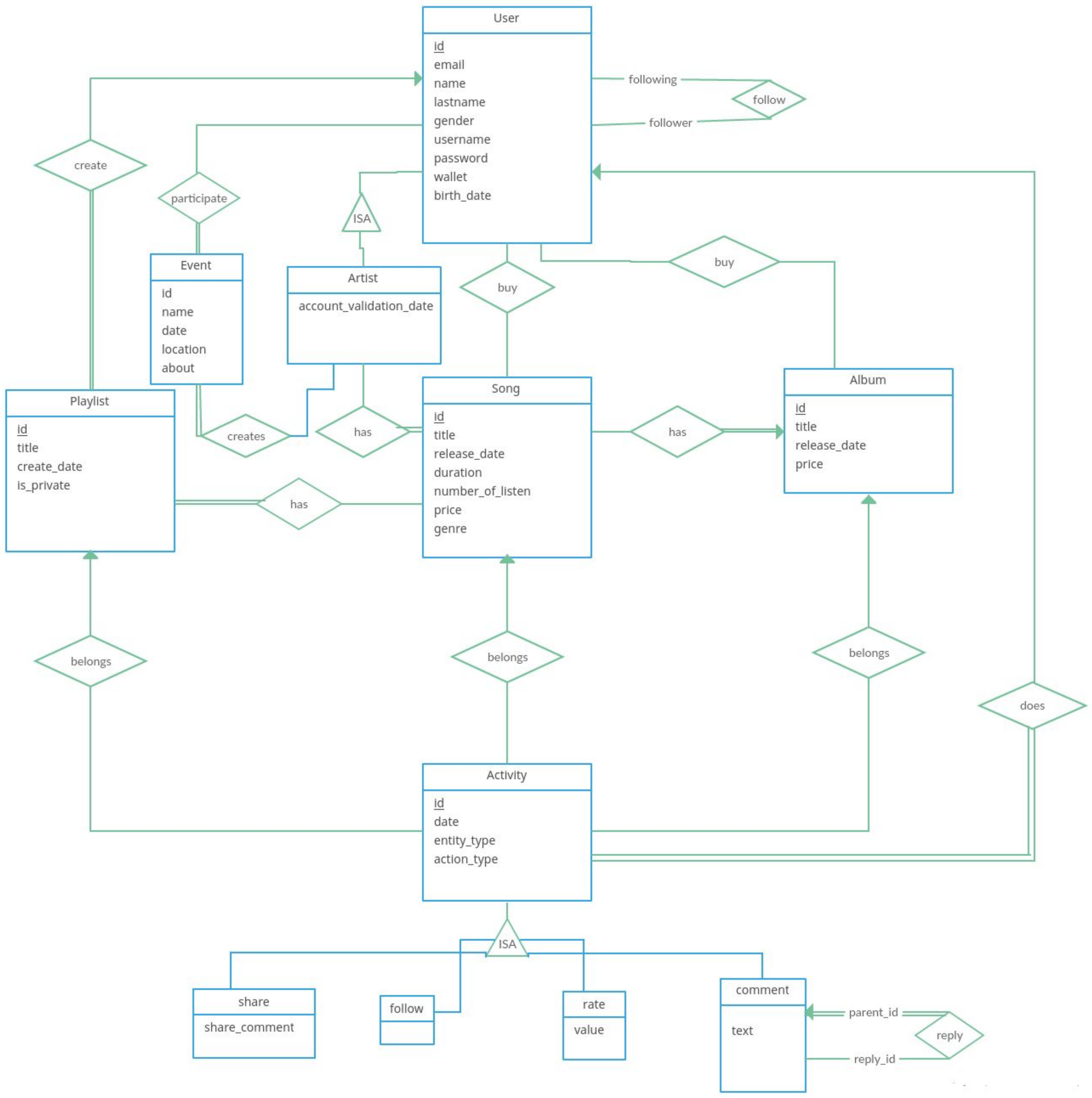
- **Musab Erayman**

  Musab helped to provide required database methods prototypes and implement some DB methods with related SQL queries. He contributed to logout authentication and helped to implement UI of top menu. He also wrote a fake data generator to fill the DB for testing.

- **Ömer Faruk Karakaya**

  Ömer helped to setup the system and MySQL server on Amazon RDS. He also worked on the connection of GUI and Database parts of the program. He wrote some of the DB methods and wrote related SQL queries, too. He also contributed to make the system play music.

2. Final E/R Diagram



3. Final Tables

Some changes are made in some tables to avoid redundancy in tables and to execute less costly queries.

Table ***participation*** divided into two seperate tables which are ***participation_artist*** and ***participation_user***.

Previous design:	participation(user id, artist id, event_id)
Revised design:	participation_artist(artist id, event_id)
	participation_user(user id, event_id)

The problem with the previous design was redundancy. For instance when an artist created an event, column user_id would be null since there is no user to participate to the event yet. Also when a user decided to participate to the event, there would be too much redundant data in terms of duplication of (artist_id, event_id) pair.

Table ***genre*** is removed and ***genre*** attribute is added to ***song*** table as it can be seen from the final revised E/R diagram.

4. Implementation

4.1. Basic Concepts and Used Technologies

The system uses client-server software architecture. We have the database, database methods, and the web server on the server side. We used ***Python*** programming language, ***Flask Micro Framework***, ***MySQL*** database on ***Amazon RDS(Relational Database Service), PrimeUI*** javascript library based on ***jQuery***, ***HTML*** and ***CSS***.

4.2. Database Hosting and Connection

We used ***MySQL Community Server*** to host the database. ***Python***'s ***pymysql*** package is imported and used to create database connection and we executed SQL queries which are written by us with ***pymysql's*** provided connection and execution functions.

4.3. GUI Implementation

To implement the GUI of the Nemo application, we used HTML, CSS, Bootstrap, Javascript and their frameworks as well. Initially, we designed the top menu and bottom music player part of the application, which is mostly common in all pages. After that we added the page specific elements for each page by combining the top and bottom part with them. While designing the elements, we firstly designed the row/column layout of the page. Then we add all the needed elements to their specific positions. This strategy makes the implementation much easier and enables us to work separately for the implementation of pages.

Also, we used templates from the Internet such as PrimeUI, Bootstrap etc. Their pre-implemented tables, elements are added to our page designs. We also modify them according to our needs. We used some usefull applications such as Pingendo, which dynamically shows the content of page while lets the code change and also gives available templates to us. It also creates the CSS files automatically.

For user interface of audio player we used wavesurferjs which is backed up with html5 audio player. Wavesurferjs automatically calculates bitrates of audio file and create waveform visualization. We ported waveformjs into our design to create audio player.

4.4. Faced Problems and Our Solutions

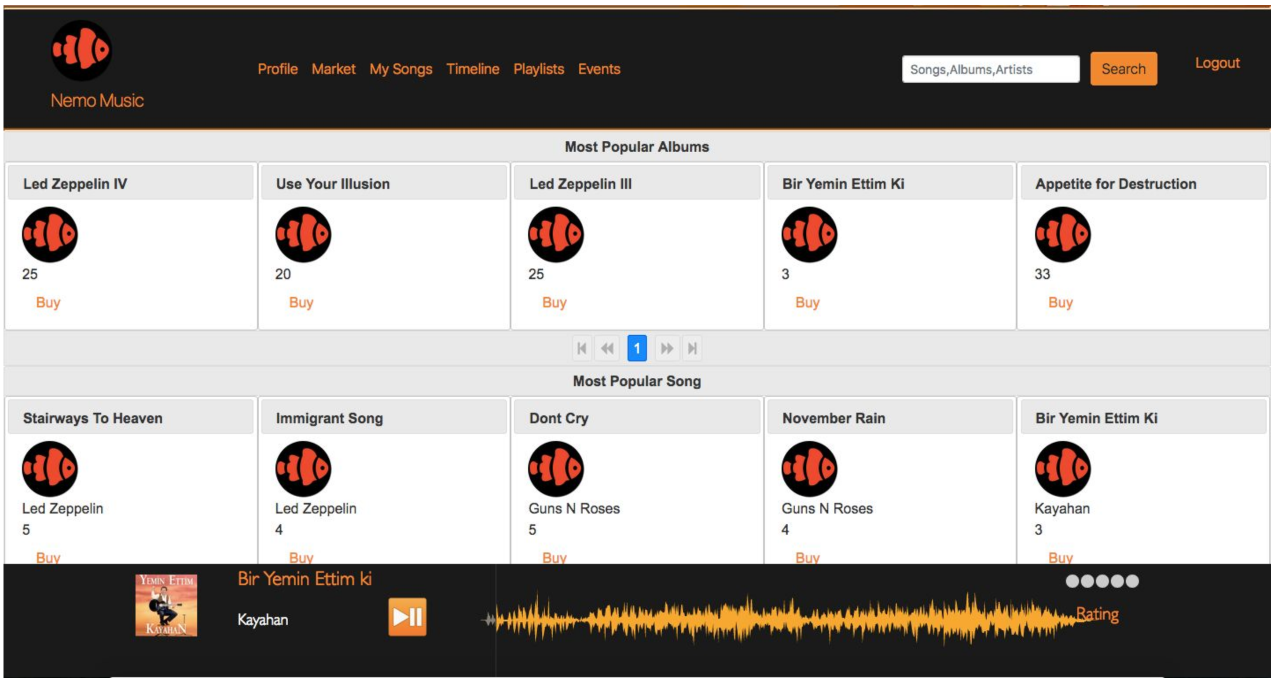
Since none of us had enough experience in web development we have faced with some difficulties. Initially, we need to open ports for web service and maintain connection html files. Also html files have some parts that has to be changed dynamically. To do these tasks we decided to use Flask microframework. It was easy to learn because it is minimalist. Also there is enough tutorials for Flask.

5. Sample Output Reports

In market page we ordered musics and albums by their rate. To do that we created rate views for all musics and albums and grouped them by their ids. According to these ordered results we provided them into market page.

```
create_view_song_rate = """create view song_rate as
        SELECT s.id as song_id, avg(r.value) as rate
        FROM song s join activity a join rate r
        WHERE a.action_type = 'RATE' and a.entity_type = 'SONG' and r.activity_id = a.id and a.entity_id = s.id
        GROUP BY s.id"""

create_view_album_rate = """create view album_rate as
        SELECT a.id album_id, avg(r.value) as rate
        FROM album a join activity ac join rate r
        WHERE ac.action_type = 'RATE' and ac.entity_type = 'ALBUM' and r.activity_id = a.id and ac.entity_id =
a.id
        GROUP BY a.id"""
```



Most Popular Albums

Led Zeppelin IV	Use Your Illusion	Led Zeppelin III	Bir Yemin Ettim Ki	Appetite for Destruction
25	20	25	3	33
Buy	Buy	Buy	Buy	Buy

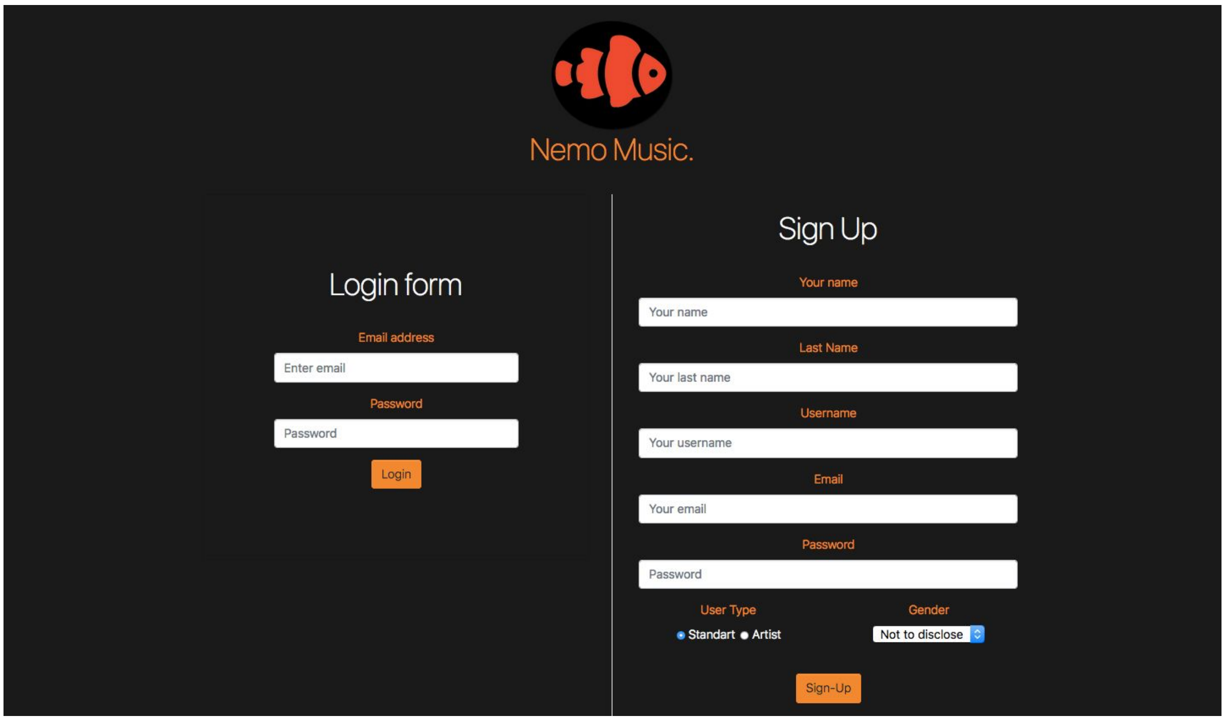
Most Popular Song

Stairways To Heaven	Immigrant Song	Dont Cry	November Rain	Bir Yemin Ettim Ki
Led Zeppelin	Led Zeppelin	Guns N Roses	Guns N Roses	Kayahan
5	4	5	4	3
Buy	Buy	Buy	Buy	Buy

Market Page

6. User's Manual

Users could start to use our system by just opening the home page of the web site. After opening that screen, user could login by entering his/her email and password and pushing Login button. Also if the user is new to the system, he/she can create an account to use Nemo. New users has to fill up all related areas below the Sign Up part. After giving needed informations the new user has to push Sign-Up button for complating of this process.


Nemo Music.

Login form

Email address

Enter email

Password

Password

Login

Sign Up

Your name

Your name

Last Name

Your last name

Username

Your username

Email

Your email

Password

Password

User Type

● Standart ● Artist

Gender

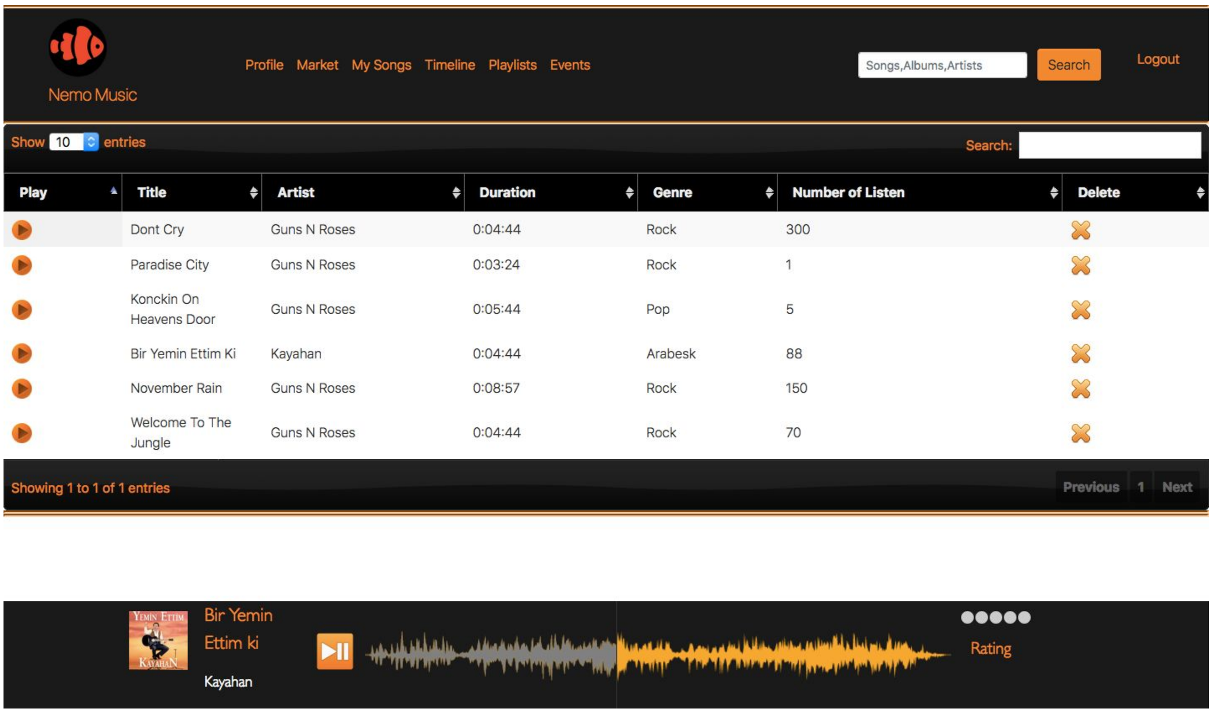
Not to disclose

Sign-Up

Login/Sign-Up Page

In Nemo, there are mainly 2 different user types. One of the type is called Standart which is the main user type of Nemo. The other user type is called "Artist", which is an account of artists who are able to create events. Standart users could buy musics, create playlists, listen songs, like/rate/comment/follow playlists, songs; follow other users and keep track of upcoming events in Nemo.

Artist type of user could do whatever the Standart user can, additionally they can create events for Standart users to attend. Artists also have songs they released in their profile page, other users could see the songs released by that particular Artist.

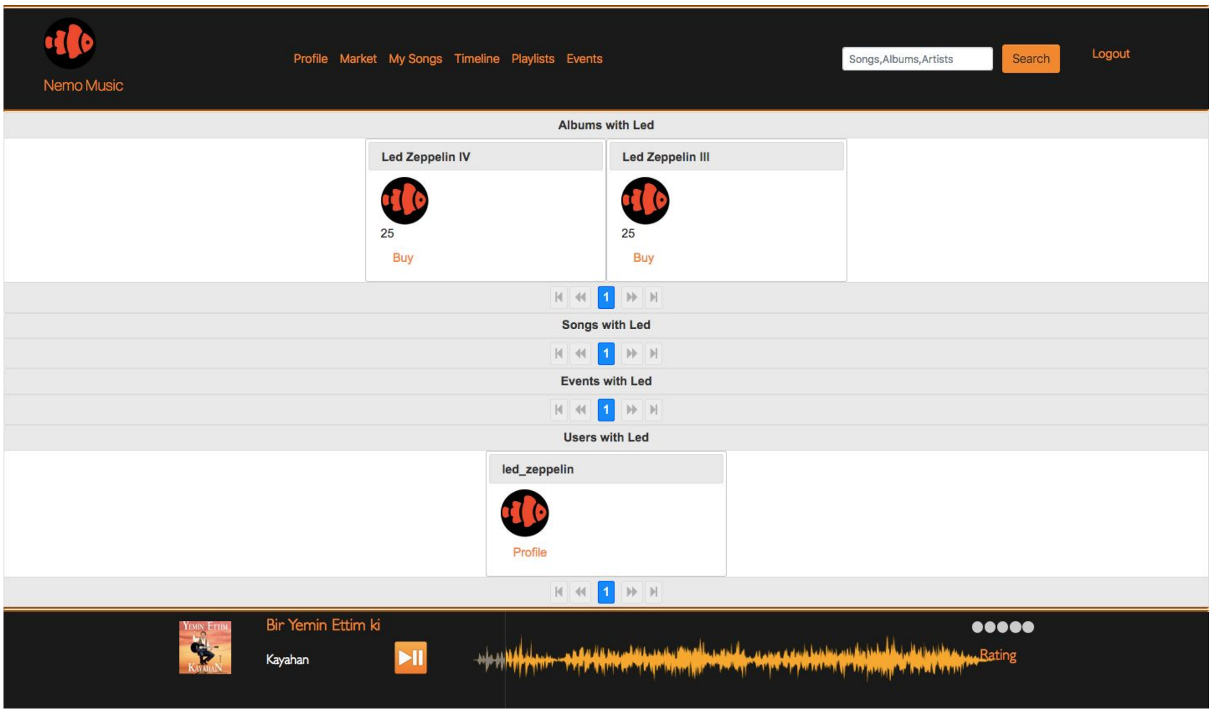

Profile Market My Songs Timeline Playlists Events

Show 10 entries

Play	Title	Artist	Duration	Genre	Number of Listen	Delete
▶	Dont Cry	Guns N Roses	0:04:44	Rock	300	✖
▶	Paradise City	Guns N Roses	0:03:24	Rock	1	✖
▶	Konckin On Heavens Door	Guns N Roses	0:05:44	Pop	5	✖
▶	Bir Yemin Ettim Ki	Kayahan	0:04:44	Arabesk	88	✖
▶	November Rain	Guns N Roses	0:08:57	Rock	150	✖
▶	Welcome To The Jungle	Guns N Roses	0:04:44	Rock	70	✖

Showing 1 to 1 of 1 entries

My Songs Page


Albums with Led

Led Zeppelin IV
25
Buy

Led Zeppelin III
25
Buy

Songs with Led

Events with Led

Users with Led

led_zeppelin
Profile

Search Page