音樂串流平台

組名：李興旺旺仙貝

組員

資訊二　111703050　李興旺

資訊二　111703054　郭中弘

資訊二　111703045　李奕辰

資訊二　111703021　宋孟庭

分工

前端程式設計：李奕辰
後端程式設計：李興旺(大部分)、郭中弘、宋孟庭
ER_diagram設計、需求分析：全部人討論而成
ER_diagram, schema製作: 郭中弘
報告製作：全部人

需求分析

1.使用者:依照自己喜好，創建playlist，playlist支持新增、刪除、播放功能

2.管理者:維護每個使用者的狀態(創建了多少playlist、limit是多少)，擴充歌曲庫

需求分析

限制用戶創建的playlist數目:

limit會對應到user的id，並根據limit限制該用戶每個playlist可創建的歌曲數。每個用戶只屬於一個level，而該用戶的所有playlist接受level影響。

playlist之於歌曲及user的關係：

1. 一首歌可同時存在於多個playlist，而playlist也可同時儲存多首歌曲
2.一個playlist只隸屬於一個user，並只有該user能編輯
3.user可創建多個playlist

需求分析

songwriter之於歌曲的關係：

一首歌只可能被一位songwriter創建，而songwriter可創建多首歌

系統功能

music:

1.包含 M_id、M_title、original_url、thumbnail_url

2.管理者負責把歌曲加入到系統中

playlist:

1.包含 P_id、P_title、P_type、P_size、is_private

2.讓使用者可以自由的加入歌曲、編輯

3.會被限制playlist的歌曲數(根據level)

系統功能

user:

1.包含 U_id、L_name、F_name

2.主要用於辨識不同的使用者使playlist歸屬關係明確

3.使用者的資料以一個帳號(U_id)為單位，便於管理

limit:

1.包含 Level、limit_num

2.用於限制使用者單一playlist的容量，管理者可透過增加使用者的level，提高容量

系統功能

songwriter:

1.紀錄每一首歌曲的歌手，管理者可透過歌手合作，加入最新的歌曲

系統架構

網頁框架

使用python的Flask當作網頁的框架

- 輕量級
- 好部屬
- 用python，有很多package可以用

系統架構

資料庫

使用SQLite

- 好架設、不用server
- SQLAlchemy，自動轉成mysql語法

```
new_music_added_to_playlist = InWhichPlaylist(M_id=which_music_id, P_id=which_playlist_id, UID=current_user.id)
db.session.add(new_music_added_to_playlist)
db.session.commit()
```

系統架構

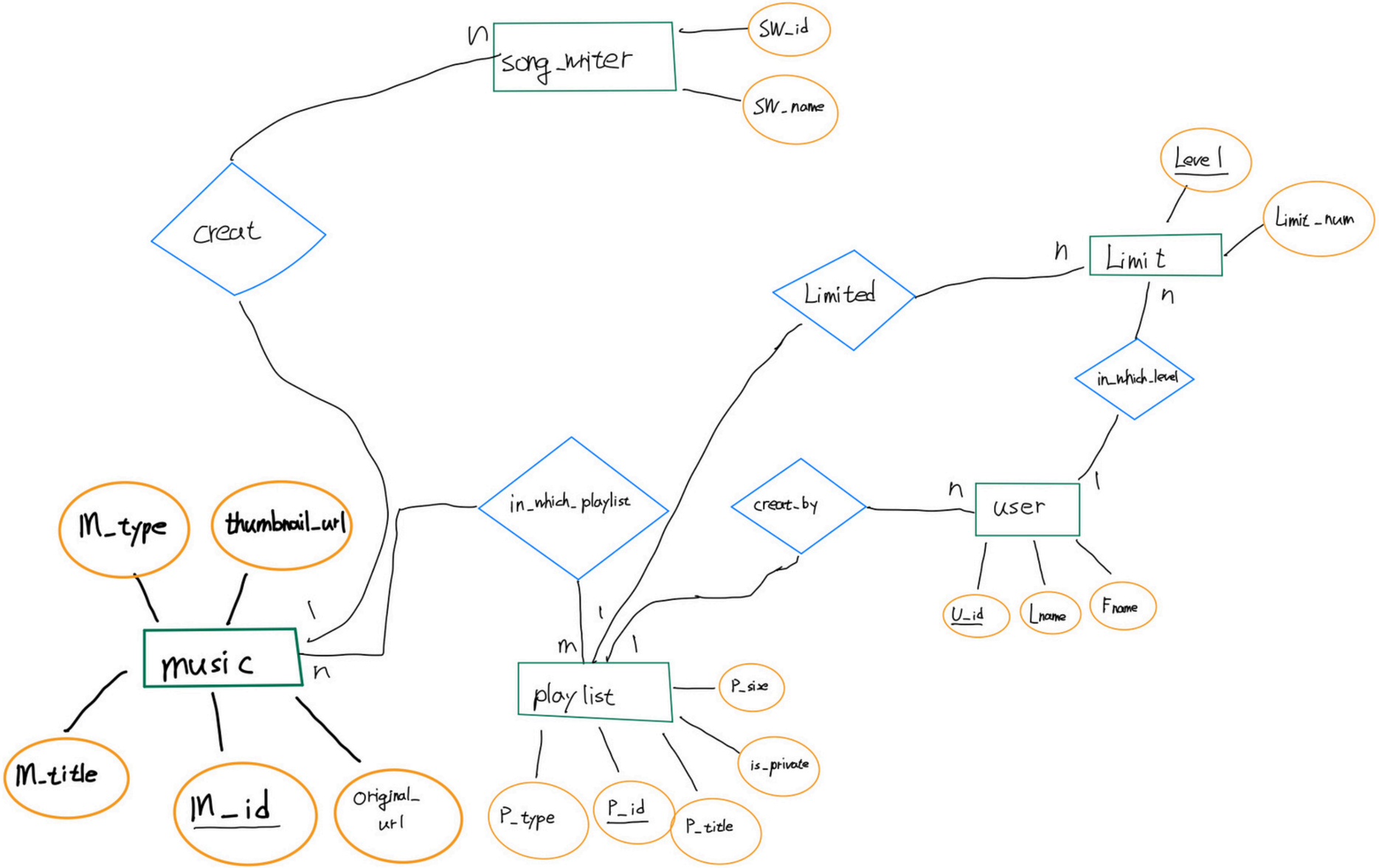
系統功能

主要架構為用flask寫api 前端用JS呼叫呈現資料
API Documents

前端

使用 jQuery、ajax 設計動態網頁音樂播放系統

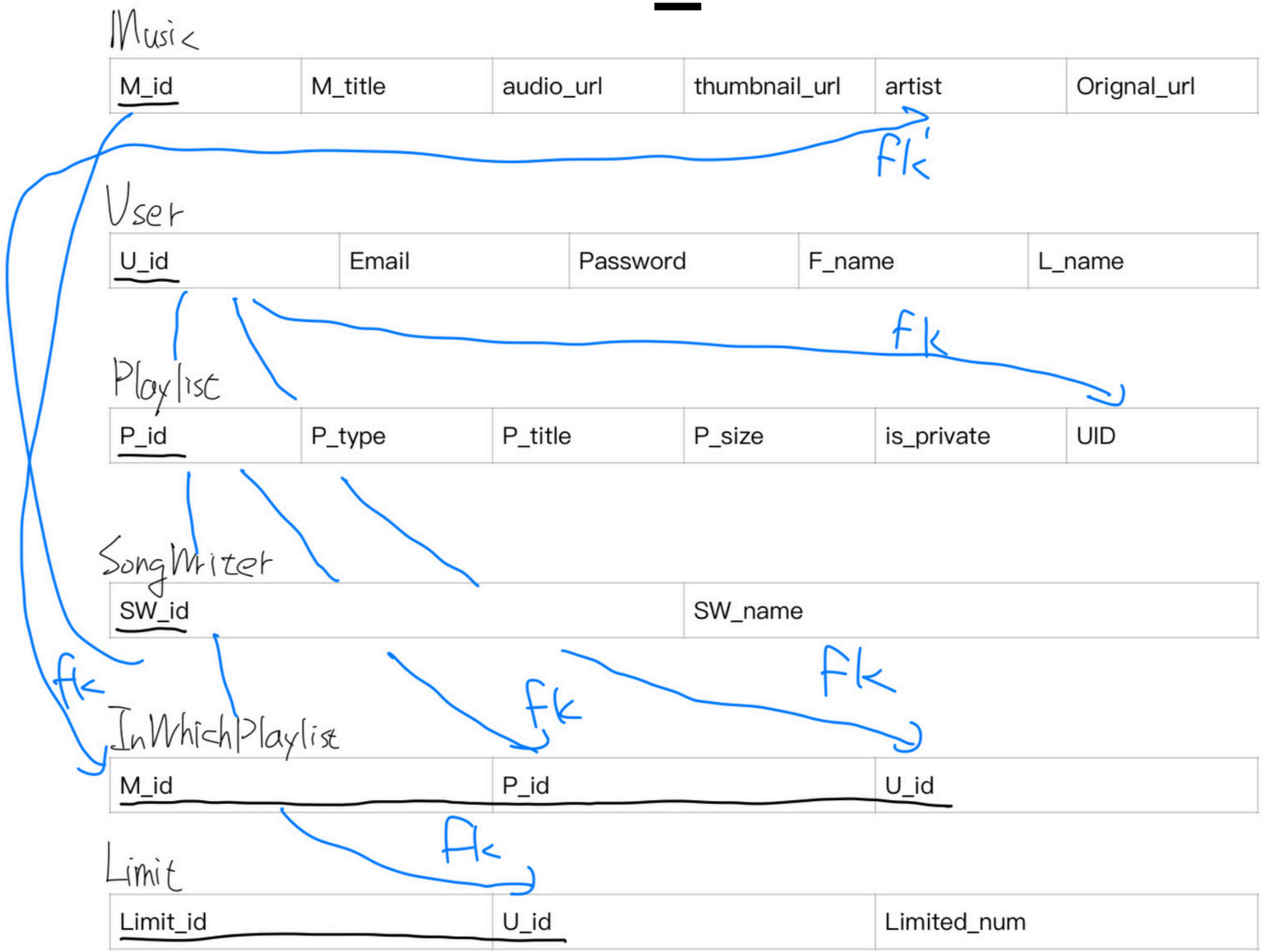
ER_MODEL


song_writer
SW_id
SW_name
creat
Level
Limit_num
Limit
Limited
in_which_level
in_which_playlist
creat_by
user
U_id
Lname
Fname
music
M_type
thumbnail_url
M_title
M_id
original_url
playlist
P_size
is_private
P_type
P_id
P_title

ER_MODEL

可參照需求分析與系統功能架構，
皆根據上述兩者設計

relational_schema



relational schema

InWhichPlaylist:

1.由於一首歌可同時存在多個playlist，一個playlist可儲存多首歌，故將InWhichPlaylist這個關係獨立成table，避免redundancy

limit:

1.須記錄user id，因此需要U_id作為F.K.

playlist:

1.playlist須記錄所有權，因此加入U_id作為F.K.

music:

1.一首歌只可能被一個songwriter創建，因此加入SW_id作為F.K.

心得

這次的final_project讓我們更深刻的理解到database在實務上的應用以及他是如何跟系統溝通，也深刻的理解到ER_model的設計是很大的一門學問，好的設計可以讓後端及前端省去很多功夫，儘管最後設計出來的系統還有許多可優化之處，仍是一次不錯的學習經驗。