健身房會員管理系統

625415

說明

健身房對很多人來說是保持健康的重要場所。為了更好地管理健身房的會員、教練、設備以及會員的訂閱和訓練計劃，我們需要設計一個全面的健身房會員管理系統。

組員

- 110703062 資訊三 周天華
- 110703054 資訊三 翁豪蔚
- 110703015 資訊三 王則正

分工

組員	工作內容	貢獻百分比
周天華	• 後端 • 撰寫專題報告	33.3%
翁豪蔚	• 前端 • 設計ER Model & Relational Schema • 撰寫專題題目說明、系統架構	33.3%
王則正	• 前端 • 撰寫專題題目說明、系統架構	33.3%

需求分析

教練 (Trainers）

- T_ID：教練編號
- T_Name：教練名稱
- Email_ID：電子郵件
- Phone：電話號碼
- Gender：性別
- Hire_Date：聘用日期
- Salary：薪水

會員 (Members）

- Member_ID：會員編號
- M_Name：會員名稱
- Phone：電話號碼
- Start_Date：開始日期
- Gender：性別
- Subs ： 訂閱項目
- Email_ID：電子郵件
- Height：身高
- Weight：體重
- Age：年齡
- Trainer_ID：教練編號

訂閱 (Subscriptions）

- Sub_ID：訂閱編號
- Price：價格
- Duration：持續時間 (月)
- Sub_Num：訂閱數量

設備 (Equipments）

- Eq_ID：設備編號
- Name：設備名稱
- Quantity：數量
- Cost： (一台或一組的)成本

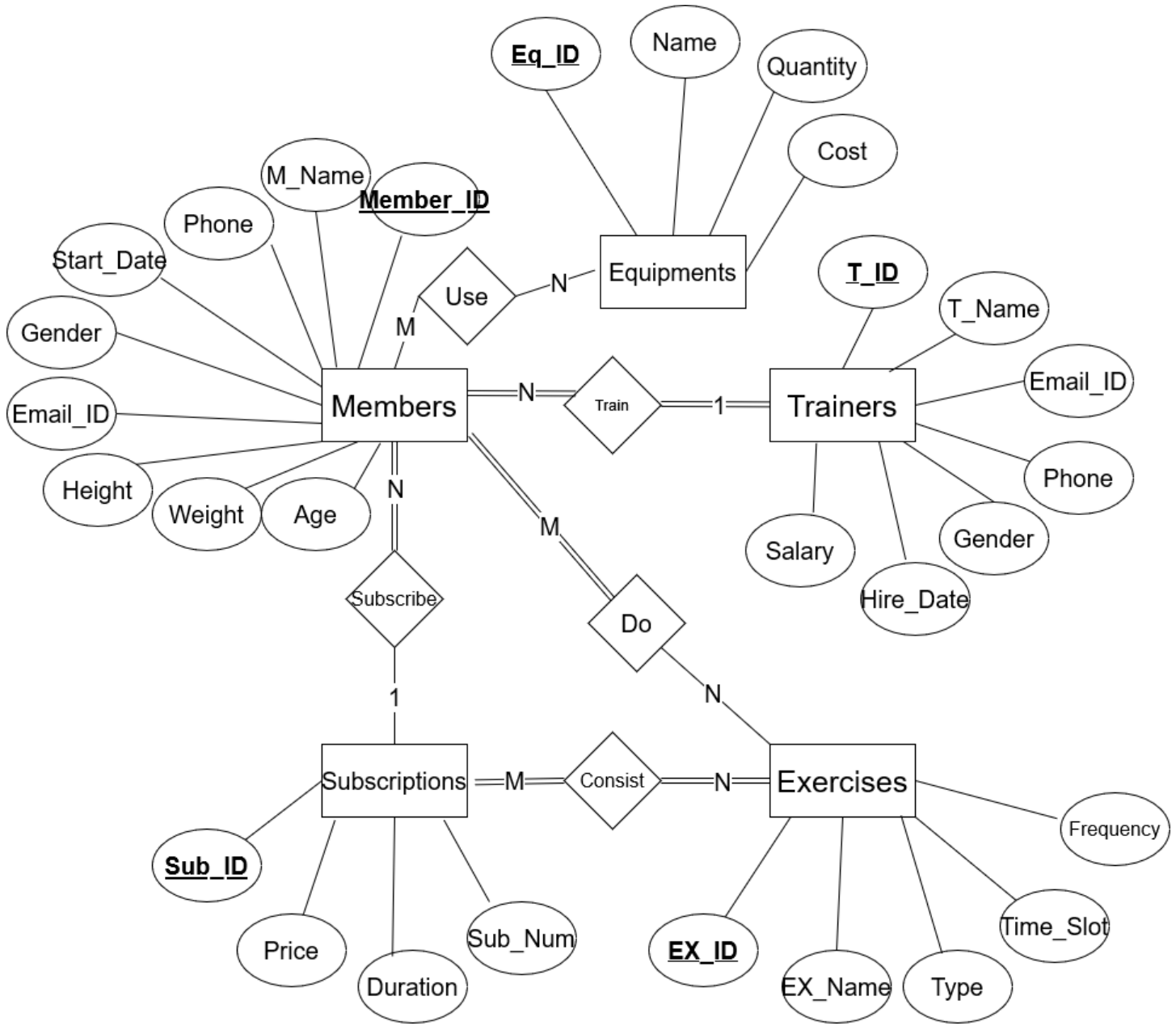
動作 (Exercises）

- EX_ID：動作編號
- EX_Name：動作名稱
- Type：動作類型 (手臂、上半身、下半身)
- Time_Slot：時間段 (分/次)
- Frequency：頻率 (次/週)

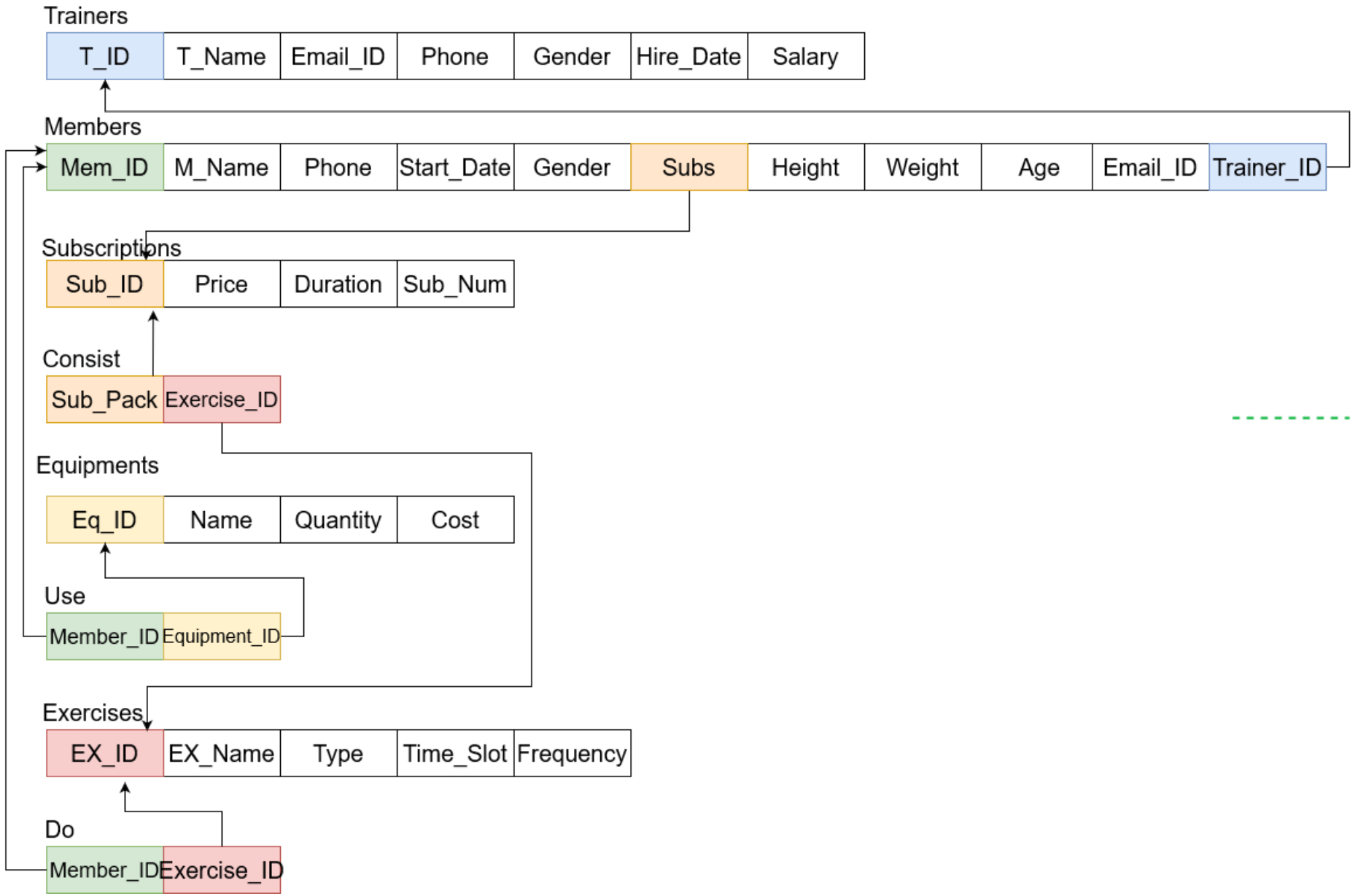
系統功能

- 主要集中在會員的個人資料管理、訂閱項目和課程參與、訓練計劃、設備使用上。
- 其餘也可看到訓練員，設備，訂閱項目，及動作的資訊並進行創建、查詢、修改、刪除

ER Model


Eq_ID
Name
Quantity
Cost
Equipments
M_Name
Member_ID
Phone
Start_Date
Gender
Email_ID
Height
Weight
Age
Use
N
M
Members
N
Train
1
Trainers
T_ID
T_Name
Email_ID
Phone
Gender
Hire_Date
Salary
N
Subscribe
1
M
Do
N
Subscriptions
M
Consist
N
Exercises
Sub_ID
Price
Duration
Sub_Num
EX_ID
EX_Name
Type
Time_Slot
Frequency

Relational Schema



系統架構

```
.
|-README.md
|-backend
| |-requirements.txt
| |-database.py
| |-app.py
| |-start.py
| |-data.py
|-frontend
| |-public
| | |-api
| |-src
| | |-main
| | |-store
| | |-App.js
| | |-index.js
| | |-style.js
```

開發

- DBMS: SQLite
- Back-End Programming Language: Python
- Front-End Programming Language: JavaScript
- 後端建立資料庫及使用*Flask*框架撰寫*API*
- 前端使用*React*框架，並透過*axios*處理資料請求

心得

在這次的期末專題實作中，我們遇到了很多挑戰，但同時也有不少的收穫。在思考題目的過程中，我們發現，要發想出一個資料庫的架構真的不是件容易的事，我們必須考慮到架構的合理性，資料是否符合我們的要求，以及這個架構是否為最好的架構。實作上我們使用SQLite當作DBMS，後端框架則使用Flask來串接資料庫，在實作中我們發現，雖然SQLite和MySQL同為RDBMS，卻有些許的不同，例如在Data Type上，MySQL對於字串有不止一種的Data Type，而SQLite只有Text這個Data Type；而MySQL自帶Foreign key的功能，SQLite則需要輸入指令才能開啟Foreign key的功能。在串接資料庫的過程中，我們並沒有碰到太大的困難，主要的挑戰在於我們需要思考如何把不符合我們資料需求的資料擋下來。前端我們使用了React框架進行開發，通過axios處理資料請求。React的組件化設計使我們能夠快速開發，提高開發效率，讓我們在前端的開發上少了很多阻礙。

收穫

這次的專案實作經歷讓我們學到了很多寶貴的經驗：

- **團隊合作的重要性**：在有限的時間內，我們每個人都盡了自己所能做到的事，分工明確，這是我們能夠完成任務的關鍵。
- **技術選型的合理性**：選擇適合的技術能夠大大提高開發效率。SQLite簡單易用，Python和Flask的強大功能，React的高效前端開發，這些都為我們的開發提供了保障。
- **時間管理與壓力應對**：在短時間內完成大量工作，我們學會了更好的時間管理和壓力應對策略，這對我們未來的工作和學習都有很大的幫助。