

# TEACHMATE

## DBMS Final Report\_Group 9

隊員：盧柏瑜 侯程麟 許馨文 藍翊齊 吳堃豪

### Table Of Contents :

#### (一) Introduction

#### (二) Requirement Analysis

1. Project Scope
2. Need Analysis
3. Economic Feasibility Analysis
4. Preliminary Schedule (Gantt Chart)

#### (三) Logical Design of the Business Transaction

1. Conceptual Schema Design
2. Relational Model
3. Interaction with the Database
4. Constraints and Functionality

#### (四) Implementation On Plans

1. FDD chart
2. Teacher
3. Student
4. DFD chart

#### (五) Appendices

### 壹、Introduction :

為學生和導師提供便捷的溝通和協作空間，該平台允許學生發布他們的補習需求，包括科目、年級、時間安排、電話號碼和年齡等信息。同樣地，導師可以在平台上創建個人檔案，展示他們的教學經驗、專業知識、期望薪資、可用性和證書等。通過篩選學生尋找導師以及導師尋找學生，該平台能夠高效地根據學生的需求和導師的背景找到最合適的候選人。

### 貳、Requirement Analysis :

#### (一) Project Scope :

1. 確定功能與特性：TEACHMATE主要是在幫助使用者，可以以老師的身分

尋找學生，或是以學生的身分尋找老師。我們將主要功能聚焦在發布需求貼文，並用過濾貼文的功能讓尋找結果更貼近使用者的需求，當使用者對特定貼文感興趣時只需要按下request按鈕讓發文者知道後，並藉由貼文內的連絡資訊，讓雙方再去做進一步的溝通。

2. 界定專案邊界:TEACHMATE網頁只設計了發布貼文的功能，至於雙方媒合過程或後續交流則需透過外部管道，此舉能避免讓TEACHMATE網頁成為類交友聊天平台，讓使用者專注於用發布貼文之功能尋找老師或學生。
3. 管理期望:我們期望利用此TEACHMATE平台，讓更多使用者滿足其需求，並設計了evaluation sheet的功能，使老師多了一份公信力，讓學生能確保教學品質與該老師的專業度。同時，在與一位老師的教學結束後，學生也可以透過與老師建立契約的方式，讓更多學生了解該老師的教學方式。
4. 評估可行性:在讓雙方使用者達成協議並成功媒合教學後，我們鼓勵老師主動建立evaluation sheet，幫助往後增加自身的教學公信力與其依據的存在，須注意的是，建立一份契約需要花費500元，同時也是此平台的經濟來源，但這可以有效地避免老師有目的性地建立契約而造成evaluation sheet失去其公信力。

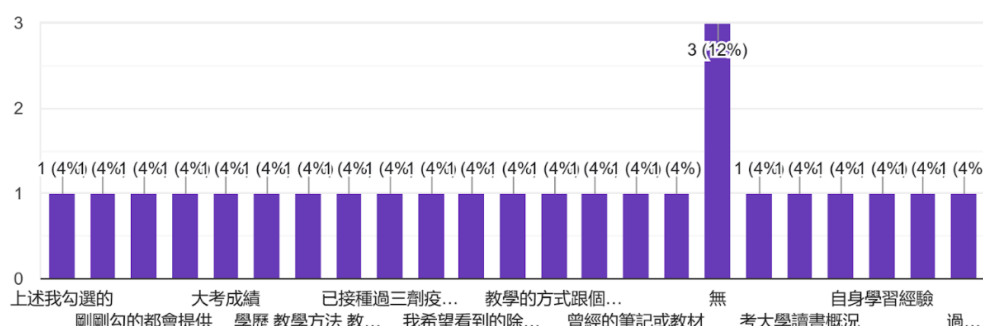
## (二)Need Analysis:

以下為我們當時Google Form蒐集到的資訊：

### 1. (圖一):

如果你是「老師」，你還會提供什麼個人資訊？

25 則回應

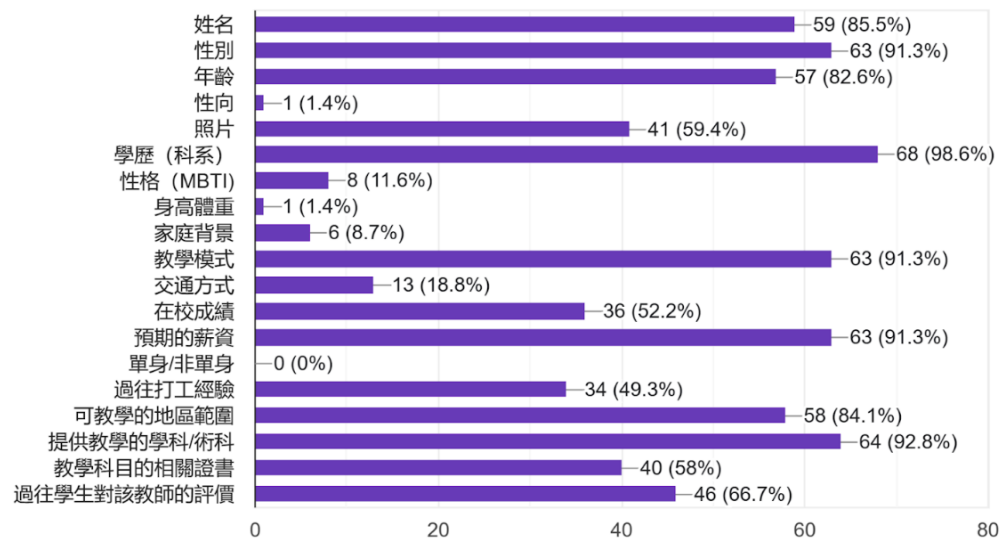


除了基本的資訊，我們透過表單發現現在的家教網缺乏老師更詳細的相關資訊，像是老師的教學方法、考大學讀書概況等等。

### 2. (圖二):

### 當您尋找家教老師時希望有哪些老師的個人資訊

69 則回應

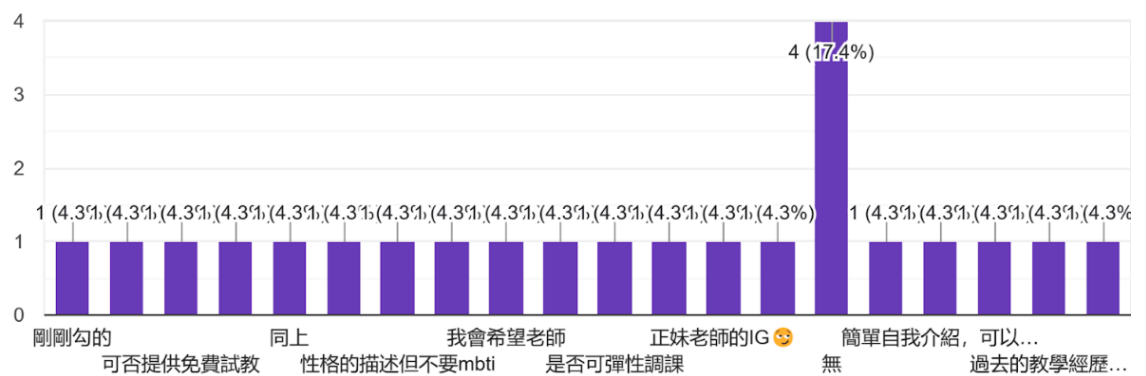


對學生而言，老師的學歷、提供的教學科目、可教學的地區範圍很重要，我們有將其納入我們的系統考量。

### 3. (圖三)

### 如果你是「學生」，你還希望有什麼老師的相關資訊？

23 則回應

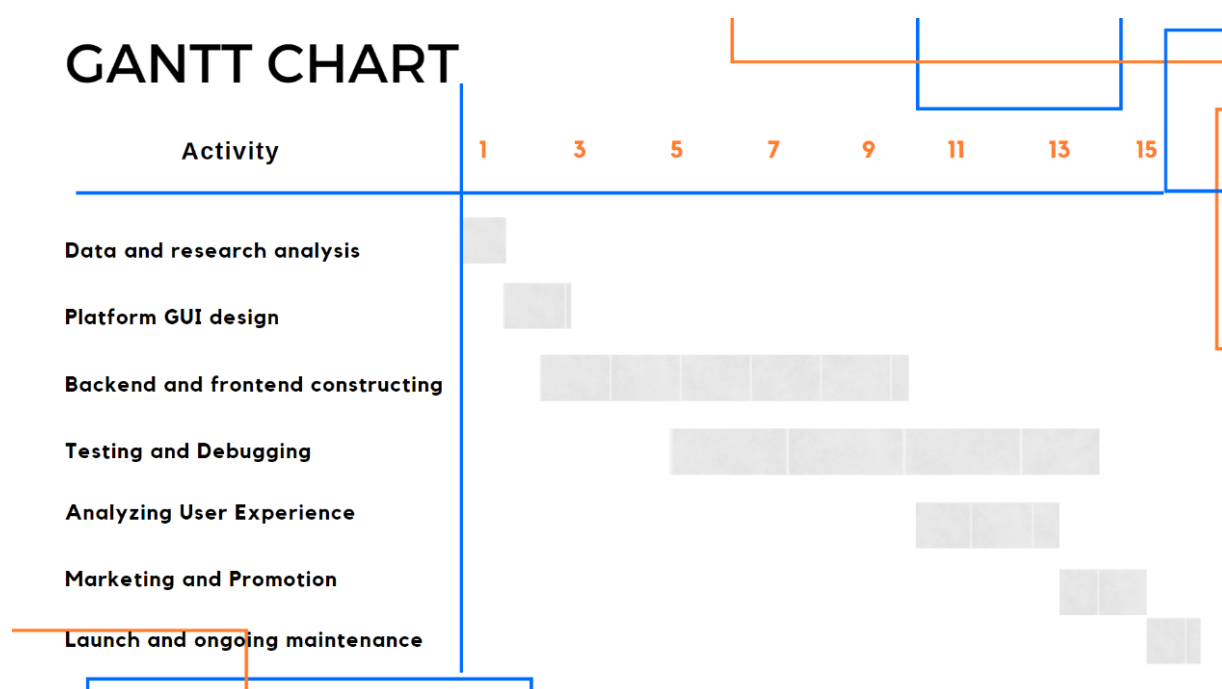


除了基本的資訊，我們透過表單發現現在的家教網缺乏老師更詳細的相關資訊，像是老師的教學方法、考大學讀書概況等等。

### (三) Economic Feasibility Analysis:

1. 市場需求:在當今社會中,許多學生和家長渴望找到一位能提升他們學業成績並培養更深入學習熱情的理想導師。因此,這樣的平台需求預計會很大。然而,值得注意的是,這個市場已經有許多成熟的競爭對手,包括FB的導師社群和其他各種應用程式。
2. 收入模式:這個平台有潛力通過多種方式產生收入,例如向教師和學生收取配對費用、獲取佣金以及實施其他賺錢策略。此外,平台還可以通過提供增值服務,如導師培訓和個性化學生學習計劃,創造額外的收入來源。為了確保我們團隊的穩定和可持續的收入,考慮將平台轉為基於會員制的模式,以獲得可循環的收入,以支持持續的維護和更新。
3. 平台維護和更新:平台運行速度優化、錯誤調試和創新的成本。
4. 客戶服務:提供實時客戶服務或線上訊息服務的成本。
5. 市場營銷:製作廣告、付費廣告、影響力行銷和社交媒體管理的成本。

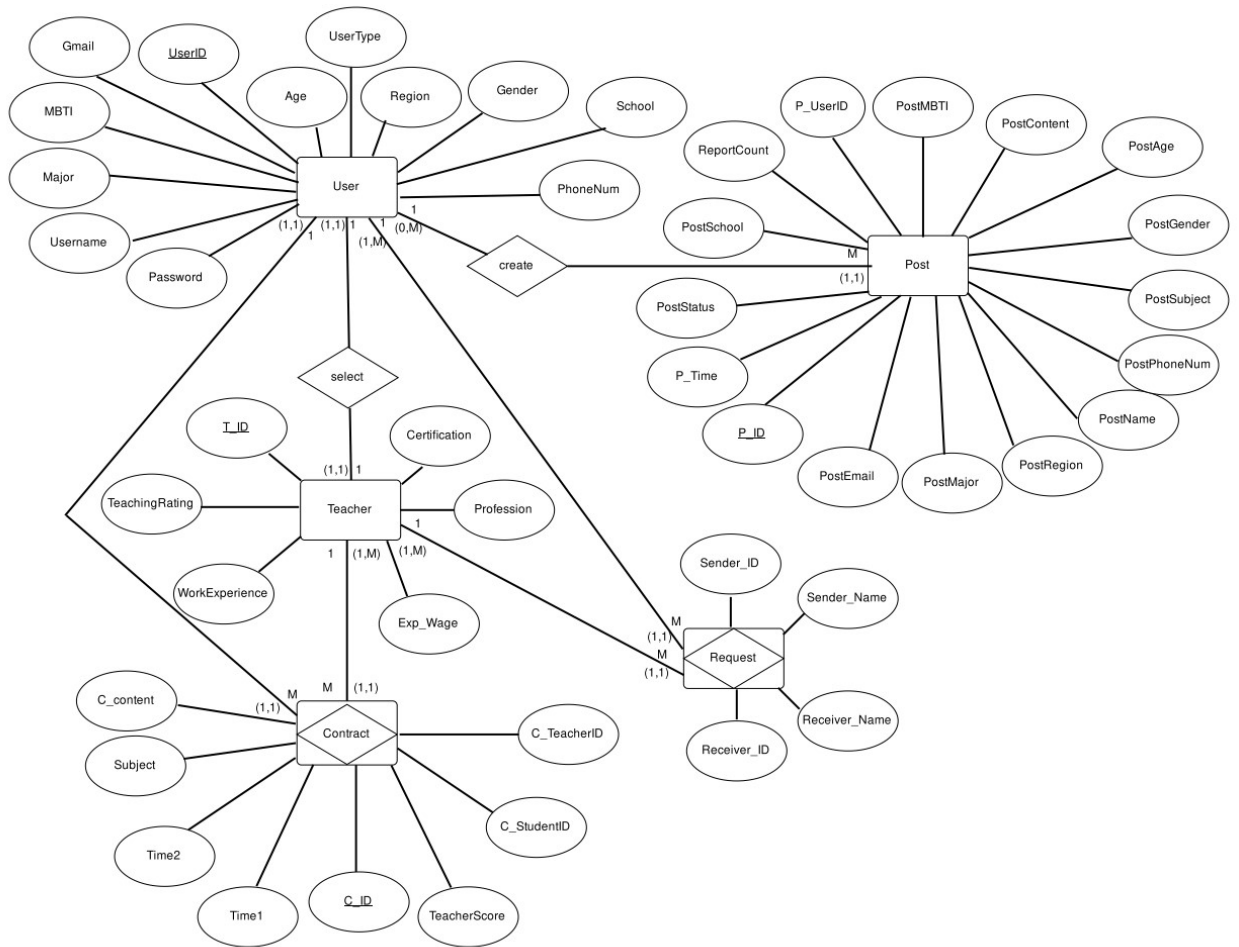
### (四) Preliminary Schedule:



## 參、Logical Design Of Business Transaction:

### (1) Conceptual Schema Design:

以下為我們的ERD圖, 之後會有更多關於attributes的詳細介紹:



## (二) Relational Model:

Tables和Attributes介紹:

**Table User**

<u>Attribute</u>	<u>Datatype</u>	<u>Primary Key</u>	<u>Not Null</u>
UserID	INT	YES	YES
Age	INT		YES
Gender	ENUM		YES

School	VARCHAR(45)		YES
MBTI	VARCHAR(45)		YES
Region	VARCHAR(60)		YES
Major	VARCHAR(45)		YES
PhoneNum	VARCHAR(10)		YES
Gmail	VARCHAR(45)		YES
Password	VARCHAR(45)		YES
Username	VARCHAR(45)		YES
UserType	VARCHAR(45)		YES

**Table Teacher**

TeacherRating為學生對老師契約評分總合平均，顯示老師的平均成績。

<u>Attribute</u>	<u>Datatype</u>	<u>Primary Key</u>	<u>Not Null</u>
T_ID	INT	YES	YES

Profession	VARCHAR(45)		YES
Exp_Wage	INT		YES
WorkExperience	VARCHAR(500)		
Certification	VARCHAR(500)		
TeachingRating	VARCHAR(500)		

Table Request

Sender\_ID和Receiver\_ID為Composite Primary Key (Sender\_ID可對應學生或者老師, Reciever ID亦同)。

<u>Attribute</u>	<u>Datatype</u>	<u>Primary Key</u>	<u>Not Null</u>
Sender_ID	INT	YES	YES
Receiver_ID	INT	YES	YES
Sender_Name	VARCHAR(45)		YES
Receiver_Name	VARCHAR(45)		YES

Table Post

PostSubject對於學生而言是想要學習的科目，對於老師而言為要教授的科目。  
 PostMajor為老師和學生的大學主修。  
 其餘的attributes為發布貼文之人的相關資訊。

Attribute	Datatype	Primary Key	Not Null
P_ID	INT	YES	YES
P_Time	VARCHAR(300)		
P_UserID	INT		
PostContent	VARCHAR(100)		
PostSubject	VARCHAR(45)		
PostRegion	VARCHAR(45)		
PostMajor	VARCHAR(45)		
PostName	VARCHAR(45)		
PostEmail	VARCHAR(45)		
PostPhoneNum	VARCHAR(45)		



PostGender	VARCHAR(45)		
PostAge	VARCHAR(45)		
PostMBTI	VARCHAR(45)		
PostSchool	VARCHAR(45)		

**Table Contract**

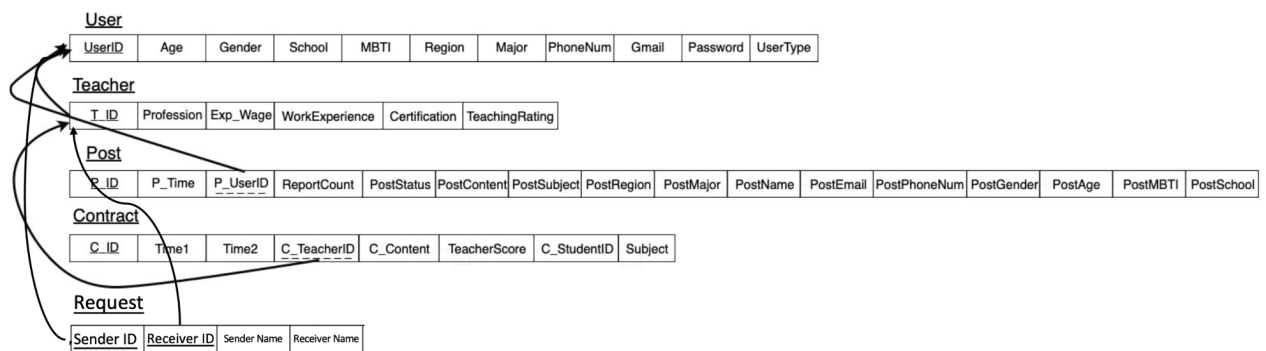
TeacherScore為學生對於老師的評價，由一到五顆星星衡量，學生可在契約上評價。

Attribute	Datatype	Primary Key	Not Null
C_ID	INT	YES	YES
TeacherScore	INT		
C_Content	VARCHAR(100)		
C_TeacherID	INT		YES
Time1	VARCHAR(300)		YES

Time2	VARCHAR(45)		YES
C_StudentID	INT		YES
Subject	VARCHAR(100)		YES

### 1. 3NF 圖：

- User: {UserID, Password, Age, Gender, School, MBTI, Region ,Major, PhoneNum, Gmail, UserType}
- Teacher: {T\_ID, Profession, Exp\_Wage, WorkExperience, Certification, TeachingRating(TeacherScore的平均)}
- Post: {P\_Time, P\_ID, P\_UserID, PostContent, PostSubject, PostRegion, PostMajor, PostName, PostEmail, PostPhoneNum, PostGender, PostAge, PostMBTI, PostSchool}
- Contract:{ C\_ID, TeacherScore, C\_Content, C\_TeacherID, C\_StudentID, Subject, Time1, Time2}
- Request:{Sender\_ID, Receiver\_ID, Sender\_Name, Receiver\_Name}



### (三) Interaction with the Database:

以下的所有功能的完整程式碼主要都在controller和UserService兩個兩個document中定義。以下只列出會對資料庫進行增刪查改的SQL指令：

1. 使用者登入功能(在程式碼UserService中的loginUser method)：

=> SQL指令:"SELECT \* FROM TeachMate.User WHERE Gmail = ? AND Password = ?"

=>系統會利用user.getEmail()和user.getPassword()取得要搜尋Gmail和Password的值，並且傳入SQL指令中。SQL指令會取得對應User的所有attributes。

2. 分辨使用者類型，是Teacher或是Student(在UserService中的checkUser method)：

=>SQL指令:"SELECT UserType FROM Teachmate.User WHERE UserID = ?"

=>系統會利用user.getUserID()取得要搜尋UserID的值，並且傳入SQL指令中。SQL指令會取得對應User的類別，分成Teacher和Student。

3. 使用者註冊功能(在程式碼UserService中的saveUser method)：

=> SQL指令：

1. "INSERT INTO User (UserID, Username, Password, Age, Gender, School, Region, Major, PhoneNum, Gmail, MBTI, UserType) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);"

2. "INSERT INTO Teacher (T\_ID) VALUES (?)"

=> 第一條SQL指令系統會取得User的所有attributes data並將其存入User Table中，其中Student的所有attributes都跟User相同，所以並沒有多設置一個Student Table。

=> 第二條SQL指令系統會透過User取得Teacher的T\_ID attribute data並將其存入Teacher Table中。

4. 使檢查UserID是否有存在User的Table中(在程式碼UserService中的isUserIDExists method)：

=> SQL指令:"SELECT COUNT(\*) FROM User WHERE UserID = ?"

=> SQL會根據傳入的UserID去搜尋資料庫中，如果回傳的count>0，代表有找到UserID，回傳True。此段method被應用在RegisterController中的generateUserID當中。使用者每註冊一個新帳號時，都會生成一個新的UserID，之後會用isUserIDExists去檢查該UserID是否已經存在於資料庫中，避免duplicate value的出現。

5. 老師建立契約(在程式碼UserService中的buildContract method)：

=> SQL指令:"INSERT INTO Contract (C\_ID, C\_TeacherID, Time1, Time2, C\_StudentID, Subject) VALUES (?, ?, ?, ?, ?, ?)"

=> SQL會根據參數Contract object去取得對應的attributes data，並且加入

Contract Table當中，完成建立契約的動作。

6. 學生契約評分留言功能(在程式碼UserService中的ContractGradeComment method)：

=>SQL指令:"INSERT INTO Contract (C\_ID, TeacherScore, C\_Content, C\_TeacherID, Time1, Time2, C\_StudentID, Subject) VALUES (?, ?, ?, ?, ?, ?, ?, ?)"

=> SQL會交將學生的星星評分(1\_5顆星)和學生對老師的評論留言分別插入attributes TeacherScore和C-Content, 另外, Contract的一些基本attributes也會被插入資料庫

7. 老師貼文建立(在程式碼UserService中的buildTeacherPost method)：

=> SQL指令：

1. "SELECT Username, Gmail, PhoneNum, Gender, Age, MBTI, School FROM User WHERE UserID=" + temp\_TID;
2. "INSERT INTO Post (P\_ID, P\_UserID, PostContent, PostSubject, PostRegion, PostMajor, PostName, PostEmail, PostPhoneNum, PostGender, PostAge, PostMBTI, PostSchool, P\_time) VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?)"

=> 第一條SQL指令系統先根據老師的ID (T\_ID, temporary ID 的意思) 取得老師的相關資訊，一些相關的attributes。

=> 第二條SQL會建立貼文，將貼文所需的attributes放入資料庫Post Table中。其中P\_UserID為T\_ID, PostSubject為老師想要教授的科目，PostMajor為老師的主修，其他的attributes為根據T\_ID取得的資訊。老師每建立一個新貼文時，都會生成一個新的PostID(定義在BuildTeacherPostController當中的generatePostID()中)，之後會用isPostIDExists去檢查該P\_ID是否已經存在於資料庫中(定義在UserService的isPostIDExists中)，避免duplicate value的出現。

8. 學生貼文建立(在程式碼UserService中的buildStudentPost method)：

=> SQL指令：

1. "SELECT Username, Gmail, PhoneNum, Gender, Age, MBTI, School FROM User WHERE UserID=" + temp\_TID;
2. "INSERT INTO Post (P\_ID, P\_UserID, PostContent, PostSubject, PostRegion, PostMajor, PostName, PostEmail, PostPhoneNum, PostGender, PostAge, PostMBTI, PostSchool, P\_time) VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?)"

=> 第一條SQL指令系統先根據學生的ID (T\_ID, temporary ID 的意思) 取得學生的相關資訊, 一些相關的attributes。

=> 第二條SQL會建立貼文, 將貼文所需的attributes放入資料庫Post Table中。其中P\_UserID為T\_ID, PostSubject為學生想要找/學習的科目, PostMajor為學生的主修, 其他的attributes為根據T\_ID取得的資訊。學生每建立一個新貼文時, 都會生成一個新的PostID (此過程定義在BuildStudentPostController當中generatePostID()), 之後會用isPostIDExists去檢查該P\_ID是否已經存在於資料庫中 (此過程定義在UserService的isPostIDExists中), 避免duplicate value的出現。

9. 檢查PostID是否有存在Post的Table中 (在程式碼UserService中的isPostIDExists method) :

=> SQL指令: "SELECT COUNT(\*) FROM Post WHERE P\_ID = ?"

=> SQL會根據傳入的PostID去搜尋資料庫中, 如果回傳的count>0, 代表有找到PostID, 回傳True。此段method被應用在BuildTeacherPostController和BuildStudentPostController中的generatePostID當中。使用者每創一個新貼文時, 都會生成一個新的PostID, 之後會用isPostIDExists去檢查該PostID是否已經存在於資料庫Post的Table中, 避免duplicate value的出現。

10. 根據使用者的gmail取得使用者名稱(name) (在程式碼UserService中的getUserNameByEmail method) :

=> SQL指令: "SELECT Username FROM Teachmate.User WHERE Gmail = ?"

=> 由於每位使用者的gmail為唯一的, 使用者在登入的時候會輸入gmail和name。登入時系統會執行LogInController中getUserAccountInfo的method, 該method中有定義getUserNameByEmail, 會根據使用者輸入的gmail取得使用者名稱。

11. 根據使用者的gmail取得使用者的類別(UserType) (在程式碼UserService中的getUserTypeByEmail method) :

=> SQL指令: "SELECT UserType FROM Teachmate.User WHERE Gmail = ?"

=> 由於每位使用者的gmail為唯一的, 使用者在登入的時候會輸入gmail和name。登入時系統會執行LogInController中getUserAccountInfo的method, 該method中有定義getUserTypeByEmail, 會根據使用者輸入的gmail取得使用者的類別 (分成學生和老師)。

12. 取得學生資訊 (在程式碼UserService中的getStudentInfo method) :

=> SQL指令:"SELECT UserID, Region, UserType, Gender, Age, School, Major, MBTI, Gmail, PhoneNum FROM Teachmate.User WHERE Username = ?"

=>此SQL會根據學生的名稱去資料庫User的Table中取得學生資訊，該POST定義在前端的一些頁面中，用axios傳送POST。

13. 取得老師資訊(在程式碼UserService中的getTeacherInfo method) :

=>SQL指令:"SELECT u.UserID, u.Region, u.UserType, u.Gender, u.Age, u.School, u.Major, u.MBTI, u.Gmail, u.PhoneNum, t.Profession, t.Exp\_Wage, t.WorkExperience, t.Certification, t.TeachingRating FROM Teachmate.User as u INNER JOIN Teachmate.Teacher as t ON u.UserID = t.T\_ID WHERE u.Username = ?"

=>此SQL會根據老師的名稱去資料庫User和Teacher的兩個Table中取得老師資訊。會用INNER JOIN 比對T\_ID和User\_ID兩者聯繫兩個Tables。

14. 老師輸入或更改Profile中的Profession欄位(在程式碼UserService中的teacherinputprofession method) :

=> SQL指令:"UPDATE Teacher SET Profession = ? WHERE T\_ID = ?"

=>此SQL會根據老師的T\_ID在資料庫Teacher的Table中找到相對應的Profession欄位，並且更新該欄位的值。該method會出現在TeacherinputProfessionController的document中。

15. 老師輸入或更改Profile中的WorkingExperience欄位(在程式碼UserService中的teacherinputworkingexperiencemethod) :

=> SQL指令:"UPDATE Teacher SET WorkExperience = ? WHERE T\_ID = ?"

=>此SQL會根據老師的T\_ID在資料庫Teacher的Table中找到相對應的WorkExperience欄位，並且更新該欄位的值。該method會出現在TeacherinputWorkExperienceController的document中。

16. 學生發送Request給老師(在程式碼UserService中的saveStudentRequestInfo method) :

=> SQL指令:

sql1 = "SELECT UserID FROM User WHERE Username=?"

sql2 = "SELECT UserID FROM User WHERE Username=?"

sql="INSERT INTO Request (Sender\_ID,Receiver\_ID, Sender\_Name, Receiver\_Name) VALUES(?,?,?,?)"

=>當學生看到契約發送Request給老師後，該SQL會根據學生的名稱和老

師的名稱去User的Table中尋找相對應的UserID。接著再insert進去資料庫Request Table中Sender\_ID (在此是Student的ID), Receiver\_ID(在此是Teacher的ID), Sender\_Name(在此是Student的Name)和Receiver\_Name(在此是Teacher的Name)。

17. 老師發送Request給學生(在程式碼UserService中的saveTeacherRequestInfo method):

=> SQL指令:

1. sql1 = "SELECT UserID FROM User WHERE Username=?"
2. sql2 = "SELECT UserID FROM User WHERE Username=?"
3. sql="INSERT INTO Request (Sender\_ID,Receiver\_ID, Sender\_Name, Receiver\_Name) VALUES(?,?,?,?)"

=>當老師看到契約發送Request給學生後, 該SQL會根據老師的名稱和學生的名稱去User的Table中尋找相對應的UserID。接著再insert進去資料庫Request Table中Sender\_ID (在此是Teacher的ID), Receiver\_ID(在此是Student的ID), Sender\_Name(在此是Teacher的Name)和Receiver\_Name(在此是Student的Name)。

18. 取得Post的Data詳細資訊(在程式碼UserService中的 getPostsData method):

=> SQL指令:"select PostSubject, PostName, PostEmail, PostPhoneNum, PostGender, PostAge, PostMBTI, PostRegion, PostSchool, PostMajor, PostContent from Teachmate.Post"

=>該指令會從資料庫取回所有的Post相關資料, 並將其存入HashMap的物件中, 利用鍵值對對應相關的data。

#### (四) Constraints and Functionality:

Log In/Sign Up Page	Function	Constraints
Log In Button	Log in the user's account	Email and password should exist in <a href="#">database</a>
Sign Up Button	Sign up new user <a href="#">account</a>	Each attribute must meet the domain requirement in <a href="#">database</a>

Post Page	Function	Constraints
Filter Button	Filter posts by multiple conditions	The options in each attribute are provided by app administrator, users can't create new <a href="#">options</a>
Request Button	Send request notification to teacher/student through <a href="#">email</a>	N/A

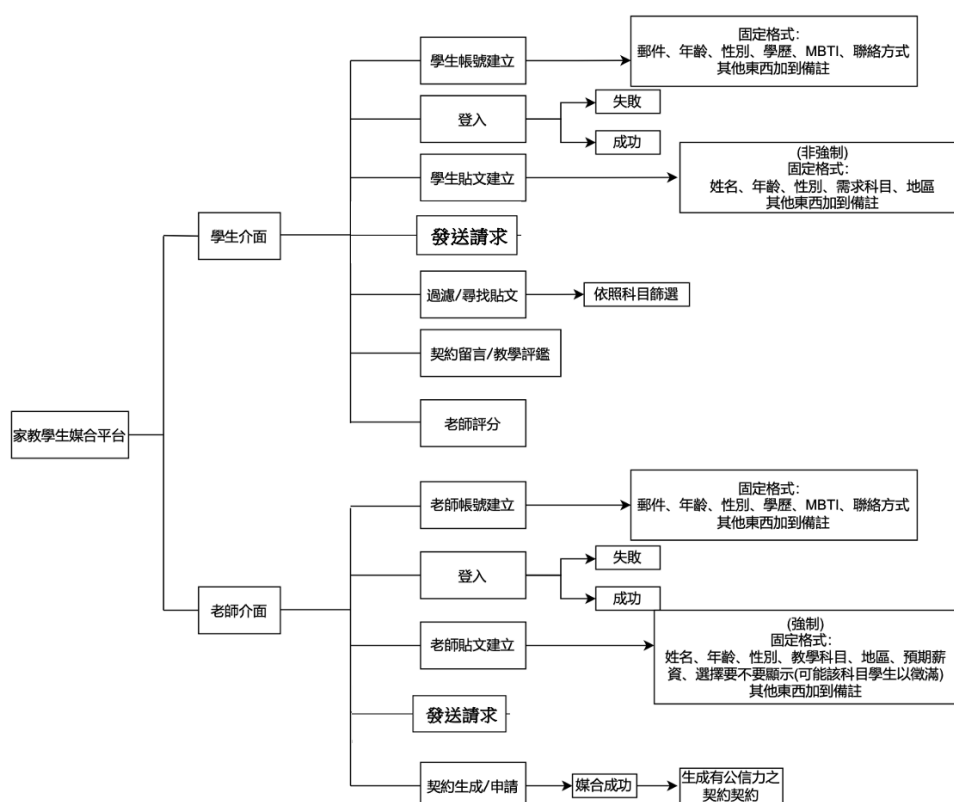
Create Post Page	Function	Constraints
Add Button	Create new post with specified subject and sidenotes (there will be an addition attribute called hourly expected wage for teacher's version)	Subject and hourly expected wage text field is required to be <a href="#">filled</a>

Evaluation Sheet Page	Function	Constraints
Stars rating	Rate the teacher's <a href="#">performance</a>	From 1 to 5 stars
Be A Teacher Button	Toggle to Log In/Sign Up Page for more <a href="#">actions</a>	N/A
My Evaluation Button	Get the info about your student's evaluation about you or your evaluation about your <a href="#">teacher</a>	N/A
確認評論 Button	Review your evaluation about the <a href="#">teacher</a>	N/A
Next Button	Head to payment window	N/A
完成繳費 Button	Pop up a dialog for users to type in the last 5 digits of their bank account for future <a href="#">reference</a>	You have to pay the fee beforehand to use this <a href="#">button</a>

## 肆、Implementation On Plans:



(一)FDD圖：



以下將以老師和學生的視角出發，分別敘述其使用系統時使用到的功能：

(二)老師：

1. 註冊功能:使用者在註冊時能選擇自己的身份為老師還是學生，選擇的身份為UserTable中的UserType。老師註冊時會輸入name、email、password、age、gender、usertype、school、region、major、phoneNumber、mbti，系統也會自動生成一個UserID給老師。詳細的功能定義在後端RegisterController和前端SigninPage中。
2. 登入功能:老師註冊過後能登入系統。老師登入系統只需輸入Email和Password即可。詳細的功能定義在後端LoginController和前端SigninPage中。
3. 檢視Teacher Profile內容:老師能在前端的TeacherProfile頁面中查看自己profile的相關資訊，後端則會用到UserService中的getTeacherInfo()方法取得資料庫中的資訊呈現在頁面上。
- 4.修改Teacher Profile 的 Profession和WorkExperience:老師能在自己的Profile中手動修改更新自己的Profession和WorkExperience。詳細的功能定義在後端TeacherInputProfessionController、TeacherInputWorkExperience和前端TeacherProfile中。

5.檢視自己的發布過的Post:在前端TeacherProfilePost中老師能檢視自己發布過的貼文。後端的取得貼文定義在GetPostController當中。

6.老師發布Post:老師能建立自己的貼文, 詳細的功能定義在後端的BuildTeacherPostController和前端的TeacherAddPost當中。老師在buildPost時會被要求輸入Subject、Major、Region和Note這四項attributes。其中Subject為希望教授的科目, Major為自己就學時期的主修, Region為希望授課的地點和Note為想要在Post上貼文的文字內容。其餘老師的詳細個人資料會被取得然後秀在Post上。

7.老師Filter學生貼文功能:老師能在前端TeacherPostPage中閱覽所有學生發布過的貼文, 並且根據最上方的filter的四項欄位標準對學生的Posts進行過濾。最上方的filter選相包含四項:subject、gender、MBTI和region。後端對filter功能的程式撰寫是在TeacherPostPage.vue的filterPosts() 方法中定義。

8.老師發送Request給學生:老師在filter過後閱覽學生貼文時, 若對某學生的貼文有興趣, 想要取得進一步聯繫時會按每篇文下方都有的REQUEST按鈕。按按鈕後, 學生會取得老師發送的Request, 當下畫面中也會顯示對話框: “Your request has been sent to student \${studentName} through gmail !”。前端的部分仍是在TeacherPostPage當中, 後端的部分則會定義在GetTeacherRequestController當中, 將老師發送的Request存入資料庫的Request Table當中作為紀錄。

9. 老師建立契約功能:老師發送Request後如果學生同意, 他們會私底下約出來授課。之後老師會建立契約, 該契約為一個證明, 證明老師和學生有一起授課過, 學生之後也能到老師的契約下留言對老師評價(用星星打分數)和留言, 能作為日後其他學生選擇老師的依據。老師在建立契約頁面中會輸入授課開始時間(time1)和結束時間(time2), 教授學生的id和教授的科目。前端的頁面在TeacherSheetBuild, 後端的功能定義在BuildContractController當中。老師建立完契約後, 該契約的詳細資訊會被輸入資料庫的Contract Table當中。

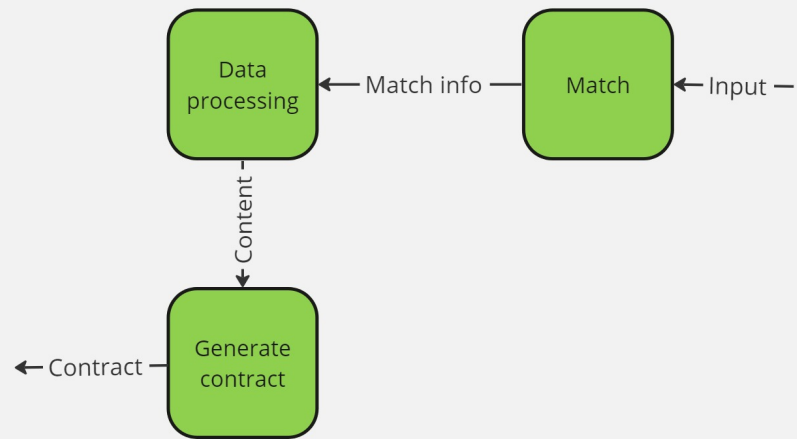
### (三)學生:

1. 註冊功能:使用者在註冊時能選擇自己的身份為老師還是學生, 選擇的身份為UserTable中的UserType。學生註冊時會輸入name、email、password、age、gender、usertype、school、region、major、phoneNumber、mbti, 系統也會自動生成一個UserID給學生。詳細的功能定義在後端RegisterController和前端SigninPage中。

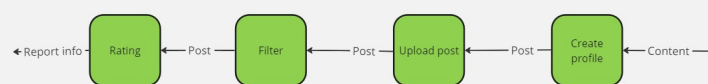
2. 登入功能:學生註冊過後能登入系統。學生登入系統只需輸入Email和Password即可。詳細的功能定義在後端LoginController和前端SigninPage中。
3. 檢視Student Profile內容:學生能在前端的StudentProfile頁面中查看自己profile的相關資訊, 後端則會用到UserService中的getStudentInfo()方法取得資料庫中的學生資訊呈現在頁面上。
4. 檢視自己的發布過的Post:在前端StudentProfilePost中老師能檢視自己發布過的貼文。後端的取得貼文定義在GetPostController當中。
6. 學生發布Post:老師能建立自己的貼文, 詳細的功能定義在後端的BuildStudentPostController和前端的StudentAddPost當中。學生在buildPost時會被要求輸入Subject、Major、Region和Note這四項attributes。其中Subject為希望學習和想要尋找家教的科目, Major為自己就學的主修, Region為希望授課的地點和Note為想要在Post上貼文的文字內容。其餘學生的詳細個人資訊會被取得然後秀在Post上。
7. 學生Filter老師貼文功能:學生能在前端StudentPostPage中閱覽所有老師發布過的貼文, 並且根據最上方的filter的四項欄位標準對老師的Posts進行過濾。最上方的filter選相包含四項: subject、gender、MBTI和region。後端對filter功能的程式撰寫是在StudentPostPage.vue的filterPosts() 方法中定義。
8. 學生發送Request給老師:學生在filter過後閱覽老師貼文時, 若對某老師的貼文有興趣, 想要取得進一步聯繫時會按每篇文下方都有的REQUEST按鈕。按按鈕後, 老師會取得學生發送的Request, 當下畫面中也會顯示對話框: “Your request has been sent to teacher \${teacherName} through gmail !”。前端的部分仍是在StudentPostPage當中, 後端的部分則會定義在GetStudentRequestController當中, 將學生發送的Request存入資料庫的Request Table當中作為紀錄。
9. 學生契約評分和留言功能:老師和學生上完課後, 老師會先建立契約, 該份契約代表某學生和某老師有一同上課的證據。之後學生會取得老師已建立好的契約, 並且在前端的StudentSheetComment的頁面中為老師的授課進行評價。評價的部分分成兩種。第一種為用星星評價, 可以為老師打1~5顆星星的評價。第二種為可以直接在老師的契約上留comment的功能。後端的程式碼定義在ContractGradeCommentController當中。

(四)DFD圖:

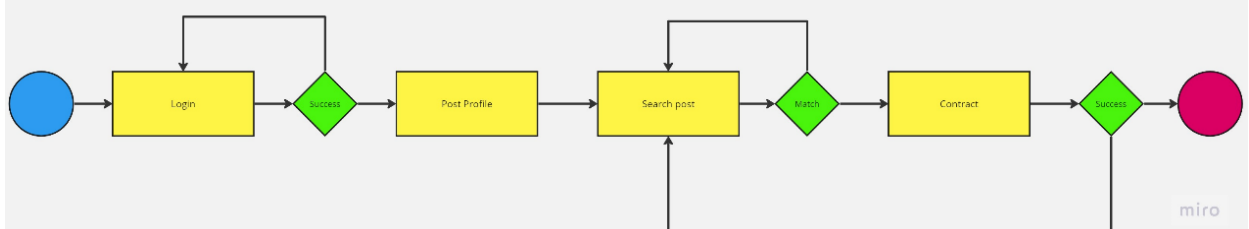




miro



miro



miro

## 伍、Appendices：

1. Github 連結：<https://github.com/lawrence0510/Teachmate>
2. 影片連結：  
demo影片：  
<https://www.youtube.com/watch?v=VV0PxIV5V-Y>  
新增Request功能的說明影片：  
<https://www.youtube.com/watch?v=P65CF0hpTCk>
3. 參考資料：<https://router.vuejs.org/guide>