

系統分析與實作

系統軟體設計規格書

第 15 組

吳亦鴻 周炯辰 鄧同恩 程子芸 李奕蓁

指導教授：許智誠 教授

目錄

第 2 章 資料庫設計

第 3 章 類別圖

第 4 章 系統循序圖

4.1 使用案例圖

4.2 Use Case 實做之循序圖

4.2.1.1 Sequence Diagram—Use Case

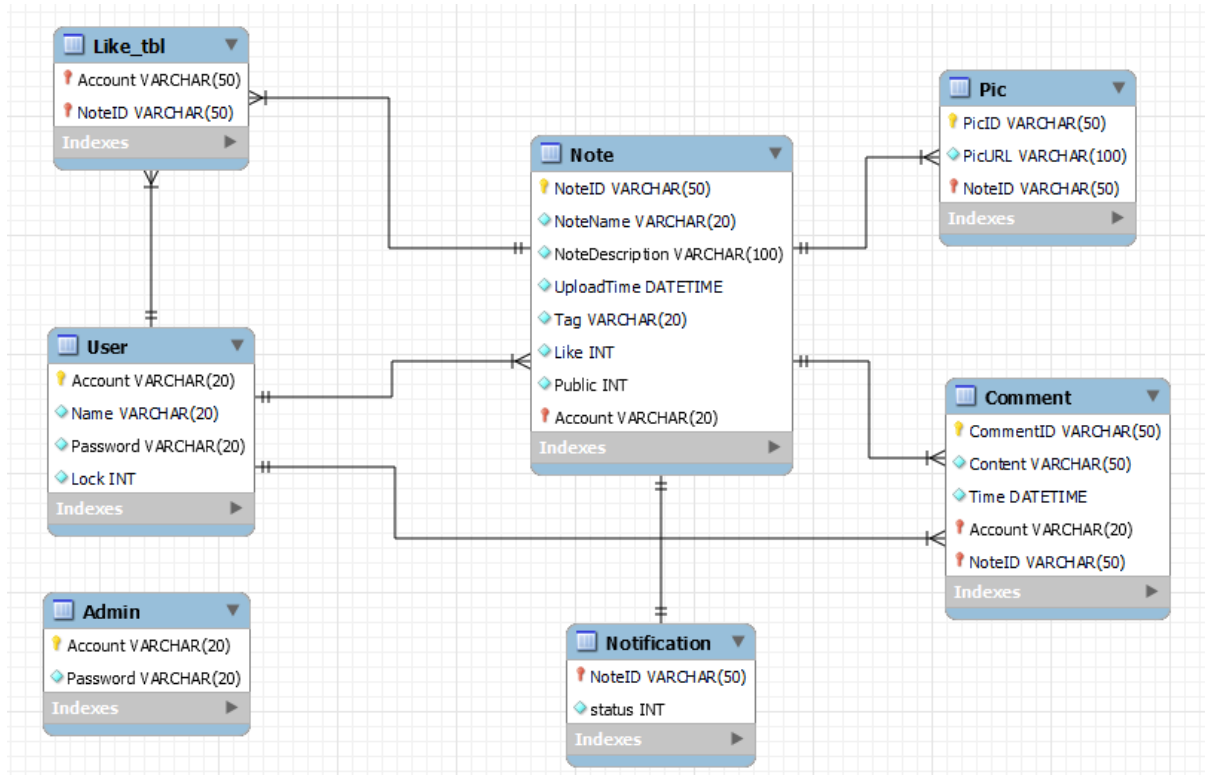
2.1 上傳筆記

4.2.1.2 Sequence Diagram—Use Case

3.3 查看筆記

第 2 章 資料庫設計

設計階段之資料庫，根據分析文件之實體關係圖 (Entity-Relation Diagram)，進行確認並依據其規劃資料庫之資料表，共計包含 6 個實體 (Entity)、5 個關係 (Relationship)、1 個複合性實體 (Compound Entity)。



用戶

Key	名稱	類型	預設值	空值	自動增加	編碼
P. K	Account	Varchar(20)	無	否		
	Name	Varchar(20)	無	否		
	Password	Varchar(20)	無	否		
	Lock	int	無	否		

- Account：用戶帳號，不可更動且不能重複。
- Name：用於紀錄用戶名稱。

- Password：用戶用於登入系統
- Lock：判定是否被封鎖，如被封鎖用戶將無法登入(0=沒被鎖，1=有被鎖)

管理員

Key	名稱	類型	預設值	空值	自動增加	編碼
P.K	Account	Varchar(20)	無	否		
	Password	Varchar(20)	無	否		

- Account：管理員帳號，不可更動且不能重複。
- Password：管理員用於登入系統

筆記

Key	名稱	類型	預設值	空值	自動增加	編碼
P.K	NoteID	Varchar(50)	無	否		
F.K	Account	Varchar(20)	無	否		
	NoteName	Varchar(20)	無	否		
	NoteDescription	Varchar(100)	無	否		
	Tag	Varchar(20)	無	否		
	UploadTime	Datetime	無	否		
	Like	int	無	否		
	Public	int	無	否		

- NoteID (筆記 ID): 主鍵，唯一識別一筆筆記的字符串，由資料庫系統自動產生。
- Account (用戶): 外鍵，用於判斷是誰的筆記。
- NoteName (筆記名稱): 筆記的名稱。
- NoteDescription (筆記描述): 筆記的詳細描述。
- Tag (標籤): 筆記的標籤，有助於分類和搜索。
- UploadTime (上傳時間): 紀錄筆記上傳的時間。
- Like (喜歡): 整數計數，表示有多少用戶喜歡這篇筆記。
- Public (公開): 整數標誌，指示筆記是否對所有用戶公開 (0=沒公開，1=有公開)

留言

Key	名稱	類型	預設值	空值	自動增加	編碼
P K	CommentID	Varchar(50)	無	否		
F K	NoteID	Varchar(50)	無	否		
F K	Account	Varchar(20)	無	否		
	Content	Varchar(50)	無	否		
	Time	Datetime	無	否		

- CommentID：評論的唯一識別碼，為主鍵，由資料庫系統自動產生。
- NoteID：關聯到特定筆記的識別碼，為外鍵。
- Account：發表評論的用戶賬號，也是外鍵。
- Content：評論的內容。
- Time：評論發表的時間。

通知

Key	名稱	類型	預設值	空值	自動增加	編碼
F K	NoteID	Varchar(50)	無	否		
	Status	int	無	否		

- NoteID：關聯到特定筆記的識別碼，為外鍵。
- Status：辨識該通知是否被閱讀過（0=未讀，1=已讀）

愛心

Key	名稱	類型	預設值	空值	自動增加	編碼
F K	Account	Varchar(20)	無	否		
F K	NoteID	Varchar(50)	無	否		

- NoteID：關聯到特定筆記的識別碼，為外鍵。
- Account：按讚者的用戶賬號，也是外鍵。

照片

Key	名稱	類型	預設值	空值	自動增加	編碼
-----	----	----	-----	----	------	----

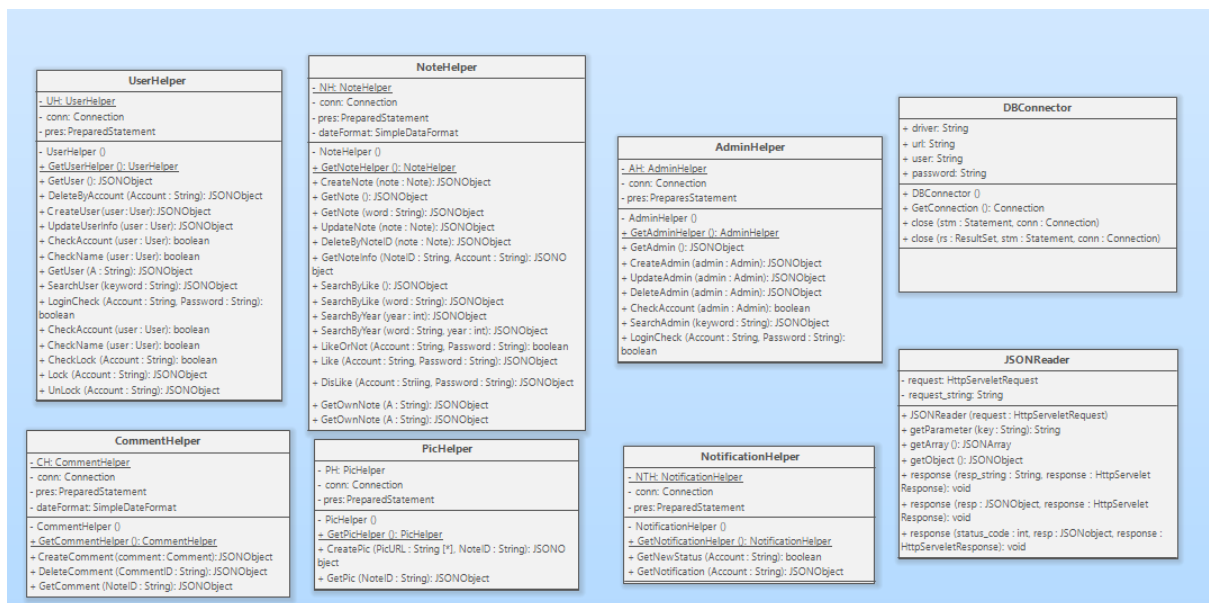
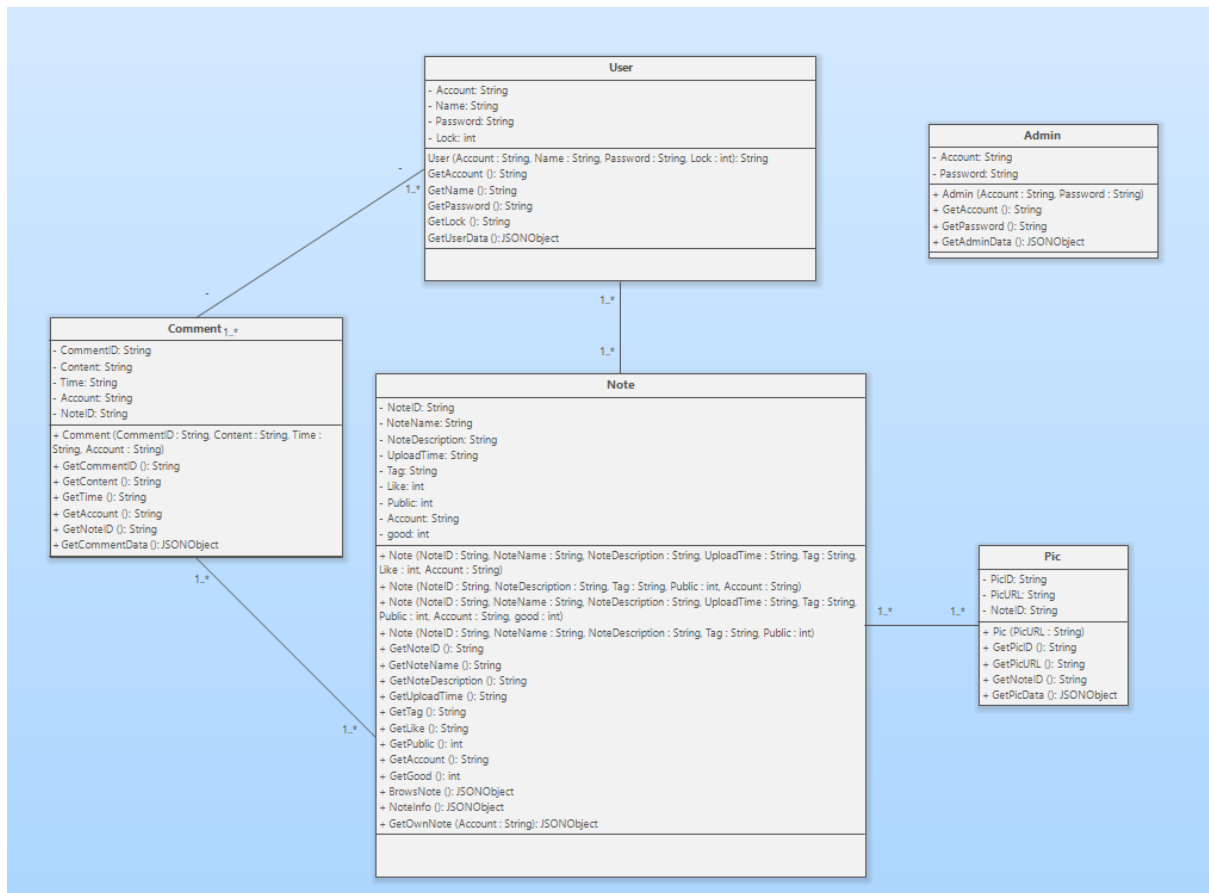
P. K	PicID	Varchar(50)	無	否	
	PicURL	Varchar(100)	無	否	
F K	NoteID	Varchar(50)	無	否	

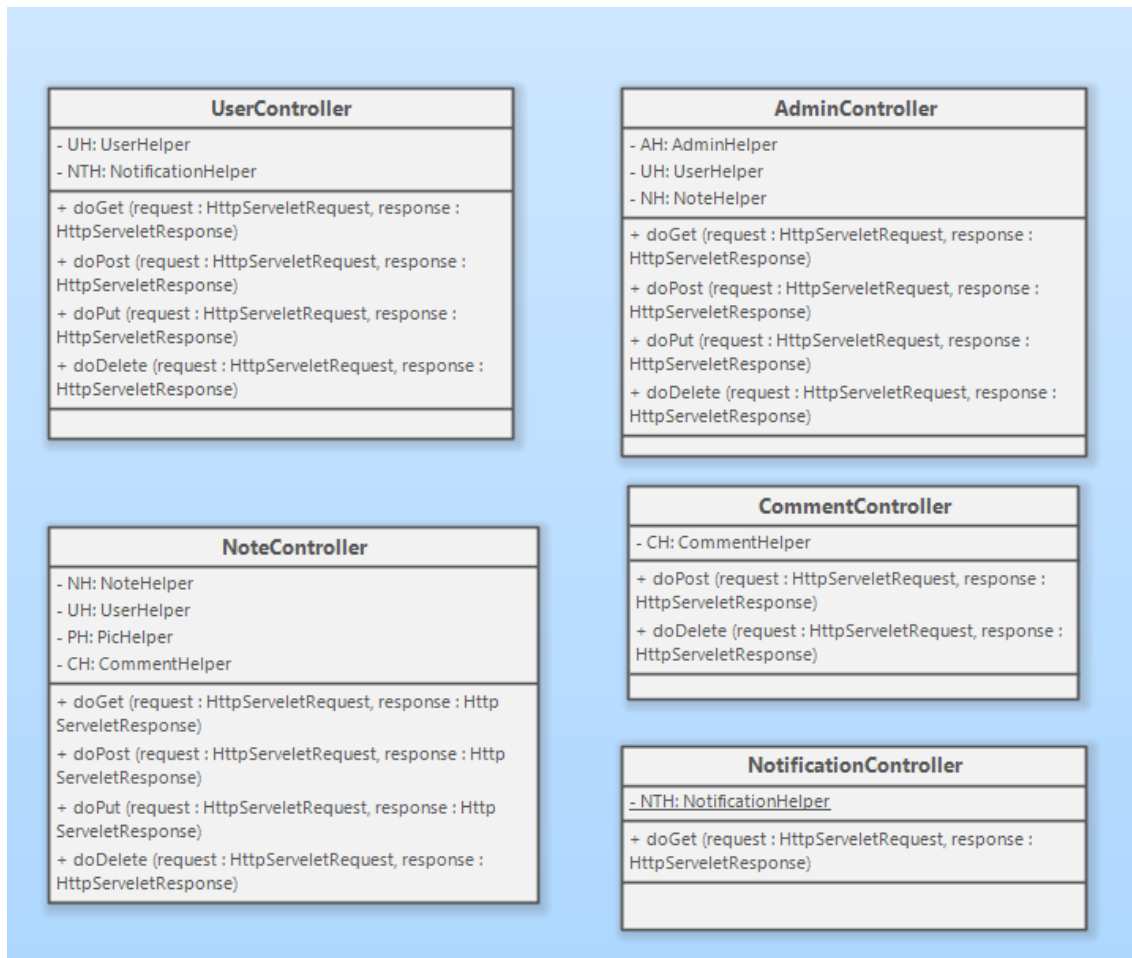
- PicID：圖片的唯一識別碼，為主鍵，由資料庫系統自動產生。
- PicURL：圖片的路徑。
- NoteID：關聯到特定筆記的識別碼，作為外鍵。

第 3 章 類別圖

下圖資管筆記平台的分析模型和建立的互動圖，以及實體關係圖（**Entity-Relation Diagram**）所繪製之設計階段之類別圖（**Class Diagram**），用於描述系統的類別集合，包含其中之屬性，與類別之間的關係。

本階段之類別圖屬於細部（detail）之設計圖，與上一份文件分析階段之類別圖需要有詳細之變數型態、所擁有之方法，依據這些設計原則，本類別圖之說明如下所列：類別圖除包含與資料庫相對應之物件外，亦包含相關之控制物件（controller）、DBMgr 與各功能相對應資料庫操作類別和相對應之類別工具（JsonReader）。





第 4 章 系統循序圖

本章節主要依照第一份文件需求所產生之使用案例圖（**use case**）與第二份文件分析之邏輯階段活動圖與強韌圖為基礎，進行設計階段之循序圖設計，將每個使用案例進行闡述。於此階段，需要有明確之類別（**class**）名稱與呼叫之方法（**method**）與傳入之變數名稱與型態等細部設計之內容。

4.1 使用案例圖

依據第一份文件針對專案之需求進行確定，本電子商務線上訂購系統預計共有 2 位動作者與 23 個使用案例，並依照不同之模組區分成不同子系統共計六個模組，其中包含以下：1.用戶管理模組、2.筆記管理模組、3.筆記資訊模組、4.留言管理模組、5.管理員模組、6.通知模組，如下圖所示



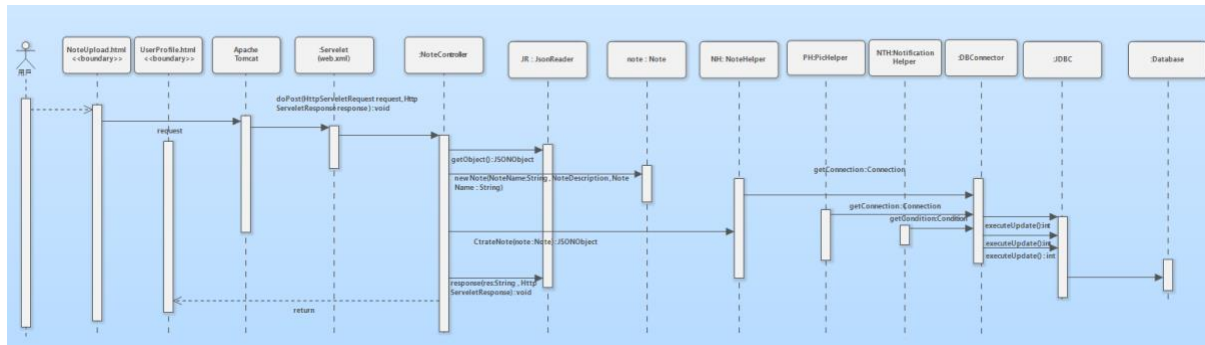
4.2 Use Case 實做之循序圖

本次作業以 Use Case 2.1 上傳筆記、Use Case 3.3 查看筆記來製作，下表為 user case 關聯頁面

HTML	關聯 Use Case	說明
------	-------------	----

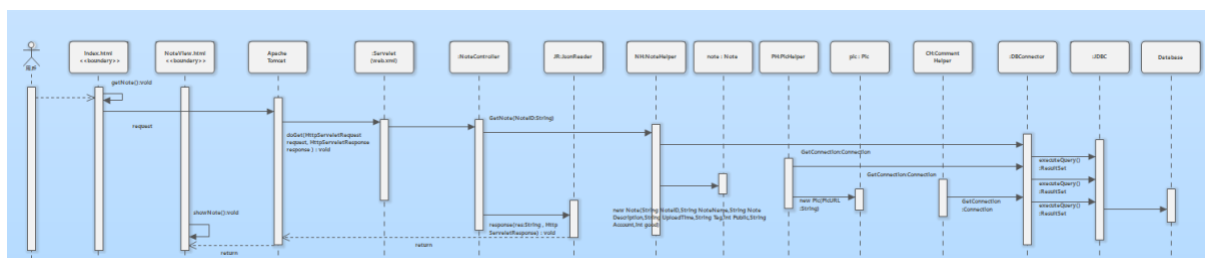
NoteUpload.html	Use Case 2.1	筆記上傳頁面
UserProfile.html	Use Case 2.1	用戶個人頁面
index.html	Use Case 3.3	平台首頁
NoteView.html	Use Case 3.3	筆記內容頁面

4.2.1.1 Sequence Diagram—Use Case 2.1 上傳筆記



1. 使用者在 **NoteUpload.html** 界面發起上傳筆記的請求。
2. 請求通過 **Apache Tomcat** 服務器傳遞。
3. **Servlet** 接收到 **HTTP POST** 請求並調用 **NoteController** 的 **doPost** 方法。
4. **NoteController** 使用 **JsonReader** 解析出請求中的 **JSON** 物件。
5. 建立一個新的 **Note** 物件並將其屬性設置為 **JSON** 物件中解析的值。
6. 透過 **NH: NoteHelper** 的 **CreateNote** 方法處理 **Note** 物件，這個方法負責將筆記數據保存到數據庫。
7. **NoteHelper** 透過 **DBConnector** 獲取數據庫連接。
8. 透過 **PH: PicHelper** 的 **CreatePic** 方法將圖片連結傳進資料庫。
9. **PicHelper** 透過 **DBConnector** 獲取數據庫連接。
10. 使用 **JDBC** 執行數據庫更新操作。
11. 數據庫操作完成後，**NoteController** 將操作結果封裝為 **JSON** 物件並回傳至客戶端。
12. 客戶端的 **UserProfile.html** 界面接收到響應並進行更新。

4.2.1.2 Sequence Diagram—Use Case 3.3 查看筆記



1. 使用者透過界面 (`index.html`) 發起獲取筆記的請求 (`getNote()`) 。
2. 請求被送到伺服器端的 **Apache Tomcat**。
3. **Tomcat** 通過配置的 **Servlet** 接收到請求，調用 **NoteController** 的 `doGet()` 方法。
4. **NoteController** 使用 **JRJsonReader** 從請求中解析出 **NoteID**。
5. **NoteController** 呼叫 **NH:NoteHelper** 的 `GetNote()`方法，傳入 **NoteID**。
6. **NHNoteHelper** 透過 **DBConnector** 使用 **JDBC** 技術與數據庫建立連接。
7. 執行對應的 **SQL** 查詢 (`executeQuery()`) ，檢索 **NoteID** 對應的筆記資料。
8. **NoteController** 呼叫 **PH:PicHelper** 的 `GetPic()`方法，傳入 **PicURL**。
9. **PH:PicHelper** 透過 **DBConnector** 使用 **JDBC** 技術與數據庫建立連接。
10. 執行對應的 **SQL** 查詢 (`executeQuery()`) ，檢索 **NoteID** 對應的圖片 **URL**。
11. **NoteController** 呼叫 **CH:CommentHelper** 的 `GetPic()`方法，傳入該筆記的留言。
12. **CH:CommentHelper** 透過 **DBConnector** 使用 **JDBC** 技術與數據庫建立連接。
13. 執行對應的 **SQL** 查詢 (`executeQuery()`) ，檢索 **NoteID** 對應的留言資料。
14. 查詢結果傳回給 **NoteController**。
15. **NoteController** 將結果封裝進 **JSONObject** 並通過 **ServletResponse** 返回給客戶端。
16. 客戶端的 **NoteView.html** 頁面接收到數據後更新，顯示筆記內容。