

研究所網路申請入學系統

Web-based Graduate Admission Application System

研究所網路申請入學系統軟體需求規格

Version 1.2

開發小組：第五小組

組員：1021421 張弘瑜 1021440 郭柏宏 1021752 吳冠嫻 1023336 鍾羽函

目錄

一、簡介.....	3
1.1 一般描述資訊.....	3
二、規格書內容描述.....	3
2.1 系統概述.....	3
2.1.1 系統目標.....	3
2.1.2 系統範圍.....	3
2.2.3 系統架構.....	4
2.2 軟體架構.....	4
2.3 軟體類別定義.....	6
2.4 軟體類別描述.....	7
2.5 軟體行為概述.....	16
三、軟體設計描述.....	21
3.1 設計概述.....	21
3.2 軟體類別設計描述.....	21
3.3 執行概念.....	58
3.4 需求追溯.....	59
四、軟體介面設計描述.....	59
4.1 介面設計概述.....	59
4.2 外部介面定義.....	59
4.3 函式介面定義表.....	61
4.4 類別介面定義.....	65

一、簡介

1.1 一般描述資訊

a) 發布日期和狀態

此版於2016/05/03發佈正式版本(version 1.0)

b) 範圍

本規格書的內容以研究所網路申請入學系統之操作介面等為主。

c) 發布組織

YZU_CSE

d) 參考

IEEE/EIA 12207.1-1997

e) 內容

本申請入學系統之標準規格以及介面設計。

f) 主體

包含系統概述、各項主要功能定義說明、使用者介面介紹、資料定義，以及案例說明等。

g) 修改歷史

2016/05/03 為第一版(version 1.0)

2016/05/05 為第一次的修正版(version 1.1)

2016/05/10 為第二次的修正版(version 1.2)

二、規格書內容描述

2.1 系統概述

2.1.1 系統目標

➤ 提供一個安全性高的平台

為了提高系統的安全性，系統在使用者登入時，會記錄使用者登入時間，用來掌握每個帳號的使用情況，以確保帳號被盜用時，能夠取得更多資訊，來保護使用者的權益。在重設密碼方面，系統則會以使用者申請的信箱為依據，寄送重設密碼的連結，以保證只能由本人修改帳號資訊，以上述兩個方法來提升整體系統的安全性。

➤ 人性化操作介面

本系統為了達到能夠讓所有使用者輕鬆上手的目的，特別針對不同權限的使用者作資料的分類，讓使用者能夠輕鬆管理、審核資料。

➤ 維持資料庫系統穩定

藉由ER-Model的輔助規劃整個資料庫的設計，以達到資料庫系統的穩定，讓使用者在讀寫資料的時候，可以提供高度穩定的運作。

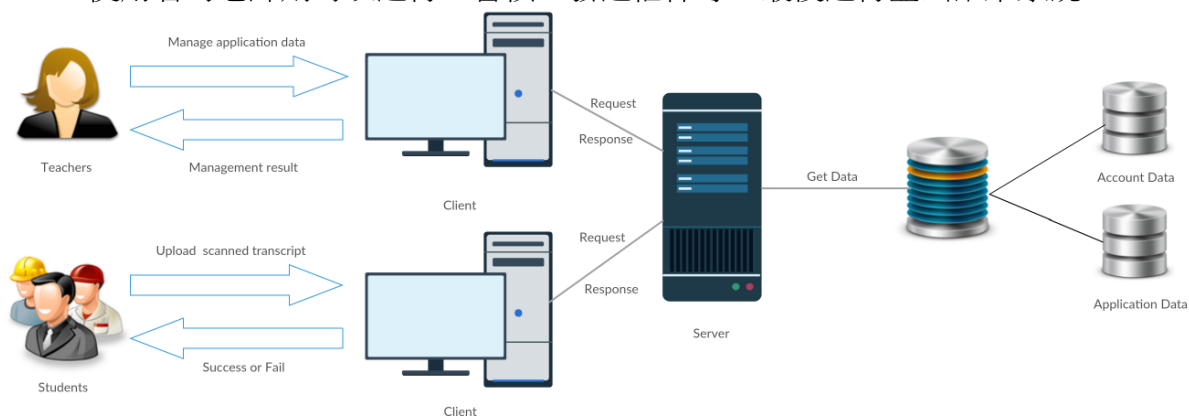
2.1.2 系統範圍

本研究所網路申請入學系統範圍包括

- 資料上傳及整合
- 管理者審核彙整
- 資料安全管理

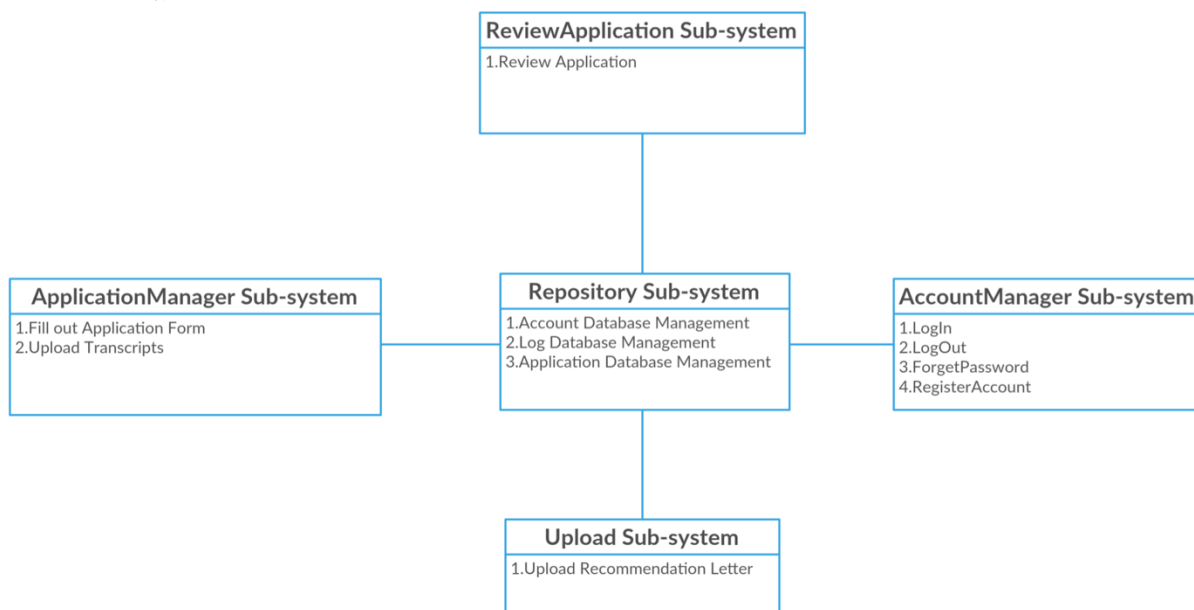
2.2.3 系統架構

使用者可以在起始頁面進行登入，進行登入時，使用者輸入之資料將會到資料庫進行比對，判斷是否為已註冊的會員，並提供忘記密碼重設服務，為維護使用者資料安全，密碼重設信件將會寄送到使用者申請的信箱，以確保只能由本人修改帳號資訊。登入成功後會依使用者不同（學生、老師），而有不同的功能可以使用，使用者為學生能夠進行：填寫資料、上傳資料、更改資料。使用者為老師則可以進行：審核、發送信件等，最後進行登出結束系統。



圖一、系統基本架構圖

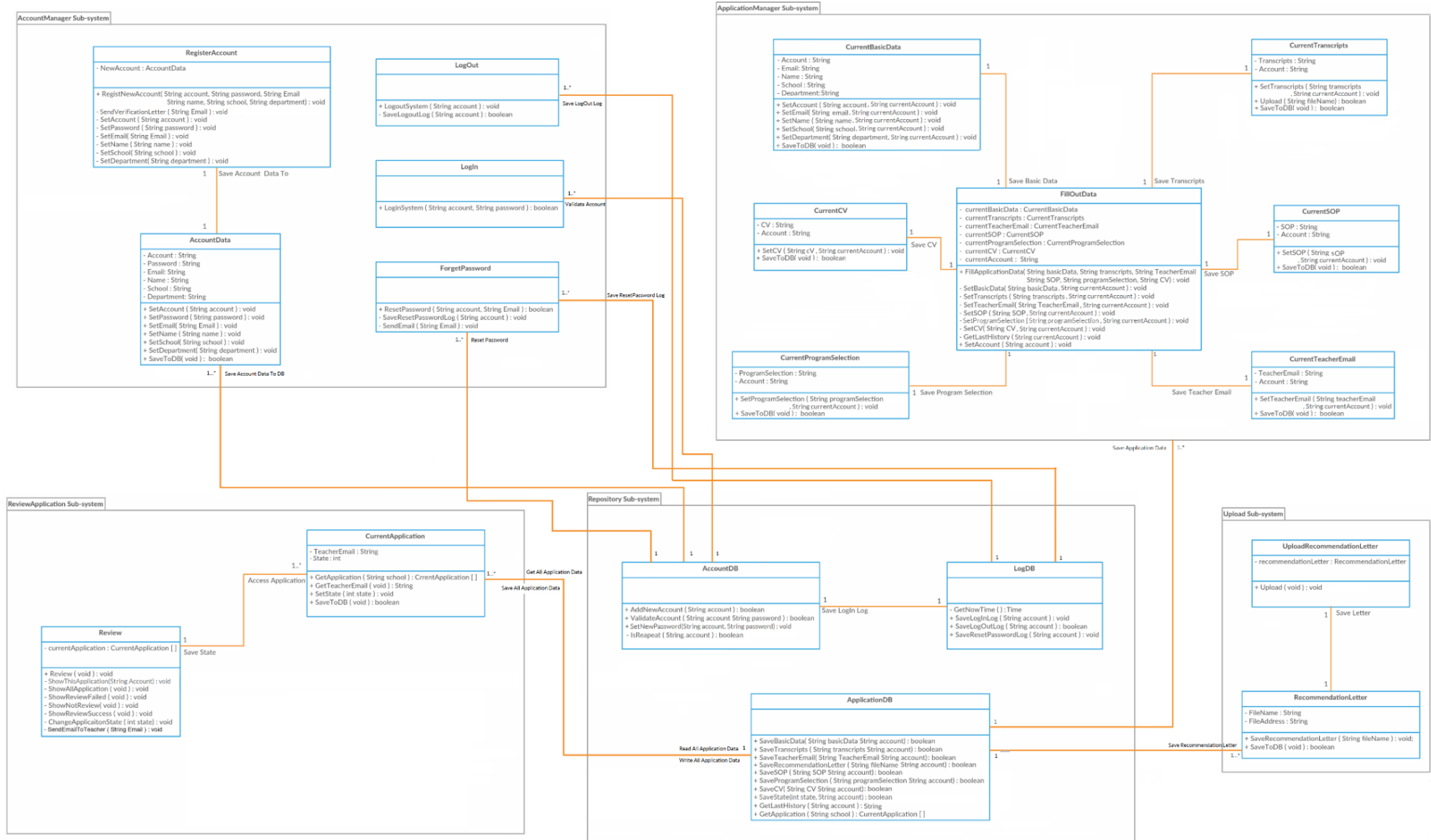
2.2 軟體架構



圖二、系統功能架構圖

- 儲存庫子系統(Repository Sub-system)
 - Account Database Management : 管理帳號資料庫
 - Log Database Management : 管理日誌資料庫
 - Application Database Management : 管理申請書資料庫
- 審核申請書子系統(ReviewApplication Sub-system)
 - Review Application : 用來審核申請書
- 帳號管理子系統(AccountManager Sub-system)
 - LogIn : 處理帳號登入，並記錄時間
 - LogOut : 處理帳號登出，並記錄時間
 - ForgetPassword : 處理使用者忘記密碼的時候，並記錄時間
 - RegisterAccount : 註冊一個新帳號
- 申請書管理子系統(ApplicationManager Sub-system)
 - Fill out Application Form : 線上填寫申請書
 - Upload Transcripts : 上傳成績單
- 上傳子系統(Upload Sub-system)
 - Upload Recommendation Letter : 讓推薦教授上傳推薦信

2.3 軟體類別定義



圖三、Package Diagram

2.4 軟體類別描述

Class RegisterAccount
類別描述：
提供使用者註冊帳號之用
類別屬性：
Private AccountData NewAccount：新註冊使用者的帳號資訊
類別函式：
Public void RegistNewAccount(String account, String password, String Email, String name, String school, String department)：註冊一個新帳號，需要使用者填入帳號、密碼、信箱、姓名、學校、科系
Private void SendVerificationLetter(String Email)：寄送驗證信
Private void SetAccount(String account)：設定使用者的帳號到 NewAccount
Private void SetPassword(String password)：設定使用者的密碼到 NewAccount
Private void SetEmail(String Email)：設定使用者的信箱到 NewAccount
Private void SetName(String name)：設定使用者的姓名到 NewAccount
Private void SetSchool(String school)：設定使用者的學校到 NewAccount
Private void SetDepartment(String department)：設定使用者的科系到 NewAccount

Class AccountData
類別描述：
將使用者資料統一在這個類別裡管理
類別屬性：
Private String Account：使用者的帳號
Private String Password：使用者的密碼
Private String Email：使用者的信箱
Private String Name：使用者的姓名
Private String School：使用者的學校
Private String Department：使用者的科系

類別函式：
Public void SetAccount(String account) : 設定使用者的帳號
Public void SetPassword(String password) : 設定使用者的密碼
Public void SetEmail(String Email) : 設定使用者的信箱
Public void SetName(String name) : 設定使用者的姓名
Public void SetSchool(String school) : 設定使用者的學校
Public void SetDepartment(String department) : 設定使用者的科系
Public boolean SaveToDB() : 將資訊存到資料庫

Class LogOut
類別描述：
處理使用者登出時的動作
類別屬性：
類別函式：
Public void LogoutSystem(String account) : 使用者登出系統
Private boolean SaveLogoutLog(String account) : 儲存使用者登出的資訊

Class LogIn
類別描述：
處理使用者登入時的動作
類別屬性：
類別函式：
Public boolean LoginSystem(String account, String password) : 使用者登入系統

Class ForgetPassword
類別描述：
使用者忘記密碼時需要重設密碼

類別屬性：
類別函式：
Public boolean ResetPassword(String account, String Email)：讓使用者重新設定密碼
Private void SaveResetPasswordLog(String account)：儲存使用者重設密碼的紀錄
Private void SendEmail(String Email)：寄出密碼重設信到使用者的信箱

Class FillOutData
類別描述：
提供一個能夠線上填寫申請書的介面，讓使用者使用
類別屬性：
Private CurrentBasicData currentBasicData：使用者基本資料
Private CurrentTranscripts currentTranscripts：使用者成績單
Private CurrentTeacherEmail currentTeacherEmail：推薦教授電子信箱
Private CurrentSOP currentSOP：使用者申請動機
Private CurrentProgramSelection currentProgramSelection：使用者選擇申請的科系
Private CurrentCV currentCV：使用者自傳
Private String currentAccount：使用者帳號
類別函式：
Public void FillApplicationData(String basicData, String transcripts, String TeacherEmail, String SOP, String programSelection, String CV)：填入申請書的資料
Private void SetBasicData(String basicData, String currentAccount)：將基本資料傳入對應的類別
Private void SetTranscripts(String transcripts, String currentAccount)：將成績單傳入對應的類別
Private void SetTeacherEmail(String TeacherEmail, String currentAccount)：將推薦教授的 Email 傳入對應的類別

Private void SetSOP(String SOP, String currentAccount) : 將使用者申請動機傳入對應的類別

Private void SetProgramSelection(String programSelection, String currentAccount) : 將使用者選擇申請的科系傳入對應的類別

Private void SetCV(String CV, String currentAccount) : 將使用者自傳傳入對應的類別

Private void GetLastHistory(String account) : 將使用者上次送出的資料載入到目前的頁面

Public void SetAccount(String account) : 紀錄目前使用者的帳號

Class CurrentBasicData

類別描述 :

將使用者填寫的基本資料傳入資料庫

類別屬性 :

Private String Account : 使用者帳號

Private String Email : 使用者信箱

Private String Name : 使用者姓名

Private String School : 使用者就讀學校

Private String Department : 使用者科系

類別函式 :

Public void SetAccount(String account, String currentAccount) : 設定使用者帳號

Public void SetEmail(String email, String currentAccount) : 設定使用者信箱

Public void SetName(String name, String currentAccount) : 設定使用者姓名

Public void SetSchool(String school, String currentAccount) : 設定使用者就讀學校

Public void SetDepartment(String department, String currentAccount) : 設定使用者科系

Public boolean SaveToDB() : 將使用者資料存入資料庫中

Class CurrentCV
類別描述：
將使用者填寫的自傳傳入資料庫
類別屬性：
Private String CV : 使用者填寫的自傳
Private String Account : 使用者帳號
類別函式：
Public void SetCV(String cV, String currentAccount) : 設定使用者的自傳
Public boolean SaveToDB() : 將使用者自傳存入資料庫中

Class CurrentProgramSelection
類別描述：
將使用者選擇申請的科系傳入資料庫
類別屬性：
Private String ProgramSelection : 使用者選擇申請的科系
Private String Account : 使用者帳號
類別函式：
Public void SetProgramSelection(String programSelection, String currentAccount) : 設定使用者所選擇的科系
Public boolean SaveToDB() : 將使用者所選擇的科系傳入資料庫中

Class CurrentTranscripts
類別描述：
將使用者的成績單傳入資料庫
類別屬性：
Private String Transcripts : 使用者的成績單
Private String Account : 使用者帳號
類別函式：
Public void SetTranscripts(String transcripts, String currentAccount) : 設定使用者的成績單

Public boolean Upload(String fileName) : 提供上傳的功能
 Public boolean SaveToDB() : 將使用者上傳的成績單傳入資料庫中

Class CurrentSOP
類別描述：
將使用者輸入的申請動機傳入資料庫
類別屬性：
Private String SOP : 使用者輸入的申請動機
Private String Account : 使用者帳號
類別函式：
Public void SetSOP(String sOP, String currentAccount) : 設定使用者輸入的申請動機
Public boolean SaveToDB() : 將使用者申請動機傳入資料庫中

Class CurrentTeacherEmail
類別描述：
將使用者的推薦教授信箱傳入資料庫
類別屬性：
Private String TeacherEmail : 使用者的推薦教授的信箱
Private String Account : 使用者帳號
類別函式：
Public void SetTeacherEmail(String teacherEmail, String currentAccount) : 設定使用者選擇的科系
Public boolean SaveToDB() : 將使用者推薦教授的信箱傳入資料庫中

Class Review
類別描述：
審查學生提出申請的申請書
類別屬性：
Private CurrentApplication[] currentApplication : 目前已呈交的申請書

類別函式：
Public void Review()：審查老師進行審查
Private void ShowThisApplication(String Account)：顯示該帳號所提出的申請書
Private void ShowAllApplication()：顯示所有申請書
Private void ShowReviewFailed()：顯示被評為不通過的申請書
Private void ShowNotReview()：顯示尚未審查的申請書
Private void ShowReviewSuccess()：顯示已被評為通過的申請書
Private void ChangeApplicationState(int state)：改變申請書之審查狀態
Private void SendEmailToTeacher(String Email)：寄信給推薦教授

Class CurrentApplication
類別描述：
管理及操作申請書
類別屬性：
Private String TeacherEmail：推薦教授的 Email
Private int state：申請書狀態
類別函式：
Public CurrentApplication[] GetApplication(String school)：取得對該學校提出申請的所有申請書
Public String GetTeacherEmail()：取得推薦教授的 Email
Public void SetState(int state)：設定該申請書之狀態
Public boolean SaveToDB()：將申請書狀態存入資料庫中

Class AccountDB
類別描述：
負責帳號資訊的管理，並且提供一個與帳戶資料庫連接的介面。
類別屬性：
類別函式：
Public boolean AddNewAccount(String account)：新增一個帳號

Public boolean ValidateAccount(String account, String password) : 判斷帳號密碼是否正確

Public void SetNewPassword(String account, String password) : 設定重設密碼後的新密碼

Private boolean IsRepeat(String account) : 判斷新註冊的帳號是否有重複

Class LogDB

類別描述：

負責記錄帳號登入、登出以及重設密碼的時間。

類別屬性：

類別函式：

Public void SaveLogInLog(String account) : 記錄使用者登入的時間

Public boolean SaveLogOutLog(String account) : 記錄使用者登出的時間

Public void SaveResetPasswordLog(String account) : 記錄使用者重設密碼的時間

Private Time GetNowTime() : 取得目前的時間

Class ApplicationDB

類別描述：

負責申請書的管理，並且提供一個與申請書資料庫連接的介面。

類別屬性：

類別函式：

Public boolean SaveBasicData(String basicData, String account) : 儲存使用者的基本資料

Public boolean SaveTranscripts(String transcripts, String account) : 儲存使用者的成績單

Public boolean SaveTeacherEmail(String TeacherEmail, String account) : 儲存使用者的推薦教授信箱

Public boolean SaveRecommendationLetter(String fileName, String account) : 儲存使用者的教授推薦信

Public boolean SaveSOP(String SOP, String account) : 儲存使用者的申請目的

Public boolean SaveProgramSelection(String programSelection, String account) : 儲存使用者的申請系所

Public boolean SaveCV(String CV, String account) : 儲存使用者的自傳

Public boolean SaveState(int state, String account) : 儲存申請書的狀態

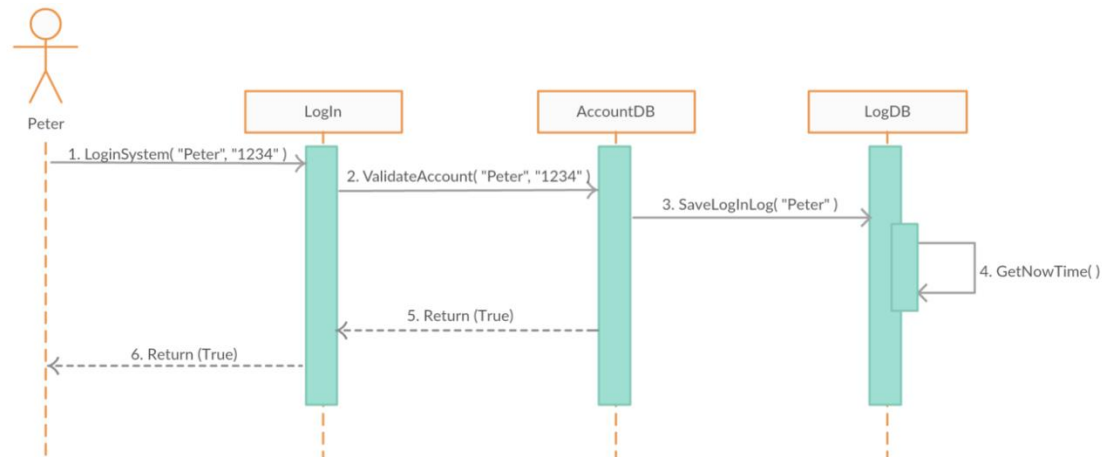
Public void GetLastHistory(String account) : 將使用者最後一次送出的資料，載入回現在的頁面

Public CurrentApplication[] GetApplication(String school) : 取得申請該學校的全部申請書

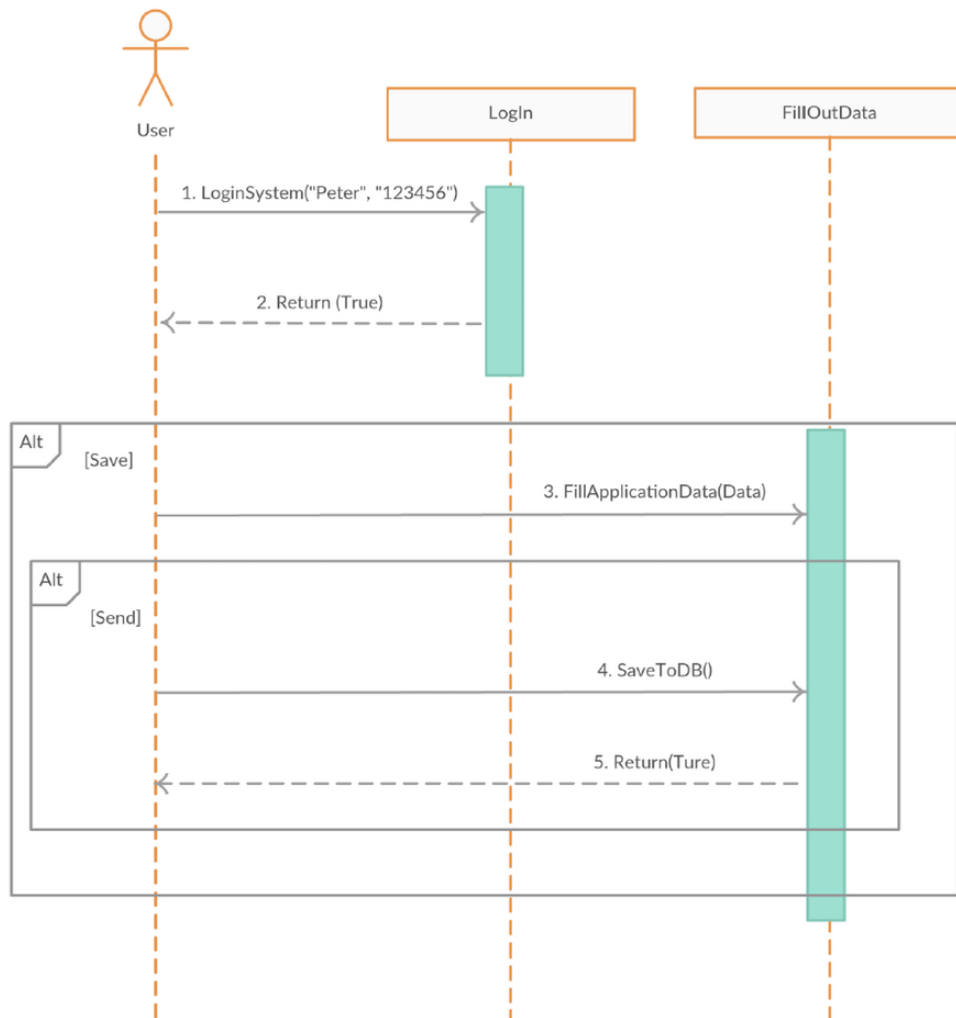
Class UploadRecommendationLetter
類別描述：
提供教授一個上傳推薦信的介面。
類別屬性：
Private RecommendationLetter recommendationLetter : 用來儲存推薦信的
類別函式：
Public void Upload() : 提供上傳的功能

Class RecommendationLetter
類別描述：
儲存推薦信的基本資料。
類別屬性：
Private String FileName : 用來儲存推薦信的檔案名稱
Private String FileAddress : 用來儲存推薦信的儲存位置
類別函式：
Public void SaveRecommendationLetter(String fileName) : 負責儲存推薦信
Public boolean SaveToDB() : 判斷推薦信有無成功存入到資料庫中

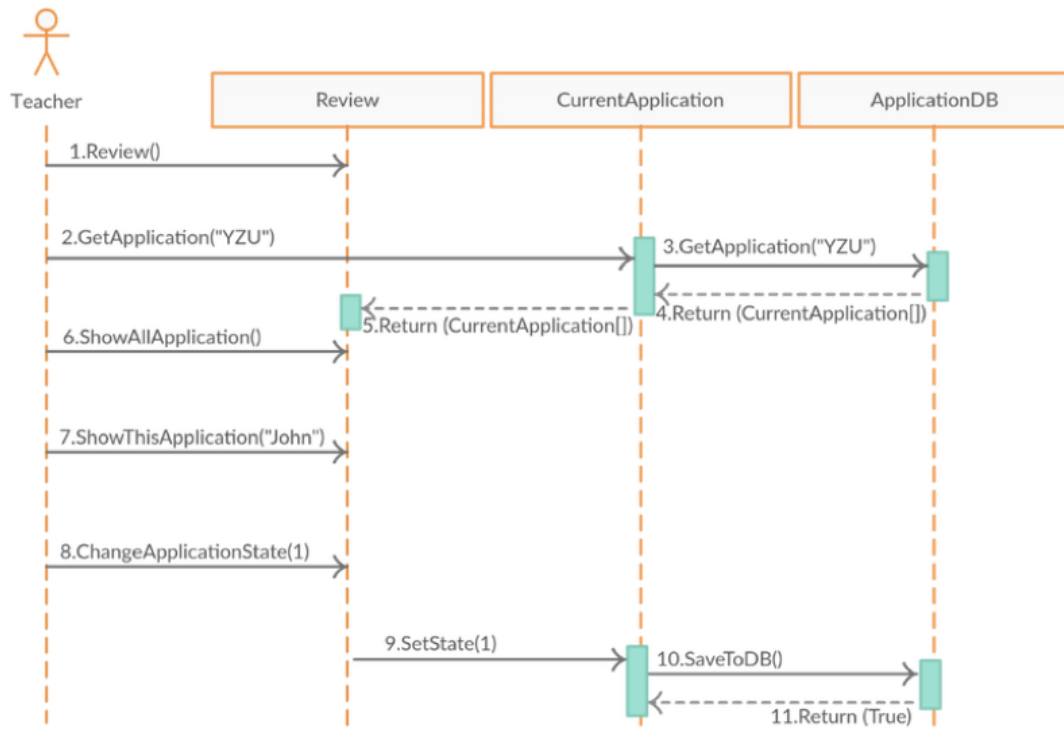
2.5 軟體行為概述



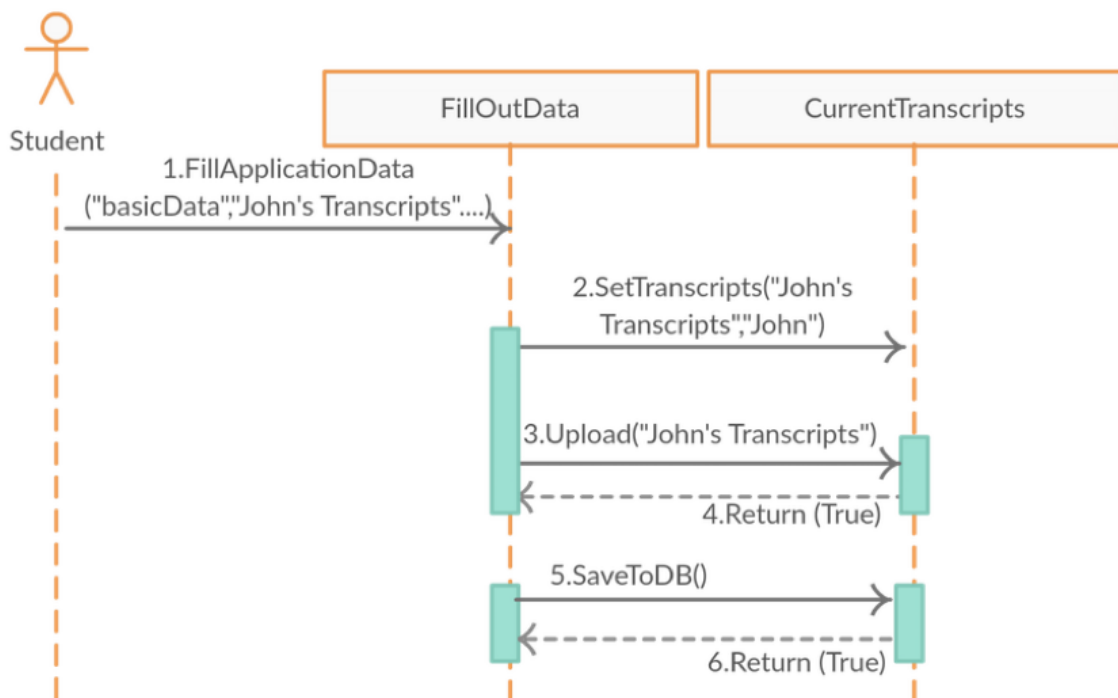
圖四、Sequence Diagram WGS_Login



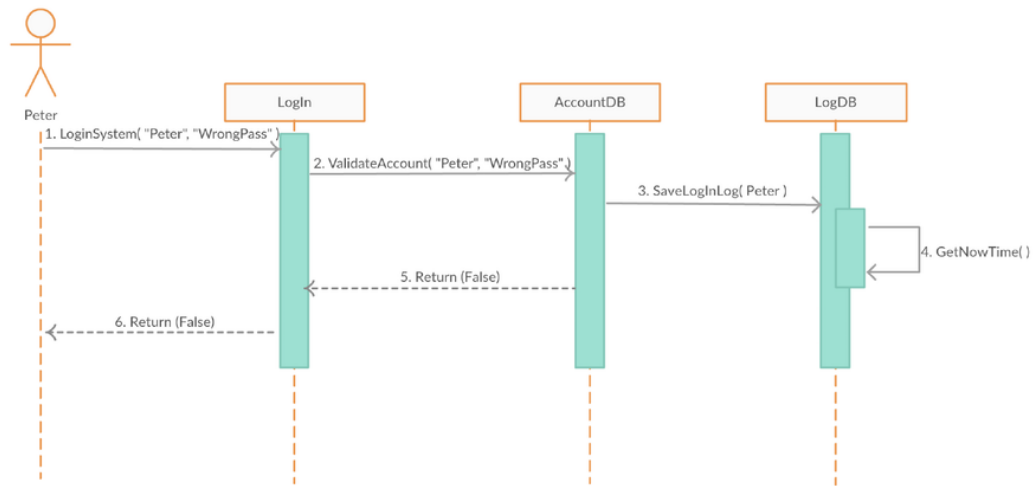
圖五、Sequence Diagram WGS_User_Info



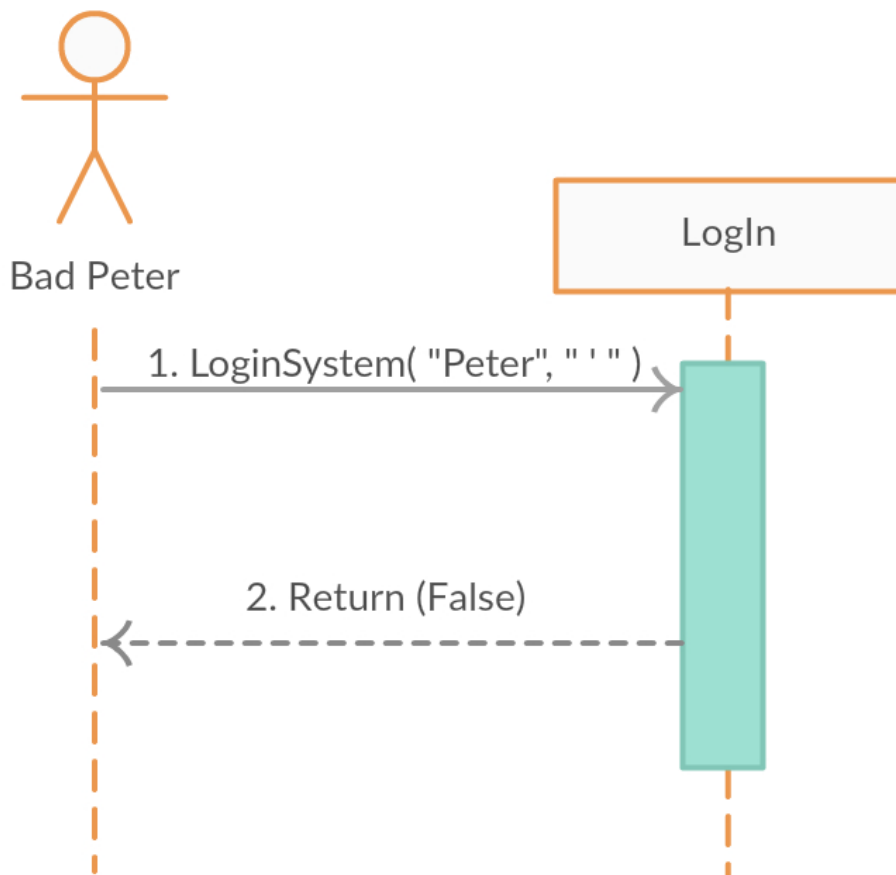
圖六、Sequence Diagram WGS_TeacherReview



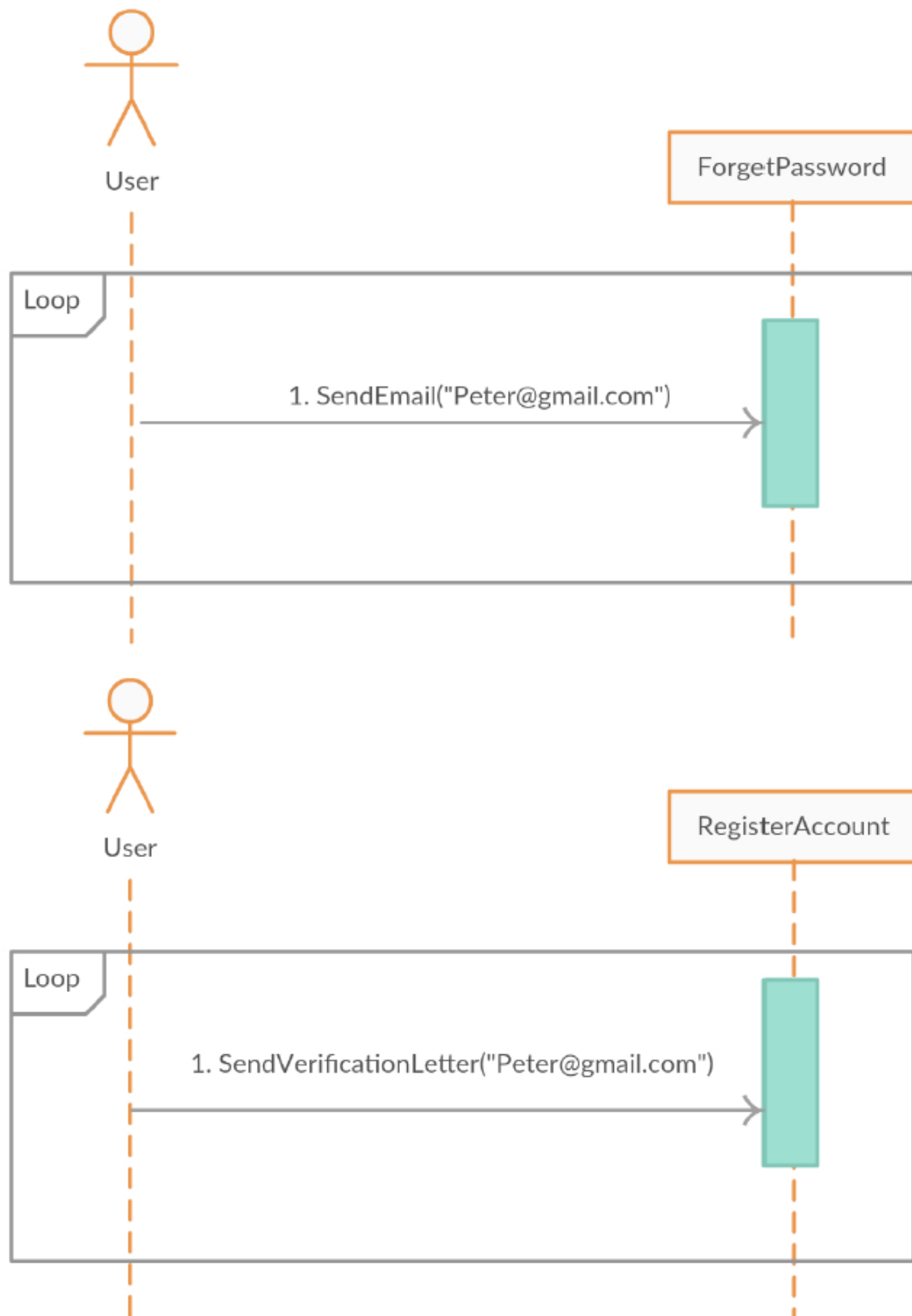
圖七、Sequence Diagram WGS_Upload_Transcripts



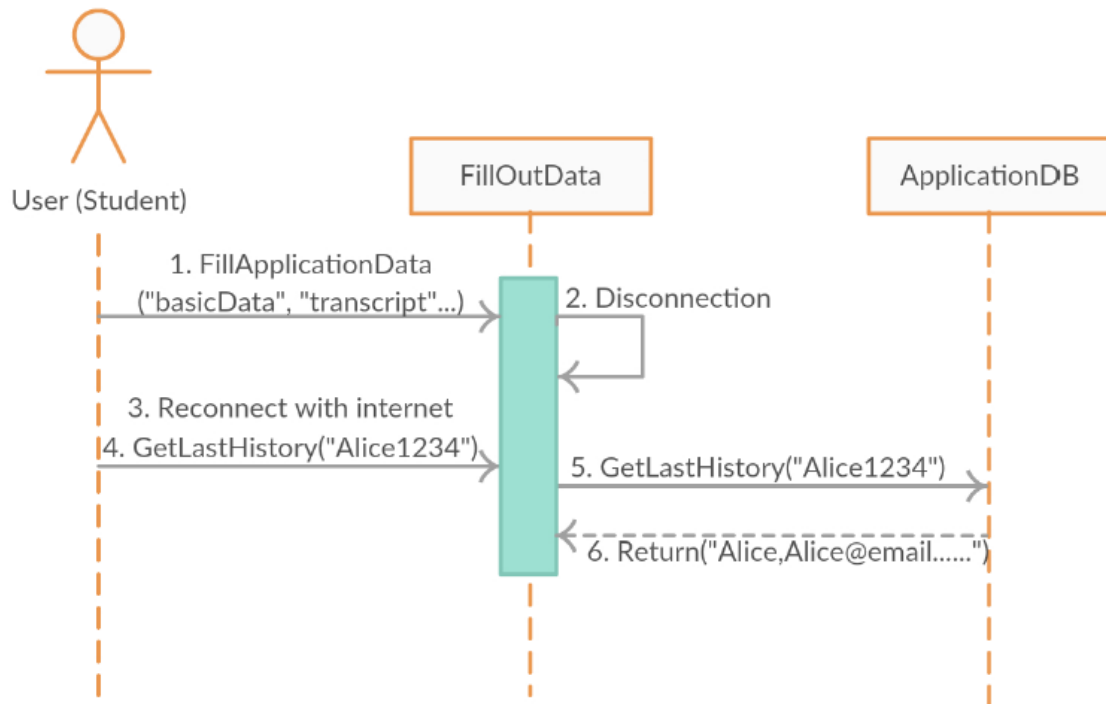
圖八、Sequence Diagram WGS_LogException01



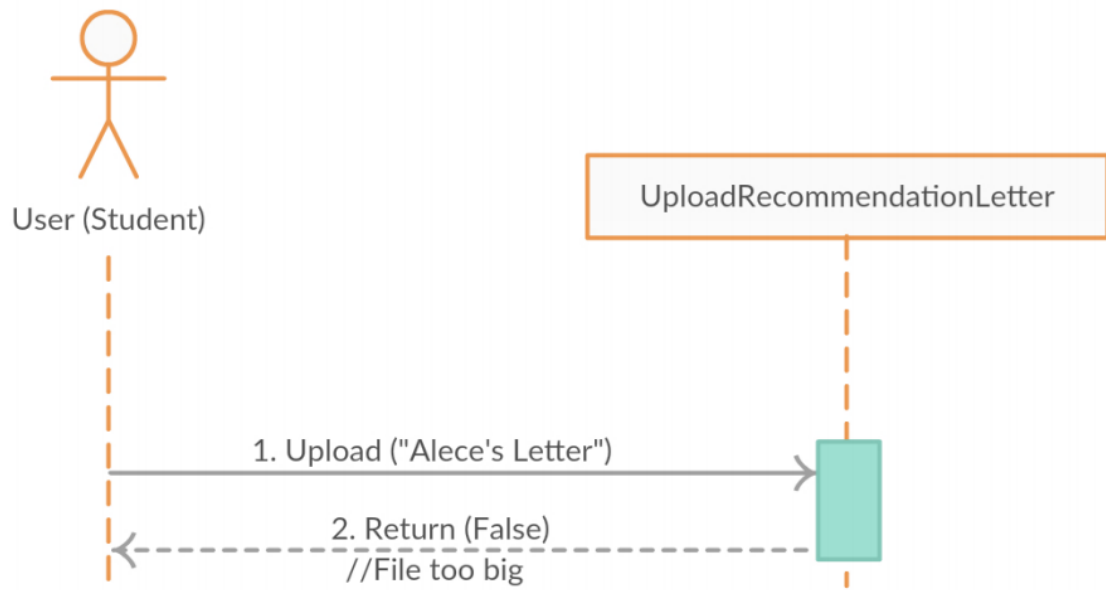
圖九、Sequence Diagram WGS_LogException02



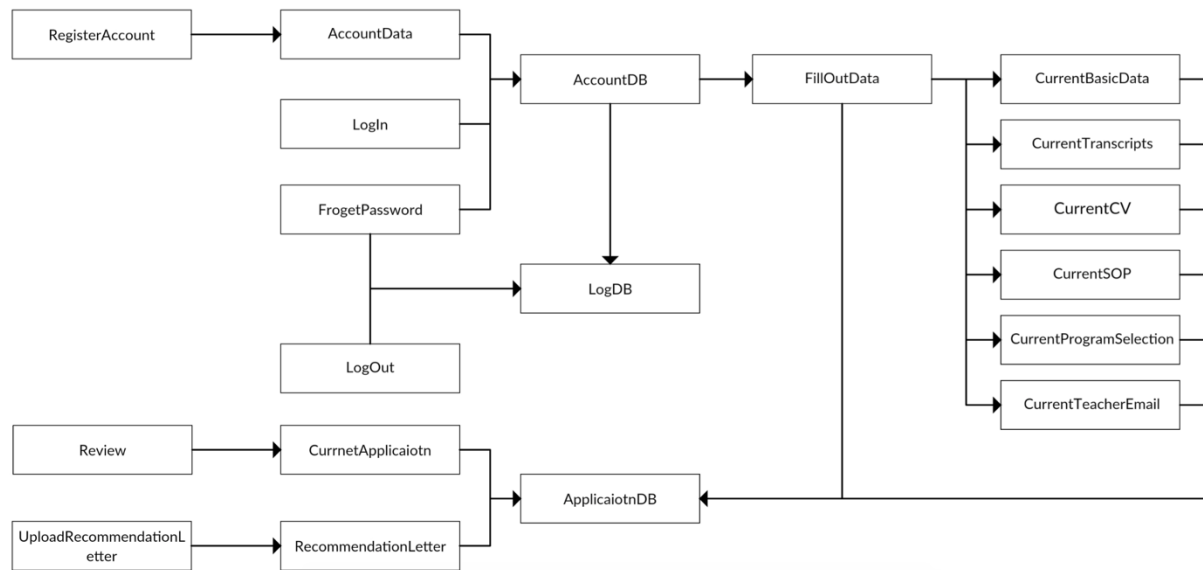
圖十、Sequence Diagram WGS_LossVerificationEmail



圖十一、Sequence Diagram WGS_Disconnection



圖十二、Sequence Diagram WGS_FileTooBig



圖十三、Client-Object Diagram

三、軟體設計描述

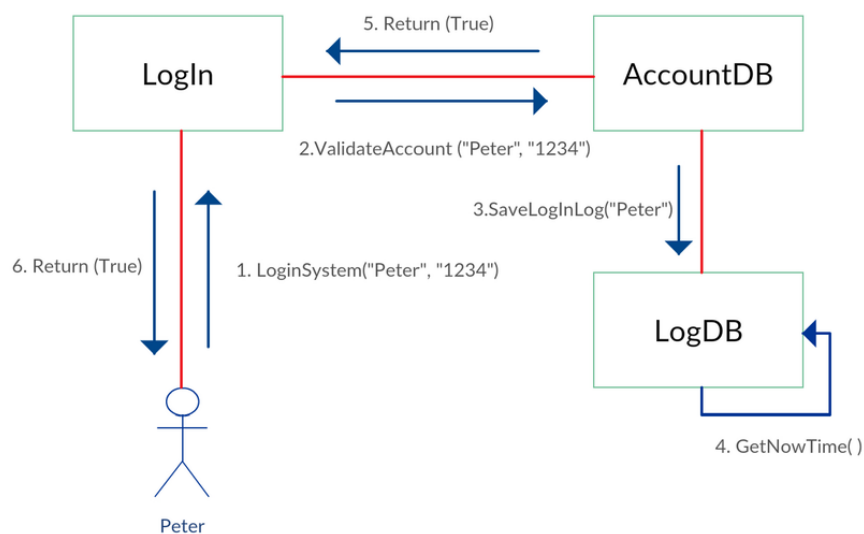
3.1 設計概述

本節內容主要是子系統之間的合作關係、各個子系統內部的類別合作關係及其類別相對應的虛擬碼等系統元件設計相關描述。

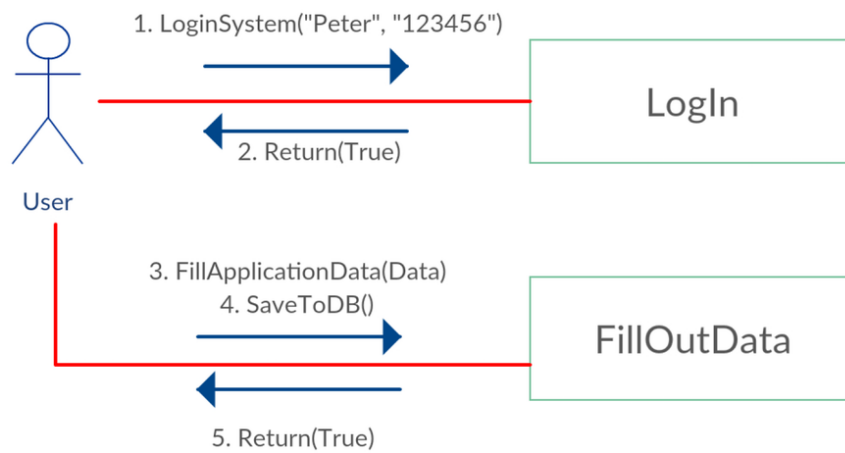
3.2 軟體類別設計描述

本系統功能包括：

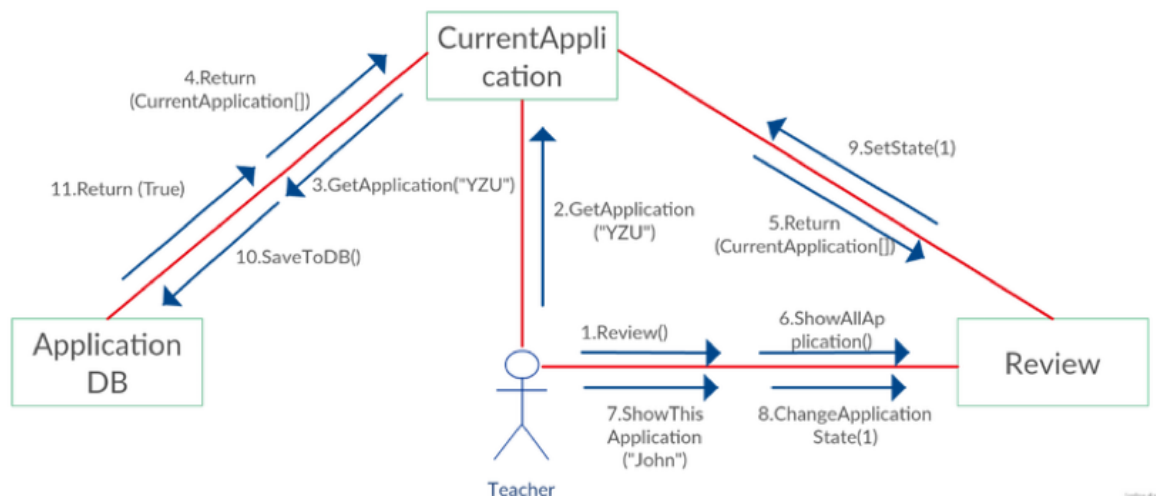
- 登入、登出本系統，以及忘記密碼查詢
- 線上填寫申請書
- 審核申請書子系統
- 上傳推薦信
- 資料庫管理



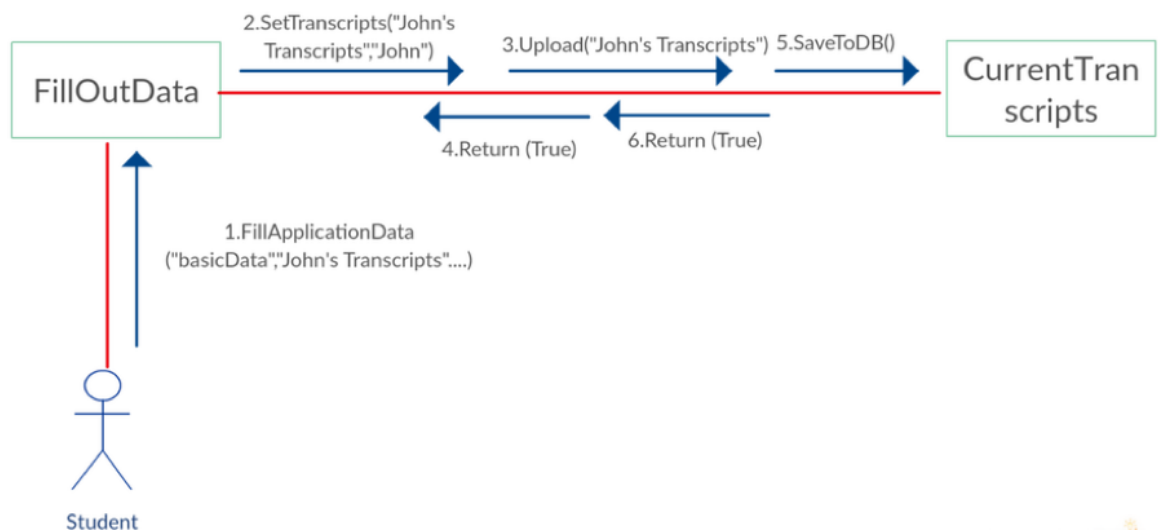
圖十四、Collaboration Diagram WGS_Login



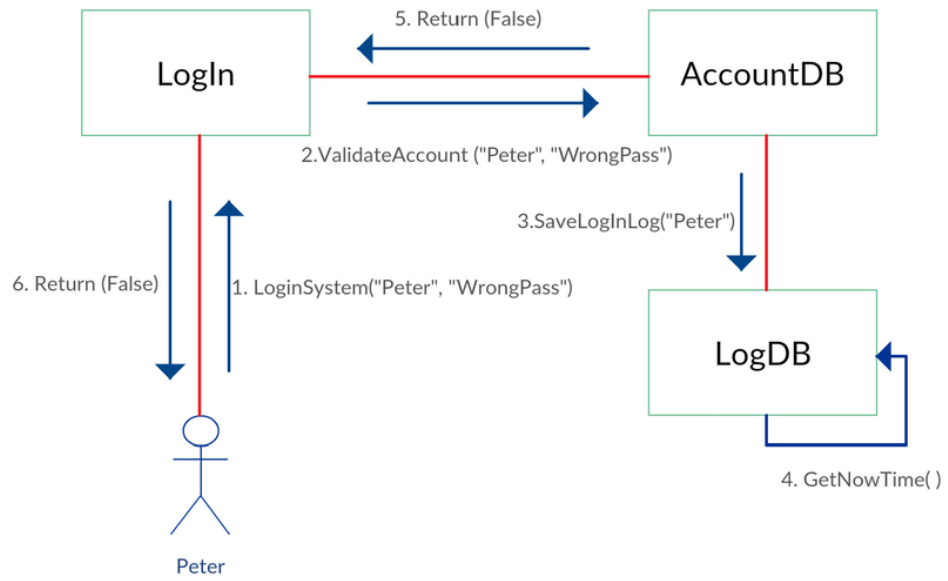
圖十五、Collaboration Diagram WGS_User_Info



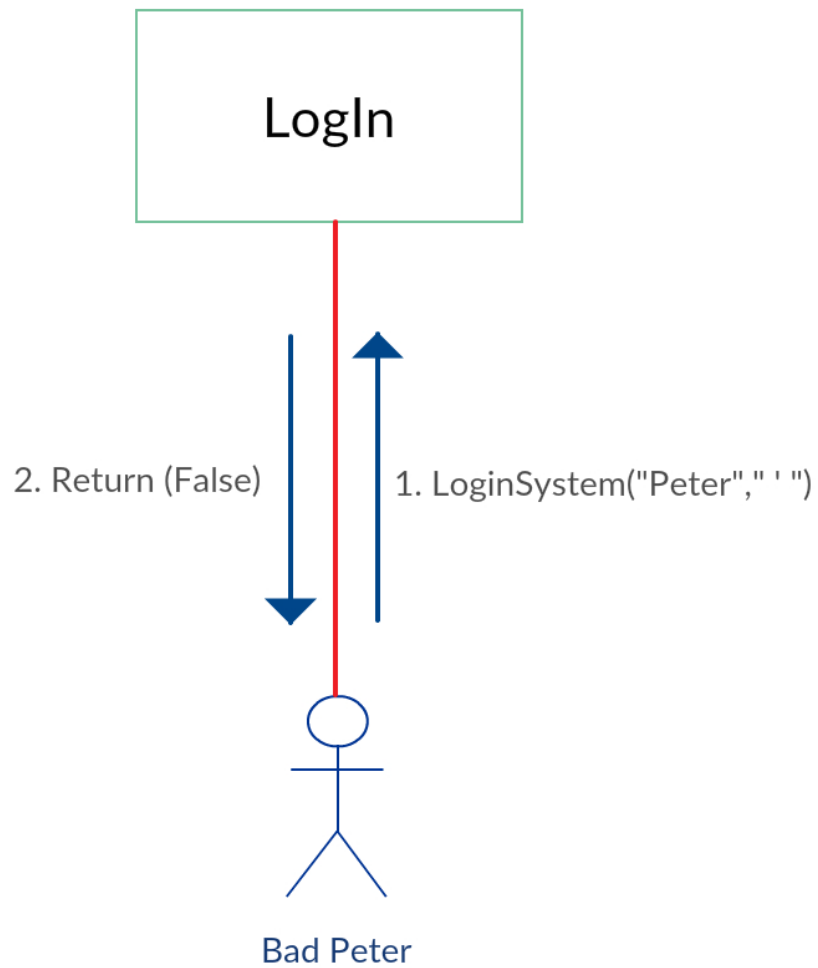
圖十六、Collaboration Diagram WGS_TeacherReview



圖十七、Collaboration Diagram WGS_Upload_Transcripts



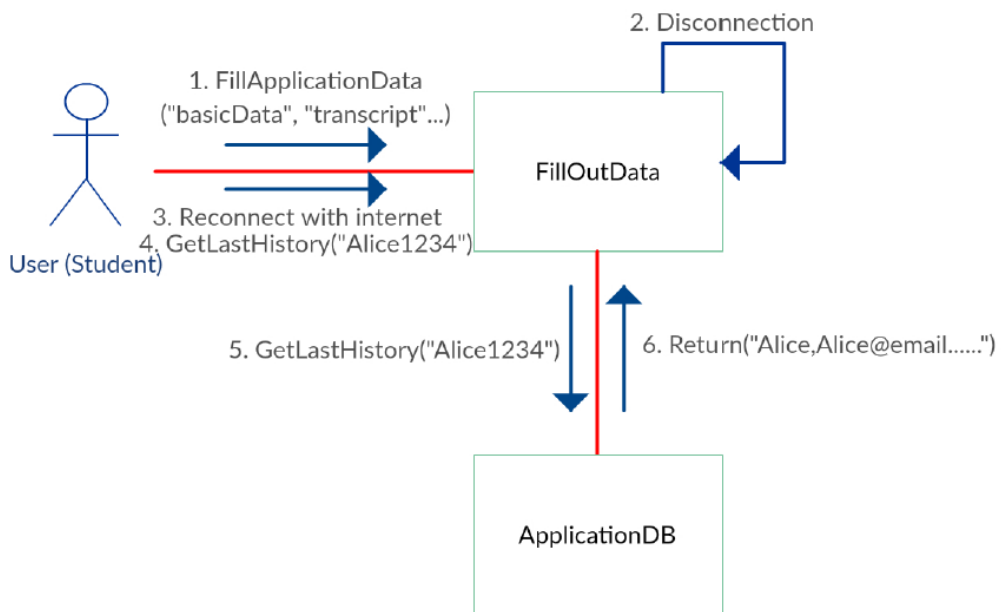
圖十八、Collaboration Diagram WGS_LogException01



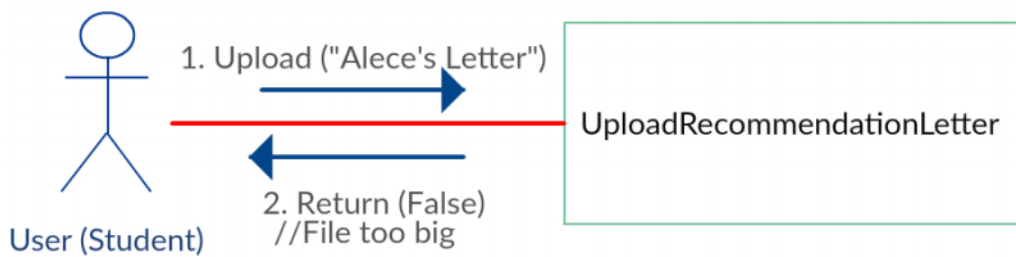
圖十九、Collaboration Diagram WGS_LogException02



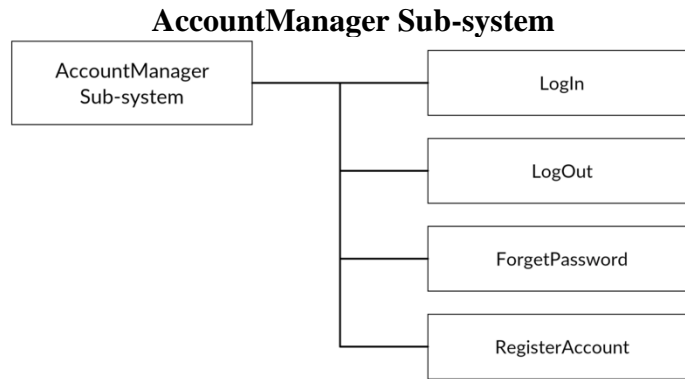
圖二十、Collaboration Diagram WGS_LossVerificationEmail



圖二十一、Collaboration Diagram WGS_Disconnection



圖二十二、Collaboration Diagram WGS_FileTooBig



圖二十三、帳號管理子系統

★登入系統(Login)

-說明：進入登入系統

-輸入：點擊登入按鈕

-處理：判斷是否登入成功

-輸出：登入成功

-類別函式：

```
Public boolean LoginSystem(String account, String password)
```

函式說明：讓使用者輸入帳號與密碼

輸入參數：帳號及密碼

輸出參數：登入是否成功

虛擬碼：

```
Public boolean LoginSystem(String account, String password)
{
    Return ValidateAccount(account,password);
}
```

-操作：

使用者進入登入畫面後，輸入帳號及密碼。

- 使用「登入」Button，系統會去判斷是否登入成功，若成功會跳出成功頁面。
- 使用「取消」Button，系統取消登入，並且不做任何動作。

★登出系統(LogOut)

-說明：進入登出系統

-輸入：點擊登出按鈕

-處理：讓使用者登出

-輸出：登出成功

-類別函式：

```
Public void LogoutSystem(String account)
```

函式說明：讓使用者登出

輸入參數：帳號

輸出參數：無

虛擬碼：

```
Public void LogoutSystem(String account)
```

```
{  
    if(SaveLogoutLog(account))  
        Show("登出成功")  
    else  
        Show("登出失敗")  
}
```

```
Private boolean SaveLogoutLog(String account)
```

函式說明：讓使用者登出

輸入參數：帳號

輸出參數：無

虛擬碼：

```
Private boolean SaveLogoutLog (String account)
```

```
{  
    Return LogDB::SaveLogOutLog(account);  
}
```

-操作：

使用者按登出後，會讓使用者登出。

- 使用「登出」button，系統會去判斷是否登入成功，若成功會跳出成功頁面。

★忘記密碼系統(ForgetPassword)

-說明：當使用者忘記密碼時，會去讓使用者重新設定密碼

-輸入：點擊忘記密碼

-處理：寄信給使用者並且重設使用者密碼

-輸出：重設成功

-類別函式：

```
Public boolean ResetPassword(String account, String email)
```

函式說明：重新設定密碼

輸入參數：使用者帳號以及信箱

輸出參數：是否有成功寄出更改密碼的信

虛擬碼：

```
Public boolean ResetPassword(String account, String email)
{
    for(AccountData 1:all)
    {
        if(AccountData.account == account && AccountData.Email == email)
            SendEmail(email);
        return 1;
    }
    return 0;
}
```

```
Private void SendEmail(String Email)
```

函式說明：寄出重設信給此信箱

輸入參數：使用者的信箱

輸出參數：無

虛擬碼：

```
Private void SendEmail(String Email)
```

```

{
    系統寄出重設信給此信箱
}

Private void SaveResetPasswordLog(String account)
函式說明：存取重設的密碼
輸入參數：使用者帳號
輸出參數：無
虛擬碼：

Private void SaveResetPasswordLog(String account)
{
    LogDB::SaveResetPasswordLog(account);
}

```

-操作：

使用者按忘記密碼畫面後，填寫帳號以及信箱，會寄重設信給此信箱去重設密碼。

- 使用「確認」button，系統會去判斷信箱與帳號是否正確，並且會寄重設信。

★註冊新帳號(RegisterAccount)

-說明：讓使用者註冊

-輸入：點擊註冊

-處理：存取使用者資訊

-輸出：註冊成功

-類別函式：

```

Public void RegistNewAccount(String account, String password, String email,
String name, String school, String department)
函式說明：使用者填寫註冊資料
輸入參數：使用者填寫的資料
輸出參數：無

```

虛擬碼：

```
Public void RegistNewAccount(String account, String password, String email,
String name, String school, String department)
{
    SetAccount(account);
    SetPassword(password);
    SetEmail(email);
    SetName(name);
    SetSchool(school);
    SetDepartment(department);
    SendVerificationLetter(email);
    NewAccount.saveToDB();
}
```

Private void SetAccount(String account)

函式說明：設定使用者的帳號

輸入參數：使用者的帳號

輸出參數：無

虛擬碼：

```
Private void SetAccount(String account)
{
    NewAccount.SetAccount(account);
}
```

Private void SetName(String name)

函式說明：設定使用者的名字

輸入參數：使用者的名字

輸出參數：無

虛擬碼：

```
Private void SetName(String name)
{
```

```
        NewAccount.SetName(name);  
    }
```

Private void SetPassword(String password)

函式說明：設定使用者的密碼

輸入參數：使用者的密碼

輸出參數：無

虛擬碼：

Private void SetPassword(String password)

```
{  
    NewAccount.SetPassword(password);  
}
```

Private void SetEmail(String Email)

函式說明：設定使用者的信箱

輸入參數：使用者的信箱

輸出參數：無

虛擬碼：

Private void SetEmail(String Email)

```
{  
    NewAccount.SetEmail(Email);  
}
```

Private void SetSchool(String school)

函式說明：設定使用者的學校

輸入參數：使用者的學校

輸出參數：無

虛擬碼：

Private void SetSchool(String school)

```
{  
    NewAccount.SetSchool(school);  
}
```

```
}
```

Private void SetDepartment(String department)

函式說明：設定使用者的學系

輸入參數：使用者的學系

輸出參數：無

虛擬碼：

Private void SetDepartment(String department)

```
{
```

```
    NewAccount.SetDepartment(department);
```

```
}
```

Private void SendVerificationLetter(String Email)

函式說明：寄出驗證信給此信箱

輸入參數：使用者的信箱

輸出參數：無

虛擬碼：

Private void SendVerificationLetter (String Email)

```
{
```

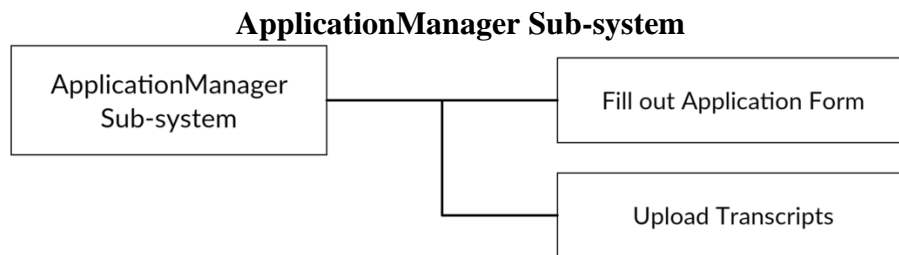
```
    系統寄出驗證信給此信箱
```

```
}
```

-操作：

使用者按註冊後，填寫所有的註冊資料，之後會寄驗證信給此信箱去驗證。

- 使用「送出」button，系統會去判斷帳號是否有重複，並且同時會寄驗證信到註冊者的信箱。



圖二十四、申請書管理子系統

★填寫申請書(Fill out Application Form)

- 說明：填寫使用者申請書
- 輸入：使用者填寫之資料
- 處理：提供使用者填寫申請書的介面
- 輸出：送出成功/失敗
- 類別函式：

Public void SetAccount(String account)

函式說明：設定使用者帳號

輸入參數：目前使用者的帳號

輸出參數：無

虛擬碼：

Public void SetAccount(String account)

{

WHEN LOGIN

Set user account ID to currentAccount

//currentAccount = account

}

Public void FillApplicationData(String basicData, String transcripts, String TeacherEmail, String SOP, String programSelection, String CV)

函式說明：填寫使用者資料

輸入參數：基本資料、成績單、推薦教授信箱、動機簡述、選擇系所、
自傳

輸出參數：無

虛擬碼：

```
Public void FillApplicationData(String basicData, String transcripts, String
TeacherEmail, String SOP, String programSelection, String CV)
{
    //When user click "Submit" button
    //CurrentAccountt has been set when user login
    SetBasicData(basicData,currentAccount);
    SetTranscripts(transcripts,currentAccount);
    SetTeacherEmail(TeacherEmail,currentAccount);
    SetSOP(SOP,currentAccount);
    SetProgramSelection(programSelection,currentAccount);
    SetCV(CV,currentAccount);

    currentBasicData.SaveToDB();
    currentTranscripts.SaveToDB();
    currentTeacherEmail.SaveToDB();
    currentSOP.SaveToDB();
    currentProgramSelection.SaveToDB();
    currentCV.SaveToDB();
}
```

Private void SetBasicData(String basicData, String currentAccount)

函式說明：設定使用者填寫的資本資料

輸入參數：使用者基本資料和目前的使用者

輸出參數：無

虛擬碼：

```
Private void SetBasicData(String basicData, String currentAccount)
{
    //CurrentAccountt has been set when user login
    Parse basicData into Email, name, school, department,THEN
    {
```

```

        currentBasicData.SetEmail(email, currentAccount);
        currentBasicData.SetName(name, currentAccount);
        currentBasicData.SetSchool(school, currentAccount);
        currentBasicData.SetDepartment(department, currentAccount);
    }
}

```

Public void SetAccount(String account, String currentAccount)

函式說明：設定使用者帳號到 **CurrentBasicData** 這個類別中

輸入參數：目前使用者的帳號和目前的使用者

輸出參數：無

虛擬碼：

Public void SetAccount(String account, String currentAccount)

```

{
    Account = currentAccount;
    Account = account;
}

```

Public void SetEmail(String email, String currentAccount)

函式說明：設定使用者信箱到 **CurrentBasicData** 這個類別中

輸入參數：目前使用者的信箱和目前的使用者

輸出參數：無

虛擬碼：

Public void SetEmail(String email, String currentAccount)

```

{
    Account = currentAccount;
    Email = email;
}

```

Public void SetName(String name, String currentAccount)

函式說明：設定使用者姓名到 **CurrentBasicData** 這個類別中

輸入參數：使用者的姓名和目前的使用者

輸出參數：無

虛擬碼：

```
Public void SetName(String name, String currentAccount)
{
    Account = CurrentAccount;
    Name = name;
}
```

```
Public void SetSchool(String school, String currentAccount)
```

函式說明：設定使用者就讀學校到 **CurrentBasicData** 這個類別中

輸入參數：使用者就讀的學校和目前的使用者

輸出參數：無

虛擬碼：

```
Public void SetSchool(String school, String currentAccount)
{
    Account = currentAccount;
    School = school;
}
```

```
Public void SetDepartment(String department, String currentAccount)
```

函式說明：設定使用者科系到 **CurrentBasicData** 這個類別中

輸入參數：使用者的科系和目前的使用者

輸出參數：無

虛擬碼：

```
Public void SetDepartment(String department, String currentAccount)
{
    Account = currentAccount;
    Department = department;
}
```

Private void SetTeacherEmail(String TeacherEmail, String currentAccount)

函式說明：設定使用者推薦人教授的電子信箱

輸入參數：推薦教授的電子信箱和目前的使用者

輸出參數：無

虛擬碼：

Private void SetTeacherEmail(String TeacherEmail, String currentAccount)

```
{  
    currentTeacherEmail.SetTeacherEmail(teacherEmail, currentAccount);  
}
```

Public void SetTeacherEmail(String teacherEmail, String currentAccount)

函式說明：設定使用者推薦人教授之電子信箱到 CurrentTeacherEmail
這個類別中

輸入參數：推薦教授的電子信箱和目前的使用者

輸出參數：無

虛擬碼：

Public void SetTeacherEmail(String teacherEmail, String currentAccount)

```
{  
    Account = currentAccount;  
    TeacherEmail = teacherEmail;  
}
```

Private void SetSOP(String SOP, String currentAccount)

函式說明：設定使用者申請動機簡述

輸入參數：申請動機簡述和目前的使用者

輸出參數：無

虛擬碼：

Private void SetSOP(String SOP, String currentAccount)

```
{  
    currentSOP.SetSOP(SOP, currentAccount);  
}
```

Public void SetSOP(String sOP, String currentAccount)

函式說明：設定使用者申請動機簡述到 **CurrentSOP** 這個類別中

輸入參數：申請動機簡述和目前的使用者

輸出參數：無

虛擬碼：

Public void SetSOP(String sOP, String currentAccount)

```
{  
    Account = currentAccount;  
    SOP = sOP;  
}
```

**Private void SetProgramSelection(String programSelection, String
currentAccount)**

函式說明：設定使用者選擇申請之系所

輸入參數：使用者申請系所和目前的使用者

輸出參數：無

虛擬碼：

**Private void SetProgramSelection(String programSelection, String
currentAccount)**

```
{  
    currentProgramSelection.SetProgramSelection(  
        programSelection, currentAccount)  
}
```

**Public void SetProgramSelection(String programSelection, String
currentAccount)**

函式說明：設定使用者選擇申請之系所到 **CurrentProgramSelection**
這個類別中

輸入參數：使用者申請系所和目前的使用者

輸出參數：無

虛擬碼：

```
Public void SetProgramSelection(String programSelection, String
currentAccount)
{
    Account = currentAccount;
    ProgramSelection = programSelection;
}
```

Private void SetCV(String CV, String currentAccount)

函式說明：設定使用者之自傳

輸入參數：使用者自傳和目前的使用者

輸出參數：無

虛擬碼：

```
Private void SetCV(String CV, String currentAccount)
{
    currentCV.SetCV(CV, currentAccount);
}
```

Public void SetCV(String cV, String currentAccount)

函式說明：設定使用者自傳到 CurrentProgramSelection 這個類別中

輸入參數：使用者自傳和目前的使用者

輸出參數：無

虛擬碼：

```
Public void SetCV(String cV, String currentAccount)
{
    Account = currentAccount;
    CV = cV;
}
```

Private void SetTranscripts(String transcripts, String currentAccount)

函式說明：設定使用者之成績單

輸入參數：使用者的成績單和目前的使用者

輸出參數：無

虛擬碼：

```
Private void SetTranscripts(String transcripts, String currentAccount)
{
    currentTranscripts.SetTranscripts(transcripts, currentAccount);
}
```

Public void SetTranscripts(String transcripts, String currentAccount)

函式說明：設定使用者之成績單到 CurrentTranscripts 這個類別中

輸入參數：使用者的成績單和目前的使用者

輸出參數：無

虛擬碼：

```
Public void SetTranscripts(String transcripts, String currentAccount)
{
    Account = currentAccount;
    CV = cV;
}
```

Private void GetLastHistory(String currentAccount)

函式說明：抓取使用者先前填寫的資料

輸入參數：使用者帳號

輸出參數：無

虛擬碼：

```
Private void GetLastHistory(String currentAccount)
{
    ApplicationDB::GetLastHistory(currentAccount);
    Use “account” as primary key to get every data in database
    Set download data to
    {
        currentBasicData
        currentTranscripts
    }
}
```

```

        currentTeacherEmail
        currentSOP
        currentProgramSelection
        currentCV
    }
}

```

Public boolean SaveToDB(void)

函式說明：上傳類別中所有資料成員到資料庫

輸入參數：無

輸出參數：成功/失敗

虛擬碼：

Public boolean SaveToDB(void)

```

{
    PUSH all user information to database
}

```

p.s 這個函式分別出現在以下類別中，而裡面進行的動作都是將資料推到儲存庫子系統，所以我們只寫一個代表他們裡面的動作。

出現的類別：CurrentBasicData, CurrentCV, CurrentProgramSelection, CurrentTranscripts, CurrentSOP, CurrentTeacherEmail

-操作：

登入後若帳號為申請者帳號直接進入填寫資料頁面。

- 使用「修改」Button 載入先前所填寫之資料。
- 使用「送出」Button 送出所有資料。
- 使用「取消」Button 放棄目前所修改之資料。

★上傳成績單(Upload Transcripts)

-說明：上傳成績單

-輸入：附加檔案成績單

-處理：將使用者的成績單送往資料庫

-輸出：送出成功/失敗

-類別函式：

Public void SetTranscripts(String transcripts, String currentAccount)

函式說明：設定使用者之成績單

輸入參數：使用者的成績單

輸出參數：無

虛擬碼：

Public void SetTranscripts(String transcripts, String currentAccount)

```
{  
    Account = currentAccount;  
    Transcripts = transcripts;  
    Result = Upload(Transcripts);  
    IF Result = 1  
        Show "Upload Success !" message  
    ELSE  
        Show "Upload Fail, please try again!" message  
}
```

Public boolean Upload(String fileName)

函式說明：上傳使用者成績單

輸入參數：成績單檔案

輸出參數：成功/失敗

虛擬碼：

Public bool Upload(String fileName)

```
{  
    Call Reposiory Sub-system  
    Upload file  
    Record Address  
    Return boolean  
}
```

-操作：

在填寫資料頁面執行。

- 使用「附加檔案」Button 選取要上傳之檔案。
- 使用「送出」Button 確認送出檔案。
- 使用「取消」Button 放棄使用附加檔案功能。

ReviewApplication Sub-system



圖二十五、申請書審查子系統

★審查子系統(Review)

-說明：對申請書進行審查

-輸入：點擊審核按鈕觸發

-處理：顯示及審查申請書

-輸出：改變申請書的狀態

-類別函式：

Public void Review()

函式說明：對所選到的申請書進行審查

輸入參數：無

輸出參數：無

虛擬碼：

Public void Review()

{

 LOOP

 while(input)//老師決定執行的動作

 ShowAllApplication()

 or

 ShowReviewFailed()

 or

 ShowNotReview()

```

        or
        ShowReviewSuccess()
        or
        Exit
        ShowThisApplication(Account)
        ChangeApplicationState(state)
    END LOOP
}

```

Public CurrentApplication[] GetApplication(String school)

函式說明：取得所有學校為 school 的申請書

輸入參數：school

輸出參數：CurrentApplication[]

虛擬碼：

```

Public CurrentApplication[] GetApplication(String school)
{
    Return CurrentApplication [] which School's name of
    application is school
}

```

Private void ShowAllApplication()

函式說明：顯示所有申請書

輸入參數：無

輸出參數：無

虛擬碼：

```

Private void ShowAllApplication()
{
    Print name, school of currentApplications.
}

```

Private void ShowReviewFailed()

函式說明：顯示審核結果為不通過的申請書

輸入參數：無

輸出參數：無

虛擬碼：

```
Private void ShowReviewFailed()
```

```
{  
    Print currentApplications which State are “not pass”  
}
```

```
Private void ShowNotReview()
```

函式說明：顯示尚未進行審核的申請書

輸入參數：無

輸出參數：無

虛擬碼：

```
Private void ShowNotReview()
```

```
{  
    Print currentApplications which State are “not review”  
}
```

```
Private void ShowReviewSuccess()
```

函式說明：顯示審核結果為通過的申請書

輸入參數：無

輸出參數：無

虛擬碼：

```
Private void ShowReviewSuccess()
```

```
{  
    Print currentApplications which State are “pass”  
}
```

```
Private void ShowThisApplication(String Account)
```

函式說明：顯示”Account”的申請書的所有資料

輸入參數：此使用者的帳號

輸出參數：無

虛擬碼：

```
Private void ShowThisApplication(String Account)
```

```
{  
    Print Account's Application  
}
```

```
Private void ChangeApplicationState(int state)
```

函式說明：改變目前審核中的申請書的狀態

輸入參數：正在進行審核中的申請書的狀態

輸出參數：無

虛擬碼：

```
Private void ChangeApplicationState(int state)
```

```
{  
    currentApplication.SetState(state);  
}
```

```
Public String GetTeacherEmail()
```

函式說明：取得推薦教授的信箱

輸入參數：無

輸出參數：無

虛擬碼：

```
Public String GetTeacherEmail()
```

```
{  
    Return currentApplication's TeacherEmail  
}
```

```
Private void SendEmailToTeacher(String Email)
```

函式說明：寄送電子郵件給推薦此學生的老師

輸入參數：推薦教授的 Email

輸出參數：無

虛擬碼：

```
Private void SendEmailToTeacher(String Email)
{
    mailto(Email)
}
```

```
Public void SetState(int state)
```

函式說明：設定 State 為 state

輸入參數：欲改變之 state 值

輸出參數：無

虛擬碼：

```
Public void SetState(int state)
{
    State = state
}
```

```
Public boolean SaveToDB()
```

函式說明：將改變儲存至資料庫

輸入參數：無

輸出參數：True/False

虛擬碼：

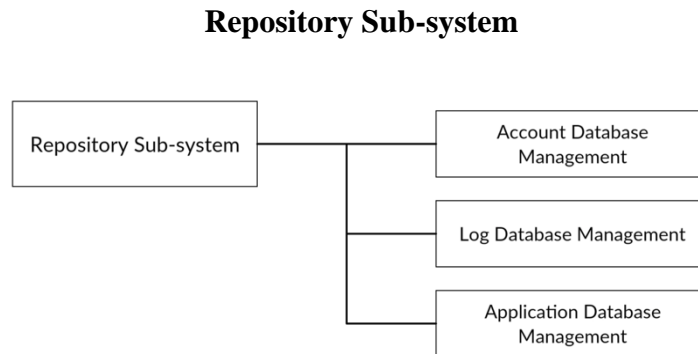
```
Public boolean SaveToDB()
{
    Save State to this currentApplication
}
```

-操作

在功能選單選擇「審查」，進入審查頁面。

- 使用「顯示所有申請書」Button 顯示所有提出申請的申請書。
- 使用「顯示審查狀態為不通過的申請書」Button 顯示審查狀態為不通過的申請書。
- 使用「顯示審查狀態為通過的申請書」Button 顯示審查狀態為通過的申請書。

- 使用「顯示審查狀態為尚未審查的申請書」Button 顯示審查狀態為尚未審查的申請書。
- 使用顯示在畫面上任一申請書進入審查。
- 使用「寄送電子郵件給推薦教授」Button 寄送電子郵件給推薦教授
- 使用「送出」Button 改變申請書的審查狀態。



圖二十六、資料庫子系統

★帳號資料庫管理(Account Database Management)

-說明：負責掌管帳號的資料庫

-輸入：由功能選項觸發不同的功能，和依要求輸入帳號或密碼

-處理：新增一筆帳號、驗證帳號、檢查帳號是否重複、更新密碼

-輸出：將要求的結果回傳給呼叫的函式

-類別函式：

Public boolean AddNewAccount(String account)

函式說明：將新註冊的帳號加到資料庫中

輸入參數：新用戶的帳號名稱、密碼、電子郵件、名字、學系等

輸出參數：是否有註冊成功

虛擬碼：

Public boolean AddNewAccount(String account)

{

IF (IsRepeat(String account)) THEN

 Return False

ELSE

 write this user's information into account database

```
        Return True
    }
```

Public boolean ValidateAccount(String account, String password)

函式說明：登入時，驗證帳號密碼有沒有錯誤

輸入參數：用戶的帳號和密碼

輸出參數：是否有登入成功

虛擬碼：

Public boolean ValidateAccount(String account, String password)

```
{
    Select this account from account database
    IF(account exist in account database) THEN
        IF( password is correct) THEN
            Return True
        ELSE
            Return False
    ELSE
        Return False
}
```

Public void SetNewPassword(String account, String password)

函式說明：忘記密碼時，將新密碼寫到資料庫中

輸入參數：用戶的帳號和新密碼

輸出參數：無

虛擬碼：

Public void SetNewPassword(String account, String password)

```
{
    Select this account from account database
    Replace old password with new password
}
```


Private boolean IsRepeat(String account)

函式說明：註冊時，判斷將要註冊的帳號，是否有其他人使用過

輸入參數：新用戶的帳號

輸出參數：是否有帳號重複

虛擬碼：

Private boolean IsRepeat(String account)

```
{  
    Select this account from account database  
    IF(have data in the account database about this account)  
        THEN  
            Return True  
    ELSE  
        Return False  
}
```

★日誌資料庫管理(Log Database Management)

-說明：帳號負責掌管日誌的資料庫

-輸入：記錄帳號登入、登出以及重設密碼的時間

-處理：記錄每個帳號的登入、登出以及重設密碼的時間

-輸出：無

-類別函式：

Public void SaveLogInLog(String account)

函式說明：將要登入的帳號和目前時間存到資料庫中

輸入參數：用戶帳號

輸出參數：無

虛擬碼：

Public void SaveLogInLog(String account)

```
{  
    Time NowTime = GetNowTime();  
    Write this account and NowTime into Log database  
}
```

Public boolean SaveLogoutLog(String account)

函式說明：將要登出的帳號和目前時間存到資料庫中

輸入參數：用戶帳號

輸出參數：無

虛擬碼：

Public boolean SaveLogoutLog(String account)

```
{  
    Time NowTime = GetNowTime();  
    Write this account and NowTime into Log database  
}
```

Public void SaveResetPasswordLog(String account)

函式說明：將要重設密碼的帳號和目前時間存到資料庫中

輸入參數：用戶帳號

輸出參數：無

虛擬碼：

Public void SaveResetPasswordLog (String account)

```
{  
    Time NowTime = GetNowTime();  
    Write this account and NowTime into Log database  
}
```

Private Time GetNowTime()

函式說明：取得目前的時間

輸入參數：無

輸出參數：現在時間

虛擬碼：

Private Time GetNowTime()

```
{  
    Return Now Time
```

}

★申請書資料庫管理(Application Database Management)

-說明：負責掌管申請書的資料庫

-輸入：申請書各個欄位的值，和所需要的申請書

-處理：將每個欄位的值都存起來，和找出對應的申請書

-輸出：回傳所要求要找的申請書

-類別函式：

```
Public boolean SaveBasicData(String basicData, String account)
```

函式說明：將基本資料寫入到指定的帳號裡

輸入參數：基本資料和帳號

輸出參數：儲存資料成功/失敗

虛擬碼：

```
Public boolean SaveBasicData(String basicData, String account)
```

```
{  
    //Because we save all basicData (like name, account,  
    Email...)into "basicData" this variable, so we have to  
    parse"basicData" this variable first.  
    Parse "basicData" this variable into name, account, Email...  
    Select this account from application database  
    IF(the account have data in the application database) THEN  
        Write basicData into Database  
        Return True  
    ELSE  
    {  
        Create a new one  
        Write basicData into Database  
        Return True  
    }  
    Return False  
}
```

Public boolean SaveTranscripts(String transcripts, String account)

函式說明：將成績單寫入到指定的帳號裡

輸入參數：成績單和使用者的帳號

輸出參數：儲存資料成功/失敗

虛擬碼：

Public boolean SaveTranscripts(String transcripts, String account)

```
{  
    Select this account from application database  
    IF(the account have data in the application database) THEN  
        Write transcripts into Database  
    Return True  
    ELSE  
    {  
        Create a new one  
        Write transcripts into Database  
        Return True  
    }  
    Return False  
}
```

Public boolean SaveTeacherEmail(String TeacherEmail, String account)

函式說明：將老師信箱寫入到指定的帳號裡

輸入參數：老師信箱和使用者的帳號

輸出參數：儲存資料成功/失敗

虛擬碼：

Public boolean SaveTeacherEmail(String TeacherEmail, String account)

```
{  
    Select this account from application database  
    IF(the account have data in the application database) THEN
```

```

        Write TeacherEmail into Database
        Return True
    ELSE
    {
        Create a new one
        Write TeacherEmail into Database
        Return True
    }
    Return False
}

```

Public boolean SaveRecommendationLetter(String fileName, String account)

函式說明：將推薦信在伺服器中所儲存的檔名寫入到指定的帳號裡

輸入參數：推薦信檔名和使用者的帳號

輸出參數：儲存資料成功/失敗

虛擬碼：

```

Public boolean SaveRecommendationLetter(String fileName, String account)
{
    Select this account from application database
    IF(the account have data in the application database) THEN
        Write fileName into Database
        Return True
    ELSE
    {
        Create a new one
        Write fileName into Database
        Return True
    }
    Return False
}

```

Public boolean SaveSOP(String SOP, String account)

函式說明：將申請的動機寫入到指定的帳號裡

輸入參數：申請動機和使用者的帳號

輸出參數：儲存資料成功/失敗

虛擬碼：

Public boolean SaveSOP(String SOP, String account)

```
{
    Select this account from application database
    IF(the account have data in the application database) THEN
        Write SOP into Database
        Return True
    ELSE
        {
            Create a new one
            Write SOP into Database
            Return True
        }
    Return False
}
```

Public boolean SaveProgramSelection(String programSelection, String account)

函式說明：將申請的動機寫入到指定的帳號裡

輸入參數：申請動機和使用者的帳號

輸出參數：儲存資料成功/失敗

虛擬碼：

Public boolean SaveProgramSelection(String programSelection, String account)

```
{
    Select this account from application database
    IF(the account have data in the application database) THEN
```

```

        Write programSelection into Database
        Return True
    ELSE
    {
        Create a new one
        Write programSelection into Database
        Return True
    }
    Return False
}

```

Public boolean SaveCV(String CV, String account)

函式說明：將自傳寫入到指定的帳號裡

輸入參數：自傳和使用者帳號

輸出參數：儲存資料成功/失敗

虛擬碼：

Public void SaveCV(String CV, String account)

```

{
    Select this account from application database
    IF(the account have data in the application database) THEN
        Write CV into Database
        Return True
    ELSE
    {
        Create a new one
        Write CV into Database
        Return True
    }
    Return False
}

```

Public Stirng GetLastHistory(String account)

函式說明：取得上一次編輯到一半的申請書

輸入參數：使用者帳號

輸出參數：回傳上一次的紀錄

虛擬碼：

Public Stirng GetLastHistory(String account)

```
{  
    Select this account from application database  
    IF(the account have data in the application database) THEN  
        Return all data in database about this account  
    ELSE  
    {  
        Return “You don’t have last history”  
    }  
}
```

Public CurrentApplication [] GetApplication(String school)

函式說明：老師在要審核申請書時，系統會藉由老師的學校來判斷要
抓取哪些申請書

輸入參數：老師所在的學校

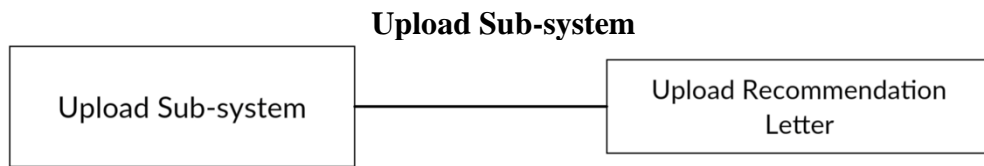
輸出參數：申請書陣列

虛擬碼：

Public CurrentApplication [] GetApplication(String school)

```
{  
    Select program selection is same as school in this database  
    IF(the school have data in the application database) THEN  
        Return all application in database about this school  
    ELSE  
    {  
        Return “No application want to apply this ” + school  
    }  
}
```


}



圖二十七、推薦信上傳子系統

★上傳推薦信(Upload Recommendation Letter)

-說明：提供老師一個上傳推薦信的介面

-輸入：推薦信

-處理：將推薦信儲存到伺服器中，並且將檔名存到資料庫中

-輸出：是否上傳成功

-類別函式：

Public void Upload()

函式說明：提供老師一個上傳推薦信的介面

輸入參數：無

輸出參數：無

虛擬碼：

Public void Upload()

```
{  
    Open upload file window  
    Wait for user choose file  
    IF(file size is smaller than 5MB)THEN  
        Save the file to the server  
    ELSE  
        Ask user to reselect file  
        SaveRecommendationLetter(String fileName)  
        //This method to record where the file be saved  
}
```

Public void SaveRecommendationLetter(String fileName)

函式說明：儲存推薦信的檔案名稱和檔案位置到資料庫中

輸入參數：檔案名稱

輸出參數：無

虛擬碼：

```
Public void SaveRecommendationLetter(String fileName)
{
    FileName = fileName;
    FileAddress = The location where the file
    Repeated calls "SaveToDB()" until the information is saved in database
}
```

Public boolean SaveToDB()

函式說明：將使用者推薦信存入資料庫中

輸入參數：無

輸出參數：儲存成功/儲存失敗

虛擬碼：

```
Public boolean SaveToDB()
{
    IF(AccountDB::SaveRecommendationLetter(filename, account)) THEN
        Return True
    ELSE
        Return False
}
```

-操作：

提供給教授一個上傳申請書的介面

- 使用「選擇檔案」Button 開起檔案選擇視窗。
- 使用「上傳」Button 將所選擇的檔案傳送到伺服器上。
- 使用「取消」Button 來放棄上傳動作。

3.3 執行概念

如圖四～圖十二所示

3.4 需求追溯

OOA 需求摘要	OOD 相關類別
account registration	- Class RegisterAccount
login/logout	- Class LogIn - Class LogOut
program selection	- Class FillOutData - Class CurrentProgramSelection
filling out basic information	- Class FillOutData - Class CurrentBasicData
recommenders' E-mails	- Class FillOutData - Class CurrentTeacherEmail
uploading of scanned transcripts	- Class FillOutData - Class CurrentTranscripts
CV(Curriculum Vitae)	- Class FillOutData - Class CurrentCV
SOP(Statement of Purpose)	- Class FillOutData - Class CurrentSOP

四、軟體介面設計描述

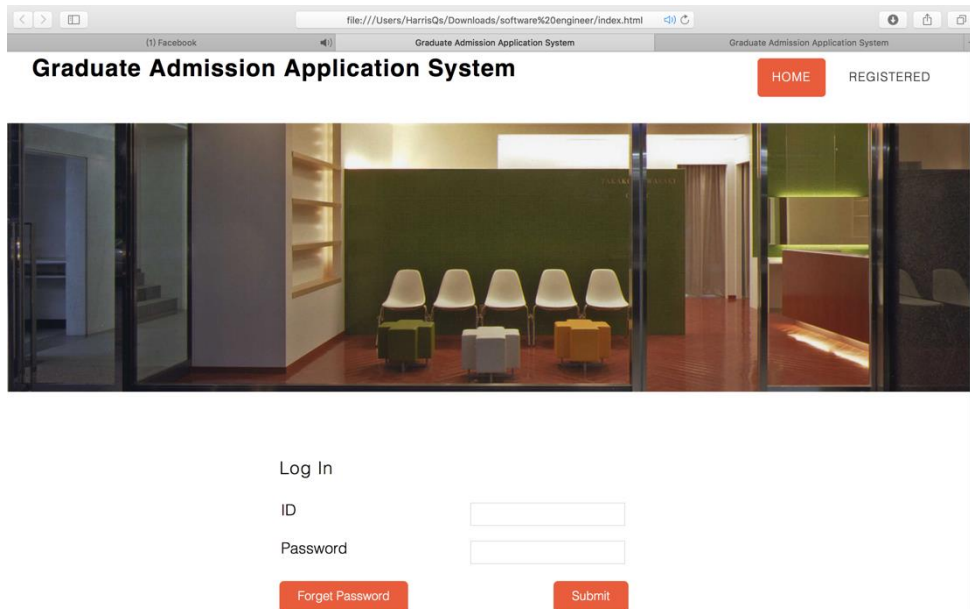
4.1 介面設計概述

本節主要描述主要控管軟體之外部介面、系統之函式介面與系統之物件介面等三部分。

4.2 外部介面定義

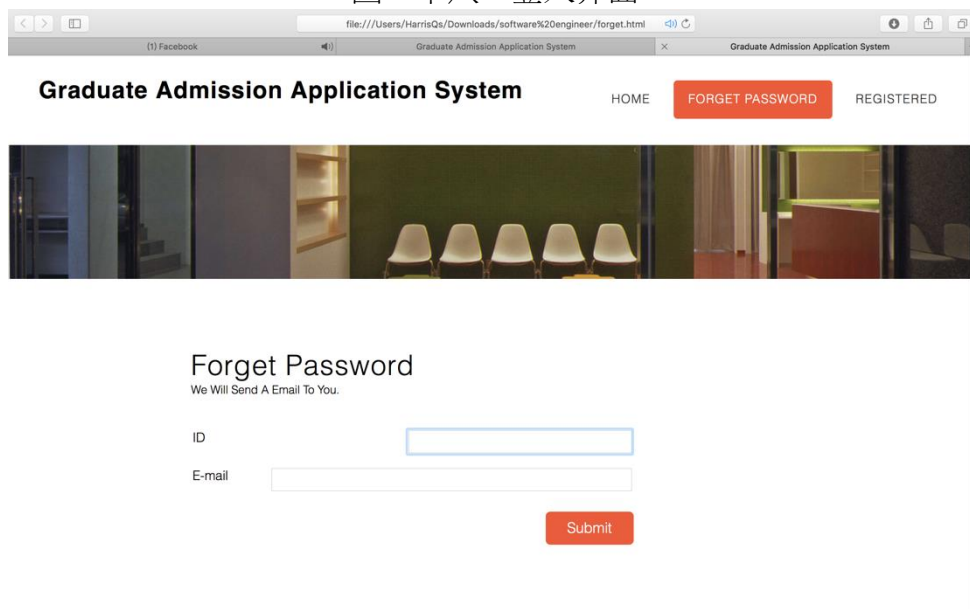
- 系統平台:
 - 作業系統：支援任何常見到的系統,例如:Windows 7 / OS X / Linux。
 - 瀏覽器：支援大部分常見到的瀏覽器,例如:Internet Explore、Chrome、Safari。
 - 硬體平台: 本軟體設定於PC 之硬體平台上執行，且該硬體具備上網的功能。
- 主畫面

- 本軟體之視窗規劃如下圖所示
 - ◆ 登入介面
 - ◆ 忘記密碼介面
 - ◆ 註冊介面
- 網頁標題名稱設為 : Graduate Admission Application System
- 本軟體執行於PC環境: 螢幕解析度不限制



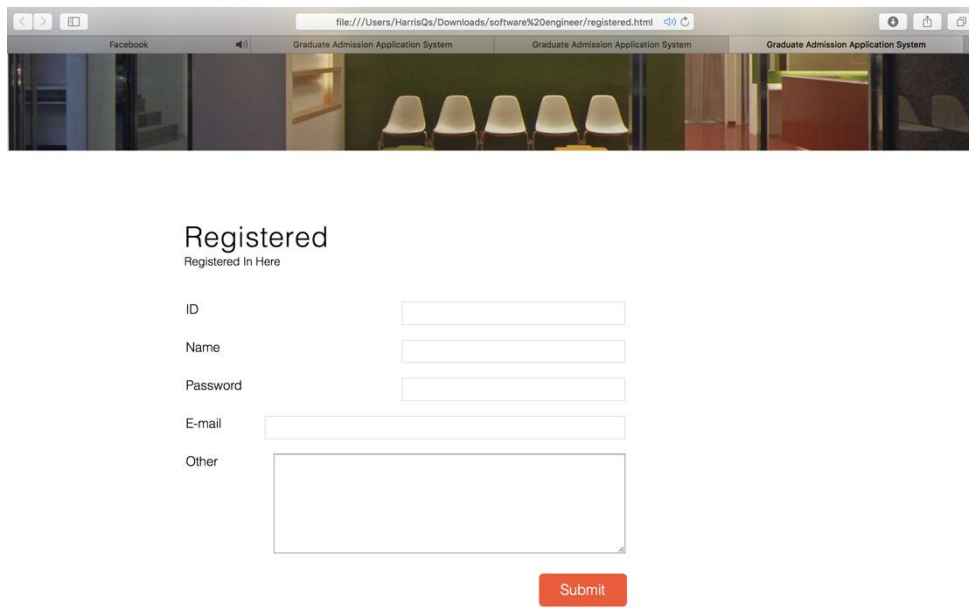
The screenshot shows a web browser window with the address bar displaying 'file:///Users/HarrisQs/Downloads/software%20Engineer/index.html'. The page title is 'Graduate Admission Application System'. The navigation bar includes a 'HOME' button and a 'REGISTERED' link. Below the navigation bar is a large banner image of a modern interior space with white chairs and a green wall. The main content area is titled 'Log In' and contains two input fields for 'ID' and 'Password'. Below these fields are two buttons: 'Forget Password' and 'Submit'.

圖二十八、登入介面



The screenshot shows a web browser window with the address bar displaying 'file:///Users/HarrisQs/Downloads/software%20Engineer/forget.html'. The page title is 'Graduate Admission Application System'. The navigation bar includes a 'HOME' button, a 'FORGET PASSWORD' button, and a 'REGISTERED' link. Below the navigation bar is a large banner image of a modern interior space with white chairs and a green wall. The main content area is titled 'Forget Password' and includes the text 'We Will Send A Email To You.' Below this text are two input fields for 'ID' and 'E-mail'. Below these fields is a 'Submit' button.

圖二十九、忘記密碼介面



圖三十、註冊介面

- 軟體的系統選單
- 本軟體主要為學生研究所網路申請入學的網頁。
- 主程式選單主要分為：
 - ◆ 網頁登入、登出，忘記密碼
 - ◆ 線上填寫申請書、上傳檔案
 - ◆ 審查申請書

4.3 函式介面定義表

Class (RegisterAccount)

- Void RegistNewAccount(String account, String password,String Email, String name, String school, String department)
- Void SendVerificationLetter(String Email)
- Void SetAccount(String account)
- Void SetPassword(String password)
- Void SetEmail(String Email)
- Void SetName(String name)
- Void SetSchool(String school)
- Void SetDepartment(String department)

Class (AccountData)

- Void SetAccount(String account)
- Void SetPassword(String password)
- Void SetEmail(String Email)
- Void SetName(String name)
- Void SetSchool(String school)
- Void SetDepartment(String department)
- Boolean SaveToDB()

Class (LogOut)

- Void LogoutSystem(String account)
- Boolean SaveLogoutLog(String account)

Class (LogIn)

- Boolean LoginSystem(String account, String password)

Class (ForgetPassword)

- Boolean ResetPassword(String account, String Email)
- Void SaveResetPasswordLog(String account)
- Void SendEmail(String Email)

Class (FillOutData)

- Void FillApplicationData(String basicData, String transcripts, String TeacherEmail, String SOP, String programSelection, String CV)
- Void SetBasicData(String basicData, String currentAccount)
- Void SetTranscripts(String transcripts, String currentAccount)
- Void SetTeacherEmail(String TeacherEmail, String currentAccount)
- Void SetSOP(String SOP, String currentAccount)
- Void SetProgramSelection(String programSelection, String currentAccount)
- Void SetCV(String CV, String currentAccount)
- Void GetLastHistory(String currentAccount)
- Void SetAccount(String account)

Class (CurrentBasicData)

<ul style="list-style-type: none"> ● Void SetAccount(String account, String currentAccount) ● Void SetEmail(String email, String currentAccount) ● Void SetName(String name, String currentAccount) ● Void SetSchool(String school, String currentAccount) ● Void SetDepartment(String department, String currentAccount) ● Boolean SaveToDB()
Class (CurrentCV)
<ul style="list-style-type: none"> ● Void SetCV(String cV, String currentAccount) ● Boolean SaveToDB()
Class (CurrentProgramSelection)
<ul style="list-style-type: none"> ● Void SetProgramSelection (String programSelection, String currentAccount) ● Boolean SaveToDB()
Class (CurrentTranscripts)
<ul style="list-style-type: none"> ● Void SetTranscripts (String transcripts, String currentAccount) ● Boolean Upload (String fileName) ● Boolean SaveToDB()
Class (CurrentSOP)
<ul style="list-style-type: none"> ● Void SetSOP (String sOP, String currentAccount) ● Boolean SaveToDB()
Class (CurrentTeacherEmail)
<ul style="list-style-type: none"> ● Void SetTeacherEmail (String teacherEmail, String currentAccount) ● Boolean SaveToDB()
● Class (Review)
<ul style="list-style-type: none"> ● Void Review() ● Void ShowThisApplication(String Account) ● Void ShowAllApplication() ● Void ShowReviewFailed() ● Void ShowNotReview() ● Void ShowReviewSuccess()

<ul style="list-style-type: none"> ● Void ChangeApplicationState(int state) ● Void SendEmailToTeacher(String Email)
● Class (CurrentApplication)
<ul style="list-style-type: none"> ● CurrentApplication[] GetApplication(String school) ● String GetTeacherEmail() ● Void SetState(int state) ● Boolean SaveToDB()
Class (AccountDB)
<ul style="list-style-type: none"> ● Boolean AddNewAccount(String account) ● Boolean ValidateAccount(String account, String password) ● Void SetNewPassword(String account, String password) ● Boolean IsRepeat(String account)
Class (LogDB)
<ul style="list-style-type: none"> ● Void SaveLogInLog(String account) ● Boolean SaveLogOutLog(String account) ● Void SaveResetPasswordLog(String account) ● Time GetNowTime()
Class (ApplicationDB)
<ul style="list-style-type: none"> ● Boolean SaveBasicData(String basicData, String account) ● Boolean SaveTranscripts(String transcripts, String account) ● Boolean SaveTeacherEmail(String TeacherEmail, String account) ● Boolean SaveRecommendationLetter(String fileName, String account) ● Boolean SaveSOP(String SOP, String account) ● Boolean SaveProgramSelection(String programSelection, String account) ● Boolean SaveCV(String CV, String account) ● Boolean SaveState(int state, String account) ● Void GetLastHistory(String account) ● CurrentApplication[] GetApplication(String school)
Class (UploadRecommendationLetter)
<ul style="list-style-type: none"> ● Void Upload()

Class (RecommendationLetter)

- Void SaveRecommendationLetter(String fileName)
- Boolean SaveToDB()

4.4 類別介面定義

如圖二十三～圖二十七所示