

Software Design Document

Virtual Room Reservation Assistant

<version 1.0>

DECEMBER 25 , 2021

Group:25

Author:

B10704101 王柏鈞

B10815043 鄭鴻翔

B10815052 洪翊方

B10832030 陳政亮

**Submitted in partial fulfillment
Of the requirements of
CS3025301 Software Engineering**

1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Definition and Acronyms	4
1.4 References	4
1.5 Overview	4
2. System Overview	4
3. System Architecture	5
3.1 Architectural Design	5
3.1.1 Layered architecture	5
3.1.2 System Package Diagram	6
3.2 Decomposition Description	7
3.2.1 Class Diagram	7
3.3 Design Rationale	7
4. Data Design	8
4.1 Data Description	8
4.2 Data Dictionary	8
4.2.1 Local data entity	8
4.2.1.1 room database data dictionary	8
4.2.1.2 appointment data dictionary	9
4.2.2 MySql database entity	10
5. Component Design	15
5.1 System Component Design	15
5.1.1 UIManager	15
5.1.2 MailSender	16
5.1.3 DatabaseManager	17
5.1.4 LocalDataManager	19
5.2 Use Case Component Design	20
5.2.1 註冊	20
5.2.2 登入	21
5.2.3 更新	22
5.2.4 查看預約的會議室	22
5.2.5 查看參與的會議室	23
5.2.6 自動更新	23
5.2.7 預約會議	24
5.2.8 取消會議	24
5.2.9 修改會議	25
6. Human Interface Design	26
6.1 Overview of User Interface	26
6.2 Screen Images	26

6.2.1 登入介面	26
6.2.2 註冊介面	27
6.2.3 首頁	27
6.2.4 選擇日期介面	28
6.2.5 選擇時段介面	28
6.2.6 輸入會議資訊界面	29
6.2.7 輸入受邀請人介面	30
6.2.8 已預約/已參與介面	31
6.2.9 會議室資訊介面	31
6.3 Screen Objects and Actions	32
6.3.1 登入介面	32
6.3.2 註冊介面	32
6.3.3 首頁	32
6.3.4 選擇日期介面	32
6.3.5 選擇時段介面	32
6.3.6 輸入會議資訊界面	32
6.3.7 輸入受邀請人介面	33
6.3.8 已預約/已參與介面	33
6.3.9 會議室資訊介面	33
7.Requirement Matrix	34

1. Introduction

1.1 Purpose

本文件的目的為提供此軟件(virtual room reservation assistant)建立的細節，並以此協助軟件開發。

設計文檔中將包含軟件以及其部件的敘述與圖形化文檔，包含但不限於use case、sequence、class diagram等以提供軟件設計的信息。

1.2 Scope

本文件提供了在基礎功能設計上，以概念證明在此架構下本項目之可行性。

本設計文件將側重於系統基礎以及關鍵部分，主要描述UI設計與資料庫資料架構部分，並以抽象文檔和圖像介紹設計架構。

1.3 Definition and Acronyms

UI: User Interface

SQL :Structured Query Language

1.4 References

IEEE Standard for Information Technology—Systems Design— Software Design Descriptions

XML Legal Document Utility Software Design Document
Software Design Document (SDD) Template

1.5 Overview

文件包含了 Data、Component、Human Interface的設計和部分細節，另外也包含了系統整體架構。

章節分別為：

1. Introduction
2. System Overview
3. System Architecture
4. Data Design
5. Component Design
6. Human Interface Design

2. System Overview

此系統使用Unity進行開發，並以Unity系統為基礎，設計不同面向功能的元件，使用Unity所提供的架構呼叫這些元件，元件包含DatabaseManager, UIManager, LocalDataManager, MailSender。

3. System Architecture

3.1 Architectural Design

3.1.1 Layered architecture

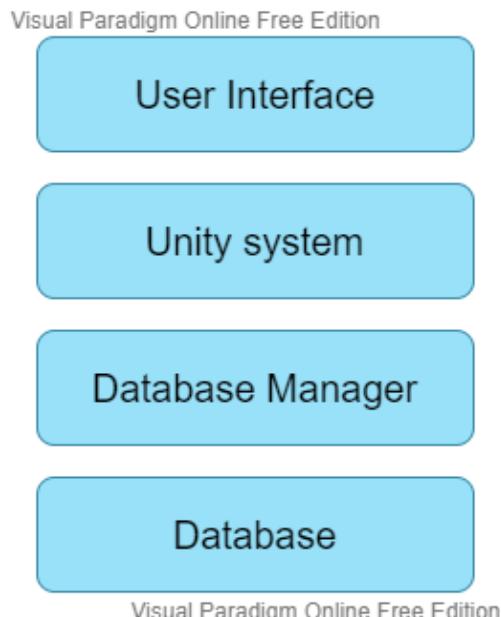
本系統主要分為以下四大層次。

User Interface: 使用者輸入資料、獲取資訊。

Unity system: 為主要運作系統，負責管理介面、本地資料，及檢驗輸入資料正確性。

Database Manager: 外部資料庫的接口。負責獲取、傳送資料。

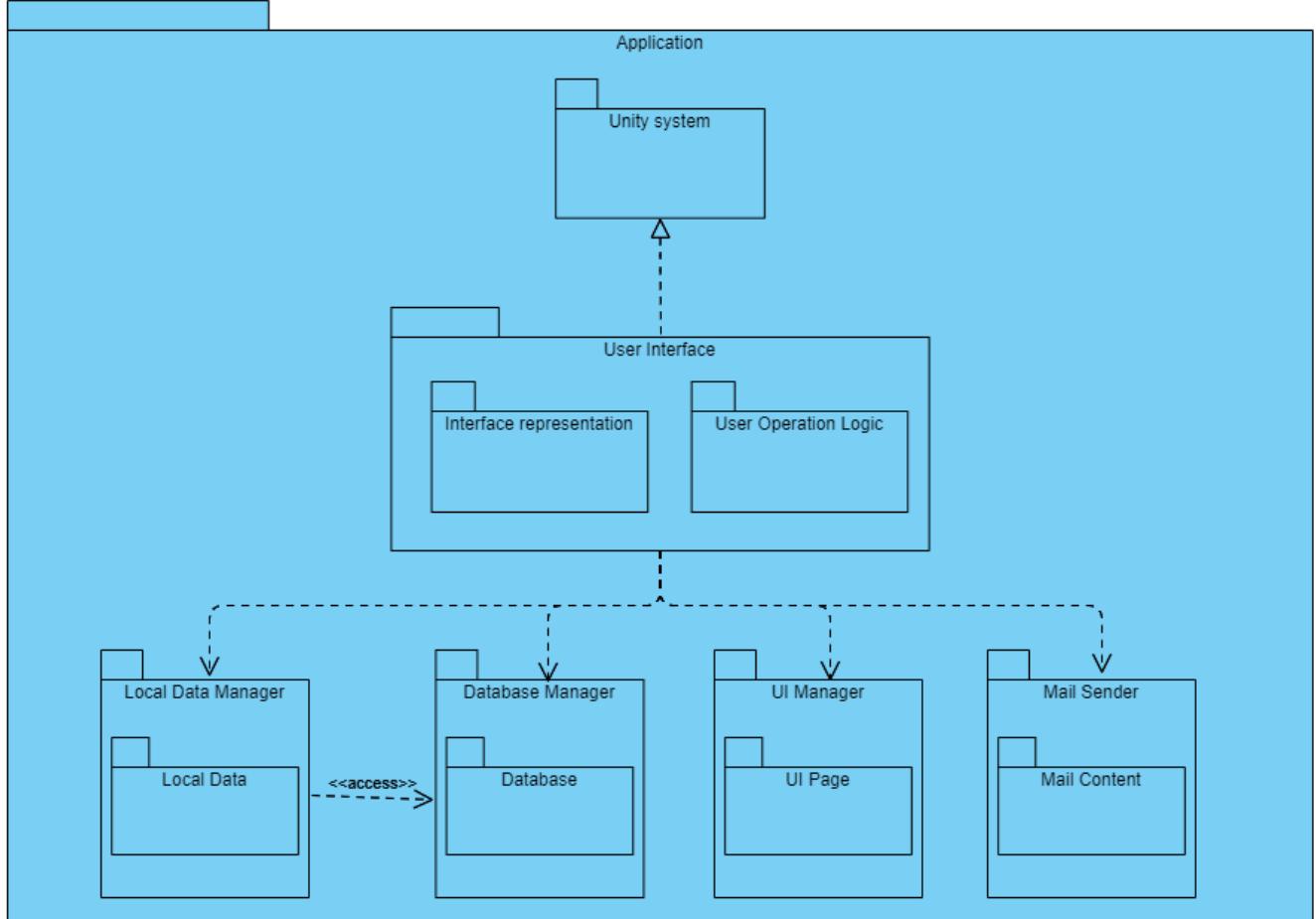
Database: 外部資料庫系統，由MySQL實作，Azure提供伺服器服務



3.1.2 System Package Diagram

此圖用以描述系統內部實作情況。UI畫面、UI事件以Unity提供的API實作。UI事件，如點擊按鈕，將會呼叫元件功能(最底層的Package)

Visual Paradigm Online Free Edition

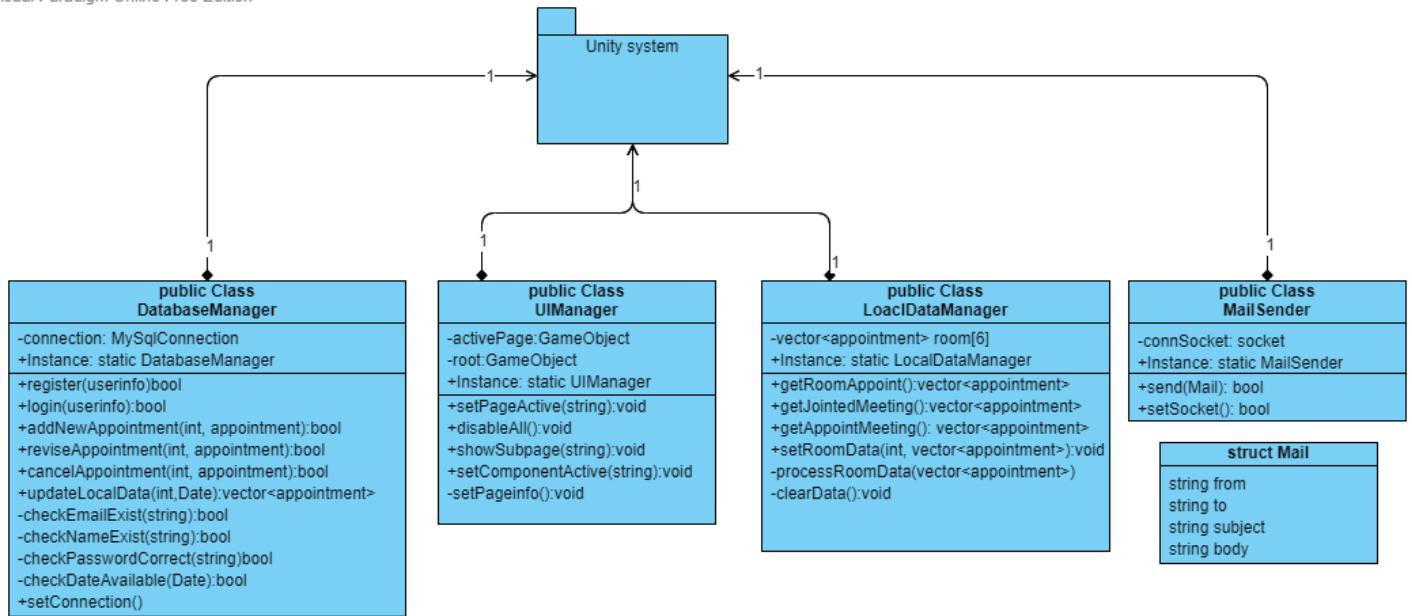


3.2 Decomposition Description

3.2.1 Class Diagram

此處描述系統的關鍵物件。物件的函式的調用由Unity system的UI event呼叫。每個Class中都會自動創建一份Instance, 使Unity System只會對每個Class維護一份Instance。

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

3.3 Design Rationale

本系統為事件驅動架構，即在系統接收到事件訊號才會做出相應動作，在本系通中通常為按鈕事件。這樣做的好處是只需要關鍵元件，並在每種事件動作中，以程式邏輯架構(如迴圈、分支、判斷)組合關鍵元件，就能搭建出所有Use Case。在Unity中已有事件驅動架構，另外也有簡單易學的UI設計架構，因此選為我們的開發平台，但在效能上需要作出割讓。

4. Data Design

4.1 Data Description

分為本地端**database**與雲端**mysql database**, 本地端只儲存請求的會議相關資訊，雲端儲存使用者資訊與所有會議資訊。

4.2 Data Dictionary

4.2.1 Local data entity

由6個**room**組成一個**roomdatabase**, 每個**room**將儲存該會議室編號與**vector<appointment>**(所有從**sql**中抓下來的會議)。每場會議資訊將儲存在**appointment class**中

4.2.1.1 room database data dictionary

roomdatabase	
Room[6]	Vector<appointment>

4.2.1.2 appointment data dictionary

Appointment(會議)		
	name	type
Date(時間)	Day(日期)	String
	Start(開始時間)	String
	End(結束時間)	String
Proposerinfo (發起人資訊)	Name(發起人姓名)	string
	Email(發起人信箱)	String
Meetinginfo (會議資訊)	Name(會議名稱)	String
	Script(會議內容)	String
Inviteeinfo[] (受邀請人資訊)	Name(受邀請人姓名)	String
	Email(受邀請人信箱)	String

4.2.2 MySql database entity

雲端資料庫採用MySQL database, 其中儲存了會議資訊與使用者帳號資訊。

database			
	name	Description	type
Room1(會議室1)	Inviteeinfo1 (受邀請人資訊表格)	Name(受邀請人姓名)	Text
		Email(受邀請人信箱)	Text
	day	格式:yyyy-mm-day	Date
	hour	使用時間, 小時為單位	Int
	Meeting_name	會議名稱	Text
	Proposer	發起人姓名	Text
	Proposer_email	發起人信箱	Text
Room2(會議室2)	Inviteeinfo2 (受邀請人資訊表格)	Name(受邀請人姓名)	Text
		Email(受邀請人信箱)	Text

	day	格式:yyyy-mm-day	Date
	hour	使用時間, 小時為單位	Int
	Meeting_name	會議名稱	Text
	Proposer	發起人姓名	Text
	Proposer_email	發起人信箱	Text
	script	會議內容	Text
Room3(會議室3) (受邀請人資訊表格)	Inviteeinfo3	Name(受邀請人姓名)	Text
		Email(受邀請人信箱)	Text
	day	格式:yyyy-mm-day	Date
	hour	使用時間, 小時為單位	Int
	Meeting_name	會議名稱	Text
	Proposer	發起人姓名	Text
	Proposer_email	發起人信箱	Text

	script	會議內容	Text
Room4(會議室4)	Inviteeinfo4 (受邀請人資訊表格)	Name(受邀請人姓名)	Text
		Email(受邀請人信箱)	Text
	day	格式:yyyy-mm-day	Date
	hour	使用時間, 小時為單位	Int
	Meeting_name	會議名稱	Text
	Proposer	發起人姓名	Text
	Proposer_email	發起人信箱	Text
Room5(會議室5)	Inviteeinfo5 (受邀請人資訊表格)	Name(受邀請人姓名)	Text
		Email(受邀請人信箱)	Text
	day	格式:yyyy-mm-day	Date
	hour	使用時間, 小時為單位	Int

	Meeting_name	會議名稱	Text
	Proposer	發起人姓名	Text
	Proposer_email	發起人信箱	Text
	script	會議內容	Text
Room6(會議室6)	Inviteeinfo6 (受邀請人資訊表格)	Name(受邀請人姓名)	Text
		Email(受邀請人信箱)	Text
	day	格式:yyyy-mm-day	Date
	hour	使用時間，小時為單位	Int
	Meeting_name	會議名稱	Text
	Proposer	發起人姓名	Text
	Proposer_email	發起人信箱	Text
	script	會議內容	Text
userinfo	Name	使用者姓名	Text

	Password	使用者密碼	Text
	Email	使用者信箱	Text
Meetinginfo	Roomnumber	會議室編號	Int
	Meet_id	會議編號	uint

5.Component Design

5.1 System Component Design

5.1.1 UIManager

Class name: UIManager			
Attributes			
Name		Description	
GameObject activePage		目前使用的UI頁面	
GameObject root		所有UI頁面的根物件，用來進行頁面搜尋	
Static UIManager Instance		本類別的實例，用來在Unity系統中全域呼叫	
Method			
Name	Input	Output	Process
setPageActive	string	void	啟用指定UI頁面
disableAll	void	void	使所有頁面失效
showSubpage	string	void	啟用指定UI子頁面

setComponentActive	string	void	啟用指定UI元件
setPageinfo	void	void	設定頁面資訊

5.1.2 MailSender

Class name: MailSender			
Attributes			
Name		Description	
Socket connSocket		連接指定郵箱的Socket位置	
Static MailSender Instance		本類別的實例，用來在Unity系統中全域呼叫	
Method			
Name	Input	Output	Process
send	Mail	bool	寄出電子郵件，並檢查是否有成功
setSocket	void	bool	設定所要連接的Socket

5.1.3 DatabaseManager

Class name: DatabaseManager			
Attributes			
Name		Description	
MysqlConnection connection		連接mysql database	
Static DatabaseManager Instance		本類別的實例，用來在Unity系統中全域呼叫	
Method			
Name	Input	Output	Process
Register	Userinfo	boolean	傳入預註冊之使用者資訊，檢查使否可用
Login	Userinfo	boolean	傳入登入時所輸入的資訊，檢查是否正確
AddNewAppointment	Int, appointment	Boolean	傳入預借用之會議室編號與會議資訊，檢查是否可借用

ReviseAppointment	Int, appointment	boolean	傳入預修改之會議室編號與會議資訊，檢查是否可修改
CancelAppointment	Int, appointment	boolean	傳入預取消之會議室編號與會議資訊，檢查是否取消成功
UpdateLocalData	Int	Vector<appointment>	根據傳入的會議室編號與相關資訊，取得該條件中所有符合的會議資訊
CheckEmailExist	string	boolean	傳入輸入之信箱，檢查信箱是否存在
CheckNameExist	string	boolean	傳入輸入之帳號，檢查帳號名稱是否存在
CheckPasswordCorrect	String, string	boolean	傳入輸入帳號密碼，檢查密碼是否符合此帳號
CheckDateAvailable	Date	boolean	根據傳入的資訊檢查是否可預約
setConnection	void	bool	建立mysql database連接

5.1.4 LocalDataManager

Class name: LocalDataManager			
Attributes			
Name	Description		
Vector<appointment> room[6]	儲存由database載回的資料		
Static LocalDataManager Instance	本類別的實例，用來在Unity系統中全域呼叫		
Method			
Name	Input	Output	Process
getRoomAppoint	int	vector<appointment>	回傳指定會議室所有預約
getJointedMeeting	void	vector<appointment>	回傳使用者參與的會議
getAppointMeeting	void	vector<appointment>	回傳使用者預約的會議
setRoomData	int, vector<appointment>	void	寫入指定會議室資料

processRoomData	vector<appointment>	void	將讀入的會議室資料進行處理，留下必要資訊
clearData	void	void	清除所有資料

5.2 Use Case Component Design

此部分將會描述該如何用上述系統元件來實現Use case, 此部分內容只會涉及到系統元件, 其他部分只會用文字簡述。

5.2.1 註冊

Use case	註冊
Scenario	當未註冊使用者西往取得使用者身分
Trigger	使用者介面註冊按鈕
Brief description	本例子展示了未註冊者如何透過和系統溝通取得使用者身分。
Actor	未註冊者
Preconditions	使用者在註冊介面
Post condition	回到登入和註冊介面
Program Description Language	<pre>//button triggered if(!DatabaseManager.Instance.Register(userInfo)) { UIManager.Instance.showSubpage("regError"); } else { UIManager.Instance.showSubpage("regDone"); }</pre>

5.2.2 登入

Use case	登入
Scenario	使用者透過介面登入系統
Trigger	使用者介面登入按鈕
Brief description	本例子展示了”已註冊使用者”如何登入系統。
Actor	已註冊使用者
Preconditions	使用者在登入介面
Post condition	回到登入介面(0)or進入會議室選擇UI(1)
Program Description Language	//button triggered if(!DatabaseManager.Instance.login(userinfo)) { UIManager.Instance.showSubpage("loginError"); } else { UIManager.Instance.setPageActivate("mainPage"); }

5.2.3 更新

Use case	更新資料
Scenario	使用者透過介面更新資料
Trigger	使用者UI介面更新資料按鈕
Brief description	本例子允許使用者透過更新按鈕獲得資料庫最新的會議資訊。
Actor	已註冊使用者
Preconditions	使用者登入成功
Post condition	資料已更新
Program Description Language	<pre>//button triggered vector<appointment> tmp; for(int idx=0; idx<room_num; idx++) { tmp=DatabaseManager.Instance.updateLocalData(idx); LocalDataManager.Instance.setRoomData(idx, tmp); tmp.clear(); }</pre>

5.2.4 查看預約的會議室

Use case	查看預約的會議室
Scenario	使用者透過介面查看與自己預約的會議
Trigger	使用者介面"已預約"按鈕
Brief description	本例子允許使用者查看所有自己預約的會議資訊
Actor	已註冊使用者
Preconditions	使用者在登入成功
Post condition	進入已預約介面
Program Description Language	<pre>//button triggered vector<appointment> tmp; tmp=LocalDataManager.Instance.getAppointment(); // use "tmp" data update user interface</pre>

5.2.5 查看參與的會議室

Use case	查看參與的會議室
Scenario	使用者透過介面查看與自己有參與的會議
Trigger	使用者介面"已參與"按鈕
Brief description	本例子允許使用者查看所有自己參與的會議資訊
Actor	已註冊使用者
Preconditions	使用者在登入成功
Post condition	進入已參與介面
Program Description Language	<pre>//button triggered vector<appoint> tmp; tmp=LocalDataManager.Instance.getJointedMeeting(); // use "tmp" data update user interface</pre>

5.2.6 自動更新

Use case	更新資料
Scenario	使用者透過介面更新資料
Trigger	計時器
Brief description	以系統的計時器觸發事件獲得資料庫最新的會議資訊。
Actor	已註冊使用者
Preconditions	使用者登入成功
Post condition	資料已更新
Program Description Language	<pre>//timer triggered vector<appointment> tmp; for(int idx=0; idx<room_num; idx++) { tmp=DatabaseManager.Instance.updateLocalData(idx); LocalDataManagr.Instance.setRoomData(idx, tmp); tmp.clear(); }</pre>

5.2.7 預約會議

Use case	預約會議室
Scenario	使用者透過介面預約要使用的會議室
Trigger	使用者介面預約會議按鈕
Brief description	本例子允許使用者預約可以借用的會議室，系統根據時段判斷各會議室是否可以借用，並發送郵件給所有受邀請人。
Actor	已註冊使用者
Preconditions	使用者於預約介面
Post condition	顯示預約結果
Program Description Language	<pre>//button triggered if(!DatabaseManager.Instance.addNewAppointment(room_idx, appointmentData)) { UIManager.Instance.showSubpage("appoint_failed"); } else { MailSender.Instance.send(appointMail); UIManager.Instance.showSubpage("appoint_success"); }</pre>

5.2.8 取消會議

Use case	取消預約
Scenario	使用者需要取消預定錯誤的會議
Trigger	使用者介面取消預約按鈕
Brief description	本例子允許會議發起人取消已預約的會議，僅限於發起人本人建立的所有會議，系統會發送郵件給所有被取消會議的會議參與者和受邀請人。
Actor	會議發起人
Preconditions	發起人於會議室資訊介面中
Post condition	顯示執行結果

Program Description Language	<pre>//button triggered if(!DatabaseManager.Instance.cancelAppointment(room_idx, appointmentData)) { UIManager.Instance.showSubpage("cancel_failed"); } else { MailSender.Instance.send(cancelMail); UIManager.Instance.showSubpage("cancel_success"); }</pre>
------------------------------	--

5.2.9 修改會議

Use case	修改會議資訊
Scenario	使用者希望改變自身發起的會議資訊。
Trigger	使用者介面修改按鈕
Brief description	本例子允許會議發起人更改已預約的會議會議資訊(如: 時段、參與人、會議內容與標題)
Actor	會議發起人
Preconditions	發起人於會議室資訊介面中
Post condition	顯示執行結果
Program Description Language	<pre>//button triggered if(!DatabaseManager.Instance.reviseAppointment(room_idx, appointmentData)) { UIManager.Instance.showSubpage("revise_failed"); } else { MailSender.Instance.send(reviseMail); UIManager.Instance.showSubpage("revise_success"); }</pre>

6. Human Interface Design

6.1 Overview of User Interface

使用者在UI系統上進行註冊、登入、預約會議等操作，而UI系統則給予、獲取資料，並同時驗證輸入資料的正確性，最後提供給外部資料庫，並與外部資料庫進行核對。本地端資料透過"更新"來獲取外部資料庫資料。

6.2 Screen Images

6.2.1 登入介面

The image shows a wireframe of a login interface. At the top left is a header labeled "登入介面". At the top right is a button labeled "結束". Below the header, there are two text input fields: one labeled "名稱" and another labeled "密碼", both with placeholder text "Text Box". Below these fields are two large buttons: a grey "登入" button on the left and a white "註冊" button on the right.

6.2.2 註冊介面

註冊介面

結束

名稱

密碼

Email

6.2.3 首頁

首頁

結束

已預約

已參與

更新

6.2.4 選擇日期介面

選擇日期 結束

March 2017						
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

上一步

6.2.5 選擇時段介面

選擇時段 結束

10	00	show available	<input checked="" type="checkbox"/> Selected
11	00		<input checked="" type="checkbox"/> Selected
12	00		<input checked="" type="checkbox"/> Selected
13	00		<input checked="" type="checkbox"/> Selected

上一步 下一步

6.2.6 輸入會議資訊界面

輸入會議資訊

結束

會議名稱

會議內容

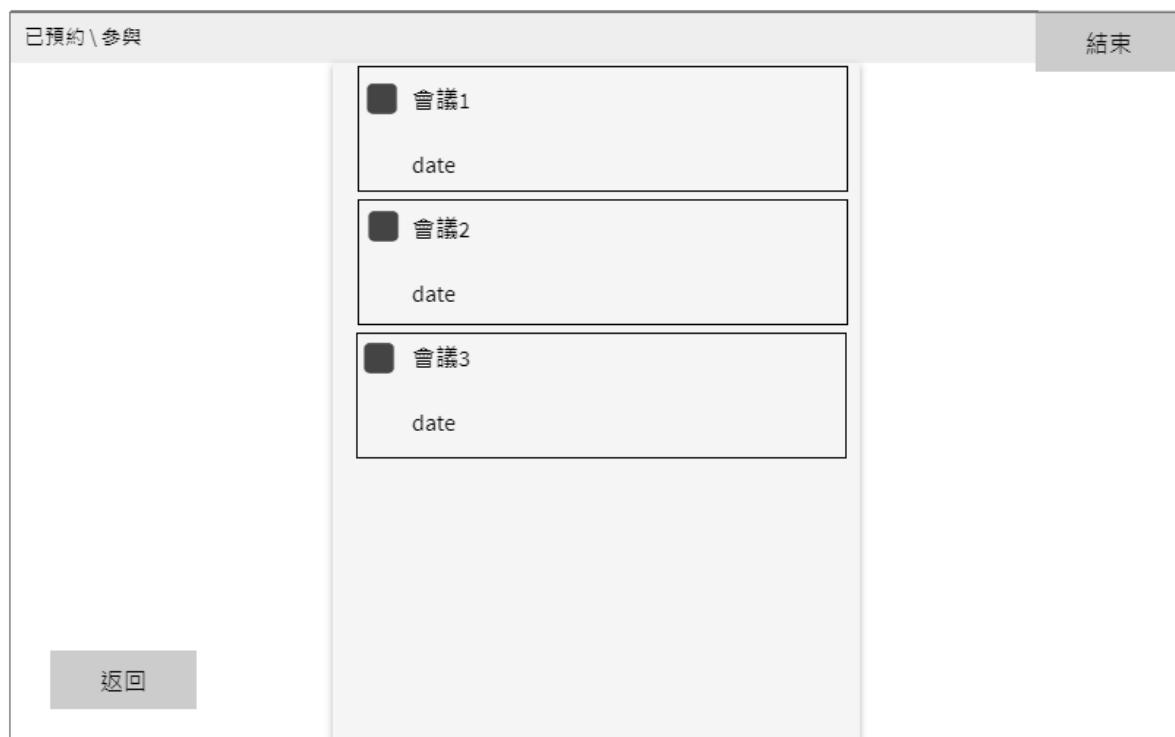
上一步

下一步

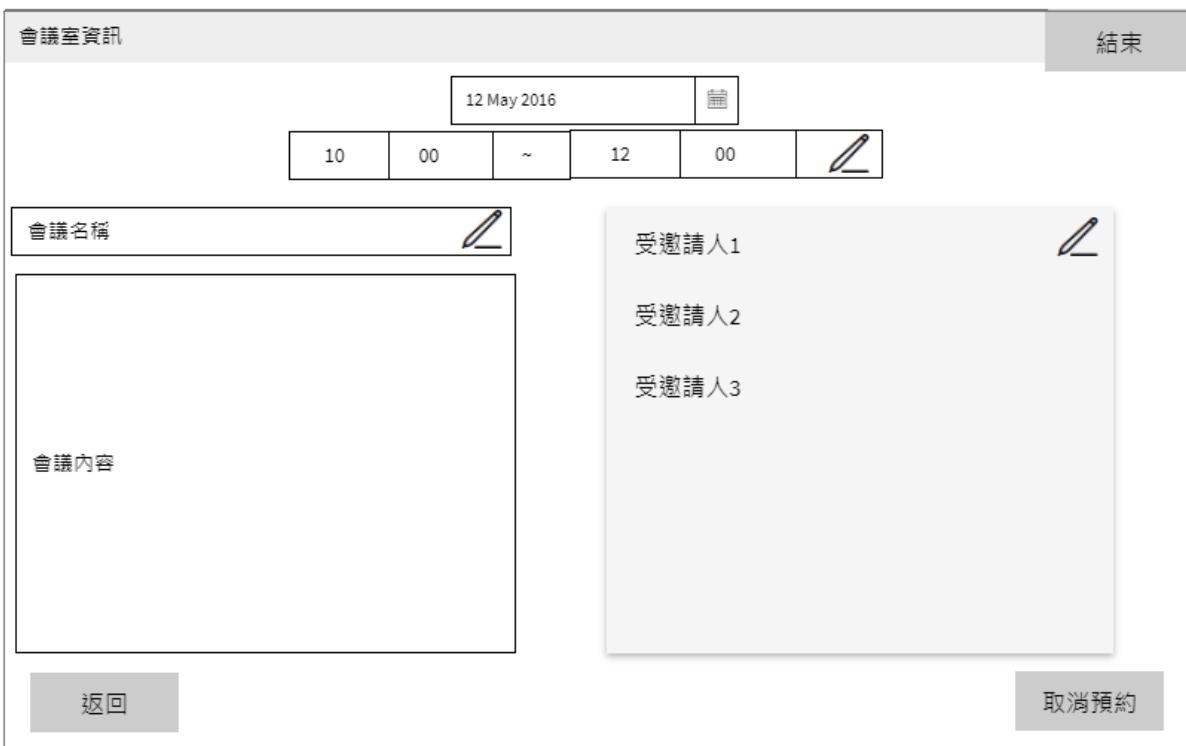
6.2.7 輸入受邀請人介面

The screenshot shows a user interface for entering invitees. At the top left is a text input field labeled "輸入受邀請人". At the top right is a button labeled "結束". Below the input field is a search bar with a magnifying glass icon. A central panel displays three entries, each consisting of a name followed by a close (X) button: "受邀請人1 X", "受邀請人2 X", and "受邀請人3 X". At the bottom left is a "上一步" (Previous Step) button, and at the bottom right is a "預約" (预约) button.

6.2.8 已預約\參與



6.2.9 會議室資訊介面



6.3 Screen Objects and Actions

6.3.1 登入介面

1. 結束按鈕: 按下後會跳出確認\取消介面, 按下確認即關閉程式
2. 名稱\密碼Text Box: 使用者可輸入相應資訊
3. 登入按鈕: 執行"登入"動作
4. 註冊按鈕: 轉移至註冊介面

6.3.2 註冊介面

1. 結束按鈕: 按下後會跳出確認\取消介面, 按下確認即關閉程式
2. 名稱\密碼\Email Text Box: 使用者可輸入相應資訊
3. 確認按鈕: 執行"註冊"動作

6.3.3 首頁

1. 結束按鈕: 按下後會跳出確認\取消介面, 按下確認即關閉程式
2. 會議室按鈕: 移至選擇日期介面
3. 已預約按鈕: 執行"查看預約的會議室"動作；移至已預約介面
4. 已參與按鈕: 執行"查看參與的會議室"動作；移至已參與介面
5. 更新按鈕: 執行更新

6.3.4 選擇日期介面

1. 結束按鈕: 按下後會跳出確認\取消介面, 按下確認即關閉程式
2. <\>按鈕: 選擇 上\下月分
3. 日期按鈕: 選擇日期, 移至選擇時段介面
4. 上一步按鈕: 移至首頁

6.3.5 選擇時段介面

1. 結束按鈕: 按下後會跳出確認\取消介面, 按下確認即關閉程式
2. 可用顯示: 顯示該時段會議室可否借用
3. 勾選: 點擊選擇借用時段, 必須為連續時段
4. scrollbar: 上下移動, 顯示不同時段
5. 上一步按鈕: 移至選擇日期介面
6. 下一步按鈕: 移至 輸入會議資訊介面

6.3.6 輸入會議資訊界面

1. 結束按鈕: 按下後會跳出確認\取消介面, 按下確認即關閉程式
2. 會議名稱 Text Box: 輸入會議名稱, 不可空白
3. 會議內容 Text Box: 輸入會議內容, 不可空白

4. 上一步按鈕: 移至選擇時段介面
5. 下一步按鈕: 移至輸入受邀請人介面

6.3.7 輸入受邀請人介面

1. 結束按鈕: 按下後會跳出確認\取消介面, 按下確認即關閉程式
2. 搜索: 輸入使用者名稱, 按下Enter鍵加入
3. 受邀請人顯示欄: 顯示目前已邀請的人員
4. X按鈕: 刪除已邀請的人員
5. 上一步按鈕: 移至會議資訊介面
6. 預約按鈕: 執行"預約"動作

6.3.8 已預約/已參與介面

1. 結束按鈕: 按下後會跳出確認\取消介面, 按下確認即關閉程式
2. 會議按鈕: 以日期排序, 移至會議室資訊介面。
3. 返回按鈕: 移至首頁

6.3.9 會議室資訊介面

1. 結束按鈕: 按下後會跳出確認\取消介面, 按下確認即關閉程式
2. 編輯按鈕: 按下後移至上述相應資訊編輯介面
3. 取消預約按鈕: 執行"取消預約"動作
4. 返回按鈕: 移至首頁

7.Requirement Matrix

本章節將會使用一個表格將Use case與其所使用的關鍵系統元件關係一一對應，確保每個Use Case有正確執行功能，並且沒有執行不必要功能。

	UIMANAGER MANAGER	LOCALDATA MANAGER	DATABASE MANAGER	MAILSENDER
註冊	✓		✓	
登入	✓		✓	
更新		✓	✓	
查看預約	✓	✓		
查看參與	✓	✓		
自動更新		✓	✓	
預約	✓		✓	✓
取消	✓		✓	✓
修改	✓		✓	✓