

Virtual Room Reservation Assistant

Final Document

Version 1.0
<Annotated Version>

January 20, 2022

Group 14

B10832023 蔣秉錕

B10832037 林軒宇

B10832038 王璽銘

B10832043 張友怡

B10832047 楊奕儒

Submitted in partial fulfillment Of the
requirements of CS3025301
Software Engineering

Table of Contents of the Final Document

Table of Contents of the Final Document	2
Revision History of Final Document	2
Software Requirements Specification	3
Software Architecture Document	25
Software Design Document	46

Revision History of Final Document

Date	Version	Description	Author
2022/01/20	1.0	關於網站型會議室預約系統的軟體 最終文件，初始版本	蔣秉錕、林軒 宇、王璽銘、 張友怡、楊奕 儒

Virtual Room Reservation Assistant Software Requirements Specification

Version 2.0
<Annotated Version>

January 19, 2022

Group 14

B10832023 蔣秉錕

B10832037 林軒宇

B10832038 王璽銘

B10832043 張友怡

B10832047 楊奕儒

Submitted in partial fulfillment Of the
requirements of CS3025301
Software Engineering

Table of Contents of SRS

Table of Contents SRS	4
Revision History of SRS	5
1. Introduction	5
1.1 Purpose	5
1.2 Scope of Project	5
1.3 Glossary	6
1.4 References	6
1.5 Overview of Document	7
2. Overall Description	8
2.1 System Environment	8
2.2 Functional Requirements Definition	9
2.2.1 Guest use case	9
Use case: 登入預約系統	9
Use case: 註冊帳號	10
2.2.2 User use case	11
Use case: 預約會議室	11
Use case: 加入已存在的會議	12
Use case: 更新我的會議資訊	13
Use case: 取消會議	14
Use case: 查詢會議室預約狀態	15
2.3 User Characteristics	16
2.4 Non-Functional Requirements	16
3. Requirement Specifications	17
3.1 External Interface Requirements	17
3.2 Functional Requirements	17
3.2.1 Room Status	17
3.2.2 Account Register	18
3.2.3 Account Login	18
3.2.4 Room Reserve	19
3.2.5 Join Meeting	20
3.2.6 Update Meeting Status	20
3.2.7 Send Notification	21
3.2.8 Cancel a meeting	21
4. Other Non-Functional Requirements	22
4.1 Logical Structure of the Data	22
4.2 Security Requirements	23

Revision History of SRS

Name	Data	Reason For Changes	Version
蔣秉錕、林軒宇、王璽銘、張友怡、楊奕儒	2021/11/09	關於網站型會議室預約系統的軟體需求文件，初始版本	1.0
蔣秉錕、林軒宇、王璽銘、張友怡、楊奕儒	2022/01/19	1. 新增 4.3 Safety Requirements 2. 新增 Revision history 3. 修改 1.4 References 4. 修改 2.2.2 database icon 5. 修改 4.2 Security Requirements	2.0

1. Introduction

1.1 Purpose

The purpose of this document is to introduce a conference rooms reservation website. Functions including registration and login, display of rooms, reservation, reminding of coming up meetings, invitation of meeting members.

1.2 Scope of project

To reserve a meeting room, users need to sign up an account first. After log in, the website interface will provide two main functions, the

calendar interface(in user' s account status)and the room reservation.

You can see arranged meeting information on the calendar interface, and reserve meeting room through room reservation web page. When doing reservation, you need to key in some information about your meeting, like when to start and end, inviting members by email, setting reminder.

1.3 Glossary

Term	Definition
Guest	訪客，尚未登入會議預約網站的操作者。
User	使用者，已登入會議預約網站的操作者。
Database	資料庫，系統儲存、管理所有資料之處。

1.4 References

1. IEEE Recommended Practice for Software Requirements

Specifications。URL：

https://moodle.ntust.edu.tw/pluginfile.php/886496/mod_resource/content/1/IEEE%20Recommended%20Practice%20for%20Software%20Requirements%20Specifications%20830-1998.pdf

2. SRSEExample-webapp。URL：

https://moodle.ntust.edu.tw/pluginfile.php/886498/mod_resource/content/1/SRSEExample-webapp.pdf

[nt/1/SRSExample-webapp.pdf](#)

3. ConnectomeSRS ◦ URL :

https://moodle.ntust.edu.tw/pluginfile.php/886499/mod_resource/content/1/ConnectomeSRS.pdf

[nt/1/ConnectomeSRS.pdf](#)

4. srsTemplateAnnotated ◦ URL :

https://moodle.ntust.edu.tw/pluginfile.php/886500/mod_resource/content/1/srsTemplate-annotated.pdf

[nt/1/srsTemplate-annotated.pdf](#)

1.5 overview of document

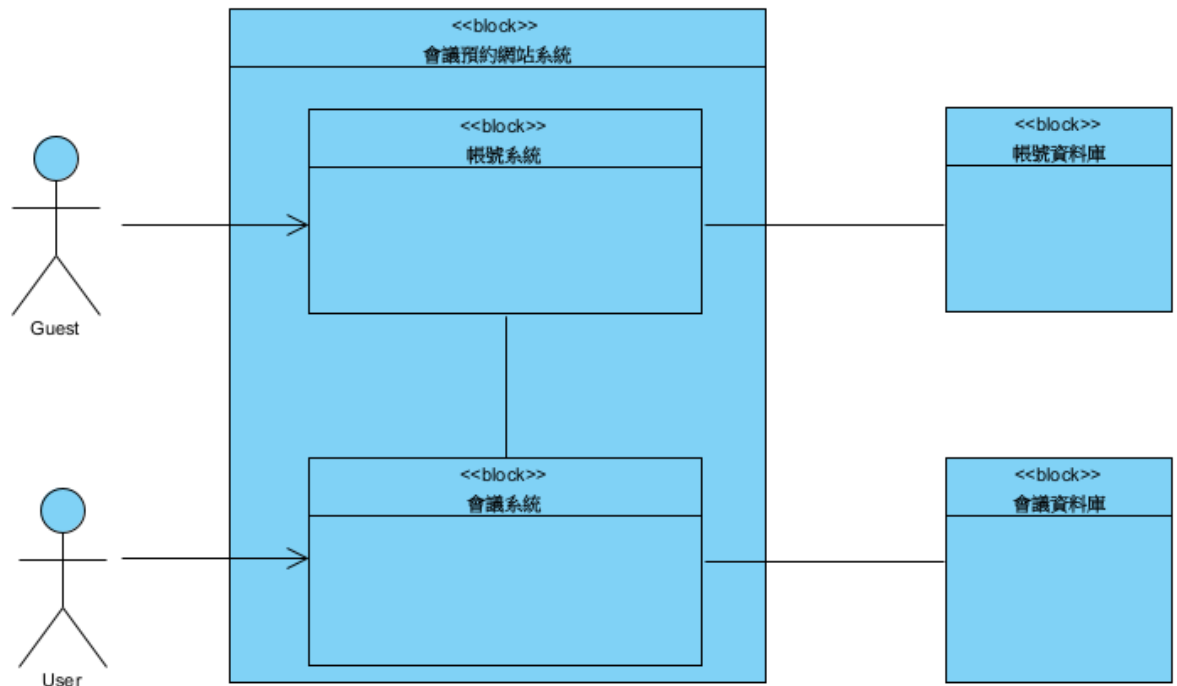
The second chapter, the Overall Description section informally describes System environment, required functions, interactions with user cases, and security, efficiency and some operational constraints concerned.

The third chapter, Requirements Specification section describes system functionality and developing details for the developers.

The fourth chapter, Other Non-Functional Requirements describes the logical Structure of the Data stored in server and security issues.

2. Overall Description

2.1 System Environment



在會議預約網站系統中有兩個系統，分別是帳號系統以及會議系統，還有兩種不同的存取身分，分別是 Guest 和 User。Guest 在還沒登入或註冊前，只能接觸到帳號系統。User 是登入後的身分，可以接觸到會議系統，User 可以查詢會議室的資料或狀況、預約會議室、查詢或取消自己目前擁有的會議。

2.2 Functional Requirements Specification

2.2.1 Guest use case

Guest (訪客) 是尚未登入的用戶在存取會議預約網站系統時的身分。

Use case: 登入預約系統

Diagram:



Brief Description:

用戶要輸入帳號資料以獲得存取會議系統的身分。

Initial Step-By-Step Description:

1. 用戶輸入帳號與密碼在欄位中後送出。
2. 將封包送入後端與資料庫中的資料作匹配。
3. 伺服器回傳結果，若失敗則傳送錯誤訊息，若成功則給予存取會議系統的身分與權限。

Xref: Section 3.2.3 Account Login

Use case: 註冊帳號

Diagram:



Brief Description:

訪客向系統申請新的使用者身分。

Initial Step-By-Step Description:

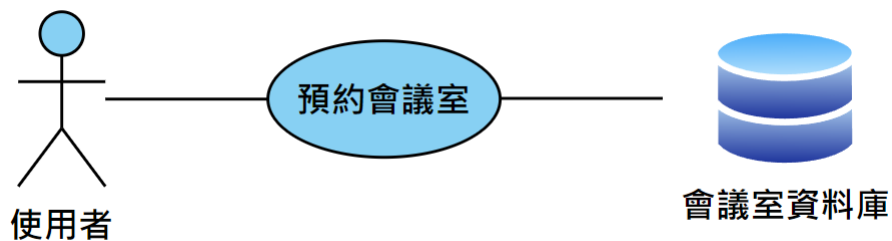
1. 用戶輸入帳號與密碼在欄位中後送出。
2. 將封包送入後端與資料庫中的資料作匹配。
3. 後端回傳結果，失敗則傳送錯誤訊息；成功則加入新的帳號資料進資料庫。

Xref: Section 3.2.2 Account Register

2.2.2 User use case

Use case: 預約會議室

Diagram:



Brief Description:

使用者選擇一間會議室進行預約。

Initial Step-By-Step Description :

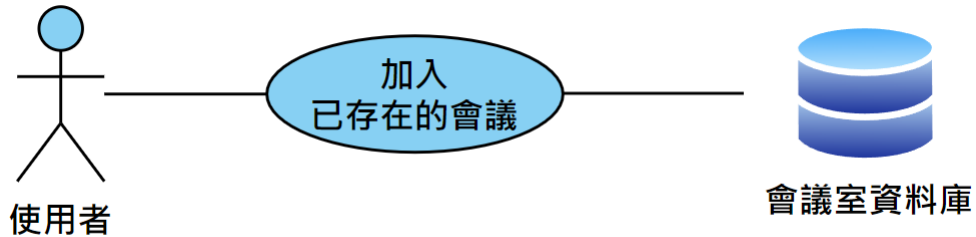
當這個用例被初始化前，使用者已經登入系統，並透過主頁連結進入會議預約頁面。

1. 使用者選擇**會議室型號**以及**時段**按鈕。
2. 使用者填寫會議目的以及備註欄位後，點擊**送出**按鈕。
3. 系統產生預約成功或預約失敗的訊息，並返回會議預約頁面。

Xref: Section 3.2.4, Room Reserve

Use case: 加入已存在的會議

Diagram:



Brief Description :

使用者選擇一間已存在的會議進行加入。

Initial Step-By-Step Description:

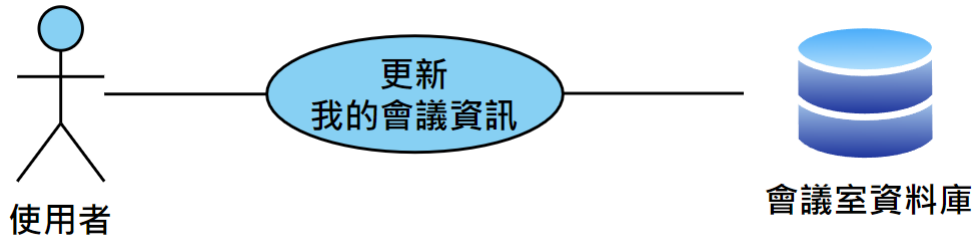
當這個用例被初始化前，使用者已經登入系統，並透過主頁連結進入會議預約頁面。

1. 使用者選擇**已存在會議**按鈕。
2. 系統依序列出該場會議的進階資訊。
3. 使用者確認無誤後，點擊**送出**按鈕。
4. 系統產生加入成功或加入失敗的訊息，並返回會議預約頁面。

Xref: Section 3.2.5, Join Meeting

Use case: 更新我的會議資訊

Diagram:



Brief Description:

使用者選擇一間已加入的會議進行編輯。

Initial Step-By-Step Description:

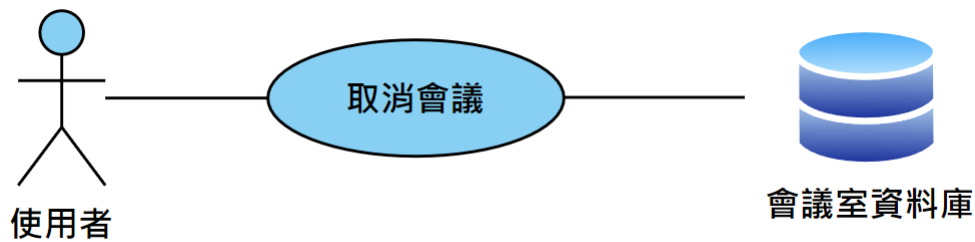
當這個用例被初始化前，使用者已經登入系統，並透過主頁連結進入我的會議頁面。

1. 使用者選擇**已加入會議**按鈕。
2. 系統依序列出該場會議的進階資訊，且資訊可被編輯。
2. 使用者對會議資訊編輯後，選擇**取消**或**送出**按鈕。
3. 如果點擊**取消**按鈕，系統將產生取消編輯的訊息，並返回我的會議頁面。
4. 如果點擊**送出**按鈕，系統將產生成功編輯的訊息，並返回我的會議頁面。

Xref: Section 3.2.6, Update Meeting Status

Use case: 取消會議

Diagram:



Brief Description:

使用者選擇一間已加入的會議進行取消。

Initial Step-By-Step Description:

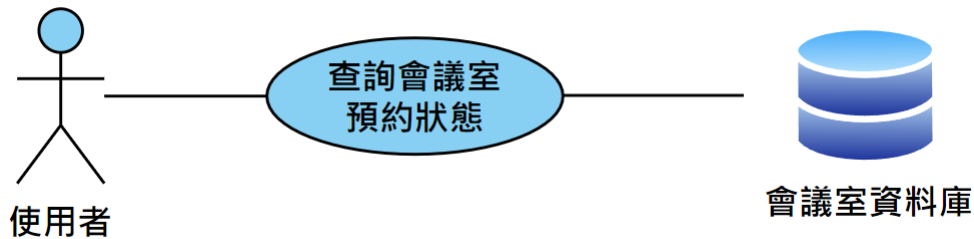
當這個用例被初始化前，使用者已經登入系統，並透過主頁連結進入我的會議頁面。

1. 使用者選擇**已加入會議**按鈕。
2. 系統依序列出該場會議的進階資訊。
3. 使用者點擊**取消會議**按鈕，系統將產生再次確認取消會議的訊息。
3. 如果點擊**取消**按鈕，系統將直接返回我的會議頁面。
4. 如果點擊**確認**按鈕，系統將產生成功取消會議的訊息，並返回我的會議頁面。

Xref: Section 3.2.8, Cancel a meeting

Use case: 查詢會議室狀態

Diagram:



Brief Description:

使用者選擇一間會議室進行預約狀態查詢。

Initial Step-By-Step Description:

當這個用例被初始化前，使用者已經登入系統，並進入主頁。

1. 使用者點擊會議室圖片上方的**會議室型號**按鈕。
2. 系統依序列出該會議室的進階資訊，並以表格呈現目前的預約狀態。
3. 使用者點擊**立即預約**或**返回**按鈕。
3. 如果點擊**立即預約**按鈕，系統將連結進入會議預約頁面。
4. 如果點擊**返回**按鈕，系統將返回主頁。

Xref: Section 3.2.1, Room Status

2.3 User Characteristics

對於本系統的訪客 (Guest) 而言，預期需要具備的知識包括基本的網路知識以及視窗知識，能夠在網頁上使用按鈕與鍵盤輸入功能，並根據彈出式提示完成帳號註冊程序並登入系統。

至於本系統的使用者 (User)，預期需要具備的知識則包括基本的網路知識、視窗知識以及電子郵件知識，能夠在網頁上使用按鈕、滾動條、拖放以及鍵盤輸入，藉此順利操作本系統的會議預約、編輯、查詢以及取消的功能，並透過電子郵件接收會議提醒或取消郵件。

關於這幾頁內容的詳細內容，會在段落 3.2 以後作進一步討論。

2.4 Non-Functional Requirements

本預約系統的伺服器將會運行於個人電腦之上，擁有一般的網路速度，並允許多位使用者同時對系統進行操作。

而軟體方面，預期會整合 Google Calendar API，讓使用者的會議預約狀況能同步呈現於 Google 日曆。除此之外，本系統也將整合 Gmail API，讓使用者能在參與的會議開始前，收到會議提醒或會議取消的通知郵件。

至於安全性議題，我們也將採取對應措施確保會議資訊以及使用者的帳戶資訊得到妥善保護。

關於非功能性需求的詳細內容，會在段落 4 以後作進一步討論。

3 Requirements Specification

3.1 External Interface Requirements

我們使用了 Google calendar 作為個別用戶的時間表，我們想讓用戶能輕易的檢查自己在甚麼時候有會議，而 Google calendar 是廣泛被使用的行事曆軟體，會議成功在創建後將自動加入用戶的 Google 行事曆中，使用者會收到通知並能清楚知道會議在甚麼時候進行。

Send Notification 用例將會在會議創建或者是變更後更新使用者行事曆中的會議時間。

3.2 Functional Requirements

3.2.1 Room Status

Use Case Name	Room Status
XRef	Section 2.2.2 查詢會議室狀態
Trigger	The user login successfully and enter the main page.
Precondition	The user has accessed the login page.
Basic Path	1. The system creates and presents a list of all rooms organized by their status. 2. The user may request to see the full information of a specific room.
Alternative Paths	None.
Postcondition	The requested information has been displayed.
Exception Paths	The user may abandon the operation at any time.

Other	None.
-------	-------

3.2.2 Account Register

Use Case Name	Account Register
XRef	Section 2.2.1 註冊帳號
Trigger	The guest selects to register an account.
Precondition	The guest has accessed the login page.
Basic Path	<ol style="list-style-type: none"> 1. The system presents a blank grid to enter the user information. 2. The guest enters the information and submits the form. 3. The system checks if each block is filled and the user name has not been used then updates the database.
Alternative Paths	<p>If in step 2, either field is blank, the guest is instructed to add an entry.</p> <p>In step 3, if the username has been used, the guest will be instructed to change another username.</p>
Postcondition	The user has been added to the database.
Exception Paths	The guest may abandon the operation anytime.
Other	None.

3.2.3 Account Login

Use Case Name	Account Login
XRef	Section 2.2.1 Account Login
Trigger	The guest selects to login.
Precondition	The guest has accessed the login page.
Basic Path	<ol style="list-style-type: none"> 1. The system presents a blank grid to enter the username and password.

	2. The system checks if the username exists and the password is correct.
Alternative Paths	In step 2, if the username didn't exist or the password isn't correct, the guest is instructed to enter the right information.
Postcondition	The user enters the main page.
Exception Paths	None.
Other	None.

3.2.4 Room Reserve

Use Case Name	Room Reserve
XRef	Section 2.2.2 預約會議室
Trigger	The user selects to reserve a specific room.
Precondition	The user has accessed the main page.
Basic Path	<ol style="list-style-type: none"> 1. The user chooses a room and requests. 2. The user enters the meeting time and chooses the people who participate in the meeting. 3. The system checks if the room has been occupied during the time interval. 4. The system displays the information of the meeting and requires the user to confirm. 5. The user confirms the reservation and the system updates the database.
Alternative Paths	In step 3, if the room has been occupied, the user is instructed to choose another time.
Postcondition	The database has been updated.
Exception Paths	The user may abandon the operation at any time.
Other	None.

3.2.5 Join Meeting

Use Case Name	Join Meeting
XRef	Section 2.2.2 加入已存在的會議
Trigger	The user selects to join an existing meeting.
Precondition	The user has accessed the room status and selects an existing meeting.
Basic Path	<ol style="list-style-type: none">1. The user chooses a meeting to join.2. The system checks if the user is free in the time interval.3. The system updates the database and displays if the request is successful.
Alternative Paths	None.
Postcondition	The database has been updated.
Exception Paths	If the user is not free in the time interval, the use case is abandoned.
Other	None.

3.2.6 Update Meeting Status

Use Case Name	Update Meeting Status
XRef	Section 2.2.2 更新我的會議資訊
Trigger	The user selects to update the information of a meeting.
Precondition	The user has accessed the meeting list.
Basic Path	<ol style="list-style-type: none">1. The user selects a meeting.2. The system display the information of the meeting and the block can be edited.3. The user edits the information and confirms his move.

	4. The system displays the updated information and update the database.
Alternative Paths	In this block, if the user chooses to cancels the meeting and confirm his move, the meeting will be cancelled.
Postcondition	The database has been updated.
Exception Paths	The user may abandon the operation at any time.
Other	None.

3.2.7 Send Notification

Use Case Name	Send Notification
XRef	None.
Trigger	When a meeting is created succesfully.
Precondition	The user reserves a room and creates a meeting.
Basic Path	<ol style="list-style-type: none"> 1. The system searches each participant and sends an email to their email address. 2. The system adds an activity in each participant' s calendar.
Alternative Paths	None.
Postcondition	The notification has been sent.
Exception Paths	None.
Other	None.

3.2.8 Cancel a meeting

Use Case Name	Cancel a meeting
XRef	Section 2.2.2 取消會議
Trigger	The user selects to update the information of a meeting.

Precondition	The user has accessed the meeting list.
Basic Path	<ol style="list-style-type: none"> 1. The user selects a meeting. 2. The system display the information of the meeting and the block can be edited. 3. The user decides to cancel the meeting and confirms his move. 4. The system displays the updated information and update the database.
Alternative Paths	None.
Postcondition	The database has been updated.
Exception Paths	The user may abandon the operation at any time.
Other	None.

4. Other Non-Functional Requirements

4.1 Logical Structure of the Data

儲存在伺服器上的帳號資料庫之資料邏輯結構如下所述：

User data Entity

Data Item	Type	Description	Comment
姓名	Text	使用者的人名	
帳號	Text	使用者的電子郵件地址	限定使用 Gmail
密碼	Text	使用者設定的密碼	需要八個字元以上

儲存在伺服器上的會議資料庫之資料邏輯結構如下所述：

Meeting room Entity

Data Item	Type	Description	Comment
型號	Text	會議室的名稱	
容量	Integer	適合容納的最大人數	
配備	Text	描述會議室擁有的配備	

Meeting Entity

Data Item	Type	Description	Comment
會議室	Pointer	Meeting room Entity	
時段	Date	預約的時段	
目的	Text	會議舉辦的目的	
備註	Text	對於本會議的額外描述	
參與者	Pointer	User data Entity	可能有多位使用者參加

4.2 Security Requirements

儲存帳號資料庫以及會議資料庫的伺服器，會擁有自己的安全性以防止資料遭到未經認證的存取與修改，且資料庫每隔一段時間將自動進行備份。

再者，當使用者從網站輸入的帳戶資料傳入後端伺服器後，該資料會先透過 Python Werkzeug 套件執行雜湊，再使用產生的新資料與資料庫溝通。

除此之外，使用者透過瀏覽器將資料傳遞至伺服器的全程，都會透過 HTTPS (HyperText Transfer Protocol Secure) 超文本傳輸安全協定來確保使用者的個人資訊安全。

4.3 Safety Requirements

當作為伺服器的個人電腦之機體溫度超過安全範圍時，伺服器將自動結束服務並關機，避免使用者 / 客戶的生命安全受到威脅。

Virtual Room Reservation Assistant Software Architecture Document

Version 2.0
<Annotated Version>

January 19, 2022

Group 14

B10832023 蔣秉錕

B10832037 林軒宇

B10832038 王璽銘

B10832043 張友怡

B10832047 楊奕儒

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

Table of Contents of SAD

Table of Contents of SAD	26
Revision History of SAD	27
1. Introduction	28
1.1 Purpose	28
1.2 Scope	28
1.3 Definitions, Acronyms and Abbreviations	28
1.4 References	29
1.5 Overview	30
2. Architectural Representation	30
3. Architectural Goals and Constraints	31
4. Use-Case View	31
4.1 Architecturally Significant Use Cases	32
4.1.1 Login	32
4.1.2 Register	32
4.1.3 Room Reserve	33
4.1.4 Join Meeting	33
4.1.5 Update Meeting Status	33
4.1.6 Cancel Meeting	34
4.1.7 Search Room Status	34
5. Logical View	35
5.1 Overview	35
5.2 Architecturally Significant Design Packages	35
5.2.1 Room Reservation web	35
5.2.2 Room Reservation server	36
5.2.3 Room Reservation database	36
6. Process View	36
6.1 Overview	36
6.2 Processes	37
6.2.1 User Application	37
6.2.2 Representation Controller	37
6.2.3 Room Reservation Process	37
6.2.4 Room Reservation Controller	38
6.2.5 Authentication Process	38
6.2.6 Authentication Controller	38
6.2.7 Meeting Room Process	38
6.2.8 Meeting Room Controller	38

6.2.9 Account Process	38
6.2.10 Account Controller	39
6.2.11 Close System Process	39
6.2.12 Close System Controller	39
6.3 Account Register Activities	39
6.4 Room Reservation Activities	40
7. Deployment View	40
7.1 Client Desktop PC (External)	41
7.2 Client Laptop (External)	41
7.3 Client Desktop PC (Internal)	41
7.4 Client Laptop (Internal)	41
7.5 Virtual Room Reservation Assistant Server	41
8. Implementation View	42
8.1 client	42
8.2 Room Reservation Web (Presentation Layer)	42
8.3 Room Reservation Server (Business Layer)	43
8.4 Room Reservation Database (Data Access Layer)	43
8.5 External Google Application	44
9. Size and Performance	44
10. Quality	45

Revision History of SAD

Date	Version	Description	Author
2021/12/16	1.0	關於網站型會議室預約系統的軟體架構文件，初始版本	蔣秉錕、林軒宇、王璽銘、張友怡、楊奕儒
2022/01/19	2.0	1. 修改 1.1 Purpose 2. 修改 5.Logical View 3. 修改 7.Deployment View 4. 修改 8.Implementation View	蔣秉錕、林軒宇、王璽銘、張友怡、楊奕儒

1. Introduction

1.1 Purpose

本文件的目的是介紹會議室預約網頁的系統架構。該系統提供註冊、登入、預約會議室、編輯會議資訊、顯示目前預約狀況、寄送會議邀請、發出會議提醒等功能。此外，在後續章節中，文件將透過多種觀點介紹本系統的架構，進而了解採取此架構的原因、成效以及限制。

1.2 Scope

通過預約會議室網站系統，可以直接從網頁上建立會議邀請，只需在網頁註冊好帳號後，就可以使用其功能，只需要從網站上就可以通知與會者何時有會議，透過行事曆確認時間，並且提供房間預覽功能。

1.3 Definitions, Acronyms and Abbreviations

Term	Definition
Guest	訪客，尚未登入會議預約網站的操作者。
User	使用者，已登入會議預約網站的操作者。
Database	資料庫，系統儲存、管理所有資料之處。
HTTP	超文本傳輸協定，一種用於分佈式、協作式和超媒體訊息系統的應用層協定。
Flask	一種輕量級 Web 應用框架，使用 Python 編寫。

SQL	一種特定目的程式語言，用於管理關聯式資料庫系統。
Calendar API	由 Google 公司提供的一種 API，支援本系統整合 Google 日曆。
Gmail API	由 Google 公司提供的一種 API，支援本系統整合 Google Gmail。

1.4 References

1. Software Architecture Document。URL:

https://moodle.ntust.edu.tw/pluginfile.php/895725/mod_resource/content/1/Artifact_%20Software%20Architecture%20Document.pdf

2. Example: Software Architecture Document。URL:

<https://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm>

3. Software Architecture in Practice。URL:

<https://sites.google.com/site/softwarearchitectureinpractice/home>

4. Analysis & Design: Concepts。URL:

https://sceweb.uhcl.edu/helm/RationalUnifiedProcess/process/workflow/ana_desi/cos_and.htm

5. The 4+1 View Model of architecture。URL:

https://moodle.ntust.edu.tw/pluginfile.php/895727/mod_resource/content/1/4+1_View_Model_of_Architecture.pdf

1.5 Overview

在章節二中，文件將敘述關於本系統架構的呈現方式。

在章節三中，文件將以條列式表達本系統架構的目標以及限制。

在章節四、五中，文件將分別以 use-case view、logical view 敘述本系統架構。

在章節六中，文件將以 process view 敘述本系統架構，並透過活動圖簡單描繪關於「註冊帳號」以及「預約會議室」的工作流程。

在章節七、八中，文件將分別以 deployment view、implementation view 敘述本系統架構。

在章節九中，文件將以條列式表達關於本系統的軟體規模以及效能。

在章節十中，文件將以條列式表達關於本系統的品質。

2. Architectural Representation

文件中透過一系列的觀點介紹本系統的結構，將在以下章節中將依序介紹 use-case view、logical view、process view、deployment view 以及 implementation view，通過 UML 模型的方式介紹。

3. Architectural Goals and Constraints

1. 本會議室預約系統會以客戶端-伺服器架構實踐。客戶端以及伺服器硬體皆為個人電腦。
2. 網頁後端為通過 python flask 框架建立，配合 SQL 資料庫。
3. 需要登入後才可以使用預約功能。
4. 必須確保資料庫不會被未經授權者存取。
5. 需要在網頁登入後才会有會議提醒，郵件通知需要通過 Gmail 發送。
6. 建立帳號不會要求個資，保障隱私且快速。
7. 通過連結至 Gmail 及 Google Calendar，也可以通過 Google 帳戶之功能查看。
8. 所有訪客、使用者都可以透過網際網路或是區域網路訪問本系統。

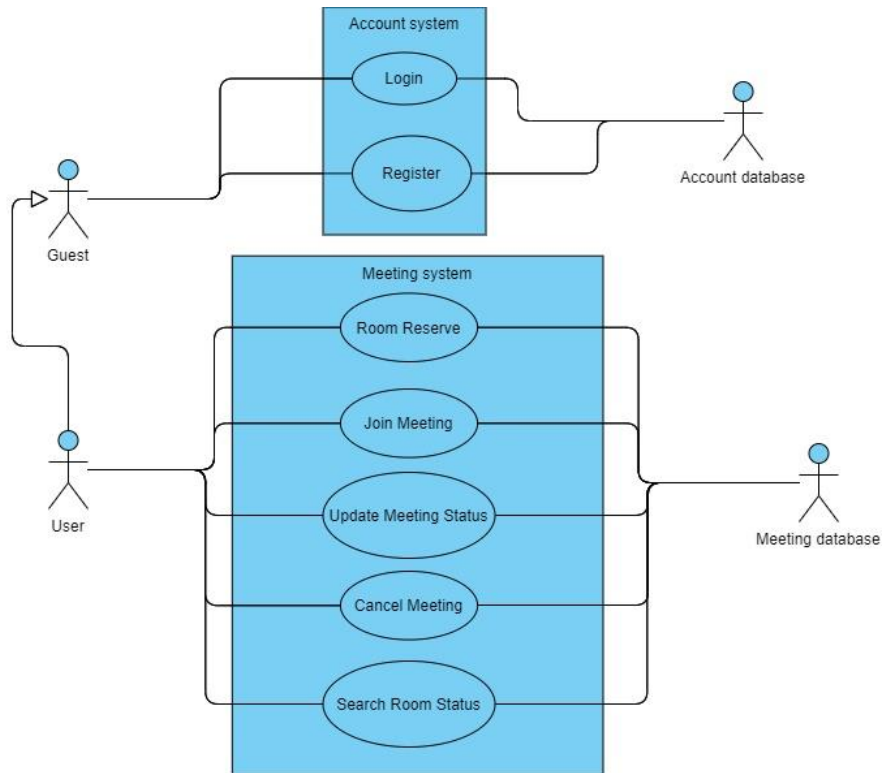
4. Use-Case View

The room reservation system use case are:

- Login
- Register
- Room Reserve
- Join meeting
- Update Meeting Status

- Cancel Meeting
- Search Room Status

4.1 Architecturally-Significant Use Cases



4.1.1 Login

This use case describes how a Guest logs into the account system and become a User. The actor starting this use case is the Guest.

Account database is an actor within this use case.

4.1.2 Register

This use case allows a new Guest to register an account. Guest

should enter their personal information and send it to account database. The actor starting this use case is Guest. Account database is an actor within this use case.

4.1.3 Room Reserve

This use case allows a User choose a room to reserve. User can select ideal time and room and enter description of meeting. Meeting room database will update and return if the reservation is success. The actor starting this use case is User. Meeting database is an actor within this use case.

4.1.4 Join Meeting

This use case allows a User to join an existing meeting. User can check the list of existing meeting to join. Meeting room database will update and return if the act success. The actor starting this use case is User. Meeting database is an actor within this use case.

4.1.5 Update Meeting Status

This use case allows a User modify the description and member

of a meeting this User has involved. Meeting room database will update and return the if the change success. The actor starting this use case is User. Meeting database is an actor within this use case.

4.1.6 Cancel Meeting

This use case describes how a User canceled an existing meeting. User can select an involved meeting and cancel it. Meeting database will update and announce User. The actor starting this use case is User. Meeting database is an actor within this use case.

4.1.7 Search Room Status

This use case allows the User request the status of a room. Meeting database will send information and reservation state to User. The actor starting this use case is User. Meeting database is an actor within this use case.

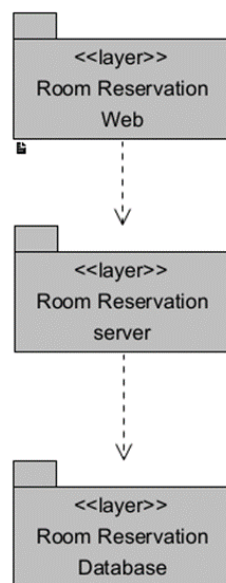
5. Logical View

5.1 Overview

Logical View 著重於 design model 關鍵部分的呈現。此視景將描述軟體拆解為 package 後的架構，以及每個 package 應負的責任。

訂房網站的 Logical view 包含三個不同的 packages : Room Reservation web, Room Reservation server, Room Reservation Database.

5.2 Architecturally Significant Design Packages



5.2.1 Room Reservation web

這一層是擁有 actors 去送出註冊會員、登入會員、查詢會議室、預約會議室、加入已存在會議、更新現有會議資訊、取消現有會議等要求的網站。此外，本分層也負責將請求、資料送往下一層 Room Reservation server layer。

5.2.2 Room Reservation server

這一層擁有處理所有從 Room Reservation web 送過來的請求 class。它可以進行5.1.1 Room Reservation web 層的所有要求，或是對資料庫送出更改或查詢的要求，以及送出會議提醒給所有參與會議的成員。或是當使用者要求網站時，將網站的檔案傳給使用者。

5.2.3 Room Reservation Database

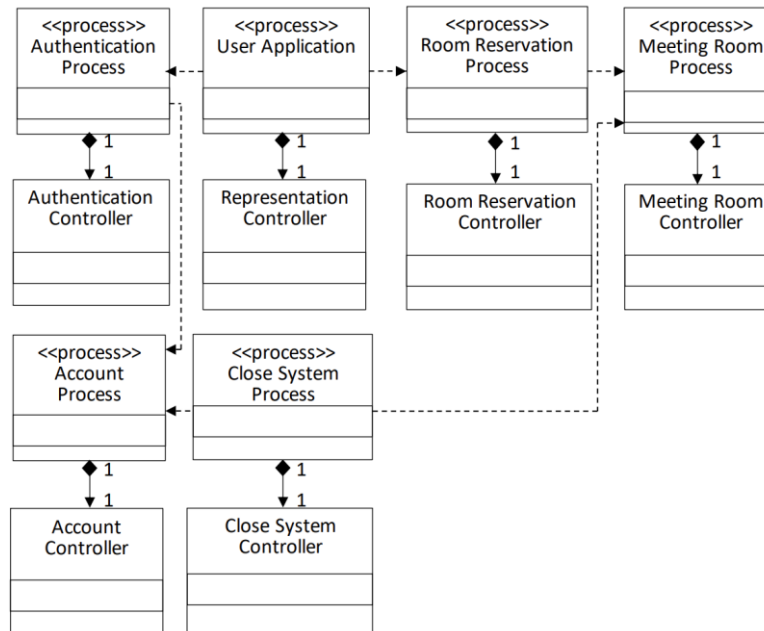
這一層擁有所有房間以及 user 帳號的資料，可以進行查詢、增加、刪除等功能。

6. Process View

6.1 Overview

本章節是關於會議室預約系統架構的 process view 描述。在6.2節中將會敘述本系統中各種程序的結構，以及程序之間的互動。而在6.3、6.4節中，將透過活動圖簡單描繪關於「註冊帳號」以及「預約會議室」的工作流程。

6.2 Processes



6.2.1 User Application

管理本系統使用者的相關功能，包括使用者界面的運作以及與伺服器程序之間的溝通、協調。

再者，對於目前正在使用本系統的每個使用者，都會產生一個此程序的實例。

6.2.2 Representation Controller

管理 User Application 的介面。

6.2.3 Room Reservation Process

對於目前正在使用本系統的每個使用者，都會產生一個此程序的實例。

6.2.4 Room Reservation Controller

支援使用者執行「預約會議室」的使用案例。

此外，如果在目標會議正式開始前，也可以支援使用者執行「更新我的會議資訊」、「加入已存在的會議」以及「取消會議」三個使用案例。

6.2.5 Authentication Process

對於目前正準備登入本系統的每個訪客，都會產生一個此程序的實例。

6.2.6 Authentication Controller

支援訪客執行「註冊帳號」、「登入帳號」兩個使用案例。

6.2.7 Meeting Room Process

在本系統運行時，只會產生唯一一個實例。

6.2.8 Meeting Room Controller

管理會議資料庫的存取，確保會議室預約以及相關資料的一致性。

6.2.9 Account Process

在本系統運行時，只會產生唯一一個實例。

6.2.10 Account Controller

管理帳號資料庫的存取，確保使用者帳號以及相關資料的一致性。

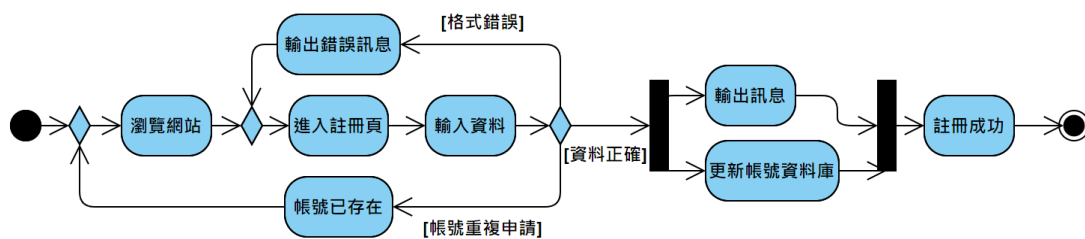
6.2.11 Close System Process

在系統準備關閉（維護、備份）時進行初始化，只會產生唯一一個實例。

6.2.12 Close System Controller

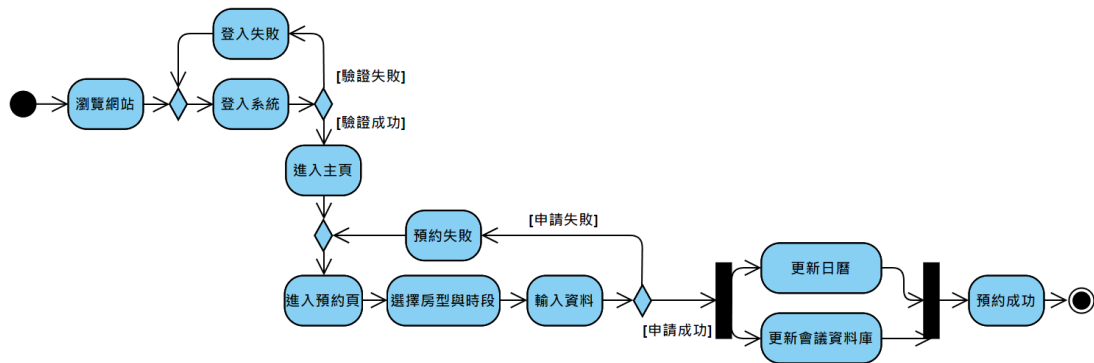
本程序負責與控制帳號資料庫以及會議資料庫的程序進行溝通，確保資料的一致性。

6.3 Account Register Activities



以上活動圖簡單描述一位訪問本會議室預約系統的訪客，經由註冊程序成為系統使用者的流程以及系統的回應。

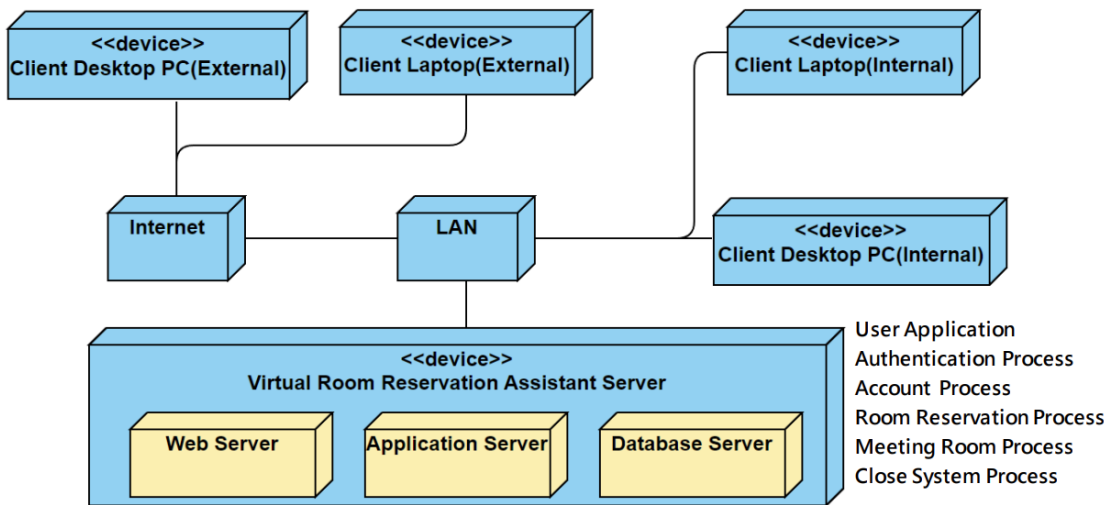
6.4 Room Reservation Activities



以上活動圖簡單描述一位登入本會議室預約系統的使用者，經由會議室預約程序成功預約會議的流程以及系統的回應。

7. Deployment view

Deployment view 著重於軟體開發完成後的佈署階段。此視景將描述軟體系統安裝於硬體時的關聯，其中也包含硬體組件之間的關係。



7.1 Client Desktop PC(External)

訪客或使用者操作外部桌上型個人電腦，並透過網頁瀏覽器以及網際網路使用本會議室預約系統。

7.2 Client Laptop(External)

訪客或使用者操作外部筆記型電腦，並透過網頁瀏覽器以及網際網路使用本會議室預約系統。

7.3 Client Desktop PC(Internal)

訪客或使用者操作本地桌上型個人電腦，並透過網頁瀏覽器以及區域網路使用本會議室預約系統。

7.4 Client Laptop(Internal)

訪客或使用者操作本地筆記型電腦，並透過網頁瀏覽器以及區域網路使用本會議室預約系統。

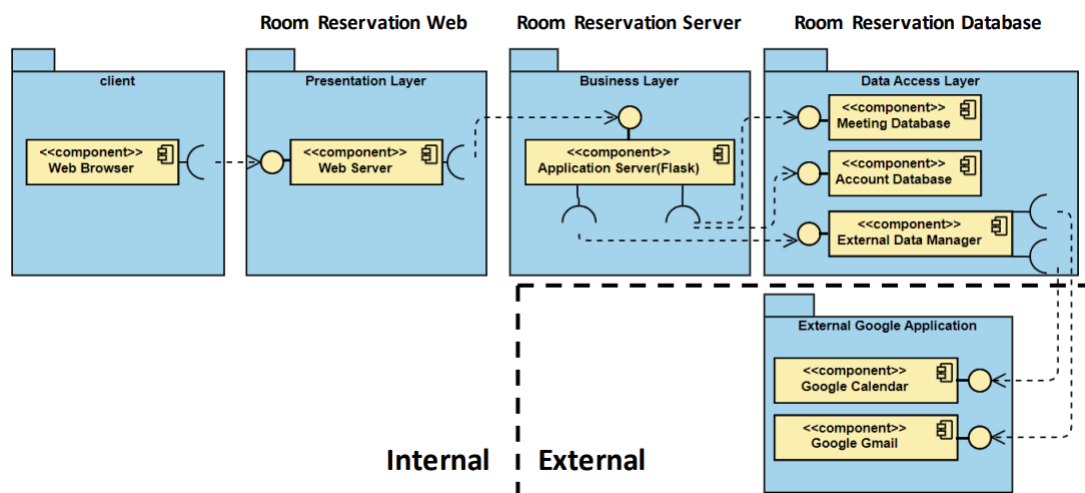
7.5 Virtual Room Reservation Assistant Server

作為伺服器的個人電腦，包含網路伺服器、應用程式伺服器以及資料庫伺服器三個軟體。此設備負責回應客戶端請求，運行本會議室預約系統的所有服務項目，並進行帳號資料庫以及會議資料庫的存取。

此外，所有外部或內部網路的訪客、使用者都可以訪問此伺服器。

8. Implementation View

Implementation View 著重於軟體的管理，將軟體系統分解成多個子系統。在本會議室預約軟體中，子系統被組織為階層式架構，且每一層都有清楚定義的介面與上層子系統溝通。



8.1 client

訪客或使用者操作個人電腦，透過網頁瀏覽器訪問本會議室預約系統。

8.2 Room Reservation Web (Presentation Layer)

該分層擁有的元件為網路伺服器。網路伺服器負責呈現靜態的網路內容 (例如：HTML 頁面、會議室圖像、背景圖像、基本文字描述)，主要功能是回

應來自客戶端瀏覽器的 HTTP 請求。

8.3 Room Reservation Server (Business Layer)

該分層擁有的元件為基於 Python Flask 網路應用框架的應用程式伺服器。

應用程式伺服器負責產生並傳遞動態內容 (例如：處理會議室預約程序、產生客製化網頁)，主要功能是啟用客戶端與伺服器端應用程式碼之間的互動。

8.4 Room Reservation Database (Data Access Layer)

該分層擁有的三個元件分別為帳戶資料庫、會議資料庫以及外部資料管理元件。

帳戶資料庫負責儲存使用者的帳號、密碼以及參與的會議，主要功能是提供上層元件的存取並做出回應。

會議資料庫負責儲存系統現有的會議資料，主要功能是提供上層元件的存取並做出回應，以及在會議結束後自動刪除該筆會議資料。

外部資料管理元件負責與系統外的 Google 應用程式溝通，主要功能是依照事件觸發或時間點，藉由 Gmail API 向使用者的 Google 帳號傳送郵件，或藉由 Calendar API 更新使用者 Google 帳號的日曆。

8.5 External Google Application

Google 公司的系統，能與本系統的外部資料管理元件溝通，主要功能是藉由其所提供的 API 讓本系統使用者的 Google 帳號接收通知或更新資訊。

9. Size and Performance

1. 在任何時間，本會議室預約系統必須要能同時支援 10 個以上的使用者同時進行操作。
2. 在會議室預約網頁下，當使用者按下預約會議的「送出」按鈕後，應該要在五秒鐘內取得伺服器的回應。
3. 在我的會議資訊之網頁下，當使用者按下更新會議資訊的「送出」按鈕後，應該要在四秒鐘內取得伺服器的回應。
4. 在我的會議資訊之網頁下，當使用者按下取消會議的「確認」按鈕後，應該要在四秒鐘內取得伺服器的回應。
5. 當使用者點擊會議室圖片下方的「會議室型號」按鈕以查詢其進階資訊時，應該要在三秒鐘內取得伺服器的回應。
6. 當使用者在目標會議室的介紹視窗下，點擊「立即預約」按鈕以前往會議預約網頁，應該要在三秒鐘內取得伺服器的回應。
7. 伺服器的系統所需硬碟空間應該要小於 2GB，RAM 應該要小於 1GB。

10. Quality

1. 訪客以及使用者應該要能透過 Google Chrome、Mozilla Firefox、Microsoft Edge、Opera、Apple Safari 五款常見的瀏覽器操作本會議室預約系統。
2. 本系統的圖形使用者介面應該要易於操作，讓未經額外訓練的一般使用者也能輕易操作。
3. 在會議室預約系統的網站上應該要提供關於本系統的線上幫助。線上幫助應該要包含本系統所有功能的逐步教學。線上幫助也應該包含專業術語的定義。
4. 平均故障間隔應該要超過 168 個小時。
5. 作為伺服器的個人電腦應該要配備 UPS 不斷電系統，確保異常斷電時的資料庫安全。
6. 自伺服器啟動起每間隔 24 小時，應該要進行一次資料庫備份。
7. 當作為伺服器的個人電腦機體溫度過高時，應該要自動儲存當下狀態並終止程式。
8. 訪客、使用者透過網頁瀏覽器將資料傳遞至伺服器的過程，應該要透過 HTTPS 超文本傳輸安全協定加密封包，確保其個人資料的安全。
9. 儲存使用者帳號的資料庫以及儲存會議資訊的資料庫伺服器，應該要防止資料免於受到未經驗證的存取。

Virtual Room Reservation Assistant Software Design Document

Version 1.0
<Annotated Version>

December 30, 2021

Group 14

B10832023 蔣秉錕

B10832037 林軒宇

B10832038 王璽銘

B10832043 張友怡

B10832047 楊奕儒

Submitted in partial fulfillment of the
requirements of NTUST CS3025301
Software Engineering

Table of Contents of SDD

Table of Contents of SDD	47
Revision History of SDD	48
1. INTRODUCTION	49
1.1 Purpose	49
1.2 Scope	49
1.3 Definitions, Acronyms and Abbreviations	49
1.4 References Material	50
1.5 Overview	51
2. SYSTEM OVERVIEW	52
3. USE CASES	52
3.1 Actors	52
3.1.1 Guest	52
3.1.2 User	52
3.1.3 Account Database	52
3.1.4 Meeting Database	52
3.2 List of Use Cases	53
3.3 Use Case Diagrams	53
3.4 Use Cases	54
3.4.1 Login	54
3.4.2 Register	54
3.4.3 Room Reserve	55
3.4.4 Join Meeting	56
3.4.5 Update Meeting Status	57
3.4.6 Cancel Meeting	57
3.4.7 Search Room Status	58
4. SYSTEM ARCHITECTURE	59
4.1 Architectural Design	59
4.2 Decomposition Description	60
4.2.1 Authentication Management	60
4.2.2 Reservation Management	61
4.2.3 Google Services Management	62
4.2.4 Account Data Management	64
4.2.5 Meeting Data Management	65
4.3 Design Rationale	66
5. Data Design	67
5.1 Data Description	67

5.2 Data Dictionary	67
5.2.1 Account Data	67
5.2.2 Meeting Room Data	68
5.2.3 Meeting Data	68
6. COMPONENT DESIGN	68
6.1 Overview	68
6.2 Authentication Management	69
6.2.1 Login Controller	69
6.2.2 Register	70
6.3 Reservation Management	72
6.3.1 Reservation Controller	72
6.4 Google Services Management	76
6.4.1 Gmail Service Controller	76
6.4.2 Calendar Service Controller	77
6.5 Account Data Management	79
6.5.1 SQL Translator	79
6.6 Meeting Data Management	81
6.6.1 Database Selector	81
6.6.2 SQL Translator	82
7. HUMAN INTERFACE DESIGN	83
7.1 Overview of User Interface	83
7.2 Screen Images	84
7.2.1 Authentication	84
7.2.2 Event calendar	84
7.2.3 Meeting room status	85
7.3 Screen Objects and Actions	85
7.3.1 Login	85
7.3.2 Register	85
7.3.3 Calendar	86
7.3.4 Meeting Room	86
8. REQUIREMENTS MATRIX	87

Revision History of SDD

Date	Version	Description	Author
2021/12/30	1.0	關於網站型會議室預約系統的軟體設計文件，初始版本	蔣秉鋸、林軒宇、王璽銘、張友怡、楊奕儒

1. INTRODUCTION

The document is to provide documentation which will be used to aid in software development by providing the details for how the software should be built, including use case models, sequence diagrams, collaboration models, object behavior models, and the other supporting requirement information.

1.1 Purpose

The purpose of the document is to provide a description of the design of a system, what is to be built and how it is expected to build.

1.2 Scope

This document is focused on the base level system and critical parts of the system, providing a base level of functionality to show feasibility for the system.

1.3 Definition, Acronyms, and Abbreviations

Term	Definition
Guest	訪客，尚未登入會議預約網站的操作者。

User	使用者，已登入會議預約網站的操作者。
Database	資料庫，系統儲存、管理所有資料之處。
HTTP	超文本傳輸協定，一種用於分佈式、協作式和超媒體訊息系統的應用層協定。
Flask	一種輕量級 Web 應用框架，使用 Python 編寫。
SQL	一種特定目的程式語言，用於管理關聯式資料庫系統。
Calendar API	由 Google 公司提供的一種 API，支援本系統整合 Google 日曆。

1.4 Reference Material

1. IEEE Standard for Information Technology—Systems Design— Software

Design Descriptions。URL：

https://moodle.ntust.edu.tw/pluginfile.php/901410/mod_resource/content/1/IEEE_1016-2009%20%28Software%20Design%20Descriptions%29.pdf

2. Software Design Document (SDD) Template。URL：

https://moodle.ntust.edu.tw/pluginfile.php/901415/mod_resource/content/1/SDDTemplate.pdf

3. XML Legal Document Utility Software Design Document。URL：

https://moodle.ntust.edu.tw/pluginfile.php/901414/mod_resource/content/1/example_design_SDD.pdf

4. 三層級架構。URL : <https://www.ibm.com/tw-zh/cloud/learn/three-tier-architecture#toc--Y5a4W75P>

5. Task: Subsystem Design。URL : http://home.iscte-iul.pt/~hro/RUPSmallProjects/core.base_rup/tasks/subsystem_design_5D1BCC0C.html

6. Activity: Subsystem Design。URL : http://www.interface.ru/rational/rup51/process/activity/ac_subds.htm

1.5 Overview

在章節二中，文件將概述關於本系統架構的結構、設計與功能。

在章節三中，文件將以結構化格式敘述本系統的用例。

在章節四中，文件將概述本系統的架構設計以及子系統的分解說明。

在章節五中，文件將敘述關於本系統的資料結構以及採用的資料庫。

在章節六中，文件將敘述各個子系統中的核心類別以及類別所擁有的方法。

在章節七中，文件將配合螢幕圖片敘述本系統的人機介面設計。

在章節八中，文件將以矩陣形式追蹤本系統的需求。

2. SYSTEM OVERVIEW

本會議室預約系統為一網路應用程式，具備登入驗證、多人同時操作、個人行事曆、當前預約狀況顯示以及 Google Calendar 整合能力。

再者，本系統採取分層架構設計，且其客戶端以及伺服器端硬體皆為個人電腦。伺服器端程式主要透過 Python Flask 框架組織，配合 MySQL Server 資料庫儲存資料。

3. USE CASES

3.1 Actors

3.1.1 Guest

訪客，尚未登入會議預約網站的操作者。

3.1.2 User

使用者，已登入會議預約網站的操作者。

3.1.3 Account Database

MySQL Server 上的帳號資料庫，儲存本系統使用者的帳戶資料。

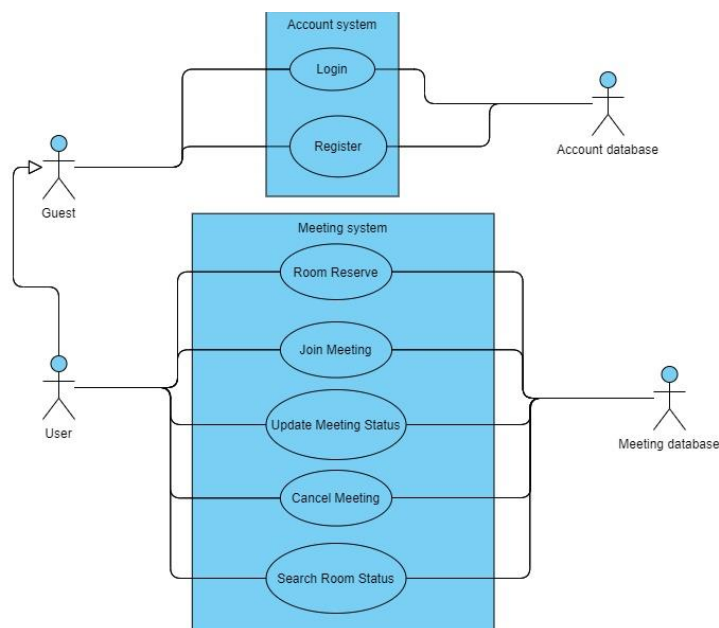
3.1.4 Meeting Database

MySQL Server 上的會議資料庫，儲存本系統的會議資料以及會議室資料。

3.2 List of Use Cases

編號	用例
1	Login
2	Register
3	Room Reserve
4	Join Meeting
5	Update Meeting Status
6	Cancel Meeting
7	Search Room Status

3.3 Use Case Diagrams



3.4 Use Cases

3.4.1 Login

Actors	1. 訪客。 2. 帳號資料庫。
Data	1. 訪客由使用者介面輸入的帳號與密碼。 2. 帳號資料庫儲存的使用者帳號與密碼。
Stimulus	訪客進入登入頁面，由使用者介面輸入帳號與密碼後，點擊「Submit」按鈕。
Response	如果訪客輸入的資料與帳號資料庫儲存的對應資料相同，則將網頁重新導引至使用者個人首頁。若不同，則留在登入頁面。
Comments	無。

3.4.2 Register

Actors	1. 訪客。 2. 帳號資料庫。
Data	1. 訪客由使用者介面輸入的帳號、密碼以及確認密碼。 2. 帳號資料庫儲存的使用者帳號。

Stimulus	訪客進入註冊頁面，由使用者介面輸入帳號、密碼以及確認密碼後，點擊「Submit」按鈕。
Response	如果訪客輸入的資料格式正確，且帳號資料庫沒有存在相同帳號的帳戶，則將網頁重新導引至登入頁面並將新資料插入帳號資料庫。若不同，則輸出錯誤訊息並留在註冊頁面。
Comments	無。

3.4.3 Room Reserve

Actors	<ol style="list-style-type: none"> 1. 使用者。 2. 會議資料庫。
Data	<ol style="list-style-type: none"> 1. 使用者由使用者介面輸入的會議開始時間、會議結束時間、會議目的、備註。 2. 會議資料庫儲存的會議資料與會議室資料。
Stimulus	使用者進入會議預約頁面，由使用者介面輸入會議開始時間、會議結束時間、會議目的以及備註後，點擊「Save」按鈕。
Response	如果使用者輸入的資料與會議資料庫內的會議室資料未產生衝突，則輸出成功訊息並將網頁重新導引至使用者個人首

	頁，且將新資料插入會議資料庫。若不同，則輸出錯誤訊息並留在註冊頁面。
Comments	使用者可以隨時放棄操作。

3.4.4 Join Meeting

Actors	<ol style="list-style-type: none"> 1. 使用者。 2. 會議資料庫。
Data	<ol style="list-style-type: none"> 1. 使用者由使用者介面選擇的已存在會議之 id。 2. 會議資料庫儲存的會議資料與會議室資料。
Stimulus	使用者進入會議預約頁面，由使用者介面輸入已存在的會議之 id，點擊「Save」按鈕。
Response	如果使用者輸入的會議 id 確實存在，則輸出成功訊息並將網頁重新導引至使用者個人首頁，且將新資料插入會議資料庫。若該會議不存在，則輸出錯誤訊息並留在註冊頁面。
Comments	使用者可以隨時放棄操作。

3.4.5 Update Meeting Status

Actors	1. 使用者。 2. 會議資料庫。
Data	1. 使用者由使用者介面選擇的已參與會議之 id。 2. 會議資料庫儲存的會議資料與會議室資料。
Stimulus	使用者進入使用者個人首頁，由使用者介面輸入已參與的會議之 id 並輸入新資料，點擊「Save」按鈕。
Response	如果使用者輸入的會議 id 確實存在且使用者的身分為會議主辦人，則輸出成功訊息並修改會議資料庫儲存的該筆資料。 若該會議不存在或使用者並非會議主辦人，則輸出錯誤訊息。
Comments	使用者可以隨時放棄操作。

3.4.6 Cancel Meeting

Actors	1. 使用者。 2. 會議資料庫。
Data	1. 使用者由使用者介面選擇的已參與會議之 id。 2. 會議資料庫儲存的會議資料與會議室資料。

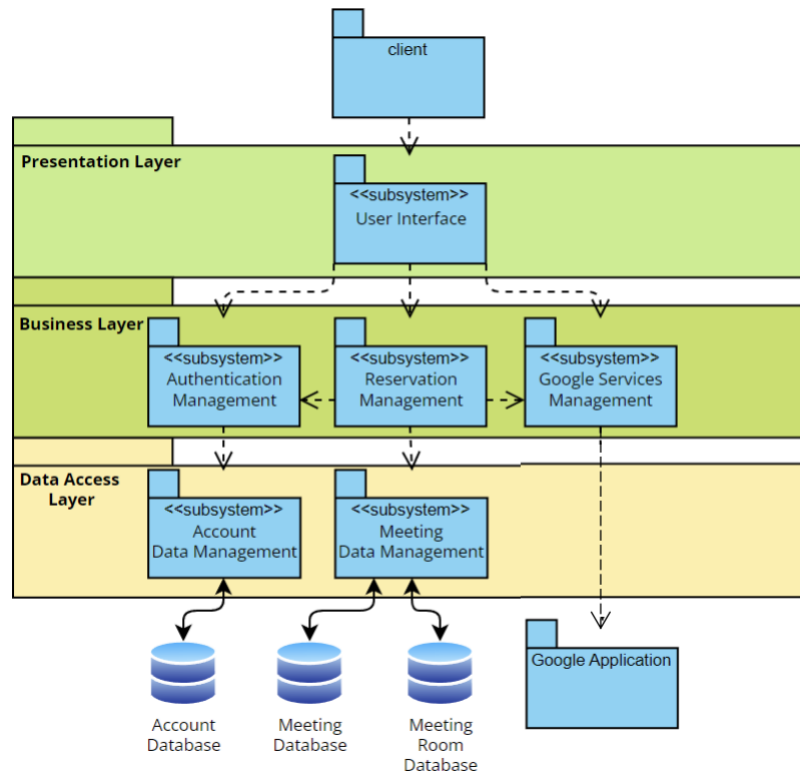
Stimulus	使用者進入使用者個人首頁，由使用者介面輸入已參與的會議之 id，點擊「Delete」按鈕。
Response	<p>如果使用者輸入的會議 id 確實存在且使用者的身分為會議主辦人，則輸出成功訊息並刪除會議資料庫儲存的該筆資料。</p> <p>若該會議不存在或使用者並非會議主辦人，則輸出錯誤訊息。</p>
Comments	使用者可以隨時放棄操作。

3.4.7 Search Room Status

Actors	<ol style="list-style-type: none"> 1. 使用者。 2. 會議資料庫。
Data	<ol style="list-style-type: none"> 1. 使用者由使用者介面選擇的會議室之 id。 2. 會議資料庫儲存的會議資料與會議室資料。
Stimulus	使用者進入使用者個人首頁，由使用者介面輸入會議室之 id，點擊「Search」按鈕。
Response	存取會議室資料庫並輸出該會議室的行事曆。
Comments	無。

4. SYSTEM ARCHITECTURE

4.1 Architectural Design



本系統採用分層架構，主要可分為最上層的 Presentation Layer、第二層的 Business Layer 以及第三層的 Data Access Layer 三個層級。

首先，最上層的 Presentation Layer 擁有的子系統為 User Interface。該子系統內的網路伺服器（Web server）負責呈現靜態的網路內容（例如：HTML 頁面、會議室圖像、背景圖像、基本文字描述）並提供使用者介面，主要功能是回應來自客戶端瀏覽器的 HTTP 請求。

再者，第二層 Business Layer 擁有的子系統為 Authentication Management、Reservation Management 以及 Google Services

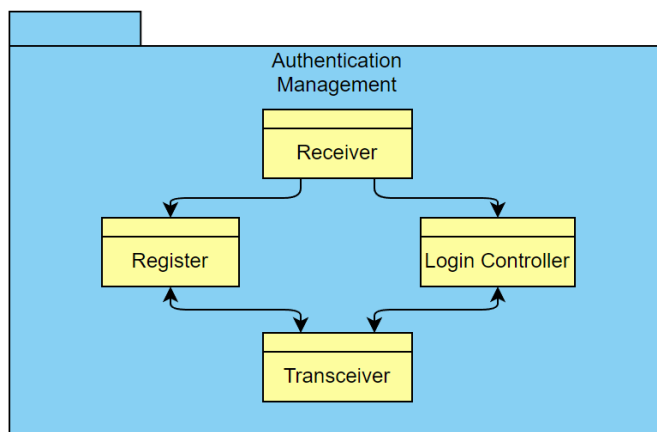
Management。本分層建立於 Python Flask 網路應用框架的應用程式伺服器之上，該伺服器負責產生並傳遞動態內容（例如：處理會議室預約程序、產生客製化網頁），主要功能是啟用客戶端與伺服器端應用程式碼之間的互動。

最後，第三層 Data Access Layer 擁有的子系統為 Account Data Management 與 Meeting Data Management。前者負責連線並存取 Account Database，而 Account Database 是負責儲存使用者帳號、密碼以及參與會議的資料庫。至於後者則是負責連線並存取 Meeting Database，而 Meeting Database 是負責儲存系統現有會議資料以及會議室資料的資料庫。

4.2 Decomposition Description

4.2.1 Authentication Management

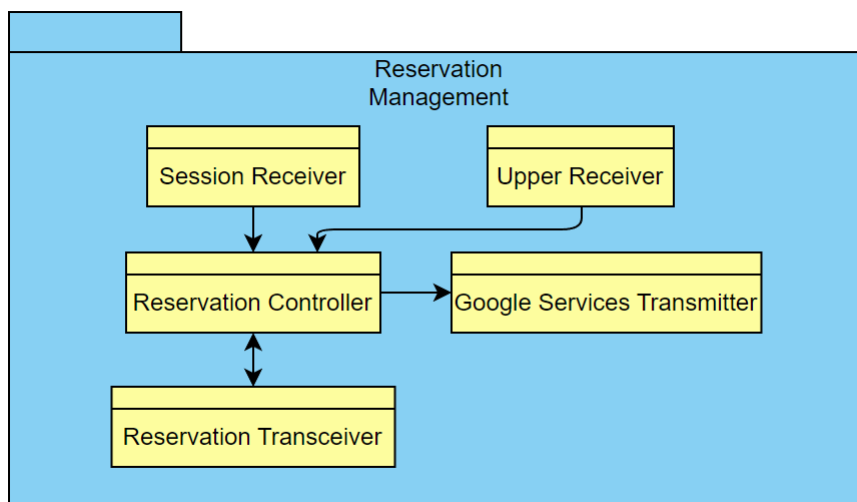
Authentication Management 子系統存在於 Business Layer，其所擁有的重要物件為負責實踐註冊功能的 Register 以及負責實踐登入功能的 Login Controller。Controller。



物件	功能描述
Receiver	接收上層子系統 (User Interface) 所傳遞的資料
Register	處理使用者註冊程序
Login Controller	處理使用者登入程序
Transceiver	接收 Register 以及 Login Controller 物件的資料並傳遞至下層子系統 (Account Data Management) · 或從下層子系統反向傳遞資料

4.2.2 Reservation Management

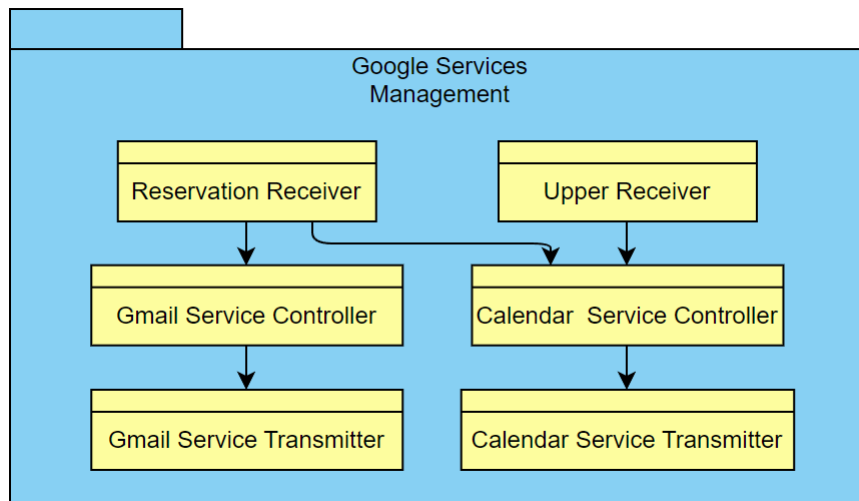
Reservation Management 子系統存在於 Business Layer，其所擁有的重要物件為負責實踐會議預約相關功能的 Reservation Controller。



物件	功能描述
Session Receiver	接收 Authentication Management 子系統所傳遞的使用者登入狀態資料
Upper Receiver	接收上層子系統 (User Interface) 所傳遞的資料
Reservation Controller	處理會議室預約的相關程序
Google Service Transmitter	傳遞資料進入 Google Services Management 子系統
Reservation Transceiver	接收 Reservation Controller 物件的資料並傳遞至下層子系統 (Meeting Data Management)，或從下層子系統反向傳遞資料

4.2.3 Google Services Management

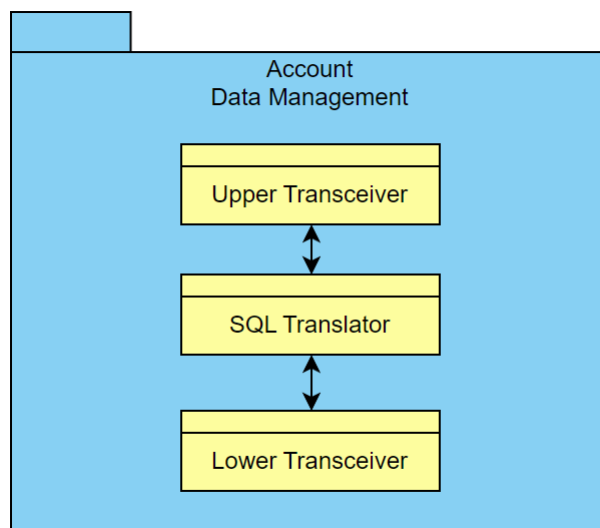
Google Services Management 子系統存在於 Business Layer，其所擁有的重要物件為負責實踐會議通知功能的 Gmail Service Controller 以及負責實踐 Google 日曆功能整合的 Calendar Service Controller。



物件	功能描述
Reservation Receiver	接收 Reservation Management 子系統所傳遞的會議資料
Upper Receiver	接收上層子系統 (User Interface) 所傳遞的資料
Gmail Service Controller	處理 Gmail 發送的程序
Calendar Service Controller	處理 Google Calendar 更新的程序
Gmail Service Transmitter	傳遞 Gmail 的發送請求至 Google 應 用程式
Calendar Service Transmitter	傳遞 Google Calendar 的更新請求 至 Google 應用程式

4.2.4 Account Data Management

Account Data Management 子系統存在於 Data Access Layer，其所擁有的重要物件為負責處理 Python 語言與 SQL 語言之間轉換程序的 SQL Translator。

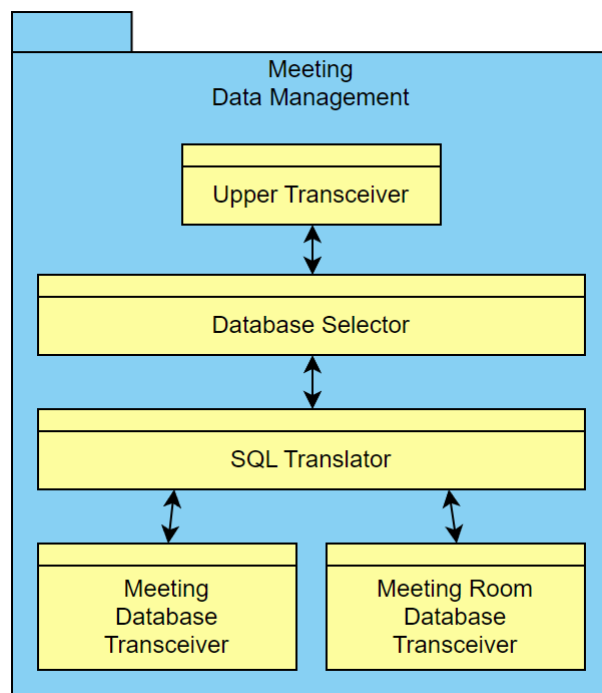


物件	功能描述
Upper Transceiver	接收上層子系統 (Authentication Management) 的資料，或反向傳遞資料
SQL Translator	處理 Python 語言與 SQL 語言之間的轉換程序
Lower Transceiver	接收底層 Account Database 的資

	料，或反向傳遞資料
--	-----------

4.2.5 Meeting Data Management

Meeting Data Management 子系統存在於 Data Access Layer，其所擁有的重要物件為負責實踐資料庫連線選擇功能的 Database Selector 以及負責處理 Python 語言與 SQL 語言之間轉換程序的 SQL Translator。



物件	功能描述
Upper Transceiver	接收上層子系統 (Reservation Management) 的資料，或反向傳遞資料

Database Selector	處理不同資料庫連線設定的程序
SQL Translator	處理 Python 語言與 SQL 語言之間的轉換程序
Meeting Database Transceiver	接收底層 Meeting Database 的資料，或反向傳遞資料
Meeting Room Database Transceiver	接收底層 Meeting Room Database 的資料，或反向傳遞資料

4.3 Design Rationale

首先，由於分層架構的程式邏輯偏向線性，主要的資料流相當單純（由上至下、由下至上），因此較容易理解與開發。而在多人合作開發的情境下，團隊內的各個成員也能專注在負責的分層中作業，不容易受到影響。

再者，就是關注點分離（Separation of concerns）原則。每一層中的組件只處理與該層相關的邏輯，藉此有效分割軟體中不同功能的程式，進而定義明確的元件介面和範圍，提高開發速度、資源重複使用能力以及可測試性。

最後，由於在分層架構中 Presentation Layer 的程式與 Data Access Layer 的程式無法直接進行溝通，因此可藉由完善 Business Layer 設計，防止 SQL 資料庫遭到惡意攻擊，進而提高系統整體的安全性。

5. DATA DESIGN

5.1 Data Description

訂房網站的 Data structure 可以分為三個：

一、帳戶資料 (Account Data)，儲存使用者的 id、帳號、密碼以及使用者目前參與的會議等四項資料。

二、會議室資料 (Meeting Room Data)，儲存會議室的 id、名稱、內部描述以及目前會議列表等四項資料。

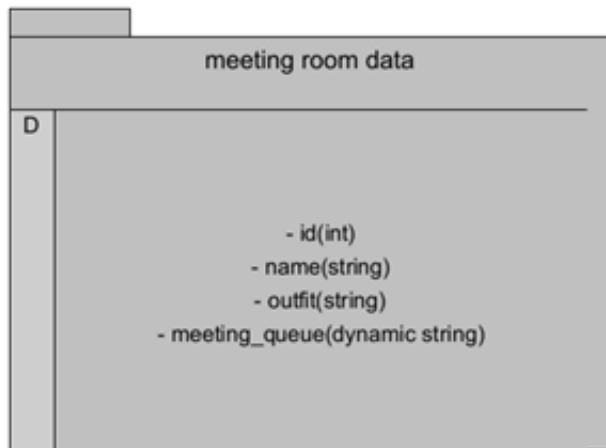
三、會議資料 (Meeting Data)，儲存會議開始時間、會議結束時間、會議主辦人、會議參與者、目的以及備註等六項資料。

5.2 Data Dictionary

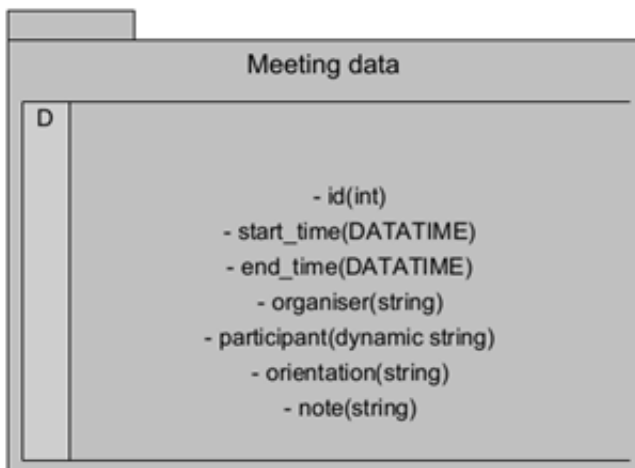
5.2.1 Account Data (帳戶資料)

Account data	
D	<ul style="list-style-type: none">- id(int)- email(string)- password(string)- attended_meeting(dynamic string)- established_time(DATETIME)

5.2.2 Meeting Room Data (會議室資料)



5.2.3 Meeting Data (會議資料)



6. COMPONENT DESIGN

6.1 Overview

本章節將著重於描述 Authentication Management 、 Reservation Management 、 Google Services Management 、 Account Data

Management 與 Meeting Data Management 等五個子系統中的核心類別以及類別所擁有的方法。

6.2 Authentication Management

6.2.1 Login Controller

方法	Login()
描述	獲取使用者輸入的帳戶資料並進行登入驗證
虛擬碼	<pre>sub Login(): 1. 獲取使用者輸入的帳戶資訊 2. if 格式不符: 輸出錯誤訊息 離開函式 end if 3. 獲取資料庫對應的帳戶資訊 4. if 資料正確: 修改使用者登入狀態變數為其 id (0 代表未登入) end if 5. 重新導引網頁進入使用者首頁</pre>

	end sub
--	---------

方法	Logout()
描述	使用者登出系統
虛擬碼	<pre> sub Logout(): 1. if 使用者目前為登入狀態: 修改使用者登入狀態變數為 0 (0 代表未登入) end if 2. 重新導引網頁進入登入頁面 end sub </pre>

6.2.2 Register

方法	Register()
描述	使用者登出系統
虛擬碼	<pre> sub Register(): 1. 獲取使用者輸入的帳戶資訊 2. if 格式錯誤: 輸出錯誤訊息 </pre>

	<p>離開函式</p> <p>end if</p> <p>3. if 密碼 ≠ 檢查密碼:</p> <p>輸出錯誤訊息</p> <p>離開函式</p> <p>end if</p> <p>4. if 資料正確:</p> <p>將帳戶資料向下層子系統傳遞</p> <p>if 帳號已存在:</p> <p>輸出錯誤訊息</p> <p>離開函式</p> <p>end if</p> <p>end if</p> <p>5. 重新導引網頁進入登入頁面</p> <p>end sub</p>
--	--

6.3 Reservation Management

6.3.1 Reservation Controller

方法	CreateMeeting ()
描述	使用者創建一個新會議
虛擬碼	<pre>sub CreateMeeting (): 1. 獲取使用者輸入的會議資訊 2. if 資料不完整: 輸出錯誤訊息 end if 3. 獲取目標會議室的資訊 4. if 該會議室在使用者預約時段已滿: 輸出錯誤訊息 else 將會議資料向下層子系統傳遞 輸出成功訊息 end if 5. 重新導引網頁進入使用者首頁 end sub</pre>

方法	ModifyMeetingInformation()
描述	使用者修改一個已存在會議的資訊
虛擬碼	<pre> sub ModifyMeetingInformation (): 1. 從網頁獲取使用者選擇的會議 id 2. 透過 id 從資料庫獲取會議資訊 3. if 會議不存在: 輸出錯誤訊息 離開函式 end if 4. 從網頁獲取使用者輸入的新會議資訊 5. if 資料正確: 將會議資料向下層子系統傳遞 輸出成功訊息 end if 6. 重新導引網頁進入使用者首頁 end sub </pre>

方法	LeaveMeeting ()
----	-----------------

描述	使用者選擇退出一個會議
虛擬碼	<pre> sub LeaveMeeting (): 1. 從網頁獲取使用者選擇的會議 id 2. 透過 id 從資料庫獲取會議資訊 3. if 會議不存在: 輸出錯誤訊息 離開函式 end if 4. 從網頁獲取使用者二次確認的資料 5. if 使用者取消操作: 離開函式 else if 離開會議的使用者為會議主辦人: 執行 CancelMeeting()函式 離開函式 else 將離開會議的指令向下層子系統傳遞 輸出成功訊息 end if end if </pre>

	<p>end if</p> <p>6. 重新導引網頁進入使用者首頁</p> <p>end sub</p>
--	--

方法	CancelMeeting ()
描述	使用者選擇取消一個會議
虛擬碼	<pre> sub CancelMeeting (): 1. 從網頁獲取使用者選擇的會議 id 2. 透過 id 從資料庫獲取會議資訊 3. if 會議不存在: 輸出錯誤訊息 離開函式 end if 4. if 使用者不是該會議的主辦人: 輸出錯誤訊息 離開函式 end if 5. 從網頁獲取使用者二次確認的資料 </pre>

	<p>6. if 使用者取消操作:</p> <p> 離開函式</p> <p>else</p> <p> 將取消會議的指令向下層子系統傳遞</p> <p> 輸出成功訊息</p> <p>end if</p> <p>7. 重新導引網頁進入使用者首頁</p> <p>end sub</p>
--	--

6.4 Google Services Management

6.4.1 Gmail Service Controller

方法	SendNotification ()
描述	發送會議通知至使用者帳號的 Gmail
虛擬碼	<p>sub SendNotification ():</p> <p> 1. 從 Reservation Management 子系統獲取會議資料</p> <p> 2. if 事件為創建會議:</p> <p> 設定事件類型變數為 0 (0 代表創建會議)</p> <p> else</p>

	<p>設定事件類型變數為 1 (1 代表取消會議)</p> <p>end if</p> <p>3. 將該會議參與者的帳號存入串列</p> <p>4. for 參與者帳號串列:</p> <p> if 事件類型變數為 0:</p> <p> 發送會議邀請通知</p> <p> else</p> <p> 發送會議取消通知</p> <p> end if</p> <p>end for</p> <p>end sub</p>
--	--

6.4.2 Calendar Service Controller

方法	ManualUpdateCalendar ()
描述	當使用者更新個人首頁的行事曆時，同步更新該使用者的 Google Calendar
虛擬碼	sub ManualUpdateCalendar (): <p>1. 從上層子系統 (User Interface) 獲取行事曆事件資料</p>

	<pre> 2. if 事件類型為新增: 設定事件類型變數為 0 (0 代表新增事件) else 設定事件類型變數為 1 (1 代表刪除事件) end if 3. if 事件類型變數為 0: 新增事件至該使用者的 Google Calendar else 從該使用者的 Google Calendar 刪除事件 end if end sub </pre>
--	---

方法	AutoUpdateCalendar ()
描述	當成功建立或刪除一個會議時，更新所有該會議參與者的 Google Calendar
虛擬碼	<pre> sub AutoUpdateCalendar (): 1. 從 Reservation Management 子系統獲取會議資料 2. if 事件為創建會議: </pre>

	<p>設定事件類型變數為 0 (0 代表創建會議)</p> <p>else</p> <p>設定事件類型變數為 1 (1 代表取消會議)</p> <p>end if</p> <p>3. 將該會議參與者的帳號存入串列</p> <p>4. for 參與者帳號串列:</p> <p>if 事件類型變數為 0:</p> <p>新增事件至該使用者的 Google Calendar</p> <p>else</p> <p>從該使用者的 Google Calendar 刪除事件</p> <p>end if</p> <p>end for</p> <p>end sub</p>
--	---

6.5 Account Data Management

6.5.1 SQL Translator

方法	Translate ()
描述	將上層子系統 (Authentication Management) 操作帳號資料

	庫的 Python 程式語言轉換為 SQL 程式語言
虛擬碼	<pre> sub Translate (): 1. 獲取資料庫存取命令 2. if 命令為查詢類型: 設定命令類型變數為 0 (0 代表查詢資料命令) else if 命令為增加類型 設定命令類型變數為 1 (1 代表增加資料命令) else if 命令為修改類型 設定命令類型變數為 2 (2 代表修改資料命令) else 設定命令類型變數為 3 (3 代表刪除資料命令) end if 3. switch(命令類型變數) case 0: 透過 SQLAlchemy 工具包查詢資料庫資料 case 1: 透過 SQLAlchemy 工具包插入資料進資料庫 case 2: 透過 SQLAlchemy 工具包修改資料庫資料 </pre>

	<p>default:</p> <p>透過 SQLAlchemy 工具包刪除資料庫資料</p> <p>end switch</p> <p>end sub</p>
--	--

6.6 Meeting Data Management

6.6.1 Database Selector

方法	SelectDatabase ()
描述	當上層子系統 (Reservation Management) 傳入操作會議資料庫的請求時，依據參數內容設定資料庫的連線
虛擬碼	<pre> sub SelectDatabase (): 1. 獲取資料庫存取命令 2. if 參數為 1 (1 代表存取 Meeting Database) : 設定 Meeting Database 的資料庫連線 else 設定 Meeting Room Database 的資料庫連線 end if end sub </pre>

6.6.2 SQLTranslator

方法	Translate ()
描述	將上層子系統 (Reservation Management) 操作帳號資料庫的 Python 程式語言轉換為 SQL 程式語言
虛擬碼	<pre>sub Translate (): 1. 獲取資料庫存取命令 2. if 命令為查詢類型: 設定命令類型變數為 0 (0 代表查詢資料命令) else if 命令為增加類型 設定命令類型變數為 1 (1 代表增加資料命令) else if 命令為修改類型 設定命令類型變數為 2 (2 代表修改資料命令) else 設定命令類型變數為 3 (3 代表刪除資料命令) end if 3. switch(命令類型變數) case 0: 透過 SQLAlchemy 工具包查詢資料庫資料</pre>

	<p>case 1:</p> <p>透過 SQLAlchemy 工具包插入資料進資料庫</p> <p>case 2:</p> <p>透過 SQLAlchemy 工具包修改資料庫資料</p> <p>default:</p> <p>透過 SQLAlchemy 工具包刪除資料庫資料</p> <p>end switch</p> <p>end sub</p>
--	---

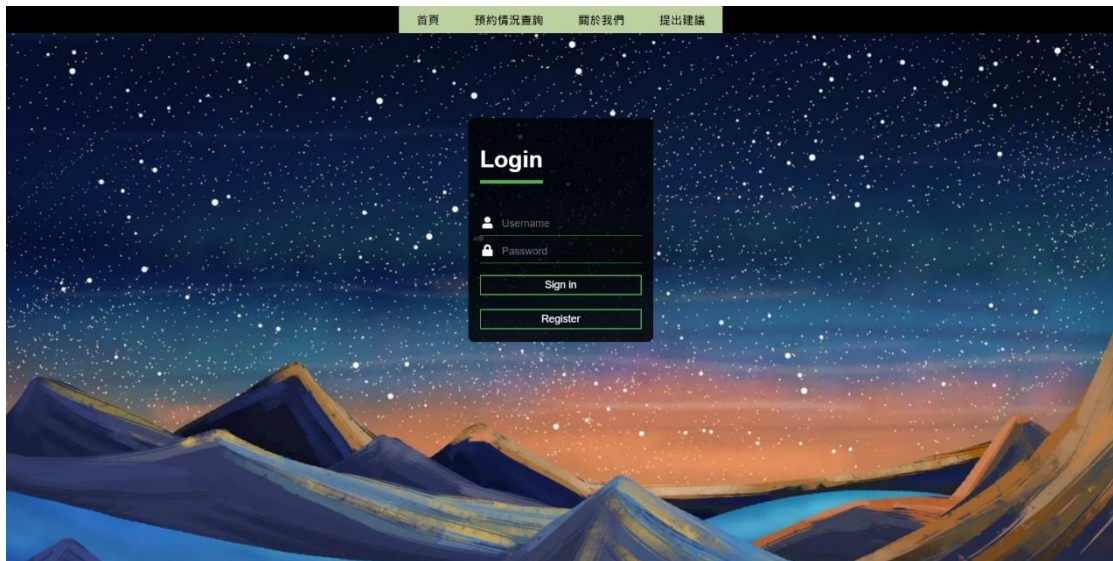
7. HUMAN INTERFACE DESIGN

7.1 Overview of User Interface

The user has to login or register an account first. After finish authentication. The user interface for the system will allow the user to easily check the time of meeting on a calendar which can be displayed in day, week or month. After the meeting time being selected, the meeting room status will be displayed to let user choose which room to reserve. User can enter the detail information of meeting and add participant when creating a meeting.

7.2 Screen Images

7.2.1 Authentication



7.2.2 Event calendar



7.2.3 Meeting room status



7.3 Screen Objects and Actions

7.3.1 Login

Username: Consist with alphabet and numbers. No special characters, space.

Password: Consist with alphabet and numbers. No special characters, space.

Submit: Send username and password to the server for authentication.

Register: Switch to the register page.

7.3.2 Register

Username: Consist with alphabet and numbers. No special characters,

space.

Password: Consist with alphabet and numbers. No special characters, space.

Submit: Send username and password to server for registration.

7.3.3 Calendar

Calendar: Displayed current time and exist event in day, week or month mode.

Event: Each tile on calendar. It can link to event page or meeting room reservation page.

7.3.4 Meeting room

Inquire: Showing status of meeting room.

CreateEvent: Creating a reservation. User should enter the detail of meeting and send to server.

8. REQUIREMENTS MATRIX

Requirements

X	Authentication	Reservation	Google	Account	Meeting Data
component	Management	Management	Services	Data	Management
			Management	Management	
Register and login	1	0	0	1	0
Room reservation	1	1	0	1	1
Search Room status	0	1	0	0	1
Update Meeting status	1	1	1	0	1
Cancel meeting	1	1	1	1	1
Join meeting	1	1	1	1	1
Send Notification	0	1	1	0	1