# Software Design Document

## for Virtual Room Reservation Assistant

**Version 1.0**

**2020.12.23**

B10715037 張家菁

B10715016 梁欣童

B10730033 宋旻芸

B10715053 吳政杰

# Table of Contents

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to describe the overall structure of our meeting room reservation system, as well as the detailed content inside of the system, the detailed functions of the objects in the system, and the detailed functions of different functions, and part of the original code. Therefore, this document is mainly used for users or managers with engineering background. Of course, if you don’t have engineering background, you are welcome to read this document and learn about the principles and architecture of the system.

## 1.2 Scope

The purpose of this meeting room reservation system is to allow people who need to use the meeting room to make online reservations, thus saving many unnecessary procedures. For example, spending lots of time commuting just to reserve meeting rooms in person, and there may be times when all rooms are full , this situation can be avoided after using this system, which saves both manpower and time.

## 1.3 Overview

The following is the overview of this document:

### Chapter 1 Introduction

This chapter is mainly used to introduce the chapters of this document, and some reference documents and explanations of terms.

### Chapter 2 System Overview

This chapter is an introduction to the system.

### Chapter 3 System Architecture

This chapter is used to introduce the main internal architecture of the system, which contains some visual graphics of the model to help users understand, such as the ER model.

### Chapter 4 Data Design

This chapter is used to introduce the data structure design of this system.

### Chapter 5 Component Design

This chapter is used to introduce some internal functions (including code) and the role of objects.

### Chapter 6 Human Interface Design

This chapter is used to introduce some system UI and user interface graphics.

### Chapter 7 Requirements Matrix

This chapter shows some system requirements, which will be presented in the form of tables.

## 1.4 Reference Material

There are two extra files here, if you are interested in this system, you can also check it out.

1. SRS document.[1]
2. Architecture document.

## 1.5 Definitions and Acronyms

1. SRS document: SRS doucment is a software requirements specification (SRS) helps you lay the groundwork for product development.[1]

# 2. System Overview

**Functional Analysis**

Input: Collect the commands that users want to perform actions on the system, such as pressing buttons, inputting data, etc.

Output: According to the user's behavior, the corresponding data presentation, appointment and other processes are carried out.

**Process**

- User query reservation information
- User (management level) queries all user reserving data
- Recall reserved data
- User cancels reservation
- Determine whether cancel reservation or not (whether it is within the time limit)
- User reserves reference room
- Determine whether the reference room can be borrowed or not (whether the number of people meets the limitation)
- The user (management level) changes the authority of other users
- Determine whether one has the authority to change the user's authority
- The user (management level) changes the room settings

**Modules**

According to the functions of this system, we can divide the modules into the following five types:

1. login module
2. reservation module
3. record module
4. setting module
5. exit module

**login module**

The module that allows users to log in.

**reservation module**

A module for users to reserve a reference room, which contains the reference room's ID number, date, time, attendees and other information.

**record module**

A module for users to inquire about reserving records, which includes the function that allow users to cancel.

**setting module**

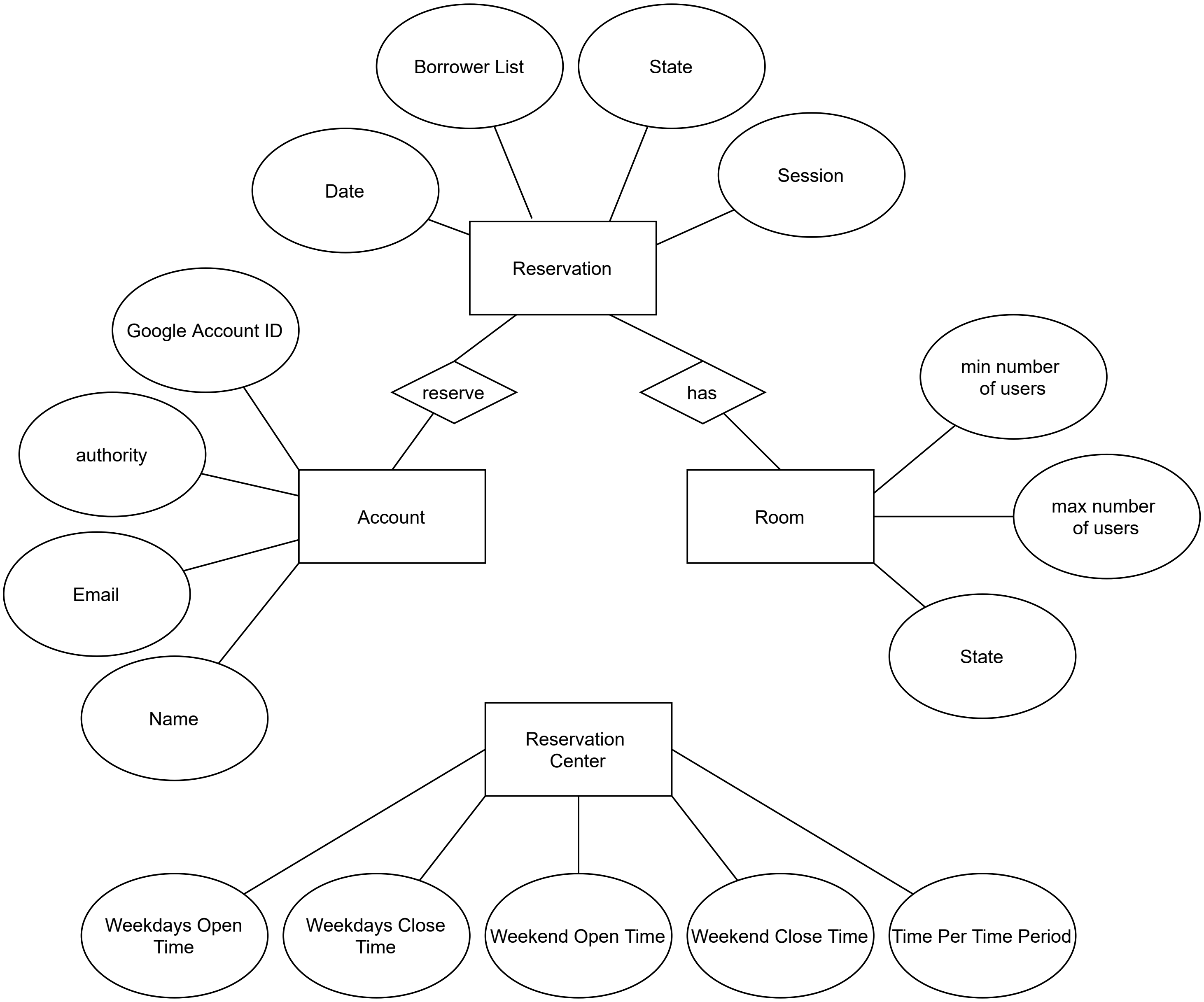
Only for managers, it is a module for managers to set room data and user's authority.

**exit module**

The module that allows users to log out.

# 3. System Architecture

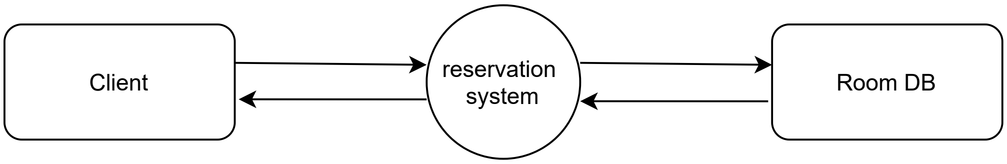
## 3.1 Architectural Design



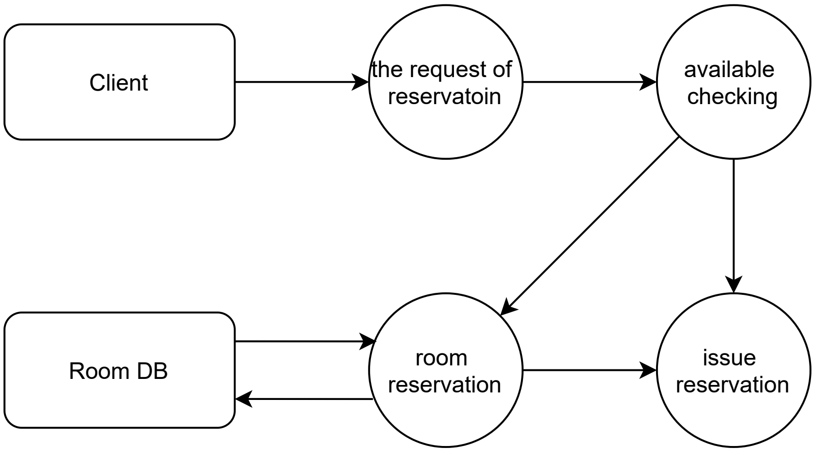
## 3.2 Decomposition Description

### Data Flow Diagram:

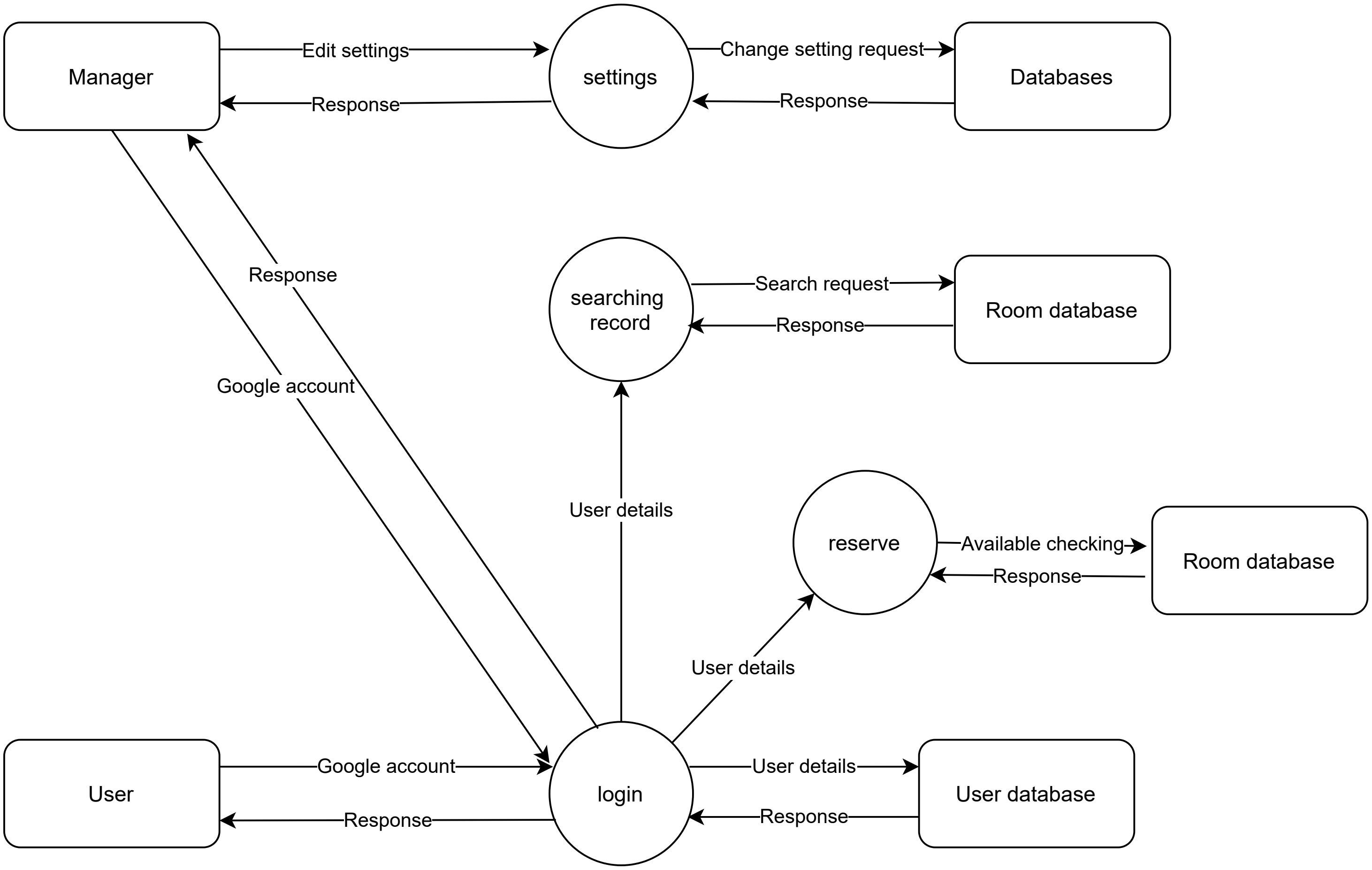
### level 0:



### level 1:



**level 2:**



## 3.3 Design Rationale

In section 3.1, we can see that there are four main entities. According to these four entities, users can know the architecture of the entire system more easily. Through the diagram in section 3.1, users can also know the attributes of each entity very easily, and could infer that what kind of data the system will need when the system is operating, and how the entire system works, too. We think it would be more convinient for users to get familiar with our system.

# 4. Data Design

This section describes the category of data required by the system. The data listed below showed the entity relationship of the system.

## 4.1 Data Description

This section explain how the information domain of the system is transformed into data structures. It also show the major data such as database and data storage items used in system.

- A List of Room including:
  - Room number (Room ID)
  - The maximum number of total attendees
  - The minimum number of total attendees
  - The condition of the room (enable or disable)
- The information about the reservation center:
  - The open hours during weekdays
  - The closed hours during weekdays
  - The open hours during weekend
  - The closed hours during weekend
- The information about the room that user reserved:
  - Room number (Room ID)
  - The password of the Room
  - Reservation's date and time
  - The rules of the room
- The information of Reservation made:
  - User's Name
  - attendees' email
  - Room number (Room ID)
  - Date and Time of the reservation
  - The Number of attendees
  - Duration
- User Information, most importantly containing:
  - Google client ID
  - User's Room reservation records
  - User Privilege
- Manager Information, most importantly containing:
  - Manager Google client ID
  - Manager Privilege

## 4.2 Data Dictionary

This section list the major data of the system by a table. As the table listed below describes the field name, data type, data format, field size, description and is it accepts null value.

**Account**

| Field Name | Data Type | Data Format | Field Size | Description | Accepts null value? |
|---|---|---|---|---|---|
| id | String | OOOOOOOO-OOOO-OOOO-OOOO-OOOOOOOOOOOO | 128 | The id of the user | N |
| privilege | Int |  | 16 bit | Defined the user's privilege | N |
| email | String | OOO@OOO.OOO | 128 | Defined the user's email | N |
| name | String |  | 256 | Defined the user's name | N |

**Room**

| Field Name | Data Type | Data Format | Field Size | Description | Accepts null value? |
|---|---|---|---|---|---|
| id | Int |  | 16 bit | The id of the room | N |
| maxNumber_of_users | Int |  | 16 bit | The maximum number of total attendees | N |
| minNumber_of_users | Int |  | 16 bit | The minimum number of total attendees | N |
| enable | Bool |  |  | The room is enable or disable | N |

**Reservation**

| Field Name | Data Type | Data Format | Field Size | Description | Accepts null value? |
|---|---|---|---|---|---|
| id | int |  | 128 | The id of this reservation | N |
| user_id | String | OOOOOOOO-OOOO-OOOO-OOOO-OOOOOOOOOOOO | 128 | The id of the user | N |

| Field Name | Data Type | Data Format | Field Size | Description | Accepts null value? |
|---|---|---|---|---|---|
| room_id | Int | | 16 bit | The id of the room | N |
| participantsEmail | String | OOO@email.com;OOO@email.com.. | 1280 | The participants' email | Y |
| session | int | | 8 bit | a period of time arranged for the reservation | N |
| date | date | yyyy-MM-dd | | The date of the reservation | N |
| disable | Bool | | | The room is enable or disable | N |

**Reservation_Center**

| Field Name | Data Type | Data Format | Field Size | Description | Accepts null value? |
|---|---|---|---|---|---|
| weekdays_open_time | time | hh:mm:ss.fffz | | The open hours of the reservation center during weekdays | N |
| weekdays_close_time | time | hh:mm:ss.fffz | | The close hours of the reservation center during weekdays | N |
| weekdend_open_time | time | hh:mm:ss.fffz | | The open hours of the reservation center during weekend | N |
| weekdend_close_time | time | hh:mm:ss.fffz | | The close hours of the reservation center during weekend | N |
| time_per_time_period | time | hh:mm:ss.fffz | | The length of duration | N |

# 5. Component Design

| Class name: Reservation |
|---|

| | |
|---|---|
| **Brief Description:** | |
| **Attributes (fields)** | **Attributes Description** |
| int id; | serial number of reservation record |
| | **Program Description Language** |
| | private int id; |
| string userID; | **Attributes Description** |
| | User that reserved conference room |
| | **Program Description Language** |
| | public string userID; |
| int roomID; | **Attributes Description** |
| | Reserved conference room |
| | **Program Description Language** |
| | public int roomID; |
| string borrowerList; | **Attributes Description** |
| | People who reserved conference room together |
| | **Program Description Language** |
| | public string borrowerList; |
| bool disable | **Attributes Description** |
| | Whether to cancel or not |
| | **Program Description Language** |
| | public bool disable; |
| int session; | **Attributes Description** |
| | Reserved time period |
| | **Program Description Language** |
| | |

| | |
|---|---|
| | public int session; |
| Datetime date; | **Attributes Description** |
| | Reserved date |
| | **Program Description Language** |
| | public Datetime date; |
| **Methods (operations)** | **Method Description** |
| void create(); | Create reservation record |
| | **Program Description Language** |
| | db.Reservations.Add(reserveModel); |
| void cancel(); | **Method Description** |
| | Cancel the reservation |
| | **Program Description Language** |
| | System.Models.Reservations reservation = (from s in db.Reservations where s.Id == id select s).First();<br>reservation.Disable = true;<br>db.SaveChanges(); |
| void show(); | **Method Description** |
| | According to the authority and query content, the reservation record will be sent to users |
| | **Program Description Language** |
| | if (user.Authority > 0) {<br>reservations = (from s in db.Reservations select s).ToList();<br>}<br>else<br>{ |

```
            reservations = (from s in db.Reservations where s.AspNetUserId == ASPNetUserID select s).ToList();

        }

        if (Date != null)

        {

            DateTime date = DateTime.ParseExact(Date, "yyyy-MM-dd", System.Globalization.CultureInfo.InvariantCulture);

            reservations = reservations.Where(x => x.Date == date).ToList();

        }

        if (roomId != null)

        {

            reservations = reservations.Where(x => x.RoomId == roomId).ToList();

        }

}
```

| | |
|---|---|
| **Class name: Account** | |
| **Brief Description:** | |
| **Attributes (fields)** | **Attributes Description** |
| int id; | User ID |
| | **Program Description Language** |
| | public int id; |
| int authority; | **Attributes Description** |
| | User’s authority |
| | **Program Description Language** |
| | public int authority; |
| string email; | **Attributes Description** |

| | |
|---|---|
| | User’s google email |
| | **Program Description Language** |
| | public string email; |
| string name; | **Attributes Description** |
| | User name |
| | **Program Description Language** |
| | public string name; |
| **Methods (operations)** | **Method Description** |
| void create(); | Create reserving record |
| | **Program Description Language** |
| | db.Account.Add(accountModel); |
| void editAuthority(); | **Method Description** |
| | Edit user’s authority |
| | **Program Description Language** |
| | System.Models.AspNetUsers user = (from s in db.AspNetUsers where s.Id == Id select s).First();<br>user.Authority = int.Parse(Authority); |
| void show(); | **Method Description** |
| | Return all users |
| | **Program Description Language** |
| | List<System.Models.AspNetUsers> users = (from s in db.AspNetUsers select s).ToList();<br>return users; |

| |
|---|
| **Class name: Room** |
| **Brief Description:** |

| **Attributes (fields)** | **Attributes Description** |
|---|---|
| int id; | serial number of conference room |
| | **Program Description Language** |
| | private int id; |
| int minNumberOfUsers; | **Attributes Description** |
| | Min number of people using conference room |
| | **Program Description Language** |
| | public int minNumberOfUsers; |
| int minNumberOfUsers; | **Attributes Description** |
| | Max number of people using conference room |
| | **Program Description Language** |
| | public int minNumberOfUsers; |
| bool enable | **Attributes Description** |
| | Is the conference room available or not |
| | **Program Description Language** |
| | public string enable; |
| **Methods (operations)** | **Method Description** |
| void create(); | Create a conference room to the system |
| | **Program Description Language** |
| | db.Reservations.Add(roomModel); |
| void edit(); | **Method Description** |
| | Edit data and settings of conference room |
| | **Program Description Language** |
| | System.Models.Rooms room = (from s in db.Rooms where s.Id == id select s).First(); |

| | |
|---|---|
| | room = roomModel; |
| void delete(); | **Method Description** |
| | Delete conference room(Only set room as unavailable) |
| | **Program Description Language** |
| | System.Models.Rooms room = (from s in db.Rooms where s.Id == id select s).First();<br>room.Enable = false; |
| void show(); | **Method Description** |
| | Return all of the information of conference room |
| | **Program Description Language** |
| | List<System.Models.Rooms> rooms = (from s in db.Rooms where s.Enable select s).ToList();<br>return rooms; |

# 6. Human Interface Design

## 6.1 Overview of User Interface

**Login**

- User Login Interface
- View any interface that need user to login. After any action is done successfully will lead the user back to this interface. After login successfully, the website will lead the user back to the main page.

**Home**

- An interface introducing the system.
- Only allowing the users who have logged in to access.

**Reserve**

- The interface that user reserves a meeting room here.
- Only allowing the users who have logged to access.

**Record**

- The interface that display the reservation record.
- Only allowing the users who have logged in to access.
- Common users can only access their own reservation record, the manager can view all the record within the database.
- General users can only view their own reservation record, the manager can view all the records in the database.

**RoomCenterSetting**

- The interface that the manager manages the system settings and selects the meeting rooms whose settings need to be changed.
- Only the managers who have logged in can access.
- It is an interface that manages the system settings and selects the meeting rooms which it's settings need to be changed.

**RoomSetting**

- The interface that the manager manages the overall setting of the meeting room and lists all the meeting room.
- Only the managers who have logged in can access.

**RoomCreate**

- The interface that the manager create a new meeting room in the system.
- Only the managers who have logged in can access.

**UsersSetting**

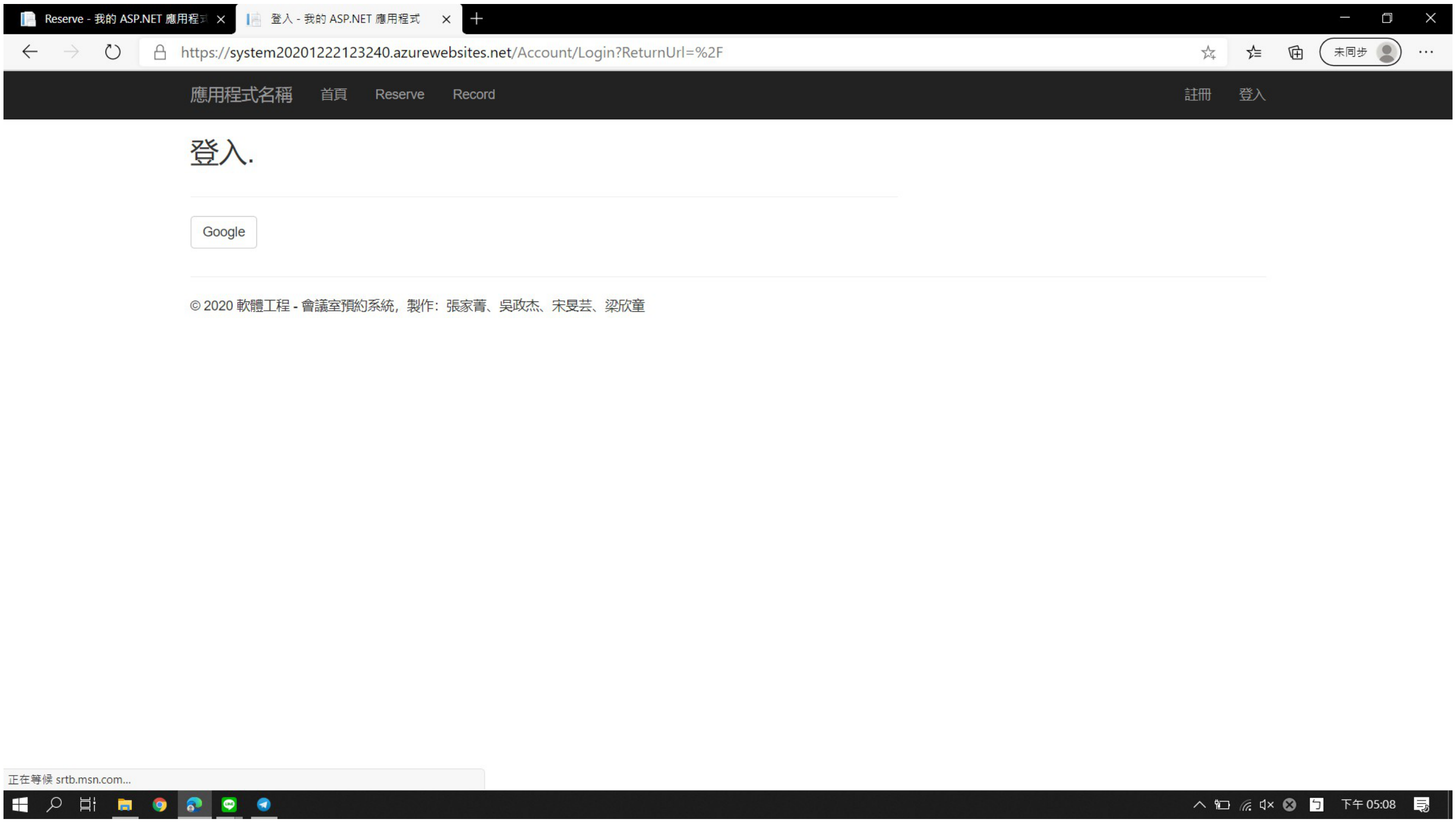
- Only the managers who have logged in can access.
- The interface that show all the users and the manager can select a user for editing his/her information.

**UserEdit**

The interface that the manager edits the user's privilege.
Only the managers who have logged in can access.

## 6.2 Screen Images

**Login**

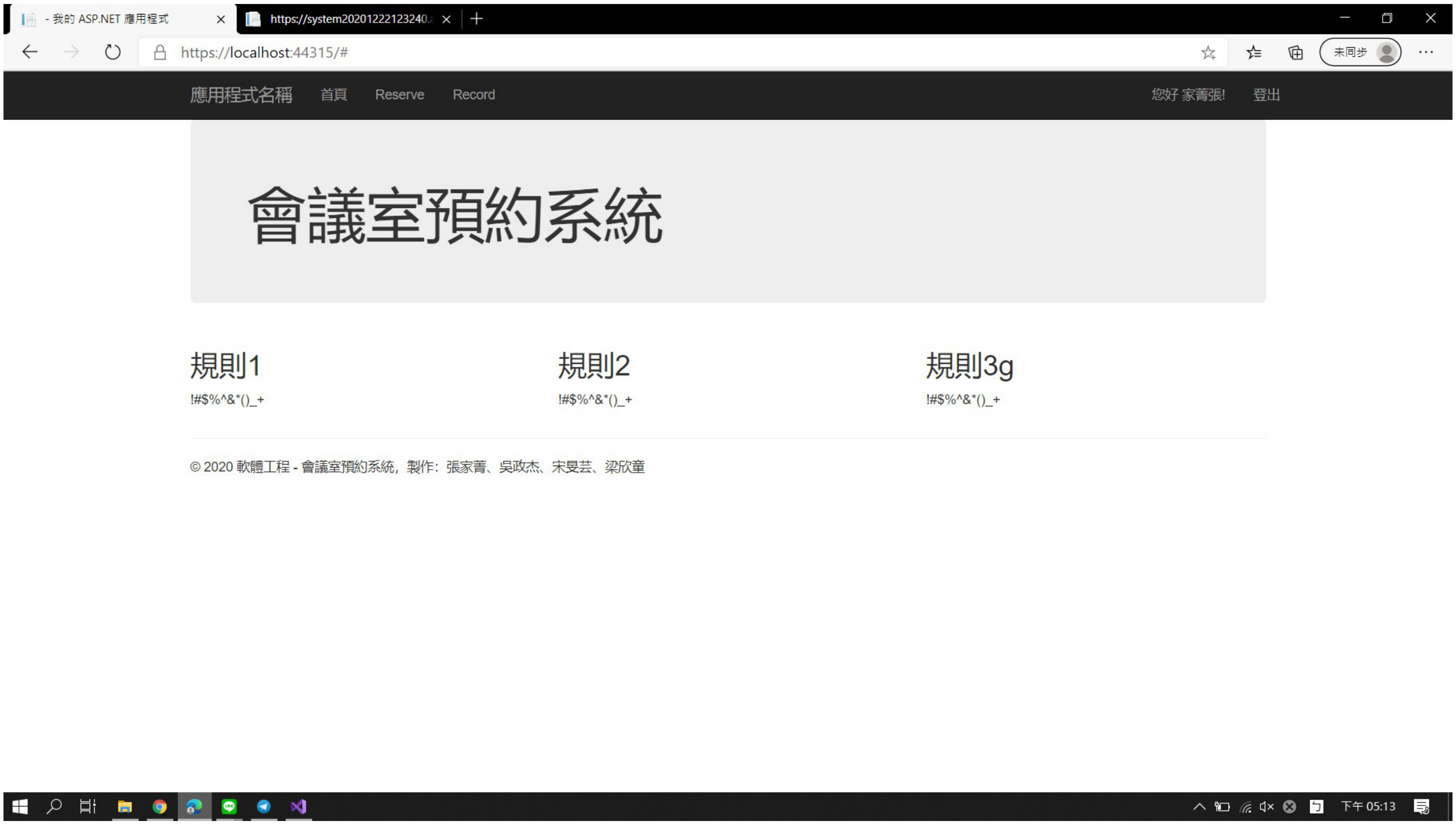

Reserve - 我的 ASP.NET 應用程式
登入 - 我的 ASP.NET 應用程式
https://system20201222123240.azurewebsites.net/Account/Login?ReturnUrl=%2F
未同步
應用程式名稱
首頁
Reserve
Record
註冊
登入
正在等候 srtb.msn.com...
下午 05:08

登入.

Google

© 2020 軟體工程 - 會議室預約系統，製作：張家菁、吳政杰、宋旻芸、梁欣童

## Home



會議室預約系統

規則1

!#$%^&*()_+

規則2

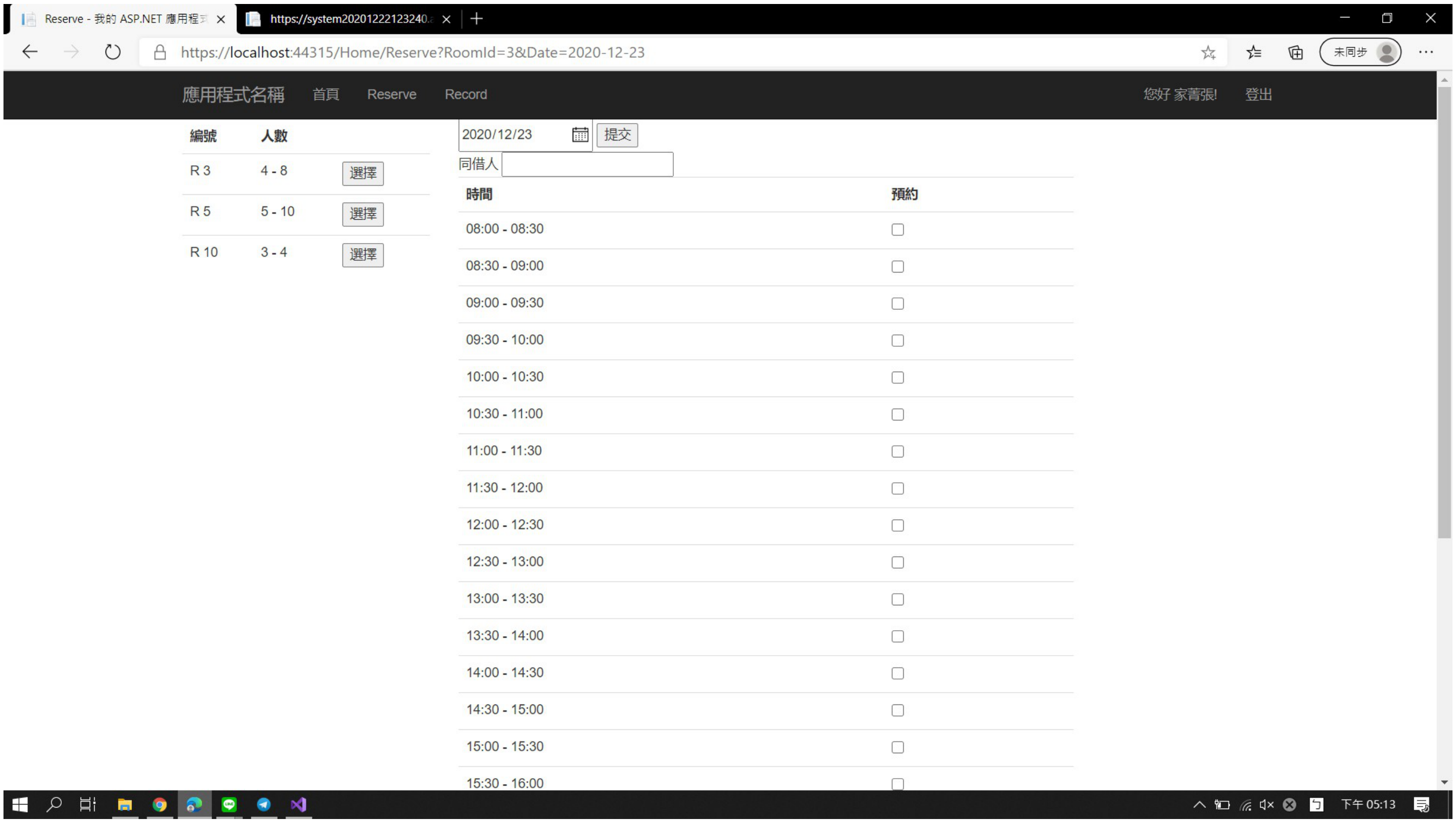
!#$%^&*()_+

規則3g

!#$%^&*()_+

© 2020 軟體工程 - 會議室預約系統，製作：張家菁、吳政杰、宋旻芸、梁欣童

## Reserve



https://localhost:44315/Home/Reserve?RoomId=3&Date=2020-12-23

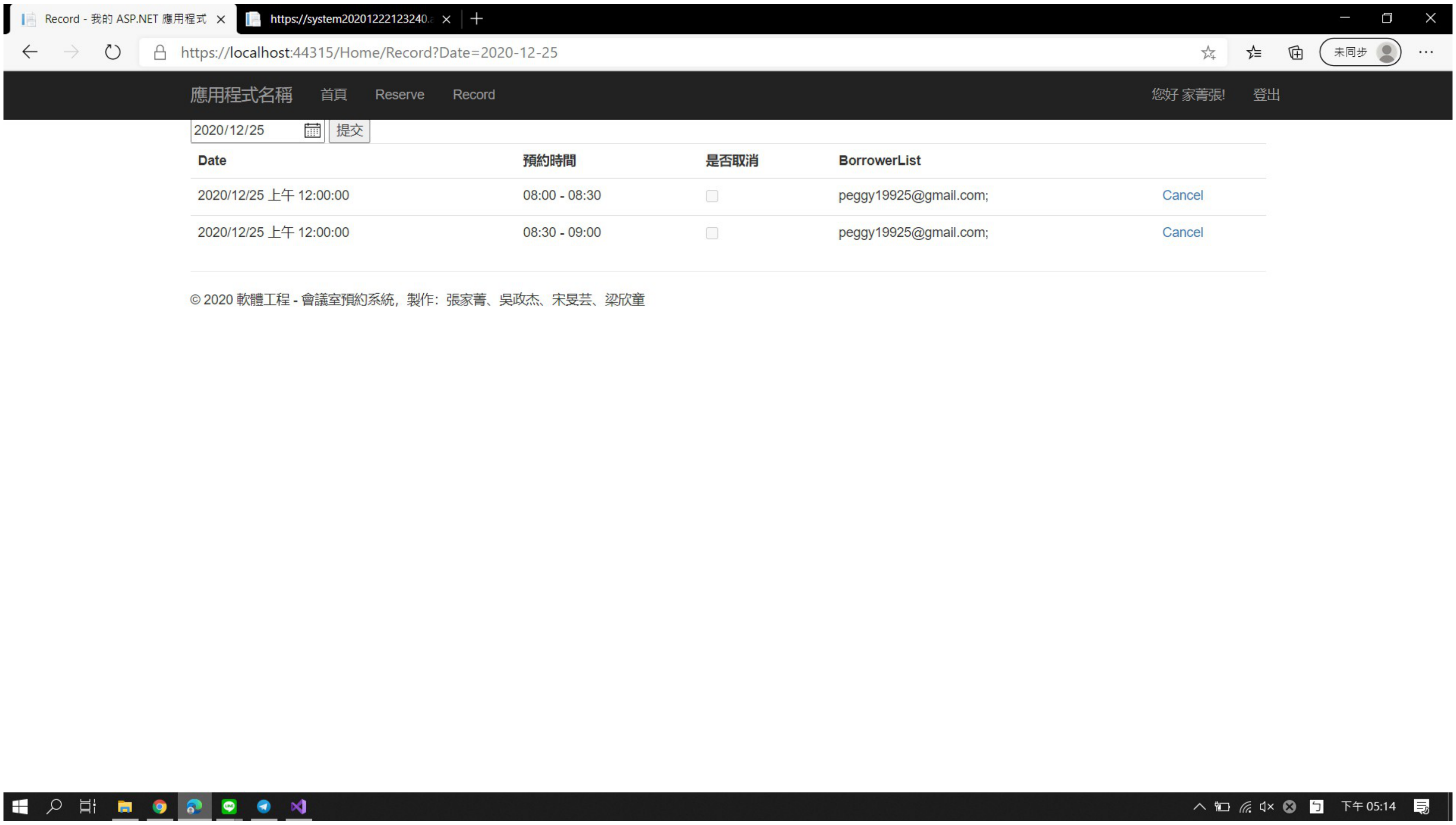
| 編號 | 人數 |  |
|---|---|---|
| R 3 | 4 - 8 | 選擇 |
| R 5 | 5 - 10 | 選擇 |
| R 10 | 3 - 4 | 選擇 |

2020/12/23 提交

同借人

| 時間 | 預約 |
|---|---|
| 08:00 - 08:30 | □ |
| 08:30 - 09:00 | □ |
| 09:00 - 09:30 | □ |
| 09:30 - 10:00 | □ |
| 10:00 - 10:30 | □ |
| 10:30 - 11:00 | □ |
| 11:00 - 11:30 | □ |
| 11:30 - 12:00 | □ |
| 12:00 - 12:30 | □ |
| 12:30 - 13:00 | □ |
| 13:00 - 13:30 | □ |
| 13:30 - 14:00 | □ |
| 14:00 - 14:30 | □ |
| 14:30 - 15:00 | □ |
| 15:00 - 15:30 | □ |
| 15:30 - 16:00 | □ |

## Record



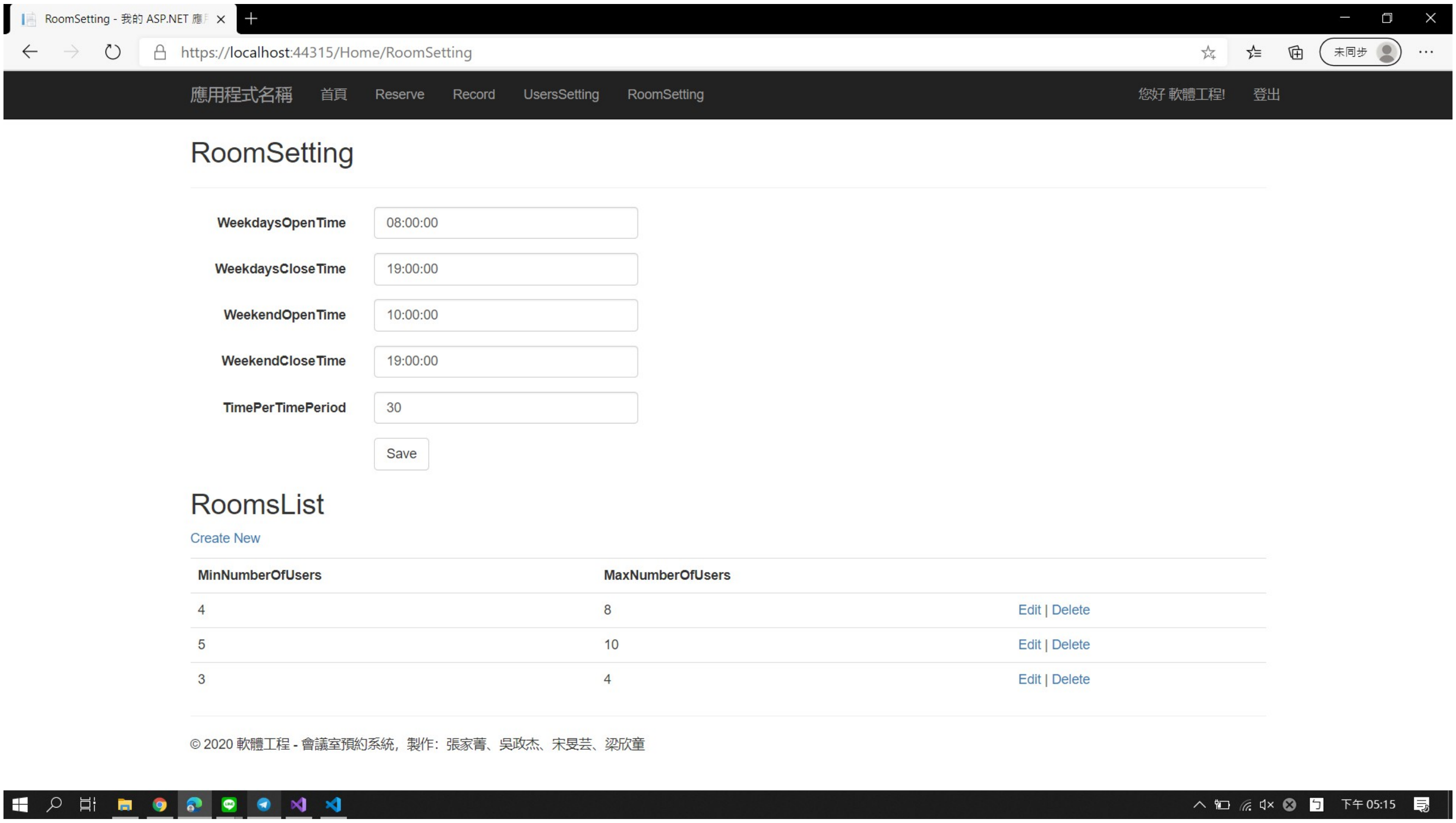
https://localhost:44315/Home/Record?Date=2020-12-25

2020/12/25 提交

| Date | 預約時間 | 是否取消 | BorrowerList |  |
|---|---|---|---|---|
| 2020/12/25 上午 12:00:00 | 08:00 - 08:30 | □ | peggy19925@gmail.com; | Cancel |
| 2020/12/25 上午 12:00:00 | 08:30 - 09:00 | □ | peggy19925@gmail.com; | Cancel |

© 2020 軟體工程 - 會議室預約系統，製作：張家菁、吳政杰、宋旻芸、梁欣童

## RoomCenterSetting



## RoomSetting

| | |
|---|---|
| WeekdaysOpenTime | 08:00:00 |
| WeekdaysCloseTime | 19:00:00 |
| WeekendOpenTime | 10:00:00 |
| WeekendCloseTime | 19:00:00 |
| TimePerTimePeriod | 30 |

Save

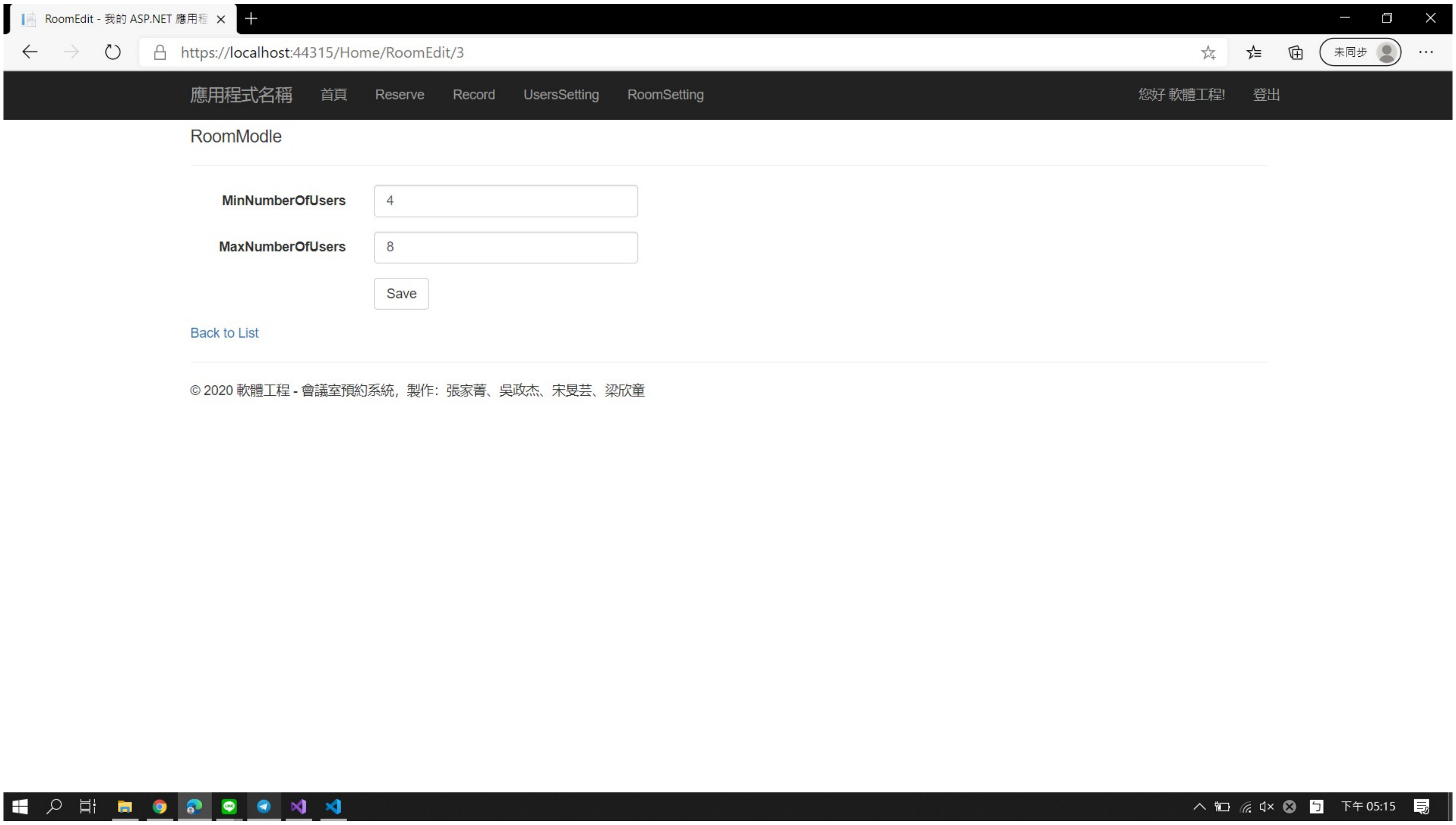
## RoomsList

Create New

| MinNumberOfUsers | MaxNumberOfUsers | |
|---|---|---|
| 4 | 8 | Edit \| Delete |
| 5 | 10 | Edit \| Delete |
| 3 | 4 | Edit \| Delete |

© 2020 軟體工程 - 會議室預約系統，製作：張家菁、吳政杰、宋旻芸、梁欣童

# RoomSetting



RoomModle

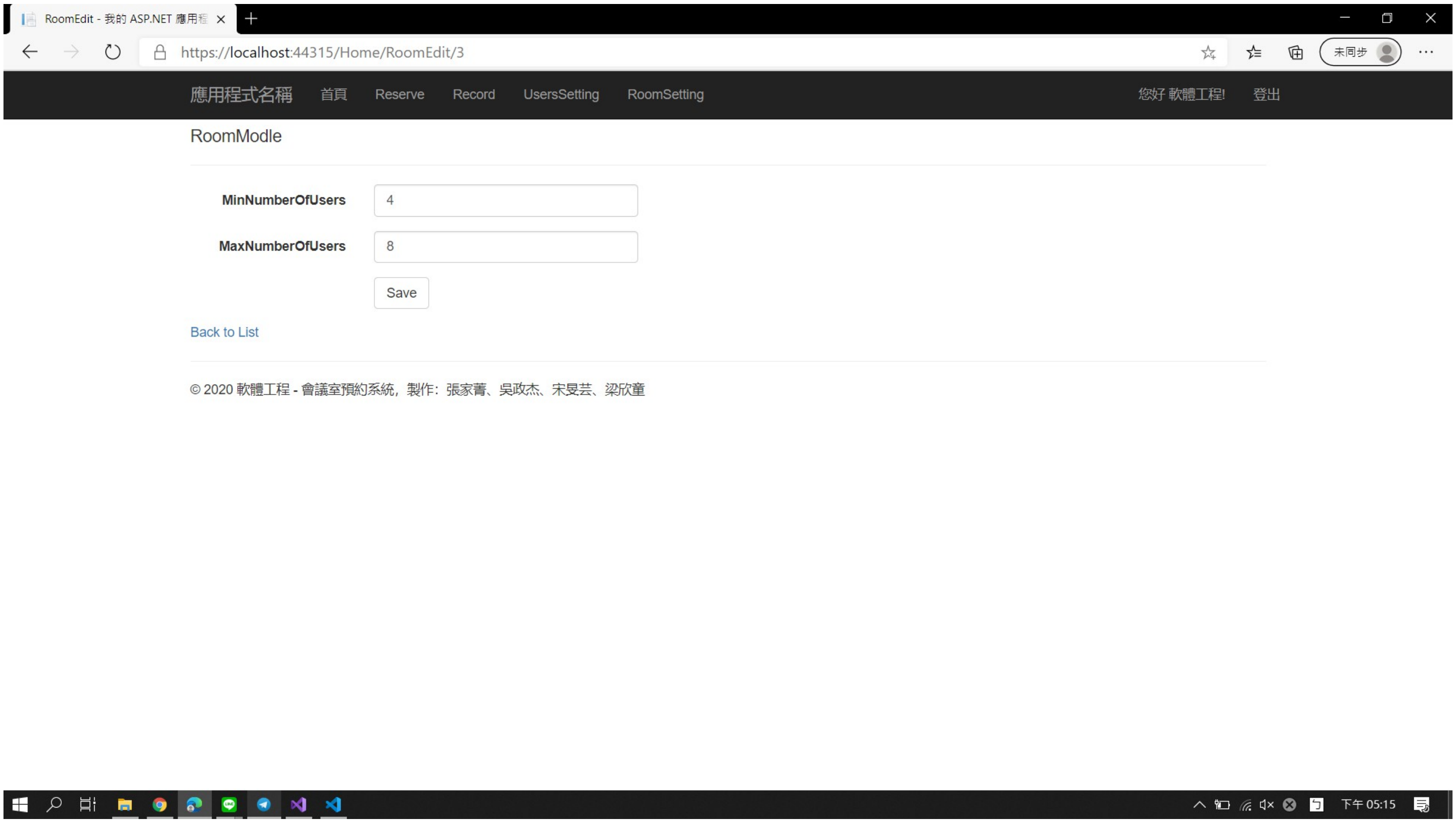
| | |
|---|---|
| MinNumberOfUsers | 4 |
| MaxNumberOfUsers | 8 |

Save

Back to List

© 2020 軟體工程 - 會議室預約系統，製作：張家菁、吳政杰、宋旻芸、梁欣童

# RoomCreate



RoomModle

MinNumberOfUsers 4

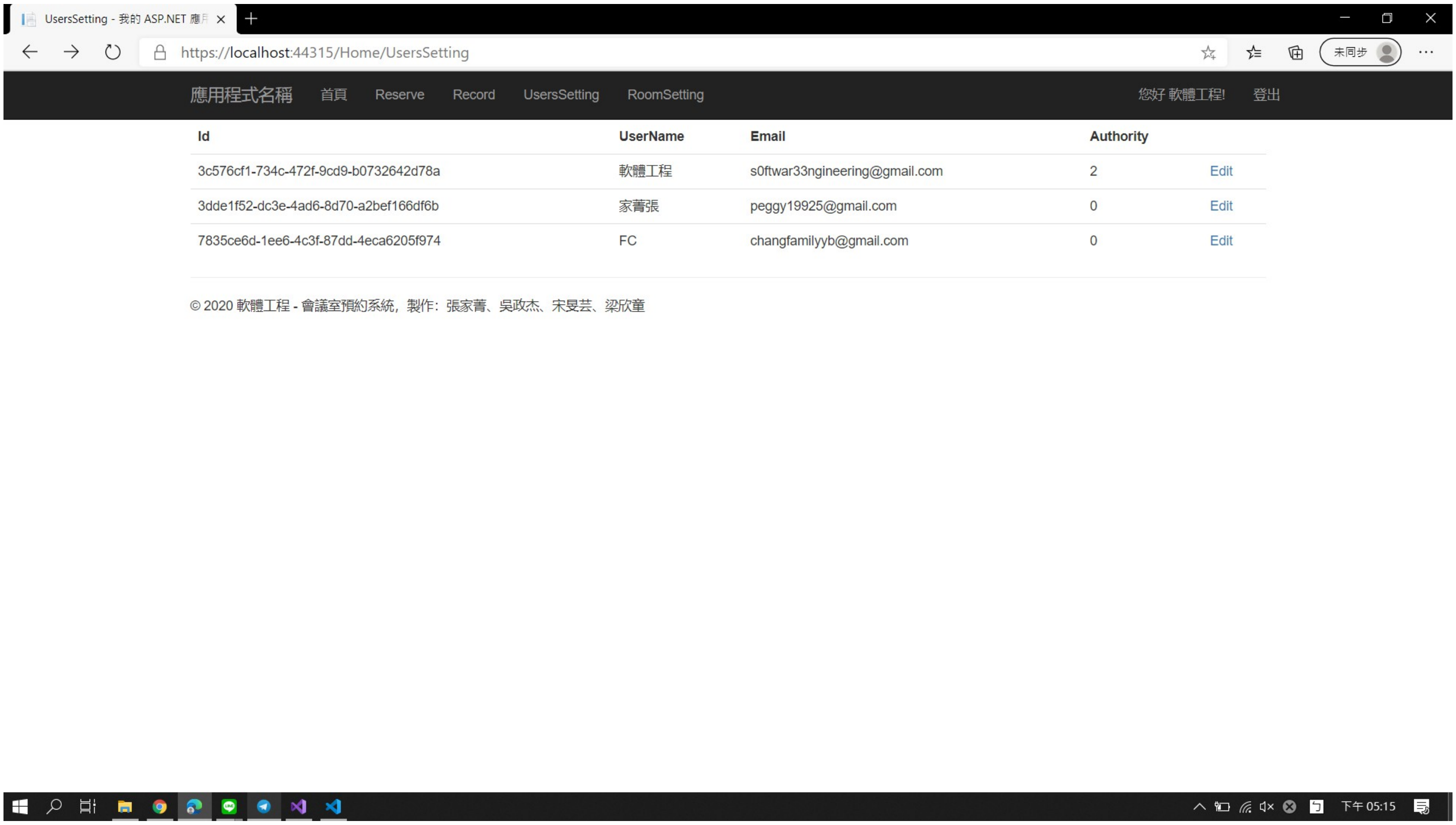
MaxNumberOfUsers 8

Save

Back to List

© 2020 軟體工程 - 會議室預約系統，製作：張家菁、吳政杰、宋旻芸、梁欣童

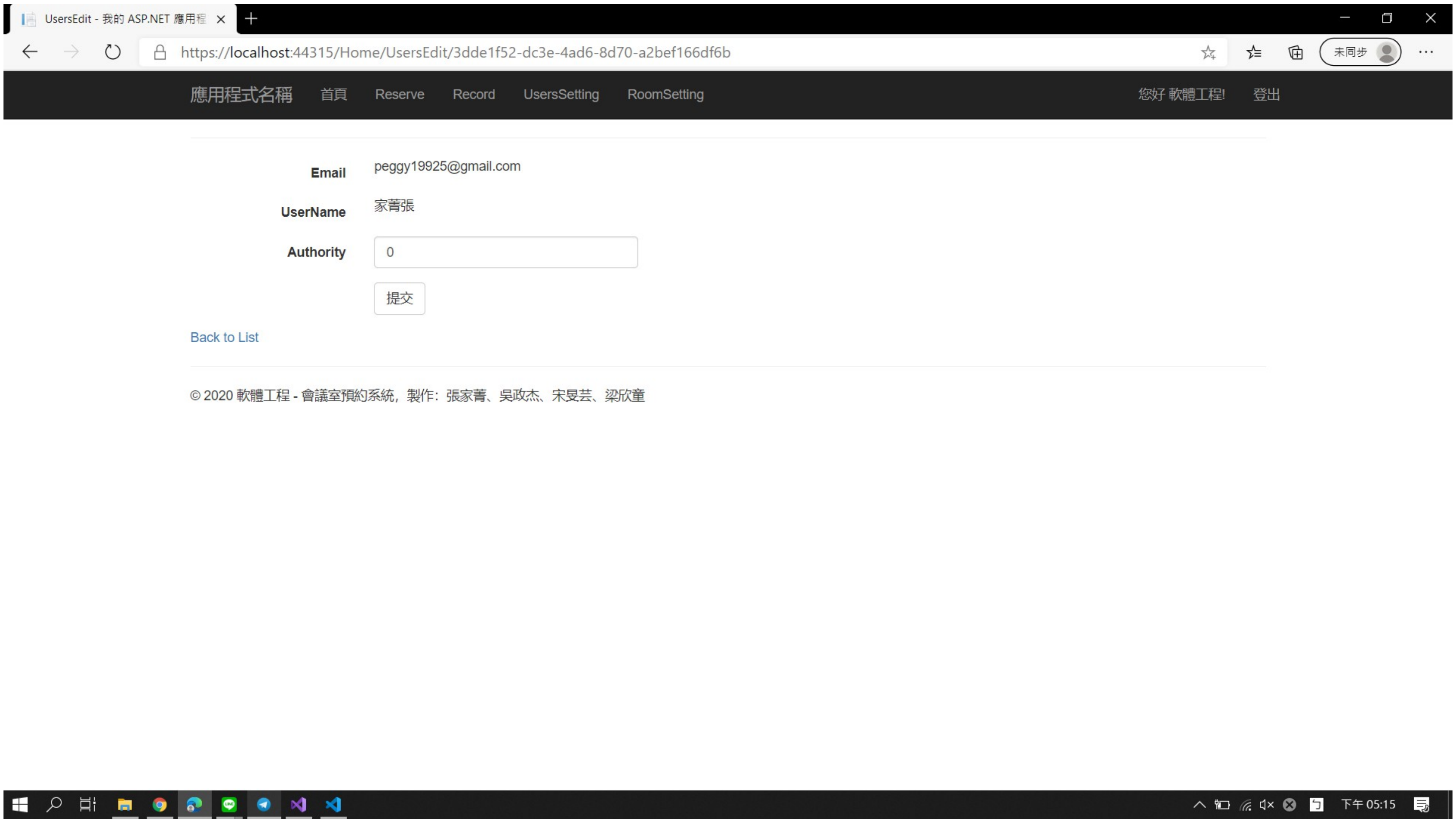
# UsersSetting



| Id | UserName | Email | Authority | |
|---|---|---|---|---|
| 3c576cf1-734c-472f-9cd9-b0732642d78a | 軟體工程 | s0ftwar33ngineering@gmail.com | 2 | Edit |
| 3dde1f52-dc3e-4ad6-8d70-a2bef166df6b | 家菁張 | peggy19925@gmail.com | 0 | Edit |
| 7835ce6d-1ee6-4c3f-87dd-4eca6205f974 | FC | changfamilyyb@gmail.com | 0 | Edit |

© 2020 軟體工程 - 會議室預約系統，製作：張家菁、吳政杰、宋旻芸、梁欣童

# UserEdit


UsersEdit - 我的 ASP.NET 應用程
https://localhost:44315/Home/UsersEdit/3dde1f52-dc3e-4ad6-8d70-a2bef166df6b
未同步
應用程式名稱
首頁
Reserve
Record
UsersSetting
RoomSetting
您好 軟體工程!
登出
提交
Back to List
下午 05:15

Email peggy19925@gmail.com

UserName 家菁張

Authority 0

© 2020 軟體工程 - 會議室預約系統，製作：張家菁、吳政杰、宋旻芸、梁欣童

# 6.3 Screen Objects and Actions

### Login

- After clicking Google, call Google's Oauth 2.0 API to jump to the Google login interface and get user's identity.
- If it is the user's first login (that is, there is no user's data in the database), add a new user and add it to the database.
- If it is not the user's first login (that is, the user's data is already in the database), the user's data in the database is obtained.
- After getting the user's identity, according to the returnURL accessed by the query, direct to the original page opened by the user.

### Home

This page can only be viewed when the user has logged in. This is the introduction page of the system.

### Reserve

- When the user selects a conference room, it will send "get request" to the server side, the server side will reply for the reservation conditions of the meeting room selected by the user.
- When the user switch the selected meeting room to others, it will send "get request" to the server side, the server side will reply for the reservation conditions of the date selected by the user.
- After pressing "submit", it will sent "post request" to the server side, the server side will add a reservation record to the database which is based on the options selected by the user.

**Record**

- The server side replies the user's own reservation records to users.
- When clicking "date query", a get request will be sent to the server side, and the server side will respond the corresponding date's reservation record.

**RoomCenterSetting**

- After the server side confirms that the user is a manager, it will return the information of all conference rooms.
- After clicking "Save", the edited content will be sent to the server side by post request, and let the server side save the changes according to the roomid.
- After clicking "Edit Room" of any of the conference rooms, system will switch to the RoomSetting interface.
- After clicking any "Delete Room" of any of the conference rooms, a request will be sent to the system side, and "disable" in the corresponding roomid will be set to "true".
- After clicking "Create Room", it will switch to the RoomCreate page.

**RoomSetting**

- After the server side confirms that the user is a manager, according to the roomid clicked by the user, it will return all of the information of the corresponding room.
- When the user finishes editing, send post request to the system side, and let the system side save the changes according to the roomid.

**RoomCreate**

- The interface will be shown after the server side confirms that the user is a manager.
- After clicking "Finish", a post request will be sent to the backend, then the server side can add the new room data to the database.

**UsersSetting**

- After the server side confirms that the user is a manager, it will return all of the user's data.
- After clicking "Edit", it will switch to the UserEdit interface.

**UserEdit**

- After the back-endthe user confirmed as a manager, according to the incoming user id, it will return the data of the corresponding user.
- After the user finish editing, use post request to return data to the server side, then the system side can save the changes according to the roomid.

# 7. Requirements Matrix

| REQ ID. | REQ Description | Components | Trace Description |
|---|---|---|---|
| REQ2.2.1 / REQ2.2.2 / REQ2.2.3 | The login and logout system is using log-in-out module provides by google | Google Client ID | Google Client module can be used in system |
| REQ2.2.4 | User Reserve Conference Room | Calendar | Select invalid date |
| | | Room selection | Select no room |
| | | | Select a room which is disable |
| | | System marks the time which is already reserved is disable | Select a room which is disable |
| | | | The table of conference room is marked automatically by the system |
| | | User confirm | Sending a confirmed email to the user's email include the information about the reserved room |
| REQ2.2.5 | User View own reserved record | Record Button | The module of record button works or not |
| | | Search | Is the database works correctly |
| | | Display Record | The table of reservation record display any error |
| | | | The table of reservation record display any error while the user have over one reservation records |
| REQ2.2.6 | User cancels room reservation | Record Button | The module of record button works or not |
| | | Search | Is the database works correctly |
| | | Display Record | The table of reservation record display any error |
| | | | The table of reservation record display any error while the user have over one reservation records |
| | | Cancel | After confirm cancelation, the database delete the reservation record correctly |
| | | Room settings Button | The module of room settings button works or not |
| | | (Single room's) Room selection | Select no room and change the overall setting, press save button |

| | | | |
|---|---|---|---|
| REQ2.2.7 / REQ2.2.8 | Manager  edits the room's / single room's overall settings | Recall | Is the database works correctly |
| | | Display Room Information | The table of room's information display any error |
| | | | The table of room's informaiton display any error while there are a lot of data or reservation records |
| | | Change | Changing the room setting with an invalid input |
| | | Save Button | The module of save button works or not |
| | | | Find the database changed after pressed save button |
| REQ2.2.9 | Manager: Search User | User settings Button | The module of user settings button works or not |
| | | Search | Is the database works correctly |
| | | | Type any special characters for searching |
| | | | Type a name or ID which is not existed |
| | | Search Button | The module of search button works or not |
| | | Display information | The information table of user display any error |
| | | | The information table of user display any error while there are a lot of user |
| | | | Display every information about the user correctly |
| REQ2.2.10 | Manager edits user's privilege settings | User settings Button | The module of user settings button works or not |
| | | Call | Is the database works correctly |
| | | Display information | The information table of user display any error |
| | | | The information table of user display any error while there are a lot of user |
| | | | Display every information about the user correctly |
| | | Change privilege | Input wrong privilege setting |
| | | Save Button | The module of save button works or not |
| | | | Find the user's privilege changed after pressed save button |

|  |  |  |  |
|---|---|---|---|
|  |  |  | after pressed save button |
| REQ2.2.11 | Manager views individual user's room reservation record | User settings Button | The module of user settings button works or not |
|  |  | Call | Is the database works correctly |
|  |  | Search | Is the database works correctly |
|  |  |  | Type any special characters for searching |
|  |  |  | Type a name or ID which is not existed |
|  |  | Search Button | The module of search button works or not |
|  |  | Display Information | The information table of user display any error |
|  |  |  | The information table of user display any error while there are a lot of user |
|  |  |  | Display every information about the user correctly |
|  |  |  | The user's information displayed is matching the user searched by the manager |
|  |  | View Button | The module of view button works or not |
|  |  | Display Record | The table of reservation record display any error |
|  |  |  | The table of reservation record display any error while the user have over one reservation records |
|  |  |  | The user's reservation record displayed is matching the user searched by the manager |