



Function Rooms Booking System For Co-working Space In Singapore

Version 2.0 → Updated to Server-Client Application

Created by
Oon Xin Ni
(Batch 2)
Go Advanced Assignment

Table of Contents

1	Background	3
2	Features	3
2.1	Users category	3
2.2	Application menu	4
2.3	Browse venue and availability	4
2.4	Search venue	5
2.5	Venue booking.....	6
2.6	Browse booking list	6
2.7	Dependency.....	8
3	Error handling & Concurrency.....	8
3.1	Error handling.....	8
3.2	Reducing room for error	9
3.3	Concurrency.....	9

1 Background

For this assignment, it is a continuity from previous assignment. The original text based application was converted to server-client application. For illustration purposes, the application was preloaded with some registered users, venue selection and also some bookings. For the scope of this project, the bookings can only be done for month of March.

Welcome to Function Rooms Booking System

Login

Sign Up

Browse as Guest

2 Features

2.1 Users category

The user of this application was divided into 3 groups. All 3 groups has different access.

1. Guest access

Only 3 features are show when the user browse as guest. When they select the book venue feature, they will be redirected back to welcome page where you can select to log in or sign up.

Welcome to Function Rooms Booking System	Function Rooms Booking System
Login	Browse Venue and Availability
Sign Up	Search for Available Venue
Browse as Guest	Book Venue (Login is required)

Please login to your account

Submit

Sign up [here](#) if you do not have an account

Create New Account

Enter the following to create a new account

First name:

Last name:

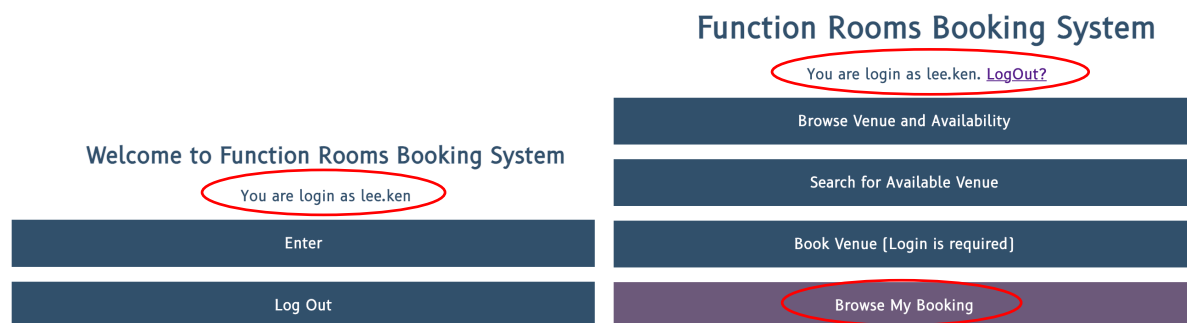
Username:

Password:

Submit

2. Registered user

After registered user is signed in, it will be reflected in their application menu. Registered user will also have an additional feature of viewing their own booking.



3. Admin

When you logged in as admin, you will have access to two additional features which are remove booking and browse booking list.



2.2 Application menu

As you can see from the screen shot above, based on the cookies we save, the menu are customised accordingly based on different category of user. There is also an option to log out and delete the saved cookies.

Multiples sessions were tried out to connect to the local host server :5221 at once. They were able to connect successfully.

2.3 Browse venue and availability

For this feature, it will show the list of venues available at this coworking space. We have different type of function rooms with different kind of capacity. For this feature, selection of room was presented in radio input type format to avoid possible errors that might exist in text input type.

Based on selection, user could view all the availability of a particular function room of their preference. This feature can give the user a quick idea to determine the availability of the slots for a particular day & time. Those booked slot will be reflected as zero.

Browse Venue and Availability

List of venues

1. Room Name: MR01 , Type: MeetingRoom , Capacity: 10
2. Room Name: MR02 , Type: MeetingRoom , Capacity: 10
3. Room Name: MR03 , Type: MeetingRoom , Capacity: 10
4. Room Name: MR04 , Type: MeetingRoom , Capacity: 20
5. Room Name: MR05 , Type: MeetingRoom , Capacity: 20
6. Room Name: AR06 , Type: ActivityRoom , Capacity: 50
7. Room Name: AR07 , Type: ActivityRoom , Capacity: 50
8. Room Name: AR08 , Type: ActivityRoom , Capacity: 50
9. Room Name: AD09 , Type: Auditorium , Capacity: 100
10. Room Name: AD10 , Type: Auditorium , Capacity: 100

Please select the room

- ☐ MR01 ☐ MR02 ☐ MR03 ☐ MR04 ☐ MR05
☐ AR06 ☐ AR07 ☐ AR08 ☐ AD09 ☐ AD10

Click [here](#) to return to main menu

Available Slots For Room 1

Click [here](#) to return to main menu

~ March ~	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
01	1	1	1	1	1	1	1	1	1
02	1	1	1	1	1	1	1	1	1
03	1	1	1	1	1	1	1	1	1
04	1	1	1	1	1	1	1	1	1
05	1	1	1	1	1	1	1	1	1
06	1	1	1	1	1	1	1	1	1
07	1	1	1	1	1	1	1	1	1
08	1	1	1	1	1	1	1	1	1
09	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1
17	0	0	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1
20	1	1	1	1	0	0	0	1	1
21	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1	1
29	1	1	1	1	1	1	1	1	1
30	1	1	1	1	1	1	1	1	1
31	1	1	1	1	1	1	1	1	1

2.4 Search venue

Based on the user input's criteria, available venues are filtered out. If there is no room found, an error message will be prompt. Besides that, the input box was all set as required except the last box, some input patterns were specified as well. This is to ensure the box are filled out and of the desired pattern before the user hit the submit button.

Search for Available Venues

Enter the following to start searching

Please enter your preferred date: {DDMM}

Please enter your preferred time:

Please enter duration of the event in hours:

Please enter total number of participant:

Please enter preferred type of function room: [1, 2, 3]

1. Meeting Room
2. Activity Room
3. Auditorium

Leave blank if there is no preference

Click [here](#) to return to main menu

The available venues are as below:

1. 18/3 14:00 Room 6 is available.
2. 18/3 14:00 Room 7 is available.
3. 18/3 14:00 Room 8 is available.

2.5 Venue booking

Venue booking feature is setup similarly to the search venue feature, however after all the criteria is entered, available venues will be show up at the side and registered user can proceed with the selection of venue. Once the submit button is click, the application will process the booking and a confirmation message will showed on top once the slot is blocked out in the system. Possible race condition is also taken care of, if the option selected was booked right before the user book the venue, it will show an error of unsuccessful booking.

Congratulations, booking for function room 6 is confirmed.

Venue Booking

Please enter your booking details

Please enter your preferred date: (DDMM)

Please enter your preferred time:

Please enter duration of the event in hours:

Please enter total number of participant:

Please enter preferred type of function room: [1, 2, 3]

1. Meeting Room
2. Activity Room
3. Auditorium

Leave blank if there is no preference

Click [here](#) to return to main menu

Please select the room

- ☐ 20/3 12:00 Room 4 is available.
- ☐ 20/3 12:00 Room 5 is available.
- ☐ 20/3 12:00 Room 6 is available.
- ☐ 20/3 12:00 Room 7 is available.
- ☐ 20/3 12:00 Room 8 is available.
- ☐ 20/3 12:00 Room 9 is available.
- ☐ 20/3 12:00 Room 10 is available.

2.6 Browse booking list

2.6.1 Browse registered user booking list

For registered users, once they are login, the option of browsing their booking list is shown on their menu. They can select this option to view all of the booking they have done. The list will be updated as well once they are done with their booking.

Browse My Booking List

Click [here](#) to return to main menu

Below is summary of the booking list

Booking ID: 2, Bookings for 4 hours: [31104 31204 31304 31404], Host: ong.ryan, Venue: Function Room No 4

2.6.2 Delete booking list (for admin only)

For the remove booking feature, admin can choose to delete the booking based on booking ID. Once the booking is removed, it will be updated on the summary list below. As you can see, in the screen shot below, booking ID 3 has been deleted.

Remove Booking

Please enter Booking ID to delete:

Click [here](#) to return to main menu

Below is summary of the booking list:

Booking ID: 0, Bookings for 3 hours: [181109 181209 181309], Host: lim.christina, Venue: Function Room No 9

Booking ID: 1, Bookings for 2 hours: [221203 221303], Host: tan.alison, Venue: Function Room No 3

Booking ID: 2, Bookings for 4 hours: [31104 31204 31304 31404], Host: ong.ryan, Venue: Function Room No 4

Booking ID: 4, Bookings for 3 hours: [201401 201501 201601], Host: lee.matthew, Venue: Function Room No 1

Booking ID: 5, Bookings for 2 hours: [171001 171101], Host: tan.alison, Venue: Function Room No 1

Booking ID: 6, Bookings for 5 hours: [201206 201306 201406 201506 201606], Host: admin, Venue: Function Room No 6

2.6.3 Browse full booking list (for admin only), filter based on user

For the last feature, admin is able to see all the booking that have been done. The admin can also filter the booking based on the host using the drop down list as shown below.

Browse Booking List

Click [here](#) to return to main menu

Please select user that you wish to view.

Below is summary of the booking

Booking ID: 0, Bookings for 3 hours: [181109 181209 181309], Host: lim.christina, Venue: Function Room No 9

Booking ID: 1, Bookings for 2 hours: [221203 221303], Host: tan.alison, Venue: Function Room No 3

Booking ID: 2, Bookings for 4 hours: [31104 31204 31304 31404], Host: ong.ryan, Venue: Function Room No 4

Booking ID: 3, Bookings for 5 hours: [281006 281106 281206 281306 281406], Host: tan.alison, Venue: Function Room No 6

Booking ID: 4, Bookings for 3 hours: [201401 201501 201601], Host: lee.matthew, Venue: Function Room No 1

Booking ID: 5, Bookings for 2 hours: [171001 171101], Host: tan.alison, Venue: Function Room No 1

Booking ID: 6, Bookings for 5 hours: [201206 201306 201406 201506 201606], Host: admin, Venue: Function Room No 6

Below is an example of booking list filtered by username.

Below is summary of the booking list

Booking ID: 1, Bookings for 2 hours: [221203 221303], Host: tan.alison, Venue: Function Room No 3

Booking ID: 3, Bookings for 5 hours: [281006 281106 281206 281306 281406], Host: tan.alison, Venue: Function Room No 6

Booking ID: 5, Bookings for 2 hours: [171001 171101], Host: tan.alison, Venue: Function Room No 1

2.7 Dependency

After the completion of the application creation, go.mod and go.sum are generated using go mod init command line to list down all the dependency of this application. go mod tidy is also called to ensure no unused dependencies is added.

```
Go > src > golnAction1Assignment > go.mod
```

```
Run go mod tidy | Create vendor directory
```

```
1 module goInAction1Assignment
```

```
2
```

```
3 go 1.16
```

```
4
```

```
Check for upgrades | Upgrade transitive dependencies | Upgrade direct dependencies
```

```
5 require (
```

```
6     github.com/satori/go.uuid v1.2.0
```

```
7     golang.org/x/crypto v0.0.0-20210220033148-5ea612d1eb83
```

```
8     gopkg.in/check.v1 v1.0.0-20201130134442-10cb98267c6c // indirect
```

```
9 )
```

```
10
```

```
Go > src > golnAction1Assignment > go.sum
```

```
1 github.com/kr/pretty v0.2.1 h1:Fmg33tUaq4/8ym9TJN1x7sLJnHVwhP33CNkpYV/7rwI=
```

```
2 github.com/kr/pretty v0.2.1/go.mod h1:ipq/a2n7PKx30Hsz4KJIIeveXtP04qwEXGdVfWzfnI=
```

```
3 github.com/kr/pty v1.1.1/go.mod h1:pFQYn66WHR0pPYNljw0Mqo10TkYh1fy3cYio2l3bCsQ=
```

```
4 github.com/kr/text v0.1.0 h1:45sCR5RtLFHMR4UwH9sdQ5TC8v0qDQCHNxt+kaKSTVE=
```

```
5 github.com/kr/text v0.1.0/go.mod h1:4Jbv+DJW3UT/Li0wJeYQe1efqtUx/iVham/4vfdArNI=
```

```
6 github.com/satori/go.uuid v1.2.0 h1:0uYX9dsZ2yD7q2RtLRtPSdGDWzjeM3TbMJP9utgA0ww=
```

```
7 github.com/satori/go.uuid v1.2.0/go.mod h1:dA0hQrYB0VpLJoorgLMZABFdXlWrHn1NE0zdhQKdks0=
```

```
8 golang.org/x/crypto v0.0.0-20190308221718-c2843e01d9a2/go.mod h1:djNgcEr1/C05ACkg1iLfiJ5U5Ep61QUkGW8qpdssI0+w=
```

```
9 golang.org/x/crypto v0.0.0-20210220033148-5ea612d1eb83 h1:/ZScEX8SfEmUGRHs0gxpqte05nfNW6axyZbBdw9A12g=
```

```
10 golang.org/x/crypto v0.0.0-20210220033148-5ea612d1eb83/go.mod h1:jdWPYTVW3xRLrWPugEBEK3UY2ZEsg3UU495nc5E+M+I=
```

```
11 golang.org/x/net v0.0.0-20190404232315-eb5bcb51f2a3/go.mod h1:t9HGtf8HONx5eT2rt9n7q6eTqICYqUVnKs3thJo3Qplg=
```

```
12 golang.org/x/sys v0.0.0-20190215142949-d0b11bdaac8a/go.mod h1:STP8DvDyc/dI5b8T5shstkjS+E42TnysNCUPdjciGhY=
```

```
13 golang.org/x/sys v0.0.0-20191026070338-33540a1f6037/go.mod h1:h1NjWce9XRlGQEsW7wpKNCjG9DtnlCvVFLZEzDNbEs=
```

```
14 golang.org/x/term v0.0.0-20201117132131-f5c789dd3221/go.mod h1:Nr5EML6q2oocZ2LXRh80K7Bx0lk5/8JxuGnuhpl+muw=
```

```
15 golang.org/x/text v0.3.0/go.mod h1:NqM8EU0U14njlkJ3fQMW+pc6Ldnwhi/IjpwHt7yyuw0Q=
```

```
16 gopkg.in/check.v1 v1.0.0-20201130134442-10cb98267c6c h1:Hei/4ADfdWqJk1ZMxUNpqntNwaWcugrBjA1HlqqRiVk=
```

```
17 gopkg.in/check.v1 v1.0.0-20201130134442-10cb98267c6c/go.mod h1:JHKPIbrfpd72SG/EVd6muEfdQjCINN0R0C8j2r3qZ4Q=
```

```
18
```

3 Error handling & Concurrency

3.1 Error handling

Error handling has been introduced to this application especially on the input requested from the user. Besides that, we also insert some error handling for search algorithm. Errors should both reflected on terminal side (server) and also browser side (client). Panic handling has also been incorporated into the application to cover any unexpected error that occurred.

3.2 Reducing room for error

One of the way to reduce error handling is to reduce room for error. These have been considered when building the application. For example, radio and list input type have been used to ensure only desired selection are presented. User has to pick the input from those list. Besides that, all the required field has been marked required so that the user will not submit empty input. Input type pattern matching has also been incorporated for suitable input box.

3.3 Concurrency

For this application, the main concurrency issue that we need to handle is to avoid double booking of a particular slot. Multiple clients could get the similar search result, however, when they submit the booking, the system has to ensure the slot has not been booked. Mutex lock has been used in the application to ensure only one booking goes through at a time, preventing race condition.

Another feature that could benefit from concurrency is to perform several go routines one shot especially in search the availability of 10 different rooms. 10 go routines have been used for this case.