Software Requirements Specification

For

Booking System

Version 1.0 approved

Prepared by B10815044 **Hsieh, Jun-Yao**
B10815054 **Huang, Chen-En**
B10815058 **Pu, Chi-Hao**
B10815057 **Liao, Sheng-Hao**

NTUST-SE-G8

目錄

Revision History

Name	Data	Reason for Change	Version
Draft	2021/11/09	First version	1.0

1. Introduction

1. Purpose

The document is mainly about the Book System's details, such as how it works, operates, and resolves problems. The Book System is suitable for companies to arrange their meeting rooms. Companies can raise meetings or check available meeting rooms through the system, the document will talk about in more detail below.

2. Glossary

Term	Definition
API	Abbreviation of Application Interfaces
UI	Abbreviation of User Interface
Database	It's a system that is used to save data
MySQL	One of SQL databases
User	Normal user can raise or modify a meeting, but can't create meeting rooms
Manager	Manager is one of users, but the manager can create or delete meeting room
Room	Corresponds to the space where the entity exists, so only one event can exist at the same time
Event	Corresponding to a period of time, may be meetings, teaching activities, etc..
Attendee	Participants of the meeting, save email address to send notification

3. Intended Audience and Reading Suggestions

In order to avoid these tragedies, a good room management system is necessary, and the system must be real-time, convenient and stable. The booking system that we have developed can perfectly manage the rooms, events, and personnel to ensure that they do not conflict with each other, and after using this system, you can focus on the content of the event instead of spending a lot of time arranging the venue and adjusting the time.

After reading the first chapter, you will understand the objective and usage situation of the system, after reading the second chapter, you will understand the operation flow and principle of the system, after reading the third chapter, you will understand how the underlying system works, after reading the fourth chapter, you will understand that the system not only has basic functions, but also contains the design made by considering many aspects.

4. Product Scope

This system provides a solution for event or personnel conflict, using information technology to integrate all data together, determine conflict or not, and allow users to quickly know which meetings they have, send reminder emails, and reduce the chance of meeting personnel absence.

5. References

Software Engineering 10th Edition by Ian Sommerville
For more information about software engineering, please check the textbook.

2. Overall Description

1. System Environment

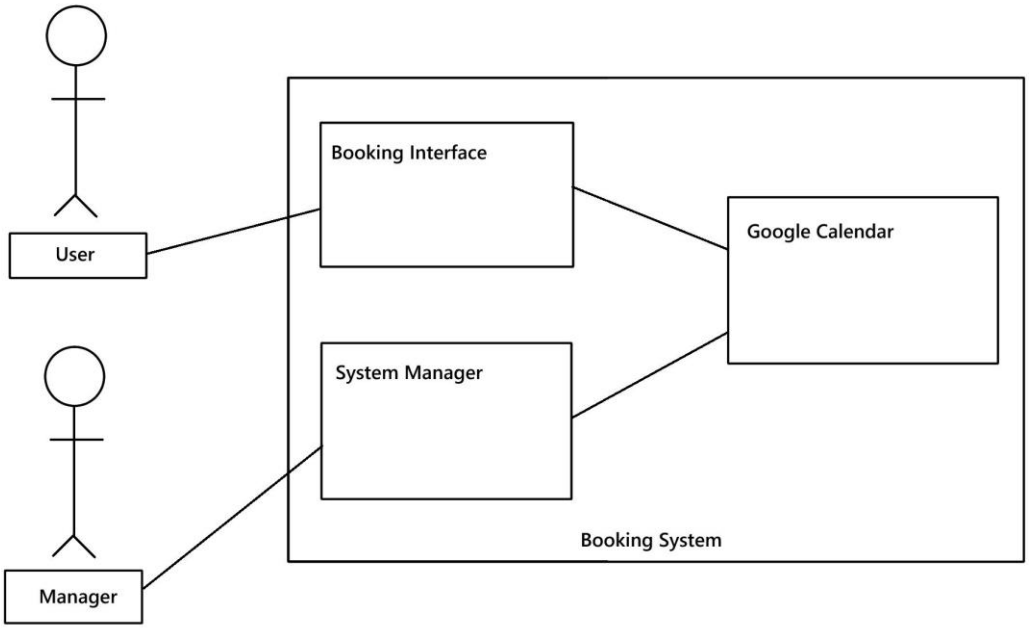


Figure 1 - System Environment

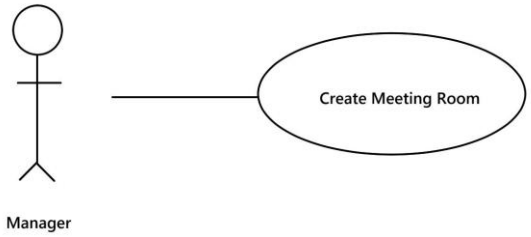
The Booking System will have two different types of users, one is normal user who can book meeting room, create meeting, check meeting they participant. Another is The System Manager who can manage all of the meeting rooms.

2. Functional Requirements Definition

The part descripts the using cases of The Book System, normal users are the main participant.

2.2.1 Manager Use Case

Use case: **Create Meeting Room**
Diagram:



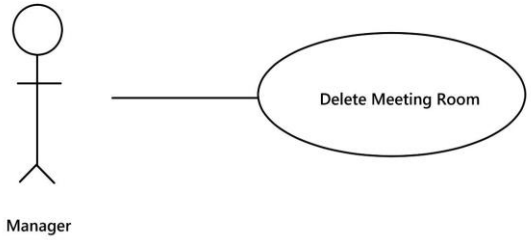
Brief Description
System manager adds the existing room into the meeting room list

Initial Step-By-Step Description
Before this use case can be initiated, the Manager has already accessed the main page of the System Manager.

1. System manager enters the name of the meeting room
2. Press the add button
3. System will add the meeting room into the database and meeting room list

Xerf: Section 3.2.1, Create Meeting Room

Use case: **Delete Meeting Room**
Diagram:



Brief Description
System manager deletes the existing room in the meeting room list

Initial Step-By-Step Description
Before this use case can be initiated, the Manager has already accessed the main page of the System Manager.

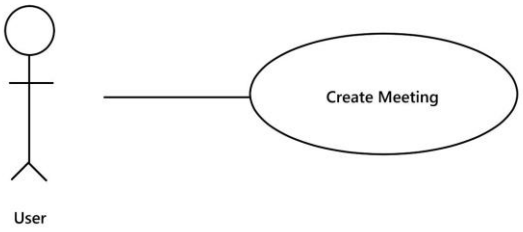
1. Select the meeting that want to be deleted from the list
2. Press the delete button

3. System will delete the meeting room from the database and meeting room list

Xerf: Section 3.2.9, Create Meeting Room

2.2.2 User Use Case

Use case: **Create Meeting**
Diagram:



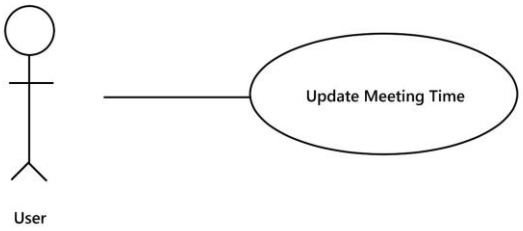
Brief Description
User raises a meeting in the meeting room

Initial Step-By-Step Description
Before this use case can be initiated, the User has already accessed the main page of the Booking Interface.

1. Select the meeting that want to be book from the list
2. UI will show the calendar of the meeting room
3. Select the date from the calendar
4. UI will show the schedule of the date
5. Select unbook date
6. User can raise events there
7. Enter event's info, such as title, hoster email, attendee's email, description
8. Press the confirm button
9. The event will be added into attendee's google calendar and invitation will be sent to attendee

Xref: Section 3.2.2, Create meeting

Use case: **Update Meeting Title**
Diagram:



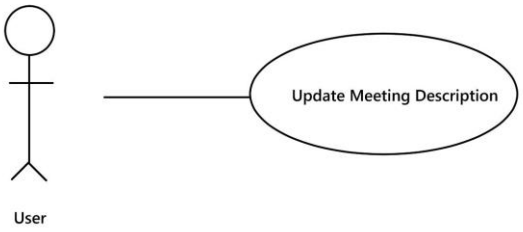
Brief Description
User updates the meeting's title

Initial Step-By-Step Description
Before this use case can be initiated, the User has already accessed the main page of the Booking Interface.

1. Select My meeting that user can browse themselves meeting there
2. UI will list the meetings that the user participant
3. Modify the event
4. UI will show the event's info
5. Modify the event's title
6. Press confirm button
7. The changing in the Book System will also apply to attendee's Google calendar

Xref: Section 3.2.3, Update Meeting title

Use case: **Update Meeting Description**
Diagram:



Brief Description
User updates the meeting's description

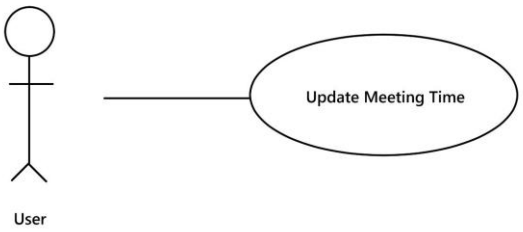
Initial Step-By-Step Description
Before this use case can be initiated, the User has already accessed the main page of the Booking Interface.

1. Select My meeting that user can browse themselves meeting there

2. UI will list the meetings that the user participant
3. Modify the event
4. UI will show the event's info
5. Modify the event's description
6. Press the confirm button
7. The changing in the Book System will also apply to attendee's Google calendar

Xref: Section 3.2.4, Update Meeting Description

Use case: **Update Meeting Time**
Diagram:



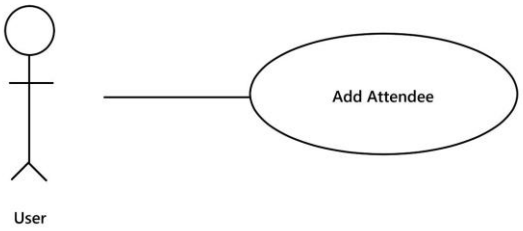
Brief Description
User updates the meeting's time

Initial Step-By-Step Description
Before this use case can be initiated, the User has already accessed the main page of the Booking Interface.

1. Select My meeting that user can browse themselves meeting there
2. UI will list the meetings that the user participant
3. Modify the event
4. UI will show the event's info
5. Modify the event's time
6. Press the confirm button
7. The changing in the Book System will also apply to attendee's Google calendar

Xref: Section 3.2.5, Update Meeting Time

Use case: **Add Attendee**
Diagram:



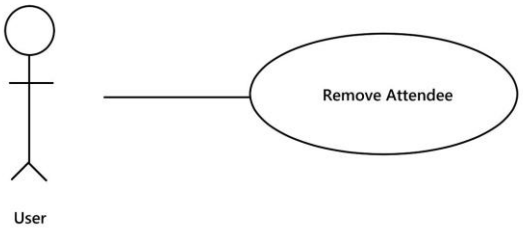
Brief Description
User add a new attendee to the event

Initial Step-By-Step Description
Before this use case can be initiated, the User has already accessed the main page of the Booking Interface.

1. Select My meeting that user can browse themselves meeting there
2. UI will list the meetings that the user participant
3. Modify the event
4. UI will show the event's info
5. Enter new attendee's email
6. Press the add button
7. Press the confirm button
8. The change in the Book System will also apply to attendee's Google calendar, and invitations will be sent to new attendees.

Xref: Section 3.2.6, Add Attendee

Use case: **Remove Attendee**
Diagram:



Brief Description
User remove the meeting in the meeting room

Initial Step-By-Step Description
Before this use case can be initiated, the User has already accessed the main page of the Booking Interface.

1. Select My meeting that user can browse themselves meeting there

2. UI will list the meetings that the user participant
3. Modify the event
4. UI will show the event's info
5. Drop down menu and select the attendee who you want to remove
6. Press the delete button
7. Press the confirm button
8. The change in the Book System will also apply to attendee's Google calendar.

Xref: Section 3.2.7, Remove Attendee

Use case: **Delete Meeting**
Diagram:

Brief Description
User remove a event

Initial Step-By-Step Description
Before this use case can be initiated, the User has already accessed the main page of the Booking Interface.

1. Select My meeting that user can browse themselves meeting there
2. UI will list the meetings that the user participant
3. Modify the event
4. UI will show the event's info
5. Press the remove button
6. The event will be removed into attendee's google calendar

Xref: Section 3.2.8, Delete Meeting

3. User Interface Specifications

Users need to be able to use button, pull-down menu, slider, textbox and other basic software operate skills.

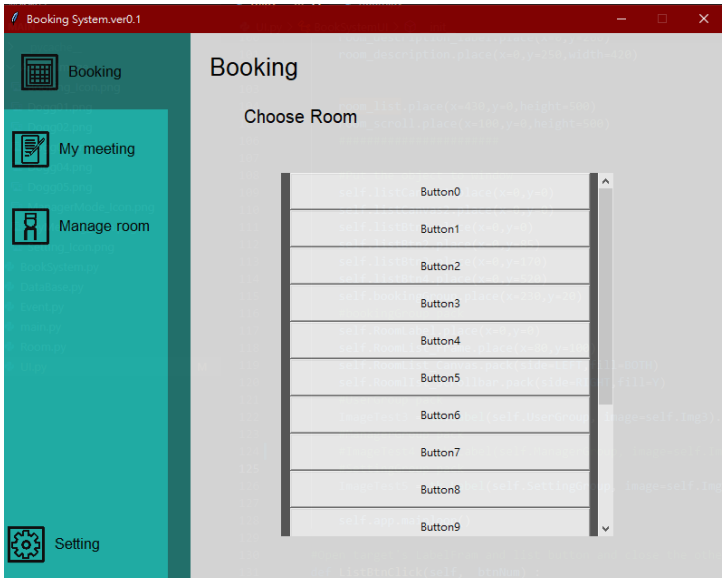


Figure 2 - User Interface

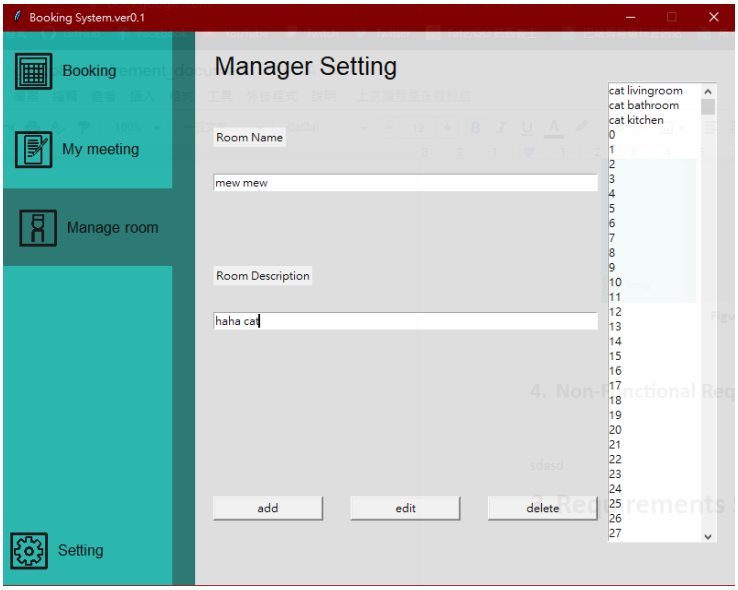


Figure 3 - Manager Interface

4. Non-Functional Requirements

Using the PC platform to execute system executable files, the system will use Google calendar for event management, so both users and administrators need to have a google account.

3. Requirements Specifications

1. External Interface Requirements

Our software will link to 2 external systems. One is the MySQL database. We use it to save data, such as meeting room data, meeting data, and meeting attendees. Another one is Google calendar API, we use this API to modify events in users' Google calendars.

2. Functional Requirements

1. Create Meeting Room

Use case name: Create meeting room

Trigger: System Manager create a meeting room

Precondition: System Manager enters the main UI.

Basic path: 1. Use Google calendar API to create a new calendar as a meeting room.

2. API will respond an ID of the calendar.

3. Using MySQL database to save calendar info.

Postcondition: The meeting room has been created.

Exception paths: System Manager may abandon the operation at any time.

Other: The calendar information includes name and meeting info (empty when created).

2. Create Meeting

Use case name: Create meeting

Trigger: User creates a meeting in the meeting room.

Precondition: User enters the meeting room UI.

Basic path: 1. Use find_calendar function to find which calendar user wants to add a meeting.

2. Function will response an ID of the calendar.

3. Create a meeting in the calendar it returned.

4. Use MySQL database to save meeting info in meeting room

Postcondition: The meeting has been created.

Exception paths: Users may abandon the operation at any time.

Other: The meeting information include name, title, description, attendees, start time and end time.

3. Update Meeting Title

Use case name: Update meeting title

Trigger: User updates title of the meeting

Precondition: User enters meeting UI.

Basic path: 1. Use find_calendar function to find calendar.

2. Use find_event function to find event that user wants to be updated.

3. Update meeting title in calendar.

4. Update the meeting information in MySQL database.

Postcondition: The title of the meeting has been updated.

Exception paths: Users may abandon the operation at any time.

4. Update Meeting Description

Use case name: Update meeting description

Trigger: User updates description of the meeting

Precondition: User enters meeting UI.

Basic path: 1. Use find_calendar function to find calendar.

2. Use find_event function to find event that user wants to be updated.

3. Update meeting description in calendar.

4. Update the meeting information in MySQL database.

Postcondition: The description of the meeting has been updated.

Exception paths: Users may abandon the operation at any time.

5. Update Meeting Time

Use case name: Update meeting time

Trigger: User updates time of the meeting.

Precondition: User enters meeting UI.

Basic path: 1. Use find_calendar function to find calendar.

2. Use find_event function to find the event that the user wants to be updated.

3. Update meeting time in calendar.

4. Update the meeting information in MySQL database.

Postcondition: The time of the meeting has been updated.

Exception paths: Users may abandon the operation at any time.

6. Add Attendee

Use case name: add attendee

Trigger: User adds attendee of the meeting.

Precondition: User meeting UI.

Basic path: 1. Use find_calendar function to find calendar.

2. Use find_event function to find events that the user wants to add attendees into.

3. Update meeting attendee in calendar.

4. Update the meeting information in MySQL database.

Postcondition: The attendee has been added into the meeting.

Exception paths: Users may abandon the operation at any time.

Other: Attendee information includes Email only.

7. Remove Attendee

Use case name: Remove attendee

Trigger: User removes attendee of the meeting.

Precondition: User enters the meeting UI.

Basic path: 1. Use find_calendar function to find calendar.

2. Use find_event function to find the event.

3. Use find_attendee function to find attendee to be removed.

4. Update meeting information in calendar.

5. Update the meeting information in MySQL database.

Postcondition: The attendee has been removed from the meeting.

Exception paths: Users may abandon the operation at any time.

Other: Attendee information includes Email only.

8. Delete Meeting

Use case name: Delete meeting

Trigger: User Deletes the meeting.

Precondition: User enters meeting UI.

Basic path: 1. Use find_calendar function to find calendar.

2. Use find_event function to find the event that the user wants to delete.

3. Use Google calendar API to delete the event.

4. Remove the meeting information in MySQL database.

Postcondition: The meeting has been deleted.

Exception paths: Users may abandon the operation at any time.

9. Delete Meeting Room

Use case name: Delete meeting room

Trigger: System Manager deletes the selected meeting room.

Precondition: System Manager enters the meeting room UI.

Basic path: 1. Use find_calendar function to find the calendar that the System Manager wants to delete.

2. Use Google calendar API to delete calendar.

3. Remove all the meeting room information includes all events in the calendar in MySQL database.

Postcondition: The meeting room has been deleted.

Exception paths: System Manager may abandon the operation at any time

4. Other Nonfunctional Requirements

1. Performance Requirements

Make sure the database won't be stuck when many users are using the program at the same time. Maybe need load balance if too many users access the database at the same time.

2. Safety Requirements

We need to try our best to prevent users from doing irreversible or illegal operations, such as delete the database or access others data. Besides, it's better to back-up the database once in a while.

3. Security Requirements

We need to protect the client and the meeting’s information. There are some ways we can do this, one is to encrypt the data we transfer, another is to verify the user's identity before we access the database.