

Software Requirements Specification

for

Virtual Room Reservation Assistant System

Version 1.3

Prepared by

Group 21

1. 李柏賢 B10815007
2. 羅翊幸 B10815009
3. 葉小婷 B10815011
4. 范恩琪 B10815013
5. 施佳妘 B10815014

National Taiwan University of Science and Technology (NTUST)

November 8, 2021

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document. Software Requirements Specification for Virtual Room Reservation Assistant System

Table of Contents

Table of Contents	2
Introduction	4
1.1 Purpose	4
1.2 Glossary	4
1.3 Intended Audience and Reading Suggestions	4
1.4 Product Scope	5
1.5 References	5
Overall Description	6
2.1 System Environment	6
2.2 Functional Requirements Definition	7
Use case: Log in	7
Use case: Register	7
Use case: Log out	7
Use case: Reservation	8
Use case: Edit	8
Use case: Room Status	9
Use case: Profile	9
Use case: Delete Account	9
2.3 User Interface Specifications	10
2.4 Non-Functional Requirements	14
Requirement Specifications	14
3.1 External Interface Requirements	14
3.2 Functional Requirements	14
Other Non-functional Requirements	20
4.1 Performance Requirements	20
4.2 Safety Requirements	20
4.3 Security Requirements	20

Revision History

Name	Date	Reason For Changes	Version
范恩琪	19/01/2022	<ol style="list-style-type: none">1. Add several new use cases and delete several use cases2. Changes in User Interface Specifications3. Changes in Functional Requirements	1.1
羅翊幸 葉小婷	20/01/2022	<ol style="list-style-type: none">1. Changes in System Environment2. Changes in External Interface Requirements3. Changes in several use cases	1.2
李柏賢	21/01/2022	<ol style="list-style-type: none">1. Add some UI images	1.3

1. Introduction

1.1 Purpose

This document serves as a thorough description of the Virtual Room Reservation Assistant System. It covers the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2 Glossary

The table below provides a brief description of the technical terms used in this document.

Term	Meaning
User Profile	An account that is created per user at the moment registration is done. This user profile can then store the user's reservation data.
Participants	People who are invited to the specified meeting session.
Database	Collection of all the information which is monitored by this system.
Graphical User Interface (GUI)	An interface which allows interaction between the user and the system.

1.3 Intended Audience and Reading Suggestions

This document is intended for individuals who are directly involved in the development of the Virtual Room Reservation Assistant System. This document need not be read sequentially; users are encouraged to jump to any section they find relevant. Below is the brief overview of each part of the document.

- Part 1 (Introduction)
This section will briefly explain Virtual Room Reservation Assistant System, including the purpose, project scope, and general system details.
- Part 2 (Overall Description)
This section will explain about the system flow and all the details related to the structure of Graphical User Interface (GUI).
- Part 3 (Requirement Specifications)
This section will cover all the requirements needed such as external interface requirements which will also include all the interface used, and functional requirements.
- Part 4 (Other Non-functional Requirements)
This section will give information about the software's performance, safety, and security requirements.

1.4 Product Scope

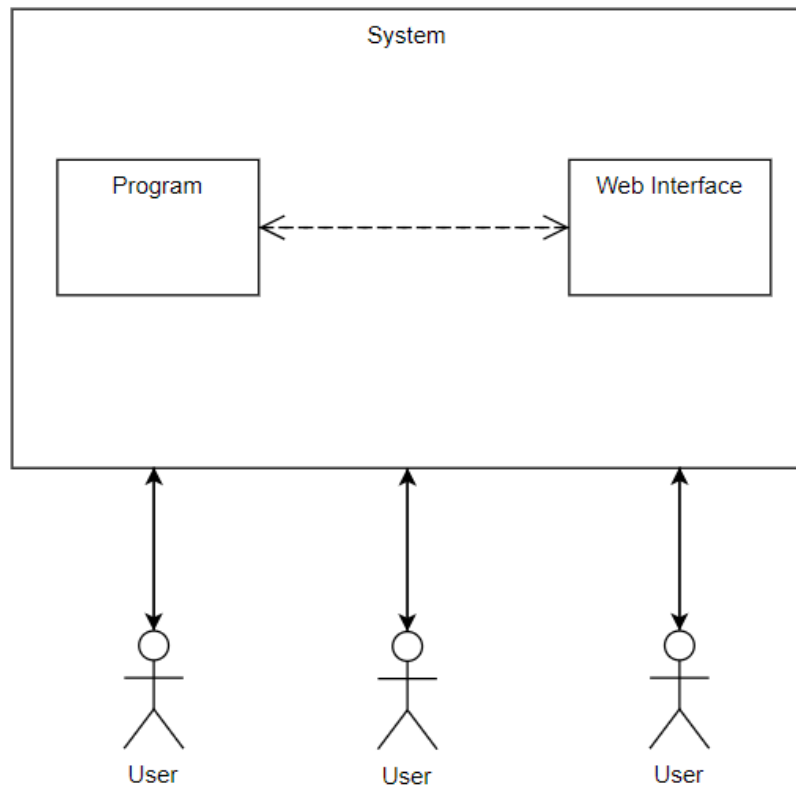
The goal of this product is to provide the users with a room reservation assistant to help manage meeting rooms. By using this product, users are able to check the rooms' current and/or upcoming availability and easily book those that are available within their time slot. The scope of interest of this product lies in the making of the booking system and not focusing on registering the available rooms to the system.

1.5 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2. Overall Description

2.1 System Environment



The interface for our software is the web interface, which runs in a local network using the web browser so that the users can interact with the running program.

2.2 Functional Requirements Definition

Use case: Log in

Brief Description

Users log into their account by entering their username and password.

Initial Step-by-Step Description

1. User click log in / register button
2. Click log in
3. Server receive user's request
4. Server ask user for username and password
5. User input username and password
6. Server receive user's inputs and log user to their account

Use case: Register

Brief Description

Users create a new account by entering their username, email and password.

Initial Step-by-Step Description

1. User click login / register button
2. Click register
3. Server receive user's request
4. Server ask user for username, email and password
5. User input username, email and password
6. Server receive user's inputs and create an account for the user

Use case: Log out

Brief Description

Users log out of their account.

Initial Step-by-Step Description

1. User click login / register button
2. Click Log out
3. Server receive user's request

4. Server log user out of their account

Use case: Reservation

Brief Description

Users reserve a room that is available.

Initial Step-by-Step Description

1. User select the room they desire, time and date
2. Server receive user's request
3. Server will check for the room availability for selected time and date
4. If available, user's reservation is success and will be recorded into their account
5. If not available, user's request will not be recorded into their account

Use case: Edit

Brief Description

Users can edit the room, date and time of their upcoming reservation. Users can also cancel their reservation from this page.

Initial Step-by-Step Description

1. User select which reservation will be edited
2. User edit selected reservation room, date or time
3. Server receive user's request
4. Server will change user's reservation

Use case: Cancel

Brief Description

Users cancel a room that they have reserved from the edit page.

Initial Step-by-Step Description

1. Server will list the user's upcoming reservations
2. User will select which reservation they want to cancel
3. Server receive user's request
4. Server will cancel the reservation for the user

Use case: Room Status**Brief Description**

Users can check the availability of each room during a certain date and time.

Initial Step-by-Step Description

1. User select the date to check the availability
2. Server receive user's request
3. Server will show the room availability based on the date

Use case: Profile**Brief Description**

Users can edit their account by changing password or delete account.

Initial Step-by-Step Description

1. User can delete account by clicking the button
2. Server receive user's request
3. Server will delete user's account from database

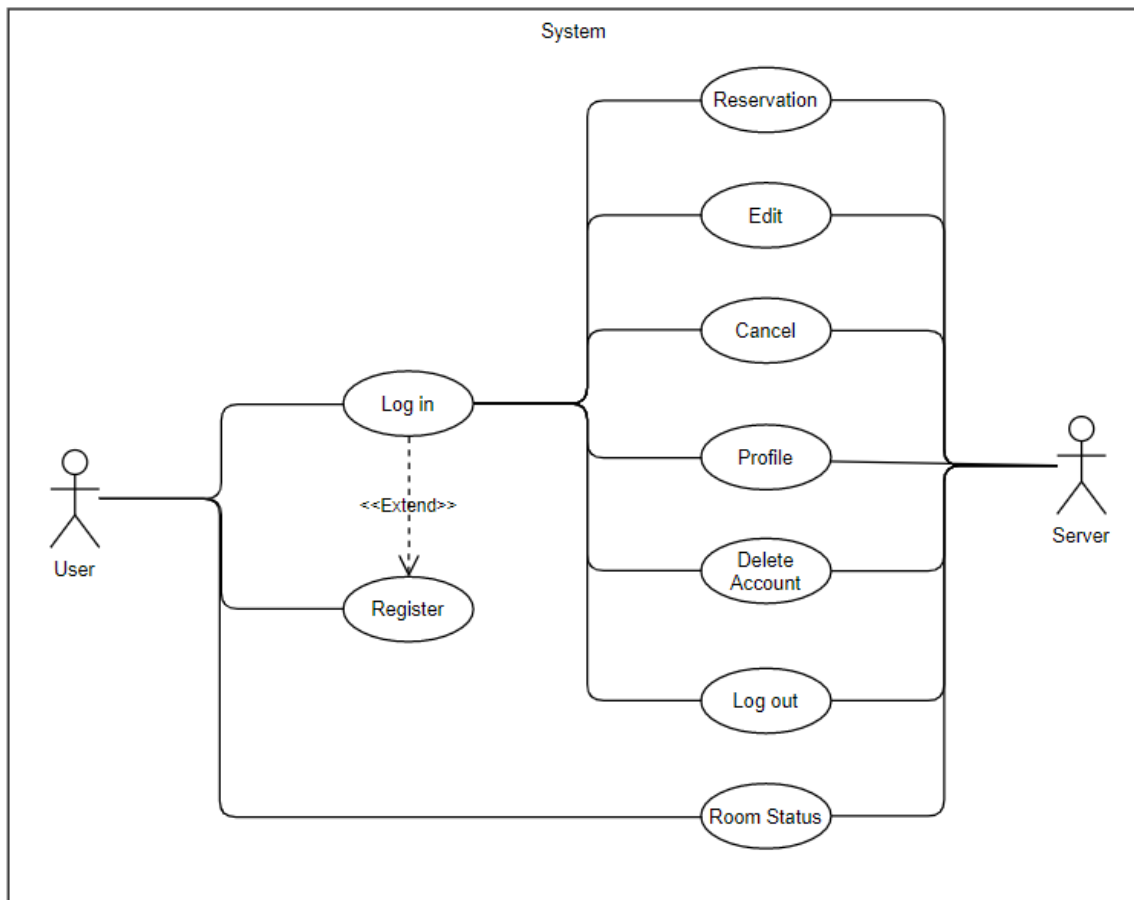
Use case: Delete Account**Brief Description**

Users can delete their account by clicking the delete account button.

Initial Step-by-Step Description

1. User can delete account by clicking the button
2. Server receive user's request
3. Server will delete user's account from database

Use case diagram



2.3 User Interface Specifications

1. Home page



Fig. 2.3.1 Home Page

- a. A “Home” button to go back to main page

- b. A “Reservation” button to go to Reservation page
- c. A “Edit” button to go to the edit page where users can edit their upcoming reservation or cancel their upcoming reservation.
- d. A “Room Status” button to go to the room status page where users can check each room availability.
- e. A “Profile” button to go to the profile page where the user can edit/delete the account.
- f. A “Login” button which will pop up, containing:
 - i. A field for users to enter username.
 - ii. A field for users to enter passwords.
 - iii. A “Login” button which will log users to their account.
 - iv. A “Cancel” button which will take the user back to the page before.
- g. A “Register” button which will pop up, containing:
 - i. A field for users to enter username.
 - ii. A field for users to enter email.
 - iii. A field for users to enter passwords.
 - iv. A “Register” button which will create a new account for users.
 - v. A “Cancel” button which will take the user back to the page before.
- h. A “Logout” button which will log the user out of their account.

2. Reservation page

The screenshot shows a web application interface for room reservations. On the left is a dark sidebar with navigation links: Home, Reservation, Edit, Room Status, and Profile. The main content area is titled 'New Reservation' and contains a form with the following fields:

- Subject:** A text input field containing 'Meeting'.
- Message/Details:** A text input field containing 'Software Engineering'.
- Room:** A radio button selection with options R1 (selected), R2, R3, R4, and R5.
- Participants:** A text input field with the placeholder 'Search for names...' and a small 'Enq' button.
- Date:** A date picker showing '01/20/2022'.
- Time:** A time range selector showing '16:00' to '17:00'.

At the bottom of the form are two buttons: a green 'Submit' button and a red 'Clear Form' button.

Fig. 2.3.2 Reservation Page

- a. A radios option where user can select which room they desire
- b. A field to select participant
- c. A field to add meeting details or message
- d. A field to select date
- e. A field to select time
- f. A “Submit” button to proceed user’s request
- g. A “Clear Form” button to clear the form from all input

3. Edit page

Edit Reservation [ID5]

Subject

Message/Details

Room

☒ R1
☐ R2
☐ R3
☐ R4
☐ R5

Participants ardine
ardine ☒

Date

Time -

Fig. 2.3.3 Edit Page

- a. A table which will show the user's upcoming reservation and the user needs to select one of the room edit buttons or click on the cancel reservation button.
- b. A radios option where user can select new room
- c. A field to select new date
- d. A field to select new time
- e. A “Submit” button to proceed user’s request
- f. A “Clear Form” button to clear the form from all input

4. Room Status page

ROOM RESERVATION

Home
 Reservation
 Edit
 Room Status
 Profile

Date:

	Room 1	Room 2	Room 3	Room 4	Room 5
8:00 - 9:00			/NA		
9:00 - 10:00			/NA		
10:00 - 11:00			/NA		
11:00 - 12:00			/NA		
12:00 - 13:00					
13:00 - 14:00		/NA			
14:00 - 15:00					
15:00 - 16:00	/NA				
16:00 - 17:00					
17:00 - 18:00					
18:00 - 19:00					
19:00 - 20:00			/NA		
20:00 - 21:00					
21:00 - 22:00		/NA			
22:00 - 23:00					
23:00 - 24:00					

@2021 Software Engineering

Group 21

Fig. 2.3.4 Room Status Page

- A table which will show the selected rooms' status corresponding to the selected date.
- A field to select a date
- A "Submit" button to proceed user's request

5. Profile page

ROOM RESERVATION

ingrid

Home

Reservation

Edit

Room Status

Profile

ingrid's Profile

History

No records available

[Make a reservation](#)

Options

Click here to delete account

Fig. 2.3.5 User's profile page with reservation history (empty)

- A list of user's reservations.
- A "Delete Account" button to delete the user's account.

2.4 Non-Functional Requirements

Software will at least require 2GB memory. Every use case should not take more than 5 seconds.

3. Requirement Specifications

3.1 External Interface Requirements

The system utilizes Gmail emailing services to be able to notify users of the reservations and other activities that have been done, as in the System Admin is logged in using a Gmail Account. The system however **does not** provide Google Account integration.

3.2 Functional Requirements

1. Login

Use case name	Login
Trigger	The user clicks on the “Login” button.
Precondition	The user clicks the “Login / Register” button, which contains a “Login” button.
Basic Path	<ol style="list-style-type: none">1. The server will redirect the user to a <i>Login</i> page.2. The user fills in their username and password.3. The user clicks on the “Login” button.4. Server will log users into their account.
Alternative Path	<p>In step 3, if the system fails to get an input from either input field, the system will flash a message on the said field, prompting the user for an input.</p> <p>In step 3, if the system fails to find a match between the user input and the database, the system will flash an error message corresponding to the missing field.</p>

Postcondition	Users will be logged into their account.
Exception Path	The operation may be abandoned at any time (the database remains unchanged).

2. Register

Use case name	Register
Trigger	The user clicks on the “Register” button
Precondition	The user clicks the “Login / Register” button, which contains a “Register” button.
Basic Path	<ol style="list-style-type: none"> 1. The server will redirect the user to a <i>Register</i> page. 2. The user fills in their name, username, email address, and password. 3. The user clicks on the “Register” button. 4. Server will create a new account for the user.
Alternative Path	<p>In step 3, if the system fails to get an input from either input field, the system will flash a message on the said field, prompting the user for an input.</p> <p>In step 3, if the system finds a duplicate of either a username or an email, the system will notify the user of the error, preventing the user from using the duplicates. (Users have to use unique usernames and email addresses.)</p>
Postcondition	User’s account will be created and they will be automatically logged in.
Exception Path	The operation may be abandoned at any time (the database remains unchanged).

3. Logout

Use case name	Logout
---------------	--------

Trigger	The user clicks the “Logout” button.
Precondition	The user clicks on their profile button, available at the top-left corner of the page, then selecting the “Logout” option.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks on the “Logout” button. 2. Server will log users out of their account.
Alternative Path	none
Postcondition	Users will be logged out of their account.
Exception Path	The operation may be abandoned at any time (the database remains unchanged).

4. Reservation

Use case name	Reservation
Trigger	The user clicks the “Submit” button.
Precondition	The user on <i>Reservation Form</i> page
Basic Path	<ol style="list-style-type: none"> 1. The user fills in all the necessary date on the reservation form. 2. Server checks for the room availability for the selected time and date. 3. Server records the reservation to the database.
Alternative Path	If the room is unavailable at that certain time and date, then the server will notify the user of the error and the recently made reservation won’t be recorded into the database.
Postcondition	User’s reservation will be recorded into the database and added into the user's profile.
Exception Path	The operation may be abandoned at any time (the

	database remains unchanged).
--	------------------------------

5. Cancel

Use case name	Cancel
Trigger	The user clicks the “Cancel” button.
Precondition	The user on the <i>Edit</i> page.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks on the “Cancel” option of the to-be-cancelled reservation. 2. The server will check the reservation. 3. The server will cancel the user's reservation and update the database.
Alternative Path	None
Postcondition	User’s reservation will be deleted from the database and the previously occupied space of that room will be available for future reservations.
Exception Path	The operation may be abandoned at any time (the database remains unchanged).

6. Edit

Use case name	Edit
Trigger	The user clicks on the “Submit” button.
Precondition	The user on the <i>Edit</i> page.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks on the “Edit” option of the to-be-edited reservation. 2. The user edits/modifies the reservation data. 3. The server will check the room availability of the modified reservation data. 4. The server will update the reservation upon the

	<p>user's request.</p> <p>5. The server will update the database.</p>
Alternative Path	The system will notify the user if the selected time slot for the room is unavailable.
Postcondition	User's reservation information will be updated.
Exception Path	The operation may be abandoned at any time (the database remains unchanged).

7. Room Status

Use case name	Room Status
Trigger	The user clicks the "Room Status" button.
Precondition	The user is on the <i>Home</i> page which has a "Room Status" button.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks on the "Room Status" button. 2. Server will show today's room availability on different time periods in a table format.
Alternative Path	The user alters the date on the date input and clicks the "Update" button. The server will then show the room availability on the new given date.
Postcondition	Table output of the room availability on different time periods.
Exception Path	The operation may be abandoned at any time (the database remains unchanged).

8. Profile

Use case name	Profile
Trigger	The user clicks the "Profile" button.

Precondition	The user is on the <i>Home</i> page which has a “Profile” button.
Basic Path	<ol style="list-style-type: none"> 1. Users click the “Profile” button. 2. Server will list all user’s reservations. 3. Users can delete accounts.
Alternative Path	none
Postcondition	List of user’s reservations with their current statuses.
Exception Path	The operation may be abandoned at any time (the database remains unchanged).

9. Delete Account

Use case name	Delete Account
Trigger	The user clicks the “Delete Account” button.
Precondition	The user is on the <i>Profile</i> page
Basic Path	<ol style="list-style-type: none"> 1. Users click the “Delete Account” button. 2. Server will delete the user's account and update the database. 3. User account deleted.
Alternative Path	None
Postcondition	User account deleted
Exception Path	The operation may be abandoned at any time (the database remains unchanged).

4. Other Non-functional Requirements

4.1 Performance Requirements

The system is available for use when run locally on the user's browser through some terminal commands. All of the UI elements will respond to the user's input immediately.

4.2 Safety Requirements

- Ensure that there no loss of personal data
- Ensure that the room is available for booking, if there is any emergency maintenance, then inform the user that the booking is canceled due to maintenance.
- Ensure that when 2 users at the same time are booking the same room for the same time slot, there should be a time duration for the first user to do their booking before the second user can do the booking.

4.3 Security Requirements

The system requires users to register an account before being given access to the reservation system. Passwords will be encrypted before recording it into the database. The system will not monitor or collect any other data, excluding the user inputs.