Software Requirements Specification

For

TEMPO

San Hae Kim (110398442)

sanhae.kim@stonybrook.edu

Kyu Hee Park (110398044)

kyuhee.park@stonybrook.edu

Dong Yeob Lee (109626747)

dongyeob.lee.1@stonybrook.edu

Table of Contents

1. Introduction

1.1 Product Description

Currently, IGC students have trouble using school facilities such as study rooms and the music room. Students have to reserve the facility by signing up on the reservation sheet (paper) before they use them. They do not know if the room is available or not before they go sign up. This is a waste of time. Sometimes, students have conflicts with other students as the reservation system is unclear.

Therefore, we are going to create an online reservation website for university facilities. It allows students to see the status of facilities and reserve them online. They can reserve space that does not require special approval such as study rooms directly. For rooms that need approval from the staff, students can send a request form to the staff. This will save students' time and energy, and they do not have to worry about conflicts due to a double reservation.

Furthermore, school staff can use the website to manage the status of facilities. They can add or remove facilities from the list. They can approve or decline the request to use rooms that need special authorization. They can view and edit the reservation status of facilities. With our website, it will be much easier for them to keep track of reservation records.

1.2 Existing Alternatives

IGC and a lot of universities use a paper for students to reserve the facilities. By using the sheet, the students can easily see the schedule and make reservations. The students, however, can only check by actually go and see if the rooms are available. For IGC, both dormitories have reserving sheets, so the students get a conflict of time and room. Also, it is hard for the staff to keep track of records since papers can be lost or damaged easily.

Some universities have online registration forms for their own campus facilities. They, however, are only limited to the facilities that require approval from the staff. Also, students cannot check the availability of the facilities.

With our system, students can reserve online by checking the status of the facilities. They do not have to go to the rooms that they want to use; they can save their time. Moreover, the conflicts between students will not happen anymore.

1.3 Scope

The system will only be provided for students and staff at the university. The system will easily accept a new university campus. The staff can manage the facilities without writing any code; they can simply log in with their account to manage them.

1.4 Documentation & Support

Our system will provide the user manual for the users to refer to. It will be two different user manuals: one for students and another one for the staff. It will contain a step-by-step guide with captured images of our system.

2. Features & Requirements

2.1 Main Features

Students

- Sign Up
- Login/logout
- Manual Page
- Reservation System
    - Reservation without Approval
    - Reservation with Approval
- Facility Status Check
- My Page
    - Check reservation status
    - Manage reservations
    - Check Past Reservation
    - View penalty status
    - Report system

Admin (Staff)
- Sign Up
- Login/logout
- Manual Page
- Reservation System
    - Add, delete, and edit facilities
    - Approve and decline reservation requests
    - Cancel reservations
    - Mark off time slots
    - Check Past Reservation
- Facility Status Check
- Penalty System
    - View students' reports
    - Give/remove penalty to/from students

Stretch features
- Location of facilities on a map
- Reservation Waitlist (up to 3 teams)
- Information page
    - School events, shuttle bus, course schedule, etc

2.2 Use Cases

Use Case:	Sign Up
Primary Actor:	User
Goal in Context:	The user wants to register for the website.
Preconditions:	The user is viewing the login page without anything typed.
Scenario:	1. The user clicks the 'Register' button on the top right corner. 2. The system gives a form to fill in. 3. After filling in information, the user clicks the 'Complete' button.
Extensions:	2a. When the user fails to fill in the form, the system gives an error message to fill in accurately.
Postconditions:	The user is viewing the login page.

Use Case:	Login
Primary Actor:	User
Goal in Context:	The user wants to log in to his/her account.
Preconditions:	The user is viewing the login page without anything typed.
Scenario:	4. The user types his/her ID(school email address) and Password. 5. The user clicks the 'Login' button.

Extensions:	2a. When the user fails to login, a message will appear 'incorrect ID or Password' on the login page.
Postconditions:	The user is viewing the main page.

Use Case:	Logout
Primary Actor:	User
Goal in Context:	The user wants to logout from his/her account.
Preconditions:	From any screens, the user can see the 'Log Out' button on the top right corner of the website.
Scenario:	1. The 'Log Out' button is always displayed on the top right of the website. 2. The user clicks 'Log Out'.
Extension:	2a. When the user press 'Log Out', any data that was being edited will be unsaved.
Postcondition:	The user is viewing the Login page.

Use Case:	User Manual Page
Primary Actor:	User
Goal in Context:	The user manual page helps the users to understand how to use the website.
Preconditions:	The user has clicked to the 'Manual Page'.

Scenario:	1. When 'Manual Page' is clicked, the manual page will appear on the display.
	2. There will be only a few manuals provided, but enough for users to understand when a similar situation occurs.
Extension:	1.1. The user can see one of the manuals by opening on a new web tab.

Use Case:	Reservation without approval (Dorm Page)
Primary Actor:	User (student)
Goal in Context:	The user wants to reserve a study room.
Preconditions:	The user has clicked the 'Dorm' from the facility list.
Scenario:	1. The user is viewing a list of study rooms available in the dorm.
	2. The user selects a study room that he/she wants.
	3. The screen shows a timetable of the study room that the user chose.
	4. The user selects a time slot that he/she wants.
	5. The user fills in the details (number of members, members' email addresses).
	6. The user clicks the 'Complete' button to complete a reservation.
Postconditions:	The screen goes back to the timetable.

Use Case:	Reservation with approval (music room)
Primary Actor:	User (student)
Goal in Context:	The user wants to reserve a music room.

Preconditions:	The user has clicked the 'Music room' from the facility list.
Scenario:	1. The user is viewing a music room from the Multi-Complex facility list. 2. The user selects the music room. 3. The screen shows a timetable of the music room. 4. The user selects a time slot that he/she wants. (max. 2 slots) 5. When selecting the time, the user will see the form on the screen. 6. The user fills the form with the details (number of people, why the room is needed etc.). 7. The user clicks the 'Submit' button to complete a reservation.
Extensions:	4-7a. The user selects the 'Cancel' button. 4-7a.1. The screen shows the timetable. 6a. The user did not fill out the details. 6a.1. An alert will show on the screen.
Postconditions:	The screen goes back to the timetable.

Use Case:	My Page (Check the Reservation)
Primary Actor:	User (student)
Goal in Context:	The user wants to check the status of the submitted form.
Preconditions:	The user has clicked the 'My Page' from the tab.
Scenario:	1. The user sees the page with different tabs on the left. 2. The user selects 'Current reservation' on the tab. 3. The user can check if the submitted form for the approval facility reservation is accepted or not.

Use Case:	My Page (Cancel the reservation)
Primary Actor:	User (student)
Goal in Context:	The user wants to cancel the reservation.
Preconditions:	The user has clicked the 'My Page' from the tab.
Scenario:	1. The user sees the page with different tabs on the left. 2. The user selects 'Current Reservation'. 3. The user selects the 'Cancel' button, then the reservation will be deleted from the list.

Use Case:	My Page (Check the past reservation)
Primary Actor:	User (student)
Goal in Context:	The user wants to see the past reservation record.
Preconditions:	The user has clicked the 'My Page' from the tab.
Scenario:	1. The user sees different tabs on the left side of the screen. 2. The user selects the 'Past Reservation' button. 3. The screen shows the record of the past reservation.

Use Case:	My Page (View the penalty status)
Primary Actor:	User (student)
Goal in Context:	The user wants to see the penalty status

Preconditions:	The user has clicked the 'My Page' from the tab.
Scenario:	1. The user sees different tabs on the left side of the screen. 2. The user selects the 'Penalty' button. 3. The screen shows the status of penalty that the user received.

Use Case:	Add the Facility to the list
Primary Actor:	User (admin)
Goal in Context:	The user wants to add the facility to the list.
Preconditions:	The user is viewing the facility list on the reservation page after clicking 'Reservation with Approval' or 'Reservation without Approval' button.
Scenario:	1. The user clicks 'Manage the Facility' button. 2. The user clicks '+' button. 3. The user fills in the details: the title and the category 4. The user clicks 'Complete' button.
Postconditions:	The user is viewing the facility list with the facility that he/she just added.

Use Case:	Delete the Facility from the list
Primary Actor:	User (admin)
Goal in Context:	The user wants to delete the facility from the list.
Preconditions:	The user is viewing the facility list on the reservation page after clicking 'Reservation with Approval' or 'Reservation without Approval' button.

Scenario:	1. The user clicks 'Manage the Facility' button.
	2. The user clicks 'X' button next to the title of facility that he/she wants to delete.
	3. The system gives the alert message: 'Do you really want to delete this facility?'
	4. The user clicks 'Yes' button.
Postconditions:	The user is viewing the facility list without the facility that the user just deleted.

Use Case:	Edit the Facility in the list
Primary Actor:	User (admin)
Goal in Context:	The user wants to edit the facility in the list.
Preconditions:	The user is viewing the facility list on the reservation page after clicking 'Reservation with Approval' or 'Reservation without Approval' button.
Scenario:	1. The user clicks 'Manage the Facility' button.
	2. The user clicks 'Edit' button next to the title of facility that he/she wants to delete.
	3. The user can edit the facility
Postconditions:	The user is viewing the facility list with the edition that the user made

Use Case:	Change the status of facility
Primary Actor:	User (admin)
Goal in Context:	The user wants to change the facility status.

Preconditions:	The user is viewing the facility list on the reservation page after clicking 'Reservation with Approval' or 'Reservation without Approval' button.
Scenario:	1. The user clicks 'Manage the Facility' button. 2. The user clicks the 'Status' button next to the facility that the user wants to change. 3. The user can select the status through radio buttons: 'Available' and 'Out of Order'.
Postconditions:	The user is viewing the facility list with the facility that he/she just changed.

Use Case:	Cancel the student's reservation (Reservation without approval)
Primary Actor:	User (admin)
Goal in Context:	The user wants to cancel the student's reservation. (Reservation without approval)
Preconditions:	The user is viewing the facility list on the reservation page after clicking 'Reservation without Approval' button.
Scenario:	1. The user clicks the facility that he/she wants to manage. 2. The timetable is shown, and the user clicks the time slot that he/she wants to manage. 3. The user clicks the 'Cancel the reservation' button. 4. The user writes the reason for canceling the reservation. 5. The user clicks the 'Complete' button.
Extensions:	2a. When the user clicks the 'Cancel' button, the user goes back to the facility list. 4a. If the user does not write the reason, the system gives an alert message.
Postconditions:	The user is viewing the timetable with the time slot that he/she just canceled empty.

Use Case:	Approve the student's reservation (Reservation with approval)
Primary Actor:	User (admin)
Goal in Context:	The user wants to approve the student's reservation. (Reservation with approval)
Preconditions:	The user is logged in.
Scenario:	1. The user clicks the 'Message Box' on the top right corner of the page. 2. The system shows the request forms that students sent. 3. The user selects 'Approve' button next to the message that he/she wants to approve.
Extensions:	3a. The user selects the 'Decline' button next to the message that he/she wants to decline.
Postconditions:	The user is viewing the message box.

Use Case:	Cancel the student's reservation (Reservation with approval)
Primary Actor:	User (admin)
Goal in Context:	The user wants to cancel the student's reservation. (Reservation with approval)
Preconditions:	The user is viewing the facility list on the reservation page after clicking the 'Reservation with Approval' button.
Scenario:	1. The user clicks the facility that he/she wants to manage. 2. The timetable is shown, and the user clicks the time slot that he/she wants to manage. 3. The user clicks the 'Cancel the reservation' button.

	4. The user writes the reason for canceling the reservation. 5. The user clicks the 'Complete' button.
Extensions:	2a. When the user clicks the 'Cancel' button, the user goes back to the facility list. 4a. If the user does not write the reason, the system gives an alert message.
Postconditions:	The user is viewing the timetable with the time slot that he/she just canceled empty.

Use Case:	View the penalty status
Primary Actor:	User (admin)
Goal in Context:	The user wants to view the penalty status of students.
Preconditions:	The user is viewing the penalty page
Scenario:	1. The user can select among the tabs: students with 3 strikes, 2 strikes, and 1 strike. 2. The screen shows the list of students with according number of strikes.
Postconditions:	The screen shows the list of students with according number of strikes.

Use Case:	Give penalty to students
Primary Actor:	User (admin)
Goal in Context:	The user wants to give the penalty to students
Preconditions:	The user is viewing the penalty page.

Scenario:	1. The user clicks 'Give a penalty' button.
	2. The user writes the details: name of the student and the reason.
	3. The user clicks the 'Complete' button.
Postconditions:	The penalty of the student that the user just sent is increased.

2.3 User Interfaces

Please refer to a separate file called, 'user_interface.pdf'

2.4 Non-functional Requirements

Usability

- The website should function in any browsers and mobile devices.
- The user manual should be easily understandable.
- The reservation system should be simple and straightforward to use.
- Students should easily see their reservation status and cancel them.
- Staff should easily see every reservation status and cancel them.
- Staff should easily manage the facilities.

Reliability

- After the user login, the system should show an appropriate reservation page that is sorted by the user's school email address. (ex. SBU user => SBU page)
- The timetable of reservation page should contain users' reservations accurately and disable these time slots for the other users so that there won't be any conflicts
- For 'Reservation with Approval', the reservation request form should be sent to the appropriate staff. (ex. SBU user => SBU staff)
- When the staff cancels a student's reservation, the reason should be sent to that student accurately.

- When the staff changes the status of the facility to 'Out of Order', the students should not be able to select that facility.

Performance

- The server should expect a large number of users at the start of the week when the timetable is renewed because they might reserve the facility ahead.

3. Schedule & Risks

3.1 Schedule

Due Date	To Do List
9/24	- Complete System Requirements Specification
10/1	- Complete System Design Specification - Set up the GitHub for the team and make sure everyone can use the program without trouble
10/7	- Set up the Authentication System - Login/Logout - Distinguish Student/Admin - Distinguish Campus
10/21	- Set up the reservation system without approval - Main reservation page with two buttons: Reservation with Approval and Reservation without Approval - Facility List - Timetable - Reservation Pop Up
11/4	- Set up the reservation system with approval - Facility List - Timetable - Reservation Request Form - POST the form
11/11	- Set up 'My Page' - View the reservation status - Canceling the reservation - Set up the manual page
11/18	- Managing the facility list (Admin function) - Add the facility - Delete the facility - Change the status of facility
11/25	- Managing the reservations (Admin function) - Approve/Reject the students' reservations - Cancel the student's reservations
12/2	- Stretch Functions

12/5	-	Final Release

3.2 Risk Analysis

Simultaneous Reservation

When more than two users are trying to reserve the same room at the same time, the system can reserve more than one user in the same time slot. To reduce this risk, we will make sure the database only accepts and receives one user who reserved the room first. For 'Reservations without Approval' rooms, the user who clicks 'Complete' first will use the room. The rest of the users who click 'Complete', a message will appear that the room is already taken by another user. For 'Reservations with Approvals' rooms, the user who clicks the 'Submit' button will only send that user's request to the admin. For the rest of the users who click 'Submit', a message will appear that the room is already taken by another user.

APIs

We might have problems to connect between the frontend and the backend. This risk will result in failure to get and connect APIs, such as a weekly time table, to the project. It is required to plan and search for suitable APIs for the project and gather enough information about how the application runs. This will help us to understand how programs and APIs work. However, designs will be changed to recover from the risk. For instance, the users will select rooms from tag functions, such as selecting a date, start time and end time, instead of making the timetable. Or, the users have to reserve non-approval rooms like approval rooms by submitting a form and wait for the admin to confirm the request.

Admin's Message Box

Every time the users request to reserve a room, the admins have to receive notifications in the message box. There is a risk that the request forms might not be sent to the message box correctly. To prevent this risk, we will run and check if the data from the request form has been saved to the database. If this risk has failed to be recovered, will change the request form to an email form.

3.3 Acquiring Feedback

Designing Stage

As our primary target is the IGC, we will have a survey of current reservation system and our prospective service to make better design from the IGC students and staff. It is more difficult to change the design after the product is created.

Beta Test

After the system is created, we will run a beta test for a few facilities within the IGC. Then, we will gather feedback from users to improve the system.