Software Design Document

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. System Architecture

1.1 Overview

We will use HTML 5, CSS 3, Javascript, MySQL 8.0, PHP 7.3, and Laravel 5 to implement the web application. We are following a model-view-controller design pattern. We chose these programming languages and environments since they are the most familiar choices for us. Also, Laravel provides many pre-built functions and architecture so that we can start building up easily.

1.2 Components and Interfaces

1.2.1 Reservation

This model contains all the information about reservations that users (students) make. This model allows users (students) to reserve the facility. This model includes the type of reservations, the name of a user who made a reservation, timeslot selected for the reservation, the name of the facility that the user selected, the purpose of reserving the room, and the number of students using the room.

Attributes

Name	Type	Description
ID	int	The unique ID is assigned to each reservation that the user (students) makes.
Type	string	There are two types of reservations. ‘Approval’ for facilities that need approval from the admin ‘Open’ for facilities that do not require approval from the admin. The user will choose either one and make a reservation.
User_ID	int	User ID of whom made a reservation
Timeslot_ID	int	ID of the selected time slot for the reservation. The time slot indicates the start date/time and end date/time.
Facility_ID	int	Facility ID is assigned when a reservation is made
Reservation_Status	string	For the ‘Approval’, after a student sends the request form to the admin, a student can check the status. If the room is not approved yet, ‘Pending’ will be displayed. If it is approved, ‘Approved’ will be displayed. If it is canceled, ‘Declined’ will

		be displayed. For the declined case, the admin will give the reason why the reservation is rejected.
Purpose	string	For ‘Approval’, students are required to fill out why they are reserving the room. The purpose will be stored in the database, and the admin will give permission to use the room after reading the purpose.
Number	int	The user needs to write about how many students are going to use the room

1.2.2 MakeReservationController

This controller allows the user (student) to make a reservation by selecting a facility, date, and time.

Functions

Name	SelectRoom
Description	This function takes a Facility_ID and queries the ‘Timeslot’ class to return all objects with passed Facility_ID. Then, it updates the interface to show the timetable with filled in timeslots.
Parameters	Facility_ID: int

Name	SelectTimeslot
Description	This function is triggered when the user selects a time slot from the timetable. It takes Start/End date and time from the timetable and requests the ‘Timeslot’ class to create an object with passed values. Then, it updates the interface to show the reservation request form.
Parameters	Time: time Date: date

Name	AddReservation
Description	This function enables the user (student) to submit the form to the user (admin) when the ‘Submit’ button is clicked and make a reservation. It saves ID, User_ID, Facility_ID, Type, Timeslot_ID, Reservation_Status, Purpose, and Number to the database. Reservation_status is ‘Pending’ in default.
Parameters	ID: int

	Type: string User_ID: int Timeslot_ID: int Facility_ID: int Reservation_Status: string Purpose: string Number: int

1.2.3 AdminCancelReservationController

This controller allows the user (admin) to cancel the user's (student's) reservations by filling out the form. If the form is not filled completely, the alert will be displayed.

Functions

Name	ClickCancelForm
Description	This function is triggered when the user (admin) clicks the time slot that is filled. This function is used to proceed to cancel the reservation by the user (admin). It updates the interface to show the cancel form.
Parameters	No Parameter

Name	CancelReservation
Description	This function enables the user (admin) to cancel the reservation of students' reservations when the 'Submit' button is clicked. It takes in the Reservation_ID of that reservation and queries 'Reservation' class to return the 'Reservation_Status' of the object with passed Reservation_ID. Then, the controller changes the status to 'Declined'.
Parameters	ID: int

Name	Alert
Description	This function is to notify the user that the form is not filled when the user clicks the 'Submit' button.
Parameters	No parameter needed

Name	CancelProcess
Description	This function is used when the user clicks the 'Cancel' button while

	canceling students' reservations. The information written on the form is not saved.
Parameters	No parameter needed

1.2.4 ManageReservationController

This controller allows the user (admin) to approve or decline users' (students') reservations. This also allows the user (student) to cancel its own reservation on My Page.

Functions

Name	ClickMessage
Description	This function enables the user to see the list of students' requests by clicking the 'Message' button. It queries the 'Reservation' class to return all objects with 'Pending' status. Then, it updates the interface to show the list of objects.
Parameters	No parameter needed

Name	ClickApprove
Description	This function approves the students' reservation requests. It queries the 'Reservation' class to return the status of the object with passed ID. Then, it changes the status to 'Approved'.
Parameters	Reservation_ID: int

Name	ClickDecline
Description	This function enables the user to decline students' requests by clicking the 'Decline' button. It queries the 'Reservation' class to return the status of the object with passed ID. Then, it changes the status to 'Declined'.
Parameters	Reservation_ID: int

Name	ClickCancel
Description	This function enables the user (student) to cancel one's own reservation by clicking the 'Cancel' button. The reservation will be deleted from the list on My Page. It queries the 'Reservation' class to return the status of the object with passed ID. Then, it changes the status to 'Declined'.

Parameters	Reservation_ID: int

1.2.5 User

This model contains all the information about the users. Each user's information, such as the name of the user, email, password, role, and penalty status, is included. There are three types of users, which are student, admin, and blacklist. The user (student) is able to reserve or cancel rooms. The other user (admin) is able to manage the reservation system, such as adding or deleting facilities and giving permission for the user (student). The user (blacklist) is not authorized to log in to the web application.

Attributes

Name	Type	Description
ID	int	The unique ID is assigned to each user
Email	String	The email of the user
Name	String	The name of the user
Password	String	The password of the user
Role	String	Indicates the type of the user (Student/Admin/Blacklist)
Penalty_Status	int	Indicates the current penalty status (0/1/2/3)

1.2.6 UserController

This controller allows the users to log in and log out from the web application.

Functions

Name	Login
Description	This function enables the user to log in to the web application when the 'Login' button is clicked.
Parameter	Email: String Password: String

Name	Logout
Description	This function enables the user to log out of the web application when the 'Logout' button is clicked.

Parameter	No parameter needed

Name	Sign Up
Description	This function enables the user to register for the website.
Parameter	Email: String Name: String Password: String Role: String

1.2.7 Report

This model contains all the information about the reports that the user (student) has reported. Each report's information, such as identification of the user, identification of reservation, reason, and picture, is sent to the user (admin). When the user (admin) receives the report, the user (admin) decides whether or not the report should be avoided or sent to the penalty model.

Attributes

Name	Type	Description
ID	int	Each report ID is assigned when the user (student) makes a report.
User_ID	int	User ID to check who reported.
Reservation_ID	int	Reservation ID
Reason	String	Indicates the reason for the user (staff) to read the report
Picture	blob	The picture of the evidence.

1.2.8 ReportController (STUDENT)

This controller allows the user (student) to report to the user (admin) when the room is used inappropriately.

Functions

Name	UploadImage
Description	This function enables the user (student) to upload an image to the field

	in the report form when the ‘Upload’ button is clicked.
Parameter	Picture: blob

Name	AddReport
Description	This function enables the user (student) to submit the form to the user (admin) when the ‘Submit’ button is clicked. The controller queries the ‘Reservation’ class to return the ID of an object with passed ‘Timeslot_ID’ and ‘Facility_ID’. Then, it queries the ‘User’ class to get current user’s User_ID. It saves User_ID, Reservation_ID, Reason, and Picture to the database.
Parameter	ID: int User_ID: int Timeslot_ID: int Facility_ID: int Reason: string Picture: blob

1.2.9 Penalty

This model contains all information about the penalties that the user (admin) has confirmed from the reports made by the users (students). The admin gives the penalty to the users when the users (student) disobeyed the rules, such as overtime usage and make the room in poor condition. Each penalty’s information, such as identification of reservation and reason, is sent to the user (student)’s my page (penalty status from the user’s my page). There are three types of penalties and each type has different penalties. When the user (student) receives the first penalty, the user (student) will only get a warning, but the user (student) will not be able to use the facilities until the user (student) has advice with the user (admin) when the user (student) gets the second penalty. If the user (student) receives the penalty for the third time, the user (student) is changed to a different user (blacklist), meaning the user (student) is not available to use the web application for the rest of the semester.

Attributes

Name	Type	Description
ID	int	Each penalty ID is assigned when the user (staff) gives a penalty to the user (student)
Reservation_ID	int	When the user (student) reports, the admin finds the reservation ID that matches the previous user.

Reason	String	The user (staff) is required to provide reasons to the user (student) for giving a penalty. The reason will be stored in the database.

1.2.10 PenaltyController (ADMIN)

This controller allows the user (admin) to give a penalty to the user (student) who has used the room inappropriately.

Functions

Name	ViewReport
Description	This function enables the user (admin) to see the report sent from the user (student). The controller queries the 'Report' class to return the object with the passed ID.
Parameter	Report_ID: int

Name	ConfirmReport
Description	This function enables the user (admin) to confirm whether or not the report sent by the user (student) is rejected or approved. Takes the report ID and sees the report text and images to confirm whether to give a penalty or not when the 'Reject' or 'Approve' button is clicked. When the 'Approve' button is clicked, the controller queries the 'Reservation' class to return the 'User_ID' of an object with 'Reservation_ID' in the 'Report' object. It queries the 'User' class to return the status of an object with returned 'User_ID'. Then, it changes the status to 'Blacklist'.
Parameter	Report_ID: int

Name	GivePenalty
Description	This function enables the user (admin) to give a penalty to the user (student) who misused the room when the 'Give Penalty' button is clicked. The controller queries the 'User' class to return the status of an object with returned 'User_ID'. Then, changes the penalty status by incrementing by one to the current penalty status.
Parameter	Reservation_ID: int Reason: String

Name	RemoveFromBlacklist
Description	This function enables the user (admin) to change the user (blacklist) who has been blocked from the application to a different user (student). It is to give access to the reservation system when the ‘Unlock’ button is clicked. The controller queries the ‘User’ class to return the status of an object with passed ‘User_ID’. Then, it changes the status to ‘Student’.
Parameter	User_ID: int

Name	CurrentPenaltyStatus1
Description	This function enables the user (student) to get a warning for using the room inappropriately.
Parameter	User_ID: int

Name	CurrentPenaltyStatus2
Description	This function enables the user (admin) to make the user (student) not be able to use the web application by changing the type of user (student) to blacklist until the user (student) has advice with the user (admin). The controller queries the ‘User’ class to return the status of an object with passed‘User_ID’. Then, it changes the status to ‘Blacklist’.
Parameter	User_ID: int

Name	CurrentPenaltyStatus3
Description	This function enables the user (student) to be changed to a different user (blacklist) and not be able to use the reservation system for the rest of the semester. The controller queries the ‘User’ class to return the status of an object with passed‘User_ID’. Then, it changes the status to ‘Blacklist’.
Parameter	User_ID: int

1.2.11 Facility

This model contains all information about facilities that the user (students) can make reservations. Each facility's information, such as name, category, location, the maximum number of capacity, and a picture of the facility, is provided by the user (admin). It can be NULL which means there is no facility.

Attributes

Name	Type	Description
ID	int	The unique ID is assigned to each facility
Name	string	The name of the facility (ex. Study Room, Music Room)
Category	string	The building or space in which the facility belongs (ex. Housing A, Multi-complex)
Type	string	Indicates whether the facility needs approval from the admin or not when students try to make a reservation ('Approval' for a facility that requires approval and 'Open' for a facility that does not require approval)
Location	string	The location of the facility (ex. A101, Near the female gate on the 1st floor of Housing A)
Capacity	int	The maximum number of people that the facility can hold
Picture	blob	The picture of the facility
Status	string	Indicates the status of the facility. Students can make a reservation when it is 'Available' and they cannot make a reservation when it is 'Out of Order'.

1.2.12 FacilityController

This controller allows the user (admin) to add, edit, and delete the facility information.

Functions

Name	AddFacility
Description	This function enables the user (admin) to add the facility. The user has to fill in the category, title, location, and the maximum number of capacity for

	mandatory. The picture is optional. The status of the facility is ‘Available’ in default. The information is saved in DB.
Parameters	Name: string Category: string Type: string Location: string Capacity: int Picture: blob Status: string

Name	EditFacility
Description	This function enables the user (admin) to edit the information of the facility. If there is no change, the original information is saved to the field. If the user wants to edit, the user has to fill in the fields that he/she wants to change among name, category, location, capacity, and picture. The information is saved in DB.
Parameters	Name: string Category: string Type: string Location: string Capacity: int Picture: blob Status: string

Name	EditFacilityStatus
Description	This function is called by clicking a ‘Change Status’ button next to the facility name. The controller queries the ‘Facility’ class to return the ‘Status’ of the facility with an ID that was passed in. If the status is ‘Available’, it is changed into ‘Out of Order’. If the status is ‘Out of Order’, it is changed into ‘Available’.
Parameters	Facility_ID: string

Name	DeleteFacility

Description	This function enables the user (admin) to delete the facility. The controller queries the 'Facility' class to return an object with the passed ID and delete the object.
Parameters	Facility_ID: string

1.2.12 Timeslot

This model indicates the date and the time range of reservation for facilities. It has date, start time of reservation, end time of reservation, and facility ID.

Attributes

Name	Type	Description
ID	int	The unique ID is assigned to each timeslot
Date	date	Indicates the date of a timeslot
Start_time	time	Indicates the starting time of a timeslot
End_time	time	Indicates the ending time of a timeslot
Facility_ID	int	ID of the facility

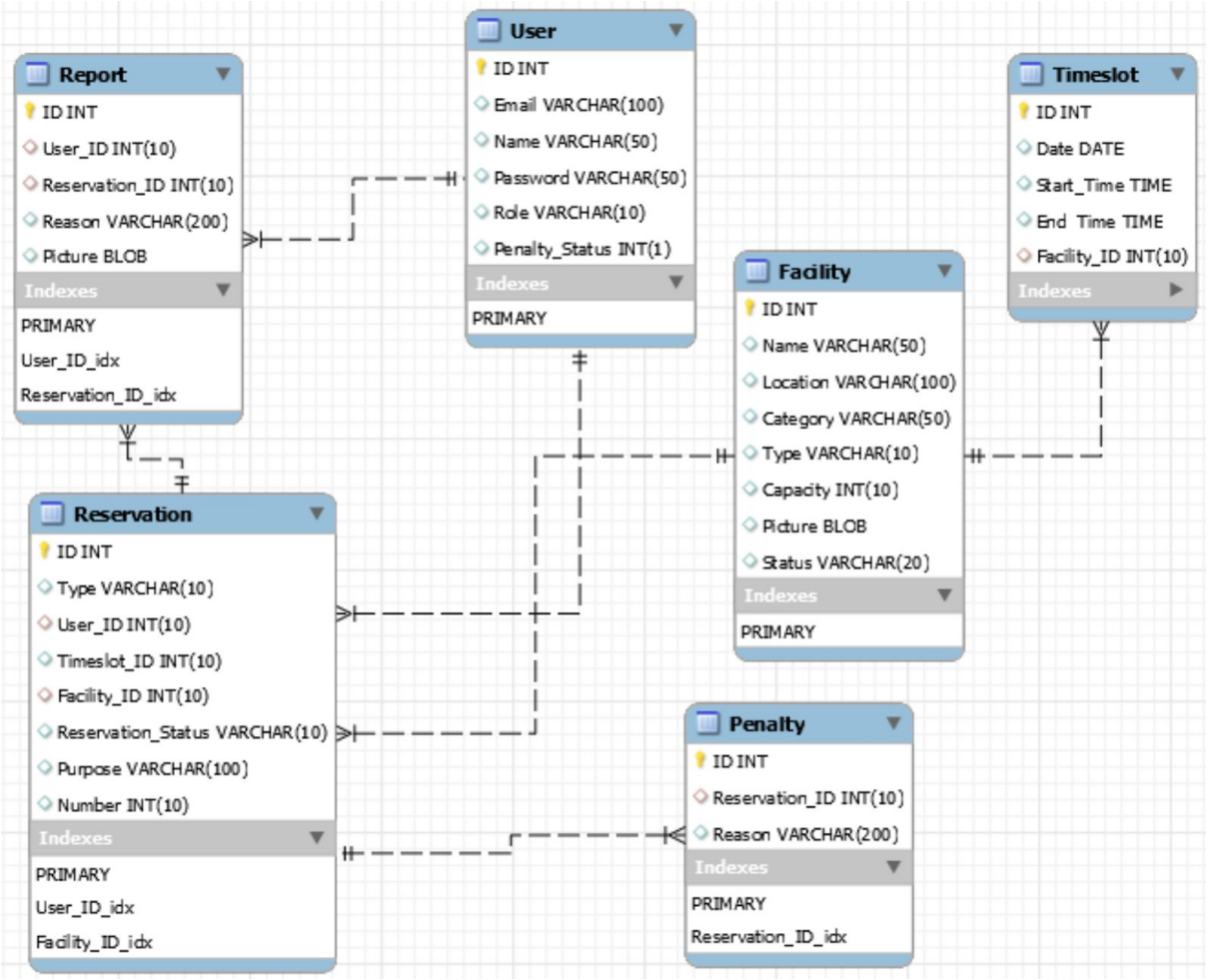
1.3 UML Class Diagram

Please refer to the file called 'UML_Class_Diagram.pdf'

1.4 UML Sequence Diagrams

Please refer to the file called 'Make a Reservation with Approval', 'Approve a Reservation', 'Cancel Reservation by Admin', and 'Room Report by Student'.

1.5 Data Storage



<Foreign Key Indexing>

Report.User_ID = User.ID

Report.Reservation_ID = Reservation.ID

Reservation.Facility_ID = Facility.ID

Reservation.Timeslot_ID = Timeslot.ID

Penalty.Reservation_ID = Reservation.ID

Report.Reservation_ID = Reservation.ID

Report.User_ID = User.ID

Timeslot.Facility_ID = Facility.ID

1.6 Deployment

We will use GitHub and Heroku to deploy our web application.

1.7 Alternatives

1.7.1 Timetable

Current decision: Make our own timetable

Pros	It will be easier to find different components of the timetable, and it will be easier to change the components if necessary. Using a framework without understanding the design pattern will be more difficult to understand the code than making it from scratch. No limit to change the style of the timetable.
Cons	Changing the components for all different timetables will be time-consuming and can give confusion while modifying.

Alternative decision: Use timetable framework/API

1.7.2 Separate 'Approval' and 'Open' page

Current decision: Having separate pages for 'Approval' and 'Open' facilities.

Pros	Easy to distinguish two types of facilities
Cons	It might be unnecessary to have separate pages for 'Approval' and 'Open' facilities.

Alternative decision: Having 'Approval' and 'Open' facilities in one page with different formats and text denoting the differences. Using one page can be easier for users to see all the facilities.

2. Schedule

Gantt Schedule Link:

Version 1 (10/10): https://app.instagantt.com/shared/5d9e2227bf40fa716de03d9f

Version 2 (11/5): https://app.instagantt.com/shared/5dc059131c8d772c7f367936