

**Team:** Ao Liu, Liangjun Song, Vivian Lai

**Title:** Facility Booking Management System

### **Project Summary**

A facility booking management systems that allows students to book and reserve facilities such as group study rooms and fitness courts.

### **Requirements**

Business Requirements	
ID	Requirement
BR-01	Log in email addresses must be colorado.edu

User Requirements	
ID	Requirement
UR-01	As a student, I want to be able to create an account using my school email address.
UR-02	As a student, I want to be able to log in to the system.
UR-03	As a student, I want to be able to view my bookings.
UR-04	As a student, I want to be able to receive email notifications on bookings made.
UR-05	As a student, I want to be able to make a booking.
UR-06	As a student, I want to be able to edit a booking.
UR-07	As a student, I want to be able to delete a booking.
UR-08	As a student, I want to be able to view locations' schedule.
UR-09	As an admin, I want to be able to add new locations that can be booked.
UR-10	As an admin, I want to be able to delete locations.
UR-11	As an admin, I want to be able to delete a student's account.
UR-12	As an admin, I want to be able to override a student's booking.

Functional Requirements	
ID	Requirement
FR-01	Populate available time-slots in tabular form when user has selected a location.

Non-Functional Requirements	
ID	Requirement
NFR-01	Upon providing correct credentials, user shall be directed to landing page in under 3 seconds.
NFR-02	Log in functionality shall behave the same on Windows and OS platforms.

### **Use Cases**

Use Case ID:	UC-01		
Use Case Name:	Create an account		
Description:	Student needs to create an account in order to log in to system.		
Actors:	Student		
Pre-conditions:	Student must have a colorado.edu email address.		
Post-conditions:	Student has a valid account.		
Frequency of Use:	Once a month		
Flow of Events:		Actor Action	System Response
	1.	Enter email	
	2.	Enter password	
	3.	Enter second password field	Display success message
	4.		
	5.		
Variations:	3. System may show failure message if email domain is not correct and/or both password fields do not contain same password.		
Notes and Issues:	-		

<b>Developer Notes:</b>	1. Check if the account has already been used
-------------------------	---

Use Case ID:	UC-02		
Use Case Name:	Log in		
Description:	Student needs to login before using Facility Booking Management System.		
Actors:	Student		
Pre-conditions:	Student must have a valid account.		
Post-conditions:	Student is logged in to system.		
Frequency of Use:	A few times a day		
Flow of Events:		Actor Action	System Response
	1.	Enter email address and password	
	2.	Click on Submit	Display home page
	3.		
	4.		
	5.		
Variations:	2. System may show failure message if email and/or password is/are incorrect.		
Notes and Issues:	-		
Developer Notes:	-		

Use Case ID:	UC-03
Use Case Name:	View booking(s)
Description:	Student can view his/her bookings.
Actors:	Student

<b>Pre-conditions:</b>	Student has logged in to the system.		
<b>Post-conditions:</b>	Student is able to view his/her bookings.		
<b>Frequency of Use:</b>	A few times a week		
<b>Flow of Events:</b>		<b>Actor Action</b>	<b>System Response</b>
	1.	Click on home button	Display booking schedule
	2.		
	3.		
	4.		
	5.		
<b>Variations:</b>	-		
<b>Notes and Issues:</b>	If you are on the homepage, no actor action is needed.		
<b>Developer Notes:</b>	-		

<b>Use Case ID:</b>	UC-04		
<b>Use Case Name:</b>	Send email notification		
<b>Description:</b>	Student can send an email notification after making a booking		
<b>Actors:</b>	Student		
<b>Pre-conditions:</b>	Student has made a booking successfully.		
<b>Post-conditions:</b>	Student is notified of booking via email.		
<b>Frequency of Use:</b>	A few times a week		
<b>Flow of Events:</b>		<b>Actor Action</b>	<b>System Response</b>
	1.	Enter email address(es)	
	2.	Click on Send	Display success message and closes pop-up
	3.		

	4.		
	5.		
<b>Variations:</b>	2. If actor clicks on Cancel, pop-up will disappear and actor is directed back to home page.		
<b>Notes and Issues:</b>	-		
<b>Developer Notes:</b>	-		

Use Case ID:	UC-05		
Use Case Name:	Make a booking		
Description:	Student can make a booking on system.		
Actors:	Student		
Pre-conditions:	Student has logged in to the system.		
Post-conditions:	Booking is made by student.		
Frequency of Use:	A few times a week		
Flow of Events:		Actor Action	System Response
	1.	Click on Make Booking	Display location field
	2.	Select booking location	Display location booking schedules
	3.	Select booking period	
	4.	Click on Save	Display success message and direct user back to home page
	5.		
Variations:	3. The process may end here if there is no ideal period for student		
Notes and Issues:	-		
Developer Notes:	-		

Use Case ID:	UC-06		
Use Case Name:	Edit booking		
Description:	Student can change booking location or booking period.		
Actors:	Student		
Pre-conditions:	Student has logged in to the system and has at least one booking.		
Post-conditions:	Booking is edited and saved in the system.		
Frequency of Use:	A few times a month		
Flow of Events:		Actor Action	System Response
	1.	Click on Edit of respective booking to be edited	Display booking details
	2.	Select booking location	Display location booking schedules
	3.	Select booking period	
	4.	Click on Save	Display success message and direct user back to home page
	5.		
Variations:	3. The process may end here if there is no ideal time for student		
Notes and Issues:	-		
Developer Notes:	-		

Use Case ID:	UC-07
Use Case Name:	Delete a booking
Description:	Student can delete a booking.
Actors:	Student
Pre-conditions:	Student has logged in to the system and has at least one booking.

<b>Post-conditions:</b>	Booking is deleted.		
<b>Frequency of Use:</b>	A few times a month		
<b>Flow of Events:</b>		<b>Actor Action</b>	<b>System Response</b>
	1.	Click on Delete of respective booking to be edited	Display confirmation pop-up
	2.	Click on Ok	Display success message and direct user back to home page
	3.		
	4.		
	5.		
<b>Variations:</b>	2. Actor may click on Cancel and be directed back to home page.		
<b>Notes and Issues:</b>	-		
<b>Developer Notes:</b>	-		

<b>Use Case ID:</b>	UC-08		
<b>Use Case Name:</b>	View schedule for location		
<b>Description:</b>	Students can check out the schedule of location so they can avoid confliction		
<b>Actors:</b>	Student		
<b>Pre-conditions:</b>	Student has logged in to the system and system must have at least one location.		
<b>Post-conditions:</b>	Student is able to view schedule for location.		
<b>Frequency of Use:</b>	A few times a week		
<b>Flow of Events:</b>		<b>Actor Action</b>	<b>System Response</b>
	1.	Click Make Booking	Display location field
	2.	Select booking location	Display location booking schedules
	3.		

	4.		
	5.		
<b>Variations:</b>	-		
<b>Notes and Issues:</b>	-		
<b>Developer Notes:</b>	-		

Use Case ID:	UC-09		
Use Case Name:	Add new location		
Description:	Admin can add new locations for bookings to be made by students.		
Actors:	Admin		
Pre-conditions:	Admin has logged in to the system.		
Post-conditions:	New locations are added into the system.		
Frequency of Use:	Once a week by admin		
Flow of Events:		Actor Action	System Response
	1.	Click on Add New Location	Display form
	2.	Enter Location name	
	3.	Click on Add	Display success message and direct user back to home page
	4.		
	5.		
Variations:	3. System may show failure message if location name contains special characters		
Notes and Issues:	-		
Developer Notes:	-		



<b>Use Case ID:</b>	UC-10		
<b>Use Case Name:</b>	Delete location		
<b>Description:</b>	Admin can delete a location from the system if the location is no longer available for booking.		
<b>Actors:</b>	Admin		
<b>Pre-conditions:</b>	Admin has logged in to the system.		
<b>Post-conditions:</b>	Selected location is deleted from the system.		
<b>Frequency of Use:</b>	Once a month		
<b>Flow of Events:</b>		<b>Actor Action</b>	<b>System Response</b>
	1.	Click on Delete Location	Display list of locations
	2.	Select location to be deleted	Display success message and direct user back to home page
	3.		
	4.		
	5.		
<b>Variations:</b>	-		
<b>Notes and Issues:</b>	-		
<b>Developer Notes:</b>	-		

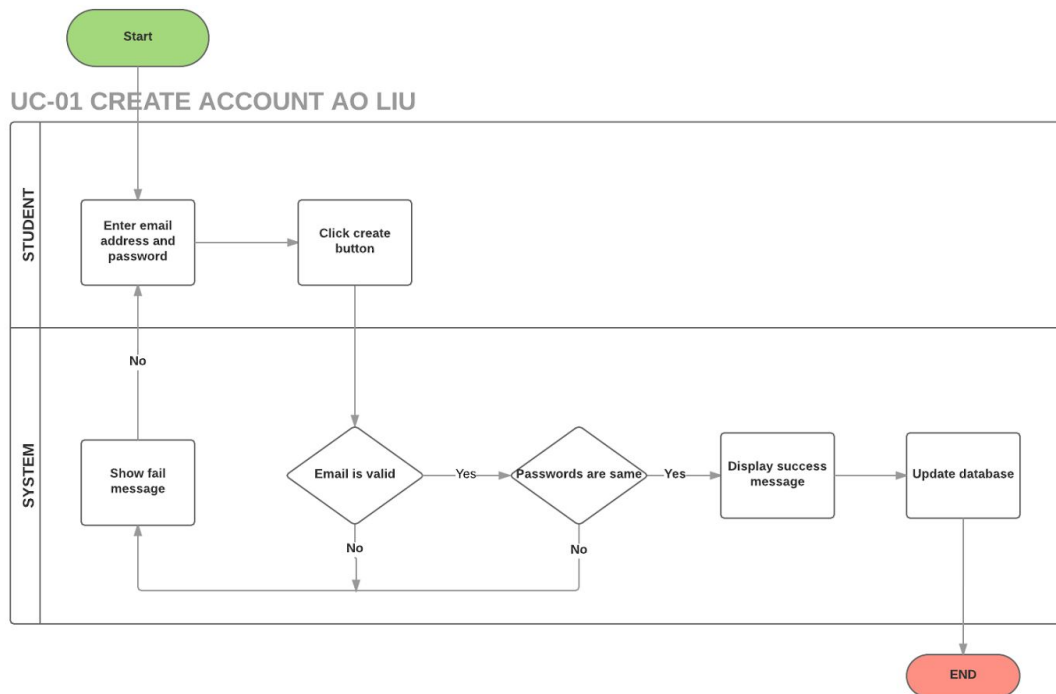
Use Case ID:	UC-11
Use Case Name:	Delete student account
Description:	Admin can delete a student account if the student has graduated from the university.
Actors:	Admin
Pre-conditions:	Admin has logged in to the system. The selected student account must exist in system.

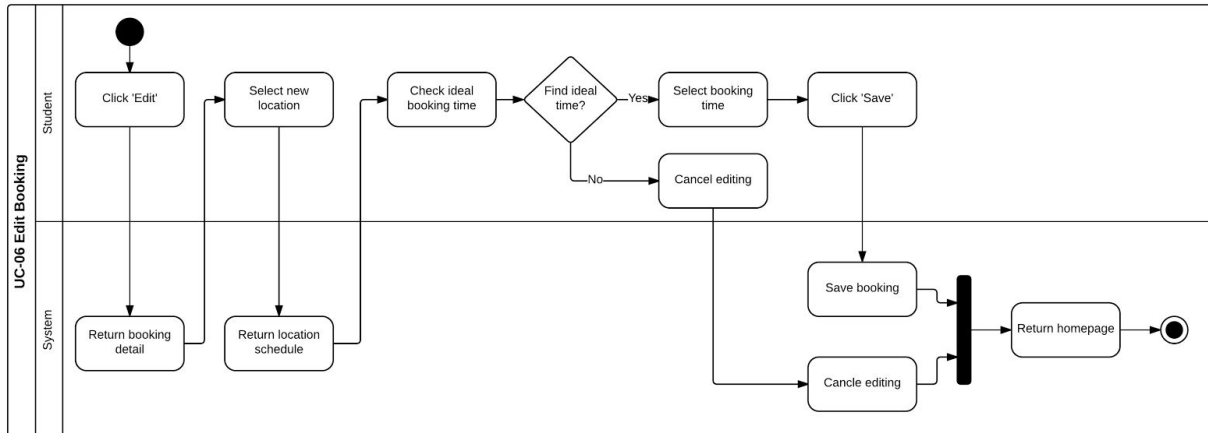
<b>Post-conditions:</b>	The selected student account is deleted from the system.		
<b>Frequency of Use:</b>	Once a month		
<b>Flow of Events:</b>		<b>Actor Action</b>	<b>System Response</b>
	1.	Click on Delete Student Account	Display form
	2.	Input student's email address in search box	
	3.	Click on Search	Display student account
	4.	Click on Delete	Display success message and direct user back to home page
	5.		
<b>Variations:</b>	3. System may not show any student account if there is not any account associated with the given email address.		
<b>Notes and Issues:</b>	-		
<b>Developer Notes:</b>	-		

<b>Use Case ID:</b>	UC-12		
<b>Use Case Name:</b>	Override a booking		
<b>Description:</b>	Admin can override a student’s booking on behalf of the university.		
<b>Actors:</b>	Admin		
<b>Pre-conditions:</b>	Admin has logged in to the system. Booking overridden must be made by student account.		
<b>Post-conditions:</b>	Student’s booking is overridden by admin’s booking.		
<b>Frequency of Use:</b>	Once a month		
<b>Flow of Events:</b>		<b>Actor Action</b>	<b>System Response</b>
	1.	Click on Make Booking	Display location field
	2.	Select booking location	Display location booking schedules

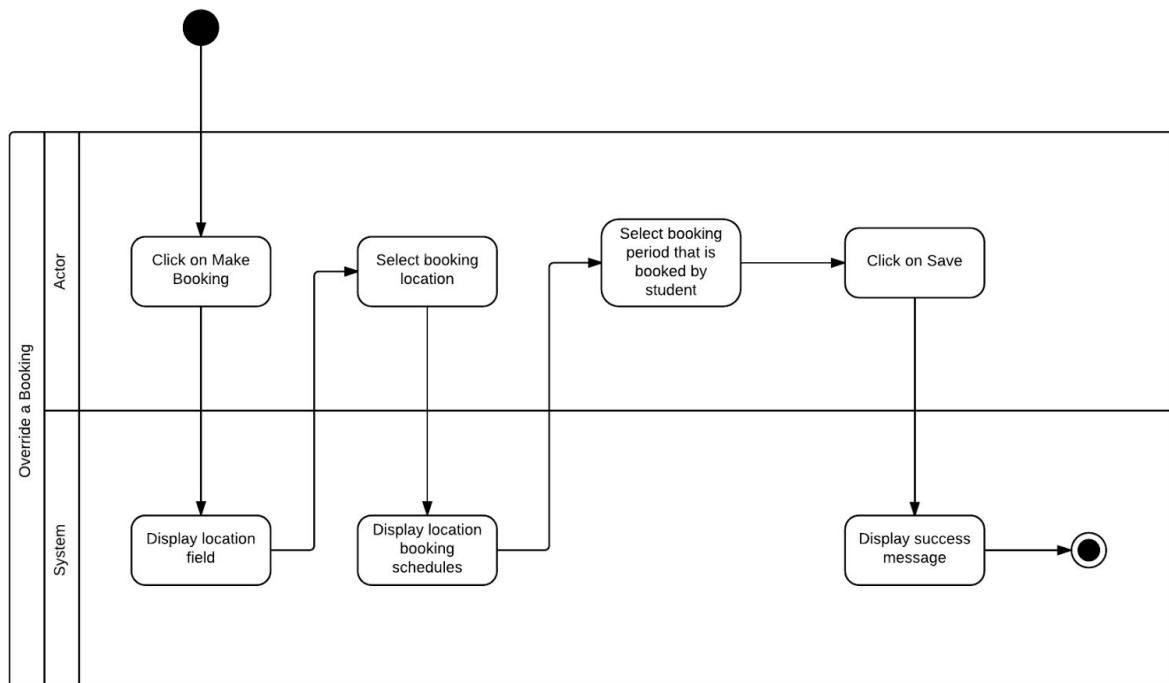
	3.	Select booking period that is booked by student	
	4.	Click on Save	Display success message and direct user back to home page
	5.		
<b>Variations:</b>	-		
<b>Notes and Issues:</b>	-		
<b>Developer Notes:</b>	-		

### Activity Diagrams





## UC-12: Override a Booking | Vivian Lai



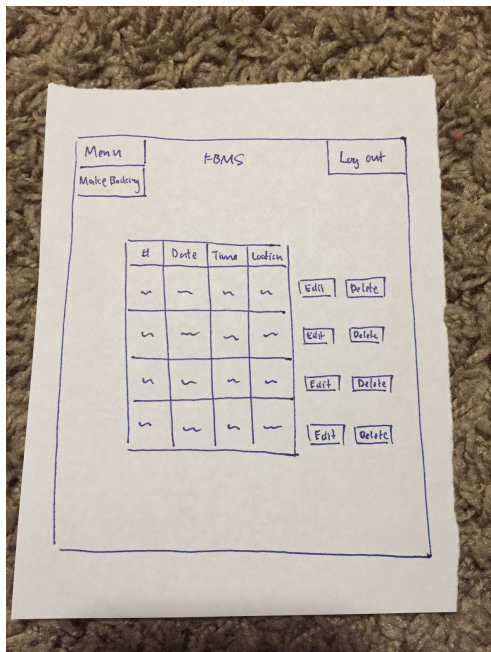
## Data Storage

To handle object persistence, three Data Access Object (DAO) classes were implemented, namely PersonDao, LocationDao and BookingDao. These classes abstract away the database connection and queries (e.g. using SQL) needed to retrieve and persist data from the underlying database.

In this project, relational database (e.g. MySQL) is used, so the various methods in DAO classes (e.g. getBooking and addBooking) when called will execute the select or insert SQL statement to cause data to be transferred to and from the database.

## UI Mockups

Student homepage.



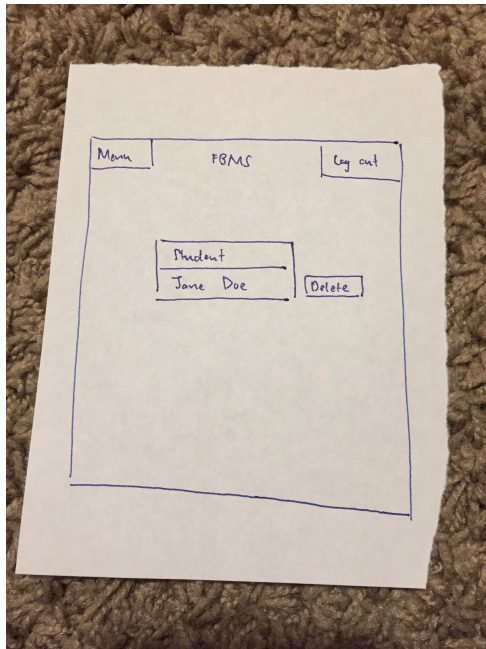
Admin homepage.

The sketch shows a web interface for an Admin homepage. At the top, there is a header bar with three sections: 'Menu' on the left, 'LMS' in the center, and 'Log out' on the right. The 'Menu' section contains three links: 'Add Location', 'Delete Location', and 'Delete Student Act'. Below the menu is a table with four columns: 'Sl', 'Date', 'Time', and 'Location'. The table contains four rows of data, each represented by a tilde (~) in the first three columns. To the right of each row are two buttons: 'Edit' and 'Delete'. The entire interface is enclosed in a rectangular border.

Enter student's email address to be deleted.

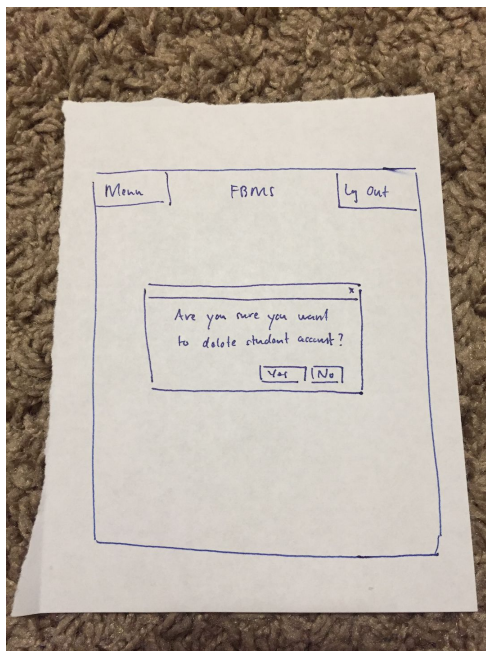
The sketch shows a web interface for entering a student's email address. At the top, there is a header bar with three sections: 'Menu' on the left, 'LMS' in the center, and 'Log out' on the right. Below the header, there is a label 'Email :' followed by a rectangular input field. Below the input field are two buttons: 'Search' and 'Cancel'. The entire interface is enclosed in a rectangular border.

Student account is found. Delete student account if necessary.



Delete confirmation pop-up

The below image depicts an example of a delete confirmation pop-up e.g. Delete student account.



Enter location name to be added.

A hand-drawn sketch of a web form. At the top, there are three buttons: "Menu", "FBMS", and "Log out". Below these, the text "Location:" is followed by a rectangular input field containing a wavy line. At the bottom of the form, there are two buttons: "Add" and "Cancel".

Confirmation message is displayed on homepage.

A hand-drawn sketch of a web form. At the top, there are three buttons: "Menu", "FBMS", and "Log out". Below these, the text "Location is successfully added!" is displayed. Below the message is a table with 4 columns: "Id", "Date", "Time", and "Loc". The table contains 4 rows of data, each with a wavy line in the "Id" column and a checkmark in the "Loc" column. To the right of the table, there are two buttons: "Edit" and "Delete".

Id	Date	Time	Loc
~	~	~	~
~	~	~	~
~	~	~	~
~	~	~	~



Add booking view. Select a location from drop down list.

Menu FBMS Log out

Location:

Add booking view. Location booking schedule is displayed in tabular form when location is selected.

Menu FBMS Log out

Location: Rm 3

	Mon	Tue	Wed	Thurs	Fri
1					
2					
3					

Add Cancel

Edit booking view.

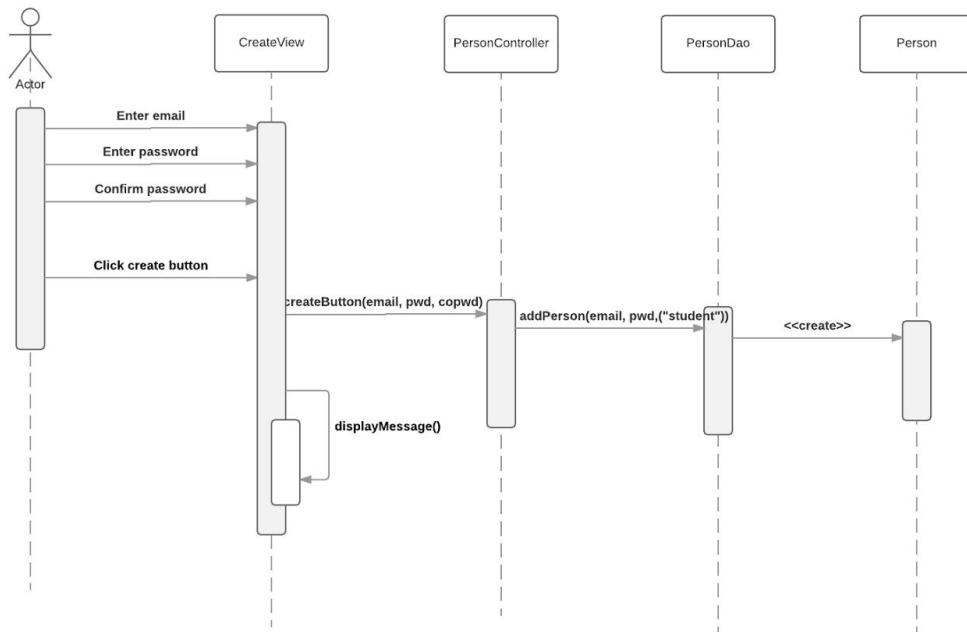
A hand-drawn sketch of a web interface for editing a booking. At the top, there are three buttons: 'Menu', 'FBMS', and 'Log out'. Below them is a 'Location:' label followed by a dropdown menu showing 'Room 7'. The main part of the interface is a calendar grid with columns for 'Mon', 'Tue', 'Wed', 'Thur', and 'Fri'. The first row of the grid has five empty cells. The second row has a shaded cell under 'Tue' and empty cells for 'Wed', 'Thur', and 'Fri'. The third row has a shaded cell under 'Tue' and empty cells for 'Wed', 'Thur', and 'Fri'. The fourth row has a shaded cell under 'Tue' and empty cells for 'Wed', 'Thur', and 'Fri'. The fifth row has a shaded cell under 'Tue' and empty cells for 'Wed', 'Thur', and 'Fri'. At the bottom right of the grid are two buttons: 'Save' and 'Cancel'.

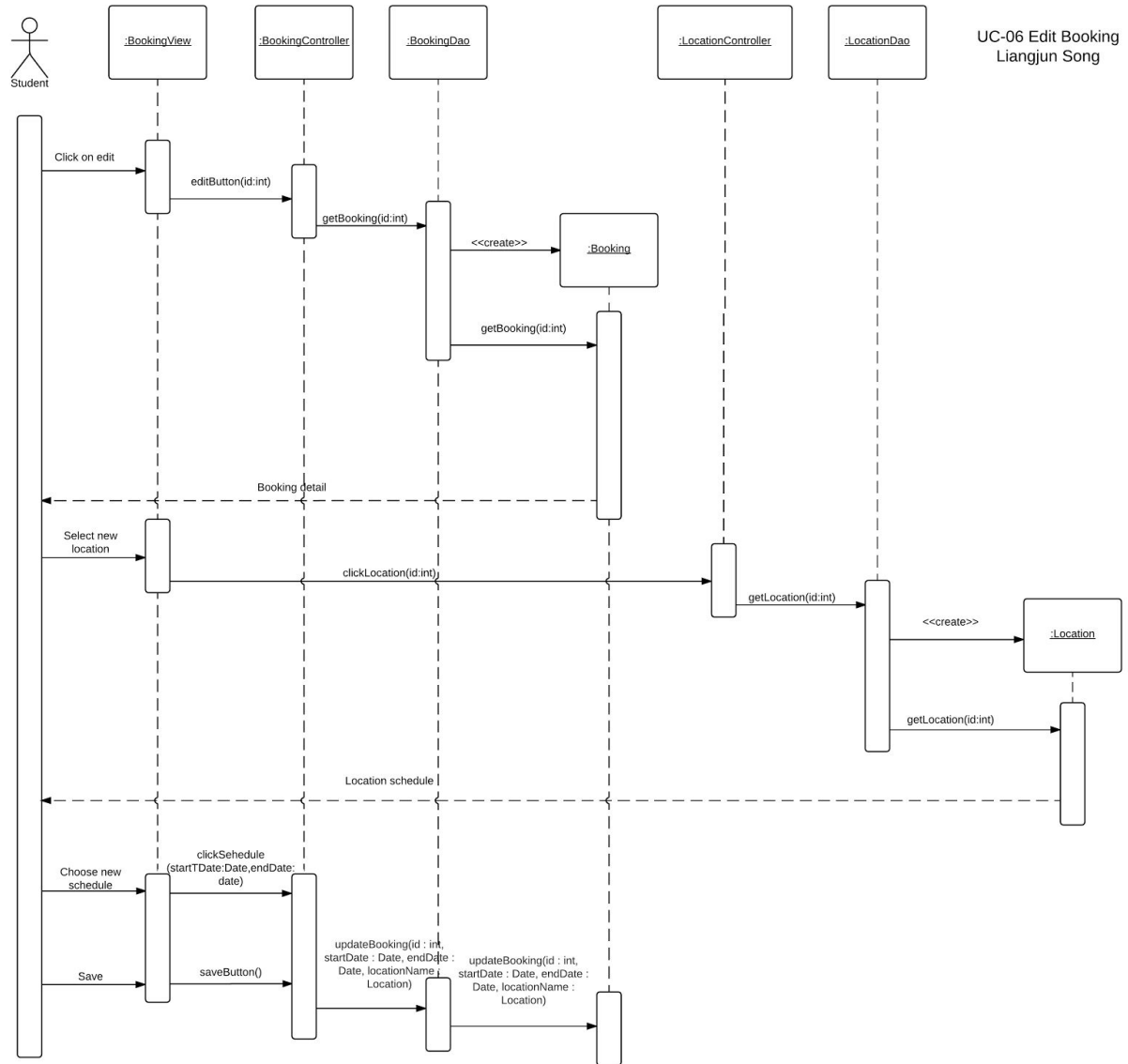
Confirmation message is displayed on pop-up.

A hand-drawn sketch of a confirmation message pop-up. The pop-up has a title bar with a close button (X). The main text inside the pop-up says 'Booking is successfully edit!'. Below this text is a label 'Send email notification:' followed by an input field. At the bottom right of the pop-up are two buttons: 'Send' and 'Cancel'.

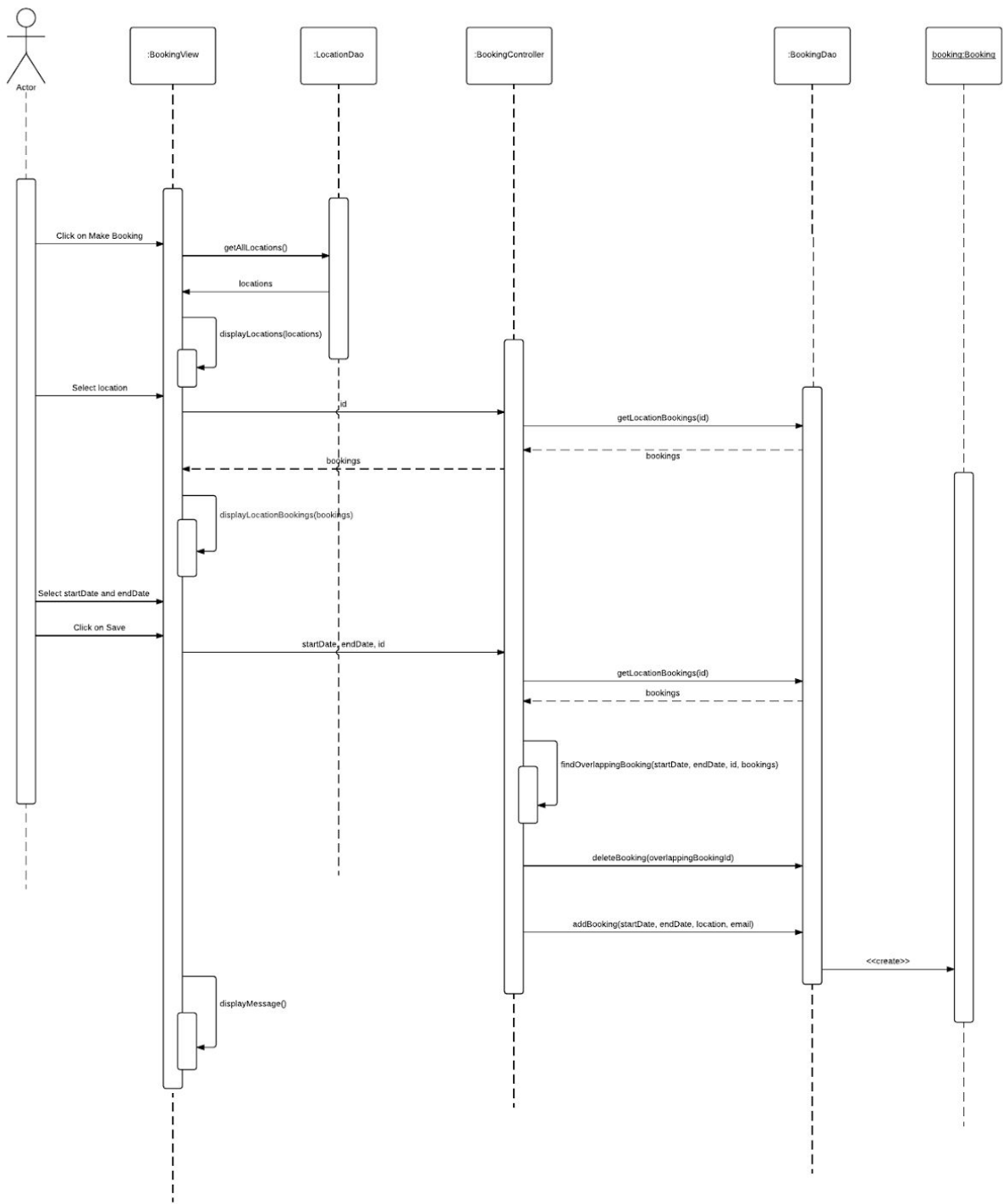
## User Interactions

### UC-1 CREATE ACCOUNT AO LIU





UC-12: Override a Booking | Vivian Lai



# Class Diagram

