

Project 1

Image Segmentation Using K-means Clustering, Gaussian Mixture model and Expectation Maximization

Zunao Hu(40258305)
Department of Electrical and
Computer Engineering
Concordia University
Montreal, Canada

Qian Sun (40232982)
Department of Electrical and
Computer Engineering
Concordia University
Montreal, Canada

Abstract—by creating and simulating a web-based application, this project intends to address the issue of heavily used textbooks at Concordia University's library during exam times. The application offers group reading areas to expose pupils to more desirable texts. User-friendly elements and an intuitive user interface are included in the design. The primary features of the program enable resource sharing, reading room bookings, and study session organization. Other academic institutions dealing with comparable problems might use the provided method as a useful and reproducible model.

Keywords—Software design and Modeling, Object-Oriented Programming, Reservation Application

I. INTRODUCTION

Concordia University has created a textbook-sharing system in response to the rising demand for course textbooks during exam season. By giving students access to textbooks and study areas, the suggested system hopes to achieve a supportive and cooperative learning environment. Students can use the system to reserve reading spaces in the library and exchange textbooks with up to ten other students at once. Students can use the system by entering the course number and necessary textbook, and it will then show any available reading rooms. The number of enrolled students will decide the maximum number of reading rooms for a given course. Students will be able to reserve reading rooms for at least an hour through the first-come, first-served method and no more than five hours. The course instructor will have access to the booking records so they can keep an eye on group dynamics and study habits. The Concordia textbook sharing system's design and execution, including the use-case diagram, user interface, system context analysis, activity diagram, model-view-controller design, functional diagram, and deployment diagram, will be covered in this report.

We will also talk about the system's shortcomings and prospective improvements [1][2].

II. DESIGN AND MODELING

A. Requirements Engineering

First, requirement engineering or software specifications perform in the form of user stories as follows:

- As a student, I want to log in to the system so that I can access the available reading rooms.
- As a student, I want to search for an available reading room based on my course number and textbook so that I can reserve a room for studying that specific textbook.
- As a student, I want to check the availability of the reading room for a specific time slot so that I can plan my study schedule accordingly.
- As a student, I want to reserve a reading room for a minimum of 1 hour and a maximum of 5 hours so that I can study based on my need.
- As a student, I want to be able to create a new reading room, if there is no available one in my desired time slot so that I can plan for my studying comfortably.
- As a student, I want to be able to book a reading room with my friends so that we can study together.
- As the librarian, I want the students to enter their textbook, course number, and their ID to reserve a reading room so that a clear record of each student's activity is recorded.

- As a course instructor, I want to access the booking logs so that I can see which students are studying together.
- As the librarian, I want to have a capacity of a maximum of ten students for each reading room, so that I can provide the necessary facilities and high-quality services.
- As the librarian, I want to ensure that the maximum number of reading rooms available for a course is based on the number of registered students in that course so that I can allocate the number of rooms needed for each course.
- As the librarian, I want to make sure that there is no new reading room being created if one is available at the same time for the same textbook so that I can manage the library space.

B. Use-Case Diagram

The use-case diagram highlights the functionality of the textbook-sharing system from the viewpoint of the user by graphically representing the interactions between actors and the system under review (Figure 1).

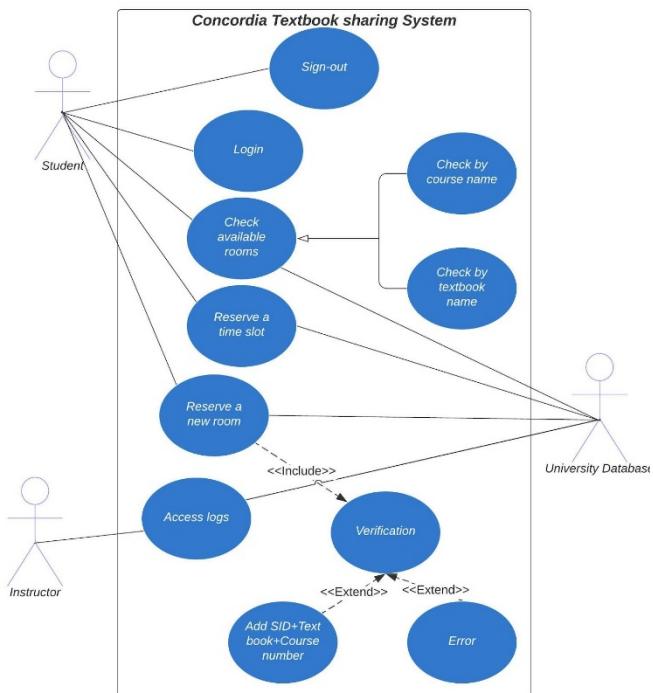


Fig. 1. The Use-Case Diagram

C. System Context Analysis

System context analysis entails determining the external entities that engage with the system, outlining its boundaries and scope, and comprehending how the system interacts with its surroundings (Figure 2).

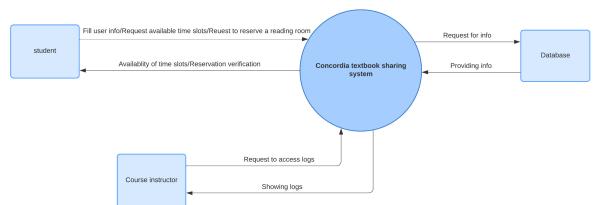


Fig. 2. System Context Diagram

D. Activity Diagram

The activity diagram is a visual representation of the steps or activities that make up a system or process. It illustrates the control flow between these steps and activities and offers a high-level perspective of the behavior of the system (Figure 3).

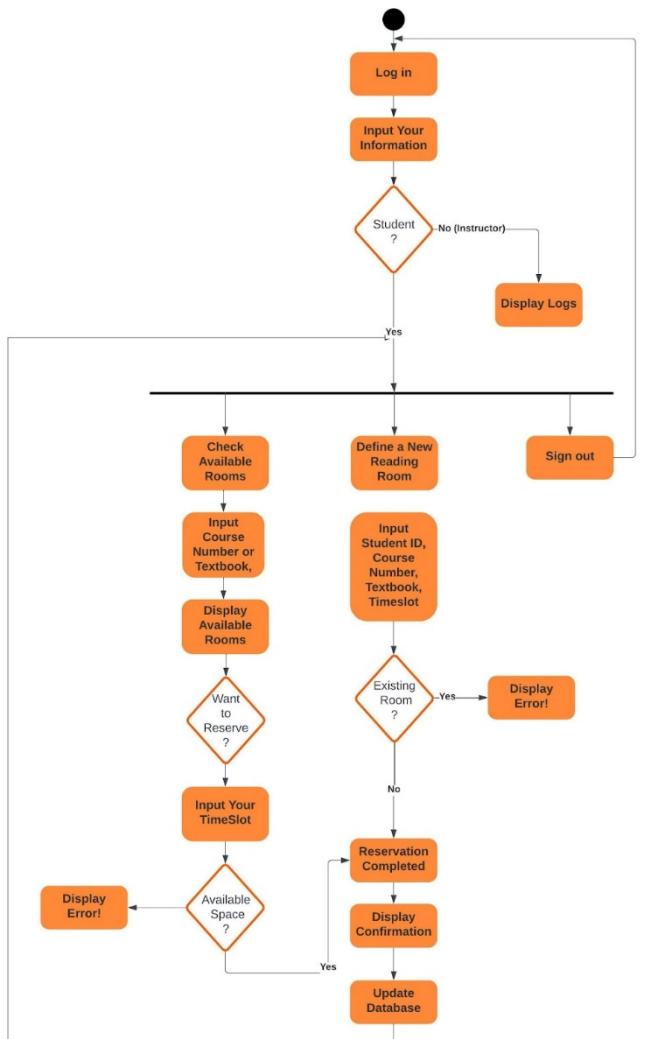


Fig. 3. Activity Diagram

E. Model-View-Controller Design

The software architecture pattern called Model-View-Controller (MVC) design divides our text-sharing application into three interconnected parts (Model, View, and Controller) to promote modularity, maintainability, and scalability (Figure 4).

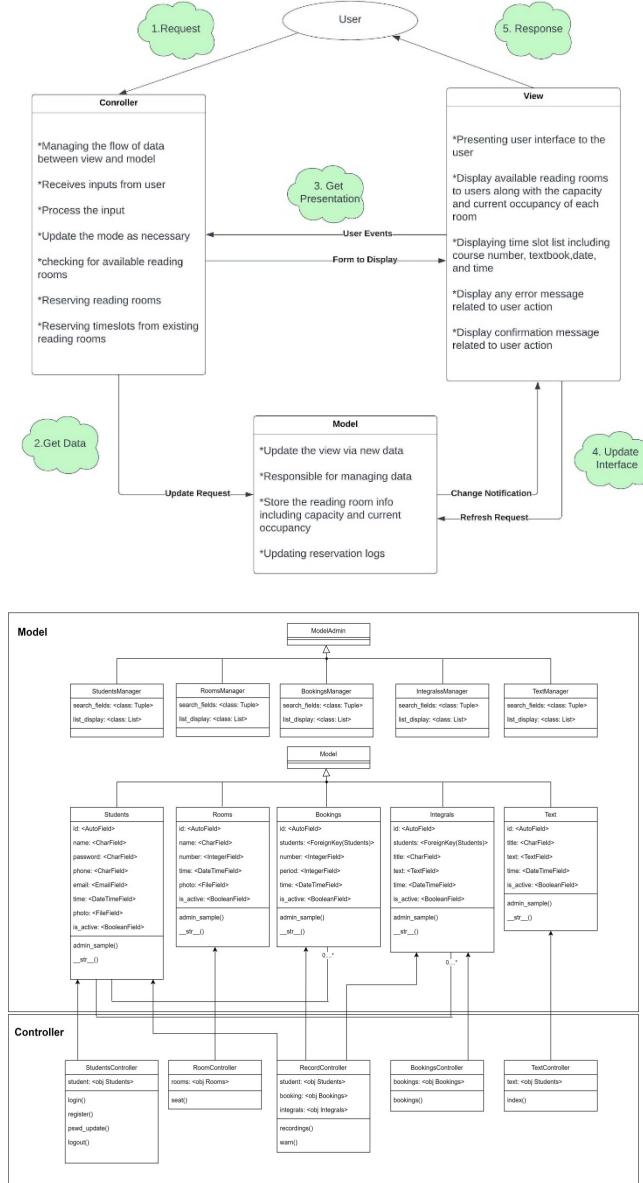


Fig. 4. Model-View-Controller Design and Class Diagram

F. User-Interface Design

The user-interface design diagram for the Concordia textbook-sharing system is created to show the various screens and different interface elements to provide a visual illustration of the system (Figure 5).

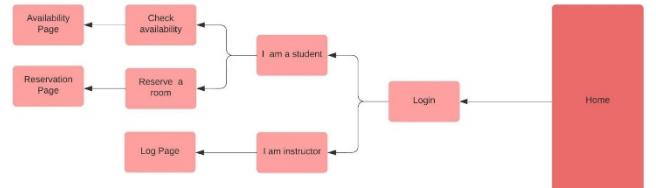


Fig. 5. User-Interface Design

G. Functional Diagram

An illustration of a system's functions and connections that emphasizes the inputs, outputs, and processing needs of the system is called a functional diagram (Figure 6).

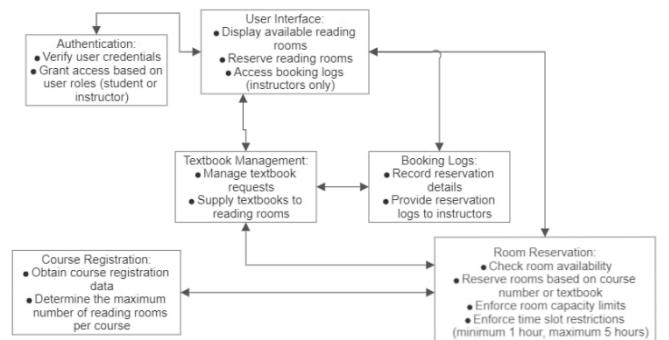


Fig. 6. Functional Diagram

H. Deployment Diagram

A deployment diagram is a form of UML diagram that depicts the actual hardware and software components of a system, demonstrating how they are connected and distributed across various nodes or devices (Figure 7).

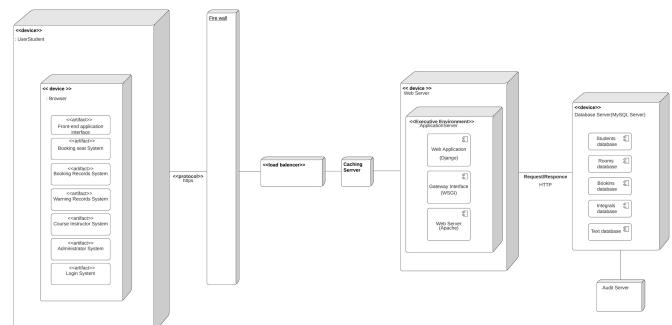


Fig. 7. Deployment Diagram

III. IMPLEMENTATION

A. Description

We have recorded a video demonstrating the functionalities that were implemented during Assignment#2. The demo video can be found in the submitted files under the name "Assignment_Demo.mp4". The following is an introduction to the testing functionalities showcased in the video.

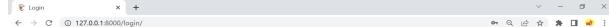
On this website, students can easily choose their own classroom by logging in or registering on the homepage. Once in the classroom, they can select their preferred study time and reserve a seat for up to five hours. Once a seat is reserved, it cannot be reserved in any other classroom during that time. If a student wishes to reserve a seat in another classroom for the same time period, they must first cancel their current reservation.

Our platform, also, allows students to help their friends from other schools to reserve seats in their classrooms. Instructors can log in to their own accounts and view booking logs for their classrooms to monitor their students' progress.

In addition to these functions, other features were added such as an administrator role that allows users to create classroom rooms for students who do not have study rooms, and a warning feature that alerts students on their home pages if they are not following discipline.

B. Implementation Functions and Code Screenshots for Our Page

B.1: As a student, I want to log in to the system so that I can access the available reading rooms (Figure 8).



```

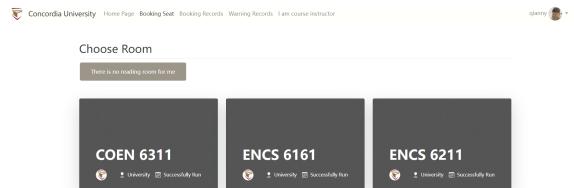
from django.shortcuts import render, HttpResponseRedirect
from login.models import *

def login(request):
    if request.method == 'POST':
        name = request.POST['username']
        password = request.POST['password']
        try:
            stu = Student.objects.filter(username=username)
        except Exception as e:
            print(e)
        if stu:
            try:
                name = stu.objects.get(username=username, password=password)
            except Exception as e:
                print(e)
            name = request.session['name'] = name.name
            stu.photo
        else:
            return HttpResponseRedirect('index')
    else:
        print("not post")

```

Fig. 8. Log-In

B.2: As a student, I want to search for an available reading room based on my course number or textbook so that I can reserve a room for studying (Figure 9).



```

# Create your views here.

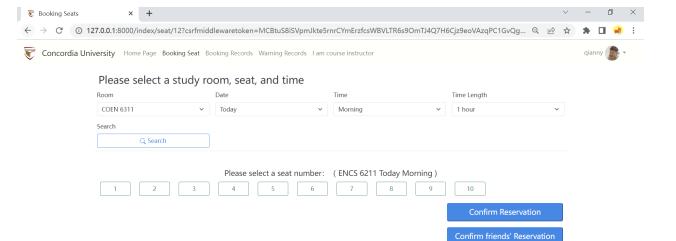
def bookings(request):
    try:
        room = Room.objects.filter(is_active=True)
    except Exception as e:
        print(e)
    return render(request, 'index/bookings.html', {'room': room})

def seat(request):
    return HttpResponseRedirect('index')

```

Fig. 9. Search for Rooms

B.3: As a student, I want to check the availability of the reading room for a specific time slot so that I can plan my study schedule accordingly (Figure 10).



© 2023 Concordia University Administrator

Project: BookReadingRoomManagementSystem

File: views.py

```
from django.shortcuts import render
from .models import Room

def room(request):
    try:
        room = Room.objects.get(id=id)
        rooms = Room.objects.filter(is_active=True)
    except Exception as e:
        print(e)
    if request.method == "POST":
        room_selected = request.GET.get('room_id')
        if room_selected:
            time_selected_r = int(request.GET.get('day')) + 1
            time_selected_y = int(request.GET.get('time')) + 1
            if time_selected_r > 1:
                time = time + 1
            else:
                time = time - 1
    else:
        day = 1
        time = 1
    return render(request, 'room.html', {'rooms': rooms, 'room': room, 'time': time, 'day': day})
```

Project: BookReadingRoomManagementSystem

File: templates/index.html

```
<div><h1>Welcome to BookReadingRoom Management System</h1></div>
```

Project: BookReadingRoomManagementSystem

File: templates/room.html

```
<div><h2>Room Details</h2><div><h3>Room ID:</h3><input type="text" value="<={{ room.id }}>"><br/><h3>Room Name:</h3><input type="text" value="<={{ room.name }}>"><br/><h3>Capacity:</h3><input type="text" value="<={{ room.capacity }}>"><br/><h3>Status:</h3><input type="checkbox" checked=""> Active <input type="checkbox"> Inactive<br/><h3>Bookings:</h3><table border="1"><thead><tr><th>Booking ID</th><th>Room ID</th><th>Booked Date</th><th>Status</th></tr></thead><tbody><tr><td><={{ booking.id }}></td><td><={{ booking.room_id }}></td><td><={{ booking.booked_date }}></td><td><={{ booking.status }}></td></tr></tbody></table></div></div>
```

Fig. 10. Search for Time Slots

B.4: As a student, I want to reserve a reading room for a minimum of 1 hour and a maximum of 5 hours so that I can study comfortably (Figure 11).

Please select a study room, seat, and time

Room	Date	Time	Time Length
CORE 6311	today	Morning	<input type="button" value="1 hour"/> <input type="button" value="2 hours"/> <input type="button" value="3 hours"/> <input type="button" value="4 hours"/> <input type="button" value="5 hours"/>
Search	<input type="text" value="C Search"/>		

Please select a seat number: (ENCS 6211 Today Morning)

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Fig. 11. Duration of Reservation

B.5: As a student, I want to book a seat for my course so that I can study in a specific time (Figure 12).

A screenshot of a web browser window titled "Booking Records". The URL is "127.0.0.1:8000/index/recording/". The page content includes the Concordia University logo and navigation links: Home Page, Booking Seat, Booking Records, Warning Records, I am course instructor, and Log Out. The main content area displays a booking record for a seat: Date: April 5, 2023; Time: Morning; Room: COEN 6311; Seat Number: 7. A note says "Please keep a good learning environment!". A "Cancel booking" button is present. The bottom of the page shows a footer with the copyright notice "© 2023 Concordia University" and a small logo.

```
views.py
1 from django.shortcuts import render
2 from django.http import HttpResponseRedirect
3 from django.core.urlresolvers import reverse
4 from .models import Bookings
5
6 def recording(request):
7     student = request.session.get('name')
8     student = Student.objects.get(name=student)
9     id = request.GET.get('id')
10    if id:
11        try:
12            booking = Bookings.objects.get(id=id)
13        except Exception as e:
14            print(e)
15            booking.is_active = False
16            booking.save()
17
18        return HttpResponseRedirect(reverse('index/recording/'))
19
20    student = Student.objects.get(name=student).id
21    booking = Bookings.objects.filter(is_on_time=True, student_id=student).order_by('-time')
22
23    except Exception as e:
24        print(e)
25
26
main.container.html
1 <!-- Modal -->
2 <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
3     <div class="modal-dialog">
4         <div class="modal-content">
5             <div class="modal-header">
6                 <h4 class="modal-title">Please confirm the information</h4>
7                 <button type="button" class="btn btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
8             </div>
9             <div class="modal-body">
10                 <p class="text-warning">Are you sure you want to cancel?</p>
11                 <input type="text" value="cancel" class="form-control" style="width: 100px;">
12                 <p>Set Number: {{ i.number }}</p>
13                 <input type="button" value="Cancel" class="btn btn-primary" style="width: 100px; margin-top: 10px;">
14             </div>
15         </div>
16     </div>
17
18
19
20
21
22
23
24
25
26
```

Fig. 12. Seat Reservation Confirmation

B.6: As a student, I want to cancel the reserved seat for one of my courses so that I can reserve another room in the same time slot (Figure 13).

Booking Seats

127.0.0.1:8000 回显
You also have a booking. Please cancel your booking before making a new one!

Booking Records

127.0.0.1:8000/index/recording/

Please confirm the information!

Are you sure you want to cancel?

Date: April 5, 2023
Time: Morning
Room: COEN 6311
Seat Number: 7
Please keep a good learning environment!

Cancel Yes

The screenshot displays two instances of the PyCharm IDE. The top window shows the Python file `views.py` with code related to handling user login and session management. The bottom window shows the HTML file `Nonclass.html` with a form for booking a room.

views.py (Top Window)

```
05     # 检查用户名和密码是否正确
06     name = request.session.get('name')
07     name = name[0]
08
09     try:
10         student = Students.objects.get(name=name)
11         book = Bookings.objects.filter(
12             student_id=student.id,
13             time_start__lt=time,
14             time_end__gt=time,
15             period=period,
16             # time_length__lt=time_length,
17             is_active=True
18         )
19
20     except Exception as e:
21         print(e)
22     if book:
23         return render(request, 'Nonclass.html', {'room': room, 'month': month, 'room_id': room_id, 'book': book})
24     else:
25         try:
```

Nonclass.html (Bottom Window)

```
11 <div class="index" style="background-color: #f0f0f0; padding: 10px; border-radius: 5px;">
12     <div style="text-align: center; margin-bottom: 10px;">
13         <script type="text/javascript">
14             (function() {
15                 var myForm = document.getElementById("myForm");
16                 myForm.addEventListener("submit", function(event) {
17                     event.preventDefault();
18                     var room = document.getElementById("room").value;
19                     var month = document.getElementById("month").value;
20                     var room_id = document.getElementById("room_id").value;
21                     var book = document.getElementById("book");
22                     var button = document.getElementById("button");
23                     var label = document.createElement("label");
24                     var input = document.createElement("input");
25                     var option = document.createElement("option");
26
27                     if (book.value === "无") {
28                         book.value = "有";
29                     } else {
30                         book.value = "无";
31                     }
32
33                     if (button.value === "预约") {
34                         button.value = "取消预约";
35                     } else {
36                         button.value = "预约";
37                     }
38
39                     if (label.value === "未选") {
40                         label.value = "已选";
41                     } else {
42                         label.value = "未选";
43                     }
44
45                     if (input.value === "未选") {
46                         input.value = "已选";
47                     } else {
48                         input.value = "未选";
49                     }
50
51                     if (option.value === "未选") {
52                         option.value = "已选";
53                     } else {
54                         option.value = "未选";
55                     }
56
57                     myForm.appendChild(book);
58                     myForm.appendChild(button);
59                     myForm.appendChild(label);
60                     myForm.appendChild(input);
61                     myForm.appendChild(option);
62
63                     myForm.submit();
64                 });
65             })();
66         </script>
67         <form id="myForm" style="border: 1px solid #ccc; padding: 10px; border-radius: 5px; width: fit-content; margin: auto; text-align: center;">
68             <div style="margin-bottom: 10px;">
69                 <label style="font-weight: bold;">房间号:
70                 <input type="text" name="room" value="101" style="width: 100px; height: 30px; border: 1px solid #ccc; border-radius: 5px; padding: 5px;"/>
71             </div>
72             <div style="margin-bottom: 10px;">
73                 <label style="font-weight: bold;">月份:
74                 <input type="text" name="month" value="2023年2月" style="width: 100px; height: 30px; border: 1px solid #ccc; border-radius: 5px; padding: 5px;"/>
75             </div>
76             <div style="margin-bottom: 10px;">
77                 <label style="font-weight: bold;">房间ID:
78                 <input type="text" name="room_id" value="101" style="width: 100px; height: 30px; border: 1px solid #ccc; border-radius: 5px; padding: 5px;"/>
79             </div>
80             <div style="margin-bottom: 10px;">
81                 <label style="font-weight: bold;">是否预约:
82                 <input type="checkbox" name="book" checked="" style="width: 10px; height: 10px;"/>
83             </div>
84             <div style="margin-bottom: 10px;">
85                 <label style="font-weight: bold;">状态:
86                 <input type="checkbox" name="status" checked="" style="width: 10px; height: 10px;"/>
87             </div>
88             <div style="margin-bottom: 10px;">
89                 <label style="font-weight: bold;">是否已选:
90                 <input type="checkbox" name="selected" checked="" style="width: 10px; height: 10px;"/>
91             </div>
92             <div style="margin-bottom: 10px;">
93                 <label style="font-weight: bold;">是否已选:
94                 <input type="checkbox" name="selected" checked="" style="width: 10px; height: 10px;"/>
95             </div>
96             <div style="margin-bottom: 10px;">
97                 <label style="font-weight: bold;">是否已选:
98                 <input type="checkbox" name="selected" checked="" style="width: 10px; height: 10px;"/>
99             </div>
100            <div style="text-align: center;">
101                <input type="button" value="预约" style="width: 100px; height: 30px; border: 1px solid #ccc; border-radius: 5px; background-color: #f0f0f0; font-size: 14px; margin-right: 10px;"/>
102                <input type="button" value="取消预约" style="width: 100px; height: 30px; border: 1px solid #ccc; border-radius: 5px; background-color: #f0f0f0; font-size: 14px;"/>
103            </div>
104        </form>
105    </div>
106
```

Fig. 13. Cancelling a Reservation

B.7: As a student, I want to require the library to create a room for my course so that I can have access to study (Figure 14).

Choose Room

COEN 6311
ENCS 6161
ENCS 6211

There is no reading room for me

Reserve New Rooms

Class Name:

Teacher Name:

Student ID:

SUBMIT

Fig. 14. Request to Create a Room

B.8: As a student, I want to be able to book a reading room with my friends so that we can study together (Figure 15).

Please select a seat number: (ENCS 6211 Today Morning)

1 2 3 4 5 6 7 8 9 10

Confirm Reservation

Confirm friends' Reservation

Fig. 15. Group Study Room with Friends

B.9: As a course instructor, I want to access the booking logs so that I can see which students are studying together (Figure 16).

Welcome to our website!

This is the Concordia Textbook Sharing System. We have developed a unique textbook-sharing model specifically for exam season. Our librarian noticed an increasing demand for course textbooks and proposed a solution to share them with a group of students in the library reading room. Join our community of Concordia students and share textbooks, study together, and succeed together!

Learn More

Faculty 1

Faculty 2

Faculty 3

Class Name	Student ID	Study Time	Time Spent
COEN 6311	Qian Sun 40258305	2023-04-01 09:00:00	2 hours
COEN 6311	Mohsen Sharifzadeh 40185127	2023-04-01 13:00:00	3 hours
COEN 6311	Hadi Ghader Azaad	2023-04-01 14:00:00	1 hour
ENCS 6161	Behrooz Montaz 40157492	2023-04-02 19:00:00	2 hours
ENCS 6161	Shaghayegh Ghaseeri 40232724	2023-04-01 10:00:00	2 hours
ENCS 6161	Pegah Shahab Davari Zadeh 40185262	2023-04-01 15:00:00	2 hours
ENCS 6161	Kherkhan Mohammadi	2023-04-02 08:00:00	5 hours

Fig. 16. Booking Logs

B.10: As an administrator, I want to receive a message from students so that I can create a new room for them without a classroom (Figure 17).

NAME	THE NUMBER OF SEATS	CREATED TIME	ACTIVE STATUS	STUDENT NAME
ENCS 6211	10	April 5, 2023, 1:23 p.m.	Green	
ENCS 6161	75	April 4, 2023, 1:11 p.m.	Green	
COEN 6311	60	April 4, 2023, 1:10 p.m.	Green	

Name:

The number of seats:

Image:

Create Status

Save and add another

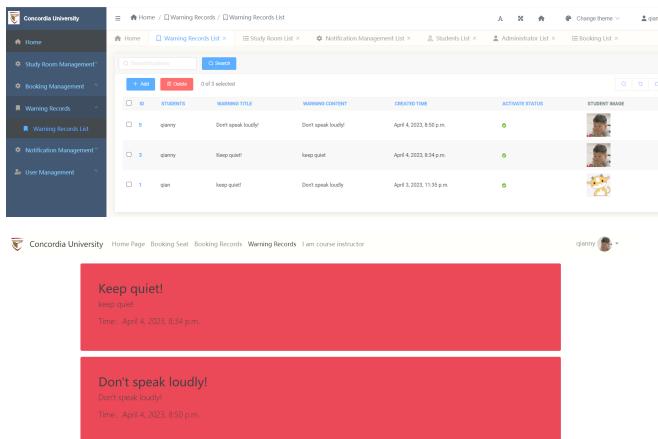
Save and continue editing

Save

```
File Edit View Manager Code Refactor Run Tools Window Help Library login admipy
Project + - × indoView.py admin.py loginView.py modispy NonclassHTML Extractor.html
D:\PycharmProjects\LibraryManagementSystem\src\views
11 admin.site.index_title = "Concordia University"
12
13 class StudentUser(AbstractUser):
14     # 管理员后台增加的字段
15     # 增加了头像、性别、生日、年级
16     # 通过下方的list_display显示出来
17     list_display = ['id', 'name', 'password', 'phone', 'email', 'time', 'is_active', 'admin_sample']
18
19 class BookInfoManager(admin.ModelAdmin):
20     # 指定要显示的字段
21     search_fields = ('student', 'number')
22     # 指定要显示的搜索框
23     list_display = ['id', 'student', 'number', 'name', 'period', 'time', 'is_active', 'admin_sample']
24
25 class RoomManager(admin.ModelAdmin):
26     # 指定要显示的字段
27     search_fields = ('name',)
28     # 指定要显示的搜索框
29     list_display = ['id', 'name']
30
31 list_display = ['id', 'name', 'number', 'time', 'is_active', 'admin_sample']
```

Fig. 17. Admin's Notification

B.11: As an administrator, I want to warn impolite students so that they can keep a quiet environment (Figure 18).



IV. CONCLUSION

By enabling students to reserve reading rooms and share textbooks with their classmates in a collaborative and supportive academic setting, the proposed Concordia textbook-sharing system offers a solution to the rising demand for course textbooks during exam season. While the course instructor can keep an eye on student study patterns and group dynamics, the system's user-friendly interface allows students to look for open reading rooms based on the course number or textbook and make reservations for a specific time slot. The method has the potential to improve student learning and collaboration while addressing issues with textbook availability and affordability with proper implementation.

REFERENCES

For this assignment, the course material is our reference for the descriptions [1]. Also, all the diagrams were drawn using the Lucid Chart tools [3].

- [1] "Course: COEN 6311 M 2224."
<https://moodle.concordia.ca/moodle/course/view.php?id=1533>
 - [2] "Sommerville I. Software Engineering, 10/E. Pearson Education India; 2011"
 - [3] "<https://www.lucidchart.com/pages/>"

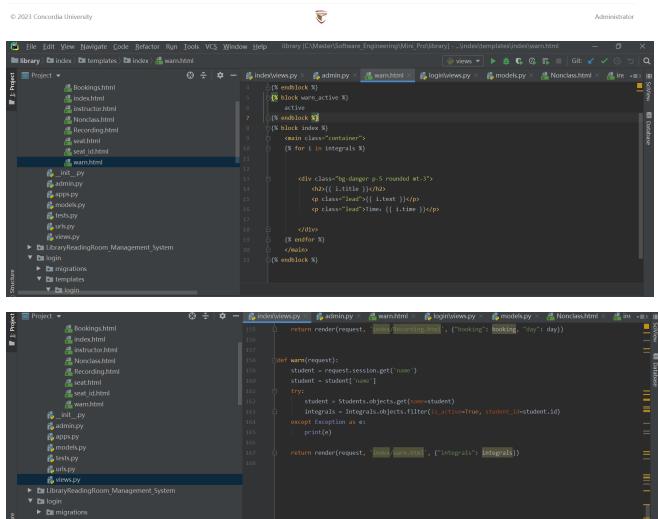


Fig. 18. Admin's Warning