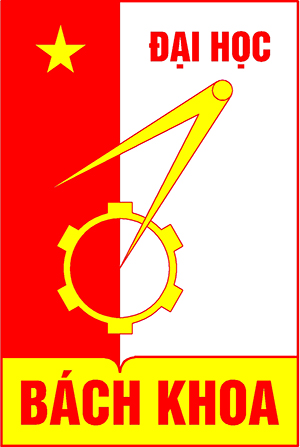
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Trường Công nghệ thông tin và truyền thông**

**----- o0o -----**



**BÁO CÁO THỰC HÀNH**

**ĐỀ TÀI: QUẢN LÝ THƯ VIỆN**

**Môn Học:** Thực hành Cơ sở Dữ Liệu

**Mã học phần:** IT3290

**Mã lớp:** 126877

**Sinh viên thực hiện:**

|  |  |  |
| --- | --- | --- |
| Đỗ Thanh Sơn | MSSV: | 20225665 |
| Đỗ Văn Vũ | MSSV: | 20225956 |
| Chu Đình Hà | MSSV: | 20225712 |

**Giảng viên hướng dẫn: Nguyễn Thị Oanh**

# **LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin được bày tỏ lòng biết ơn chân thành và sâu sắc nhất tới cô giáo NGUYỄN THỊ OANH – người luôn tận tình, nhiệt huyết giúp đỡ, hướng dẫn chúng em trong thời gian học môn Thực hành Cơ sở Dữ Liệu, đã truyền đạt cho chúng em những kiến thức vô cùng hữu ích, giúp chúng em có đủ kiến thức về cơ sở dữ liệu để hoàn thành project này.

# **LÝ DO CHỌN ĐỀ TÀI**

Tại đại học Bách Khoa Hà Nội, ngôi trường đào tạo hàng chục nghìn sinh viên mỗi năm học, nhu cầu sử dụng thư viện Tạ Quang Bửu mỗi ngày của sinh viên là vô cùng lớn. Đó là lúc hình thức mượn, trả sách truyền thống trực tiếp tại thư viện không còn hoạt động một cách tối ưu nữa. Hơn nữa, nếu chỉ sử dụng nhân lực, việc kiểm soát số lượng và định danh các sinh viên ra vào thư viện cũng rất khó khăn.

Nắm bắt được vấn đề trên, nhóm chúng em đã xây dựng một phần mềm quản lý thư viện Tạ Quang Bửu dành riêng cho sinh viên trong trường. Qua đó, sinh viên có thể xem được những thông tin cá nhân trên ứng dụng, theo dõi những tựa sách mới được nhập về hoặc loại bỏ trên mục tin tức, đặt lịch mượn sách trước qua ứng dụng, xem được thời hạn trả sách. Những sinh viên ra vào cũng sẽ được giám sát bởi camera nhận diện khuôn mặt. Người thủ thư cũng sẽ dễ dàng quản lý các đầu sách được nhập về/loại bỏ, và xác nhận các yêu cầu mượn/trả sách của sinh viên qua các tính năng trên ứng dụng.

# **CÁC TÀI NGUYÊN SỬ DỤNG**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Tài nguyên** | **Đặc điểm** |
| FRONT END  &  BACK END | Ngôn ngữ | Python | Tương thích với các trình duyệt, trang web hiện có, chuẩn hóa các kỹ thuật không chính thức. |
| Framework | PyQt | PyQt là một bộ công cụ GUI (Graphical User Interface - Giao diện người dùng đồ họa) cho Python. Nó là một bộ ràng buộc Python cho Qt, một bộ công cụ C++ mạnh mẽ và phổ biến để phát triển ứng dụng đa nền tảng với giao diện người dùng đồ họa. PyQt cho phép viết các ứng dụng GUI trong Python bằng cách sử dụng các tính năng và tiện ích của Qt. |
| Model Face recogintion | Framework | OpenCV | **OpenCV** (cv2): Thư viện này được sử dụng để xử lý hình ảnh và video. Trong mã của bạn, OpenCV được sử dụng để chụp ảnh từ camera, phát hiện khuôn mặt, và xử lý hình ảnh trước khi đưa vào mô hình học máy. |
| Tensorflow và keras | Xây dựng mô hình học sâu (deep learning) với các lớp Neural Network khác nhau để xây dựng mô hình CNN |
| Numpy | Thư viện này được sử lý các mảng số học, tiền xử lý dữ liệu ( preprocessing data) trước khi đưa vào mô hình học máy |

1. **MÔ TẢ BÀI TOÁN**

* Ứng dụng được thiết kế dành cho sinh viên trong trường và thủ thư làm việc tại thư viện.
* Ứng dụng cho phép người dùng xem thông tin về thư viện, nội quy, liên hệ với phòng tư vấn.
* Đối với sinh viên:

o Đăng nhập vào ứng dụng bằng tài khoản do nhà trường cung cấp, mật khẩu có thể được thay đổi.

o Email do nhà trường cấp sẽ có đuôi @sis.hust.edu.vn.

o Xem được thông tin cá nhân trong ứng dụng.

* Email, họ tên, số điện thoại, ngày sinh, giới tính.
* MSSV: gồm 8 kí tự sinh ngẫu nhiên.

o Ra vào thư viện được giám sát bởi camera nhận diện khuôn mặt

* Sinh viên ra vào thư viện được kiểm soát qua camera
* Sinh viên cũng có thể xem được lịch sử check-in/check-out tại các phòng tự đọc & học cũng như tổng thời gian ở trong thư viện của bản thân.

o Xem được những tựa sách mới được nhập về hoặc loại bỏ trên mục tin tức.

* Biết được cụ thể tên các bộ sưu tập sách, báo, đầu sách, thể loại, tác giả, xuất xứ, tên nhà xuất bản, số lượng và ngày nhập về/loại bỏ.
* Thông tin sẽ được đăng lên 1 tuần trước ngày nhập về/loại bỏ.
* Biết được những sự kiện đã đang và sẽ được tổ chức ở thư viện trên mục tin tức.

o Mượn sách và đặt mượn trước qua ứng dụng:

* Sinh viên có thể dùng ứng dụng để đăng ký mượn sách online. Sinh viên phải nhập cụ thể tên đầu sách và số lượng, tối đa là 5 đầu sách 1 lần mượn, không giới hạn số lượng. Một đơn mượn sách sẽ được tạo, ghi rõ thời gian làm đơn online, tên sách, số lượng, và thông tin cá nhân của người mượn gồm MSSV, họ và tên và đưa lên hệ thống.
* Ngoài ra sinh viên có thể mượn sách trực tiếp tại quầy.

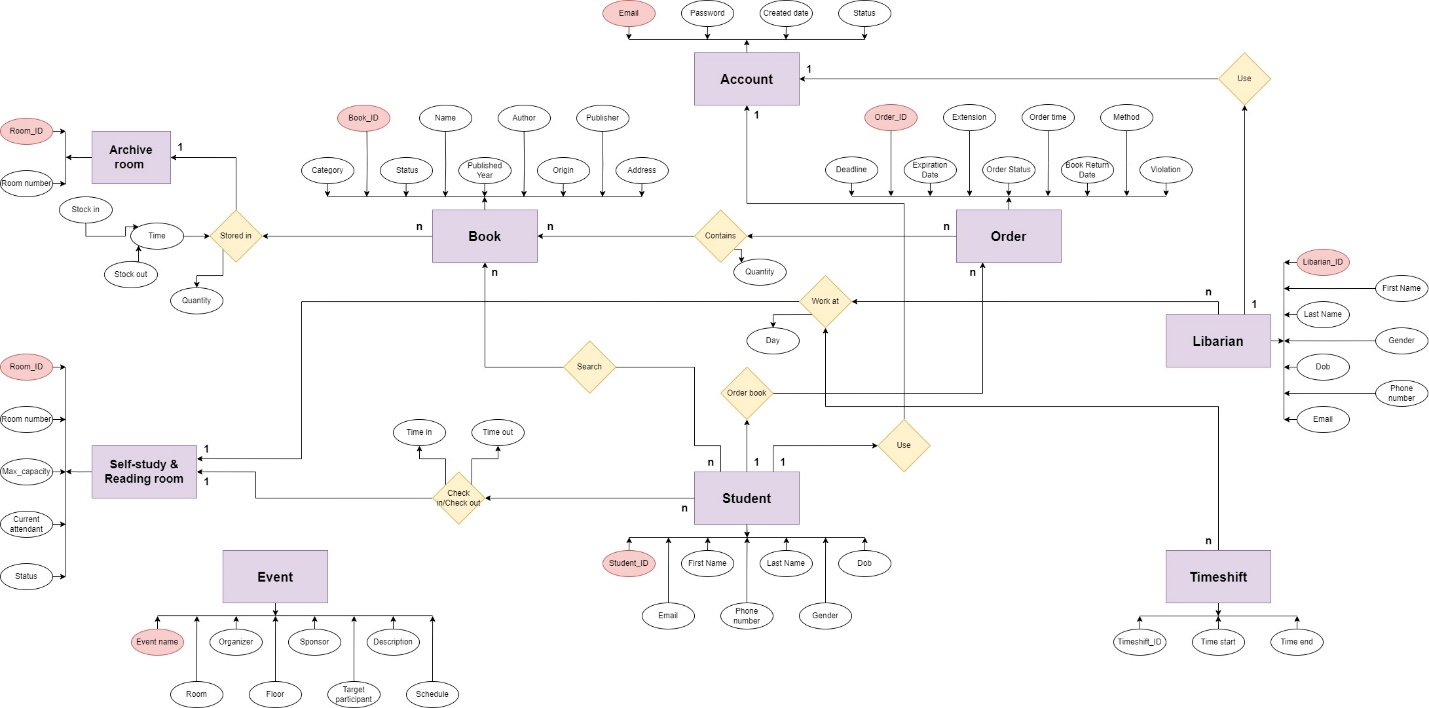
o Thư viện cũng sẽ tổ chức các các sự kiện như triển lãm tranh sách, giới thiệu sách, các buổi thảo luận với tác giả …

* Đối với thủ thư:
* Thủ thư thực hiện nhận đơn trực tiếp và đưa sách cho sinh viên tại quầy.
* Đối với những sinh viên vi phạm nội quy thư viện quá 3 lần, thủ thư có quyền cấm mượn trong khoảng thời gian hoặc xóa sinh viên đó khỏi csdl của thư viện, đồng nghĩa sinh viên đó không thể mượn tài liệu tại thư viện nữa.
* Thủ thư sẽ xem được thời gian biểu(thời gian, địa điểm) của mình: ai làm ở ca nào thì xử lý đơn tại quầy tại thời điểm đó.
* Sách:
* Sách trong thư viện sẽ được đánh mã, mỗi sách sẽ có 1 mã duy nhất.
* Quản lí các thông tin của sách: mã sách, tên sách, tên tác giả, xuất xứ, năm xuất bản, nhà xuất bản, thể loại.
* Sách trong phòng lưu trữ được ghi rõ số lượng, ngày nhập kho, ngày loại khỏi kho.
* Đối với những sách được thanh lí, do độc giả làm mất, hoặc mới được nhập về, thủ thư có thể tìm kiếm, thêm, sửa, xóa thông tin của những cuốn sách đó.
* Đơn mượn sách:
  + Các đơn mượn sách đều sẽ được quản lí bởi một mã duy nhất, tất cả đơn mượn sách đều phải có ngày mượn, trả rõ ràng.
  + Trạng thái của đơn sau khi sinh viên đặt đơn trên ứng dụng lúc này sẽ là “đang chờ”. Sinh viên có thể xem được các đơn mượn sách của mình và có thể chỉnh sửa đơn cũng như hủy đơn đang chờ trong 8 giờ kể từ khi điền đơn. 1 đơn có thể mượn được nhiều sách.
  + Thời gian nhận sách sau khi làm đơn online chậm nhất sẽ là 3 ngày, nếu quá 3 ngày này thì đơn sẽ bị hủy. Sinh viên có thể nhận sách khi trạng thái hiển thị “đã nhận” trên ứng dụng.Sau khi đến nhận sách, nhân viên sẽ cập nhật ngày nhận sách vào đơn và trạng thái lúc này sẽ là “thành công”, hạn mượn sách là 7 ngày tính từ thời điểm nhận sách.
  + Hệ thống sẽ thông báo cho sinh viên khi sắp đến hạn trả sách trước 2 ngày. Nếu đến hạn mà vẫn chưa trả sách thì sẽ bị phạt theo quy định của thư viện.
  + Sinh viên có thể sử dụng ứng dụng để gia hạn online bằng cách vào mục các đơn đã hoàn thành và chọn gia hạn, số ngày gia hạn tối đa là 7 ngày.
* Trong trường hợp sinh viên đến làm thủ tục mượn sách tại quầy:

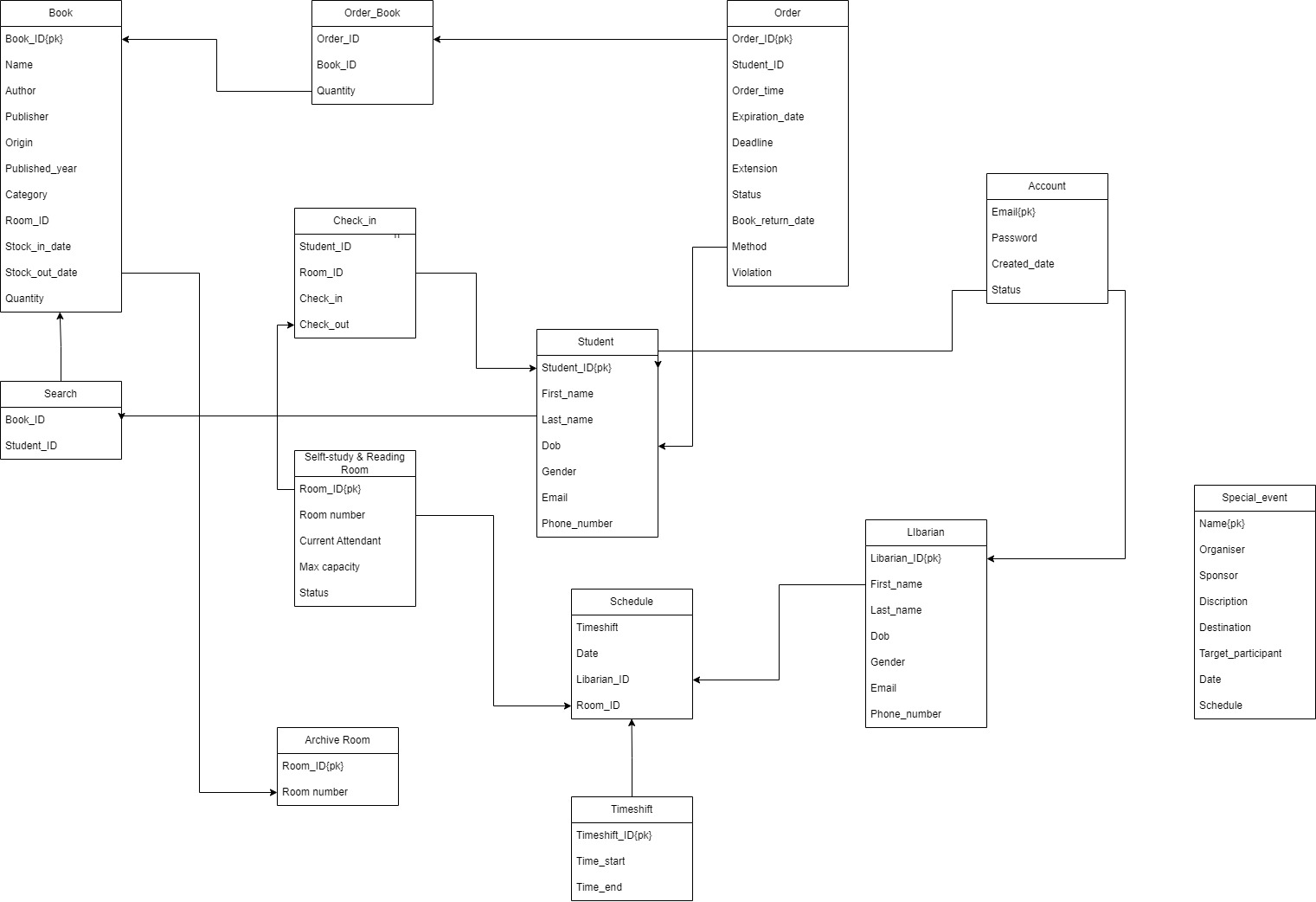
Nhân viên tại quầy mượn sẽ đưa cho sinh viên 1 đơn mượn sách, sinh viên sẽ phải điền họ tên, MSSV, tên sách và số lượng muốn mượn. Sau đó nhân viên sẽ nhập những thông tin đó vào đơn trên hệ thống và một đơn mượn sách sẽ được tạo trong ứng dụng y hệt với đơn mượn sách online, chỉ khác trạng thái lúc này sẽ là “thành công” vì sinh viên sẽ nhận sách trực tiếp ngay sau đó. Khi không có sẵn sách, trạng thái cho cuốn sách đó là “sách hiện đang được mượn”

* Đối với sách mà sinh viên muốn mượn nhưng đang cho mượn thì sinh viên có thể đặt trước để mượn đầu sách đó ngay khi đầu sách đó được trả về, yêu cầu cần phải đề rõ ngày nhận và ngày trả
* Phòng:
  + Phòng tự học và đọc: các sinh viên có thể ra vào các phòng. Các lượt ra vào sẽ được camera dùng công nghệ nhận diện khuôn mặt khi lại và đăng lên hệ thống thời gian chính xác sinh viên ra/vào. Dựa vào số lượng sinh viên hiện tại ở trong phòng/ sức chứa tối đa của phòng mà hệ thống sẽ đưa ra nhận xét. Ngoài tổng thời gian ở trong thư viện, sinh viên cũng có thể xem được thống kê số lần mình vào các phòng tự đọc/tự học.
  + Phòng lưu trữ: Là nơi lưu trữ sách, hệ thống sẽ ghi lại số lượng trong kho/ có thể được mượn/ đang được mượn/ đang nằm trong đơn(chưa nhận sách). Khi xử lý đơn, sách sẽ được lấy ra từ các phòng này.

1. **Mô hình thực thể liên kết và mô hình quan hệ**
2. **Mô hình thực thể liên kết (ERD)**

****

1. **Mô hình quan hệ bảng**

****

# **CÂU LỆNH SQL VÀ TRUY VẤN CỦA TỪNG THÀNH VIÊN**

**\*Các câu lệnh truy vấn cho database cũ**

1. **Chu Đình Hà - 20225712**

**1. Hiển thị số lượt vào của các phòng từ ngày 23/05/2024 - 30/05/2024**

SELECT COUNT(student\_id), room\_id

FROM check\_in

GROUP BY room\_id, check\_in

HAVING DATE(check\_in) BETWEEN '05-23-2024' AND '05-30-2024';

**2. Hiển thị danh sách các sách có số lần được mượn ít nhất trong tháng 4/2024(tối đa 10)**

SELECT b.\*, COUNT(book\_id), order\_time

FROM orders o JOIN book b USING(book\_id)

JOIN orderline ol USING(order\_id)

GROUP BY b.book\_id, ol.order\_time

HAVING EXTRACT(MONTH FROM(order\_time)) = 4

ORDER BY count(o.book\_id) ASC

LIMIT 10;

**3. Tạo view hiển thị lịch sử mượn sách của sinh viên có email là kristie61@sis.hust.edu.vn**

CREATE OR REPLACE VIEW kristie61\_order\_history AS

SELECT ol.order\_id, o.book\_id, b.name, quantity, order\_time

FROM orderline ol JOIN orders o USING(order\_id)

JOIN student USING(student\_id)

JOIN book b USING(book\_id)

WHERE email = 'kristie61@sis.hust.edu.vn';

**4. Tạo trigger chỉ cho phép đơn online mới có trạng thái 'In queue'**

CREATE OR REPLACE FUNCTION func\_orderline\_status()

RETURNS TRIGGER AS

$$

DECLARE

method varchar(10);

stat varchar(15);

BEGIN

SELECT NEW.method, NEW.status INTO method, stat

FROM orderline;

IF(method = 'Offline' AND stat = 'In queue')

THEN

RAISE NOTICE 'Cannot insert';

RETURN NULL;

END IF;

RETURN NEW;

END;

$$

LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER trig\_order\_status\_insert

BEFORE INSERT ON orderline

FOR EACH ROW

EXECUTE PROCEDURE func\_orderline\_status();

**5. Hiển thị danh sách đơn mượn mà có ít nhất 1 đầu sách có số lượng lớn hơn 5(tối đa 10)**

SELECT ol.order\_id, o.book\_id, ol.student\_id, o.quantity

FROM orderline ol JOIN orders o USING(order\_id)

WHERE o.quantity > 5

LIMIT 10;

**6. Hiển thị danh sách các đầu sách có số lượng trong kho lớn nhất**

SELECT b.book\_id, b.name, a.stock

FROM book b JOIN archiving a USING(book\_id)

GROUP BY b.book\_id, a.stock

ORDER BY a.stock DESC;

**7. Tạo trigger tính số lượng sách trong kho mỗi khi có đơn được cập nhật**

CREATE OR REPLACE FUNCTION func\_order\_quantity()

RETURNS TRIGGER AS

$$

DECLARE

x int4;

y varchar(15);

bookid char(6);

orderid char(5);

BEGIN

FOR orderid IN

SELECT order\_id FROM orderline

LOOP

SELECT status INTO y

FROM orderline

WHERE order\_id = orderid;

SELECT book\_id, quantity INTO bookid, x

FROM orders

WHERE order\_id = orderid;

IF(y = 'Accepted') THEN

UPDATE archiving SET In\_order = In\_order + x, Available = Available - x

WHERE book\_id = bookid;

ELSEIF(y = 'Cancelled') THEN

UPDATE archiving SET In\_order = In\_order - x, Available = Available + x

WHERE book\_id = bookid;

ELSEIF(y = 'Successful') THEN

UPDATE archiving SET Away = Away + x, In\_order = In\_order - x

WHERE book\_id = bookid;

ELSEIF(y = 'Returned') THEN

UPDATE archiving SET Away = Away - x, Available = Available + x

WHERE book\_id = bookid;

END IF;

END LOOP;

RAISE NOTICE 'Update successful';

RETURN NEW;

END;

$$

LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER trig\_order\_quantity

AFTER UPDATE OF status OR INSERT ON orderline

FOR EACH ROW

EXECUTE PROCEDURE func\_order\_quantity();

**8. Hiển thị danh sách sinh viên ra khỏi phòng 173 sau 18h ngày 18/04/2024**

SELECT s.student\_id, s.first\_name, s.last\_name, check\_out

FROM check\_in JOIN student s USING(student\_id)

JOIN self\_study\_and\_reading\_room USING(room\_id)

WHERE room\_number = '173' AND check\_out > '2024-04-18 18:00:00';

**9. In ra tổng số lượng sách được mượn trong 3 tháng đầu năm 2024**

SELECT SUM(quantity)

FROM orderline JOIN orders USING(order\_id)

WHERE DATE(order\_time) BETWEEN '01-01-2024' AND '03-31-2024';

**10. Hiển thị danh sách 10 sinh viên có số lượng đơn mượn nhiều nhất theo từng tháng (kèm theo số lượng sách đã mượn sắp xếp theo thứ tự giảm dần) trong 6 tháng đầu năm 2024**

SELECT s.student\_id, s.first\_name, s.last\_name, COUNT(ol.order\_id), SUM(quantity)

FROM orderline ol JOIN orders o USING(order\_id)

RIGHT JOIN student s USING(student\_id)

GROUP BY s.student\_id, order\_time

HAVING DATE(order\_time) BETWEEN '01-01-2024' AND '06-30-2024'

ORDER BY COUNT(ol.order\_id) DESC, SUM(quantity) DESC

LIMIT 10;

1. **Đỗ Thanh Sơn - 20225665**

**1. Liệt kê các đầu sách có số lượng > 5 quyển, nhưng không có ai mượn**

**-- Cách 1:**

select book\_id, name, author, category from book

join stock st using (book\_id)

join orders ord using (book\_id)

where st.quantity > 5 and ord.quantity = 0

group by (book\_id);

**-- Cách 2:**

select book\_id, quantity from stock st

where st.quantity > 5

except

select book\_id, st.quantity from stock st

join orders ord using (book\_id)

where ord.quantity >= 1;

**2. Đưa ra danh sách 10 đầu sách được mượn nhiều nhất trong 6 tháng vừa rồi**

**-- Cách 1:**

select bk.book\_id, bk.name, count(bk.book\_id) as so\_sach

from book bk

join orders ord ON (bk.book\_id = ord.book\_id)

join orderline ol ON (ord.order\_id = ol.order\_id)

where ol.order\_time >= (CURRENT\_DATE - INTERVAL '6 month')

group by(bk.book\_id)

ORDER BY so\_sach DESC

LIMIT 10;

**-- Cách 2:**

WITH count\_borrowed\_books AS (

SELECT bk.book\_id, bk.name, COUNT(\*) AS so\_sach

FROM book bk

join orders ord ON (bk.book\_id = ord.book\_id)

join orderline ol ON (ord.order\_id = ol.order\_id)

where ol.order\_time >= (CURRENT\_DATE - INTERVAL '6 month')

GROUP BY bk.book\_id, bk.name

)

SELECT book\_id, name, so\_sach

FROM count\_borrowed\_books

ORDER BY so\_sach DESC

LIMIT 10;

-- Cách viết sử dụng with này (Common Table Expression - CTE) làm cho truy vấn dễ đọc và dễ hiểu hơn, đặc biệt là khi cần mở rộng hoặc bảo trì truy vấn trong tương lai.

**3. In ra các tài khoản email đã bị cấm do vi phạm quy tắc**

select email from student

join account acc using (email)

where acc.status = 'Suspended';

**4.Tạo trigger cập nhật số lượng sinh viên đang ở trong phòng nếu có sinh viên ra/vào**

CREATE OR REPLACE FUNCTION func\_room\_checkin\_out()

RETURNS TRIGGER AS

$$

BEGIN

IF(tg\_op = 'INSERT') THEN

UPDATE self\_study\_and\_reading\_room

SET current\_attendant = current\_attendant + 1

WHERE room\_id = NEW.room\_id;

ELSEIF(tg\_op = 'UPDATE') THEN

UPDATE self\_study\_and\_reading\_room

SET current\_attendant = current\_attendant - 1

WHERE room\_id = NEW.room\_id;

END IF;

RAISE NOTICE 'Update successful';

RETURN NEW;

END;

$$

LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER trig\_checkin\_out

AFTER UPDATE OF check\_out OR INSERT ON check\_in

FOR EACH ROW

EXECUTE PROCEDURE func\_room\_checkin\_out();

**5. Liệt kê tất cả các đầu sách mà thủ thư cho mượn trong 3 tháng vừa qua và đã trả lại đúng thời hạn**

**-- Cách 1:**

select book\_id, order\_id

from orders ord

join orderline ol using (order\_id)

where book\_return\_date BETWEEN (CURRENT\_DATE - INTERVAL '3 months') and CURRENT\_DATE

except

select book\_id, order\_id

from orders ord

join orderline ol using (order\_id)

where violation = 'X';

**-- Cách 2 (tối ưu hơn):**

select book\_id, order\_id

from orders ord

join orderline ol using (order\_id)

where book\_return\_date BETWEEN (CURRENT\_DATE - INTERVAL '3 months') and CURRENT\_DATE

and book\_return\_date <= deadline;

**6. Tạo hàm để tính số lượng sinh viên hiện tại đang ở trong phòng**

CREATE OR REPLACE FUNCTION func\_room\_current\_attendant()

RETURNS VOID AS

$$

DECLARE

x int4;

y char(6);

BEGIN

FOR y IN

SELECT room\_id FROM self\_study\_and\_reading\_room

LOOP

SELECT COUNT(student\_id) INTO x

FROM self\_study\_and\_reading\_room

WHERE room\_id = y AND check\_out is NULL;

UPDATE self\_study\_and\_reading\_room

SET current\_attendant = x

WHERE room\_id = y;

END LOOP;

RAISE NOTICE 'Update successful';

END;

$$

LANGUAGE plpgsql;

**7. Đưa ra thông tin về các sự kiện đã tổ chức trong ngày hội sách Việt Nam hàng năm (21/4) từ năm 2010 và những người tham gia, các nhà tài trợ cho các sự kiện đó**

select \* from special\_event

where date >= '2010-04-21';

**8. Đưa ra danh sách 5 tác giả có nhiều đầu sách nhất trong thư viện**

select count(distinct bk.book\_id) as so\_sach, bk.author from book bk

join stock st ON bk.book\_id = st.book\_id

group by (author)

order by so\_sach DESC

LIMIT 5;

**9. Đưa ra 7 thể loại sách ít nhất trong thư viện**

select count(category) as so\_theloai, bk.book\_id

from book bk

join stock st ON bk.book\_id = st.book\_id

group by (bk.book\_id)

order by so\_theloai ASC

Limit 7;

**10. Đưa ra danh sách các thủ thư làm việc trong khung giờ từ 8h-13h hàng ngày**

select \*

from librarian

join schedule using (librarian\_id)

where timeshift between '08:00:00' and '13:00:00';

1. **Đỗ Văn Vũ - MSSV:20225956**

**1. Lấy danh sách tất cả các đơn đặt hàng đã quá hạn trả sách và chi tiết sinh viên thực hiện đơn**

SELECT

o.order\_id,

o.book\_id,

s.first\_name,

s.last\_name,

ol.deadline,

ol.book\_return\_date

FROM

orders o

JOIN

orderline ol ON o.order\_id = ol.orderline\_id

JOIN

student s ON ol.student\_id = s.student\_id

WHERE

ol.book\_return\_date > ol.deadline;

**2. Lấy thông tin các sự kiện đặc biệt và các thủ thư đang làm việc trong khoảng thời gian đó**

SELECT

se.name AS event\_name,

se.organiser,

se.sponsor,

se.description,

se.destination,

se.target\_participant,

se.date,

se.schedule,

l.first\_name AS librarian\_first\_name,

l.last\_name AS librarian\_last\_name

FROM

special\_event se

LEFT JOIN

schedule s ON se.schedule = s.timeshift AND se.date = s.date

LEFT JOIN

librarian l ON s.librarian\_id = l.librarian\_id;

**3. Truy vấn tìm các thủ thư có lịch làm việc trong phòng tự học và đọc sách vào một ngày cụ thể**

SELECT

l.librarian\_id,

l.first\_name,

l.last\_name,

s.room\_id,

ssr.room\_number,

s.date,

s.timeshift

FROM

librarian l

JOIN

schedule s ON l.librarian\_id = s.librarian\_id

JOIN

self\_study\_and\_reading\_room ssr ON s.room\_id = ssr.room\_id

WHERE

s.date = '2023-12-15';

**4. Truy vấn danh sách các phòng tự học và đọc sách có số lượng người hiện tại vượt quá sức chứa tối đa, cùng với chi tiết thủ thư phụ trách**

SELECT

ssr.room\_id,

ssr.room\_number,

ssr.current\_attendant,

ssr.max\_capacity,

l.librarian\_id,

l.first\_name,

l.last\_name

FROM

self\_study\_and\_reading\_room ssr

JOIN

schedule s ON ssr.room\_id = s.room\_id

JOIN

librarian l ON s.librarian\_id = l.librarian\_id

WHERE

ssr.current\_attendant > ssr.max\_capacity;

**5. Truy vấn danh sách các sinh viên có đơn đặt hàng bị hủy nhiều nhất và số lượng đơn hàng bị hủy**

SELECT

s.student\_id,

s.first\_name,

s.last\_name,

COUNT(ol.orderline\_id) AS cancelled\_orders

FROM

student s

JOIN

orderline ol ON s.student\_id = ol.student\_id

WHERE

ol.status = 'Cancelled'

GROUP BY

s.student\_id, s.first\_name, s.last\_name

ORDER BY

cancelled\_orders DESC;

**6. Tạo trigger khi mà một đơn được tạo thì ngày hết hạn sẽ được tự động tính**

CREATE OR REPLACE FUNCTION func\_order\_date()

RETURNS TRIGGER AS

$$

DECLARE

BEGIN

UPDATE orderline SET expiration\_date = DATE(order\_time) + 3

WHERE order\_id = NEW.order\_id;

RAISE NOTICE 'Update successful';

RETURN NEW;

END;

$$

LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER trig\_order\_date\_insert

AFTER INSERT ON orderline

FOR EACH ROW

EXECUTE PROCEDURE func\_order\_date();

**7. Truy vấn thông tin về các sách được xuất kho trong khoảng thời gian nhất định và chi tiết đơn đặt hàng liên quan**

SELECT

b.book\_id,

b.name,

s.stock\_out\_date,

o.order\_id,

ol.order\_time,

ol.status

FROM

book b

JOIN

stock s ON b.book\_id = s.book\_id

JOIN

orders o ON b.book\_id = o.book\_id

JOIN

orderline ol ON o.order\_id = ol.orderline\_id

WHERE

s.stock\_out\_date BETWEEN '2023-01-01' AND '2023-12-31';

**8. tính tổng thời gian trong thư viện của một sinh viên trong năm 2023**

SELECT

s.student\_id,

s.first\_name,

s.last\_name,

ROUND(SUM(EXTRACT(EPOCH FROM (ci.check\_out - ci.check\_in)) / 3600), 2) AS total\_hours\_in\_self\_study\_room

FROM

student s

JOIN

check\_in ci ON s.student\_id = ci.student\_id

JOIN

self\_study\_and\_reading\_room ssr ON ci.room\_id = ssr.room\_id

WHERE

ci.check\_in BETWEEN '2023-01-01' AND '2023-12-31'

GROUP BY

s.student\_id, s.first\_name, s.last\_name;

**9. tạo index**

CREATE INDEX idx\_archiving\_room\_id ON archiving(room\_id);

CREATE INDEX idx\_archiving\_book\_id ON archiving(book\_id);

CREATE INDEX idx\_archive\_room\_room\_id ON archive\_room(room\_id);

EXPLAIN ANALYZE

SELECT

ar.room\_id,

ar.room\_number,

b.book\_id,

b.name,

SUM(a.available) AS total\_available\_books

FROM

archive\_room ar

JOIN

archiving a ON ar.room\_id = a.room\_id

JOIN

book b ON a.book\_id = b.book\_id

GROUP BY

ar.room\_id, ar.room\_number, b.book\_id, b.name;

**10. Tạo hàm giúp hệ thống đưa ra nhận xét về trạng thái của phòng dựa vào số lượng người trong phòng tự học**

CREATE OR REPLACE FUNCTION func\_room\_status()

RETURNS VOID AS

$$

DECLARE

x varchar(20);

y int4;

z int4;

room char(6);

BEGIN

FOR room IN

(SELECT room\_id FROM self\_study\_and\_reading\_room)

LOOP

SELECT current\_attendant, max\_capacity INTO y, z

FROM self\_study\_and\_reading\_room

WHERE room\_id = room;

IF(y > z)

THEN x := 'Overly crowded';

ELSEIF(y >= (z\*3/4))

THEN x := 'Crowded';

ELSEIF(y >= (z\*3/5))

THEN x := 'Fairly occupied';

ELSEIF(y >= (z\*2/5))

THEN x := 'Barely occupied';

ELSEIF(y >= (z\*1/5))

THEN x := 'Fairly empty';

ELSE x := 'Deserted';

END IF;

UPDATE self\_study\_and\_reading\_room SET status = x

WHERE room\_id = room;

END LOOP;

RAISE NOTICE 'Update successful';

END;

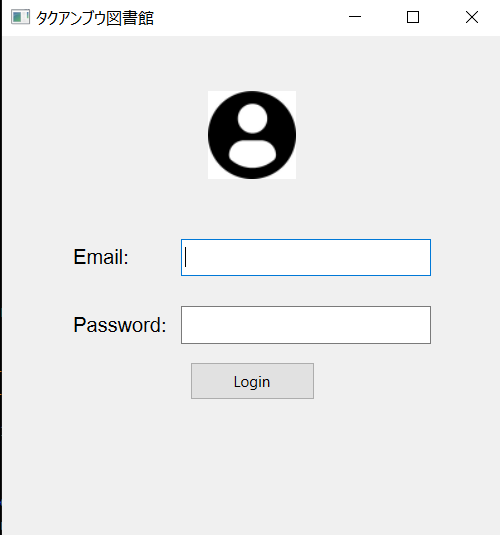
$$

LANGUAGE plpgsql;

# **CÁC CHỨC NĂNG CHÍNH**

1. **Giao diện đăng nhập.**

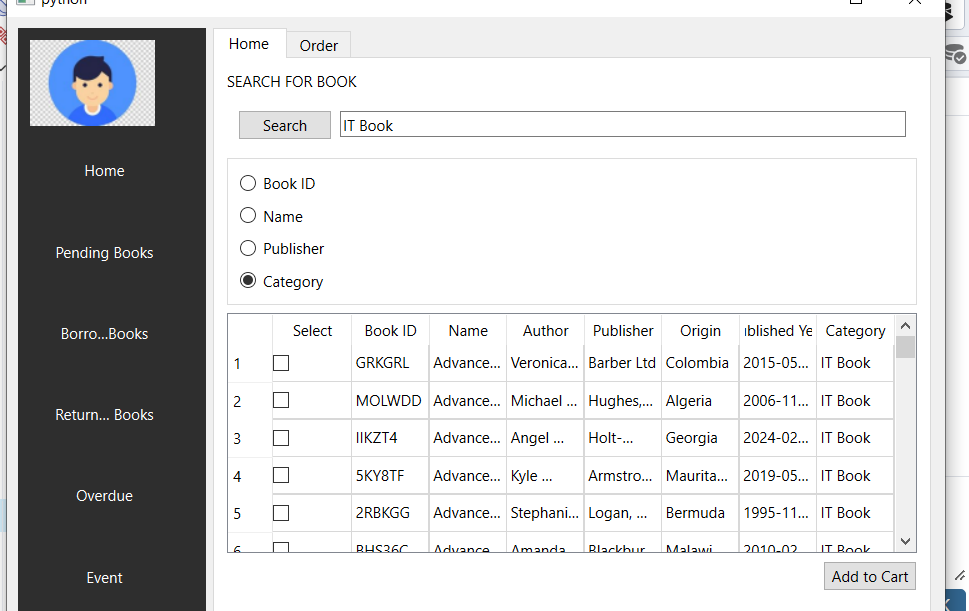
* Khi người dùng muốn đăng nhập thì sẽ hiển thị giao diện đăng nhập, nếu là thủ thư đăng nhập thì sẽ chuyển đến giao diện của thủ thư để quản lý, khi sinh viên đăng nhập thì sẽ chuyển đến giao diện của sinh viên để thực hiện các tác vụ mong muốn



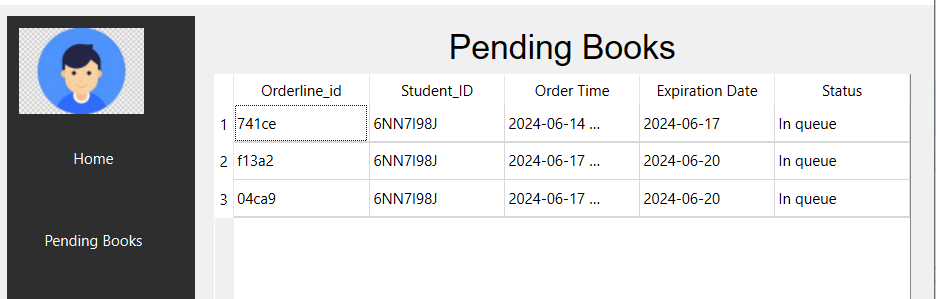
1. **Giao diện sinh viên**
   1. **Đăng nhập với tư cách là sinh viên**

****

* 1. **Tìm kiếm sách:**
* Sinh viên có thể tìm kiếm sách theo từng mục, sau đó có thể thêm vào giỏ hàng và chuyển đến tab Order để tiến hành hoàn thiện đơn hàng.

****

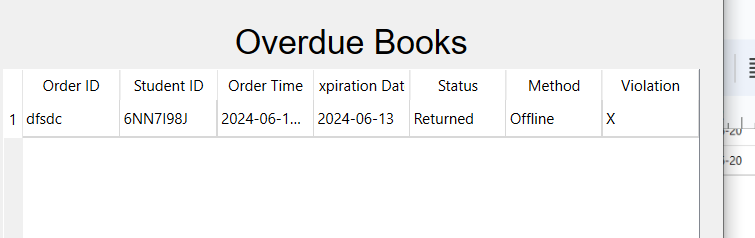
* 1. **Xem thông tin**
* Sinh viên có thể xem thông tin các đơn hàng.
  + Đơn hàng đang đợi và đã được duyệt:

****

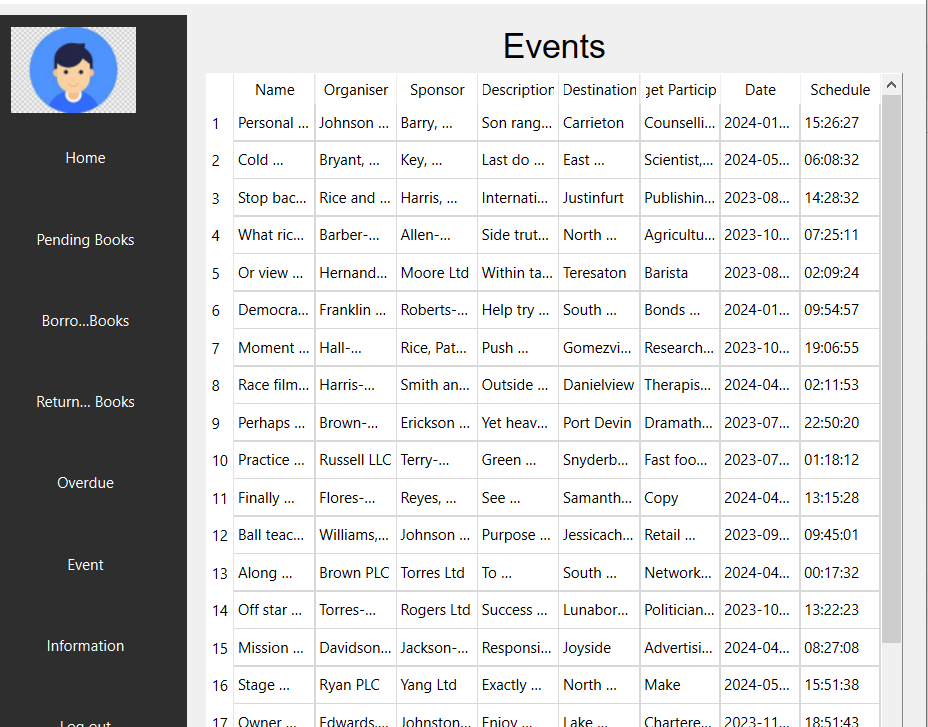
* Đơn hàng đang mượn:



* Đơn hàng quá hạn:



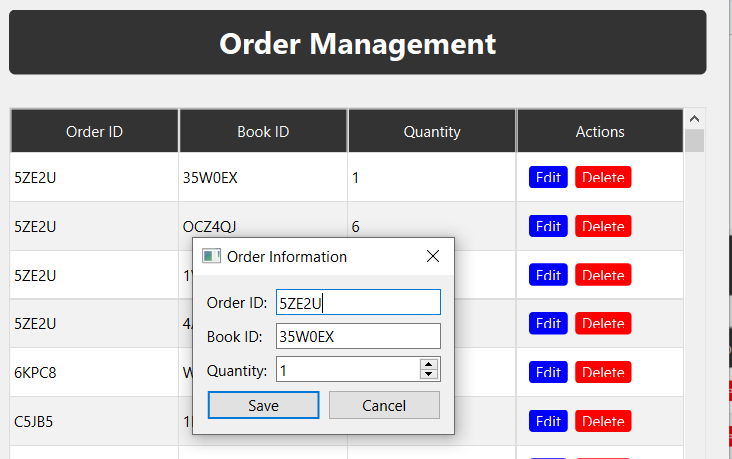
* Có thể xem được thông tin chi tiết các sự kiện được tổ chức bên trong thư viện:



1. **Giao diện cho nhà quản lý (admin)**

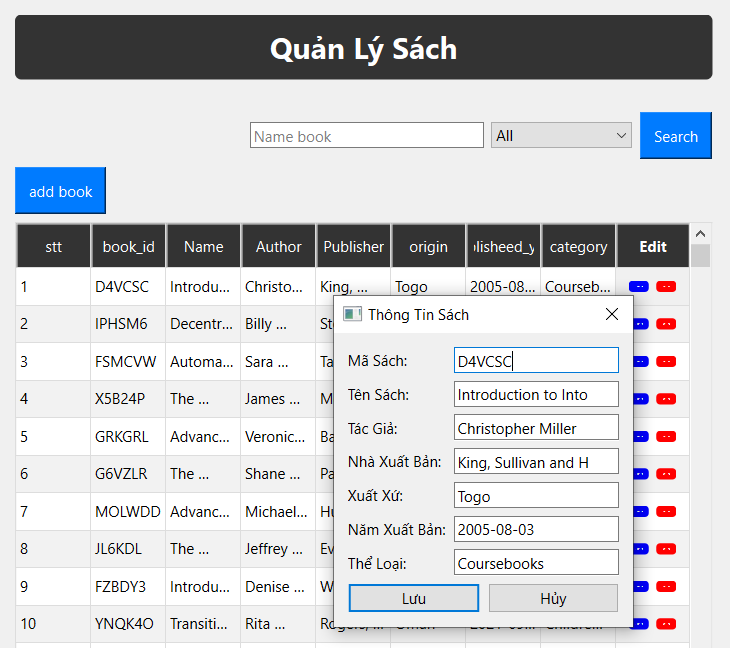
* Đây là giao diện khi đăng nhập vào với tài khoản admin. Tại đây quản lí có thể thực hiện các tác vụ quản lí sách, quản lí order, các tài khoản, theo dõi trạng thái các phòng học, đưa ra các thông báo,...

**3.1. Quản lí order:**

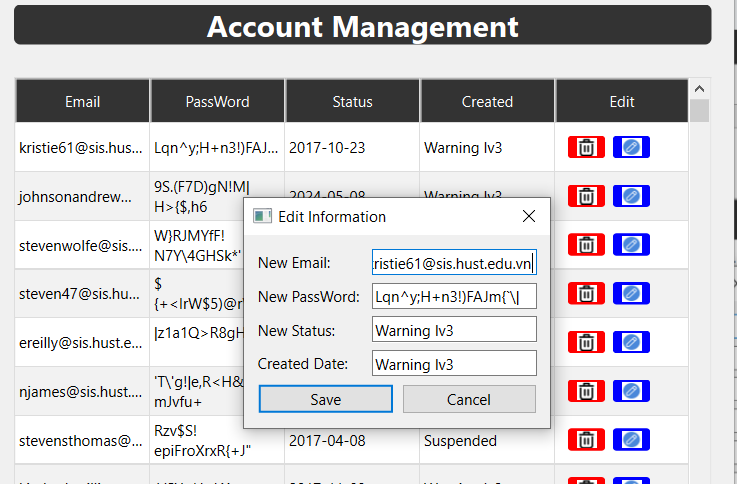


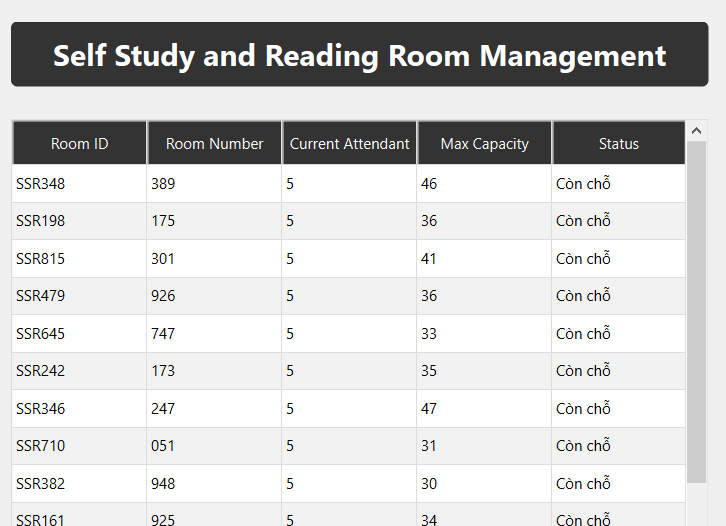
Tại đây admin có thể chỉnh sửa các thông tin của từng đơn hàng.

**3.2. Thêm hoặc chỉnh sửa thông tin sách**

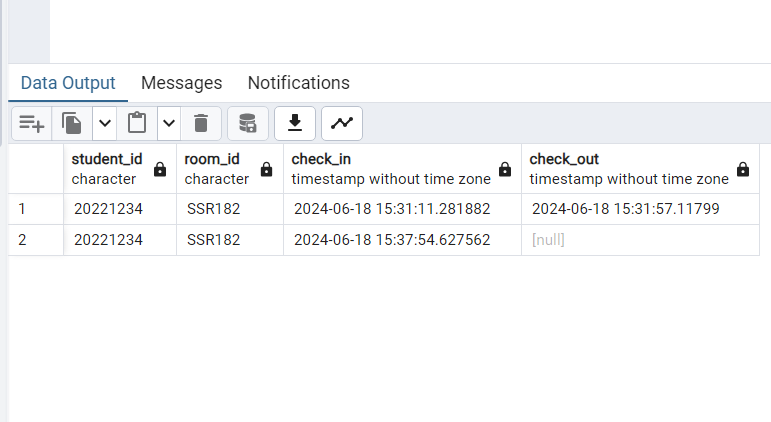
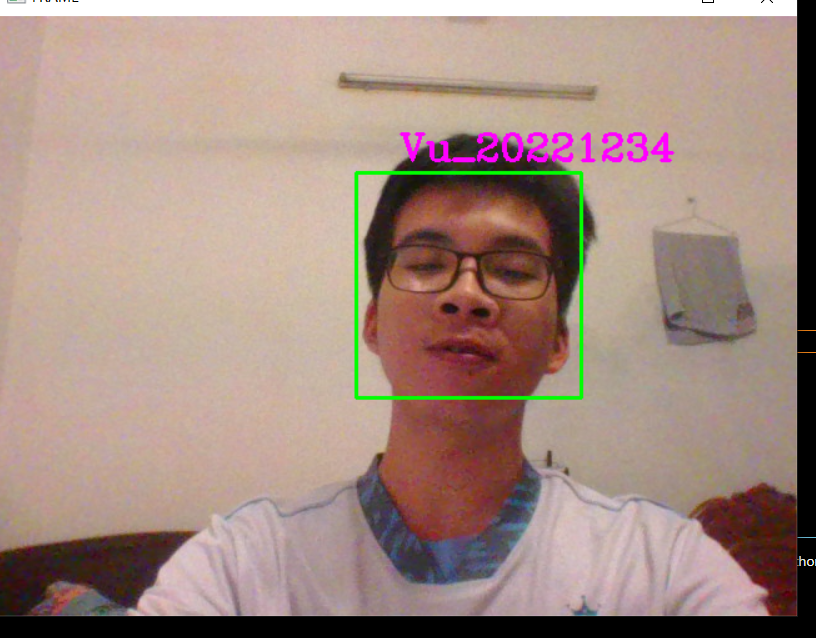
****

**3.3. Quản lý account:**

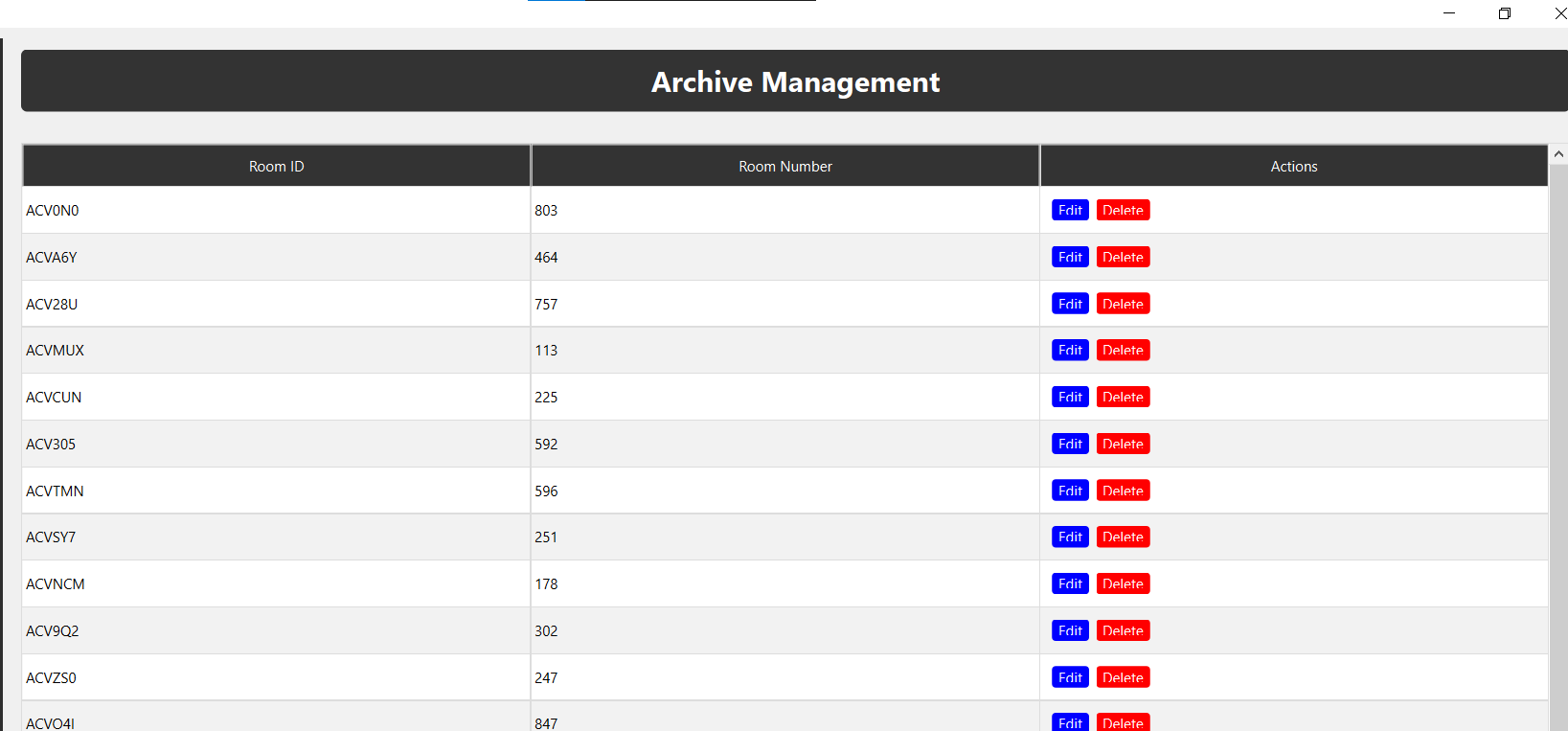
****

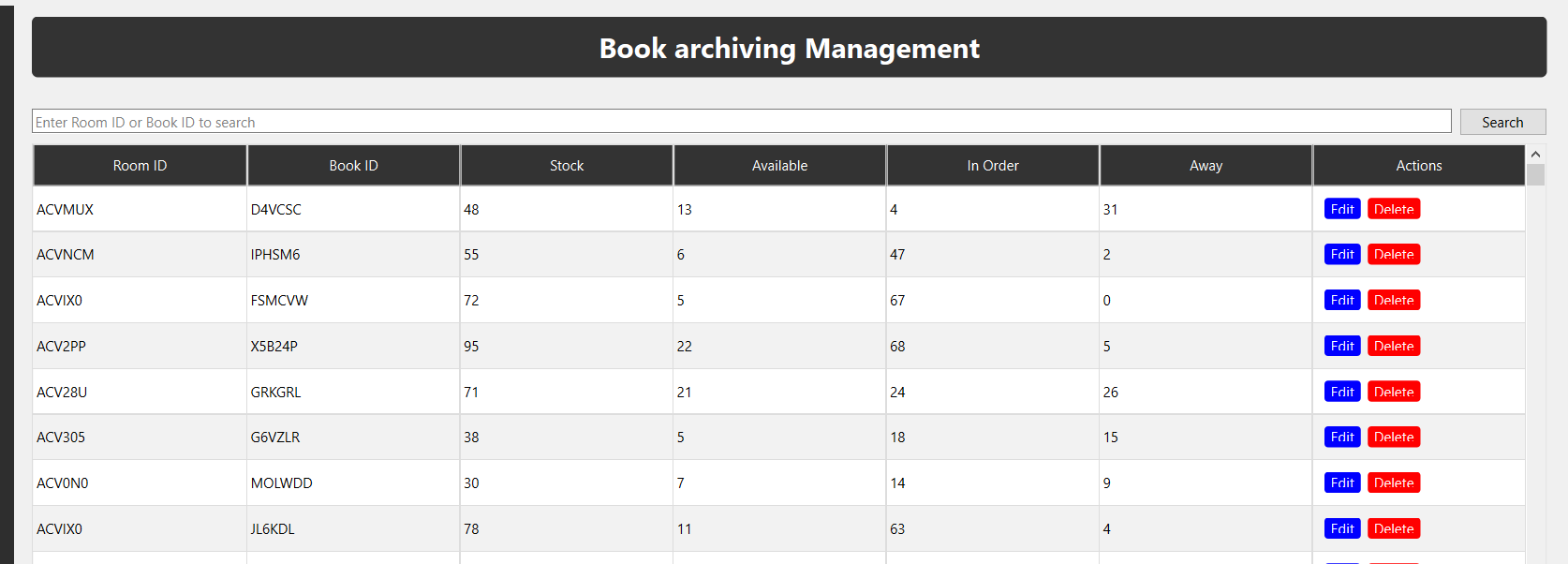
**3.4. Quản lý các phòng tự học **

* Đồng thời admin cũng có thể xem được những sinh viên nào đã ra, vào phòng tự học tại thời điểm nào nhờ bảng check\_in, mỗi phòng sẽ camera nhận diện khuôn mặt để nhận diện thời điểm ra, vào của sinh viên:

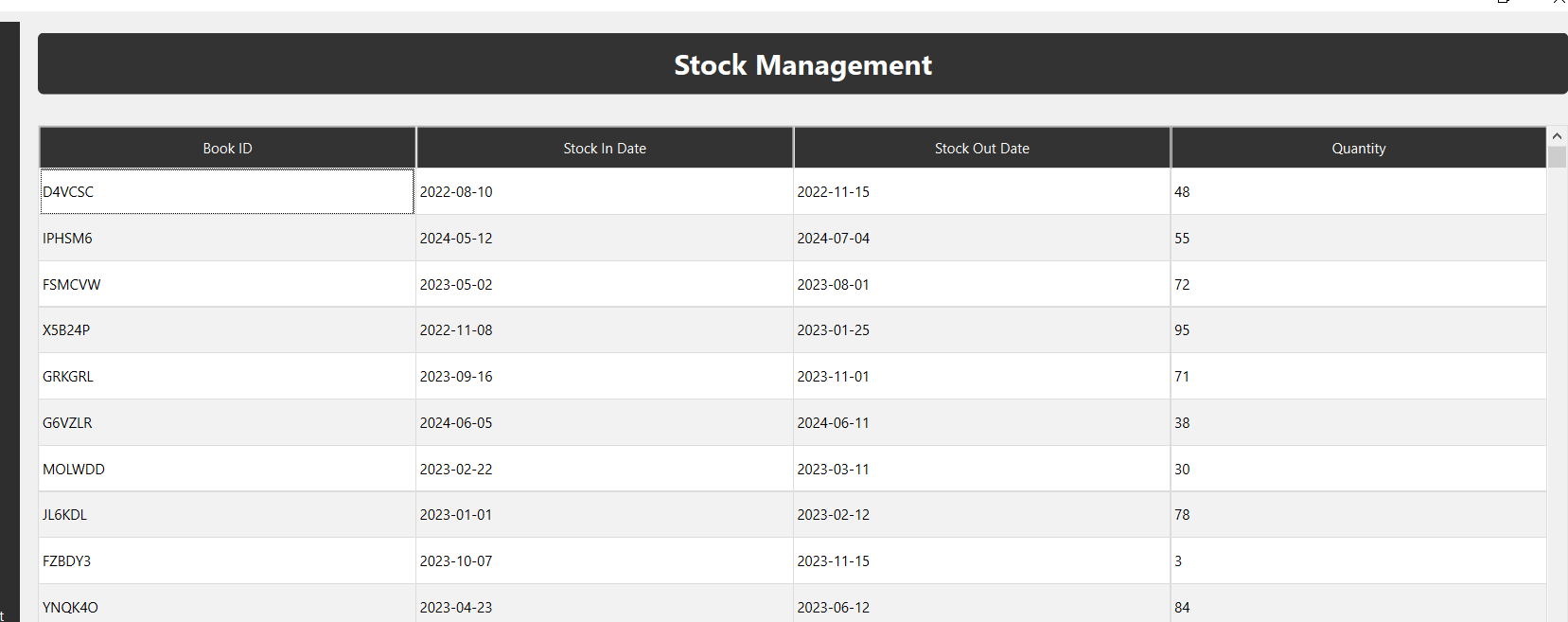
****

**3.5. Quản lý các phòng kho chứa sách.**

****

**3.6. Quản lí sách trong kho: **

**3.7. Quản lý xuất, nhập kho:**

****

1. **KẾT LUẬN**
2. **Về dự án**

Bọn em đã rất cố gắng hoàn thiện dự án này và đảm bảo dự án được vận hành một cách đơn giản và hiệu quả nhất!

1. **Những gì đã học được qua dự án**

Qua dự án này, cũng như qua quá trình học môn Thực hành cơ sở dữ liệu, bọn em đã được tiếp nhận những kiến thức rất bổ ích về hệ quản trị cơ sở dữ liệu PostgreSQL, được tìm hiểu cặn kẽ về database và cách viết các câu lệnh truy vấn. Đồng thời, trong quá trình làm việc nhóm, bọn em cũng được học hỏi thêm về ngôn ngữ Python và framework PyQt. Bên cạnh đó mỗi thành viên trong nhóm em đều hiểu biết thêm về mảng Computer Vision thông qua mô hình nhận diện khuôn mặt. Hơn nữa, bọn em có thêm kinh nghiệm để làm việc nhóm hiệu quả hơn, quản lý tiến độ công việc và phân chia nhiệm vụ hợp lý cho các thành viên. Đây thật sự là một bài tập rất giá trị đối với chúng em.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_HẾT\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_