

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
CÔNG NGHỆ THÔNG TIN TRONG CHUYỂN ĐỔI SỐ

MODULE QUẢN LÝ PHÒNG LAB

Sinh viên thực hiện : Nguyễn Quý Trung
Trình Quốc Đạt
Trần Quốc Việt

Ngành : Công nghệ thông tin

Giảng viên hướng dẫn : ThS. Lê Trung Hiếu
ThS. Nguyễn Thái Khánh

Lời cảm ơn

Chúng em xin bày tỏ lòng biết ơn vô cùng to lớn đến Thạc sĩ Lê Trung Hiếu và Thạc sĩ Nguyễn Thái Khánh, người đã tận tình dìu dắt, giảng dạy và truyền đạt những kiến thức chuyên sâu về môn học Công nghệ thông tin trong chuyển đổi số trong suốt quá trình học. Thầy không chỉ là người thầy truyền đạt kiến thức mà còn là người truyền cảm hứng, luôn sẵn lòng giải đáp mọi thắc mắc và đưa ra những lời khuyên quý báu, giúp chúng em có cái nhìn toàn diện và sâu sắc hơn về đề tài. Sự hướng dẫn tận tâm của thầy chính là kim chỉ nam giúp nhóm chúng em vượt qua mọi khó khăn và hoàn thành tốt bài tập lớn này.

Chúng em cũng xin chân thành cảm ơn Phòng Lab đã tạo mọi điều kiện thuận lợi về tài liệu, cơ sở vật chất và các nguồn thông tin cần thiết, giúp quá trình nghiên cứu và thực hiện bài tập được diễn ra suôn sẻ.

Cuối cùng, chúng em xin gửi lời cảm ơn đến gia đình, bạn bè và những người thân đã luôn là nguồn động viên lớn lao, là điểm tựa vững chắc để chúng em có đủ nghị lực và quyết tâm hoàn thành nhiệm vụ học tập của mình.

Đây là một bài tập lớn mang nhiều tâm huyết và công sức, nhưng nhờ có sự hỗ trợ của tất cả mọi người, chúng em đã tích lũy được những kiến thức và kinh nghiệm vô cùng quý giá, giúp chúng em tự tin hơn trên con đường học vấn và sự nghiệp sau này.

Xin chân thành cảm ơn!

Mục lục

1	Giới thiệu đề tài	1
1.1	Bối cảnh nghiên cứu	1
1.2	Lý do chọn đề tài	1
1.3	Mục tiêu nghiên cứu	2
1.4	Đối tượng và phạm vi nghiên cứu	2
1.5	Ý nghĩa thực tiễn của đề tài	2
1.6	Kết cấu báo cáo	3
2	Phân tích hệ thống	4
2.1	Khảo sát hiện trạng	4
2.2	Phân tích yêu cầu hệ thống	5
2.2.1	Yêu cầu chức năng	5
2.2.2	Yêu cầu phi chức năng	5
2.3	Biểu đồ Use Case minh họa	6
2.4	Biểu đồ hoạt động (Activity Diagram)	6
2.5	Kịch bản sử dụng hệ thống (Scenario)	6
3	Thiết kế hệ thống	8
3.1	Kiến trúc tổng thể của hệ thống	8
3.2	Thiết kế cơ sở dữ liệu	9
3.2.1	Các bảng dữ liệu chính	9
3.3	Thiết kế chức năng	10
3.3.1	Chức năng của Sinh viên	10
3.3.2	Chức năng của Giảng viên (Admin)	10
3.4	Thiết kế giao diện người dùng (UI/UX)	11
4	Triển khai và Thử nghiệm hệ thống	12
4.1	Môi trường phát triển và công cụ sử dụng	12
4.2	Cài đặt và thiết lập hệ thống	12
4.2.1	Cài đặt môi trường Django	12

4.2.2	Cấu hình cơ sở dữ liệu (SQLite hoặc PostgreSQL)	13
4.2.3	Tạo bảng và siêu người dùng	14
4.3	Các chức năng chính đã triển khai	14
4.3.1	Chức năng: Sinh viên đặt lịch thiết bị	14
4.3.2	Chức năng: Giảng viên xét duyệt đơn đặt lịch	15
4.3.3	Giao diện quản lý đơn chờ duyệt (dành cho giảng viên)	16
4.4	Kiểm thử hệ thống	17
4.4.1	Mục tiêu kiểm thử	17
4.4.2	Kịch bản kiểm thử (Test Scenarios)	17
4.4.3	Đánh giá kết quả thử nghiệm	17
5	Ứng dụng AI và Kết luận	18
5.1	Ứng dụng AI trong hệ thống quản lý đặt lịch thiết bị	18
5.1.1	Dự đoán nhu cầu sử dụng thiết bị (AI Forecasting)	18
5.1.2	Phân tích hành vi người dùng (User Behavior Analytics)	18
5.1.3	Hệ thống gợi ý (Recommendation System)	19
5.1.4	Chatbot hỗ trợ (AI-powered Chatbot)	19
5.1.5	Phát hiện và cảnh báo bất thường (Anomaly Detection)	19
5.2	Lợi ích khi áp dụng AI	19
5.3	Hạn chế của hệ thống hiện tại	20
5.4	Định hướng phát triển	20
5.5	Kết luận chung	20

Chương 1

Giới thiệu đề tài

1.1 Bối cảnh nghiên cứu

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trên toàn cầu, các cơ sở giáo dục, đặc biệt là các trường đại học, ngày càng chú trọng việc áp dụng công nghệ thông tin để tối ưu hóa hoạt động quản lý và giảng dạy. Một trong những vấn đề nổi bật hiện nay là việc quản lý và sử dụng thiết bị thí nghiệm, dụng cụ học tập.

Ở nhiều trường, sinh viên khi muốn mượn thiết bị phải thực hiện qua quy trình thủ công: viết giấy mượn, ký nhận, hoặc đăng ký qua file Excel. Cách làm này tiềm ẩn nhiều hạn chế:

- Dễ xảy ra trùng lịch sử dụng thiết bị.
- Khó khăn trong việc theo dõi trạng thái mượn – trả.
- Thiếu tính minh bạch và kịp thời trong việc xét duyệt.
- Gây áp lực cho cán bộ, giảng viên quản lý thiết bị.

Chính vì vậy, việc xây dựng một hệ thống quản lý đặt lịch thiết bị trực tuyến là cần thiết, góp phần hiện thực hóa mục tiêu số hóa hoạt động quản lý giáo dục.

1.2 Lý do chọn đề tài

Các mục tiêu chính của đề tài bao gồm:

- Đáp ứng nhu cầu thực tế trong việc quản lý và tối ưu hóa thiết bị tại trường học.
- Góp phần thúc đẩy chuyển đổi số trong giáo dục, phù hợp với xu thế chung của xã hội.

- Đề tài mang tính ứng dụng cao, có thể triển khai thực tế, đồng thời giúp sinh viên rèn luyện kỹ năng lập trình web, quản lý cơ sở dữ liệu, và triển khai hệ thống thông tin.
- Có thể dễ dàng mở rộng sang các lĩnh vực khác như: quản lý phòng học, quản lý thư viện, hay quản lý đặt lịch phòng thí nghiệm.

1.3 Mục tiêu nghiên cứu

Mục tiêu chính của đề tài là xây dựng một ứng dụng web “Hệ thống Quản lý Đặt lịch Thiết bị” với các tính năng sau:

- Cho phép sinh viên đăng ký, đăng nhập và đặt lịch mượn thiết bị trực tuyến.
- Hỗ trợ giảng viên (admin) trong việc xét duyệt đơn, quản lý người dùng, quản lý thiết bị và phòng ban.
- Tích hợp chức năng gửi email thông báo cho sinh viên khi đơn được duyệt.
- Xây dựng hệ thống có giao diện thân thiện, dễ sử dụng, đồng thời đảm bảo tính bảo mật dữ liệu và phân quyền người dùng rõ ràng.

1.4 Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Các hệ thống quản lý thiết bị hiện nay.
- Các công nghệ hỗ trợ xây dựng ứng dụng web: Python, Django/Flask, Bootstrap 5, PostgreSQL/SQLite.

Phạm vi nghiên cứu:

- Quản lý đặt lịch thiết bị trong phạm vi trường/khoa.
- Hệ thống tập trung vào hai vai trò chính: Sinh viên và Giảng viên (Admin).
- Chưa tích hợp các chức năng nâng cao như: kết nối IoT để quản lý thiết bị tự động, thanh toán trực tuyến, hay tích hợp AI dự đoán nhu cầu.

1.5 Ý nghĩa thực tiễn của đề tài

Đối với sinh viên:

- Chủ động trong việc đặt lịch mượn thiết bị.
- Theo dõi được trạng thái yêu cầu của mình (chờ duyệt, được duyệt, bị từ chối).
- Giảm thời gian chờ đợi, hạn chế việc di chuyển nhiều lần để đăng ký.

Đối với giảng viên/admin:

- Hỗ trợ quản lý thiết bị khoa học, minh bạch.
- Giảm tải công việc xét duyệt, vì hệ thống tự động lưu trữ, phân loại và thông báo.
- Dễ dàng thống kê, báo cáo tình hình sử dụng thiết bị.

Đối với nhà trường:

- Thúc đẩy quá trình chuyển đổi số trong công tác quản lý.
- Góp phần xây dựng hình ảnh một trường đại học hiện đại, ứng dụng công nghệ trong giảng dạy và nghiên cứu.
- Có thể mở rộng, tích hợp với các hệ thống quản lý khác (quản lý sinh viên, thư viện số, e-learning).

1.6 Kết cấu báo cáo

Báo cáo gồm 5 chương chính:

- Chương 1: Giới thiệu đề tài
- Chương 2: Phân tích hệ thống
- Chương 3: Thiết kế hệ thống
- Chương 4: Triển khai và thử nghiệm
- Chương 5: Ứng dụng AI và kết luận

Chương 2

Phân tích hệ thống

2.1 Khảo sát hiện trạng

Hiện nay, tại nhiều trường đại học, việc quản lý thiết bị học tập, thí nghiệm chủ yếu thực hiện theo phương pháp thủ công:

Quy trình hiện tại:

- Sinh viên muốn mượn thiết bị phải liên hệ trực tiếp với giảng viên hoặc quản lý phòng ban.
- Giảng viên/nhân viên quản lý ghi chép thông tin mượn – trả vào sổ hoặc bảng Excel.
- Việc xét duyệt hoàn toàn thủ công, phụ thuộc vào sự có mặt của giảng viên.

Khó khăn gặp phải:

- Không kiểm soát được trùng lịch sử dụng thiết bị.
- Thiếu sự minh bạch trong quản lý.
- Tốn thời gian, gây quá tải cho giảng viên khi có nhiều đơn mượn.
- Sinh viên khó theo dõi được trạng thái yêu cầu của mình.

Chính vì vậy, cần có một hệ thống quản lý đặt lịch trực tuyến, giúp tự động hóa quy trình mượn – trả thiết bị, đảm bảo minh bạch, thuận tiện và tiết kiệm thời gian.

2.2 Phân tích yêu cầu hệ thống

2.2.1 Yêu cầu chức năng

Đối với sinh viên:

- Đăng ký tài khoản và đăng nhập hệ thống.
- Chọn thiết bị theo phòng ban, thời gian nhận và trả.
- Gửi yêu cầu đặt lịch để giảng viên xét duyệt.
- Theo dõi danh sách các đơn đặt lịch cá nhân, kèm trạng thái (đang chờ duyệt, đã được duyệt, bị từ chối).

Đối với giảng viên (Admin):

- Đăng nhập vào trang quản trị.
- Quản lý thiết bị: thêm, sửa, xóa thông tin.
- Quản lý phòng ban.
- Quản lý người dùng (danh sách sinh viên, giảng viên).
- Xem và xét duyệt các đơn đặt lịch: phê duyệt hoặc từ chối.
- Gửi thông báo tự động (qua email) cho sinh viên sau khi phê duyệt.

Đối với hệ thống:

- Phân quyền rõ ràng giữa sinh viên và giảng viên.
- Lưu trữ toàn bộ thông tin đơn đặt lịch, lịch sử mượn – trả.
- Gửi email tự động khi trạng thái đơn thay đổi.

2.2.2 Yêu cầu phi chức năng

- Hiệu năng: Hệ thống phải phản hồi dưới 2 giây cho mỗi thao tác thông thường.
- Bảo mật: Mật khẩu người dùng được mã hóa; phân quyền chặt chẽ.
- Tính khả dụng: Giao diện thân thiện, hỗ trợ trên trình duyệt phổ biến (Chrome, Edge, Firefox).
- Tính mở rộng: Có thể dễ dàng mở rộng để quản lý nhiều phòng ban, nhiều loại thiết bị

khác nhau.

- Tính ổn định: Đảm bảo dữ liệu không bị mất mát khi nhiều người truy cập đồng thời.

2.3 Biểu đồ Use Case minh họa

Tác nhân: Sinh viên. Kịch bản:

- Sinh viên đăng nhập vào hệ thống.
- Sinh viên chọn thiết bị → chọn thời gian nhận và trả.
- Hệ thống kiểm tra tính hợp lệ (thiết bị có sẵn, không trùng lịch).
- Nếu hợp lệ → lưu đơn đặt lịch ở trạng thái "Chờ duyệt".
- Nếu không hợp lệ → thông báo lỗi.

2.4 Biểu đồ hoạt động (Activity Diagram)

Ví dụ cho quá trình Xét duyệt đơn đặt lịch:

- Giảng viên đăng nhập vào hệ thống.
- Chọn menu “Đơn chờ duyệt”.
- Hệ thống hiển thị danh sách các đơn đặt lịch.
- Giảng viên chọn 1 đơn → hành động Phê duyệt hoặc Từ chối.
- Hệ thống cập nhật trạng thái đơn.
- Hệ thống gửi email thông báo đến sinh viên.

2.5 Kịch bản sử dụng hệ thống (Scenario)

Kịch bản 1: Sinh viên đặt lịch thành công

- Input: Sinh viên đăng nhập, chọn thiết bị, chọn thời gian hợp lệ.
- Output: Hệ thống lưu đơn ở trạng thái “Chờ duyệt”.

Kịch bản 2: Giảng viên phê duyệt đơn

- Input: Giảng viên mở danh sách đơn, chọn phê duyệt.

- Output: Đơn chuyển sang trạng thái “Đã duyệt”, sinh viên nhận email thông báo.

Kịch bản 3: Trùng lịch

- Input: Sinh viên đặt lịch cho thiết bị đã có người khác đăng ký trong cùng thời gian.
- Output: Hệ thống báo lỗi, yêu cầu chọn thời gian khác.

Chương 3

Thiết kế hệ thống

3.1 Kiến trúc tổng thể của hệ thống

Hệ thống Quản lý Đặt lịch Thiết bị được xây dựng theo mô hình 3 lớp (Three-Tier Architecture), dựa trên framework Django (Python), gồm: Lớp giao diện (Presentation Layer):

- HTML, CSS (Bootstrap 5), JavaScript.
- Hiển thị form đăng nhập, đặt lịch, bảng quản lý đơn.
- Cung cấp trải nghiệm thân thiện cho người dùng.

Lớp xử lý nghiệp vụ (Business Logic Layer):

- Django Views và Controllers.
- Xử lý yêu cầu của sinh viên (đặt lịch) và giảng viên (xét duyệt).
- Kiểm tra tính hợp lệ (không trùng lịch, thiết bị còn khả dụng).

Lớp dữ liệu (Data Layer):

- CSDL: PostgreSQL (môi trường sản phẩm) hoặc SQLite (môi trường phát triển).
- Các bảng chính: User, Department, Device, Booking.
- Lưu trữ an toàn, hỗ trợ truy vấn nhanh.

3.2 Thiết kế cơ sở dữ liệu

3.2.1 Các bảng dữ liệu chính

Bảng User (Người dùng)

- id: khóa chính
- username: tên đăng nhập
- password: mật khẩu (được mã hóa)
- email: email
- role: phân quyền (student, admin)
- full-name: họ tên

Bảng Department (Phòng ban)

- id: khóa chính
- name: tên phòng ban
- description: mô tả

Bảng Device (Thiết bị)

- id: khóa chính
- name: tên thiết bị
- department-id: FK → Department
- status: trạng thái (hoạt động, hỏng, bảo trì)

Bảng Booking (Đơn đặt lịch)

- id: khóa chính
- student-id: FK → User
- device-id: FK → Device
- start-time: thời gian bắt đầu
- end-time: thời gian kết thúc
- status: pending, approved, rejected

3.3 Thiết kế chức năng

3.3.1 Chức năng của Sinh viên

Đăng nhập/Đăng ký

- Input: username, password, email.
- Output: tạo tài khoản, đăng nhập vào hệ thống.

Đặt lịch thiết bị

- Input: chọn thiết bị, thời gian mượn – trả.
- Process: hệ thống kiểm tra trùng lịch.
- Output: tạo đơn đặt lịch ở trạng thái “pending”.

Xem lịch sử đặt lịch

- Input: yêu cầu xem danh sách.
- Output: hiển thị tất cả đơn, trạng thái tương ứng.

3.3.2 Chức năng của Giảng viên (Admin)

Xét duyệt đơn đặt lịch

- Input: chọn đơn → phê duyệt hoặc từ chối.
- Process: cập nhật trạng thái trong CSDL.
- Output: sinh viên nhận email thông báo.

Quản lý thiết bị

- Thêm/sửa/xóa thiết bị.

Quản lý phòng ban

- Thêm/sửa/xóa phòng ban.

Quản lý người dùng

- Xem danh sách sinh viên, giảng viên.

3.4 Thiết kế giao diện người dùng (UI/UX)

Giao diện đăng nhập: Form nhập username, password. Sinh viên:

- Trang đặt lịch thiết bị (chọn thiết bị, chọn thời gian).
- Trang danh sách đơn đặt lịch cá nhân.

Giảng viên:

- Dashboard quản lý.
- Trang xét duyệt đơn (danh sách pending).
- Trang quản lý phòng ban, thiết bị, người dùng..

Chương 4

Triển khai và Thử nghiệm hệ thống

4.1 Môi trường phát triển và công cụ sử dụng

Để triển khai hệ thống “Hệ thống Quản lý Đặt lịch Thiết bị”, nhóm sử dụng các công nghệ và công cụ như sau:

Thành phần	Công nghệ / Phiên bản	Ghi chú
Ngôn ngữ lập trình	Python 3.10+	Dùng để phát triển toàn bộ hệ thống
Framework	Django 4.x	Web framework chính
Cơ sở dữ liệu (dev)	SQLite 3	Dùng trong môi trường phát triển
Cơ sở dữ liệu (prod)	PostgreSQL 14+	Dùng trong triển khai thực tế
Giao diện	HTML5 + CSS3 + Bootstrap 5	Thiết kế giao diện responsive
JavaScript	ES6	Xử lý tương tác giao diện
IDE	PyCharm / VS Code	Soạn thảo và quản lý mã nguồn
Công cụ quản lý gói	pip	Cài đặt thư viện Python
Thư viện gửi email	Django send_mail	Gửi thông báo cho sinh viên

4.2 Cài đặt và thiết lập hệ thống

4.2.1 Cài đặt môi trường Django

Hướng dẫn Cài đặt Môi trường

```
# 1. Tạo môi trường ảo
python -m venv venv
```



```
# 2. Kích hoạt môi trường
# Trên Windows:
venv\Scripts\activate
# Trên macOS/Linux:
source venv/bin/activate

# 3. Cài đặt Django
pip install django

# 4. Tạo project mới
django-admin startproject booking_system

# 5. Tạo ứng dụng con
cd booking_system
python manage.py startapp main
```

4.2.2 Cấu hình cơ sở dữ liệu (SQLite hoặc PostgreSQL)

SQLite (mặc định cho môi trường phát triển):

```
# booking_system/settings.py
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

PostgreSQL (triển khai thực tế):

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'bookingdb',
        'USER': 'postgres',
        'PASSWORD': 'your_password',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

```
}
```

4.2.3 Tạo bảng và siêu người dùng

```
# Tạo các bảng
python manage.py makemigrations
python manage.py migrate

# Tạo tài khoản admin
python manage.py createsuperuser
```

4.3 Các chức năng chính đã triển khai

4.3.1 Chức năng: Sinh viên đặt lịch thiết bị

View trong Django:

```
@login_required
def book_device(request):
    if request.method == 'POST':
        form = BookingForm(request.POST)
        if form.is_valid():
            booking = form.save(commit=False)
            booking.student = request.user
            booking.status = 'pending'
            booking.save()
            messages.success(request, "Yêu cầu đặt lịch đã được gửi và đang chờ duyệt")
            return redirect('my_bookings')
        else:
            form = BookingForm()
    return render(request, 'student/book_device.html', {'form': form})
```

Form đặt lịch:

```
class BookingForm(forms.ModelForm):
    class Meta:
        model = Booking
        fields = ['device', 'start_time', 'end_time']
```

Giao diện HTML (Bootstrap 5):

```
<form method="post">
  {% csrf_token %}
  {{ form.as_p }}
  <button type="submit" class="btn btn-primary">Đặt lịch</button>
</form>
```

4.3.2 Chức năng: Giảng viên xét duyệt đơn đặt lịch

View phê duyệt:

```
@login_required
def approve_booking(request, booking_id):
    booking = get_object_or_404(Booking, id=booking_id)
    booking.status = 'approved'
    booking.save()

    send_mail(
        subject="Yêu cầu đặt lịch đã được duyệt",
        message=f"Thiết bị {booking.device.name} đã được duyệt từ {booking.start_time}",
        from_email="admin@university.edu",
        recipient_list=[booking.student.email],
        fail_silently=False,
    )

    messages.success(request, "Đơn đặt lịch đã được phê duyệt.")
    return redirect('manage_bookings')
```

View từ chối:

```
@login_required
def reject_booking(request, booking_id):
    booking = get_object_or_404(Booking, id=booking_id)
    booking.status = 'rejected'
    booking.save()

    send_mail(
        subject="Yêu cầu đặt lịch bị từ chối",
```

```
        message=f"Thiết bị {booking.device.name} đã bị từ chối.",
        from_email="admin@university.edu",
        recipient_list=[booking.student.email],
        fail_silently=False,
    )

    messages.warning(request, "Đơn đặt lịch đã bị từ chối.")
    return redirect('manage_bookings')
```

4.3.3 Giao diện quản lý đơn chờ duyệt (dành cho giảng viên)

```
<table class="table table-bordered">
    <thead>
        <tr>
            <th>Sinh viên</th>
            <th>Thiết bị</th>
            <th>Thời gian</th>
            <th>Trạng thái</th>
            <th>Hành động</th>
        </tr>
    </thead>
    <tbody>
        {% for b in bookings %}
        <tr>
            <td>{{ b.student.full_name }}</td>
            <td>{{ b.device.name }}</td>
            <td>{{ b.start_time }} - {{ b.end_time }}</td>
            <td>{{ b.get_status_display }}</td>
            <td>
                <a href="{% url 'approve_booking' b.id %}" class="btn btn-success btn-sm">Đ
                <a href="{% url 'reject_booking' b.id %}" class="btn btn-danger btn-sm">Từ
            </td>
        </tr>
        {% endfor %}
    </tbody>
</table>
```

4.4 Kiểm thử hệ thống

4.4.1 Mục tiêu kiểm thử

Đảm bảo hệ thống hoạt động đúng với yêu cầu chức năng.

Kiểm tra tính ổn định và bảo mật cơ bản.

Kiểm tra quy trình từ sinh viên đặt lịch → giảng viên duyệt → gửi email.

4.4.2 Kịch bản kiểm thử (Test Scenarios)

STT	Mô tả kiểm thử	Kết quả mong đợi	Kết quả thực tế
1	Sinh viên đăng ký tài khoản mới	Tài khoản được tạo thành công	✓
2	Sinh viên đăng nhập hệ thống	Truy cập vào dashboard cá nhân	✓
3	Sinh viên đặt lịch thiết bị hợp lệ	Đơn lưu trạng thái "pending"	✓
4	Sinh viên đặt lịch trùng thời gian	Hệ thống báo lỗi	✓
5	Giảng viên đăng nhập vào hệ thống	Truy cập dashboard quản trị	✓
6	Giảng viên duyệt đơn đặt lịch	Trạng thái đổi sang "approved", email gửi	✓
7	Giảng viên từ chối đơn đặt lịch	Trạng thái đổi sang "rejected", email gửi	✓
8	Sinh viên xem danh sách đơn cá nhân	Hiển thị đầy đủ trạng thái từng đơn	✓

4.4.3 Đánh giá kết quả thử nghiệm

Hệ thống hoạt động ổn định với 10–20 người dùng thử.

Email thông báo gửi thành công khi phê duyệt/từ chối.

Không phát hiện lỗi nghiêm trọng trong quy trình.

Giao diện hoạt động tốt trên desktop và mobile.

Chương 5

Ứng dụng AI và Kết luận

5.1 Ứng dụng AI trong hệ thống quản lý đặt lịch thiết bị

Để nâng cao hiệu quả quản lý và trải nghiệm người dùng, nhóm nghiên cứu đề xuất tích hợp một số ứng dụng AI vào hệ thống trong tương lai:

5.1.1 Dự đoán nhu cầu sử dụng thiết bị (AI Forecasting)

Thu thập dữ liệu lịch sử đặt thiết bị (loại thiết bị, khung giờ, ngày trong tuần, đối tượng sử dụng).

Sử dụng mô hình học máy như Random Forest hoặc LSTM để dự đoán thời gian nào thiết bị sẽ được đặt nhiều nhất.

Giúp giảng viên và quản trị viên:

- Bố trí lịch hợp lý.
- Chủ động bảo trì, nâng cấp thiết bị.

5.1.2 Phân tích hành vi người dùng (User Behavior Analytics)

Sử dụng AI để phân tích tần suất, loại thiết bị mà từng nhóm sinh viên hay đăng ký.

Từ đó hệ thống gợi ý nhanh cho sinh viên những thiết bị thường sử dụng.

Giảm thời gian thao tác và nâng cao trải nghiệm người dùng.

5.1.3 Hệ thống gợi ý (Recommendation System)

Áp dụng AI Recommendation Engine tương tự như Netflix/Amazon.

Khi sinh viên chọn một thiết bị, hệ thống gợi ý thêm:

- Các thiết bị liên quan (ví dụ: khi chọn máy chiếu → gợi ý thêm loa hoặc micro).
- Khung giờ trống phù hợp nhất.

Giúp tăng tính thông minh và linh hoạt trong việc quản lý lịch.

5.1.4 Chatbot hỗ trợ (AI-powered Chatbot)

Tích hợp chatbot dựa trên NLP (xử lý ngôn ngữ tự nhiên).

Chức năng:

- Trả lời câu hỏi thường gặp: “Làm sao để đặt lịch?”, “Thiết bị X còn trống không?”.
- Hỗ trợ sinh viên thực hiện thao tác đặt lịch trực tiếp qua chatbot.

Công nghệ gợi ý: Rasa, Dialogflow, GPT API.

5.1.5 Phát hiện và cảnh báo bất thường (Anomaly Detection)

AI theo dõi các mẫu đặt lịch bất thường, ví dụ:

- Một sinh viên đặt nhiều thiết bị cùng lúc.
- Đặt liên tục trong thời gian dài không hợp lý.

Khi phát hiện bất thường, hệ thống gửi cảnh báo đến quản trị viên.

Góp phần nâng cao tính an toàn và công bằng trong quản lý thiết bị.

5.2 Lợi ích khi áp dụng AI

Việc áp dụng AI sẽ mang lại những giá trị thiết thực sau:

- Tối ưu hóa tài nguyên: Phân bổ thiết bị hợp lý, tránh lãng phí hoặc quá tải.
- Tăng trải nghiệm người dùng: Sinh viên đặt lịch nhanh chóng nhờ gợi ý thông minh.
- Tiết kiệm thời gian cho giảng viên: Giảm bớt thao tác duyệt, nhờ AI gợi ý và cảnh báo.
- Nâng cao chất lượng quản lý: Dự báo, phân tích, phát hiện gian lận kịp thời.

- Hướng tới tự động hóa: Tích hợp chatbot, giảm phụ thuộc vào con người trong hỗ trợ thường xuyên.

5.3 Hạn chế của hệ thống hiện tại

Mặc dù hệ thống đã hoạt động ổn định, nhưng vẫn còn một số hạn chế:

- Chưa áp dụng AI, việc gợi ý và duyệt lịch còn hoàn toàn thủ công.
- Chưa hỗ trợ tính năng tìm kiếm, thống kê trực quan (biểu đồ).
- Bảo mật ở mức cơ bản (xác thực bằng username/password, chưa có xác thực đa lớp).
- Giao diện mới dừng ở mức cơ bản, cần cải tiến UX/UI thân thiện hơn.

5.4 Định hướng phát triển

Trong thời gian tới, hệ thống có thể phát triển theo các hướng sau:

- Tích hợp AI để gợi ý và dự đoán nhu cầu.
- Mở rộng quy mô sang quản lý phòng học, phòng lab, hội trường.
- Cải thiện bảo mật với xác thực hai lớp (2FA) hoặc đăng nhập bằng SSO (Google/Microsoft).
- Ứng dụng Cloud Computing để triển khai hệ thống trên AWS/Azure, đảm bảo khả năng mở rộng.
- Xây dựng API mở cho phép tích hợp với các hệ thống khác của trường (quản lý đào tạo, điểm danh, thư viện).

5.5 Kết luận chung

Đề tài “Xây dựng hệ thống quản lý đặt lịch thiết bị” đã đạt được các kết quả sau:

- Phân tích, thiết kế hệ thống bằng các sơ đồ UML.
- Xây dựng và triển khai thử nghiệm bằng Django + Bootstrap.
- Hệ thống cho phép sinh viên đặt lịch, giảng viên duyệt lịch, và quản lý trạng thái dễ dàng.
- Quá trình kiểm thử cho thấy hệ thống hoạt động ổn định, đáp ứng yêu cầu chức năng.

Hướng phát triển trong tương lai là tích hợp AI để nâng cao tính thông minh và hiệu quả quản lý. Điều này không chỉ giúp trường đại học tiết kiệm nguồn lực, mà còn mang lại trải nghiệm tốt hơn cho sinh viên, giảng viên và nhà quản lý.

Tài liệu tham khảo