

ĐẠI HỌC PHENIKAA

TRƯỜNG CÔNG NGỆ THÔNG TIN

PHÂN TÍCH THIẾT KẾ PHẦN MỀM

(N05)

“Dự án: Hệ thống Quản lý đăng kí học theo tín chỉ”

Nhóm 13

|  |  |
| --- | --- |
| Đỗ Hoài Nam | [22014002@st.phenikaa-uni.edu.vn](mailto:22010392@st.phenikaa-uni.edu.vn) |

Giảng viên hướng dẫn: ThS. Vũ Quang Dũng, ThS. Nguyễn Xuân Quế

Nội dung

[**Lời cảm ơn** 3](#_Toc212209965)

[**Chương 1. Giới thiệu đề tài** 4](#_Toc212209966)

[1.1. Mục tiêu 4](#_Toc212209967)

[1.1.1. Mục tiêu tổng quát 4](#_Toc212209968)

[1.1.2. Mục tiêu chức năng 4](#_Toc212209969)

[1.1.3. Mục tiêu phi chức năng 5](#_Toc212209970)

[1.2. Phạm vi dự án (Rút gọn) 5](#_Toc212209971)

[1.2.1. Phạm vi chức năng 5](#_Toc212209972)

[1.2.2. Phạm vi dữ liệu 6](#_Toc212209973)

[1.2.3. Triển khai & môi trường 6](#_Toc212209974)

[1.2.4. Người dùng & bên liên quan 6](#_Toc212209975)

[1.3. Mô hình quy trình đã chọn và lý do 6](#_Toc212209976)

[1.3.1. Mô hình quy trình 6](#_Toc212209977)

[1.3.2. Lý do lựa chọn 7](#_Toc212209978)

[**Chương 2. Phân tích yêu cầu** 8](#_Toc212209979)

[2.1. Mô tả dự án (tóm tắt) 8](#_Toc212209980)

[2.1.1. Các use case Diagram 9](#_Toc212209981)

[2.1.2. Mô tả actors 9](#_Toc212209982)

[2.2 Yêu cầu chức năng 10](#_Toc212209983)

[2.2.1. Danh mục Use Case 10](#_Toc212209984)

[2.2.2 Đặc tả chi tiết cho từng Use Case 11](#_Toc212209985)

[**Chương 3. Thiết kế hệ thống** 17](#_Toc212209986)

[3.1. Thiết kế kiến trúc 17](#_Toc212209987)

[3.1.1. Lựa chọn mô hình kiến trúc 17](#_Toc212209988)

[3.1.2. Mô hình triển khai (gợi ý) 17](#_Toc212209989)

[3.2. Thiết kế chi tiết (UML) 17](#_Toc212209990)

[3.2.1. Sơ đồ lớp (Class Diagram) 17](#_Toc212209991)

[3.2.2. Sơ đồ trình tự (Sequence Diagram) 19](#_Toc212209992)

[3.2.3. Biểu đồ Trạng thái (State Machine) 21](#_Toc212209993)

[3.2.4. Thuật toán & ràng buộc nghiệp vụ 22](#_Toc212209994)

[3.2.5. Bảo mật & nhật ký 22](#_Toc212209995)

[**Chương 4. Thiết kế giao diện** 23](#_Toc212209996)

[4.1. Mục tiêu & phạm vi 23](#_Toc212209997)

[4.1.1. Mục tiêu UX (Trải nghiệm người dùng) 23](#_Toc212209998)

[4.1.2. Mục tiêu UI (Giao diện) 23](#_Toc212209999)

[4.2. Giao diện chương trình 23](#_Toc212210000)

[**Chương 5. Tổng kết và hướng phát triển** 27](#_Toc212210001)

[**Chương 6. Tài liệu tham khảo** 28](#_Toc212210002)

**Lời cảm ơn**

Đối với bất kì bất kì sinh viên nào thì bài tập lớn kết thúc môn luôn là thành quả quan trọng của một môn học. Và để bài tập lớn này được đạt kết quả tốt như hiện nay chúng em đã nhận được rất nhiều sự hỗ trợ, hướng dẫn vô cùng tận tâm và nhiệt huyết của ThS. Vũ Quang Dũng, ThS. Nguyễn Xuân Quế. Xuất phát từ sự chân thành, chúng em xin bày tỏ sự biết ơn sâu sắc nhất đến các thầy.

Qua những bài giảng vô cùng thú vị ở lớp, những tiết học cuốn hút, các thầy đã không ngừng tạo những điều kiện tốt để chúng em có được môi trường học tập tốt và là người trang bị cho chúng em tất cả những kiến thức, kỹ năng cần có để chúng em có thể hoàn thành tốt đồ án này.

Trong quá trình học tập và nghiên cứu, do bản thân chúng em vẫn còn chưa vững kiến thức về chuyên ngành và kinh nghiệm thực tế nên có nhiều điều chúng em còn thiếu sót và cần học hỏi nhiều hơn nữa.

Chúng em rất cảm ơn các thầy vẫn tận tâm, tỉ mỉ, dành nhiều thời gian để hướng dẫn, chỉ bảo cho chúng em, giúp chúng em ngày một hoàn thiện bản thân mình hơn. Chúng em kính chúc các thầy luôn luôn mạnh khỏe.

Chúng em xin chân thành cảm ơn!

**Chương 1. Giới thiệu đề tài**

1.1. Mục tiêu

Bản phân tích và thiết kế này mô tả chi tiết Hệ thống Quản lý Đăng ký Tín chỉ Học phần (gọi tắt là “Hệ thống”), bao gồm mục tiêu, phạm vi, nhóm người dùng chính, các chức năng cốt lõi và các tiêu chí chất lượng cần đạt. Hệ thống hướng tới chuẩn hóa và tự động hóa toàn bộ quy trình đăng ký học phần theo học chế tín chỉ, đảm bảo công bằng – minh bạch – chính xác – tiện dụng cho sinh viên, giảng viên và bộ phận quản lý đào tạo.

Trong bối cảnh số hóa giáo dục, số lượng môn/lớp, ràng buộc (tiên quyết, học trước, song hành, tương đương), cùng các chính sách học vụ ngày càng đa dạng, việc xử lý thủ công dễ sai sót và tốn nguồn lực. Hệ thống được xây dựng để đăng ký – kiểm tra điều kiện – xếp lớp theo thời gian thực, giảm tải cho phòng đào tạo, tối ưu sử dụng phòng học và lịch giảng dạy. Bản cập nhật hợp nhất bổ sung các khả năng đồng bộ kỳ hoạt động, cửa sổ đăng ký theo đợt/mốc ưu tiên, lọc xung đột ca–thứ–phòng–giảng viên, duyệt bởi giảng viên, waitlist tự động, và quản lý mở rộng Trường/Môn/Phòng.

1.1.1. Mục tiêu tổng quát

* Chuẩn hóa quy trình đăng ký tín chỉ theo quy chế học vụ và CTĐT của nhà trường.
* Tự động hóa kiểm tra ràng buộc: tiên quyết, học trước, song hành, tương đương; trùng lịch (ca/thứ/phòng/GV), giới hạn tín chỉ/học kỳ, chống trùng lớp/ca thi, chính sách tái đăng ký và số lần học/thi lại.
* Tối ưu phân bổ nguồn lực: chỉ tiêu lớp, năng lực phòng học, lịch giảng dạy – lịch thi; hạn chế nghẽn trong giờ cao điểm.
* Minh bạch & thời gian thực: cập nhật sĩ số, waitlist và trạng thái đăng ký (pending/approved/rejected/cancelled) cùng thông báo tức thì.
* Vận hành theo kỳ: mọi thao tác bám kỳ đang hoạt động (đồng bộ kỳ), có cửa sổ đăng ký theo đợt/mốc ưu tiên.
* Nâng cao trách nhiệm học thuật: đăng ký lớp do giảng viên phụ trách có bước duyệt/từ chối kèm lý do, bảo đảm chất lượng lớp.

1.1.2. Mục tiêu chức năng

* Quản lý danh mục đào tạo
  + CTĐT/khóa học, học phần, phiên bản CTĐT theo niên khóa.
  + Thiết lập/áp dụng ràng buộc: tiên quyết, song hành, tương đương; chính sách tái đăng ký và khống chế số lần học/thi lại.
* Kế hoạch học kỳ & mở lớp
  + Lập kế hoạch lớp học phần theo kỳ (môn, mã lớp, giảng viên, thời khóa biểu, phòng học, chỉ tiêu, hình thức dạy–học).
  + Lọc xung đột trước khi lưu: tránh trùng ca/thứ/phòng/giảng viên; hỗ trợ điều chỉnh chỉ tiêu, chuyển phòng/đổi lịch có thông báo.
  + Quản lý mở rộng Trường (School)/Môn (Subject)/Phòng (Room) để thống nhất dữ liệu và năng lực.
* Đăng ký học phần cho sinh viên
  + Đăng ký/hủy/đổi lớp; tự vào waitlist khi lớp đủ chỗ; cửa sổ thời gian theo đợt/mốc ưu tiên.
  + Cảnh báo xung đột theo thời gian thực (trùng ca/thứ…), hỗ trợ tự chuyển lớp cùng môn (khi chính sách bật).
  + Theo dõi lịch sử và trạng thái đăng ký; đồng bộ với kỳ đang hoạt động.
* Quản lý lớp học phần cho giảng viên
  + Xem danh sách lớp phụ trách, danh sách SV, điểm danh/xuất danh sách; theo dõi sĩ số và nhóm thực hành/thảo luận.
  + Duyệt/từ chối đăng ký đối với lớp mình phụ trách (bắt buộc lý do khi từ chối).
* Quản trị & báo cáo cho phòng đào tạo
  + Quản trị người dùng/giảng viên, học kỳ (mốc mở/đóng đăng ký), chính sách tín chỉ tối thiểu/tối đa.
  + Hủy lớp cascade cập nhật trạng thái đăng ký (cancelled) và xử lý nâng waitlist phù hợp.
  + Báo cáo sĩ số, năng lực phòng, tải hệ thống; tích hợp dashboard phục vụ ra quyết định.
* Tích hợp – mở rộng
  + Tích hợp SSO/cổng định danh; học phí & hóa đơn (đồng bộ công nợ theo tín chỉ đăng ký); LMS, thư viện.
  + API nội bộ cho đồng bộ & phân tích BI; webhook/thông báo đẩy cho sự kiện đăng ký/duyệt/đổi lịch.

1.1.3. Mục tiêu phi chức năng

* Hiệu năng & khả dụng cao: chịu tải giờ cao điểm (nghìn+ yêu cầu/giây); P95 < 2 giây cho thao tác đăng ký; uptime ≥ 99.9% trong mùa đăng ký.
* Mở rộng tuyến tính: kiến trúc scale-out, cân bằng tải, cache đọc; chống nghẽn qua rate limiting và debounce tìm kiếm.
* Tính nhất quán & toàn vẹn dữ liệu: giao dịch nguyên tử khi ghi đăng ký/cập nhật chỉ tiêu để tránh overbooking; hỗ trợ waitlist FIFO an toàn.
* Bảo mật & riêng tư: xác thực JWT/SSO, RBAC theo vai trò (SV/GV/Admin), audit log cho thao tác nhạy cảm; mã hóa dữ liệu nhạy cảm; chính sách khóa tài khoản khi bất thường.
* Khả năng vận hành: giám sát log có cấu trúc, cảnh báo lỗi 5xx và độ trễ; truy vết sự kiện đăng ký/duyệt/đổi lịch; hỗ trợ rollback và phục hồi sau sự cố.

1.2. Phạm vi dự án (Rút gọn)

1.2.1. Phạm vi chức năng

* Danh mục đào tạo: Quản lý CTĐT theo phiên bản/khóa tuyển, khối kiến thức, chuẩn đầu ra; học phần & nhóm tương đương; ràng buộc học vụ (tiên quyết/học trước/song hành/chống trùng), áp dụng theo ngành/khóa.
* Danh mục nền (mở rộng): Quản lý Trường/School, Môn học/Subject, Phòng/Room (loại phòng, sức chứa) để phục vụ lập lịch & kiểm tra xung đột.
* Kế hoạch & mở lớp: Lập kế hoạch/mở lớp học phần (chỉ tiêu, giảng viên, lịch–phòng), lịch thi tránh trùng.
* Đăng ký theo học kỳ: Quy trình SV với giỏ đăng ký/đổi/hủy, kiểm tra ràng buộc tức thì, danh sách chờ & giữ chỗ, cảnh báo trùng lịch/giới hạn tín chỉ; duyệt bởi GV đối với lớp phụ trách (lý do khi từ chối); cửa sổ đăng ký do Admin cấu hình cho kỳ hoạt động.

1.2.2. Phạm vi dữ liệu

Gồm người dùng (SV/GV/Cán bộ), CTĐT, học phần/nhóm tương đương, ràng buộc, học kỳ, lớp học phần, lịch học–thi, phòng, đăng ký, trường/môn học (danh mục), công nợ/học phí *(nếu áp dụng)*, nhật ký/audit log. Bảo đảm toàn vẹn (khóa ngoại/quan hệ), phiên bản hóa CTĐT, lịch sử thay đổi, kiểm soát đồng thời. Lưu trữ dữ liệu cho các khóa đang học và tối thiểu N năm sau tốt nghiệp.

1.2.3. Triển khai & môi trường

Môi trường DEV – UAT/Staging – PROD (dữ liệu ẩn danh ở UAT); hạ tầng on‑prem hoặc cloud, ưu tiên container, autoscale, HA; ứng dụng web responsive, có thể bổ sung app di động nhẹ; bảo đảm hiệu năng giờ cao điểm với CDN/cache đọc, giới hạn tốc độ (rate‑limit), giám sát & cảnh báo.

1.2.4. Người dùng & bên liên quan

Sinh viên (học chế tín chỉ), Giảng viên (giảng dạy & duyệt), Cán bộ đào tạo/Khoa/Bộ môn (lập kế hoạch, vận hành, báo cáo), bộ phận CNTT/Vận hành (triển khai, giám sát).

1.3. Mô hình quy trình đã chọn và lý do

1.3.1. Mô hình quy trình

Mô hình lặp–tăng trưởng (Iterative & Incremental) dựa trên Agile Scrum, có quản trị rủi ro Spiral nhẹ và kiểm thử kiểu V-Model trong từng vòng lặp.

Cấu trúc chính

* Scrum làm khung quản lý: Sprint 2–3 tuần; vai trò *Product Owner* (đại diện đào tạo), *Scrum Master*, *Dev/QA*; nghi lễ *Planning – Daily – Review – Retro*; *Product Backlog* được ưu tiên theo mốc mở kỳ.
* Phát hành tăng trưởng theo Increment: Mỗi Sprint tạo ra Increment có thể triển khai (ví dụ: quản lý Phòng/Trường/Môn → mở lớp → đăng ký → duyệt GV → báo cáo). Các *feature flags* cho phép bật/tắt dần theo đơn vị phòng/ban.
* Quản trị rủi ro Spiral nhẹ: Trong mỗi sprint thực hiện chu kỳ Identify → Mitigate → Validate cho các rủi ro trọng điểm như: hiệu năng giờ cao điểm, nhất quán giao dịch (tránh overbooking), xung đột lịch/phòng/GV, đồng bộ kỳ hoạt động, tích hợp SSO/học phí (nếu áp dụng).

Chu trình mẫu cho một Sprint

1. Refine yêu cầu (ví dụ: quy tắc cửa sổ đăng ký, lọc trùng GV/phòng/ca/thứ).
2. Thiết kế chi tiết (API, schema, UI trạng thái/waitlist).
3. Phát triển & code review (bảo đảm giao dịch/locking khi cập nhật capacity).
4. Kiểm thử: unit + integration (đăng ký trong/ngoài cửa sổ, duyệt GV có lý do), hiệu năng với dữ liệu giả lập giờ cao điểm.
5. Triển khai UAT cho cán bộ đào tạo/GV thử nghiệm; thu phản hồi.
6. Retro: ghi nhận vấn đề, cập nhật *Definition of Done* (ví dụ bắt buộc audit log cho thao tác hủy lớp).

1.3.2. Lý do lựa chọn

* Biến động nghiệp vụ cao (quy chế học vụ, ràng buộc tiên quyết, cửa sổ đăng ký, chính sách waitlist) ⇒ Agile giúp thích ứng nhanh, cấu hình hóa *rules* và phản hồi sớm từ phòng đào tạo/GV/SV.
* Ràng buộc kỹ thuật phức tạp (ACID ở các thao tác đăng ký, chống overbooking, scale giờ cao điểm) ⇒ lặp–tăng trưởng cho phép *build–measure–learn* với dữ liệu giả lập, giảm rủi ro dồn về cuối.
* Nhiều bên liên quan & phạm vi rộng (SV, GV, phòng đào tạo, tài vụ) ⇒ phát hành theo module (Danh mục → Mở lớp → Đăng ký → Duyệt → Báo cáo) để thu phản hồi thật, giảm sai lệch kỳ vọng.
* Yêu cầu phi chức năng nghiêm ngặt (P95 < 2s, 99.9% uptime mùa đăng ký) ⇒ kiểm thử hiệu năng/độ tin cậy lặp lại trong từng Sprint, trước mỗi mốc phát hành.
* Áp lực thời gian theo học kỳ ⇒ mỗi Increment có giá trị sử dụng sớm (ví dụ bật trước module quản lý lịch/phòng), tránh *big‑bang release* và cho phép cuộn dần dữ liệu/schemas qua *migration* có kiểm soát.

**Chương 2. Phân tích yêu cầu**

2.1. Mô tả dự án (tóm tắt)

Mục tiêu

Xây dựng hệ thống quản lý đăng ký tín chỉ bao quát toàn bộ quy trình học vụ theo học chế tín chỉ: mở lớp → tư vấn lộ trình → đăng ký/huỷ/đổi → kiểm tra tiên quyết & xung đột TKB → tính/chốt học phí (nếu áp dụng) → theo dõi kết quả. Hệ thống kế thừa kiến trúc 3 lớp, kiểm tra ràng buộc theo thời gian thực, đồng thời mở rộng miền dữ liệu và quy tắc vận hành để phù hợp triển khai theo kỳ học đang hoạt động.

Ba đích đến trọng tâm

1. Sinh viên đăng ký nhanh, chính xác, đúng lộ trình và trong cửa sổ đăng ký do nhà trường cấu hình.
2. Giảng viên/Phòng Đào tạo vận hành, kiểm duyệt, tối ưu sĩ số; phát hiện sớm xung đột ca/thứ/phòng/giảng viên.
3. Nhà trường giám sát số liệu theo thời gian thực để ra quyết định (mở thêm/gộp/huỷ lớp, phân bổ phòng, điều chỉnh chỉ tiêu).

Chức năng theo vai trò

Sinh viên

* Tìm & đăng ký học phần theo kỳ/khoa/giảng viên/khung giờ/cơ sở; xem lịch học chi tiết.
* Giỏ đăng ký – huỷ – đổi với kiểm tra tức thì các ràng buộc: tiên quyết, giới hạn tín chỉ, xung đột TKB; cảnh báo đỏ khi trùng ca/thứ.
* Theo dõi trạng thái: *đã đăng ký / chờ duyệt / waitlist / đã huỷ*; có thể tự động chuyển giữa các lớp cùng môn khác mã khi chính sách bật.

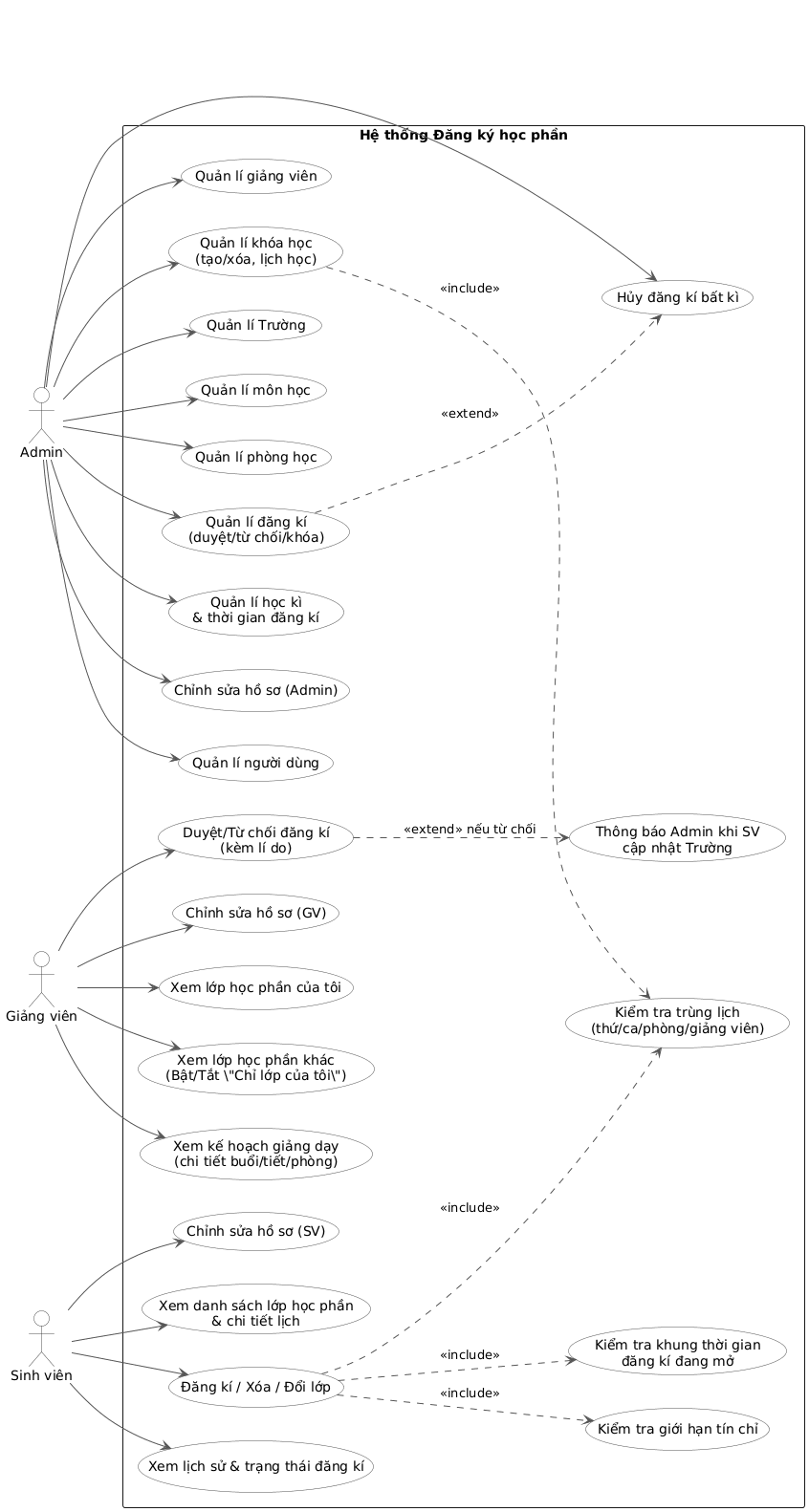
Giảng viên

* Quản lý lớp phụ trách (có toggle “Chỉ hiển thị lớp của tôi”, xem lớp khác ở chế độ đọc).
* Phê duyệt đăng ký của SV vào lớp mình phụ trách; nếu từ chối phải nêu lý do (đồng bộ cho Admin theo dõi).
* Theo dõi nhanh: tình trạng đăng ký, mức độ quan tâm học phần, chỉ dấu trùng lịch phổ biến của SV để phối hợp mở/gộp lớp.

Quản trị viên

* Cấu hình & lập kế hoạch: tạo/cập nhật kỳ học và mốc mở–đóng đăng ký; quản lý CTĐT, học phần, ràng buộc (tiên quyết/song hành/chống trùng); lập chỉ tiêu – lịch – phòng; chính sách waitlist.
* Điều phối & kiểm duyệt: phân công giảng viên; xét duyệt ngoại lệ (vượt trần tín chỉ, học chéo khoa… nếu áp dụng); phát hiện & xử lý xung đột phòng/GV; gộp/huỷ/mở thêm lớp; phát hành thông báo.
* Danh mục nền mở rộng: quản lý Trường (School) / Môn học (Subject) / Phòng (Room); khi tạo/lập lịch lớp học phần hệ thống lọc trùng theo GV/phòng/ca/thứ trước khi lưu; khi huỷ lớp, các đăng ký liên quan được chuyển trạng thái phù hợp (*ví dụ:* cancelled).

2.1.1. Các use case Diagram



*Hình 1. Use case tổng quát hệ thông*

2.1.2. Mô tả actors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Actor | Nhiệm vụ chính | Quyền / Thao tác | Luồng thao tác | Ràng buộc / Quy tắc | Màn hình chính |
| Admin | - Quản lí người dùng, vai trò và phân quyền - Quản lí trường/khoa, môn học, phòng học - Tạo/lập lịch lớp học phần, cấu hình học kì & cửa sổ đăng ký - Giám sát/duyệt/hủy đăng ký, khóa/mở chức năng | - Tạo/Sửa/Xóa: người dùng, giảng viên, môn học, lớp học phần, phòng - Cấu hình học kì, thời gian đăng ký, giới hạn tín chỉ - Duyệt/Từ chối đăng ký; hủy đăng ký bất kỳ khi cần - Khóa/mở đăng ký; xuất báo cáo, nhật ký thao tác | 1) Đăng nhập → 2) Vào Bảng điều khiển quản trị → 3) Thiết lập học kì & cửa sổ đăng ký → 4) Tạo lớp học phần (kiểm tra trùng lịch) → 5) Quản lí đăng ký (duyệt/từ chối/hủy) → 6) Theo dõi báo cáo | - Chỉ Admin mới thay đổi cấu hình hệ thống - Tạo lịch phải không trùng (giảng viên/phòng/thứ-ca) - Mở/đóng cửa sổ đăng ký theo mốc thời gian - Hủy đăng ký cần lý do và ghi log | Bảng điều khiển Admin; Quản lí học kì; Quản lí môn; Quản lí lớp học phần; Quản lí phòng; Duyệt đăng ký; Báo cáo & nhật ký |

2.2 Yêu cầu chức năng

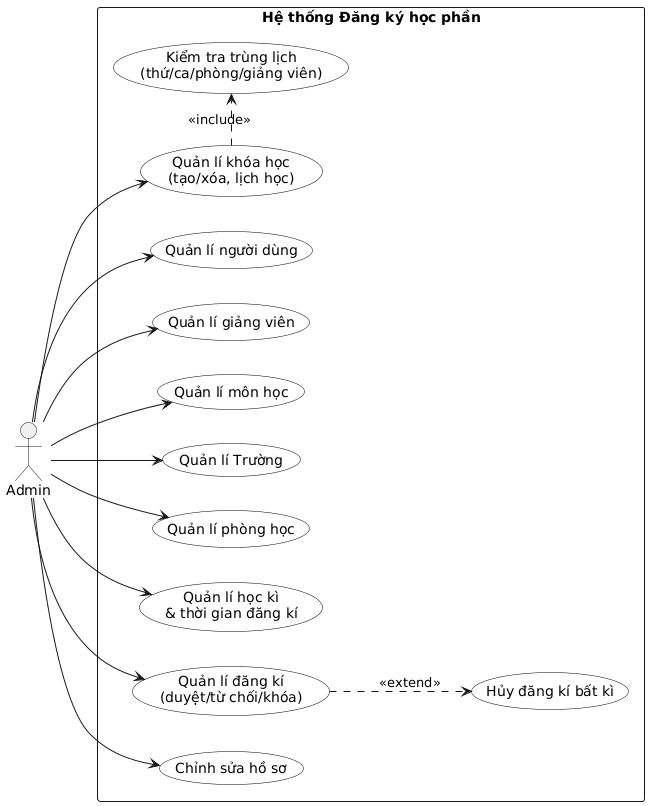
2.2.1. Danh mục Use Case

2.2.1.1. Use case admin

|  |  |  |  |
| --- | --- | --- | --- |
| Mã | Use case | Mô tả ngắn | Ghi chú |
| A1 | Quản lí người dùng | Tạo/sửa/xóa, đặt lại mật khẩu, phân quyền. | — |
| A2 | Quản lí giảng viên | Thêm/sửa thông tin GV, gán khoa/bộ môn. | — |
| A3 | Quản lí khóa học (lớp HP) | Tạo/xóa lớp học phần, cấu hình lịch/tiết/phòng. | «include» Kiểm tra trùng lịch |
| A4 | Quản lí Trường | Thêm/sửa thông tin trường/khoa/bộ môn. | — |
| A5 | Quản lí môn học | Tạo/sửa mã môn, số TC, điều kiện tiên quyết. | — |
| A6 | Quản lí phòng học | Thêm/sửa/xóa phòng, sức chứa, thiết bị. | — |
| A7 | Quản lí đăng kí | Duyệt/từ chối, khóa/mở đăng kí, xử lý khiếu nại. | «extend» Hủy đăng kí bất kì |
| A8 | Quản lí học kì & thời gian | Tạo học kì, cấu hình cửa sổ đăng kí. | — |
| A9 | Hủy đăng kí bất kì | Gỡ đăng kí của SV trong trường hợp đặc biệt. | Được «extend» từ A7 |
| A10 | Chỉnh sửa hồ sơ Admin | Cập nhật thông tin cá nhân, đổi mật khẩu. | — |

2.2.2 Đặc tả chi tiết cho từng Use Case

NHÓM ADMIN



*Hình 2. Use case admin*

A1. Quản lí người dùng

1. Mục tiêu: Tạo/sửa/xóa tài khoản, đặt lại mật khẩu, gán vai trò (Admin/Giảng viên/Sinh viên).
2. Tác nhân: Chính: Admin. Phụ: Hệ thống xác thực/LDAP (nếu có).
3. Tiền điều kiện: Admin đã đăng nhập; có quyền USER\_ADMIN.
4. Kích hoạt: Admin mở màn hình “Người dùng”.
5. Luồng chính:
   1. Admin tìm kiếm (theo email/mã định danh).
   2. Admin tạo mới hoặc chọn người dùng để chỉnh sửa.
   3. Nhập/thay đổi: họ tên, email, mssv/mscb, vai trò, trạng thái.
   4. Lưu; hệ thống kiểm tra trùng email/mã.
   5. (Tuỳ chọn) Gửi email kích hoạt/đặt lại mật khẩu.
6. Ngoại lệ/nhánh:
   1. E1: Email/mã trùng → hiển thị lỗi, yêu cầu đổi.
   2. E2: Không đủ quyền thay đổi vai trò → chặn và log.
   3. E3: Gửi email thất bại → vẫn lưu hồ sơ, hiển thị cảnh báo.
7. Hậu điều kiện: Hồ sơ và vai trò cập nhật nhất quán; audit log ghi nhận.
8. Quy tắc nghiệp vụ: Email duy nhất; mỗi tài khoản ≥ 1 vai trò; trạng thái ACTIVE/LOCKED.
9. Phi chức năng: Ghi log bảo mật; thời gian phản hồi < 2s cho tìm kiếm 10k bản ghi.

A2. Quản lí giảng viên

1. Mục tiêu: Quản lý hồ sơ GV, gán khoa/bộ môn, liên kết tài khoản.
2. Tác nhân: Admin (chính); Hệ thống (phụ).
3. Tiền điều kiện: Có danh mục Khoa/Bộ môn.
4. Kích hoạt: Mở màn hình “Giảng viên”.
5. Luồng chính:
   1. Tạo/sửa hồ sơ GV (mã, tên, học vị, liên hệ).
   2. Gán Khoa/Bộ môn, lĩnh vực.
   3. (Tuỳ chọn) Liên kết tài khoản người dùng.
6. Ngoại lệ: E1: Mã GV trùng; E2: Khoa/Bộ môn không tồn tại; E3: Liên kết tài khoản đã liên kết nơi khác.
7. Hậu điều kiện: Hồ sơ GV hợp lệ, sẵn sàng phân công lớp.
8. Quy tắc: Một GV thuộc ≥1 Bộ môn; mã GV duy nhất.
9. Phi chức năng: Import CSV tối đa 5k dòng/lượt.

A3. Quản lí khóa học (lớp học phần)

1. Mục tiêu: Tạo/xoá lớp học phần (LHP), cấu hình lịch/tiết/phòng, phân công GV.
2. Tác nhân: Admin (chính); GV (tham chiếu); Use case dùng chung: «include» C1 Kiểm tra trùng lịch.
3. Tiền điều kiện: Có danh mục Môn học, Phòng, Học kỳ còn hiệu lực.
4. Kích hoạt: Mở “Lớp học phần”.
5. Luồng chính:
   1. Chọn Môn học, Học kỳ; tạo LHP (mã LHP, sĩ số).
   2. Chọn lịch dạy (thứ/tiết/tuần), phòng học.
   3. Phân công giảng viên.
   4. Gọi C1 để kiểm tra trùng (phòng/GV/tiết).
   5. Lưu & công bố (trạng thái OPEN).
6. Ngoại lệ: E1: Trùng lịch/phòng/GV → đề xuất slot thay thế; E2: Phòng không đủ sức chứa; E3: Học kỳ đã khoá.
7. Hậu điều kiện: LHP ở trạng thái OPEN hoặc DRAFT nếu chưa hợp lệ.
8. Quy tắc: Sĩ số ≤ sức chứa phòng; 1 GV không dạy 2 LHP trùng tiết.
9. Phi chức năng: Tính trùng lịch trong < 1s/ thao tác.

A4. Quản lí Trường/Khoa/Bộ môn

1. Mục tiêu: Danh mục tổ chức.
2. Tác nhân: Admin.
3. Tiền điều kiện: —
4. Kích hoạt: Mở “Danh mục tổ chức”.
5. Luồng chính: Tạo/sửa/xóa Trường, Khoa, Bộ môn; ràng buộc cha–con.
6. Ngoại lệ: E1: Xoá khi đang được tham chiếu → từ chối.
7. Hậu điều kiện: Danh mục nhất quán.
8. Quy tắc: Không được xoá mục đang có GV/Môn/LHP dùng.
9. Phi chức năng: Kiểm tra ràng buộc khoá ngoại.

A5. Quản lí môn học

1. Mục tiêu: Danh mục môn, số tín chỉ, điều kiện tiên quyết.
2. Tác nhân: Admin.
3. Tiền điều kiện: —
4. Kích hoạt: “Môn học”.
5. Luồng chính: Tạo/sửa mã môn, tên, TC, học phần tiên quyết.
6. Ngoại lệ: E1: Mã môn trùng; E2: Vòng lặp tiên quyết.
7. Hậu điều kiện: Môn dùng được khi tạo LHP.
8. Quy tắc: Không cho phép chu trình trong đồ thị tiên quyết.
9. Phi chức năng: Validate theo thời gian thực.

A6. Quản lí phòng học

1. Mục tiêu: Danh mục phòng, sức chứa, thiết bị.
2. Tác nhân: Admin.
3. Tiền điều kiện: —
4. Kích hoạt: “Phòng học”.
5. Luồng chính: Tạo/sửa/xoá phòng; gán sức chứa, khu nhà.
6. Ngoại lệ: E1: Xoá phòng đang có lịch → từ chối/đề xuất chuyển.
7. Hậu điều kiện: Sẵn sàng lập lịch.
8. Quy tắc: Sức chứa > 0; tên phòng duy nhất trong cùng cơ sở.
9. Phi chức năng: Tìm kiếm tức thời.

A7. Quản lí đăng kí

1. Mục tiêu: Duyệt/từ chối/khóa đăng kí; xử lý khiếu nại.
2. Tác nhân: Admin (chính); GV (tham gia); «extend» A9 Hủy đăng kí bất kì.
3. Tiền điều kiện: Cửa sổ học kỳ tồn tại.
4. Kích hoạt: “Quản lí đăng kí”.
5. Luồng chính:
   1. Xem hàng đợi yêu cầu SV (đăng kí/đổi/xoá).
   2. Bộ lọc theo LHP/GV/SV/Trạng thái.
   3. Chọn yêu cầu → duyệt hoặc từ chối (ghi lý do).
   4. Khoá/mở đăng kí theo LHP hoặc toàn kỳ.
6. Ngoại lệ: E1: LHP đầy; E2: Hết hạn cửa sổ; E3: Đổi lớp vượt tín chỉ.
7. Hậu điều kiện: Trạng thái yêu cầu cập nhật; SV nhận thông báo.
8. Quy tắc: Quyết định Admin ghi nhận không thể sửa, chỉ hoàn tác bằng A9.
9. Phi chức năng: Giao diện xử lý 1000+ yêu cầu/giờ.

A8. Quản lí học kì & thời gian đăng kí

1. Mục tiêu: Tạo học kì, cấu hình registration window.
2. Tác nhân: Admin.
3. Tiền điều kiện: Lịch học tập trung.
4. Kích hoạt: “Học kì”.
5. Luồng chính: Tạo học kì (mã, ngày bắt đầu/kết thúc); đặt khung thời gian mở/đóng đăng kí theo đợt/nhóm.
6. Ngoại lệ: E1: Chồng chéo cửa sổ; E2: Trùng mã kỳ.
7. Hậu điều kiện: Cấu hình có hiệu lực cho C3.
8. Quy tắc: Thời gian mở < thời gian đóng; có thể nhiều đợt.
9. Phi chức năng: Đồng bộ giờ hệ thống.

A9. Hủy đăng kí bất kì (Admin override)

1. Mục tiêu: Gỡ đăng kí của SV trong tình huống đặc biệt.
2. Tác nhân: Admin.
3. Tiền điều kiện: Yêu cầu hợp lệ; lý do được ghi.
4. Kích hoạt: Từ A7 hoặc màn hình chi tiết SV/LHP.
5. Luồng chính: Chọn bản ghi đăng kí → nhập lý do → xác nhận → cập nhật trạng thái.
6. Ngoại lệ: E1: Hết hạn học kỳ (bị khoá điểm) → yêu cầu quy trình học vụ.
7. Hậu điều kiện: Đăng kí bị huỷ; chỗ trống trả về LHP.
8. Quy tắc: Ghi audit bắt buộc; thông báo GV & SV.
9. Phi chức năng: Hoàn tất < 1s.

A10. Chỉnh sửa hồ sơ Admin

1. Mục tiêu: Cập nhật thông tin cá nhân, đổi mật khẩu.
2. Tác nhân: Admin.
3. Tiền điều kiện: Đã đăng nhập.
4. Kích hoạt: “Hồ sơ của tôi”.
5. Luồng chính: Chỉnh sửa thông tin; đổi mật khẩu (yêu cầu mật khẩu cũ);
6. Ngoại lệ: E1: Mật khẩu cũ sai; E2: Không đạt chính sách mật khẩu.
7. Hậu điều kiện: Hồ sơ cập nhật; phiên tiếp tục hợp lệ.
8. Quy tắc: Mật khẩu ≥ 8 ký tự, gồm chữ hoa, thường, số.
9. Phi chức năng: 2FA (nếu bật), mã hoá mạnh.

2.3. Yêu cầu phi chức năng

2.3.1. Hiệu năng

* Tìm kiếm/lọc phía client: Nhiều danh sách (người dùng, giảng viên, lớp học phần, đăng ký) hỗ trợ tìm kiếm theo tên/email/role và lọc theo trạng thái/kỳ/giảng viên ngay trên dữ liệu đã tải — giúp thao tác nhanh.

Báo cáo về những chức năng vào …

* Phản hồi kiểm tra tức thời khi nhập liệu/lựa chọn: Khi tạo/sửa lịch học phần, hệ thống kiểm soát trùng ca học/thứ, giảng viên, phòng để giảm lỗi ngay từ bước nhập.

Báo cáo về những chức năng vào …

* Tối ưu thao tác đăng ký: Nếu sinh viên đăng ký cùng môn nhưng khác mã lớp, hệ thống tự chuyển sang lớp mới thay vì giữ song song nhiều lớp — giảm số lần gọi/ghi thay đổi.

Báo cáo về những chức năng vào …

* Giới hạn hiện tại: Chưa thấy cơ chế phân trang (pagination), debounce cho ô tìm kiếm, cache/prefetch hay mục tiêu SLA cụ thể (vd. “< 2s”).

2.3.2. Bảo mật & phân quyền

* Đăng nhập & đổi mật khẩu: Có trang đăng nhập; người dùng có thể đổi mật khẩu trong trang hồ sơ.

Báo cáo về những chức năng vào …

* Phân vai rõ ràng (RBAC) trong UI:
  + Admin: quản lý người dùng (vô hiệu hóa/mở khóa), giảng viên, trường, môn, khóa học/phòng học, học kỳ (thiết lập khung thời gian đăng ký), quản lý/huỷ đăng ký.

Báo cáo về những chức năng vào …

* + Giảng viên: xem lớp của mình (có nút “Chỉ hiển thị lớp của tôi”), duyệt/từ chối đăng ký của lớp mình (khi từ chối phải nhập lý do để gửi admin xem xét).

Báo cáo về những chức năng vào …

* + Sinh viên: xem danh sách lớp học phần, đăng ký/hủy/đổi lớp, xem lịch sử & trạng thái đăng ký; học kỳ trong hồ sơ đồng bộ với kỳ đang hoạt động.

Báo cáo về những chức năng vào …

* Ràng buộc truy cập theo nghiệp vụ:
  + Chặn đăng ký ngoài khung thời gian do Admin cấu hình;
  + Admin có quyền hủy bất kỳ đăng ký trước khi giảng viên can thiệp;
  + Với lớp không thuộc quyền giảng dạy, giảng viên chỉ xem trạng thái (không duyệt).

Báo cáo về những chức năng vào …

* Giới hạn hiện tại: Chưa thấy chặn theo vai trò ở tầng route hoặc khẳng định kiểm soát phân quyền phía server cho mọi đường gọi (UI chủ yếu ẩn/hiện và ràng buộc theo luồng nghiệp vụ).

2.3.3. Khả năng sử dụng (UI/UX)

* Cấu trúc thao tác rõ ràng: Form tạo/sửa khóa học (chọn môn, mã lớp, sĩ số, kỳ, giảng viên, lịch học: thứ/ca/phòng), quản lý trường/môn/phòng/học kỳ có luồng nhập liệu nhất quán.

Báo cáo về những chức năng vào …

* Phản hồi & cảnh báo:
  + Cảnh báo trùng lịch (ca/thứ) khi sinh viên đăng ký;
  + Yêu cầu lý do khi giảng viên từ chối;
  + Khi Admin xóa lớp, các đăng ký liên quan (kể cả lịch sử) tự chuyển trạng thái cancelled — người dùng dễ hiểu hệ quả thao tác.

Báo cáo về những chức năng vào …

* Tính tiện dụng khi duyệt/xem:
  + Nút “Chỉ hiển thị lớp của tôi” giúp giảng viên lọc nhanh lớp được phân công;
  + Danh sách có tìm kiếm/lọc theo vai trò/giảng viên/trạng thái; lớp của giảng viên được đánh dấu để dễ nhận biết.

Báo cáo về những chức năng vào …

* Quản lý hồ sơ cá nhân: Người dùng chỉnh sửa thông tin (họ tên, email, điện thoại, ngày sinh, giới tính, địa chỉ…) và đổi mật khẩu trong một nơi tập trung.

Báo cáo về những chức năng vào …

* Giới hạn hiện tại: Chưa thấy thuộc tính ARIA, phím tắt, hay hướng dẫn accessibility chuyên sâu.

Nguồn đối chiếu chức năng & luồng nghiệp vụ:

**Chương 3. Thiết kế hệ thống**

Tài liệu này mô tả thiết kế chương 3 tập trung cho vai trò Quản trị viên (Admin) của hệ thống quản lý đăng ký học phần. Nội dung kế thừa kiến trúc N‑Tier ở bản cũ và cập nhật theo tính năng của mã nguồn mới.

## 3.1. Thiết kế kiến trúc

### 3.1.1. Lựa chọn mô hình kiến trúc

* UI (React SPA): trang quản trị dành riêng cho Admin với các phân hệ: người dùng, giảng viên, học kỳ, môn học, trường, phòng học, lớp học phần, phê duyệt đăng ký, thông báo.
* Service/API (Node.js + Express): các API REST kèm middleware JWT + RBAC (role: admin, lecturer, student).
* Data (MongoDB + Mongoose): các collection: users, lecturers, schools, subjects, classrooms, courses (lớp học phần), semesters, registrations, notifications.

Vì sao phù hợp

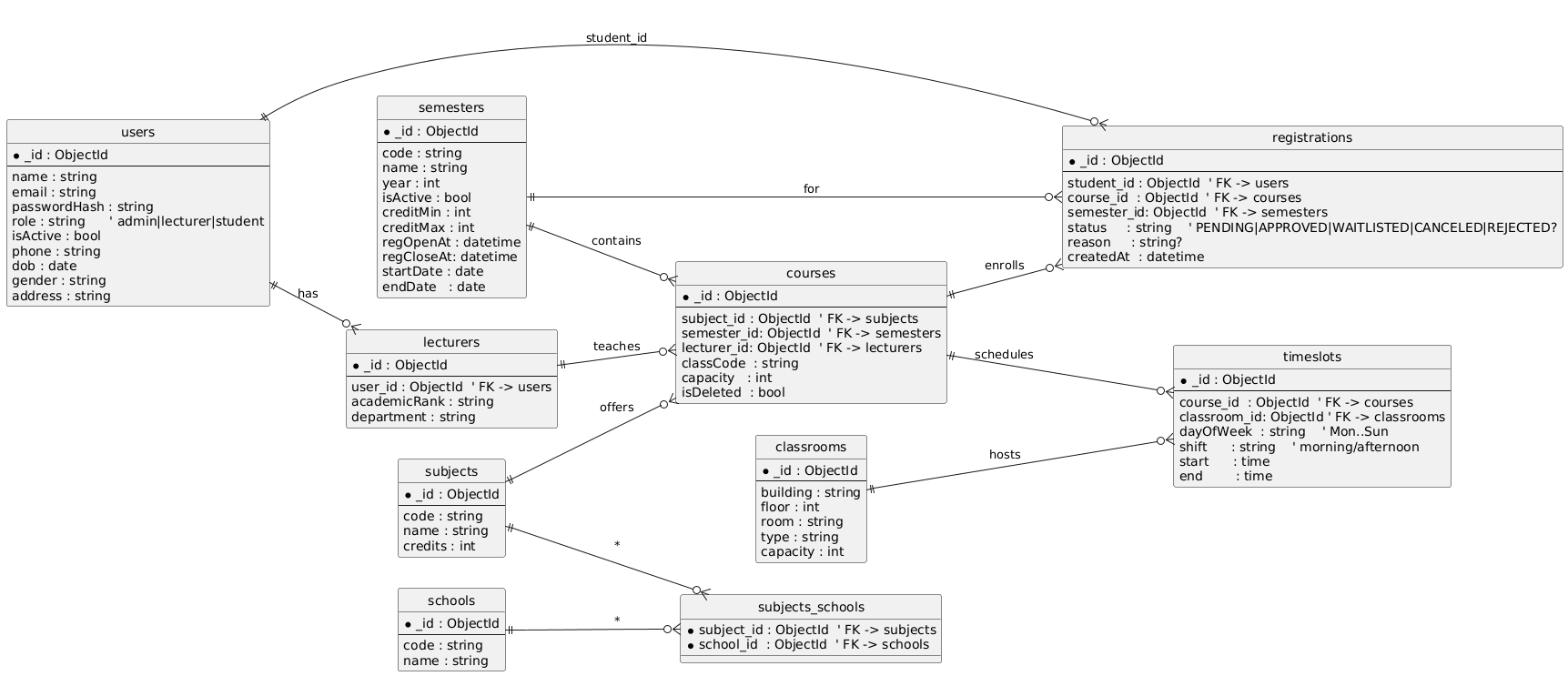
* Tách biệt rõ UI/Service/Data giúp mở rộng từng lớp độc lập.
* Đáp ứng tốt nghiệp vụ CRUD lớn và các ràng buộc nghiệp vụ khi nhập lịch học (lọc trùng ca, phòng, giảng viên).
* Dễ áp dụng RBAC: Admin có toàn quyền cấu hình học kỳ, quản lý danh mục và điều phối đăng ký.

### 3.1.2. Mô hình triển khai (gợi ý)

* Frontend: build tĩnh, phục vụ qua Nginx/CDN; cấu hình .env cho API base URL.
* Backend: Node.js chạy qua PM2/Docker sau Nginx reverse proxy (TLS).
* DB: MongoDB Replica Set/Atlas; snapshot/backup theo lịch.
* Biến môi trường: JWT\_SECRET, MONGO\_URI, REGISTRATION\_WINDOW, MAX\_CREDIT\_{MIN,MAX}.

## 3.2. Thiết kế chi tiết (UML)

### 3.2.1. Sơ đồ lớp (Class Diagram)



Bên dưới là danh sách lớp và thuộc tính/hành vi chính.

3.2.1.1. Domain entities

* User: \_id, name, email, passwordHash, role, isActive, phone, dob, gender, address.
* Lecturer: \_id, userId→User, academicRank, department.
* School: \_id, code, name.
* Subject: \_id, code, name, credits, schools:School[].
* Classroom: \_id, building, floor, room, type, capacity.
* Semester: \_id, code, name, year, isActive, creditMin, creditMax, regOpenAt, regCloseAt, startDate, endDate.
* Course (Lớp học phần): \_id, subject:Subject, classCode, capacity, semester:Semester, lecturer:Lecturer, schedule:TimeSlot[], isDeleted.
* TimeSlot (value object): dayOfWeek, shift (sáng/chiều), start, end, room:Classroom.
* Registration: \_id, student:User, course:Course, semester:Semester, status (PENDING/APPROVED/WAITLISTED/CANCELED), reason?, createdAt.

3.2.1.2. Service layer

* AuthService: login(), issueToken(), verify().
* AdminUserService: list/create/update/disable/enable, search by role/name/email.
* LecturerService: create/update/resetPassword/delete.
* SchoolService: CRUD.
* SubjectService: CRUD + gán School.
* ClassroomService: CRUD + kiểm tra trùng phòng/slot.
* SemesterService: create/activate/deactivate/close/update hạn đăng ký + cấu hình tín chỉ.
* CourseService: create/update/delete; kiểm tra xung đột (giảng viên/ca/phòng/học kỳ), tính seats còn trống.
* RegistrationModerationService: approve/reject (lưu lý do), cancel(anytime), promoteWaitlist(courseId) FIFO.
* NotificationService: gửi thông báo đến admin khi SV đề nghị cập nhật Trường, thông báo trạng thái đăng ký.

3.2.1.3. DTO/Contracts

* UserPublicDTO: id, name, email, role, isActive.
* CourseDTO (Admin view): id, subject{code,name,credits}, classCode, lecturer{name,email}, semester{code}, capacity, schedule[], seatsLeft, isDeleted.
* RegistrationModerationDTO: id, student{name,email}, course{subject, classCode}, status, reason, createdAt.

3.2.1.4. Quan hệ chính (khi vẽ UML dùng association/aggregation):

* Course → Subject/Semester/Lecturer/TimeSlot[].
* Registration → User(student)/Course/Semester.
* Subject ↔ School (nhiều‑nhiều).
* Classroom được tham chiếu trong Course.schedule[].

3.2.1.5. Ghi chú khi vẽ biểu đồ lớp: hiển thị thuộc tính + kiểu, phương thức dịch vụ ở lớp Service; phân hệ UI/Service/Repository tách gói.

### 3.2.2. Sơ đồ trình tự (Sequence Diagram)

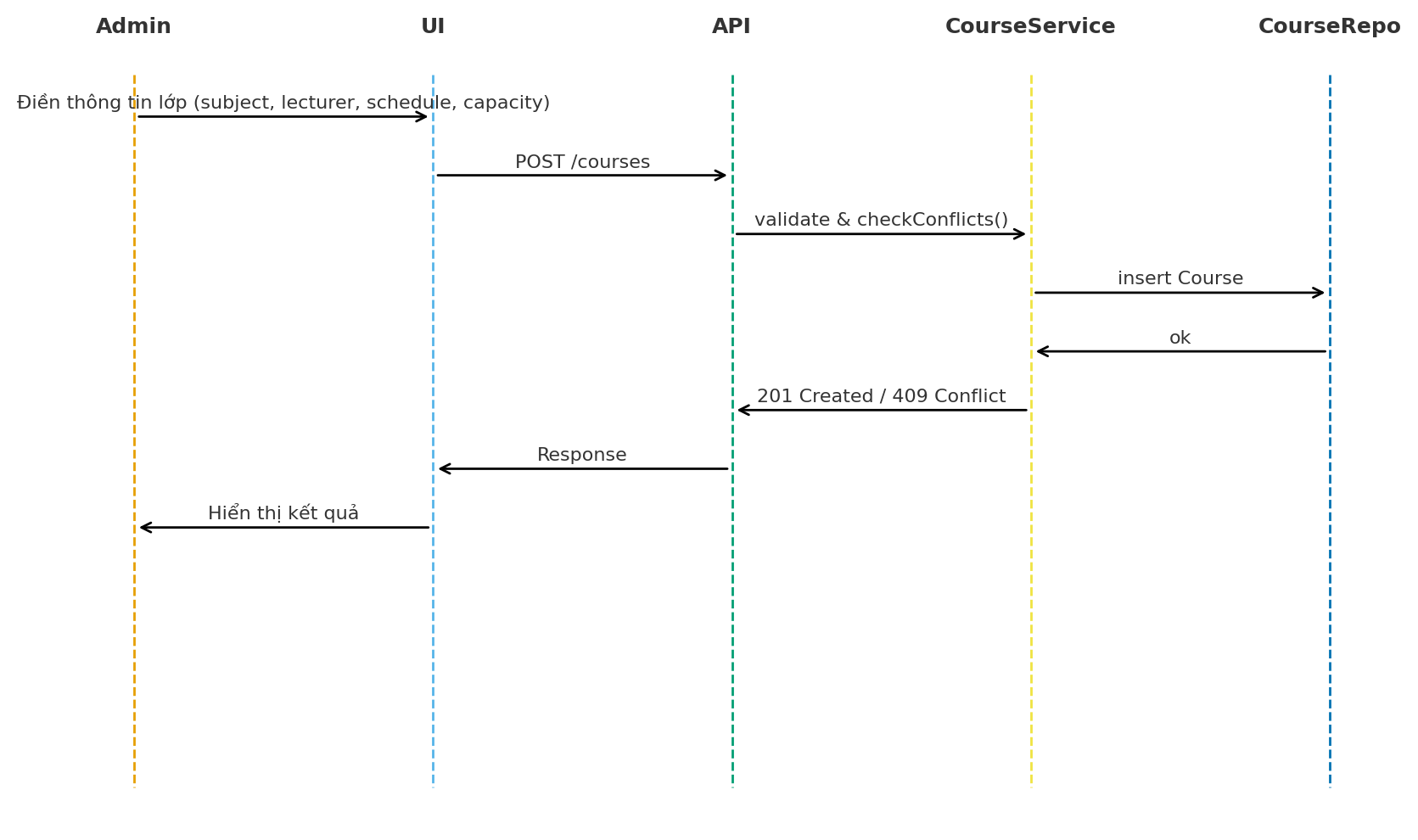
3.2.2.1. Admin tạo lớp học phần

Actor: Admin → UI → API → CourseService → Repos.

Luồng: Admin điền Subject, mã lớp, sĩ số tối đa, học kỳ, giảng viên, lịch học (thứ/ca/tiết/phòng).

Validation: kiểm tra học kỳ active; lịch không trùng (cùng phòng/cùng giảng viên/cùng lớp‑mã) theo dayOfWeek & time range; capacity ≥ 0.

Kết quả: tạo Course, trả 201; nếu trùng lịch → 409 CONFLICT + thông báo chi tiết slot.



3.2.2.2. Admin phê duyệt/ từ chối đăng ký

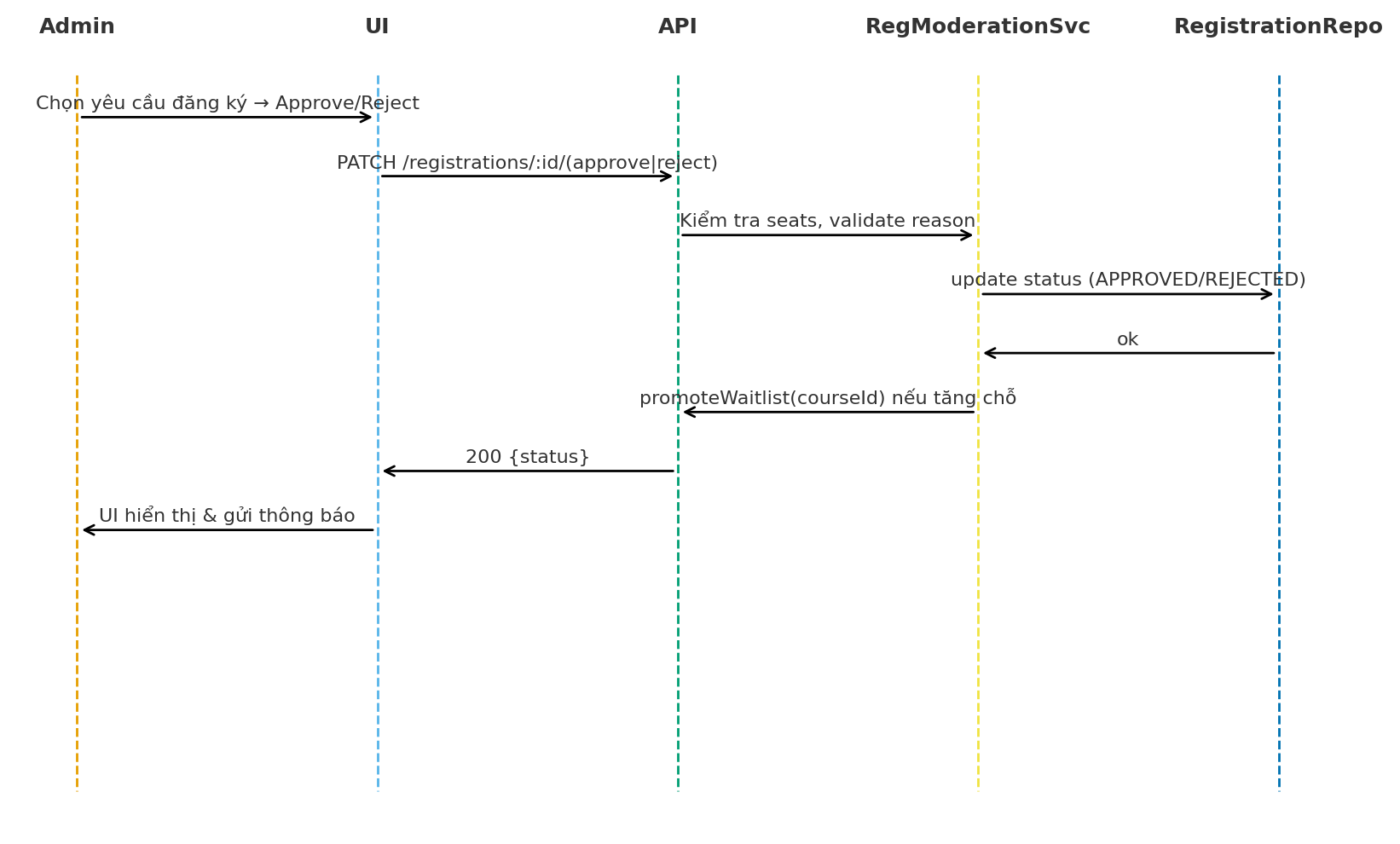
Actor: Admin/ Lecturer (đối với lớp của mình).

UI lọc theo tên SV/khóa học/trạng thái, bật 'Chỉ hiển thị lớp của tôi' cho GV.

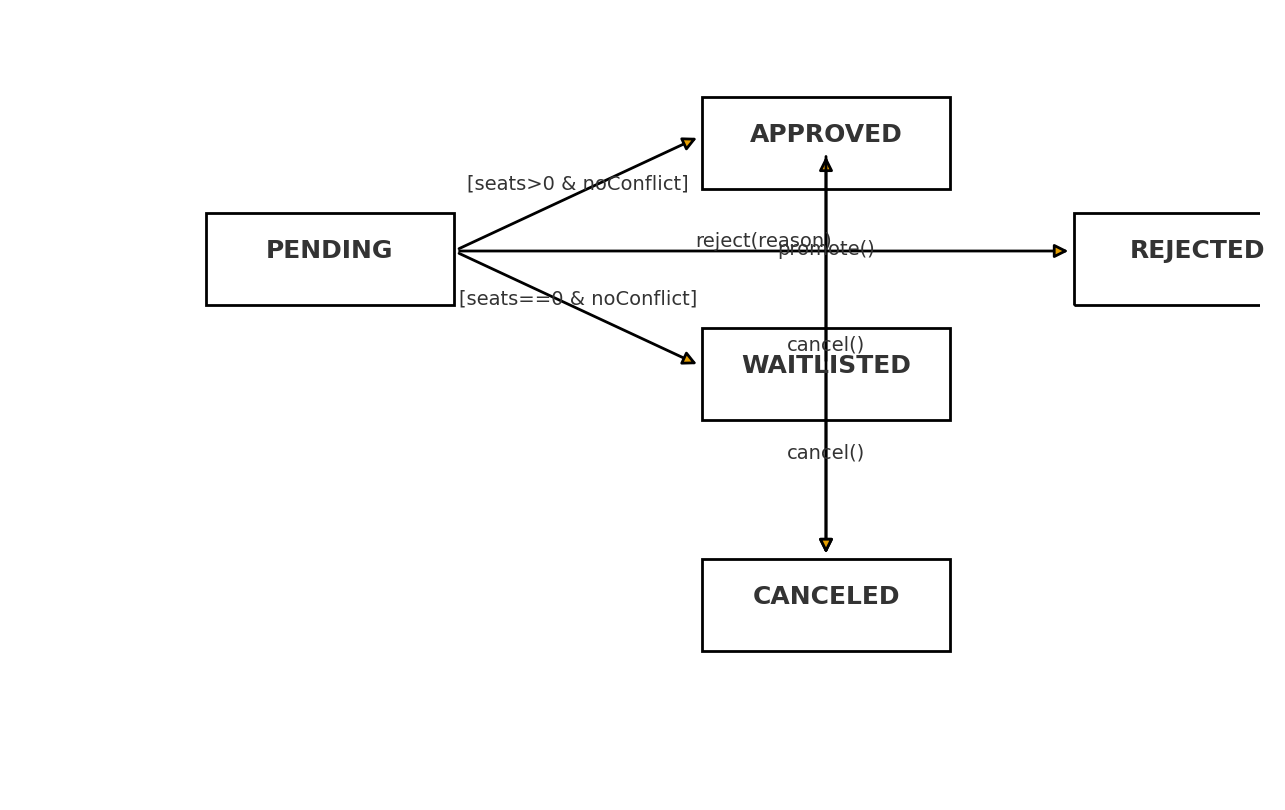
Approve: nếu còn chỗ → APPROVED; hết chỗ → WAITLISTED.

Reject: chuyển REJECTED, yêu cầu ghi lý do; hệ thống gửi thông báo cho SV và log cho Admin xem xét.

Admin có thể hủy bất kỳ đăng ký trước khi giảng viên can thiệp.



### 3.2.3. Biểu đồ Trạng thái (State Machine)



Mô tả: PENDING → APPROVED/WAITLISTED; từ WAITLISTED có thể promote lên APPROVED; mọi trạng thái có thể chuyển CANCELED; có nhánh REJECTED.

## 3.2.4. Thuật toán & ràng buộc nghiệp vụ

* Cửa sổ đăng ký: chỉ cho phép thao tác khi now∈[regOpenAt, regCloseAt] và semester.isActive==true.
* Lọc trùng lịch khi tạo Course: (giảng viên, phòng, ca, thứ) không trùng; sử dụng kiểm tra giao nhau của khoảng thời gian: max(startA,startB)<min(endA,endB).
* Xử lý capacity và chờ: seats=capacity-ApprovedCount; nếu tăng capacity → promoteWaitlist(courseId) nâng dần WAITLISTED theo FIFO; khi xóa Course → mọi Registration chuyển CANCELED (lưu audit).
* Quy tắc duyệt đăng ký: ưu tiên còn chỗ; nếu hết chỗ → WAITLISTED; Reject yêu cầu nhập reason và gửi thông báo.
* Ràng buộc tín chỉ học kỳ: creditMin/Max áp dụng ở client + server khi tổng tín chỉ đã duyệt vượt ngưỡng.

## 3.2.5. Bảo mật & nhật ký

* JWT + RBAC: admin/lecturer/student; chỉ backend quyết định vai trò từ token, không nhận role từ client.
* Audit log cho thao tác nhạy cảm: approve/reject/cancel registration, đổi capacity/xóa course.
* Chính sách mật khẩu mạnh, resetPassword cho giảng viên qua admin.

**Chương 4. Thiết kế giao diện**

4.1. Mục tiêu & phạm vi

* Mục tiêu: Thiết kế giao diện cho Hệ thống Quản lý Tín Chỉ (CREDIT) theo hướng dễ dùng, nhất quán và phù hợp bối cảnh sử dụng của sinh viên và quản trị.
* Phạm vi: Bộ Design System (màu, chữ, lưới, component), cấu trúc thông tin, luồng thao tác, màn hình chính và quy tắc tương tác.
* Ghi chú: Do giới hạn thời gian, phần kiểm thử hộp đen sẽ được thực hiện ở giai đoạn sau.

4.1.1. Mục tiêu UX (Trải nghiệm người dùng)

* Rút ngắn thao tác: Hoàn thành tác vụ chính (Đăng ký môn, Hủy môn, Tra cứu TKB) trong ≤ 4 bước.
* Tốc độ hoàn thành: Người dùng mới đăng ký 1 học phần trong ≤ 30–45 giây.
* Dễ tìm chức năng: ≥ 85% người thử nghiệm tìm đúng tính năng ngay lần đầu (không cần hướng dẫn).
* Giảm lỗi thao tác: Tỷ lệ nhập sai/misclick ≤ 5% mỗi phiên; có nhắc lỗi và gợi ý sửa tại chỗ.

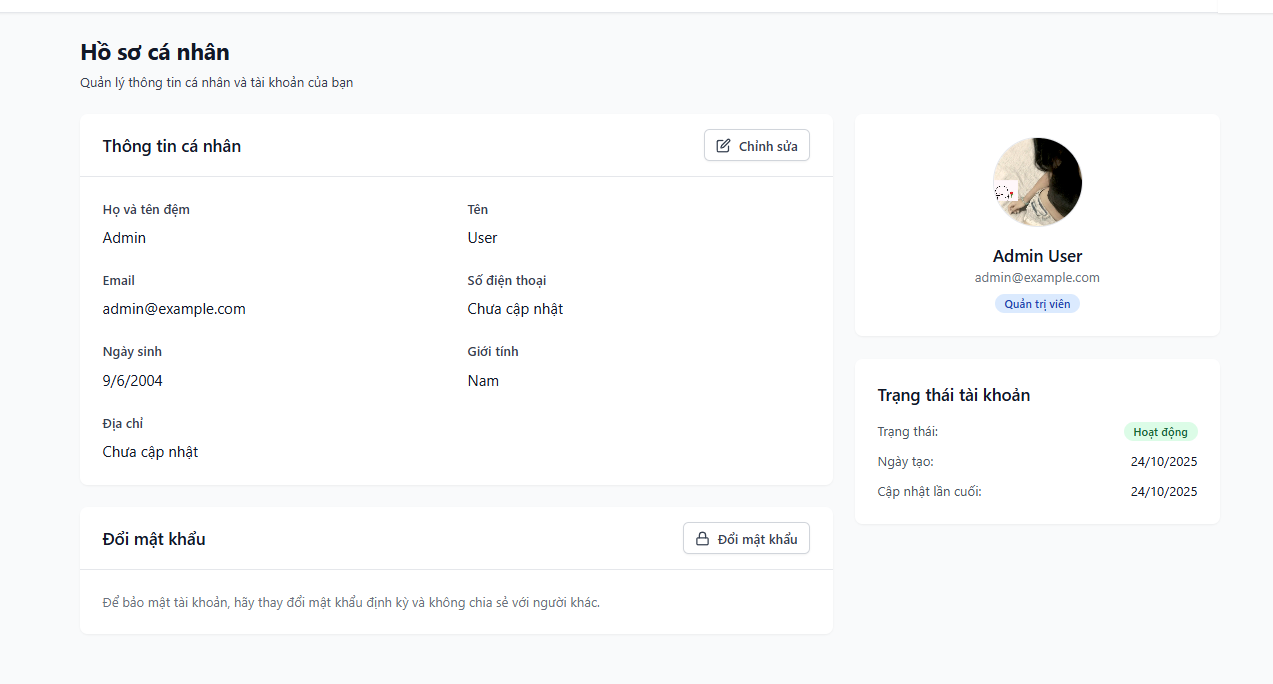
4.1.2. Mục tiêu UI (Giao diện)

* Nhất quán thị giác: Màu, font, spacing, icon tuân thủ Design System trên 100% màn hình của CREDIT.
* Khả năng truy cập: Tương phản văn bản ≥ 4.5:1; vùng chạm tối thiểu 44×44 px; hỗ trợ bàn phím với focus ring rõ ràng.
* Responsive chuẩn: 12 màn hình trọng yếu hiển thị đúng trên Desktop/Tablet/Mobile (không vỡ layout, không tràn chữ).
* Đủ trạng thái hệ thống: Mỗi màn hình có Loading / Empty / Error / Success theo chuẩn thành phần dùng chung.

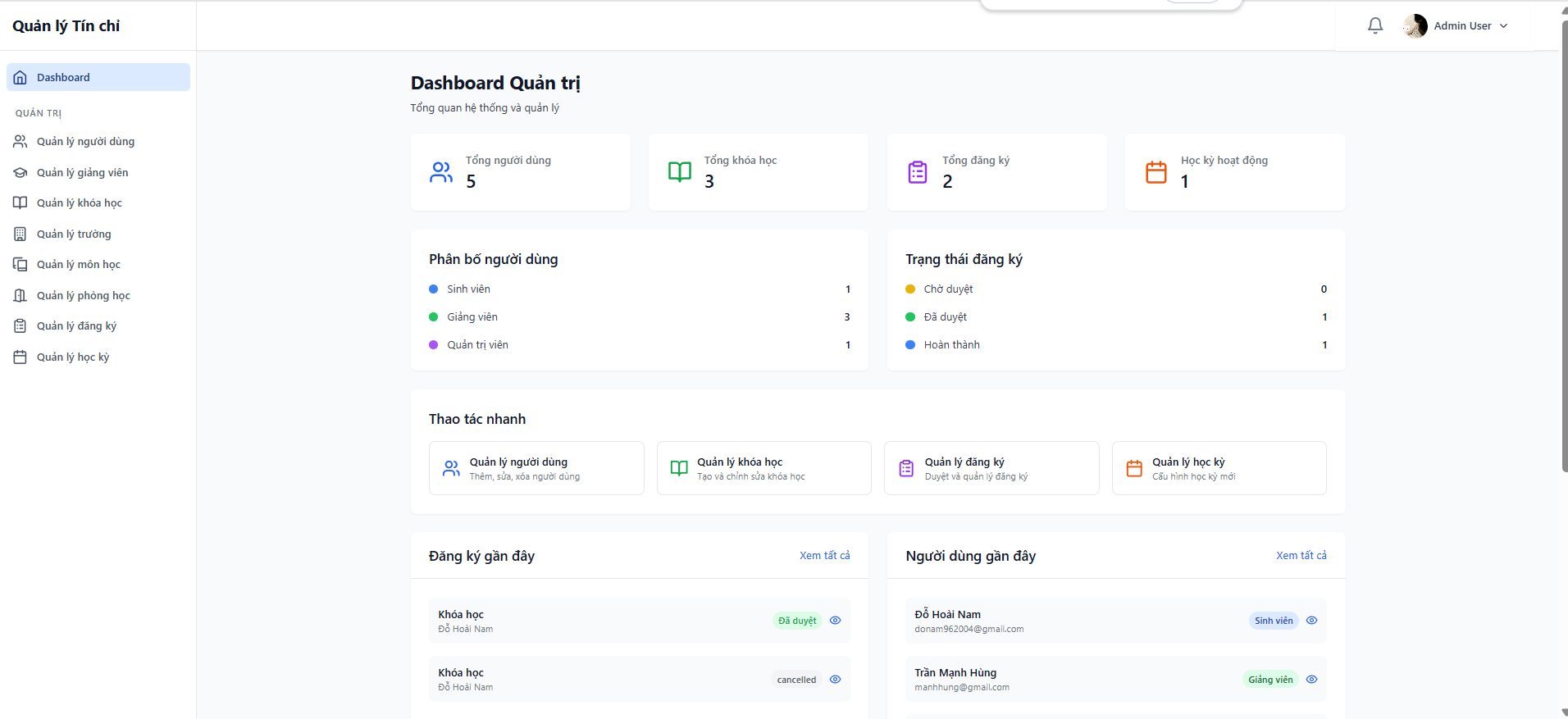
4.2. Giao diện chương trình

Trang đăng kí tài khoản

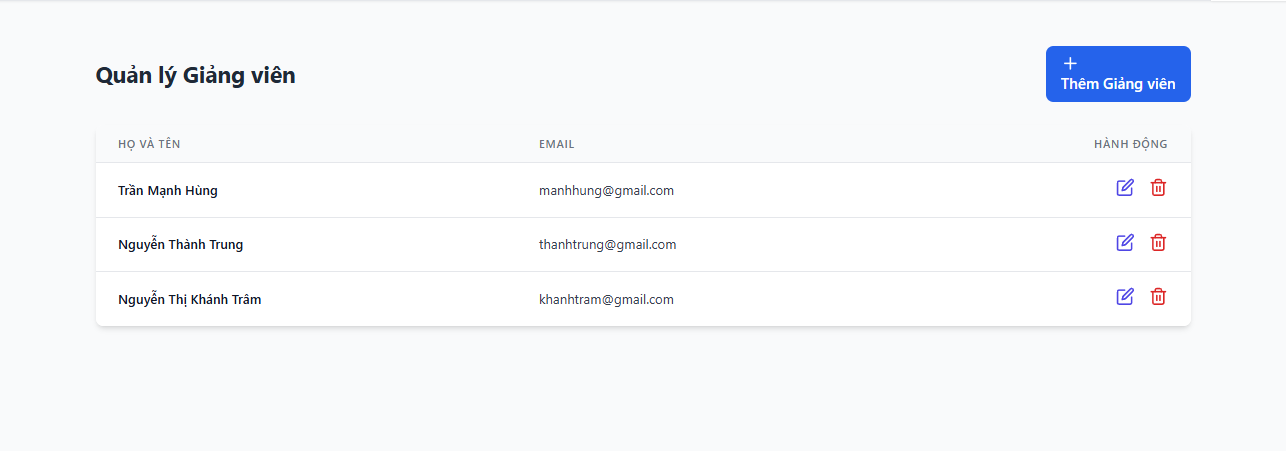
Giao diện trang admin:



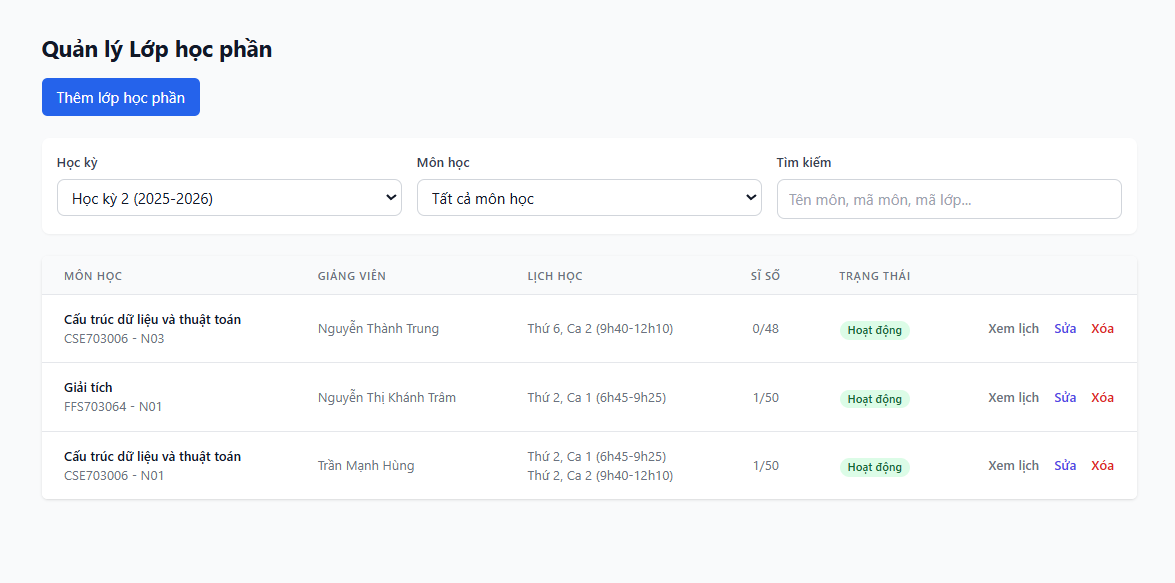
Hồ sơ cá nhân admin



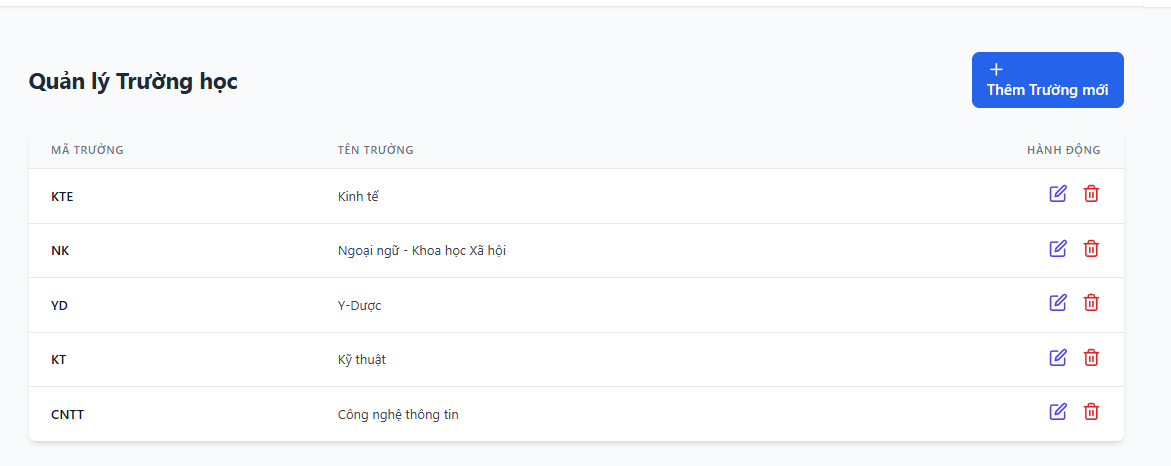
Dashboard admin



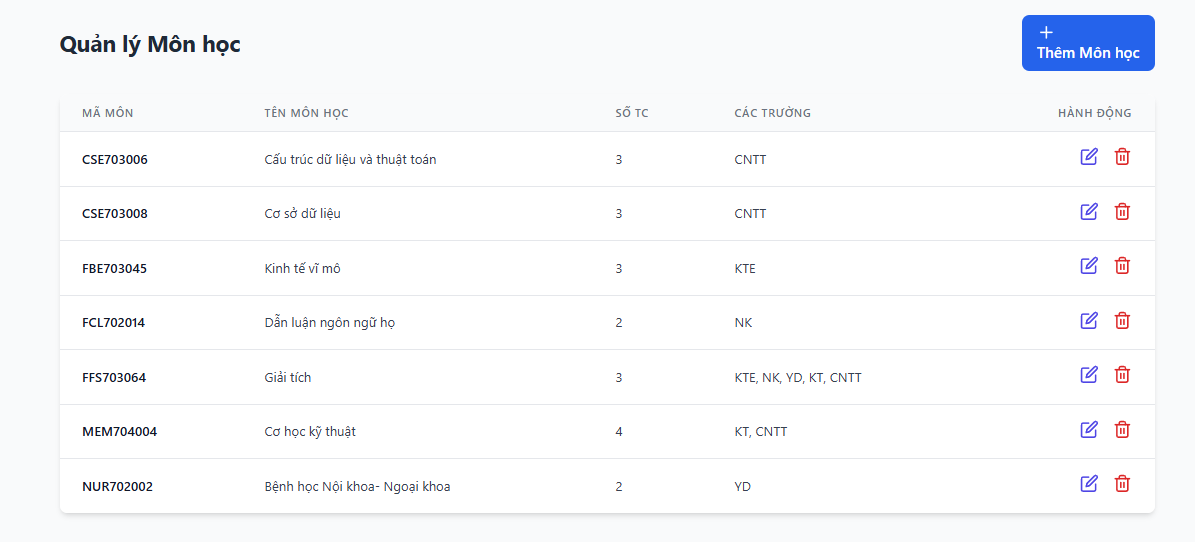
Quản lý giảng viên

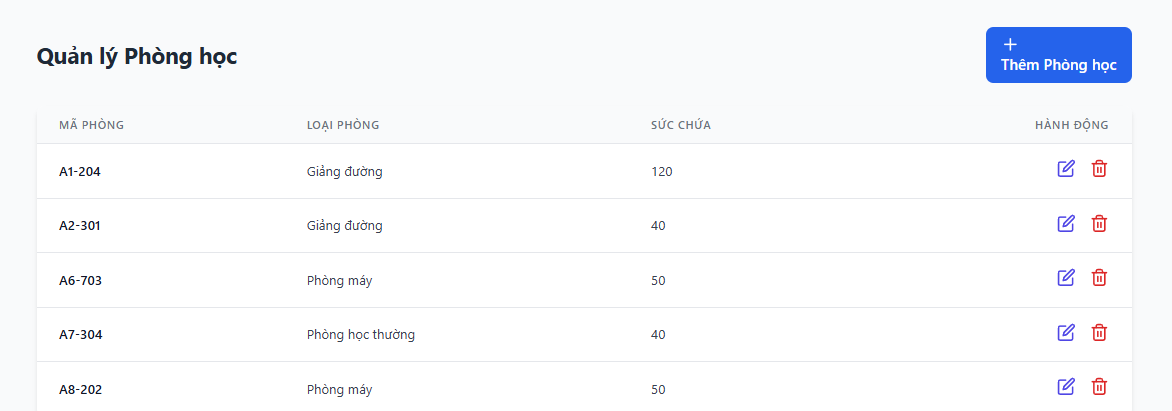


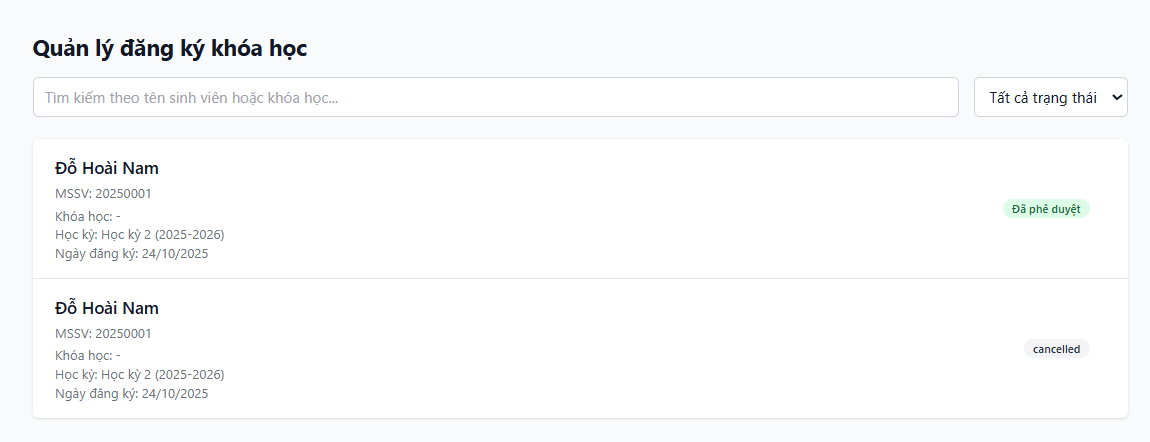
Quản lý lớp học phần

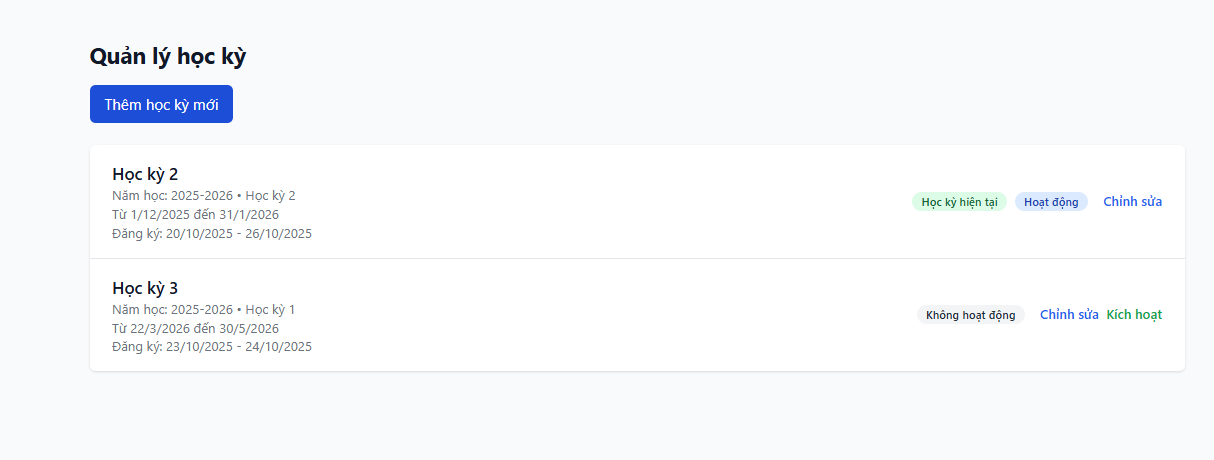


Quản lý trường học









**Chương 5. Tổng kết và hướng phát triển**

* Hệ thống đã mô hình hóa miền tốt (User–Subject–Course–Semester–Registration), ràng buộc tiên quyết, lịch học, sĩ số và quy trình đăng ký có duyệt/đợi.
* Phân quyền + JWT bảo vệ API; dữ liệu có kiểm tra đầu vào và chỉ mục phục vụ lọc/phân trang.
* Có trường thông tin phục vụ quản lý học tập (điểm, điểm danh, học phí cơ bản).
* Hạn chế: chưa có cơ chế giao dịch/atomic khi lớp gần đầy; policy thời gian/trần tín chỉ còn phân tán; test tự động và tài liệu API chưa đầy đủ; bảo vệ vận hành (rate-limit, helmet, audit log) còn thiếu.
* Bài học: cần chuẩn hoá múi giờ/khung chính sách, thêm kiểm thử E2E, và tăng lớp phòng vệ an toàn.

**Ngắn hạn:**

* Giao dịch khi đăng ký/hủy + tự động nâng waitlist; củng cố bảo mật (helmet, rate-limit, 2FA, audit log); chuẩn hóa policy hạn đăng ký/trần tín chỉ; công bố Swagger; CI chạy test; logging/monitoring.

**Trung hạn:**

* Billing–thanh toán học phí; tìm môn thông minh + cảnh báo xung đột realtime; dashboard báo cáo; quy trình học vụ mở rộng (phúc khảo/đổi lớp); quản lý lịch phòng–định mức giảng viên.

**Dài hạn:**

* Tích hợp SIS/LMS + SSO; hỗ trợ multi-tenant đa cơ sở; ứng dụng di động + thông báo đẩy; tối ưu hiệu năng (cache, tách dịch vụ, sharding); quản trị dữ liệu & tuân thủ.

**Kết luận:**

* Nền tảng chức năng cốt lõi đã sẵn sàng; để lên quy mô production, ưu tiên an toàn giao dịch, bảo mật–giám sát, bao phủ kiểm thử, và nâng UX lịch/cảnh báo thời gian thực.

**Chương 6. Tài liệu tham khảo**

6.1. Tài liệu kỹ thuật chính

1. Node.js Foundation, “Node.js Documentation,” Online: <https://nodejs.org/en/docs>
2. Express.js, “Express Guide & API,” Online: <https://expressjs.com>
3. React Team, “React Docs,” Online: <https://react.dev>
4. MongoDB Inc., “MongoDB Manual,” Online: <https://www.mongodb.com/docs>
5. Mongoose, “Mongoose ODM Docs,” Online: <https://mongoosejs.com/docs>
6. Jones, M. et al., “JSON Web Token (JWT),” *RFC 7519*, IETF, 2015. Online: <https://www.rfc-editor.org/rfc/rfc7519>
7. bcrypt, “bcrypt Library Documentation,” Online: <https://github.com/kelektiv/node.bcrypt.js>
8. Axios, “Axios HTTP Client,” Online: <https://axios-http.com>
9. Postman, “API Platform Docs,” Online: <https://learning.postman.com>
10. Jest, “Jest Testing Framework,” Online: <https://jestjs.io/docs>
11. Cypress, “Cypress E2E Testing,” Online: <https://docs.cypress.io>
12. OWASP, “Cheat Sheet Series (Authentication, Authorization, Input Validation…),” Online: <https://cheatsheetseries.owasp.org>

6.2. Chuẩn & thực hành khuyến nghị  
13. OWASP, “ASVS – Application Security Verification Standard,” Online: https://owasp.org/www-project-application-security-verification-standard  
14. Mozilla, “HTTP Security Guidelines (CORS, Cookies, Headers),” Online: https://developer.mozilla.org