

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



Vũ Trung Anh

**NGHIÊN CỨU DJANGO VÀ
PHÁT TRIỂN ỨNG DỤNG WEB DỰ BÁO KẾT QUẢ
HỌC TẬP**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

HÀ NỘI - 2021

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

Vũ Trung Anh

**NGHIÊN CỨU DJANGO VÀ
PHÁT TRIỂN ỨNG DỤNG WEB DỰ BÁO KẾT QUẢ
HỌC TẬP**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ thông tin**

Cán bộ hướng dẫn: PGS.TS Nguyễn Việt Anh

HÀ NỘI – 2021

LỜI CẢM ƠN

Em xin chân thành cảm ơn đến thầy hướng dẫn PGS.TS Nguyễn Việt Anh, người đã nhiệt tình giúp đỡ, hướng dẫn, đóng góp ý kiến, định hướng và chỉ dạy cho em những bài học, kinh nghiệm để em có thể hoàn thành Khóa luận tốt nghiệp này.

Em xin cảm ơn bạn Nguyễn Đức Lộc, người đã đưa ra số liệu về dự đoán điểm môn học để hệ thống có thể đưa ra dự đoán kết quả học tập cho sinh viên.

Em xin cảm ơn Ban giám hiệu đã tạo điều kiện thuận lợi để em có thể hoàn thành Đồ án tốt nghiệp này. Em cảm ơn toàn thể các thầy cô giáo, giảng viên đang công tác tại trường Đại học Công Nghệ - Đại học Quốc gia Hà Nội đã chỉ dạy, định hướng, truyền đạt cho em những kiến thức hữu ích và kinh nghiệm quý báu trong suốt 4 năm học để em có thể hoàn thành Khóa luận tốt nghiệp này.

Trong quá trình hoàn thành khóa luận tốt nghiệp vì kiến thức chuyên môn còn bị hạn chế và chưa có nhiều kinh nghiệm nên nội dung của khóa luận không tránh khỏi những thiếu sót và hạn chế, em rất mong nhận được sự đóng góp, chỉ bảo của quý thầy cô để khóa luận này được hoàn thiện hơn.

Em xin chân thành cảm ơn!

TÓM TẮT

Tóm tắt: Hiện nay, các tiện ích hỗ trợ con người ngày càng nhiều, ngày càng phát triển trong toàn bộ các lĩnh vực. Đặc biệt với nghiệp vụ tư vấn, dự đoán học tập thì dữ liệu và cách thức đưa thông tin cho người dùng quyết định đến hiệu quả của việc tư vấn học tập. Khóa luận này triển khai nội dung là tìm hiểu, nghiên cứu công nghệ Django trong phát triển ứng dụng website và phát triển hệ thống dự báo kết quả học tập cho sinh viên đang theo học tại các trường Đại học Công nghệ hay rộng hơn là các trường thành viên trực thuộc Trường Đại học Quốc Gia Hà Nội. Đối với hệ thống sẽ cung cấp các con số tư vấn về môn học, điểm số dự đoán cụ thể và các thống kê cần thiết để sinh viên đưa ra quyết định học.

Hệ thống hoạt động gồm hai yếu tố cốt lõi là:

- Nguồn dữ liệu về môn học, sinh viên, điểm số của sinh viên.
- Hoạt động huấn luyện, tính toán dự báo điểm.

Để xây dựng được hệ thống dự đoán kết quả học tập cho sinh viên thì cần thực hiện:

- Tìm hiểu công cụ Django.
- Thiết kế cấu trúc hệ thống.
- Tiến hành phát triển hệ thống.

Từ khóa: *Django, Dự báo kết quả học tập, Gợi ý môn học.*

LỜI CAM ĐOAN

Em xin cam đoan những nội dung nghiên cứu và phát triển của em về đề tài “Nghiên cứu công cụ Django và áp dụng phát triển front-end cho bài toán dự báo kết quả học tập” là hoàn toàn không sao chép của bất kỳ ai và tài liệu nào. Mọi tài liệu tham khảo đều được trình dẫn rõ ràng ở mục Tài liệu tham khảo.

Em xin cam đoan hệ thống mà em trình bày trong khóa luận này là do em tự phát triển, không sao chép mã nguồn của người khác. Nếu có bất cứ nội dung cam đoan nào em vi phạm thì em hoàn toàn chịu trách nhiệm theo quy định của Trường Đại học Công nghệ - Đại học Quốc gia Hà Nội.

Hà Nội, ngày 30 tháng 6 năm 2021

Sinh viên

Vũ Trung Anh

Mở đầu.....	11
Chương 1. Bài toán và Công nghệ	13
1.1. Framework Django	13
1.1.1. Cơ chế hoạt động.....	13
1.1.1.1. Thông tin chung.....	13
1.1.1.2. Mô hình hoạt động.....	13
1.1.2. Cách thức phát triển.....	14
1.1.2.1. Biểu đồ tuần tự	14
1.1.2.2. Chi tiết	14
1.2. Cách thức sử dụng Framework Django.....	15
1.2.1. Đối tượng.....	15
1.2.1.1. Dữ liệu của thuộc tính	15
1.2.1.2. Mối quan hệ.....	16
1.2.2. Xử lý logic	17
1.2.2.1. Dữ liệu liên quan đến các đối tượng.....	17
1.2.2.2. Các thư viện khác	18
1.2.2.3. Dữ liệu hiển thị.....	19
1.2.3. URL	20
1.2.3.1. Đường dẫn thành phần.....	20
1.2.3.2. Đường dẫn chi tiết	20
1.2.4. Giao diện.....	20
1.2.4.1. Xây dựng khung giao diện	20
1.2.4.2. Xử lý dữ liệu.....	21
1.2.5. Phân quyền	21
1.2.6. API.....	22
Chương 2. Phân tích thiết kế hệ thống	23
2.1. Phân tích	23
2.1.1. Các tác nhân hệ thống.....	23
2.1.1.1. Các hạng mục người quản lý có quyền thao tác.....	23
2.1.1.2. Yếu tố ảnh hưởng đến người quản lý	23
2.1.2. Sinh viên	23
2.1.2.1. Các hạng mục sinh viên có thể thao tác	23
2.1.2.2. Yêu cầu thao tác với thông tin tư vấn của sinh viên	23

2.1.3. Quản trị	24
2.1.3.1. Việc phân quyền chính của hệ thống.....	24
2.1.3.2. Cập nhật dữ liệu liên quan đến tư vấn học tập	24
2.2. Ca sử dụng	25
2.2.1. Sơ đồ chính	25
2.2.2. Chi tiết ca sử dụng	25
2.2.2.1. Quản lý các danh mục	25
2.2.2.2. Quản lý chức năng hệ thống	31
2.2.2.3. Lựa chọn mô hình.....	33
2.2.2.4. Thống kê	34
2.2.2.5. Dự đoán kết quả học tập	37
2.2.2.6. Gợi ý môn học	40
2.2.2.7. Chỉnh sửa thông tin hồ sơ.....	41
2.2.2.8. Đăng nhập.....	42
2.2.2.9. Đăng xuất.....	44
2.3. Thiết kế	46
2.3.1. Thiết kế kiến trúc hệ thống.....	46
2.3.1.1. Tổng quan về cấu trúc	46
2.3.1.2. Mô tả chi tiết.....	47
2.3.2. Thiết kế cơ sở dữ liệu	48
2.3.2.1. Mô hình ER	48
2.3.2.2. Mô hình quan hệ	50
2.3.2.3. Cơ sở dữ liệu vật lý.....	52
Chương 3. Cài đặt thử nghiệm và đánh giá	60
3.1. Sử dụng API	60
3.1.1. Thống kê	60
3.1.1.1. Thống kê phổ điểm theo môn và khóa	60
3.1.1.2. Thống kê trung bình điểm môn học qua các năm	61
3.1.1.3. Thống kê GPA theo khóa và ngành.....	61
3.1.1.4. Thống kê GPA theo sinh viên	63
3.1.2. Dự đoán	64
3.1.2.1. Dự đoán điểm môn học	64
3.1.2.2. Dự đoán điểm toàn khóa.....	64

3.1.3. Gợi ý	64
3.2. Cài đặt và thử nghiệm.....	65
3.2.1. Cài đặt.....	65
3.2.1.1. Cài đặt python.....	65
3.2.1.2. Cài đặt project Django.....	65
3.2.1.3. Cài đặt các thư viện khác.....	66
3.2.2. Khởi động chương trình	67
3.2.2.1. Makemigrations	67
3.2.2.2. Migrate.....	67
3.2.3. Kết quả thu được	68
3.3. Đánh giá.....	71
Kết luận.....	72

Danh mục hình ảnh

Hình 1.1. Biểu đồ tuần tự mô hình hoạt động.	14
Hình 1.2. Đối tượng có chứa khóa chính.....	16
Hình 1.3. Đối tượng sử dụng khóa ngoại tham chiếu.....	16
Hình 1.4. Đối tượng Unit Serialize.....	18
Hình 1.5. Hàm trả dữ liệu dạng đã biến đổi.	18
Hình 1.6. Đối tượng biểu mẫu.....	19
Hình 1.7. Phương thức xử lý tạo mới và cập nhật.....	19
Hình 1.8. Phương thức hiển thị giao diện có kiểm tra đăng nhập.	19
Hình 1.9. Phương thức trả về số trang về gửi hạn bản ghi để hiển thị.	20
Hình 1.10. Các thư viện sử dụng và định nghĩa middleware.	21
Hình 1.11. Hàm xử lý phương thức GET của API.....	22
Hình 1.12. Đường dẫn của API.	22
Hình 1.13. Hàm xử phương thức POST của API.....	22
Hình 2.1. Tổng quan toàn bộ ca sử dụng.....	25
Hình 2.2. Biểu đồ các thao tác khi thực hiện thêm mới dữ liệu.	27
Hình 2.3. Biểu đồ các thao tác khi thực hiện cập nhật dữ liệu.	27
Hình 2.4. Biểu đồ các thao tác khi thực hiện xóa dữ liệu.....	27
Hình 2.5. Biểu đồ tuần tự tính năng xem danh sách.....	28
Hình 2.6. Biểu đồ tuần tự tính năng tạo mới và sửa nội dung.....	29
Hình 2.7. Biểu đồ tuần tự tính năng xóa nội dung.	30
Hình 2.8. Biểu đồ tuần tự tính năng xuất file csv.....	30
Hình 2.9. Biểu đồ tuần tự tính năng nhập file csv.....	31
Hình 2.10. Biểu đồ các thao tác khi thực hiện thêm mới chức năng hệ thống.....	32
Hình 2.11. Biểu đồ các thao tác khi thực hiện cập nhật chức năng hệ thống.....	33
Hình 2.12. Biểu đồ các thao tác khi thực hiện thêm mới chức năng hệ thống.....	33
Hình 2.13. Biểu đồ các thao tác khi thực hiện lựa chọn mô hình.	34
Hình 2.14. Biểu đồ các thao tác khi thực hiện chọn mô hình cho hệ thống.....	34
Hình 2.15. Biểu đồ các thao tác khi thực hiện xem thống kê.....	35
Hình 2.16. Biểu đồ tuần tự tính năng thống kê theo môn học và khóa.	35
Hình 2.17. Biểu đồ tuần tự tính năng thống kê điểm trung bình môn học qua các năm.	36
Hình 2.18. Biểu đồ tuần tự tính năng thống kê GPA theo khóa và ngành.	36
Hình 2.19. Biểu đồ tuần tự tính năng thống kê GPA sinh viên.....	37
Hình 2.20. Biểu đồ các thao tác khi thực hiện xem dự báo điểm.....	38
Hình 2.21. Biểu đồ các thao tác khi thực hiện tính dự báo theo môn.	38
Hình 2.22. Biểu đồ các thao tác khi thực hiện tính dự báo theo khóa.....	38
Hình 2.23. Biểu đồ tuần tự tính năng dự đoán điểm số.....	39
Hình 2.24. Biểu đồ các thao tác khi thực hiện xem gợi ý môn học.	40
Hình 2.25. Biểu đồ tuần tự tính năng gợi ý môn học.	41
Hình 2.26. Biểu đồ các thao tác khi thực hiện cập nhật hồ sơ cá nhân.	42

Hình 2.27. Biểu đồ tuần tự thao tác chỉnh sửa hồ sơ.....	42
Hình 2.28. Biểu đồ các thao tác khi thực hiện đăng nhập.	43
Hình 2.29. Biểu đồ tuần tự tính năng đăng nhập.....	44
Hình 2.30. Biểu đồ các thao tác khi thực hiện đăng xuất.....	45
Hình 2.31. Biểu đồ tuần tự tính năng đăng xuất.....	45
Hình 2.32. Cấu trúc thư mục.	46
Hình 2.33. Tổng quan mô hình ER.....	48
Hình 2.34. Tổng quan quan hệ của các thực thể.	50
Hình 3.1. Kết quả API thống kê phổ điểm.	60
Hình 3.2. Kết quả API thống kê điểm trung bình môn học.....	61
Hình 3.3. Kết quả API thống kê GPA theo khóa và ngành.	62
Hình 3.4. Kết quả API thống kê GPA theo sinh viên.....	63
Hình 3.5. Các thư viện sử dụng trong toàn bộ dự án.....	67
Hình 3.6. Giao diện đăng nhập của hệ thống.	68
Hình 3.7. Thống kê phổ điểm.....	68
Hình 3.8. Thống kê điểm trung bình.	69
Hình 3.9. Thống kê GPA.....	69
Hình 3.10. Thống kê GPA sinh viên.	70
Hình 3.11. Kết quả dự đoán điểm của sinh viên.	70
Hình 3.12. Các môn học gợi ý cho sinh viên.	71

Danh mục bảng biểu

Bảng 1.1. Danh sách các kiểu dữ liệu trong Models	15
Bảng 1.2. Bảng các giá trị của thuộc tính on_delete	17
Bảng 1.3. Các phương thức trong middleware.....	22
Bảng 2.1. Bảng mô tả các hành động trong luồng chính.....	26
Bảng 2.2. Bảng mô tả các hành động trong luồng chính quản lý chức năng.	32
Bảng 2.3. Bảng mô tả các hành động trong luồng chính lựa chọn mô hình.	33
Bảng 2.4. Bảng mô tả các hành động trong luồng chính xem thống kê.....	34
Bảng 2.5. Bảng mô tả các hành động trong luồng chính dự đoán điểm.....	37
Bảng 2.6. Bảng mô tả các hành động trong luồng chính gợi ý môn học.	40
Bảng 2.7. Bảng mô tả các hành động trong luồng chính chỉnh sửa thông tin hồ sơ.	41
Bảng 2.8. Bảng mô tả các hành động trong luồng chính đăng nhập.	43
Bảng 2.9. Danh sách các bảng dựa trên các thực thể của hệ thống.....	52
Bảng 2.10. Cấu trúc bảng auth_group.	52
Bảng 2.11. Cấu trúc bảng auth_group_permissions.	52
Bảng 2.12. Cấu trúc bảng auth_permission.....	52
Bảng 2.13. Cấu trúc bảng auth_user_groups.....	52
Bảng 2.14. Cấu trúc bảng auth_user_user_permissions.....	53
Bảng 2.15. Cấu trúc bảng django_admin_log.	53
Bảng 2.16. Cấu trúc bảng django_content_type.....	53
Bảng 2.17. Cấu trúc bảng mainapp_courses.	54
Bảng 2.18. Cấu trúc bảng mainapp_customer.....	54
Bảng 2.19. Cấu trúc bảng mainapp_customser_function.....	54
Bảng 2.20. Cấu trúc bảng mainapp_dumpmodel.	55
Bảng 2.21. Cấu trúc bảng mainapp_functions.....	55
Bảng 2.22. Cấu trúc bảng mainapp_generations.....	55
Bảng 2.23. Cấu trúc bảng mainapp_gpa.....	55
Bảng 2.24. Cấu trúc bảng mainapp_gradepredicted.....	56
Bảng 2.25. Cấu trúc bảng mainapp_logs.....	56
Bảng 2.26. Cấu trúc bảng mainapp_majors.....	56
Bảng 2.27. Cấu trúc bảng mainapp_major_course.....	56
Bảng 2.28. Cấu trúc bảng mainapp_predicthistory.	57
Bảng 2.29. Cấu trúc bảng mainapp_profiles.	57
Bảng 2.30. Cấu trúc bảng mainapp_roles.....	57
Bảng 2.31. Cấu trúc bảng mainapp_role_function.....	57
Bảng 2.32. Cấu trúc bảng mainapp_semesters.....	58
Bảng 2.33. Cấu trúc bảng mainapp_studentgroups.....	58
Bảng 2.34. Cấu trúc bảng mainapp_traindata.	58
Bảng 2.35. Cấu trúc bảng mainapp_transcript.	58
Bảng 2.36. Cấu trúc bảng mainapp_transcriptfile.....	59
Bảng 2.37. Cấu trúc bảng mainapp_units.....	59

Bảng 2.38. Cấu trúc bảng mainapp_years.	59
Bảng 3.1. Bảng chú thích API thống kê phổ điểm.	60
Bảng 3.2. Bảng chú thích API thống kê điểm môn học.	61
Bảng 3.3. Bảng chú thích API thống kê GPA.	62
Bảng 3.4. Bảng chú thích API thống kê GPA sinh viên.	63
Bảng 3.5. Bảng chú thích API tính năng gợi ý môn học.	64

Mở đầu

Tính cấp thiết

Theo mô hình đăng ký học theo tín chỉ hiện tại của trường Đại học Công Nghệ hay là toàn thể các trường thành viên trực thuộc Đại học Quốc Gia Hà Nội thì việc công tác tư vấn học tập đóng vai trò quan trọng nhất định trong định hướng học tập và định hướng nghề nghiệp tương lai cho sinh viên. Tư vấn học tập giúp sinh viên tiếp nhận được thông tin để tìm được vấn đề của sinh viên đang gặp phải. Khi tìm ra vấn đề thì sự tư vấn hợp lý cũng giúp giải quyết vấn đề bằng cách hoạch định phương hướng học tập cho sinh viên hoặc đưa ra lời khuyên để sinh viên nhận thức và thay đổi hành vi của bản thân. Chính vì vậy, tư vấn học tập giúp sinh viên tiếp thu kinh nghiệm học tập để phát triển bản thân hơn.

Từ trước đến nay, sinh nhận được sự cố vấn học tập, lời khuyên về học tập thông qua các thầy, cô cố vấn học tập, thầy cô giáo giảng dạy trực tiếp hoặc là nghe kinh nghiệm từ các bạn bè, anh, chị khóa trên.

Đến nay, xoay quanh vấn đề học tập thì công tác tư vấn học tập đã và đang đáp ứng được hầu hết nhu cầu của sinh viên và có thể đưa ra các phương hướng giải quyết vấn đề cho sinh viên. Nhất là các vấn đề liên quan đến môn học và điểm số học tập thì các thầy cô cố vấn đã hỗ trợ rất nhiều trong quá trình học tập và hỗ trợ định hướng cho sinh viên. Thầy cô cố vấn cung cấp thông tin về môn học, bài tập xoay quanh môn học cũng như ứng dụng của môn học và định hướng của môn học. Nhưng do quy mô của toàn bộ trường đại học cũng như khả năng của thầy cô cố vấn không phải lúc nào cũng có thể lấy dẫn chứng bằng điểm số các môn học cho tất cả các sinh viên được.

Vì vậy, khi sinh viên có nguồn số liệu thống kê về môn học và hơn nữa là nhận được những môn học đề xuất phù hợp thì sinh viên sẽ đưa ra quyết định phù hợp với khả năng, học lực của bản thân sinh viên hơn. Hơn nữa khi có số liệu thống kê cụ thể thì sinh viên tự tham chiếu với kết quả của bản thân để có động lực phấn đấu làm tốt hơn.

Bài toán dự báo và gợi ý là một bước tiếp cận, phương thức tư vấn cải tiến hơn. Từ nhu cầu về các số liệu tham khảo của các môn học dành cho sinh viên thì giải pháp là tạo ra một công cụ hỗ trợ, một hệ thống cung cấp các thông tin về thống kê các môn học và đề xuất môn học cho sinh viên.

Mô tả bài toán

Để giải quyết được nhu cầu về các dẫn chứng tham khảo khi nhận được sự tư vấn môn học cũng như các số liệu cụ thể khi nhận sự tư vấn môn học thì mỗi sinh viên cần được xem lại các đầu điểm dự báo về môn học sinh viên đang có nhu cầu đăng ký học. Cũng như từ những thang điểm dự báo như vậy thì sinh viên tự đưa ra lựa chọn cho việc đăng ký học của mình để. Cũng từ các kết quả học tập mà dự báo được thì cũng góp phần để gợi ý xem là sinh viên nên học môn nào để đạt thành tích trung bình cao nhất hay là cải thiện kết quả tổng kết của bản thân.

Để có sự tư vấn môn học chính xác và tin cậy thì các biểu đồ thống kê lại kết quả của môn học ở các khóa trước, các ngành hay các năm học trước như thế nào. Việc có hình ảnh cụ thể sẽ giúp sinh viên hình dung ra bức tranh tổng thể về điểm số của môn học.

Những mỗi sinh viên trực thuộc một trường xác định và theo học một ngành ở thời điểm xác định. Sinh viên tham gia học các môn học thuộc chương trình học và từ điểm kết quả của môn học sẽ được lưu lại. Với những dữ liệu điểm môn học của sinh viên thì làm cơ sở để tính toán, dữ liệu huấn luyện cho hệ thống để có thể cho các kết quả tư vấn hợp lý đối với từng sinh viên theo môn học, theo kỳ học tiếp theo. Hơn nữa còn tạo ra những thống kê về môn học, điểm số để sinh viên tham khảo.

Mục tiêu

Cung cấp thông tin tư vấn hữu ích cho sinh viên: Cung cấp các thông tin về phổ điểm, điểm trung bình của các môn học, thống kê điểm của các khóa trước cho sinh viên để có thể đưa ra quyết định lựa chọn môn học. Ngoài ra còn đề xuất các môn học và dự đoán điểm phù hợp đối với từng sinh viên.

Sử dụng tốt công cụ django để hoàn thành bài toán: Nghiên cứu công cụ phát triển ứng dụng web django và áp dụng công cụ django để phát triển ứng dụng web giải quyết bài toán tư vấn và dự báo kết quả học tập.

Phạm vi nghiên cứu

Nghiên cứu cơ chế hoạt động và quy trình phát triển ứng dụng web với công cụ Django.

Phát triển ứng dụng trên nền tảng web bao gồm giao diện, xử lý luồng nghiệp vụ và sử dụng kết quả của các module dự báo và gợi ý để giải quyết bài toán tư vấn và dự báo kết quả học tập.

Đóng góp khóa luận

Dù công tác tư vấn học tập cho sinh viên trường Đại học Công nghệ đã hỗ trợ rất tốt cho công việc học tập và định hướng công việc cho các bạn sinh viên. Nhưng khi khóa luận này hoàn thành và đi vào phục vụ công tác hỗ trợ, tư vấn sinh viên sẽ trở thành yếu tố tham khảo xác thực và đa dạng về điểm số, môn học. Từ những thông tin tiếp nhận được thì cũng sẽ góp phần giúp sinh viên đưa ra lựa chọn môn học, kế hoạch học tập cho tương lai phù hợp.

Bố cục khóa luận

Chương 1 trình bày về công cụ phát triển web django và cách thức sử dụng Framework Django khi áp dụng vào bài toán thực tế. Chương 2 phân tích chi tiết các yếu tố cấu thành và hoàn thiện bài toán và từ đó để thiết kế các kiến trúc hệ thống, cơ sở dữ liệu. Chương 3 trình bày cách thức sử dụng API chính của hệ thống, triển khai thử nghiệm và thu về kết quả để đánh giá hiệu quả hoạt động. Cuối cùng là kết luận cùng với các tài liệu tham khảo sẽ được trình bày.

Chương 1. Bài toán và Công nghệ

1.1. Framework Django

1.1.1. Cơ chế hoạt động

1.1.1.1. Thông tin chung

Python là ngôn ngữ lập trình bậc cao. Python được sử dụng trong nhiều lĩnh vực như: khoa học máy tính, phát triển trí tuệ nhân tạo, phát triển ứng dụng web, phân tích dữ liệu...

Python là ngôn ngữ lập trình đặc biệt vì cấu trúc code đơn giản, không quá ràng buộc về khai báo và sử dụng biến, cú pháp không đơn giản và không sử dụng dấu chấm phẩy để đánh dấu kết thúc câu lệnh như đa số ngôn ngữ lập trình khác mà sử dụng các dấu cách để hiểu được câu lệnh khi thông dịch chương trình. Với cách tiếp cận đơn giản như vậy thì Python rất hiệu quả để tiếp cận lập trình hướng đối tượng.

Python cũng là ngôn ngữ lập trình dễ cài đặt và phát triển. Vì Python phù hợp với tất cả các hệ điều hành. Mỗi tất cả các hệ điều hành thì python đều có viết hướng dẫn cài đặt cụ thể. Việc cài đặt có thể thông qua vài câu lệnh hoặc tải trực tiếp trên trang chủ của python là python.org rồi thực hiện cài đặt trên giao diện của gói đã tải về.

Django là framework dành cho việc phát triển ứng dụng web và sử dụng ngôn ngữ lập trình Python. Cấu trúc mô hình phát triển ứng dụng web thì cơ bản dựa trên mô hình phát triển MVC. Vì là framework của python nên việc cài đặt trên các hệ điều hành cũng đều được hỗ trợ cụ thể trong tài liệu của framework.

1.1.1.2. Mô hình hoạt động

Mô hình MVT là mô hình hoạt động dựa trên ba thành phần là Model, View, Template.

Model là các thực thể đối tượng thực được thể hiện của các bảng trong cơ sở dữ liệu. Thông qua model thì lấy được dữ liệu và sử dụng và tính toán logic. Đây cũng chính là kỹ thuật ORM để ánh xạ từ cơ sở đối tượng sang các đối tượng. Mỗi thao tác tương tác đến với cơ sở dữ liệu thì các đối tượng ở model như là một trung gian. Khi sử dụng kết hợp với kỹ thuật ORM thì sẽ tránh được một số lỗi tấn công với cơ sở dữ liệu và giúp tăng hiệu suất truy xuất dữ liệu. Ngoài ra nó còn giúp việc xây dựng hệ thống đơn giản mà đạt hiệu quả cao hơn.

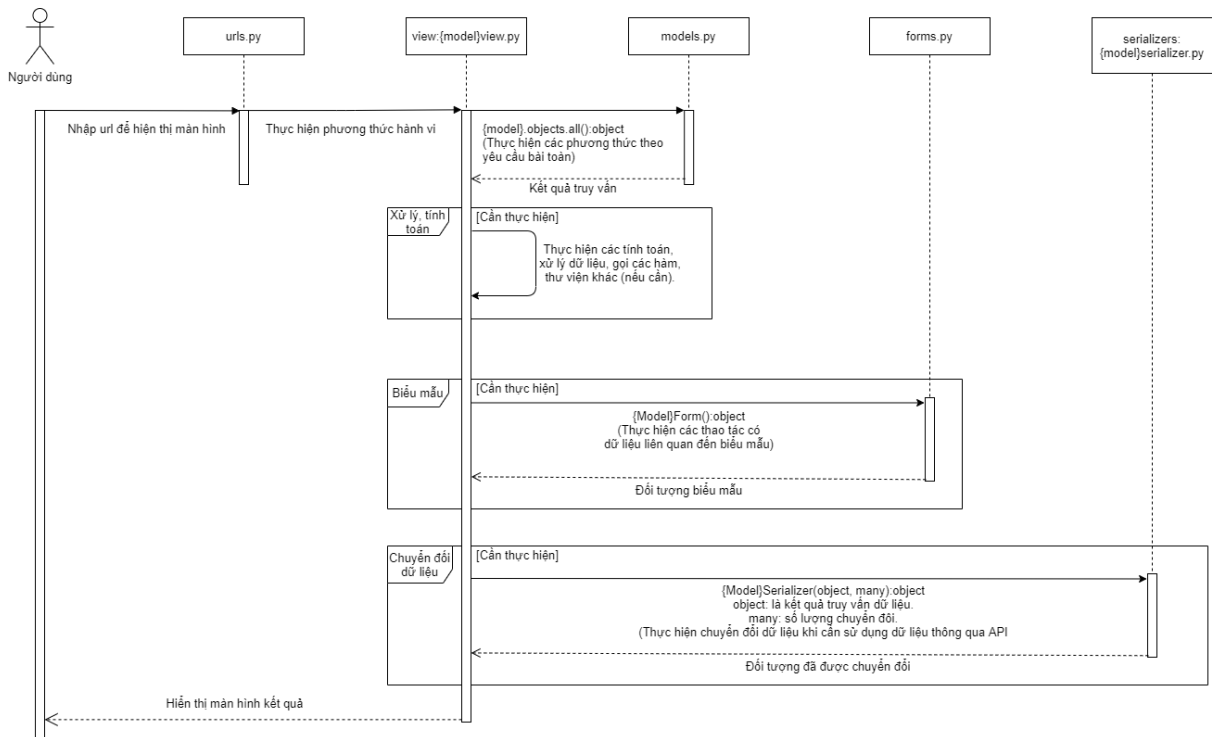
View ở trong django có thể hiểu là controller ở các mô hình hay sử dụng. View gọi đến các model để lấy dữ liệu của các đối tượng từ cơ sở dữ liệu và xử lý dữ liệu. Hơn nữa nhiệm vụ chính của view là để hiển thị các trang giao diện.

Template là các trang giao diện, thông qua giao diện thì người dùng có thể tương tác chính với hệ thống. Người dùng gửi yêu cầu, gửi thông tin đến hệ thống.

Ngoài ra tùy vào cách thức thiết kế và phương pháp giải quyết bài toán cụ thể thì cũng có thể sử dụng các kiến trúc như 2-tier, 3-tier, n-tier và các pattern phổ biến.

1.1.2. Cách thức phát triển

1.1.2.1. Biểu đồ tuần tự



Hình 1.1. Biểu đồ tuần tự mô hình hoạt động.

1.1.2.2. Chi tiết

Đường dẫn: Thông qua các đường dẫn của màn hình và API thì các ở file urls.py thì có cơ chế tự ánh xạ từ đường dẫn đến phương thức ở phần views.py.

Xử lý và điều hướng: Để tránh những lỗi không đáng có và làm cho cấu trúc của hệ thống sạch sẽ, rõ ràng hơn thì mỗi đường dẫn sẽ ánh xạ đến một và chỉ một phương thức. Mỗi phương thức sẽ nhận xử lý chính các dữ liệu được gửi kèm theo yêu cầu của người dùng và mỗi yêu cầu xử lý thì trả về một phản hồi, phản hồi này có thể chỉ là dữ liệu dạng thuần hoặc là trang giao diện để hiển thị cho người dùng. Trong phương thức xử lý thì có thể thực hiện lấy dữ liệu có trong cơ sở dữ liệu hoặc là thực hiện thông qua biểu mẫu để tạo và gửi dữ liệu. Và có thể ra lệnh để chuyển đổi dữ liệu để hiển thị phù hợp với dạng của API.

Nhưng còn tùy vào điều kiện và yêu cầu của bài toán thì trong các phương thức này còn có thể thực hiện thêm các hành vi khác để giải quyết yêu cầu.

1.2. Cách thức sử dụng Framework Django

1.2.1. Đối tượng

1.2.1.1. Dữ liệu của thuộc tính

Các kiểu dữ liệu: Để khai báo và sử dụng các thuộc tính có trong model của django thì cần import models từ thư viện django.db và mỗi đối tượng thì cần chuyển đối tượng Model có trong models vào.

Các kiểu đối tượng có trong models như là ForeignKey, AutoField, CharField, DateField, ManyToManyField, OneToOneField, IntegerField, TextField, PositiveIntegerField, BooleanField.

Kiểu dữ liệu	Mô tả
ForeignKey	Chỉ định đây là khóa ngoại tham chiếu đến khóa chính của đối tượng khác.
AutoField	Đây là thuộc tính linh hoạt không cần phải định trước kiểu lưu trữ dữ liệu của thuộc tính. mà sẽ được chỉ định trong quá trình sử dụng. Cũng có thể sử dụng thuộc tính <code>primary_key=True</code> để cài đặt đây là khóa chính.
CharField	Chỉ định kiểu dữ liệu là văn bản. Khi sử dụng kiểu dữ liệu này thường sử dụng thêm với thuộc tính <code>max_length</code> để giới hạn chiều dài của văn bản.
DateField	Chỉ định kiểu dữ liệu là kiểu thời gian. Định dạng mặc định của kiểu dữ liệu này là yyyy-mm-dd. Để cài đặt giá trị mặc định là giá trị thời gian tại thời điểm thực hiện thì sử dụng thuộc tính <code>auto_now_add=True</code> .
ManyToManyField	Chỉ định đây là mối quan hệ của bảng hiện tại với bảng được tham chiếu là mối quan hệ nhiều nhiều. Tùy vào trường hợp bài toán đặc biệt thì mới sử dụng đến kiểu giá trị này.
OneToOneField	Chỉ định đây là mối quan hệ của bảng hiện tại với bảng được tham chiếu là mối quan hệ một một.
IntegerField	Chỉ định kiểu dữ liệu lưu ở thuộc tính là kiểu số nguyên.
TextField	Chỉ định kiểu dữ liệu ở thuộc tính là văn bản. TextField thì không giới hạn chiều dài của nội dung văn bản còn CharField thì bị giới hạn chiều dài nội dung văn bản.
PositiveIntegerField	Chỉ định kiểu dữ liệu lưu của thuộc tính là kiểu số nguyên dương.
BooleanField	Chỉ định kiểu dữ liệu lưu của thuộc tính là giá trị đúng và sai.

Bảng 1.1. Danh sách các kiểu dữ liệu trong Models.

1.2.1.2. Mối quan hệ

Khóa chính: Để khai báo đối tượng thì trong tệp đối tượng thì cần sử dụng thư viện `django.db` để nhập vào đối tượng `models` sử dụng cho việc định nghĩa ra các đối tượng.

```
from django.db import models

class Units(models.Model):
    unitID = models.AutoField(primary_key=True)
    unitName = models.CharField(max_length=100)
    unitDescription = models.CharField(max_length=500, default="", null=True)

    def __str__(self):
        return self.unitName
```

Hình 1.2. Đối tượng có chứa khóa chính.

Ở đối tượng `Trường` thì có định nghĩa `unitID` là khóa chính và đối tượng này không tham chiếu đến các đối tượng khác. Ngoài ra nên định nghĩa lại phương thức `__str__` để tiện cho việc phát triển và rõ ràng khi hiển thị thông tin dữ liệu của đối tượng. Khi không định nghĩa lại phương thức này thì mặc định sẽ là thông tin dữ liệu của khóa chính hoặc là địa chỉ vị trí lưu trữ (trong trường hợp không có khóa chính).

Khóa ngoại: Định nghĩa một đối tượng khác như bình thường nhưng khóa ngoại tham chiếu đến đối tượng khác thì cần sử dụng kiểu dữ liệu `ForeignKey` để chỉ ra đối tượng tham chiếu đến là đối tượng nào. Kiểu dữ liệu của thuộc tính sẽ phụ thuộc vào khóa chính mà nó tham chiếu đến.

```
class Majors(models.Model):
    majorID = models.AutoField(primary_key=True)
    majorName = models.CharField(max_length=100)
    majorDescription = models.CharField(max_length=255, default="", null=True)
    unit = models.ForeignKey(Units, on_delete=models.CASCADE)

    def __str__(self):
        return self.majorName
```

Hình 1.3. Đối tượng sử dụng khóa ngoại tham chiếu.

Khi truy xuất dữ liệu truy vấn thông qua các đối tượng thì từ khóa tham chiếu có thể truy vấn đến dữ liệu của được tham chiếu mà không cần sử dụng các phương thức tự định nghĩa mà thông trực tiếp qua thuộc tính.

CASCADE	Khi đối tượng này bị xóa thì tất cả các đối tượng có tham chiếu đến đối tượng này thì đều bị xóa hết.
PROTECT	Không cho phép xóa đối tượng này nếu còn tồn tại các đối tượng khác đang tham chiếu đến đối tượng này.
SET_NULL	Đặt giá trị tham chiếu là NULL để khi các đối tượng khác tham chiếu đến và nhận biết là dữ liệu đã bị xóa.
SET_DEFAULT	Đặt giá trị tham chiếu là giá trị mặc định khi bị xóa.

SET	Đặt giá trị nhất định mà do người lập trình quyết định.
DO_NOTHING	Không thực hiện thao tác khi xóa và để thực hiện ở cấp cơ sở dữ liệu.

Bảng 1.2. Bảng các giá trị của thuộc tính on_delete.

1.2.2. Xử lý logic

1.2.2.1. Dữ liệu liên quan đến các đối tượng

Select: Thông qua đối tượng thì sử dụng phương thức `all()` để truy vấn dữ liệu trong bảng tương ứng.

```
units = Units.object.all()
```

Để thêm điều kiện truy vấn bản ghi thì sử dụng phương thức `filter()`.

```
units = Units.object.filter(unitID=1)
```

Ngoài ra có thể truy vấn dữ liệu bằng ID trực tiếp thông qua phương thức `get()`.

```
unit = Units.objects.get(pk=1)
```

Sử dụng điều kiện sắp xếp khi truy vấn thì sử dụng phương thức `order_by()`.

```
units = Units.object.all().order_by("unitID")
```

Sắp xếp giảm dần thì sử dụng thêm phương thức `reverse()`.

```
units = Units.object.all().order_by("unitID").reverse()
```

Update: Khi đã sử dụng truy vấn `get` để lấy ra bản ghi cụ thể thì có thể cập nhật trực tiếp bằng giá trị của thuộc tính, sau đó thực hiện lưu lại dữ liệu bản ghi đó. Hành động như vậy thì đã là thao tác tự cập nhật dữ liệu cho bản ghi `unit = Units.objects.get(pk=1)`.

```
unit.unitName = "Đại học Kinh tế"
unit.save()
```

Ngoài ra django còn cung cấp thêm phương thức `update()` để cập nhật trực tiếp dữ liệu.

```
unit = Units.objects.filter(unitID=1).update(unitName="Đại học Kinh tế")
```

Delete: Sử dụng phương thức delete() để thực hiện xóa bản ghi tương ứng.

```
Units.objects.get(pk=1).delete()
```

Hoặc

```
units = Units.objects.filter(unitName="Đại học Công nghệ")
```

```
units.delete()
```

1.2.2.2. Các thư viện khác

Thư viện serializer: Serialization là một quá trình chuyển đổi dữ liệu thuần của đối tượng được lưu trong cơ sở dữ liệu sang định dạng phù hợp để xử lý, hiển thị hơn. Ngoài ra nó cũng làm quá trình ngược lại là biến đổi dữ liệu tính toán thành dữ liệu trạng thái ban đầu. Đối tượng này được định nghĩa ở trong thư viện rest_framework nên cần truyền vào khi sử dụng. Dưới đây là định nghĩa ra phương thức để biến đổi dữ liệu.

```
from rest_framework import serializers
from mainApp.models import Units

class UnitSerializer(serializers.ModelSerializer):
    class Meta:
        model = Units
        fields = '__all__'
```

Hình 1.4. Đối tượng Unit Serialize.

Ở trong luồng xử lý khi muốn nhận được dữ liệu đã qua xử lý thì truyền đối tượng chứa dữ liệu vào trong đối tượng chuyển đổi này và gửi kèm để hiển thị trên giao diện.

```
def unit_getList(request):
    if request.method == 'GET':
        unitList = Units.objects.all()
        unitSerializer = UnitSerializer(unitList, many = True)
        return Response(unitSerializer.data)
```

Hình 1.5. Hàm trả dữ liệu dạng đã biến đổi.

Thư viện forms: Thư viện forms là tiện ý mở rộng để hỗ trợ quá trình biểu diễn biểu mẫu đến thực hiện xác thực và cập nhật dữ liệu thông qua các đối tượng form.

```

from django import forms
from .models import Units

class UnitForm(forms.ModelForm):
    class Meta:
        model = Units
        fields = '__all__'
        labels = {
            'unitName': 'Tên trường',
            'unitDescription': 'Thông tin chi tiết'
        }

    def __init__(self, *args, **kwargs):
        super(UnitForm, self).__init__(*args, **kwargs)
        self.fields['unitDescription'].required = False
        self.fields['unitDescription'].initial = 'Chưa có thông tin'

```

Hình 1.6. Đối tượng biểu mẫu.

Trong luồng xử lý nhận thông tin đối tượng thông qua biểu mẫu thì dễ dàng sử dụng đối tượng biểu mẫu này cả trong trường hợp thêm mới và cập nhật.

```

def unit_form(request, unit_id=0):
    try:
        if request.method == 'GET':
            if unit_id == 0:
                unitForm = UnitForm()
            else:
                unit = Units.objects.get(pk=unit_id)
                unitForm = UnitForm(instance=unit)
            return TemplateResponse(request, 'adminuet/unitform.html', {'form': unitForm})
        else:
            if unit_id == 0:
                unitForm = UnitForm(request.POST)
            else:
                unit = Units.objects.get(pk=unit_id)
                unitForm = UnitForm(request.POST, instance=unit)
            if unitForm.is_valid():
                unitForm.save()
            return redirect('/adminuet/unit/')
    except Exception:
        return redirect('/adminuet/unit/')

```

Hình 1.7. Phương thức xử lý tạo mới và cập nhật.

Thư viện auth decorators: Sử dụng thư viện này để xác thực phiên làm việc bắt đầu bằng hành động đăng nhập. Thư viện này cung cấp các điều kiện tiên quyết là đăng nhập trước khi thực hiện nội dung đoạn xử lý phía dưới bằng từ khóa `@login_required`.

```

@login_required(login_url='/login/')
def unitPagination_page(request, num=1, limit=10):
    return TemplateResponse(request, 'adminuet/unit.html', {'page': num, 'limit': limit})

```

Hình 1.8. Phương thức hiển thị giao diện có kiểm tra đăng nhập.

1.2.2.3. Dữ liệu hiển thị

Để có thể hiển thị trang giao diện sau quá trình xử lý logic thì sử dụng phương thức `TemplateResponse()` để hiển thị trang giao diện và có thể gửi dữ liệu để hiển thị bằng đối tượng chuyển vào dạng json. Như khi sử dụng phương thức `TemplateResponse()` để hiển thị trang giao diện thì có thể gửi kèm đối số context. Ở trang giao diện có thể trực tiếp sử dụng dữ liệu của biến bên trong đối số context.

```
def unitPagination_page(request, num=1, limit=10)
    context = {
        'page': num,
        'limit': limit
    }
    return TemplateResponse(request, 'adminuet/unit.html', context = context)
```

Hình 1.9. Phương thức trả về số trang về gửi hạn bản ghi để hiển thị.

Để chuyển hướng hiển thị một trang thì có thể sử dụng phương thức `HttpResponseRedirect()` và chuyển vào đó là đường dẫn của trang đó.

1.2.3. URL

1.2.3.1. Đường dẫn thành phần

Trong thư mục chính của dự án thì có một tệp để cấu hình đường dẫn của toàn bộ dự án đó là `urls.py`. Trong tệp này định nghĩa các `urlpatterns` chính là đường dẫn sau tên miền của dự án. Mỗi đường dẫn được coi là một `path`. Trong `path` chỉ định rõ là đường dẫn là gì, nơi nào nhận xử lý hiển thị khi mà người dùng gọi đến đường dẫn. Ngoài ra mỗi đường dẫn còn có thể đặt tên để dễ dàng cho việc tái sử dụng và thuận tiện khi gọi ở giao diện.

```
urlpatterns = [
    path('home', views.index, name='home'),
]
```

1.2.3.2. Đường dẫn chi tiết

Để đường dẫn chính có thể tham chiếu đến đường dẫn của các thành phần khác thì sử dụng phương thức `include()` thông báo cho `django` ra đó là bao gồm toàn bộ đường dẫn mà thành phần được chỉnh định trong `include()` định nghĩa ra. Khi muốn đường dẫn chính là đường dẫn bắt đầu của toàn bộ một thành phần trong dự án thì thực hiện như sau:

```
path('', include('mainApp.urls'));
```

1.2.4. Giao diện

1.2.4.1. Xây dựng khung giao diện

Tạo file `html`, xây dựng bố cục các file `template` để kế thừa và sử dụng lại các thành phần. Xây dựng các trang mã giao diện cơ bản và các trang chính có thể kế thừa lại các thành phần của các thành phần `html` khác. Sử dụng cú pháp `extend` để kế thừa lại những nội dung từ file `html`.

Ngoài ra còn có thể tự định nghĩa những khối mà hiển thị khác và các trang khác khi cần chỉ cần định nghĩa lại khối đó là có thể hiển thị nội dung tương tự đầy đủ.

```
{% block content% } {% endblock % }
```

1.2.4.2. Xử lý dữ liệu

Sử dụng các ký hiệu python trong file.html như `{{ }}` `{% %}`. Để sử dụng các tệp định nghĩa giao diện hay định nghĩa hành động thao tác thì sử dụng.

```
{% load static % }
```

Ngoài ra trong tệp html còn có những đường dẫn, url mà url đó là những đường dẫn được định nghĩa trong file urls.py và được định danh với name thì trong file html sử dụng.

```
{% url "units" %}
```

Cú pháp này sẽ được hiểu là liên kết của thẻ liên kết là giá trị của url có tên là units. Ngoài ra django có cung cấp các câu lệnh rẽ nhánh và vòng lặp để hỗ trợ các thao tác logic cơ bản, các cú pháp như:

```
{% if condition % }{% endif % }
```

```
{% for i in list % }{% endfor % }
```

1.2.5. Phân quyền

Trong Django có hỗ trợ xử lý yêu cầu và phản hồi của người dùng trước khi máy chủ nhận xử lý dữ liệu chính và tên gọi của nó middleware. Khi nó nhận yêu cầu thì có thể chèn thêm thông tin cần thiết như là thông tin người dùng, thông tin phiên làm việc. Khi tự định nghĩa ra một middleware thì cần khai báo nó cả ở trong file setting.py là file cấu hình chính của dự án.

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
    'mainApp.middlewares.FunctionMiddleware',  
]
```

Hình 1.10. Các thư viện sử dụng và định nghĩa middleware.

Cấu trúc các phương thức có thể tự ghi đè cho phù hợp mới yêu cầu của bài toán.

Phương thức	Mô tả
<code>__init__</code>	Thực hiện nạp dữ liệu và khởi chạy duy nhất lần đầu tiên khi thực hiện bắt đầu chương trình.
<code>__call__</code>	Thực hiện gọi khi mỗi lần trước khi thực hiện yêu cầu và phản hồi
<code>process_exception</code>	Thực hiện tiến trình khi xảy ra ngoại lệ để thực hiện.
<code>process_template_response</code>	Thực hiện sau quá trình phản hồi để hiển thị phản hồi

Bảng 1.3. Các phương thức trong middleware.

1.2.6. API

Để sử dụng API thì cũng phải sử dụng serializer để chuyển đổi dữ liệu. Để viết api thì trong luồng xử lý dữ liệu thì cần sử dụng đối tượng `api_view` của thư viện `rest_framework`. Khi định nghĩa phương thức sử dụng cho API thì sẽ theo đoạn code mẫu ở dưới.

```
from rest_framework.decorators import api_view

@api_view(['GET'])
def unit_getlist(request):
    if request.method == 'GET':
        unitlist = Units.objects.all()
        unitserializer = UnitSerializer(unitlist, many = True)
        return Response(unitserializer.data)
```

Hình 1.11. Hàm xử lý phương thức GET của API

Sau đó định dạng thiết kế đường dẫn trong file urls.

```
path('unit/page<int:num>/limit<int:limit>/', unitview.unit_getlist, name = 'unitpage'),
```

Hình 1.12. Đường dẫn của API.

Với phương thức post thì cần lấy dữ liệu từ yêu cầu của người dùng và thực hiện xử lý với dữ liệu đó như yêu cầu của bài toán đề ra. Nếu không cần kiểm tra hay xử lý thì có thể thực hiện lưu ngay thông qua đối tượng `UnitSerializer()`.

```
@api_view(['POST'])
def snippet_list(request):
    if request.method == 'POST':
        serializer = UnitSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

Hình 1.13. Hàm xử lý phương thức POST của API.

Chương 2. Phân tích thiết kế hệ thống

2.1. Phân tích

2.1.1. Các tác nhân hệ thống

2.1.1.1. Các hạng mục người quản lý có quyền thao tác

Trong hệ thống cho phép nhiều trường hoạt động độc lập với nhau nên vì vậy vài trò người quản lý là yếu tố để quản lý các công việc, danh mục liên quan đến công tác tư vấn điểm số và môn học cho sinh viên. Cụ thể các đầu mục mà người quản lý cấp trường có liên quan tới là các sinh viên của trường mà quản lý, các ngành đào tạo, các khóa đào tạo, khung chương trình đào tạo tương ứng lên quan tới các ngành đào tạo, các môn học có trong chương trình giảng dạy. Ngoài ra còn những ý tổ quản lý khác như quản lý điểm, năm học, kỳ học, các lớp để có thể tuân theo tuần tự của nghiệp vụ quản lý tại trường Đại học.

Các tài khoản có chức năng là quản trị viên chỉ có phạm vi hoạt động liên quan đến hoạt động của đơn vị trường trực thuộc hiện tại, không thể sử dụng các dữ liệu về sinh viên, chương trình học của các trường khác có cùng trên hệ thống.

2.1.1.2. Yếu tố ảnh hưởng đến người quản lý

Người quản lý cấp trường cần là một tài khoản do quản trị viên của hệ thống (người quản trị toàn bộ hệ thống) cung cấp. Tất cả các quyền, phạm vi hoạt động của tài khoản người quản lý là do quản trị hệ thống định ra và có thể thay đổi. Người quản lý cấp trường chỉ có quyền tương tác với tài khoản thuộc trường trực thuộc và tài khoản người quản lý không thể thay đổi được quyền của tài khoản quản trị hệ thống và các tài khoản quản lý cùng cấp.

2.1.2. Sinh viên

2.1.2.1. Các hạng mục sinh viên có thể thao tác

Hệ thống cung cấp cho sinh viên thông tin về điểm số các môn học thông qua các thống kê, các dự đoán về số điểm và đưa ra đề xuất môn học phù hợp để sinh viên có kết quả học tập tốt nhất. Vì thế sinh viên cần phải xem được xác thống kê về môn học, xem được kết quả điểm mà hệ thống dự báo được nếu mà sinh viên tham gia môn học và xem được các đề xuất các môn học phù hợp khả năng của bản thân và thời điểm phù hợp để học.

Các con số thống kê sinh viên có thể cần thiết như là thống kê về phổ điểm của các môn học có trong chương trình đào tạo của các khóa, thống kê điểm trung bình của môn học qua các năm học, thống kê về gpa theo khóa và ngành đào tạo, thống kê gpa theo từng kỳ của các khóa và ngành đào tạo và thống kê điểm của bản thân sinh viên.

2.1.2.2. Yêu cầu thao tác với thông tin tư vấn của sinh viên

Các thông tin dự đoán về điểm số môn học và môn học đề xuất thì hệ thống sẽ cần tiếp nhận thông tin về yêu cầu dự đoán điểm của môn học nào hay cần chọn số

lượng môn và cách gợi ý khi muốn nhận các môn học đề xuất. Các thông tin về môn học, điểm số dự báo mà hệ thống đề xuất qua những lần khác nhau thì sẽ có sai số.

2.1.3. Quản trị

2.1.3.1. Việc phân quyền chính của hệ thống

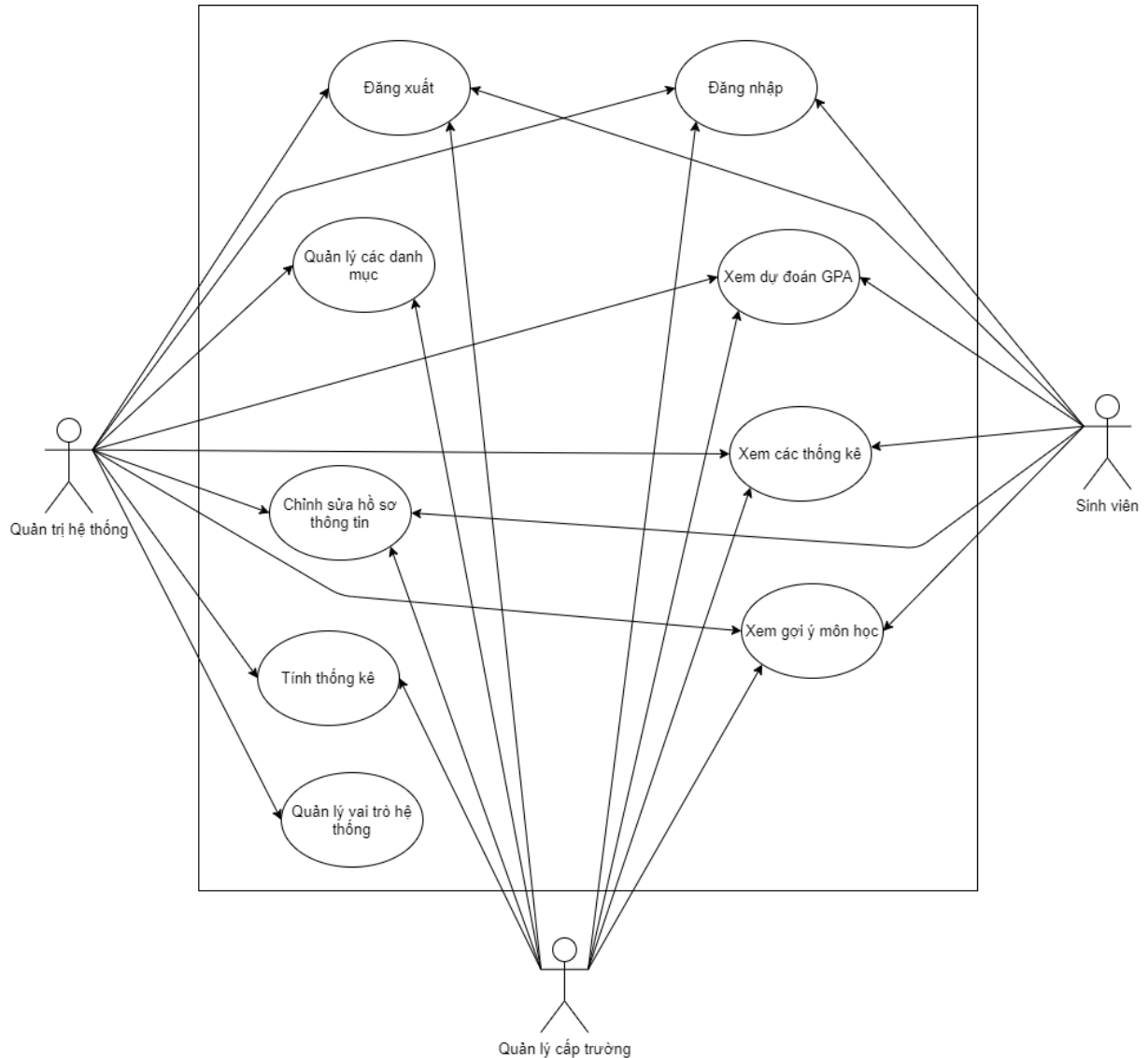
Quản trị hệ thống là vai trò quản lý cao nhất của hệ thống. Quản trị hệ thống có quyền thao tác với tất cả phạm vi của tất cả các trường có trong hệ thống. Từ việc quản lý tất cả các môn học, chương trình đào tạo...

2.1.3.2. Cập nhật dữ liệu liên quan đến tư vấn học tập

Hệ thống cung cấp kết quả tư vấn dựa trên những điểm số về các môn học của sinh viên đã tham gia môn học vì thế quản trị hệ thống cần được vào dữ liệu điểm số liên quan đến môn học, sinh viên theo khóa và theo ngành.

2.2. Ca sử dụng

2.2.1. Sơ đồ chính



Hình 2.1. Tổng quan toàn bộ ca sử dụng.

2.2.2. Chi tiết ca sử dụng

2.2.2.1. Quản lý các danh mục

Mô tả tóm tắt: Người dùng quản trị hệ thống cập nhật thông tin về các cơ sở đào tạo, trường đại học, ngành đào tạo, môn học, chương trình đào tạo, kết quả học tập, khóa học, lớp, năm học, kỳ học, lựa chọn mô hình, vai trò hệ thống. Các thao tác là: thêm mới, cập nhật dữ liệu, xóa dữ liệu.

Luồng chính:

Mục đích	Hành động người dùng	Hệ thống phản hồi	Dữ liệu
----------	----------------------	-------------------	---------

1. Thêm mới	1.1. Nhấn vào nút Thêm mới	1.2. Hệ thống hiển thị màn hình có biểu mẫu để nhập thông tin của đối tượng cần thêm mới.	Dữ liệu người dùng nhập.
	1.3. Điền thông tin theo hướng dẫn và nhấn nút Thêm mới	1.4. Hệ thống thực hiện kiểm tra dữ liệu người dùng nhập vào và thực hiện thêm mới dữ liệu vào hệ thống. Khi thực hiện thành công thì hệ thống hiển thị màn hình danh sách dữ liệu.	Dữ liệu trong danh mục tương ứng.
2. Cập nhật	2.1. Nhấn nút Sửa ở dữ liệu muốn cập nhật dữ liệu	2.2. Hệ thống hiển thị màn hình có biểu mẫu chứa thông tin của dữ liệu muốn cập nhật.	Dữ liệu trong danh mục tương ứng.
	2.3. Cập nhật thông tin trong biểu mẫu và nhấn nút Cập nhật	2.4. Hệ thống thực hiện kiểm tra dữ liệu người dùng nhập vào và thực hiện cập nhật thông tin trong biểu mẫu vào hệ thống.	Dữ liệu trong hệ thống.
3. Xóa	3.1. Nhấn nút Xóa ở hàng muốn xóa.	3.2. Hệ thống hiển thị lên thông báo xác nhận chắc chắn muốn xóa.	
	3.2. Nhấn nút xác nhận xóa	3.4. Hệ thống thực hiện kiểm tra tồn tại rồi xóa dữ liệu.	ID của bản ghi trong hệ thống.

Bảng 2.1. Bảng mô tả các hành động trong luồng chính.

Luồng phụ:

Tại bước 1.4: Khi dữ liệu của người dùng nhập vào đã tồn tại trong hệ thống hoặc sai cú pháp thì hệ thống sẽ thông báo lỗi tương ứng để người dùng thao tác lại.

Tại bước 2.2: Khi bản ghi muốn cập nhật không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách dữ liệu của hạng mục và thông báo lỗi cho người dùng.

Tại bước 2.4: Khi dữ liệu của người dùng nhập đã tồn tại trong hệ thống hoặc sai cú pháp thì hệ thống sẽ thông báo lỗi tương ứng để người dùng thao tác lại. Khi bản ghi muốn cập nhật không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách dữ liệu của hạng mục và thông báo lỗi cho người dùng.

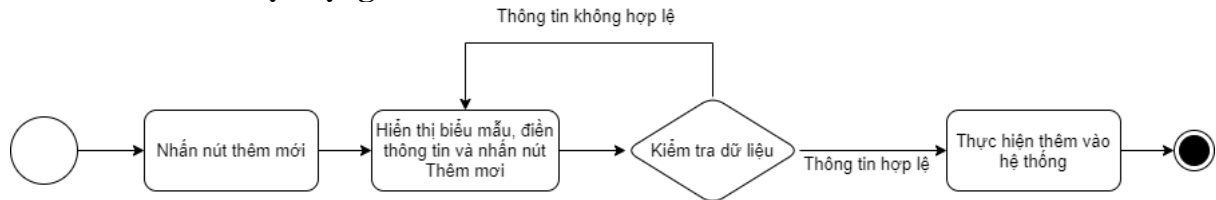
Tại bước 3.4: Khi bản ghi người dùng muốn xóa không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách các bản ghi của danh mục tương ứng và thông báo lỗi cho người dùng.

Yêu cầu đặc biệt: Người dùng bắt buộc là tài khoản quản trị hệ thống, quản lý cấp trường. Các quản lý cấp trường chỉ có thể thao tác trong phạm vi của trường mà tài khoản quản lý, không thể tác động đến những lại dữ liệu của trường khác.

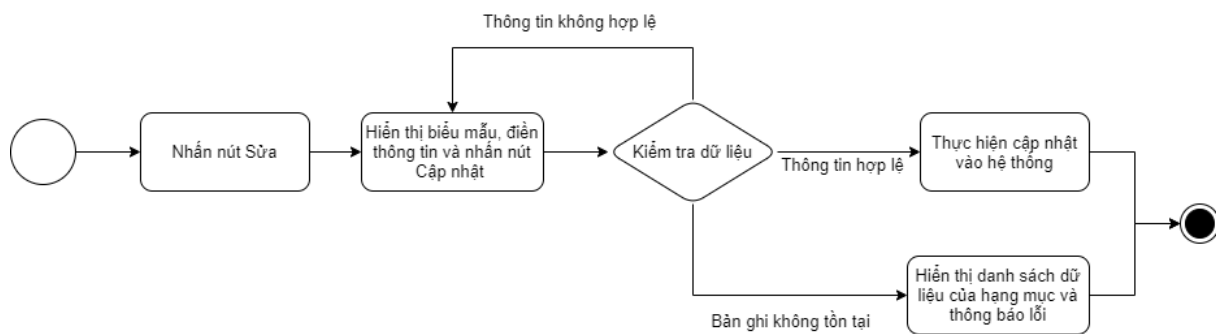
Điều kiện tiên quyết: Người dùng phải đăng nhập vào hệ thống.

Các vấn đề mở: Danh mục người dùng là danh mục đặc biệt do phụ thuộc vào hệ thống LDAP để xác thực sinh viên hay không nên việc cập nhật, tạo mới tài khoản sinh viên sẽ được tự động tạo ra dựa trên thông tin đăng nhập và đã được xác thực bởi LDAP.

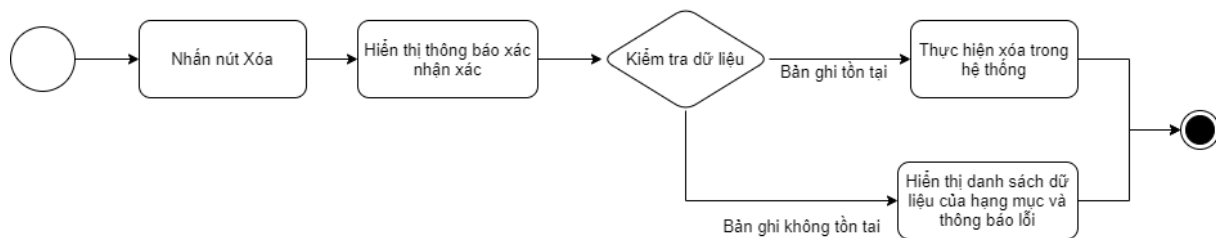
Biểu đồ hoạt động:



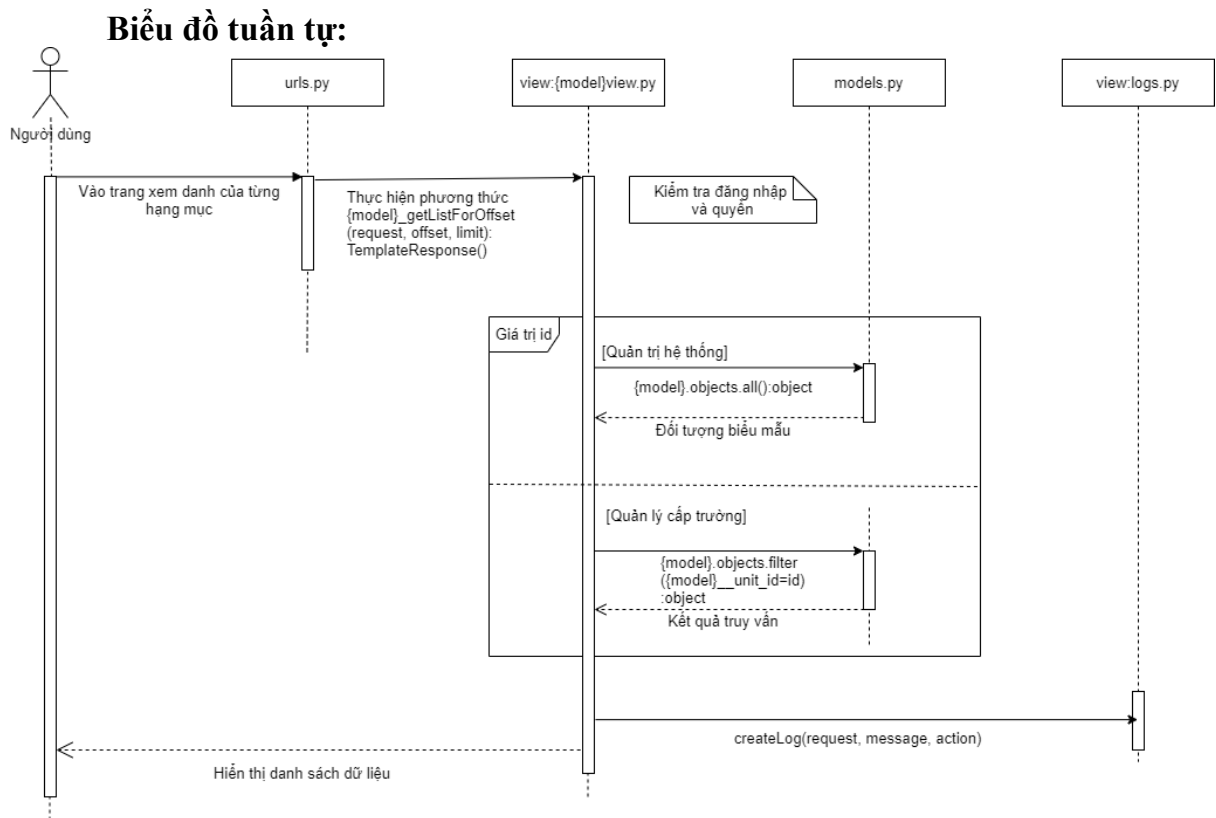
Hình 2.2. Biểu đồ các thao tác khi thực hiện thêm mới dữ liệu.



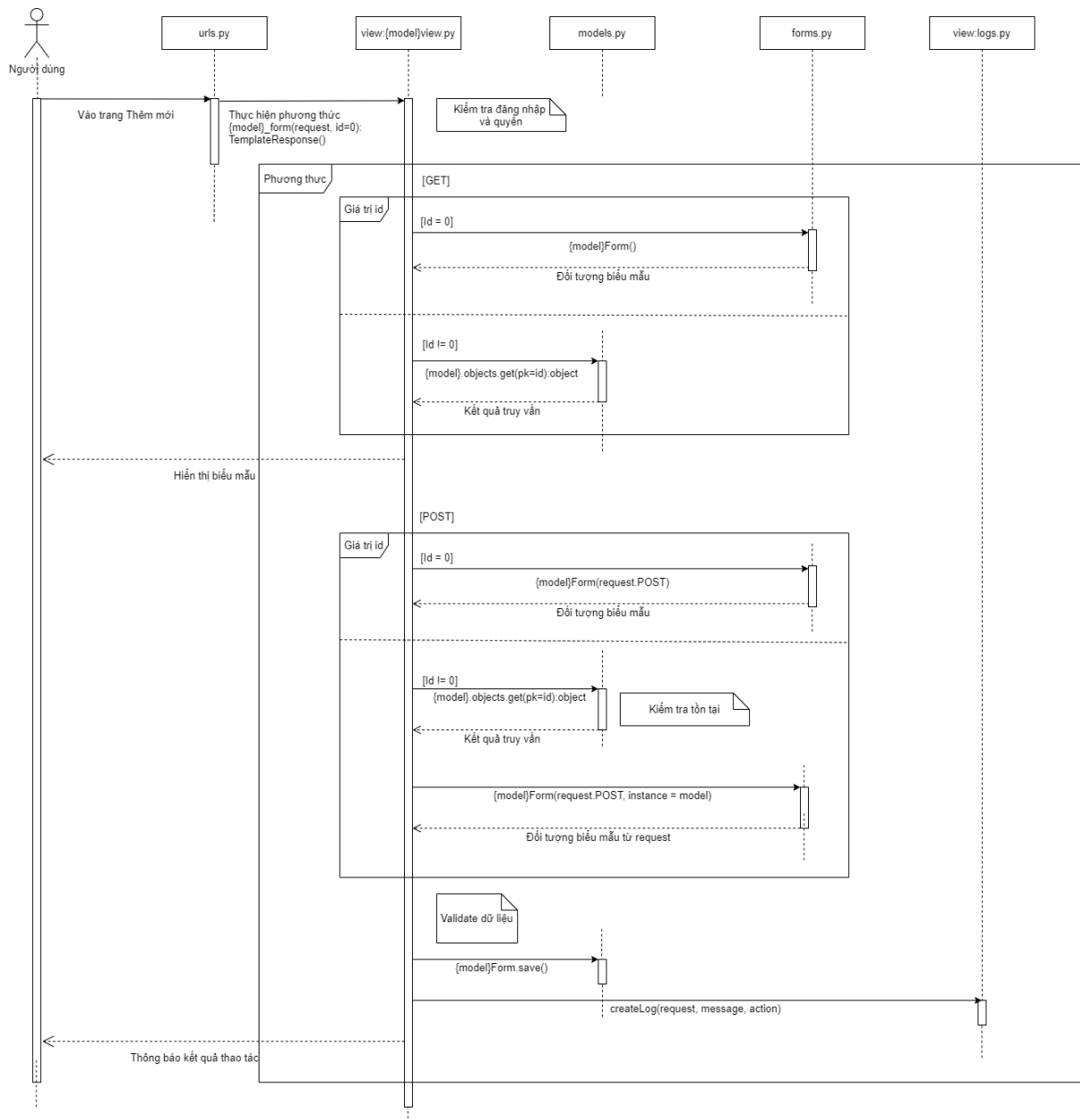
Hình 2.3. Biểu đồ các thao tác khi thực hiện cập nhật dữ liệu.



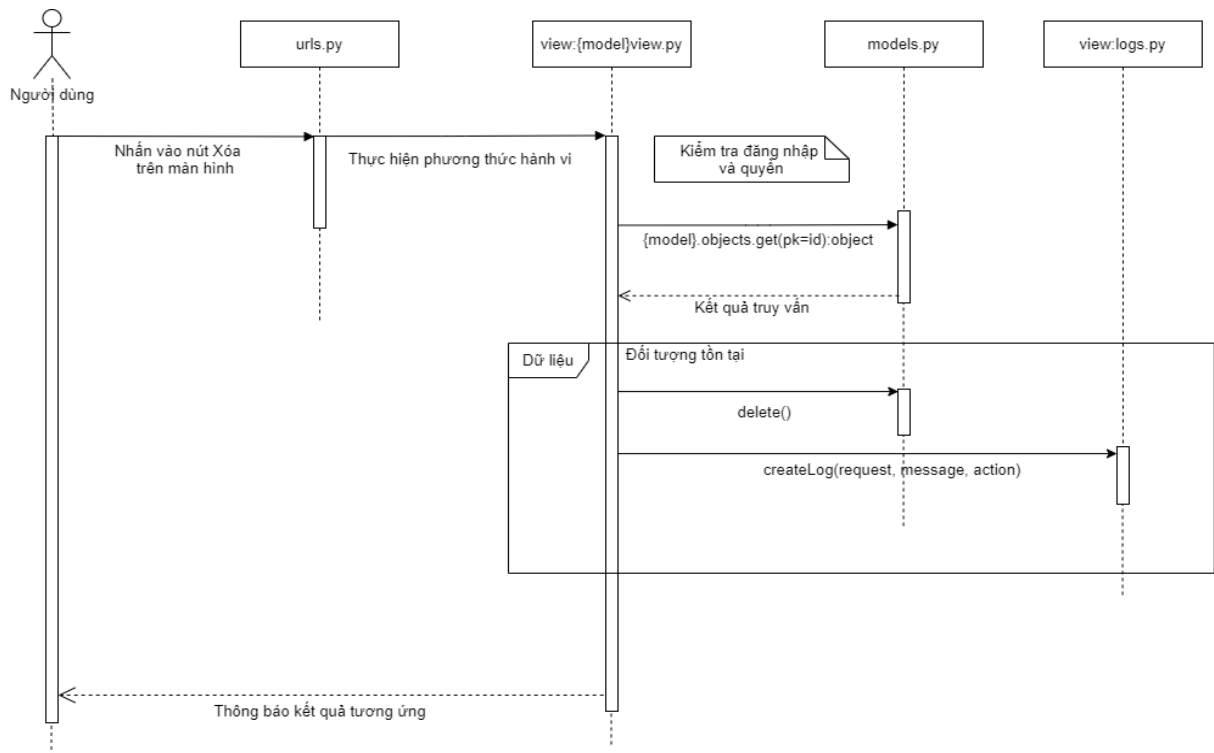
Hình 2.4. Biểu đồ các thao tác khi thực hiện xóa dữ liệu.



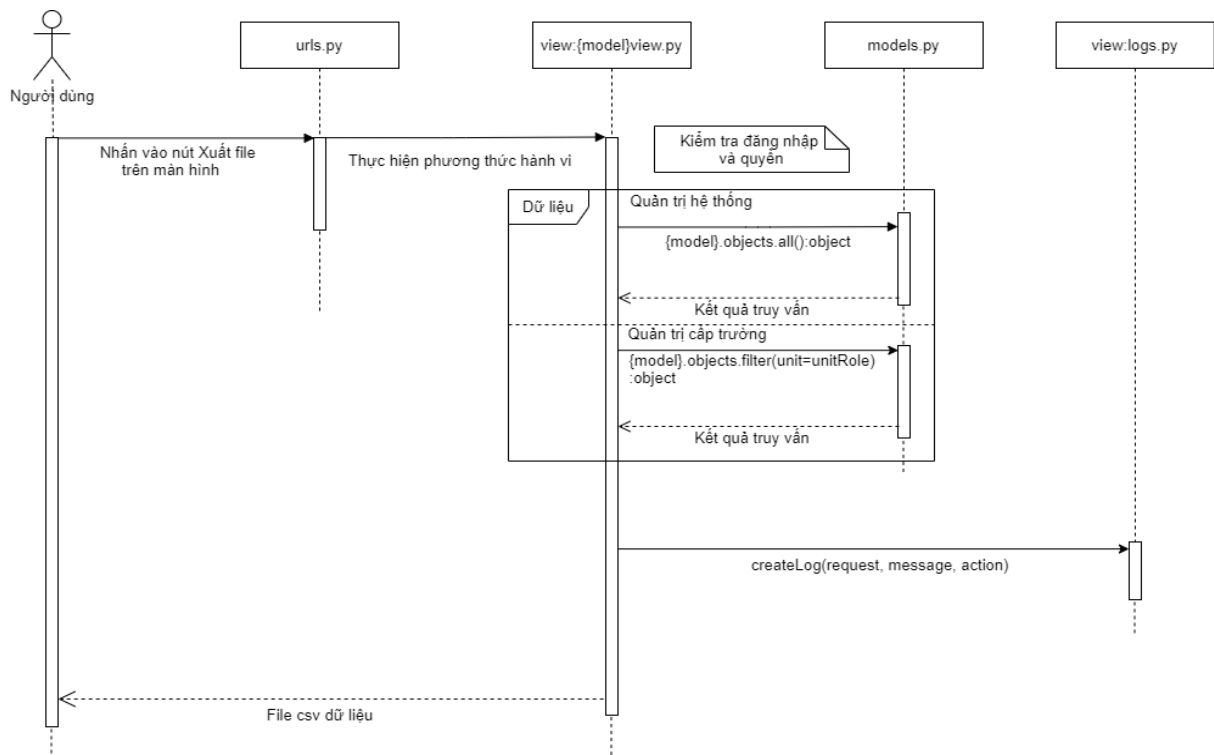
Hình 2.5. Biểu đồ tuần tự tính năng xem danh sách.



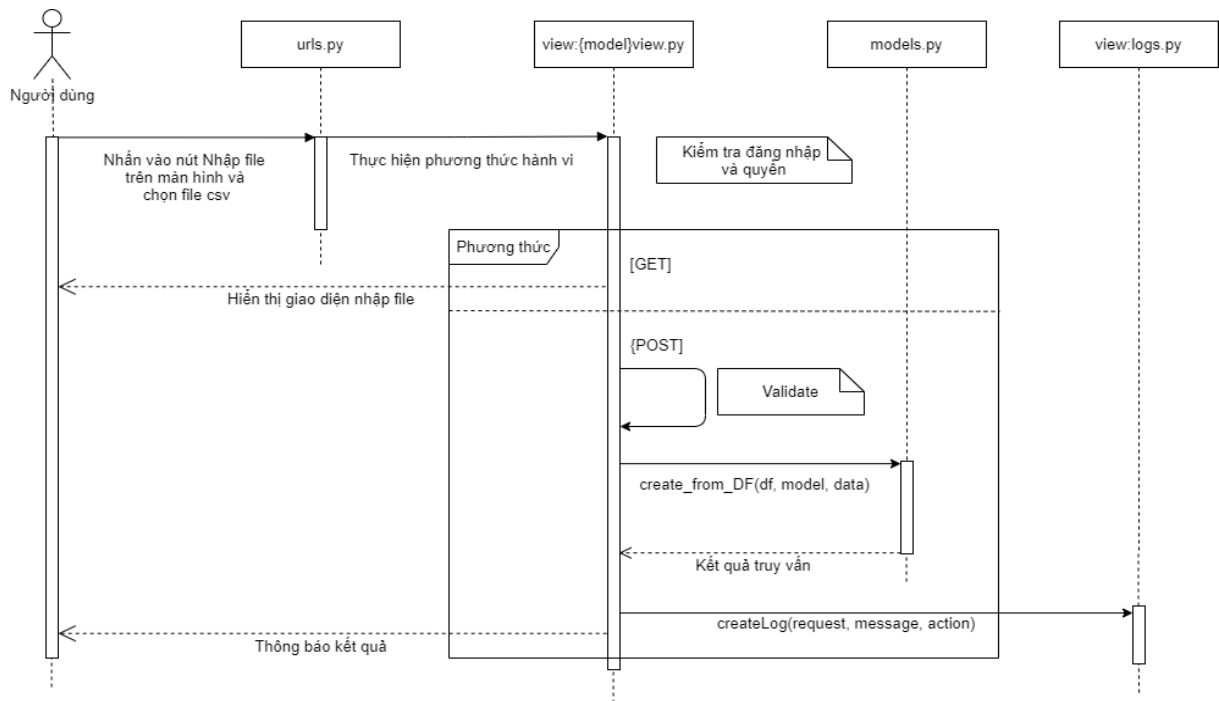
Hình 2.6. Biểu đồ tuần tự tính năng tạo mới và sửa nội dung.



Hình 2.7. Biểu đồ tuần tự tính năng xóa nội dung.



Hình 2.8. Biểu đồ tuần tự tính năng xuất file csv.



Hình 2.9. Biểu đồ tuần tự tính năng nhập file csv.

2.2.2.2. Quản lý chức năng hệ thống

Mô tả tóm tắt: Người dùng cập nhật thông tin về các chức năng hệ thống như: thêm mới chức năng, cập nhật thông tin chức năng, xóa chức năng khỏi hệ thống.

Luồng chính:

Mục đích	Hành động người dùng	Hệ thống phản hồi	Dữ liệu
1. Thêm mới	1.1. Nhấn vào nút Thêm mới.	1.2. Hệ thống hiển thị màn hình có biểu mẫu để nhập thông tin về năm học.	Dữ liệu người dùng nhập.
	1.3. Điền thông tin theo hướng dẫn và nhấn nút Thêm mới.	1.4. Hệ thống thực hiện kiểm tra dữ liệu người dùng nhập vào và thực hiện thêm mới chức năng. Khi thực hiện thành công thì hệ thống hiển thị màn hình danh sách chức năng.	Dữ liệu chức năng.
2. Cập nhật	2.1. Nhấn nút Sửa ở chức năng muốn cập nhật dữ liệu.	2.2. Hệ thống hiển thị màn hình có biểu mẫu chứa thông tin chức năng.	Dữ liệu chức năng.
	2.3. Cập nhật thông tin chức	2.4. Hệ thống thực hiện kiểm tra dữ liệu người dùng nhập vào và thực hiện	Dữ liệu chức năng.

	năng trong biểu mẫu và nhấn nút Cập nhật.	cập nhật thông tin trong biểu mẫu vào hệ thống.	
3. Xóa	3.1. Nhấn nút Xóa ở chức năng muốn xóa dữ liệu.	3.2. Hệ thống hiển thị lên thông báo xác nhận chắc chắn muốn xóa.	
	3.2. Nhấn nút xác nhận xóa.	3.4. Hệ thống thực hiện kiểm tra tồn tại rồi xóa chức năng.	Dữ liệu là id của chức năng trong hệ thống.

Bảng 2.2. Bảng mô tả các hành động trong luồng chính quản lý chức năng.

Luồng phụ:

Tại bước 1.4: Khi dữ liệu của người dùng nhập vào đã tồn tại trong hệ thống hoặc sai cú pháp thì hệ thống sẽ thông báo lỗi tương ứng để người dùng thao tác lại.

Tại bước 2.2: Khi chức năng đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách chức năng và thông báo lỗi cho người dùng.

Tại bước 2.4: Khi chức năng mà người dùng nhập đã tồn tại trong hệ thống hoặc sai cú pháp thì hệ thống sẽ thông báo lỗi tương ứng để người dùng thao tác lại. Khi kỳ học đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách chức năng và thông báo lỗi cho người dùng.

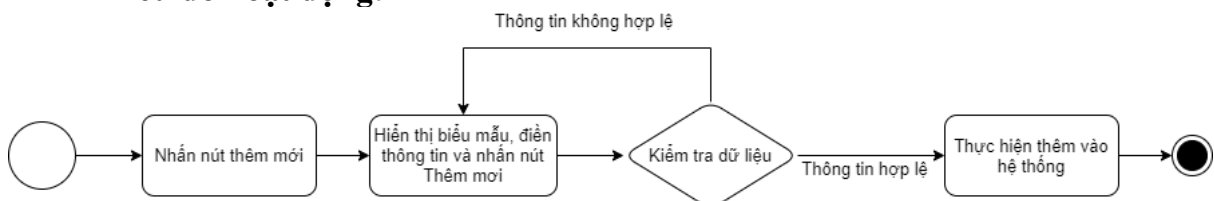
Tại bước 3.4: Khi chức năng mà người dùng muốn xóa không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách chức năng và thông báo lỗi cho người dùng.

Yêu cầu đặc biệt: Người dùng bắt buộc là tài khoản quản trị hệ thống.

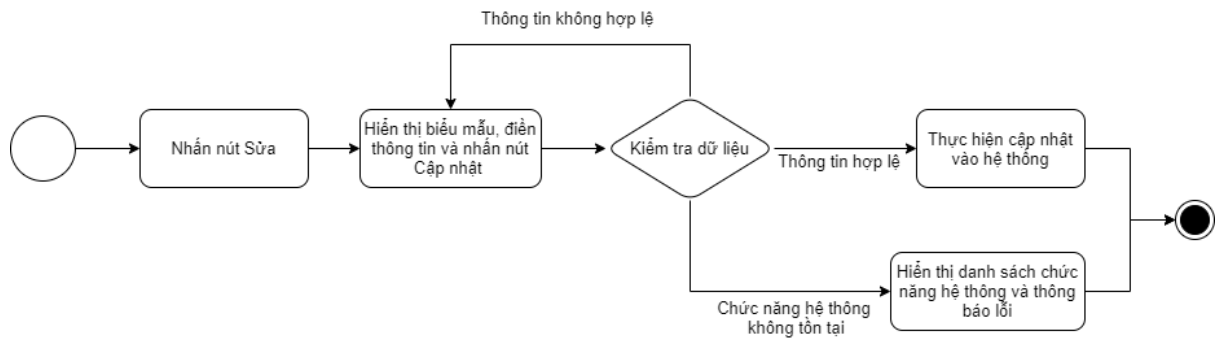
Điều kiện tiên quyết: Người dùng phải đăng nhập vào hệ thống.

Các vấn đề mở: Đây là đối tượng đặc biệt của hệ thống, trong quá trình xây dựng hệ thống đã cố định các tính năng và chức năng của hệ thống nên việc quản trị việc tuyệt đối không thể tác động đến các tính năng của hệ thống. Khi có sự thay đổi để dẫn tới sai khác trong các tính năng thì có thể gây lỗi không đáng có.

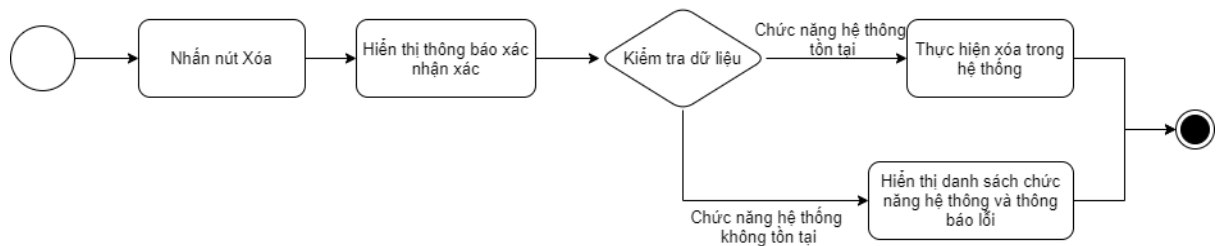
Biểu đồ hoạt động:



Hình 2.10. Biểu đồ các thao tác khi thực hiện thêm mới chức năng hệ thống.



Hình 2.11. Biểu đồ các thao tác khi thực hiện cập nhật chức năng hệ thống.



Hình 2.12. Biểu đồ các thao tác khi thực hiện thêm mới chức năng hệ thống.

Biểu đồ tuần tự: Tham chiếu các biểu đồ tuần tự của ca sử dụng Quản lý các danh mục.

2.2.2.3. Lựa chọn mô hình

Mô tả tóm tắt: Người dùng có thể lựa chọn mô hình cho cách thức dự đoán điểm và gợi ý môn học.

Luồng chính:

Mục đích	Hành động người dùng	Hệ thống phản hồi	Dữ liệu
1. Lựa chọn mô hình	1.1. Lựa chọn mô hình dự đoán, lựa chọn cài đặt phù hợp và nhấn nút Sử dụng	1.2. Hệ thống lưu lại mô hình và thông báo thành công	Dữ liệu người dùng nhập.

Bảng 2.3. Bảng mô tả các hành động trong luồng chính lựa chọn mô hình.

Luồng phụ: Không có.

Yêu cầu đặc biệt: Người dùng bắt buộc là tài khoản quản trị hệ thống.

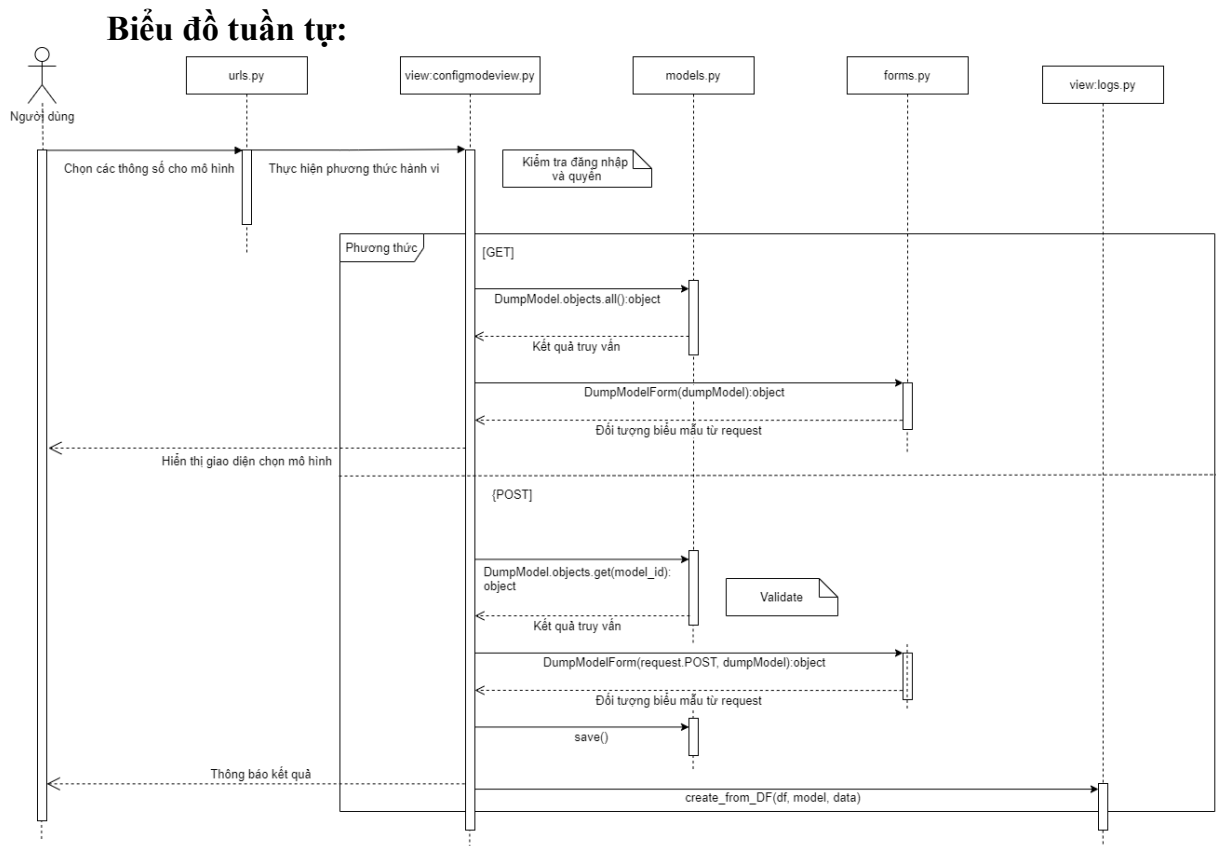
Điều kiện tiên quyết: Người dùng phải đăng nhập vào hệ thống.

Các vấn đề mở: Không có.

Biểu đồ hoạt động:



Hình 2.13. Biểu đồ các thao tác khi thực hiện lựa chọn mô hình.



Hình 2.14. Biểu đồ các thao tác khi thực hiện chọn mô hình cho hệ thống.

2.2.2.4. Thống kê

Mô tả tóm tắt: Người dùng có thể xem các thống kê như là về phổ điểm trung bình môn học của các khóa, điểm trung bình của môn học qua các năm, điểm GPA trung bình của khóa và ngành, điểm GPA của sinh viên.

Luồng chính:

Hành động người dùng	Hệ thống phản hồi	Dữ liệu
Lựa chọn các thông tin muốn xem thống kê	Hiển thị biểu đồ cho theo những thông tin người dùng chọn	Dữ liệu người dùng nhập. Dữ liệu trường, ngành, khóa, môn, năm học, sinh viên, môn học.

Bảng 2.4. Bảng mô tả các hành động trong luồng chính xem thống kê.

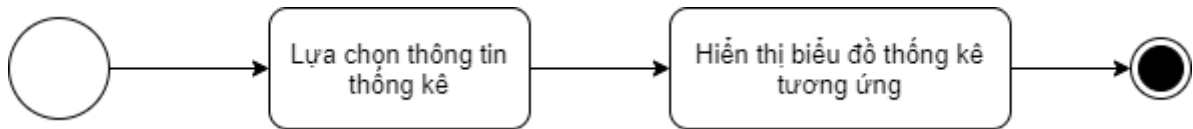
Luồng phụ: Khi người dùng nhập chưa đủ thông tin cho các thống kê thì các biểu đồ thống kê sẽ không hiển thị và sẽ có thông báo lỗi.

Yêu cầu đặc biệt: Không có.

Điều kiện tiên quyết: Người dùng phải đăng nhập vào hệ thống.

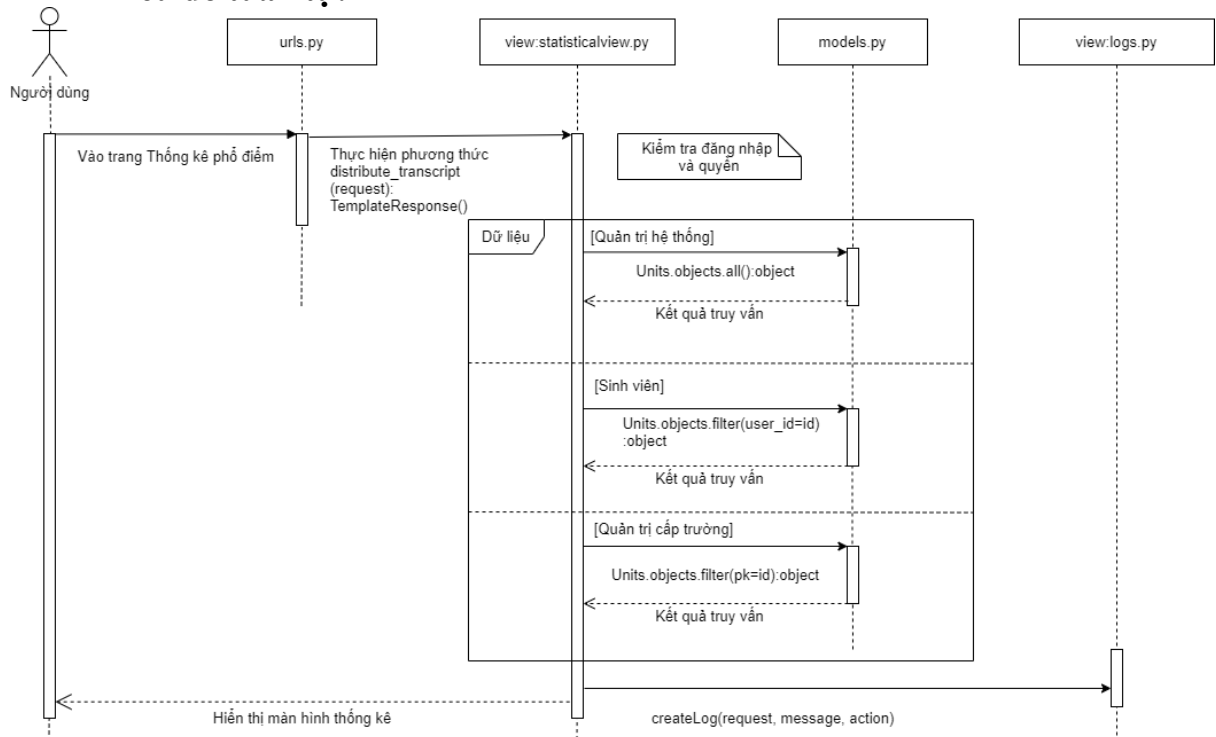
Các vấn đề mở: Không có.

Biểu đồ hoạt động:

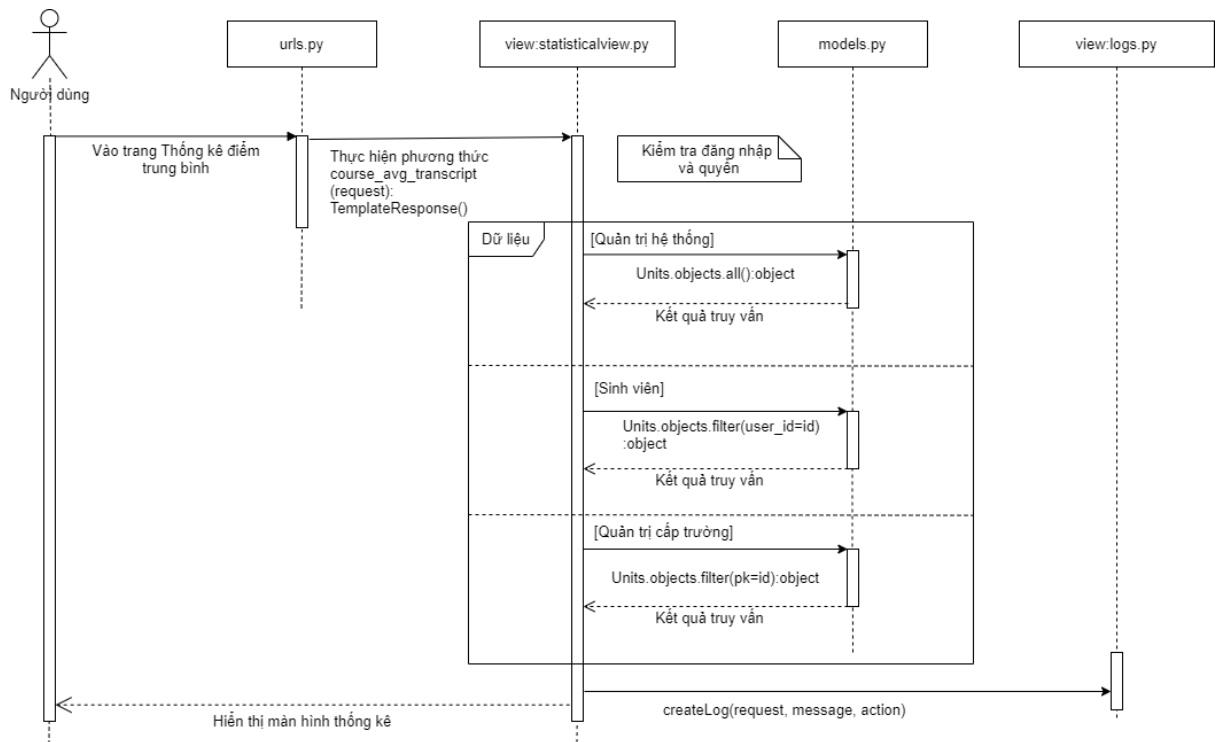


Hình 2.15. Biểu đồ các thao tác khi thực hiện xem thống kê.

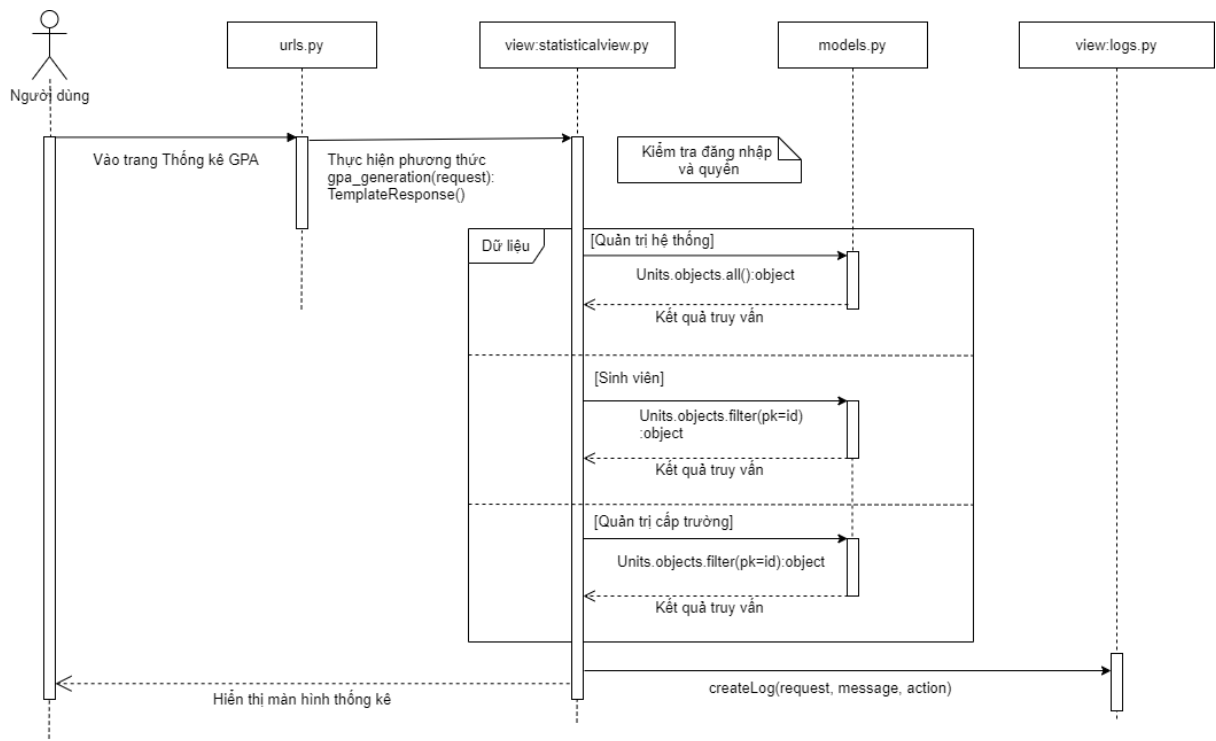
Biểu đồ tuần tự:



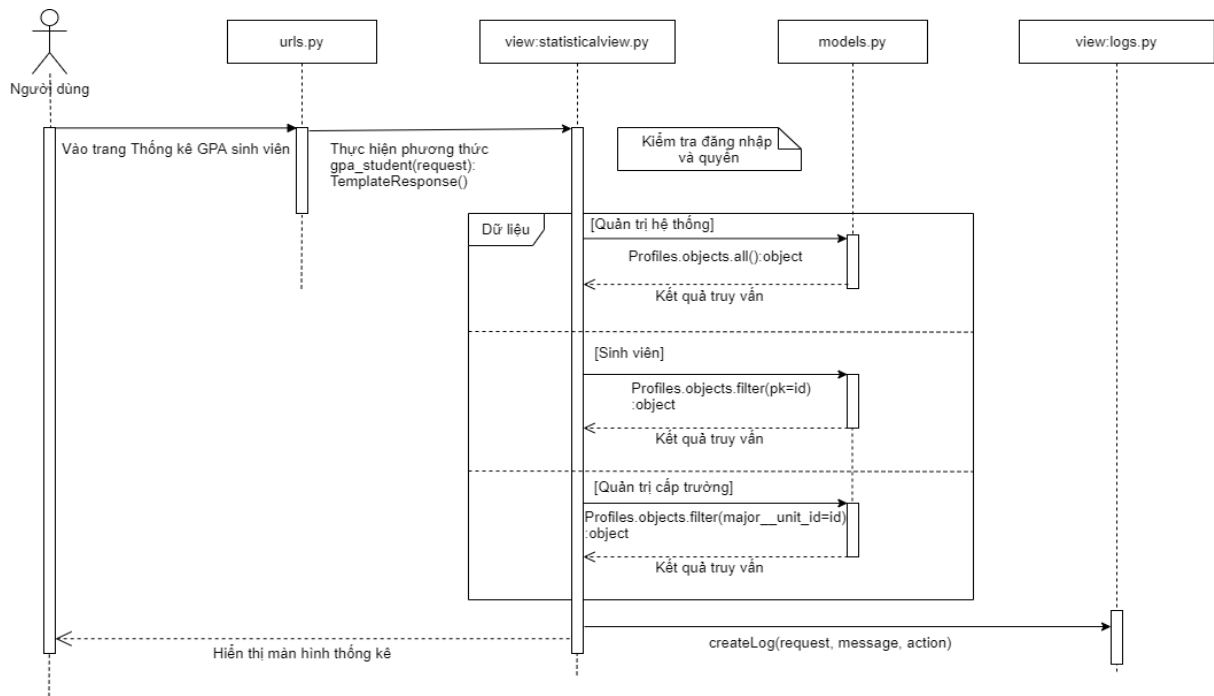
Hình 2.16. Biểu đồ tuần tự tính năng thống kê theo môn học và khóa.



Hình 2.17. Biểu đồ tuần tự tính năng thống kê điểm trung bình môn học qua các năm.



Hình 2.18. Biểu đồ tuần tự tính năng thống kê GPA theo khóa và ngành.



Hình 2.19. Biểu đồ tuần tự tính năng thống kê GPA sinh viên.

2.2.2.5. Dự đoán kết quả học tập

Mô tả tóm tắt: Người dùng có thể xem kết quả dự đoán điểm của sinh viên và tính dự báo điểm theo môn. Ngoài ra quản trị viên và quản lý cấp trường có thể tính toán các dự báo theo khóa.

Luồng chính:

Hành động người dùng	Hệ thống phản hồi	Dữ liệu
Xem kết quả	Kết quả điểm mà hệ thống dự đoán.	Dữ liệu từ sinh viên.
Tính dự báo theo môn học	Kết quả điểm mà hệ thống dự đoán.	Dữ liệu từ sinh viên và môn học.
Tính dự báo theo khóa	Kết quả điểm mà hệ thống dự đoán.	Dữ liệu từ ngành, khóa.

Bảng 2.5. Bảng mô tả các hành động trong luồng chính dự đoán điểm.

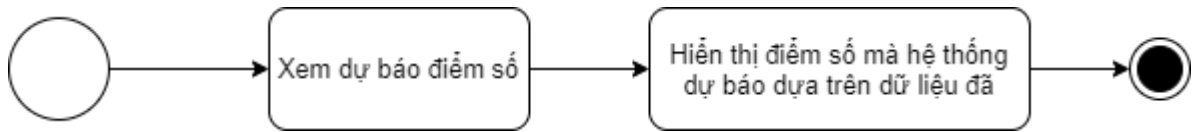
Luồng phụ: Không có.

Yêu cầu đặc biệt: Khi thực hiện tính toán dự báo theo khóa thì bắt buộc phải là tài khoản quản trị hệ thống hoặc là tài khoản quản lý cấp trường.

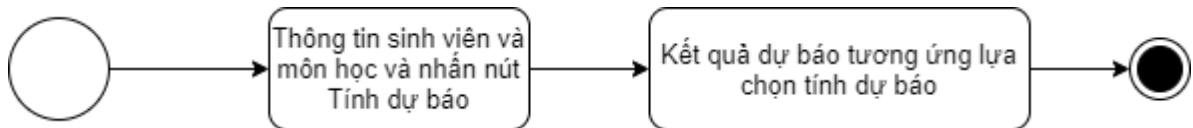
Điều kiện tiên quyết: Người dùng phải đăng nhập vào hệ thống.

Các vấn đề mở: Không có.

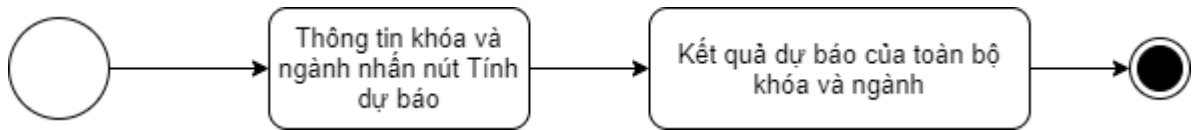
Biểu đồ hoạt động:



Hình 2.20. Biểu đồ các thao tác khi thực hiện xem dự báo điểm.

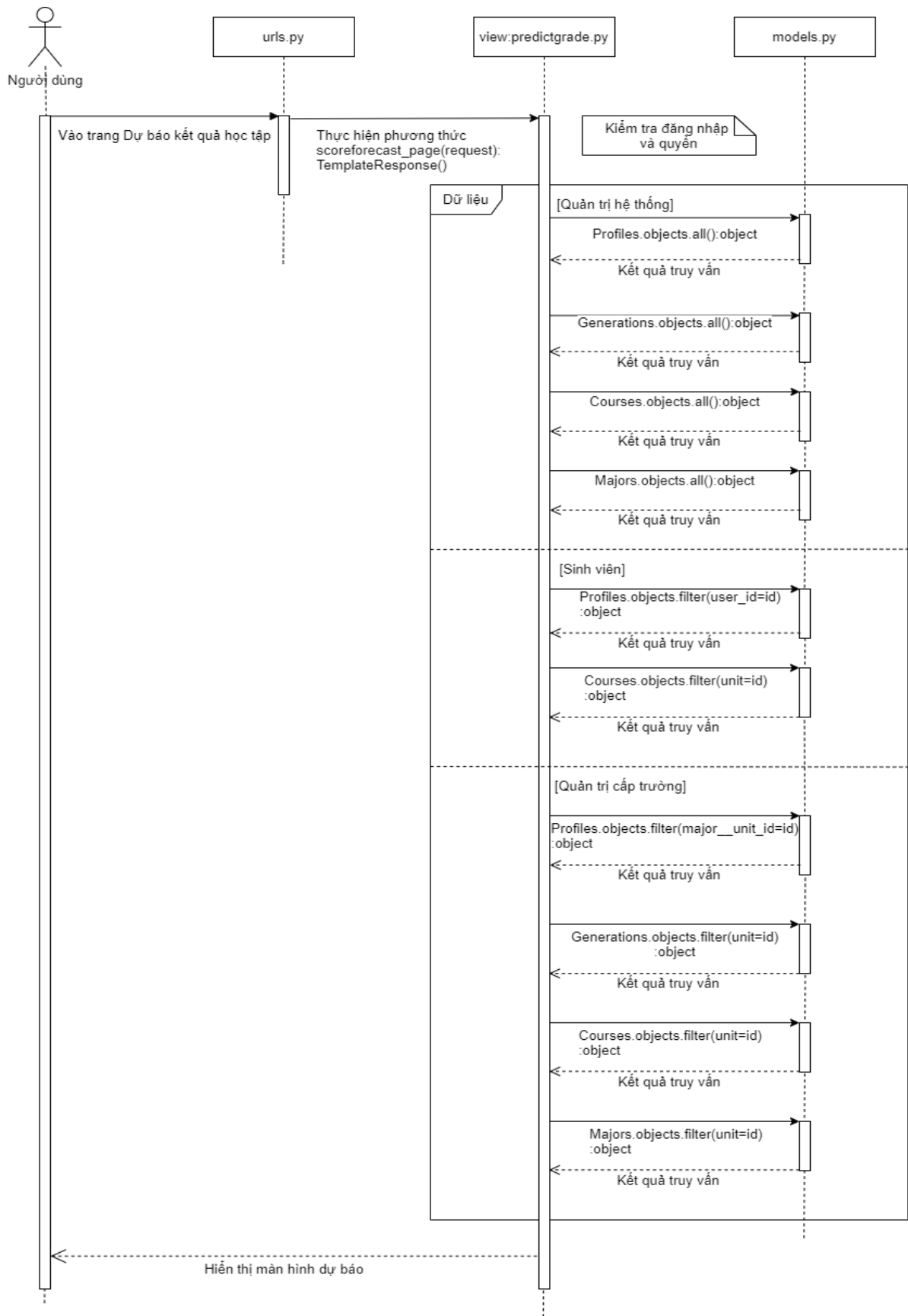


Hình 2.21. Biểu đồ các thao tác khi thực hiện tính dự báo theo môn.



Hình 2.22. Biểu đồ các thao tác khi thực hiện tính dự báo theo khóa.

Biểu đồ tuần tự:



Hình 2.23. Biểu đồ tuần tự tính năng dự đoán điểm số.

2.2.2.6. Gợi ý môn học

Mô tả tóm tắt: Người dùng có thể xem các môn học được hệ thống gợi ý phù hợp với điểm số.

Luồng chính:

Hành động người dùng	Hệ thống phản hồi	Dữ liệu
Lựa chọn số lượng môn, cách gợi ý môn học	Danh sách môn học và điểm số đã dự báo cho môn học đó	Dữ liệu từ sinh viên.

Bảng 2.6. Bảng mô tả các hành động trong luồng chính gợi ý môn học.

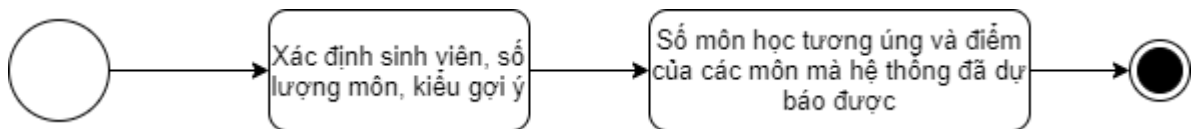
Luồng phụ: Không có.

Yêu cầu đặc biệt: Không có.

Điều kiện tiên quyết: Người dùng phải đăng nhập vào hệ thống.

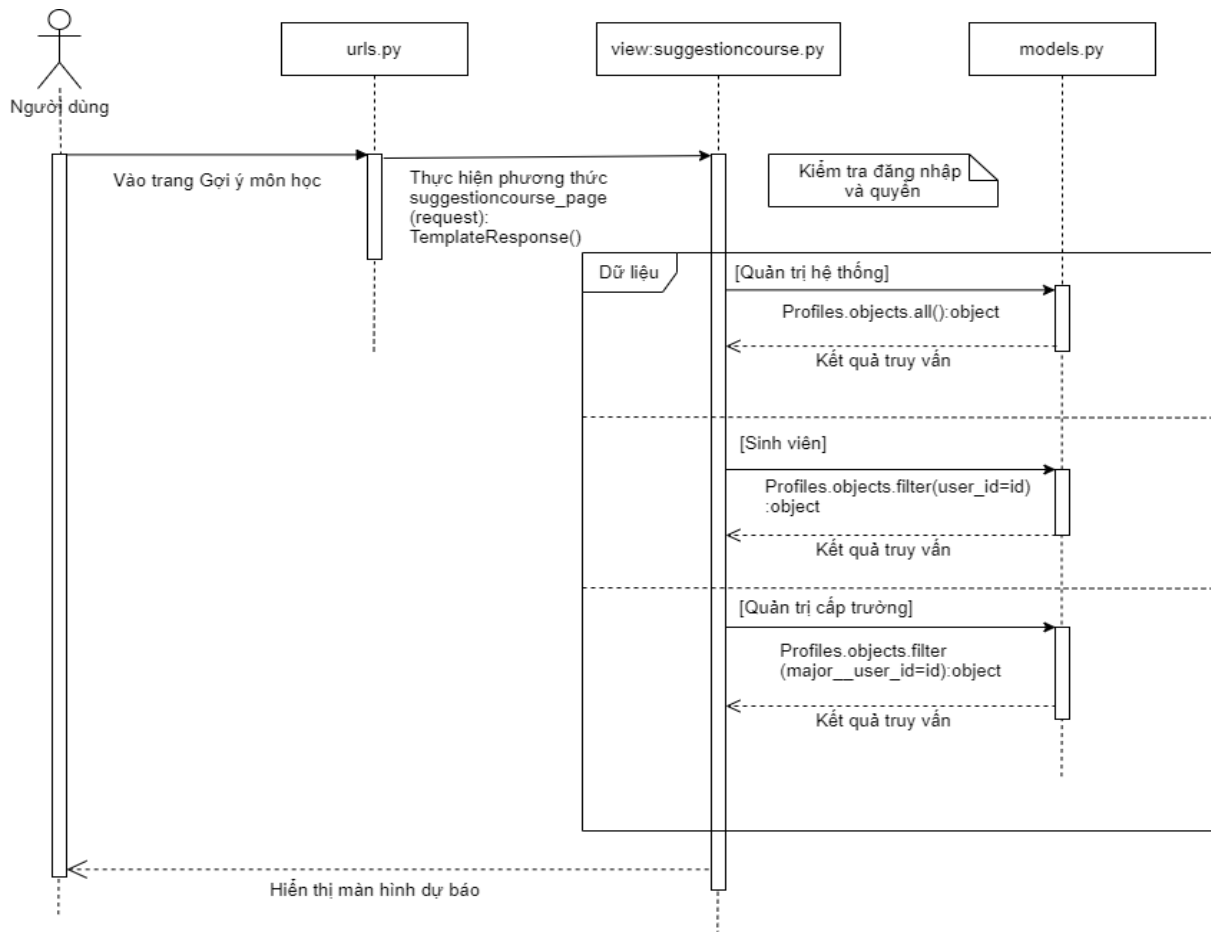
Các vấn đề mở: Không có.

Biểu đồ hoạt động:



Hình 2.24. Biểu đồ các thao tác khi thực hiện xem gợi ý môn học.

Biểu đồ tuần tự:



Hình 2.25. Biểu đồ tuần tự tính năng gợi ý môn học.

2.2.2.7. Chỉnh sửa thông tin hồ sơ

Mô tả tóm tắt: Người dùng có thể xem các thông tin về tên, mã sinh viên, lớp, ngành, khoa đang theo học và có thể thay đổi ngày sinh, email cá nhân.

Luồng chính:

Hành động người dùng	Hệ thống phản hồi	Dữ liệu
Nhập thông tin cho các hạng mục có thể thay đổi	Thực hiện kiểm tra dữ liệu và cập nhật dữ liệu vào hệ thống.	Dữ liệu về người dùng, trường, ngành, điểm, khóa, lớp.
	Thông báo trạng thái thành công cho người dùng	

Bảng 2.7. Bảng mô tả các hành động trong luồng chính chỉnh sửa thông tin hồ sơ.

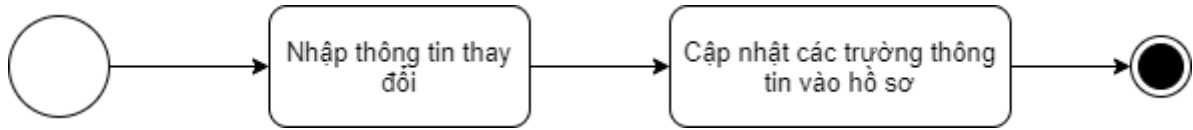
Luồng phụ: Không có.

Yêu cầu đặc biệt: Không có.

Điều kiện tiên quyết: Người dùng phải đăng nhập vào hệ thống.

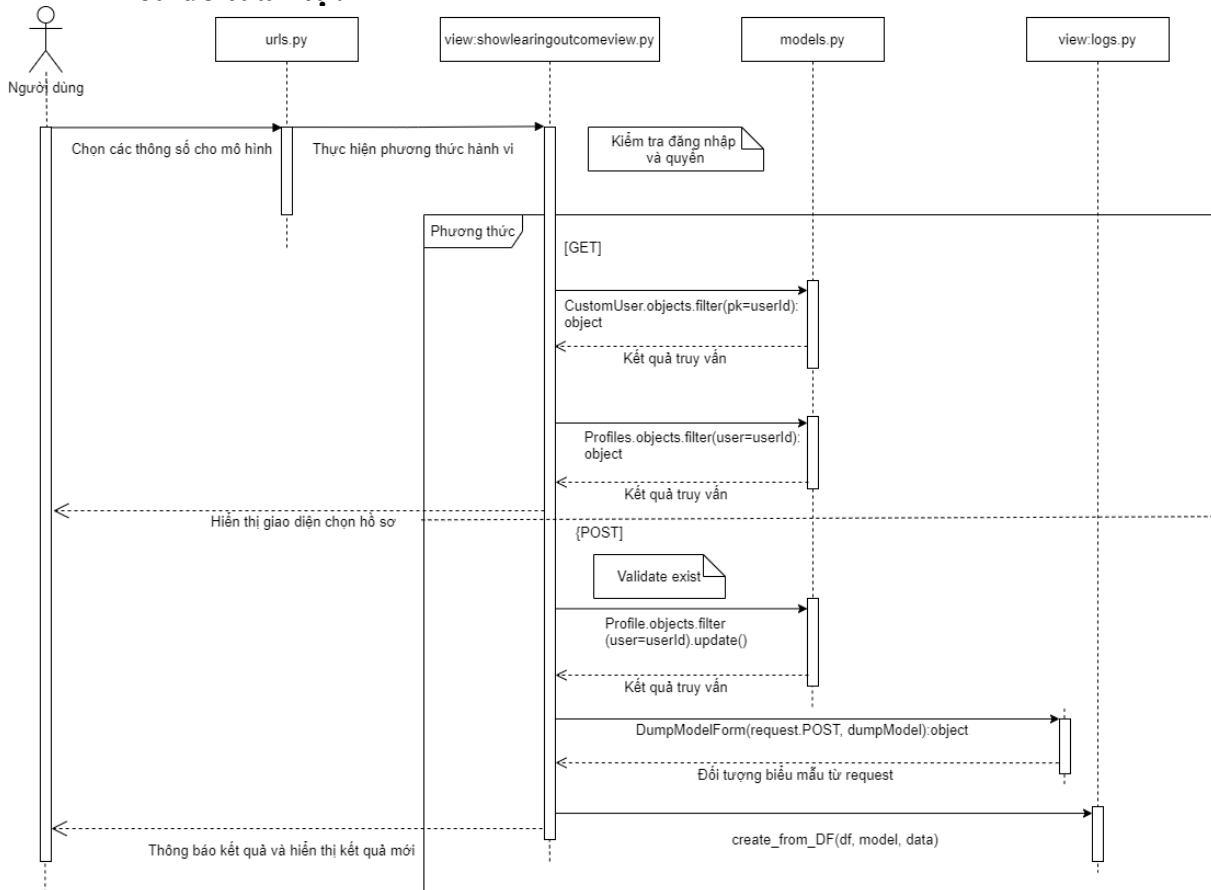
Các vấn đề mở: Không có.

Biểu đồ hoạt động:



Hình 2.26. Biểu đồ các thao tác khi thực hiện cập nhật hồ sơ cá nhân.

Biểu đồ tuần tự:



Hình 2.27. Biểu đồ tuần tự thao tác chỉnh sửa hồ sơ.

2.2.2.8. Đăng nhập

Mô tả tóm tắt: Người dùng thực hiện đăng nhập để có thể sử dụng hệ thống. Đây là điều kiện bắt buộc để sử dụng hệ thống.

Luồng chính:

Hành động người dùng	Hệ thống phản hồi	Dữ liệu
----------------------	-------------------	---------

1. Nhập tên đăng nhập và mật khẩu.	1.1. Thực hiện kiểm tra thông tin đăng nhập.	Dữ liệu về tài khoản người dùng (quản trị cấp trường, quản trị hệ thống).
	1.2. Hiện thị trang giao diện sử dụng chức năng.	Dữ liệu về vai trò và quyền của người dùng.

Bảng 2.8. Bảng mô tả các hành động trong luồng chính đăng nhập.

Luồng phụ:

Tại bước 1.1: Khi người dùng nhập sai thông tin đăng nhập thì không thực hiện di chuyển vào trang đăng nhập thành công mà hiển thị lại trang đăng nhập và thông báo lỗi.

Tại bước 1.1: Khi tài khoản sinh viên thực hiện xác thực thì hệ thống kiểm tra sự tồn tại trong hệ thống và đưa ra quyết định có thêm mới người dùng vào hệ thống khi mà hệ thống xác thực LDAP trả về kết quả thành công. Nếu LDAP trả về sai thì không thực hiện vào phiên hoạt động và hiển thị lại trang đăng nhập và thông báo lỗi.

Yêu cầu đặc biệt: Không có.

Điều kiện tiên quyết: Người dùng chưa đăng nhập vào hệ thống.

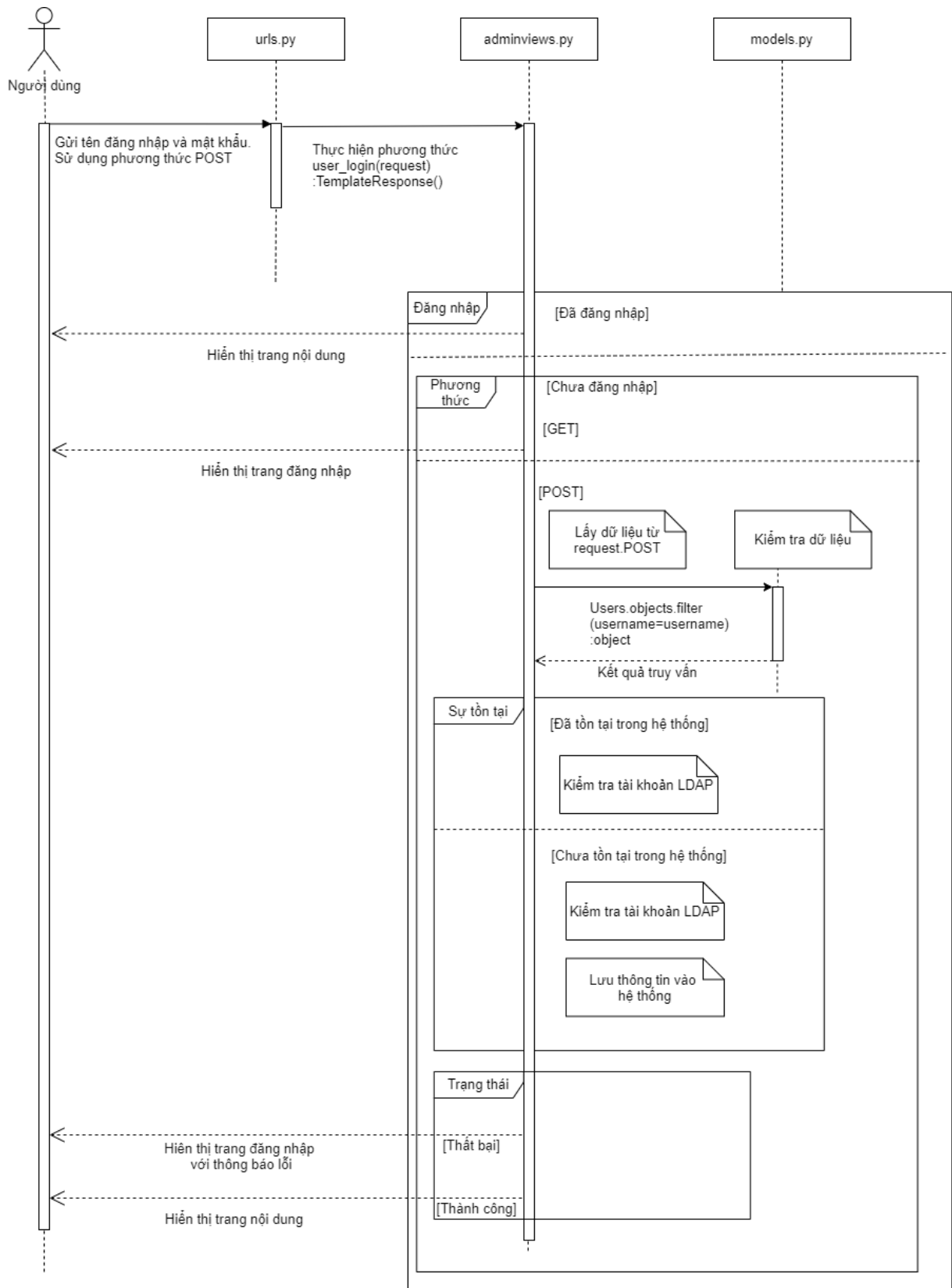
Các vấn đề mở: Không có.

Biểu đồ hoạt động:



Hình 2.28. Biểu đồ các thao tác khi thực hiện đăng nhập.

Biểu đồ tuần tự:



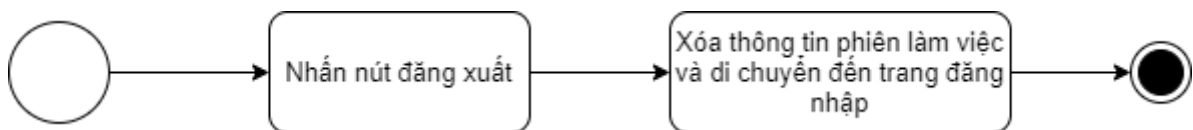
Hình 2.29. Biểu đồ tuần tự tính năng đăng nhập.

2.2.2.9. Đăng xuất

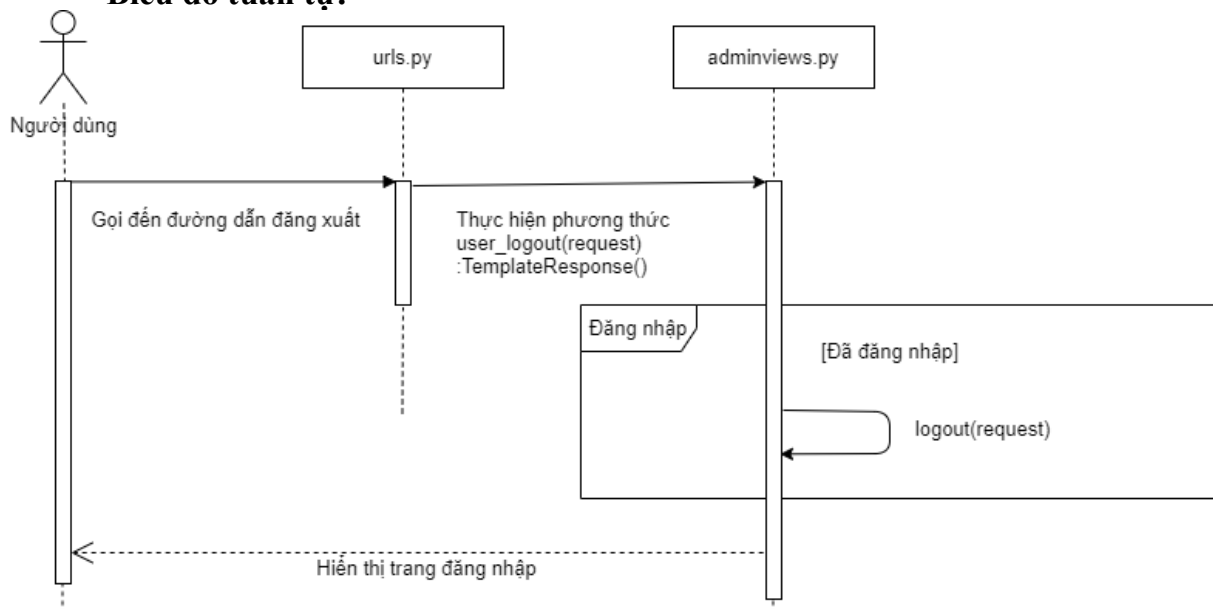
Mô tả tóm tắt: Người dùng thực hiện đăng xuất khỏi hệ thống.

Luồng chính:

Hành động người dùng	Hệ thống phản hồi	Dữ liệu
Nhấn nút đăng xuất.	Thực hiện xóa phiên làm việc hiện tại của người dùng. Và hiện thị trang đăng nhập.	Phiên làm việc thời điểm của người dùng.

Luồng phụ: Không có.**Yêu cầu đặc biệt:** Không có.**Điều kiện tiên quyết:** Người dùng đã đăng nhập vào hệ thống.**Các vấn đề mở:** Không có.**Biểu đồ hoạt động:**

Hình 2.30. Biểu đồ các thao tác khi thực hiện đăng xuất.

Biểu đồ tuần tự:

Hình 2.31. Biểu đồ tuần tự tính năng đăng xuất.

2.3. Thiết kế

2.3.1. Thiết kế kiến trúc hệ thống

2.3.1.1. Tổng quan về cấu trúc

```
root
|   gunicorn.conf.py
|   manage.py
|   output.txt
|   requirements.txt
|
+---mainApp
|   |   admin.py
|   |   adminurls.py
|   |   adminviews.py
|   |   apps.py
|   |   forms.py
|   |   managers.py
|   |   middlewares.py
|   |   models.py
|   |   tests.py
|   |   urls.py
|   |   views.py
|   |   __init__.py
|   |
|   +---dataprocessing
|   |
|   +---serializers
|   |
|   +---static
|   |   |   +---css
|   |   |   |
|   |   |   +---images
|   |   |   |
|   |   |   +---js
|   |   |
|   |   +---templates
|   |   |   |   bootstrap.html
|   |   |   |
|   |   |   +---adminuet
|   |   |   |
|   |   |   +---student
|   |   |
|   |   +---viewapi
|   |
+---model
|   |   CF_IB.pickle
|   |   CF_UB.pickle
|   |   MF.pickle
|   |
+---project
|   |   asgi.py
|   |   settings.py
|   |   urls.py
|   |   views.py
|   |   wsgi.py
|   |   __init__.py
|   |
+---recommendation
|   |   admin.py
|   |   apps.py
|   |   middlewares.py
|   |   models.py
|   |   tests.py
|   |   urls.py
|   |   views.py
|   |   __init__.py
|   |
|   +---core
|   |   |
|   |   +---lib
|   |   |
|   |   +---migrations
|   |
+---static
|   |   +---admin
|   |   |   +---css
|   |   |   |
|   |   |   +---fonts
|   |   |   |
|   |   |   +---img
|   |   |   |
|   |   |   +---js
|   |   |
|   |   +---customm
|   |   |   style.css
|   |
+---statistic
|   |   admin.py
|   |   apps.py
|   |   lib.py
|   |   middlewares.py
|   |   models.py
|   |   tests.py
|   |   urls.py
|   |   views.py
|   |   __init__.py
|   |
+---data
|   |   |   load_data.py
|   |   |
|   |   +---input
|   |   |   |   course_info.csv
|   |   |   |   semester_courses.json
|   |   |   |   semester_courses.txt
|   |   |
|   |   +---data
|   |
+---migrations
|
+---train_data
|   1_train_data.csv
```

Hình 2.32. Cấu trúc thư mục.

2.3.1.2. Mô tả chi tiết

Thư mục project: là cấu trúc ban đầu của dự án và là cài đặt, cấu hình để dự án sử dụng các cấu hình để bắt đầu chương trình làm việc. Nhiệm vụ chính là lưu thông số, lưu tên thư viện liên quan, cấu hình máy chủ, cấu hình cơ sở dữ liệu và những thông tin cần thiết khác. Dự án khi khởi tạo thì đã có sẵn một app chính của dự án để toàn bộ dự án tập trung tài nguyên và cấu hình. Vì thế trong app thành sẽ cấu hình các đường dẫn đến các app thành phần và thêm các cài đặt về cơ sở dữ liệu, thư viện và các cài đặt cần thiết khác dành cho máy chủ.

Thư mục mainApp: xử lý toàn bộ dữ liệu chính từ các app đến tiếp nhận, phản hồi của người dùng. Cấu trúc bao gồm controller, model, template, serializer, form, urls... Trong thành phần xử lý dữ liệu chính của toàn bộ hệ thống thì nhận thức được là sử dụng mô hình theo cấu trúc thì cần phát triển các đối tượng trong file models.py và việc xử lý các luồng nghiệp vụ thì sẽ tạo thư mục riêng biệt để mỗi thực thể sẽ là một file. Khi cần sử dụng thì sử dụng lẫn nhau và tránh sự chồng chéo và quá lớn của các đoạn mã. Và vì các đoạn code xử lý này cần đến serializer thì cũng tạo ra mỗi đối tượng một file riêng. Để tận dụng những công cụ mà Django hỗ trợ về biểu mẫu thì có tận dụng một biểu mẫu và định nghĩa rõ ràng biểu mẫu của các đối tượng. File này là forms.py. Để lưu các file giao diện html thì lưu trong thư mục templates. Các file các như ảnh, định nghĩa giao diện, thao tác, cử chỉ trên giao diện và các thư viện của các bên khác.

Thư mục recommendation: thực hiện các phân tích, thống kê từ những dữ liệu theo mô hình được định nghĩa ở thư mục mainApp. Từ đó thực hiện tính toán, phân tích số liệu để phục vụ cho chức năng dự báo điểm và gợi ý môn học. Tại đây các thao tác thực hiện riêng biệt với thư mục mainApp và giao tiếp thông qua API giúp các phần có thể phát triển song song và không ảnh hưởng đến phần còn lại.

Thư mục statistic: thực hiện phân tích và thống kê dữ liệu điểm số, môn học, sinh viên để đem lại cái nhìn bao quát, hữu ích cho người dùng.

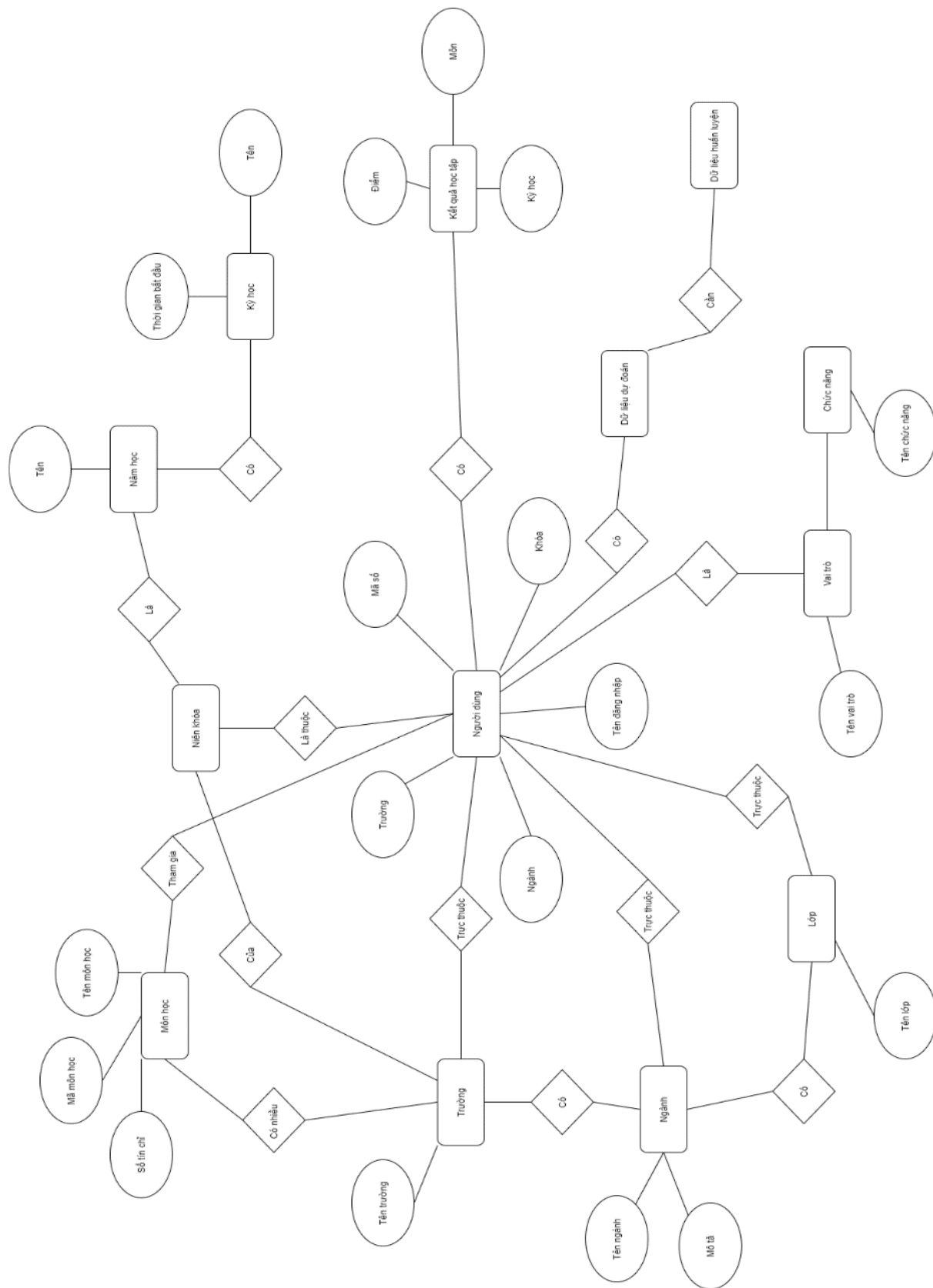
Thư mục train_data: Thực hiện lưu các file dữ liệu đã qua huấn luyện để làm kết quả huấn luyện.

Thư mục model: Thực hiện lưu các file nén model cho tính toán.

Thư mục khác: Khi phát triển rộng hơn thì các thành phần phát triển thêm sẽ ở trong các thư mục khác riêng biệt và không làm ảnh hưởng đến các thành phần đã có.

2.3.2. Thiết kế cơ sở dữ liệu

2.3.2.1. Mô hình ER



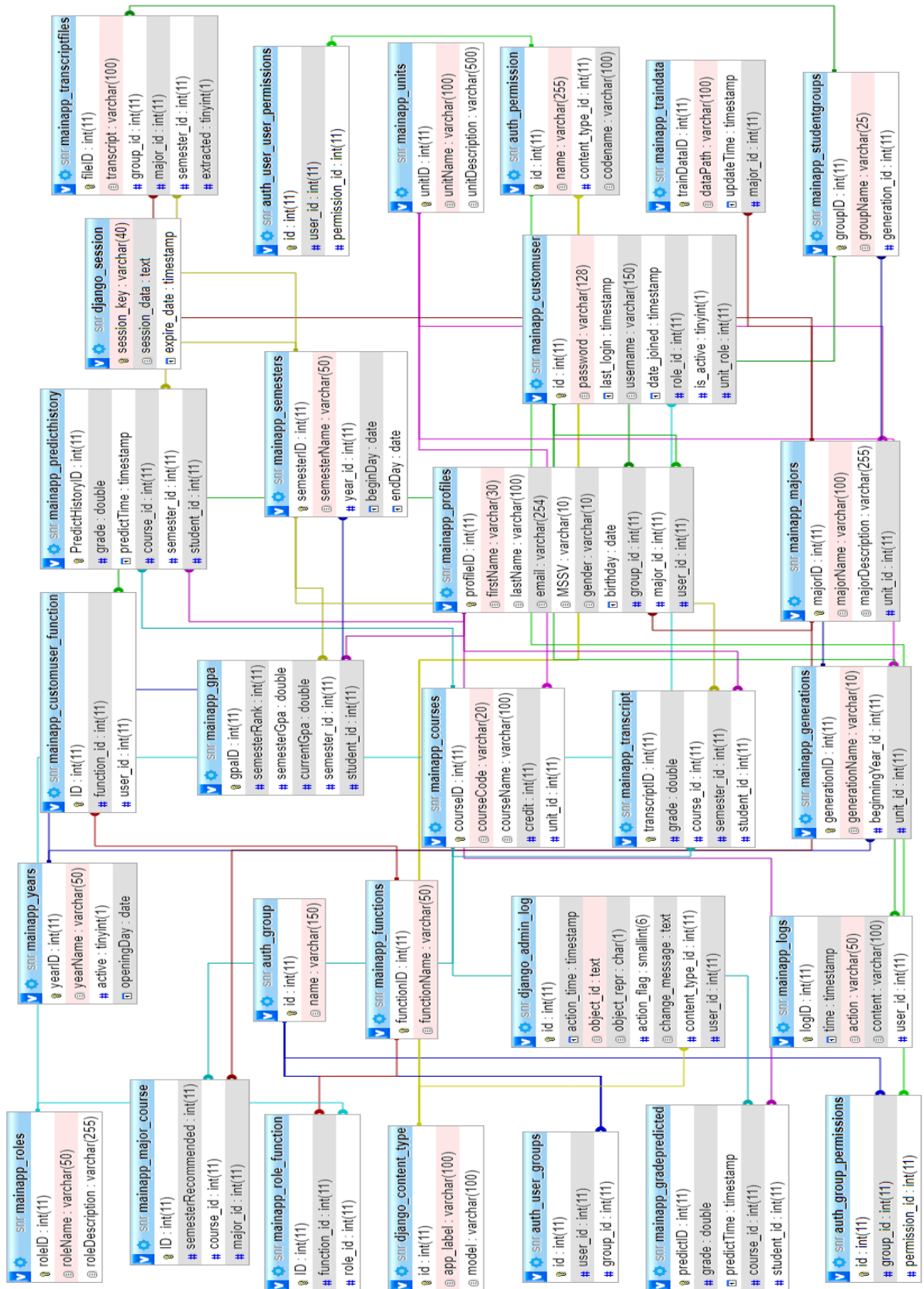
Hình 2.33. Tổng quan mô hình ER

Người dùng có thể sẽ là sinh viên, quản trị hệ thống, quản trị cấp trường và đều thuộc một đơn vị đào tạo cấp trường duy nhất. Trong cấp trường thì sẽ có nhiều ngành, nhiều niên khóa và các môn học. Mỗi sinh viên, quản lý cấp trường trực thuộc một ngành cụ thể, ví dụ như là Công nghệ thông tin, Điện tử viễn thông,... Trong các ngành đều có chia ra các lớp để tiện cho việc quản lý sinh viên. Đây cũng là lý do người dùng sẽ không trực tiếp có thuộc tính lớp vì quản lý cấp trường không nhất thiết thuộc một lớp nào cả. Các môn học được chia theo ngành để phù hợp với khung đào tạo. Vì thế môn học ở các ngành khác nhau được hệ thống hiểu là khác nhau.

Các vai trò trong hệ thống xác định đúng vai trò của người dùng thực tế hệ thống. Với mỗi vai trò thì sẽ có các khả năng, phạm vi hoạt động khác nhau trong hệ thống. Mỗi khả năng được hiểu là chức năng. Mỗi vai trò sẽ có tập hợp một số chức năng cần thiết để thao tác tốt với hệ thống.

Dữ liệu dự đoán là thực thể lưu những thông tin về điểm số mà dự đoán cho sinh viên. Để có được khả năng dự đoán tốt nhất thì cần tập dữ liệu để huấn luyện cho việc dự đoán nên cần tập dữ liệu huấn luyện để xác định rõ vai trò và tăng hiệu quả của hệ thống khi mô hình hóa các thực thể.

2.3.2.2. Mô hình quan hệ



Hình 2.34. Tổng quan quan hệ của các thực thể.

Bảng	Mô tả
auth_group	Bảng Django lưu các nhóm.
auth_group_permissions	Bảng Django lưu nhóm các quyền của người dùng.
auth_permission	Bảng Django lưu các quyền của người dùng.
auth_user_groups	Bảng Django lưu các nhóm người dùng.
auth_user_user_permissions	Bảng Django lưu quyền người dùng.
django_admin_log	Bảng Django lưu lịch sử cho quản trị viên.
django_content_type	Bảng Django lưu các loại nội dung.
mainapp_courses	Bảng lưu thông tin các môn học.
mainapp_customer	Bảng lưu thông tin cơ sở người dùng của Django.
mainapp_customser_function	Bảng lưu các chức năng chỉ dành người dùng nhất định.
mainapp_dumpmodel	Bảng lưu các mô hình dự đoán của hệ thống.
mainapp_functions	Bảng lưu các chức năng của hệ thống.
mainapp_generations	Bảng lưu các niên khóa của trường.
mainapp_gpa	Bảng lưu điểm tổng kết quả sinh viên.
mainapp_gradepredicted	Bảng lưu dữ liệu điểm số dự đoán.
mainapp_logs	Bảng lưu các lịch sử log.
mainapp_majors	Bảng lưu các ngành học của trường.
mainapp_major_course	Bảng lưu các môn học tương ứng với ngành học.
mainapp_predicthistory	Bảng lưu lịch sử các lần tạo ra dữ liệu dự đoán, gợi ý.
mainapp_profiles	Bảng lưu thông tin về ngành, khóa của người dùng.
mainapp_roles	Bảng lưu thông tin các vai trò.
mainapp_role_function	Bảng lưu ánh xạ chức năng và vai trò.
mainapp_semesters	Bảng lưu các kỳ học.
mainapp_studentgroups	Bảng lưu các lớp của ngành.
mainapp_traindata	Bảng lưu các điểm dùng để làm dữ liệu huấn luyện.

mainapp_transcript	Bảng lưu chi tiết điểm của sinh viên.
mainapp_transcriptfile	Bảng lưu các file kết quả học tập của sinh viên.
mainapp_units	Bảng lưu các thông tin về trường, đơn vị đào tạo.
mainapp_years	Bảng lưu các thông tin về năm học.

Bảng 2.9. Danh sách các bảng dựa trên các thực thể của hệ thống.

2.3.2.3. Cơ sở dữ liệu vật lý

Thuộc tính	Mô tả
id	ID của nhóm auth có sẵn trong django
name	Tên của nhóm auth có sẵn trong django

Bảng 2.10. Cấu trúc bảng auth_group.

Thuộc tính	Mô tả
id	ID của nhóm auth permission có sẵn trong django
group_id	ID của nhóm auth là khóa ngoại tham chiếu đến bảng auth_group
permission_id	ID của quyền

Bảng 2.11. Cấu trúc bảng auth_group_permissions.

Thuộc tính	Mô tả
id	ID là khóa chính của bảng
name	Tên của quyền định danh
content_type_id	ID của kiểu content tương ứng
codename	Mã của quyền định danh

Bảng 2.12. Cấu trúc bảng auth_permission.

Thuộc tính	Mô tả
id	ID là khóa chính của bảng
user_id	ID định danh người dùng
group_id	ID nhóm quyền của người dùng

Bảng 2.13. Cấu trúc bảng auth_user_groups.

Thuộc tính	Mô tả
id	ID là khóa chính của bảng
user_id	ID định danh người dùng
permission_id	ID của quyền

Bảng 2.14. Cấu trúc bảng auth_user_user_permissions.

Thuộc tính	Mô tả
id	ID là khóa chính của bảng
action_time	Thời gian thực hiện hành động
object_id	ID của đối tượng hành động
object_repr	Phản hồi của đối tượng khi nhận được hành động
action_flag	Đánh dấu action
change_message	Nội dung ghi lịch sử
content_type_id	ID của loại thông tin nội dung
user_id	ID của người dùng

Bảng 2.15. Cấu trúc bảng django_admin_log.

Thuộc tính	Mô tả
id	ID là khóa chính của bảng
app_label	Nhãn của nội dung
model	Đối tượng, thực thể chứa nội dung

Bảng 2.16. Cấu trúc bảng django_content_type.

Thuộc tính	Mô tả
courseID	ID của môn học
courseCode	Mã môn học
courseName	Tên môn học
credit	Số tín chỉ cho môn học
unit_id	ID của trường có môn học, tham chiếu tới khóa chính của bảng units

Bảng 2.17. Cấu trúc bảng mainapp_courses.

Thuộc tính	Mô tả
id	ID của người dùng
password	Mật khẩu của người dùng
last_login	Lần đăng nhập gần nhất
username	Tên đăng nhập của người dùng
date_joined	Thời gian tài khoản được tạo
role_id	ID của vai trò người dùng
is_active	Trạng thái hoạt động
unit_role	Người dùng thuộc đơn vị nào

Bảng 2.18. Cấu trúc bảng mainapp_customer.

Thuộc tính	Mô tả
id	ID của người dùng
password	Mật khẩu của người dùng
last_login	Lần đăng nhập gần nhất
username	Tên đăng nhập của người dùng
date_joined	Thời gian tài khoản được tạo
role_id	ID của vai trò người dùng
is_active	Trạng thái hoạt động
unit_role	Người dùng thuộc đơn vị nào

Bảng 2.19. Cấu trúc bảng mainapp_customser_function.

Thuộc tính	Mô tả
dumpModeID	ID của người dùng
dumpFile	Đường dẫn vị trí lưu file dump
updateTime	Thời gian thực hiện
active	Model nào được sử dụng dự đoán thì sẽ active là true

args	Các tham số và giá trị trong model lưu dưới dạng json
modelName	Tên model
param	Là kiểu mô tả chi tiết về các tham số

Bảng 2.20. Cấu trúc bảng mainapp_dumpmodel.

Thuộc tính	Mô tả
functionID	ID của chức năng
functionName	Tên chức năng

Bảng 2.21. Cấu trúc bảng mainapp_functions.

Thuộc tính	Mô tả
generationID	ID của khóa
generationName	Tên của khóa
beginningYear_id	ID của năm học mà khóa học bắt đầu
unit_id	ID của trường mà khóa theo học

Bảng 2.22. Cấu trúc bảng mainapp_generations.

Thuộc tính	Mô tả
gpaID	ID của bảng điểm
semesterRank	Vị trí xếp hạng của kỳ
semesterGpa	Điểm gpa chỉ tính trong kỳ
currnetGpa	Tổng điểm gpa của toàn bộ từ khi bắt đầu học đến hiện tại
semester_id	ID của kỳ học mà có điểm gpa
student_id	ID của người dùng sinh viên

Bảng 2.23. Cấu trúc bảng mainapp_gpa.

Thuộc tính	Mô tả
predictID	ID của dữ liệu dự đoán
predictTime	Thời gian thực hiện dự đoán
course_id	ID của môn học

student_id	ID của người dùng sinh viên
------------	-----------------------------

Bảng 2.24. Cấu trúc bảng mainapp_gradepredicted.

Thuộc tính	Mô tả
logID	ID của dữ liệu lịch sử
time	Thời gian thực hiện hành động
action	Hành động
content	Nội dung mà hành động thực hiện
user_id	ID của người dùng

Bảng 2.25. Cấu trúc bảng mainapp_logs.

Thuộc tính	Mô tả
majorID	ID của ngành
majorName	Tên của ngành
majorDescription	Mô tả về ngành
unit_id	ID của trường mà ngành thuộc sự quản lý

Bảng 2.26. Cấu trúc bảng mainapp_majors.

Thuộc tính	Mô tả
ID	ID của môn học tương ứng với ngành
semesterRecommended	Kỳ học gợi ý học
course_id	ID của môn học nên học vào kỳ semesterRecommended
major_id	ID của kỳ mà bắt buộc phải học môn có ID là course_id

Bảng 2.27. Cấu trúc bảng mainapp_major_course.

Thuộc tính	Mô tả
PredictHistoryID	ID của mỗi lần dự đoán
grade	Điểm dự đoán
predictTime	Thời gian thực hiện dự đoán
course_id	ID của môn học mà thực hiện dự đoán

semester_id	ID của kỳ mà dự đoán nên học môn course_id
student_id	ID của sinh viên mà thực hiện dự đoán

Bảng 2.28. Cấu trúc bảng mainapp_predicthistory.

Thuộc tính	Mô tả
profileID	ID của profile người dùng
firstName	Là tên của người dùng
lastName	Là họ của người dùng
email	Địa chỉ thư điện tử của người dùng
MSSV	Mã số sinh viên của người dùng nếu là sinh viên
gender	Giới tính của sinh viên
birthday	Ngày tháng năm sinh của người dùng
group_id	ID của lớp mà người dùng thuộc vào
major_id	ID của ngành mà người dùng theo học
user_id	ID của người dùng

Bảng 2.29. Cấu trúc bảng mainapp_profiles.

Thuộc tính	Mô tả
roleID	ID của vai trò
roleName	Tên của vai trò
roleDescription	Mô tả vai trò

Bảng 2.30. Cấu trúc bảng mainapp_roles.

Thuộc tính	Mô tả
ID	ID khóa chính của bảng
function_id	ID của chức năng, là khóa ngoại tham chiếu đến bảng mainapp_functions
role_id	ID của vai trò, là khóa ngoại tham chiếu đến bảng mainapp_roles

Bảng 2.31. Cấu trúc bảng mainapp_role_function.

Thuộc tính	Mô tả
------------	-------

semesterID	ID của kỳ học
semesterName	Tên của kỳ học
year_id	ID của năm học, là khóa ngoại tham chiếu đến bảng mainapp_years
beginDay	Ngày bắt đầu học kỳ
endDay	Ngày kết thúc kỳ học

Bảng 2.32. Cấu trúc bảng mainapp_semesters.

Thuộc tính	Mô tả
groupID	ID của lớp học
groupName	Tên của lớp học
generation_id	ID của khóa, là khóa ngoại tham chiếu đến bảng mainapp_generations

Bảng 2.33. Cấu trúc bảng mainapp_studentgroups.

Thuộc tính	Mô tả
trainDataID	ID là khóa chính của bảng
dataPath	Đường dẫn đến file chứa dữ liệu
updateTime	Thời gian thực hiện
major_id	ID của ngành, là khóa ngoại tham chiếu đến bảng mainapp_majors

Bảng 2.34. Cấu trúc bảng mainapp_traindata.

Thuộc tính	Mô tả
transcriptID	ID là khóa chính của bảng
grade	Điểm
course_id	ID của môn học, là khóa ngoại tham chiếu đến mainapp_courses
semester_id	ID của kỳ học, là khóa ngoại tham chiếu đến bảng mainapp_semesters
student_id	ID của người dùng sinh viên, là khóa ngoại tham chiếu đến bảng mainapp_profiles

Bảng 2.35. Cấu trúc bảng mainapp_transcript.

Thuộc tính	Mô tả
fileID	ID là khóa chính của bản ghi
transcript	Đường dẫn file của điểm dữ liệu huấn luyện
group_id	ID của lớp, là khóa ngoại tham chiếu đến bảng mainapp_groups
major_id	ID của ngành, là khóa ngoại tham chiếu đến bảng mainapp_majors
semester_id	ID của kỳ học, là khóa ngoại tham chiếu đến bảng mainapp_semesters
extracted	Trạng thái đã extract file dữ liệu điểm cho tương ứng với lớp và ngành.

Bảng 2.36. Cấu trúc bảng mainapp_transcriptfile.

Thuộc tính	Mô tả
unitID	ID của trường, là khóa chính của bảng
unitName	Tên trường
unitDescription	Mô tả trường

Bảng 2.37. Cấu trúc bảng mainapp_units.

Thuộc tính	Mô tả
yearID	ID của năm học, là khóa chính của bảng
yearName	Tên năm học
active	Trạng thái của năm học
openingDay	Ngày bắt đầu năm học

Bảng 2.38. Cấu trúc bảng mainapp_years.

Chương 3. Cài đặt thử nghiệm và đánh giá

3.1. Sử dụng API

3.1.1. Thống kê

3.1.1.1. Thống kê phổ điểm theo môn và khóa

Để lấy được dữ liệu về thống kê phổ điểm môn học theo toàn bộ khóa học thì cần thực hiện truyền dữ liệu vào đường dẫn sau:

```
domain/statistic/distribute/<int:unit_id>/<int:generation_id>/<int:major_id>/<int:course_id>/<str:boundary_type>/<int:sem_start>/<int:sem_end>
```

Tham số	Mô tả
unit_id	ID của trường (bắt buộc)
generation_id	ID của các khóa (bắt buộc)
major_id	ID chuyên ngành (bắt buộc)
course_id	ID của môn học muốn thống kê (bắt buộc)
boundary_type	Kiểu thống kê của khoảng điểm, kiểu dữ liệu điểm. Có 3 thang phân chia điểm là a-f, 10 và 20
sem_start	ID của semester đầu tiên muốn thống kê (không bắt buộc)
sem_end	ID của semester cuối cùng muốn thống kê (không bắt buộc)

Bảng 3.1. Bảng chú thích API thống kê phổ điểm.

```
HTTP 200 OK
Allow: OPTIONS, GET
Content-Type: application/json
Vary: Accept

{
  "unit_id": 1,
  "generation_id": 1,
  "major_id": 1,
  "course_id": 18,
  "boundary_type": "a-f",
  "sem_start": 1,
  "sem_end": 5,
  "number_of_grade": 50,
  "grade_distribute": {
    "F": 4,
    "D": 2,
    "C": 16,
    "B": 21,
    "A": 7
  },
  "statistics": {
    "avg": 7.02,
    "min": 0.0,
    "q1": 6.55,
    "q2": 7.25,
    "q3": 8.2,
    "max": 9.2
  }
}
```

Hình 3.1. Kết quả API thống kê phổ điểm.

grade_distribute là thống kê kết quả số lượng sinh viên đạt điểm với mỗi điểm số. Khi chọn {boundary_type} là a-f thì thống kê số lượng sinh viên đạt điểm từ a đến sinh viên đạt điểm f. Nếu chọn kiểu là 10 thì thống kê số lượng sinh viên đạt từ điểm 1 đến điểm 10 và khoảng cách là 1 điểm, nếu chọn kiểu 20 thì thống kê số lượng sinh viên đạt điểm 1 đến 10 và khoảng cách là 0.5 điểm.

statistics là các thống kê khác như là điểm trung bình, điểm thấp nhất, điểm cao nhất, tứ phân vị thứ nhất, tứ phân vị thứ hai và tứ phân vị thứ ba.

3.1.1.2. Thống kê trung bình điểm môn học qua các năm

Để lấy được dữ liệu thống kê về trung bình điểm số của môn học qua các năm học thì cần thực hiện truyền dữ liệu vào các tham số trong đường dẫn dưới đây:

domain/statistic/course_avg/<int:course_id>/<int:year_start>/<int:year_end>

Tham số	Mô tả
course_id	ID của môn học (bắt buộc)
year_start	Năm bắt đầu tính thống kê (bắt buộc)
year_end	Năm kết thúc tính thống kê (bắt buộc)

Bảng 3.2. Bảng chú thích API thống kê điểm môn học.

```

HTTP 200 OK
Allow: OPTIONS, GET
Content-Type: application/json
Vary: Accept

{
  "course_id": 1,
  "course_code": "MAT1093",
  "course_name": "Đại số",
  "year": {
    "2015-2016": {
      "avg_grade": 6.55,
      "num_of_grade": 127
    },
    "2016-2017": {
      "avg_grade": 6.82,
      "num_of_grade": 219
    },
    "2017-2018": {
      "avg_grade": 5.4,
      "num_of_grade": 9
    },
    "2018-2019": {
      "avg_grade": 2.65,
      "num_of_grade": 8
    },
    "2019-2020": {
      "avg_grade": 2.27,
      "num_of_grade": 4
    }
  }
}

```

Hình 3.2. Kết quả API thống kê điểm trung bình môn học.

avg_grade là điểm trung bình của môn học trong năm học tương ứng.

num_of_grade là số lượng điểm của sinh viên tham gia môn học đó ở năm học tương ứng.

3.1.1.3. Thống kê GPA theo khóa và ngành

Để lấy được dữ liệu thống kê về điểm GPA của toàn bộ khóa thuộc ngành thì cần thực hiện truyền dữ liệu vào các tham số trong đường dẫn dưới đây:

domain/statistic/gpa/<int:unit_id>/<int:major_id>/<int:generation_id>/<str:method>

Tham số	Mô tả
unit_id	ID của trường (bắt buộc)
major_id	ID chuyên ngành (bắt buộc)
generation_id	ID của các khóa (bắt buộc)
method	Hành động cần thực hiện gồm get (lấy GPA của tất cả các kỳ đã tính), update (tính GPA ở các kỳ còn thiếu rồi đưa ra số GPA đã thay đổi), reevaluation (tính lại GPA ở tất cả các kỳ học của rồi đưa ra kết quả số GPA đã thay đổi)

Bảng 3.3. Bảng chú thích API thống kê GPA.

HTTP 200 OK
Allow: OPTIONS, GET
Content-Type: application/json
Vary: Accept

```
{
  "major_id": 1,
  "generation_id": 1,
  "semester_number": {
    "0": {
      "semesterGpa": {
        "max": 9.68,
        "min": 0.0,
        "avg": 6.15
      },
      "currentGpa": {
        "max": 9.68,
        "min": 0.0,
        "avg": 6.15
      }
    },
    "1": {
      "semesterGpa": {
        "max": 9.36,
        "min": 0.0,
        "avg": 6.57
      },
      "currentGpa": {
        "max": 9.33,
        "min": 1.49,
        "avg": 6.79
      }
    },
    "2": {
      "semesterGpa": {
        "max": 9.73,
        "min": 0.0,
        "avg": 5.98
      },
      "currentGpa": {
        "max": 9.49,
        "min": 0.0,
        "avg": 6.61
      }
    },
    "3": {
      "semesterGpa": {
        "max": 9.24,
        "min": 0.0,
        "avg": 6.91
      },
      "currentGpa": {
        "max": 9.42,
        "min": 3.17,
        "avg": 6.82
      }
    }
  }
}
```

Hình 3.3. Kết quả API thống kê GPA theo khóa và ngành.

semester_number tương ứng với thứ tự kỳ trong quá trình học đại học của sinh viên. Tính bắt đầu từ kỳ số 0.

semesterGpa điểm trung bình của toàn bộ khóa và ngành tính theo các môn mà sinh viên tham gia vào thời điểm kỳ đó.

currentGpa điểm trung bình của toàn bộ khóa và ngành tính theo tất cả các môn mà sinh đã tham gia học từ kỳ học đầu tiên.

3.1.1.4. Thống kê GPA theo sinh viên

Để lấy được dữ liệu thống kê về điểm GPA sinh viên thì cần thực hiện truyền dữ liệu vào các tham số trong đường dẫn dưới đây:

domain/statistic/gpa/student/<int:profile_id>/<str:method>

Tham số	Mô tả
profile_id	ID của sinh viên muốn tính điểm GPA
method	Hành động cần thực hiện bao gồm: get(lấy GPA của tất cả các kỳ đã tính), update(tính GPA ở các kỳ còn thiếu rồi đưa ra GPA của tất cả kỳ. Nên gọi sau khi cập nhập điểm và sang kỳ học mới), reevaluation(tính lại GPA ở tất cả các kỳ học của sinh viên rồi đưa ra).

Bảng 3.4. Bảng chú thích API thống kê GPA sinh viên.

```
HTTP 200 OK
Allow: OPTIONS, GET
Content-Type: application/json
Vary: Accept

{
  "student_info": {
    "id": 72,
    "unit_id": 1,
    "generation_id": 1,
    "major_id": 1
  },
  "semester_number": {
    "0": {
      "semester_id": 1,
      "semesterGpa": 6.84,
      "currentGpa": 6.84
    },
    "1": {
      "semester_id": 2,
      "semesterGpa": 5.79,
      "currentGpa": 6.33
    },
    "2": {
      "semester_id": 3,
      "semesterGpa": 5.38,
      "currentGpa": 6.02
    },
    "3": {
      "semester_id": 4,
      "semesterGpa": 6.35,
      "currentGpa": 6.31
    },
    "4": {
      "semester_id": 5,
      "semesterGpa": 6.98,
      "currentGpa": 6.59
    },
    "5": {
      "semester_id": 6,
      "semesterGpa": 7.1,
      "currentGpa": 6.87
    },
    "6": {
      "semester_id": 7,
      "semesterGpa": 6.7,
      "currentGpa": 6.94
    },
    "7": {
      "semester_id": 8,
      "semesterGpa": 8.39,
      "currentGpa": 7.18
    }
  }
}
```

Hình 3.4. Kết quả API thống kê GPA theo sinh viên.

semester_number là số tương ứng với thứ tự kỳ trong quá trình học đại học của sinh viên. Tính bắt đầu từ kỳ số 0. Với thuộc tính **semester_id** cũng là thứ tự của kỳ học. **semesterGpa** là điểm trung bình của sinh viên tính theo các môn mà sinh viên tham gia vào thời điểm kỳ đó.

currentGpa là điểm trung bình của sinh viên tính theo tất cả các môn mà sinh đã tham gia học từ kỳ học đầu tiên.

3.1.2. Dự đoán

3.1.2.1. Dự đoán điểm môn học

Để lấy được kết quả điểm môn học đã dự đoán của sinh viên thì cần truyền dữ liệu vào đường dẫn và sử dụng gửi đi với phương thức get:

```
domain/recommend/predict_grade/student/<int:student_id>
```

student_id là id của sinh viên thực hiện dự báo điểm.

Để gửi đi yêu cầu dự đoán điểm về một môn học thì cần truyền dữ liệu các môn học trong phương thức post. Phần thân có dạng: course_id: 1, course_id: 2...

3.1.2.2. Dự đoán điểm toàn khóa

Để thực hiện tính điểm dự đoán của tất cả sinh viên thuộc một khóa trong cùng chuyên ngành theo các môn đã định thì cần truyền ngành vào đường dẫn:

```
domain/recommend/predict_grade/generation/<int:major_id>
```

major_id là id của ngành.

Khi thực hiện phương thức post thì cần đặt thân là generation_id= id của khóa.

3.1.3. Gợi ý

Để lấy được dữ liệu gợi ý các môn học cho sinh viên trong kỳ học tới của sinh viên cần truyền các giá trị vào đường dẫn sau:

```
domain/recommend/course/<int:student_id>/<str:method>/<int:k>
```

Tham số	Mô tả
student_id	ID của sinh viên muốn nhận gợi ý môn học
method	Hành động cần thực hiện bao gồm: greatest(môn học mà có kết quả cao nhất), similar(dựa trên điểm số kinh nghiệm của các sinh viên tương đồng)
k	Số lượng môn học muốn được gợi ý môn học

Bảng 3.5. Bảng chú thích API tính năng gợi ý môn học.

3.2. Cài đặt và thử nghiệm

3.2.1. Cài đặt

3.2.1.1. Cài đặt python

Các cài đặt để phát triển là sử dụng hệ điều hành ubuntu, sử dụng giao diện câu lệnh để install và cấu hình. Trên hệ điều hành Window thì có thể tham khảo tại các nguồn khác.

Bước 1: Cài đặt software-properties-common để quản lý các gói trong PPA (Personal Package Archive) tốt hơn.

```
sudo apt install software-properties-common
```

Bước 2: Thêm deadsnakes là PPA với phiên bản mới hơn.

```
sudo add-apt-repository ppa:deadsnakes/ppa
```

Bước 3: Cài đặt python3.

```
sudo apt install python3.8
```

Sau khi cài đặt python3 thành công thì có thể kiểm tra phiên bản trên máy bằng câu lệnh:

```
python3 --version
```

3.2.1.2. Cài đặt project Django

Bước 1: Cài đặt pip

Để có thể các thư viện, hay các thư viện liên quan đến python thì cần cài đặt pip để quản lý các gói cài đặt. Cài đặt bằng câu lệnh:

```
sudo apt install python3-pip
```

Kiểm tra phiên bản:

```
pip3 --version
```

Bước 2: Cài đặt django

Sử dụng câu lệnh để cài đặt framework django

```
pip3 install Django
```

Kiểm tra phiên bản django trên máy bằng câu lệnh

```
django-admin --version
```

Bước 3: Cài đặt môi trường sử dụng python độc lập

Python có thể được sử dụng nhiều nơi trong các dự án và cả hệ điều hành nên mỗi dự án nên tạo môi trường sử dụng các gói, thư viện liên quan đến python độc lập. Để các gói, thư viện của các dự án không dư thừa, tăng hiệu năng khi sử dụng và dễ quản lý các thư viện bên ngoài mà dự án có sử dụng. Cài đặt môi trường:

```
sudo install python3-venv
```

Tạo ra một thư mục chính để lưu dự án và tiến hành tạo môi trường độc lập cho dự án trong thư mục đó bằng câu lệnh

```
python3 -m venv my_env
```

Sau khi thực hiện thành công câu lệnh thì sẽ tạo ra thư mục là my_env. Đây chính là môi trường chứa các thư viện riêng của dự án mà chỉ có thể sử dụng trong môi trường này. Mỗi lần sử dụng môi trường thì chỉ ra là sử dụng môi trường nào:

```
source my_env/bin/activate
```

Để thoát khỏi môi trường thì sử dụng câu lệnh

```
Deactivate
```

Bước 4: Tạo dự án

Tạo ra khung của dự án sử dụng framework django:

```
django-admin startproject statistic_recommendation
```

3.2.1.3. Cài đặt các thư viện khác

Bước 1: Cài đặt

Cách 1: Cài đặt từng thư viện

Khi muốn sử dụng thư viện python nào thì sử dụng câu lệnh:

```
pip3 install <thư viện python>
```

Cách 2: Cài đặt theo file requirements

Tạo một file requirements.txt và thêm các tên thư viện trong file requirements.txt. Mỗi file ở trên một dòng. Và thực hiện cài đặt toàn bộ thư viện có trong file requirements bằng câu lệnh:

```
pip3 install requirements.txt
```

Bước 2: Khai báo sử dụng

Thêm tên thư viện sử dụng trong file requirements.txt vào file setting.py ở thư mục chính.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'recommendation',  
    'statistic',  
    'bootstrap4',  
    'mainApp',  
    'rest_framework',  
    'crispy_forms',  
]
```

Hình 3.5. Các thư viện sử dụng trong toàn bộ dự án.

3.2.2. Khởi động chương trình

3.2.2.1. Makemigrations

Lệnh makemigrations để tạo file migration. File migration là cách django quản lý các thay đổi liên quan đến đối tượng hoặc cấu trúc cơ sở dữ liệu được viết trong models.py. Vì vậy đầu tiên cần thực hiện lệnh makemigrations.

```
python3 manage.py makemigrations
```

3.2.2.2. Migrate

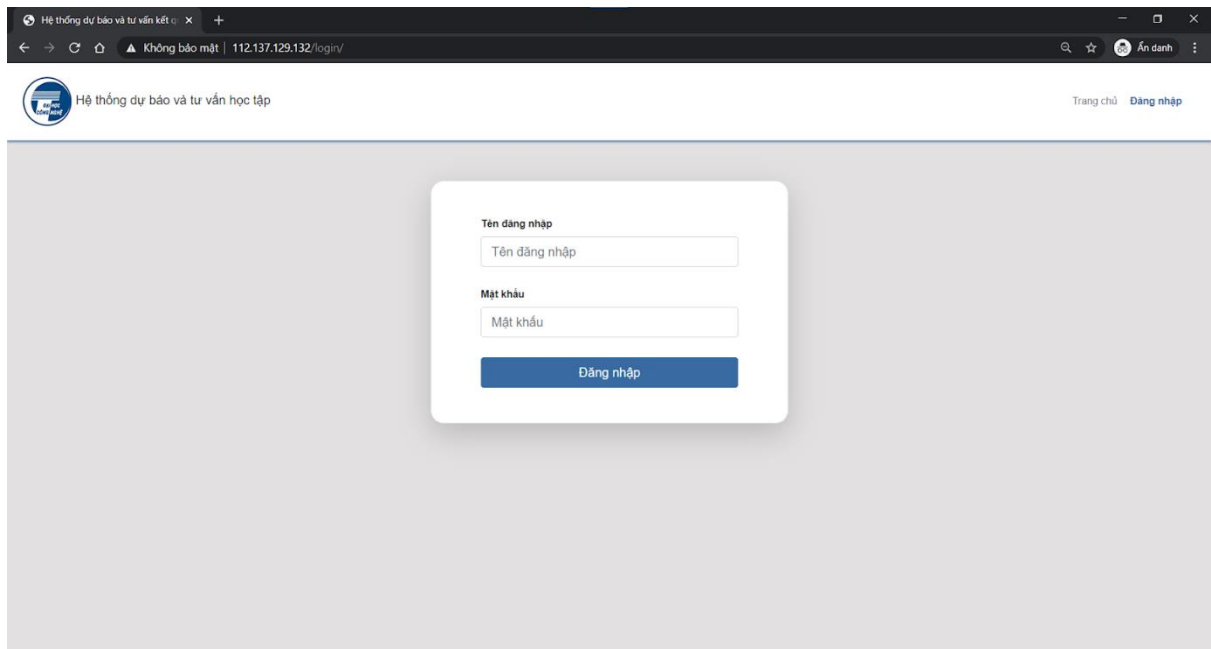
Sau mỗi lần thay đổi cấu trúc liên quan đến cơ sở dữ liệu trong models.py thì cần migrate để django lưu lại thay đổi và cập nhật vào cấu trúc của cơ sở dữ liệu.

```
python3 manage.py migrate
```

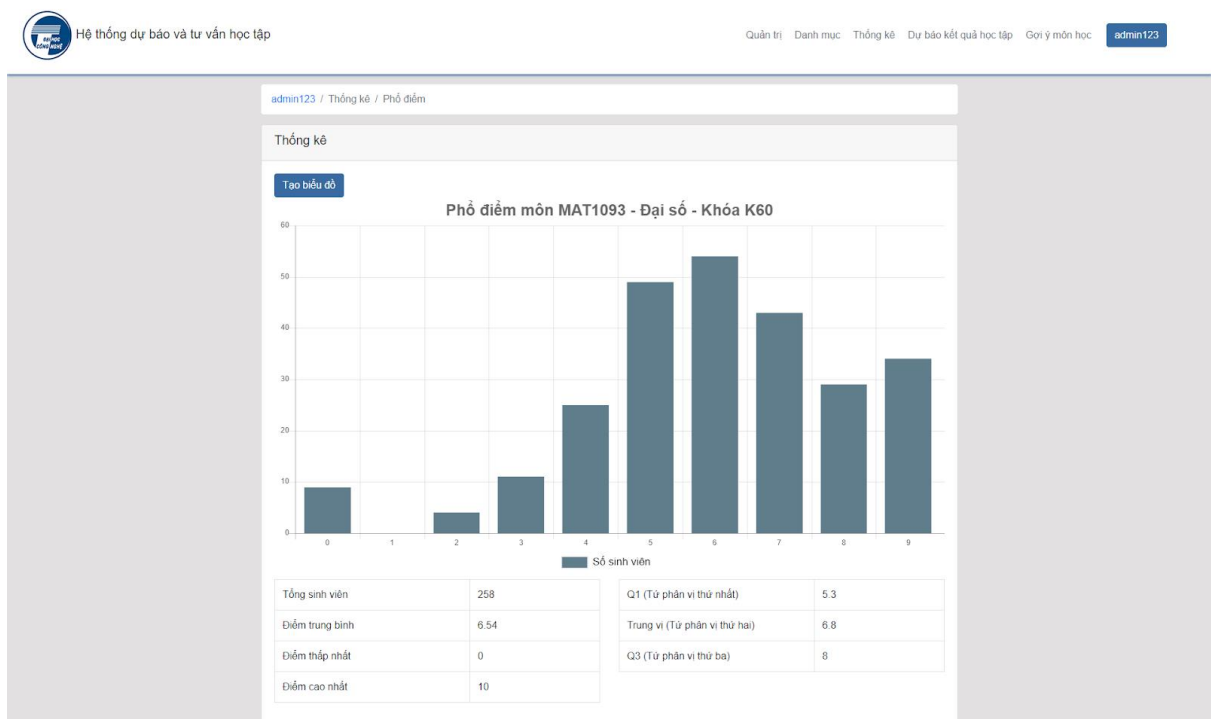
Sau khi đồng nhất cấu trúc dữ liệu thì thực hiện chạy để xem hoạt động của ứng dụng thì sử dụng câu lệnh.

```
python3 manage.py runserver
```

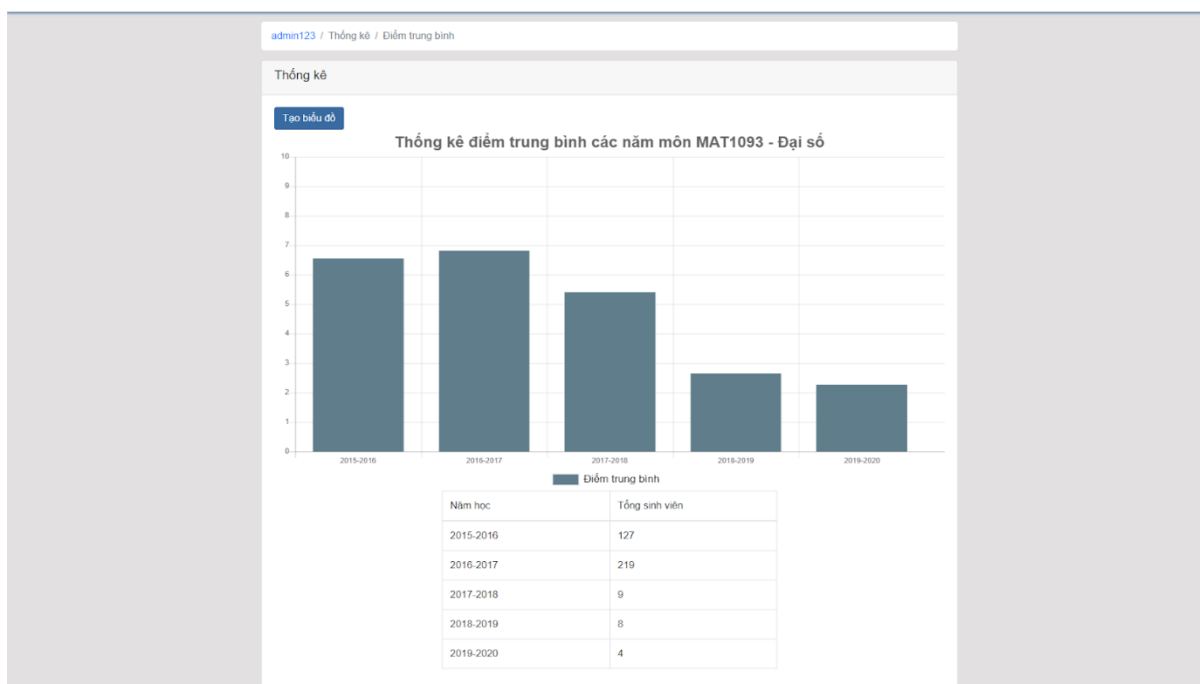
3.2.3. Kết quả thu được



Hình 3.6. Giao diện đăng nhập của hệ thống.



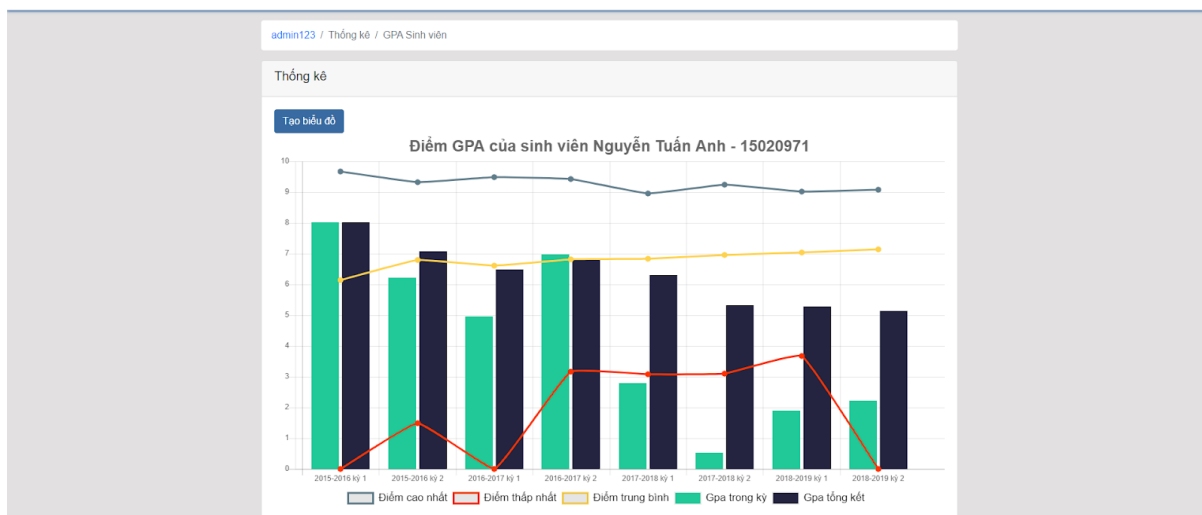
Hình 3.7. Thống kê phổ điểm.



Hình 3.8. Thống kê điểm trung bình.



Hình 3.9. Thống kê GPA.



Hình 3.10. Thống kê GPA sinh viên.



admin123 / Dự báo kết quả học tập

Điểm GPA dự đoán

Sinh viên: 20020072 - Tạ Hữu Vương

Tính dự báo theo khóa: Tính dự báo theo khóa

Tính dự báo theo môn: Tính dự báo theo môn

STT	Mã môn học	Tên môn học	Điểm dự đoán
1	FLF2102	Tiếng Anh cơ sở 2	8.45
2	HIS1002	Đường lối cách mạng của Đảng Cộng sản Việt Nam	5.55
3	INT2208	Công nghệ phần mềm	6.42
4	FLF2105	Tiếng Anh cơ sở 5	7.28
5	PHY1103	Điện và Quang	7.51
6	MAT1042	Giải tích 2	7.22
7	INT2204	Lập trình hướng đối tượng	7.64
8	INT3512	Lập trình thi đấu	3.27
9	POL1001	Tư tưởng Hồ Chí Minh	6.69
10	FLF2104	Tiếng Anh cơ sở 4	6.22

Hình 3.11. Kết quả dự đoán điểm của sinh viên.



admin123 / Gợi ý môn học

Môn học đề xuất

Sinh viên: 20020072 - Tạ Hữu Vượng

Số môn: 5

Cách gợi ý: Điểm cao nhất

Xem gợi ý

STT	Mã môn học	Tên môn học	Số tín chỉ
1	FLF2102	Tiếng Anh cơ sở 2	5
2	FLF2105	Tiếng Anh cơ sở 5	5
3	PHY1103	Điện và Quang	3
4	MAT1042	Giải tích 2	4
5	INT2204	Lập trình hướng đối tượng	3

Phiên bản thử nghiệm sản phẩm phần mềm của Đề tài QG.20.57.
© Bản quyền thuộc Đại học Quốc gia Hà Nội, Trường Đại học Công nghệ
Nhóm tác giả thực hiện đề tài.

Hình 3.12. Các môn học gợi ý cho sinh viên.

3.3. Đánh giá

Ưu điểm

Hệ thống đã hoàn thiện các chức năng để đáp ứng được nhu cầu về nguồn số liệu thống kê, và số liệu hiệu quả để tư vấn cho môn học và gợi ý điểm cho các bạn sinh viên. Các thao tác thực hiện tra cứu cũng như tìm kiếm, cập nhật thông tin cho sinh viên, quản trị cấp trường và quản trị hệ thống dễ sử dụng và có hướng dẫn thao tác đầy đủ.

Giao diện và chức năng tương ứng với quyền của người dùng. Phạm vi hoạt động của người dùng chỉ trong giới hạn quyền của vai trò. Vì thế thông tin cá nhân của sinh viên sẽ được đảm bảo an toàn và bí mật.

Hiện tại trên hệ thống đã có bộ dữ liệu điểm của K60, K61, K62, K63, K64, K65. Tài khoản sinh viên đã bước đầu kết hợp với hệ thống xác thực LDAP của trường Đại học Quốc gia Hà Nội.

Nhược điểm

Hệ thống có dự báo điểm cho sinh viên những hệ thống chưa có phát triển thêm tính năng lấy phản hồi, xác thực thông tin phản hồi để tính sai lệch của điểm dự đoán và điểm thực tế.

Nguồn điểm nhập vào hệ thống chưa đa dạng vì hiện tại hệ thống chỉ nhận nguồn dữ liệu định dạng csv.

Kết luận

Kiến thức sử dụng

Áp dụng các kiến thức, kỹ thuật phát triển và kiểm thử phần mềm để xây dựng trang web giải quyết bài toán tư vấn học tập.

Sử dụng framework để giải quyết được bài toán nghiệp vụ trong bài toán tư vấn học tập. Hơn nữa còn tận dụng tối đa các thư viện, công cụ hỗ trợ của framework để giải quyết và tối ưu các bài toán con trong tổng thể khóa luận.

Để có thêm giao diện đẹp và các thao tác mượt mà hợp tích hợp và sử dụng thành công các thư viện bootstrap, jquery. Ngoài ra để có đa dạng biểu đồ cho các thống kê thì sử dụng chartjs.

Kiến thức đạt được

Hiểu thêm về các hoạt động, kiến thức về framework Django và ngôn ngữ python. Từ đó nhận biết được ưu và nhược điểm của framework cũng như ngôn ngữ để biết cách áp dụng trong công việc.

Các kỹ năng giúp việc phát triển sản phẩm hạn chế các lỗi người dùng do có áp dụng các kiến thức về kiểm thử chất lượng sản phẩm.

Hướng cải thiện và phát triển

Để sản phẩm hoàn thiện hơn thì cần thêm tính năng tính toán hoặc thu thập phản hồi của sinh viên đã nhận kết quả điểm dự đoán và môn học gợi ý. Từ đó có thêm một trong số là tính đúng đắn của các con số mà hệ thống gợi ý.

Để tiện lợi hơn cho các quản trị hệ thống và quản lý cấp trường trong việc nhập liệu các file dữ liệu về điểm số, môn học thì cần phát triển thêm việc tiếp nhận các file dữ liệu là file excel.

TÀI LIỆU THAM KHẢO

- [1]. Holovaty A, Kaplan-Moss J. *The Definitive Guide to Django: Web Development Done Right*. 2009.
- [2]. Rubio D. *Beginning Django*. 2017.
- [3]. Read the Docs, Django Documentation Release 3.1.8.dev: Django Software Foundation.