**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Vũ Trung Anh**

**NGHIÊN CỨU CÔNG CỤ DJANGO VÀ**

**ÁP DỤNG PHÁT TRIỂN FRONT-END**

**CHO BÀI TOÁN 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 CÔNG CỤ DJANGO VÀ**

**ÁP DỤNG PHÁT TRIỂN FRONT-END**

**CHO BÀI TOÁN DỰ BÁO KẾT QUẢ HỌC TẬP**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

C**án 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. Để thật sự tiện ích thì ngoài sự dễ sử dụng, thao tác sử dụng mượt mà thì các dữ liệu cũng góp phần tạo nên sự hữu ích cho trông thông tin cung cấp cho người dùng. Đặt biệt với những nghiệp vụ tư vấn, dự đoán thì dữ liệu và cách thức đưa thông tin cho người dùng quyết định đến hiệu quả của việc sử dụng ứng dụng. Khóa luận này triển khai nội dung là dựa vào công nghệ phát triển ứng dụng website để 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.

**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 tháng năm 2021

Sinh viên

Vũ Trung Anh

[Mở đầu 13](#_Toc67783380)

[Chương 1. Bài toán và Công nghệ 15](#_Toc67783381)

1. [1.1. Framework Django 15](#_Toc67783382)

[1.1.1. Cơ chế hoạt động 15](#_Toc67783383)

[1.1.1.1. Thông tin chung 15](#_Toc67783384)

[1.1.1.2. Mô hình hoạt động 15](#_Toc67783385)

[1.1.2. Cách thức phát triển 16](#_Toc67783386)

[1.1.2.1. Biểu đồ tuần tự 16](#_Toc67783387)

[1.1.2.2. Chi tiết 16](#_Toc67783388)

1. [1.2. Cách thức sử dụng Framework Django 16](#_Toc67783389)

[1.2.1. Đối tượng 16](#_Toc67783390)

[1.2.1.1. Dữ liệu của thuộc tính 16](#_Toc67783391)

[1.2.1.2. Mối quan hệ 17](#_Toc67783392)

[1.2.2. Xử lý logic 19](#_Toc67783393)

[1.2.2.1. Dữ liệu liên quan đến các đối tượng 19](#_Toc67783394)

[1.2.2.2. Các thư viện khác 20](#_Toc67783395)

[1.2.2.3. Dữ liệu hiển thị 21](#_Toc67783396)

[1.2.3. URL 22](#_Toc67783397)

[1.2.3.1. Đường dẫn thành phần 22](#_Toc67783398)

[1.2.3.2. Đường dẫn chi tiết 22](#_Toc67783399)

[1.2.4. Giao diện 22](#_Toc67783400)

[1.2.4.1. Xây dựng khung giao diện 22](#_Toc67783401)

[1.2.4.2. Xử lý dữ liệu 23](#_Toc67783402)

[1.2.5. Phân quyền 23](#_Toc67783403)

[1.2.6. API 24](#_Toc67783404)

[Chương 2. Phân tích và thiết kế 25](#_Toc67783405)

1. [2.1. Phân tích 25](#_Toc67783407)

[2.1.1. Các tác nhân hệ thống 25](#_Toc67783408)

[2.1.1.1. Các hạng mục người quản lý có quyền thao tác 25](#_Toc67783409)

[2.1.1.2. Yếu tố ảnh hưởng đến người quản lý 25](#_Toc67783410)

[2.1.2. Sinh viên 25](#_Toc67783411)

[2.1.2.1. Các hạng mục sinh viên có thể thao tác 25](#_Toc67783412)

[2.1.2.2. Yêu cầu thao tác với thông tin tư vấn của sinh viên 25](#_Toc67783413)

[2.1.3. Quản trị 26](#_Toc67783414)

[2.1.3.1. Việc phân quyền chính của hệ thống 26](#_Toc67783415)

[2.1.3.2. Cập nhật dữ liệu liên quan đến tư vấn học tập 26](#_Toc67783416)

1. [2.2. Ca sử dụng 27](#_Toc67783417)

[2.2.1. Sơ đồ chính 27](#_Toc67783418)

[2.2.2. Chi tiết ca sử dụng 28](#_Toc67783419)

[2.2.2.1. Quản lý người dung 28](#_Toc67783420)

[2.2.2.2. Quản lý danh mục trường 28](#_Toc67783421)

[2.2.2.3. Quản lý ngành đào tạo 30](#_Toc67783422)

[2.2.2.4. Quản lý môn học 32](#_Toc67783423)

[2.2.2.5. Quản lý chương trình đào tạo 34](#_Toc67783424)

[2.2.2.6. Quản lý kết quả học tập 36](#_Toc67783425)

[2.2.2.7. Quản lý khóa học 38](#_Toc67783426)

[2.2.2.8. Quản lý lớp 40](#_Toc67783427)

[2.2.2.9. Quản lý năm học 42](#_Toc67783428)

[2.2.2.10. Quản lý kỳ học 44](#_Toc67783429)

[2.2.2.11. Quản lý chức năng hệ thống 46](#_Toc67783430)

[2.2.2.12. Lựa chọn mô hình 48](#_Toc67783431)

[2.2.2.13. Thống kê phổ điểm theo môn và khóa 49](#_Toc67783432)

[2.2.2.14. Thống kê điểm trung bình môn học qua các năm 49](#_Toc67783433)

[2.2.2.15. Thống kê điểm GPA theo khóa và ngành 50](#_Toc67783434)

[2.2.2.16. Thống kê điểm GPA theo sinh viên 51](#_Toc67783435)

[2.2.2.17. Dự đoán kết quả học tập 51](#_Toc67783436)

[2.2.2.18. Gợi ý môn học 52](#_Toc67783437)

1. [2.3. Thiết kế 54](#_Toc67783438)

[2.3.1. Thiết kế kiến trúc hệ thống 54](#_Toc67783439)

[2.3.1.1. Tổng quan về cấu trúc 54](#_Toc67783440)

[2.3.1.2. Mô tả chi tiết 55](#_Toc67783441)

[2.3.2. Thiết kế cơ sở dữ liệu 56](#_Toc67783442)

[2.3.2.1. Tổng quan về cơ sở dữ liệu 56](#_Toc67783443)

[2.3.2.2. Mô tả chi tiết 56](#_Toc67783444)

[2.3.3. Thiết kế biểu đồ tuần tự 64](#_Toc67783445)

[2.3.3.1. Tính năng xem danh sách 64](#_Toc67783446)

[2.3.3.2. Tính năng thêm mới, chỉnh sửa nội dung 65](#_Toc67783447)

[2.3.3.3. Tính năng xóa nội dung 66](#_Toc67783448)

[2.3.3.4. Tính năng xuất file csv 67](#_Toc67783449)

[2.3.3.5. Tính năng nhập file csv 68](#_Toc67783450)

[2.3.3.6. Tính năng thống kê theo môn và khóa 68](#_Toc67783451)

[2.3.3.7. Tính năng thống kê điểm trung bình môn học qua các năm 68](#_Toc67783452)

[2.3.3.8. Thống kê GPA theo khóa và ngành 69](#_Toc67783453)

[2.3.3.9. Tính năng dự đoán điểm số 70](#_Toc67783454)

[2.3.3.10. Thống kê GPA sinh viên 71](#_Toc67783455)

[2.3.3.11. Tính năng gợi ý môn học 72](#_Toc67783456)

[2.3.3.12. Tính năng đăng nhập 73](#_Toc67783457)

[2.3.3.13. Tính năng đăng xuất 74](#_Toc67783458)

[Chương 3. Cài đặt thử nghiệm và đánh giá 74](#_Toc67783459)

1. [3.1. Sử dụng API 74](#_Toc67783461)

[3.1.1. Thống kê 74](#_Toc67783462)

[3.1.1.1. Thống kê phổ điểm theo môn và khóa 74](#_Toc67783463)

[3.1.1.2. Thống kê trung bình điểm môn học qua các năm 75](#_Toc67783464)

[3.1.1.3. Thống kê GPA theo khóa và ngành 76](#_Toc67783465)

[3.1.1.4. Thống kê GPA theo sinh viên 77](#_Toc67783466)

[3.1.2. Dự đoán 78](#_Toc67783467)

[3.1.2.1. Dự đoán điểm môn học 78](#_Toc67783468)

[3.1.2.2. Dự đoán điểm toàn khóa 79](#_Toc67783469)

[3.1.3. Gợi ý 79](#_Toc67783470)

1. [3.2. Cài đặt và thử nghiệm 79](#_Toc67783471)

[3.2.1. Cài đặt 79](#_Toc67783472)

[3.2.1.1. Cài đặt python 79](#_Toc67783473)

[3.2.1.2. Cài đặt project Django 80](#_Toc67783474)

[3.2.1.3. Cài đặt các thư viện khác 81](#_Toc67783475)

[3.2.2. Khởi động chương trình 82](#_Toc67783476)

[3.2.2.1. Makemigrations 82](#_Toc67783477)

[3.2.2.2. Migrate 82](#_Toc67783478)

[3.2.3. Kết quả thu được 82](#_Toc67783479)

1. [3.3. Đánh giá 85](#_Toc67783480)

[Kết luận 85](#_Toc67783481)

**Danh mục hình ảnh**

[Hình 1.1. Biểu đồ tuần tự mô hình hoạt động 16](#_Toc67783482)

[Hình 1.2. Đối tượng có chứa khóa chính 18](#_Toc67783483)

[Hình 1.3. Đối tượng sử dụng khóa ngoại tham chiếu 18](#_Toc67783484)

[Hình 1.4. Đối tượng Unit Serialize 20](#_Toc67783485)

[Hình 1.5. Hàm trả dữ liệu dạng đã biến đối 20](#_Toc67783486)

[Hình 1.6. Đối tượng biểu mẫu 21](#_Toc67783487)

[Hình 1.7. Phương thức xử lý tạo mới và cập nhật 21](#_Toc67783488)

[Hình 1.8. Phương thức hiển thị giao diện có kiểm tra đăng nhập 21](#_Toc67783489)

[Hình 1.9. Phương thức trả về số trang về gửi hạn bản ghi để hiển thị 22](#_Toc67783490)

[Hình 1.10. Các thư viện sử dụng và định nghĩa middleware 23](#_Toc67783491)

[Hình 1.11. Hàm xử lý phương thức GET của API 24](#_Toc67783492)

[Hình 1.12. Đường dẫn của API. 24](#_Toc67783493)

[Hình 1.13. Hàm xử phương thức POST của API 24](#_Toc67783494)

[Hình 2.1. 27](#_Toc67783495)

[Hình 2. 2. 29](#_Toc67783496)

[Hình 2. 3. 30](#_Toc67783497)

[Hình 2. 4. 30](#_Toc67783498)

[Hình 2. 5. 31](#_Toc67783499)

[Hình 2. 6. 32](#_Toc67783500)

[Hình 2. 7. 32](#_Toc67783501)

[Hình 2. 8. 33](#_Toc67783502)

[Hình 2. 9. 34](#_Toc67783503)

[Hình 2.10. 34](#_Toc67783504)

[Hình 2. 11. 36](#_Toc67783505)

[Hình 2. 12. 36](#_Toc67783506)

[Hình 2. 13. 36](#_Toc67783507)

[Hình 2. 14. 38](#_Toc67783508)

[Hình 2. 15. 38](#_Toc67783509)

[Hình 2. 16. 38](#_Toc67783510)

[Hình 2. 17. 40](#_Toc67783511)

[Hình 2. 18. 40](#_Toc67783512)

[Hình 2. 19. 40](#_Toc67783513)

[Hình 2. 20. 42](#_Toc67783514)

[Hình 2. 21. 42](#_Toc67783515)

[Hình 2. 22. 42](#_Toc67783516)

[Hình 2. 23. 44](#_Toc67783517)

[Hình 2. 24. 44](#_Toc67783518)

[Hình 2. 25. 44](#_Toc67783519)

[Hình 2. 26. 46](#_Toc67783520)

[Hình 2. 27. 46](#_Toc67783521)

[Hình 2. 28. 46](#_Toc67783522)

[Hình 2. 29. 48](#_Toc67783523)

[Hình 2. 30. 48](#_Toc67783524)

[Hình 2. 31. 48](#_Toc67783525)

[Hình 2. 32. 49](#_Toc67783526)

[Hình 2. 33. 49](#_Toc67783527)

[Hình 2. 34. 50](#_Toc67783528)

[Hình 2. 35. 51](#_Toc67783529)

[Hình 2. 36. 51](#_Toc67783530)

[Hình 2. 37. 52](#_Toc67783531)

[Hình 2. 38. 52](#_Toc67783532)

[Hình 2. 39. 52](#_Toc67783533)

[Hình 2. 40. 53](#_Toc67783534)

[Hình 2.41. Cấu trúc thư mục 54](#_Toc67783535)

[Hình 2.42. Tổng quan về các bảng trong cơ sở dữ liệu 56](#_Toc67783536)

[Hình 2.43. Biểu đồ tuần tự tính năng xem danh sách 64](#_Toc67783537)

[Hình 2.44. Biểu đồ tuần tự tính năng tạo mới và sửa nội dung 65](#_Toc67783538)

[Hình 2.45. Biểu đồ tuần tự tính năng xóa nội dung 66](#_Toc67783539)

[Hình 2.46. Biểu đồ tuần tự tính năng xuất file csv 67](#_Toc67783540)

[Hình 2.47. Biểu đồ tuần tự tính năng thống kê theo môn học và khóa 68](#_Toc67783541)

[Hình 2.48. Biểu đồ tuần từ tính năng thống kê điểm trung bình môn học qua các năm 68](#_Toc67783542)

[Hình 2.49. Biểu đồ tuần tự tính năng thống kê GPA theo khóa và ngành 69](#_Toc67783543)

[Hình 2.50. Biểu đồ tuần tự tính năng dự đoán điểm số 70](#_Toc67783544)

[Hình 2.51. Biểu đồ tuần tự tính năng thống kê GPA sinh viên 71](#_Toc67783545)

[Hình 2.52. Biểu đồ tuần tự tính năng gợi ý môn học 72](#_Toc67783546)

[Hình 2.53. Biểu đồ tuần tự tính năng đăng nhập 73](#_Toc67783547)

[Hình 2.54. Biểu đồ tuần tự tính năng đăng xuất 74](#_Toc67783548)

[Hình 3.1. Kết quả API thống kê phổ điểm 75](#_Toc67783549)

[Hình 3.2. Kết quả API thống kê điểm trung bình môn học 76](#_Toc67783550)

[Hình 3.3. Kết quả API thống kê GPA theo khóa và ngành 77](#_Toc67783551)

[Hình 3.4. Kết quả API thống kê GPA theo sinh viên 78](#_Toc67783552)

[Hình 3.5. Các thư viện sử dụng trong toàn bộ dự án 82](#_Toc67783553)

[Hình 3.6. Các giao diện kết quả của hệ thống 83](#_Toc67783554)

[Hình 3.7. Thống kê phổ điểm 83](#_Toc67783555)

[Hình 3.8. Thống kê điểm trung bình 84](#_Toc67783556)

[Hình 3.9. Thống kê GPA 84](#_Toc67783557)

[Hình 3.10. Thống kê GPA sinh viên 85](#_Toc67783558)

**Danh mục bảng biểu**

[Bảng 1.1. Danh sách các kiểu dữ liệu trong Models 17](#_Toc67783559)

[Bảng 1.2. Bảng các giá trị của thuộc tính on\_delete 19](#_Toc67783560)

[Bảng 1.3. Các phương thức trong middleware 24](#_Toc67783561)

[Bảng 2.1. 29](#_Toc67783562)

[Bảng 2. 2. 31](#_Toc67783563)

[Bảng 2.3. 33](#_Toc67783564)

[Bảng 2.4. 35](#_Toc67783565)

[Bảng 2.5. 37](#_Toc67783566)

[Bảng 2.6. 39](#_Toc67783567)

[Bảng 2.7. 41](#_Toc67783568)

[Bảng 2.8. 43](#_Toc67783569)

[Bảng 2.9. 45](#_Toc67783570)

[Bảng 2.10. 47](#_Toc67783571)

[Bảng 2.11. 48](#_Toc67783572)

[Bảng 2.12. 49](#_Toc67783573)

[Bảng 2.13. 50](#_Toc67783574)

[Bảng 2.14. 50](#_Toc67783575)

[Bảng 2.15. 51](#_Toc67783576)

[Bảng 2.16. 52](#_Toc67783577)

[Bảng 2.17. 53](#_Toc67783578)

[Bảng 2.18. Cấu trúc bảng auth\_group 56](#_Toc67783579)

[Bảng 2.19. Cấu trúc bảng auth\_group\_permissions 56](#_Toc67783580)

[Bảng 2.20. Cấu trúc bảng auth\_permission 57](#_Toc67783581)

[Bảng 2.21 Cấu trúc bảng auth\_user\_groups 57](#_Toc67783582)

[Bảng 2.22. Cấu trúc bảng auth\_user\_user\_permissions 57](#_Toc67783583)

[Bảng 2.23. Cấu trúc bảng django\_admin\_log 57](#_Toc67783584)

[Bảng 2.24. Cấu trúc bảng django\_content\_type 58](#_Toc67783585)

[Bảng 2.25. Cấu trúc bảng mainapp\_courses 58](#_Toc67783586)

[Bảng 2.26. Cấu trúc bảng mainapp\_customer 58](#_Toc67783587)

[Bảng 2.27. Cấu trúc bảng mainapp\_customser\_function 59](#_Toc67783588)

[Bảng 2.28. Cấu trúc bảng mainapp\_dumpmodel 59](#_Toc67783589)

[Bảng 2.29. Cấu trúc bảng mainapp\_functions 59](#_Toc67783590)

[Bảng 2.30. Cấu trúc bảng mainapp\_generations 59](#_Toc67783591)

[Bảng 2.31. Cấu trúc bảng mainapp\_gpa 60](#_Toc67783592)

[Bảng 2.32. Cấu trúc bảng mainapp\_gradepredicted 60](#_Toc67783593)

[Bảng 2.33. Cấu trúc bảng mainapp\_logs 60](#_Toc67783594)

[Bảng 2.34. Cấu trúc bảng mainapp\_majors 60](#_Toc67783595)

[Bảng 2.35. Cấu trúc bảng mainapp\_major\_course 61](#_Toc67783596)

[Bảng 2.36. Cấu trúc bảng mainapp\_predicthistory 61](#_Toc67783597)

[Bảng 2.37. Cấu trúc bảng mainapp\_profiles 61](#_Toc67783598)

[Bảng 2.38. Cấu trúc bảng mainapp\_roles 62](#_Toc67783599)

[Bảng 2.39. Cấu trúc bảng mainapp\_role\_function 62](#_Toc67783600)

[Bảng 2.40. Cấu trúc bảng mainapp\_semesters 62](#_Toc67783601)

[Bảng 2.41. Cấu trúc bảng mainapp\_studentgroups 62](#_Toc67783602)

[Bảng 2.42. Cấu trúc bảng mainapp\_traindata 63](#_Toc67783603)

[Bảng 2.43. Cấu trúc bảng mainapp\_transcript 63](#_Toc67783604)

[Bảng 2.44. Cấu trúc bảng mainapp\_transcriptfile 63](#_Toc67783605)

[Bảng 2.45. Cấu trúc bảng mainapp\_units 63](#_Toc67783606)

[Bảng 2.46. Cấu trúc bảng mainapp\_years 64](#_Toc67783607)

[Bảng 3.1. Bảng chú thích API 75](#_Toc67783608)

[Bảng 3.2. Bảng chú thích API 75](#_Toc67783609)

[Bảng 3.3. Bảng chú thích API 76](#_Toc67783610)

[Bảng 3.4. Bảng chú thích API 78](#_Toc67783611)

[Bảng 3.5. Bảng chú thích API 79](#_Toc67783612)

# Mở đầu

**Tính cấp thiết**

Sự tư vấn là cần thiết: Theo mô hình đăng ký học theo tín chỉ hiển 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.

Các phương pháp tư vấn phổ biến: 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 để 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 bài toán cần đến những dữ liệu liên quan đến sinh viên và các 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 pháp 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 vị 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 lại phù hợp.

**Bố cục khóa luận**

Chương 1 trình bày về công cụ pháp triển web django và quy trình phát triển khi dựa 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 chỉnh của hệ thống, triển khai thử nghiệm và thu về kết quả để đánh giá hiệu quả hoạt động. Cuối cùng là kết luận.

# Bài toán và Công nghệ

## **Framework** **Django**

### **Cơ chế hoạt động**

#### 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 đến 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.

#### 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 hiển 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 xuất truy xuất dữ liệu. Ngoài ra nó còn giúp việc xây dựng hệ thống đơn giản mà đặt 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 quan giao điện thì người dùng có thể tương tác chính với hệ thống. Người dùng gửi yêu cầu, gửi thông tin đến hệ thống.

### **Cách thức phát triển**

#### Biểu đồ tuần tự



Hình 1.1. Biểu đồ tuần tự mô hình hoạt động

#### 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 xứ 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.

## **Cách thức sử dụng Framework Django**

### **Đối tượng**

#### 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 primary\_key=True 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 max\_length 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 auto\_now\_add=True. |
| 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

#### 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.



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.



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.

**Các thao tác liên quan**

|  |  |
| --- | --- |
| 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

### **Xử lý logic**

#### 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() |

#### 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, hiện 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 thực để biến đổi dữ liệu



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.



Hình 1.5. Hàm trả dữ liệu dạng đã biến đối

**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.



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.



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.



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

#### 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.



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.

### **URL**

#### Đường dẫn thành phần

Trong thư mục chính của dự án thì có một tệp để cầu 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'),  ] |

#### 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’)); |

### **Giao diện**

#### 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 “./header.html” %} để kế thừa lại những nội dung từ file header.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 %} |

#### Xử lý dữ liệu

Sử dụng các ký thiệu 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 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 %} |

### **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 báo nó cả ở trong file setting.py là file cấu hình chinh của dự án.



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 |
| \_\_init\_\_ | Thực hiện nạp dữ liệu và khởi chạy duy nhất lần đầu tiên khi thực hiện bắt đầu chương trình. |
| \_\_call\_\_ | Thực hiện gọi khi mỗi lần trước khi thực hiện yêu cầu và phản hồi |
| process\_exception | Thực hiện tiến trình khi xảy ra ngoại lệ để thực hiện. |
| process\_template\_response | Thực hiện sau quá trình phản hồi để hiển thị phản hồi |

Bảng 1.3. Các phương thức trong middleware

### **API**

Để sử dụng api thì cùng 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 sự theo đoạn code mẫu ở dưới.



Hình 1.11. Hàm xử lý phương thức GET của API

Sau đó định dạng thiết kế đường dẫn trong file urls



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()



Hình 1.13. Hàm xử phương thức POST của API

# Phân tích và thiết kế



## **Phân tích**

### **Các tác nhân hệ thống**

#### 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 tố 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 vị 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.

#### 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 vị 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.

### **Sinh viên**

#### 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 để 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.

#### **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.

### **Quản trị**

#### Việc phân quyền chính của hệ thống

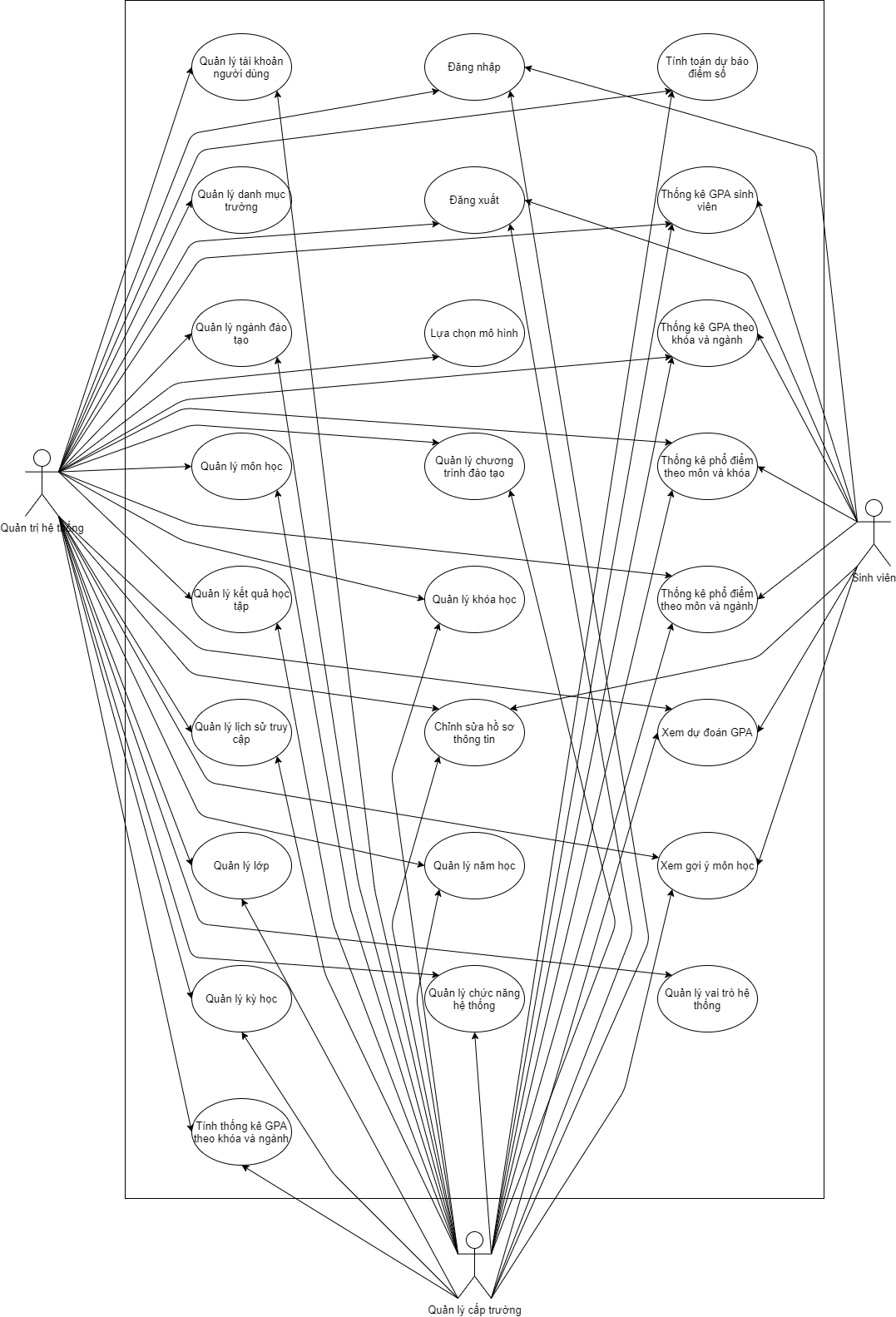
Quản trị hệ thống là vai trò quản lý cao nhất của hệ thống. Quản trị hệ thống có quyền thao tác với tất cả phạm vi của tất cả các trường có trong hệ thống. Từ việc quản lý tất cả các môn học, chương trình đào tạo...

#### **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.

## **Ca sử dụng**

### **Sơ đồ chính**



Hình 2.1.

### **Chi tiết ca sử dụng**

#### **Quản lý người dung**

**Mô tả tóm tắt**

Người dùng quản trị hệ thống cập nhật các thông tin liên quan đến tài khoản người dùng như: thêm tài khoản, cập nhật tài khoản, xóa tài khoản.

**Luồng sự kiện**

luồng cơ bản (luồng chính khi thao tác)

luồng thay thế (các luồng điều kiện có thể xuất hiện trong luồng chính

**Thông tin liên quan**

**Yêu cầu đặc biệt**

**Điều kiện đầu**

**Điều kiện cuối**

**Các vấn đề mở**

**Biểu đồ hoạt động**

#### **Quản lý danh mục trường**

**Mô tả tóm tắt**

Người dùng quản trị hệ thống cập nhật thông tin về các cơ sở đào tạo, trường đại học như: thêm mới trường, cập nhật tên và mô tả về trường, xóa trường khỏi hệ thống.

**Luồng sự kiện**

* 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 trường 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 tên trường người dùng nhập vào và thực hiện thêm Trường vào hệ thống.Khi thực hiện thành công thì hệ thống hiển thị màn hình danh sách các trường. | Dữ liệu trường trong hệ thống. |
| 2. Cập nhật | 2.1. Nhấn nút Sửa ở trường muốn cập nhật dữ liệu | 2.2. Hệ thống hiển thị màn hình có biểu mẫu chứa thông tin của trường muốn cập nhật | Dữ liệu trường gồm tên trường, mô tả. |
| 2.3. Cập nhật thông tin trường trong biểu mẫu và nhấn nút Cập nhật | 2.4. Hệ thống thực hiện kiểm tra tên trường người dùng nhập vào và thực hiện cập nhật thông tin trong biểu mẫu vào hệ thống. | Dữ liệu trường trong hệ thống. |
| 3. Xóa | 3.1. Nhấn nút Xóa ở trường 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 trường | Id của trường trong hệ thống. |

Bảng 2.1.

* 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 trường đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách các trường và thông báo lỗi cho người dùng.

Tại bước 2.4: Khi tên trường 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 trường đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách các trường và thông báo lỗi cho người dùng.

Tại bước 3.4: Khi trường người dùng muốn xóa không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách các trường 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**

Không có

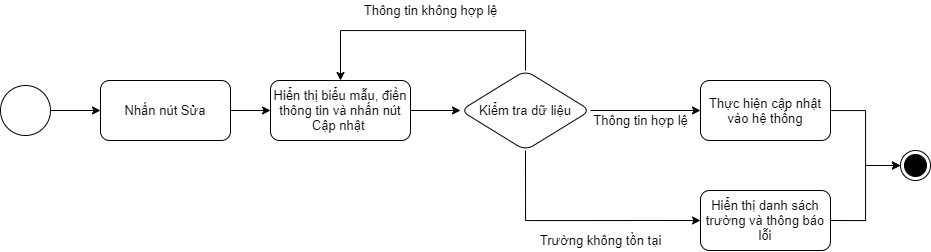
**Biểu đồ hoạt động**

Thêm mới



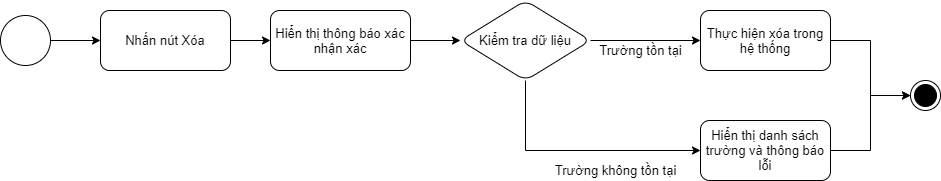
Hình 2. 2.

Cập nhật



Hình 2. 3.

Xóa



Hình 2. 4.

#### **Quản lý ngành đào tạo**

**Mô tả tóm tắt**

Người dùng cập nhật thông tin về các ngành đào tạo như: thêm mới ngành, cập nhật tên ngành, thuộc trường và mô tả về ngành, xóa ngành khỏi hệ thống.

**Luồng sự kiện**

* 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 ngành mới. | Dữ liệu người dùng nhập.  Dữ liệu về trường. |
| 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 ngành vào hệ thống.Khi thực hiện thành công thì hệ thống hiển thị màn hình danh sách các ngành. | Dữ liệu ngành và trường trong hệ thống. |
| 2. Cập nhật | 2.1. Nhấn nút Sửa ở ngành muốn cập nhật dữ liệu. | 2.2. Hệ thống hiển thị màn hình có biểu mẫu chứa thông tin của ngành muốn cập nhật. | Dữ liệu trường, ngành tương ứng. |
| 2.3. Cập nhật thông tin ngành 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 ngành và trường trong hệ thống. |
| 3. Xóa | 3.1. Nhấn nút Xóa ở ngành 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 ngành. | Dữ liệu là id của ngành trong hệ thống. |

Bảng 2. 2.

* 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 ngành đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách các ngành và thông báo lỗi cho người dùng.

Tại bước 2.4: Khi tên ngành 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 ngành đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách các ngành và thông báo lỗi cho người dùng.

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

**Yêu cầu đặc biệt**

Người dùng bắt buộc là tài khoản quản trị hệ thống hoặc là quản lý cấp trường.

Quản lý cấp trường chỉ thao tác được ngành thuộc trường mà phạm vi tài khoản hoạt động.

Các ngành có tên khác nhau.

**Điều kiện tiên quyết**

Người dùng phải đăng nhập vào hệ thống.

**Các vấn đề mở**

Không có

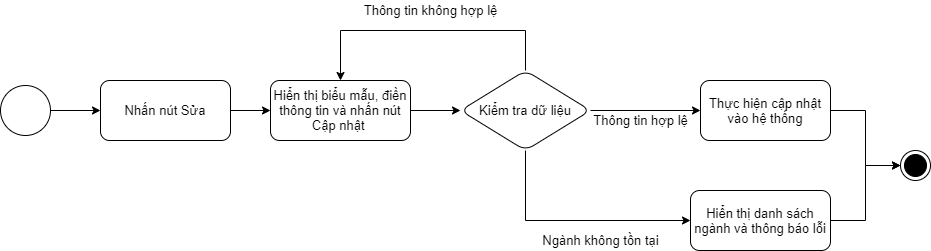
**Biểu đồ hoạt động**

Thêm mới

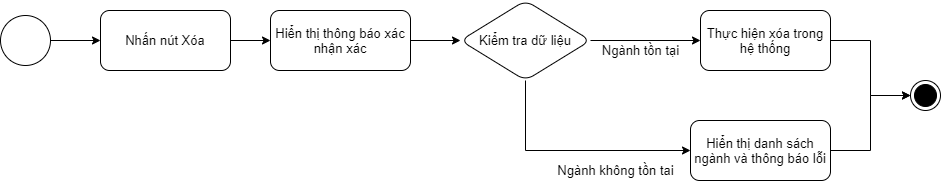


Hình 2. 5.

Cập nhật



Hình 2. 6.

Xóa

Hình 2. 7.

#### **Quản lý môn học**

**Mô tả tóm tắt**

Người dùng cập nhật thông tin về các môn học như: thêm mới môn học, cập nhật thông tin môn học, xóa môn học khỏi hệ thống.

**Luồng sự kiện**

* 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 môn học mới. | Dữ liệu người dùng nhập.  Dữ liệu về trường. |
| 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 môn học vào hệ thống.Khi thực hiện thành công thì hệ thống hiển thị màn hình danh sách các môn học. | Dữ liệu trường và môn học trong hệ thống. |
| 2. Cập nhật | 2.1. Nhấn nút Sửa ở môn học muốn cập nhật dữ liệu. | 2.2. Hệ thống hiển thị màn hình có biểu mẫu chứa thông tin của môn học muốn cập nhật. | Dữ liệu trường, môn học tương ứng. |
| 2.3. Cập nhật thông tin ngành 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 ngành và trường trong hệ thống. |
| 3. Xóa | 3.1. Nhấn nút Xóa ở môn học 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 môn học. | Dữ liệu là id của môn học trong hệ thống. |

Bảng 2.3.

* 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 môn học đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách các môn học và thông báo lỗi cho người dùng.

Tại bước 2.4: Khi tên môn học 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 môn học đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách các môn và thông báo lỗi cho người dùng.

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

**Yêu cầu đặc biệt**

Người dùng bắt buộc là tài khoản quản trị hệ thống hoặc là quản lý cấp trường.

Quản lý cấp trường chỉ thao tác được môn học thuộc trường mà phạm vi tài khoản hoạt động.

Các môn có mã môn, tên khác nhau.

**Điều kiện tiên quyết**

Người dùng phải đăng nhập vào hệ thống.

**Các vấn đề mở**

Không có

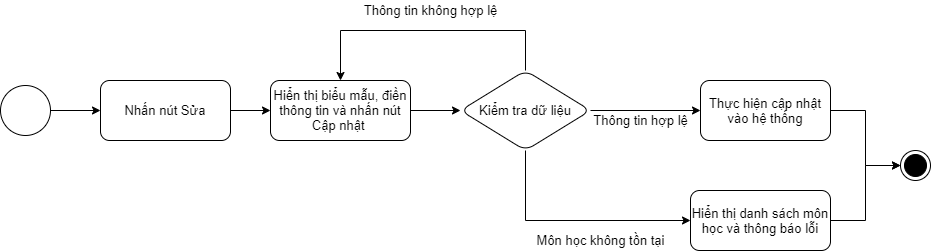
**Biểu đồ hoạt động**

Thêm mới



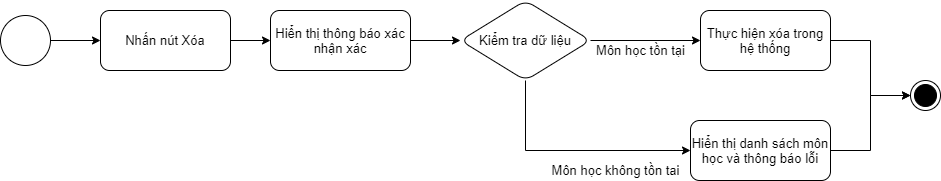
Hình 2. 8.

Cập nhật



Hình 2. 9.

Xóa



Hình 2.10.

#### **Quản lý chương trình đào tạo**

**Mô tả tóm tắt**

Người dùng cập nhật thông tin về các chương trình đào tạo như: thêm mới môn học cho chương trình đào tạo, cập nhật thông tin ngành đào tạo, xóa môn học trong ngành đào tạo khỏi hệ thống.

**Luồng sự kiện**

* 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 môn học cho ngành. | Dữ liệu người dùng nhập.  Dữ liệu về môn học, ngành. |
| 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 vào chương trình đào tạo. Khi thực hiện thành công thì hệ thống hiển thị màn hình danh sách các chương trình đào tạo. | Dữ liệu ngành và môn học trong hệ thống. |
| 2. Cập nhật | 2.1. Nhấn nút Sửa ở môn học muốn cập nhật dữ liệu. | 2.2. Hệ thống hiển thị màn hình có biểu mẫu chứa thông tin của môn học muốn cập nhật. | Dữ liệu trường, môn học tương ứng. |
| 2.3. Cập nhật thông tin chương trình đào tạo 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 ngành và môn học, chương trình đào tạo  trong hệ thống. |
| 3. Xóa | 3.1. Nhấn nút Xóa ở chương trình đào tạo 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 chương trình đào tạo. | Dữ liệu là id của chương trình đào tạo trong hệ thống. |

Bảng 2.4.

* 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 chương trình đào tạo đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách các chương trình đào tạo và thông báo lỗi cho người dùng.

Tại bước 2.4: Khi chương trình đào tạo 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 chương trình đào tạo đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách các chương trình đào tạo và thông báo lỗi cho người dùng.

Tại bước 3.4: Khi chương trình đào tạo người dùng muốn xóa không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách các chương trình đào tạo và thông báo lỗi cho người dùng.

**Yêu cầu đặc biệt**

Người dùng bắt buộc là tài khoản quản trị hệ thống hoặc là quản lý cấp trường.

Quản lý cấp trường chỉ thao tác được chương trình đào tạo thuộc trường mà phạm vi tài khoản hoạt động.

**Điều kiện tiên quyết**

Người dùng phải đăng nhập vào hệ thống.

**Các vấn đề mở**

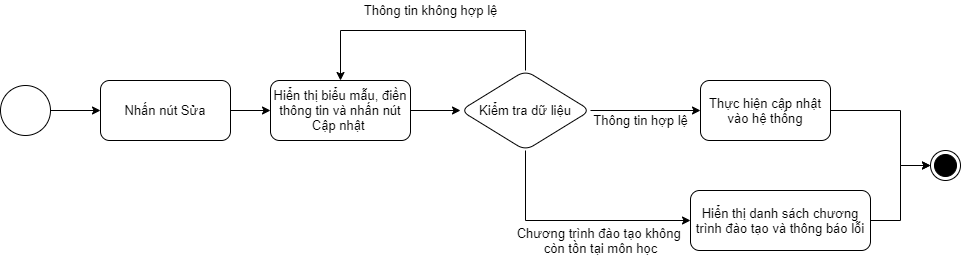
Không có

**Biểu đồ hoạt động**

Thêm mới

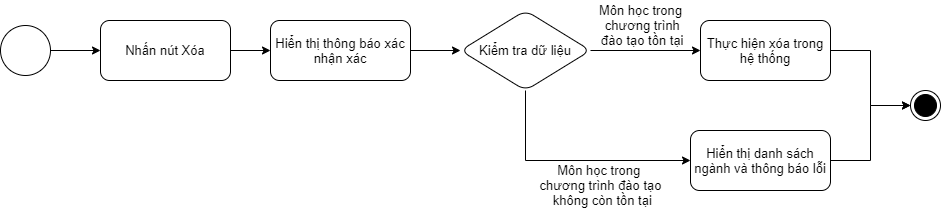
Hình 2. 11.

Cập nhật



Hình 2. 12.

Xóa



Hình 2. 13.

#### **Quản lý kết quả học tập**

**Mô tả tóm tắt**

Người dùng cập nhật thông tin về các kết quả học tập của sinh viên như: thêm mới kết quả học tập môn học của sinh viên, cập nhật thông tin kết quả môn học của sinh viên, xóa kết quả môn học của sinh viên khỏi hệ thống.

**Luồng sự kiện**

* 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 kết quả môn học của sinh viên. | Dữ liệu người dùng nhập.  Dữ liệu về môn học, sinh viên, kỳ học. |
| 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 kết quả môn học của sinh. Khi thực hiện thành công thì hệ thống hiển thị màn hình danh sách kết quả học tập. | Dữ liệu môn học, sinh viên, kỳ học. |
| 2. Cập nhật | 2.1. Nhấn nút Sửa ở kết quả môn học của sinh viên muốn cập nhật dữ liệu. | 2.2. Hệ thống hiển thị màn hình có biểu mẫu chứa thông tin điểm số môn học của sinh viên. | Dữ liệu môn học, sinh viên, kỳ học, kết quả học tập. |
| 2.3. Cập nhật thông tin kết quả môn học của sinh viên 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 môn học, sinh viên, kỳ học, kết quả học tập. |
| 3. Xóa | 3.1. Nhấn nút Xóa ở kết quả môn học của sinh viên 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 kết quả đào tạo. | Dữ liệu là id của kết quả môn học của sinh viên trong hệ thống. |

Bảng 2.5.

* 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 kết quả học tập của sinh viên đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách kết quả học tập và thông báo lỗi cho người dùng.

Tại bước 2.4: Khi kết quả học tập của sinh viên 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 kết quả học tập của sinh viên đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách kết quả học tập và thông báo lỗi cho người dùng.

Tại bước 3.4: Khi kết quả môn học của sinh viên mà người dùng muốn xóa không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách kết quả học tập và thông báo lỗi cho người dùng.

**Yêu cầu đặc biệt**

Người dùng bắt buộc là tài khoản quản trị hệ thống hoặc là quản lý cấp trường.

Quản lý cấp trường chỉ thao tác được với kết quả môn học của sinh viên thuộc trường mà phạm vi tài khoản hoạt động.

**Điều kiện tiên quyết**

Người dùng phải đăng nhập vào hệ thống.

**Các vấn đề mở**

Không có

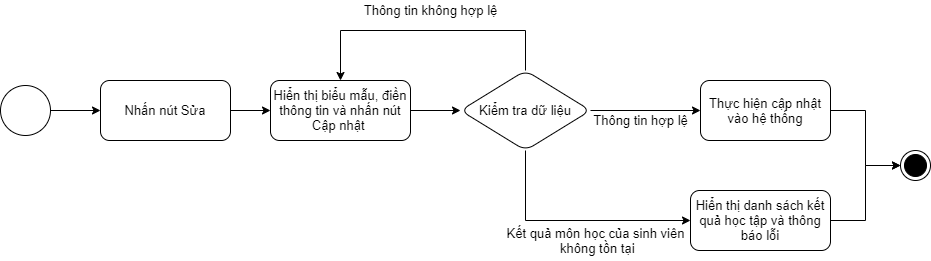
**Biểu đồ hoạt động**

Thêm mới



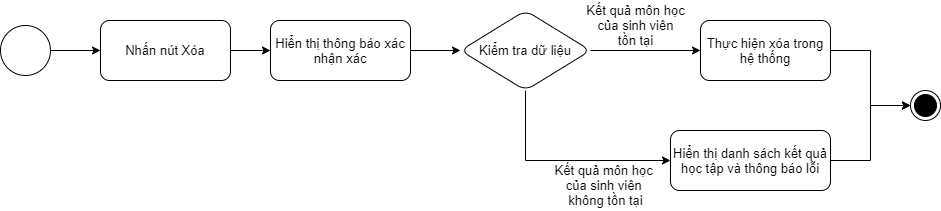
Hình 2. 14.

Cập nhật



Hình 2. 15.

Xóa



Hình 2. 16.

#### **Quản lý khóa học**

**Mô tả tóm tắt**

Người dùng cập nhật thông tin về các khóa học như: thêm mới khóa học, cập nhật thông tin khóa học, xóa khóa học khỏi hệ thống.

**Luồng sự kiện**

* 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 khóa học. | Dữ liệu người dùng nhập.  Dữ liệu về trường, năm học. |
| 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 khóa. Khi thực hiện thành công thì hệ thống hiển thị màn hình danh sách khóa. | Dữ liệu trường, năm học, khóa. |
| 2. Cập nhật | 2.1. Nhấn nút Sửa ở khóa muốn cập nhật dữ liệu. | 2.2. Hệ thống hiển thị màn hình có biểu mẫu chứa thông tin khóa. | Dữ liệu trường, năm học, khóa. |
| 2.3. Cập nhật thông tin khóa 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 trường, năm học, khóa. |
| 3. Xóa | 3.1. Nhấn nút Xóa ở khóa 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 khóa. | Dữ liệu là id của khóa trong hệ thống. |

Bảng 2.6.

* 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 khóa đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách khóa học và thông báo lỗi cho người dùng.

Tại bước 2.4: Khi khóa học 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 khóa học đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách khóa học và thông báo lỗi cho người dùng.

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

**Yêu cầu đặc biệt**

Người dùng bắt buộc là tài khoản quản trị hệ thống hoặc là quản lý cấp trường.

Quản lý cấp trường chỉ thao tác được với khóa học thuộc trường mà phạm vi tài khoản hoạt động.

**Điều kiện tiên quyết**

Người dùng phải đăng nhập vào hệ thống.

**Các vấn đề mở**

Không có

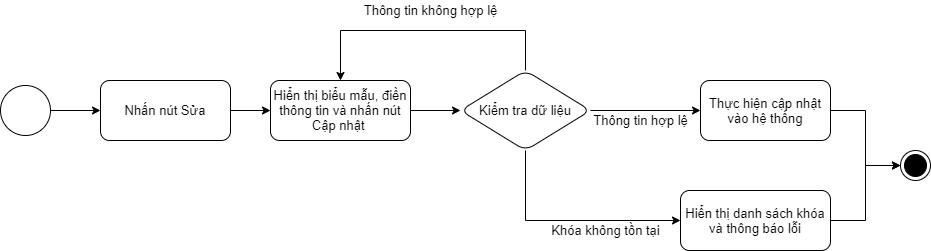
**Biểu đồ hoạt động**

Thêm mới



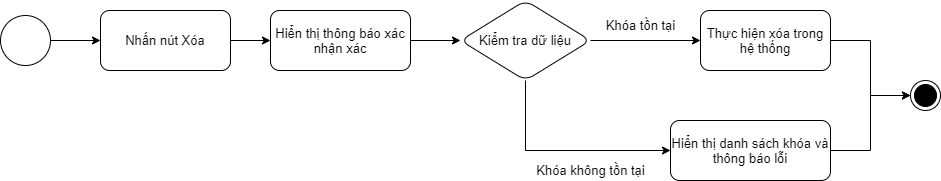
Hình 2. 17.

Cập nhật



Hình 2. 18.

Xóa



Hình 2. 19.

#### **Quản lý lớp**

**Mô tả tóm tắt**

Người dùng cập nhật thông tin về các lớp học như: thêm mới lớp học, cập nhật thông tin lớp học, xóa lớp học khỏi hệ thống.

**Luồng sự kiện**

* 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 lớp học. | Dữ liệu người dùng nhập.  Dữ liệu về khóa. |
| 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 lớp. Khi thực hiện thành công thì hệ thống hiển thị màn hình danh sách lớp học. | Dữ liệu khóa, lớp học. |
| 2. Cập nhật | 2.1. Nhấn nút Sửa ở lớp học muốn cập nhật dữ liệu. | 2.2. Hệ thống hiển thị màn hình có biểu mẫu chứa thông tin lớp. | Dữ liệu khóa, lớp học. |
| 2.3. Cập nhật thông tin lớp học 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 khóa, lớp học. |
| 3. Xóa | 3.1. Nhấn nút Xóa ở lớp học 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 lớp học. | Dữ liệu là id của lớp học trong hệ thống. |

Bảng 2.7.

* 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 lớp học đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách lớp học và thông báo lỗi cho người dùng.

Tại bước 2.4: Khi lớp học 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 lớp học đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách lớp học và thông báo lỗi cho người dùng.

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

**Yêu cầu đặc biệt**

Người dùng bắt buộc là tài khoản quản trị hệ thống hoặc là quản lý cấp trường.

Quản lý cấp trường chỉ thao tác được với lớp học thuộc trường mà phạm vi tài khoản hoạt động.

**Điều kiện tiên quyết**

Người dùng phải đăng nhập vào hệ thống.

**Các vấn đề mở**

Không có

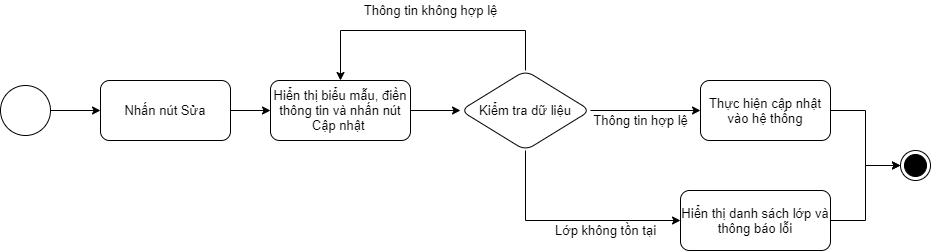
**Biểu đồ hoạt động**

Thêm mới



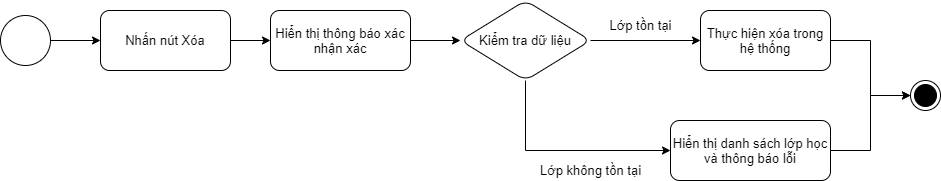
Hình 2. 20.

Cập nhật



Hình 2. 21.

Xóa



Hình 2. 22.

#### **Quản lý năm học**

**Mô tả tóm tắt**

Người dùng cập nhật thông tin về các năm học như: thêm mới năm học, cập nhật thông tin năm học, xóa năm học khỏi hệ thống.

**Luồng sự kiện**

* 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 năm học. Khi thực hiện thành công thì hệ thống hiển thị màn hình danh sách năm học. | Dữ liệu năm học. |
| 2. Cập nhật | 2.1. Nhấn nút Sửa ở năm học muốn cập nhật dữ liệu. | 2.2. Hệ thống hiển thị màn hình có biểu mẫu chứa thông tin năm học. | Dữ liệu năm học. |
| 2.3. Cập nhật thông tin năm học 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 năm học. |
| 3. Xóa | 3.1. Nhấn nút Xóa ở năm học 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 năm học. | Dữ liệu là id của năm học trong hệ thống. |

Bảng 2.8.

* 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 năm học đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách năm học và thông báo lỗi cho người dùng.

Tại bước 2.4: Khi năm học 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 năm học đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách năm học và thông báo lỗi cho người dùng.

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

**Yêu cầu đặc biệt**

Người dùng bắt buộc là tài khoản quản trị hệ thống hoặc là quản lý cấp trường.

**Điều kiện tiên quyết**

Người dùng phải đăng nhập vào hệ thống.

**Các vấn đề mở**

Không có

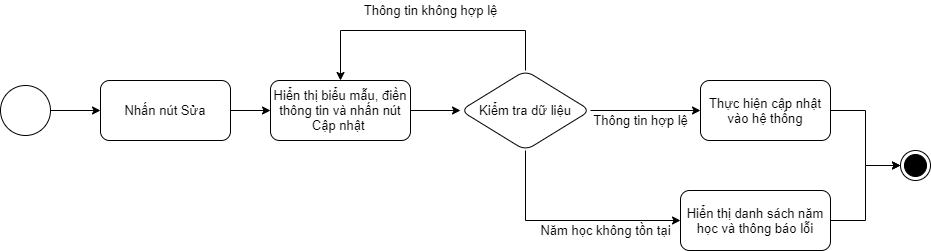
**Biểu đồ hoạt động**

Thêm mới



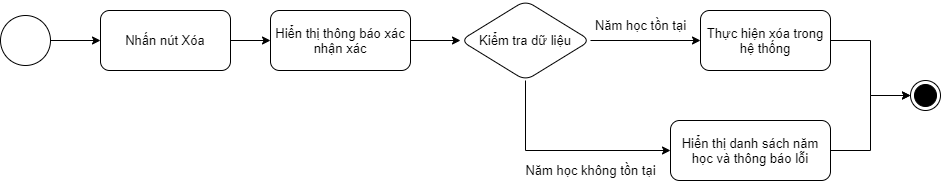
Hình 2. 23.

Cập nhật



Hình 2. 24.

Xóa



Hình 2. 25.

#### **Quản lý kỳ học**

**Mô tả tóm tắt**

Người dùng cập nhật thông tin về các kỳ học như: thêm mới kỳ học, cập nhật thông tin kỳ học, xóa kỳ học khỏi hệ thống.

**Luồng sự kiện**

* 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.  Dữ liệu năm học. |
| 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 kỳ học. Khi thực hiện thành công thì hệ thống hiển thị màn hình danh sách kỳ học. | Dữ liệu kỳ học, năm học. |
| 2. Cập nhật | 2.1. Nhấn nút Sửa ở kỳ học muốn cập nhật dữ liệu. | 2.2. Hệ thống hiển thị màn hình có biểu mẫu chứa thông tin kỳ học. | Dữ liệu kỳ học, năm học. |
| 2.3. Cập nhật thông tin kỳ học 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 kỳ học, năm học. |
| 3. Xóa | 3.1. Nhấn nút Xóa ở kỳ học 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 kỳ học. | Dữ liệu là id của kỳ học trong hệ thống. |

Bảng 2.9.

* 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 kỳ học đó không còn tồn tại thì hệ thống sẽ hiển thị màn hình danh sách kỳ học và thông báo lỗi cho người dùng.

Tại bước 2.4: Khi kỳ học 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 kỳ học và thông báo lỗi cho người dùng.

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

**Yêu cầu đặc biệt**

Người dùng bắt buộc là tài khoản quản trị hệ thống hoặc là quản lý cấp trường.

**Điều kiện tiên quyết**

Người dùng phải đăng nhập vào hệ thống.

**Các vấn đề mở**

Không có

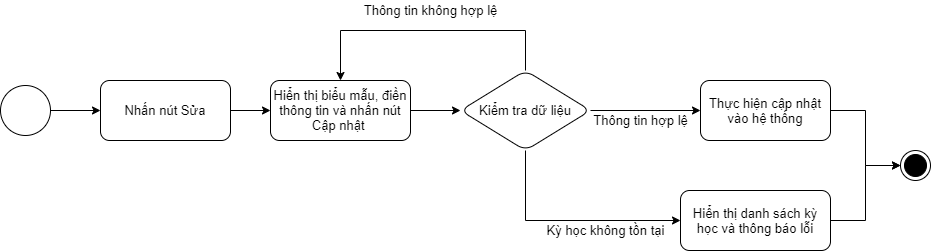
**Biểu đồ hoạt động**

Thêm mới



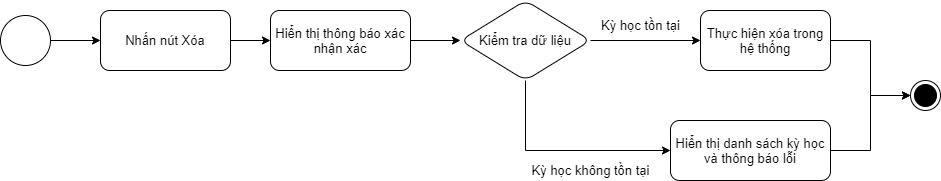
Hình 2. 26.

Cập nhật



Hình 2. 27.

Xóa



Hình 2. 28.

#### **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 sự kiện**

* 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 năng 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 chức năng. |
| 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 2.10.

* 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**

Thêm mới



Hình 2. 29.

Cập nhật



Hình 2. 30.

Xóa



Hình 2. 31.

#### **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 sự kiện**

* 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 2.11.

* 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. 32.

#### **Thống kê phổ điểm theo môn và khóa**

**Mô tả tóm tắt**

Người dùng có thể xem thống kê qua biểu đồ cột về điểm của môn học của khóa các khóa.

**Luồng sự kiện**

* 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. |

Bảng 2.12.

* 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. 33.

#### **Thống kê điểm trung bình môn học qua các năm**

**Mô tả tóm tắt**

Người dùng có thể xem thống kê qua biểu đồ cột về điểm trung bình của môn học qua các năm.

**Luồng sự kiện**

* 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, môn, năm học. |

Bảng 2.13.

* 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. 34.

#### **Thống kê điểm GPA theo khóa và ngành**

**Mô tả tóm tắt**

Người dùng có thể xem thống kê qua biểu đồ đường về điểm GPA các sinh viên theo toàn bộ khóa và theo từng kỳ.

**Luồng sự kiện**

* 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. |

Bảng 2.14.

* 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. 35.

#### **Thống kê điểm GPA theo sinh viên**

**Mô tả tóm tắt**

Người dùng có thể xem thống kê qua biểu đồ đường về điểm GPA các sinh viên.

**Luồng sự kiện**

* 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 từ sinh viên. |

Bảng 2.15.

* 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. 36.

#### **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 sự kiện**

* 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 2.16.

* 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**

Xem dự báo



Hình 2. 37.

Tính dự báo theo môn



Hình 2. 38.

Tính dự báo theo khóa



Hình 2. 39.

#### **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 sự kiện**

* 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 2.17.

* 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. 40.

## **Thiết kế**

### **Thiết kế kiến trúc hệ thống**

#### Tổng quan về cấu trúc



Hình 2.41. Cấu trúc thư mục

#### 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.

**Thư mục mainApp**: xử lý toàn bộ dữ liệu chính từ các app đến tiếp nhận, phản hồi của người dùng.

Lưu trữ controller, model, template của hệ thống

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.

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 thế sẽ là một file. Khi cần sử dụng thì sử dụng lẫn nhau và tránh sự cồng kềnh 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 tiếng. 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**: sử dụng các đối tượng của app chính để gọi các thư viện khác và tính toán về dự báo

**Thư mục statistic**: sử dụng các đối tượng của app chính để gọi các thư viện khác và tính toán về dự báo

**Thư mục train\_data:** Thực hiện lưu các file dữ liệu đã qua huấn luyện để làm kết quả huấn luyện.

### **Thiết kế cơ sở dữ liệu**

#### **Tổng quan về cơ sở dữ liệu**



Hình 2.42. Tổng quan về các bảng trong cơ sở dữ liệu

#### **Mô tả chi tiết**

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

Bảng 2.18. 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 2.19. 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 2.20. 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 2.21 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 2.22. 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 2.23. 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 2.24. 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 2.25. 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 2.26. 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 2.27. Cấu trúc bảng mainapp\_customser\_function

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| dumpModeID | ID của người dùng |
| dumpFile |  |
| updateTime |  |
| active |  |
| args |  |
| modelName |  |
| param |  |

Bảng 2.28. Cấu trúc bảng mainapp\_dumpmodel

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

Bảng 2.29. 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 2.30. 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 2.31. Cấu trúc bảng mainapp\_gpa

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| predictID | ID của dữ liệu dự đoán |
| predictTime | Thời gian thực hiện dữ đoán |
| course\_id | ID của môn học |
| student\_id | ID của người dùng sinh viên |

Bảng 2.32. 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 2.33. 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 2.34. 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 2.35. Cấu trúc bảng mainapp\_major\_course

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| PredictHistoryID | ID của mỗi lần dữ 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 2.36. 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 2.37. 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 2.38. 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 2.39. 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 2.40. 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 2.41. 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 2.42. 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 2.43. 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 2.44. 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 2.45. 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 2.46. Cấu trúc bảng mainapp\_years

### **Thiết kế biểu đồ tuần tự**

#### **Tính năng xem danh sách**



Hình 2.43. Biểu đồ tuần tự tính năng xem danh sách

Biểu đồ tuần tự cho chức năng áp dụng với các danh mục: Người dùng, Kết quả học tập, Trường, Ngành, Năm học, Kỳ học, Khóa, Môn học, Khung đào tạo, Lớp, Chức năng, Vai trò, Logs.

#### **Tính năng thêm mới, chỉnh sửa nội dung**

****

Hình 2.44. Biểu đồ tuần tự tính năng tạo mới và sửa nội dung

#### **Tính năng xóa nội dung**



Hình 2.45. Biểu đồ tuần tự tính năng xóa nội dung

Biểu đồ tuần tự cho chức năng áp dụng với các danh mục: Người dùng, Kết quả học tập, Trường, Ngành, Năm học, Kỳ học, Khóa, Môn học, Khung đào tạo, Lớp, Chức năng, Vai trò, Logs.

#### **Tính năng xuất file csv**



Hình 2.46. Biểu đồ tuần tự tính năng xuất file csv

Biểu đồ tuần tự cho chức năng áp dụng với các danh mục: Người dùng, Kết quả học tập, Trường, Ngành, Năm học, Kỳ học, Khóa, Môn học, Khung đào tạo, Lớp, Chức năng, Vai trò.

#### **Tính năng nhập file csv**

#### **Tính năng thống kê theo môn và khóa**



Hình 2.47. Biểu đồ tuần tự tính năng thống kê theo môn học và khóa

#### **Tính năng thống kê điểm trung bình môn học qua các năm**

****

Hình 2.48. Biểu đồ tuần từ tính năng thống kê điểm trung bình môn học qua các năm

#### **Thống kê GPA theo khóa và ngành**



Hình 2.49. Biểu đồ tuần tự tính năng thống kê GPA theo khóa và ngành

#### **Tính năng dự đoán điểm số**



Hình 2.50. Biểu đồ tuần tự tính năng dự đoán điểm số

#### **Thống kê GPA sinh viên**

****

Hình 2.51. Biểu đồ tuần tự tính năng thống kê GPA sinh viên

#### **Tính năng gợi ý môn học**



Hình 2.52. Biểu đồ tuần tự tính năng gợi ý môn học

#### **Tính năng đăng nhập**



Hình 2.53. Biểu đồ tuần tự tính năng đăng nhập

#### **Tính năng đăng xuất**



Hình 2.54. Biểu đồ tuần tự tính năng đăng xuất

# Cài đặt thử nghiệm và đánh giá



## Sử dụng API

### Thống kê

#### 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 cua 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

Kết quả tương ứng khi gọi api



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 đặt 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.

#### 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

Kết quả tương ứng khi gọi API



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.

#### 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 cua 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 các kỳ học của rồi đưa ra kết quả số GPA đã thay đổi) |

Bảng 3.3. Bảng chú thích API

Kết quả tương ứng khi gọi API



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.

#### 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 các kỳ học của sinh viên rồi đưa ra). |

Bảng 3.4. Bảng chú thích API

Kết quả tương ứng khi gọi API



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

semester\_number: 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: điểm trung bình của sinh viên tính theo các môn mà sinh viên tham gia vào thời điểm kỳ đó.

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

### Dự đoán

#### 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: 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,...

#### 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: ID của ngành.

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

### 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 mô học muốn được gợi ý môn học |

Bảng 3.5. Bảng chú thích API

## Cài đặt và thử nghiệm

### Cài đặt

#### 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 |

#### 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 diango 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 |

#### 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



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

### Khởi động chương trình

#### Makemigrations

Lệch 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 |

#### 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 |

### Kết quả thu được



Hình 3.6. Các giao diện kết quả của hệ thống



Hình 3.. Thống kê phổ điểm



Hình 3.. Thống kê điểm trung bình



Hình 3.. Thống kê GPA



Hình 3.. Thống kê GPA sinh viên

## 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 quyển của vai trò. Vì thế thông tin cá nhân của sinh viên sẽ được đảm bảo an toàn và bí mật.

**Nhược điểm**

Hệ thống có dự báo điểm cho sinh viên những hệ thống chưa có phát triển thêm tính năng lấy phản hồi và 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 chart js.

**Kiến thức đạt được**

Hiểu thêm về các hoạt động, kiến thức về framework django nói chung 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.

**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. doi:10.1007/978-1-4842-2787-9