

Trường đại học Bách khoa Hà Nội

Viện công nghệ thông tin và truyền thông

ĐỒ ÁN TỐT NGHIỆP

Xây dựng hệ thống gợi ý đăng ký học tập ứng dụng tại trường Đại học Bách khoa Hà Nội

NGUYỄN BÁ NGỌC

ngoc.nb173287@sis.hust.edu.vn

Ngành Công nghệ thông tin

Giảng viên hướng dẫn: PGS.TS.Nguyễn Bình Minh _____
Chữ ký của GVHD

Bộ môn: Hệ thống thông tin

Viện: Công nghệ thông tin và truyền thông

Hà Nội, 04/2021

Phiếu giao nhiệm vụ đồ án

1. Thông tin sinh viên

Họ và tên sinh viên: Nguyễn Bá Ngọc

Điện thoại liên lạc: 0868681623

Email: ngoc.nb173287@sis.hust.edu.vn

Lớp: CNTT.10 - K62

Hệ đào tạo: Cử nhân công nghệ thông tin

2. Mục đích nội dung của đồ án

Xây dựng hệ thống gợi ý đăng ký học tập áp dụng trong trường đại học Bách khoa Hà Nội.

3. Các nhiệm vụ cụ thể của đồ án tốt nghiệp

(1) Phân tích yêu cầu và thiết kế hệ thống.

(2) Tìm hiểu các mô hình gợi ý, lựa chọn, xây dựng và đánh giá mô hình, tích hợp mô hình vào hệ thống gợi ý.

(3) Xây dựng hệ thống website với hai phân hệ sinh viên và quản lý.

(4) Triển khai và kiểm thử toàn bộ hệ thống.

4. Lời cam đoan của sinh viên

Tôi – *Nguyễn Bá Ngọc* - cam kết Đồ án tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của PGS.TS. Nguyễn Bình Minh. Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày tháng năm 2021

Tác giả ĐATN

Sinh viên Nguyễn Bá Ngọc

Xác nhận của giảng viên hướng dẫn:

.....
.....
.....
.....
.....

Hà Nội, ngày tháng năm 2021

Giảng viên hướng dẫn

PGS.TS. Nguyễn Bình Minh

Lời cảm ơn

Đầu tiên, em xin gửi lời cảm ơn chân thành tới nhà trường, các thầy cô giáo trường Đại học Bách Khoa Hà Nội nói chung và thầy cô giáo trong Viện Công nghệ thông tin và Truyền thông nói riêng đã đem đến cho em một môi trường học tập bổ ích, truyền đạt những kiến thức, tạo điều kiện giúp đỡ em trong suốt quá trình học tập. Đặc biệt, em xin gửi lời cảm ơn sâu sắc nhất tới PGS.TS. Nguyễn Bình Minh, Bộ môn Hệ thống thông tin – Viện Công nghệ thông tin và Truyền thông – Trường Đại học Bách Khoa Hà Nội. Trong suốt quá trình thực hiện ĐATN, thầy đã tận tình hướng dẫn, chỉ bảo, đưa ra những góp ý giúp em có được định hướng tốt hơn, đúng đắn hơn cho đề tài và hoàn thành đồ án một cách trọn vẹn nhất.

Tiếp theo, em xin dành lời cảm ơn tới các anh chị, bạn bè, các thành viên trong BKC Lab đã quan tâm, chia sẻ, ủng hộ về tinh thần lớn lao để em có động lực học tập và hoàn thành ĐATN này.

Cuối cùng, em xin chân thành cảm ơn gia đình đã luôn tạo điều kiện, giúp đỡ động viên trong suốt quá trình thực hiện ĐATN.

Trong quá trình thực hiện ĐATN, em đã cố gắng tìm hiểu, tìm ra hướng giải quyết tốt cho bài toán của mình, giảm thiểu, hạn chế các sai sót. Mặc dù vậy, đồ án không thể tránh khỏi những thiếu sót, vì vậy em rất mong nhận được các góp ý từ các thầy cô và bạn đọc để có thể hoàn thiện hơn nữa sản phẩm này.

Em xin chân thành cảm ơn!

Tóm tắt nội dung đề án

Nội dung chính của đề án tập trung quanh các vấn đề cùng các giải pháp xây dựng một hệ thống gợi ý đăng ký học tập trong trường Đại học Bách khoa Hà Nội. Mục đích chính là nhằm hỗ trợ sinh viên trong học tập bằng cách đưa ra các gợi ý môn học để sinh viên tham khảo, cùng các phân tích tình hình học tập để từ đó giúp sinh viên có được định hướng, quyết định học tập đúng đắn.

Đề án trình bày chi tiết về các vấn đề, mục tiêu, giải pháp, phân tích yêu cầu, thiết kế và xây dựng hệ thống. Đề án cũng tập trung tìm hiểu các phương pháp được sử dụng trong các hệ gợi ý, so sánh đánh giá để lựa chọn phương pháp phù hợp, từ đó xây dựng mô hình, giải thuật và triển khai chức năng gợi ý của hệ thống.

Nội dung đề án bao gồm 6 chương:

Chương 1. Tổng quan đề tài

Chương 2. Công nghệ sử dụng và cơ sở lý thuyết

Chương 3. Xây dựng hệ thống website gợi ý đăng ký học tập

Chương 4. Xây dựng chức năng gợi ý đăng ký học tập

Chương 5. Triển khai và kiểm thử hệ thống

Chương 6. Kết luận và hướng phát triển

Mục lục

Lời cảm ơn	1
Tóm tắt nội dung đồ án	2
Danh mục hình ảnh	6
Danh mục bảng	8
Danh mục từ viết tắt	9
Danh mục thuật ngữ	10
1 Tổng quan đề tài	11
1.1 Đặt vấn đề	11
1.2 Mục tiêu	11
1.3 Phạm vi đề tài	12
1.4 Định hướng giải pháp	12
1.5 Bố cục của đồ án	13
2 Công nghệ sử dụng và cơ sở lý thuyết	14
2.1 Các công nghệ sử dụng	14
2.1.1 Frontend	14
2.1.2 Backend	15
2.1.3 Giao thức kết nối	17
2.1.4 Triển khai hệ thống với docker	18
2.2 Cơ sở lý thuyết	18
2.2.1 Utility matrix	18
2.2.2 Lọc cộng tác dựa trên láng giềng (NBCF)	19
2.2.3 Lọc cộng tác dựa trên phân tích ma trận (MFCF)	22
2.2.4 Phương pháp đánh giá mô hình dự đoán	26
3 Xây dựng hệ thống website gợi ý đăng ký học tập	27
3.1 Phân tích hệ thống	27
3.1.1 Tổng quan chức năng	27
3.1.2 Đặc tả chức năng	29
3.1.3 Yêu cầu phi chức năng	39

3.2	Thiết kế hệ thống	41
3.2.1	Thiết kế kiến trúc	41
3.2.2	Thiết kế chi tiết frontend	43
3.2.3	Thiết kế chi tiết backend	48
3.2.4	Thiết kế mô hình dữ liệu	56
4	Xây dựng chức năng gợi ý đăng ký học tập	63
4.1	Xây dựng tập dữ liệu	63
4.2	Xây dựng, đánh giá mô hình huấn luyện	69
4.2.1	Xây dựng mô hình	69
4.2.2	Thử nghiệm, đánh giá mô hình	71
4.3	Cài đặt và tích hợp mô hình	73
5	Triển khai và kiểm thử hệ thống	75
5.1	Cài đặt và triển khai hệ thống	75
5.2	Kiểm thử hệ thống	80
5.2.1	Kiểm thử tương thích	80
5.2.2	Kiểm thử hộp đen	80
6	Kết luận và hướng phát triển	83
6.1	Kết luận	83
6.2	Hướng phát triển	83
	Tài liệu tham khảo	84

Danh mục hình ảnh

2.1	Utility matrix biểu diễn đánh giá của người dùng cho các bài hát (0 - 5)	19
2.2	Xây dựng user similarity matrix [6]	21
2.3	Utility matrix Y xấp xỉ bằng tích của hai ma trận U và V	23
3.1	Biểu đồ use case tổng quan hệ thống gợi ý học tập	27
3.2	Biểu đồ phân rã use case Huấn luyện	28
3.3	Biểu đồ phân rã use case Lịch sử huấn luyện	28
3.4	Biểu đồ phân rã use case Cài đặt hệ thống	29
3.5	Thiết kế tổng thể của hệ thống gợi ý học tập	42
3.6	Chuyển đổi dữ liệu trong Object Relational Database	42
3.7	Kiến trúc tổng thể frontend	43
3.8	Mockup layout của hệ thống gợi ý đăng ký học tập	44
3.9	Thiết kế màn hình điểm môn học	45
3.10	Thiết kế màn hình biểu đồ học tập	45
3.11	Thiết kế màn hình gợi ý học tập	46
3.12	Thiết kế màn hình huấn luyện mô hình	46
3.13	Thiết kế màn hình lịch sử huấn luyện	47
3.14	Thiết kế màn hình chi tiết huấn luyện	47
3.15	Thành phần kiến trúc chính và luồng hoạt động của backend	48
3.16	Biểu đồ trình tự ca sử dụng Huấn luyện mô hình	52
3.17	Biểu đồ trình tự ca sử dụng Lưu kết quả huấn luyện	54
3.18	Biểu đồ trình tự ca sử dụng Xem gợi ý học tập	55
3.19	Sơ đồ thực thể liên kết cho hệ thống gợi ý đăng ký học tập	56
3.20	Sơ đồ cơ sở dữ liệu	58
4.1	Ví dụ về dữ liệu sinh viên	64
4.2	Dữ liệu loại học phần dưới định dạng json	65
4.3	Các giá trị cài đặt cho hàm sinh dữ liệu	66
4.4	Biểu đồ cột phân bố các giá trị điểm cho học phần <i>Tư tưởng Hồ Chí Minh</i>	67
4.5	Biểu đồ phân bố giá trị của một số học phần khác	68
4.6	Ví dụ phân rã ma trận trên dữ liệu sinh viên - môn học với $k=2$	70
4.7	Biểu diễn ma trận Y dưới dạng <i>sparse matrix</i>	71
4.8	Mô hình sử dụng bias với $k = 50, \lambda = 0.02, \beta = 0.0001, max_iter = 100$	73
4.9	Biểu đồ so sánh giữa các giải thuật	73

4.10	Thiết kế chi tiết package <i>predict_model</i>	74
5.1	Sơ đồ triển khai hệ thống gợi ý đăng ký học tập	75
5.2	Giao diện gợi ý học tập	77
5.3	Giao diện điểm môn học	78
5.4	Giao diện biểu đồ học tập	78
5.5	Giao diện huấn luyện mô hình với tham số mặc định	79
5.6	Giao diện huấn luyện mô hình với tham số điều chỉnh	79

Danh mục bảng

3.1	Danh sách use case	29
3.2	Đặc tả use case Đăng nhập	30
3.4	Đặc tả use case Xem kết quả học tập	31
3.6	Đặc tả use case Xem biểu đồ học tập	32
3.7	Đặc tả use case Xem gợi ý học tập	32
3.8	Đặc tả use case Xem biểu đồ học tập	33
3.9	Đặc tả use case Huấn luyện mô hình	33
3.11	Đặc tả use case Điều chỉnh tham số	34
3.13	Đặc tả use case Lưu kết quả huấn luyện	35
3.14	Đặc tả use case Xem danh sách phiên bản huấn luyện	36
3.15	Đặc tả use case Xem chi tiết huấn luyện	36
3.16	Đặc tả use case Xóa phiên bản huấn luyện	37
3.17	Đặc tả use case Sử dụng kết quả huấn luyện làm gợi ý	38
3.18	Đặc tả use case Xem cài đặt hệ thống	38
3.19	Đặc tả use case Thay đổi cài đặt	39
3.21	Danh sách API của hệ thống gợi ý đăng ký học tập	50
3.22	Định dạng dữ liệu gửi từ frontend	50
3.23	Mô tả các action	50
3.24	Thiết kế chi tiết bảng Program	58
3.25	Thiết kế chi tiết bảng Subject	59
3.26	Thiết kế chi tiết bảng Program_Subject	59
3.27	Thiết kế chi tiết bảng Course	59
3.28	Thiết kế chi tiết bảng Student	59
3.29	Thiết kế chi tiết bảng Learn	60
3.30	Thiết kế chi tiết bảng Learn	60
3.31	Thiết kế chi tiết bảng User	61
3.32	Thiết kế chi tiết bảng TrainVersion	61
3.33	Thiết kế chi tiết bảng Setting	62
3.34	Thiết kế chi tiết bảng Predict	62
4.1	Định dạng dữ liệu cho đối tượng sinh viên	63
4.2	Bảng kết quả sinh dữ liệu sinh viên	64
4.3	Mô tả các trường của đối tượng <i>loại học phần</i>	64

4.4	Mô tả các trường của đối tượng <i>học phần</i>	65
4.5	Bảng tổng hợp số lượng dữ liệu đã tạo cho từng đối tượng	69
4.6	Bảng quy đổi điểm số từ thang 10 về thang 4 của Đại học Bách khoa Hà Nội	71
4.7	Độ đo RMSE trên tập train và tập test với các tham số khác nhau trên mô hình kết hợp bias	72
4.8	Độ đo RMSE trên tập train và test với các tham số khác nhau trên mô hình không kết hợp bias	72
5.1	Thông tin cấu hình thiết bị triển khai hệ thống	76
5.2	Thống kê kiểm thử tương thích trên các thiết bị	80
5.3	Danh sách kịch bản kiểm thử	80
5.3	Danh sách kịch bản kiểm thử	81
5.3	Danh sách kịch bản kiểm thử	82

Danh mục từ viết tắt

DATN	Đồ án tốt nghiệp
API	Application Programing Interface (giao diện lập trình ứng dụng)
NBCF	Neighborhood-Based Collaborative Filtering
CF	Collaborative Filtering
MF	Matrix Factorization
DOM	Document Object Model
RMSE	Root Mean Squared Error
SGD	Stochastic gradient descent
ASGI	Asynchronous Server Gateway Interface
RBAC	Role-based access control
HTML	HyperText Markup Language
MVT	Model-View-Template
MVC	Model-Vew-Controller

Danh mục thuật ngữ

similarity	Sự tương đồng
database	Cơ sở dữ liệu
client	Máy khách, người dùng
server	Máy chủ
Recommendation Systems	Hệ gợi ý
User-based Collaborative Filtering	Lọc cộng tác dựa trên Users
Item-based Collaborative Filtering	Lọc cộng tác dựa trên Items

Chương 1: Tổng quan đề tài

1.1. Đặt vấn đề

Hiện nay phần lớn các trường đại học đã triển khai theo hình thức học tập tín chỉ, với hình thức này, người học trở thành trung tâm, được lựa chọn những môn học mình yêu thích, phát huy tính chủ động, sáng tạo của bản thân; tạo sự linh hoạt trong các môn học; giảm thiểu chi phí giảng dạy; thời gian học tập linh hoạt hơn.

Bên cạnh những ưu điểm thì phương pháp đào tạo tín chỉ cũng tạo ra những khó khăn trong quá trình dạy và học, có thể kể đến như lượng kiến thức không đầy đủ, không liên mạch bởi các môn học thường bị chia nhỏ, giáo viên không thể truyền tải hết kiến thức đến cho sinh viên; sinh viên khó gắn kết với nhau hơn bởi mỗi người sẽ lựa chọn các môn học, thời gian, lớp học khác nhau. Việc được tự do lựa chọn môn học cũng tiềm ẩn nhiều rủi ro khi sinh viên lựa chọn sai môn học, định hướng không rõ ràng, dẫn đến kết quả học tập thấp, bản thân chán nản và từ đó mất phương hướng, mất động lực học tập, thậm chí có thể bỏ học hoặc buộc thôi học... gây nên những tổn thất lớn cho sinh viên, gia đình và xã hội. Do vậy việc sinh viên có thể lựa chọn môn học và lộ trình học phù hợp để có kết quả học tập tốt là một nhu cầu cấp thiết.

Xét riêng trong trường đại học Bách khoa Hà Nội, cũng là ngôi trường đào tạo theo hình thức tín chỉ, hằng năm có rất nhiều sinh viên bị buộc thôi học, tình trạng học lại, nợ môn diễn ra khá phổ biến. Có nhiều nguyên nhân dẫn đến tình trạng này, có thể kể đến như môi trường sống, môi trường học tập có sự khác biệt lớn giữa cấp ba và đại học, nhiều bạn sinh viên không kịp thích nghi, hay nguyên nhân phổ biến hơn như đã nói là sinh viên không có định hướng học tập rõ ràng, lựa chọn sai môn học, mãi chơi dẫn đến bỏ bê học tập, kết quả yếu kém. Nhà trường cũng đã xây dựng nhiều hệ thống điện tử nhằm hỗ trợ sinh viên nhưng chưa có một hệ thống thực sự hỗ trợ sinh viên về mặt định hướng, lựa chọn đăng ký môn học. Thay vào đó, nhà trường đã thành lập các đội ngũ thầy cô tư vấn, định hướng học tập để giải quyết vấn đề này, tuy nhiên số lượng sinh viên là rất lớn, thời gian có hạn nên việc tư vấn còn hạn chế, hơn nữa nhiều bạn sinh viên có tâm lý ngại không muốn hoặc không dám gặp các thầy cô tư vấn. Từ thực tiễn đó, em đã hình thành ý tưởng ***xây dựng hệ thống gợi ý đăng ký học tập*** để hỗ trợ các bạn sinh viên có được định hướng học tập đúng đắn hơn, đồng thời như là một công cụ hỗ trợ đội ngũ tư vấn, giảm bớt khối lượng công việc, chi phí và thời gian.

1.2. Mục tiêu

Với yêu cầu đặt ra ở phần 1.1, có ba mục tiêu chính đặt ra đó là:

1. Xây dựng một hệ thống gợi ý đăng ký học tập trên nền tảng website với hai phân hệ sinh viên và quản lý

2. Xây dựng chức năng gợi ý cho hệ thống đó, trong đó chia ra các mục tiêu:
 - (a) Tìm hiểu các phương pháp phổ biến sử dụng trong một hệ gợi ý
 - (b) Xây dựng, thử nghiệm, đánh giá mô hình gợi ý
 - (c) Tích hợp mô hình gợi ý vào hệ thống website
3. Triển khai và kiểm thử toàn bộ hệ thống

1.3. Phạm vi đề tài

Đề tài tập trung vào xây dựng và triển khai hệ thống gợi ý đăng ký học tập trong phạm vi trường đại học Bách khoa Hà Nội, với các chức năng cơ bản và cần thiết cho các đối tượng sinh viên và quản lý hệ thống. Do các dữ liệu về sinh viên, dữ liệu học tập đều là các dữ liệu nhạy cảm nên đồ án sẽ thực hiện sinh dữ liệu để có thể chạy thử nghiệm được hệ thống, tuy nhiên để đảm bảo hệ thống có thể chạy tốt trong trường đại học Bách khoa Hà Nội các dữ liệu về sinh viên, chương trình học, môn học, hay kết quả học tập sẽ được xây dựng dựa trên cấu trúc dữ liệu của trường.

1.4. Định hướng giải pháp

Để giải quyết được các yêu cầu đặt ra trong phần 1.2, ĐATN cần phải (i) phân định vai trò của từng người dùng với các chức năng của hệ thống gợi ý đăng ký học tập, (ii) lựa chọn công nghệ lập trình phù hợp, (iii) xây dựng mô hình huấn luyện cho chức năng gợi ý, (iv) Tích hợp mô hình đó để xây dựng và triển khai hệ thống.

Để phân định rõ vai trò của từng người dùng trong hệ thống, ứng dụng sử dụng mô hình phân quyền RBAC. RBAC là mô hình phân quyền truy nhập dựa trên vai trò, mỗi người dùng trong ứng dụng sẽ được phân vai trò tương ứng với vai trò trong doanh nghiệp hay tổ chức và có quyền truy cập vào những tài nguyên được chỉ định.

Đồ án sử dụng (i) MySQL làm cơ sở dữ liệu, (ii) ReactJS, Redux để xây dựng frontend, (iii) Framework Django để phát triển backend. Các công nghệ này đều dễ dàng được triển khai trên hệ thống máy chủ truyền thống và trên các nền tảng dịch vụ cloud hiện nay. MySQL là hệ quản trị cơ sở dữ liệu quan hệ thường được sử dụng phổ biến trong việc phát triển các ứng dụng, đặc biệt là các ứng dụng web, nó đem tới sự ổn định, độ bảo mật thông tin cao, tốc độ truy vấn tốt và có thể tương thích với nhiều hệ điều hành. ReactJS là một thư viện với nhiều tính năng mạnh mẽ giúp xây dựng giao diện ứng dụng Web linh hoạt và dễ dàng, nó phân chia giao diện ứng dụng thành các thành phần thuận lợi cho việc tái sử dụng và mở rộng ứng dụng. ReactJS kết hợp với Redux sẽ giúp xây dựng và quản lý các thành phần giao diện tốt hơn. Về phía backend, Django là một web framework với đầy đủ các thư viện, module hỗ trợ giúp việc xây dựng các ứng dụng trở nên đơn giản, nhanh chóng, dễ dàng cài đặt và triển khai. Backend sẽ cung cấp các API để có thể giao tiếp, trao đổi truyền nhận dữ liệu với frontend.

Đối với mô hình gợi ý, đồ án sẽ thực hiện tìm hiểu đánh giá một số giải thuật trong mô hình lọc cộng tác để lựa chọn và xây dựng được mô hình phù hợp với hệ thống gợi ý đăng ký học tập. Mô hình sau đó sẽ được tích hợp vào chức năng gợi ý của hệ thống.

1.5. Bố cục của đồ án

Phần còn lại của đồ án tốt nghiệp được tổ chức như sau:

Chương 2 trình bày một số cơ sở lý thuyết, các phương pháp phổ biến sử dụng trong hệ gợi ý và giới thiệu về các công nghệ được sử dụng trong hệ thống cùng lý do sử dụng các công nghệ này.

Chương 3 trình bày về quá trình phân tích, thiết kế, xây dựng hệ thống.

Chương 4 trình bày chi tiết về việc xây dựng chức năng gợi ý cho hệ thống. Trong phần này, ta sẽ đi lựa chọn, xây dựng mô hình gợi ý phù hợp và tiến hành thử nghiệm đánh giá mô hình vừa xây dựng. Cuối cùng là việc tích hợp mô hình đó để xây dựng tính năng gợi ý cho hệ thống.

Chương 5 sẽ trình bày chi tiết về cách thức triển khai, cài đặt và kiểm thử hệ thống.

Chương 6 trình bày về các kết quả đạt được, đưa ra kết luận và định hướng phát triển cho ứng dụng trong tương lai.

Chương 2: Công nghệ sử dụng và cơ sở lý thuyết

2.1. Các công nghệ sử dụng

Phần này sẽ giới thiệu về các công nghệ được sử dụng trong hệ thống gợi ý đăng ký học tập. Hệ thống được xây dựng thành hai thành phần riêng biệt frontend và backend. Phần frontend sử dụng ReactJS kết hợp với Redux và thư viện Antdesign để xây dựng giao diện người dùng. Phần backend sử dụng framework Django để xây dựng nên các API và sử dụng MySQL làm cơ sở dữ liệu.

2.1.1 Frontend

Hiện nay có rất nhiều công nghệ hỗ trợ việc xây dựng giao diện người dùng, có thể kể đến như: ReactJS, AngularJS, VueJS... Đây đều là những thư viện, framework rất nổi tiếng và được sử dụng nhiều để xây dựng các trang web hiện nay. Đồ án sử dụng ReactJS cùng thư viện AntDesign để xây dựng giao diện, kết hợp với Redux để quản lý dữ liệu của ứng dụng. Lý do chọn ReactJS là bởi nó có cấu trúc rõ ràng, cú pháp quen thuộc, gần gũi với cú pháp HTML, CSS thông thường, hệ thống thư viện hỗ trợ và cộng đồng phát triển mạnh mẽ.

2.1.1.1 ReactJS

ReactJS ¹ (hay React) là một thư viện JavaScript mã nguồn mở được sử dụng để xây dựng giao diện người dùng và được phát triển bởi Facebook. ReactJS tạo ra những ứng dụng web hấp dẫn với tốc độ nhanh và hiệu quả cao với những mã hóa tối thiểu. React thường được sử dụng làm cơ sở để phát triển các ứng dụng *single-page* hoặc ứng dụng di động. Dưới đây là một vài tính năng nổi bật của React:

JSX (Javascript XML) - là một phần mở rộng cú pháp JavaScript. Đây là một cú pháp cho phép viết các mã HTML trong JavaScript để xác định thành phần giao diện người dùng và chức năng của chúng.

Component-Based – React được xây dựng dựa trên Component. Mỗi component được thiết kế với một logic cụ thể. Chính vì thế, việc phát triển và tìm lỗi đơn giản hơn vì chỉ cần tập trung vào một phần nhỏ thay vì toàn bộ ứng dụng. Hơn nữa, khi sử dụng component sẽ giúp tăng khả năng tái sử dụng khi viết code.

Virtual DOM - là tính năng chính cho phép ReactJS xây dựng các ứng dụng nhanh và có thể mở rộng. React tạo một bộ đệm cấu trúc dữ liệu trong bộ nhớ, tính toán sự khác biệt kết quả và sau đó cập nhật lại DOM trên trình duyệt một cách hiệu quả.

¹<https://reactjs.org/>, truy cập lần cuối 24/5/2021.

2.1.1.2 Redux

Như đã trình bày ở trên, có rất nhiều những thư viện được xây dựng để hỗ trợ phát triển các ứng dụng React, đem đến nhiều tiện ích cho trang web, tiện lợi cho việc phát triển, mở rộng ứng dụng. Trong đó, nổi bật nhất phải kể đến là bộ đôi React-Redux.

Redux JS là một thư viện Javascript hỗ trợ quản lý trạng thái (state) hay chính là dữ liệu của ứng dụng. Redux giúp ta viết các ứng dụng hoạt động nhất quán, chạy trong các môi trường khác nhau (client, server, và native) và dễ dàng để test. Redux ra đời lấy cảm hứng từ tư tưởng của ngôn ngữ Elm và kiến trúc Flux của Facebook.

2.1.1.3 Ant Design

Thư viện Ant Design được sử dụng để giúp cho việc xây dựng các thành phần giao diện được nhanh chóng và đồng nhất. Ant Design là tập hợp các components của React được xây dựng theo chuẩn thiết kế của Ant UED Team. Ant Design rất phù hợp để xây dựng các ứng dụng React một cách nhanh chóng bởi các components Ant-design cung cấp cho chúng ta là vô cùng đa dạng và đầy đủ, đáp ứng được hầu hết các components thông dụng như Layout, Button, Textfield, Icon, DatePicker, v.v.

2.1.2 Backend

Với hệ thống gợi ý đăng ký học tập, quá trình xử lý logic không cần phải có hiệu năng hay tốc độ xử lý quá cao bởi hệ thống không có nhiều tác vụ cần đến khả năng tính toán mạnh mẽ, chỉ trừ quá trình huấn luyện mô hình. Nhưng quá trình này diễn ra không thường xuyên, thường chỉ sử dụng sau khi có các dữ liệu về điểm được thêm hoặc cập nhật. Do đó việc sử dụng một ngôn ngữ lập trình có cú pháp nhanh gọn, dễ dàng cài đặt, triển khai và kiểm thử là rất phù hợp, vì vậy em quyết định lựa chọn ngôn ngữ lập trình python, web framework Django để xây dựng phần backend.

Dữ liệu của hệ thống có cấu trúc rất rõ ràng, các trường dữ liệu cố định, các thành phần có quan hệ chặt chẽ với nhau và nhất quán về mặt dữ liệu. Dựa vào các đặc điểm này thì một hệ quản trị cơ sở dữ liệu quan hệ là phù hợp nhất. Em sử dụng MySQL làm cơ sở dữ liệu bởi nó là mã nguồn mở nên không mất phí, nó phù hợp triển khai với các hệ thống web, mặt khác kích thước dữ liệu của hệ thống và tần suất truy cập dữ liệu là không quá lớn, chỉ ở mức trung bình.

2.1.2.1 Ngôn ngữ lập trình Python

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và

cách gõ linh động làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.

Python còn có hệ thống các thư viện mạnh mẽ và tiện ích hỗ trợ xây dựng và tích hợp các ứng dụng.

2.1.2.2 Cython

Cython² là một ngôn ngữ lập trình được thiết kế để đem lại hiệu năng như C hoặc C++ với mã được viết gần giống Python cộng thêm một số cú pháp được lấy cảm hứng từ C, được xây dựng với mục đích trở thành một *superset* của ngôn ngữ lập trình Python. Cython là một ngôn ngữ biên dịch thường được sử dụng để sinh ra các CPython modules mà trong đó mã sẽ được biên dịch thành C hoặc C++ và được bọc trong một *interface code*³. Các modules này có thể được sử dụng trong mã nguồn Python như bình thường.

Trong đồ án, Cython được sử dụng để triển khai phần giải thuật huấn luyện để tăng tốc độ thực thi.

2.1.2.3 Django

Django là một web framework khá nổi tiếng được viết hoàn toàn bằng ngôn ngữ Python. Nó là một framework với đầy đủ các thư viện, module hỗ trợ các web-developer. Django được phát triển bởi Django Software Foundation (DSF một tổ chức phi lợi nhuận độc lập). Mục tiêu chính của Django là đơn giản hóa việc tạo các website phức tạp có sử dụng cơ sở dữ liệu. Django tập trung vào tính năng “có thể tái sử dụng” và “có thể tự chạy” của các component, tính năng phát triển nhanh, không làm lại những gì đã làm. Một số website phổ biến được xây dựng từ Django là Pinterest, Instagram, Mozilla, và Bitbucket.

2.1.2.4 MySQL

MySQL là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến trên thế giới và được sử dụng rộng rãi trong việc phát triển ứng dụng, đặc biệt là các ứng dụng web. Các đặc điểm nổi bật của MySQL là tốc độ cao, ổn định dễ sử dụng, hoạt động trên nhiều hệ điều hành và có hệ thống các hàm tiện ích mạnh mẽ. Dưới đây là một số ưu điểm của MySQL:

Sử dụng dễ dàng: MySQL là cơ sở dữ liệu tốc độ cao và ổn định, công cụ này dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp hệ thống lớn các hàm tiện ích.

Tính bảo mật cao: MySQL phù hợp với các ứng dụng có truy cập cơ sở dữ liệu trên internet vì nó sở hữu nhiều tính năng bảo mật, thậm chí là bảo mật cấp cao.

Đa tính năng: MySQL có thể hỗ trợ hàng loạt các chức năng SQL từ hệ quản trị cơ sở dữ liệu quan hệ trực tiếp và cả gián tiếp.

²<https://en.wikipedia.org/wiki/Cython>, truy cập lần cuối 9/6/2021

³https://en.wikipedia.org/wiki/Wrapper_library

Khả năng mở rộng và sự mạnh mẽ: Công cụ MySQL có khả năng xử lý khối dữ liệu lớn và có thể mở rộng khi cần thiết.

Tương thích trên nhiều hệ điều hành: MySQL tương thích để chạy trên nhiều hệ điều hành, như Novell NetWare, Windows, Linux ... MySQL cũng cung cấp phương tiện mà các máy khách có thể chạy trên cùng một máy tính với máy chủ hoặc trên một máy tính khác (giao tiếp qua mạng cục bộ hoặc Internet).

Cho phép khôi phục: MySQL cho phép các transaction được khôi phục, cam kết và phục hồi sau sự cố.

2.1.3 Giao thức kết nối

2.1.3.1 Rest API

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu cần cho ứng dụng ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, v.v đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các tài nguyên. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

2.1.3.2 WebSocket

Trong quá trình huấn luyện mô hình cần có các chức năng, giao diện giúp theo dõi chất lượng mô hình, cập nhật các trạng thái của mô hình trong thời gian thực để có thể lựa chọn được mô hình tốt nhất làm kết quả gợi ý. Do vậy cần phải có một giải pháp để server có thể chủ động gửi các tin nhắn tới client đồng thời duy trì được một kết nối đủ lâu giữa client, server và WebSocket là công nghệ phù hợp giúp giải quyết vấn đề trên.

WebSocket là công nghệ hỗ trợ giao tiếp hai chiều giữa client và server bằng cách sử dụng một TCP socket để tạo một kết nối hiệu quả và ít tốn kém. Với kết nối này client có thể gửi tin nhắn đến server bất kỳ lúc nào và nhận phản hồi theo hướng sự kiện mà không cần gửi yêu cầu đến server. Nói cách khác, server có thể chủ động gửi thông tin đến client mà không cần có yêu cầu từ client và client có thể gửi tin nhắn đến server bất kỳ lúc nào, khác với HTTP là client chủ động gửi yêu cầu đến server sau đó chờ đợi để nhận được dữ liệu phản hồi.

2.1.4 Triển khai hệ thống với docker

Docker ⁴ là nền tảng phần mềm cho phép ta dựng, kiểm thử và triển khai ứng dụng một cách nhanh chóng. Docker đóng gói phần mềm vào các đơn vị tiêu chuẩn hóa được gọi là container có mọi thứ mà phần mềm cần để chạy, trong đó có thư viện, công cụ hệ thống, mã và thời gian chạy. Các container chạy trực tiếp trên nhân của máy chủ, ta có thể có nhiều container cùng chạy trên một tổ hợp phần cứng nhất định. Bằng cách sử dụng Docker, ta có thể nhanh chóng triển khai và thay đổi quy mô ứng dụng vào bất kỳ môi trường nào và biết chắc rằng mã của ta sẽ chạy được.

2.2. Cơ sở lý thuyết

Phần này sẽ trình bày về một số phương pháp thường được sử dụng trong các hệ gợi ý, đánh giá ưu nhược điểm của các phương pháp đó.

Có hai thực thể chính trong hệ gợi ý là *users* và *items*. Trong đó *users* là người dùng. *Items* là sản phẩm, ví dụ như các bộ phim, bài hát, cuốn sách, clip, hoặc cũng có thể là các *users* khác trong bài toán gợi ý kết bạn. Mục đích chính của các hệ gợi ý là dự đoán mức độ quan tâm của một *user* tới một *item* nào đó, qua đó có chiến lược gợi ý phù hợp.

Các hệ thống gợi ý thường sử dụng nhiều loại công nghệ khác nhau. Theo [5] các hệ gợi ý có thể được chia thành hai nhóm lớn:

- Gợi ý dựa trên nội dung (Content-based systems): đánh giá đặc tính của *items* được gợi ý. Ví dụ: một *user* xem rất nhiều các bộ phim hành động, vậy thì gợi ý một bộ phim trong cơ sở dữ liệu có chung đặc tính "hành động" tới *user* này, ví dụ phim Fast and Furious.
- Gợi ý dựa trên lọc cộng tác (Collaborative filtering): hệ thống gợi ý *items* dựa trên sự tương quan (similarity) giữa các *users* và/hoặc *items*. Có thể hiểu rằng ở nhóm này một *item* được gợi ý tới một *user* dựa trên những *users* có hành vi tương tự.

Đồ án này tập trung vào các phương pháp thuộc nhóm Collaborative filtering.

2.2.1 Utility matrix

Như đã đề cập, có hai thực thể chính trong các hệ gợi ý là *users* và *items*. Mỗi *user* sẽ có mức độ quan tâm (degree of preference) tới từng *item* khác nhau. Mức độ quan tâm này, nếu đã biết trước, được gán cho một giá trị ứng với mỗi cặp *user-item*. Giả sử rằng mức độ quan tâm được đo bằng giá trị *user rate* cho *item*, ta tạm gọi giá trị này là *rating*. Tập hợp tất cả các *ratings*, bao gồm cả những giá trị chưa biết cần được dự đoán, tạo nên một ma trận gọi là utility matrix.

Xét ví dụ (tham khảo [6]): Hình 2.1 biểu diễn đánh giá của 6 *users* A, B, C, D, E, F cho 5 bài hát. Các ô màu xanh thể hiện việc một *user* đã đánh giá một bài hát với *ratings*

⁴<https://docs.docker.com/get-started/overview/>, truy cập lần cuối 4/6/2021

từ 0 (không thích) đến 5 (rất thích). Các ô có dấu ‘?’ màu xám tương ứng với các ô chưa có dữ liệu. Công việc của một Hệ gợi ý là dự đoán giá trị tại các ô màu xám này, từ đó đưa ra gợi ý cho người dùng. Hệ gợi ý vì vậy đôi khi cũng được coi là bài toán Matrix Completion (Hoàn thiện ma trận).

	A	B	C	D	E	F
Mưa nửa đêm	5	5	0	0	1	?
Cỏ úa	5	?	?	0	?	?
Vùng lá me bay	?	4	1	?	?	1
Con cò bé bé	1	1	4	4	4	?
Em yêu trường em	1	0	5	?	?	?

Hình 2.1: Utility matrix biểu diễn đánh giá của người dùng cho các bài hát (0 - 5)

Thông thường, có rất nhiều *users* và *items* trong hệ thống, và mỗi *user* thường chỉ rate một số lượng rất nhỏ các *item*, thậm chí có những *user* không rate *item* nào (với những *users* này thì cách tốt nhất là gợi ý các *items* phổ biến nhất). Vì vậy, lượng ô màu xám của utility matrix trong các bài toán đó thường là rất lớn, và lượng các ô đã được điền là một số rất nhỏ.

2.2.2 Lọc cộng tác dựa trên láng giềng (NBCF)

Ý tưởng cơ bản của NBCF là xác định mức độ quan tâm của một *user* tới một *item* dựa trên các *users* khác gần giống với *user* này (*similar users*). Việc gần giống nhau giữa các *users* có thể được xác định thông qua mức độ quan tâm của các *users* này tới các *items* khác mà hệ thống đã biết. Ví dụ, A, B đều thích phim *Fast and Furious*, tức đều rate bộ phim này 5 sao. Ta đã biết A cũng thích *The Terminator*, vậy nhiều khả năng B cũng thích bộ phim này.

Việc xác định mức độ quan tâm của mỗi *user* tới một *item* dựa trên mức độ quan tâm của *similar users* tới *item* đó còn được gọi là *user-based collaborative filtering* (lọc cộng tác dựa trên *users*). Có một hướng tiếp cận khác được cho là làm việc hiệu quả hơn là *item-based collaborative filtering* (lọc cộng tác dựa trên *items*). Trong hướng tiếp cận này, thay vì xác định *user similarities*, hệ thống sẽ xác định *item similarities*. Từ đó, hệ thống gợi ý những *items* gần giống với những *items* mà *user* có mức độ quan tâm cao.

2.2.2.1 Lọc cộng tác dựa trên Users

Công việc quan trọng nhất phải làm trước tiên trong NBCF là phải xác định được sự giống nhau (*similarity*) giữa hai *users* hoặc *items*. Dữ liệu duy nhất chúng ta có là Utility matrix, vậy nên sự giống nhau này phải được xác định dựa trên các cột hoặc các hàng tương ứng với hai *users* (*items*) trong ma trận này.

Lọc cộng tác dựa trên users lần đầu tiên được giới thiệu bởi nghiên cứu của GroupLens để cung cấp các dự đoán được cá nhân hoá cho các bài báo trên Usenet. Với mỗi active user U_a trong tập các users U , bài toán đặt ra là dự đoán giá trị rating mà user sẽ đánh giá cho item I_t (chưa được đánh giá bởi U_a) trong tập items. Các bước trong lọc cộng tác dựa trên users được thực hiện như sau:

1. Tính toán sự tương đồng (similarity) giữa user U_a và các user khác.
2. Xác định tập k users láng giềng có giá trị tương đồng cao nhất với user U_a .
3. Cuối cùng, dự đoán cho item I_t được xác định bằng cách lấy trung bình có trọng số của các *ratings* cho item I_t trong tập k users láng giềng.

Trong bước 1, để tính toán sự tương đồng giữa các users ta sử dụng độ đo Pearson correlation, đây là độ đo tương đồng phổ biến cho user-based CF. Đặt tập các items được đánh giá bởi cả users u và v là I , hệ số tương đồng $Sim_{u,v}$ giữa chúng được tính như sau:

$$Sim_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}} \quad (2.1)$$

trong đó $r_{u,i}$ là rating của user u cho item i và \bar{r}_u là trung bình rating của user u trên tất cả các items mà user này đã đánh giá. Tương tự, $r_{v,i}$ là rating của user v cho item i và \bar{r}_v là trung bình rating của user v trên tất cả các items mà user này đã đánh giá. Ngoài ra có thể sử dụng độ đo cosine để đánh giá sự tương đồng. Tuy nhiên các đánh giá trong [3] chỉ ra rằng các kết quả của Pearson correlation đem lại độ chính xác tốt hơn đối với lọc cộng tác dựa trên users.

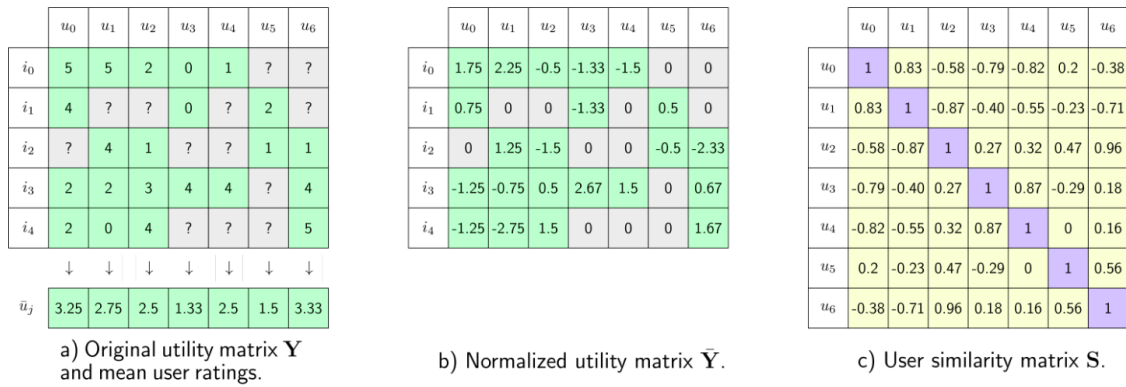
Sau khi các giá trị tương đồng được tính, tập các user tương đồng nhất với user U_a được xác định trong bước 2.

Trong bước 3, ta thực hiện tính toán các dự đoán đánh giá $P_{u,i}$ của user u cho item i , công thức 2.2 là công thức tính $P_{u,i}$ đã được điều chỉnh để tính đến thực tế là những user khác nhau có các đánh giá khác nhau (user dễ tính hoặc user khó tính) .

$$P_{u,i} = \bar{r}_u + \frac{\sum_{v \in V} Sim_{u,v}(r_{v,i} - \bar{r}_v)}{\sum_{v \in V} |Sim_{u,v}|} \quad (2.2)$$

trong đó V là tập k users láng giềng của user u mà đã đánh giá item i .

Có thể thấy, với mỗi cặp $(user, item)$ ta sẽ phải thực hiện qua các bước đã trình bày ở trên. Tuy nhiên ở bước 1, việc thực hiện tính toán giá trị tương đồng giữa các cặp user có thể sẽ bị lặp lại, do đó ta sẽ thực hiện tính toán toàn bộ các giá trị này trước và lưu lại, khi đó ta chỉ việc lấy các giá trị tương ứng với user U_a và các user khác mà không cần phải thực hiện lại quá trình tính toán. Toàn bộ các giá trị này sẽ hình thành nên một ma trận tương đồng gọi là *Similarity matrix* (Hình 2.2).



Hình 2.2: Xây dựng user similarity matrix [6]

Hàng cuối cùng trong Hình 2.2a) là giá trị trung bình của *ratings* cho mỗi *user*. Giá trị cao tương ứng với các *user* dễ tính và ngược lại. Khi đó, nếu tiếp tục trừ từ mỗi *rating* đi giá trị này và thay các giá trị chưa biết bằng 0, ta sẽ được *normalized utility matrix* như trong Hình 2.2b). Hình 2.2c) là ma trận *Similarity matrix*, chính là kết quả của quá trình tính toán sự tương đồng giữa các cặp *user*.

2.2.2.2 Lựa chọn tác dựa trên Items

Một số hạn chế của User-based CF:

- Trên thực tế, số lượng *users* luôn lớn hơn số lượng *items* rất nhiều. Kéo theo đó là *Similarity matrix* là rất lớn với số phần tử phải lưu giữ là hơn 1 nửa của bình phương số lượng *users* (chú ý rằng ma trận này là đối xứng). Việc này, như đã đề cập ở trên, khiến cho việc lưu trữ ma trận này trong nhiều trường hợp là không khả thi.
- Ma trận Utility \mathbf{Y} thường là rất thưa. Với số lượng *users* rất lớn so với số lượng *items*, rất nhiều cột của ma trận này sẽ rất thưa, tức chỉ có một vài phần tử khác 0. Lý do là *users* thường lười *rating*. Cũng chính vì việc này, một khi *user* đó thay đổi *rating* hoặc rate thêm *items*, trung bình cộng các *ratings* cũng như vector chuẩn hoá tương ứng với *user* này thay đổi nhiều. Kéo theo đó, việc tính toán ma trận *Similarity*, vốn tốn nhiều bộ nhớ và thời gian, cũng cần được thực hiện lại.

Ngược lại, nếu chúng ta tính toán *similarity* giữa các *items* rồi *recommend* những *items* gần giống với *item* yêu thích của một *user* thì sẽ có những lợi ích sau:

- Vì số lượng *items* thường nhỏ hơn số lượng *users*, *Similarity matrix* trong trường hợp này cũng nhỏ hơn nhiều, thuận lợi cho việc lưu trữ và tính toán ở các bước sau.
- Vì số lượng phần tử đã biết trong *Utility matrix* là như nhau nhưng số hàng (*items*) ít hơn số cột (*users*), nên trung bình, mỗi hàng của ma trận này sẽ có nhiều phần tử đã biết hơn số phần tử đã biết trong mỗi cột. Việc này cũng dễ hiểu vì mỗi *item* có thể được rated bởi nhiều *users*. Kéo theo đó, giá trị trung bình của mỗi hàng ít bị thay đổi hơn khi có thêm một vài *ratings*. Như vậy, việc cập nhật ma trận *Similarity Matrix* có thể được thực hiện ít thường xuyên hơn.

Cách tiếp cận thứ hai này được gọi là *Lọc cộng tác dựa trên Items*. Hướng tiếp cận này được sử dụng nhiều trong thực tế hơn. Điểm khác biệt chính giữa lọc cộng tác dựa trên items và lọc cộng tác dựa trên users là lọc cộng tác dựa trên items đưa ra các dự đoán dựa trên mô hình tương quan giữa item-item thay vì tương quan giữa user-user. Các bước thực hiện trong lọc cộng tác dựa trên items cũng tương tự như trên users:

1. Tính toán giá trị tương đồng giữa các item.
2. Từ tập các items được đánh giá bởi user mục tiêu (U_a), chọn ra k items tương đồng nhất với item mục tiêu (I_t) (k items láng giềng).
3. Tính toán giá trị dự đoán cho item mục tiêu bằng cách lấy trung bình có trọng số các *ratings* của user mục tiêu trên k items đã xác định ở bước 2. Các trọng số được sử dụng là các giá trị tương đồng giữa item mục tiêu và k items láng giềng đã được đánh giá bởi user mục tiêu.

Trong lọc cộng tác dựa trên items thì độ đo tương đồng cosin thường sử dụng. Đặt U là tập users đã đánh giá cả items i và j , hệ số tương đồng $Sim_{i,j}$ giữa hai item được xác định bởi:

$$Sim_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_u)^2}} \quad (2.3)$$

trong đó $r_{u,i}$ là giá trị rating của user u cho item i và \bar{r}_u là giá trị trung bình rating của user u trên các items mà user này đã đánh giá. Tương tự, $r_{u,j}$ là giá trị rating của user u cho item j .

Việc tính toán các giá trị dự đoán $P_{u,i}$ của user u cho item i được thực hiện theo công thức 2.4:

$$P_{u,i} = \frac{\sum_{j \in I} Sim_{i,j} \cdot r_{u,j}}{\sum_{j \in I} |Sim_{i,j}|} \quad (2.4)$$

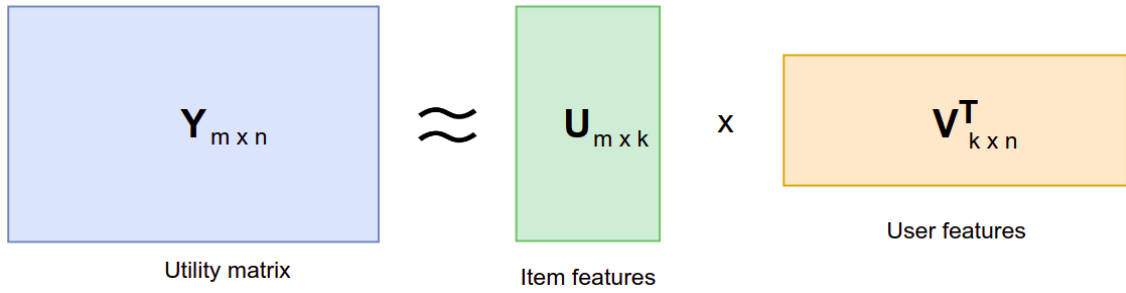
trong đó I là tập k items tương đồng nhất với item mục tiêu i , được chọn từ tập các items đã được đánh giá bởi user u và $r_{u,j}$ là rating của user u cho item j .

2.2.3 Lọc cộng tác dựa trên phân tích ma trận (MF/CF)

Lọc cộng tác dựa trên phân tích ma trận (Matrix Factorization collaborative filtering) là cách tiếp cận dựa trên mô hình với giả thiết *users* và *items* được biểu diễn trên cùng không gian ẩn mà thể hiện đặc trưng của chúng (latent factor space).

Theo hướng tiếp cận này, ta xấp xỉ *Utility Matrix* $\mathbf{Y} \in \mathbb{R}^{m \times n}$ bằng tích của hai ma trận có bậc thấp hơn $\mathbf{U} \in \mathbb{R}^{m \times k}$ và $\mathbf{V} \in \mathbb{R}^{n \times k}$. (Hình 2.3). Lúc này, mỗi phần tử y_{ij} (rating của user j lên item i) trong ma trận \mathbf{Y} được xấp xỉ bằng tích của hàng i (u_i) trong ma trận \mathbf{U} và hàng j (v_j) trong ma trận \mathbf{V} (hay cột j trong ma trận \mathbf{V}^T):

$$y_{ij} \approx u_i \cdot v_j^T \quad (2.5)$$



Hình 2.3: Utility matrix Y xấp xỉ bằng tích của hai ma trận U và V

Thông thường, k được chọn là một số nhỏ hơn rất nhiều so với m, n . Khi đó, cả hai ma trận U và V đều có rank không vượt quá k .

Ý tưởng chính đằng sau Matrix Factorization cho Recommendation Systems là tồn tại các *latent features* (tính chất ẩn) mô tả sự liên quan giữa các *items* và *users*. Ví dụ với hệ thống gợi ý môn học, tính chất ẩn có thể là số tín chỉ học phần, số tín chỉ học phí, chuyên ngành, lĩnh vực..., cũng có thể là một sự kết hợp nào đó của các thuộc tính này, hoặc cũng có thể là bất cứ điều gì mà chúng ta không thực sự cần đặt tên. Mỗi *item* i sẽ mang tính chất ẩn ở một mức độ nào đó tương ứng với vector hệ số u_i của nó, hệ số càng cao tương ứng với việc mang tính chất đó càng cao. Tương tự, mỗi *user* j cũng sẽ có xu hướng thích những tính chất ẩn nào đó và được mô tả bởi vector hệ số v_j của nó. Hệ số cao tương ứng với việc các môn học có tính chất ẩn đó phù hợp với users đó. Giá trị của biểu thức $u_i \cdot v_j^T$ (2.5) sẽ cao nếu các thành phần tương ứng của u_i, v_j đều cao. Điều này nghĩa là *item* mang các tính chất ẩn mà *user* thích, vậy nên gợi ý *item* đó cho *user*.

Ngoài ra, MFCF còn giúp giảm lượng bộ nhớ lưu trữ khi chỉ cần lưu hai ma trận U và V thay vì lưu toàn bộ *Similarity matrix* trong NBCF. Cụ thể ta chỉ cần bộ nhớ để lưu trữ $k \times (m + n)$ phần tử so với m^2 hoặc n^2 phần tử của *Similarity matrix*.

2.2.3.1 Hàm mất mát

Giả sử số *users* là n , số *items* là m , utility matrix được mô tả bởi ma trận Y . Thành phần ở hàng i cột j (y_{ij}) của Y là *rating* của *user* thứ j lên *item* i . Ma trận Y bị khuyết nhiều vị trí tương ứng với các giá trị mà hệ thống cần phải dự đoán.

Từ biểu thức 2.5 ta coi hiệu $u_i v_j^T - y_{ij}$ là giá trị mất mát tại phần tử (i, j) , việc tìm hai ma trận U và V được đưa về việc tối thiểu các giá trị mất mát này trên tập các *ratings* đã có. Ta xây dựng được hàm mất mát như sau:

$$loss(U, V) = \sum_{j=1}^n \sum_{i:r_{ij}=1} \underbrace{(u_i v_j^T - y_{ij})^2}_{\text{thành phần 1}} + \underbrace{\lambda(\|u_i\| + \|v_j\|)}_{\text{thành phần 2}} \quad (2.6)$$

trong đó:

- Thành phần thứ nhất chính là sai số của mô hình. Thành phần thứ hai trong hàm mất mát phía trên là regularization term, giúp tránh overfitting.
- $r_{ij} = 1$ nếu *item* thứ i đã được đánh giá bởi *user* thứ j .

$\|\cdot\|$ là Frobenius norm, tức căn bậc hai của tổng bình phương tất cả các phần tử của ma trận.

2.2.3.2 Vấn đề thiên lệch

Một lợi thế của hướng tiếp cận Matrix Factorization cho Collaborative Filtering là khả năng linh hoạt của nó khi có thêm các điều kiện ràng buộc khác, các điều kiện này có thể liên quan đến quá trình xử lý dữ liệu hoặc đến từng ứng dụng cụ thể.

Các giá trị *ratings* thực tế đều có những thiên lệch về *users* hoặc *items*. Lấy ví dụ trong hệ thống gợi ý môn học, sẽ có các yếu tố ảnh hưởng làm cho kết quả gợi ý không còn chính xác, có thể kể đến như yêu cầu môn học cao hay thấp, khó hay dễ (item effect) hoặc do một số ảnh hưởng của sinh viên như sinh viên giỏi, thông minh, lười biếng khác nhau (user effect). Vấn đề thiên lệch có thể được giải quyết bằng các biến gọi là *bias*, phụ thuộc vào mỗi *user* và *item* và có thể được tối ưu cùng với U và V . Khi đó, *rating* của *user* j lên *item* i không chỉ được xấp xỉ bằng $u_i v_j^T$ mà còn phụ thuộc vào các *biases* của *item* i và *user* j nữa. Ngoài ra giá trị này cũng có thể phụ thuộc vào giá trị trung bình của toàn bộ *ratings* nữa:

$$y_{ij} \approx u_i v_j^T + b_i + d_j + \mu$$

với b_i, d_j, μ lần lượt là bias của *item* i , *user* j và giá trị trung bình của toàn bộ các *ratings* (hằng số).

Hàm mất mát 2.6 trở thành:

$$loss(U, V, b, d) = \sum_{j=1}^n \sum_{i:r_{ij}=1} (u_i v_j^T + b_i + d_j + \mu - y_{ij})^2 + \lambda(\|u_i\| + \|v_j\| + b_i^2 + d_j^2) \quad (2.7)$$

2.2.3.3 Tối ưu hàm mất mát

Xét sự mất mát tại phần tử i, j , ta đặt

$$e_{ij} = u_i v_j^T + b_i + d_j + \mu - y_{ij}$$

Khi đó hàm mất mát có thể được viết lại thành:

$$loss(U, V, b, d) = \sum_{j=1}^n \sum_{i:r_{ij}=1} e_{ij}^2 + \lambda(\|u_i\| + \|v_j\| + b_i^2 + d_j^2)$$

Việc tối ưu đồng thời U, V, b, d là tương đối phức tạp, thay vào đó, phương pháp được sử dụng là tối ưu lần lượt từng phần tử trong công thức 2.7, tức ta đi tối ưu:

$$loss(u_i, v_j, b_i, d_j) = e_{ij}^2 + \lambda(\|u_i\| + \|v_j\| + b_i^2 + d_j^2)$$

Khi cố định u_i, b_i, d_j việc tối ưu v_j được đưa về tối ưu hàm (chú ý: $u_i v_j^T = v_j u_i^T$):

$$loss(v_j) = e_{ij}^2 + \lambda\|v_j\| = (v_j u_i^T + b_i + d_j + \mu - y_{ij})^2 + \lambda\|v_j\|$$

Tương tự:

$$loss(d_j) = e_{ij}^2 + \lambda d_j^2 = (u_i v_j^T + b_i + d_j + \mu - y_{ij})^2 + \lambda d_j^2$$

$$loss(u_i) = e_{ij}^2 + \lambda \|u_i\|^2 = (u_i v_j^T + b_i + d_j + \mu - y_{ij})^2 + \lambda \|u_i\|^2$$

$$loss(b_i) = e_{ij}^2 + \lambda b_i^2 = (u_i v_j^T + b_i + d_j + \mu - y_{ij})^2 + \lambda b_i^2$$

Các bài toán này đều được tối ưu được bằng phương pháp Gradient Descent [7].

Thực hiện tính toán đạo hàm (tham khảo [1]), ta được:

$$\frac{\partial loss(v_j)}{\partial v_j} = 2(v_j u_i^T + b_i + d_j + \mu - y_{ij})u_i + 2\lambda v_j = 2e_{ij}u_i + 2\lambda v_j$$

$$\frac{\partial loss(d_j)}{\partial d_j} = 2(u_i v_j^T + b_i + d_j + \mu - y_{ij}) + 2\lambda d_j = 2e_{ij} + 2\lambda d_j$$

$$\frac{\partial loss(u_i)}{\partial u_i} = 2(u_i v_j^T + b_i + d_j + \mu - y_{ij})v_j + 2\lambda u_i = 2e_{ij}v_j + 2\lambda u_i$$

$$\frac{\partial loss(b_i)}{\partial b_i} = 2(u_i v_j^T + b_i + d_j + \mu - y_{ij}) + 2\lambda b_i = 2e_{ij} + 2\lambda b_i$$

Vậy công thức cập nhật cho các thành phần u_i, v_j, b_i, d_j với tốc độ học β là:

$$v_j = v_j - 2\beta(e_{ij}u_i + \lambda v_j) \quad (2.8)$$

$$d_j = d_j - 2\beta(e_{ij} + \lambda d_j) \quad (2.9)$$

$$u_i = u_i - 2\beta(e_{ij}v_j + \lambda u_i) \quad (2.10)$$

$$b_i = b_i - 2\beta(e_{ij} + \lambda b_i) \quad (2.11)$$

2.2.3.4 Điều kiện dừng

Trong thực nghiệm, có một vài phương pháp như dưới đây:

1. Giới hạn số vòng lặp: đây là phương pháp phổ biến nhất và cũng để đảm bảo rằng chương trình chạy không quá lâu. Tuy nhiên, một nhược điểm của cách làm này là có thể thuật toán dừng lại trước khi đủ gần với nghiệm.
2. So sánh gradient của nghiệm tại hai lần cập nhật liên tiếp, khi nào giá trị này đủ nhỏ thì dừng lại. Phương pháp này cũng có một nhược điểm lớn là việc tính đạo hàm đôi khi trở nên quá phức tạp (ví dụ như khi có quá nhiều dữ liệu), nếu áp dụng phương pháp này thì coi như ta không được lợi khi sử dụng SGD và mini-batch GD.
3. So sánh giá trị của hàm mất mát của nghiệm tại hai lần cập nhật liên tiếp, khi nào giá trị này đủ nhỏ thì dừng lại. Nhược điểm của phương pháp này là nếu tại một thời điểm, đồ thị hàm số có dạng bằng phẳng tại một khu vực nhưng khu vực đó không chứa điểm local minimum (khu vực này thường được gọi là saddle points), thuật toán cũng dừng lại trước khi đạt giá trị mong muốn. Trong SGD và mini-batch GD, cách thường dùng là so sánh nghiệm sau một vài lần cập nhật.

2.2.3.5 Dự đoán kết quả

Sau khi việc tối ưu hoàn thành, quá trình dự đoán được tính theo công thức:

$$\hat{y}_{ij} = u_i v_j^T + b_i + d_j + \mu \quad (2.12)$$

trong đó \hat{y}_{ij} là giá trị *rating* dự đoán của *user j* lên *item i*; u_i là vector hàng thứ i của ma trận U ; v_j là vector hàng thứ j của ma trận V ; b_i, d_j lần lượt là giá trị bias của *item i* và *user j*.

2.2.4 Phương pháp đánh giá mô hình dự đoán

Hiện nay có rất nhiều tiêu chuẩn khác nhau để đánh giá mức độ phù hợp của các mô hình, nhưng không thể khẳng định được tiêu chuẩn nào tốt hơn. Trong phạm vi đồ án này, em sử dụng tiêu chuẩn sai số RMSE (Root-mean-square error) để so sánh các mô hình dự đoán.

Root-mean-square error (RMSE) là một phép đo thường được sử dụng để đánh giá sự khác biệt giữa các giá trị được dự đoán bởi một mô hình hoặc một công cụ ước lượng và các giá trị quan sát được. RMSE là căn bậc hai của trung bình moment bậc hai của sai số giữa các giá trị dự đoán và giá trị quan sát. RMSE là thước đo độ chính xác để so sánh lỗi dự đoán của các mô hình khác nhau trên một tập dữ liệu cụ thể (Theo Wikipedia [8]). RMSE càng nhỏ tức là sai số càng bé thì độ tin cậy của mô hình càng cao.

Công thức tính RMSE:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (2.13)$$

trong đó \hat{y}_i là giá trị dự đoán, y_i là giá trị quan sát được.

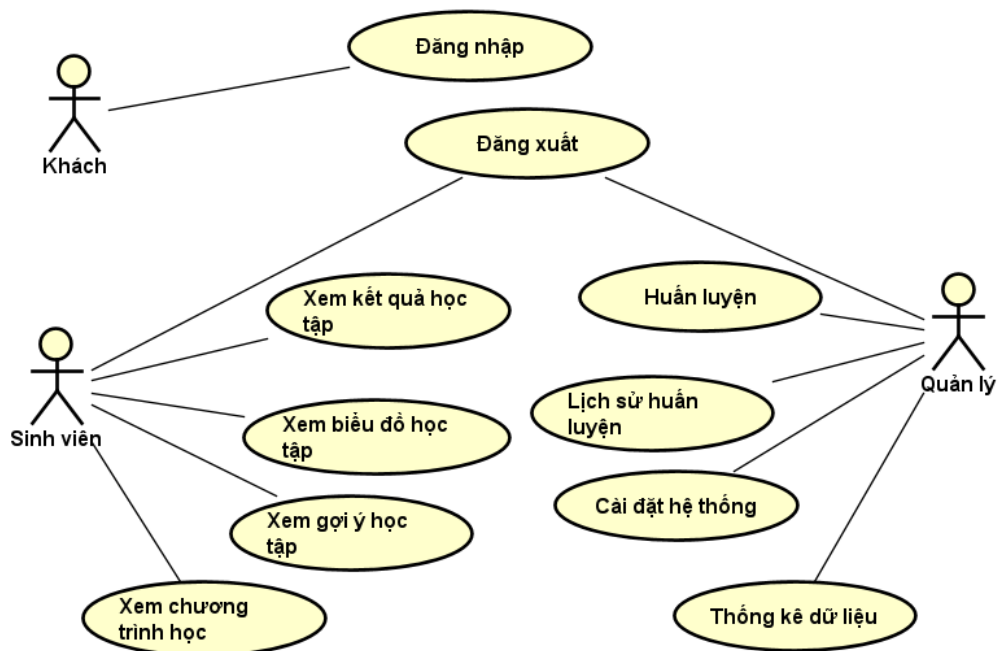
Chương 3: Xây dựng hệ thống website gợi ý đăng ký học tập

3.1. Phân tích hệ thống

3.1.1 Tổng quan chức năng

3.1.1.1 Biểu đồ use case tổng quan

Hệ thống gợi ý đăng ký học tập gồm nhiều thành phần và chức năng khác nhau. **Hình 3.1** mô tả tổng quan chức năng của toàn bộ hệ thống.



Hình 3.1: Biểu đồ use case tổng quan hệ thống gợi ý học tập

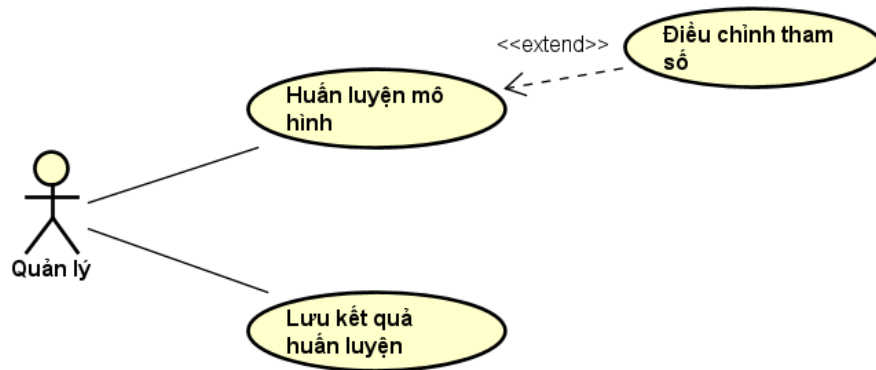
Hệ thống gồm có ba tác nhân là *khách*, *sinh viên* và *quản lý*, mỗi tác nhân có những vai trò nhất định:

- Tác nhân *khách* có thể đăng nhập vào hệ thống bằng tài khoản đã được cấp trước đó với các vai trò khác nhau, là sinh viên hoặc quản lý.
- Mỗi *sinh viên* sau khi đăng nhập có thể theo dõi các kết quả học tập, biểu đồ học tập của mình, xem chương trình học hiện tại và xem các gợi ý đăng ký học tập do hệ thống đề xuất.
- Bên cạnh đó, hệ thống cần một người với vai trò quản lý để có thể giám sát và cập nhật hệ thống.

3.1.1.2 Biểu đồ use case phân rã

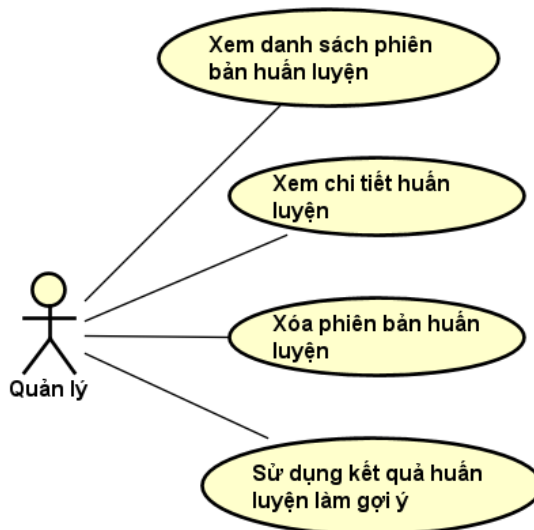
Biểu đồ phân rã use case Huấn luyện

Chức năng *huấn luyện* chỉ có thể được sử dụng bởi *quản lý* và được phân rã thành các chức năng nhỏ hơn: quản lý có thể *huấn luyện mô hình* với bộ tham số huấn luyện mặc định hoặc tự *điều chỉnh tham số huấn luyện*, sau khi quá trình huấn luyện kết thúc hệ thống sẽ hiển thị các kết quả và đánh giá chất lượng mô hình. Lúc này quản lý có thể lưu lại các kết quả huấn luyện và sử dụng kết quả này cho chức năng gợi ý của hệ thống.



Hình 3.2: Biểu đồ phân rã use case Huấn luyện

Biểu đồ phân rã use case Lịch sử huấn luyện

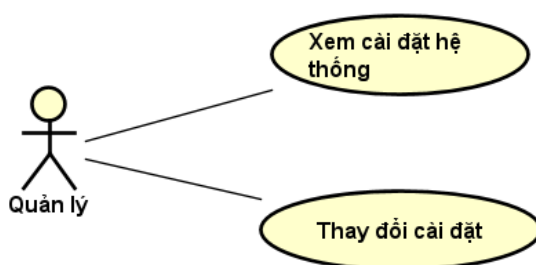


Hình 3.3: Biểu đồ phân rã use case Lịch sử huấn luyện

Quản lý có thể theo dõi lịch sử các phiên bản huấn luyện đã lưu trước đó, có thể xem chi tiết các huấn luyện, sử dụng kết quả của một phiên bản huấn luyện phù hợp làm dữ liệu đánh giá môn học cho hệ thống. Với những phiên bản huấn luyện không còn phù hợp nữa quản lý có thể xóa khỏi hệ thống.

Biểu đồ phân rã use case Cài đặt hệ thống

Với chức năng này, người quản lý có thể xem và điều chỉnh các cài đặt hệ thống như tham số huấn luyện mặc định, phiên bản huấn luyện đang sử dụng...



Hình 3.4: Biểu đồ phân rã use case Cài đặt hệ thống

3.1.2 Đặc tả chức năng

Bảng 3.1: Danh sách use case

Nhóm use case	Mã use case	Tên use case
Tổng quan	UC001	Đăng nhập
	UC002	Đăng xuất
	UC003	Xem kết quả học tập
	UC004	Xem biểu đồ học tập
	UC005	Xem gợi ý học tập
	UC006	Xem chương trình học
	UC007	Thống kê dữ liệu
Huấn luyện	UC008	Huấn luyện mô hình
	UC009	Điều chỉnh tham số
	UC010	Lưu kết quả huấn luyện
Lịch sử huấn luyện	UC011	Xem danh sách phiên bản huấn luyện
	UC012	Xem chi tiết huấn luyện
	UC013	Xóa phiên bản huấn luyện
	UC014	Sử dụng kết quả huấn luyện làm gợi ý
Cài đặt hệ thống	UC015	Xem cài đặt hệ thống
	UC016	Thay đổi cài đặt

3.1.2.1 Đặc tả use case Đăng nhập

Bảng 3.2: Đặc tả use case Đăng nhập

Mã use case	UC001	Tên use case	Đăng nhập
Tác nhân	Sinh viên, quản lý		
Tiền điều kiện	Không		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu đăng nhập hệ thống
	2.	Hệ thống	Hiển thị giao diện đăng nhập
	3.	Người dùng	Chọn vai trò rồi nhập tài khoản và mật khẩu
	4.	Người dùng	Nhấn nút đăng nhập
	5.	Hệ thống	Kiểm tra các trường thông tin bắt buộc
	6.	Hệ thống	Kiểm tra tài khoản có trong hệ thống hay không
	7.	Hệ thống	Tạo menu chức năng người dùng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a.	Hệ thống	Thông báo lỗi nếu các trường bắt buộc chưa được nhập
	6a.	Hệ thống	Thông báo lỗi "tài khoản không chính xác" nếu không tồn tại tài khoản trong hệ thống
Hậu điều kiện	Không		

* Dữ liệu đầu vào khi đăng nhập

STT	Trường dữ liệu	Mô tả	Điều kiện hợp lệ	Bắt buộc
1.	Vai trò	xác định quyền của user	nhận giá trị 'student' hoặc 'manager'	có
2.	Tên tài khoản		không chứa khoảng trắng	có
3.	Mật khẩu			có

3.1.2.2 Đặc tả use case Xem kết quả học tập

Bảng 3.4: Đặc tả use case Xem kết quả học tập

Mã use case	UC003	Tên use case	Xem kết quả học tập
Tác nhân	Sinh viên		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò sinh viên		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu xem kết quả học tập
	2.	Hệ thống	Lấy dữ liệu học tập và thông tin của sinh viên
	3.	Hệ thống	Hiển thị giao diện thông tin sinh viên và kết quả học tập
Luồng sự kiện thay thế			
Hậu điều kiện	Không		

** Dữ liệu đầu ra khi hiển thị thông tin sinh viên

STT	Trường dữ liệu	Mô tả
1.	Sinh viên	tên của sinh viên
2.	MSSV	
3.	Email	
4.	Chương trình	chương trình đào tạo sinh viên đang theo học
5.	Năm nhập học	
6.	Trạng thái	tình học học tập của sinh viên: Chưa tốt nghiệp hay Đã tốt nghiệp

3.1.2.3 Đặc tả use case Xem biểu đồ học tập

Bảng 3.6: Đặc tả use case Xem biểu đồ học tập

Mã use case	UC004	Tên use case	Xem biểu đồ học tập
Tác nhân	Sinh viên		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò sinh viên		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu xem biểu đồ học tập
	2.	Hệ thống	Tổng kết dữ liệu học tập của sinh viên
	3.	Hệ thống	Hiển thị giao diện các biểu đồ phân tích
Luồng sự kiện thay thế			
Hậu điều kiện	Không		

3.1.2.4 Đặc tả use case Xem gợi ý học tập

Bảng 3.7: Đặc tả use case Xem gợi ý học tập

Mã use case	UC005	Tên use case	Xem gợi ý học tập
Tác nhân	Sinh viên		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò sinh viên		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu xem gợi ý học tập
	2.	Hệ thống	Kiểm tra sinh viên đã tốt nghiệp hay chưa
	3.	Hệ thống	Lấy dữ liệu các học phần còn nợ, các học phần bắt buộc
	4.	Hệ thống	Tính toán dữ liệu dự đoán các học phần tự do
	5.	Hệ thống	Hiển thị danh sách các môn học gợi ý
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a.	Hệ thống	Hiển thị giao diện thông báo sinh viên đã tốt nghiệp và kết thúc use case
Hậu điều kiện	Không		

3.1.2.5 Đặc tả use case Xem chương trình học

Bảng 3.8: Đặc tả use case Xem biểu đồ học tập

Mã use case	UC006	Tên use case	Xem chương trình học
Tác nhân	Sinh viên		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò sinh viên		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu xem chương trình học
	2.	Hệ thống	Lấy dữ liệu chương trình học của sinh viên
	3.	Hệ thống	Hiển thị chương trình học của sinh viên
Luồng sự kiện thay thế			
Hậu điều kiện	Không		

3.1.2.6 Đặc tả use case Huấn luyện mô hình

Bảng 3.9: Đặc tả use case Huấn luyện mô hình

Mã use case	UC008	Tên use case	Huấn luyện mô hình
Tác nhân	Quản lý		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò quản lý		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu chức năng huấn luyện
	2.	Hệ thống	Hiển thị giao diện huấn luyện với tham số mặc định
	3.	Người dùng	Nhấn nút huấn luyện
	4.	Hệ thống	Chuẩn bị dữ liệu, chuẩn hóa và thực hiện huấn luyện mô hình.
	5.	Hệ thống	Hiển thị các thông báo, quá trình thực thi huấn luyện qua giao diện log messages.
	6.	Hệ thống	Hiển thị kết quả khi huấn luyện hoàn tất.

Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a.	Người dùng	Chọn chức năng điều chỉnh tham số
	3b.	Hệ thống	Gọi đến use case Điều chỉnh tham số
	3c.	Hệ thống	Quay lại bước 3 ở luồng chính
Hậu điều kiện	Không		

**** Dữ liệu đầu ra khi hiển thị kết quả huấn luyện**

STT	Trường dữ liệu	Mô tả
1.	Data size	kích thước dữ liệu huấn luyện
2.	Test size	kích thước dữ liệu dùng để test
3.	RMSE	đánh giá sai số mô hình
4.	Loss	giá trị mất mát của mô hình

3.1.2.7 Đặc tả use case Điều chỉnh tham số

Bảng 3.11: Đặc tả use case Điều chỉnh tham số

Mã use case	UC009	Tên use case	Điều chỉnh tham số
Tác nhân	Quản lý		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò quản lý và đã chọn mục điều chỉnh tham số trong giao diện huấn luyện		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Hệ thống	Hiển thị form điều chỉnh tham số huấn luyện
	2.	Người dùng	Thay đổi giá trị các tham số
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a.	Hệ thống	Thông báo lỗi nếu tham số không hợp lệ
Hậu điều kiện	Không		

* Dữ liệu đầu vào khi huấn luyện với tham số điều chỉnh

STT	Trường dữ liệu	Mô tả	Điều kiện hợp lệ	Bắt buộc
1.	Number factor (K)	số nhân tố tiềm ẩn	nhận giá trị nguyên dương	có
2.	Max iterator	số vòng lặp tối đa	nhận giá trị nguyên dương	có
3.	Test size	kích thước test sẽ được chia từ tập dữ liệu	nằm trong đoạn [0,1]	có
4.	Lambda	tham số regularization	giá trị không âm	có
5.	Learning rate	tốc độ học	giá trị không âm	có
6.	Print every	theo dõi sự thay đổi chất lượng mô hình sau mỗi "print every" vòng lặp	giá trị nguyên dương	có

3.1.2.8 Đặc tả use case Lưu kết quả huấn luyện

Bảng 3.13: Đặc tả use case Lưu kết quả huấn luyện

Mã use case	UC010	Tên use case	Lưu kết quả huấn luyện
Tác nhân	Quản lý		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò quản lý UC008 Huấn luyện mô hình vừa hoàn thành		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Nhấn nút lưu kết quả huấn luyện
	2.	Hệ thống	Kiểm tra kết quả đã được lưu hay chưa
	3.	Hệ thống	Hệ thống tính toán các kết quả dự đoán và thực hiện lưu lại vào cơ sở dữ liệu
	4.	Hệ thống	Thông báo lưu thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a.	Hệ thống	Thực hiện bước 4 ở luồng chính nếu kết quả huấn luyện đã được lưu
Hậu điều kiện	Không		

3.1.2.9 Đặc tả use case Xem danh sách phiên bản huấn luyện

Bảng 3.14: Đặc tả use case Xem danh sách phiên bản huấn luyện

Mã use case	UC011	Tên use case	Xem danh sách phiên bản huấn luyện
Tác nhân	Quản lý		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò quản lý		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu xem lịch sử huấn luyện
	2.	Hệ thống	Lấy dữ liệu các phiên bản huấn luyện
	3.	Hệ thống	Hiển thị danh sách huấn luyện
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a.	Hệ thống	Hiển thị thông báo không có dữ liệu nếu danh sách rỗng
Hậu điều kiện	Không		

3.1.2.10 Đặc tả use case Xem chi tiết huấn luyện

Bảng 3.15: Đặc tả use case Xem chi tiết huấn luyện

Mã use case	UC012	Tên use case	Xem chi tiết huấn luyện
Tác nhân	Quản lý		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò quản lý		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Nhấn vào một phiên bản trong danh sách lịch sử huấn luyện
	2.	Hệ thống	Hiển thị chi tiết các thông tin của phiên bản huấn luyện
Luồng sự kiện thay thế			
Hậu điều kiện	Không		

3.1.2.11 Đặc tả use case Xóa phiên bản huấn luyện

Bảng 3.16: Đặc tả use case Xóa phiên bản huấn luyện

Mã use case	UC013	Tên use case	Xóa phiên bản huấn luyện
Tác nhân	Quản lý		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò quản lý và đang ở giao diện xem chi tiết huấn luyện		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu xóa phiên bản huấn luyện
	2.	Hệ thống	Hiển thị form xác nhận xóa
	3.	Người dùng	Xác nhận xóa
	4.	Hệ thống	Kiểm tra phiên bản huấn luyện có tồn tại hay không
	5.	Hệ thống	Xóa dữ liệu liên quan đến phiên bản huấn luyện
	6.	Hệ thống	Thông báo xóa thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a.	Người dùng	Kết thúc use case nếu người dùng xác nhận hủy bỏ
	4a.	Hệ thống	Thông báo lỗi phiên bản không tồn tại nếu không tìm thấy trong cơ sở dữ liệu và kết thúc use case
Hậu điều kiện	Không		

3.1.2.12 Đặc tả use case Sử dụng kết quả huấn luyện làm gợi ý

Bảng 3.17: Đặc tả use case Sử dụng kết quả huấn luyện làm gợi ý

Mã use case	UC014	Tên use case	Sử dụng kết quả huấn luyện làm gợi ý
Tác nhân	Quản lý		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò quản lý và đang ở giao diện xem chi tiết huấn luyện		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Nhấn nút sử dụng phiên bản
	2.	Hệ thống	Kiểm tra phiên bản huấn luyện có tồn tại hay không
	3.	Hệ thống	Thay đổi phiên bản huấn luyện sử dụng cho gợi ý
	4.	Hệ thống	Thông báo thay đổi thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a.	Hệ thống	Thông báo lỗi phiên bản không tồn tại nếu không tìm thấy trong cơ sở dữ liệu và kết thúc use case
Hậu điều kiện	Không		

3.1.2.13 Đặc tả use case Xem cài đặt hệ thống

Bảng 3.18: Đặc tả use case Xem cài đặt hệ thống

Mã use case	UC015	Tên use case	Xem cài đặt hệ thống
Tác nhân	Quản lý		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò quản lý		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu chức năng cài đặt hệ thống
	2.	Hệ thống	Lấy dữ liệu cài đặt từ cơ sở dữ liệu
	3.	Hệ thống	Hiển thị giao diện cài đặt
Luồng sự kiện thay thế			
Hậu điều kiện	Không		

3.1.2.14 Đặc tả use case Thay đổi cài đặt

Bảng 3.19: Đặc tả use case Thay đổi cài đặt

Mã use case	UC016	Tên use case	Thay đổi cài đặt
Tác nhân	Quản lý		
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống với vai trò quản lý		
Luồng sự kiện chính (thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Nhấn nút chỉnh sửa ở mỗi mục cài đặt
	2.	Hệ thống	Hiển thị giao diện chỉnh sửa cho mục cài đặt tương ứng
	3.	Người dùng	Thực hiện chỉnh sửa cài đặt
	4.	Người dùng	Yêu cầu lưu lại thay đổi
	5.	Hệ thống	Kiểm tra giá trị cài đặt
	6.	Hệ thống	Lưu lại thay đổi vào cơ sở dữ liệu
	7.	Hệ thống	Thông báo thay đổi thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a.	Người dùng	Yêu cầu hủy bỏ thay đổi
	4b.	Hệ thống	Trở lại giao diện xem cài đặt
	5a.	Hệ thống	Hiển thị thông báo lỗi nếu giá trị cài đặt không hợp lệ
	5b.	Hệ thống	Quay lại bước 3 của luồng chính
Hậu điều kiện	Không		

3.1.3 Yêu cầu phi chức năng

3.1.3.1 Yêu cầu chung

Ứng dụng có cơ chế phân quyền người dùng theo từng chức năng, vai trò đảm bảo từng người dùng chỉ được phép sử dụng các tính năng theo quyền hạn của mình. Ứng dụng cần hoạt động ổn định trên các nền tảng, trình duyệt khác nhau như Google Chrome, Microsoft Edge, Firefox v.v.

3.1.3.2 Yêu cầu về giao diện người dùng

Ứng dụng luôn đặt yếu tố dễ dùng và có tính tương tác cao, đơn giản hóa các thao tác và gợi ý cho người dùng về các thành phần chức năng trên giao diện bằng cách sử dụng các icon, button v.v phổ biến. Giao diện của ứng dụng cũng cần được xây dựng sao cho

phù hợp với các hệ thống hiện tại của nhà trường.

3.1.3.3 Yêu cầu về bảo mật

Các tiêu chuẩn về bảo mật thông tin được sử dụng: (i) các dữ liệu quan trọng đều được mã hóa trước khi truyền trên internet, (ii) xác thực người dùng khi truy cập vào các tài nguyên quan trọng của hệ thống.

3.1.3.4 Yêu cầu khác

Ngoài ra ứng dụng cần đảm bảo (i) tính khả thi - ứng dụng được xây dựng đáp ứng được các yêu cầu thực tế của người dùng, (ii) tính linh động - ứng dụng cần được xây dựng để có thể dễ dàng triển khai, bảo trì, sửa đổi và nâng cấp cho phù hợp với môi trường hoạt động và các yêu cầu mới.

3.2. Thiết kế hệ thống

3.2.1 Thiết kế kiến trúc

Hệ thống được xây dựng theo kiến trúc client-server. Trong mô hình này, server cung cấp API cho phép client thực hiện request thông qua các API đó. Khi nhận được yêu cầu từ client, server thực hiện xác thực thông tin người dùng, xử lý dữ liệu, lưu trữ, cập nhật và trả về thông tin cho client. Còn client đóng vai trò cung cấp giao diện tương tác để thu thập thông tin và thực hiện gửi request đến server để yêu cầu xử lý thông tin.

Hình 3.5 mô tả kiến trúc tổng thể của hệ thống gọi ý đăng ký học tập. Hệ thống được chia làm hai phần: Frontend và Backend. Trong đó *frontend* là phần giao diện, chịu trách nhiệm hiển thị, tương tác và thực hiện các yêu cầu của người dùng, sau đó gửi các yêu cầu xử lý tới *backend* thông qua các API (trên giao thức HTTP) hoặc qua websocket, *backend* sẽ chịu trách nhiệm xử lý các yêu cầu, logic nghiệp vụ và trả về kết quả (response).

Backend được thiết kế có thể phục vụ các yêu cầu HTTP hoặc các kết nối websocket, các yêu cầu khi được gửi đến sẽ đi qua một thành phần gọi là Router (Http router hoặc websocket router) để có thể xác định được "nơi" sẽ xử lý yêu cầu này, là các thành phần ở Controller layer. Các yêu cầu này có thể sẽ phải đi qua thành phần Middleware để tiền xử lý, lọc các yêu cầu không hợp lệ hoặc điều chỉnh các kết quả trước khi trả về cho frontend.

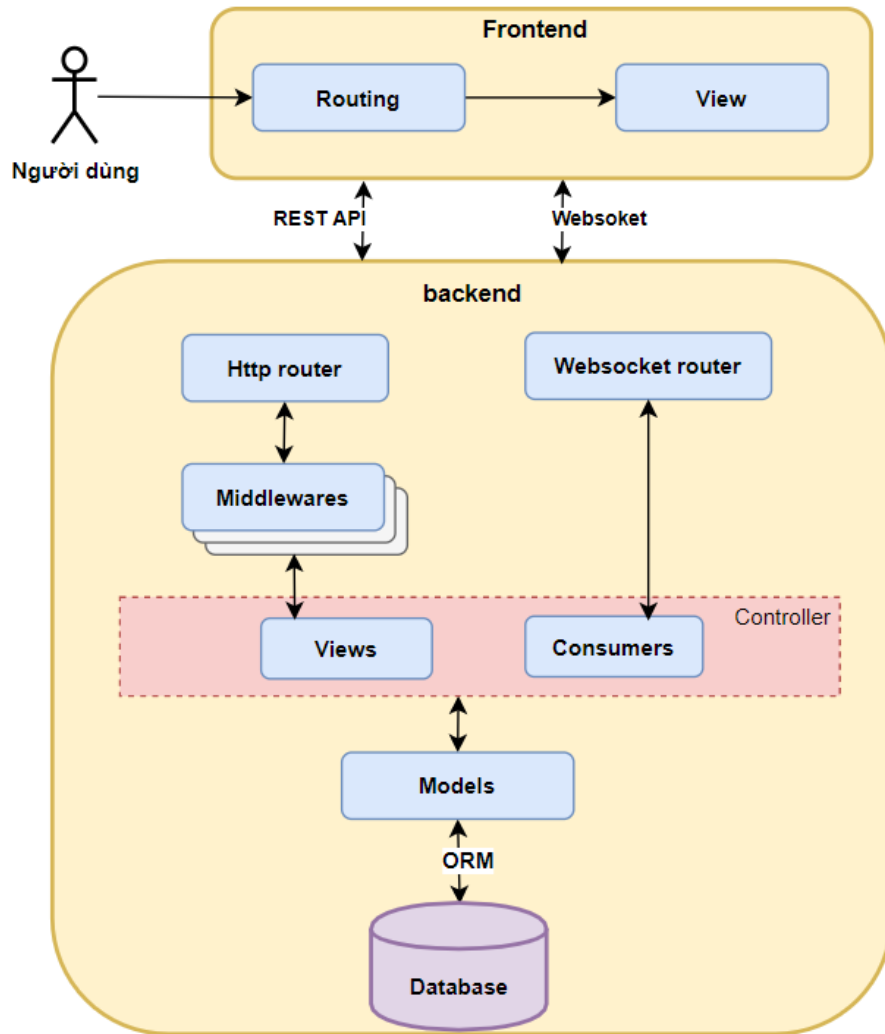
Việc sử dụng middleware sẽ giúp ta có thể tái sử dụng mã nguồn, hệ thống rõ ràng, tăng tính bảo mật và dễ bảo trì hơn. Lấy ví dụ ta có một số lượng các yêu cầu (request) gọi đến các hàm xử lý khác nhau của hệ thống và ta cần phải xác thực các yêu cầu này trước khi thực hiện các hàm đó thì thay vì phải viết mã kiểm tra các yếu tố xác thực tại đầu mỗi hàm xử lý, ta đưa đoạn mã này vào một middleware có chức năng kiểm tra các yếu tố xác thực, nếu hợp lệ ta thực hiện hàm tương ứng, nếu không sẽ trả về một thông báo lỗi. Khi đó yêu cầu trước khi đi đến hàm xử lý sẽ phải đi qua lớp middleware này, các yêu cầu không hợp lệ sẽ bị bỏ qua.

Thành phần xử lý (Controller) có thể sẽ phải truy cập đến dữ liệu được lưu trữ trong database và việc tương tác với cơ sở lưu trữ dữ liệu sẽ do thành phần Model đảm nhiệm. Đứng trung gian giữa Model và Database là ORM (Object Relational Mapping), đây là một kỹ thuật giúp ánh xạ các bản ghi trong cơ sở dữ liệu sang dạng đối tượng được định nghĩa trong model (Hình 3.6) giúp cho việc lập trình, bảo trì trở nên thuận lợi hơn.

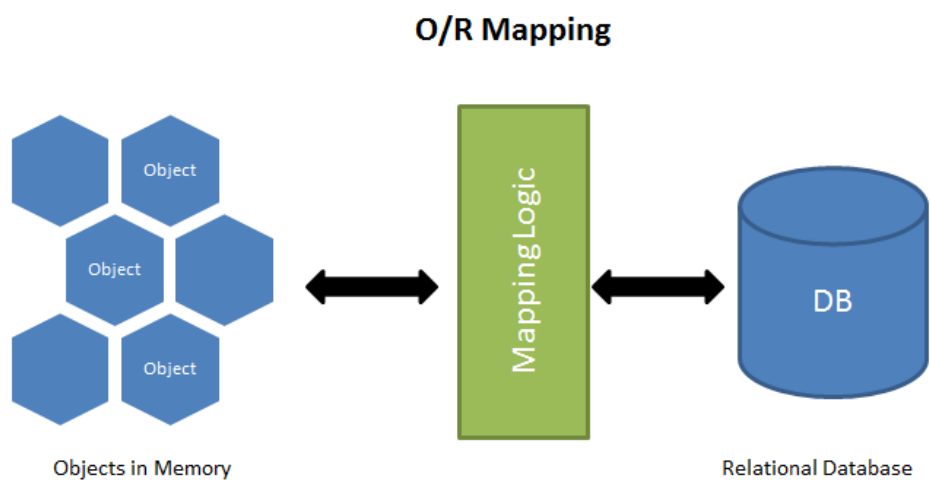
Trong Hình 3.5, thành phần View ở frontend và backend là khác nhau. Ở frontend, View chính là các thành phần giao diện sẽ hiển thị cho người dùng còn ở backend, View giữ vai trò xử lý các logic nghiệp vụ của hệ thống, đây là kiến trúc được sử dụng trong Django framework, mô hình MVT ¹.

Việc chia hệ thống thành hai phần cũng đem lại những lợi ích khi mỗi phần có thể được phát triển độc lập, có thể sử dụng nhiều công nghệ khác nhau mà không sợ ảnh hưởng đến thành phần kia, việc tối ưu, bảo trì cũng trở nên thuận tiện hơn.

¹MVT: Viết tắt của Model-View-Template là mô hình được sử dụng trong Django



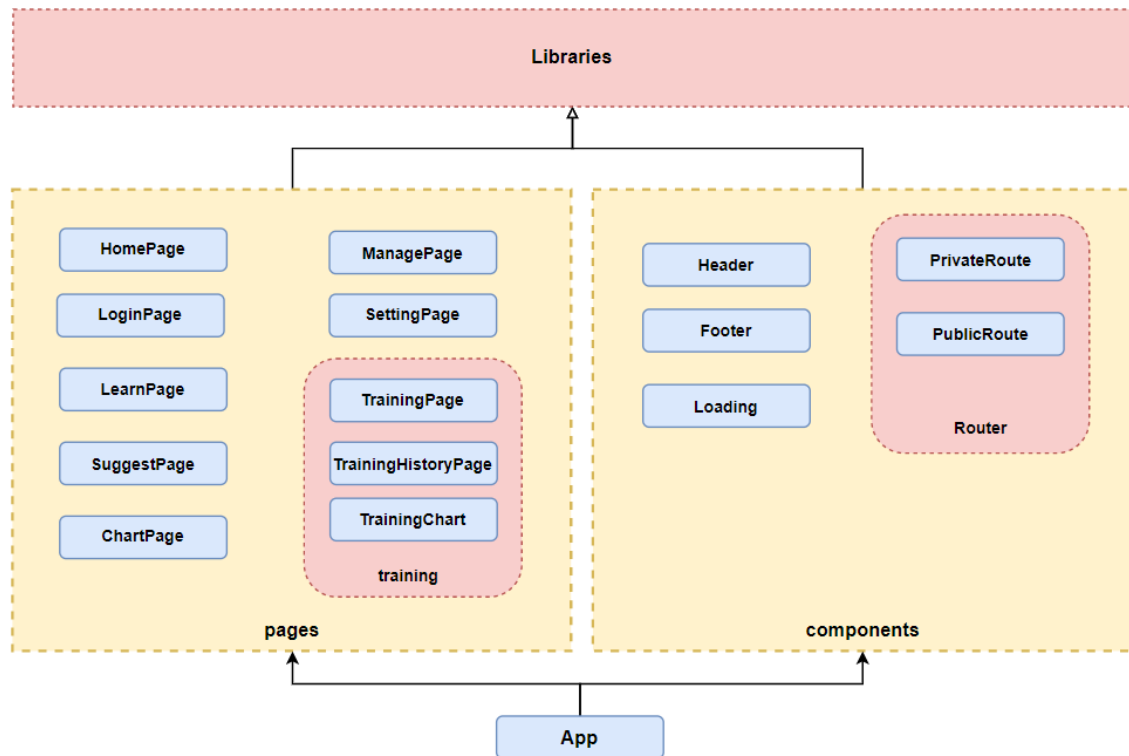
Hình 3.5: Thiết kế tổng thể của hệ thống gợi ý học tập



Hình 3.6: Chuyển đổi dữ liệu trong Object Relational Database

3.2.2 Thiết kế chi tiết frontend

3.2.2.1 Kiến trúc



Hình 3.7: Kiến trúc tổng thể frontend

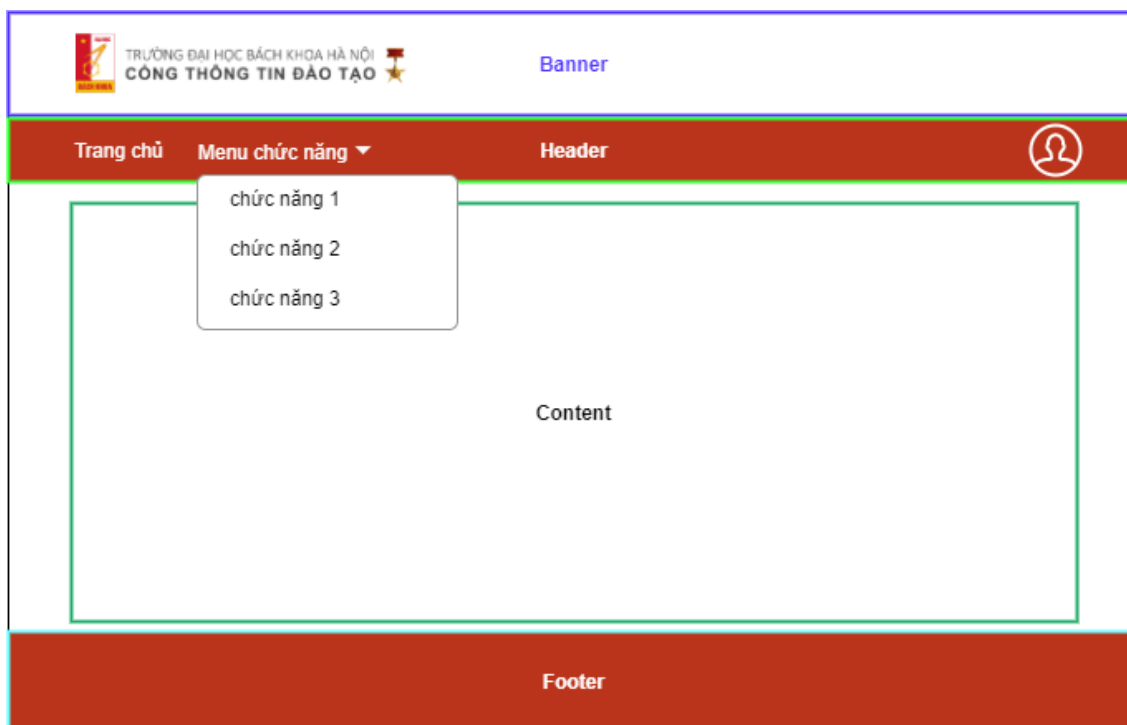
Thiết kế frontend được chia làm hai thành phần chính: pages và components. Phần components bao gồm các thành phần được xây dựng để có thể sử dụng lại nhiều lần, có thể được gọi bởi App hoặc các thành phần trong pages, bao gồm 4 thành phần: Header, Footer, Loading, Router. Trong đó Router gồm có PrivateRoute và PublicRoute, được sử dụng trong App nhằm mục đích điều hướng người dùng đến các thành phần giao diện có trong pages dựa vào đường dẫn trên trình duyệt. Để có thể truy cập vào các PrivateRoute, người dùng sẽ phải đảm bảo các yếu tố xác thực hoặc vai trò phù hợp.

Phần pages chịu trách nhiệm kết xuất ra các giao diện người dùng. Trong đó (i) HomePage kết xuất ra giao diện trang chủ của hệ thống, (ii) LoginPage kết xuất giao diện đăng nhập; các giao diện cho phân hệ sinh viên gồm: (iii) LearnPage kết xuất giao diện kết quả học tập, (iv) ChartPage kết xuất giao diện biểu đồ học tập, (v) SuggestPage kết xuất giao diện gợi ý học tập; các giao diện cho phân hệ quản lý: (vi) ManagePage kết xuất giao diện quản lý, (vii) SettingPage kết xuất giao diện cài đặt hệ thống, (viii) TrainingPage kết xuất giao diện huấn luyện và (ix) TrainingHistoryPage kết xuất giao diện lịch sử huấn luyện, thành phần TrainingChart được xây dựng để có thể được sử dụng trong nhóm training (bởi thành phần viii và ix).

Ngoài ra các thành phần trên cũng có thể phụ thuộc hoặc sử dụng các thư viện bên thứ ba (Libraries) để việc xây dựng trang web trở nên thuận lợi hơn, giao diện thân thiện, dễ sử dụng hơn.

3.2.2.2 Thiết kế mockup

Giao diện của hệ thống được thiết kế theo các component. Các trang đều có chung một bố cục như Hình 3.8 bao gồm 4 thành phần **Banner**, **Header**, **Content**, **Footer**. Với mỗi trang nội dung phần Content sẽ được hiển thị khác nhau, các thành phần ở Header sẽ cho phép người dùng điều hướng đến các trang khác nhau, riêng với phần *Menu chức năng* ở Header sẽ hiển thị các chức năng tương ứng với các tác nhân trong hệ thống.



Hình 3.8: Mockup layout của hệ thống gọi ý đăng ký học tập

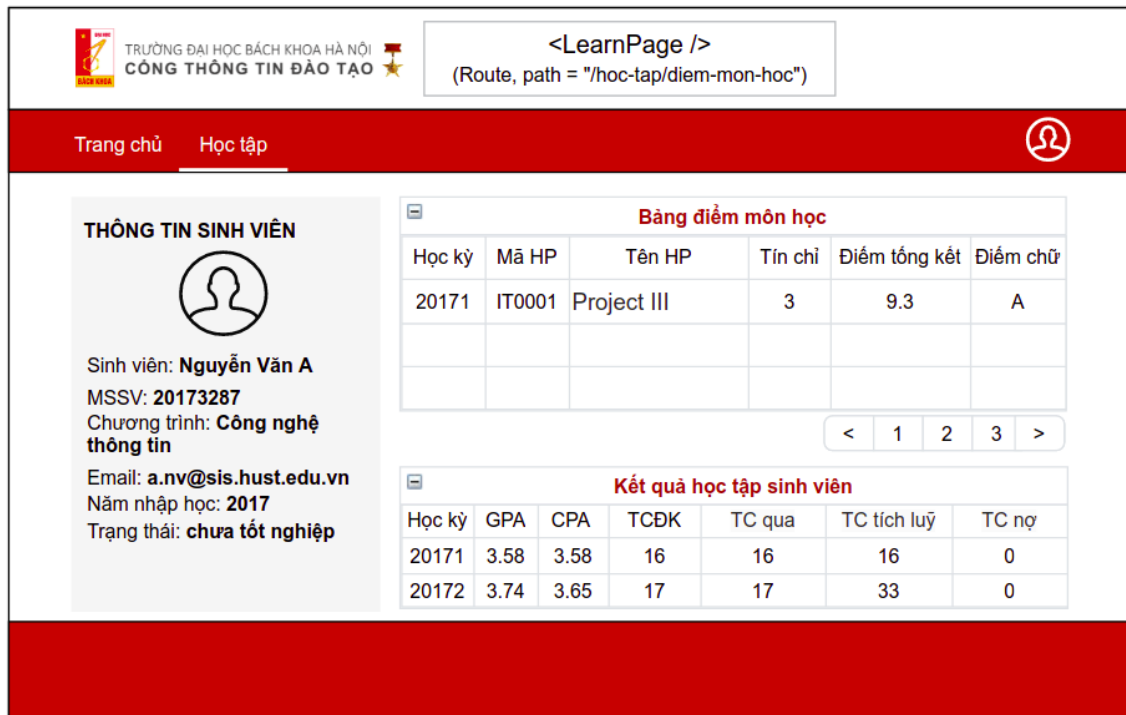
Màu sắc, font chữ, các button, thông báo đều được thiết kế đồng nhất, đảm bảo tính nhất quán cho toàn bộ hệ thống. Chi tiết các thuộc tính được trình bày trong bảng 7.7:

Thuộc tính	Giá trị
Màu sắc	Màu nền header, footer, button: #9a2b17 Màu chữ: #000000, #ffffff Màu khác: #1890ff, #be4a35
Font chữ	Segoe UI, Arial, sans-serif
Vị trí hiển thị popup dữ liệu	Chính giữa màn hình
Vị trí hiển thị thông báo	Góc trên bên phải màn hình

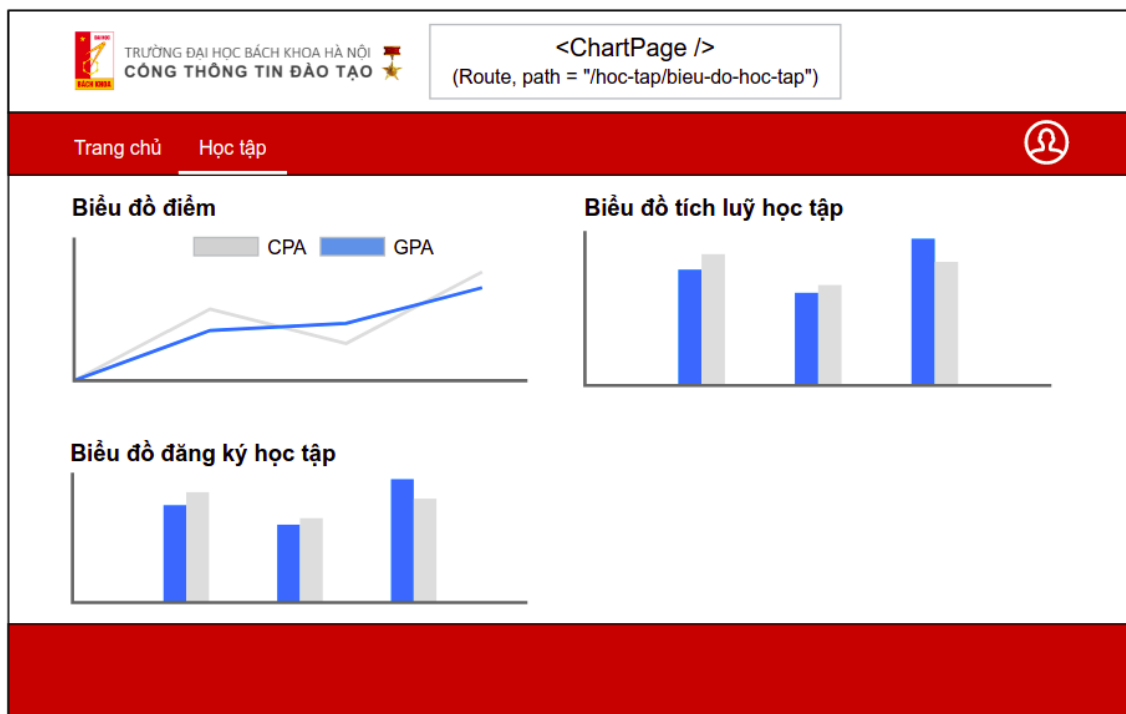
3.2.2.3 Thiết kế giao diện

Ở phần này, em sẽ trình bày giao diện thiết kế của một số màn hình chính cho hai tác nhân sinh viên và quản lý. Với sinh viên sẽ có các màn hình: Điểm môn học, Biểu đồ học

tập, Gợi ý học tập, với quản lý sẽ là màn hình Huấn luyện, Lịch sử huấn luyện, Chi tiết huấn luyện.



Hình 3.9: Thiết kế màn hình điểm môn học



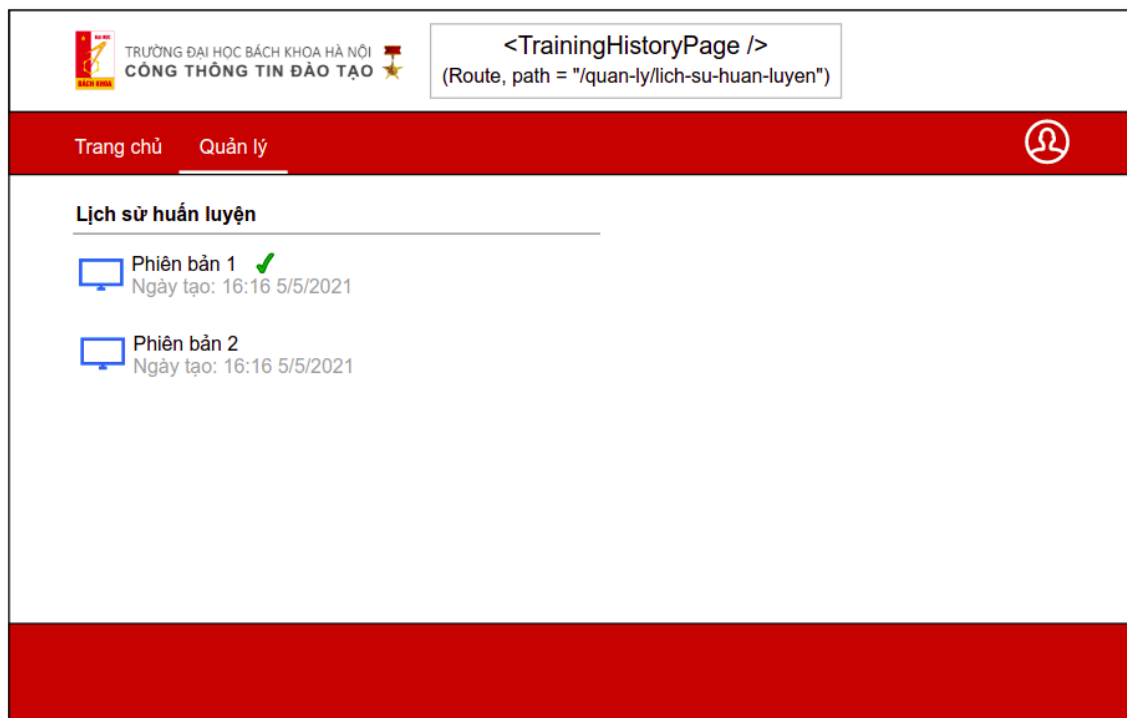
Hình 3.10: Thiết kế màn hình biểu đồ học tập



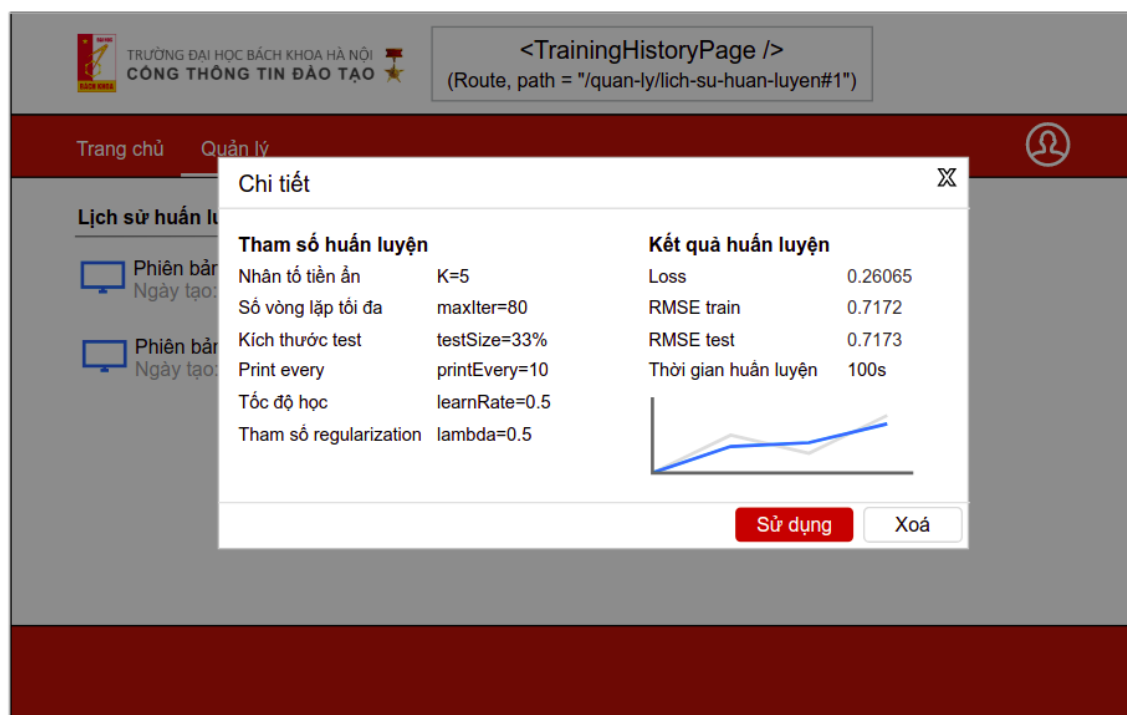
Hình 3.11: Thiết kế màn hình gợi ý học tập



Hình 3.12: Thiết kế màn hình huấn luyện mô hình



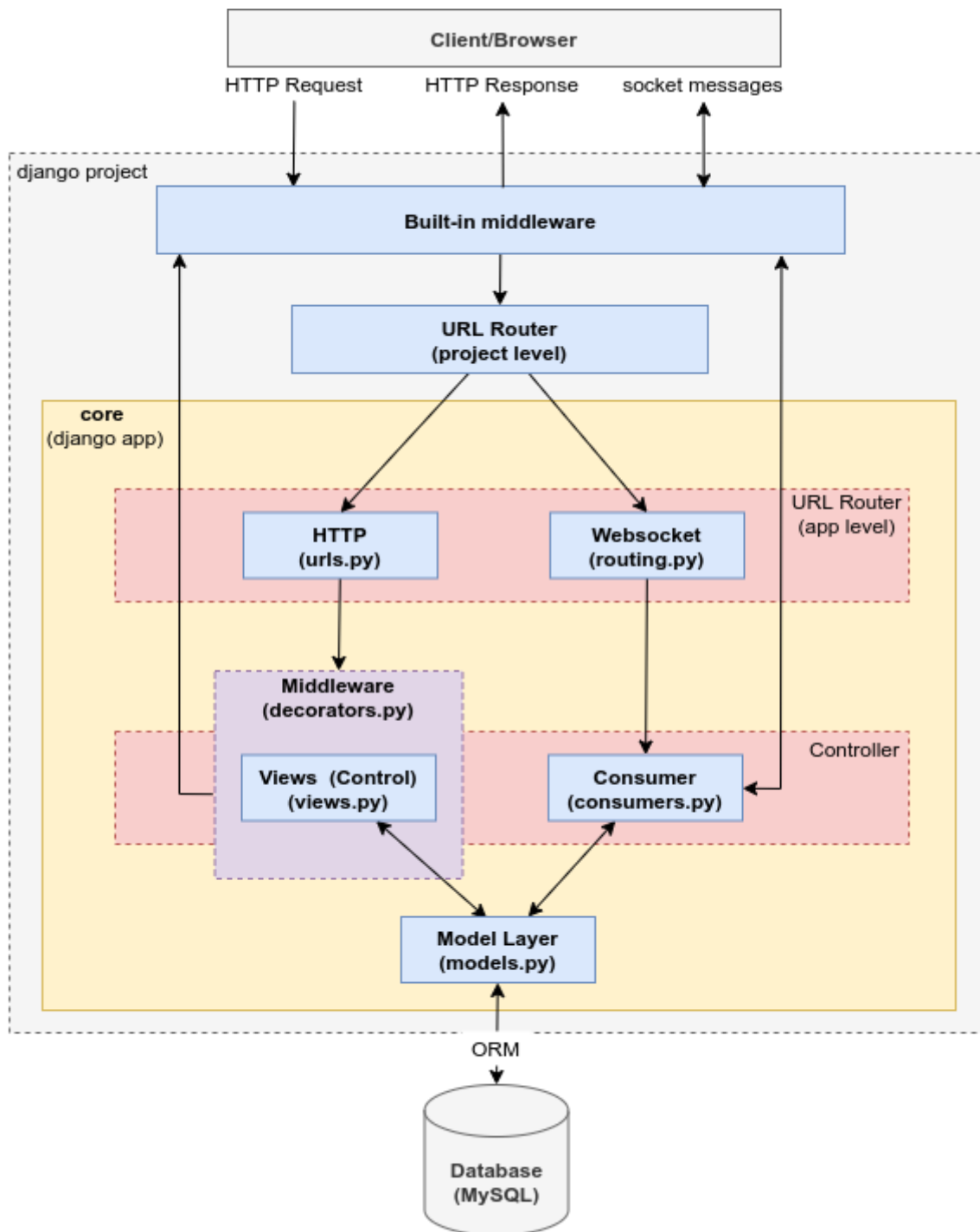
Hình 3.13: Thiết kế màn hình lịch sử huấn luyện



Hình 3.14: Thiết kế màn hình chi tiết huấn luyện

3.2.3 Thiết kế chi tiết backend

3.2.3.1 Kiến trúc



Hình 3.15: Thành phần kiến trúc chính và luồng hoạt động của backend

Với Django framework, backend sẽ được xây dựng thành một *django project*, trong đó sẽ bao gồm nhiều *django app* cùng các thành phần cấu hình, cài đặt ở mức project. Mỗi *django app* sẽ được xây dựng theo mô hình MVT với ba thành phần Model, View, Template. Mô hình này tương tự với mô hình MVC² với Model chịu trách nhiệm tương

²Viết tắt của Model-View-Controller

tác với cơ sở dữ liệu; Template chịu trách nhiệm hiển thị dữ liệu cho người dùng thông qua các thành phần giao diện; View là nơi chứa các xử lý logic, nghiệp vụ của hệ thống, là cầu nối giữa Model và Template.

Hình 3.15 mô tả các thành phần kiến trúc và luồng hoạt động của backend trong hệ thống gợi ý đăng ký học tập. Hệ thống được xây dựng với một *django app* là *core*, trong đó thành phần Model được triển khai trong module *models.py*, thành phần View được triển khai trong module *views.py*. Do backend được xây dựng để cung cấp các API nên thành phần Template trong các *django app* là không có, thay vào đó có thể coi thành phần này được đặt ở phía frontend bởi nó nhận dữ liệu từ API và hiển thị ra giao diện người dùng. Với các kết nối websocket, các yêu cầu sẽ được xử lý bởi thành phần Consumer được triển khai trong module *consumers.py*. Backend sẽ mở ra các kết nối tới các đường dẫn được định nghĩa trong URL Router ở cả mức project và app. Nhiệm vụ của các thành phần Router này là điều hướng request tới các thành phần xử lý tương ứng ở Controller layer (tương ứng với thành phần View hoặc Consumer). Ở mức app, cũng có thể xây dựng các middleware xử lý riêng cho nhóm các View, làm tăng tính bảo mật cho hệ thống. Các middleware này được triển khai trong module *decorators.py*.

Về luồng hoạt động chung, các yêu cầu khi được gửi tới backend sau khi đi qua các thành phần middleware ở mức project sẽ được URL Router điều hướng tới các thành phần xử lý tương ứng, có thể là View hoặc Consumer. Các thành phần này có thể cần tương tác với Model để lấy dữ liệu từ database, thực hiện các logic nghiệp vụ và trả về dữ liệu cho client. Với websocket, sau khi đã thiết lập được kết nối, phía client và server có thể gửi các messages qua lại với nhau.

3.2.3.2 Thiết kế API

Như đã trình bày trong phần 3.2.1, hai thành phần frontend và backend của hệ thống sẽ giao tiếp qua các API và kết nối websocket. Bảng 3.21 mô tả danh sách các API của hệ thống hoạt động trên giao thức HTTP, đường dẫn đầy đủ tới mỗi api là sự kết hợp giữa host, port của server và địa chỉ các api **<protocol>://host:port/<địa_chi_api>** (protocol có thể là http hoặc https).

STT	Mục đích	Phương thức	Địa chỉ
1	Đăng nhập hệ thống	POST	/api/login
2	Đăng xuất hệ thống	POST	/api/logout
3	Kiểm tra token có hợp lệ?	GET	/api/check-token
4	Lấy dữ liệu học tập của sinh viên	GET	/api/user-learn-data
5	Lấy dữ liệu chương trình học của sinh viên	GET	/api/program

6	Lấy dữ liệu tổng kết học tập GPA, CPA của sinh viên	GET	/api/get-gpa-cpa
7	Lấy dữ liệu gợi ý môn học	GET	/api/suggest
8	Lấy dữ liệu lịch sử huấn luyện	GET	/api/train-version/all
9	Xoá phiên bản huấn luyện	DELETE	/api/train-version/delete/:id
10	Lấy dữ liệu cài đặt hệ thống	GET	/api/setting/get
11	Cập nhật cài đặt hệ thống	PUT	/api/setting/update
12	Lấy dữ liệu thống kê	GET	/api/get-statistic

Bảng 3.21: Danh sách API của hệ thống gợi ý đăng ký học tập

Với websocket frontend sẽ thiết lập kết nối tới backend thông qua địa chỉ:

<protocol>://host:port/ws/training/ (protocol có thể là ws hoặc wss).

Kết nối này được sử dụng cho chức năng huấn luyện mô hình, hai bên sẽ trao đổi các dữ liệu định dạng json và được chuyển sang dạng string khi gửi đi. Định dạng dữ liệu gửi từ frontend được cho trong Bảng 3.22. Dựa vào trường *action* backend sẽ biết cần phải xử lý gì, trường *data* là các dữ liệu liên quan tới *action* đó.

Trường dữ liệu	Kiểu dữ liệu
action	string
data	json

Bảng 3.22: Định dạng dữ liệu gửi từ frontend

Action	Mô tả
train	Thực hiện huấn luyện mô hình với tham số cho trong trường data
save	Lưu lại kết quả từ bước train

Bảng 3.23: Mô tả các action

3.2.3.3 Luồng hoạt động

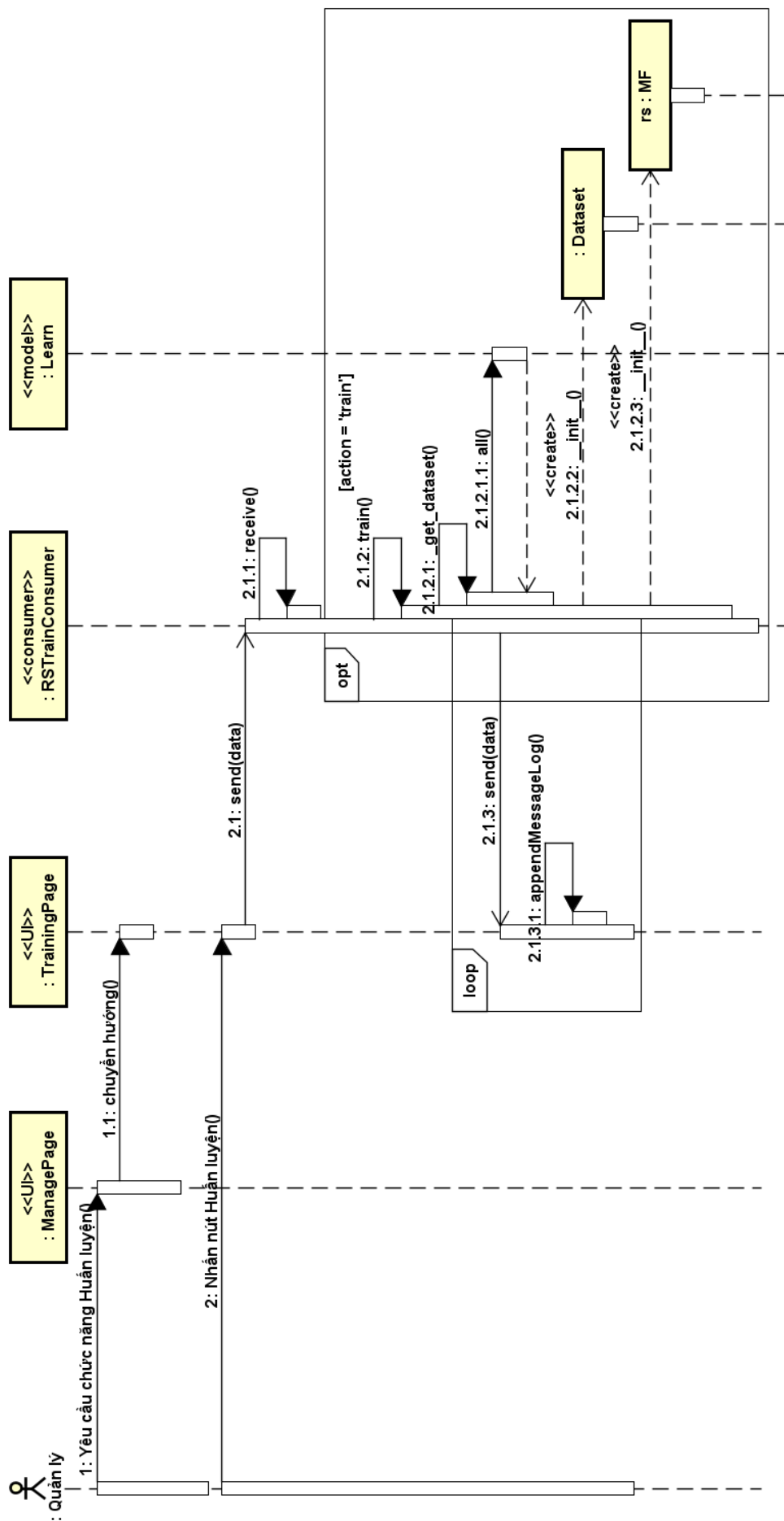
Phần 3.2.3.1 đã mô tả về luồng hoạt động chung của hệ thống backend, ở phần này sẽ đi trình bày cụ thể hơn về luồng hoạt động của một số chức năng chính thông qua các biểu

đồ trình tự, đó là chức năng Huấn luyện mô hình, Lưu kết quả huấn luyện và Xem gợi ý học tập.

Luồng hoạt động ca sử dụng Huấn luyện mô hình

Hình 3.16 mô tả quá trình thực hiện ca sử dụng Huấn luyện mô hình với điều kiện trước đó hệ thống (frontend) phải tạo thành công kết nối socket tới server, kết nối này sẽ được sử dụng để thực hiện chức năng huấn luyện. Chi tiết như sau:

1. Ca sử dụng bắt đầu khi người quản lý chuyển hướng đến giao diện huấn luyện và nhấn nút Huấn luyện.
2. Frontend thông qua kết nối socket đã thiết lập trước đó gửi dữ liệu với định dạng được cho trong Bảng 3.22, trong đó trường **action** là *train*. Dữ liệu được gửi tới consumer ở phía server backend (đối tượng RSTrainConsumer).
3. Đối tượng consumer nhận dữ liệu thông qua phương thức *receive*, dựa vào trường **action** rồi gọi tới phương thức *train*. Phương thức này thực hiện chuẩn bị các dữ liệu điểm số được lấy thông qua model Learn (thông qua phương thức *_get_dataset*), thực hiện chuẩn hóa, chuyển đổi dữ liệu thành đối tượng Dataset và cùng với các tham số cài đặt nhận từ frontend để tạo đối tượng MF và bắt đầu thực hiện giải thuật để tìm mô hình phù hợp. Quá trình thực hiện ở consumer sẽ liên tục được cập nhật tới frontend thông qua phương thức *send* gửi dữ liệu qua socket.
4. Frontend khi nhận được dữ liệu từ server sẽ thực hiện hiển thị ra cửa sổ logs thông qua phương thức *appendMessageLog*. Sau khi huấn luyện kết thúc frontend hiển thị giao diện kết quả.



Hình 3.16: Biểu đồ trình tự ca sử dụng Huấn luyện mô hình

Luồng hoạt động ca sử dụng Lưu kết quả huấn luyện

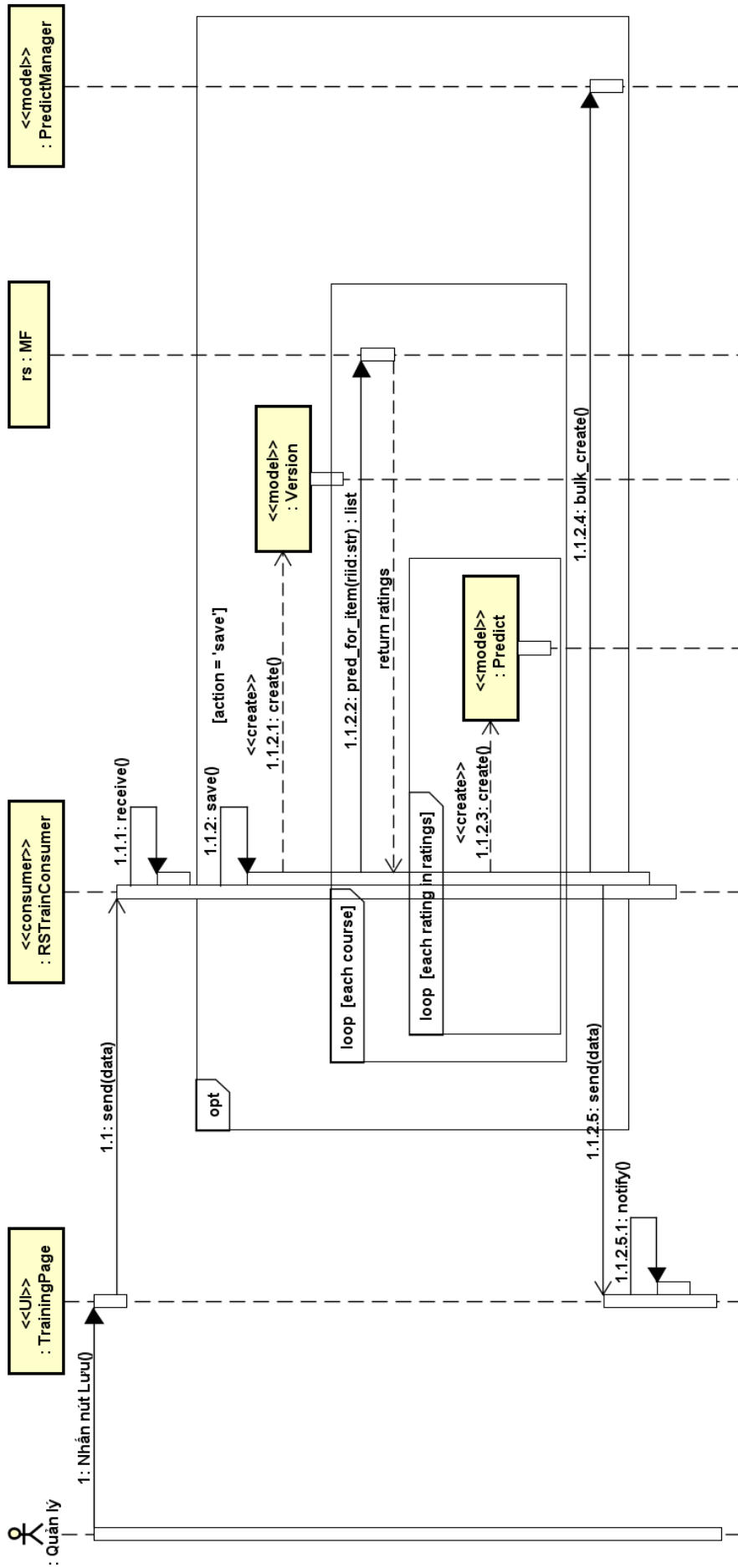
Sau khi luồng hoạt động ca sử dụng Huấn luyện mô hình kết thúc, frontend sẽ hiển thị giao diện kết quả kèm theo chức năng lưu lại kết quả. **Hình 3.17** mô tả chi tiết luồng hoạt động của ca sử dụng Lưu kết quả huấn luyện.

1. Người quản lý nhấn nút Lưu trên giao diện
2. Frontend gửi dữ liệu thông qua kết nối socket với trường **action** là *save*.
3. Đối tượng consumer nhận dữ liệu thông qua phương thức *receive*, dựa vào trường **action** rồi gọi tới phương thức *save*. Phương thức này thực hiện tạo một phiên bản huấn luyện mới (đối tượng Version), tính toán các kết quả dự đoán cho mỗi môn học (course) và với mỗi kết quả này tương ứng với một đối tượng Predict và được lưu lại trong cơ sở dữ liệu. Sau khi lưu xong, server gửi message thành công về cho frontend.
4. Frontend nhận dữ liệu từ server thông qua socket và thông báo lưu thành công.

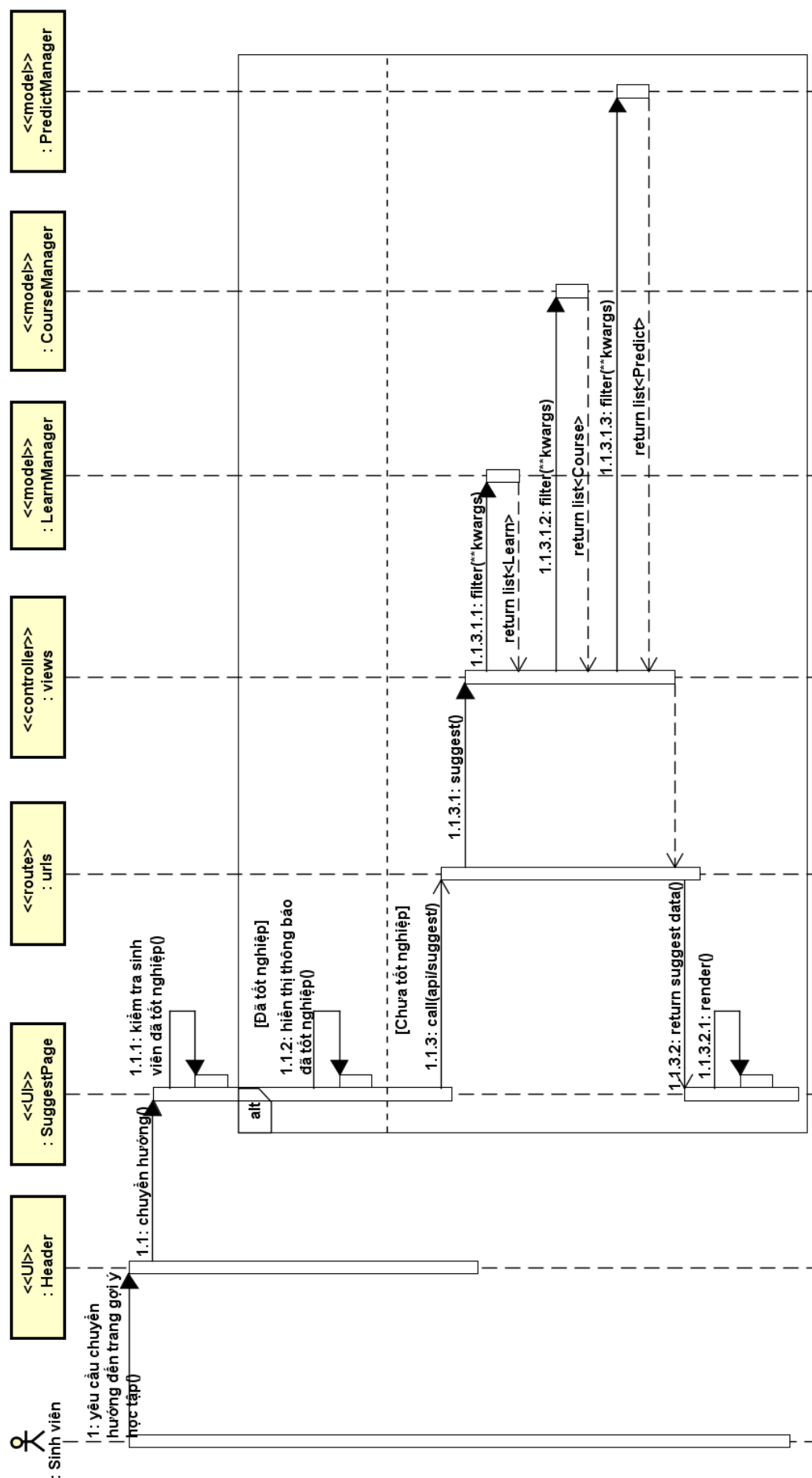
Luồng hoạt động ca sử dụng Xem gợi ý học tập

Hình 3.18 mô tả luồng hoạt động của ca sử dụng Xem gợi ý học tập. Khi tác nhân sinh viên chuyển hướng đến trang gợi ý học tập (SuggestPage), trước tiên hệ thống sẽ thực hiện kiểm tra trạng thái học tập của sinh viên là đã tốt nghiệp hay chưa, các kết quả gợi ý sẽ chỉ hiển thị khi sinh viên chưa tốt nghiệp.

Nếu sinh viên có trạng thái chưa tốt nghiệp hệ thống (frontend) sẽ thực hiện gọi đến api gợi ý học tập để lấy dữ liệu gợi ý và hiển thị ra giao diện. Dữ liệu gợi ý học tập bao gồm các môn học chưa qua, các môn học bắt buộc trong chương trình học và các môn học tự do được sắp xếp dựa trên các kết quả dự đoán (Predict) đã lưu trước đó.



Hình 3.17: Biểu đồ trình tự ca sử dụng Lưu kết quả huấn luyện

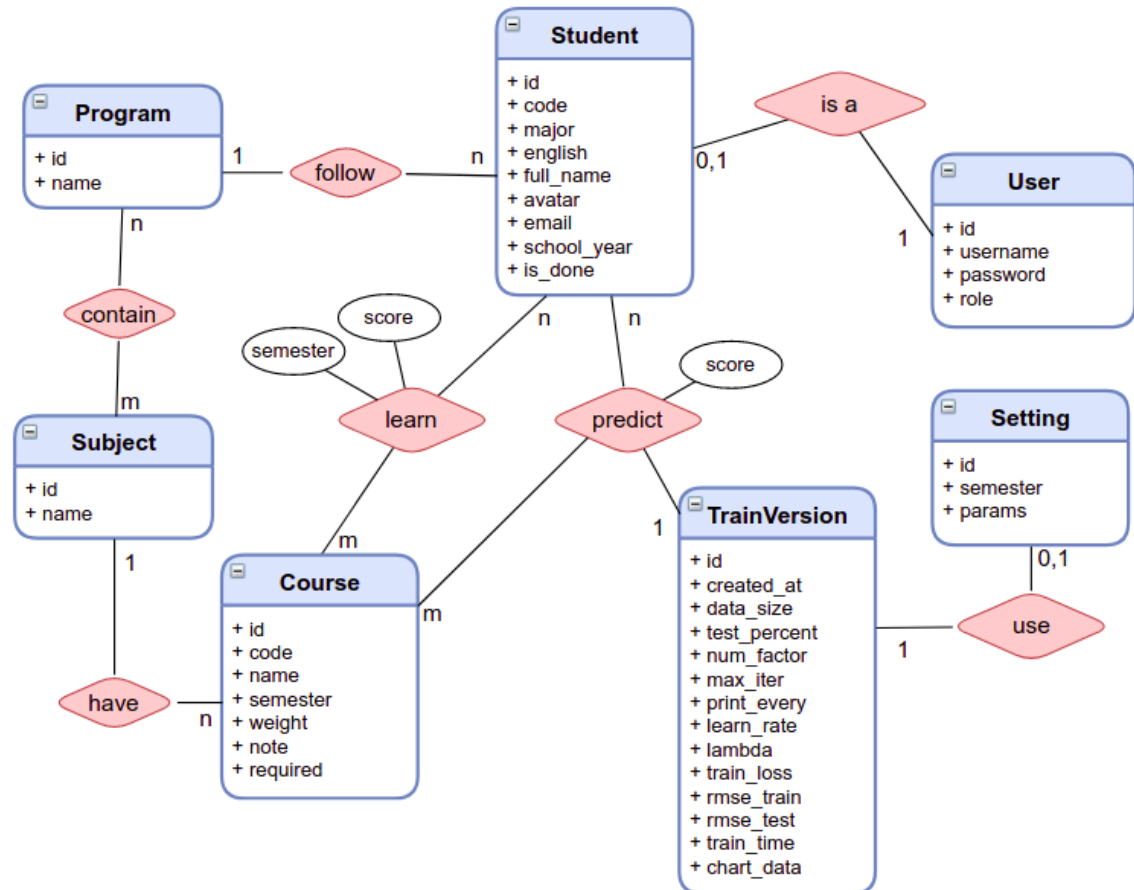


Hình 3.18: Biểu đồ trình tự ca sử dụng Xem gợi ý học tập

3.2.4 Thiết kế mô hình dữ liệu

3.2.4.1 Mô hình dữ liệu mức khái niệm

Hình 3.19 là sơ đồ thực thể - liên kết của hệ thống gợi ý đăng ký học tập với các thực thể và thuộc tính của nó được thể hiện dưới dạng bảng, hình thoi thể hiện mối quan hệ giữa các thực thể với nhau.



Hình 3.19: Sơ đồ thực thể liên kết cho hệ thống gợi ý đăng ký học tập

Hệ thống gồm có 7 thực thể: Program (chương trình học), Subject (loại học phần), Course (học phần), Student (sinh viên), TrainVersion (phiên bản huấn luyện), Setting và User. Trong đó Program sẽ được thiết kế gồm nhiều Subject khác nhau tùy vào các chuyên ngành của trường, mỗi Subject cũng có thể thuộc nhiều Program khác nhau và bao gồm nhiều Course trong đó. Mỗi Student sẽ đi theo một Program riêng tùy thuộc vào chuyên ngành của sinh viên đó. Student có thể học nhiều Course khác nhau và ngược lại mỗi Course có thể được học bởi nhiều Student. Mỗi User trong hệ thống có thể là một sinh viên (Student) hoặc quản lý. Với mỗi phiên bản huấn luyện (TrainVersion) sẽ cho ta các kết quả dự đoán (predict) học tập của sinh viên (Student) với các môn học tự do (Course). Thực thể Setting sẽ lưu lại các cài đặt hệ thống như kì học, tham số huấn luyện mặc định và TrainVersion đang sử dụng để gợi ý học tập.

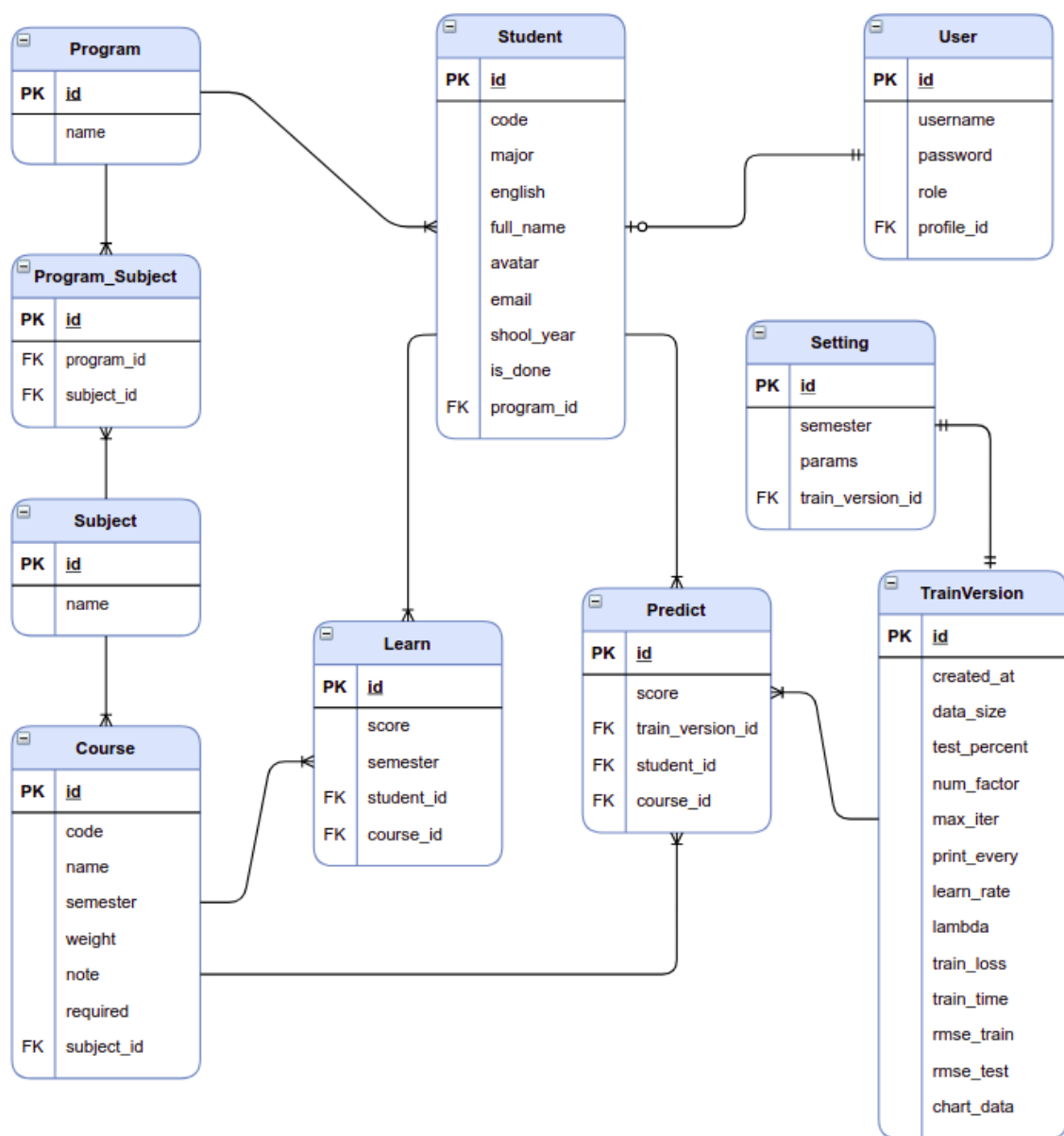
3.2.4.2 Mô hình dữ liệu mức logic

Sơ đồ tổng quan

Từ sơ đồ thực thể liên kết đã trình bày ở phần 3.2.4.1, ta thực hiện chuyển đổi sang mô hình dữ liệu mức logic và được thể hiện trong Hình 3.20.

Các thao tác chuyển đổi được thực hiện như sau:

- Mỗi thực thể và các thuộc tính của nó trong sơ đồ thực thể-liên kết sẽ tương ứng với một bảng trong cơ sở dữ liệu.
- Với quan hệ một-một ta thêm khoá chính của một bên vào bên còn lại làm thuộc tính khoá ngoại. Trong Hình 3.19, các cặp thực thể User-Student, Setting-TrainVersion có quan hệ một-một, ta thêm trường *profile_id* vào bảng User làm khoá ngoại tới bảng Student, thêm trường *train_version_id* vào bảng Setting làm khoá ngoại tới bảng TrainVersion.
- Với quan hệ một-nhiều ta thêm khoá chính của quan hệ bên một vào làm khoá ngoại của quan hệ bên nhiều. Các thực thể có quan hệ một-nhiều: Program-Student, Subject-Course, ta thêm khoá chính của bên một (Program, Subject) vào bên nhiều (Student, Course) làm khoá ngoại (*program_id*, *subject_id*).
- Với quan hệ nhiều-nhiều ta tạo một bảng mới xác định bởi khoá của các thực thể liên quan và các thuộc tính nằm trong quan hệ đó. Ví dụ quan hệ giữa Course-Student, ta tách ra một bảng mới Learn có hai thuộc tính khoá ngoại (*course_id*, *student_id*) chứa khoá của hai bảng Course và Student cùng các thuộc tính riêng *semester*, *score*. Tương tự với quan hệ Program-Subject, ta tách thành bảng Program_Subject như trong Hình 3.20.



Hình 3.20: Sơ đồ cơ sở dữ liệu

Thiết kế chi tiết các bảng

Bảng 3.24: Thiết kế chi tiết bảng Program

Tên bảng			Program		
STT	PK	FK	Tên cột	Kiểu dữ liệu	Mô tả
1	x		id	int	Định danh cho bản ghi
2			name	nvarchar	Tên của chương trình học

Bảng 3.25: Thiết kế chi tiết bảng Subject

Tên bảng			Subject		
STT	PK	FK	Tên cột	Kiểu dữ liệu	Mô tả
1	x		id	int	Định danh cho bản ghi
2			name	nvarchar	Tên của loại học phần

Bảng 3.26: Thiết kế chi tiết bảng Program_Subject

Tên bảng			Program_Subject		
STT	PK	FK	Tên cột	Kiểu dữ liệu	Mô tả
1	x		id	int	Định danh cho bản ghi
2		x	program_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng Program
3		x	subject_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng Subject

Bảng 3.27: Thiết kế chi tiết bảng Course

Tên bảng			Course		
STT	PK	FK	Tên cột	Kiểu dữ liệu	Mô tả
1	x		id	int	Định danh cho bản ghi
2			code	varchar	Mã học phần
3			name	nvarchar	Tên học phần
4			semester	int	Kì học
5			weight	int	Khối lượng học tập (số tín chỉ)
6			note	nvarchar	Chú thích
7			required	int	Môn học bắt buộc? (1: bắt buộc, 0: không bắt buộc)
8		x	subject_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng Subject

Bảng 3.28: Thiết kế chi tiết bảng Student

Tên bảng			Student		
STT	PK	FK	Tên cột	Kiểu dữ liệu	Mô tả

1	x		id	int	Định danh cho bản ghi
2			code	varchar	Mã số sinh viên
3			major	nvarchar	Chuyên ngành
4			english	nvarchar	Thông tin về điểm tiếng anh
5			full_name	nvarchar	Họ tên sinh viên
6			avatar	varchar	Đường dẫn đến ảnh đại diện
7			email	varchar	Địa chỉ email sinh viên
8			school_year	int	Năm nhập học
9			is_done	int	Sinh viên đã tốt nghiệp? (1: đã tốt nghiệp, 0: chưa tốt nghiệp)
10		x	program_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng Program

Bảng 3.29: Thiết kế chi tiết bảng Learn

Tên bảng			Learn		
STT	PK	FK	Tên cột	Kiểu dữ liệu	Mô tả
1	x		id	int	Định danh cho bản ghi
2			score	float	Điểm tổng kết học phần trên thang 10
3			semester	varchar	Kì đăng ký học
4		x	student_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng Student
5		x	course_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng Course

Bảng 3.30: Thiết kế chi tiết bảng Learn

Tên bảng			Learn		
STT	PK	FK	Tên cột	Kiểu dữ liệu	Mô tả
1	x		id	int	Định danh cho bản ghi
2			score	float	Điểm tổng kết học phần trên thang 10
3			semester	varchar	Kì đăng ký học

4		x	student_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng Student
5		x	course_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng Course

Bảng 3.31: Thiết kế chi tiết bảng User

Tên bảng			User		
STT	PK	FK	Tên cột	Kiểu dữ liệu	Mô tả
1	x		id	int	Định danh cho bản ghi
2			username	varchar	Tên tài khoản đăng nhập
3			password	varchar	Mã hash của mật khẩu
4			role	varchar	Vai trò người dùng: student hoặc manager
5		x	profile_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng Student (nếu role là student)

Bảng 3.32: Thiết kế chi tiết bảng TrainVersion

Tên bảng			TrainVersion		
STT	PK	FK	Tên cột	Kiểu dữ liệu	Mô tả
1	x		id	int	Định danh cho bản ghi
2			created_at	float	Thời điểm tạo phiên bản
3			data_size	int	Số lượng bản ghi huấn luyện
4			test_per-cent	float	Phần trăm dữ liệu dành cho test
5			num_factor	int	Số nhân tố tiềm ẩn
6			max_iter	int	Số vòng lặp tối đa (epochs)
7			print_every	int	Tính toán chất lượng mô hình sau mỗi print_every vòng lặp
8			learn_rate	float	Tốc độ học
9			lambda	float	Tham số regularization
10			train_loss	float	Giá trị mất mát của mô hình
11			train_time	float	Thời gian huấn luyện mô hình

12			rmse_train	float	Độ đo RMSE trên tập train
13			rmse_test	float	Độ đo RMSE trên tập test
14			chart_data	varchar	Dữ liệu về sự thay đổi giá trị hàm mất mát và độ đo RMSE

Bảng 3.33: Thiết kế chi tiết bảng Setting

Tên bảng			Setting		
STT	PK	FK	Tên cột	Kiểu dữ liệu	Mô tả
1	x		id	int	Định danh cho bản ghi
2			semester	varchar	Kỳ học hiện tại
3			params	varchar	Tham số huấn luyện mặc định (đã chuyển từ json về string)
4		x	train_version_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng TrainVersion

Bảng 3.34: Thiết kế chi tiết bảng Predict

Tên bảng			Predict		
STT	PK	FK	Tên cột	Kiểu dữ liệu	Mô tả
1	x		id	int	Định danh cho bản ghi
2			score	float	Điểm gợi ý môn học
3		x	train_version_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng TrainVersion
4		x	student_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng Student
5		x	course_id	int	Khoá ngoại tham chiếu tới bản ghi thuộc bảng Course

Chương 4: Xây dựng chức năng gợi ý đăng ký học tập

Chương này sẽ trình bày chi tiết về quá trình xây dựng và triển khai chức năng gợi ý đăng ký học tập cho hệ thống, từ việc lựa chọn mô hình, xây dựng và phát triển giải thuật, đánh giá mô hình lựa chọn và tích hợp mô hình vào chức năng gợi ý đăng ký học tập của hệ thống. Tuy nhiên trước đó cần phải có được tập dữ liệu để huấn luyện, do dữ liệu về sinh viên và các kết quả học tập đều là những dữ liệu khá nhạy cảm nên trong đồ án này em sẽ tự sinh ra tập dữ liệu, quá trình xây dựng tập dữ liệu sẽ được trình bày chi tiết trong chương này.

4.1. Xây dựng tập dữ liệu

Dữ liệu được sinh sẽ bao gồm ba đối tượng chính: sinh viên, chương trình học, điểm số.

Với đối tượng sinh viên:

Định dạng dữ liệu cho đối tượng sinh viên gồm các trường:

Trường	Kiểu dữ liệu	Mô tả
id	string	mã số sinh viên
major	string	chuyên ngành
english	string	điểm tiếng anh
fullName	string	họ tên sinh viên
avatar	string	đường dẫn tới ảnh đại diện
email	string	địa chỉ email
program	string	chương trình đào tạo

Bảng 4.1: Định dạng dữ liệu cho đối tượng sinh viên

Từng trường sẽ được sinh một cách ngẫu nhiên, ví dụ về một bản ghi dưới định dạng json được mô tả trong Hình 4.1:

```
{
  "mssv": "20173287",
  "major": "Công nghệ thông tin",
  "english": "535",
  "fullName": "Nguyễn Bá Ngọc",
  "avatar": "",
  "email": "ngoc.nb173287@sis.hust.edu.vn",
  "program": "CNKT"
}
```

Hình 4.1: Ví dụ về dữ liệu sinh viên

Kết quả sinh dữ liệu sinh viên:

Năm nhập học	Số lượng sinh viên được tạo
2015	786
2016	828
2017	1054
2018	1035
2019	995
2020	1210
Tổng	5908

Bảng 4.2: Bảng kết quả sinh dữ liệu sinh viên

Với chương trình học:

Trong phạm vi đồ án, em sử dụng chương trình học cho cử nhân kỹ thuật (khóa 2017) ngành công nghệ thông tin của trường Đại học Bách khoa Hà Nội gồm 11 loại học phần với tổng cộng 58 học phần. Dữ liệu được lấy từ phần chương trình đào tạo trên trang web <https://ctt-sis.hust.edu.vn/> của sinh viên Nguyễn Bá Ngọc - 20173287 và đưa về định dạng json.

Mô tả các trường dữ liệu:

Trường	Kiểu dữ liệu	Mô tả
id	string	mã loại học phần
name	string	tên loại
courses	array	danh sách các học phần

Bảng 4.3: Mô tả các trường của đối tượng *loại học phần*

Trường	Kiểu dữ liệu	Mô tả
course_id	string	mã học phần
name	string	tên học phần
semester	integer	kỳ học
weight	integer	số tín chỉ
note	string	ghi chú
required	integer	môn học bắt buộc (0 hoặc 1)

Bảng 4.4: Mô tả các trường của đối tượng *học phần*

Ví dụ về loại học phần "230 - Giáo dục Quốc phòng - An ninh" định dạng json:

```
{
  "id": "230",
  "name": "Giáo dục Quốc phòng - An ninh",
  "courses": [
    {
      "course_id": "MIL1110",
      "name": "Đường lối quân sự",
      "semester": 1,
      "weight": 0,
      "note": "Giáo dục Quốc phòng - An ninh",
      "required": 1
    },
    {
      "course_id": "MIL1120",
      "name": "Công tác quốc phòng-An ninh",
      "semester": 2,
      "weight": 0,
      "note": "Giáo dục Quốc phòng - An ninh",
      "required": 1
    },
    {
      "course_id": "MIL1130",
      "name": "QS chung và KCT bắn súng AK",
      "semester": 3,
      "weight": 0,
      "note": "Giáo dục Quốc phòng - An ninh",
      "required": 1
    }
  ]
}
```

Hình 4.2: Dữ liệu loại học phần dưới định dạng json

Về điểm số

Đây là phần dữ liệu quan trọng được sử dụng trong mô hình gợi ý. Với mỗi học phần sẽ có một số lượng sinh viên tham gia học, và ta sẽ sinh điểm học phần này (trên thang 10) cho mỗi sinh viên theo phân phối chuẩn. Công cụ sử dụng là phần mềm Excel với chức năng Data Analysis.

Các bước thực hiện sinh điểm cho môn học A:

1. Xác định số sinh viên học môn học A (chỉ lấy mã số sinh viên) và vào đưa vào một cột trên phần mềm Excel.
2. Xác định giá trị trung bình (mean) và độ lệch chuẩn (standard deviation) cho phân phối sao cho phù hợp với môn học A
3. Sử dụng chức năng sinh các giá trị theo phân phối chuẩn của Excel: *Data > Data Analysis > Random Number Generation*. Nhập số lượng biến là 1 (vì ta chỉ cần sinh điểm), số lượng giá trị sinh bằng số lượng sinh viên, chọn *Distribution* là *Normal*, sau đó cung cấp 2 giá trị mean và standard deviation, chọn vị trí *output range* rồi nhấn OK.

Ví dụ, với học phần *Tư tưởng Hồ Chí Minh (SSH1050)* ta lựa chọn được 4678 sinh viên, ta đưa các mã số sinh viên vào một cột trong Excel. Sau đó thực hiện sinh dữ liệu theo các bước ở trên.

Random Number Generation

Number of Variables: 1

Number of Random Numbers: 4698

Distribution: Normal

Parameters

Mean = 5.2

Standard deviation = 1.3

Random Seed:

Output options

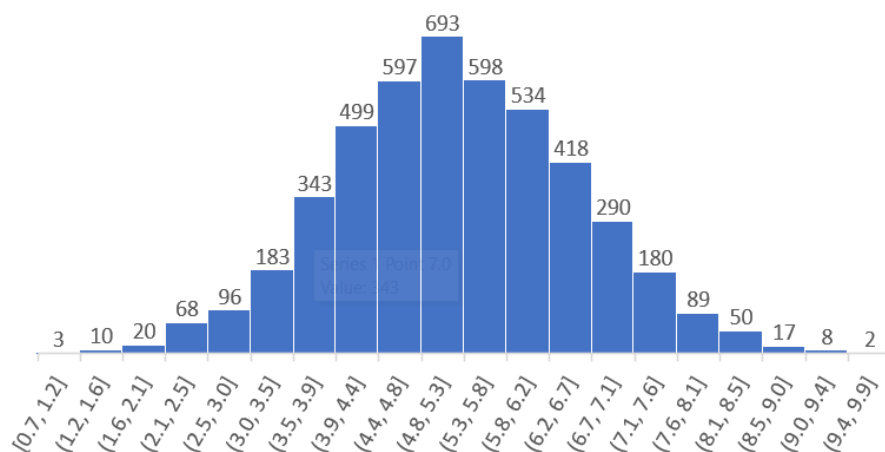
☒ Output Range: \$B\$2

☐ New Worksheet Ply:

☐ New Workbook

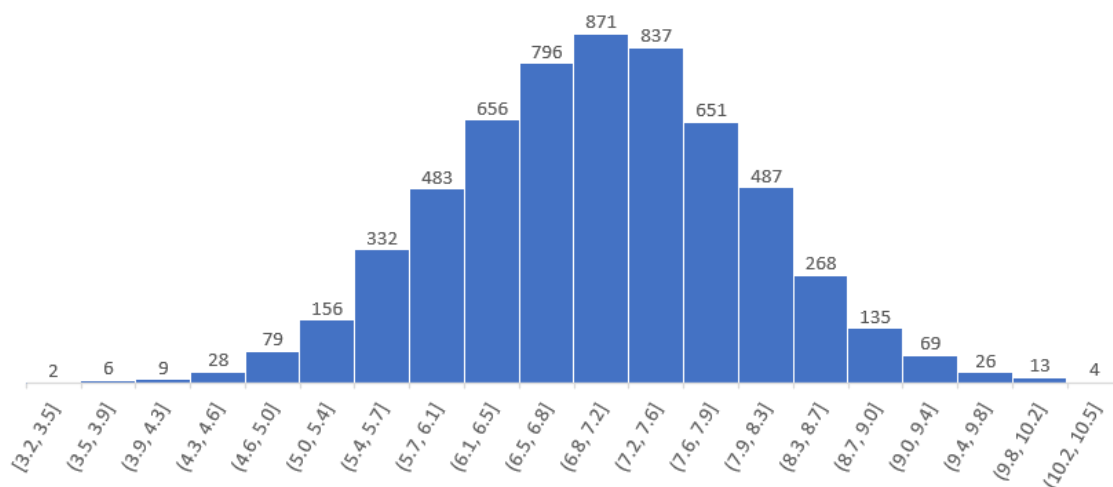
OK Cancel Help

Hình 4.3: Các giá trị cài đặt cho hàm sinh dữ liệu

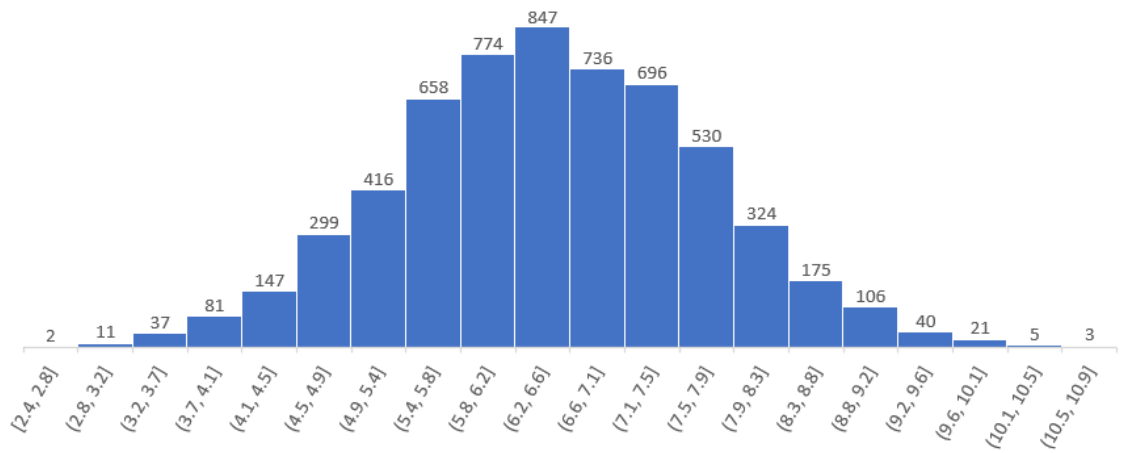


Hình 4.4: Biểu đồ cột phân bố các giá trị điểm cho học phần *Tư tưởng Hồ Chí Minh*

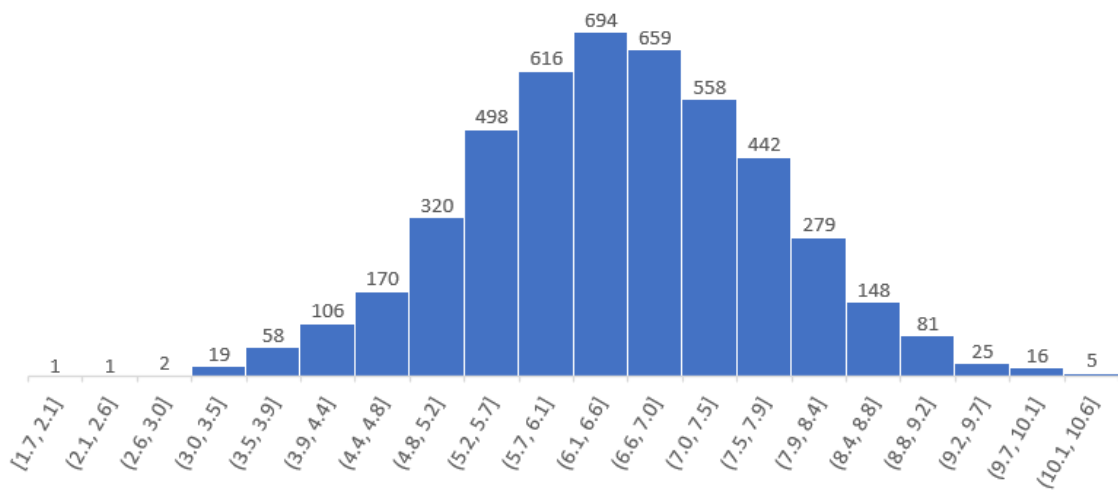
Trong ví dụ trên, hàm phân phối chuẩn nhận giá trị trung bình là 5.4 với độ lệch chuẩn 1.3 (hình 4.3), các giá trị được sinh ra cũng tập trung quanh giá trị 5.4 như hình 4.4. Các học phần khác cũng được sinh một cách tương tự.



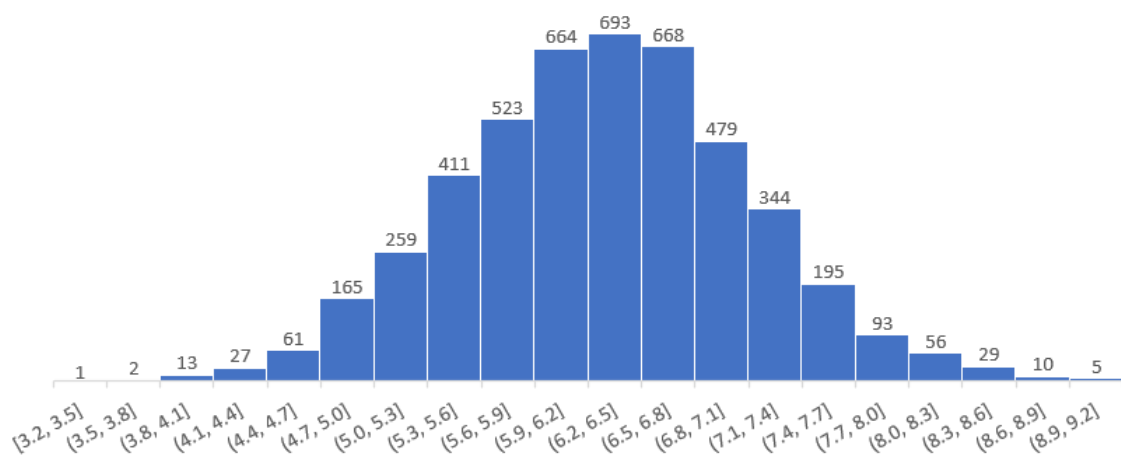
(a) Học phần Giải tích I, số lượng: 5908, $mean = 7$, $\sigma = 1$



(b) Học phần Đại số, số lượng: 5908, $mean = 6.5$, $\sigma = 1.2$



(c) Học phần Giải tích II, số lượng: 4698, $mean = 6.5$, $\sigma = 1.2$



(d) Học phần Vật lý đại cương I, số lượng: 4698, $mean = 6.3$, $\sigma = 0.8$

Hình 4.5: Biểu đồ phân bố giá trị của một số học phần khác

Sau khi sinh xong, dữ liệu điểm này sẽ được lưu lại trong một file định dạng .csv gồm 3 cột theo thứ tự: mã học phần, mã số sinh viên, điểm môn học. Trước khi lưu ta cần chuẩn hóa lại điểm của mỗi sinh viên trên thang 10 bởi trong một số học phần xuất hiện một vài giá trị lớn hơn 10 hoặc nhỏ hơn 0, với những giá trị này ta chuẩn hóa như sau: những giá trị lớn hơn 10 làm tròn xuống 10, còn những giá trị nhỏ hơn 0 thì đặt bằng 0.

Kết quả tổng hợp dữ liệu đã sinh được mô tả trong bảng 4.5:

Đối tượng	Số lượng dữ liệu
Sinh viên	5908
Môn học	58
Điểm	211843

Bảng 4.5: Bảng tổng hợp số lượng dữ liệu đã tạo cho từng đối tượng

4.2. Xây dựng, đánh giá mô hình huấn luyện

4.2.1 Xây dựng mô hình

Ở phần này, mục tiêu là đưa ra được một mô hình phù hợp với bài toán gợi ý môn học. Như đã trình bày trong Chương 2, các phương pháp lọc cộng tác dựa trên láng giềng có thời gian huấn luyện nhỏ bởi chỉ cần tính similarity matrix và lưu lại nhưng thời gian dự đoán lại khá lâu. Với các tập dữ liệu lớn việc lưu trữ similarity matrix là không khả thi, cho nên các phương pháp này chỉ phù hợp với các tập dữ liệu vừa và nhỏ. Với phương pháp lọc cộng tác dựa trên phân tích ma trận (Matrix Factorization), thời gian huấn luyện là khá lớn vì phải thực hiện lặp và tính toán qua rất nhiều mô hình để tìm ra được mô hình phù hợp, tuy nhiên thời gian dự đoán lại rất nhanh, hơn nữa việc tách ma trận thành hai ma trận có bậc nhỏ hơn giúp giảm không gian lưu trữ. Vì vậy, em quyết định lựa chọn mô hình gợi ý theo phương pháp *Matrix Factorization* kết hợp với *bias* bởi nó phù hợp với tập dữ liệu lớn và bộ nhớ để lưu trữ kết quả.

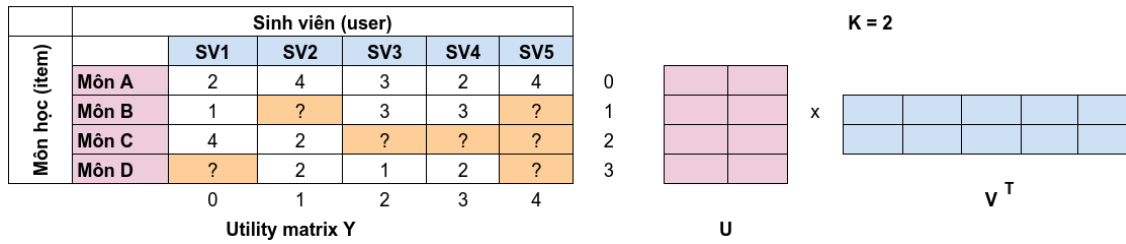
Đầu tiên, ta cần xác định được các đối tượng tương ứng với các thành phần của mô hình này: *users*, *items*, *ratings*. Trong hệ thống gợi ý môn học, tập *users* tương ứng với các dữ liệu về sinh viên, tập *items* tương ứng với các dữ liệu về môn học và cuối cùng các dữ liệu đánh giá của sinh viên đến môn học tương ứng với tập *ratings*. Tập dữ liệu đánh giá này có thể xác định được từ nhiều yếu tố, ví dụ như điểm số, lĩnh vực của môn học, định hướng, sở thích của sinh viên... Tuy nhiên, trong phạm vi đồ án này điểm số được lựa chọn làm tập *ratings*.

4.2.1.1 Xây dựng giải thuật

Sau khi đã xác định được các đối tượng của mô hình, ta xây dựng được Utility matrix Y là ma trận điểm số của sinh viên trên từng môn học. Theo như phương pháp MF, ta

phân rã Y thành hai ma trận U và V sao cho có thể xây dựng lại Y từ hai ma trận này.

$$Y \approx UV^T$$



Hình 4.6: Ví dụ phân rã ma trận trên dữ liệu sinh viên - môn học với k=2

Giai đoạn huấn luyện ta sẽ thực hiện tối ưu hai ma trận U và V. Phương pháp thực hiện là ban đầu ta khởi tạo giá trị trong hai ma trận này là ngẫu nhiên (thường theo phân phối chuẩn). Hàm mất mát đã được chỉ ra trong công thức 2.7:

$$loss(U, V, b, d) = \sum_{j=1}^n \sum_{i:r_{ij}=1} (u_i v_j^T + b_i + d_j + \mu - y_{ij})^2 + \lambda (\|u_i\|^2 + \|v_j\|^2 + b_i^2 + d_j^2)$$

Tiếp theo ta thực hiện tối ưu hàm mất mát này và cập nhật giá trị lại cho U, V và các giá trị bias b, d qua mỗi bước lặp theo phương pháp gradient descent (giải thuật Stochastic gradient descent [2]).

Các công thức cập nhật đã được chỉ ra ở 2.8, 2.9, 2.10, 2.11

$$e_{ij} = u_i v_j^T + b_i + d_j + \mu - y_{ij}$$

$$v_j = v_j - \beta(e_{ij} u_i + \lambda v_j)$$

$$d_j = d_j - \beta(e_{ij} + \lambda d_j)$$

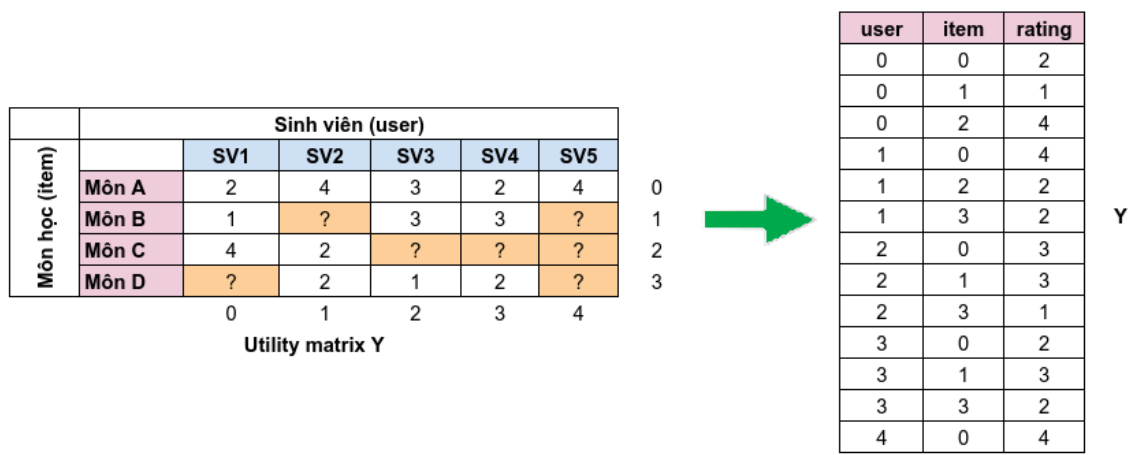
$$u_i = u_i - \beta(e_{ij} v_j + \lambda u_i)$$

$$b_i = b_i - \beta(e_{ij} + \lambda b_i)$$

Sau khi quá trình huấn luyện kết thúc ta thu được 2 ma trận U, V và các giá trị bias b, d đã được tối ưu. Các kết quả dự đoán được thực hiện theo công thức 2.12 và lưu lại trong cơ sở dữ liệu của hệ thống để thuận tiện cho việc tính toán và hiển thị các kết quả gợi ý cho sinh viên.

$$\hat{y}_{ij} = u_i v_j^T + b_i + d_j + \mu$$

Trong thực tế, ma trận Y thường thưa và việc lưu trữ Y như trong hình 4.6 là không tối ưu do có quá nhiều giá trị *ratings* chưa biết (dấu '?') và ta chỉ sử dụng các giá trị *ratings* đã biết trong giải thuật. Do đó ta sẽ tốt hơn nếu ta biểu diễn ma trận Y dưới dạng *sparse matrix*, tức chỉ lưu các giá trị đã biết và vị trí của chúng. Ma trận Y lúc này sẽ có 3 cột: *sinh viên*, *môn học*, *điểm số* (Hình 4.7)



Hình 4.7: Biểu diễn ma trận Y dưới dạng *sparse matrix*

Mã giả cho giải thuật SGD:

Algorithm 1: MF_fit ($Y, k, \beta, \lambda, \text{stopping condition}$)

$U, V, b, d \leftarrow \mathcal{N}(0, \sigma^2)$

$\mu \leftarrow \frac{\sum_{\text{all rating } r \text{ in } Y} r}{|Y|}$

while *stopping criterion is NOT met* **do**

for each *rating* r_{ij} **in** Y **do**

$e_{ij} \leftarrow u_i v_j^T + b_i + d_j + \mu - y_{ij}$

$v_j \leftarrow v_j - \beta(e_{ij} u_i + \lambda v_j)$

$u_i \leftarrow u_i - \beta(e_{ij} v_j + \lambda u_i)$

$b_i \leftarrow b_i - \beta(e_{ij} + \lambda b_i)$

$d_j \leftarrow d_j - \beta(e_{ij} + \lambda d_j)$

end

end

4.2.2 Thử nghiệm, đánh giá mô hình

Trong phần này ta sẽ đi thử nghiệm mô hình đã được xây dựng trong phần 4.2.1 trên tập dữ liệu điểm đã được tạo trước đó (ở phần 4.1).

Trước tiên, dữ liệu cần được chuẩn hóa về dạng phù hợp với mô hình:

- Chuyển điểm môn học về hệ 4 theo cách thức quy đổi của trường Đại học Bách khoa Hà Nội. Cụ thể:

Thang 10	0.0 ÷ 3.9	4.0 ÷ 4.9	5.0 ÷ 5.4	5.5 ÷ 6.4	6.5 ÷ 6.9	7 ÷ 7.9	8 ÷ 8.4	8.5 ÷ 10
Thang 4	0	1	1.5	2	2.5	3	3.5	4

Bảng 4.6: Bảng quy đổi điểm số từ thang 10 về thang 4 của Đại học Bách khoa Hà Nội

- Mỗi sinh viên và môn học sẽ được đánh số nguyên từ 0 (tương ứng với vị trí hàng

và cột trong ma trận Utility matrix) và ma trận Utility matrix Y sẽ được biểu diễn dưới dạng *sparse matrix* như hình 4.7.

Với quá trình huấn luyện, dữ liệu sẽ được chia làm 2 tập: train và test theo tỷ lệ 7:3.

Một số kết quả thử nghiệm với các tham số đầu vào khác nhau: k - số nhân tố tiềm ẩn, β - tốc độ học, λ - tham số regularization, max_iter - số vòng lặp tối đa (điều kiện dừng) với mô hình có sử dụng bias.

STT	Tham số	RMSE train	RMSE test
1	$k = 50, \lambda = 0, \beta = 0.005, max_iter = 50$	0.1391	0.8921
2	$k = 50, \lambda = 0.02, \beta = 0.005, max_iter = 50$	0.3700	0.7353
3	$k = 50, \lambda = 0.02, \beta = 0.001, max_iter = 50$	0.6704	0.7035
4	$k = 50, \lambda = 0.02, \beta = 0.0005, max_iter = 50$	0.6808	0.6990
5	$k = 50, \lambda = 0.02, \beta = 0.0001, max_iter = 100$	0.6893	0.6967
6	$k = 50, \lambda = 0.1, \beta = 0.005, max_iter = 50$	0.6874	0.7000

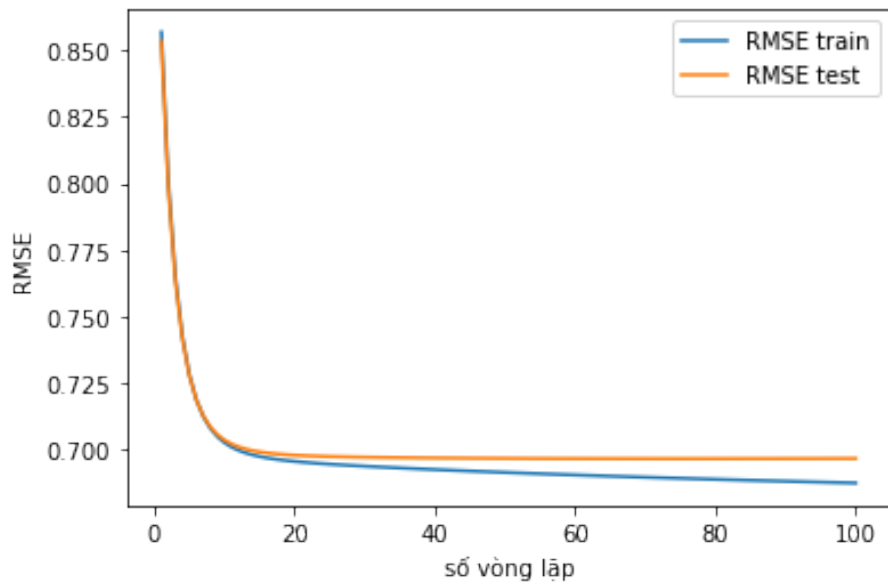
Bảng 4.7: Độ đo RMSE trên tập train và tập test với các tham số khác nhau trên mô hình kết hợp bias

Ở trường hợp đầu tiên, $\lambda = 0$ tức không sử dụng regularization, ta thấy mô hình bị overfitting do mô hình quá fit với tập huấn luyện (RMSE train rất thấp) trong khi đó giá trị RMSE trên tập kiểm thử khá cao. Ở các trường hợp khác có sử dụng regularization, chất lượng mô hình được cải thiện rõ rệt.

Bảng 4.8 là kết quả thử nghiệm mô hình khi không sử dụng bias. Nhìn chung ta cần nhiều vòng lặp hơn để có thể tìm được mô hình tối ưu, tuy nhiên chất lượng mô hình không tốt như khi sử dụng bias.

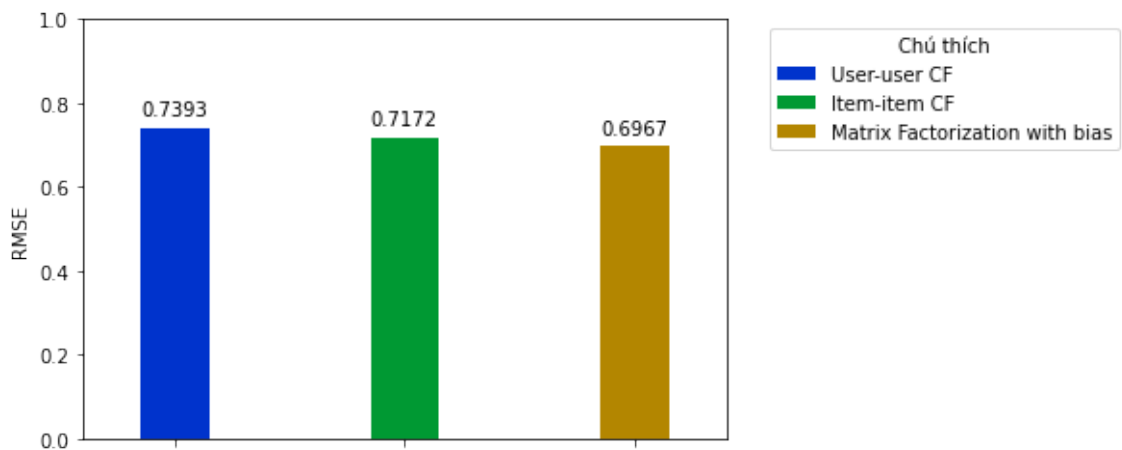
STT	Tham số	RMSE train	RMSE test
1	$k = 50, \lambda = 0, \beta = 0.005, max_iter = 50$	0.1379	0.9116
2	$k = 50, \lambda = 0.02, \beta = 0.005, max_iter = 50$	0.3638	0.7418
3	$k = 50, \lambda = 0.02, \beta = 0.001, max_iter = 50$	0.6752	0.7175
4	$k = 50, \lambda = 0.02, \beta = 0.0005, max_iter = 115$	0.6694	0.7170
5	$k = 50, \lambda = 0.02, \beta = 0.0001, max_iter = 300$	0.6877	0.7235
6	$k = 50, \lambda = 0.1, \beta = 0.005, max_iter = 110$	0.6344	0.7160

Bảng 4.8: Độ đo RMSE trên tập train và test với các tham số khác nhau trên mô hình không kết hợp bias



Hình 4.8: Mô hình sử dụng bias với $k = 50$, $\lambda = 0.02$, $\beta = 0.0001$, $max_iter = 100$

Để đánh giá được giải thuật chạy trên tập dữ liệu đã xử lý, em thực hiện so sánh với các giải thuật khác. Hình 4.9 dưới đây mô tả giá trị RMSE khi sử dụng các giải thuật khác nhau trên cùng tập dữ liệu. Hai giải thuật thuộc nhóm *Lọc cộng tác dựa trên láng giềng* User-user CF và Item-item CF cho các kết quả tương ứng là 0.7393 và 0.7172, trong khi đó giải thuật mà ta xây dựng cho kết quả tốt hơn một chút 0.6967.



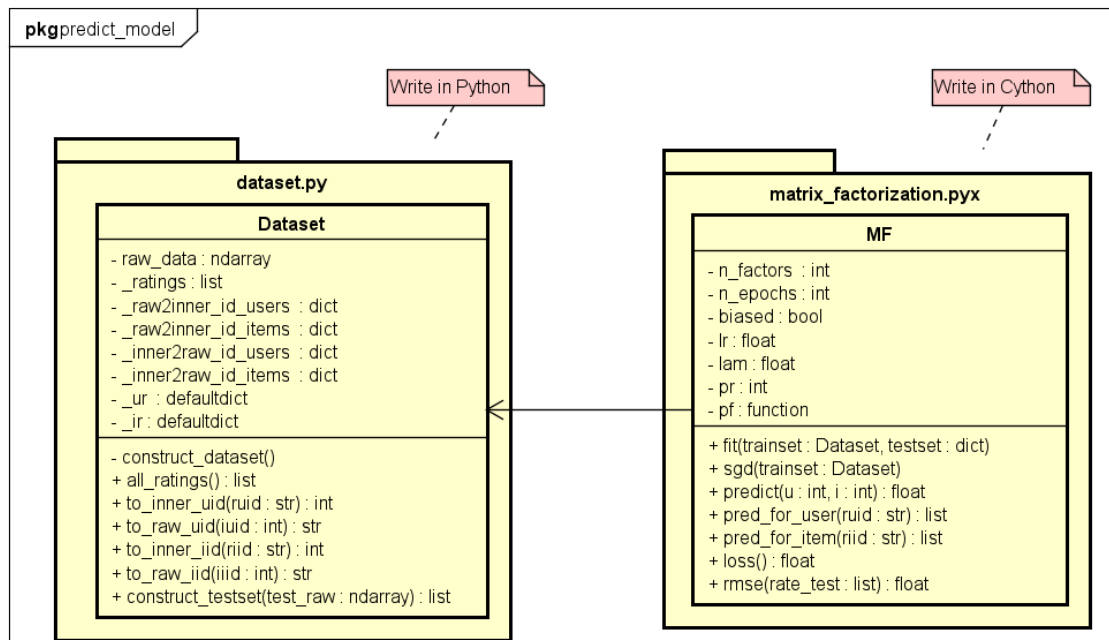
Hình 4.9: Biểu đồ so sánh giữa các giải thuật

4.3. Cài đặt và tích hợp mô hình

Sau khi đã lựa chọn được mô hình huấn luyện và xây dựng giải thuật, công việc tiếp theo chính là cài đặt, tích hợp mô hình đó để triển khai chức năng gợi ý đăng ký học tập cho hệ thống. Mô hình này cần được cài đặt bằng ngôn ngữ lập trình Python để có thể dễ dàng tích hợp vào phần backend của hệ thống. Tuy nhiên Python không phải là một ngôn ngữ lập trình có tốc độ nhanh nên khi chạy mô hình sẽ mất khá nhiều thời gian. Do vậy,

phần lõi của mô hình, tức phần giải thuật SGD em sẽ triển khai bằng Cython để có hiệu năng cao hơn, trong khi vẫn có thể sử dụng giải thuật này tại bất kỳ đoạn mã python nào.

Hình 4.10 dưới đây là sơ đồ thiết kế cho mô hình gợi ý, được đóng gói vào trong một python package `predict_model` và được đặt trong phần backend của hệ thống.



Hình 4.10: Thiết kế chi tiết package `predict_model`

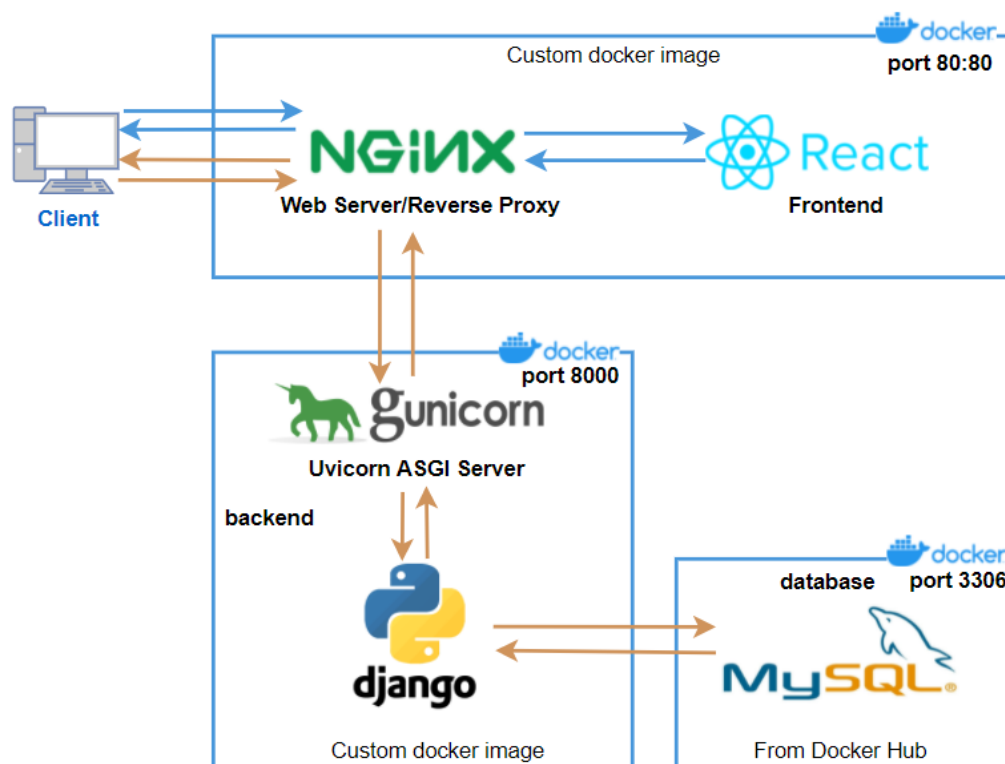
Thiết kế này gồm hai thành phần chính: `dataset` và `matrix_factorization`. Thành phần `dataset` đóng vai trò tiền xử lý, có nhiệm vụ chuyển đổi dữ liệu đầu vào là ma trận Utility matrix Y sang định dạng được sử dụng trong giải thuật huấn luyện, cụ thể sẽ chuyển về dạng *sparse matrix* như trong Hình 4.7, mỗi rating sẽ được biểu diễn dưới dạng (*user_index*, *item_index*, *rating*) trong đó các *user* và *item* sẽ được đánh số 0,1,2,3... đồng thời cung cấp các chức năng giúp chuyển đổi qua lại giữa hai kiểu dữ liệu.

Thành phần `matrix_factorization` là phần quan trọng nhất của package. Thành phần này nhận các tham số cài đặt đầu vào và dữ liệu đã được xử lý bởi `dataset`, thực hiện giải thuật huấn luyện. Bên cạnh đó còn một số chức năng khác như lấy kết quả dự đoán cho *user* hoặc *item*, tính toán RMSE và giá trị hàm mất mát.

Package `predict_model` sẽ được sử dụng trong chức năng huấn luyện của tác nhân quản lý. Sau khi quá trình huấn luyện kết thúc, hệ thống tính toán các kết quả gợi ý môn học cho từng sinh viên và lưu lại trong cơ sở dữ liệu (chỉ lưu các kết quả ứng với các môn học tự do). Chức năng gợi ý môn học cho sinh viên sẽ gợi ý các môn học chưa qua, các môn học bắt buộc, và với các môn học tự do sẽ dựa trên các kết quả dự đoán đã được lưu lại trước đó.

Chương 5: Triển khai và kiểm thử hệ thống

5.1. Cài đặt và triển khai hệ thống



Hình 5.1: Sơ đồ triển khai hệ thống gợi ý đăng ký học tập

Hệ thống được chia làm ba thành phần triển khai: frontend, backend và database. Mỗi thành phần sẽ được cài đặt với docker, toàn bộ mã nguồn, môi trường thực thi đều được đóng gói thành các docker image.

Với frontend được xây dựng bằng ReactJS, khi triển khai sẽ được build thành các static files (html, css, js), do đó cần phải có một máy chủ web để có thể đáp ứng các yêu cầu truy xuất đến các tài nguyên này. Em lựa chọn nginx để làm máy chủ web, bên cạnh đó nginx còn đóng vai trò là một reverse proxy trung gian giữa client và backend server khi client gửi các yêu cầu gọi đến api ở phía backend, điều này giúp kiểm soát tập trung các yêu cầu (request) và tăng tính bảo mật, độ tin cậy của hệ thống.

Với backend, django app được quản lý bởi một ASGI server (Uvicorn) đáp ứng các kết nối HTTP và WebSocket. Gunicorn được sử dụng để quản lý nhiều Uvicorn server cung cấp khả năng chạy các tiến trình song song, giúp tăng tốc độ xử lý yêu cầu của hệ thống.

Với cơ sở dữ liệu, hệ thống sử dụng docker image có sẵn của MySQL trên Docker Hub ¹.

¹<https://hub.docker.com/>

Sau khi cài đặt và đóng gói hoàn tất, việc triển khai hệ thống trở nên hoàn toàn đơn giản với docker, ta chỉ cần chạy các docker image đã xây dựng trước đó, mỗi thành phần được chạy lúc này sẽ tương ứng với một docker container. **Bảng 5.1** mô tả thông tin cấu hình của thiết bị triển khai hệ thống thử nghiệm.

Tên cấu hình	Thông số
Hệ điều hành	Ubuntu 20.04
CPU	Intel Core i5-6300HQ 2.3GHz
RAM	8GB

Bảng 5.1: Thông tin cấu hình thiết bị triển khai hệ thống


Dưới đây là một vài hình ảnh minh họa cho các chức năng của hệ thống:

Hình 5.2 minh họa cho chức năng gợi ý học tập của sinh viên. Giao diện hiển thị các môn học gợi ý với những giải thích, đánh giá cụ thể, bên cạnh đó là hiển thị một vài thông tin liên quan đến tình hình học tập, chương trình đào tạo để sinh viên dễ theo dõi, đưa ra quyết định học tập phù hợp.


Hình 5.3 minh họa cho chức năng xem điểm môn học, theo dõi các kết quả học tập của sinh viên. Dữ liệu học tập được hiển thị dưới dạng bảng, sinh viên có thể sắp xếp, tìm kiếm, lọc theo ý muốn.

Hình 5.4 là giao diện biểu đồ học tập, hiển thị các dữ liệu thống kê của sinh viên dưới dạng biểu đồ trực quan.

Hình 5.5 và **Hình 5.6** minh họa cho chức năng huấn luyện với tham số mặc định và với tham số tự điều chỉnh dành cho quản lý. Cửa sổ logs trực quan cập nhật quá trình huấn luyện trong thời gian thực. Kết quả huấn luyện được hiển thị ở phần bên phải kèm theo chức năng cho phép lưu lại các kết quả này.


TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
 HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Trang chủ
 Học tập


 20173287

Gợi ý học tập kỳ 20211

▶ Học phần chưa qua

▼ Học phần bắt buộc

PE1024 - Bơi lội
Kỳ học: 2 • Tín chỉ: 0

IT4995 - Đồ án tốt nghiệp cử nhân
Kỳ học: 8 • Tín chỉ: 6

IT4244 - Quản trị dự án CNTT
Kỳ học: 8 • Tín chỉ: 2

▼ Học phần tự do

IT4735 - IoT và ứng dụng
Kỳ học: 7 • Tín chỉ: 2
Mức độ gợi ý: ★★★★★★

IT4409 - Công nghệ Web và dịch vụ trực tuyến
Kỳ học: 7 • Tín chỉ: 3
Mức độ gợi ý: ★★★★★☆

IT4263 - An ninh mạng
Kỳ học: 7 • Tín chỉ: 3
Mức độ gợi ý: ★★★☆☆☆

CPA
2.52


TC tích lũy
114

TC nợ
5

Chương trình đào tạo

	Mã loại HP	Tên loại HP	Số lượng HP	Tổng TC	Tổng đạt
+	7	Đồ án/Khóa luận Tốt nghiệp	1	6	0
+	210	Lý luận chính trị	5	12	12
+	230	Giáo dục Quốc phòng - An ninh	3	0	0
+	240	Tiếng Anh	2	0	0
+	250	Khối kiến thức Toán và Khoa học cơ bản	10	32	27
+	300	Cơ sở và cốt lõi ngành	20	50	48
+	320	Khối kiến thức bổ trợ	6	12	12
+	360	Thực tập	1	2	2
+	400	Mô đun 1	3	8	8
+	401	Mô đun 2	5	13	5
+	220	Giáo dục thể chất	2	0	0

HỆ THỐNG GỢI Ý ĐĂNG KÍ HỌC TẬP
 Email: hustrs.dev@gmail.com - Tel: +84(0)8 6868 8686
 Copyright © by nguyenvangoc



Trường đại học Bách khoa Hà Nội
 Số 1 Đại Cồ Việt, Hai Bà Trưng, Hà Nội

Hình 5.2: Giao diện gợi ý học tập

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Trang chủ Học tập

20173287

THÔNG TIN SINH VIÊN

Sinh viên: **Nguyễn Bá Ngọc**
MSSV: **20173287**
Chương trình: **Công nghệ thông tin**
Email: **ngoc.nb173287@sis.hust.edu.vn**
Năm nhập học: **2017**
Trạng thái: **Chưa tốt nghiệp**

Bảng điểm môn học

Học kỳ	Mã HP	Tên HP	Tin chỉ	Điểm tổng kết	Điểm chữ
20202	ET3262	Tư duy công nghệ và thiết kế kỹ thuật	2	7.8	B
20201	IT3940	Project III	3	9.6	A+
20201	IT4350	Kiến trúc các hệ thống thông tin và ứng dụng	3	6.5	C+
20201	IT4490	Thiết kế và xây dựng phần mềm	3	7.4	B
20201	EM1180	Văn hóa kinh doanh và tinh thần khởi nghiệp	2	5.8	C
20201	IT4501	Đảm bảo chất lượng Phần mềm	2	8.1	B+
20192	IT3120	Phân tích và thiết kế hệ thống	2	6.8	C+
20192	IT3930	Project II	2	8.2	B+
20192	IT4015	Nhập môn an toàn thông tin	3	5.9	C
20192	IT4060	Lập trình mạng	2	4.8	D

< 1 2 3 4 5 6 > 10 / page

Kết quả học tập sinh viên

Học kỳ	GPA	CPA	TC ĐK	TC qua	TC tích lũy	TC nợ
20202	3.00	2.52	2	2	114	5
20201	3.38	2.51	13	13	112	5
20192	2.93	2.40	15	15	99	5
20191	2.23	2.30	22	22	84	5
20182	2.38	2.33	24	24	62	5
20181	1.59	2.30	16	11	38	5
20172	2.73	2.82	11	11	22	0
20171	2.91	2.91	11	11	11	0

HỆ THỐNG GỢI Ý ĐĂNG KÍ HỌC TẬP
Email: husts.dev@gmail.com - Tel: +84(0)6 6868 8686
Copyright © by nguyennangoc

Trường đại học Bách khoa Hà Nội
Số 1 Đại Cồ Việt, Hai Bà Trưng, Hà Nội

Hình 5.3: Giao diện điểm môn học

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Trang chủ Học tập

20173287

Biểu đồ điểm

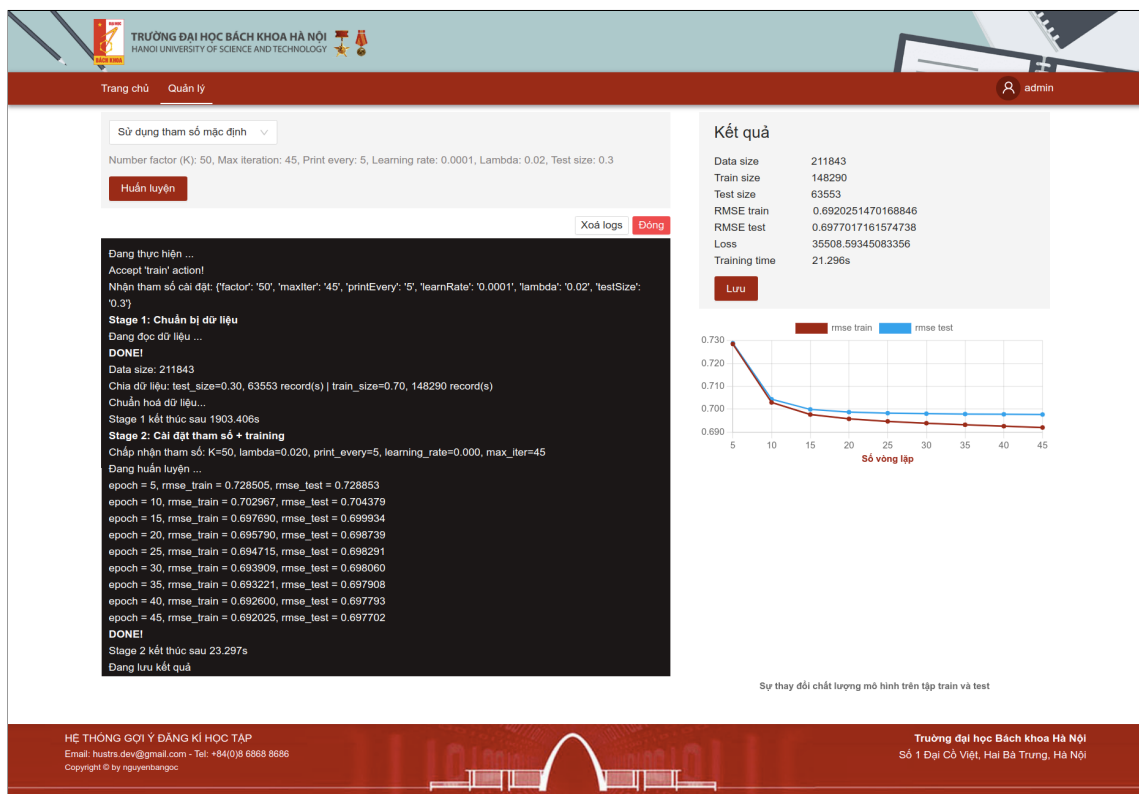
Biểu đồ tích lũy học tập

Biểu đồ đăng ký học tập

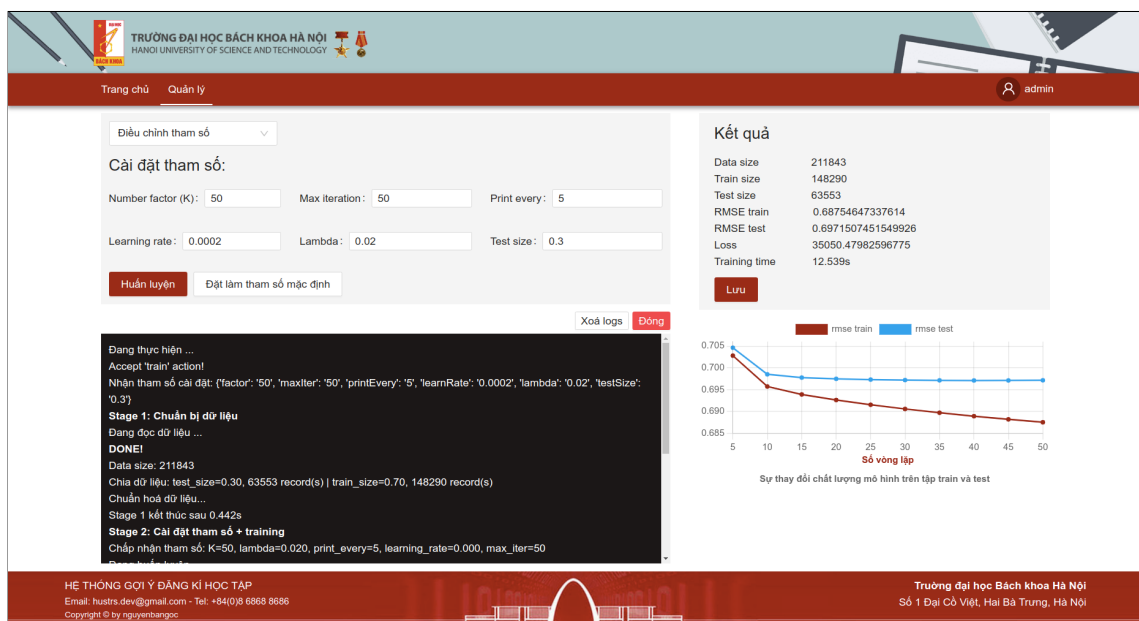
HỆ THỐNG GỢI Ý ĐĂNG KÍ HỌC TẬP
Email: husts.dev@gmail.com - Tel: +84(0)6 6868 8686
Copyright © by nguyennangoc

Trường đại học Bách khoa Hà Nội
Số 1 Đại Cồ Việt, Hai Bà Trưng, Hà Nội

Hình 5.4: Giao diện biểu đồ học tập



Hình 5.5: Giao diện huấn luyện mô hình với tham số mặc định



Hình 5.6: Giao diện huấn luyện mô hình với tham số điều chỉnh

5.2. Kiểm thử hệ thống

5.2.1 Kiểm thử tương thích

Hệ thống sẽ được chạy thử nghiệm trên các thiết bị với cấu hình, hệ điều hành khác nhau để đánh giá mức độ tương thích của phần mềm, chi tiết được mô tả trong Bảng 5.2.

Thiết bị	Thông số kỹ thuật	Giao diện	Chức năng
Dell inspiron 7559	Màn hình: 15.6" FHD, RAM 8GB	Đạt	Đạt
MSI GP62	Màn hình: 15.6" FHD, RAM 8GB	Đạt	Đạt

Bảng 5.2: Thông kê kiểm thử tương thích trên các thiết bị

5.2.2 Kiểm thử hộp đen

Để đánh giá chức năng cho hệ thống gọi ý đăng ký học tập em đã tiến hành kiểm thử hộp đen cho từng chức năng. Các kịch bản kiểm thử được mô tả cụ thể trong Bảng 5.3.

Bảng 5.3: Danh sách kịch bản kiểm thử

Chức năng	Đầu vào	Đầu ra mong muốn	Kết quả
Đăng nhập	Nhập tài khoản và mật khẩu với vai trò sinh viên hoặc quản lý	Đăng nhập thành công vào hệ thống	Đạt
	Nhập thiếu trường	Highlight đỏ ô input của trường bị thiếu	Đạt
	Nhập sai tài khoản và mật khẩu	Hiện thị popup thông báo lỗi ở góc trên bên phải màn hình	Đạt
Huấn luyện mô hình	Tham số huấn luyện hợp lệ, nhấn Huấn luyện	Hiện thị log message quá trình huấn luyện và giao diện kết quả huấn luyện	Đạt
	Chọn Điều chỉnh tham số	Hiện thị giao diện điều chỉnh các tham số huấn luyện	Đạt
	Tham số huấn luyện không hợp lệ, nhấn huấn luyện	Hiện thị thông báo lỗi ở ô input nhập sai giá trị	Đạt
Đặt tham số huấn luyện mặc định	Tham số huấn luyện hợp lệ, nhấn Đặt làm tham số mặc định	Hiện thị popup thông báo Lưu thành công ở góc trên bên phải màn hình	Đạt

Bảng 5.3: Danh sách kịch bản kiểm thử

Chức năng	Đầu vào	Đầu ra mong muốn	Kết quả
	Tham số huấn luyện không hợp lệ, nhấn Đặt làm tham số mặc định	Hiển thị thông báo lỗi ở ô input nhập sai giá trị	Đạt
Lưu kết quả huấn luyện	Nhấn Lưu	Hiển thị popup thông báo Lưu thành công ở góc trên bên phải màn hình	Đạt
Thay đổi cài đặt	Giá trị trường kỳ học hiện tại	Hiển thị popup thông báo Cập nhật thành công ở góc trên bên phải màn hình	Đạt
	Giá trị các tham số huấn luyện hợp lệ	Hiển thị popup thông báo Cập nhật thành công ở góc trên bên phải màn hình	Đạt
	Giá trị các tham số huấn luyện không hợp lệ	Hiển thị thông báo lỗi ở ô input nhập sai giá trị	Đạt
Xem lịch sử huấn luyện	Nhấn button Lịch sử huấn luyện	Hiển thị danh sách huấn luyện hoặc danh sách trống	Đạt
Chi tiết huấn luyện	Chọn một phiên bản trong danh sách lịch sử huấn luyện	Hiển thị giao diện chi tiết của phiên bản huấn luyện	Đạt
	Chọn Sử dụng	Hiển thị popup thông báo Cập nhật thành công ở góc trên bên phải màn hình, button Sử dụng chuyển thành Bỏ sử dụng	Đạt
	Chọn Bỏ sử dụng	Hiển thị popup thông báo Cập nhật thành công ở góc trên bên phải màn hình, button Bỏ sử dụng chuyển thành Sử dụng	Đạt
	Chọn Xóa phiên bản	Hiển thị modal xác nhận xóa	Đạt
	Xác nhận xóa	Hiển thị danh sách không chứa phiên bản vừa xóa	Đạt

Bảng 5.3: Danh sách kịch bản kiểm thử

Chức năng	Đầu vào	Đầu ra mong muốn	Kết quả
Xem gợi ý học tập	Chọn xem gợi ý học tập với tài khoản đăng nhập là sinh viên có trạng thái đã tốt nghiệp	Hiển thị thông báo không có kết quả gợi ý	Đạt
	Chọn xem gợi ý học tập với tài khoản đăng nhập là sinh viên có trạng thái chưa tốt nghiệp	Hiển thị các gợi ý môn học	Đạt

Chương 6: Kết luận và hướng phát triển

6.1. Kết luận

Sau quá trình tìm hiểu, phân tích em đã xây dựng thành công hệ thống gợi ý đăng ký học tập trong phạm vi trường Đại học Bách khoa Hà Nội, giải quyết được các mục tiêu đã đặt ra trong phần 1.2. Đồ án đặt ra ba mục tiêu chính, trong đó Chương 3 giải quyết mục tiêu thứ nhất *”Xây dựng một hệ thống gợi ý đăng ký học tập trên nền tảng website với hai phân hệ sinh viên và quản lý”*; mục tiêu thứ hai *”Xây dựng chức năng gợi ý cho hệ thống”* được chia làm ba mục tiêu nhỏ hơn cần hoàn thành, trong đó mục tiêu (2a) được trình bày chi tiết trong phần Cơ sở lý thuyết của Chương 2, mục tiêu (2b) và (2c) được giải quyết trong Chương 4; và mục tiêu thứ ba *”Triển khai và kiểm thử toàn bộ hệ thống”* được giải quyết trong Chương 5. Ứng chạy tốt trên nhiều môi trường, có giao diện đồng nhất với các hệ thống của trường Bách khoa, các công nghệ được sử dụng giúp triển khai và tích hợp hệ thống nhanh chóng và dễ dàng.

Đồ án đã tạo điều kiện cho em được thực hành các bước trong quy trình xây dựng một hệ thống phần mềm hoàn chỉnh, có tính ứng dụng trong thực tiễn, thúc đẩy bản thân tìm hiểu các kiến thức, công nghệ, xây dựng các giải pháp để giải quyết các vấn đề đặt ra. Quá trình thực hiện đồ án đã giúp em tích lũy thêm nhiều kinh nghiệm, là nền tảng phát triển bản thân sau này.

Tuy nhiên, đồ án vẫn còn một số điểm chưa được hoàn thiện xuất phát từ những hạn chế về thời gian cũng như kinh nghiệm của bản thân. Thứ nhất, phạm vi đáp ứng của hệ thống hiện còn hạn chế trong ngành Công nghệ thông tin, lý do là sự hạn chế về mặt dữ liệu chương trình học, dữ liệu sinh viên, dữ liệu học tập không sẵn có và phải sinh thủ công. Do vậy khi hệ thống được mở rộng ra phạm vi toàn trường, sẽ phải có những điều chỉnh, thay đổi để có thể hoạt động ổn định, đưa ra kết quả gợi ý phù hợp với từng chuyên ngành, từng sinh viên. Thứ hai là với mô hình gợi ý, khi có các dữ liệu cập nhật thì mô hình cần phải được cập nhật. Điều này đồng nghĩa với việc ta phải tiếp tục thực hiện quá trình huấn luyện vốn tốn khá nhiều thời gian.

6.2. Hướng phát triển

Trong tương lai, hệ thống cần được mở rộng phạm vi đáp ứng ra toàn bộ trường Đại học Bách khoa Hà Nội, chức năng gợi ý cần xây dựng chi tiết hơn cho từng ngành, từng khoa. Mô hình gợi ý cũng cần được cải thiện để có thể tự động cập nhật khi có các dữ liệu thay đổi, làm tăng độ tin cậy của các kết quả gợi ý, một giải pháp có thể hướng đến đó là sử dụng Incremental Matrix Factorization [4].

Tài liệu tham khảo

- [1] Matrix calculus. <https://ccrma.stanford.edu/~dattorro/matrixcalc.pdf>. [truy cập lần cuối 25/5/2021].
- [2] Yehuda Koren, Robert Bell and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer Society*, 2009.
- [3] Breese J., Heckerman D., Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.
- [4] Vinagre J., Jorge A.M., Gama J. Fast incremental matrix factorization for recommendation with positive-only feedback. *UMAP*, 2011.
- [5] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Recommendation Systems*, page 307–339. Cambridge University Press, 3 edition, 2020.
- [6] Vũ Hữu Tiệp. Content-based recommendation systems.
<https://machinelearningcoban.com/2017/05/17/contentbasedrecommendersys/>, 2017. [truy cập lần cuối 12/5/2021].
- [7] Vũ Hữu Tiệp. Gradient descent.
<https://machinelearningcoban.com/2017/01/12/gradientdescent/>, 2017. [truy cập lần cuối 5/6/2021].
- [8] Wikipedia. Root-mean-square deviation.
https://en.wikipedia.org/wiki/Root-mean-square_deviation. [truy cập lần cuối 12/5/2021].