TRƯỜNG ĐẠI HỌC THĂNG LONG

**KHOA TOÁN TIN**

-----🙞🙜🕮🙞🙜-----

**DỰ ÁN PHẦN MỀM**

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM LỚP THỜI KHÓA BIỂU**

|  |  |
| --- | --- |
| **GIÁO VIÊN HƯỚNG DẪN** | **SINH VIÊN THỰC HIỆN** |
| **Mai Thúy Nga** | **Vũ Văn Minh - A29774**  **Đặng Anh Tú -\_A29378**  **Trần Xuân Vũ -\_A26828**  **Trần Văn Phương -\_A28800**  **Ngành: Khoa học máy tính** |

**HÀ NỘI – 2019**

**LỜI CẢM ƠN**

Qua 4 năm học tập và rèn luyện tại trường Trường ĐH Thăng Long, được sự chỉ bảo và giảng dạy nhiệt tình của quý thầy cô, đặc biệt là quý thầy cô khoa Công nghệ thông tin đã truyền đạt cho em những kiến thức về lý thuyết và thực hành trong suốt thời gian học ở trường. Và trong thời gian làm bài tập lớn môn project này chúng em xin bày tỏ lời cảm ơn sâu sắc nhất tới tất cả các tập thể, cá nhân đã tạo điều kiện giúp đỡ cho nhóm em trong suốt quá trình thực hiện nghiên cứu bài tập lớn này. Cùng với sự nổ lực của các thành viên trong nhóm, chúng em đã hoàn thành bài tập lớn của mình.

Từ những kết quả đạt được này, chúng em xin chân thành [cám ơn](http://baocaothuctap.net/loi-cam-on-hay-trong-viet-bai-bao-cao-thuc-tap-tot-nghiep/): Trước hết chúng em xin trân trọng cảm ơn Ban giám hiệu trường Đại học Thăng Long cùng các thầy cô giáo, những người đã trang bị kiến thức cho tôi trong suốt quá trình học tập. Gửi tới biết ơn chân thành và sâu sắc nhất với cô Mai Thúy Nga đã tận tình hướng dẫn chỉ bảo chúng em hoàn thành tốt bài tập lớn này. Do kiến thức còn hạn hẹp nên không tránh khỏi những thiếu sót trong cách hiểu, lỗi trình bày. Chúng em rất mong nhận được sự đóng góp ý kiến của quý thầy cô cùng toàn thể bạn học để đạt được kết quả tốt hơn.

**MỤC LỤC**

[LỜI GIỚI THIỆU 6](#_Toc24068263)

[DANH MỤC BẢNG BIỂU, HÌNH ẢNH 7](#_Toc24068264)

[CHƯƠNG 1. TỔNG QUAN VỀ PHẦN MỀM XẾP THỜI KHÓA BIỂU 10](#_Toc24068265)

[1.1. Mô tả bài toán 10](#_Toc24068266)

[1.1.1 Thực trạng 10](#_Toc24068267)

[1.1.2 Phân tích bài toán 10](#_Toc24068268)

[1.2. Yêu cầu nghiệp vụ 10](#_Toc24068269)

[1.3. Sơ đồ Usecase các chức năng chính 11](#_Toc24068270)

[1.3.1. Web 11](#_Toc24068271)

[1.3.1.1. Sơ đồ UseCase 11](#_Toc24068272)

[1.3.1.2. Các tác nhân hệ thống 12](#_Toc24068273)

[1.3.1.3. Các UseCase 12](#_Toc24068274)

[1.3.1.4. Bảng ánh xạ yêu cầu nghiệp vụ với các Usecase 13](#_Toc24068275)

[1.3.2. Mobile 13](#_Toc24068276)

[1.3.2.1. Sơ đồ user case 13](#_Toc24068277)

[1.3.2.2. Các tác nhân của hệ thống 14](#_Toc24068278)

[1.3.2.4. Bảng ánh xạ yêu cầu nghiệp vụ với các Usecase 15](#_Toc24068279)

[1.4. Kế hoạch thực hiện 15](#_Toc24068280)

[CHƯƠNG 2. KIẾN TRÚC TỔNG THỂ CỦA HỆ THỐNG 16](#_Toc24068281)

[2.1. Sơ đồ kiến trúc tổng thể 16](#_Toc24068282)

[2.1.1. Sơ đồ kiến trúc 16](#_Toc24068283)

[2.1.2. Mô tả kiến trúc 16](#_Toc24068284)

[2.1.3. Công nghệ sử dụng 17](#_Toc24068285)

[2.2. Server 19](#_Toc24068286)

[2.3. Web Client 19](#_Toc24068287)

[2.4. Cấu hình sử dụng 20](#_Toc24068288)

[CHƯƠNG 3. ĐẶC TẢ CÁC CHỨC NĂNG 21](#_Toc24068289)

[3.1. Web 21](#_Toc24068290)

[3.1.1. UC #01 - Đăng nhập 21](#_Toc24068291)

[3.1.2. UC #02 - Quản lý tài khoản 22](#_Toc24068292)

[3.1.3. UC #03 - Phân quyền 23](#_Toc24068293)

[3.1.4. UC #04 - Quản lý TKB trường 25](#_Toc24068294)

[3.1.5. UC #05 - Quản lý TKB khoa 27](#_Toc24068295)

[3.1.6. UC #06 - Quản lý phòng học 29](#_Toc24068296)

[3.1.7. UC #07 - Quản lý môn học 31](#_Toc24068297)

[3.1.8. UC #08 - Quản lý khối giờ học 33](#_Toc24068298)

[3.1.9. UC #09 - Quản lý ngành học 34](#_Toc24068299)

[3.1.10. UC #010 - Phân công giảng dạy 35](#_Toc24068300)

[3.1.11. UC #011 - Xem TKB cá nhân 36](#_Toc24068301)

[3.1.12. UC #012 - Xem TKB khoa 36](#_Toc24068302)

[3.2. Mobile 37](#_Toc24068303)

[3.2.1. UC #01 - Login 37](#_Toc24068304)

[3.2.2. UC #02 - Xem TKB toàn trường 41](#_Toc24068305)

[3.2.3. UC #03 - Xem TKB khoa 44](#_Toc24068306)

[3.2.4. UC #04 Xem TKB cá nhân 45](#_Toc24068307)

[CHƯƠNG 4: THIẾT KẾ API VÀ CƠ SỞ DỮ LIỆU 49](#_Toc24068308)

[4.1. Tổng thể 49](#_Toc24068309)

[4.2. Cấu trúc chung của một API 50](#_Toc24068310)

[4.3. Các API của hệ thống 51](#_Toc24068311)

[4.3.1. API dùng để quản lý các lớp học (/api/class) 51](#_Toc24068312)

[4.3.2. API dùng để quản lý các phòng học (/api/room) 52](#_Toc24068313)

[4.3.3. API dùng để quản lý các giáo viên (/api/teacher) 53](#_Toc24068314)

[4.3.4. API dùng để quản lý các môn học (/api/subject) 54](#_Toc24068315)

[4.3.5. API dùng để quản lý các học kỳ (/api/semester) 55](#_Toc24068316)

[4.3.6. API dùng để quản lý các ca học (/api/shift) 56](#_Toc24068317)

[4.3.7. API dùng để quản lý các bộ môn (/api/group-subject) 57](#_Toc24068318)

[4.3.8. API dùng để quản lý quyền hạn của các giáo viên (/api/role) 58](#_Toc24068319)

[4.3.9. API dùng để tạo token xác thực tài khoản (/api/class/login) 59](#_Toc24068320)

[4.4. Kiến trúc cơ sở dữ liệu 59](#_Toc24068321)

[4.4.1. Ý nghĩa các bảng cơ sở dữ liệu 59](#_Toc24068322)

[4.4.2. Cơ sở dữ liệu quản lý thời khóa biểu 65](#_Toc24068323)

[4.4.3. Cơ sơ dữ liệu phân quyền 65](#_Toc24068324)

[CHƯƠNG 5. THIẾT KẾ VÀ CÀI ĐẶT CÁC CHỨC NĂNG 67](#_Toc24068325)

[5.1. Thiết kế web 67](#_Toc24068326)

[5.1.1. Chức năng quản lý TKB trường 67](#_Toc24068327)

[5.1.2. Chức năng quản lý phòng học 67](#_Toc24068328)

[5.1.3. Chức năng quản lý môn học 68](#_Toc24068329)

[5.1.4. Chức năng quản lý khối giờ học 69](#_Toc24068330)

[5.1.5. Chức năng quản lý ngành học 70](#_Toc24068331)

[5.1.6. Chức năng xem TKB cá nhân 71](#_Toc24068332)

[5.1.7. Chức năng xem TKB khoa 71](#_Toc24068333)

[5.1.8. Chức năng phân quyền 72](#_Toc24068334)

[5.1.9. Chức năng quản lý tài khoản 73](#_Toc24068335)

[5.1.10. Chức năng phân công giảng dạy 74](#_Toc24068336)

[5.1.11. Chức năng quản lý TKB khoa 75](#_Toc24068337)

[5.1.12. Chức năng đăng nhập 76](#_Toc24068338)

[5.2. Thiết kế mobile 76](#_Toc24068339)

[5.2.1. Chức năng Login 76](#_Toc24068340)

[5.2.2. Chức năng Xem TKB trường 77](#_Toc24068341)

[5.2.3. Chức năng Xem TKB khoa 78](#_Toc24068342)

[5.2.4. Chức năng Xem TKB cá nhân 79](#_Toc24068343)

[5.3. Cài đặt chức năng web 80](#_Toc24068344)

[5.3.1. Dự án sử dụng Spring MVC 80](#_Toc24068345)

[5.3.2. Các lợi ích của Spring MVC 82](#_Toc24068346)

[5.3.3. Cấu trúc file mã nguồn 82](#_Toc24068347)

[5.3.4. Cấu trúc chung của 1 module 83](#_Toc24068348)

[5.3.5. Cài đặt các chức năng 84](#_Toc24068349)

[5.3.6. Ý nghĩa 85](#_Toc24068350)

[5.4. Cài đặt chức năng mobile 85](#_Toc24068351)

[5.4.1. Framework sử dụng: React Native 85](#_Toc24068352)

[5.4.2. Cấu trúc file mã nguồn 86](#_Toc24068353)

[CHƯƠNG 6: KIỂM THỬ 87](#_Toc24068354)

[6.1. Giới thiệu chung 87](#_Toc24068355)

[6.2. Kiểm thử 87](#_Toc24068356)

[6.3. Quy trình 87](#_Toc24068357)

[6.3.1. Requirement analysis- Phân tích yêu cầu 88](#_Toc24068358)

[6.3.2. Test planning- Lập kế hoạch kiểm thử 89](#_Toc24068359)

[6.3.3. Test case development- Thiết kế kịch bản kiểm thử 90](#_Toc24068360)

[6.3.4. Test environment set up- Thiết lập môi trường kiểm thử 90](#_Toc24068361)

[6.3.5. Test excution-Thực hiện kiểm thử 91](#_Toc24068362)

[6.3.6. Test cycle closure- Đóng chu trình kiểm thử 91](#_Toc24068363)

[6.4. Các module kiểm thử 92](#_Toc24068364)

[6.4.1. Login 92](#_Toc24068365)

[6.4.2. Quản lý thời khóa biểu trường 93](#_Toc24068366)

[6.4.3. Quản lý phòng học 94](#_Toc24068367)

[6.4.4. Quản lý môn học 95](#_Toc24068368)

[CHƯƠNG 7: THUẬT TOÁN TẠO THỜI KHÓA BIỂU 97](#_Toc24068369)

[7.1. Quy trình tạo thời khóa biểu 97](#_Toc24068370)

[7.2. Thuật toán 98](#_Toc24068371)

[7.2.1. Lấy danh sách các ca học còn trống 98](#_Toc24068372)

[7.2.2. Lấy danh sách các phòng học còn trống 99](#_Toc24068373)

# LỜI GIỚI THIỆU

Tài liệu này là tài liệu báo cáo bài tập lớn môn Project học kì I nhóm 2 năm học 2019-2020. Tài liệu được sử dụng để mô tả lại toàn bộ quá trình xây dựng phần mềm thời khóa biểu của trường Đại học Thăng Long. Tài liệu gồm 6 chương chính bao gồm:

* Chương 1 - Tổng quan về phần mềm thời khóa biểu: Chương này giúp người dùng có cách nhìn tổng quát về phần mềm được sử dụng trong tài liệu này
* Chương 2 - Kiến trúc tổng thể của hệ thống: Chương này giúp người dùng biết được sơ đồ kiến trúc tổng thể của phần mềm cũng như công nghệ được sử dụng trong phần mềm.
* Chương 3 - Đặc tả các chức năng: Chương này trình bày các chức năng chính của hệ thống với sơ đồ user-case và các luồng nghiệp vụ chính của phần mềm thông qua sơ đồ luồng
* Chương 4 - Thiết kế API và cơ sở dữ liệu: Chương này trình bày hệ thống API được sử dụng trong phần mềm và các cơ sở dữ liệu cơ bản được đưa vào phần mềm thời khóa biểu.
* Chương 5 - Thiết kế và cài đặt các chức năng: Chương này trình bày chi tiết hơn về các chức năng được phần mềm xây dựng
* Chương 6 - Kiểm thử: Chương này dùng để kiểm tra chất lượng phần mềm qua các test-case.

# DANH MỤC BẢNG BIỂU, HÌNH ẢNH

[Bảng 1: Bảng ánh xạ người dùng đối với các User-case web 13](#_Toc24055730)

[Bảng 2: Bảng ánh xạ yêu cầu nghiệp vụ đối với các Usecase mobile 15](#_Toc24055731)

[Bảng 3: Kế hoạch thực hiện 15](#_Toc24055732)

[Bảng 4: Bảng hạng mục cấu hình tối thiểu sử dụng 20](#_Toc24055733)

[Bảng 5: Cơ sở dữ liệu phòng học 60](#_Toc24055734)

[Bảng 6: Ý nghĩa cơ sở dữ liệu phòng học 60](#_Toc24055735)

[Bảng 7: Cơ sở dữ liệu môn học 60](#_Toc24055736)

[Bảng 8: Ý nghĩa cơ sở dữ liệu môn học 61](#_Toc24055737)

[Bảng 9: Cơ sở dữ liệu lớp học 61](#_Toc24055738)

[Bảng 10: Ý nghĩa cơ sở dữ liệu lớp học 61](#_Toc24055739)

[Bảng 11: Cơ sở dữ liệu ca học 62](#_Toc24055740)

[Bảng 12: Ý nghĩa cơ sở dữ liệu ca học 62](#_Toc24055741)

[Bảng 13: Cơ sở dữ liệu học kỳ 62](#_Toc24055742)

[Bảng 14: Ý nghĩa cơ sở dữ liệu học kỳ 63](#_Toc24055743)

[Bảng 15: Cơ sở dữ liệu bộ môn 63](#_Toc24055744)

[Bảng 16: Ý nghĩa cơ sở dữ liệu bộ môn 63](#_Toc24055745)

[Bảng 17: Cơ sở dữ liệu giáo viên 63](#_Toc24055746)

[Bảng 18: Ý nghĩa cơ sở dữ liệu giáo viên 63](#_Toc24055747)

[Bảng 19: Cơ sở dữ liệu quyền hạn 64](#_Toc24055748)

[Bảng 20: Ý nghĩa cơ sở dữ liệu quyền hạn 64](#_Toc24055749)

[Bảng 21: Cấu trúc thư mục các chức năng 84](#_Toc24055750)

[Bảng 22: Kết quả test chức năng login 93](#_Toc24055751)

[Bảng 23: Kết quả test chức năng quản lý thời khóa biểu trường 94](#_Toc24055752)

[Bảng 24: Kết quả test chức năng quản lý phòng học 95](#_Toc24055753)

[Bảng 25: Kết quả test chức năng quản lý môn học 96](#_Toc24055754)

[Hình 1: Sơ đồ Usecase trên web 11](#_Toc24068374)

[Hình 2: Sơ đồ User-case mobile 14](#_Toc24068375)

[Hình 3: Sơ đồ kiến trúc tổng thể 16](#_Toc24068376)

[Hình 4: Sơ đồ web-client 19](#_Toc24068377)

[Hình 5: Giao diện minh họa đăng nhập trên web 22](#_Toc24068378)

[Hình 6: Giao diện minh họa TKB trường 27](#_Toc24068379)

[Hình 7: Giao diện minh họa 31](#_Toc24068380)

[Hình 8: Giao diện minh họa quản lý môn học 33](#_Toc24068381)

[Hình 9: Giao diện đăng nhập trên mobile 39](#_Toc24068382)

[Hình 10: Sơ đồ lớp phân tích login trên mobile 40](#_Toc24068383)

[Hình 11: Sơ đồ trình tự login trên mobile 40](#_Toc24068384)

[Hình 12: Sơ đồ lớp phân tích xem TKB trường trên mobile 43](#_Toc24068385)

[Hình 13: Sơ đồ trình tự xem TKB trường trên mobile 43](#_Toc24068386)

[Hình 14: Sơ đồ lớp phân tích xem TKB khoa trên mobile 45](#_Toc24068387)

[Hình 15: Sơ đồ trình tự xem TKB khoa trên mobile 45](#_Toc24068388)

[Hình 16: Giao diện TKB cá nhân trên mobile 47](#_Toc24068389)

[Hình 17: Sơ đồ lớp phân tích xem TKB cá nhân trên mobile 48](#_Toc24068390)

[Hình 18: Sơ đồ trình tự xem TKB cá nhân trên mobile 48](#_Toc24068391)

[Hình 19: Sơ đồ tổng thể thiết kế API 49](#_Toc24068392)

[Hình 20: Các phương thức sử dụng trong API 50](#_Toc24068393)

[Hình 21: Cơ sở dữ liệu 59](#_Toc24068394)

[Hình 22: Các bảng quản lý thời khóa biểu 65](#_Toc24068395)

[Hình 23: Các bảng quản lý phân quyền 65](#_Toc24068396)

[Hình 24: Sơ đồ phân tích chức năng quản lý TKB trường 67](#_Toc24068397)

[Hình 25: Sơ đồ trình tự chức năng quản lý TKB trường 67](#_Toc24068398)

[Hình 26: Sơ đồ phân tích chức năng quản lý phòng học 67](#_Toc24068399)

[Hình 27: Sơ đồ trình tự chức năng thêm phòng học 68](#_Toc24068400)

[Hình 28: Sơ đồ lớp phân tích chức năng quản lý môn học 68](#_Toc24068401)

[Hình 29: Sơ đồ trình tự chức năng hiển thị danh sách các môn học 69](#_Toc24068402)

[Hình 30: Sơ đồ lớp phân tích chức năng quản lý giờ học 69](#_Toc24068403)

[Hình 31: Sơ đồ trình tự chức năng thêm giờ học 70](#_Toc24068404)

[Hình 32: Sơ đồ lớp phân tích chức năng quản lý ngành học 70](#_Toc24068405)

[Hình 33: Sơ đồ trình tự chức năng hiển thị danh sách ngành học 71](#_Toc24068406)

[Hình 34: Sơ đồ lớp phân tích chức năng xem TKB cá nhân 71](#_Toc24068407)

[Hình 35: Sơ đồ trình tự chức năng xem TKB cá nhân 71](#_Toc24068408)

[Hình 36: Sơ đồ lớp phân tích chức năng xem TKB khoa 71](#_Toc24068409)

[Hình 37: Sơ đồ trình tự chức năng xem TKB khoa 72](#_Toc24068410)

[Hình 38: Sơ đồ lớp phân tích chức năng phân quyền 72](#_Toc24068411)

[Hình 39: Sơ đồ trình tự chức năng thêm quyền 73](#_Toc24068412)

[Hình 40: Sơ đồ lớp phân tích chức năng quản lý tài khoản 73](#_Toc24068413)

[Hình 41: Sơ đồ trình tự chức năng hiển thị thông tin tài khoản 74](#_Toc24068414)

[Hình 42: Sơ đồ lớp phân tích chức năng phân công giảng dạy 74](#_Toc24068415)

[Hình 43: Sơ đồ trình tự hiển thị danh sách phân công 75](#_Toc24068416)

[Hình 44: Sơ đồ lớp phân tích chức năng quản lý TKB khoa 75](#_Toc24068417)

[Hình 45: Sơ đồ trình tự chức năng quản lý TKB khoa 75](#_Toc24068418)

[Hình 46: Sơ đồ lớp phân tích chức năng đăng nhập 76](#_Toc24068419)

[Hình 47: Sơ đồ trình tự chức năng đăng nhập 76](#_Toc24068420)

[Hình 48: Sơ đồ lớp phân tích chức năng đăng nhập 76](#_Toc24068421)

[Hình 49: Sơ đồ trình tự chức năng đăng nhập 77](#_Toc24068422)

[Hình 50: Sơ đồ lớp phân tích chức năng Xem TKB 77](#_Toc24068423)

[Hình 51: Sơ đồ trình tự chức năng xem TKB toàn trường 78](#_Toc24068424)

[Hình 52: Sơ đồ lớp phân tích chức năng Xem TKB khoa 78](#_Toc24068425)

[Hình 53: Sơ đồ trình tự chức năng Xem TKB toàn trường 79](#_Toc24068426)

[Hình 54: Sơ đồ lớp phân tích chức năng Xem TKB cá nhân 79](#_Toc24068427)

[Hình 55: Sơ đồ trình tự chức năng Xem TKB cá nhân 80](#_Toc24068428)

[Hình 56: Mô hình dự án sử dụng chức năng Spring MVC 80](#_Toc24068429)

[Hình 57: Mô hình dự án sử dụng chức năng Spring MVC 81](#_Toc24068430)

[Hình 58: Cấu trúc file mã nguồn 82](#_Toc24068431)

[Hình 59: Cấu trúc file mã nguồn 83](#_Toc24068432)

[Hình 60: Cấu trúc file mã nguồn 86](#_Toc24068433)

[Hình 61: Quy trình kiểm thử phần mềm 88](#_Toc24068434)

[Hình 62: Module kiểm thử login 92](#_Toc24068435)

[Hình 63: Module kiểm thử quản lý thời khóa biểu trường 93](#_Toc24068436)

[Hình 64: Module kiểm thử quản lý phòng học 94](#_Toc24068437)

[Hình 65: Module kiểm thử quản lý môn học 95](#_Toc24068438)

[Hình 67: Tạo thời khóa biểu 97](#_Toc24068439)

[Hình 68: Thuật toán danh sách các ca học 98](#_Toc24068440)

[Hình 69: Thuật toán danh sách phòng trống 99](#_Toc24068441)

# CHƯƠNG 1. TỔNG QUAN VỀ PHẦN MỀM XẾP THỜI KHÓA BIỂU

* 1. **Mô tả bài toán**

Trong cuộc sống ta thường gặp các bài toán liên quan đến sắp xếp lịch như lịch làm việc, xếp lịch cho một bài toán kiểm thử, xếp lịch thi,….Đối với loại bài toán này ta cần phải tìm một phương pháp xếp lịch thỏa mãn các yêu cầu và tối ưu hiệu quả những tài nguyên hiện có.

Bài toán xếp thời khóa biểu trong trường học nói chung và trong trường học nói riêng là một trong những bài toán như thế. Có rất nhiều yêu cầu được đặt ra trong bài toàn này như đối tượng tham gia (Phòng đào tạo, Quản lý bộ môn, Giáo viên), tài nguyên phục vụ giảng dạy (Phòng học), thời gian (ca học, ngày học), các yêu cầu khác nhau tùy thuộc vào từng trường. Vấn đề đặt ra là cần xây dựng một thời khóa biểu thỏa mãn các yêu cầu mà nhà trường nêu ra.

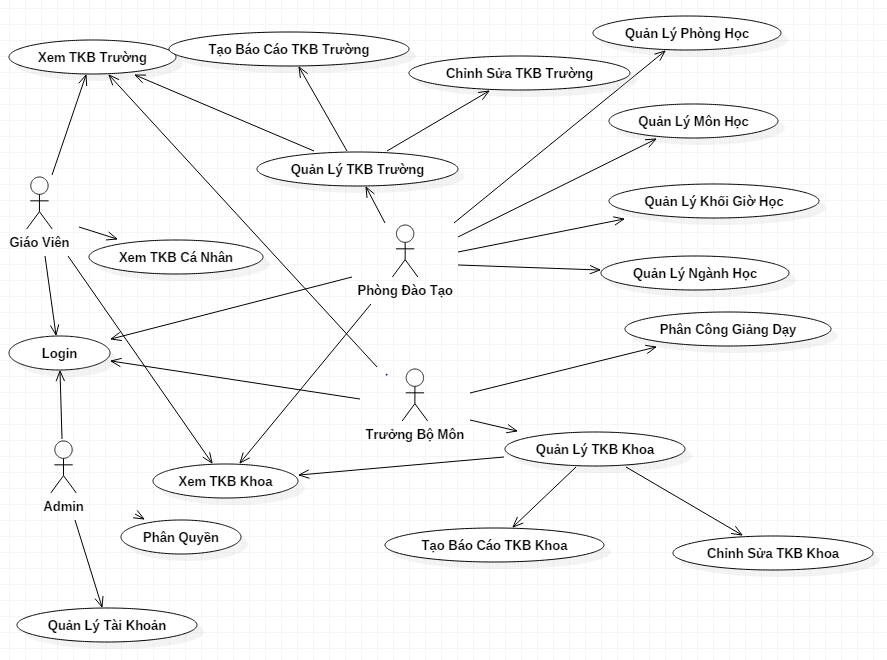
* + 1. **Thực trạng**

Hiện nay, các trường đại học ở Việt Nam đang đào tạo theo hệ thống tín chỉ. Điều này rất tốt cho công tác giảng dạy. Trường Đại học Thăng Long may mắn là 1 trong những trường đi đầu trong việc áp dụng hệ thống tín chỉ vào việc dạy và học cho sinh viên. Trong những năm gần đây, công nghệ thông tin phát triển dẫn đến việc dạy và học trong trường cũng đã thay đổi theo. Đặc biệt là việc sắp xếp thời khóa biểu giảng dạy cho giáo viên. Việc thiếp lập thời khóa biểu toàn trường vẫn còn bằng giấy nên rất mất thời gian và công sức của phòng đào tạo cũng như phòng bộ môn. Sau đó lại dùng giấy để giúp thông báo đến các khoa cho giáo viên. thay vì như vậy, một hệ thống quản lý thời khóa biểu trên internet sẽ giải quyết những vấn đề này.

* + 1. **Phân tích bài toán**

Vì xếp lịch TKB là một bài toán nên khó nên phần mềm sẽ có một số chức năng chính sau đây:

* Chuẩn bị dữ liệu thời khóa biểu
* Chuẩn bị các file excel để import (môn học, phòng học, giáo viên, học kì, ca học,)
* Xây dựng thời khóa biểu
  + Các chức năng quản lý (thời khóa biểu, phòng học, môn học)
* Xem TKB
* Xem TKB cá nhân
* Xem TKB bộ môn
* Xem TKB toàn trường
* Thống kê, báo cáo.
* Thống kê tần suất sử dụng phòng học
  1. **Yêu cầu nghiệp vụ**
* BR1: Chuẩn bị dữ liệu TKB
  + BR1.1: Quản lý môn học
  + BR1.2: Quản lý phòng học
  + BR1.3: Quản lý giáo viên
* BR2: Xây dựng thời khóa biểu
  + BR2.1: Đăng nhập
  + BR2.2: Quản lý tài khoản
  + BR2.3: Phân quyền
  + BR2.4: Quản lý TKB trường
  + BR2.5: Quản lý TKB khoa
  + BR2.6: Quản lý phòng học
  + BR2.7: Quản lý môn học
  + BR2.8: Quản lý khối giờ học
  + BR2.9: Quản lý ngành học
  + BR2.10: Phân công giảng dạy
* BR3: Xem TKB
  + BR3.1: Xem TKB trường
  + BR3.2: Xem TKB cá nhân
* BR4: Thống kê, báo cáo
  1. **Sơ đồ Usecase các chức năng chính**
     1. **Web**
        1. **Sơ đồ UseCase**

****

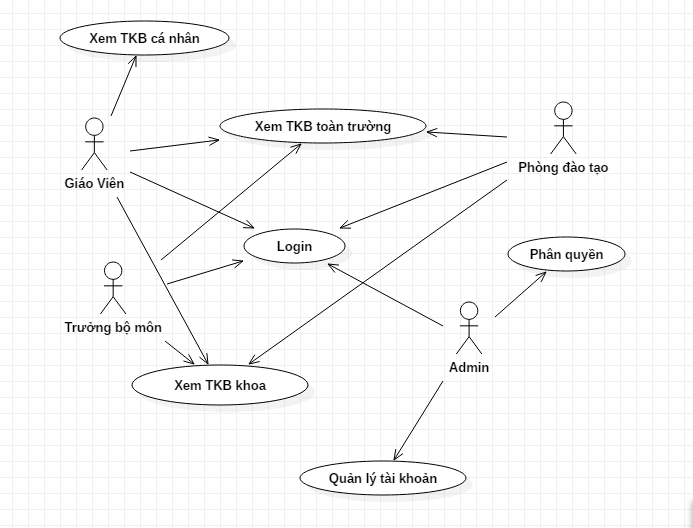
Hình 1: Sơ đồ Usecase trên web

* + - 1. **Các tác nhân hệ thống**
* Phòng đào tạo: là người trực tiếp sử dụng hệ thống. Phòng Đào Tạo có chức năng to lớn nhất là tổng hợp thời khóa biểu của Bộ Môn sau đó sắp xếp thời khóa biểu cho toàn trường.
* Trưởng bộ môn: cũng là người trực tiếp sử dụng hệ thống nhưng chỉ dừng lại ở mức độ nhất định. Bộ Môn có chức năng phân công cho Giáo Viên giảng dạy môn học.
* Giáo viên: cũng là người trực tiếp sử dụng hệ thống. Giáo viên hang ngày phải lên kiểm tra thời khóa biểu của mình trên hệ thống để nắm bắt được thông tin giảng dạy trong ngày đó.
* Admin: là người có chức năng quản lý tài khoản của toàn bộ hệ thống
  + - 1. **Các UseCase**
* UC #01 - Đăng nhập: người dùng đăng nhập để thực hiện các chức năng của hệ thống
* UC #02 - Quản lý tài khoản: chức năng cho phép admin quản lý thêm mới, sửa, xóa tài khoản đăng nhập của cả hệ thống.
* UC #03 - Phân quyền: chức năng cho phép admin phân quyền sử dụng các chức năng của hệ thống tới các thành viên
* UC #04 - Quản lý TKB trường: Hệ thống cho phép phòng đạo tạo sử dụng chức năng để quản lý các chức năng trong hệ thống.
  + Tạo báo cáo TKB trường: phòng đạo tạo tổng hợp thời khóa biểu từ các bộ môn rồi tạo ra danh sách thời khóa biểu của trường
  + Chỉnh sửa TKB trường: phòng đào tạo chỉnh sửa thời khóa biểu nếu có sai sót và cập nhật các môn học, xóa môn học.
* UC #05 - Quản lý TKB khoa: chức năng này cho phép các trưởng bộ môn quản lý các chức năng như
  + Tạo báo cáo TKB khoa
  + Chỉnh sửa TKB khoa
* UC #06 - Quản lý phòng học: Hệ thống cho phép người dùng sử dụng chức năng để quản lý tình trạng phòng, địa điểm như phòng máy, phòng iot, phòng học thường, quản lý trạng thái bảo trì,…
* UC #07 - Quản lý môn học: Chức năng này cho phép người dùng có thể kiểm tra thông tin về môn học, tra cứu mã môn học,giáo viên giảng dạy và số tiết học
* UC #08 - Quản lý khối giờ học: chức năng cho phép tra cứu thông tin phòng học còn trống hay không trong khoảng giờ nhất định để đăng ký dạy.
* UC #09 - Quản lý ngành học: chức năng cho phép hiển thị các môn cần dạy và đăng ký dạy trong kỳ của từng ngành học.
* UC #010 - Phân công giảng dạy: chức năng cho phép các trưởng bộ môn phân công lịch dạy các môn dạy đến giáo viên trong kì.
* UC #011 - Xem TKB cá nhân: hệ thống cho phép giáo viên xem thời khóa biểu cá nhân theo ngày theo tuần theo kỳ.
* UC #012 - Xem TKB khoa: hệ thống cho phép giáo viên và trưởng bộ môn xem thời khóa biểu khoa nếu có sai sót báo cáo với từng bộ phận liên quan.
  + - 1. **Bảng ánh xạ yêu cầu nghiệp vụ với các Usecase**

|  |  |  |
| --- | --- | --- |
| **BR** | **Mô tả** | **UC** |
| BR1 | Chuẩn bị TKB |  |
| BR1.1 | Quản lý môn học |  |
| BR1.2 | Quản lý phòng học |  |
| BR1.3 | Quản lý giáo viên |  |
| BR2 | Xây dựng TKB |  |
| BR2.1 | Đăng nhập | UC #01 |
| BR2.2 | Quản lý tài khoản | UC #02 |
| BR2.3 | Phân quyền | UC #03 |
| BR2.4 | Quản lý TKB trường | UC #04 |
| BR2.5 | Quản lý TKB khoa | UC #05 |
| BR2.6 | Quản lý phòng học | UC #06 |
| BR2.7 | Quản lý môn học | UC #07 |
| BR2.8 | Quản lý khối giờ học | UC #08 |
| BR2.9 | Quản lý ngành học | UC #09 |
| BR2.10 | Phân công giảng dạy | UC #010 |
| BR3 | Xem TKB |  |
| BR3.1 | Xem TKB trường | UC #011 |
| BR3.2 | Xem TKB cá nhân | UC #012 |
| BR4 | Thống kê,báo cáo |  |

Bảng 1: Bảng ánh xạ người dùng đối với các User-case web

* + 1. **Mobile**
       1. **Sơ đồ user case**



Hình 2: Sơ đồ User-case mobile

* + - 1. **Các tác nhân của hệ thống**
* Phòng đào tạo: là người trực tiếp sử dụng hệ thống. Phòng Đào Tạo có chức năng to lớn nhất là tổng hợp thời khóa biểu của Bộ Môn sau đó sắp xếp thời khóa biểu cho toàn trường.
* Trưởng bộ môn: cũng là người trực tiếp sử dụng hệ thống nhưng chỉ dừng lại ở mức độ nhất định. Bộ Môn có chức năng phân công cho Giáo Viên giảng dạy môn học.
* Giáo viên: cũng là người trực tiếp sử dụng hệ thống. Giáo viên hang ngày phải lên kiểm tra thời khóa biểu của mình trên hệ thống để nắm bắt được thông tin giảng dạy trong ngày đó.
* Admin: là người có chức năng quản lý tài khoản của toàn bộ hệ thống
  + - 1. **Các usercase**
* UC #01- Login: người dùng đăng nhập để thực hiện các chức năng của hệ thống
* UC #02 - Xem TKB trường: hệ thống cho phép các thành viên xem thời khóa biểu của trường.
* UC #03 - Xem TKB khoa: hệ thống cho phép giáo viên và trưởng bộ môn xem thời khóa biểu khoa nếu có sai sót báo cáo với từng bộ phận liên quan.
* UC #04 - Xem TKB cá nhân: hệ thống cho phép giáo viên xem thời khóa biểu cá nhân theo ngày theo tuần theo kỳ
  + - 1. **Bảng ánh xạ yêu cầu nghiệp vụ với các Usecase**

|  |  |  |
| --- | --- | --- |
| **BR** | **Mô tả** | **UC** |
| BR1 | Login | UC #01 |
| BR2 | Xem TKB trường | UC #02 |
| BR3 | Xem TKB khoa | UC #03 |
| BR4 | Xem TKB cá nhân | UC #04 |

Bảng 2: Bảng ánh xạ yêu cầu nghiệp vụ đối với các Usecase mobile

* 1. **Kế hoạch thực hiện**

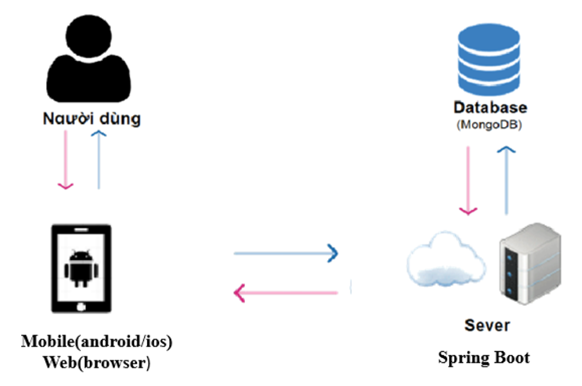
Thời gian thực hiện: 3 tháng (1/9/2019 – 30/11/2019)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Task Name |  | Duration | Start | Finish |
| 1 | Phân tích yêu cầu bài toán |  | 16 days | Sun 01/09/19 | Mon 16/09/19 |
| 2 | Thiết kế CSDL |  | 7 days | Tue 17/09/19 | Mon 23/09/19 |
| 3 | Thiết kế giao diện cho người dùng |  | 7 days | Tue 24/09/19 | Mon 30/09/19 |
| 4 | Thiết kế giao diện cho admin |  | 14 days | Tue 01/10/19 | Mon 14/10/19 |
| 5 | Lập trình |  | 42 days | Tue 15/10/19 | Mon 25/11/19 |
| 6 | Kiểm thử |  | 5 days | Tus 26/11/19 | Sat 30/11/19 |

Bảng 3: Kế hoạch thực hiện

# CHƯƠNG 2. KIẾN TRÚC TỔNG THỂ CỦA HỆ THỐNG

* 1. **Sơ đồ kiến trúc tổng thể**
     1. **Sơ đồ kiến trúc**



Hình 3: Sơ đồ kiến trúc tổng thể

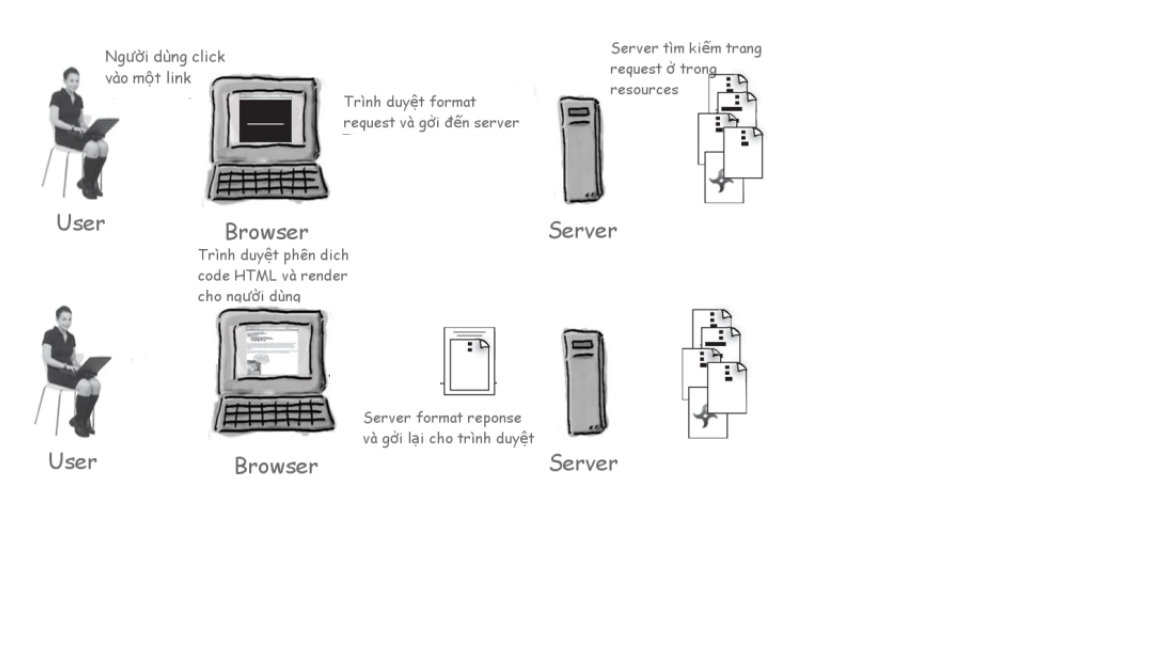
* Kiến trúc tổng thể sẽ gồm có Người dùng, Ứng dụng, Server và Database.
* Người dùng sẽ thông qua các ứng dụng (ios, android, browser) để thao tác với frontend để gửi các resquest đến server, server sẽ trả lại các response cho ứng dụng và trả hiển thị lên cho người dùng.
* Khi người dùng có các request thao tác với datasebase, frontend sẽ gửi request đến server và server sẽ thao tác với database theo request của người dùng.
  + 1. **Mô tả kiến trúc**

Hệ thống được thiết kế với 2 phần chính, bao gồm: Client-side và Server-side. Cụ thể như sau:

* Client-side: Ứng dụng Android / IOS / Browser
* Người dùng tương tác với ứng dụng.
* Ứng dụng trao đổi dữ liệu với server thông qua phương thức HTTP.
* Server-side: Sử dụng SpringBoot kết nối CSDL MongoDB
* Server trao đổi dữ liệu với ứng dụng Android thông qua phương thức HTTP
* Server sử dụng SpringBoot kết nối tới cơ sở dữ liệu MongoDB.
  + 1. **Công nghệ sử dụng**
       1. **JavaScript**
* JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ (small and lightweight). Khi nằm bên trong một môi trường (host environment), JavaScript có thể kết nối tới các object của môi trường đó và cung cấp các cách quản lý chúng (object).
* JavaScript chứa các thư viện tiêu chuẩn cho các object, ví dụ như: Array, Date, và Math, và các yếu tố cốt lõi của ngôn ngữ lập trình như: toán tử (operators), cấu trúc điều khiển (control structures), và câu lệnh. JavaScript có thể được mở rộng cho nhiều mục đích bằng việc bổ sung thêm các object; ví dụ:
* Client-side JavaScript - JavaScript phía máy khách, JavaScript được mở rộng bằng cách cung cấp các object để quản lý trình duyệt và Document Object Model (DOM) của nó. Ví dụ, phần mở rộng phía máy khách cho phép một ứng dụng tác động tới các yếu tố trên một trang HTML và phản hồi giống các tác động của người dùng như click chuột, nhập form, và chuyển trang.
* Server-side JavaScript - JavaScript phía Server, JavaScript được mở rộng bằng cách cung cấp thêm các đối tượng cần thiết để để chạy JavaScript trên máy chủ. Ví dụ, phần mở rộng phía server này cho phép ứng dụng kết nối với cơ sở dữ liệu (database), cung cấp thông tin một cách liên tục từ một yêu cầu tới phần khác của ứng dụng, hoặc thực hiện thao tác với các tập tin trên máy chủ.
* Trong ứng dụng, ta sẽ sử dụng Javascript để viết kịch bản ở phía frontend cho một trang web tĩnh. Javascript sẽ gửi các request đến của người sử dụng đến API và nhận lại các response từ phía server
  + - 1. **SpringBoot**
* Spring là framework phát triển ứng dụng phổ biến nhất dành cho Java Enterprise. Ban đầu nó được viết bởi Rod Johnson và lần đầu tiên được phát hành theo giấy phép Apache 2.0 vào tháng 6 năm 2003. Spring có kích thướng nhẹ, phiên bản cơ bản của Spring framework có kích thước khoảng 2MB.
* Spring framework là một Java Platform mã nguồn mở, một giải pháp gọn nhẹ dành cho Java Enterprise. Với Spring Framework các nhà phát triển có thể tạo ra các mã có hiệu suất cao, dễ kiểm thử và có thể sử dụng lại được.
* Các tính năng core của Spring Framework có thể được sử dụng trong việc phát triển bất kỳ ứng dụng Java nào. Bên cạnh đó, phần mở rộng được sử dụng để xây dựng các ứng dụng web trên nền tảng Java EE. Mục tiêu của Spring Framework là làm cho việc phát triển ứng dụng J2EE dễ dàng hơn và thúc đẩy việc lập trình tốt hơn bằng mô hình POJO-based.
* Dưới đây là các ưu điểm của Spring Boot:
* Nó rất dễ dàng để phát triển các ứng dụng dựa trên Spring với Java hoặc Groovy.
* Nó giảm thiểu thời gian phát triển và tăng năng xuất
* Nó tránh việc phải viết nhiều mã nguyên mẫu (boilerplate Code), Annotations và các cấu hình XML.
* Nó dễ dàng để bạn tương tác các ứng dụng Spring Boot với các hệ sinh thái của Spring như Spring JDBC, Spring ORM, Spring Data, Spring Security etc.
* Nó đi theo cách tiếp cận “Nguyên tắc cấu hình mặc định” để giảm thiểu thời gian và nỗ lực cần thiết để phát triển ứng dụng.
* Nó cung cấp các Server nhúng (Embedded HTTP servers) như là Tomcat, Jetty.... để phát triển và test các ứng dụng web nhanh chóng và dễ dàng.
* Nó cung cấp công cụ CLI (Command Line Interface) dể phát triển và test các ứng dụng Spring Boot (Java hoặc Groovy) từ các dòng lệnh (command prompt) rất dễ dàng và nhanh chóng.
* Nó cung cấp rất nhiều các plugin để phát triển và test các ứng dụng Spring Boot nhanh chóng sử dụng các công cụ Build như Maven và Gradle
* Nó cung cấp nhiều plugin để làm việc với các cơ sở dữ liệu nhúng (embedded database) và các cơ sở dữ liệu lưu trữ trên bộ nhớ (in-memory Databases) một cách dễ dàng.
* Trong ứng dụng, Spring Boot sẽ được sử dụng để tạo ra một server gồm có các model, các interface để thao tác với cơ sở dữ liệu và cung cấp các API để các ứng dụng thao tác với server.
  + - 1. **MongoDB**
* MongoDB là một chương trình cơ sở dữ liệu mã nguồn mở được thiết kế theo kiểu hướng đối tượng trong đó các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt này nên MongoDB có thể được dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data).

- Ưu điểm của MongoDB:

* Document oriented
* Hiệu suất cao
* Tính sẵn sàng cao – Nhân rộng
* Khả năng mở rộng cao – Sharding
* Năng động – Không có lược đồ cứng nhắc.
* Linh hoạt – thêm / xóa trường có ít hoặc không ảnh hưởng đến ứng dụng
* Dữ liệu không đồng nhất
* Không joins
* Phân phối được
* Biểu diễn dữ liệu trong JSON hoặc BSON
* Tích hợp dễ dàng với BigData Hadoop
* Ngôn ngữ truy vấn dựa trên tài liệu mạnh mẽ như SQL
* Trong ứng dụng, MongoDB sẽ được sử dụng để làm database lưu trữ dữ liệu của các Model trong server.
* Database là nơi lưu trữ dữ liệu của toàn bộ hệ thống. Hệ thống sử dụng MongoDB. Database sẽ được cài trên cùng một máy với máy chủ chứa web service và sử dụng các kết nối nội bộ để thao tác.
  1. **Server**
* Server (máy chủ) là một chương trình máy tính hoặc một thiết bị cung cấp chức năng cho các chương trình hoặc các thiết bị khác được gọi là “Clients” (khách hàng), cấu trúc này được gọi là mô hình client - server. Một máy chủ có thể cung cấp nhiều chức năng khác nhau được gọi là “dịch vụ”, chẳng hạn như chia sẻ dữ liệu, thực hiện công việc tính toán cho một khách hàng… Một máy chủ có thể phục vụ nhiều khách hàng và một khách hàng có thể sử dụng nhiều máy chủ.
* Sử dụng laptop HP, cấu hình máy gồm:
  + Core(TM): i5−
  + CPU: 2.70GHz
  + RAM: 4GB
  + System type: 64-bit
  + Operating System: Windows 7
  1. **Web Client**
* Web client giúp người dùng gửi request đến server và hiển thị cho người dùng kết quả của request trả về



Hình 4: Sơ đồ web-client

* Quy trình thực hiện:
  + User (máy khách) sẽ gửi yêu cầu (request) đến Server (máy chủ, cung ứng các dịch vụ cho các máy khách) yêu cầu tài nguyên thông tin. Truy vấn này được gửi dưới định dạng HTTP, trong khi địa chỉ của nguồn cài nguyên yêu cầu được xác định trong định dạng URL.
  + Web server nhận được yêu cầu, và ánh xạ đến đường dẫn đến đúng tệp tin có thực trên máy chủ. Nếu trang yêu cầu được tìm thấy, nó phản hồi đến client và hiển thị nội dung lên trình duyệt web của client. Nếu trang web yêu cầu không tìm thấy, web server sẽ gửi phản hồi http response đến client với thông báo: “404 Not Found”.
* Ưu điểm:
  + Dữ liệu và tài nguyên được tập trung lại và tăng tính toàn vẹn của các dữ liệu.
  + Sự linh động trong việc mở rộng được hệ thống mạng.
  + Không phụ thuộc vào cùng một nền tảng, chỉ cần chung một định dạng giao tiếp (protocol) là có thể hoạt động được.
  1. **Cấu hình sử dụng**

|  |  |
| --- | --- |
| Hạng mục | Cấu hình tối thiểu |
| Máy sever | * Cpu: intel core i5, 2.3 Ghz * Ram: 4 gb * Ổ cứng: ssd, 250gb * Cấu hình: 64bits |
| Máy client | * Intel pen III trở lên |
|  | * RAM 128 trở lên Máy Sever * Hệ điều hành: Win 2000 Server/NT/2003 |

Bảng 4: Bảng hạng mục cấu hình tối thiểu sử dụng

# CHƯƠNG 3. ĐẶC TẢ CÁC CHỨC NĂNG

* 1. **Web**
     1. **UC #01 - Đăng nhập**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#01** | | **Đăng nhập** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng đăng nhập vào hệ thống | |
| **Tác nhân** | | Admin, Giáo viên, Trưởng bộ môn, Phòng đào tạo | |
| **Tiền điều kiện** | | Admin, Giáo viên, Trưởng bộ môn, Phòng đào tạo phải đăng nhập vào hệ thống | |
| **Hậu điều kiện** | **Thành công** | Đăng nhập thành công vào hệ thống | |
| **Lỗi** | Đăng nhập không thành công | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Use case bắt đầu khi người dùng muốn đăng nhập để sử dụng các chức năng của hệ thống   * Usecase này bắt đầu khi tác nhân muốn đăng nhập vào hệ thống. * Hệ thống yêu cầu tác nhân nhập tài khoản và mật khẩu đăng nhập. * Tác nhân nhập tài khoản và mật khẩu đăng nhập của mình. * Hệ thống xác nhận tài khoản và mật khẩu đăng nhập có hợp lệ không? Nếu không hợp lệ thì thực hiện luồng A. * Hệ thống ghi lại quá trình đăng nhập.   **Đăng xuất khỏi hệ thống:**  Chức năng này bắt đầu khi người dùng muốn rời khỏi hệ thống.   * Người dùng chọn đăng xuất trên trang website. * Hệ thống đăng xuất người dùng ra khỏi hệ thống.   Sau khi người dùng chọn đăng xuất. Hệ thống sẽ hiển thị ra trang đăng nhập. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Luồng A: Nhập sai tài khoản/ mật khẩu đăng nhập** | | | |
| * Hệ thống hiển thị thông báo lỗi * Người sử dụng có thể chọn hoặc là đăng nhập lại hoặc là hủy bỏ đăng nhập, khi đó usecase này sẽ kết thúc. | | | |
| **Luồng A1: Nhập thông tin không hợp lệ** | | | |
| * Hệ thống hiển thị thông báo “Thông tin không hợp lệ” và yêu cầu nhập lại thông tin. * Người sử dụng có thể chọn nhập lại thông tin hoặc hủy bỏ thay đổi mật khẩu, khi đó usecase này kết thúc. | | | |
| **Giao diện minh họa** | | | |
| Hình 5: Giao diện minh họa đăng nhập trên web | | | |

* + 1. **UC #02 - Quản lý tài khoản**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#02** | | **Quản lý tài khoản** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng quản lý tài khoản | |
| **Tác nhân** | | Admin | |
| **Tiền điều kiện** | | Admin phải đăng nhập vào hệ thống và chọn tài khoản | |
| **Hậu điều kiện** | **Thành công** | Hiển thị, thay đổi dữ liệu của tài khoản trong hệ thống | |
| **Lỗi** | Không hiển thị, thay đổi dữ liệu của tài khoản trong hệ thống | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng gồm các chức năng nhỏ hơn gồm: hiển thị chi tiết; sửa thông tin tài khoản.  1. Chức năng hiển thị chi tiết thông tin tài khoản.  1.1. Chức năng bắt đầu khi người dùng chọn quản lý tài khoản tại thanh công cụ.  1.2. Hệ thống hiển thị thông tin chi tiết của người dùng, thông tin chi tiết của thuộc tính trong bảng thuộc tính của đặc tả tương ứng:  1.3. Chức năng kết thúc khi người dùng chọn một chức năng khác.  2. Chức năng sửa thông tin tài khoản.  2.1. Chức năng bắt đầu khi người dùng chọn sửa ở chức năng hiển thị chi tiết thông tin tài khoản.  2.2. Hệ thống hiển thị ra các ô với dữ liệu của thông tin cũ để người dùng sửa: Dữ liệu trong ô được miêu tả trong bảng thuộc tính của đặc tả tương ứng:  2.3. Người dùng nhập dữ liệu và chọn lưu.  2.4. Hệ thống sửa dữ liệu.  2.5. Chức năng kết thúc, hệ thống chuyển đến chức năng hiển thị chi tiết thông tin tài khoản. | | | |

* + 1. **UC #03 - Phân quyền**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#03** | | **Phân quyền** | **Độ phức tạp: cao** |
| **Mô tả** | | Cho phép người dùng phân quyền | |
| **Tác nhân** | | Admin | |
| **Tiền điều kiện** | | Admin phải đăng nhập vào hệ thống và chọn phân quyền | |
| **Hậu điều kiện** | **Thành công** | Hiển thị, thay đổi dữ liệu của phân quyền trong hệ thống | |
| **Lỗi** | Không hiển thị, thay đổi dữ liệu của phân quyền trong hệ thống | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng gồm các chức năng nhỏ hơn gồm: Hiển thị danh sách; tạo; chuyển trạng thái; hiển thị chi tiết; sửa; xóa quyền.  1. Chức năng hiển thị danh sách quyền.  1.1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý quyền  1.2. Hệ thống hiển thị danh sách 10 các quyền sắp xếp theo alphabet.  1.3. Người dùng có thể sử dụng bộ lọc phía trên danh sách để lọc dữ liệu theo ý mình. Thành phần bộ lọc được mô tả ở bảng bộ lọc trong đặc tả tương ứng:  1.4. Người dùng có thể chọn chuyển trang để hệ thống hiển thị trang tương ứng.  1.5. Chức năng kết thúc khi người dùng chọn một chức năng khác.  2. Chức năng thêm quyền  2.1. Chức năng bắt đầu khi người dùng chọn thêm quyền trong chức năng hiển thị danh sách quyền.  2.2. Hệ thống hiển thị ra các ô để người dùng nhập dữ liệu. Kiểu dữ liệu được định nghĩa trong bảng thuộc tính của đặc tả tương ứng:  2.3. Người dùng nhập dữ liệu và chọn lưu.  2.4. Hệ thống thêm mới dữ liệu.  2.5. Chức năng kết thúc, hệ thống chuyển đến chức năng hiển thị.  3. Chức năng thay đổi trạng thái quyền.  3.1. Chức năng bắt đầu khi người dùng chọn trạng thái ở chức năng hiển thị danh sách quyền. Thông tin trạng thái được hiển thị trong bảng trạng thái của đặc tả tương ứng.  3.2. Hệ thống hiển thị ra thông báo xác nhận việc đổi trạng thái.  3.3. Người dùng chọn “OK”, hệ thống sẽ thực hiện chuyển trạng thái quyền và tắt thông báo. Chức năng kết thúc.  3.4. Người dùng chọn “Hủy”, hệ thống tắt thông báo. Chức năng kết thúc.  4. Chức năng hiển thị chi tiết quyền.  4.1. Chức năng bắt đầu khi người dùng chọn quyền trong danh sách ở chức năng hiển thị danh sách quyền.  4.2. Hệ thống hiển thị thông tin chi tiết của quyền, thông tin chi tiết của thuộc tính trong bảng thuộc tính của đặc tả tương ứng:  4.3. Chức năng kết thúc khi người dùng chọn một chức năng khác.  5. Chức năng sửa quyền.  5.1. Chức năng bắt đầu khi người dùng chọn sửa ở chức năng hiển thị chi tiết quyền.  5.2. Hệ thống hiển thị ra các ô với dữ liệu của quyền cũ để người dùng sửa: Dữ liệu trong ô được miêu tả trong bảng thuộc tính của đặc tả tương ứng:  5.3. Người dùng nhập dữ liệu và chọn lưu.  5.4. Hệ thống sửa dữ liệu.  5.5. Chức năng kết thúc, hệ thống chuyển đến chức năng hiển thị chi tiết quyền.  6. Chức năng xóa quyền.  6.1. Chức năng bắt đầu khi người dùng chọn xóa ở chức năng hiển thị chi tiết quyền.  6.2. Hệ thống hiển thị ra thông báo xác nhận việc xóa.  6.3. Người dùng chọn “OK”, hệ thống sẽ thực hiện xóa quyền và chuyển đến chức năng hiển thị danh sách quyền.  6.4. Người dùng chọn “Hủy”, hệ thống tắt thông báo. | | | |

* + 1. **UC #04 - Quản lý TKB trường**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#04** | | **Quản lý TKB trường** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng quản lý TKB trường | |
| **Tác nhân** | | Phòng đào tạo | |
| **Tiền điều kiện** | | Cần đăng nhập với tài khoản có quyền truy cập chức năng trước khi thực hiện chức năng. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị, thay đổi dữ liệu của TKB trường có trong hệ thống. | |
| **Lỗi** | Không hiển thị hay thay đổi dữ liệu TKB trường trong hệ thống. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng gồm các chức năng nhỏ hơn gồm: Hiển thị, thêm, sửa, xóa TKB trường.   1. Chức năng hiển thị TKB trường    1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý TKB trường    2. Hệ thống hiển thị danh sách các môn học trong TKB trường    3. Chức năng kết thúc khi người dùng chọn một chức năng khác. 2. Chức năng thêm TKB trường    1. Chức năng băt đầu khi người dùng chọn import TKB trường trên thanh công cụ    2. Hệ thống hiển thị bảng đường dẫn để người dùng chọn file tương ứng để import dữ liệu    3. Người dùng chọn file import và chọn open    4. Hệ thống thêm mới dữ liệu    5. Chức năng kết thúc, hệ thống chuyển đến chức năng hiển thị 3. Chức năng sửa TKB trường    1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý TKB trường    2. Hệ thống hiển thị ra các ô dữ liệu của môn học cũ để người dùng sửa: Dữ liệu trong ô dược miêu tả trong bảng thuộc tính của đặc tả tương ứng    3. Người dùng nhập dữ liệu và chọn lưu    4. Hệ thống sửa dữ liệu    5. Chức năng kết thúc, hệ thống chuyển đến chức năng hiển thị TKB trường. 4. Chức năng xóa TKB trường    1. Chức năng bắt đầu khi người dùng chọn xóa ở chức năng hiển thị TKB trường    2. Hệ thống hiển thị ra thông báo xác nhận việc xóa.    3. Người dùng chọn “OK”, hệ thống sẽ thực hiện môn học và chuyển đến chức   năng hiển thị danh sách TKB trường   * 1. Người dùng chọn “Hủy”, hệ thống tắt thông báo. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Chuyển chức năng trước khi hoàn thành sửa.** | | | |
| 1. Người dùng chuyển chức năng trước khi thực hiện thành công việc sửa TKB trường.  2. Hệ thống chuyển sang chức năng được chỉ định mà không thay đổi trạng thái TKB trường. Chức năng kết thúc | | | |
| **Nhập sai điều kiện dữ liệu khi thực hiện sửa.** | | | |
| 1. Người dùng thực hiện lưu thông tin. 2. Hệ thống hiển thị lại chức năng với các cảnh báo ở dưới ô nhập có dữ liệu sai tương ứng.   Người dùng sửa lại dữ liệu và tiếp tục chức năng. | | | |
| **Giao diện minh họa** | | | |
| Hình 6: Giao diện minh họa TKB trường | | | |

* + 1. **UC #05 - Quản lý TKB khoa**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#05** | | **Quản lý TKB khoa** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng quản lý TKB khoa | |
| **Tác nhân** | | Phòng đào tạo | |
| **Tiền điều kiện** | | Cần đăng nhập với tài khoản có quyền truy cập chức năng trước khi thực hiện chức năng. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị, thay đổi dữ liệu của TKB khoa trong hệ thống | |
| **Lỗi** | Không hiển thị, thay đổi dữ liệu của TKB khoa trong hệ thống | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng gồm các chức năng nhỏ hơn gồm: Hiển thị, thêm, sửa, xóa TKB khoa.   1. Chức năng hiển thị TKB khoa    1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý TKB khoa    2. Hệ thống hiển thị danh sách các môn học trong TKB khoa    3. Chức năng kết thúc khi người dùng chọn một chức năng khác. 2. Chức năng thêm TKB khoa    1. Chức năng băt đầu khi người dùng chọn import TKB khoa trên thanh công cụ    2. Hệ thống hiển thị bảng đường dẫn để người dùng chọn file tương ứng để import dữ liệu    3. Người dùng chọn file import và chọn open    4. Hệ thống thêm mới dữ liệu    5. Chức năng kết thúc, hệ thống chuyển đến chức năng hiển thị 3. Chức năng sửa TKB khoa    1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý TKB khoa    2. Hệ thống hiển thị ra các ô dữ liệu của môn học cũ để người dùng sửa: Dữ liệu trong ô dược miêu tả trong bảng thuộc tính của đặc tả tương ứng    3. Người dùng nhập dữ liệu và chọn lưu    4. Hệ thống sửa dữ liệu    5. Chức năng kết thúc, hệ thống chuyển đến chức năng hiển thị TKB khoa. 4. Chức năng xóa TKB khoa    1. Chức năng bắt đầu khi người dùng chọn xóa ở chức năng hiển thị TKB khoa    2. Hệ thống hiển thị ra thông báo xác nhận việc xóa.    3. Người dùng chọn “OK”, hệ thống sẽ thực hiện môn học và chuyển đến chức năng hiển thị danh sách TKB khoa   Người dùng chọn “Hủy”, hệ thống tắt thông báo. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Chuyển chức năng trước khi hoàn thành sửa.** | | | |
| 1. Người dùng chuyển chức năng trước khi thực hiện thành công việc sửa TKB khoa.  2. Hệ thống chuyển sang chức năng được chỉ định mà không thay đổi trạng thái TKB khoa. Chức năng kết thúc | | | |
| **Nhập sai điều kiện dữ liệu khi thực hiện sửa.** | | | |
| 1. Người dùng thực hiện lưu thông tin. 2. Hệ thống hiển thị lại chức năng với các cảnh báo ở dưới ô nhập có dữ liệu sai tương ứng.   Người dùng sửa lại dữ liệu và tiếp tục chức năng. | | | |

* + 1. **UC #06 - Quản lý phòng học**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#06** | | **Quản lý phòng học** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng quản lý phòng học | |
| **Tác nhân** | | Phòng đào tạo | |
| **Tiền điều kiện** | | Cần đăng nhập với tài khoản có quyền truy cập chức năng trước khi thực hiện chức năng. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị, thay đổi dữ liệu của phòng học có trong hệ thống. | |
| **Lỗi** | Không hiển thị hay thay đổi dữ liệu phòng học trong hệ thống. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng gồm các chức năng nhỏ hơn gồm: Hiển thị, thêm, sửa, xóa phòng học.   1. Chức năng hiển thị phòng học    1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý phòng học    2. Hệ thống hiển thị danh sách các môn học trong phòng học    3. Chức năng kết thúc khi người dùng chọn một chức năng khác. 2. Chức năng thêm phòng học    1. Chức năng băt đầu khi người dùng chọn import phòng học trên thanh công cụ    2. Hệ thống hiển thị bảng đường dẫn để người dùng chọn file tương ứng để import dữ liệu    3. Người dùng chọn file import và chọn open    4. Hệ thống thêm mới dữ liệu    5. Chức năng kết thúc, hệ thống chuyển đến chức năng hiển thị 3. Chức năng sửa phòng học    1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý phòng học    2. Hệ thống hiển thị ra các ô dữ liệu của phòng học cũ để người dùng sửa: Dữ liệu trong ô dược miêu tả trong bảng thuộc tính của đặc tả tương ứng    3. Người dùng nhập dữ liệu và chọn lưu    4. Hệ thống sửa dữ liệu    5. Chức năng kết thúc, hệ thống chuyển đến chức năng hiển thị phòng học 4. Chức năng xóa phòng học    1. Chức năng bắt đầu khi người dùng chọn xóa ở chức năng hiển thị phòng học    2. Hệ thống hiển thị ra thông báo xác nhận việc xóa.    3. Người dùng chọn “OK”, hệ thống sẽ thực hiện xóa phòng học và chuyển đến chức năng hiển thị danh sách phòng học   Người dùng chọn “Hủy”, hệ thống tắt thông báo. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Chuyển chức năng trước khi hoàn thành sửa.** | | | |
| 1. Người dùng chuyển chức năng trước khi thực hiện thành công việc sửa TKB trường.  2. Hệ thống chuyển sang chức năng được chỉ định mà không thay đổi trạng thái TKB trường. Chức năng kết thúc | | | |
| **Nhập sai điều kiện dữ liệu khi thực hiện sửa.** | | | |
| 1. Người dùng thực hiện lưu thông tin.  2. Hệ thống hiển thị lại chức năng với các cảnh báo ở dưới ô nhập có dữ liệu sai tương ứng.  Người dùng sửa lại dữ liệu và tiếp tục chức năng. | | | |
| **Giao diện minh họa** | | | |
| Hình 7: Giao diện minh họa | | | |

* + 1. **UC #07 - Quản lý môn học**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#07** | | **Quản lý môn học** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng quản lý môn học | |
| **Tác nhân** | | Phòng đào tạo | |
| **Tiền điều kiện** | | Cần đăng nhập với tài khoản có quyền truy cập chức năng trước khi thực hiện chức năng. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị, thay đổi dữ liệu của môn học có trong hệ thống. | |
| **Lỗi** | Không hiển thị hay thay đổi dữ liệu môn học trong hệ thống. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng gồm các chức năng nhỏ hơn gồm: Hiển thị, thêm, sửa, xóa phòng học.   1. Chức năng hiển thị học phần    1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý học phần    2. Hệ thống hiển thị danh sách các môn học trong học phần    3. Chức năng kết thúc khi người dùng chọn một chức năng khác. 2. Chức năng thêm học phần    1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý môn học    2. Hệ thống hiển thị biểu mẫu thêm học phần    3. Người dùng nhập dữ liệu và chọn save    4. Hệ thông thêm mới dữ liệu    5. Chức năng kết thúc, hệ thống chuyển đến chức năng quản lý học phần 3. Chức năng sửa học phần    1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý học phần    2. Hệ thống hiển thị ra các ô dữ liệu của học phần cũ để người dùng sửa: Dữ liệu trong ô dược miêu tả trong bảng thuộc tính của đặc tả tương ứng    3. Người dùng nhập dữ liệu và chọn lưu    4. Hệ thống sửa dữ liệu    5. Chức năng kết thúc, hệ thống chuyển đến chức năng hiển thị học phần 4. Chức năng xóa học phần    1. Chức năng bắt đầu khi người dùng chọn xóa ở chức năng hiển thị học phần    2. Hệ thống hiển thị ra thông báo xác nhận việc xóa.    3. Người dùng chọn “OK”, hệ thống sẽ thực hiện xóa học phầnvà chuyển đến chức năng hiển thị danh sách học phần   Người dùng chọn “Hủy”, hệ thống tắt thông báo. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Nhập sai điều kiện dữ liệu.** | | | |
| 1.Người dùng thực hiện lưu thông tin.  2.Hệ thống hiển thị lại chức năng với các cảnh báo ở dưới ô nhập có dữ liệu sai tương ứng.  Người dùng sửa lại dữ liệu và tiếp tục chức năng. | | | |
| **Giao diện minh họa** | | | |
| Hình 8: Giao diện minh họa quản lý môn học | | | |

* + 1. **UC #08 - Quản lý khối giờ học**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#08** | | **Quản lý khối giờ học** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng quản lý khối giờ học | |
| **Tác nhân** | | Phòng đào tạo | |
| **Tiền điều kiện** | | Cần đăng nhập với tài khoản có quyền truy cập chức năng trước khi thực hiện chức năng. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị, thay đổi dữ liệu của môn học có trong hệ thống. | |
| **Lỗi** | Không hiển thị hay thay đổi dữ liệu môn học trong hệ thống. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng gồm các chức năng nhỏ hơn gồm: Thêm, xóa giờ học   1. Chức năng thêm giờ học    1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý giờ học    2. Hệ thống hiển thị biểu mẫu thêm giờ học    3. Người dùng nhập dữ liệu và chọn save    4. Hệ thông thêm mới dữ liệu    5. Chức năng kết thúc, hệ thống chuyển đến chức năng quản lý giờ học 2. Chức năng xóa giờ học    1. Chức năng bắt đầu khi người dùng chọn xóa ở chức năng hiển thị giờ học    2. Hệ thống hiển thị ra thông báo xác nhận việc xóa.    3. Người dùng chọn “OK”, hệ thống sẽ thực hiện xóa giờ học và chuyển đến chức năng hiển thị danh sách giờ học | | | |

* + 1. **UC #09 - Quản lý ngành học**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#09** | | **Quản lý nghành học** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng quản lý ngành học | |
| **Tác nhân** | | Phòng đào tạo | |
| **Tiền điều kiện** | | Cần đăng nhập với tài khoản có quyền truy cập chức năng trước khi thực hiện chức năng. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị, thay đổi dữ liệu của môn học có trong hệ thống. | |
| **Lỗi** | Không hiển thị hay thay đổi dữ liệu môn học trong hệ thống. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| 1. Chức năng thêm ngành học    1. Chức năng bắt đầu khi người dùng truy cập vào chức năng quản lý ngành học    2. Hệ thống hiển thị biểu mẫu thêm ngành học    3. Người dùng nhập dữ liệu và chọn save    4. Hệ thông thêm mới dữ liệu    5. Chức năng kết thúc, hệ thống chuyển đến chức năng quản lý ngành học 2. Chức năng xóa ngành học    1. Chức năng bắt đầu khi người dùng chọn xóa ở chức năng hiển thị ngành học    2. Hệ thống hiển thị ra thông báo xác nhận việc xóa.    3. Người dùng chọn “OK”, hệ thống sẽ thực hiện xóa ngành học và chuyển đến chức năng hiển thị danh sách ngành học | | | |

* + 1. **UC #010 - Phân công giảng dạy**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#010** | | **Phân công giảng dạy** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng phân công giảng dạy | |
| **Tác nhân** | | Trưởng bộ môn | |
| **Tiền điều kiện** | | Cần đăng nhập với tài khoản có quyền truy cập chức năng trước khi thực hiện chức năng. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị, thay đổi dữ liệu của phân công giảng dạy trong hệ thống | |
| **Lỗi** | Không hiển thị, thay đổi dữ liệu của phân công giảng dạy trong hệ thống | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng gồm các chức năng nhỏ hơn gồm: hiển thị chi tiết; sửa thông tin tài khoản.  1. Chức năng hiển thị chi tiết thông tin phân công  1.1. Chức năng bắt đầu khi người dùng chọn phân công giảng dạy  1.2. Hệ thống hiển thị thông tin chi tiết phân công giảng dạy  1.3. Chức năng kết thúc khi người dùng chọn một chức năng khác.  2. Chức năng sửa phân công.  2.1. Chức năng bắt đầu khi người dùng chọn sửa ở chức năng hiển thị bảng phân công  2.2. Hệ thống hiển thị ra các ô với dữ liệu của thông tin cũ để người dùng sửa: Dữ liệu trong ô được miêu tả trong bảng thuộc tính của đặc tả tương ứng:  2.3. Người dùng nhập dữ liệu và chọn lưu.  2.4. Hệ thống sửa dữ liệu.  2.5. Chức năng kết thúc, hệ thống chuyển đến chức năng hiển thị bảng phân công giảng dạy | | | |

* + 1. **UC #011 - Xem TKB cá nhân**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#011** | | **Xem TKB cá nhân** | **Độ phức tạp: thấp** |
| **Mô tả** | | Cho phép người dùng xem TKB cá nhân | |
| **Tác nhân** | | Giáo viên | |
| **Tiền điều kiện** | | Cần đăng nhập với tài khoản có quyền truy cập chức năng trước khi thực hiện chức năng. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị dữ liệu của TKB cá nhân có trong hệ thống | |
| **Lỗi** | Không hiển thị dữ liệu TKB cá nhân trong hệ thống | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Hiển thị danh sách TKB cá nhân có trong hệ thống | | | |

* + 1. **UC #012 - Xem TKB khoa**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#012** | | **Xem TKB khoa** | **Độ phức tạp: thấp** |
| **Mô tả** | | Cho phép người dùng xem TKB khoa | |
| **Tác nhân** | | Giáo viên, Trưởng bộ môn, Phòng đào tạo | |
| **Tiền điều kiện** | | Cần đăng nhập với tài khoản có quyền truy cập chức năng trước khi thực hiện chức năng. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị dữ liệu của TKB khoa có trong hệ thống. | |
| **Lỗi** | Không hiển thị dữ liệu TKB khoa trong hệ thống | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Hiển thị danh sách TKB khoa có trong hệ thống | | | |

* 1. **Mobile**
     1. **UC #01 - Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #01** | | **Login** | **Độ phức tạp: Normal** |
| **Mô tả** | | App cho phép các Giáo viên, Phòng đào tạo, Trưởng bộ môn cũng như quản lý người dùng thông qua tên đăng nhập và mật khẩu được cung cấp. Sau khi đăng nhập thành công, có thể sử dụng các chức năng được cho phép. | |
| **Tác nhân** | | Giáo viên, Phòng đào tạo, Quản lý người dùng(admin), Trưởng bộ môn | |
| **Tiền điều kiện** | | Admin(Quản lý người dùng)- được cấp tài khoản đăng nhập vào hệ thống và quản lý tài khoản  Nhũng thành viên còn lại được admin cấp cho 1 tài khoản. | |
| **Hậu điều kiện** | **Thành công** | Sau khi đăng nhập thành công, giảng viên, phòng đào tạo chỉ được sử dụng các chức năng được cho phép: xem thời khóa biểu cá nhân, xem thời khóa biểu toàn trường. | |
| **Lỗi** | Hiển thị thông báo dữ liệu nhập vào chưa đúng,thực hiện lại việc đăng nhập hoặc hủy bỏ thao tác | |
| ĐẶC TẢ CHỨC NĂNG | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| ***Luồng chính:***  Chức năng này bắt đầu khi Phòng đào tạo, Giáo viên, Trưởng bộ môn muốn vào trong và sử dụng hệ thống.  1. Chọn đăng nhập trên trang web của hệ thống.  2. Hệ thống yêu cầu nhập đầy đủ các thông tin:   * Tên đăng nhập; * Mật khẩu.   3. Sau khi Giáo viên, Phòng đào tạo, Trưởng nhập các thông tin và chọn nút đăng nhập, hệ thống sẽ kiểm tra thông tin nhập vào với cơ sở dữ liệu và thông báo kết quả.  3.1. Nếu khớp với cơ sở dữ liệu thì hệ thống sẽ hiện thông báo đăng nhập thành công.  3.2. Nếu không khớp với cơ sở dữ liệu thì hệ thống sẽ thông báo để Giáo viên, Phòng đào tạo, Trưởng bộ môn có thể nhập lại. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Nhập thiếu thông tin** | | | |
| Khi Giảng viên, Phòng đào tạo, Trưởng bộ môn nhập thiếu các thông tin được yêu cầu thì hệ thống sẽ thông báo để Giảng viên, Phòng đào tạo điền đầy đủ. | | | |
| **Các yêu cầu đặc biệt khác** | | | |
| Hệ Thống không cho phép tài khoản người dùng đăng nhập ở hai thiết bị khác nhau.  Nếu có từ hai thiết bị trở lên đăng nhập vào hệ thống thì thiết bị đăng nhập sau cùng được phép truy cập vào hệ thống. Đồng thời, tài khoản trên các thiết bị khác bị đăng xuất. | | | |
| **Giao diện minh họa** | | | |
| Hình 9: Giao diện đăng nhập trên mobile | | | |
| THIẾT KẾ UML | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
| ClassLogin  Hình 10: Sơ đồ lớp phân tích login trên mobile | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
| **C:\Users\VanMinh\Desktop\project\mobile1.png**  Hình 11: Sơ đồ trình tự login trên mobile | | | |

* + 1. **UC #02 - Xem TKB toàn trường**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #02** | | **Xem TKB toàn trường** | **Độ phức tạp: Normal** |
| **Mô tả** | | Hệ thống cho phép Giáo viên, Phòng đào tạo, Trưởng bộ môn xem TKB toàn trường. | |
| **Tác nhân** | | Giáo viên, Phòng đào tạo, Trưởng bộ môn | |
| **Tiền điều kiện** | | Giáo viên, Phòng đào tạo phải đăng nhập vào hệ thống | |
| **Hậu điều kiện** | **Thành công** | Sau khi đăng nhập thành công Giáo viên, Phòng đào tạo, Trưởng bộ môn xem TKB toàn trường, không được phép chỉnh sửa, thêm, sửa, xóa TKB toàn trường | |
| **Lỗi** | Không hiển thị thời khóa biểu toàn trường | |
| ĐẶC TẢ CHỨC NĂNG | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| ***Luồng chính:*** Xem TKB toàn trường  Chức năng này bắt đầu khi Giáo viên, Phòng đào tạo, Trưởng bộ môn muốn xem thông tin TKB toàn trường trong hệ thống.  1. Giáo viên chọn chức năng xem TKB toàn trường trong hệ thống.  2. Hệ thống hiển thị danh sách gồm có:   * Mã môn * Tên môn * Giáo viên dạy * Số tín chỉ * Phòng học * Ca học * Thứ * STT | | | |
| **Giao diện minh họa** | | | |
| Giao diện 2.1. Hiển thị giao diện  . | | | |
| THIẾT KẾ UML | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
| Hình 12: Sơ đồ lớp phân tích xem TKB trường trên mobile | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
| Hình 13: Sơ đồ trình tự xem TKB trường trên mobile | | | |

* + 1. **UC #03 - Xem TKB khoa**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #03** | | **Xem TKB khoa** | **Độ phức tạp: Normal** |
| **Mô tả** | | Hệ thống cho phép Giáo viên, Phòng đào tạo, Trưởng bộ môn xem TKB khoa | |
| **Tác nhân** | | Giáo viên, Phòng đào tạo, Trưởng bộ môn | |
| **Tiền điều kiện** | | Giáo viên, Phòng đào tạo phải đăng nhập vào hệ thống | |
| **Hậu điều kiện** | **Thành công** | Sau khi đăng nhập thành công Giáo viên, Phòng đào tạo, Trưởng bộ môn xem TKB khoa, không được phép chỉnh sửa, thêm, sửa, xóa TKB khoa | |
| **Lỗi** | Không hiển thị thời khóa biểu khoa | |
| ĐẶC TẢ CHỨC NĂNG | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| ***Luồng chính:*** Xem TKB khoa  Chức năng này bắt đầu khi Giáo viên, Phòng đào tạo, Trưởng bộ môn muốn xem thông tin TKB khoa trong hệ thống.  1. Giáo viên chọn chức năng xem TKB khoa trong hệ thống.  2. Hệ thống hiển thị danh sách gồm có:   * Mã môn * Tên môn * Giáo viên dạy * Phòng học * Ca học * Thứ/ngày/tháng/học kì | | | |
| THIẾT KẾ UML | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
| Hình 14: Sơ đồ lớp phân tích xem TKB khoa trên mobile | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
| Hình 15: Sơ đồ trình tự xem TKB khoa trên mobile | | | |

* + 1. **UC #04 Xem TKB cá nhân**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #04** | | **Xem TKB cá nhân** | **Độ phức tạp: Normal** |
| **Mô tả** | | Hệ thống cho phép Giáo viên xem TKB cá nhân của riêng Giáo viên đó. Mỗi giáo viên có một Mã giáo viên riêng. | |
| **Tác nhân** | | Giáo viên | |
| **Tiền điều kiện** | | Giáo viên phải đăng nhập vào hệ thống | |
| **Hậu điều kiện** | **Thành công** | Sau khi đăng nhập thành công Giáo viên chỉ được xem thời khóa biểu cá nhân, không được phép chỉnh sửa, thêm, sửa, xóa TKB… | |
| **Lỗi** | Không hiển thị thời khóa biểu cá nhân | |
| ĐẶC TẢ CHỨC NĂNG | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| ***Luồng chính:*** Xem TKB cá nhân  Chức năng này bắt đầu khi Giáo viên muốn xem thông tin TKB cá nhân trong hệ thống.  1. Giáo viên chọn chức năng xem TKB cá nhân trong hệ thống.  2. Hệ thống hiển thị danh sách các ngày dạy trong tuần gồm có:   * Thứ * Ngày * Tên môn học | | | |
| Hình 16: Giao diện TKB cá nhân trên mobile | | | |
| THIẾT KẾ UML | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
| Hình 17: Sơ đồ lớp phân tích xem TKB cá nhân trên mobile | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
| Hình 18: Sơ đồ trình tự xem TKB cá nhân trên mobile | | | |

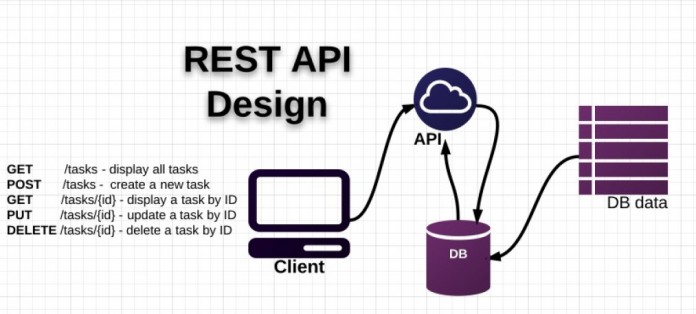
# CHƯƠNG 4: THIẾT KẾ API VÀ CƠ SỞ DỮ LIỆU

* 1. **Tổng thể**
* Rest API với 2 thành phần chính là Controller (thao tác với resource) và Model. Phần View được lược bỏ. Lúc này, phần Controller chỉ trả về các dữ liệu Json hoặc Xml.

**

Hình 19: Sơ đồ tổng thể thiết kế API

* Hệ thống hoạt động theo mô hình client-server, trong đó server là tập hợp các server nhỏ lắng nghe các request từ client. Với từng các request khác nhau thì có thể có một hoặc nhiều service xử lý.
* Stateless (phi trạng thái). Đơn giản server và client không lưu trạng thái của nhau -> mỗi request lên server thì client phải đóng gói thông tin đầy đủ để thằng server hiểu được. Điều này giúp hệ thống của bạn dễ phát triển, bảo trì, mở rộng vì không cần tốn công CRUD trạng thái của client. Hệ thống phát triển theo hướng này có ưu điểm nhưng cũng có khuyết điểm là gia tăng lượng thông tin cần truyền tải giữa client và server.
* Khả năng caching: Các response có thể lấy ra từ cache. Bằng cách cache các response, server giảm tải việc xử lý request, còn client cũng nhận được thông tin nhanh hơn. Ở đây ta đặt 1 thằng cache vào giữa: client- cache- server.
* Chuẩn hóa các interface: Đây là một trong những đặc tính quan trọng của hệ thống REST. Bằng cách tạo ra các quy ước chuẩn để giao tiếp giữa các thành phần trong hệ thống, đơn giản hóa việc client có thể tương tác với server. Các quy ước này áp dụng cho toàn bộ các service giúp cho người sử dụng hệ thống của bạn dễ dụng hơn. Dễ hiểu hơn trên hệ thống đặt ra 1 chuẩn API để người dùng dù là mobile, web đều có thể kết nối vào được. Hệ thống REST có yếu điểm ở đây vì khi chuẩn hóa rồi ta không thế tối ưu từng kết nối.
* Phân lớp hệ thống: trong hệ thống REST chia tách các thành phần hệ thống theo từng lớp, mỗi lớp chỉ sử dụng lớp ở dưới nó và giao tiếp với lớp ở ngay trên nó mà thôi. Điều này giúp giảm độ phức tạp của hệ thống, giúp các thành phần tách biệt nhau từ đó dễ dàng mở rộng từng thành phần.
* Controller được chia thành các Action khác nhau. Mỗi Action thực hiện môn nghiệp vụ duy nhất trong nhóm nghiệp vụ được bao bọc bởi Controller.



Hình 20: Các phương thức sử dụng trong API

* Có 4 kiểu Request chính là: GET; POST; PUT, DELETE và tham số đường dẫn truyền vào.
* Ngoài các API quản lý các resource cơ bản, hệ thống sẽ có API để tạo token xác thực tài khoản
  1. **Cấu trúc chung của một API**

Cấu trúc chung của một API sẽ gồm có:

* Phương thức truy cập API: Có 4 phương thức sau:
* GET để truy vấn dữ liệu
* POST để tạo mới dữ liệu
* PUT để thay thế hoặc cập nhật một dữ liệu
* DELETE để loại bỏ một dữ liệu
* Truy cập API resource: Gồm URL có hoặc không có tham số đầu vào để truy cập đến địa chỉ website
* Return type: Kiểu dữ liệu trả về là một tập tin dạng JSON
  1. **Các API của hệ thống**
     1. **API dùng để quản lý các lớp học (/api/class)**

| Url | Chi tiết |
| --- | --- |
| /api/class | * Mục đích: Lấy danh sách lớp học trong hệ thống. * Phương thức: GET. * Tham số:   + page (url) – kiểu int, mặc định 1   + take (url) – kiểu int, mặc định 10   + text (url) – kiểu string, độ dài 128 ký tự, không bắt buộc |
| * Mục đích: Tạo lớp học mới trong hệ thống. * Phương thức: POST * Tham số:   + FullName (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Gender (body) – kiểu bool, không bắt buộc.   + DateOfBrith (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Avatar (body) – kiểu string, độ dài 256 ký tự, bắt buộc.   + Introduction (body) – kiểu string, độ dài 2048, bắt buộc |
| /api/class/{id} | * Mục đích: Sửa lớp trong hệ thống. * Phương thức: PUT. * Tham số:   + id (url) – kiểu int, bắt buộc   + FullName (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Gender (body) – kiểu bool, không bắt buộc.   + DateOfBrith (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Avatar (body) – kiểu string, độ dài 256 ký tự, bắt buộc.   Introduction (body) – kiểu string, độ dài 2048, bắt buộc. |
| * Mục đích: Xóa thông tin lớp học trong hệ thống. * Phương thức: DELETE. * Tham số:   + id (url) – kiểu int, bắt buộc |

* + 1. **API dùng để quản lý các phòng học (/api/room)**

| Url | Chi tiết |
| --- | --- |
| /api/rooms | * Mục đích: Lấy danh sách phòng học và tình trạng phòng học trong hệ thống. * Phương thức: GET. * Tham số:   + page (url) – kiểu int, mặc định 1   + take (url) – kiểu int, mặc định 10   + text (url) – kiểu string, độ dài 128 ký tự, không bắt buộc |
| * Mục đích: Cập nhật các trạng thái phòng học trong hệ thống. * Phương thức: POST * Tham số:   + Code (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Large (body) – kiểu float, bắt buộc.   + Capacity (body) – kiểu int, bắt buộc.   + Description (body) – kiểu string, độ dài 1024 ký tự, bắt buộc. |
| /api/rooms/{id} | * Mục đích: Lấy thông tin phòng học theo id trong hệ thống. * Phương thức: GET. * Tham số:   + id (url) – kiểu int, bắt buộc |
| * Mục đích: Sửa thông tin trạng thái phòng theo id trong hệ thống. * Phương thức: PUT. * Tham số:   + id (url) – kiểu int, bắt buộc   + Code (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Large (body) – kiểu float, bắt buộc.   + Capacity (body) – kiểu int, bắt buộc.   + Description (body) – kiểu string, độ dài 1024 ký tự, bắt buộc. |
| * Mục đích: Xóa thông tin phòng học theo id trong hệ thống. * Phương thức: DELETE. * Tham số:   + id (url) – kiểu int, bắt buộc |

* + 1. **API dùng để quản lý các giáo viên (/api/teacher)**

| Url | Chi tiết |
| --- | --- |
| /api/teacher | * Mục đích: Lấy danh sách giáo viên trong hệ thống. * Phương thức: GET. * Tham số:   + page (url) – kiểu int, mặc định 1   + take (url) – kiểu int, mặc định 10   + text (url) – kiểu string, độ dài 128 ký tự, không bắt buộc |
| * Mục đích: Tạo giáo viên mới trong hệ thống. * Phương thức: POST * Tham số:   + FullName (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Gender (body) – kiểu bool, không bắt buộc.   + DateOfBrith (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Avatar (body) – kiểu string, độ dài 256 ký tự, bắt buộc.   + Introduction (body) – kiểu string, độ dài 2048, bắt buộc |
| /api/teacher/{id} | * Mục đích: Sửa giáo viên trong hệ thống. * Phương thức: PUT. * Tham số:   + id (url) – kiểu int, bắt buộc   + FullName (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Gender (body) – kiểu bool, không bắt buộc.   + DateOfBrith (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Avatar (body) – kiểu string, độ dài 256 ký tự, bắt buộc.   Introduction (body) – kiểu string, độ dài 2048, bắt buộc. |
| * Mục đích: Xóa thông tin giáo viên trong hệ thống. * Phương thức: DELETE. * Tham số:   + id (url) – kiểu int, bắt buộc |

* + 1. **API dùng để quản lý các môn học (/api/subject)**

| Url | Chi tiết |
| --- | --- |
| /api/subject | * Mục đích: Lấy danh sách môn học trong hệ thống. * Phương thức: GET. * Tham số:   + page (url) – kiểu int, mặc định 1   + take (url) – kiểu int, mặc định 10   + text (url) – kiểu string, độ dài 128 ký tự, không bắt buộc |
| * Mục đích: Tạo môn học mới trong hệ thống. * Phương thức: POST * Tham số:   + FullName (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Gender (body) – kiểu bool, không bắt buộc.   + DateOfBrith (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Avatar (body) – kiểu string, độ dài 256 ký tự, bắt buộc.   + Introduction (body) – kiểu string, độ dài 2048, bắt buộc |
| /api/subject/{id} | * Mục đích: Sửa môn học trong hệ thống. * Phương thức: PUT. * Tham số:   + id (url) – kiểu int, bắt buộc   + FullName (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Gender (body) – kiểu bool, không bắt buộc.   + DateOfBrith (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Avatar (body) – kiểu string, độ dài 256 ký tự, bắt buộc.   Introduction (body) – kiểu string, độ dài 2048, bắt buộc. |
| * Mục đích: Xóa thông tin môn học trong hệ thống. * Phương thức: DELETE. * Tham số:   + id (url) – kiểu int, bắt buộc |

* + 1. **API dùng để quản lý các học kỳ (/api/semester)**

| Url | Chi tiết |
| --- | --- |
| /api/semeter | * Mục đích: Lấy danh sách các học kỳ trong hệ thống. * Phương thức: GET. * Tham số:   + page (url) – kiểu int, mặc định 1   + take (url) – kiểu int, mặc định 10   + text (url) – kiểu string, độ dài 128 ký tự, không bắt buộc |
| * Mục đích: Tạo mới học kỳ trong hệ thống. * Phương thức: POST * Tham số:   + Name (body) – kiểu string, độ dài 128 ký tự, bắt buộc. |
| /api/semeter/{id} | * Mục đích: Lấy thông tin các học kỳ theo id trong hệ thống. * Phương thức: GET. * Tham số:   + id (url) – kiểu int, bắt buộc |
| * Mục đích: Sửa thông tin học kỳ theo id trong hệ thống. * Phương thức: PUT. * Tham số:   + id (url) – kiểu int, bắt buộc   + Name (body) – kiểu string, độ dài 128 ký tự, bắt buộc. |
| * Mục đích: Xóa thông học kỳ theo id trong hệ thống. * Phương thức: DELETE. * Tham số:   + id (url) – kiểu int, bắt buộc |

* + 1. **API dùng để quản lý các ca học (/api/shift)**

| Url | Chi tiết |
| --- | --- |
| /api/shift | * Mục đích: Lấy danh sách ca học trong hệ thống. * Phương thức: GET. * Tham số:   + page (url) – kiểu int, mặc định 1   + take (url) – kiểu int, mặc định 10   + text (url) – kiểu string, độ dài 128 ký tự, không bắt buộc |
| * Mục đích: Tạo ca học mới trong hệ thống. * Phương thức: POST * Tham số:   + FullName (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Gender (body) – kiểu bool, không bắt buộc.   + DateOfBrith (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Avatar (body) – kiểu string, độ dài 256 ký tự, bắt buộc.   + Introduction (body) – kiểu string, độ dài 2048, bắt buộc |
| /api/shift/{id}  Url | * Mục đích: Sửa thông tin ca học theo id trong hệ thống. * Phương thức: PUT. * Tham số:   + id (url) – kiểu int, bắt buộc   + FullName (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Gender (body) – kiểu bool, không bắt buộc.   + DateOfBrith (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Avatar (body) – kiểu string, độ dài 256 ký tự, bắt buộc.   Introduction (body) – kiểu string, độ dài 2048, bắt buộc. |
| * Mục đích: Xóa thông tin ca học trong hệ thống. * Phương thức: DELETE. * Tham số:   + id (url) – kiểu int, bắt buộc |
| * + Chi tiết |

* + 1. **API dùng để quản lý các bộ môn (/api/group-subject)**

| Url | Chi tiết |
| --- | --- |
| /api/group-subject | * Mục đích: Lấy danh sách các bộ môn trong hệ thống. * Phương thức: GET. * Tham số:   + page (url) – kiểu int, mặc định 1   + take (url) – kiểu int, mặc định 10   + text (url) – kiểu string, độ dài 128 ký tự, không bắt buộc |
| * Mục đích: Tạo mới bộ môn trong hệ thống. * Phương thức: POST * Tham số:   + Name (body) – kiểu string, độ dài 128 ký tự, bắt buộc. |
| /api/group-subject/{id} | * Mục đích: Lấy thông tin các bộ môn theo id trong hệ thống. * Phương thức: GET. * Tham số:   + id (url) – kiểu int, bắt buộc |
| * Mục đích: Sửa thông tin bộ môn theo id trong hệ thống. * Phương thức: PUT. * Tham số:   + id (url) – kiểu int, bắt buộc   + Name (body) – kiểu string, độ dài 128 ký tự, bắt buộc. |
| * Mục đích: Xóa thông học kỳ theo id trong hệ thống. * Phương thức: DELETE. * Tham số:   + id (url) – kiểu int, bắt buộc |

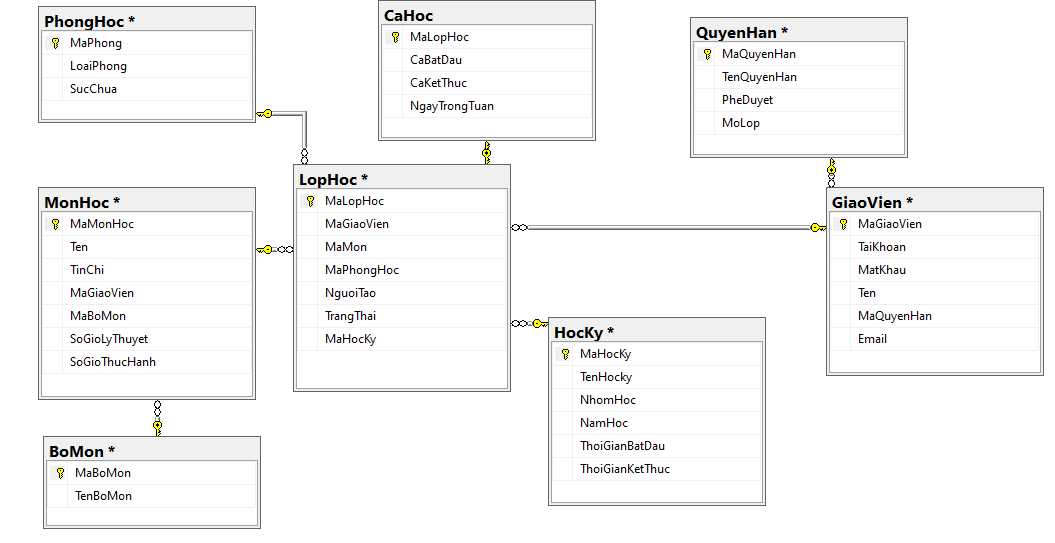
* + 1. **API dùng để quản lý quyền hạn của các giáo viên (/api/role)**

| Url | Chi tiết |
| --- | --- |
| /api/role | * Mục đích: Lấy danh sách phân trang quyền trong hệ thống. * Phương thức: GET. * Tham số:   + Page (url) – kiểu int, mặc định 1.   + Take (url) – kiểu int, mặc định 10.   + Text (url) - kiểu string, độ dài 128 ký tự |
| * Mục đích: Tạo mới quyền trong hệ thống. * Phương thức: POST * Tham số:   + Name (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Description (body) – kiểu string, độ dài 1024 ký tự, bắt buộc.   + JsonActions (body) – kiểu json, bắt buộc. |
| /api/role/{id} | * Mục đích: Lấy thông tin quyền trong hệ thống. * Phương thức: GET. * Tham số:   + id (url) – kiểu int, bắt buộc |
| * Mục đích: Sửa quyền trong hệ thống. * Phương thức: PUT. * Tham số:   + id (url) – kiểu int, bắt buộc   + Name (body) – kiểu string, độ dài 128 ký tự, bắt buộc.   + Description (body) – kiểu string, độ dài 1024 ký tự, bắt buộc.   + JsonActions (body) – kiểu json, bắt buộc. |
| * Mục đích: Xóa thông tin bộ môn trong hệ thống hệ thống. * Phương thức: DELETE. * Tham số:   + id (url) – kiểu int, bắt buộc |

* + 1. **API dùng để tạo token xác thực tài khoản (/api/class/login)**

| Url | Chi tiết |
| --- | --- |
| /api/class/login/{id} | * Mục đích: Lấy token để xác thực cho tài khoản * Phương thức: GET. * Tham số:   + Page (url) – kiểu int, mặc định 1.   + Take (url) – kiểu int, mặc định 10.   + Text (url) - kiểu string, độ dài 128 ký tự |

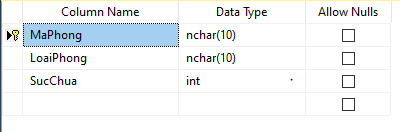
* 1. **Kiến trúc cơ sở dữ liệu**



Hình 21: Cơ sở dữ liệu

Trong cơ sở dữ liệu của nhóm em có 8 bảng. Các bảng đó là: PhongHoc, MonHoc, BoMon, CaHoc, LopHoc, HocKy, QuyenHan, GiaoVien. Để thực hiện yêu cầu của bài toán sắp xếp thời khóa biểu cho phần mềm và lưu trữ dữ liệu của bài toán.

* + 1. **Ý nghĩa các bảng cơ sở dữ liệu**
    - Bảng Phòng học

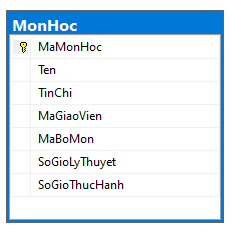
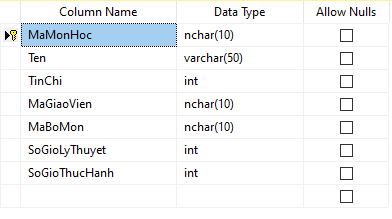


Bảng 5: Cơ sở dữ liệu phòng học

|  |  |  |
| --- | --- | --- |
| STT | Tên | Ý nghĩa |
| 1 | MaPhong | Mã phòng học dùng để phân biệt các phòng học với nhau. |
| 2 | LoaiPhong | Loại phòng của phòng học |
| 3 | SucChua | Sức chứa tối đa của phòng học |

Bảng 6: Ý nghĩa cơ sở dữ liệu phòng học

* + - Bảng Môn học

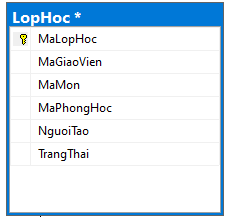
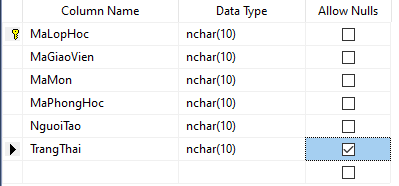


Bảng 7: Cơ sở dữ liệu môn học

|  |  |  |
| --- | --- | --- |
| STT | Tên | Ý nghĩa |
| 1 | MaMonHoc | Mã môn học dùng để phân biệt các môn học với nhau |
| 2 | Ten | Tên của môn học |
| 3 | TinChi | Số tín chỉ của môn học |
| 4 | MaGiaoVien | Mã giáo viên dạy môn học |
| 5 | MaBoMon | Mã bộ môn của môn học |
| 6 | SoGioLyThuyet | Số giờ lý thuyết của môn học |
| 7 | SoGioThucHanh | Số giờ thực hành của môn học |

Bảng 8: Ý nghĩa cơ sở dữ liệu môn học

* + - Bảng Lớp học

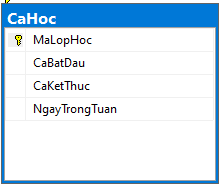
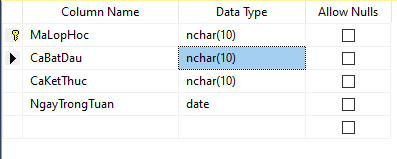


Bảng 9: Cơ sở dữ liệu lớp học

|  |  |  |
| --- | --- | --- |
| STT | Tên | Ý nghĩa |
| 1 | MaLopHoc | Mã lớp học dùng để phân biệt các lớp học với nhau |
| 2 | MaGiaoVien | Giáo viên sử dụng lớp học |
| 3 | MaMon | Môn học được dạy trong lớp học |
| 4 | MaPhongHoc | Mã phòng của lớp học |
| 5 | NguoiTao | Người tạo lớp học |
| 6 | TrangThai | Trạng thái lớp học |

Bảng 10: Ý nghĩa cơ sở dữ liệu lớp học

* + - Ca học

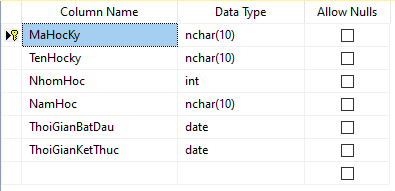
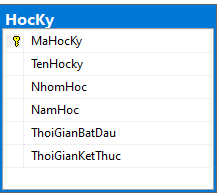


Bảng 11: Cơ sở dữ liệu ca học

|  |  |  |
| --- | --- | --- |
| STT | Tên | Ý nghĩa |
| 1 | MaLopHoc | Mã lớp học dùng để phân biệt các ca học trong lớp học |
| 2 | CaBatDau | Giờ học bắt đầu |
| 3 | CaKetThuc | Giờ học kết thúc |
| 4 | NgayTrongTuan | Ngày ca học mở trong tuần |

Bảng 12: Ý nghĩa cơ sở dữ liệu ca học

* + - Học kỳ

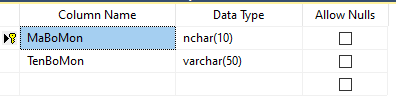


Bảng 13: Cơ sở dữ liệu học kỳ

|  |  |  |
| --- | --- | --- |
| STT | Tên | Ý nghĩa |
| 1 | MaHocKy | Mã học kỳ |
| 2 | TenHocKy | Tên học kỳ |
| 3 | NhomHoc | Nhóm học trong học kỳ |
| 4 | NamHoc | Năm học của học kỳ |
| 5 | ThoiGianBatDau | Thời gian bắt đầu của học kỳ |
| 6 | ThoiGianKetThuc | Thời gian kết thúc của học kỳ |

Bảng 14: Ý nghĩa cơ sở dữ liệu học kỳ

* + - Bộ môn

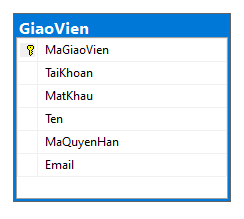
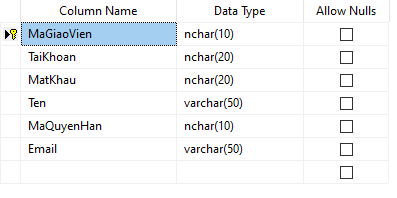


Bảng 15: Cơ sở dữ liệu bộ môn

|  |  |  |
| --- | --- | --- |
| STT | Tên | Ý nghĩa |
| 1 | MaBoMon | Mã bộ môn dùng để phân biệt các bộ môn với nhau |
| 2 | TenBoMon | Tên của bộ môn |

Bảng 16: Ý nghĩa cơ sở dữ liệu bộ môn

* + - Giáo viên

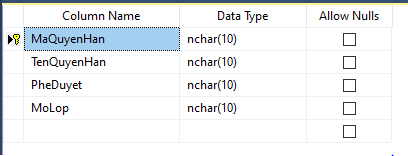
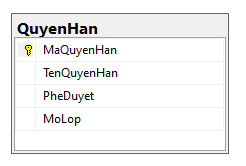
 

Bảng 17: Cơ sở dữ liệu giáo viên

|  |  |  |
| --- | --- | --- |
| STT | Tên | Ý nghĩa |
| 1 | MaGiaoVien | Mã giáo viên dùng để phân biệt các giáo viên với nhau |
| 2 | TaiKhoan | Tài khoản của giáo viên |
| 3 | MatKhau | Mật khẩu của tài khoản giáo viên |
| 4 | Ten | Tên giáo viên |
| 5 | MaQuyenHan | Mã quyền hạn của tài khoản giáo viên |
| 6 | Email | Email của giáo viên |

Bảng 18: Ý nghĩa cơ sở dữ liệu giáo viên

* + - Quyền hạn

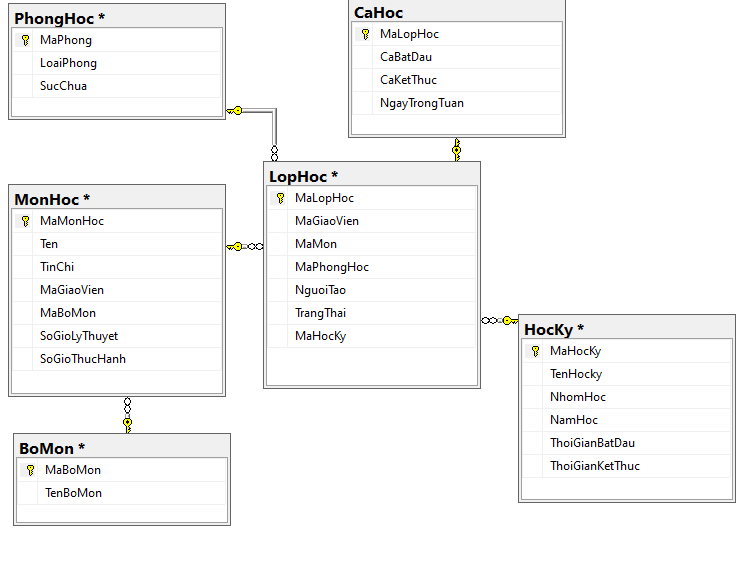


Bảng 19: Cơ sở dữ liệu quyền hạn

|  |  |  |
| --- | --- | --- |
| STT | Tên | Ý nghĩa |
| 1 | MaQuyenHan | Mã quyền hạn dùng để phân biệt các quyền hạn với nhau |
| 2 | TenQuyenHan | Tên quyền hạn |
| 3 | PheDuyet | Chức năng phê duyệt của quyền hạn |
| 4 | MoLop | Chức năng mở lớp của quyền hạn |

Bảng 20: Ý nghĩa cơ sở dữ liệu quyền hạn

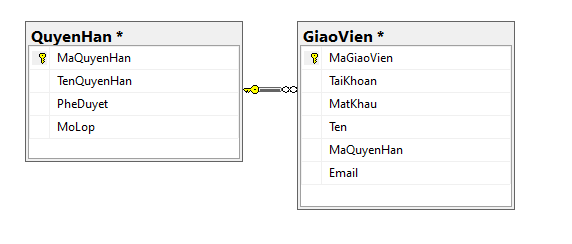
* + 1. **Cơ sở dữ liệu quản lý thời khóa biểu**



Hình 22: Các bảng quản lý thời khóa biểu

Trong cơ sở dữ liệu quản lý thời khóa biểu có 6 bảng.Để tạo nên được 1 thời khóa biểu chúng ta phải có nhiều môn học của mỗi bộ môn, các lớp học, phòng học được liên kết với ca học, một lớp học có thể có nhiều ca học và phòng học cũng vậy. Bảng học kỳ để phân ra môn học có trong từng học kỳ và các lớp học được sử dụng.

* + 1. **Cơ sơ dữ liệu phân quyền**



Hình 23: Các bảng quản lý phân quyền

Trong cơ sở dữ liệu phân quyền có 2 bảng, bảng quyền hạn và giáo viên. Bảng GiaoVien sẽ chưa các thông tin liên quan đến tài khoản và thông tin cá nhân của giáo viên. Mã quyền hạn và Mã giáo viên n-n với nhau.

# CHƯƠNG 5. THIẾT KẾ VÀ CÀI ĐẶT CÁC CHỨC NĂNG

* 1. **Thiết kế web**
     1. **Chức năng quản lý TKB trường**
        1. **Sơ đồ lớp phân tích**

****

Hình 24: Sơ đồ phân tích chức năng quản lý TKB trường

* + - 1. **Sơ đồ trình tự**

Hiển thị danh sách TKB trường



Hình 25: Sơ đồ trình tự chức năng quản lý TKB trường

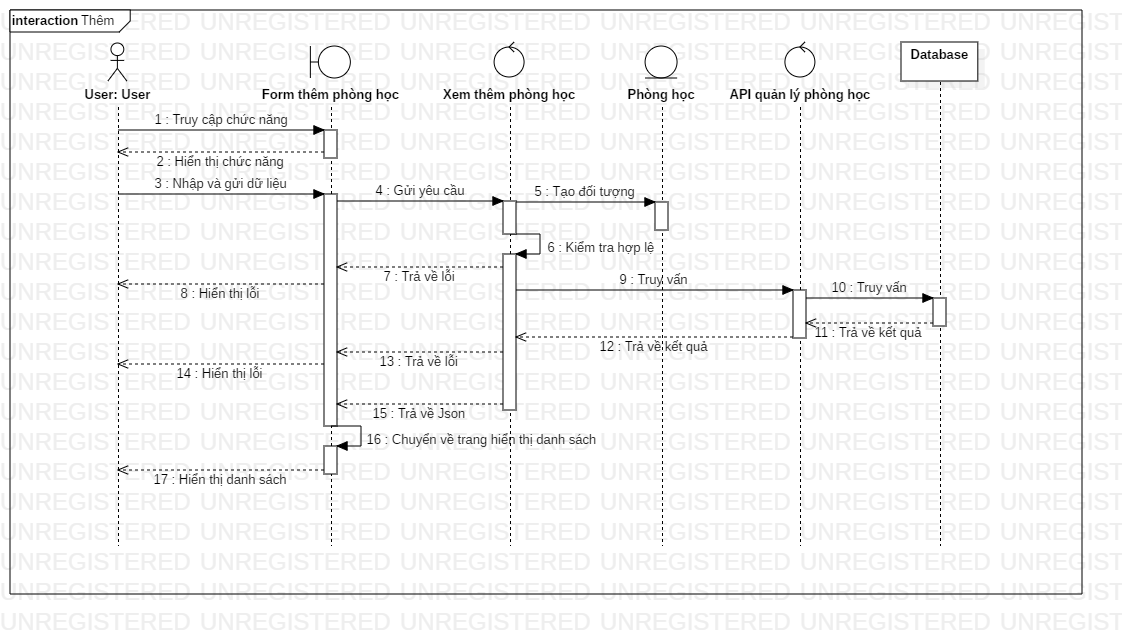
* + 1. **Chức năng quản lý phòng học**
       1. **Sơ đồ lớp phân tích**

****

Hình 26: Sơ đồ phân tích chức năng quản lý phòng học

* + - 1. **Sơ đồ trình tự**

Thêm phòng học



Hình 27: Sơ đồ trình tự chức năng thêm phòng học

* + 1. **Chức năng quản lý môn học**
       1. **Sơ đồ lớp phân tích**

****

Hình 28: Sơ đồ lớp phân tích chức năng quản lý môn học

* + - 1. **Sơ đồ trình tự**

Hiển thị danh sách môn học



Hình 29: Sơ đồ trình tự chức năng hiển thị danh sách các môn học

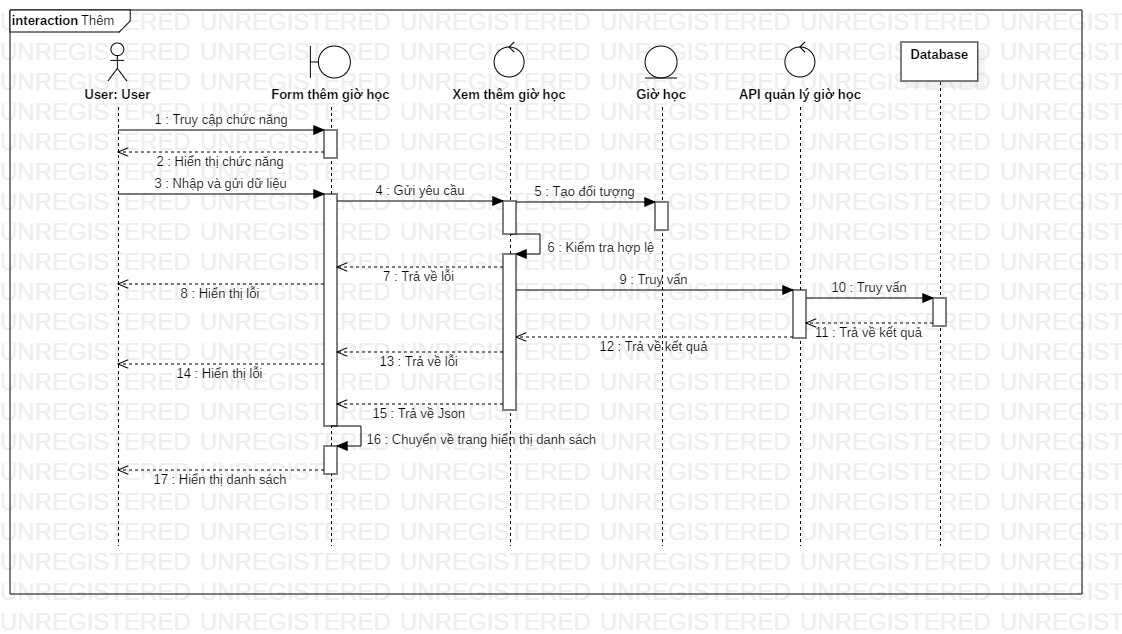
* + 1. **Chức năng quản lý khối giờ học**
       1. **Sơ đồ lớp phân tích**

****

Hình 30: Sơ đồ lớp phân tích chức năng quản lý giờ học

* + - 1. **Sơ đồ trình tự**

Thêm giờ học



Hình 31: Sơ đồ trình tự chức năng thêm giờ học

* + 1. **Chức năng quản lý ngành học**
       1. **Sơ đồ lớp phân tích**

****

Hình 32: Sơ đồ lớp phân tích chức năng quản lý ngành học

* + - 1. **Sơ đồ trình tự**

Hiển thị danh sách ngành học



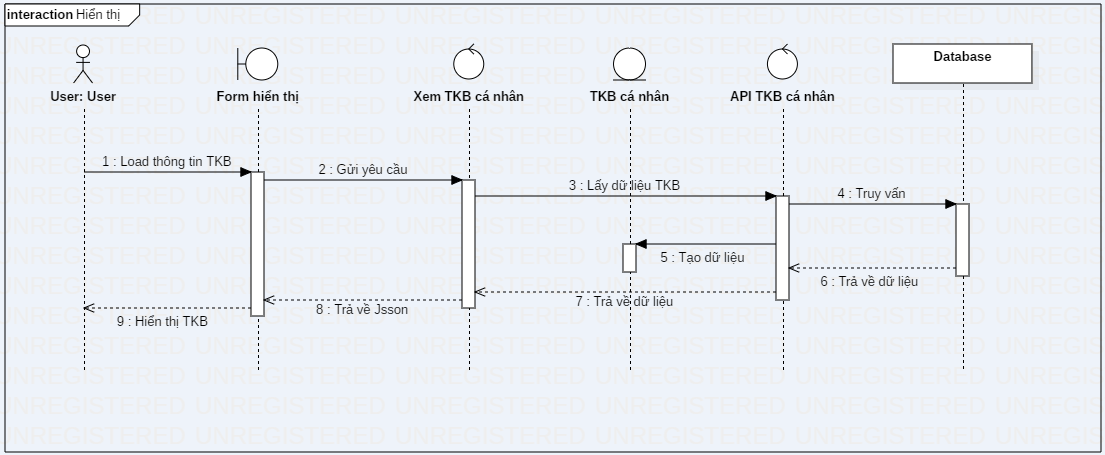
Hình 33: Sơ đồ trình tự chức năng hiển thị danh sách ngành học

* + 1. **Chức năng xem TKB cá nhân**
       1. **Sơ đồ lớp phân tích**

****

Hình 34: Sơ đồ lớp phân tích chức năng xem TKB cá nhân

* + - 1. **Sơ đồ trình tự**

****

Hình 35: Sơ đồ trình tự chức năng xem TKB cá nhân

* + 1. **Chức năng xem TKB khoa**
       1. **Sơ đồ lớp phân tích**

****

Hình 36: Sơ đồ lớp phân tích chức năng xem TKB khoa

* + - 1. **Sơ đồ trình tự**

****

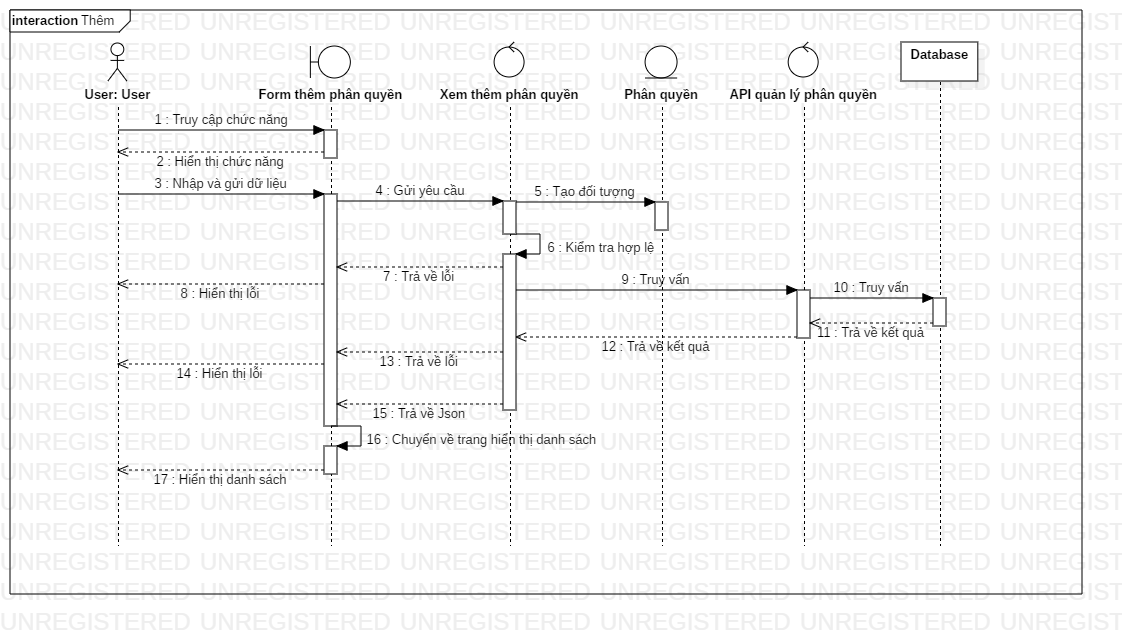
Hình 37: Sơ đồ trình tự chức năng xem TKB khoa

* + 1. **Chức năng phân quyền**
       1. **Sơ đồ lớp phân tích**

****

Hình 38: Sơ đồ lớp phân tích chức năng phân quyền

* + - 1. **Sơ đồ trình tự**
    - Thêm quyền



Hình 39: Sơ đồ trình tự chức năng thêm quyền

* + 1. **Chức năng quản lý tài khoản**
       1. **Sơ đồ lớp phân tích**

****

Hình 40: Sơ đồ lớp phân tích chức năng quản lý tài khoản

* + - 1. **Sơ đồ trình tự**

Hiển thị thông tin tài khoản



Hình 41: Sơ đồ trình tự chức năng hiển thị thông tin tài khoản

* + - Sửa tài khoản



* + 1. **Chức năng phân công giảng dạy**
       1. **Sơ đồ lớp phân tích**

****

Hình 42: Sơ đồ lớp phân tích chức năng phân công giảng dạy

* + - 1. **Sơ đồ trình tự**

Hiển thị danh sách phân công



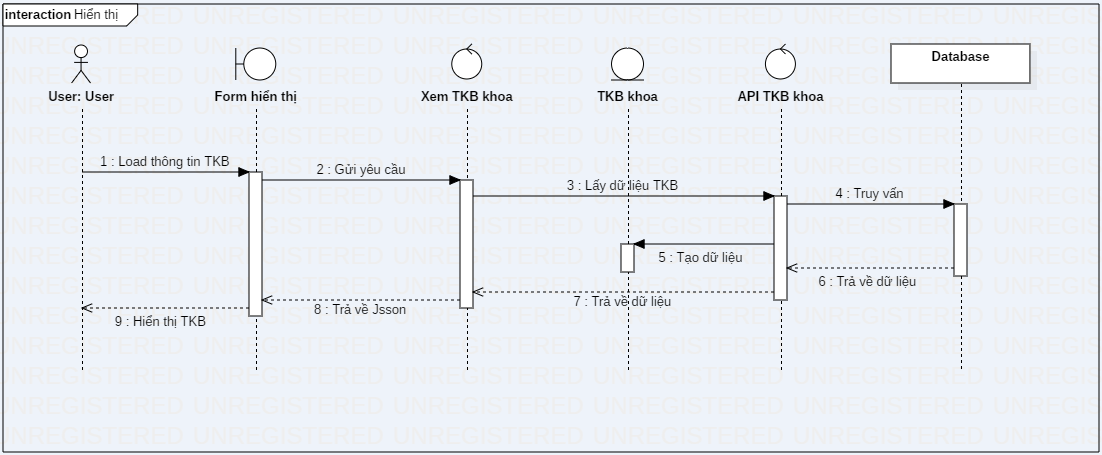
Hình 43: Sơ đồ trình tự hiển thị danh sách phân công

* + 1. **Chức năng quản lý TKB khoa**
       1. **Sơ đồ lớp phân tích**

****

Hình 44: Sơ đồ lớp phân tích chức năng quản lý TKB khoa

* + - 1. **Sơ đồ trình tự**
    - Hiển thị TKB khoa



Hình 45: Sơ đồ trình tự chức năng quản lý TKB khoa

* + 1. **Chức năng đăng nhập**
       1. **Sơ đồ lớp phân tích**

****

Hình 46: Sơ đồ lớp phân tích chức năng đăng nhập

* + - 1. **Sơ đồ trình tự**

****

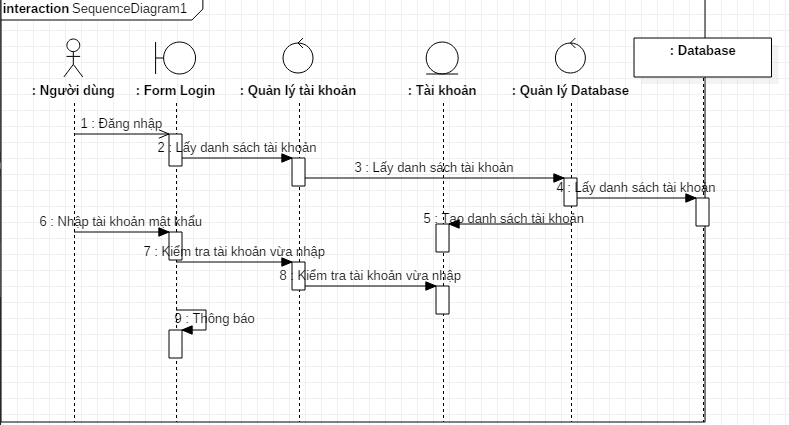
Hình 47: Sơ đồ trình tự chức năng đăng nhập

* 1. **Thiết kế mobile**
     1. **Chức năng Login**
        1. **Sơ đồ lớp phân tích**



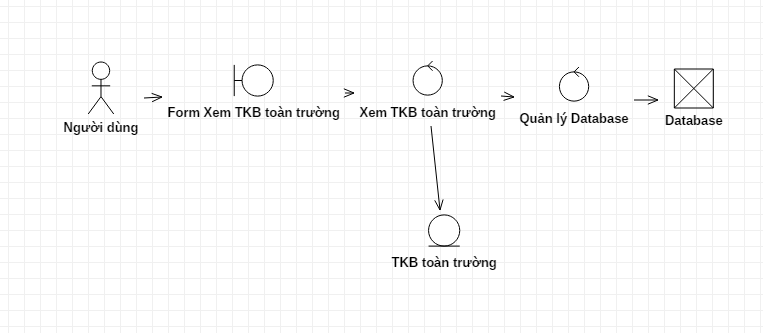
Hình 48: Sơ đồ lớp phân tích chức năng đăng nhập

* + - 1. **Sơ đồ trình tự**

****

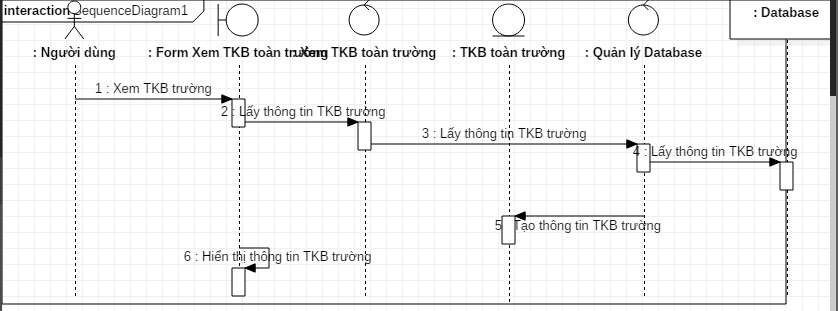
Hình 49: Sơ đồ trình tự chức năng đăng nhập

* + 1. **Chức năng Xem TKB trường**
       1. **Sơ đồ lớp phân tích**



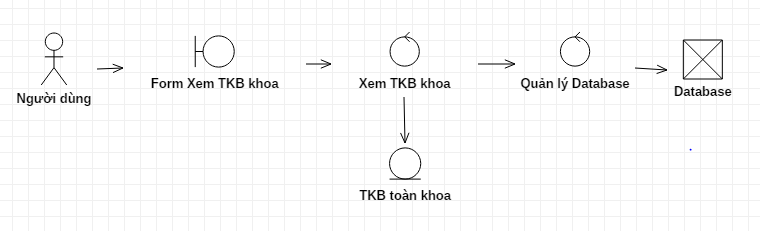
Hình 50: Sơ đồ lớp phân tích chức năng Xem TKB

* + - 1. **Sơ đồ trình tự**



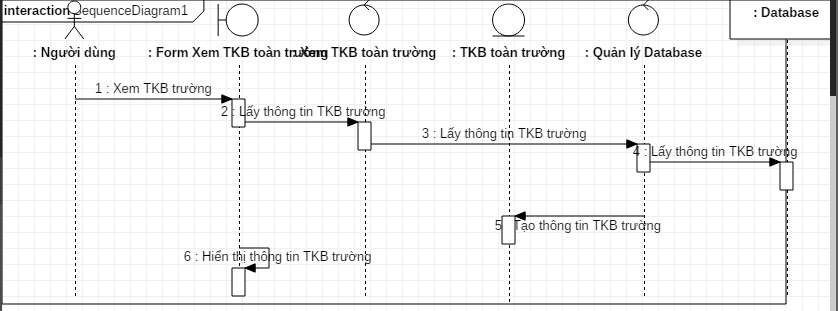
Hình 51: Sơ đồ trình tự chức năng xem TKB toàn trường

* + 1. **Chức năng Xem TKB khoa**
       1. **Sơ đồ lớp phân tích**



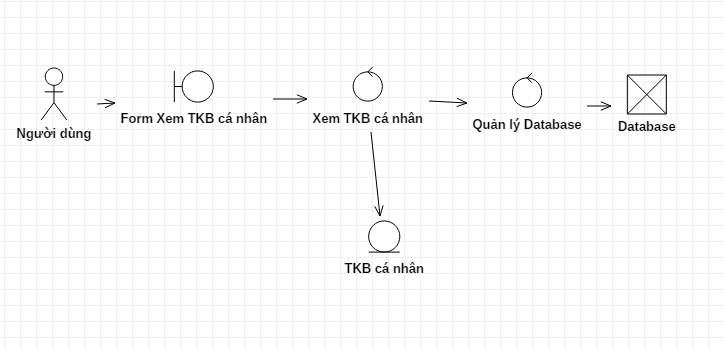
Hình 52: Sơ đồ lớp phân tích chức năng Xem TKB khoa

* + - 1. **Sơ đồ trình tự**



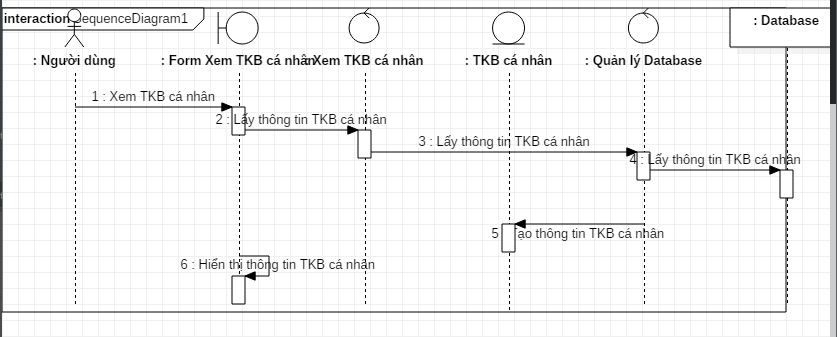
Hình 53: Sơ đồ trình tự chức năng Xem TKB toàn trường

* + 1. **Chức năng Xem TKB cá nhân**
       1. **Sơ đồ lớp phân tích**



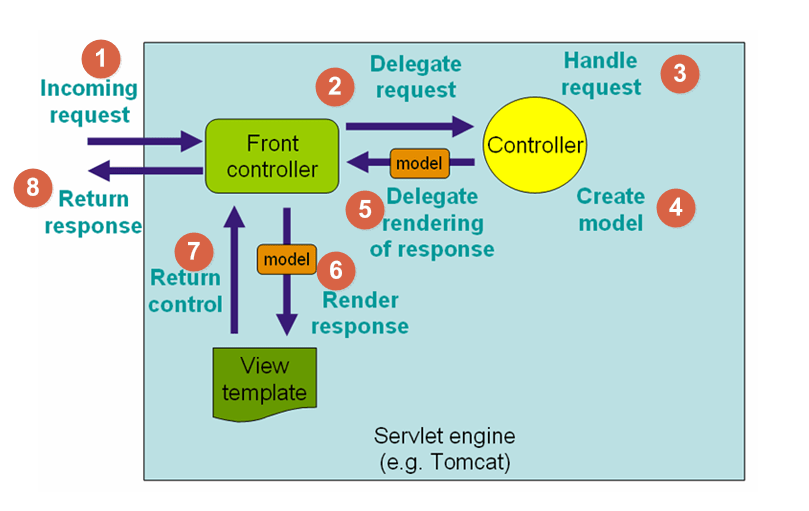
Hình 54: Sơ đồ lớp phân tích chức năng Xem TKB cá nhân

* + - 1. **Sơ đồ trình tự**



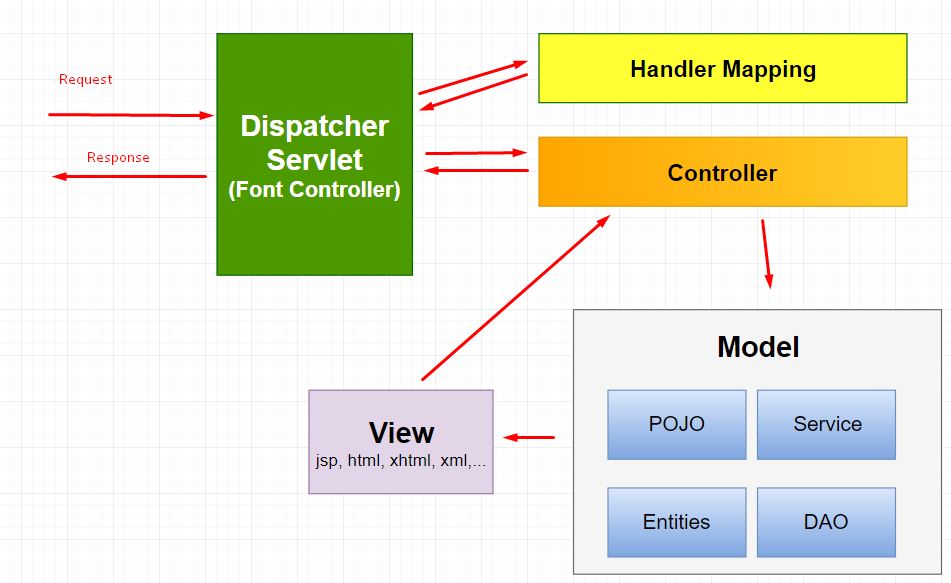
Hình 55: Sơ đồ trình tự chức năng Xem TKB cá nhân

* 1. **Cài đặt chức năng web**
     1. **Dự án sử dụng Spring MVC**
* Spring MVC là một Framework/1 Project mã nguồn mở của Spring. Spring MVC Framewrok cung cấp kiến truc MVC (Model-View-Controller) và các component được sử dụng để phát triển các ứng dụng web một cách linh hoạt
* Mô hình dự án:



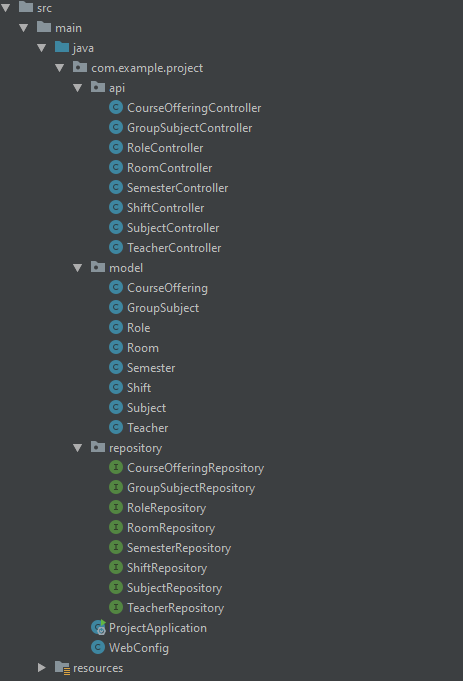
Hình 56: Mô hình dự án sử dụng chức năng Spring MVC

* Bất kỳ request nào tới ứng dụng web đều sẽ được gửi tới Front Controller (Dispatcher Servlet)
* Front Controller sẽ sử dụng Handler Mapping để biết được controller nào sẽ xử lý request đó
* Controller nhận request, gọi tới các class service thích hợp để xử lý yêu cầu.
* Sau khi xử lý xong, Controller sẽ nhận được model từ tầng Service hoặc tầng DAO.
* Controller gửi model vừa nhận được tới Front Controller (Dispatcher Servlet)
* Dispatcher Servlet sẽ tìm các mẫu view, sử dụng view resolver và truyền model vào nó.
* View template, model, view page được build và gửi trả lại Front Controller
* Front Controller gửi một page view tới trình duyệt để hiển thị nó cho người dùng.



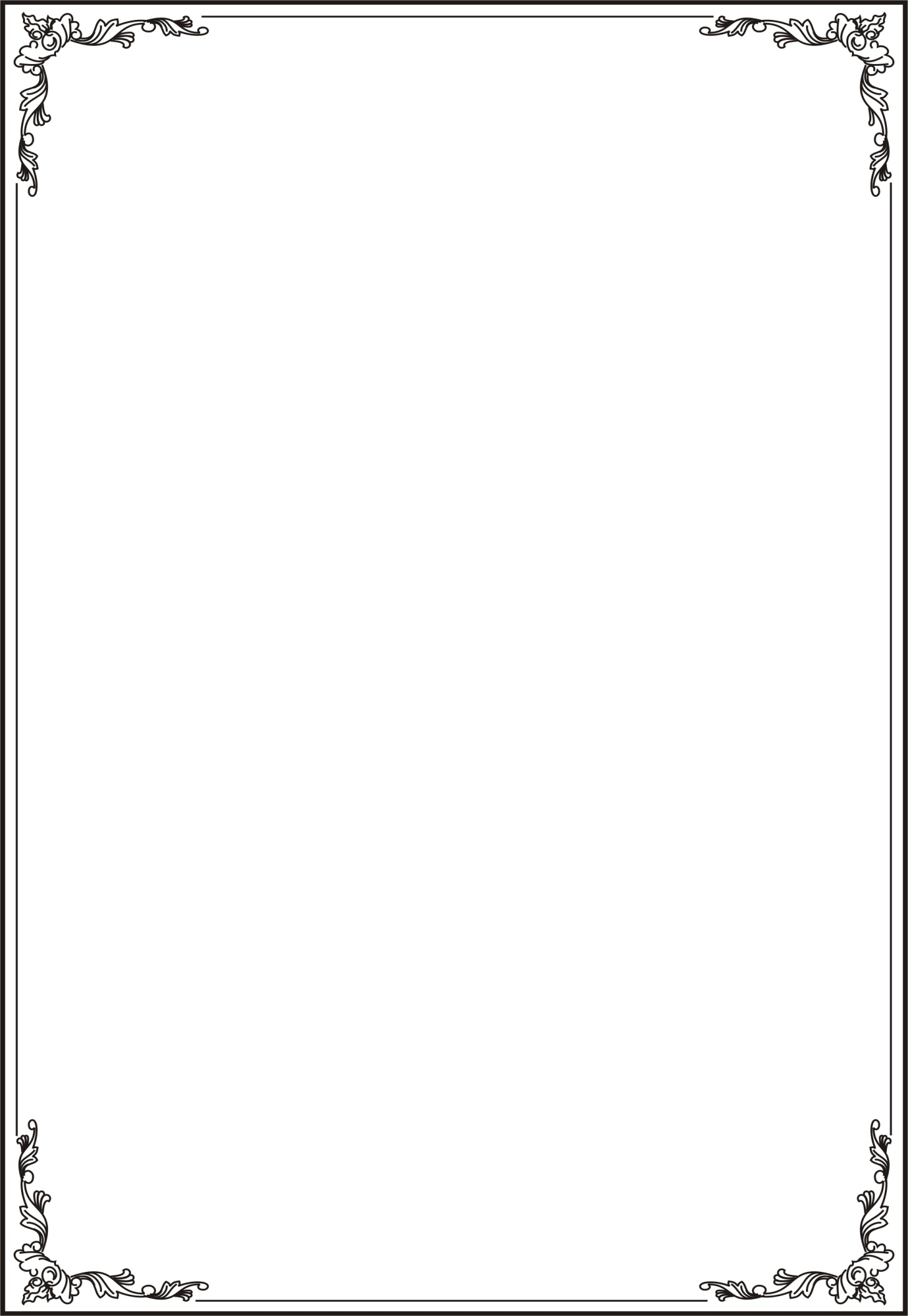
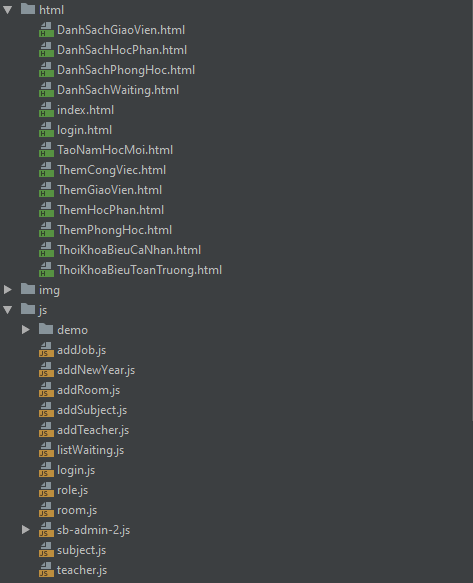
Hình 57: Mô hình dự án sử dụng chức năng Spring MVC

* **Model**: là các file POJO, Service, DAO thực hiện truy cập database, xử lý business
* **View**: là các file JSP, html…
* **Control**: là Dispatcher Controller, Handler Mapping, Controller – thực hiện điều hướn các request.
  + 1. **Các lợi ích của Spring MVC**
* Các tầng trong Spring MVC độc lập nên việc unit test dễ dàng hơn.
* Phần view có thể tích hợp với nhiều Framework về UI như JSF, Freemarker, Themeleaf…
* Spring MVC base trên các POJO class nên các hành động của nó khá đơn giản
* Hỗ trợ cả Annotation và XML config giúp việc phát triển nhanh hơn và sạch hơn.
* Cung cấp việc phân chia một cách rõ ràng, linh hoạt giữa controller, service, data acces layer.
  + 1. **Cấu trúc file mã nguồn**



Hình 58: Cấu trúc file mã nguồn

Cấu trúc thư mục phía backend gồm có 3 tầng chính: tầng model (entity) là tầng định nghĩa các thực thể, tầng repository giúp các model tương tác với database và tầng controller gồm các api để xử lý các yêu cầu của người dùng và trả dữ liệu về frontend để hiển thị lên view

Hình 59: Cấu trúc file mã nguồn

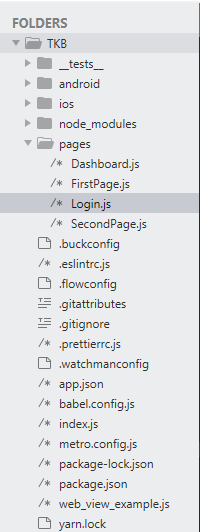
Cấu trúc thư mục phía fontend gồm các file .html, .css để tạo trang web và cái file .js (javascript) để xử lý kịch bản phía trang web.

* + 1. **Cấu trúc chung của 1 module**
* Để tạo được một module sẽ cần:
* Các file cấu hình để thực thể tương tác với dữ liệu trong database
* Các thực thể tương tác trong module
* Các api trong controller để điều hướng và xử lý các yêu cầu từ phía frontend, trả dữ liệu về cho frontend
* Các file .js để gửi yêu cầu, nhận phản hồi từ phía backend và tạo các kịch bản trên trang web
* Các file .html và css để tạo trang web
  + 1. **Cài đặt các chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên file | Ý nghĩa | Ánh xạ sang lớp phân tích |
| 1 | ..src/main/repository/SubjectRepository.java | Kết nối và thao tác thực thể Môn học với database | Database |
| 2 | ..src/main/model/Subject.java | Thực thể Môn học để tương tác trong module | Thực thể Môn học |
| 3 | ..src/main/controller/SubjectController | Nhận request từ frontend, chứa logic xử lý yêu cầu và trả dữ liệu về môn học cho frontend | Api Quản lý Môn học |
| 4 | ../html/DanhSachMonHoc.html  ../html/ThemMonHoc.html | Tạo trang giao diện quản lý Môn học để người dùng sử dụng | Người dùng (User) |
| 5 | ../js/Subject.js  ../js/AddSubject.js | Xử lý các thao tác của người dùng trên giao diện, gửi yêu cầu và nhận dữ liệu với backend | Form chức năng |

Bảng 21: Cấu trúc thư mục các chức năng

* + 1. **Ý nghĩa**
* **Model**: là các thực thể, đóng gói các đối tượng sử dụng trong dự án
* **Repository**: là các interface kế thừa từ MongoRepository, được cấu hình để thao tác với cơ sở dữ liệu MongoDB
* **Service**: xử lý các nghiệp vụ của bài toán. (đây là bài toán nhỏ nên phần xử lý nghiệp vụ được gộp vào phần controller)
* **Controller:** Điều hướng các request tới các service tương ứng (Mapping) và trả kết quả về frontend thông qua các api
  1. **Cài đặt chức năng mobile**
     1. **Framework sử dụng: React Native**
* React Native là một framework do công ty công nghệ nổi tiếng Facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động.
* Ưu điểm:
  + Hiệu quả về mặt thời gian khi mà bạn muốn phát triển một ứng dụng nhanh chóng.
  + Hiệu năng tương đối ổn định.
  + Cộng đồng phát triển mạnh.
  + Tiết kiệm tiền.
  + Team phát triển nhỏ.
  + Ứng dụng tin cậy và ổn định.
  + Xây dựng cho nhiều hệ điều hành khác nhau với ít native code nhất.
  + Trải nghiệm người dùng tốt hơn là hybrid app.
    1. **Cấu trúc file mã nguồn**

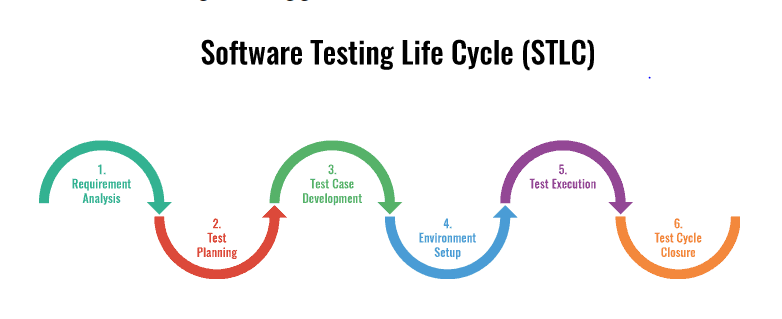


Hình 60: Cấu trúc file mã nguồn

* Fide index.js: file root, khi ứng dụng bắt đầu sẽ mở file này đầu tiên
* File Login.js: Xử lý chức năng login
* File Dashboad.js: Tạo khung trang chính của mobile
* File FirstPage.js: Màn hình “Thời khóa biểu toàn trường”
* File SecondPage.js: Màn hình “Thời khóa biểu cá nhân”

# CHƯƠNG 6: KIỂM THỬ

* 1. **Giới thiệu chung**
* Kiểm thử (Test) là một quá trình đánh giá một hệ thống hay là các thành phần của nó với mục đích là xác định xem nó có thỏa mãn những yêu cầu được đưa ra hay không. Hiểu một cách đơn giản, kiểm thử - test là chạy một chương trình để xác nhận bất kì lỗ hổng, lỗi sai hay những yêu cầu bị bỏ quên, những yêu cầu không đúng so với yêu cầu thực tế đề ra
* Theo tiêu chuẩn ANSI/IEEE 1059, kiểm thử - test có thể được định nghĩa như - một quá trình của việc phân tích các thành phần của phần mềm để dò tìm sự khác biệt giữa phần mềm thực tế đang tồn tại và những điều kiện được yêu cầu – requirement (đó là thiếu sót – defect, sai sót - error, lỗi - bug). Từ đó đánh giá được chất lượng của sản phẩm phần mềm.
  1. **Kiểm thử**
* Kiểm thử là 1 cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng sản phẩm hoặc dịch vụ được kiểm thử. Kiểm thử có thể cung cấp cho doanh nghiệp 1 quan điểm, 1 cách nhìn độc đáo về phần mềm để từ đó có thể cho phép đánh giá và thấu hiểu những rủi ro trong quá trình triển khai phần mềm.
* Trong kỹ thuật kiểm thử không chỉ giới hạn ở việc thực hiện một chương trình hoặc ứng dụng với mục đích đi tìm các lỗi phần mềm (bao gồm các lỗi và các thiếu sót) mà còn là một quá trình phê chuẩn và xác minh một chương trình máy tính / ứng dụng / sản phẩm nhằm:
  + - Đáp ứng được mọi yêu cầu hướng dẫn khi thiết kế và phát triển phần mềm.
    - Thực hiện công việc đúng như kỳ vọng.
    - Có thể triển khai được với những đặc tính tương tự.
    - Và đáp ứng được mọi nhu cầu của các bên liên quan.
* Tùy thuộc vào từng phương pháp, việc kiểm thử có thể được thực hiện bất cứ lúc nào trong quá trình phát triển phần mềm. Theo truyền thống thì các nỗ lực kiểm thử được tiến hành sau khi các yêu cầu được xác định và việc lập trình được hoàn tất nhưng trong Agile (là một tập hợp các phương pháp phát triển phần mềm linh hoạt dựa trên việc lặp đi lặp lại và gia tăng giá trị) thì việc kiểm thử được tiến hành liên tục trong suốt quá trình xây dựng phần mềm. Như vậy, mỗi một phương pháp kiểm thử bị chi phối theo một quy trình phát triển phần mềm nhất định.
  1. **Quy trình**
* Quy trình kiểm thử phần mềm xác định các giai đoạn/ pha trong kiểm thử phần mềm. Tuy nhiên, không có STLC tiêu chuẩn cố định nào trên thế giới, nhưng về cơ bản quy trình kiểm thử bao gồm những giai đoạn sau:



Hình 61: Quy trình kiểm thử phần mềm

1. *Requirenment analysis - Phân tích yêu cầu*
2. *Test planning - Lập kế hoạch kiểm thử*
3. *Test case development - Thiết kế kịch bản kiểm thử*
4. *Test environment set up - Thiết lập môi trường kiểm thử*
5. *Test execution - Thực hiện kiểm thử*
6. *Test cycle closure - Đóng chu trình kiểm thử*

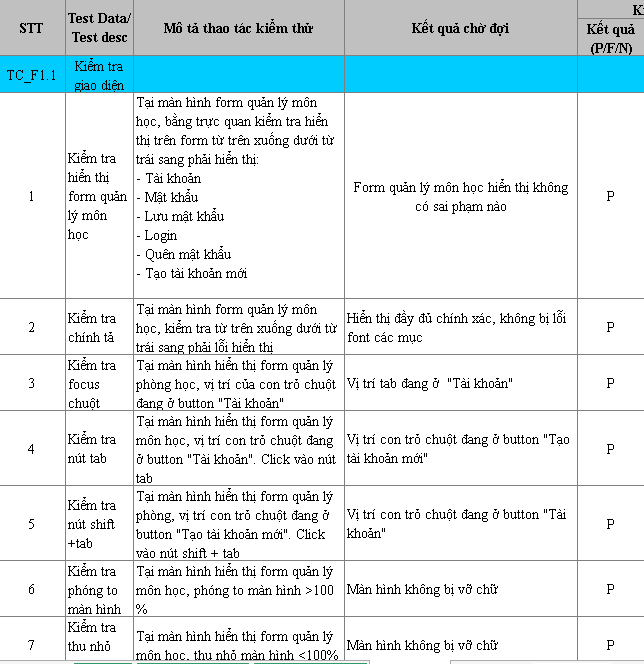
* Các giai đoạn kiểm thử được thực hiện một cách tuần tự. Mỗi giai đoạn sẽ có những mục tiêu khác nhau, đầu vào và kết quả đầu ra khác nhau nhưng mục đích cuối cùng vẫn là đảm bảo chất lượng sản phẩm phần mềm tốt nhất. Sau đây, chúng ta sẽ tìm hiểu chi tiết thông tin về các hoạt động, ai là người thực hiện, đầu vào, đầu ra của từng giai đoạn trong quy trình kiểm thử phần mềm.

**Phân tích quy trình kiểm thử**

* + 1. **Requirement analysis- Phân tích yêu cầu**
* Đầu vào: Đầu vào của giai đoạn phân tích yêu cầu bao gồm các tài liệu như: tài liệu đặc tả yêu cầu, tài liệu thiết kế hệ thống, tài liệu khách hàng yêu cầu về các tiêu chí chấp nhận của sản phẩm, bản prototype của khách hàng yêu cầu (nếu có), ...
* Hoạt động:
  + Phân tích yêu cầu là giai đoạn đầu tiên trong quy trình kiểm thử phần mềm.
  + QA team sẽ thực hiện đọc hiểu, nghiên cứu và phân tích cụ thể các yêu cầu trong tài liệu đặc tả của dự án hoặc tài liệu khách hàng. Qua hoạt động này, QA team sẽ nắm bắt được các yêu cầu mà dự án đưa ra bao gồm yêu cầu kiểm thử chức năng/ phi chức năng nào.
  + Ngoài ra, trong quá trình phân tích, nghiên cứu tài liệu, nếu có câu hỏi phát sinh hay đề xuất giải quyết, QA team sẽ đưa ra câu hỏi với các bên liên quan như BA (Business Analysis), PM (Project Manager), team leader, khách hàng để hiểu chính xác hơn về yêu cầu của sản phẩm. Những câu hỏi này sẽ được lưu trữ vào file Q&A (Question and Answer). Các câu hỏi nên được đưa ra dưới dạng Yes/No question hoặc các lựa chọn để tiết kiệm thời gian trả lời cũng như hỗ trợ đưa ra những gợi ý hay để xây dựng sản phẩm ngay từ đầu. Như vậy, đương nhiên là chúng ta không nên nêu ra những câu hỏi dạng là gì, như thế nào, tại sao, Những câu hỏi như thế thường mất thời gian để giải thích và cũng khó có thể giải thích một cách chi tiết nhất có thể. Hơn nữa, đối với khách hàng không có sự hiểu biết về lĩnh vực phần mềm mà họ yêu cầu thì càng không thể trả lời những câu hỏi mang tính chuyên môn cao. Chính chúng ta sẽ là người hỗ trợ và đưa ra giải pháp thích hợp cho khách hàng lựa chọn.
* Đầu ra: Đầu ra của giai đoạn phân tích yêu cầu bao gồm tài liệu chứa các câu hỏi và câu trả lời liên quan đến nghiệp vụ của hệ thống, tài liệu báo cáo tính khả thi, phân tích rủi ro của việc kiểm thử phần mềm.
  + 1. **Test planning- Lập kế hoạch kiểm thử**
* Đầu vào: Đầu vào của giai đoạn lập kế hoạch kiểm thử là các tài liệu đặc tả đã được cập nhật thông qua các câu hỏi và trả lời được đưa ra trong giai đoạn phân tích yêu cầu, tài liệu báo cáo tính khả thi, phân tích rủi ro của việc kiểm thử phần mềm.
* Hoạt động:

Dựa vào các tài liệu được cung cấp và cập nhật mới nhất, thông thường, test manager hoặc test leader sẽ là người lập kế hoạch kiểm thử cho cả QA team. Lập kế hoạch kiểm thử nhằm xác định một số yếu tố quan trọng sau:

* + Xác định phạm vi (Scope) dự án: Dự án thực hiện trong thời gian bao lâu? Bao gồm những công việc gì cho từng khoảng thời gian xác định? Từ đó đưa ra lịch trình thực hiện cho từng công việc nhỏ sao cho phù hợp với toàn bộ đội dự án.
  + Xác định phương pháp tiếp cận: Nói về cách tiếp cận để kiểm thử cho một đối tượng nào đó, thì phải dựa vào nhiều thứ, ví dụ như: Thời gian cho phép test có phù hợp với con số ước lượng, nhiều hay ít, yêu cầu chất lượng từ phía khách hàng thế nào? Cao, thấp hay khắc khe hay sao cũng được? Công nghệ / kỹ thuật sử dụng để phát triển ứng dụng này là gì? Lĩnh vực của hệ thống/sản phẩm đang được test (domain) là gì? Từ đó, test manager có thể đưa ra những phương pháp và kế hoạch phù hợp nhất cho cả quá trình thực hiện dự án sao cho đúng với các tiêu chí chấp nhận của sản phẩm và kịp tiến độ với các mốc thời gian bàn giao, phát hành.
  + Xác định các nguồn lực:
    - Con người: Bao nhiêu người tham gia dự án, ai sẽ test phần nào, bao nhiêu tester tham gia? Tester và nhóm phát triển có kinh nghiệm về lĩnh vực này không?
    - Thiết bị: số lượng server, version, máy tính, mobile để thực hiện test là bao nhiêu.
  + Lên kế hoạch thiết kế công việc test: Bản kế hoạch kiểm thử sẽ bao gồm các nội dung:
    - Liệt kê các chức năng cần kiểm thử: Để thực hiện test chức năng này thì cần làm những công việc gì, trong thời gian bao lâu, cái nào thực hiện trước, cái nào thực hiện sau, ai là người thực hiện.
  + Xác định điều kiện bắt đầu: xác định những điều kiện tối thiểu để bắt đầu hoạt động kiểm thử cho từng chức năng.
  + Xác định điều kiện kết thúc: khi có những điều kiện nào thì sẽ kết thúc việc kiểm thử.
* Đầu ra: Đầu ra của giai đoạn lập kế hoạch bao gồm các tài liệu như test plan, test estimation, test schedule.
  + 1. **Test case development- Thiết kế kịch bản kiểm thử**
* Đầu vào: Đầu vào của giai đoạn thiết kế kịch bản kiểm thử là test plan, test estimation, test schedule, các tài liệu đặc tả đã được cập nhật.
* Hoạt động:
  + Review tài liệu: Đầu tiên, các kiểm thử viên cần review lại tất cả các tài liệu để xác định công việc cần làm, các công việc có khác gì so với dự án trước khách hàng đưa cho, chức năng nào cần test, chức năng nào không cần test lại nữa. Từ đó, vừa có thể tiết kiệm thời gian mà vẫn đưa ra được một kịch bản kiểm thử đầy đủ và hiệu quả.
  + Viết test case/ check list: Sau đó, tester bắt tay vào việc viết test case chi tiết dựa vào kế hoạch đã đưa ra và vận dụng các kỹ thuật thiết kế kịch bản kiểm thử. Test case cần bao phủ được tất cả các trường hợp kiểm thử có thể xảy ra cũng như đáp ứng đầy đủ các tiêu chí của sản phẩm. Đồng thời tester cũng cần đánh giá mức độ ưu tiên cho từng test case.
  + Chuẩn bị dữ liệu kiểm thử: Cùng với việc tạo ra các test case chi tiết, đội kiểm thử cũng cần chuẩn bị trước các dữ liệu kiểm thử cho các trường hợp cần thiết như test data, test script.
  + Review test case/ check list: Sau khi hoàn thành, các thành viên trong đội kiểm thử hoặc test leader cũng cần review lại test case đã tạo để có thể bổ sung, hỗ trợ lẫn nhau nhằm tránh những sai sót trong thiết kế test case và rủi ro về sau.
    1. **Test environment set up- Thiết lập môi trường kiểm thử**
* Đầu vào: Đầu vào của giai đoạn cài đặt môi trường kiểm thử là test plan, smoke test case, test data.
* Hoạt động:
  + Việc cài đặt môi trường kiểm thử là giai đoạn cũng rất quan trọng trong vòng đời phát triển phần mềm. Môi trường kiểm thử sẽ được quyết định dựa trên những yêu cầu của khách hàng, hay đặc thù của sản phẩm ví dụ như server/ client/ network...
  + Tester cần chuẩn bị một vài test case để kiểm tra xem môi trường cài đặt đã sẵn sàng cho việc kiểm thử hay chưa. Đây chính là việc thực thi các smoke test case.
* Đầu ra: Đầu ra của giai đoạn này là môi trường đã được cài đặt đúng theo yêu cầu, sẵn sàng cho việc kiểm thử và kết quả của smoke test case.
  + 1. **Test excution-Thực hiện kiểm thử**
* Đầu vào: Tài liệu đầu vào của giai đoạn này là test plan, test design, test case, check list, test data, test automation script.
* Hoạt động:
  + Thực hiện các test case như thiết kế và mức độ ưu tiên đã đưa ra trên môi trường đã được cài đặt.
  + So sánh với kết quả mong đợi sau báo cáo các bug xảy ra lên tool quản lý lỗi và theo dõi trạng thái của lỗi đến khi được sửa thành công.
  + Thực hiện re-test để verify các bug đã được fix và regression test khi có sự thay đổi liên quan.
  + Trong quá trình thực hiện kiểm thử, kiểm thử viên cũng có thể hỗ trợ, đề xuất cho cả đội dự án để có giải pháp hợp lý và kết hợp công việc hiệu quả.
  + Đo và phân tích tiến độ: kiểm thử viên cũng cần kiểm soát chặt chẽ tiến độ công việc của mình bằng cách so sánh tiến độ thực tế với kế hoạch, nếu chậm cần phải điều chỉnh sao cho kịp tiến độ dự án, nếu nhanh cũng cần điều chỉnh vì có thể test lead lên kế hoạch chưa sát với thực tế dự án. Từ đó có thể sửa chữa test plan cần điều chỉnh để phù hợp với tiến độ dự án đưa ra.
  + Report thường xuyên cho PM và khách hàng về tình hình thực hiện dự án: Cung cấp thông tin trong quá trình kiểm thử đã làm được những chức năng nào, còn chức năng nào, hoàn thành được bao nhiều phần trăm công việc, báo cáo các trường hợp phát sinh sớm, tránh ảnh hưởng tiến độ công việc của cả ngày
* Đầu ra: Đầu ra của giai đoạn này là test results ( kết quả kiểm thử), defect reports( danh sách các lỗi tìm được).
  + 1. **Test cycle closure- Đóng chu trình kiểm thử**
* Đầu vào: Đầu vào của giai đoạn đóng chu trình kiểm thử là bao gồm tất cả những tài liệu liên quan đã được tổng hợp, ghi chép và hoàn thiện đầy đủ trong suốt quy trình kiểm thử của dự án: tài liệu phân tích đặc tả yêu cầu, test plan, test results, defect reports, tài liệu Q&A...
* Hoạt động:
  + Đây là giai đoạn cuối cùng trong quy trình kiểm thử phần mềm.
  + Ở giai đoạn này, QA team thực hiện tổng kết, báo cáo kết quả về việc thực thi test case, bao nhiêu case pass/ fail, bao nhiêu case đã được fix, mức độ nghiêm trọng của lỗi, bao nhiêu lỗi cao/ thấp, lỗi còn nhiều ở chức năng nào, dev nào nhiều lỗi. Chức năng nào đã hoàn thành test/ chưa hoàn thành test/ trễ tiến độ bàn giao.
  + Đánh giá các tiêu chí hoàn thành như phạm vi kiểm tra, chất lượng, chi phí, thời gian, mục tiêu kinh doanh quan trọng.
  + Ngoài ra, giai đoạn này cũng thảo luận tất cả những điểm tốt, điểm chưa tốt và rút ra bài học kinh nghiệm cho những dự án sau, giúp cải thiện quy trình kiểm thử.
* Đầu ra: Đầu ra của giai đoạn này bao gồm các tài liệu: Test report, Test results( final)
  1. **Các module kiểm thử**
     1. **Login**

****

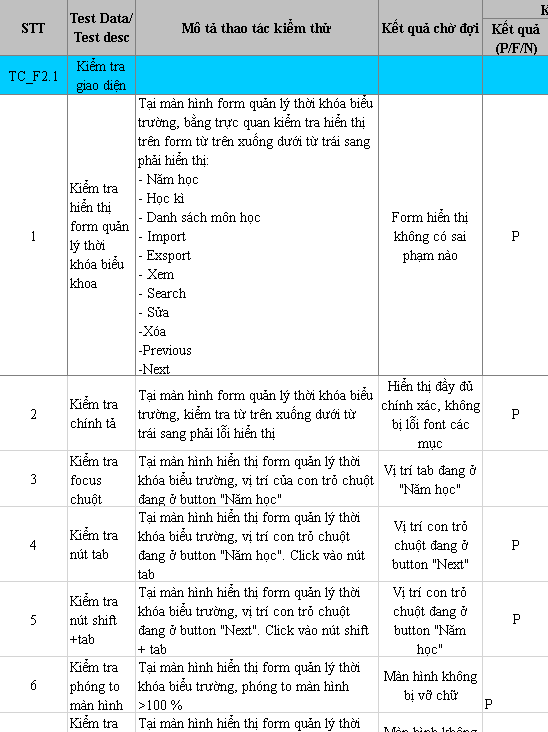
Hình 62: Module kiểm thử login

* Nhận xét

|  |  |
| --- | --- |
| Tên test case | Login |
| Mã Test case | TC\_F01 |
| Tổng số TC thực hiện | 13 |
| Tổng số TC Pass | 11 |
| Tổng số TC Fail | 2 |
| Test case Fail | - Button Quên mật khẩu  - Button Tạo tài khoản |
| Nguyên nhân | - Button Quên mật khẩu: Hệ thống chưa hoàn thiện, hiện tại không thể truy cập vào chức năng này  - Button Tạo tài khoản: Hệ thống chưa hoàn thiện, hiện tại không thể truy cập vào chức năng này |

Bảng 22: Kết quả test chức năng login

* + 1. **Quản lý thời khóa biểu trường**

****

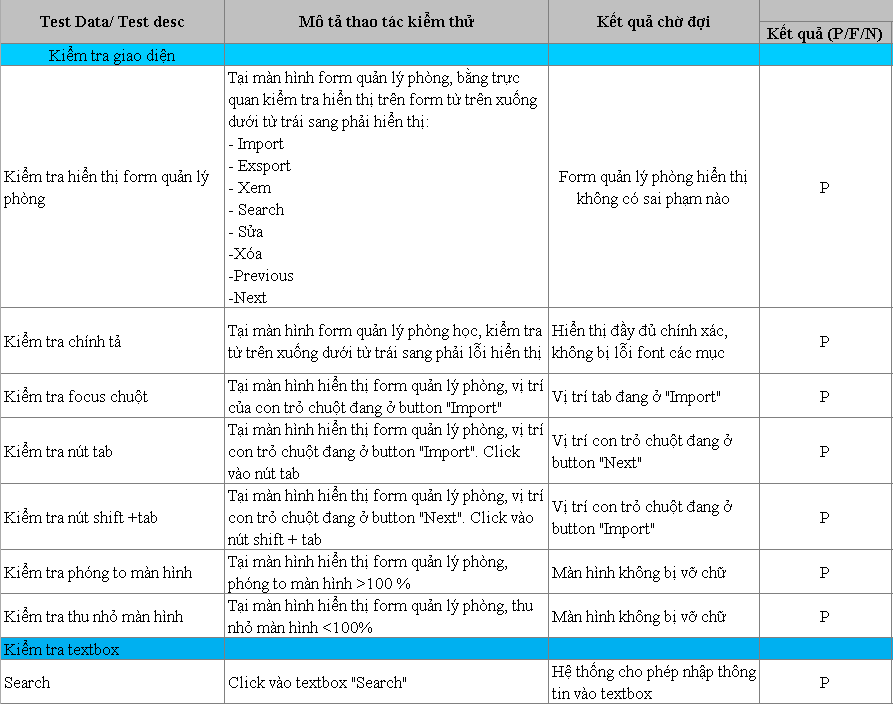
Hình 63: Module kiểm thử quản lý thời khóa biểu trường

* Nhận xét:

|  |  |
| --- | --- |
| Tên test case | Quản lý thời khóa biểu trường |
| Mã Test case | TC\_F02 |
| Tổng số TC thực hiện | 15 |
| Tổng số TC Pass | 15 |
| Tổng số TC Fail | 0 |

Bảng 23: Kết quả test chức năng quản lý thời khóa biểu trường

* + 1. **Quản lý phòng học**

****

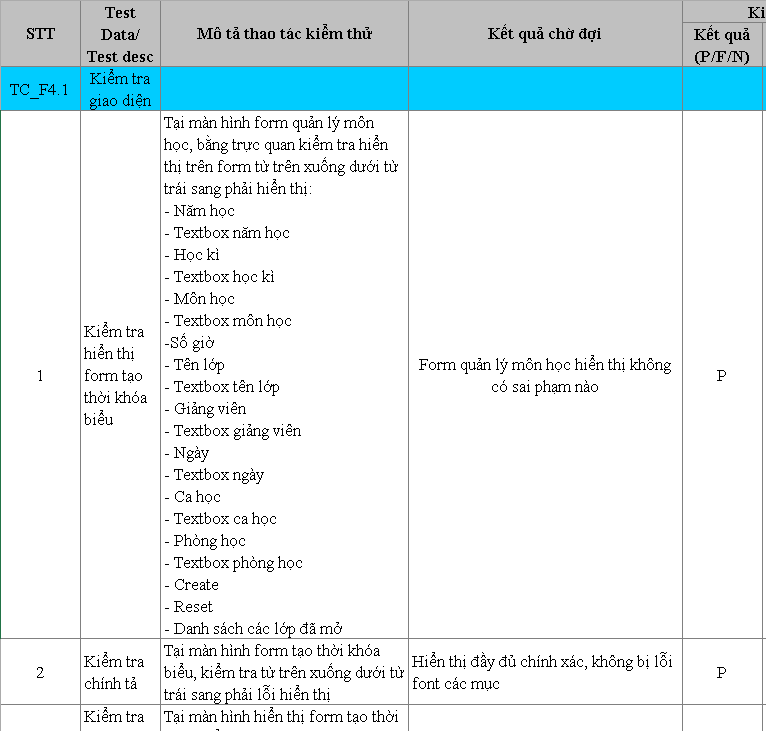
Hình 64: Module kiểm thử quản lý phòng học

* Nhận xét:

|  |  |
| --- | --- |
| Tên test case | Quản lý phòng học |
| Mã Test case | TC\_F03 |
| Tổng số TC thực hiện | 14 |
| Tổng số TC Pass | 12 |
| Tổng số TC Fail | 2 |
| Tese case Fail | - Button Previous  - Button Next |
| Nguyên nhân | - Button Previous: khi click vào button Previous hệ thống không hiển thị thông tin trang trước đó  - Button Next: khi click vào button Next hệ thống không hiển thị thông tin trang tiếp theo |

Bảng 24: Kết quả test chức năng quản lý phòng học

* + 1. **Quản lý môn học**

****

Hình 65: Module kiểm thử quản lý môn học

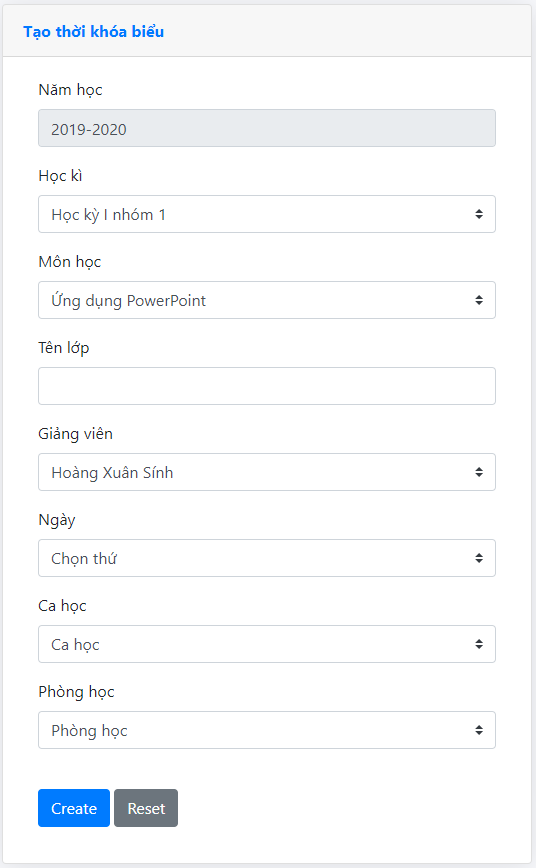
* Nhận xét:

|  |  |
| --- | --- |
| Tên test case | Quản lý môn học |
| Mã Test case | TC\_F04 |
| Tổng số TC thực hiện | 19 |
| Tổng số TC Pass | 19 |
| Tổng số TC Fall | 0 |

Bảng 25: Kết quả test chức năng quản lý môn học

# CHƯƠNG 7: THUẬT TOÁN TẠO THỜI KHÓA BIỂU

# 7.1. Quy trình tạo thời khóa biểu



Hình 67: Tạo thời khóa biểu

Trường năm học sẽ tự cập nhật năm học hiện tại và không cho chỉnh sửa

Trường học kì là một selectbox sẽ tự cập nhật các giá trị theo năm tương ứng

Trường môn học sẽ cập nhật các giá trị là các môn học sẽ mở trong kì

Trường “Tên lớp” sẽ cho phé người dùng nhập dữ liệu là tên của lớp học

Trường “Giảng viên” sẽ tự cập nhật các giảng viên trong bộ môn

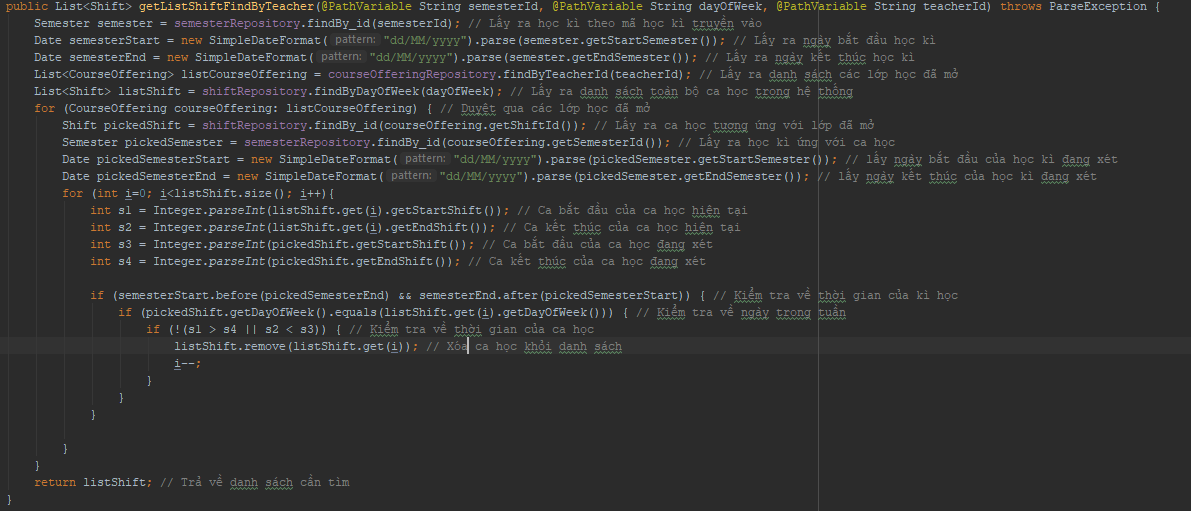
Trường “Ngày” sẽ có các giá trị là các ngày trong tuần

Khi trường “Học kì”, “Giảng viên” và “Ngày” được chọn giá trị, trường “Ca học” sẽ tự cập nhật các giá trị là các ca học còn trống theo 3 trường đã chọn

Khi trường “Học kì” và “Ca học” được chọn giá trị, trường “Phòng học” sẽ tự cập nhật các giá trị là các phòng học còn trống theo 2 trường đã chọn

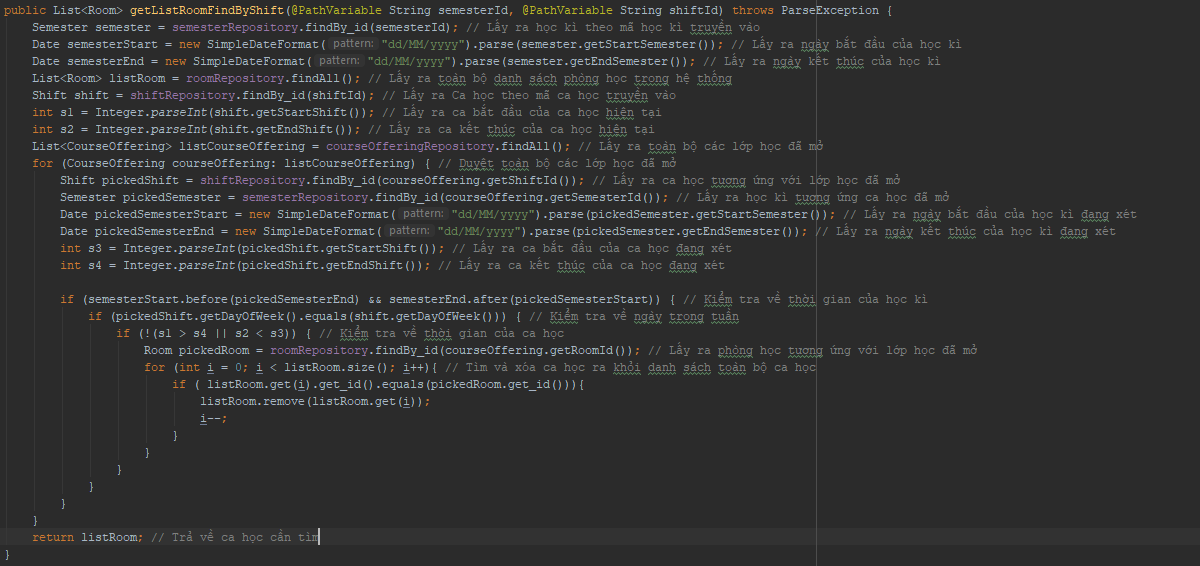
# 7.2. Thuật toán

* + 1. **Lấy danh sách các ca học còn trống**
* Các tham số cần truyền vào hàm: Học kì, Ngày trong tuần và Giảng viên
* Ý tưởng: Lấy toàn bộ danh sách các ca học và toàn bộ danh sách các ca học đã mở. Với mỗi ca học đã mở, đối chiếu với các ca học trong danh sách, nếu trùng thì loại ra khỏi danh sách toàn bộ ca học. Sau khi đối chiếu, danh sách toàn bộ ca học sẽ chỉ còn những ca học thỏa mãn điều kiện



Hình 68: Thuật toán danh sách các ca học

* Thực hiện:
  + B1: Lấy danh sách toàn bộ các ca học trong hệ thống
  + B2: Lấy toàn bộ danh sách các lớp học đã mở
  + B3: Duyệt toàn bộ danh sách ca học đã mở. Với mỗi lớp học đã mở, ta lấy ra ca học của lớp đó và kiểm tra với danh sách các ca học:
    - Kiểm tra về học kì: Nếu ngày bắt đầu kì học hiện tại bắt đầu sau ngày kết thúc của kì học ứng với ca học đang xét và ngày kết thúc của kì học hiện tại bắt đầu sau ngày bắt đầu của kì học ứng với ca học đang xét thì tiếp tục xét, nếu không thì bỏ qua
    - Kiểm tra về ngày trong tuần: Nếu ngày trong tuần của ca học hiện tại trùng với ngày trong tuần của ca học đang xét thì tiếp tục, nếu không thì bỏ qua
    - Kiểm tra về ca học: Nếu ca học hiện tại có ca bắt đầu lớn hơn ca kết thúc của ca đang xét hoặc có ca kết thúc bé hơn ca bắt đầu của ca học đang xét thì sẽ không bị trùng. Nếu bị trùng thì sẽ bỏ ca học đó ra khỏi danh sách ca học
  + B4: Sau khi vòng lặp kết thúc, danh sách toàn bộ ca học sẽ chỉ còn các ca học thỏa mãn.
    1. **Lấy danh sách các phòng học còn trống**
* Các tham số truyền vào: Học kì và Ca học
* Ý tưởng: Lấy toàn bộ danh sách ca học đã mở của các lớp đã mở và danh sách toàn bộ các phòng học trong trường. Với mỗi ca học thì đối chiếu với ca học đang chọn (ca học theo tham số truyền vào), nếu trùng thì tìm phòng học tương ứng với ca học đó và loại phòng học đó ra khổi danh sách phòng học. Sau khi duyệt qua tất cả các lớp học đã mở, danh sách phòng học sẽ loại được các phòng học đã mở nếu bị trùng



Hình 69: Thuật toán danh sách phòng trống

* Thực hiện:
  + B1: Lấy toàn bộ danh sách các phòng học trong hệ thống
  + B2: Duyệt toàn bộ danh sách các lớp học đã mở. Với mỗi lớp học đã mở lấy ra ca học của lớp đó và kiểm tra:
    - Kiểm tra về học kì: : Nếu ngày bắt đầu kì học hiện tại bắt đầu sau ngày kết thúc của kì học ứng với ca học đang xét và ngày kết thúc của kì học hiện tại bắt đầu sau ngày bắt đầu của kì học ứng với ca học đang xét thì tiếp tục xét, nếu không thì bỏ qua
    - Kiểm tra về ngày trong tuần: Nếu ngày trong tuần của ca học hiện tại trùng với ngày trong tuần của ca học đang xét thì tiếp tục, nếu không thì bỏ qua
    - Kiểm tra về ca học: Nếu ca học hiện tại có ca bắt đầu lớn hơn ca kết thúc của ca đang xét hoặc có ca kết thúc bé hơn ca bắt đầu của ca học đang xét thì sẽ không bị trùng. Nếu bị trùng, lấy ra phòng học của lớp học đang xét và loại phòng đó ra khỏi danh sách phòng học
  + B3: Sau khi vòng lặp kết thúc, danh sách toàn bộ phòng học ban đầu sẽ chỉ còn các phòng học trống ứng với học kì và ca học đã chọn

***Danh mục tài liệu tham khảo***

1. Javascript: <https://developer.mozilla.org/vi/docs/Web/JavaScript>
2. Springboot: <https://spring.io/guides>
3. MongoDB: <https://docs.mongodb.com/>
4. API: <https://restfulapi.net/>
5. React Native: <https://facebook.github.io/react-native/>
6. Java 8: <https://www.mkyong.com/tutorials/java-8-tutorials/>