

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



Hệ quản trị cơ sở dữ liệu **Student Management System**

Giảng viên hướng dẫn: ThS. Đỗ Thị Minh Phụng

Sinh viên thực hiện:

STT	MSSV	Họ và tên
1	16520241	Nguyễn Minh Đức
2	16521490	Nguyễn Trần Ngọc Anh
3	15521001	Lộc Sinh Vây

Lớp học: IS210.I22.HTCL

TP. HỒ CHÍ MINH – 6/2018

MỤC LỤC

LỜI CẢM ƠN	7
NHẬN XÉT CỦA GIÁO VIÊN.....	8
Chương 1: Khảo sát và phân tích yêu cầu	9
1.1. Tổng quan đề tài.....	9
1.2. Phát biểu bài toán.....	10
1.3. Phân tích yêu cầu	12
1.3.1. Yêu cầu chức năng.....	12
1.3.2. Yêu cầu phi chức năng	14
1.3.3. Yêu cầu quản trị, bảo mật hệ thống	15
1.4. Biểu mẫu, báo cáo	16
1.4.1 Sổ sách, hồ sơ	24
1.4.1.1 Hồ sơ của nhà trường	24
1.4.1.1.1 Hồ sơ chuyên môn	25
1.4.1.1.2 Hồ sơ của giáo viên	25
1.5. Thủ tục, quy trình	25
Chương 2: Các công nghệ sử dụng	27
2.1. Tổng quan về NodeJS	27
2.2. Tổng quan về Sequelize	27
2.3. Tổng quan về Bootstrap và Html/css.....	28
2.4. Quản lý thư viện mã nguồn mở NPM	30
2.5. Hệ thống quản lý mã nguồn Github	31
2.6. Quản lý dự án bằng nền tảng Trello	31
Chương 3: Hệ quản trị cơ sở dữ liệu MySQL.....	34
3.1. Tổng quan về hệ quản trị cơ sở dữ liệu MySQL	34
3.2. Tổng quan về InnoDB.....	34
4.1. Xây dựng và thiết kế cơ sở dữ liệu.....	36
4.1.1. Sơ đồ logic.....	36
4.1.1. Mô hình quan hệ	38
4.1.2. Mô tả dữ liệu	39

4.1.3.	Ràng buộc toàn vẹn.....	50
4.2.	Trigger.....	51
4.2.1.	Trigger trong MySQL	51
4.2.2.	Danh sách các Trigger	52
4.2.3.	Mô tả một số Trigger	54
4.3.	Store Procedure.....	59
4.3.1.	Stored procedure trong MySQL	59
4.3.2.	Danh sách các Store Procedure	60
4.3.3.	Mô tả một số Store Procedure	62
Chương 5: Các vấn đề liên quan đến xử lý đồng thời.....		75
5.1.	Tổng quan về các mức cô lập trong MySQL và sự khác biệt	75
5.1.1.	Read uncommitted.....	75
5.1.2.	Read committed	75
5.1.3.	Repeatable read.....	76
5.1.4.	Serializable	76
5.2.	Kịch bản và giải pháp các vấn đề liên quan đến xử lý đồng thời	77
5.2.1.	Unrepeatable Data	78
5.2.2.	Uncommitted data.....	81
5.2.3.	Lost update	85
5.2.4.	Phantom	90
5.2.5.	Deadlock	91
Chương 6: Thiết kế giao diện		94
6.1.	Màn hình “Đăng nhập”	95
6.2.	Màn hình “Bảng điều khiển”	96
6.3.	Màn hình “Học sinh”	97
6.4.	Màn hình “Thông tin học sinh”	98
6.5.	Màn hình “Mượn phòng”	99
6.6.	Màn hình “Kiểm duyệt phòng”	100
6.7.	Màn hình “Trả phòng”	101
6.8.	Màn hình “Các khối”	102

6.9.	Màn hình “Các lớp học theo khối”	103
6.10.	Màn hình “Lớp học”	104
6.11.	Màn hình “Hoạt động”	105
6.12.	Màn hình “Quản trị người dùng”	106
6.13.	Màn hình “Thông tin người dùng”	107
6.14.	Màn hình “Báo cáo/thống kê điểm TBM theo lớp”	108
Chương 7: Kết luận		109
7.1.	Kết quả đạt được	109
7.2.	Hạn chế.....	110
7.3.	Hướng phát triển.....	110
7.4.	Bảng phân công công việc	110
Chương 8: Phụ lục		112
8.1.	Biểu mẫu khác	112

DANH MỤC BẢNG

Bảng 1: Danh sách các yêu cầu chức năng của hệ thống	12
Bảng 2: Danh sách các yêu cầu ứng với biểu mẫu và qui định.....	14
Bảng 3: Biểu mẫu - Bảng điểm học sinh	17
Bảng 4: Biểu mẫu - Bảng điểm lớp học	18
Bảng 5: Biểu mẫu - Báo cáo hạnh kiểm	19
Bảng 6: Biểu mẫu - Phiếu yêu cầu mượn phòng.....	20
Bảng 7: Biểu mẫu - phiếu xác nhận trả phòng	21
Bảng 8: Biểu mẫu - PHIẾU TRA CỨU THÔNG TIN HỌC SINH.....	22
Bảng 9: Biểu mẫu - PHIẾU TRA CỨU CÁC PHÒNG ĐANG ĐƯỢC MƯỢN.....	23
Bảng 10: Mô tả dữ liệu - PERSONS	39
Bảng 11: Mô tả dữ liệu - EMPLOYEES	40
Bảng 12: Mô tả dữ liệu - TEACHERS	40
Bảng 13: Mô tả dữ liệu - SUPERVISORS	40
Bảng 14: Mô tả dữ liệu - STUDENTS	41
Bảng 15: Mô tả dữ liệu - ROOMS	41
Bảng 16: Mô tả dữ liệu - ROOMRENTALS.....	41
Bảng 17: Mô tả dữ liệu - ACADEMICYEAR	42
Bảng 18: Mô tả dữ liệu - CLASSES	42
Bảng 19: Mô tả dữ liệu - CLASSHASACADEMICYEAR.....	43
Bảng 20: Mô tả dữ liệu - STUDIESAT.....	43
Bảng 21: Mô tả dữ liệu - EVENTS	43
Bảng 22: Mô tả dữ liệu - SUBJECTS	44
Bảng 23: Mô tả dữ liệu - GRADES.....	44
Bảng 24: Mô tả dữ liệu - REPORTDETAILS.....	45
Bảng 25: Mô tả dữ liệu - REPORTCLASS.....	45
Bảng 26: Mô tả dữ liệu - REPORTCLASSDETAIL	46
Bảng 27: Mô tả dữ liệu - ROLES	47
Bảng 28: Mô tả dữ liệu - REPORTSTUDENT	48
Bảng 29: Mô tả dữ liệu - USERS	48
Bảng 30: Mô tả dữ liệu - LOGINACTIVITY	48
Bảng 31: Mô tả dữ liệu - ATTENTION	49
Bảng 32: Danh sách các Trigger	52
Bảng 33: Danh sách các Store Procedure.....	61
Bảng 34: Các vấn đề liên quan đến xử lý đồng thời.....	77
Bảng 35: Danh sách các màn hình và mô tả.....	94
Bảng 36: Bảng phân công công việc	110
Bảng 37: Biểu mẫu - Đơn xin rút hồ sơ.....	113
Bảng 38: Biểu mẫu - đơn xin chuyển trường	114

Bảng 39: Biểu mẫu - giấy xác nhận.....	116
Bảng 40: Biểu mẫu - mượn hồ sơ học bạ	117
Bảng 41: Biểu mẫu – đơn xin phép	118
Bảng 42: Biểu mẫu - KẾ HOẠCH THỰC HIỆN NHIỆM VỤ NĂM HỌC.....	119

DANH MỤC HÌNH ẢNH

Hình 1: Sự khác biệt giữa quản lý học sinh của trường và sở giáo dục	9
Hình 2: Logo NodeJS	27
Hình 3: Logo Sequelize	28
Hình 4: Logo Bootstrap	28
Hình 5: Cấu trúc cơ bản của HTML	29
Hình 6: Cấu trúc CSS	30
Hình 7: Cấu trúc cơ bản của CSS	30
Hình 8: Logo NPM	30
Hình 9: Logo Github.....	31
Hình 10: Một giao diện của Trello	32
Hình 11: Logo MySQL.....	34
Hình 12: Sơ đồ logic cơ sở dữ liệu	37
Hình 13: Màn hình Đăng nhập	95
Hình 14: Màn hình Bảng điều khiển	96
Hình 15: Màn hình Học sinh	97
Hình 16: Màn hình Thông tin học sinh	98
Hình 17: Màn hình Mượn phòng.....	99
Hình 18: Màn hình Kiểm duyệt phòng.....	100
Hình 19: Màn hình Trả phòng	101
Hình 20: Màn hình Các khối	102
Hình 21: Màn hình Các lớp học theo khối	103
Hình 22: Màn hình Lớp học	104
Hình 23: Màn hình Hoạt động.....	105
Hình 24: Màn hình Quản trị người dùng	106
Hình 25: Màn hình Thông tin người dùng	107
Hình 26: Màn hình Báo cáo/thống kê điểm TBM theo lớp.....	108

LỜI CẢM ƠN

Lời đầu tiên, nhóm sinh viên thực hiện xin gửi lời cảm ơn chân thành đến quý thầy cô Khoa Hệ Thống Thông Tin, trường Đại học Công Nghệ Thông Tin; là những người đã dùng trí thức và tâm huyết của mình để truyền đạt cho chúng em vốn kiến thức vô cùng quý báu trong khoảng thời gian học tập tại trường. Những kiến thức mà thầy cô truyền đạt là nền tảng quan trọng để nhóm có thể hoàn thành đề tài này. Nhóm xin chân thành cảm ơn Ths. Đỗ Thị Minh Phụng đã tận tình giúp đỡ, trực tiếp chỉ bảo, giải đáp các thắc mắc và hướng dẫn nhóm chúng em trong suốt quá trình làm đồ án môn học. Nhờ đó, chúng em đã tiếp thu được nhiều kiến thức bổ ích, đó sẽ là hành trang cho chúng em trong quá trình học tập và công tác sau này.

Trong thời gian một học kỳ thực hiện đề tài, nhóm đã vận dụng những kiến thức nền tảng, đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, nhóm đã vận dụng tối đa những gì thu thập được để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, trong quá trình thực hiện, nhóm không thể tránh khỏi những thiếu sót. Vì vậy, nhóm rất mong nhận được sự góp ý từ phía cô, nhằm hoàn thiện vốn kiến thức của mình, và là hành trang quý báu cho nhóm trong việc thực hiện những đồ án tiếp theo.

Xin chân thành cảm ơn cô!

[illegible]

Chương 1: Khảo sát và phân tích yêu cầu

1.1. Tổng quan đề tài

- **Tên đề tài:** Student Management System
- **Phạm vi thực hiện các chức năng sau:**
 - o Quản lý học sinh
 - o Quản lý lớp học
 - o Quản lý cơ sở vật chất
- **Tổng quan và lý do thực hiện đề tài:**

Student Management System là một ứng dụng web xây dựng hướng đến hỗ trợ công tác quản lý học sinh và các nghiệp vụ khác tại các trường THPT trên địa bàn thành phố Hồ Chí Minh. Ứng dụng web này được phát triển với mục tiêu thay thế và thống nhất các hệ thống quản lý giữa các trường THPT trên địa bàn từ đó tiết kiệm chi phí xây dựng và nâng cấp, đảm bảo tính thống nhất dữ liệu giữa các trường, từ đó cắt giảm các thủ tục gây tốn thời gian khác.

Theo khảo sát, hiện nay trên địa bàn tồn tại nhiều hệ thống quản lý học sinh nhưng đa phần những hệ thống này đã cũ từ đó không những không đáp ứng được các yêu cầu nghiệp vụ hằng ngày của các trường mà còn làm tăng các nguy cơ về bảo mật. Do đó, để thay đổi tình trạng này nhóm đã xây dựng lộ trình làm việc được chia thành các giai đoạn xây dựng và phát triển.



Hình 1: Sự khác biệt giữa quản lý học sinh của trường và sở giáo dục

Đầu tiên, nhóm tập trung về phân tích các quy trình nghiệp vụ trong lĩnh vực liên quan và thu thập các tài liệu cần thiết như biểu mẫu, giấy tờ, ... phục vụ việc xây dựng hệ thống. Từ đó xây dựng kế hoạch phát triển hệ thống theo nhiều giai đoạn, với mỗi giai đoạn tập trung xây dựng một số tính năng chủ chốt để phục vụ cho việc thử nghiệm thực tế và viết các tài liệu liên quan trước.

Student Management System được xây dựng lại từ đầu theo hướng module ¹ hoá các chức năng kết hợp sử dụng các nền tảng công nghệ mới nhất hiện nay giúp cải thiện tốc độ xử lý và tính bảo mật dữ liệu nhưng vẫn đảm bảo sự nhất quán các chức năng như quản lý điểm học sinh, lớp học, phòng học, học phí, thời khoá biểu, lương nhân viên. Hoạt động Đoàn – Hội, ... Từ đó, giúp dễ dàng quản lý các công việc, tiết kiệm chi phí và nhân lực.

1.2. Phát biểu bài toán

Trường THPT Trần Hưng Đạo là một trường THPT có tiếng tại TPHCM, với lượng học sinh ngày càng tăng. Để công tác quản lý học sinh tốt hơn, nhà trường đang muốn tin học hóa các quy trình quản lý. Trong đó sẽ tập trung chính quản lý các quy trình như quản lý thông tin học sinh, điểm số các môn học, quản lý lớp học, quản lý giáo viên, quản lý cơ sở vật chất², ... Bởi vì công tác quản lý thủ công trước đó ngày càng gặp khó khăn và bộc lộ nhiều hạn chế như:

- Các công văn nhiều, nhưng khu vực lưu trữ chưa đảm bảo.
- Các báo cáo thống kê thường sai sót các số liệu
- Quy trình cập nhập điểm lâu, nhưng thường xuyên bị sai sót điểm số
- Thống kê thu chi có nhiều sai sót, chưa quản lý được con số cụ thể
- Các văn bản do Đoàn trường không được lưu trữ.
- Quy trình thông báo hoạt động đến học sinh còn rườm rà
- Thông tin chuyên cần, hạnh kiểm học sinh thường bị bỏ qua và không ghi chép hằng ngày
- Chưa có mẫu báo cáo chung cho các sự kiện, vấn đề.
- Chưa có quy trình quản lý cơ sở vật chất của nhà trường.
- Tra cứu thông tin về học sinh, điểm số mất nhiều thời gian và nhiều khi không chính xác.
- Cập nhật các thông tin hằng ngày tốn nhiều thời gian và khó khăn trong việc thực hiện báo cáo thống kê, nhất là khi có sự việc đột xuất.
- Khi lượng học sinh ngày một tăng thì công việc của giáo viên sẽ rất vất vả, có khi dẫn đến sai sót. Việc lập báo cáo thống kê định kì gặp khó khăn.

¹ Module thường bao gồm một hoặc nhiều trang mô tả nội dung và được chia thành nhiều khối khác nhau trên website. Các module thường thấy là module tìm kiếm, module logo, module menu, module map, module slideshow...

² Chức năng chính được sử dụng để trình bày các vấn đề liên quan đến cơ sở dữ liệu

Do đó cần xây dựng một hệ thống thông tin mới nhằm đáp ứng nhu cầu quản lý cơ bản sau:

- Giảm khối lượng ghi chép nhằm lưu trữ thông tin.
- Cập nhật dữ liệu nhanh chóng, chính xác và kịp thời.
- Thống kê được số lượng học sinh và tình hình học lực học hạnh kiểm, chuyên cần trong nhà trường.
- In các báo cáo cần thiết như: phiếu điểm, phiếu liên lạc, báo cáo đánh giá.
- Có khả năng lưu trữ thông tin lâu dài, đảm bảo truy vấn nhanh khi cần thiết.

Danh sách lớp sẽ được tổ văn phòng sắp xếp theo thứ tự điểm số tuyển sinh đầu vào và năng lực học tập ngang nhau. Sĩ số lớp tối đa là 45 bạn / lớp. Với mỗi lớp, sẽ có 1 giáo viên chủ nhiệm. Lớp tuyển là lớp có số từ A1 -> A9, giáo viên chủ nhiệm các lớp này phải là giáo viên có chức vụ từ tổ phó bộ môn trở lên.

Ở mỗi học kì, các môn học sẽ có các cột điểm: điểm miệng (hệ 1), điểm 15p (hệ 1), 1 tiết (hệ 2), cuối kì (hệ 3). Các điểm này sẽ được giáo viên các môn cập nhật lên hệ thống, học sinh có thể xem điểm của mình trên hệ thống thông qua tài khoản. Hết thời hạn nhập điểm, các điểm số sẽ bị khoá lại.

Hàng tuần, trợ lý thanh niên thường tổ chức hoạt động cho các học sinh. Các hoạt động được tổ chức theo khối hoặc theo trường. Và được thông báo với các em bí thư, phó bí thư qua email và qua hệ thống. Các em bí thư, phó bí thư đều có tài khoản để truy cập hệ thống, cho đến khi các em không còn học tập hoặc hết làm các chức vụ trên. Hoạt động được chia thành các loại: học tập, sinh hoạt tập thể, từ thiện...

Mỗi lớp được cấp cho 1 phòng học, học đến hết năm học. Tuy nhiên, thứ và chủ nhật trường không tổ chức học tập, các lớp hoặc câu lạc bộ đội nhóm có thể đăng ký với giám thị về việc mượn phòng học. Thời gian mượn tối thiểu là 2 ngày trước ngày mượn phòng học. Giám thị sẽ là người có quyền chấp nhận hoặc từ chối yêu cầu mượn phòng, cũng như quản lý các vấn đề liên quan đến cơ sở vật chất.

Sau mỗi học kì hệ thống sẽ thống kê, báo cáo điểm số, học lực, hạnh kiểm của học sinh để đánh giá năng lực của mỗi học sinh cũng như cách giảng dạy của giáo viên để nâng cao phương pháp giáo dục hiệu quả hơn.

1.3. Phân tích yêu cầu

1.3.1. Yêu cầu chức năng

Bảng 1: Danh sách các yêu cầu chức năng của hệ thống

STT	Tên chức năng	Mô tả	Thực hiện
1	Quản lý thông tin học sinh	Hỗ trợ người dùng lưu giữ thông tin, lý lịch học sinh, hạnh kiểm, chuyên cần của học sinh.	X
2	Quản lý nhập điểm	Hỗ trợ người dùng lưu giữ điểm số của các học sinh trong hệ thống và cập nhật điểm trong thời gian cho phép.	
3	Quản lý xem điểm	Hỗ trợ giáo viên xem danh sách điểm của các học sinh mà giáo viên đó dạy	
4	Quản lý điểm cá nhân	Hỗ trợ học sinh xem điểm cá nhân của mình thông qua tài khoản được cấp.	
5	Quản lý mượn phòng học	Hỗ trợ cho học sinh mượn phòng học cho mục đích học tập nhóm, các hoạt động Đoàn-Hội, được đăng ký trước thông qua tài khoản được cấp.	X
6	Quản lý kiểm duyệt phòng	Hỗ trợ người dùng xem và xác nhận thông tin mượn phòng của học sinh.	X
7	Quản lý danh sách các lớp khối 10.	Hỗ trợ người dùng quản lý danh sách học sinh(thêm, xóa , cập nhật thông tin học sinh) các khối lớp 10 tại năm học đó	X
8	Quản lý danh sách các lớp khối 11.	Hỗ trợ người dùng quản lý danh sách học sinh(thêm, xóa , cập nhật thông tin học sinh) các khối lớp 11 tại năm học đó	X
9	Quản lý danh sách các lớp khối 12.	Hỗ trợ người dùng quản lý danh sách học sinh(thêm, xóa , cập nhật thông tin học sinh) các khối lớp 12 tại năm học đó	X
10	Quản lý giáo viên bộ môn	Hỗ trợ người dùng xem thông tin và cập nhật các thông tin của các giáo viên bộ môn.	
11	Quản lý các vi	Hỗ trợ giám thị xem, cập nhật các thông tin vi	

	phạm của giáo viên	phạm của giáo viên.	
12	Quản lý thống kê chung	Hỗ trợ người dùng thống kê tổng quát về học lực, hạnh kiểm theo cá nhân, theo lớp, theo khối.	
13	Quản lý báo cáo điểm theo môn	Hỗ trợ giáo viên, giám thị báo cáo điểm theo môn của học sinh tại học kỳ đó	
14	Quản lý báo cáo điểm lớp	Hỗ trợ giáo viên, giám thị báo cáo điểm theo từng lớp tại học kỳ đó	
15	Quản lý báo cáo điểm từng học sinh	Hỗ trợ giáo viên, giám thị báo cáo điểm theo từng học sinh	
16	Quản lý xem các hoạt động	Hỗ trợ người dùng xem các hoạt động được đăng trên hệ thống.	
17	Quản lý tạo các hoạt động	Hỗ trợ người dùng tạo các hoạt động trên hệ thống.	
18	Quản lý góp ý kiến cho trường	Hỗ trợ người dùng đóng góp các ý kiến cá nhân, các mong muốn tích cực cho nhà trường.	
19	Quản lý báo cáo các sự cố kỹ thuật	Hỗ trợ người dùng báo cáo các sự cố về kỹ thuật, vật chất của nhà trường.	
20 ³	Quản trị người dùng	Hỗ trợ quản trị viên phân quyền và kiểm soát hệ thống, đảm bảo sự ổn định của hệ thống.	X

³ Chức năng được xây dựng để hỗ trợ cấp quyền truy cập người dùng

Bảng 2: Danh sách các yêu cầu ứng với biểu mẫu và qui định

STT	Tên yêu cầu	Biểu mẫu	Qui định	Ghi chú
1	Lập bảng điểm số học sinh	BM1	QĐ-1, QĐ-2, QĐ-7, QĐ-9	
2	Lập bảng điểm lớp học	BM2	QĐ-5, QĐ-6, QĐ-7, QĐ-8, QĐ-9	
3	Lập báo cáo hạnh kiểm lớp	BM3		
4	Lập phiếu mượn phòng	BM4	QĐ-3, QĐ-4, QĐ-10	
5	Lập phiếu trả phòng	BM5		
6	Tra cứu thông tin học sinh	BM6	QĐ-12	
7	Tra cứu danh sách các phòng đang mượn	BM7		

Đáp ứng các yêu cầu cơ bản:

- Quản lý các môn học của các lớp theo học kỳ và kết quả học tập của học sinh đối với các môn học đó. Hệ thống còn phải cho phép tạo ra các báo cáo từ kết quả học tập của học sinh nhằm phục vụ công tác điều hành tập huấn như: Tổng kết kết quả học tập theo kỳ, theo năm, theo khối; In bản điểm học kỳ; In Bảng điểm cá nhân...
- Ngoài các chức năng trên, hệ thống còn cần thêm một số chức năng khác nhau: cập nhật loại danh mục dữ liệu (danh mục lớp, danh mục học sinh ...); các chức năng sao lưu và phục hồi dữ liệu; các chức năng trợ giúp, chức năng liên kết website khác của trường,..., hệ thống cần được bảo trì và backup hàng tháng, hệ thống cần đảm bảo lưu trữ được số lượng văn bản lớn trong vòng tối đa 10 năm

1.3.2. Yêu cầu phi chức năng

Khả năng hoạt động, sử dụng: hệ thống hoạt động 24/7 kể cả ngày nghỉ, ngày lễ, hệ thống cần được vận hành một cách ổn định, đảm bảo tốc độ truy xuất cao và liên tục, hạn chế thấp nhất về sai sót dữ liệu, lỗi hệ thống và các chính sách bảo mật hệ thống

Bên cạnh việc lưu trữ thông tin, phần mềm phải đáp ứng yêu cầu sau:

- Chức năng tra cứu: đáp ứng nhu cầu tìm kiếm thông tin từ người dùng như: tra cứu thông tin học sinh, tra cứu điểm, tra cứu giáo viên, tra cứu mượn phòng học.

- Chức năng thống kê, báo cáo: đáp ứng nhu cầu của người dùng trong việc lập báo cáo về tình hình học tập của các học sinh.
- Chức năng cập nhật thông tin: thêm, sửa, xóa thông tin của các đối tượng lưu trữ: thông tin học sinh, điểm số, thông tin giáo viên, các hoạt động Đoàn-Hội, danh sách các khối lớp học, mượn phòng.
- Chức năng phân quyền truy cập.
- Chức năng sao lưu, phục hồi.

1.3.3. Yêu cầu quản trị, bảo mật hệ thống

STT	Tên chức năng	Giáo viên	Giám thị	Học sinh
1	Quản lý thông tin học sinh		X	
2	Quản lý nhập điểm	X		
3	Quản lý xem điểm	X	X	
4	Quản lý điểm cá nhân			X
5	Quản lý mượn phòng học	X	X	X
6	Quản lý kiểm duyệt phòng		X	
7	Quản lý danh sách các lớp khối 10.		X	
8	Quản lý danh sách các lớp khối 11.		X	
9	Quản lý danh sách các lớp khối 12.		X	
10	Quản lý giáo viên bộ môn		X	
11	Quản lý các vi phạm của giáo viên		X	
12	Quản lý thống kê chung	X	X	
13	Quản lý báo cáo điểm theo môn	X	X	
14	Quản lý báo cáo điểm lớp	X	X	

15	Quản lý báo cáo điểm từng học sinh	X	X	
16	Quản lý xem các hoạt động	X	X	X
17	Quản lý tạo các hoạt động	X	X	X
18	Quản lý góp ý kiến cho trường			X
19	Quản lý báo cáo các sự cố kỹ thuật			X
20 ⁴	Quản trị người dùng			

1.4. Biểu mẫu, báo cáo

Ngoài ra, trong quá trình phân tích nghiệp vụ, nhóm đã liên hệ với nhà trường để tìm thêm một số biểu mẫu khác liên quan đến các nghiệp vụ và quy trình của nhà trường được nằm trong phần [phụ lục 9.1](#) các biểu mẫu dưới đây được sử dụng để phục vụ trong phạm vi của đề án.

⁴ Chức năng được xây dựng để hỗ trợ cấp quyền truy cập người dùng

Bảng 3: Biểu mẫu - Bảng điểm học sinh

SỞ GDĐT TP HỒ CHÍ MINH
Trường THPT Trần Hưng Đạo

(BMI)
Tp. Hồ Chí Minh, ngày tháng năm

BẢNG ĐIỂM HỌC SINH
Học kỳ x | Năm học 201x – 201x

Mã số học sinh: **Họ và tên:**
Lớp:

STT	Môn học	Miệng		1 Tiết	Cuối kỳ	Tổng kết	Ghi chú
1	Toán học						
2	Vật lí						
3	Hóa học						
4	Sinh học						
5	Tin học						
6	Ngữ văn						
7	Lịch sử						
8	Địa lí						
9	Ngoại ngữ						
10	GDCD						
11	Công nghệ						
12	Thể dục						
13	GDQP						
Kết quả HK		ĐBQ	XH	Hạnh kiểm	Danh hiệu	Phép	Không phép
Kết quả CN							

Tp. Hồ Chí Minh, ngày tháng năm
 Giáo viên chủ nhiệm

Bảng 4: Biểu mẫu - Bảng điểm lớp học

SỞ GDĐT TP HỒ CHÍ MINH
Trường THPT Trần Hưng Đạo

(BM2)
Tp. Hồ Chí Minh, ngày tháng năm

BẢNG ĐIỂM LỚP HỌC 1xAxx
Học kỳ x | Năm học 201x – 201x

Mã giáo viên: Giáo viên chủ nhiệm:

STT	Họ và tên	Toán	Ngữ văn	Sinh học	Vật lý	Hoá học	Lịch sử	Địa lý	Ngoại ngữ	Giáo dục công dân	Quốc phòng	Thể dục	Công nghệ	Tin học

Xác nhận của BGH trường

Tp. Hồ Chí Minh, ngày tháng năm
Giáo viên chủ nhiệm

Bảng 5: Biểu mẫu - Báo cáo hạnh kiểm

SỞ GD&ĐT TP HỒ CHÍ MINH
Trường THPT Trần Hưng Đạo

(BM3)
Tp. Hồ Chí Minh, ngày tháng năm

BÁO CÁO HẠNH KIỂM LỚP
Học kỳ x | Năm học 201x – 201x

Mã giám thị: **Giám thị:**

STT	Học sinh	Số ngày nghỉ	Số lần vi phạm	Số lần cảnh cáo	Hạnh kiểm	Ghi chú

Xác nhận của BGH trường

Tp. Hồ Chí Minh, ngày tháng năm
Trưởng phòng giám thị

SỞ GDĐT TP HỒ CHÍ MINH
Trường THPT Trần Hưng Đạo

(BM4)
Tp. Hồ Chí Minh, ngày tháng năm

PHIẾU YÊU CẦU MƯỢN PHÒNG

Mã người mượn: Họ và tên:
Lớp:
Phòng mượn: Ngày mượn phòng:
Lý do mượn phòng:

Loại hình:

☐ Học lớp

☐ Học nhóm

☐ Tổ chức sự kiện

☐ Khác

Xác nhận của phòng giám thị

Tp. Hồ Chí Minh, ngày tháng năm
Người mượn phòng

Bảng 7: Biểu mẫu - phiếu xác nhận trả phòng

SỞ GD&ĐT TP HỒ CHÍ MINH
Trường THPT Trần Hưng Đạo

(BM5)
Tp. Hồ Chí Minh, ngày tháng năm

PHIẾU XÁC NHẬN TRẢ PHÒNG

Mã người trả: **Họ và tên:**
Lớp:
Phòng mượn: **Ngày trả phòng:**

Xác nhận của phòng giám thị

Tp. Hồ Chí Minh, ngày tháng năm
Người trả phòng

Bảng 8: Biểu mẫu - PHIẾU TRA CỨU THÔNG TIN HỌC SINH

SỞ GDĐT TP HỒ CHÍ MINH
Trường THPT Trần Hưng Đạo

(BM6)
Tp. Hồ Chí Minh, ngày tháng năm

PHIẾU TRA CỨU THÔNG TIN HỌC SINH

Mã người tra cứu: Họ và tên:
Mã học sinh: Họ và tên:
Lớp: Ngày tháng năm sinh:
Địa chỉ nhà:
Họ và tên phụ huynh: Số điện thoại:

Xác nhận của phòng giám thị

Bảng 9: Biểu mẫu - PHIẾU TRA CỨU CÁC PHÒNG ĐANG ĐƯỢC MƯỢN

SỞ GD&ĐT TP HỒ CHÍ MINH
Trường THPT Trần Hưng Đạo

(BM7)
Tp. Hồ Chí Minh, ngày tháng năm

PHIẾU TRA CỨU CÁC PHÒNG ĐANG ĐƯỢC MƯỢN
Thời gian từ ngày → tháng

STT	Ngày mượn	Mã phòng	Tên phòng	Sức chứa	Người mượn	Số điện thoại	Loại hoạt động	Ghi chú

Xác nhận của phòng giám thị

1.4.1 Sổ sách, hồ sơ

1.4.1.1 Hồ sơ của nhà trường

○ Kế hoạch:

- Kế hoạch chiến lược phát triển nhà trường (theo giai đoạn);
- Kế hoạch thực hiện nhiệm vụ năm học;
- Kế hoạch công tác của Hiệu trưởng, Phó Hiệu trưởng;
- Kế hoạch phân công giảng dạy và công tác kiêm nhiệm;
- Kế hoạch thanh kiểm tra nội bộ trường học.
- Kế hoạch và hồ sơ công tác Kiểm định chất lượng giáo dục.

○ Hồ sơ, sổ sách theo dõi hoạt động giáo dục trong nhà trường:

- Thời khóa biểu áp dụng trong năm học;
- Sổ nghị quyết của nhà trường và nghị quyết của Hội đồng trường;
- Hồ sơ thi đua;
- Hồ sơ thanh kiểm tra, đánh giá đơn vị trường học;
- Hồ sơ kỷ luật;
- Sổ quản lý và hồ sơ lưu trữ các văn bản, công văn đi, đến;
- Hồ sơ quản lý tài sản, thiết bị;
- Hồ sơ quản lý tài chính;
- Hồ sơ quản lý thư viện;
- Hồ sơ bàn giao và cam kết thực hiện chất lượng giáo dục;
- Hồ sơ thực hiện cuộc vận động “Xây dựng trường học thân thiện, học sinh tích cực”;
- Hồ sơ quản lý dạy thêm - học thêm trong và ngoài nhà trường (nếu có);
- Hồ sơ theo dõi phổ cập giáo dục (nếu có).

○ Hồ sơ quản lý học sinh:

- Hồ sơ quản lý, cấp phát văn bằng, chứng chỉ;
- Sổ đăng bộ;
- Sổ theo dõi học sinh chuyển đi, chuyển đến;
- Sổ gọi tên và ghi điểm;
- Sổ ghi đầu bài;
- Học bạ học sinh;
- Phiếu phối hợp giáo dục học sinh;
- Hồ sơ tuyển sinh lớp đầu cấp (lớp 6 hoặc lớp 10);
- Hồ sơ kiểm tra lại, rèn luyện thêm hạnh kiểm, xét lưu ban và lên lớp;

- Hồ sơ quản lý Hoạt động GD nghề phổ thông, Hướng nghiệp và HD NGLL;
- Hồ sơ theo dõi sức khỏe học sinh;
- Hồ sơ giáo dục đối với học sinh khuyết tật (nếu có);

1.4.1.1.1 Hồ sơ chuyên môn

- Kế hoạch Tổ chuyên môn;
- Kế hoạch thực hiện chế độ kiểm tra các môn học / học kỳ;
- Sổ biên bản sinh hoạt Tổ, nhóm chuyên môn;
- Sổ biên bản đánh giá kết quả thực hiện chuyên đề chuyên môn;
- Kế hoạch thực hiện thí nghiệm – thực hành, sử dụng thiết bị, đồ dùng dạy học, ứng dụng công nghệ thông tin.

1.4.1.1.2 Hồ sơ của giáo viên

- Giáo án (bài soạn);
- Sổ ghi kế hoạch giảng dạy theo tuần;
- Sổ dự giờ, thăm lớp;
- Sổ điểm cá nhân;
- Sổ ghi chép hội họp và sinh hoạt chuyên môn;
- Hồ sơ chủ nhiệm (đối với giáo viên làm công tác chủ nhiệm lớp).

1.5. Thủ tục, quy trình

- Quy trình soạn thảo, trình ký văn bản
- Quy trình nhận, chuyển văn bản đến từ Email
- Quy trình chuyển giao văn bản đi
- Quy trình đóng dấu
- Quy trình cấp giấy giới thiệu, giấy công tác, giấy đi đường
- Quy trình tham gia BHYT của CB, GV, NV
- Quy trình xét nâng lương hằng năm
- Quy trình đóng BHXH hằng tháng
- Quy trình thanh toán chế độ ốm đau, thai sản
- Quy trình tiếp nhận, gia hạn, quản lý hộ chiếu của học sinh quốc tế
- Quy trình giáo viên đăng ký mượn đồ dùng học cụ
- Quy trình giáo viên đăng ký sử dụng phòng đa năng
- Quy trình giáo viên đăng ký sử dụng phòng thí nghiệm thực hành
- Quy trình giáo viên đăng ký sử dụng phòng vi tính
- Quy trình quản lý và sử dụng khu vực hội trường
- Quy trình cung cấp nước uống đóng chai
- Quy trình tổ chức hoạt động thư viện

- Quy trình phục vụ bạn đọc tại thư viện
- Quy trình trực phòng và phục vụ bạn đọc
- Quy trình tổng hợp báo cáo tuần
- Quy trình tổng hợp báo cáo tháng
- Quy trình pho-to tài liệu
- Quy trình đăng ký thao giảng
- Quy trình sử dụng, nhập điểm sổ cái
- Quy trình liên thẩm sỹ số học sinh hằng tháng
- Quy trình mua sắm CSVN, trang thiết bị, cải tạo, sửa chữa, trường, lớp
- Quy trình thủ tục tuyển sinh
- Quy trình thủ tục học sinh nhập học
- Quy trình thủ tục học sinh chuyển trường
- Quy trình học sinh xin xác nhận là học sinh của trường
- Quy trình làm thẻ học sinh
- Quy trình thủ tục học sinh nhận bằng tốt nghiệp
- Quy trình học sinh xin nghỉ
- Quy trình xử lý học sinh vi phạm
- Quy trình thủ tục thanh toán chế độ BHYT cho học sinh

Chương 2: Các công nghệ sử dụng

2.1. Tổng quan về NodeJS



Hình 2: Logo NodeJS

NodeJS là nền tảng framework được xây dựng về kế thừa từ Javascript nên được thừa hưởng những ưu điểm của Javascript như chạy đa nền tảng, nhanh, cú pháp đơn giản, thư viện module đa dạng và phong phú. Do đó NodeJS được cộng đồng rất yêu thích trong việc xây dựng các ứng dụng web giúp đơn giản hoá việc lập trình và giảm thời gian ở mức thấp nhất.

Bên cạnh các ưu điểm trên, nhắc đến NodeJS là nhắc đến xử lý các yêu cầu theo thời gian thực, do đó có thể đáp ứng được số lượng lớn các request đồng thời tại một thời điểm mà không làm ảnh hưởng đến server. Vì vậy NodeJS được đánh giá là nền tảng nhẹ mà thực thi rất nhanh. Các tiến trình đơn giản nhưng hiệu năng cao, Node.js sử dụng một mô hình luồng đơn (single thread) với các sự kiện lặp. Các cơ chế sự kiện giúp Server trả lại các phản hồi với một cách không khóa và tạo cho Server hiệu quả cao ngược lại với các cách truyền thống tạo ra một số lượng luồng hữu hạn để quản lý request

2.2. Tổng quan về Sequelize

Để sử dụng được MySQL trên NodeJS, nền tảng này phải sử dụng thư viện Sequelize để kết nối từ NodeJS đến MySQL. Có thể định nghĩa, Sequelize là một ORM (Object/Relational Mapper) dành cho NodeJS hỗ trợ việc kết nối đến các hệ quản trị cơ sở dữ liệu MySQL, PostgreSQL, SQLite, ... Mặc dù, Sequelize là thư viện mới được cộng đồng xây dựng, nhưng đã được cộng đồng mạng cũng như các công ty

lớn khác sử dụng bởi khả năng tiện lợi và mở rộng cao. Tức có thể chạy được các hệ cơ sở dữ liệu quan hệ trên các nền tảng chạy bất đồng bộ vốn thường được biết đến việc sử dụng No-SQL. Do đó, việc sử dụng Sequelize giúp cho các công ty có thể nâng cấp công nghệ cho trang web của mình mà không phải bị tác động đến việc làm sạch lại dữ liệu.



Hình 3: Logo Sequelize

2.3. Tổng quan về Bootstrap và Html/css



Hình 4: Logo Bootstrap

Bootstrap là một framework cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn Bootstrap bao gồm các HTML templates, CSS templates và Javascript tạo ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin Javascript trong nó. Giúp cho việc thiết kế reponsive của bạn dễ dàng hơn và

nhanh chóng hơn.

Một số điểm thuận lợi khi sử dụng bootstrap:

- Rất dễ để sử dụng: Nó đơn giản vì nó được base trên HTML, CSS và Javascript chỉ cần có kiến thức cơ bản về 3 ngôn ngữ đó là có thể sử dụng bootstrap tốt.
- Responsive: Bootstrap xây dựng sẵn responsive css trên các thiết bị Iphones, tablets, và desktops. Tính năng này khiến cho người dùng tiết kiệm được rất nhiều thời gian trong việc tạo ra một website thân thiện với các thiết bị điện tử, thiết bị cầm tay.
- Tương thích với trình duyệt: Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera). Tuy nhiên, với IE browser, Bootstrap chỉ hỗ trợ từ IE9 trở lên. Điều này vô cùng dễ hiểu vì IE8 không support HTML5 và CSS3.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Hình 5: Cấu trúc cơ bản của HTML

HTML là chữ viết tắt của Hypertext Markup Language, là ngôn ngữ đánh dấu siêu văn bản được sử dụng rộng rãi nhất để viết các trang Web.

- Hypertext là cách mà các trang Web (các tài liệu HTML) được kết nối với nhau. Và như thế, đường link có trên trang Web được gọi là Hypertext.
- Như tên gọi đã gợi ý, HTML là ngôn ngữ đánh dấu bằng thẻ (Markup Language), nghĩa là bạn sử dụng HTML để đánh dấu một tài liệu text bằng

các thẻ (tag) để nói cho trình duyệt Web cách để cấu trúc nó để hiển thị ra màn hình.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightblue;
}
h1 {
  color: white;
  text-align: center;
}
p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
<body>

<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

CSS là viết tắt của Cascading Style Sheet, là một design language đơn giản được sử dụng để làm đơn giản hóa tiến trình trình bày các trang web.

Có thể hiểu đơn giản là, CSS xử lý phần Look và Feel của một trang web. Sử dụng CSS, bạn có thể điều khiển màu văn bản (text color), font style, khoảng cách giữa các đoạn văn, kích cỡ các cột, hình nền hoặc màu nền,

Hình 6: Cấu trúc CSS

2.4. Quản lý thư viện mã nguồn mở NPM



Hình 7: Cấu trúc cơ bản của CSS

NPM viết tắt của Node package manager là một công cụ tạo và quản lý các thư viện lập trình Javascript cho Node.js, công cụ này là thật sự cần thiết cho thế giới mã nguồn mở. Sử dụng NPM để cài đặt, chia sẻ và phân phối mã nguồn; quản lý vào các dự án của bạn; chia sẻ và nhận thông tin phản hồi với những nhà phát triển khác.

Trong cộng đồng Javascript, các lập trình viên chia sẻ hàng trăm nghìn các đoạn code giúp cho các dự án mới tránh phải viết lại các thành phần cơ bản, các thư viện lập trình hay thậm chí cả các framework. Mỗi đoạn code này có thể phụ thuộc vào rất nhiều các mã nguồn mở khác, thật may mắn khi các công cụ quản lý thư viện ra đời, nếu không sẽ mất rất nhiều công sức trong việc quản lý các thư viện này.

Nếu bạn đã làm việc với Javascript một thời gian, bạn có thể đã nghe nói về NPM. NPM làm cho nó dễ dàng cho các nhà phát triển Javascript để chia sẻ mã mà họ đã tạo ra để giải quyết vấn đề cụ thể, và cho các nhà phát triển khác sử dụng lại mã đó và rất dễ dàng để kiểm tra xem họ đã thực hiện bất kỳ bản cập nhật nào cho nó

2.5. Hệ thống quản lý mã nguồn Github

GitHub là một dịch vụ lưu trữ trên web dành cho các dự án có sử dụng hệ thống kiểm soát Git revision. GitHub cung cấp chức năng social network như feeds, followers và

network graph để các developer học hỏi kinh

nghiệm làm việc thông qua lịch sử commit. Nếu như comment dùng để mô tả chức năng của đoạn code, thì commit message trên Git dùng để mô tả hành động mà dev vừa thực hiện trên code.



Hình 9: Logo Github

Tại sao lại sử dụng GitHub???

- GitHub dễ sử dụng, an toàn và nhanh chóng.
- Có thể giúp quy trình làm việc code theo nhóm đơn giản hơn rất nhiều bằng việc kết hợp các phân nhánh (branch).
- Bạn có thể làm việc ở bất cứ đâu vì chỉ cần clone mã nguồn từ kho chứa hoặc clone một phiên bản thay đổi nào đó từ kho chứa, hoặc một nhánh nào đó từ kho chứa.

Dễ dàng trong việc triển khai sản phẩm.

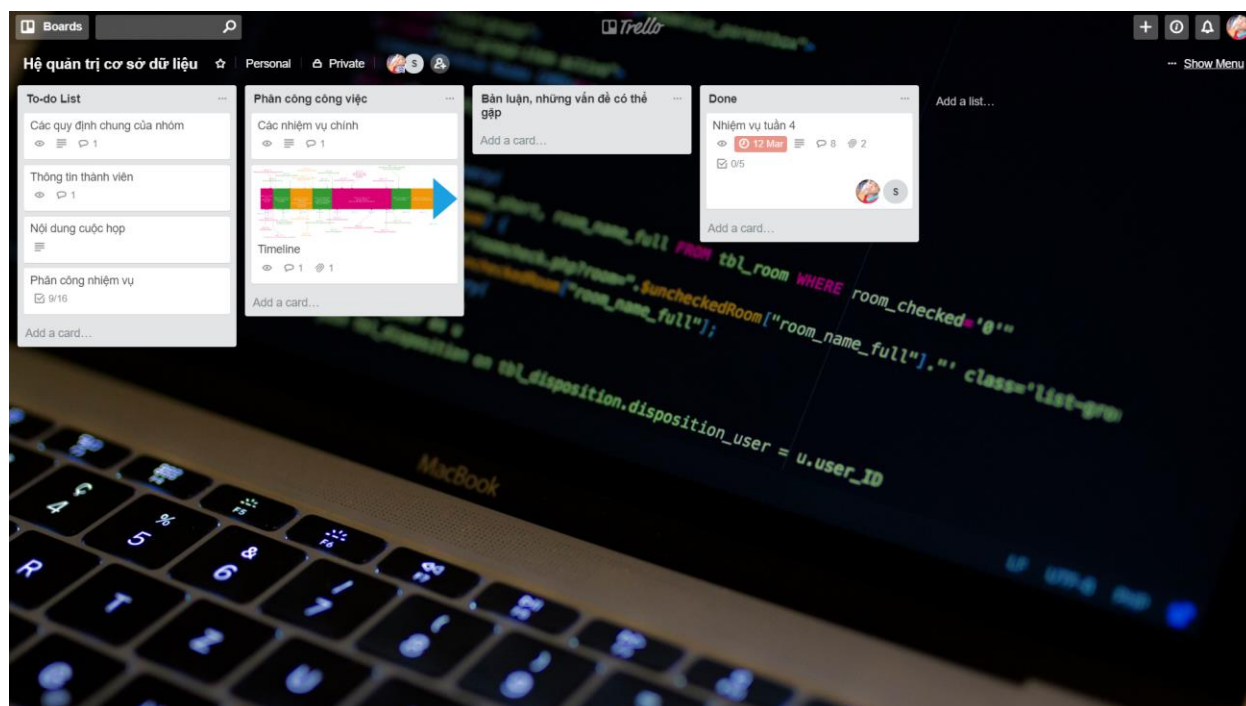
2.6. Quản lý dự án bằng nền tảng Trello

Trello là một công cụ để phối hợp công việc hiệu quả giúp cho mọi người trong team chỉ cần nhìn qua là biết được có những đầu việc nào, ai đang làm gì, và làm đến giai đoạn nào rồi. Trello là một công cụ Freemium, tức là bạn có thể xài miễn phí cho đến khi cần những

tích năng nâng cao và đặc biệt chỉ có bản trả tiền mới có thì bạn mới phải trả.

Trello có những thành phần cơ bản sau:

- **Card:** là một thẻ thông tin có Title, Description dùng để lưu trữ công việc. Một Card có thể dùng để ghi task (công việc), feature (tính năng sản phẩm), question (câu hỏi), note (ghi chú)... tùy cách bạn sử dụng. Mỗi card sẽ có checklist để bạn chia đầu việc nhỏ hơn. Bạn có thể comment, attach hình ảnh trên card để thuận tiện cho việc trao đổi giữa các thành viên.
- **List:** là một danh sách tổng hợp các card thuộc chung 1 tình trạng hoặc 1 tính chất khác nhau. Ví dụ: List Doing bao gồm những card đang được thực hiện. List Done chứa những card đã hoàn thành.
- **Board:** là một bảng thường tương đương với 1 project hoặc 1 mảng công việc. Trong một board có thể có nhiều List để chia tình trạng của những đầu việc của project đó cho tiện theo dõi.
- **Organization:** là tổng hợp những board và member của 1 công ty



Hình 10: Một giao diện của Trello

Tại sao lại là Trello???

- **Simple**

Thật ra Trello cũng như tập hợp 1 đồng Post-It note mà mọi người ai cũng đều sử dụng qua để ghi chú lại những việc mình cần làm. Trello không có quá nhiều khái niệm phức tạp, nhìn vào có thể hiểu liền và sử dụng ngay được. Tính đơn giản của Trello còn nằm ở chỗ những thao tác trong đó cũng được đơn giản hóa cực kỳ:

- **Sophisticated**

Trello không có phức tạp (complex) mà là tinh vi (sophisticated). Mặc dù bạn sử dụng rất dễ dàng, nhưng Trello cũng tạo điều kiện để bạn quản lý công việc hiệu quả thông qua hàng loạt những tính năng và tiện ích bên dưới – rất tinh vi nhưng không làm bạn rối.

- **Flexible**

Vì vừa Simple và Sophisticated, cho nên Trello rất linh hoạt. Linh hoạt ở chỗ là có thể hỗ trợ cho bạn quản lý theo cách bạn muốn. Nó không bắt ép bạn phải theo một quy chuẩn nào cả. Hơn nữa bạn có thể dùng Trello cho nhiều thứ khác nhau chứ không phải chỉ là quản lý dự án thông thường.

- **Free**

Mặc dù Trello có gói Business Class (50\$/tháng/công ty) tuy nhiên phiên bản Free vẫn rất đầy đủ và thoải mái để sử dụng cho tất cả những nhu cầu thông thường.

Chương 3: Hệ quản trị cơ sở dữ liệu MySQL

3.1. Tổng quan về hệ quản trị cơ sở dữ liệu MySQL

MySQL là một hệ quản trị cơ sở dữ liệu được nhóm tác giả MySQL AB công bố lần đầu tiên vào ngày 23-05-1995. Trải qua những năm tháng thăng trầm với nhiều chủ khác nhau như MySQL AB (1995 - 2007), Sun Microsystems (02/2008 – 01/2010) và hiện nay là Oracle (từ 27-01-2010), MySQL đã nâng cấp, cải tiến và cho ra đời nhiều phiên bản khác nhau, được rất nhiều người lựa chọn.



Hình 11: Logo MySQL

Sau nhiều năm nỗ lực phát triển và đi vào cộng đồng công nghệ thông tin, MySQL ngày nay đã trở thành một hệ quản trị CSDL nổi tiếng và thông dụng. Cơ sở dữ liệu MySQL là sự lựa chọn phổ biến cho các ứng dụng web, thương mại điện tử (E-commerce), ứng dụng xử lý giao dịch trực tuyến (Online Transaction Processing - OLTP). Với khả năng giao dịch an toàn, tích hợp đầy đủ các tiện ích, dễ sử dụng, mở rộng, tăng hiệu suất, chi phí thấp ... đã đưa MySQL trở thành một CSDL mã nguồn mở phổ biến nhất thế giới. Rất nhiều trang web có lượng truy cập nhiều nhất trên thế giới như Facebook, Google, Twitter, Uber, Amazon, Digg và Booking.com ... đã chọn MySQL cho các ứng dụng kinh doanh quan trọng.

Sự ra đời của MySQL 5.7 vào tháng 10-2015 đã đánh dấu thành công vượt bậc của MySQL với tốc độ nhanh hơn, bảo mật hơn, nhiều tiện ích hơn so với version 5.6. MySQL 5.7 có những tính năng nổi trội như: hiệu suất hơn MySQL đến 3 lần, cải tiến InnoDB, định tuyến (MySQL Router), cải tiến replication (đa nguồn, đa luồng).

3.2. Tổng quan về InnoDB

InnoDB là một loại kiểu lưu trữ bảng (storage engine) trong MySQL dùng cho nhiều mục đích với độ tin cậy và hiệu năng cao. Từ phiên bản MySQL 5.5.5, kiểu lưu trữ bảng mặc định cho bảng mới tạo là InnoDB thay vì MyISAM. InnoDB thích hợp

cho các ứng dụng hoặc trang web có tần suất cập nhật dữ liệu cao như ngân hàng, mạng xã hội, diễn đàn, ...

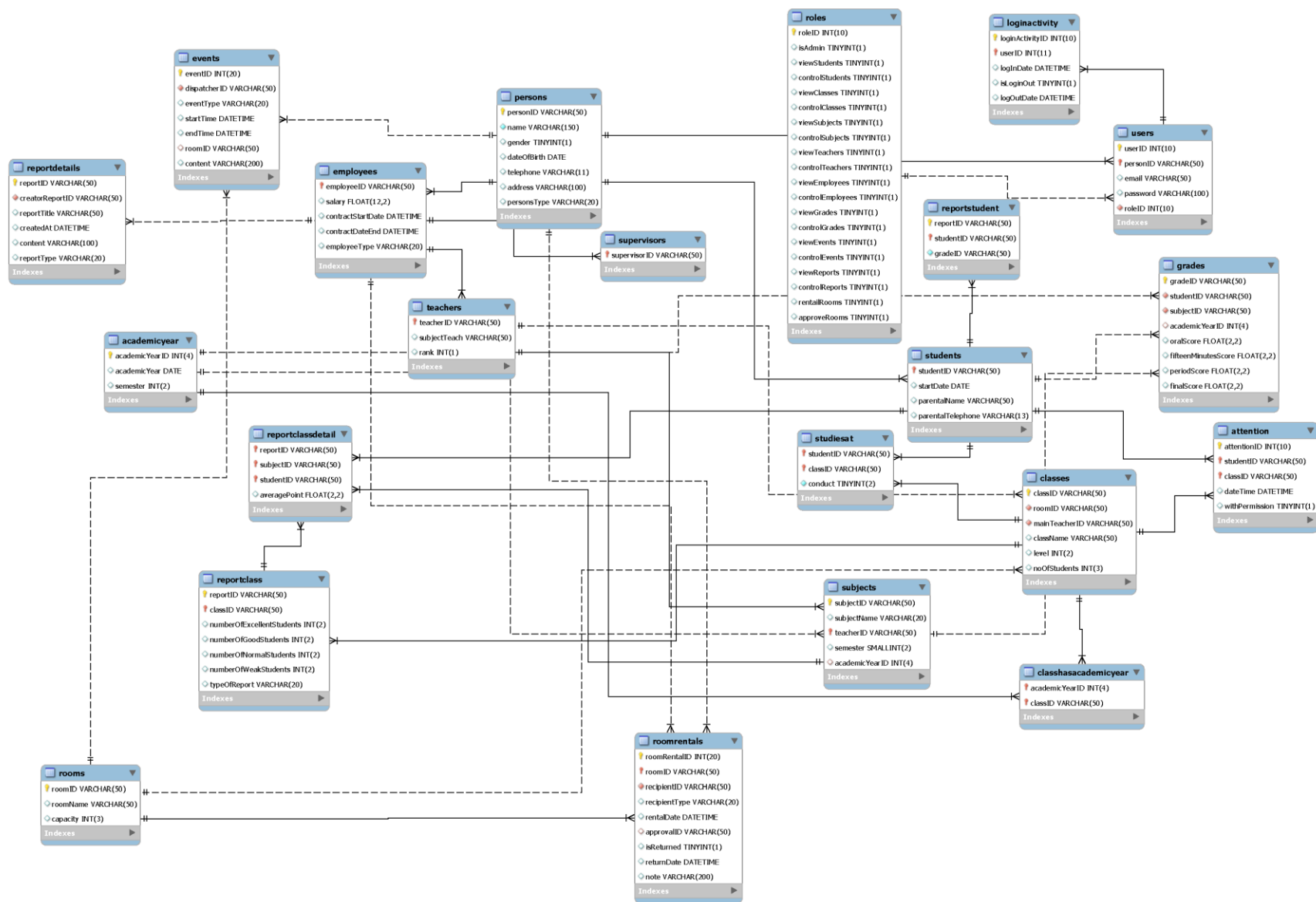
Lợi ích của InnoDB:

- Các tác vụ với CSDL đều tuân theo chuẩn ACID, hỗ trợ transactions với các chức năng commit, rollback, and phục hồi sau sự cố để bảo vệ dữ liệu người dùng.
- Hoạt động theo cơ chế Row-level locking, khi có thao tác trên bảng thì chỉ có bản ghi đang thao tác bị khóa, các hoạt động khác trên bảng vẫn diễn ra bình thường.
- Bảng theo kiểu InnoDB được lưu trữ để tối ưu hóa câu truy vấn theo khóa chính. Mỗi bảng sẽ có chỉ số khóa chính gọi là clustered index giúp tổ chức dữ liệu nhằm giảm thiểu tác vụ nhập xuất tìm khóa chính.
- Không như các storage engine khác trong MySQL, InnoDB hỗ trợ ràng buộc FOREIGN KEY để duy trì tính toàn vẹn của dữ liệu. Các tác vụ insert, update, delete sẽ được kiểm tra để đảm bảo rằng dữ liệu được thống nhất giữa các bảng khác nhau.

Chương 4: Thiết kế và xây dựng cơ sở dữ liệu

4.1. Xây dựng và thiết kế cơ sở dữ liệu

4.1.1. Sơ đồ logic



Hình 12: Sơ đồ logic cơ sở dữ liệu

4.1.1. Mô hình quan hệ

persons(<u>personID</u> , name, gender, dateOfBirth, telephone, address, personsType)
employees(<u>employeeID</u> , salary, contracStartDate, contracDateEnd, employeeType)
teachers(<u>teacherID</u> , subjectTeach, rank)
supervisors(<u>supervisorID</u>)
students(<u>studentID</u> , startDate, parentalName, parentalTelephone)
rooms(<u>roomID</u> , roomName, capacity)
roomRentals(<u>roomRentalID</u> , <u>roomID</u> , <u>recipientID</u> , recipientType, rentalDate, <u>approvalID</u> , isReturned, returnDate, Note)
academicYear(<u>academicYearID</u> , academicYear, semester)
classes(<u>classID</u> , <u>roomID</u> , <u>mainTeacherID</u> , className, level, noOfStudents)
classHasAcademicYear(<u>academicYearID</u> , <u>classID</u>)
studiesAt(<u>studentID</u> , <u>classID</u> , conduct)
events(<u>eventID</u> , <u>dispatcherID</u> , eventType, startTime, endTime, <u>roomID</u> , content)
subjects(<u>subjectID</u> , subjectName, <u>teacherID</u> , smester, <u>academicYearID</u>)
grades(<u>gradeID</u> , <u>studentID</u> , <u>subjectID</u> , <u>academicYearID</u> , oralScore, fifteenMinutesScore, periodScore, finalScore)
reportDetails(<u>reportID</u> , <u>creatorReportID</u> , reportTitle, createdAt, content, reportType)
reportClass(<u>reportID</u> , <u>classID</u> , numberOfExcellentStudents, numberOfNormalStudents, numberOfWeakStudents, typeOfReport)
reportClassDetail(<u>reportID</u> , <u>subjectID</u> , <u>studentID</u> , averagePoint)
reportStudent(<u>reportID</u> , <u>studentID</u> , <u>gradeID</u>)

roles(roleID, isAdmin, viewStudents, controlStudents, viewClasses, controlClasses, viewSubjects, controlSubjects, viewTeaches, controlTeachers, viewEmployees, controlEmployee, viewGrades, controlGrades, viewEvents, controlEvents, viewReports, controlReports, rentailRooms, approveRooms)

users(userID, personID, email, password, roleID)

loginActivity(loginActivityID, userID, logInDate, isLoginOut, logOutDate)

attention(attentionID, studentID, classID, dateTime, withPermission)

4.1.2. Mô tả dữ liệu

Bảng 10: Mô tả dữ liệu - PERSONS

Bảng: PERSONS					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	PERSONSID	VARCHAR(50)	NOT NULL		
2	NAME	VARCHAR(50)	NOT NULL		
3	GENDER	TINYINT(1)		1	
4	DATEOFBIRTH	DATE			
5	TELEPHONE	VARCHAR(13)			
6	ADDRESS	VARCHAR(100)			
7	PERSONSTYPE	VARCHAR(20)			

Bảng 11: Mô tả dữ liệu - EMPLOYEES

Bảng: EMPLOYEES					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	EMPLOYEEID	VARCHAR(50)	NOT NULL		
2	SALARY	FLOAT(12,2)			
3	CONTRACTSTARTDATE	DATETIME			
4	CONTRACTDATEEND	DATETIME			
5	EMPLOYEEYPE	VARCHAR(20)			

Bảng 12: Mô tả dữ liệu - TEACHERS

Bảng: TEACHERS					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	TEACHERID	VARCHAR(50)	NOT NULL		
2	SUBJECTTEACH	VARCHAR(50)			
3	RANK	INT(1)			

Bảng 13: Mô tả dữ liệu - SUPERVISORS

Bảng: SUPERVISORS					
STT	Thuộc tính	Type	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	SUPERVISORID	VARCHAR(50)	NOT NULL		

Bảng 14: Mô tả dữ liệu - STUDENTS

Bảng: STUDENTS					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	STUDENTID	VARCHAR(50)	NOT NULL		
2	STARTDATE	DATE			
3	PARENTALNAME	VARCHAR(50)			
4	PARENTALTELEPHONE	VARCHAR(13)			

Bảng 15: Mô tả dữ liệu - ROOMS

Bảng: ROOMS					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	ROOMID	VARCHAR(50)	NOT NULL		
2	ROOMNAME	VARCHAR(50)			
3	CAPACITY	INT(3)			

Bảng 16: Mô tả dữ liệu - ROOMRENTALS

Bảng: ROOMRENTALS					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	ROOMRENTALID	INT(20)	NOT NULL	AUTO_INCREMENT	
2	ROOMID	VARCHAR(50)	NOT NULL		
3	RECIPIENTID	VARCHAR(50)	NOT NULL		
4	RECIPIENTTYPE	VARCHAR(20)			
5	RENTALDATE	DATETIME			

6	APPROVALID	VARCHAR(50)			
7	ISRETURNED	TINYINT(1)			
8	RETURNDATE	DATETIME			
9	NOTE	VARCHAR(200)			

Bảng 17: Mô tả dữ liệu - ACADEMICYEAR

Bảng: ACADEMICYEAR					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	ACADEMICYEARID	INT(4)	NOT NULL	AUTO_INCREMENT	
2	ACADEMICYEAR	DATE			
3	SEMESTER	INT(2)			

Bảng 18: Mô tả dữ liệu - CLASSES

Bảng: CLASSES					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	CLASSID	VARCHAR(50)	NOT NULL		
2	ROOMID	VARCHAR(50)	NOT NULL		
3	MAINTeacherID	VARCHAR(50)	NOT NULL		
4	CLASSNAME	VARCHAR(50)			
5	LEVEL	INT(2)		10	
6	NOOFSTUDENT	INT(3)			

Bảng 19: Mô tả dữ liệu - CLASSHASACADEMICYEAR

Bảng: CLASSHASACADEMICYEAR					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	ACADEMICYEARID	INT(4)	NOT NULL		
2	CLASSID	VARCHAR(50)	NOT NULL		

Bảng 20: Mô tả dữ liệu - STUDIESAT

Bảng: STUDIESAT					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	STUDENTID	VARCHAR(50)	NOT NULL		
2	CLASSID	VARCHAR(50)	NOT NULL		
3	CONDUCT	TINYINT(2)	NOT NULL	10	

Bảng 21: Mô tả dữ liệu - EVENTS

Bảng: EVENTS					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	EVENTID	INT(10)	NOT NULL	AUTO_INCREMENT	
2	DISPATCHERID	VARCHAR(50)	NOT NULL		
3	EVENTTYPE	VARCHAR(20)			
4	STARTTIME	DATETIME			
5	ENDTIME	DATETIME			
6	ROOMID	VARCHAR(50)	NOT NULL		
7	CONTENT	VARCHAR(200)			

Bảng 22: Mô tả dữ liệu - SUBJECTS

Bảng: SUBJECTS					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	SUBJECTID	VARCHAR(50)	NOT NULL		
2	SUBJECTNAME	VARCHAR(20)			
3	TEACHERID	VARCHAR(50)	NOT NULL		
4	SEMESTER	SMALLINT(2)			
5	ACADEMICYEARID	INT(4)	NOT NULL		

Bảng 23: Mô tả dữ liệu - GRADES

Bảng: GRADES					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	GRADEID	VARCHAR(50)	NOT NULL		
2	STUDENTID	VARCHAR(50)	NOT NULL		
3	SUBJECTID	VARCHAR(50)	NOT NULL		
4	ACADEMICYEARID	INT(4)	NOT NULL		
5	ORALSCORE	FLOAT(2,2)			
6	FIFTEENMINUTEESSCORE	FLOAT(2,2)			
7	PERIODSCORE	FLOAT(2,2)			
8	FINALSCORE	FLOAT(2,2)			

Bảng 24: Mô tả dữ liệu - REPORTDETAILS

Bảng: REPORTDETAILS					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	REPORTID	VARCHAR(50)	NOT NULL		
2	CREATORREPORTID	VARCHAR(50)	NOT NULL		
3	REPORTTITLE	VARCHAR(50)			
4	CREATEDAT	DATETIME			
5	CONTENT	VARCHAR(100)			

Bảng 25: Mô tả dữ liệu - REPORTCLASS

Bảng: REPORTCLASS					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	REPORTID	VARCHAR(50)	NOT NULL		
2	CLASSID	VARCHAR(50)	NOT NULL		
3	NUMBEROFEXCELLENTSTUDENTS	INT(2)			
4	NUMBEROFGOODSTUDENTS	INT(2)			
5	NUMBEROFWEAKSTUDENTS	INT(2)			
6	TYPEOFREPORT	VARCHAR(20)			

Bảng 26: Mô tả dữ liệu - REPORTCLASSDETAIL

Bảng: REPORTCLASSDETAIL					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	REPORTID	VARCHAR(50)	NOT NULL		
2	SUBJECTID	VARCHAR(50)	NOT NULL		
3	STUDENTID	VARCHAR(50)	NOT NULL		
4	AVERAGEPOINT	FLOAT(2,2)			

Bảng 27: Mô tả dữ liệu - ROLES

Bảng: ROLES					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	ROLEID	INT(10)	NOT NULL	AUTO_INCREMENT	
2	ISADMIN	TINYINT(1)		0	
3	VIEWSTUDENTS	TINYINT(1)		0	
4	CONTROLSTUDENTS	TINYINT(1)		0	
5	VIEWCLASSES	TINYINT(1)		0	
6	CONTROLCLASSES	TINYINT(1)		0	
7	VIEWSUBJECTS	TINYINT(1)		0	
8	CONTROLSUBJECT	TINYINT(1)		0	
9	VIEWTEACHERS	TINYINT(1)		0	
10	CONTROLTEACHER	TINYINT(1)		0	
11	VIEWEMPLOYEES	TINYINT(1)		0	
12	CONTROLEMPLOYEES	TINYINT(1)		0	
13	VIEWGRADES	TINYINT(1)		0	
14	CONTROLGRADES	TINYINT(1)		0	
15	VIEWEVENTS	TINYINT(1)		0	
16	CONTROLEVENTS	TINYINT(1)		0	
17	VIEWREPORTS	TINYINT(1)		0	
18	CONTROLREPORT	TINYINT(1)		0	
19	RENTAILROOMS	TINYINT(1)		0	
20	APPROVEROOMS	TINYINT(1)		0	

Bảng 28: Mô tả dữ liệu - REPORTSTUDENT

Bảng: REPORTSTUDENT					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	REPORTID	VARCHAR(50)	NOT NULL		
2	STUDENTID	VARCHAR(50)	NOT NULL		
3	GRADEID	VARCHAR(50)	NOT NULL		

Bảng 29: Mô tả dữ liệu - USERS

Bảng: USERS					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	USERID	INT(10)	NOT NULL	AUTO_INCREMENTS	
2	PERSONid	VARCHAR(50)	NOT NULL		
3	EMAIL	VARCHAR(50)			
4	PASSWORD	VARCHAR(100)			
5	ROLEID	INT(10)	NOT NULL		

Bảng 30: Mô tả dữ liệu - LOGINACTIVITY

Bảng: LOGINACTIVITY					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	LOGINACTIVITYID	INT(10)	NOT NULL	AUTO_INCREMENT	

2	USERID	INT	NOT NULL		
3	LOGINDATE	DATETIME		NOW()	
4	ISLOGINOUT	TINYINT(1)		0	
5	LOGOUTDATE	DATETIME			

Bảng 31: Mô tả dữ liệu - ATTENTION

Bảng: ATTENTION					
STT	Thuộc tính	Loại	Ràng buộc	Giá trị khởi tạo	Ghi chú
1	ATTENTIONID	INT(10)	NOT NULL	AUTO_INCREMENT	
2	STUDENTID	VARCHAR(50)	NOT NULL		
3	CLASSID	VARCHAR(50)	NOT NULL		
4	DATETIME	DATETIME		NOW()	
5	WITHPERMISSION	TINYINT(1)		0	

4.1.3. Ràng buộc toàn vẹn

- **Ràng buộc 1: Các điểm miệng, 15ph, 1 tiết, cuối kì phải là những điểm lớn hơn hoặc bằng 0 và bé hơn hoặc bằng 10.**

Nội dung:

$\forall g \in \text{GRADES: } (g.\text{oralScore} \geq 0 \wedge g.\text{oralScore} \leq 10) \vee (g.\text{fifteenMinutesScore} \geq 0 \wedge g.\text{fifteenMinutesScore} \leq 10) \vee (g.\text{periodScore} \geq 0 \wedge g.\text{periodScore} \leq 10) \vee (g.\text{finalScore} \geq 0 \wedge g.\text{finalScore} \leq 10)$

Bối cảnh: Quan hệ GRADES

Bảng tầm ảnh hưởng:

	Thêm	Xóa	Sửa
GRADES	+ (oralScore, fifteenMinutesScore, periodScore, finalScore)	-	+ (oralScore, fifteenMinutesScore, periodScore, finalScore)

- **Ràng buộc 2: Mượn phòng phải mượn trước ngày mượn 2 ngày**

Nội dung:

$\forall r \in \text{ROOMRENTALS: } r.\text{rentalDate} - \text{CURDATE}() \geq 2$

Bối cảnh: Quan hệ ROOMRENTALS

Bảng tầm ảnh hưởng:

	Thêm	Xóa	Sửa
ROOMRENTALS	+ (rentalDate)	-	+ (rentalDate)

- **Ràng buộc 3: Mỗi giáo viên chủ nhiệm 1 lớp học duy nhất và lớp đó có số ≤ 45 học sinh**

Nội dung:

$\forall c1, c2 \in \text{CLASSES: } (c1.\text{mainTeacherID} \neq c2.\text{mainTeacherID}) \wedge (\text{noOfStudents} \leq 45)$

Bối cảnh: Quan hệ CLASSES

Bảng tầm ảnh hưởng:

	Thêm	Xóa	Sửa
CLASSES	+ (mainTeacherID, noOfStudents)	-	+ (mainTeacherID, noOfStudents)

- **Ràng buộc 4: Mã ID của 1 người là số cmnd gồm 9 số hoặc 12 số**

Nội dung:

$\forall p \in \text{PERSONS}: (\text{length}(p.\text{personID}) = 9) \vee (\text{length}(p.\text{personID}) = 12)$

Bối cảnh: Quan hệ PERSONS

Bảng tầm ảnh hưởng:

	Thêm	Xóa	Sửa
PERSONS	+ (personID)	-	+ (personID)

4.2. Trigger

4.2.1. Trigger trong MySQL

Trigger là một đối tượng được định danh trong CSDL và được gắn chặt với một sự kiện xảy ra trên một bảng nào đó (điều này có nghĩa là nó sẽ được tự động thực thi khi xảy ra một sự kiện trên một bảng). Các sự kiện này bao gồm: chèn (Insert), xóa (Delete) hay cập nhật (Update) một bảng. Trigger được đưa vào MySQL từ phiên bản 5.0.2.

Trigger được sử dụng để tạo ra các công việc tự động thay cho việc phải làm thủ công bằng tay như: kiểm tra dữ liệu, đồng bộ hóa dữ liệu, đảm bảo các mối quan hệ giữa các bảng, ...

Trigger trong MySQL chỉ được kích hoạt thông qua các câu lệnh SQL, chúng không được kích hoạt bằng các thay đổi được thực thi bởi các hàm API trong bảng mà không thông qua việc truyền các câu lệnh SQL tới máy chủ MySQL. Trong thực tế, chúng không được kích hoạt bằng các lệnh cập nhật thông qua NDB API.

Cú pháp của trigger trong MySQL:

```
CREATE [DEFINER = { user | CURRENT_USER }]
```

```
    TRIGGER trigger_name trigger_time trigger_event
```

Hệ

ON tbl_name FOR EACH ROW trigger_stmt

- *DEFINER*: xác định ngữ cảnh bảo mật sẽ được sử dụng khi kiểm tra quyền truy cập tại thời điểm trigger được kích hoạt.
- *trigger_name*: Tên của trigger.
- *trigger_time*: Xác định thời điểm thực thi của trigger. Thời điểm này có thể là BEFORE (trước) hoặc AFTER (sau) để chỉ báo rằng trigger sẽ được kích hoạt trước hoặc sau mỗi dòng được chỉnh sửa.
- *trigger_event*: xác định kiểu câu lệnh sẽ kích hoạt trigger. Nó chỉ có thể nhận một trong các giá trị sau đây:
 - + INSERT: Trigger sẽ được kích hoạt khi có một dòng mới được chèn vào bảng thông qua các câu lệnh như: INSERT, LOAD DATA, và REPLACE.
 - + UPDATE: Trigger được kích hoạt khi có một dòng bị chỉnh sửa bởi câu lệnh UPDATE
 - + DELETE: Trigger được thực hiện khi một dòng bị xóa khỏi bảng. Chú ý rằng các câu lệnh DROP TABLE và TRUNCATE sẽ không kích hoạt được trigger này.
- *tbl_name*: Tên bảng mà trigger sẽ hoạt động trên đó
- *trigger_stmt*: Câu lệnh sẽ được thực thi khi trigger được kích hoạt. Nếu như cần kích hoạt nhiều câu lệnh, chúng ta cần phải đưa chúng vào cấu trúc khối lệnh BEGIN ... END

4.2.2. Danh sách các Trigger

Bảng 32: Danh sách các Trigger

STT	TÊN	THAO TÁC	BẢNG	NỘI DUNG
1	<i>trig_checkins_garades</i>	insert	<i>garades</i>	Kiểm tra điểm số mà giáo viên nhập vào phải nằm trong khoảng từ 0 đến 10 điểm
2	<i>trg_checkins_timeroomRentals</i>	insert	<i>roomRentals</i>	Kiểm tra ngày mà người mượn phòng

				phải sớm ít nhất 2 ngày so với ngày đăng kí.
3	<i>trig_checkins_idpersons</i>	insert	persons	Kiểm tra số CMND của học sinh phải số đủ 9 số hoặc 12 số(nếu có thêm 3 số 0 đầu)
4	<i>trig_checkupd_garades</i>	update	<i>garades</i>	Kiểm tra điểm số mà giáo viên cập nhập lại phải nằm trong khoảng từ 0 đến 10 điểm
5	<i>trg_checkupd_timeroomRentals</i>	update	roomRentals	Kiểm tra ngày mà người mượn phòng cập nhập lại ngày mượn phải sớm ít nhất 2 ngày so với ngày cập nhập.
6	<i>trig_checkupd_idpersons</i>	update	persons	Kiểm tra số CMND của học sinh cần cập nhập lại phải số đủ 9 số hoặc 12 số(nếu có thêm 3 số 0 đầu)
7	<i>trg_insert_students</i>	insert	<i>students</i>	Thêm học sinh mới vào trong 1 lớp học
8	<i>trg_update_students</i>	update	<i>students</i>	Sửa lại thông tin của một học sinh
9	<i>trg_delete_students</i>	delete	<i>students</i>	Xóa lại thông tin của một học sinh

4.2.3. Mô tả một số Trigger

- **Tên:** *trig_checkins_garades*

Thao tác: insert

Trên bảng: *garades*

Nội dung: Kiểm tra điểm số mà giáo viên nhập vào phải nằm trong khoảng từ 0 đến 10 điểm.

Các bước thực hiện:

[1] Giáo viên nhập điểm số của học sinh vào hệ thống.

[2] Kiểm tra xem thông tin các cột điểm mà giáo viên nhập có thỏa: từ 0 đến 10 điểm hay không.

[2.1] Nếu thông tin nhập thỏa, thực hiện thao tác thêm, lưu lại.

[2.1] Ngược lại, thông báo lỗi cho giáo viên.

Code SQL:

```
DELIMITER $$
DROP TRIGGER IF EXISTS `trig_checkins_garades`;$$
CREATE DEFINER=`root`@`localhost`
TRIGGER `trig_checkins_garades`
BEFORE INSERT ON `grades`
FOR EACH ROW
BEGIN

IF (NEW. oralScore) <0 OR (NEW. oralScore) > 10
THEN
    SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Error: diem nhap phai lon hoac bang 0 va phai be hon hoac bang 10!';

IF (NEW. fifteenMinutesScore) <0 OR (NEW. fifteenMinutesScore) > 10
THEN
    SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Error: diem nhap phai lon hoac bang 0 va phai be hon hoac bang 10!';

IF (NEW. finalScore) <0 OR (NEW. finalScore) > 10
THEN
    SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Error: diem nhap phai lon hoac
```

bang 0 va phai be hon hoac bang 10!';

IF (NEW. periodScore) <0 OR (NEW. periodScore) > 10
THEN

SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Error: diem nhap phai lon hoac bang 0 va phai be hon hoac bang 10!';

END IF;

END IF;

END IF;

ENDIF;

END\$\$

delimiter ;

- **Tên: *trg_checkins_timeroomRentals***

Thao tác: insert

Trên bảng: *roomRentals*

Nội dung: Kiểm tra ngày mượn phòng phải sớm ít nhất là 2 ngày so với ngày đăng kí hiện tại.

Các bước thực hiện:

[1] Người đăng kí phòng chọn ngày muốn mượn.

[2]Kiểm tra xem ngày mượn có cách ít nhất 2 ngày so với ngày đăng kí hiện tại không.

[2.1] Nếu ngày mượn hợp lệ, cho phép thực hiện thao tác đăng kí.

[2.1] Ngược lại, thông báo lỗi cho người đang thực hiện đăng kí.

Code SQL:


```

DELIMITER $$
DROP TRIGGER IF EXISTS `trg_checkins_roomRentals`;
CREATE DEFINER=`root`@`localhost`
TRIGGER `trg_checkins_roomRentals`
BEFORE INSERT ON `roomrentals`
FOR EACH ROW
BEGIN

IF EXISTS ( SELECT * FROM roomrentals
            WHERE DATEDIFF(curdate(),rentalDate)) < 2
THEN
    SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Warning: ngay muOn phong phai
som hon it nhat 2 ngay so ngay hien tai!';
END IF;

END$$
delimiter ;

```

- **Tên: *trig_checkins_idpersons***

Thao tác: insert

Trên bảng: *persons*

Nội dung: Kiểm tra mã CMND được nhập có phải thuộc 9 số hoặc 12 số hay không(nếu thêm 3 số 0 đầu)

Các bước thực hiện:

[1].Nhập mã CMND của học sinh vào hệ thống

[2]Kiểm tra xem mã vừa nhập có thỏa 9 số hoặc 12 số hay không

[2.1] Nếu khóa, thực hiện thao tác thêm thông tin

[2.1] Ngược lại, thông báo lỗi cho người nhập.

Code SQL:

```

DELIMITER $$
DROP TRIGGER IF EXISTS `trig_checkins_persons`;
CREATE DEFINER=`root`@`localhost`

```

```

TRIGGER `trig_checkins_persons`
BEFORE INSERT ON `persons`
FOR EACH ROW
BEGIN

IF CHAR_LENGTH(NEW. personID) != 9
OR CHAR_LENGTH(NEW. personID) != 12
THEN
    SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'personID phai co 9 hoac 12 so';
END IF;

END$$
delimiter ;

```

- **Tên:** *trig_checkiupd_garades*

Thao tác: update

Trên bảng: *garades*

Nội dung: Kiểm tra điểm số mà giáo viên cập nhật lại phải nằm trong khoảng từ 0 đến 10 điểm.

Các bước thực hiện:

[1] Giáo viên cập nhật điểm số của học sinh vào hệ thống.

[2] Kiểm tra xem thông tin các cột điểm mà giáo viên cập nhật có thỏa: từ 0 đến 10 điểm hay không.

[2.1] Nếu thông tin nhập thỏa, thực hiện thao tác cập nhật.

[2.1] Ngược lại, thông báo lỗi cho giáo viên.

Code SQL:

```

DELIMITER $$
DROP TRIGGER IF EXISTS `trig_checkupd_garades`;$$
CREATE DEFINER=`root`@`localhost`

```

```

TRIGGER `trig_checkup_garades`
BEFORE UPDATE ON `grades`
FOR EACH ROW
BEGIN

IF (NEW. oralScore) <0 OR (NEW. oralScore) > 10
THEN
    SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Error: diem nhap phai lon hoac
bang 0 va phai be hon hoac bang 10!';

IF (NEW. fifteenMinutesScore) <0 OR (NEW. fifteenMinutesScore) > 10
THEN
    SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Error: diem nhap phai lon hoac
bang 0 va phai be hon hoac bang 10!';

IF (NEW. finalScore) <0 OR (NEW. finalScore) > 10
THEN
    SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Error: diem nhap phai lon hoac
bang 0 va phai be hon hoac bang 10!';

IF (NEW. periodScore ) <0 OR (NEW. periodScore ) > 10
THEN
    SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Error: diem nhap phai lon hoac
bang 0 va phai be hon hoac bang 10!';

END IF;
END IF;
END IF;
ENDIF;

END$$
delimiter ;

```

- **Tên:** *trig_checkupd_idpersons*

Thao tác: update

Trên bảng: *persons*

Nội dung: Kiểm tra mã CMND được cập nhật lại phải thuộc 9 số hoặc 12 số hay không(nếu thêm 3 số 0 đầu)

Các bước thực hiện:

[1].Cập nhật lại mã CMND của học sinh vào hệ thống

[2]Kiểm tra xem mã vừa cập nhật có thỏa 9 số hoặc 12 số hay không

[2.1] Nếu khóa, thực hiện thao tác cập nhật thông tin

[2.1] Ngược lại, thông báo lỗi cho người nhập.

Code SQL:

```
DELIMITER $$
DROP TRIGGER IF EXISTS `trig_checkupd_persons`;
CREATE DEFINER=`root`@`localhost`
TRIGGER `trig_checkupd_persons`
BEFORE UPDATE ON `persons`
FOR EACH ROW
BEGIN

IF CHAR_LENGTH(NEW. personID) != 9
OR CHAR_LENGTH(NEW. personID) != 12
THEN
    SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'personID phai co 9 hoac 12 so';
END IF;

END$$
delimiter ;
```

4.3. Store Procedure

4.3.1. Stored procedure trong MySQL

Stored procedure là một hoặc một tập các lệnh SQL được biên dịch để thực hiện cùng lúc mỗi khi được gọi đến. Về bản chất giống với hàm, thủ tục trong các ngôn ngữ lập trình khác. Stored procedure đã được đưa vào MySQL từ phiên bản 5.0.

Stored procedure giúp giảm thời gian giao tiếp giữa các ứng dụng với hệ quản trị MySQL, bởi vì thay vì gửi nhiều câu lệnh dài thì ta chỉ cần gọi tới một thủ tục và trong thủ tục này sẽ thực hiện nhiều câu lệnh SQL.

Stored procedure sẽ giúp các ứng dụng nhìn minh bạch hơn, nghĩa là khi ta định nghĩa các thao tác xử lý vào một Stored thì công việc của các ngôn ngữ lập trình khác chỉ quan tâm đến tên thủ tục, các tham số truyền vào chứ không cần biết nó thực hiện như thế nào. Mỗi thủ tục sẽ có các mức độ truy cập, nghĩa là ta có thể cấp quyền sử dụng cho một User nào đó trong hệ quản trị.

Stored procedure cũng có một số nhược điểm. Nếu tạo ra quá nhiều Procedure thì hệ quản trị sẽ sử dụng bộ nhớ để lưu trữ các thủ tục này khá nhiều. Ngoài ra nếu thực hiện quá nhiều xử lý trong mỗi thủ tục thì đồng nghĩa với việc CPU sẽ làm việc nặng hơn. Một số hệ quản trị CSDL có những tool hỗ trợ Debug Store nhưng MySQL thì không có nên gây khó khăn trong quá trình kiểm lỗi.

Cú pháp khai báo Stored procedure:

```
DELIMITER $$
```

```
CREATE PROCEDURE procName([definition paramName typeName], ...)
```

```
BEGIN
```

```
-- xử lý
```

```
END; $$
```

```
DELIMITER ;
```

- DELIMITER \$\$ và DELIMITER; : dùng để phân cách bộ nhớ lưu trữ thủ tục cache và mở ra một ô lưu trữ mới cho stored này. Đây là cú pháp bắt buộc.
- procName: tên của stored procedure.
- [definition paramName typeName]: các khai báo tham số và kiểu dữ liệu. definition biểu hiện bởi IN, OUT hoặc INOUT.
- BEGIN ... END \$\$: nội dung xử lý của Stored procedure nằm trong đoạn lệnh này.

4.3.2. Danh sách các Store Procedure

Bảng 33: Danh sách các Store Procedure

STT	Tên trigger	Bảng	Nội dung
1	sp_findPersonDetail	Persons, employees, supervisors, teachers	Xem thông tin người dùng
2	sp_findClassDetail	Classes, persons, teachers	Xem thông tin lớp học
3	sp_showClassDetailWithYear	Classes, persons, teachers, classHasAcademicYear, academicYear	Xem danh sách các lớp học của năm học
4	sp_findStudentStudyAtClass	Students, persons, studiesAt	Xem danh sách học sinh theo lớp
5	sp_findRole	roles	Xem các quyền hạn của người dùng
6	sp_findStudentDetailByName	Persons, students, studiesat, classes, academicyear	Tìm kiếm thông tin học sinh trong hệ thống bằng tên
7	sp_insertStudent	Persons, students	Thêm thông tin học sinh
8	sp_showRoom	Rooms, roomrentals	Liệt kê những phòng còn trống
9	sp_registerRoom	roomrentals	Thêm một lệnh mượn phòng
10	sp_showRoomWaiting	Roomrentals, rooms, persons	Liệt kê những phòng đang chờ kiểm duyệt
11	sp_approveRoom	roomRentals	Đồng ý cho

			mượn phòng
12	sp_disapproveRoom	roomRentals	Không đồng ý cho mượn phòng
13	sp_showRoomRentaling	Roomrentals, rooms, persons	Liệt kê những phòng đang mượn
14	sp_returnRoom	roomRentals	Trả phòng đã mượn
15	sp_findStudentAtClassAndUpdateConduct	Students, studiesAt, classes, classHasAcademicYear, academicYear	Cập nhật toàn bộ hạnh kiểm của học sinh tại 1 lớp học khi bắt đầu mỗi học kỳ

4.3.3. Mô tả một số Store Procedure

4.3.3.1. Xem thông tin người dùng

- **Tên:** sp_findPersonDetail
- **Nội dung:** Liệt kê thông tin người dùng ứng với số chứng minh nhân dân. Nếu người dùng là học sinh thì hiển thị thông tin học sinh đó, thông tin phụ huynh và số điện thoại phụ huynh. Nếu người dùng là nhân viên trường thì hiển thị thông tin nhân viên đó kèm theo mức lương, ngày bắt đầu làm việc. Nếu nhân viên đó là giáo viên thì hiển thị thêm bộ môn giảng dạy và chức vụ của giáo viên đó.

- **Các bước thực hiện:**

[1] Kiểm tra thông tin người dùng là học sinh hay là nhân viên

[2] Nếu là học sinh thì:

[2.1] In thông tin học sinh và thông tin phụ huynh, số điện thoại liên lạc.

[3] Nếu là nhân viên thì kiểm tra là giám thị hay là giáo viên

[3.1] Nếu là nhân viên thì in thông tin nhân viên đó kèm mức lương, ngày bắt đầu làm việc

[3.2] Nếu là giáo viên thì in thông tin giáo viên đó, kèm mức lương, ngày bắt đầu làm việc, bộ môn giảng dạy và chức vụ.

- **Code thực hiện:**

```

DELIMITER $$
CREATE PROCEDURE findPersonDetail(personID
varchar(50))
BEGIN
    DECLARE typeOfPerson VARCHAR(20);
    SELECT P.personsType INTO typeOfPerson FROM
persons P WHERE P.personID = personID;
    IF (typeOfPerson = 'employee') THEN
    BEGIN
        DECLARE typeOfEmployee VARCHAR(20);
        SELECT E.employeeID INTO typeOfEmployee
FROM employees E WHERE E.employeeID = personID;
        IF (typeOfEmployee = 'supervisor') THEN
        BEGIN
            SELECT *
            FROM persons P
            INNER JOIN employees E ON P.personID =
E.employeeID
            INNER JOIN supervisors S ON
S.supervisorID = E.employeeID
            WHERE P.personID = personID;
        END;
        ELSE BEGIN
            SELECT *
            FROM persons P
            INNER JOIN employees E ON P.personID =
E.employeeID
            INNER JOIN teachers T ON T.teacherID =
E.employeeID
            WHERE P.personID = personID;
        END;
        END IF;
    END;
    ELSE BEGIN
        SELECT *
        FROM persons INNER JOIN students
        ON persons.personID = students.studentID
        WHERE persons.personID = personID;
    END;
    END IF;
END;

```



```
$$  
DELIMITER ;
```

4.3.3.2. Xem thông tin lớp học

- **Tên:** sp_findClassDetail
- **Nội dung:** Liệt kê thông tin của lớp học đó bao gồm thông tin giáo viên chủ nhiệm, thông tin năm học, sĩ số, khối học
- **Các bước thực hiện:**
 - [1] Liệt kê thông tin lớp học đó bao gồm thông tin năm học, sĩ số trình độ, mã giáo viên chủ nhiệm
 - [2] Tìm thông tin của giáo viên và in ra bộ môn giảng dạy, chức vụ
- **Code thực hiện:**

```
DELIMITER $$  
CREATE PROCEDURE findClassDetail(classID  
VARCHAR(50))  
BEGIN  
    SELECT *  
    FROM `classes` C  
    INNER JOIN `persons` P ON P.personID =  
C.mainTeacherID  
    INNER JOIN `teachers` T ON T.teacherID =  
P.personID  
    WHERE C.classID = classID;  
END;  
$$  
DELIMITER ;
```

4.3.3.3. Xem danh sách các lớp học của năm học

- **Tên:** sp_showClassDetailWithYear
- **Nội dung:** Liệt kê danh sách các lớp theo năm học và khối học
- **Các bước thực hiện:**
 - [1] Tìm mã các lớp học của năm học đó và khối học đó
 - [2] Xuất thông tin lớp học ứng với các mã lớp học tìm được bao gồm thông tin năm học, sĩ số, trình độ, mã giáo viên chủ nhiệm
 - [3] Tìm thông tin của giáo viên và in ra bộ môn giảng dạy, chức vụ ứng với mã giáo viên
- **Code thực hiện:**

```
DELIMITER $$
```

```

CREATE PROCEDURE
showClassDetailWithYear(academicYear INT,
levelClass INT)
BEGIN
    SELECT *
    FROM `classes` C
    INNER JOIN `persons` P ON P.personID =
C.mainTeacherID
    INNER JOIN `teachers` T ON T.teacherID =
P.personID
    WHERE C.level = levelClass
    AND C.classID IN (
        SELECT CA.classID
        FROM `classHasAcademicYear` CA
        INNER JOIN `academicYear` AY ON
CA.academicYearID = AY.academicYearID
        WHERE YEAR(AY.academicYear) = academicYear;
    );
END;
$$
DELIMITER ;

```

4.3.3.4. Xem danh sách học sinh theo lớp

- **Tên:** sp_findStudentStudyAtClass
- **Nội dung:** Liệt kê danh sách các thông tin học sinh đang theo học tại lớp học ứng với mã lớp bao gồm họ tên, ngày tháng năm sinh, số điện thoại phụ huynh, họ tên phụ huynh.
- **Các bước thực hiện**
 - [1] Liệt kê danh sách các mã học sinh trong bảng `studyAt` ứng với mã lớp học
 - [2] Tìm và xuất các thông tin học sinh ứng với các mã học sinh đã tìm được
- **Code thực hiện:**

```

DELIMITER $$
CREATE PROCEDURE findStudentStudyAtClass(classID
VARCHAR(50))
BEGIN
    SELECT *
    FROM `persons` P

```

```

        INNER JOIN `students` S ON S.studentID =
P.personID
        WHERE P.personID IN (
            SELECT S.studentID
            FROM `studiesAt` S
            WHERE S.classID = classID
        );
    END;
    $$
    DELIMITER ;

```

4.3.3.5. Xem các quyền hạn của người dùng

- **Tên:** sp_findRole
- **Nội dung:** Liệt kê những quyền hạn mà người dùng đó được truy cập vào
- **Các bước thực hiện:**
 - [1] Tìm mã quyền truy cập ứng với tài khoản người dùng
 - [2] Liệt kê những quyền truy cập mà tài khoản này được sử dụng
- **Code thực hiện:**

```

- DELIMITER $$
CREATE PROCEDURE findRole(personID varchar(50))
BEGIN
    SELECT * FROM roles
    WHERE roles.roleID IN (SELECT U.roleID FROM
users U WHERE U.personID = personID);
END;
$$
DELIMITER ;

```

4.3.3.6. Tìm kiếm thông tin học sinh trong hệ thống bằng tên

- **Tên:** sp_findStudentDetailByName
- **Nội dung:** Liệt kê những thông tin của học sinh bằng tên
- **Các bước thực hiện:**
 - [1] Tìm personID mà những tên này có chứa 1 phần so với tên cần tìm
 - [2] Dùng personID tìm kiếm thông tin trong các bảng dữ liệu khác để lấy thông tin về lớp học, năm học
- **Code thực hiện:**

```

DELIMITER $$
CREATE PROCEDURE findStudentDetailByName(personName
varchar(50), academicYearID INT)

```

```

BEGIN
    SELECT *
    FROM `persons` P
    INNER JOIN `students` S ON P.personID =
S.studentID
    INNER JOIN `studiesat` SA ON S.studentID =
SA.studentID
    INNER JOIN `classes` C ON C.classID =
SA.classID
    INNER JOIN `academicyear` A ON A.academicYearID
= C.academicYearID
    WHERE P.name LIKE CONCAT('%', personName , '%')
    AND A.academicYearID = academicYearID;
END;
$$
DELIMITER ;

```

4.3.3.7. Thêm thông tin học sinh

- **Tên:** sp_insertStudent
- **Nội dung:** Thêm mới một học sinh trong trường
- **Các bước thực hiện:**
 - [1] Kiểm tra mã học sinh (mã CMND) đã tồn tại trong hệ thống chưa
 - [2] Nếu chưa tồn tại thì:
 - [2.1] Nhập các thông tin của học sinh
 - [2.2] Nhập thông tin phụ huynh của học sinh
 - [2.3] Thông báo nhập thông tin mới thành công
 - [3] Nếu đã tồn tại thì thông báo học sinh đã tồn tại trên hệ thống
- **Code thực hiện:**

```

DELIMITER $$
CREATE PROCEDURE insertStudentInfo(personID
VARCHAR(50), gender TINYINT(1), nameA
VARCHAR(150),dateOfBirth DATE, telephone
VARCHAR(11), parentalName VARCHAR(50),
parentalTelephone VARCHAR(13), address1
VARCHAR(100))
BEGIN
    -- declare handler
    declare exit handler for sqlexception
    begin

```

```

rollback;
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'ERROR';
end;

IF NOT EXISTS (SELECT * FROM `persons` P WHERE
P.personID = personID) THEN
BEGIN
INSERT INTO `persons` (`personID`, `name`,
`gender`, `dateOfBirth`, `telephone`, `address`,
`personsType`)
VALUES(personID, nameA, gender, dateOfBirth,
telephone, address1, 'student');
INSERT INTO `students` (`studentID`,
`startDate`, `parentalName`, `parentalTelephone`)
VALUES(personID, CURRENT_TIMESTAMP,
parentalName, parentalTelephone);
END;
END IF;
END;
$$
DELIMITER ;

```

4.3.3.8. Liệt kê những phòng còn trống

- **Tên:** sp_showRoom
- **Nội dung:** Liệt kê những phòng còn trống trong ngày hôm đó
- **Các bước thực hiện:**
 - [1] Liệt kê danh sách tất cả các phòng
 - [2] Kết danh sách trên ứng với thông tin những phòng đang mượn của ngày hôm đó và xuất kết quả
- **Code thực hiện:**

```

DELIMITER $$
CREATE PROCEDURE showRoom(weekTime DATE)
BEGIN
SELECT R.*, RR.roomRentalID
FROM rooms R
LEFT JOIN roomrentals RR ON RR.roomID =
R.roomID
AND DATEDIFF(RR.rentalDate, weekTime) = 0;
END;

```

```
$$  
DELIMITER ;
```

4.3.3.9. Thêm một lệnh mượn phòng

- **Tên:** sp_registerRoom
- **Nội dung:** Thêm một yêu cầu mượn phòng vào danh sách chờ
- **Các bước thực hiện:**
 - [1] Kiểm tra thông tin phòng đang mượn trong danh sách chờ mượn
 - [2] Nếu phòng chưa có ai mượn thì:
 - [2.1] Thêm một yêu cầu mượn phòng với các thông tin người mượn, loại hoạt động tổ chức, ngày mượn
 - [3] Nếu phòng đã có người mượn thì:
 - [3.1] Rollback và xuất thông báo không đăng ký được phòng
- **Code thực hiện:**

```
DELIMITER $$  
CREATE PROCEDURE registerRoom(roomID VARCHAR(50),  
recipientID VARCHAR(50), recipientType VARCHAR(20),  
rentalDate DATE, note VARCHAR(200))  
BEGIN  
    SET AUTOCOMMIT = 0;  
    START TRANSACTION;  
    IF NOT EXISTS(SELECT RR.roomID FROM  
`roomrentals` RR WHERE RR.roomID = roomID AND  
DATEDIFF(RR.rentalDate, rentalDate) = 0)  
    THEN  
        INSERT INTO `roomrentals` (`roomRentalID`,  
`roomID`, `recipientID`, `recipientType`,  
`rentalDate`, `approvalID`, `isReturned`,  
`returnDate`, `note`)  
        VALUES (NULL, roomID, recipientID,  
recipientType, rentalDate, NULL, NULL, NULL, NULL);  
    ELSE  
        ROLLBACK;  
    END IF;  
    COMMIT;  
END;  
$$  
DELIMITER ;
```

4.3.3.10. Liệt kê những phòng đang chờ kiểm duyệt

- **Tên:** sp_showRoomWaiting
- **Nội dung:** Liệt kê những phòng đang chờ kiểm duyệt để giám thị thực hiện nhiệm vụ kiểm duyệt phòng
- **Các bước thực hiện:**
 - [1] Tìm kiếm những roomID trong bảng các phòng đang mượn mà có chưa có người kiểm duyệt
 - [2] Kết thông tin vừa tìm được với thông tin người mượn và thông tin phòng đó.
- **Code thực hiện:**

```
DELIMITER $$
CREATE PROCEDURE showRoomWaiting()
BEGIN
    SET AUTOCOMMIT = 0;
    SELECT *
    FROM `roomrentals` R
    INNER JOIN `rooms` RO ON RO.roomID = R.roomID
    INNER JOIN `persons` P ON P.personID =
    R.recipientID
    WHERE R.approvalID IS NULL;
END;
$$
DELIMITER ;
```

4.3.3.11. Đồng ý cho mượn phòng

- **Tên:** sp_approveRoom
- **Nội dung:** Cập nhập thông tin người đồng ý duyệt cho mượn phòng trong danh sách các phòng đang chờ mượn
- **Các bước thực hiện:**
 - [1] Cập nhật thông tin người đồng ý kiểm duyệt vào danh sách các phòng đang chờ mượn tương ứng với mã đăng ký mượn phòng
- **Code thực hiện:**

```
DELIMITER $$
CREATE PROCEDURE approveRoom(roomRentalID INT(20),
approvalID VARCHAR(50))
BEGIN
    SET AUTOCOMMIT = 0;
    UPDATE `roomRentals` R
```

```

        SET R.approvalID = approvalID
        WHERE R.roomRentalID = roomRentalID;
    END
    $$
    DELIMITER ;

```

4.3.12. Không đồng ý cho mượn phòng

- **Tên:** sp_disapproveRoom
- **Nội dung:** Giám thị từ chối cho mượn phòng học
- **Các bước thực hiện:**
 - [1] Xóa thông tin mượn phòng ứng với mã đăng ký mượn phòng
- **Code thực hiện:**

```

DELIMITER $$
CREATE PROCEDURE disapproveRoom(roomRentalIDA INT(20))
BEGIN
    SET AUTOCOMMIT = 0;
    DELETE FROM `roomRentals`
    WHERE `roomRentalID` = roomRentalIDA;
END
$$
DELIMITER ;

```

4.3.13. Liệt kê những phòng đang mượn

- **Tên:** sp_showRoomRentaling
- **Nội dung:** Liệt kê những phòng đang được mượn và đã có người kiểm duyệt
- **Các bước thực hiện:**
 - [1] Liệt kê những phòng mà đã có người kiểm duyệt nhưng chưa được trả
 - [2] Kết với thông tin người đang mượn và xuất ra
- **Code thực hiện:**

```

DELIMITER $$
CREATE PROCEDURE showRoomRentaling()
BEGIN
    SET AUTOCOMMIT = 0;
    SELECT *
    FROM `roomrentals` R
    INNER JOIN `rooms` RO ON RO.roomID = R.roomID
    INNER JOIN `persons` P ON P.personID =

```



```

R.recipientID
WHERE R.isReturned IS NULL
AND R.approvalID IS NOT NULL;
END;
$$
DELIMITER ;

```

4.3.14. Trả phòng đã mượn

- **Tên:** sp_returnRoom
- **Nội dung:** Cập nhật thông tin ngày mượn và ngày trả ứng với mã mượn phòng tương ứng
- **Các bước thực hiện:**
 - [1] Cập nhật thông tin ngày mượn và ngày trả ứng với mã mượn phòng tương ứng
- **Code thực hiện:**

```

DELIMITER $$
CREATE PROCEDURE returnRoom(roomRentalIDA INT(20),
returnDate DATETIME)
BEGIN
    SET AUTOCOMMIT = 0;
    UPDATE `roomRentals` R
    SET R.isReturned = 1, R.returnDate = returnDate
    WHERE R.roomRentalID = roomRentalIDA;
END
$$
DELIMITER ;

```

4.3.15. Cập nhật hạnh kiểm học sinh của một lớp học

- **Tên:** sp_findStudentAtClassAndUpdateConduct
- **Nội dung:** Cập nhật toàn bộ hạnh kiểm của học sinh tại 1 lớp học khi bắt đầu mỗi học kỳ.
- **Các bước thực hiện:**
 - [1] Khởi tạo con trỏ với các thông tin dùng để xác định lớp học, năm học và hạnh kiểm đó để lấy ra danh sách các ID của học sinh
 - [2] Khởi tạo vòng lặp để lấy danh sách các ID của học sinh
 - [3] Ứng với mỗi ID học sinh thì cập nhật hạnh kiểm lại
 - [4] Chuyển sang ID học sinh tiếp theo
 - [5] Nếu ID học sinh không phải là ID cuối cùng thì quay lại bước 3

[6] Thoát khỏi vòng lặp và hiển thị danh sách hạnh kiểm và thông tin học sinh sau khi cập nhật xong

- **Code thực hiện:**

```
DELIMITER $$
CREATE PROCEDURE
findStudentAtClassAndUpdateConduct(classID
VARCHAR(50), academicYearID INT, conduct
TINYINT(2))
BEGIN

    DECLARE personIDLooking VARCHAR(50);
    DECLARE done INT DEFAULT FALSE;

    DECLARE dataPersonID CURSOR FOR
        SELECT P.personID
        FROM `persons` P
        INNER JOIN `students` S ON P.personID =
S.studentID
        INNER JOIN `studiesAt` SA ON S.studentID =
SA.studentID
        INNER JOIN `classes` C ON C.classID =
SA.classID
        INNER JOIN `classHasAcademicYear` CH ON
CH.classID = C.classID
        INNER JOIN `academicYear` A ON
A.academicYearID = CH.academicYearID
        WHERE SA.classID = classID
        AND A.academicYearID = academicYearID;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET
done=TRUE;
    declare exit handler for sqlexception
    begin
        rollback;
    end;
    -- do sleep(10);
    OPEN dataPersonID;
    dataPersonID : LOOP
        FETCH dataPersonID INTO personIDLooking;
        IF done THEN
            LEAVE dataPersonID;
        END IF;
```

```

        UPDATE `studiesAt`
        SET `studiesAt`.`conduct` = conduct
        WHERE `studiesAt`.`studentID` =
personIDLooking;
    END LOOP;

    CLOSE dataPersonID;
    SELECT *
        FROM `persons` P
        INNER JOIN `students` S ON P.personID =
S.studentID
        INNER JOIN `studiesAt` SA ON S.studentID =
SA.studentID
        INNER JOIN `classes` C ON C.classID =
SA.classID
        INNER JOIN `classHasAcademicYear` CH ON
CH.classID = C.classID
        INNER JOIN `academicYear` A ON
A.academicYearID = CH.academicYearID
        WHERE SA.classID = classID
        AND A.academicYearID = academicYearID;
END
$$
DELIMITER ;

```

Chương 5: Các vấn đề liên quan đến xử lý đồng thời

Ngày nay, hầu như tất cả hệ thống được xây dựng với yêu cầu có khả năng hỗ trợ nhiều người dùng cùng có thể truy cập và thao tác trên các dữ liệu chung nhằm phục vụ cho các công việc của họ, công ty. Tuy nhiên, để một hệ thống có thể vận hành tốt không những đáp ứng nhu cầu trên mà còn đòi hỏi dữ liệu phải chính xác và tin cậy. Vậy để làm cho các dữ liệu đảm bảo lượng truy cập nhiều và vẫn thoả mãn các tiêu chí về cơ chế ACID đòi hỏi người thiết kế phải xây dựng các cơ chế liên quan đến xử lý đồng thời.

5.1. Tổng quan về các mức cô lập trong MySQL và sự khác biệt

Trong tất cả các hệ quản trị cơ sở dữ liệu quan hệ hiện nay, cơ chế ACID⁵ không những đóng một vai trò quan trọng trong việc đảm bảo dữ liệu tin cậy mà còn là tiền đề đối xứng để kiểm tra các giao tác có thể thực hiện một cách đồng thời mà không ảnh hưởng đến các giao tác khác mà vẫn đảm bảo độ tin cậy dữ liệu.

5.1.1. Read uncommitted

Đây là mức cô lập dữ liệu thấp nhất, do không thiết lập shared lock (khóa đọc) nên các giao tác khác không phải chờ khi đọc dữ liệu kể cả khi dữ liệu đang bị khóa bởi một giao tác khác.

Vì vậy, mức cô lập read-uncommitted cho tốc độ xử lý rất nhanh tuy nhiên cũng có rất nhiều khuyết điểm như đọc dữ liệu bẩn (dirty read), không thể đọc lại (unrepeatable reads), mất dữ liệu cập nhật (lost update), dữ liệu bóng ma (phantoms).

Giải quyết được	Không giải quyết được
	Lost updates
	Dirty reads
	Unrepeatable reads
	Phantoms

5.1.2. Read committed

Read-committed là mức cô lập mặc định của một số hệ quản trị cơ sở dữ liệu quan hệ hiện nay như SQL Server. Với cơ chế tạo ra khóa đọc

⁵ Atomicity – tính nguyên tử: một giao tác sẽ có nhiều giao tác khác nhau, giao tác này sẽ được hoàn thành khi tất cả các thao tác khác được hoàn thành hoặc không có một thao tác nào được hoàn thành

Consistency – tính nhất quán: khi một giao tác được thực hiện sẽ tạo ra một trạng thái mới hợp lệ, nếu bị lỗi thì sẽ trở lại trạng thái như cũ

Isolation – tính độc lập: các giao tác có thể thực hiện đồng thời mà không ảnh hưởng đến nhau

Durability – tính bền vững: Dữ liệu sẽ được hệ thống sao lưu trong các trường hợp liên quan đến lỗi, các dữ liệu này đều có thể được khôi phục để đảm bảo sự chuẩn xác

trên đơn vị dữ liệu được đọc và giải phóng ngay sau khi đọc dữ liệu xong, tuy nhiên khoá ghi sẽ được giữ cho đến hết giao tác.

Với mức cô lập này, vấn đề đọc dữ liệu rác sẽ được giải quyết bởi mức cô lập này sẽ không cho phép đọc dữ liệu mà các dữ liệu này đang được thao tác trên một giao tác khác.

Tuy nhiên, ở khác với các hệ quản trị khác, với mức cô lập này câu truy vấn SELECT sẽ cho ra kết quả *snapshot* ở trạng thái dữ liệu cũ. Vì vậy việc truy vấn lệnh SELECT sẽ có thể cho ra nhiều kết quả khác nhau nếu truy vấn nhiều lần trên cùng một giao tác (hiện tượng non-consistent read). Do đó vấn đề dữ liệu bóng ma và không thể đọc lại dữ liệu có thể xảy ra với những tiến trình khác khi thực hiện thao tác ghi dữ liệu.

Giải quyết được	Không giải quyết được
Dirty read	Lost updates
	Unrepeatable reads
	Phantoms

5.1.3. Repeatable read

Mức cô lập mặc định của InnoDB trên hệ quản trị MySQL là repeatable-read. Với mức cô lập này, các kết quả của câu truy vấn SELECT trên cùng một giao tác sẽ cho kết quả giống nhau thể hiện tính nhất quán dữ liệu.

Chính vì vậy, ở mức cô lập này, các giao tác khác khi muốn thực hiện việc cập nhật thay đổi giá trị trên đơn vị dữ liệu này phải chờ cho đến khi giao tác này hoàn tất. Bên cạnh đó, khi một giao tác khác đang chiếm khoá ghi trên đơn vị dữ liệu đó, thì giao tác ở mức cô lập repeatable-read này phải chờ đợi cho đến khi giao tác đó kết thúc. Tuy nhiên, ở mức cô lập này vẫn xảy ra tình trạng dữ liệu bóng ma do vẫn cho phép thêm mới dữ liệu trên đơn vị dữ liệu đó.

Giải quyết được	Không giải quyết được
Lost updates	Phantoms
Dirty read	
Unrepeatable reads	

5.1.4. Serializable

Với mức cô lập cao nhất của InnoDB, các vấn đề liên quan đến dữ liệu đều được giải quyết. Do khi chạy ở mức cô lập dữ liệu này Shared Lock sẽ được tạo trên đơn vị dữ liệu được đọc và giữ shared lock này đến

hết giao tác. Bên cạnh đó, mức cô lập này không cho phép Insert những dòng dữ liệu thỏa mãn điều kiện thiết lập Shared Lock. Ngoài ra Exclusive Lock cũng sẽ được tạo trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác. Tất cả các khoá sẽ bị khoá lại do đó các giao tác khác sẽ phải chờ đợi cho đến khi giao tác với mức cô lập serializable được commit hoặc rollback.

Giải quyết được	Không giải quyết được
Lost updates	
Dirty read	
Unrepeatable reads	
Phantoms	

Tuy nhiên với mức cô lập này, các giao tác phải chờ nếu đơn vị dữ liệu cần đọc đang giữ khoá ghi và làm cản trở nhiều đến việc cập nhật dữ liệu của các giao tác khác.

5.2. Kịch bản và giải pháp các vấn đề liên quan đến xử lý đồng thời

Bảng 34: Các vấn đề liên quan đến xử lý đồng thời

STT	Trường hợp	Vấn đề có thể xảy ra	Loại
1	Người dùng A đang xem danh sách thông tin học sinh của lớp để in ra thì người dùng B đang cập nhật lại thông tin	Giáo viên A đang xem danh sách thông tin học sinh thì giám thị thêm một học sinh mới thông tin, khiến thông tin học sinh khi được in ra khác với danh sách đang hiển thị trên màn hình	Unrepeatable data
2	Giám thị đang cập nhật thông tin học sinh thì học sinh đó cũng đang xem thông tin của mình	Giám thị đang cập nhật thông tin của học sinh A nhưng bị lỗi và phải huỷ thông tin cập nhật đó, trong khi đó học sinh A đó đang xem thông tin cá nhân	Uncommitted data
3	Giáo viên và giám thị cùng đồng thời tìm kiếm cập nhật thông tin của học sinh đó	Giáo viên A và giám thị B cùng truy cập và cập nhật vào một đơn vị dữ liệu (trường hợp này là hạnh kiểm học sinh C) trên 2 thời điểm khác nhau nhưng liên	Lost update, Dirty reads

		kề.	
4	Giám thị A đang làm báo cáo danh sách các học sinh có điểm tổng trên 7.5, và giáo viên B thêm học sinh đó vào lớp học đó	Giám thị A đang làm báo cáo những học sinh có điểm tổng trên 7.5. Tuy nhiên, trong quá trình làm báo cáo thì có một giáo viên B thêm học sinh mới vào lớp học với điểm số trên 7.5 điểm, khiến cho kết quả xuất ra bị khác so với dữ liệu ban đầu nhận được.	Phantom
5	Cả 2 học sinh cùng đăng ký mượn phòng tại cùng một thời điểm.	Học sinh A đăng ký mượn 1 phòng học, tại thời điểm đó học sinh B cũng đăng ký mượn 1 phòng học khác. Tuy nhiên sau khi đăng ký xong chỉ có học sinh A đăng ký thành công, còn học sinh B bị lỗi phải đăng ký lại dù 2 phòng đăng ký mượn khác nhau	Deadlock

5.2.1. Unrepeatable Data

Trường hợp: Người dùng A đang xem danh sách thông tin học sinh của lớp để in ra thì người dùng B đang cập nhật lại thông tin

Mô tả tình huống: Giáo viên A đang xem danh sách thông tin học sinh thì giám thị thêm một học sinh mới thông tin, khiến thông tin học sinh khi được in ra khác với danh sách đang hiển thị trên màn hình

Nhận diện loại vấn đề: Unrepeatable data

Mô tả trường hợp chi tiết:

T1	T2
DELIMITER \$\$	

<pre> CREATE PROCEDURE findStudentStudyAtClass (classID VARCHAR(50)) BEGIN start transaction; -- declare handler declare exit handler for sqlexception begin rollback; SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'ERROR'; end; SELECT * FROM `persons` P INNER JOIN `students` S ON S.studentID = P.personID WHERE P.personID IN (SELECT S.studentID FROM `studiesAt` S WHERE S.classID = classID); do sleep(5); </pre>	<pre> DELIMITER \$\$ CREATE PROCEDURE updateStudentInfo(personID VARCHAR(50),dateOfBirth DATE, telephone VARCHAR(11), parentalName VARCHAR(50), parentalTelephone VARCHAR(13), address1 VARCHAR(100)) BEGIN START TRANSACTION; -- declare handler declare exit handler for sqlexception begin </pre>
---	--

<pre> SELECT * FROM `persons` P INNER JOIN `students` S ON S.studentID = P.personID WHERE P.personID IN (SELECT S.studentID FROM `studiesAt` S WHERE S.classID = classID); </pre>	<pre> -- ERROR rollback; SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'ERROR'; end; UPDATE `persons` SET `persons`.`dateOfBirth` = dateOfBirth, `persons`.`telephone` = telephone, `persons`.`address` = address1 WHERE `persons`.`personID` = personID; UPDATE `students` SET `students`.`parentalTelephone` = parentalTelephone, `students`.`parentalName` = parentalName WHERE `students`.`studentID` = personID; COMMIT; END; \$\$ DELIMITER ; </pre>
---	---

commit; END; \$\$ DELIMITER ;	
--	--

Vấn đề xảy ra: T1 thực hiện thao tác xuất ra danh sách các học sinh đang học tại một lớp học. Tuy nhiên vì một sự cố khiến cho quá trình bị chậm đi khoảng một thời gian. Song song đó, T2 thực hiện thao tác thêm mới thông tin của một học sinh trùng với học sinh đang có trong lớp học đó được hoàn tất. Từ đó khiến cho kết quả khi T1 thực hiện lại thao tác xuất ra danh sách để in thì danh sách này lại khác so với danh sách hiển thị trên màn hình.

Nguyên nhân: Ở hai mức độ cô lập read-uncommitted và read-committed, kết quả trả ra của câu truy vấn SELECT sẽ cho ra kết quả snapshot ở trạng thái dữ liệu chính nó (do khoá đọc được giải phóng ngay sau khi hoàn tất câu lệnh truy vấn nên kết quả câu truy vấn SELECT sẽ không đồng nhất dù trên cùng một giao tác). Bên cạnh đó T1 không có thao tác ghi dữ liệu lên hệ thống nên không được cấp khoá ghi. Nên T2 được phép thêm mới lên cùng đơn vị dữ liệu mà không phải chờ đợi T1 kết thúc giao tác.

Giải pháp: Chuyển mức cô lập T1 thành repeatable read. Khi thiết lập mức cô lập này thì tất cả các câu lệnh SELECT trong cùng 1 thao tác đọc trên những đơn vị dữ liệu giống nhau sẽ đọc snapshot của câu truy vấn SELECT đầu tiên

5.2.2. Uncommitted data

Trường hợp: Giám thị đang cập nhật thông tin học sinh thì học sinh đó cũng đang xem thông tin của mình.

Mô tả tình huống: Giám thị đang cập nhật thông tin của học sinh A nhưng bị lỗi và phải huỷ thông tin cập nhật đó, trong khi đó học sinh A đó đang xem thông tin cá nhân làm xảy ra tình trạng uncommitted data.

Nhận diện loại vấn đề: Uncommitted data

Mô tả trường hợp chi tiết:

T1	T2

<pre> DELIMITER \$\$ CREATE PROCEDURE updateStudentInfo(personID VARCHAR(50), dateOfBirth DATE, telephone VARCHAR(11), parentalName VARCHAR(50), parentalTelephone VARCHAR(13), address1 VARCHAR(100)) BEGIN Start trasaction; -- declare handler declare exit handler for sqlexception begin rollback; SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'ERROR'; end; Declare controlStudents TINYINT(1); SELECT R.controlStudents INTO controlStudents FROM `roles` R WHERE R.roleID IN (SELECT U.roleID FROM `users` U WHERE U.personID = userRequest); IF (controlStudents = 1) THEN BEGIN UPDATE `persons` SET `persons`.`dateOfBirth` = dateOfBirth, `persons`.`telephone` = telephone, `persons`.`address` = address1 </pre>	
---	--

<pre> WHERE `persons`.`personID` = personID; UPDATE `students` SET `students`.`parentalTelephone` = parentalTelephone, `students`.`parentalName` = parentalName WHERE `students`.`studentID` = personID; Do sleep(5); </pre>	<pre> DELIMITER \$\$ CREATE PROCEDURE findPersonDetail(personID varchar(50)) BEGIN DECLARE typeOfPerson VARCHAR(20); START TRANSACTION; SELECT P.personsType INTO typeOfPerson FROM persons P WHERE P.personID = personID; IF (typeOfPerson = 'employee') THEN BEGIN DECLARE typeOfEmployee VARCHAR(20); SELECT E.employeeID INTO typeOfEmployee FROM employees E WHERE E.employeeID = personID; IF (typeOfEmployee = 'supervisor') THEN BEGIN SELECT * FROM persons P INNER JOIN employees E ON P.personID = E.employeeID INNER JOIN supervisors S ON </pre>
---	---

<pre> --Lỗi nên bị rollback Commit; END; ELSE BEGIN Rollback; END; END IF; END; \$\$ </pre>	<pre> S.supervisorID = E.employeeID WHERE P.personID = personID; END; ELSE BEGIN SELECT * FROM persons P INNER JOIN employees E ON P.personID = E.employeeID INNER JOIN teachers T ON T.teacherID = E.employeeID WHERE P.personID = personID; END; END IF; END; ELSE BEGIN SELECT * FROM persons INNER JOIN students ON persons.personID = students.studentID WHERE persons.personID = personID; END; END IF; COMMIT; END; \$\$ DELIMITER ; </pre>
--	--

DELIMITER ;	
-------------	--

Vấn đề xảy ra: Thông tin cá nhân của học sinh đó bị hiển thị sai trên màn hình tại thời điểm đó

Nguyên nhân: Trong quá trình cập nhật thông tin học sinh của T1, vì một số sự cố nên quá trình hoàn tất transaction bị chậm lại. Tại thời điểm đó lại có một giao tác T2 khác truy cập vào xem cùng thông tin học sinh đó. Và giao tác T1 bị lỗi nên phải rollback toàn bộ thay đổi. Từ đó thông tin mà T2 đang nắm giữ sẽ không chính xác.

Giải pháp: Chuyển mức cô lập của T2 thành read committed vì mức cô lập này chỉ cho phép transaction đọc dữ liệu của các transaction đã commit

5.2.3. Lost update

Trường hợp: Giáo viên và giám thị cùng đồng thời tìm kiếm cập nhật thông tin của học sinh đó

Mô tả tình huống: Giáo viên A và giám thị B cùng truy cập và cập nhật vào một đơn vị dữ liệu (trường hợp này là hạnh kiểm học sinh C) trên 2 thời điểm khác nhau nhưng liên kế.

Nhận diện loại vấn đề: Lost update, dirty read.

Mô tả trường hợp chi tiết:

T1	T2
DELIMITER \$\$ CREATE PROCEDURE findStudentAtClassAndUpdateConduct (classID VARCHAR(50), academicYearID INT, conduct TINYINT(2)) BEGIN START TRANSACTION; DECLARE personIDLooking VARCHAR(50); DECLARE done INT DEFAULT FALSE; DECLARE dataPersonID	

<pre> CURSOR FOR SELECT P.personID FROM `persons` P INNER JOIN `students` S ON P.personID = S.studentID INNER JOIN `studiesAt` SA ON S.studentID = SA.studentID INNER JOIN `classes` C ON C.classID = SA.classID INNER JOIN `classHasAcademicYear` CH ON CH.classID = C.classID INNER JOIN `academicYear` A ON A.academicYearID = CH.academicYearID WHERE SA.classID = classID AND A.academicYearID = academicYearID; DECLARE CONTINUE HANDLER FOR NOT FOUND SET done=TRUE; declare exit handler for sqlexception begin rollback; end; -- do sleep(10); OPEN dataPersonID; dataPersonID : LOOP FETCH dataPersonID INTO personIDLooking; IF done THEN LEAVE dataPersonID; END IF; UPDATE `studiesAt` SET `studiesAt`.`conduct` = conduct WHERE `studiesAt`.`studentID` = </pre>	<pre> DELIMITER \$\$ CREATE PROCEDURE findStudentAtClassAndUpdateConduct (classID VARCHAR(50), academicYearID INT, conduct TINYINT(2)) BEGIN START TRANSACTION; DECLARE personIDLooking VARCHAR(50); DECLARE done INT DEFAULT FALSE; DECLARE dataPersonID </pre>
--	---

<pre> personIDLooking; END LOOP; CLOSE dataPersonID; DO SLEEP(10); </pre>	<pre> CURSOR FOR SELECT P.personID FROM `persons` P INNER JOIN `students` S ON P.personID = S.studentID INNER JOIN `studiesAt` SA ON S.studentID = SA.studentID INNER JOIN `classes` C ON C.classID = SA.classID INNER JOIN `classHasAcademicYear` CH ON CH.classID = C.classID INNER JOIN `academicYear` A ON A.academicYearID = CH.academicYearID WHERE SA.classID = classID AND A.academicYearID = academicYearID; DECLARE CONTINUE HANDLER FOR NOT FOUND SET done=TRUE; declare exit handler for sqlexception begin rollback; end; -- do sleep(10); OPEN dataPersonID; dataPersonID : LOOP FETCH dataPersonID INTO personIDLooking; IF done THEN LEAVE dataPersonID; END IF; UPDATE `studiesAt` SET `studiesAt`.`conduct` = conduct WHERE `studiesAt`.`studentID` = </pre>
--	---

<pre> SELECT * FROM `persons` P INNER JOIN `students` S ON P.personID = S.studentID INNER JOIN `studiesAt` SA ON S.studentID = SA.studentID INNER JOIN `classes` C ON C.classID = SA.classID INNER JOIN `classHasAcademicYear` CH ON CH.classID = C.classID INNER JOIN `academicyear` A ON A.academicYearID = CH.academicYearID WHERE SA.classID = classID AND A.academicYearID = academicYearID; COMMIT; END \$\$ DELIMITER ; </pre>	<pre> personIDLooking; END LOOP; CLOSE dataPersonID; SELECT * FROM `persons` P INNER JOIN `students` S ON P.personID = S.studentID INNER JOIN `studiesAt` SA ON S.studentID = SA.studentID INNER JOIN `classes` C ON C.classID = SA.classID INNER JOIN `classHasAcademicYear` CH ON CH.classID = C.classID INNER JOIN `academicyear` A ON A.academicYearID = CH.academicYearID WHERE SA.classID = classID AND A.academicYearID = academicYearID; COMMIT; END \$\$ DELIMITER ; </pre>
---	--

Vấn đề xảy ra: Thông tin hạnh kiểm của giao tác T1 sau khi cập nhật bị mất vì T2 chen đè lên

Nguyên nhân: Trong quá trình cập nhật hạnh kiểm của học sinh của T1 vì bị trì hoãn nên việc commit dữ liệu bị hoãn lại, trong khi đó T2 thực hiện và hoàn tất việc cập nhật khiến cho dữ liệu của T1 cập nhật bị mất do T2 ghi đè lên.

Giải pháp: Chuyển mức cô lập của T2 thành read committed vì mức cô lập này chỉ cho phép transaction đọc dữ liệu của các transaction đã commit

5.2.4. Phantom

Trường hợp: Giám thị A đang làm báo cáo danh sách các học sinh có điểm tổng trên 7.5, và giáo viên B thêm học sinh đó vào lớp học đó

Mô tả tình huống: Giám thị A đang làm báo cáo những học sinh có điểm tổng trên 7.5. Tuy nhiên, trong quá trình làm báo cáo thì có một giáo viên B thêm học sinh mới vào lớp học với điểm số trên 7.5 điểm, khiến cho kết quả xuất ra bị khác so với dữ liệu ban đầu nhận được.

Nhận diện loại vấn đề: Phantom

Mô tả trường hợp chi tiết:

T1	T2
<pre>DELIMITER \$\$ CREATE PROCEDURE `createReport`(`subjectID` VARCHAR(50), `academicYearID` INT(4)) BEGIN SELECT G.studentID, G.subjectID, G.oralScore, G.fifteenMinutesScore, G.periodScore, G.finalScore FROM `grades` G WHERE G.academicYearID = academicYearID AND G.subjectID = subjectID;</pre>	<pre>DELIMITER \$\$ CREATE PROCEDURE `insertGrade`(`gradeID` VARCHAR(50), `studentID` VARCHAR(50), `subjectID` VARCHAR(50), `academicYearID` INT(4), `oralScore` FLOAT(4, 2),`fifteenMinutesScore` FLOAT(4, 2),`periodScore` FLOAT(4, 2), `finalScore` FLOAT(4, 2)) BEGIN INSERT INTO `grades`</pre>

<pre> SELECT SUM(G.studentID) FROM `grades` G WHERE G.academicYearID = academicYearID AND G.subjectID = subjectID AND ((G.oralScore + G.fifteenMinutesScore + G.periodScore*2 + G.finalScore*3)/7 >= 7.5); END\$\$ DELIMITER ; </pre>	<pre> VALUES (gradeID, studentID, subjectID, academicYearID, oralScore, fifteenMinutesScore, periodScore, finalScore); END\$\$ DELIMITER ; </pre>
---	---

Vấn đề xảy ra: Số lượng học sinh có điểm số trên 7.5 khác với danh sách các học sinh có điểm trung bình trên 7.5

Nguyên nhân: T1 thiết lập mức cô lập read committed nên khi tính tổng danh sách học sinh có điểm trung bình trên 7.5 thì sẽ cho ra kết quả snapshot bao gồm cả kết quả của việc thêm điểm số học sinh ở T2.

Giải pháp: Chuyển mức cô lập của T1 thành repeatable read. Khi thiết lập mức cô lập này thì tất cả các câu lệnh SELECT trong cùng 1 thao tác đọc trên những đơn vị dữ liệu giống nhau sẽ đọc snapshot của câu truy vấn SELECT đầu tiên.

5.2.5. Deadlock

Trường hợp: Cả 2 học sinh cùng đăng ký mượn phòng tại cùng một thời điểm. Tuy nhiên các giao tác này được thực hiện ở mức cô lập serializable.

Mô tả tình huống: Học sinh A đăng ký mượn 1 phòng học, tại thời điểm đó học sinh B cũng đăng ký mượn 1 phòng học khác. Tuy nhiên sau khi đăng ký xong chỉ có học sinh A đăng ký thành công, còn học sinh B bị lỗi phải đăng ký lại dù 2 phòng đăng ký mượn khác nhau

Nhận diện loại vấn đề: Deadlock

Mô tả trường hợp chi tiết:

T1	T2
<pre>DELIMITER \$\$ CREATE PROCEDURE registerRoom(roomID VARCHAR(50), recipientID VARCHAR(50), recipientType VARCHAR(20), rentalDate DATE, note VARCHAR(200)) BEGIN SET AUTOCOMMIT = 0; START TRANSACTION; -- declare handler declare exit handler for sqlexception begin -- ERROR rollback; SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'ERROR'; end; IF NOT EXISTS(SELECT RR.roomID FROM `roomrentals` RR WHERE RR.roomID = roomID AND DATEDIFF(RR.rentalDate, rentalDate) = 0) THEN</pre>	<pre>DELIMITER \$\$ CREATE PROCEDURE registerRoom(roomID VARCHAR(50), recipientID VARCHAR(50), recipientType VARCHAR(20), rentalDate DATE, note VARCHAR(200)) BEGIN SET AUTOCOMMIT = 0; START TRANSACTION; -- declare handler declare exit handler for sqlexception begin -- ERROR rollback; SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'ERROR'; end; IF NOT EXISTS(SELECT RR.roomID FROM `roomrentals` RR WHERE RR.roomID = roomID AND DATEDIFF(RR.rentalDate, rentalDate) = 0) THEN INSERT INTO `roomrentals`</pre>

<pre> INSERT INTO `roomrentals` (`roomRentalID`, `roomID`, `recipientID`, `recipientType`, `rentalDate`, `approvalID`, `isReturned`, `returnDate`, `note`) VALUES (NULL, roomID, recipientID, recipientType, rentalDate, NULL, NULL, NULL, NULL); ELSE ROLLBACK; END IF; COMMIT; END; \$\$ DELIMITER ; </pre>	<pre> (`roomRentalID`, `roomID`, `recipientID`, `recipientType`, `rentalDate`, `approvalID`, `isReturned`, `returnDate`, `note`) VALUES (NULL, roomID, recipientID, recipientType, rentalDate, NULL, NULL, NULL, NULL); ELSE ROLLBACK; END IF; COMMIT; END; \$\$ DELIMITER ; </pre>
---	---

Vấn đề xảy ra: Chỉ 1 trong 2 đăng ký thành công

Nguyên nhân: Ở mức Serializable, lệnh SELECT sẽ giữ khóa đọc. Vì vậy, T1 muốn thực hiện lệnh insert phải yêu cầu khóa ghi và phải chờ T2 giải phóng khóa đọc. Khóa đọc mà T2 đang giữ chỉ được giải phóng sau khi T2 commit. Để commit, T2 phải thực hiện lệnh insert, phải xin khóa ghi nhưng không được phê duyệt vì yêu cầu khóa ghi trước đó cả T1.

Giải pháp: Chuyển mức cô lập từ serializable về thành repeatable read.

Chương 6: Thiết kế giao diện

Bảng 35: Danh sách các màn hình và mô tả

STT	Tên màn hình	Chức năng
1	Đăng nhập	Truy cập vào hệ thống
2	Bảng điều khiển	Hiển thị các chức năng của hệ thống
3	Học sinh	Tra cứu thông tin học sinh, quản lý hạnh kiểm, chuyên cần
4	Thông tin học sinh	Tra cứu thông tin học sinh
5	Mượn phòng	Thực hiện nghiệp vụ mượn phòng
6	Kiểm duyệt phòng	Xác nhận phòng có được mượn hay không
7	Trả phòng	Sau khi phòng được sử dụng xong sẽ trả phòng
8	Các khối	Hiển thị các khối trong trường
9	Các lớp học theo khối	Hiển thị các lớp học theo khối
10	Lớp học	Hiển thị danh sách 1 lớp học cụ thể
11	Hoạt động	Tạo và tra cứu hoạt động
12	Quản trị người dùng	Chỉnh sửa quyền hạn của người dùng
13	Thông tin người dùng	Tra cứu thông tin người dùng
14	Báo cáo/thống kê điểm TBM theo lớp	Hiển thị báo cáo/thống kê điểm TBM của học sinh trong 1 lớp học cụ thể

6.1. Màn hình “Đăng nhập”

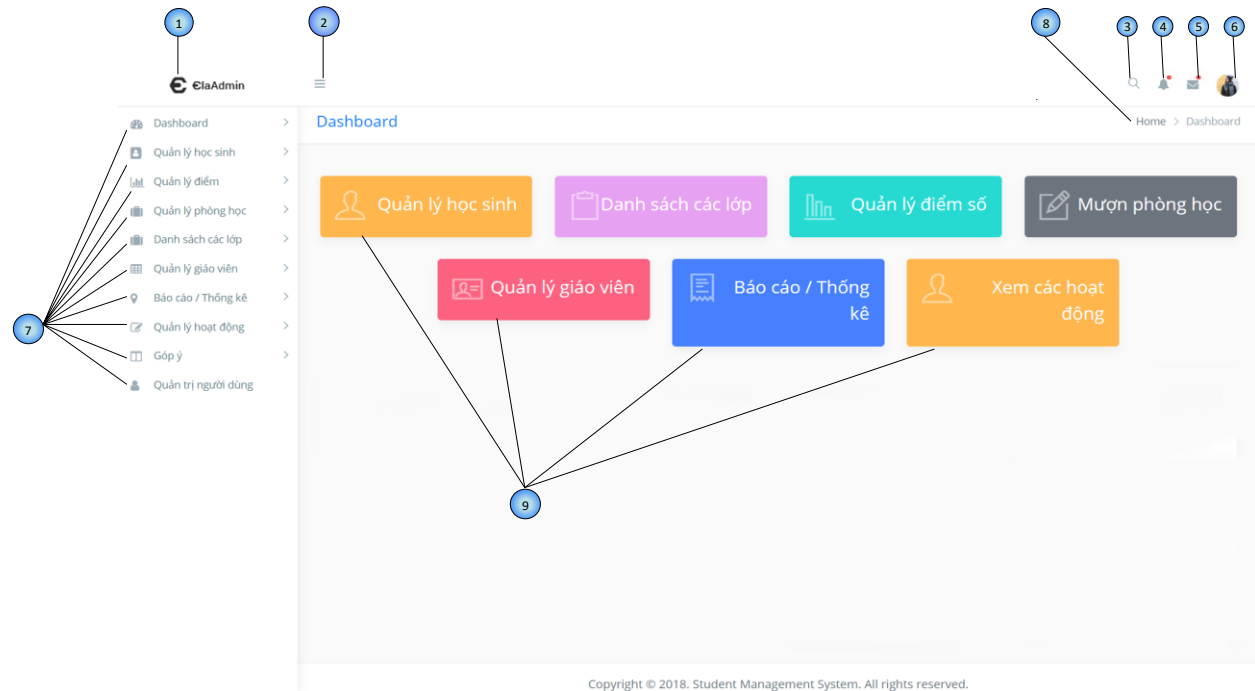
The diagram shows a login form with the following elements and their corresponding callout numbers:

- 1**: Identify Number (Text input field)
- 2**: Password (Text input field)
- 3**: Remember Me (Checkbox)
- 4**: Forgotten Password? (Link)
- 5**: SIGN IN (Button)

Hình 13: Màn hình Đăng nhập

STT	Tên	Kiểu	Chức năng
1	Identify number	Text	Nhập chứng minh nhân dân
2	Password	Text	Nhập mật khẩu
3	Remember me	CheckBox	Lưu tên đăng nhập
4	Forgotten password	Button	Chuyển sang trang quên mật khẩu
5	Sign in	Button	Đăng nhập vào hệ thống

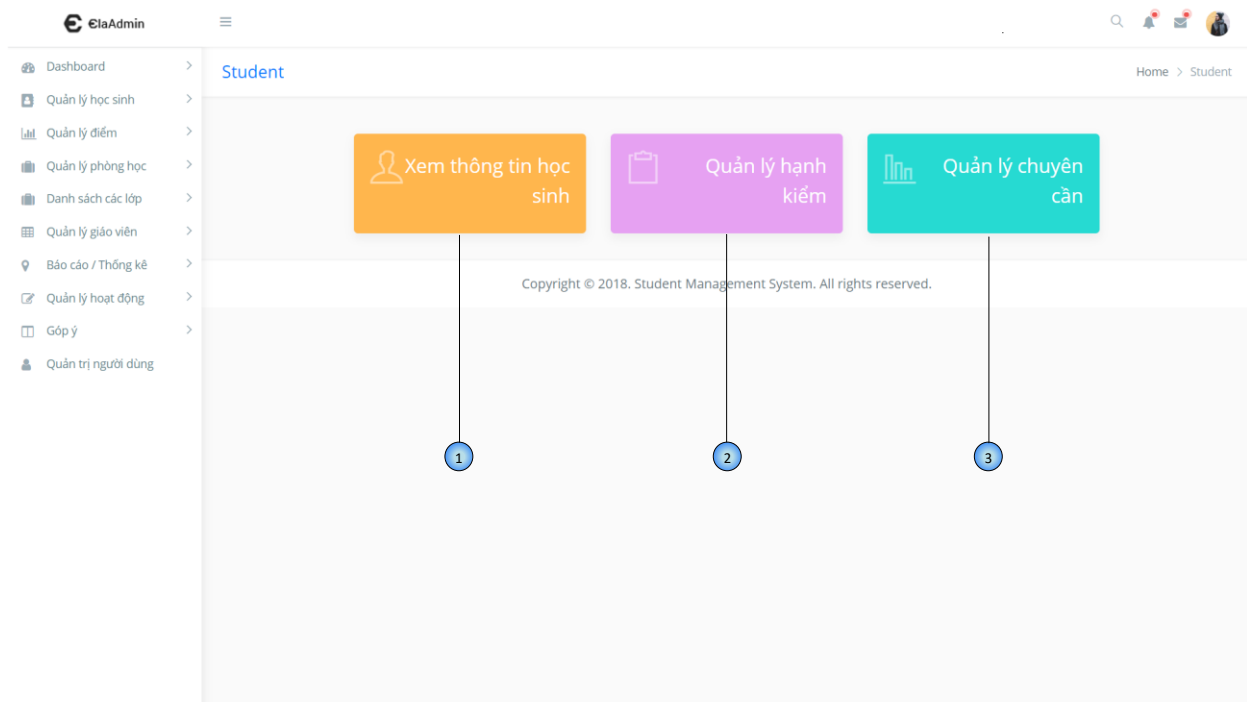
6.2. Màn hình “Bảng điều khiển”



Hình 14: Màn hình Bảng điều khiển

STT	Tên	Kiểu	Chức năng
1	ElaAdmin	Image	Logo phần mềm
2	Nút thu nhỏ	Button	Thu nhỏ thanh menu
3	Tìm kiếm	Buton	Tìm kiếm thông tin
4	Thông báo	Button	Hệ thống thông báo khi có gì thay đổi
5	Tin nhắn	Button	Nhận tin nhắn từ người dùng khác
6	Người dùng	Button	Xem thông tin người dùng
7	Menu hệ thống	Button	Hiển thị danh sách chức năng hệ thống
8	Thanh điều hướng	Label	Thanh điều hướng
9	Các chức năng của hệ thống	Button	Hiển thị chức năng muốn sử dụng

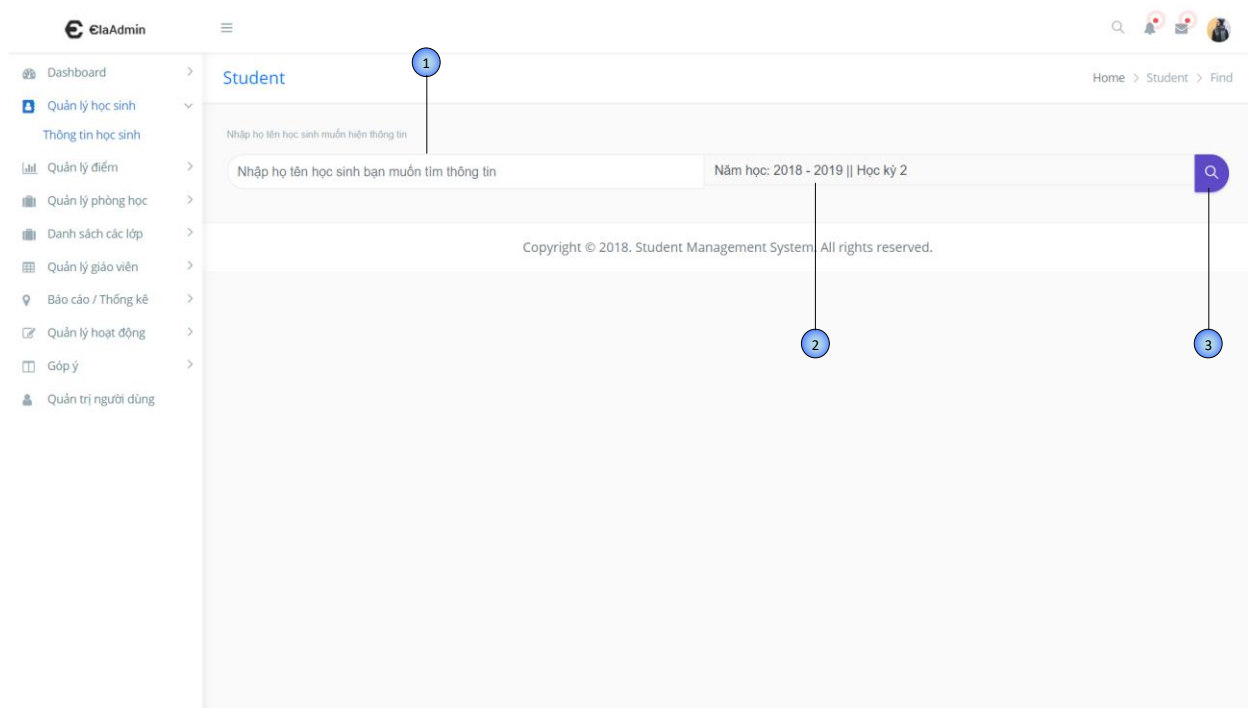
6.3. Màn hình “Học sinh”



Hình 15: Màn hình Học sinh

STT	Tên	Kiểu	Chức năng
1	Xem thông tin học sinh	Button	Chuyển hướng sang xem thông tin học sinh
2	Quản lý hạnh kiểm	Button	Chuyển hướng sang quản lý hạnh kiểm học sinh
3	Quản lý chuyên cần	Button	Chuyển hướng sang quản lý chuyên cần học sinh

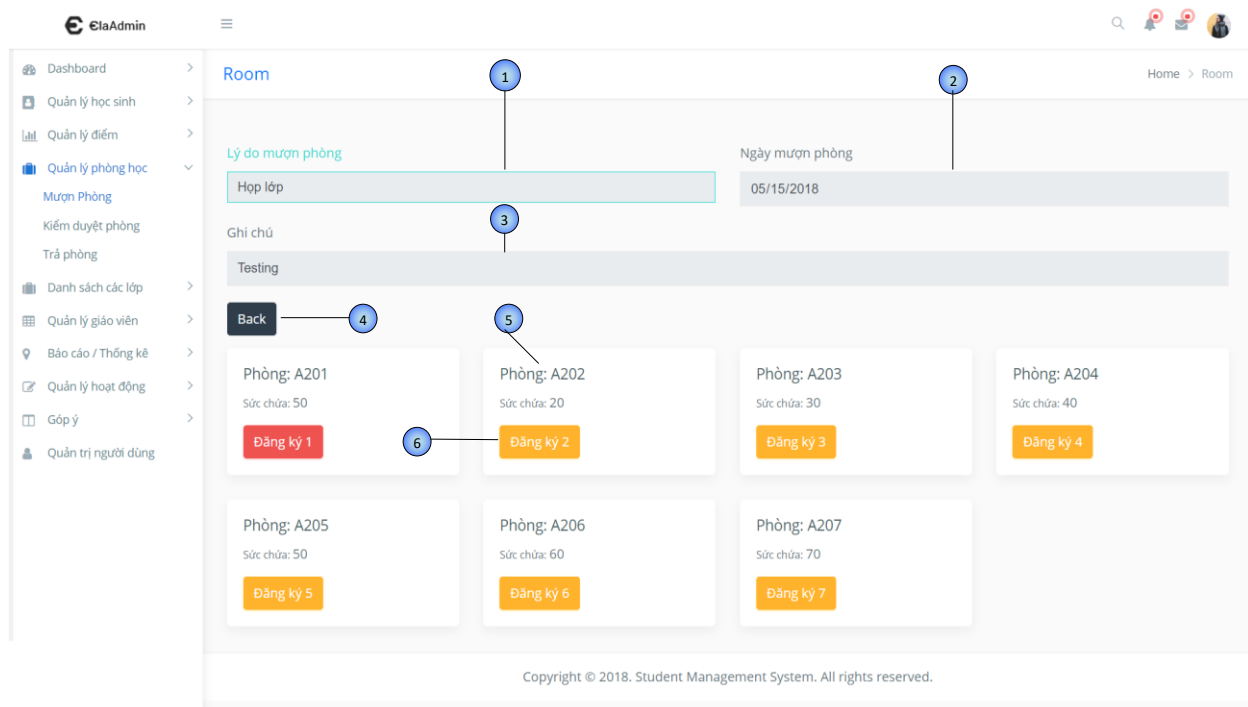
6.4. Màn hình “Thông tin học sinh”



Hình 16: Màn hình Thông tin học sinh

STT	Tên	Kiểu	Chức năng
1	Họ tên học sinh	Button	Nhập họ tên học sinh muốn tra cứu
2	Niên khóa - học kì	Dropdown	Hiển thị danh sách niên khóa – học kì
3	Tra cứu	Button	Tra cứu thông tin học sinh

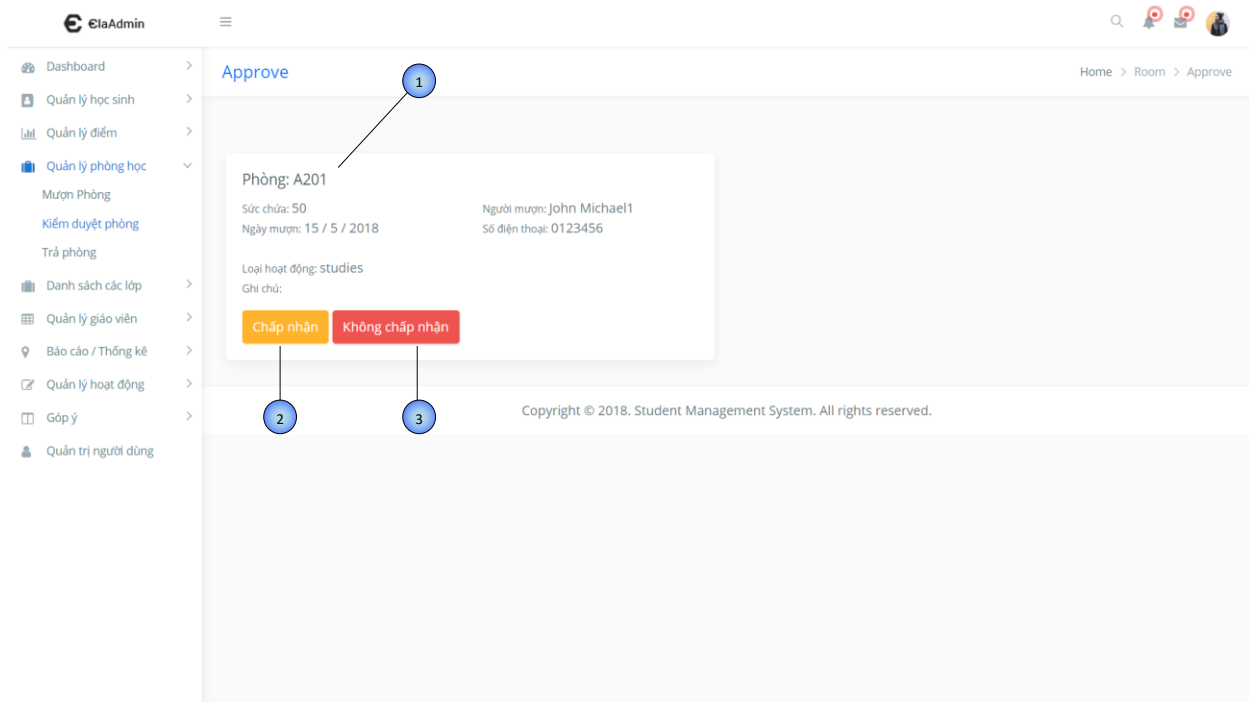
6.5. Màn hình “Mượn phòng”



Hình 17: Màn hình Mượn phòng

STT	Tên	Kiểu	Chức năng
1	Lý do mượn phòng	Drop down	Phân loại lý do mượn phòng
2	Ngày mượn phòng	DateTime	Hiển thị ngày muốn mượn phòng
3	Ghi chú	Text	Nhập ghi chú
4	Back	Button	Quay trở lại quản lý phòng học
5	Phòng mượn	Label	Hiển thị thông tin phòng
6	Đăng kí	Button	Chọn đăng kí phòng muốn mượn

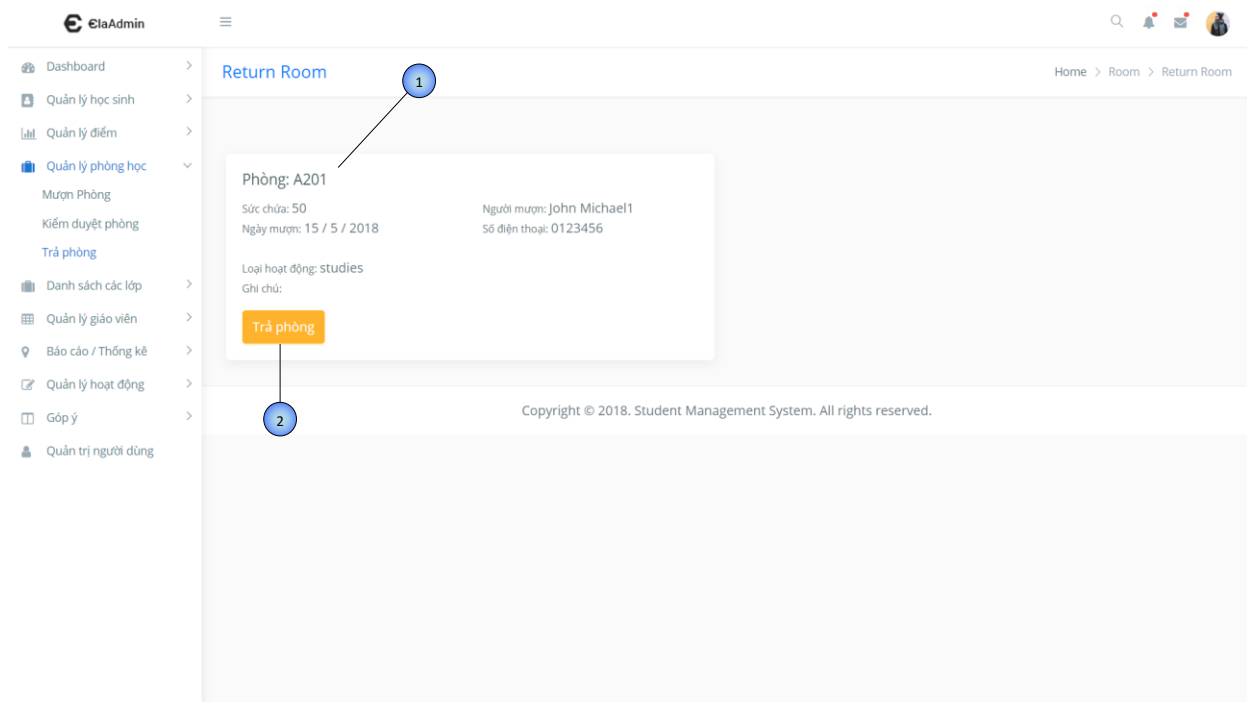
6.6. Màn hình “Kiểm duyệt phòng”



Hình 18: Màn hình Kiểm duyệt phòng

STT	Tên	Kiểu	Chức năng
1	Phòng mượn	Label	Hiển thị thông tin phòng mượn
2	Chấp nhận	Button	Cho phép mượn phòng
3	Không chấp nhận	Button	Không cho phép mượn phòng

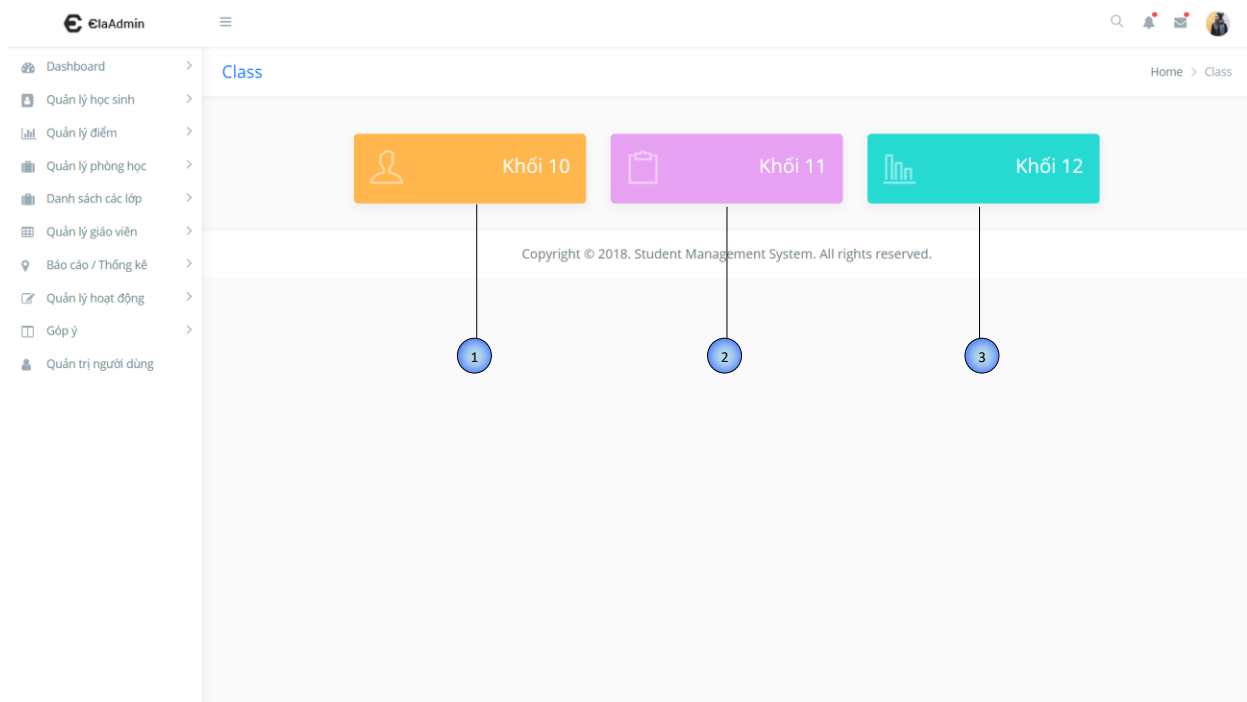
6.7. Màn hình “Trả phòng”



Hình 19: Màn hình Trả phòng

STT	Tên	Kiểu	Chức năng
1	Phòng mượn	Label	Hiển thị thông tin phòng mượn
2	Trả phòng	Button	Trả phòng sau khi sử dụng phòng xong

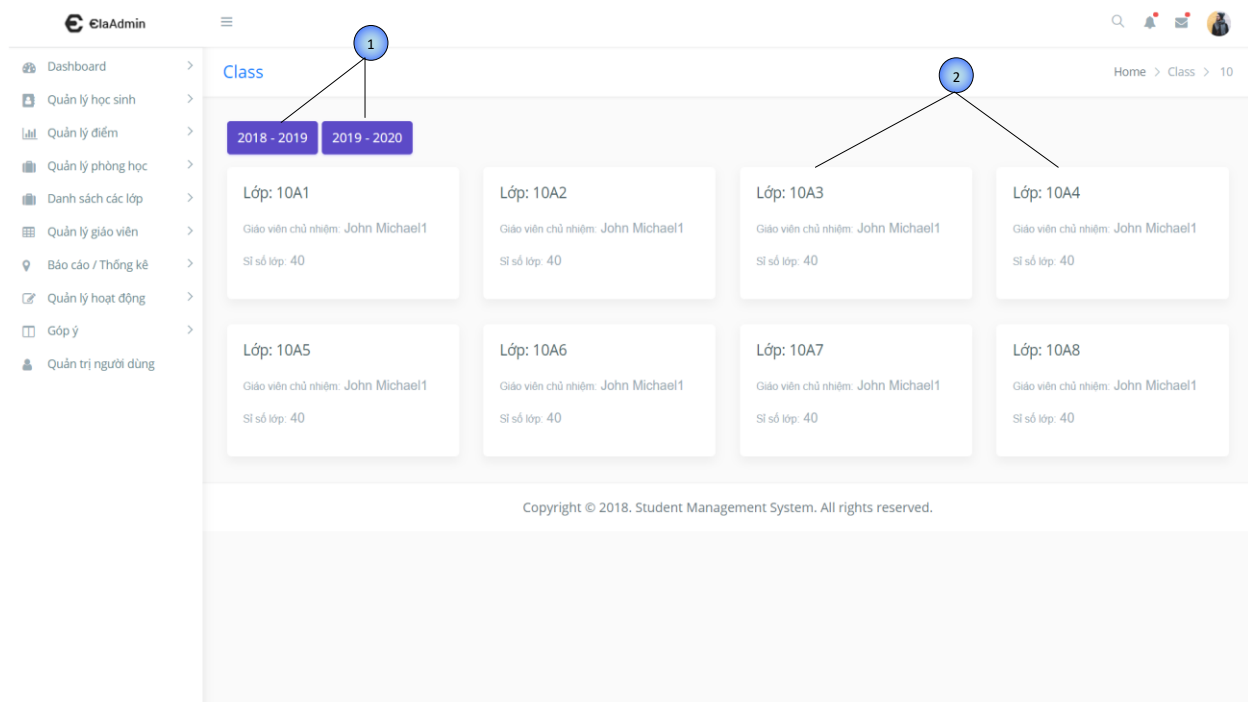
6.8. Màn hình “Các khối”



Hình 20: Màn hình Các khối

STT	Tên	Kiểu	Chức năng
1	Khối 10	Button	Chuyển hướng sang thông tin các lớp học trong khối 10
2	Khối 11	Button	Chuyển hướng sang thông tin các lớp học trong khối 11
3	Khối 12	Button	Chuyển hướng sang thông tin các lớp học trong khối 12

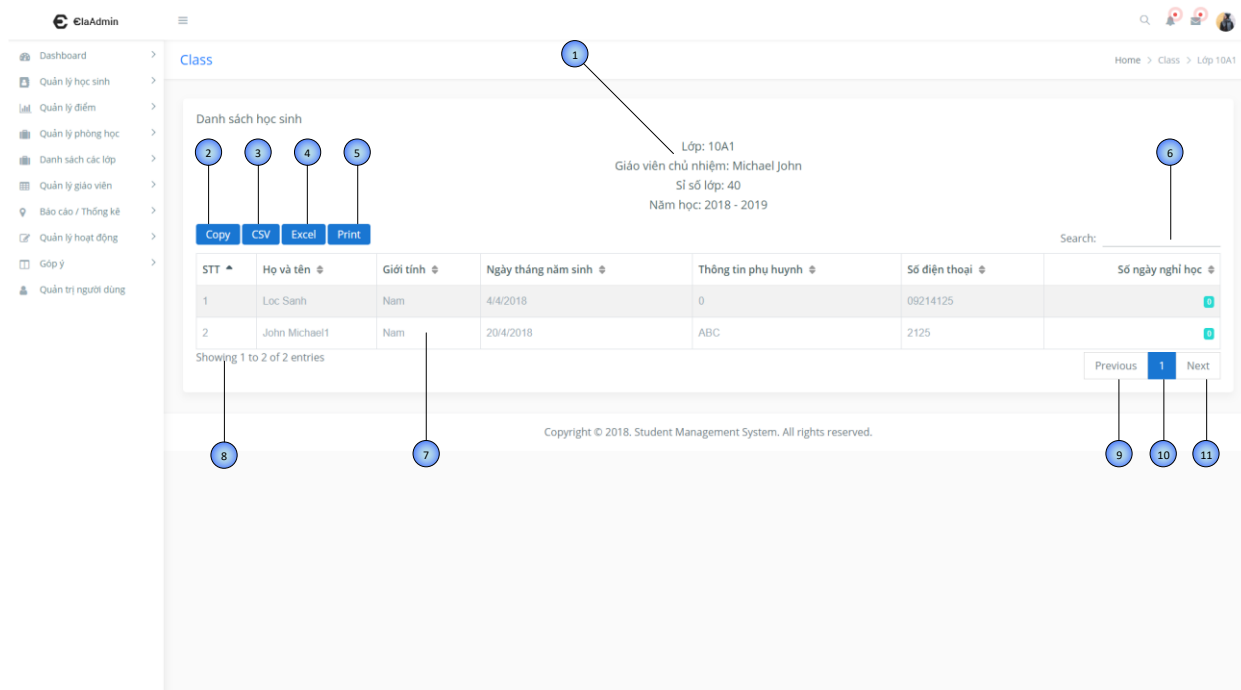
6.9. Màn hình “Các lớp học theo khối”



Hình 21: Màn hình Các lớp học theo khối

STT	Tên	Kiểu	Chức năng
1	Niên khóa	Button	Xem thông tin các lớp theo niên khóa
2	Các lớp học	Label	Thông tin các lớp học trong khóa đó

6.10. Màn hình “Lớp học”



Hình 22: Màn hình Lớp học

STT	Tên	Kiểu	Chức năng
1	Lớp	Label	Hiển thị thông tin lớp học
2	Copy	Button	Sao chép danh sách lớp học
3	CSV	Button	Xuất danh sách lớp sang định dạng csv
4	Excel	Button	Xuất danh sách lớp ra file excel
5	Print	Button	In danh sách lớp
6	Search	Text	Tra cứu thông tin học sinh tồn tại trong danh sách sinh viên
7	Danh sách sinh viên	TableViewer	Hiển thị danh sách sinh viên trong lớp
8	Tổng số lượng học sinh trong lớp	Label	Hiển thị tổng số lượng học sinh trong lớp
9	Previous	Button	Trở về trang trước
10	Page	Label	Hiển thị trang hiện tại
11	Next	Button	Chuyển sang trang kế tiếp

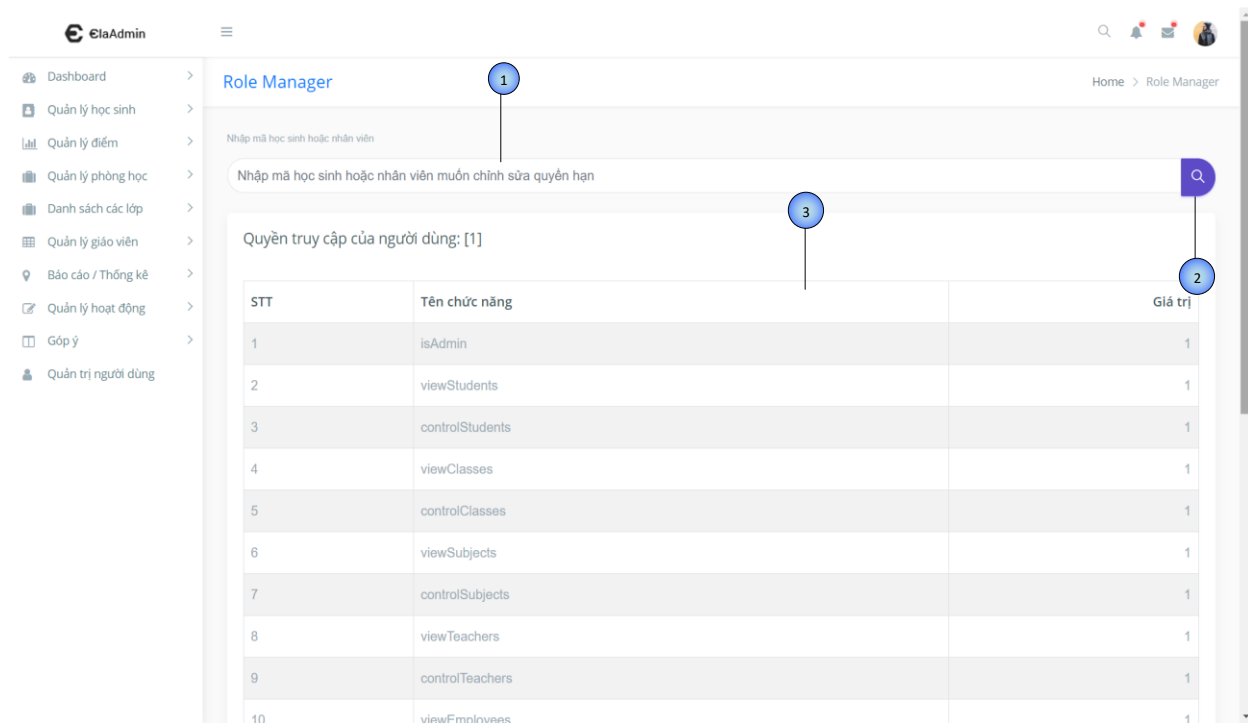
6.11. Màn hình “Hoạt động”



Hình 23: Màn hình Hoạt động

STT	Tên	Kiểu	Chức năng
1	Tạo hoạt động	Button	Chuyển hướng sang trang tạo hoạt động cho học sinh tham gia
2	Xem hoạt động	Button	Chuyển hướng sang xem thông tin hoạt động đã tạo

6.12. Màn hình “Quản trị người dùng”



Hình 24: Màn hình Quản trị người dùng

STT	Tên	Kiểu	Chức năng
1	Nhập mã học sinh hoặc nhân viên	Text	Nhập mã học sinh hoặc nhân viên muốn chỉnh sửa quyền hạn
2	Tra cứu	Button	Tra cứu thông tin mã học sinh hoặc nhân viên muốn chỉnh sửa quyền hạn
3	Danh sách quyền hạn	TableViewer	Hiển thị danh sách quyền hạn của 1 người sau khi được tìm kiếm

6.13. Màn hình “Thông tin người dùng”

Hình 25: Màn hình Thông tin người dùng

STT	Tên	Kiểu	Chức năng
1	Person info	Label	
2	Họ và tên	Text	Hiển thị họ và tên
3	Giới tính	Text	Nhập giới tính
4	Ngày tháng năm sinh	Text	Hiển thị ngày tháng năm sinh
5	Địa chỉ	Text	Hiển thị địa chỉ
6	Ngày bắt đầu làm việc	Text	Hiển thị ngày bắt đầu làm việc
7	Số điện thoại cá nhân	Text	Nhập số điện thoại cá nhân
8	Thông tin nhân viên	Label	
9	Loại nhân viên	Text	Nhập loại nhân viên
10	Lương	Text	Hiển thị lương
11	Lưu thông tin	Button	Lưu thông tin người dùng
12	Cancel	Button	Hủy không lưu những thay đổi
13	Thanh cuộn		

6.14. Màn hình “Báo cáo/thống kê điểm TBM theo lớp”



Hình 26: Màn hình Báo cáo/thống kê điểm TBM theo lớp

STT	Tên	Kiểu	Chức năng
1	Biểu đồ tỉ lệ điểm	eChart	Hiển thị biểu đồ tỉ lệ điểm của học sinh lớp 10A1
2	Biểu đồ phân loại học sinh	eChart	Hiển thị biểu đồ phân loại học sinh lớp 10A1
3	Các thông tin khác	Lable	Hiển thị các thông tin liên quan

Chương 7: Kết luận

7.1. Kết quả đạt được

Trong học kỳ vừa qua, nhóm đã tìm hiểu vận dụng kiến thức về hệ quản trị cơ sở dữ liệu và đã đạt được một số kết quả như sau:

- Nắm vững kiến thức và có thể vận dụng, cài đặt stored procedure, trigger, transaction trong nhiều hệ quản trị cơ sở dữ liệu như MySQL.
- Trang bị thêm kiến thức về các chế độ khóa và mức cô lập trong My SQL.
- Giải quyết được các trường hợp xử lý đồng thời trên hệ quản trị trong môi trường giả lập.
- Nắm được kinh nghiệm hợp tác làm nhóm, báo cáo.
- Xây dựng được phần mềm “Quản lý học sinh trường THPT” với các chức năng sau:

STT	Tên chức năng
1	Quản lý thông tin học sinh
2	Quản lý nhập điểm
3	Quản lý xem điểm
4	Quản lý điểm cá nhân
5	Quản lý mượn phòng học
6	Quản lý kiểm duyệt phòng
7	Quản lý danh sách các lớp khối 10.
8	Quản lý danh sách các lớp khối 11.
9	Quản lý danh sách các lớp khối 12.
10	Quản lý giáo viên bộ môn
11	Quản lý các vi phạm của giáo viên
12	Quản lý thống kê chung
13	Quản lý báo cáo điểm theo môn
14	Quản lý báo cáo điểm lớp
15	Quản lý báo cáo điểm từng học sinh

16	Quản lý xem các hoạt động
17	Quản lý tạo các hoạt động
18	Quản lý góp ý kiến cho trường
19	Quản lý báo cáo các sự cố kỹ thuật
20	Quản trị người dùng

7.2. Hạn chế

Do thời gian ngắn cộng với khối lượng công việc lớn nên trong quá trình thực hiện đồ án nhóm còn gặp phải một số hạn chế :

- Các kịch bản xử lý đồng thời chưa thực tế.
- Thông tin học sinh trong phần mềm phải nhập thủ công.
- Mã mỗi học sinh chưa được mã hóa mã vạch.
- Nhóm không có nhiều thời gian trực tiếp gặp mặt để khảo luận các vấn đề.

7.3. Hướng phát triển

Để khắc phục các hạn chế còn tồn tại cũng như phát triển phần mềm nhóm có một số đề xuất như sau:

- Hỗ trợ import dữ liệu phim từ Excel.
- Nghiên cứu hướng phát triển quét mã thẻ học sinh thông qua tích hợp mã vạch để điểm danh.
- Phần mềm hỗ trợ đa ngôn ngữ.

7.4. Bảng phân công công việc

Bảng 36: Bảng phân công công việc

STT	Nội dung	Nguyễn Minh Đức	Lộc Sinh Vỹ	Nguyễn Trần Ngọc Anh
1	Tổng quan đề tài	Tìm hiểu và định hướng dự án		
2	Phát biểu bài toán	Liên hệ khảo sát các bên liên quan		Tổng hợp các yêu cầu và phát biểu bài toán

3	Phân tích yêu cầu		Dựa trên các kết quả, phân tích yêu cầu	
4	Biểu mẫu, báo cáo	Tổng hợp các biểu mẫu		
5	Thủ tục, quy trình		Liệt kê các thủ tục, quy trình	
6	Các công nghệ sử dụng	Lên danh sách các công nghệ sử dụng		Tìm hiểu và phân tích các công nghệ
7	Hệ quản trị CSDL MySQL		Giới thiệu về MySQL	
8	Xây dựng và thiết kế CSDL	Xây dựng sơ đồ logic. Viết mô hình quan hệ	Viết ràng buộc toàn vẹn	Mô tả dữ liệu
9	Trigger	Giới thiệu về trigger	Lập danh sách và viết các trigger	
10	Store procedure	Mô tả các store procedure	Giới thiệu store procedure	Lập danh sách các store procedure
11	Tổng quan về các mức cô lập trong MySQL	Tìm hiểu về các mức cô lập		
12	Kịch bản và giải pháp các vấn đề liên quan đến xử lý đồng thời	Viết, mô tả các kịch bản xử lý đồng thời		Tìm các trường hợp liên quan đến xử lý đồng thời
13	Thiết kế giao diện	Thiết kế các giao diện	Đặc tả cách thức giao tiếp giữa người dùng với phần mềm	Mô tả giao diện

Chương 8: Phụ lục

8.1. Biểu mẫu khác

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

ĐƠN XIN RÚT HỒ SƠ

Kính gửi: BGH trường THPT Trần Hưng Đạo,

Tôi tên:

Ngày, tháng, năm sinh: / /

Nơi sinh:

Cư trú tại: Ấp, xã huyện tỉnh.....

Là của emsinh năm / /

Học sinh lớp Năm học 201...- 201..... của trường THPT Trần Hưng Đạo,

Nay tôi làm đơn nay kính đến BGH trường THPT Trần Hưng Đạo cho tôi rút hồ sơ của em

Lý do:

.....

Hồ sơ gồm:

1./

4./.....

2./.....

5./.....

3./.....

6./.....

Rất mong được sự pháp thuận của BGH trường, tôi chân thành cảm ơn!

Tam Nông, ngàytháng năm 201...

Văn phòng giao hồ sơ

(Ký, ghi rõ họ tên)

Người viết đơn

(Ký, ghi rõ họ tên)

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

ĐƠN XIN CHUYỂN TRƯỜNG

(Mẫu dành cho học sinh cấp THPT)

Kính gửi:

- Lãnh đạo Sở GD-ĐT:
- Lãnh đạo Sở GD-ĐT:
- Hiệu trưởng trường:
- Hiệu trưởng trường:

Tôi tên là:Hiện ngụ tại:.....

.....
Là phụ huynh của em Sinh ngày tháng năm

Đã trúng tuyển vào lớp 10 năm học 20 - 20 nay là học sinh lớp

.....năm học 20 - 20 thuộc trường

Ngoại ngữ

Kết quả cuối năm : Học lực : Hạnh kiểm :

Tôi làm đơn này gửi đến Hiệu trưởng trường.....

Lãnh đạo Sở GD&ĐTcho tôi được chuyểntôi về học lớp

.....năm học 20 - 20 tại trường

thuộc huyện.....

Lý do :

.....

Được sự chấp thuận của Quý thầy (cô) chúng tôi chân thành cảm ơn./.

Hồ sơ đính kèm

Ngày tháng..... năm 20...

- Học bạ
- Khai sinh
- Bằng TNTHCS
- Giấy CNTT vào lớp 10

Người làm đơn

Ý kiến tiếp nhận Hiệu trưởng trường
(Nơi chuyển đến)

Ý kiến tiếp nhận Hiệu trưởng trường
(Nơi đang học)

SỞ GDĐT TP HỒ CHÍ MINH CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
TRƯỜNG THPT TRẦN HƯNG ĐẠO Độc lập- Tự do- Hạnh phúc

Tam Nông, ngày tháng năm 20

GIẤY XÁC NHẬN

HIỆU TRƯỞNG TRƯỜNG THPT TRẦN HƯNG ĐẠO

Xác nhận học sinh: Giới tính:.....
Ngày sinh:/...../....., Nơi sinh:.....
Hộ khẩu thường trú: Ấp: xã
Huyện:....., tỉnh:.....
Năm học: 201.... – 201.... đang học lớp trường THPT Trần Hưng Đạo.
Nay em làm đơn này đến BGH trường xác nhận cho em đang học tại trường.
Lý do:
Rất mong được sự chấp thuận em xin chân thành cảm ơn!

HIỆU TRƯỞNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

ĐƠN XIN MƯỢN HỒ SƠ HỌC BẠ

Kính gửi: BGH trường THPT Trần Hưng Đạo

Tôi tên là:Hiện ngụ tại:.....

.....
Là phụ huynh của emSinh ngày tháng năm

Đang học lớp.....năm học 20 - 20 thuộc trường THPT Trần Hưng Đạo.

Nay tôi làm đơn này đến BGH trường cho tôi được mượn hồ sơ.

Lý do :

Rất mong được sự chấp thuận của Quý thầy (cô) tôi chân thành cảm ơn./.

Ngày tháng..... năm 20...

Người làm đơn

SỞ GDĐT TP HỒ CHÍ MINH CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
TRƯỜNG THPT TRẦN HƯNG ĐẠO Độc lập- Tự do- Hạnh phúc

ĐƠN XIN PHÉP

Kính gửi: - BGH trường THPT Trần Hưng Đạo,

- Tổ trưởng tổ

Tôi tên:..... GVBM :.....

Hiện đang công tác tại trường THPT Trần Hưng Đạo

Nay tôi làm đơn này kính đến BGH trường cho tôi được nghỉ phép từ ngày
..... đến ngày...../...../20

Lý do:

.....

Trong thời gian được nghỉ tôi sẽ nhờ giáo viên
trong Tổ dạy thay hoặc tôi sẽ dạy bù vào thời gian gần nhất.

Rất mong được sự chấp thuận của Tổ trưởng bộ môn và BGH. Tôi xin chân thành
cảm ơn!.

Tràm Chim, ngày tháng năm 20

Ý KIẾN CỦA TỔ

NGƯỜI VIẾT ĐƠN

DUYỆT CỦA BGH TRƯỞNG

SỞ GD&ĐT TP HỒ CHÍ MINH
TRƯỜNG: THPT TRẦN HƯNG ĐẠO

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Số:...../KH-....

....., ngày tháng năm.....

KẾ HOẠCH THỰC HIỆN NHIỆM VỤ NĂM HỌC 20.... - 20.....

Căn cứ công văn số /SGDDĐT-GDTrH, ngày .../.../2018 về việc hướng dẫn thực hiện nhiệm vụ giáo dục trung học năm học 2018 - 2019 của Sở GD&ĐT (và của Phòng GD&ĐT – đối với trường THCS); Căn cứ vào điều kiện, tình hình cụ thể của đơn vị, trường xây dựng kế hoạch thực hiện nhiệm vụ năm học 2018- 2019 như sau:

I. Đặc điểm tình hình:

1. **Bối cảnh năm học** (của tổ (trọng tâm) và của trường):
2. **Thuận lợi** (mạnh/thời cơ):
3. **Khó khăn** (yếu/thách thức):
4. **Thông kê số liệu:** (Cơ sở vật chất, trang thiết bị, phòng học, phòng làm việc, GV, HS,...)

II. Các nhiệm vụ trọng tâm:

III. Các chỉ tiêu phấn đấu:

IV. Một số giải pháp cơ bản:

Nơi nhận:

- Sở (Phòng) GD&ĐT;
- UBND huyện, TP (xã, phường, thị trấn);
- Lưu.

HIỆU TRƯỞNG

(Ký tên và đóng dấu)