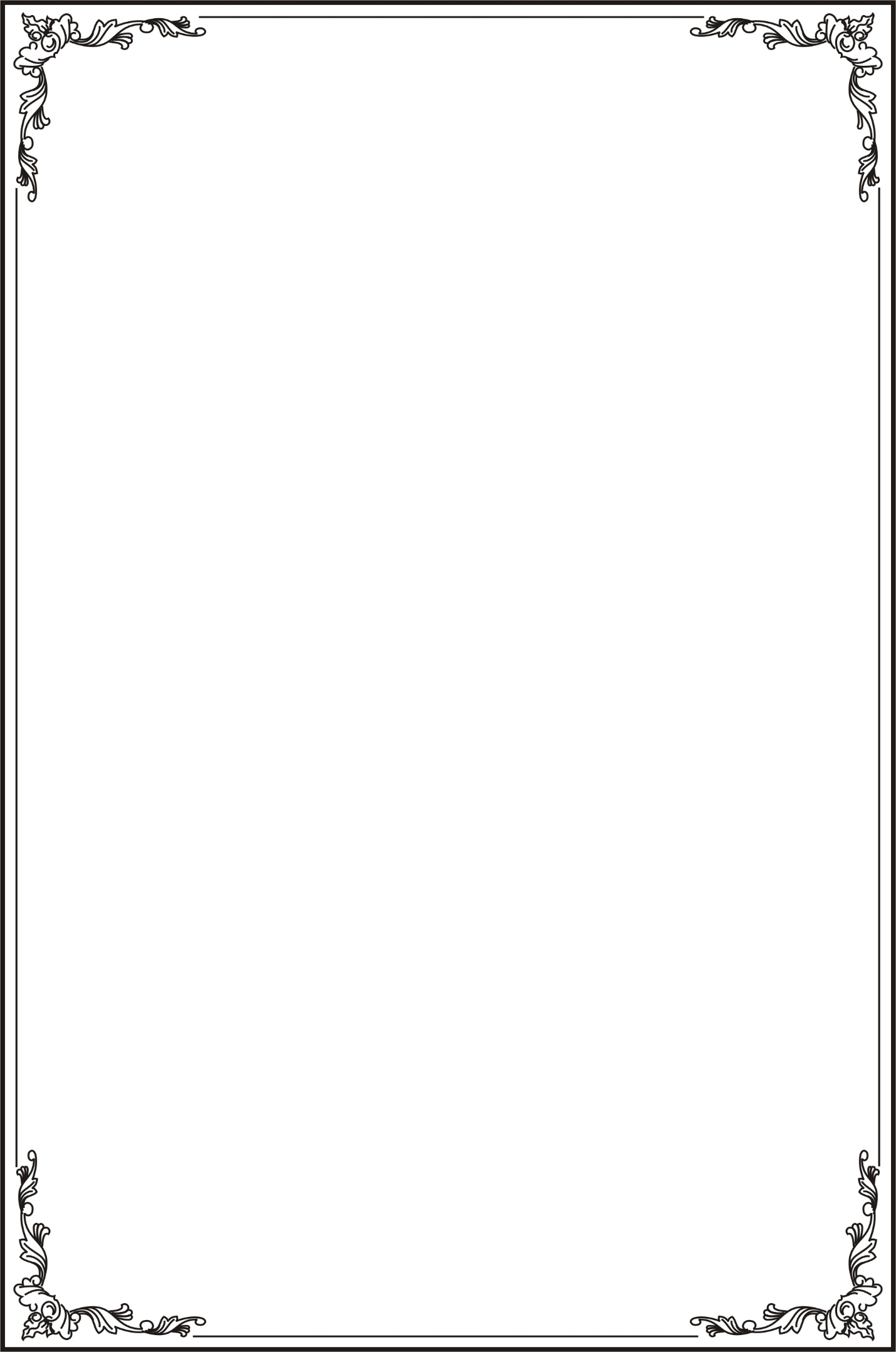
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

---------------🙦 🕮 🙤---------------



*THIẾT KẾ PHẦN MỀM HƯỚNG ĐỐI TƯỢNG*

*XÂY DỰNG ỨNG DỤNG QUẢN LÝ SINH VIÊN*

**GIẢNG VIÊN HƯỚNG DẪN:** Th.S NGUYỄN TRẦN THI VĂN

**SINh VIÊN THỰC HIỆN:**

**nhóm 11:**

vÕ PHẠM HOÀNG QUÂN 18110184

TRẦN VŨ QUỐC 18110187

TRẦN LIỄU NHỰT ANH 18110077

HoÀNG NGỌC DOANH 18110089

*---Tp Hồ Chí Minh, tháng 6 năm 2021---*

**-----------------------**

***LỜI CẢM ƠN***

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn thầy Nguyễn Trần Thi Văn, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm ơn thầy đã hướng dẫn nhóm em trong quá trình học trên lớp, đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành tốt cũng như đúng thời hạn đã đề ra.

Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Đào tạo Chất Lượng Cao nói chung và ngành Công Nghệ Thông Tin nói riêng đã tận tình truyền đạt những kiến thức cần thiết giúp chúng em có nền tảng để làm nên đề tài này, đã tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian khá khó khăn với tất cả mọi người do tình hình dịch bệnh phức tạp ngày nay, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kĩ thuật và kinh nghiệm trong việc thực hiện. Do đó, trong quá trình làm đề tài có những thiếu sót là điều không thể tránh khỏi, nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý thầy cô để củng cố thêm kiến thức, hoàn thiện hơn và có thể làm tốt hơn nữa trong những lần sau. Chúng em xin chân thành cảm ơn.

Cuối lời, nhóm em kính chúc quý thầy, quý cô dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa nhóm em xin chân thành cảm ơn.

**Tp. Hồ Chí Minh, ngày 12 tháng 6 năm 2021**

**Nhóm sinh viên thực hiện**

MỤC LỤC

[MỤC LỤC 1](#_Toc74593183)

[DANH MỤC BẢNG BIỂU 3](#_Toc74593184)

[DANH MỤC HÌNH ẢNH 5](#_Toc74593185)

[CHƯƠNG I:KHẢO SÁT HIỆN TRẠNG VÀ XÁC ĐỊNH YÊU CẦU 7](#_Toc74593186)

[1. Giới thiệu đề tài 7](#_Toc74593187)

[2. Mục tiêu nghiên cứu 7](#_Toc74593188)

[3. Khảo sát hiện trạng 8](#_Toc74593189)

[3.1. Phần mềm quản lý học sinh, học viên Mona eLMS 8](#_Toc74593190)

[3.2. Phần mềm quản lý học viên TeacherKit 9](#_Toc74593191)

[3.3. Phần mềm quản lý Nino App 10](#_Toc74593192)

[4. Mô tả tổng thể 11](#_Toc74593193)

[4.1. Phối cảnh 11](#_Toc74593194)

[4.2. Chức năng ứng dụng 11](#_Toc74593195)

[4.3. Các lớp người dùng và đặc điểm người dùng 11](#_Toc74593196)

[4.4. Môi trường hoạt động 12](#_Toc74593197)

[CHƯƠNG II:PHÂN TÍCH VÀ MÔ HÌNH HOÁ YÊU CẦU 13](#_Toc74593198)

[1. Phân tích yêu cầu chức năng Giảng viên 13](#_Toc74593199)

[1.1. Chức năng quản lý thông tin môn 13](#_Toc74593200)

[1.2. Chức năng quản lý lớp học 13](#_Toc74593201)

[1.3. Chức năng quản lý tài khoản 14](#_Toc74593202)

[1.4. Chức năng quản lý sinh viên 14](#_Toc74593203)

[1.5. Chức năng quản lý điểm 15](#_Toc74593204)

[1.6. Chức năng xem trang của giáo viên 15](#_Toc74593205)

[2. Phân tích yêu cầu chức năng Khách 16](#_Toc74593206)

[3. Phân tích yêu cầu chức năng Sinh viên 16](#_Toc74593207)

[4. Phân tích yêu cầu phi chức năng 17](#_Toc74593208)

[5. Mô hình hoá yêu cầu 17](#_Toc74593209)

[5.1. Usecase tổng quát 17](#_Toc74593210)

[5.2. Usecase chi tiết chức năng giảng viên 18](#_Toc74593211)

[5.3. Usecase chi tiết chức năng khách 43](#_Toc74593212)

[5.4. Usecase chi tiết chức năng sinh viên 46](#_Toc74593213)

[CHƯƠNG III:THIẾT KẾ 51](#_Toc74593214)

[1. Thiết kế cơ sở dữ liệu 51](#_Toc74593215)

[1.1. Mô hình ERD 51](#_Toc74593216)

[1.2. Mô tả các bảng 51](#_Toc74593217)

[2. Thiết kế giao diện 55](#_Toc74593218)

[3. Thiết kế xử lý 75](#_Toc74593219)

[3.1. Sequence Diagram 75](#_Toc74593220)

[3.2. Class Diagram 84](#_Toc74593221)

[CHƯƠNG IV:ÁP DỤNG CÁC MẪU THIẾT KẾ 87](#_Toc74593224)

[1. Singleton Pattern 87](#_Toc74593225)

[1.1. Định nghĩa 87](#_Toc74593226)

[1.2. Lý do sử dụng 88](#_Toc74593227)

[1.3. Áp dụng trong ứng dụng 88](#_Toc74593228)

[2. Façade Pattern 90](#_Toc74593229)

[2.1. Định nghĩa 90](#_Toc74593230)

[2.2. Lý do sử dụng 91](#_Toc74593231)

[2.3. Áp dụng trong ứng dụng 91](#_Toc74593232)

[3. State Pattern 97](#_Toc74593233)

[3.1. Định nghĩa 97](#_Toc74593234)

[3.2. Lý do sử dụng 98](#_Toc74593235)

[3.3. Áp dụng trong ứng dụng 98](#_Toc74593236)

[CHƯƠNG V:KẾT QUẢ 100](#_Toc74593237)

[1. Ưu điểm phần mềm 100](#_Toc74593238)

[2. Nhược điểm phần mềm 100](#_Toc74593239)

[3. Khó khăn gặp phải 100](#_Toc74593240)

[4. Hướng phát triển đề tài 101](#_Toc74593241)

[LỜI KẾT 102](#_Toc74593242)

[TÀI LIỆU THAM KHẢO 103](#_Toc74593243)

# DANH MỤC BẢNG BIỂU

[Bảng biểu II‑1: Chức năng quản lý thông tin môn học 13](#_Toc74593244)

[Bảng biểu II‑2: Chức năng quản lý lớp học 14](#_Toc74593245)

[Bảng biểu II‑3: Chức năng quản lý tài khoản 14](#_Toc74593246)

[Bảng biểu II‑4: Chức năng quản lý sinh viên 15](#_Toc74593247)

[Bảng biểu II‑5: Chức năng quản lý điểm 15](#_Toc74593248)

[Bảng biểu II‑6: Chức năng xem trang giáo viên 15](#_Toc74593249)

[Bảng biểu II‑7: Chức năng khách 16](#_Toc74593250)

[Bảng biểu II‑8: Chức năng sinh viên 16](#_Toc74593251)

[Bảng biểu II‑9: Usecase quản lý môn học 19](#_Toc74593252)

[Bảng biểu II‑10: Usecase thêm môn học 20](#_Toc74593253)

[Bảng biểu II‑11: Usecase xoá môn học 22](#_Toc74593254)

[Bảng biểu II‑12: Usecase Sửa môn học 23](#_Toc74593255)

[Bảng biểu II‑13: Usecase Xem thông tin lớp học 24](#_Toc74593256)

[Bảng biểu II‑14: Usecase Thêm lớp học 26](#_Toc74593257)

[Bảng biểu II‑15: Usecase Xoá lớp học 27](#_Toc74593258)

[Bảng biểu II‑16: Usecase Sửa lớp học 28](#_Toc74593259)

[Bảng biểu II‑17: Usecase Xem thông tin tài khoản 30](#_Toc74593260)

[Bảng biểu II‑18: Usecase Thêm tài khoản 31](#_Toc74593261)

[Bảng biểu II‑19: Usecase Xoá tài khoản 33](#_Toc74593262)

[Bảng biểu II‑20: Usecase Sửa thông tin tài khoản 34](#_Toc74593263)

[Bảng biểu II‑21: Usecase Thêm sinh viên 35](#_Toc74593264)

[Bảng biểu II‑22: Usecase Xoá sinh viên 36](#_Toc74593265)

[Bảng biểu II‑23: Usecase Sửa thông tin sinh viên 37](#_Toc74593266)

[Bảng biểu II‑24: Usecase Tra cứu thông tin sinh viên 38](#_Toc74593267)

[Bảng biểu II‑25: Usecase Nhập điểm 40](#_Toc74593268)

[Bảng biểu II‑26: Usecase Sửa điểm 41](#_Toc74593269)

[Bảng biểu II‑27: Usecase Tính điểm trung bình 42](#_Toc74593270)

[Bảng biểu II‑28: Usecase Xem trang chủ Giảng viên 43](#_Toc74593271)

[Bảng biểu II‑29: Usecase Đăng nhập 45](#_Toc74593272)

[Bảng biểu II‑30: Usecase Quên mật khẩu 46](#_Toc74593273)

[Bảng biểu II‑31: Usecase Xem trang chủ Sinh viên 47](#_Toc74593274)

[Bảng biểu II‑32: Usecase Xem thông tin môn học 48](#_Toc74593275)

[Bảng biểu II‑33: Usecase Xem điểm 49](#_Toc74593276)

[Bảng biểu II‑34: Usecase Xem thời khoá biểu 50](#_Toc74593277)

[Bảng biểu III‑1: Mô tả bảng student 52](#_Toc74593278)

[Bảng biểu III‑2: Mô tả bảng user 52](#_Toc74593279)

[Bảng biểu III‑3: Mô tả bảng teacher 53](#_Toc74593280)

[Bảng biểu III‑4: Mô tả bảng classes 53](#_Toc74593281)

[Bảng biểu III‑5: Mô tả bảng scores 54](#_Toc74593282)

[Bảng biểu III‑6: Mô tả bảng subject 54](#_Toc74593283)

[Bảng biểu III‑7: Mô tả bảng teach 55](#_Toc74593284)

[Bảng biểu III‑8: Thiết kế Login 56](#_Toc74593285)

[Bảng biểu III‑9: Thiết kế ForgetPass 57](#_Toc74593286)

[Bảng biểu III‑10: Thiết kế Verify OTP 58](#_Toc74593287)

[Bảng biểu III‑11: Thiết kế Change Pass 59](#_Toc74593288)

[Bảng biểu III‑12: Thiết kế Màn hình chính của Sinh viên 60](#_Toc74593289)

[Bảng biểu III‑13: Thiết kế Personal Sinh viên 61](#_Toc74593290)

[Bảng biểu III‑14: Thiết kế Hiển thị môn học của sinh viên 61](#_Toc74593291)

[Bảng biểu III‑15: Thiết kế hiển thị điểm của sinh viên 62](#_Toc74593292)

[Bảng biểu III‑16: Màn hình chính của Admin 63](#_Toc74593293)

[Bảng biểu III‑17: Thiết kế Thêm môn học của giáo viên 64](#_Toc74593294)

[Bảng biểu III‑18: Thiết kế Xoá môn học của giáo viên 65](#_Toc74593295)

[Bảng biểu III‑19: Thiết kế Sửa môn học của giáo viên 66](#_Toc74593296)

[Bảng biểu III‑20: Thiết kế quản lý lớp 66](#_Toc74593297)

[Bảng biểu III‑21: Thiết kế Thêm lớp của giáo viên 68](#_Toc74593298)

[Bảng biểu III‑22: Thiết kế Sửa lớp của giáo viên 69](#_Toc74593299)

[Bảng biểu III‑23: Thiết kế Thêm sinh viên của Giáo viên 70](#_Toc74593300)

[Bảng biểu III‑24: Thiết kế sửa thông tin sinh viên của giáo viên 71](#_Toc74593301)

[Bảng biểu III‑25: Thiết kế Thêm giáo viên 72](#_Toc74593302)

[Bảng biểu III‑26: Thiết kế nhập điểm của giáo viên 73](#_Toc74593303)

[Bảng biểu III‑27: Thiết kế sửa điểm của giáo viên 74](#_Toc74593304)

[Bảng biểu V‑1: Bảng phân công công việc 101](#_Toc74593305)

# DANH MỤC HÌNH ẢNH

[Hình I‑1: Mona eLMS 9](#_Toc74593306)

[Hình I‑2: TeacherKit 10](#_Toc74593307)

[Hình I‑3: Nino App 10](#_Toc74593308)

[Hình II‑1: Usecase Tổng thể hệ thống 17](#_Toc74593309)

[Hình II‑2: Usecase Quản lý môn học 18](#_Toc74593310)

[Hình II‑3: Usecase Quản lý lớp học 23](#_Toc74593311)

[Hình II‑4: Usecase Quản lý tài khoản 29](#_Toc74593312)

[Hình II‑5: Usecase Quản lý sinh viên 34](#_Toc74593313)

[Hình II‑6: Usecase Quản lý điểm 39](#_Toc74593314)

[Hình II‑7: Usecase xem trang giảng viên 42](#_Toc74593315)

[Hình II‑8: Usecase chức năng khách 43](#_Toc74593316)

[Hình II‑9: Usecase chức năng sinh viên 46](#_Toc74593317)

[Hình III‑1: Lược đồ cơ sở dữ liệu 51](#_Toc74593318)

[Hình III‑2: Màn hình Đăng nhập 55](#_Toc74593319)

[Hình III‑3: Màn hình quên mật khẩu 56](#_Toc74593320)

[Hình III‑4: Màn hình xác nhận mã OTP 57](#_Toc74593321)

[Hình III‑5: Màn hình đổi mật khẩu 58](#_Toc74593322)

[Hình III‑6: Màn hình chính của sinh viên 59](#_Toc74593323)

[Hình III‑7: Màn hình thông tin sinh viên 60](#_Toc74593324)

[Hình III‑8: Màn hình môn học của sinh viên 61](#_Toc74593325)

[Hình III‑9: Màn hình điểm của sinh viên 62](#_Toc74593326)

[Hình III‑10: Màn hình trang chủ admin 62](#_Toc74593327)

[Hình III‑11: Màn hình thêm môn học của giáo viên 63](#_Toc74593328)

[Hình III‑12: Màn hình xoá môn của giáo viên 64](#_Toc74593329)

[Hình III‑13: Màn hình sửa môn của giáo viên 65](#_Toc74593330)

[Hình III‑14: Màn hình quản lý lớp của giáo viên 66](#_Toc74593331)

[Hình III‑15: Màn hình thêm lớp học của giáo viên 67](#_Toc74593332)

[Hình III‑16: Màn hình sửa lớp học của giáo viên 68](#_Toc74593333)

[Hình III‑17: Màn hình thêm sinh viên của giáo viên 69](#_Toc74593334)

[Hình III‑18: Màn hình sửa thông tin sinh viên của giáo viên 70](#_Toc74593335)

[Hình III‑19: Màn hình thêm giáo viên 71](#_Toc74593336)

[Hình III‑20: Màn hình nhập điểm của giáo viên 72](#_Toc74593337)

[Hình III‑21: Màn hình sửa điểm của giáo viên 73](#_Toc74593338)

[Hình III‑22: Màn hình quản lý tài khoản của giáo viên 74](#_Toc74593339)

[Hình III‑25: Sequence Diagram của Login 75](#_Toc74593340)

[Hình III‑26: Sequence Diagram của Forgetpass 76](#_Toc74593341)

[Hình III‑27: : Sequence Diagram của Add Class 77](#_Toc74593342)

[Hình III‑28: : Sequence Diagram của Edit Class 77](#_Toc74593343)

[Hình III‑29: Sequence Diagram của Delete Class 78](#_Toc74593344)

[Hình III‑30: Sequence Diagram của Add Student 78](#_Toc74593345)

[Hình III‑31: Sequence Diagram của Edit Student 79](#_Toc74593346)

[Hình III‑32: Sequence Diagram của Delete Student 79](#_Toc74593347)

[Hình III‑33: Sequence Diagram của Add Account 80](#_Toc74593348)

[Hình III‑34: Sequence Diagram của Edit Account 80](#_Toc74593349)

[Hình III‑35: Sequence Diagram của Delete Account 81](#_Toc74593350)

[Hình III‑36: Sequence Diagram của Add Score 81](#_Toc74593351)

[Hình III‑37: Sequence Diagram của Edit Score 82](#_Toc74593352)

[Hình III‑38: Sequence Diagram của Delete Score 82](#_Toc74593353)

[Hình III‑39: Sequence Diagram của Add Subject 83](#_Toc74593354)

[Hình III‑40: Sequence Diagram của Edit Subject 83](#_Toc74593355)

[Hình III‑41: Sequence Diagram của Delete Subject 84](#_Toc74593356)

[Hình III‑42: Class Diagram 84](#_Toc74593357)

[Hình III‑43: Class Diagram của Singleton Pattern 85](#_Toc74593358)

[Hình III‑44: Class Diagram của State Pattern 85](#_Toc74593359)

[Hình III‑45: Class Diagram của Façade Pattern 86](#_Toc74593360)

[Hình IV‑1: Singleton Pattern 87](#_Toc74593361)

[Hình IV‑2: Façade Pattern 90](#_Toc74593362)

[Hình IV‑3: State Pattern 97](#_Toc74593363)

# KHẢO SÁT HIỆN TRẠNG VÀ XÁC ĐỊNH YÊU CẦU

1. Giới thiệu đề tài

Suốt hơn nghìn năm kéo dài từ quá khứ đến hiện tại, lịch sử loài người đã ghi nhận tổng cộng ba cuộc cách mạng công nghiệp chính thức,là ba cột mốc làm thay đổi không chỉ thay đổi nền kinh tế - sản xuất đời sống xã hội thế giới nói chung, mà còn ảnh hưởng tới lối sống của từng cá nhân,xã hội. Từ những năm cuối thế kỉ XX, chúng ta đang từng bước cảm nhận hơi thở mới của một thời đại mới, khi mà tác động của ngành Công Nghệ ảnh hưởng đến hầu hết hoạt động sinh hoạt và sản xuất của tổ chức và cá nhân, Tuy nhiên, khi sự phát triển của xã hội cũng như điều kiện sống được cải tiến khiến cho nhu cầu sản phẩm gia tăng, và nguồn nhân lực cũng phát triển không kém. Việc quản lý trên giấy tờ trở nên rắc rối và không đáp ứng được lượng thông tin được mang lại. Đây là sự bất tiện lớn đối với việc quản lý của tổ chức và cá nhân trong thời đại hiện nay.

Điều này khiến cho con người hiện đại dần trở nên bận rộn hơn trong cuộc sống thường nhật và việc quản lý những số liệu trở nên ngày càng khó khăn hơn do lượng dữ liệu ngày càng nhiều. Tuy vậy, với nền tảng công nghệ ngày càng phát triển và vững mạnh, đó là một tiền đề vững mạnh cho ta có thể sử dụng để tạo ra những công cụ hỗ trợ cho trong công việc, giúp tối ưu hơn và tăng năng suất làm việc của con người thời đại mới này. Từ những lý do trên nhóm tác giả muốn áp dụng những kiến thức của mình việc quản lý môi trường Đại học nên đã chọn đề tài : “Xây dựng ứng dụng quản lý sinh viên” nhằm hỗ trợ cho quản lý sinh viên trong trường học.

1. Mục tiêu nghiên cứu

* Nghiên cứu về cấu trúc Design Patter.
* Áp dụng kiến thức thiết kế phần mềm hướng đối tượng.
* Tìm hiểu về quy trình quản lý sinh viên trong nhà trường.
* Tạo dựng ứng dụng Android trên điện thoại với Java.

1. Khảo sát hiện trạng

Trên thị trường hiện nay, có rất nhiều các đơn vị chuyên cung cấp các phần mềm quản lý sinh viên. Nhưng trong số các phần mềm toàn diện thì cũng tồn tại một số phần mềm khác nhau, được thiết kế nhằm thực hiện các nhiệm vụ ngày càng tốt hơn. Các phần mềm đó sẽ có trách nhiệm quản lý sinh viên hoặc còn được dùng trong quản lý tài chính. Một trong những lý do chúng ta cần đến những ứng dụng quản lý sinh viên vì :

* Tính năng nắm bắt thông tin nhanh, thông báo đến cho mọi học sinh, học viên các lịch nghỉ đột xuất, thông báo lịch khai giảng, lịch học, lịch thi định kỳ và lịch nghỉ học,.. mọi thứ đều được thông báo qua phần mềm. Đặc biệt trong thời điểm dịch nhạy cảm ngày nay.
* Quản lý mọi thông tin hiệu quả: Là công cụ quản lý các thông tin toàn bộ học sinh đang học tập trong trường bao gồm họ tên, mã số, học lực, học phí, địa chỉ,ngành học, lớp học… Hay là những thông tin nhập học, đăng kí môn, chuyển trường, nghỉ học, bảo lưu kết quả. Tất cả mọi thông tin đều được phân tích rõ ràng theo thứ tự cho việc dễ quản lý.
* Phần mềm có khả năng phân quyền cho phép nhiều đối tượng sử dụng và có thể truy cập và thực hiện các nhiệm vụ theo trách nhiệm phân công của mỗi người,
* Thông tin được bảo mật cao không để lộ thông tin ra ngoài. Ngoài ra còn bản back up phòng ngừa trường hợp phần mềm gặp rủi ro mất hết dữ liệu.

Hiện nay nước ta đang đối mặt với tình hinh dịch bệnh rất là phức tạp, điều này sẽ khiến cho công việc học tập bị trì trệ cũng như giáo dục bị đứng, do đó các tổ chức chịu trách nhiệm đã đưa ra hình thức học tập online để giải quyết vấn đề. Và để đảm bảo nguyên tắc công bằng và trách nhiệm thì những ứng dụng quản lý từ xa là điều cần thiết nhất nhưng hiện tại trên thị trường vẫn chưa có nhiều ứng dụng hỗ trợ tốt được. Sau đây là vài ví dụ :

* 1. *Phần mềm quản lý học sinh, học viên Mona eLMS*

Đây là phần mềm giúp cho tăng khả năng lưu trữ thông tin học sinh cũng như server hỗ trợ nhu cầu tìm kiếm, xuất file , thống kê. Giao diện đẹp cũng như tương tác mượt mà cho nhiều đối tượng người dùng. Có chức năng hoạt động tự động quản lý các dữ liệu học tập giúp cho việc tìm kiếm, tra cứu nhanh và đơn giản, phần mềm giúp tiết kiệm lên đến 50% chi phí mà hiệu quả lại tăng cao hơn so với cách quản lý truyền thống, sẽ chủ động hơn trong việc tìm kiếm các thông tin, bài giảng, mọi kiến thức chỉ cần được kết nối với internet. Tuy nhiên vẫn chưa phổ biến do thực chất là không có quá nhiều chức năng hỗ trợ giáo viên và giúp học sinh giải quyết vấn đề thông tin cá nhân.



Hình ‑: Mona eLMS

* 1. *Phần mềm quản lý học viên TeacherKit*

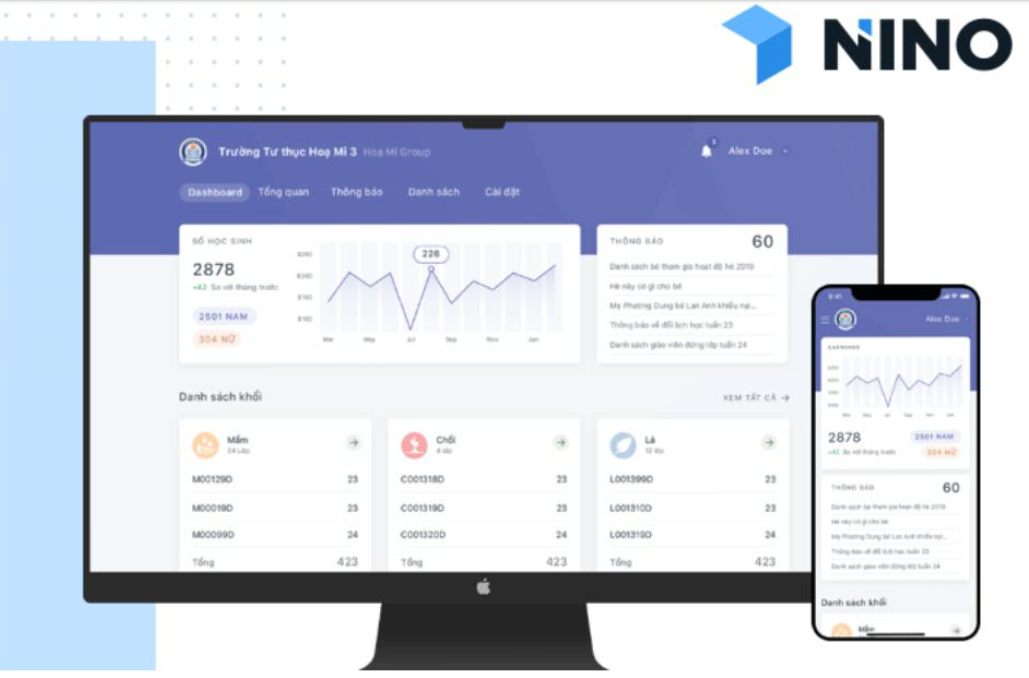
Phần mềm quản lý học sinh, học viên TeacherKit là phần mềm được thiết kế với các tính năng hiện đại cùng với các giao diện trực quan, cho phép người dùng thêm dữ liệu bằng hình ảnh, có thể điểm danh qua scan hình ảnh, gửi email đến cho phụ huynh học sinh. Phần mềm được sử dụng trên các thiết bị thông minh như ipad, iphone, pc hay hệ điều hành Android. Chức năng cơ bản như thêm danh sách lớp học, điểm số, biểu đồ chỗ ngồi, sơ đồ lớp học, kiểm tra, giám sát thái độ và hành vi học tập trong giờ, xuất các báo cáo qua hệ thống từ tổng quát đến chi tiết, điểm danh thông qua hình ảnh. Tuy nhiên ứng dụng chỉ phục vụ được trong môi trường mẫu giáo là chủ yếu do lượng thông tin lưu được rất ít và đôi lúc có hiện tượng giật lag, yêu cầu mạng Internet ổn định liên tục và có trường hợp bị thất thoát dữ liệu do đường mạng.



Hình ‑: TeacherKit

* 1. *Phần mềm quản lý Nino App*

Phần mềm Nino App chính là phần mềm quản lý giúp kết nối được với nhà trường, phụ huynh học sinh phù hợp được với hệ thống giáo dục mầm non và cần sự giám sát cao. Khi đó, phần mềm sẽ giúp cho việc điểm danh từng học sinh dễ dàng hơn, trả điểm qua sổ liên lạc điện tử, ghi chú từng học sinh qua ứng dụng, trao đổi, chia sẻ với phụ huynh nhanh chóng và dễ dàng qua ứng dụng. Với ứng dụng này thì có thể quản lý được mọi hoạt động của học sinh qua ứng dụng, nắm bắt mọi phản hồi của phụ huynh để có những cải thiện cần thiết, nâng cao hình ảnh chuyên nghiệp của nhà trường. Tuy nhiên như đã nói phần mềm chỉ phục vụ cho nhu cầu mẫu giáo.



Hình ‑: Nino App

1. Mô tả tổng thể
   1. *Phối cảnh*

Ứng dụng quản lý sinh viên được xây dựng nhằm mục đích quản lý sinh viên trong môi trường đại học. Sẽ có những đối tượng chính được hướng tới :

* Khách : từ việc đăng nhập sẽ cho ta biết đối tượng đang sử dụng app thuộc phân quyền nào.
* Sinh viên : sau khi được xác nhận là sinh viên những điều sinh viên có thể làm là xem thông tin của bản thân , xem lớp học, xem lịch học và điểm số.
* Giáo viên : người sẽ quản lý thông tin của sinh viên , điểm số và lớp học. Ngoài ra giáo viên được chọn để quản lý tài khoản chung.

Sinh viên có thể biết được những thông tin cá nhân cũng như điểm số trong quá trình học, giảng viên được quyền cập nhật cũng như sửa đổi thông tin cần thiết. Ngoài ra chỉ có giảng viên mới được thêm tài khoản cho hệ thống vì giảng viên có thông tin học sinh học giảng viên mới được quyền cấp tài khoản nhằm đảm bảo không có người ngoài được phép truy cập thông tin hệ thống nếu không phải là sinh viên trường.

* 1. *Chức năng ứng dụng*

Chức năng chính của ứng dụng :

* Quản lý chi tiết hệ thống tài khoản
* Quản lý chi tiết thông tin sinh viên
* Quản lý chi tiết thông tin lớp học
* Quản lý danh sách điểm của sinh viên
* Hiển thị thông tin cụ thể cho cả sinh viên, giảng viên.
  1. *Các lớp người dùng và đặc điểm người dùng*

Với thông tin ta có thể thấy các lớp người dùng của hệ thống được chia làm 2, dựa trên phân quyền, mục đích người dùng có thể sử dụng:

* Lớp sinh viên: Xem thông tin lớp học, điểm, thông tin cá nhân, lịch học.
* Lớp giảng viên : Quản lý tài khoản, quản lý danh sách lớp học, thông tin sinh viên, quản lý điểm số của sinh viên.
  1. *Môi trường hoạt động*
* Môi trường hoạt động của ứng dụng : Android.
* Phần mềm xây dựng ứng dụng : Android Studio.
* Ngôn ngữ sử dụng : Java
* Hệ quản trị cơ sở dữ liệu : SQLLite.

# PHÂN TÍCH VÀ MÔ HÌNH HOÁ YÊU CẦU

1. Phân tích yêu cầu chức năng Giảng viên
   1. *Chức năng quản lý thông tin môn*

Giảng viên có thể thêm môn học, chỉnh sửa thông tin môn học đã có hoặc loại bỏ môn học trên hệ thống. Ngoài ra còn hiển thị danh sách môn học.

|  |  |
| --- | --- |
| Yêu cầu | Mô tả |
| 1. Quản lý môn học | Hiển thị thông tin môn học bao gồm mã môn, tên môn, tên giáo viên phụ trách, số lượng lớp hiện có, số lượng lớp tối đa, số tín chỉ |
| 1. Thêm môn học | Nhập thông tin môn học mới bao gồm mã môn, tên môn, tên giáo viên phụ trách, số lượng lớp hiện có, số lượng lớp tối đa, số tín chỉ |
| 1. Xoá môn học | Nhập mã môn học muốn huỷ để thực hiện xoá môn |
| 1. Sửa thông tin môn | Chỉnh sửa thông tin môn học hiện có, |

Bảng biểu ‑: Chức năng quản lý thông tin môn học

* 1. *Chức năng quản lý lớp học*

Giảng viên có thể thêm lớp học mới , chỉnh sửa thông tin lớp học đã có hoặc loại bỏ lớp học trên hệ thống. Ngoài ra còn hiển thị danh sách lớp học.

|  |  |
| --- | --- |
| Yêu cầu | Mô tả |
| 1. Quản lý lớp học | Hiển thị thông tin lớp học bao gồm mã lớp, mã môn học, mã user, số lượng sinh viên trong lớp, số lượng sinh viên tối đa của từng lớp có mã môn giống với mã môn người dùng chọn. |
| 1. Thêm lớp học | Nhập thông tin lớp học mới bao gồm mã lớp, mã môn học, tên giáo viên phụ trách, số lượng sinh viên trong lớp, số lượng sinh viên tối đa. |
| 1. Xoá lớp học | Nhập mã lớp học muốn huỷ để thực hiện xoá lớp |
| 1. Sửa thông tin lớp | Chỉnh sửa thông tin lớp học hiện có, |

Bảng biểu ‑: Chức năng quản lý lớp học

* 1. *Chức năng quản lý tài khoản*

Giảng viên có thể thêm tài khoản mới , chỉnh sửa thông tin tài khoản đã có hoặc loại bỏ tài khoản trên hệ thống. Ngoài ra còn hiển thị danh sách tài khoản.

|  |  |
| --- | --- |
| Yêu cầu | Mô tả |
| 1. Quản lý tài khoản | Hiển thị thông tin tài khoản username, password, gmail, vai trò của tất cả các tài khoản hiện có. |
| 1. Thêm tài khoản | Nhập thông tin tài khoản mới gồm username, password, gmail, vai trò. |
| 1. Xoá tài khoản | Nhập username của tài khoản muốn huỷ để thực hiện xoá |
| 1. Sửa thông tin tài khoản | Chỉnh sửa thông tin tài khoản hiện có, |

Bảng biểu ‑: Chức năng quản lý tài khoản

* 1. *Chức năng quản lý sinh viên*

Giảng viên có thể thêm sinh viên mới , chỉnh sửa thông tin sinh viên đã có hoặc loại bỏ sinh viên trên hệ thống. Ngoài ra còn tìm kiếm thông tin sinh viên.

|  |  |
| --- | --- |
| Yêu cầu | Mô tả |
| 1. Thêm sinh viên | Nhập thông tin sinh viên mới bao gồm MSSV, họ tên, quê quán, ngày sinh, lớp |
| 1. Xoá sinh viên | Nhập MSSV của sinh viên muốn huỷ để thực hiện xoá |
| 1. Sửa thông tin sinh viên | Chỉnh sửa thông tin theo MSSV được nhập hiện có, |
| 1. Tra cứu thông tin sinh viên | Dựa trên MSSV được nhập để tìm kiếm toàn bộ thông tin sinh viên bao gồm : MSSV, họ tên, ngày sinh, lớp |

Bảng biểu ‑: Chức năng quản lý sinh viên

* 1. *Chức năng quản lý điểm*

Giảng viên có thể nhập điểm cho sinh viên , chỉnh sửa điểm đã nhập trên hệ thống. Ngoài ra còn tính toán điểm trung bình.

|  |  |
| --- | --- |
| Yêu cầu | Mô tả |
| 1. Nhập điểm | Nhập thông tin sinh viên bao gồm MSSV, mã môn học để thêm điểm cho sinh viên |
| 1. Sửa điểm | Chỉnh sửa điểm theo MSSV, mã môn học được nhập hiện có, |
| 1. Tính điểm trung bình | Dựa trên điểm cuối kì và giữa kì đã nhập để hệ thống tính toán điểm trung bình và lưu lại điểm trung bình đã tính. |

Bảng biểu ‑: Chức năng quản lý điểm

* 1. *Chức năng xem trang của giáo viên*

Giảng viên sau khi đăng nhập thành công ứng dụng với tài khoản được cấp trước có thể xem trang chủ tương ứng.

|  |  |
| --- | --- |
| Yêu cầu | Mô tả |
| 1. Xem trang chủ | Tuỳ theo Username đăng nhập kiểm tra liệu có phải admin không để chuyển đến trang tương ứng. |

Bảng biểu ‑: Chức năng xem trang giáo viên

1. Phân tích yêu cầu chức năng Khách

Khách có thể đăng nhập ứng dụng với tài khoản được Giảng viên cấp trước hoặc có thể xin cấp lại mật khẩu nếu quên mật khẩu.

|  |  |
| --- | --- |
| Yêu cầu | Mô tả |
| 1. Đăng nhập | Nhập thông tin Username, Password đã được cấp để đăng nhập vào giao diện chức năng tương ứng |
| 1. Quên mật khẩu | Khách nhập Username, hệ thống sẽ dựa trên thông tin được lưu truy tới Email cá nhân và yêu cầu khách truy cập Gmail để có thể thay đổi mật khẩu. |

Bảng biểu ‑: Chức năng khách

1. Phân tích yêu cầu chức năng Sinh viên

Sinh viên sau khi đăng nhập thành công ứng dụng với tài khoản được cấp trước có thể xêm thông tin điểm, thông tin lớp học, thời khoá biểu, xem thông tin cá nhân.

|  |  |
| --- | --- |
| Yêu cầu | Mô tả |
| 1. Xem trang chủ sinh viên | Tuỳ theo Username sinh viên đăng nhập sẽ tải lên màn hình thông tin của sinh viên. |
| 1. Xem thông tin môn học của sinh viên | Tuỳ theo ID User sẽ gửi lên thông tin môn học của tất cả môn học bao gồm mã môn, tên môn, tên giáo viên. |
| 1. Xem điểm | Tuỳ theo ID User sẽ gửi lên thông tin điểm môn học của tất cả môn học bao gồm mã môn, tên môn, điểm giữa kì, điểm cuối kì, điểm TB. |
| 1. Xem thời khoá biểu | Tuỳ theo ID User sẽ gửi lên thông tin môn bao gồm mã môn, tên môn, tên giáo viên, tiết học, phòng học. |

Bảng biểu ‑: Chức năng sinh viên

1. Phân tích yêu cầu phi chức năng
   1. *Bảo mật*

Phân quyền đối tượng truy cập trong hệ thống, gửi tài khoản theo email, không cho phép đăng kí tài khoản nằm ngoài kiểm soát.

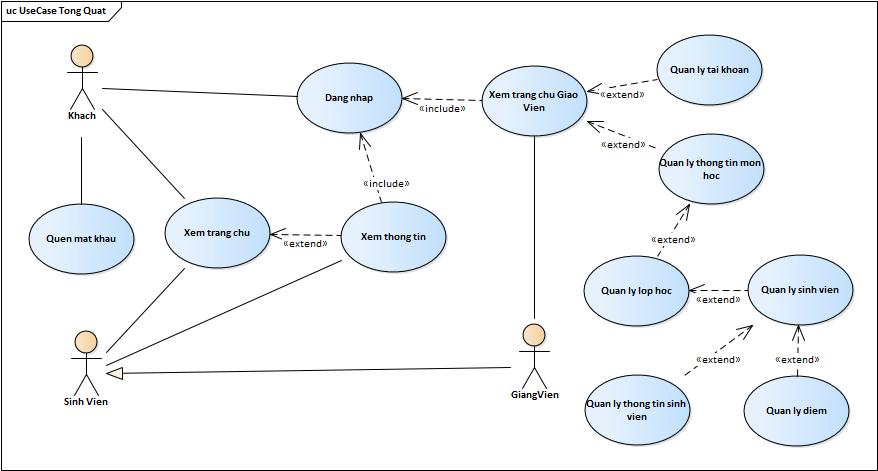
* 1. *An toàn*

Tài khoản bảo mật thông tin kín, hạn chế truy cập dữ liệu nhiều, sử dụng công nghệ tiên tiến giúp dữ liệu được bảo mật.

* 1. *Giao diện*

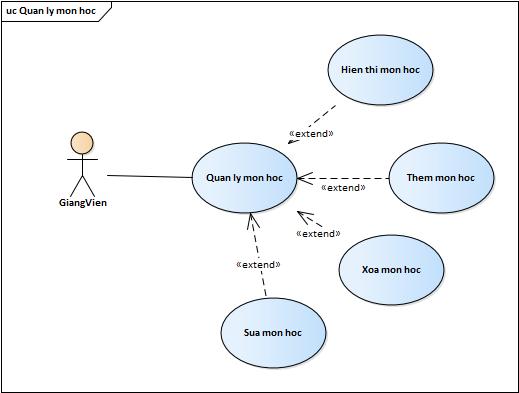
Giao diện dễ dùng, thích ứng tốt với những bản Android mới.

1. Mô hình hoá yêu cầu
   1. *Usecase tổng quát*



Hình ‑: Usecase Tổng thể hệ thống

* 1. *Usecase chi tiết chức năng giảng viên*
     1. *Quản lý môn học*



Hình ‑: Usecase Quản lý môn học

* Quản lý môn học

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Hiển thị môn học | |
| Mục đích | Hiển thị thông tin môn học bao gồm mã môn, tên môn, tên giáo viên phụ trách, số lượng lớp hiện có, số lượng lớp tối đa, số tín chỉ | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Lấy thông tin môn học học (mã môn, tên môn, tên giáo viên phụ trách, số lượng lớp hiện có, số tính chỉ, số lượng lớp tối đa) của từng môn |
|  | 3 | Nếu lấy thành công thì trả thông tin của tất cả môn học ra listview của trang quản lý môn học.  Ngược lại quay lại trang quản lý môn học và xuất thông báo lỗi kết nối. |
| 4 | Đóng cơ sở dữ liệu |
| 5 | Kết thúc |
| Kết quả | Hiển thị trang quản lý thông tin môn học | |

Bảng biểu ‑: Usecase quản lý môn học

* Thêm môn học

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Thêm lớp học | |
| Mục đích | Tạo mới một thông tin môn học bao gồm mã môn, tên môn, tên giáo viên phụ trách, số lượng lớp hiện có, số lượng lớp tối đa, số tín chỉ | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Không bỏ sót thông tin khi thêm. * Nhập thông tin cần thêm | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Kiểm tra dữ liệu đầu vào đủ và đúng kiểu hay không. Nếu có thực hiện tiếp bước 2, nếu không thì yêu cầu người dùng nhập lại |
| 2 | Mở cơ sở dữ liệu. |
| 3 | Lấy toàn bộ mã môn trong cơ sở dữ liệu để so sánh với mã môn người dùng nhập vào. |
| 4 | Nếu mã môn tồn tại thì quay lại trang thêm môn, xuất thông báo môn học tồn tại và yêu cầu người dùng nhập lại.  Ngược lại thực hiện bước 5 |
| 5 | Lưu thông tin môn học người dùng nhập vào row mới trong cơ sở dữ liệu. Nếu lưu thành công thì thực hiện bước 6.  Ngược lại quay lại trang thêm môn học xuất thông báo lỗi kết nối |
| 6 | Lấy toàn bô thông tin (mã môn, tên môn, tên giáo viên phụ trách, số lượng lớp hiện có) của tất cả môn trong cơ sở dữ liệu. |
| 7 | Quay lại trang quản lý môn học và cập nhật lại listview bằng dữ liệu lấy được ở bước 6. |
| 8 | Đóng cơ sở dữ liệu |
| 9 | Kết thúc |
| Kết quả | Hiển thị trang quản lý thông tin lớp học và thông báo thêm dữ liệu thành công. | |

Bảng biểu ‑: Usecase thêm môn học

* Xoá môn học

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Xoá môn học | |
| Mục đích | Nhập mã môn học muốn huỷ để thực hiện xoá môn | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Nhập mã môn học. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu. |
| 2 | Lấy row thông tin có mã môn học trong cơ sở dữ liệu giống với mã người dùng nhập vào. Nếu không có mã trong cơ sở dữ liệu thì quay lại trang xóa môn học và xuất thông báo môn học không tồn tại.  Ngược lại thực hiện B3 |
| 3 | Xóa row lấy ở B2 trong cơ sở dữ liệu. Nếu xóa không thành công thì trở về trang xóa môn và thông báo lỗi kết nối  Ngược lại thực hiện B4 |
| 4 | Quạy lại trang quản lý môn, lấy thông tin môn học (mã môn, tên môn, tên giáo viên phụ trách, số lượng lớp hiện có) của tất cả các môn hiện có và xuất lên listview |
| 5 | Đóng cơ sở dữ liệu |
| 6 | Kết thúc |
| Kết quả | Hiển thị trang quản lý thông tin môn học và thông báo xoá thành công | |

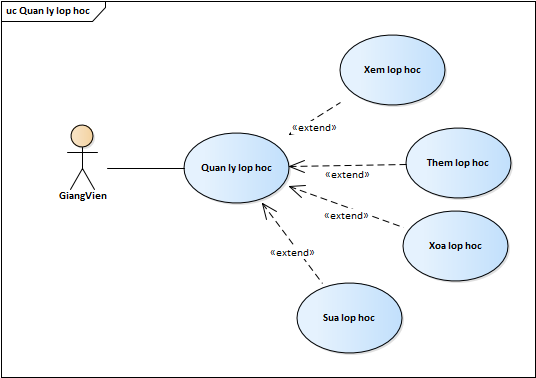
Bảng biểu ‑: Usecase xoá môn học

* Sửa môn học

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Sửa môn học | |
| Mục đích | Chỉnh sửa thông tin môn học hiện có. | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Không bỏ sót thông tin khi sửa. * Nhập thông tin cần sửa | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Kiểm tra dữ liệu đầu vào đủ và đúng kiểu hay không. Nếu có thực hiện tiếp bước 2, nếu không thì yêu cầu người dùng nhập lại |
| 2 | Mở cơ sở dữ liệu. |
| 3 | Lấy row chứa thông tin môn học (mã môn, tên môn, tên giáo viên phụ trách, số lượng lớp hiện có, số lượng lớp tối đa) có mã môn khớp với mã người dùng nhập |
| 4 | Nếu mã môn không tồn tại thì quay lại trang sửa môn, xuất thông báo môn học không tồn tại và yêu cầu người dùng nhập lại  Ngược lại thực hiện bước 5 |
| 5 | Lưu thông tin môn học người dùng nhập vào vị trí row vừa lấy trong cơ sở dữ liệu. Nếu lưu thành công thì thực hiện bước 6.  Ngược lại quay lại trang sửa môn học xuất thông báo lỗi kết nối |
| 6 | Lấy toàn bô thông tin (mã môn, tên môn, tên giáo viên phụ trách, số lượng lớp hiện có) của tất cả môn trong cơ sở dữ liệu. |
| 7 | Quay lại trang quản lý môn học và cập nhật lại listview bằng dữ liệu lấy được ở bước 6 |
| 8 | Đóng cơ sở dữ liệu |
| 9 | Kết thúc |
| Kết quả | Hiển thị trang quản lý thông tin môn học và cập nhật lại dữ liệu, thông báo sửa thành công. | |

Bảng biểu ‑: Usecase Sửa môn học

* + 1. *Quản lý lớp học*



Hình ‑: Usecase Quản lý lớp học

* Xem thông tin lớp học

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Xem lớp học | |
| Mục đích | Hiển thị trang quản lý thông tin lớp học bao gồm mã lớp, mã môn học, mã user, số lượng sinh viên trong lớp, số lượng sinh viên tối đa của từng lớp có mã môn giống với mã môn người dùng chọn. | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Mã môn được nhập. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Lấy thông tin lớp học (mã lớp, mã môn học, mã user, số lượng sinh viên trong lớp, số lượng sinh viên tối đa) của từng lớp có mã môn giống với mã môn người dùng chọn. |
|  | 3 | Nếu lấy thành công thì trả thông tin của tất cả lớp học ra listview của trang quản lý lớp học.  Ngược lại quay lại trang quản lý lớp học và xuất thông báo lỗi kết nối |
| 4 | Đóng cơ sở dữ liệu |
| 5 | Kết thúc |
| Kết quả | Hiển thị trang quản lý thông tin lớp học | |

Bảng biểu ‑: Usecase Xem thông tin lớp học

* Thêm lớp học

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Thêm lớp học | |
| Mục đích | Tạo mới một thông tin lớp học bao gồm mã lớp, mã môn học, tên giáo viên phụ trách, số lượng sinh viên trong lớp, số lượng sinh viên tối đa. | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Không bỏ sót thông tin khi thêm. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Kiểm tra dữ liệu đầu vào đủ và đúng kiểu hay không. Nếu có thực hiện tiếp bước 2, nếu không thì yêu cầu người dùng nhập lại |
| 2 | Mở cơ sở dữ liệu. |
| 3 | Lấy toàn bộ mã lớp trong cơ sở dữ liệu để so sánh với mã lớp người dùng nhập vào |
| 4 | Nếu mã lớp tồn tại, quay lại trang thêm lớp, xuất thông báo lớp học tồn tại và yêu cầu người dùng nhập lại .  Ngược lại thực hiện bước 5 |
| 5 | Lưu thông tin lớp học người dùng nhập vào row mới trong cơ sở dữ liệu. Nếu lưu thành công thì thực hiện bước 6.  Ngược lại quay lại trang thêm lớp học xuất thông báo lỗi kết nối |
| 6 | Lấy toàn bô thông tin (mã lớp, tên giáo viên phụ trách, số lượng sinh viên trong lớp, số lượng sinh viên tối đa) của tất cả môn trong cơ sở dữ liệu |
| 7 | Quay lại trang quản lý lớp học và cập nhật lại listview bằng dữ liệu lấy được ở bước 6 |
| 8 | Đóng cơ sở dữ liệu |
| 9 | Kết thúc |
| Kết quả | Hiển thị trang quản lý thông tin lớp học | |

Bảng biểu ‑: Usecase Thêm lớp học

* Xoá lớp học

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Xoá lớp học | |
| Mục đích | Xoá lớp học dựa trên mã lớp đã tồn tại | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Nhập mã lớp học. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu. |
| 2 | Lấy row thông tin có mã lớp học trong cơ sở dữ liệu giống với mã người dùng nhập vào. Nếu không có mã trong cơ sở dữ liệu thì quay lại trang xóa lớp học và xuất thông báo lớp học không tồn tại.  Ngược lại thực hiện B3 |
| 3 | Xóa row lấy ở B2 trong cơ sở dữ liệu. Nếu xóa không thành công thì trở vè trang xóa lớp và thông báo lỗi kết nối.  Ngược lại thực hiện B4. |
| 4 | Quạy lại trang quản lý lớp , Lấy thông tin lớp học (mã lớp, tên giáo viên phụ trách, số lượng sinh viên trong lớp, số lượng sinh viên tối đa) của tất cả các lớp hiện có và xuất lên listview |
| 5 | Đóng cơ sở dữ liệu |
| 6 | Kết thúc |
| Kết quả | Hiển thị trang quản lý thông tin lớp học và thông báo xoá thành công | |

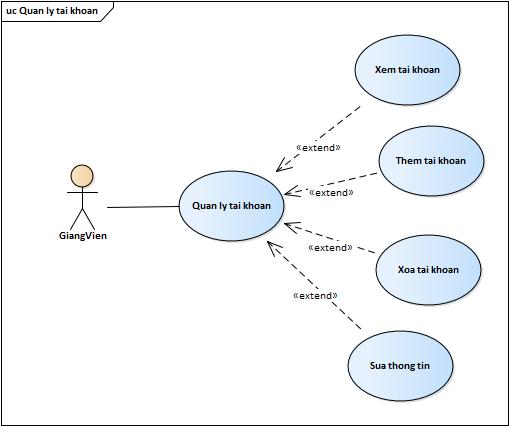
Bảng biểu ‑: Usecase Xoá lớp học

* Sửa lớp học

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Sửa lớp học | |
| Mục đích | Chỉnh sửa thông tin lớp học đã tồn tại | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Không bỏ sót thông tin khi sửa. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Kiểm tra dữ liệu đầu vào đủ và đúng kiểu hay không. Nếu có thực hiện tiếp bước 2, nếu không thì yêu cầu người dùng nhập lại |
| 2 | Mở cơ sở dữ liệu. |
| 3 | Lấy row chứa thông tin lớp học (mã lớp, mã môn học, tên giáo viên phụ trách, số lượng sinh viên trong lớp, số lượng sinh viên tối đa) có mã lớp khớp với mã người dùng nhập |
| 4 | Nếu mã lớp không tồn tại thì quay lại trang sửa lớp, xuất thông báo môn học không tồn tại và yêu cầu người dùng nhập lại.  Ngược lại thực hiện bước 5 |
| 5 | Lưu thông tin lớp học người dùng nhập vào vị trí row vừa lấy trong cơ sở dữ liệu. Nếu lưu thành công thì thực hiện bước 6.  Ngược lại quay lại trang sửa lớp học xuất thông báo lỗi kết nối |
| 6 | Lấy toàn bô thông tin (mã lớp, tên giáo viên phụ trách, số lượng sinh viên trong lớp, số lượng sinh viên tối đa) của tất cả môn trong cơ sở dữ liệu. |
| 7 | Quay lại trang quản lý lớp học và cập nhật lại listview bằng dữ liệu lấy được ở bước 6. |
| 8 | Đóng cơ sở dữ liệu |
| 9 | Kết thúc |
| Kết quả | Hiển thị trang quản lý thông tin lớp học và cập nhật lại dữ liệu, thông báo sửa thành công. | |

Bảng biểu ‑: Usecase Sửa lớp học

* + 1. *Quản lý tài khoản*



Hình ‑: Usecase Quản lý tài khoản

* Xem thông tin tài khoản

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Xem tài khoản | |
| Mục đích | Hiển thị thông tin tài khoản username, password, gmail, vai trò của tất cả các tài khoản hiện có. | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Chọn chức năng hiển thị danh sách | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Lấy thông tin tài khoản (username, password, gmail, vai trò) của tất cả tài khoản. |
| 3 | Nếu lấy thành công thì trả thông tin của tất cả tài khoản ra listview của trang quản lý tài khoản.  Ngược lại quay lại trang quản lý tài khoản và xuất thông báo lỗi kết nối. |
| 4 | Đóng cơ sở dữ liệu |
| 5 | Kết thúc |
| Kết quả | Hiển thị trang quản lý thông tin tài khoản | |

Bảng biểu ‑: Usecase Xem thông tin tài khoản

* Thêm tài khoản

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Thêm tài khoản | |
| Mục đích | Tạo mới một thông tin tài khoản gồm username, password, gmail, vai trò. | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Không bỏ sót thông tin khi thêm. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Kiểm tra dữ liệu đầu vào đủ và đúng kiểu hay không. Nếu có thực hiện tiếp bước 2, nếu không thì yêu cầu người dùng nhập lại |
| 2 | Mở cơ sở dữ liệu. |
| 3 | Lấy toàn bộ username trong cơ sở dữ liệu để so sánh với username người dùng nhập vào |
| 4 | Nếu username tồn tại thì quay lại trang thêm tài khoản, xuất thông báo tài khoản đã tồn tại và yêu cầu người dùng nhập lại.  Ngược lại thực hiện bước 5 |
| 5 | Lưu thông tin tài khoản người dùng nhập vào row mới trong cơ sở dữ liệu.Nếu lưu thành công thì thực hiện bước 6.  Ngược lại quay lại trang thêm tài khoản xuất thông báo lỗi kết nối |
| 6 | Lấy toàn bộ Thông tin tài khoản (username, password, gmail, vai trò) của tất cả tài khoản trong cơ sở dữ liệu. |
| 7 | Quay lại trang quản lý tài khoản và cập nhật lại listview bằng dữ liệu lấy được ở bước 6 |
| 8 | Đóng cơ sở dữ liệu |
| 9 | Kết thúc |
| Kết quả | Hiển thị trang quản lý tài khoản và thông báo thành công | |

Bảng biểu ‑: Usecase Thêm tài khoản

* Xoá tài khoản

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Xoá tài khoản | |
| Mục đích | Nhập username của tài khoản muốn huỷ để thực hiện xoá | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Nhập username. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu. |
| 2 | Lấy row thông tin có username trong cơ sở dữ liệu giống với username dùng nhập vào. Nếu không có mã trong cơ sở dữ liệu thì quay lại trang xóa tài khoản và xuất thông báo tài khoản không tồn tại.  Ngược lại thực hiện B3 |
| 3 | Xóa row lấy ở B2 trong cơ sở dữ liệu. Nếu xóa không thành công thì trở vè trang xóa tài khoản và thông báo lỗi kết nối.  Ngược lại thực hiện B4. |
| 4 | Quạy lại trang quản lý tài khoản. Lấy Thông tin tài khoản (username, password, gmail, vai trò) của tất cả các tài khoản hiện có và xuất lên listview |
| 5 | Đóng cơ sở dữ liệu |
| 6 | Kết thúc |
| Kết quả | Hiển thị trang quản lý thông tin tài khoản và thông báo xoá thành công | |

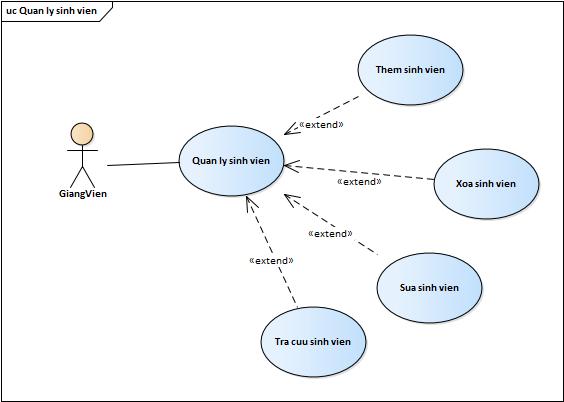
Bảng biểu ‑: Usecase Xoá tài khoản

* Sửa thông tin tài khoản

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Sửa thông tin | |
| Mục đích | Chỉnh sửa thông tin tài khoản đã tồn tại | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Không bỏ sót thông tin khi sửa. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Kiểm tra dữ liệu đầu vào đủ và đúng kiểu hay không. Nếu có thực hiện tiếp bước 2, nếu không thì yêu cầu người dùng nhập lại |
| 2 | Mở cơ sở dữ liệu. |
| 3 | Lấy row chứa thông tin tài khoản (username, password, gmail, vai trò) có username khớp với mã người dùng nhập |
| 4 | Nếu username không tồn tại thì quay lại trang sửa tài khoản, xuất thông báo tài khoản không tồn tại và yêu cầu người dùng nhập lại.  Ngược lại thực hiện bước 5 |
| 5 | Lưu thông tin tài khoản người dùng nhập vào vị trí row vừa lấy trong cơ sở dữ liệu. Nếu lưu thành công thì thực hiện bước 6.  Ngược lại quay lại trang sửa tài khoản xuất thông báo lỗi kết nối |
| 6 | Lấy toàn bộ thông tin tài khoản (username, password, gmail, vai trò) của tất cả tài khoản trong cơ sở dữ liệu. |
| 7 | Quay lại trang quản lý tài khoản và cập nhật lại listview bằng dữ liệu lấy được ở bước 6. |
| 8 | Đóng cơ sở dữ liệu |
| 9 | Kết thúc |
| Kết quả | Hiển thị trang quản lý thông tin tài khoản và cập nhật lại dữ liệu, thông báo sửa thành công. | |

Bảng biểu ‑: Usecase Sửa thông tin tài khoản

* + 1. *Quản lý sinh viên*



Hình ‑: Usecase Quản lý sinh viên

* Thêm sinh viên

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Thêm sinh viên | |
| Mục đích | Nhập thông tin sinh viên mới bao gồm MSSV, họ tên, quê quán, ngày sinh, lớp | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Không bỏ sót thông tin khi thêm. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu. |
| 2 | Lấy MSSV kiểm tra xem trong bảng sinh viên đã tồn tại hay chưa |
| 3 | Nếu đã tồn tại thì hiện thông báo đã tồn tại MSSV |
| 4 | Nếu chưa có thì tiến hành lưu MSSV, họ tên, quê quán, ngày sinh, lớp vào bảng sinh viên. Xuất hiện màn hình thông báo đã thêm thành công |
| 5 | Đóng cơ sở dữ liệu |
| 6 | Kết thúc |
| Kết quả | Hiển thị màn hình thông báo thành công hoặc thất bại | |

Bảng biểu ‑: Usecase Thêm sinh viên

* Xoá sinh viên

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Xoá sinh viên | |
| Mục đích | Nhập MSSV của sinh viên muốn huỷ để thực hiện xoá | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Nhập MSSV | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu. |
| 2 | Lấy MSSV kiểm tra xem có tồn tại trong bảng sinh viên hay không |
| 3 | Nếu không hiện thông báo không tìm thấy |
| 4 | Nếu có thì tiến hành xóa toàn bộ thông tin theo hàng trong bảng sinh viên. Xuất hiện màn hình thông báo xóa thành công |
| 5 | Đóng cơ sở dữ liệu |
| 6 | Kết thúc |
| Kết quả | Màn hình thông báo thành công hoặc thất bại | |

Bảng biểu ‑: Usecase Xoá sinh viên

* Sửa thông tin sinh viên

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Sửa sinh viên | |
| Mục đích | Chỉnh sửa thông tin theo MSSV được nhập hiện có | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Không bỏ sót thông tin khi sửa. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu. |
| 2 | Kiểm tra xem MSSV có tồn tại trong bảng sinh viên hay chưa |
| 3 | Nếu chưa có thì thông báo không tồn tại |
| 4 | Nếu đã có thì tiến hành lưu thay đổi thông tin ( không sửa MSSV) vào bảng sinh viên. Còn nếu thay đổi cả MSSV thì tiến hành kiểm trả xem trong bảng sinh viên đã tồn tại MSSV này chưa nếu rồi thì thông báo sửa thất bại, còn nếu chưa thì tiến hành lưu MSSV, địa chỉ, ngày sinh, lớp, quê quán và thông báo đã thêm thành công  Ngược lại thực hiện bước 5 |
| 5 | Đóng cơ sở dữ liệu |
| 6 | Kết thúc |
| Kết quả | Hiển thị màn hình thông báo thành công hoặc thất bại | |

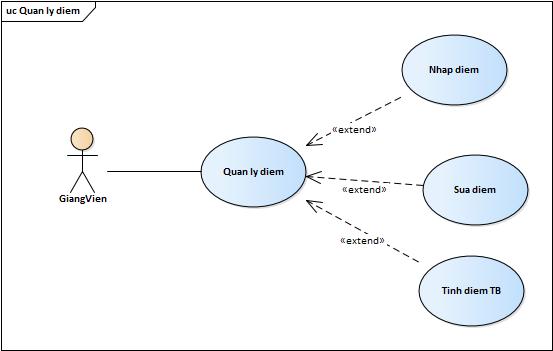
Bảng biểu ‑: Usecase Sửa thông tin sinh viên

* Tra cứu thông tinh sinh viên

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Tra cứu sinh viên | |
| Mục đích | Dựa trên MSSV được nhập để tìm kiếm toàn bộ thông tin sinh viên bao gồm : MSSV, họ tên, ngày sinh, lớp | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Nhập MSSV | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Kiểm tra xem MSSV có tồn tại trong bảng sinh viên hay không |
| 3 | Nếu không thông báo không tồn tại.  Nếu có thì nạp dữ diệu từ cơ sở dữ liệu lên theo từng hàng ứng với MSSV tương ứng |
| 4 | Đóng cơ sở dữ liệu |
| 5 | Kết thúc |
| Kết quả | Hiển thị thông tin sinh viên tương ứng và thông báo thành công hoặc chỉ riêng thông báo thất bại. | |

Bảng biểu ‑: Usecase Tra cứu thông tin sinh viên

* + 1. *Quản lý điểm*



Hình ‑: Usecase Quản lý điểm

* Nhập điểm

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Nhập điểm | |
| Mục đích | Nhập thông tin sinh viên bao gồm MSSV, mã môn học để thêm điểm cho sinh viên | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Nhập MSSV, mã môn học | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Nhập mã môn học cần sửa. Kiểm tra trong bảng môn học có tồn tại hay không.  Nếu không thông báo không tồn tại |
| 3 | Nếu có tiến hành kiểm tra MSSV có tồn tại trong bảng sinh viên hay không.  Nếu không thông báo không tồn tại |
| 4 | Nếu tồn tại thì tiến hành lưu lại điểm đã sửa theo hàng ứng với MSSV. Xuất hiện màn hình thông báo thành công |
| 5 | Đóng cơ sở dữ liệu |
| 6 | Kết thúc |
| Kết quả | Màn hình thông báo thành công hoặc thất bại. | |

Bảng biểu ‑: Usecase Nhập điểm

* Sửa điểm

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Sửa điểm | |
| Mục đích | Chỉnh sửa điểm theo MSSV, mã môn học được nhập hiện có | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Nhập mã môn học, MSSV | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu. |
| 2 | Nhập mã môn học cần sửa. Kiểm tra trong bảng môn học có tồn tại hay không.  Nếu không thông báo không tồn tại |
| 3 | Nếu có tiến hành kiểm tra MSSV có tồn tại trong bảng sinh viên hay không. Sang bước 4  Nếu không thông báo không tồn tại |
| 4 | Nếu tồn tại thì tiến hành thêm điểm đã sửa theo hàng ứng với MSSV |
| 5 | Sau đó tiến hành sửa điểm. Xuất hiện màn hình thông báo thành công |
| 6 | Đóng cơ sở dữ liệu |
| 7 | Kết thúc |
| Kết quả | Hiển thị màn hình thông báo thành công hoặc thất bại | |

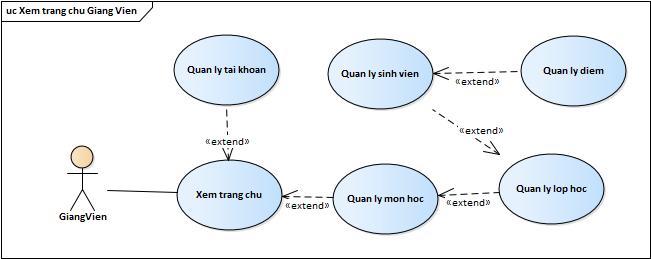
Bảng biểu ‑: Usecase Sửa điểm

* Tính điểm trung bình

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Tính điểm TB | |
| Mục đích | Dựa trên điểm cuối kì và giữa kì đã nhập để hệ thống tính toán điểm trung bình và lưu lại điểm trung bình đã tính. | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. * Nhập điểm giữa kì, điểm cuối kì | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu. |
| 2 | Nhập điểm giữa kì và cuối kì theo MSSV |
| 3 | Lưu điểm xuống cơ sở dữ liệu |
| 4 | Tình điểm trung bình bằng cách lấy điểm giữa kì cộng cuối kì chia đôi. |
| 5 | Màn hình hiển thị điểm trung bình của sinh vừa mới được nhập điểm. |
| 6 | Đóng cơ sở dữ liệu |
| 7 | Kết thúc |
| Kết quả | Hiển thị màn hình thông báo thành công và hiển thị điểm trung bình sinh viên hoặc thất bại. | |

Bảng biểu ‑: Usecase Tính điểm trung bình

* + 1. *Xem trang giảng viên*

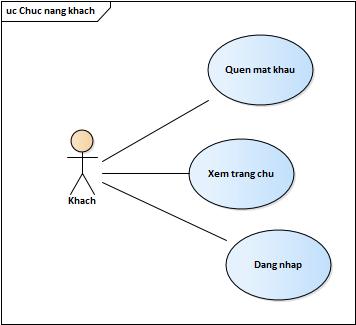


Hình ‑: Usecase xem trang giảng viên

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Xem trang chủ | |
| Mục đích | Tuỳ theo Username đăng nhập kiểm tra liệu có phải admin không để chuyển đến trang tương ứng. | |
| Tác nhân chính | Giảng viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Giảng viên * Kết nối mạng ổn định. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Lấy thông tin user (vai trò) có username tương ứng trong bảng user sau khi đăng nhập thành công |
| 3 | Kiểm tra vai trò có phải là Giảng viên hay không. Nếu phải chuyển đến trang Giảng viên  Ngược lại chuyển đến trang dành cho sinh viên |
| 4 | Đóng cơ sở dữ liệu |
| 5 | Kết thúc |
| Kết quả | Thông báo thành công và trang chủ giảng viên hoặc trang sinh viên | |

Bảng biểu ‑: Usecase Xem trang chủ Giảng viên

* 1. *Usecase chi tiết chức năng khách*



Hình ‑: Usecase chức năng khách

* Đăng nhập

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Đăng nhập | |
| Mục đích | Nhập thông tin Username, Password đã được cấp để đăng nhập vào giao diện chức năng tương ứng | |
| Tác nhân chính | Người dùng | |
| Yêu cầu | * Người dùng đang sử dụng ứng dụng. * Nhập Username, Password. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Lấy user và password nhập vào tra với bảng cơ sở dữ liệu xem tên user và password có khớp với nhau và tồn tại hay không từ cột user và password. |
| 3 | Nếu tồn tại user trong cột user thì ta tiếp tục kiểm tra password không khớp thì xuất ra thông báo user hoặc pass chưa đúng.  Nếu trường hợp user không tồn tại thì xuất ra thông báo user hoặc pass chưa đúng. Quay lại trang đăng nhập |
| 4 | Nếu tồn tại user trong cột user thì ta tiếp tục kiểm tra password tại cột user khớp với mật khẩu đăng nhập thì sẽ chuyển user tới trang admin hoặc trang sinh viên |
| 5 | Đóng cơ sở dữ liệu |
| 6 | Kết thúc |
| Kết quả | Hiển thị thông báo đăng nhập thành công và dẫn người dùng sang trang chủ của sinh viên hoặc giáo viên. Còn không sẽ thông báo thất bại và vẫn ở trang đăng nhập. | |

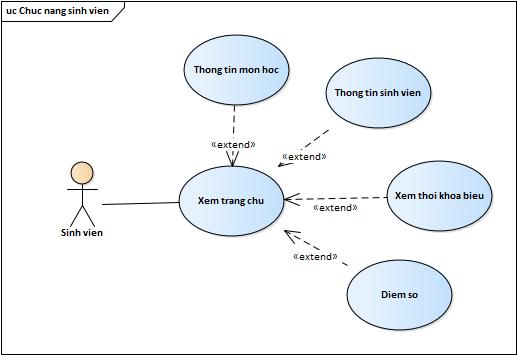
Bảng biểu ‑: Usecase Đăng nhập

* Quên mật khẩu

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Quên mật khẩu | |
| Mục đích | Khách nhập Username, hệ thống sẽ dựa trên thông tin được lưu truy tới Email cá nhân và yêu cầu khách truy cập Gmail để có thể thay đổi mật khẩu. | |
| Tác nhân chính | Người dùng | |
| Yêu cầu | * Người dùng đã có tài khoản được cấp * Kết nối mạng ổn định. * Nhập Username | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Đăng nhập thông tin mail và so sánh với bảng thông tin cá nhân cột mail |
| 3 | Nếu thông tin mail không khớp với cột mail với bảng User thì xuất ra thông báo tài khoản chưa được kích hoạt. Quay lại trang Quên mật khẩu.  Nếu đúng sang bước 4 |
| 4 | Nếu thông tin mail khớp với cột mail với cơ sở dữ liệu thì xuất ra thông báo bạn vui lòng đăng nhập mail để nhận mật khẩu ( mật khẩu được cấp và không thể thay đổi mật khẩu khi cấp mật khẩu sẽ là mật khẩu hiện đang dùng nên không cần bước xác nhận đổi qua link mail chỉ đơn giản gửi lại mật khẩu đang dùng). Quay lại trang đăng nhập |
| 5 | Đóng cơ sở dữ liệu |
| 6 | Kết thúc |
| Kết quả | Quay lại trang đăng nhập và thông báo gửi thành công. | |

Bảng biểu ‑: Usecase Quên mật khẩu

* 1. *Usecase chi tiết chức năng sinh viên*



Hình ‑: Usecase chức năng sinh viên

* Xem trang chủ sinh viên.

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Xem trang chu | |
| Mục đích | Tuỳ theo Username sinh viên đăng nhập sẽ tải lên màn hình thông tin của sinh viên. | |
| Tác nhân chính | Sinh viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Sinh viên * Kết nối mạng ổn định. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Lấy thông tin user (vai trò) có username tương ứng trong bảng user sau khi đăng nhập thành công |
| 3 | Sau khi đăng nhập thành công lấy cột mã user lấy kiểm tra bảng thông tin User và xuất ra thông báo chào mừng User)  Chuyển đến trang chủ sinh viên (các button thông tin môn học, thông tin cá nhân, thời khóa biểu, xem điểm) |
| 4 | Đóng cơ sở dữ liệu |
| 5 | Kết thúc |
| Kết quả | Thông báo thành công và trang chủ sinh viên | |

Bảng biểu ‑: Usecase Xem trang chủ Sinh viên

* Xem thông tin môn học của sinh viên.

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Thông tin môn học | |
| Mục đích | Tuỳ theo ID User sẽ gửi lên thông tin môn học của tất cả môn học bao gồm mã môn, tên môn, tên giáo viên. | |
| Tác nhân chính | Sinh viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Sinh viên * Kết nối mạng ổn định. * Thông tin User. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Lấy ID user so sánh với bảng môn học cột (ID user) lấy tất cả các thông tin bảng môn học( Mã môn học, tên môn học, tên giao viên ) có ID user khớp trong bảng môn học. |
| 3 | Xuất ra màn hình thông tin các môn học (mã môn, tên môn, tên giáo viên phụ trách) trùng với mã ID user của người dùng chứa trong listview của màn hình. |
| 4 | Đóng cơ sở dữ liệu |
| 5 | Kết thúc |
| Kết quả | Thông báo thành công và hiển thị màn hình thông tin môn học.Hoặc hiện thông báo thất bại. | |

Bảng biểu ‑: Usecase Xem thông tin môn học

* Xem điểm.

|  |  |  |
| --- | --- | --- |
| Tên Usecase | Điểm số | |
| Mục đích | Tuỳ theo ID User sẽ gửi lên thông tin điểm môn học của tất cả môn học bao gồm mã môn, tên môn, điểm giữa kì, điểm cuối kì, điểm TB. | |
| Tác nhân chính | Sinh viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Sinh viên * Kết nối mạng ổn định. * Thông tin User | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Lấy ID user so sánh với bảng điểm cột (ID user) lấy tất cả các thông tin bảng môn học(mã môn, tên môn, điểm giữa kì, điểm cuối kì, điểm trung bình ) có ID user khớp trong bảng môn học |
| 3 | Xuất ra màn hình thông tin các môn học (mã môn, tên môn, điểm giữa kì, điểm cuối kì, điểm trung bình) trùng với mã ID user của người dùng chứa trong listview của màn hình |
| 4 | Đóng cơ sở dữ liệu |
| 5 | Kết thúc |
| Kết quả | Thông báo thành công và đưa ra màn hình xem thông tin môn học. Hoặc thông báo thất bại | |

Bảng biểu ‑: Usecase Xem điểm

* Xem thời khoá biểu.

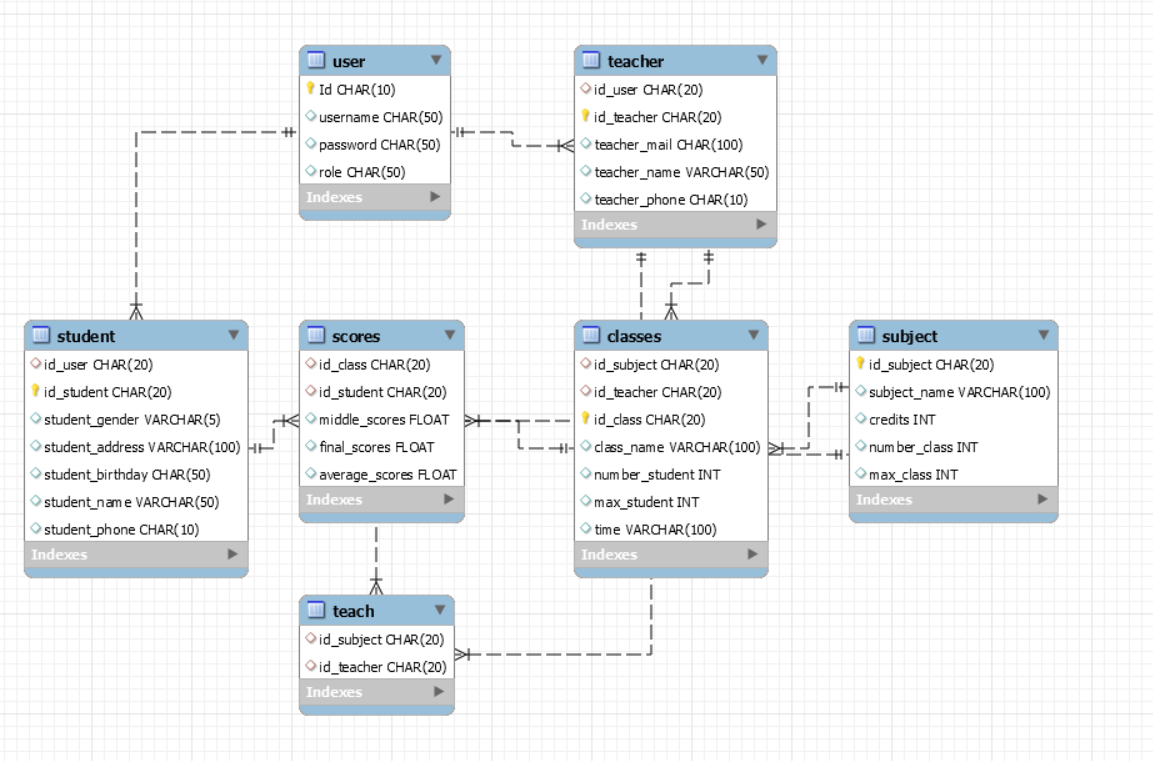
|  |  |  |
| --- | --- | --- |
| Tên Usecase | Xem thời khoá biểu | |
| Mục đích | Tuỳ theo ID User sẽ gửi lên thông tin môn bao gồm mã môn, tên môn, tên giáo viên, tiết học, phòng học. | |
| Tác nhân chính | Sinh viên | |
| Yêu cầu | * Người dùng đã đăng nhập vào tài khoản phân quyền dành riêng cho Sinh viên * Kết nối mạng ổn định. * Thông tin user. | |
| Thuật toán | STT | Công việc thực hiện |
| 1 | Mở cơ sở dữ liệu |
| 2 | Lấy ID user so sánh với bảng môn học cột (ID user) lấy tất cả các thông tin bảng môn học(mã môn, tên môn, Tên giáo viên, tiết học, phòng học) có ID user khớp trong bảng môn học |
| 3 | Xuất ra màn hình thông tin các môn học (mã môn, tên môn, Tên giáo viên, tiết học, phòng học) trùng với mã ID user của người dùng chứa trong listview của màn hình |
| 4 | Đóng cơ sở dữ liệu |
| 5 | Kết thúc |
| Kết quả | Thông báo thành công và đưa ra màn hình xem thông tin môn học.  Hoặc thông báo thất bại | |

Bảng biểu ‑: Usecase Xem thời khoá biểu

.

# THIẾT KẾ

1. Thiết kế cơ sở dữ liệu
   1. *Mô hình ERD*



Hình ‑: Lược đồ cơ sở dữ liệu

* 1. *Mô tả các bảng*
     1. *Bảng student*

Bảng student được sử dụng để lưu trữ thông tin chi tiết củasinh viên

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** |
| id\_user | Mã người dùng | char (20) | Khoá ngoại |
| id\_student | Mã sinh viên | char (20) | Khoá chính |
| student\_gender | Giới tính sinh viên | varchar (5) |  |
| student\_address | Địa chỉ sinh viên | varchar (100) |  |
| student\_birth | Ngày sinh sinh viên | char (50) |  |
| student \_name | Tên sinh viên | varchar (50) |  |
| student\_phone | Số điện thoại sinh viên | char (10) |  |

Bảng biểu ‑: Mô tả bảng student

* + 1. *Bảng user*

Bảng user được dùng để lưu thông tin tài khoản mật khẩu và phân quyền người dùng

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** |
| Id | Mã người dùng | char (10) | Khoá chính |
| username | Tên đăng nhập | char (50) |  |
| password | Mật khẩu | char (50) |  |
| role | Chức vụ | char (50) |  |

Bảng biểu ‑: Mô tả bảng user

* + 1. *Bảng teacher*

Bảng teacher được dùng để lưu thông tin chi tiết của giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** |
| id\_user | Mã người dùng | char (20) | Khoá ngoại |
| id\_teacher | Mã giáo viên | char (20) | Khoá chính |
| teacher\_mail | Mail giáo viên | char (100) |  |
| teacher \_name | Tên giáo viên | varchar (50) |  |
| teacher\_phone | Số điện thoại giáo viên | char (10) |  |

Bảng biểu ‑: Mô tả bảng teacher

* + 1. *Bảng classes*

Bảng classes được dùng để lưu thông tin chi tiết của lớp học

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** |
| id\_subject | Mã môn học | char (20) | Khoá ngoại |
| id\_teacher | Mã giáo viên | char (20) | Khoá ngoại |
| id\_class | Mã lớp | char (20) | Khoá chính |
| class \_name | Tên lớp | varchar (100) |  |
| number\_student | Số lượng học viên | int |  |
| max\_student | Tối đa học viên | int |  |
| time | Thời gian | varchar (100) |  |

Bảng biểu ‑: Mô tả bảng classes

* + 1. *Bảng scores*

Bảng scores được dùng để lưu thông tin chi tiết của điểm số

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** |
| id\_class | Mã lớp | char (20) | Khoá ngoại |
| id\_student | Mã sinh viên | char (20) | Khoá ngoại |
| Middle\_score | Điểm giữa kì | Float |  |
| Final\_score | Điểm cuối kì | Float |  |
| Avenrage\_score | Điểm trung bình | Float |  |

Bảng biểu ‑: Mô tả bảng scores

* + 1. *Bảng subject*

Bảng subject được dùng để lưu thông tin chi tiết của môn học

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** |
| id\_subject | Mã môn | char (20) | Khoá chính |
| subject\_name | Tên môn | char (100) |  |
| credits | Tín chỉ | int |  |
| number\_class | Số lượng lớp | int |  |
| max\_class | Số lượng lớp tối đa | int |  |

Bảng biểu ‑: Mô tả bảng subject

* + 1. *Bảng teach*

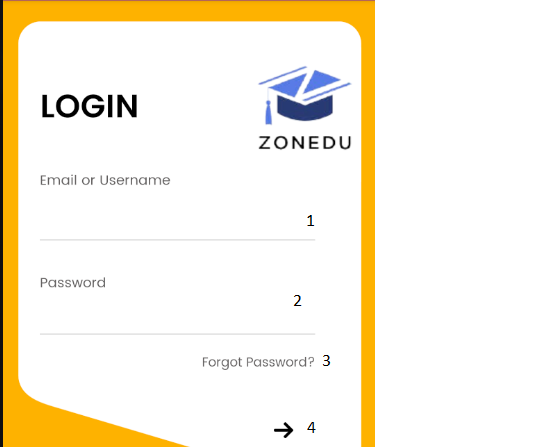
Bảng teach được dùng để tổng hợp danh sách giáo viên dạy theo môn

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** |
| id\_subject | Mã môn | char (20) | Khoá ngoại |
| id\_teacher | Mã giáo viên | char (20) | Khoá ngoại |

Bảng biểu ‑: Mô tả bảng teach

1. Thiết kế giao diện

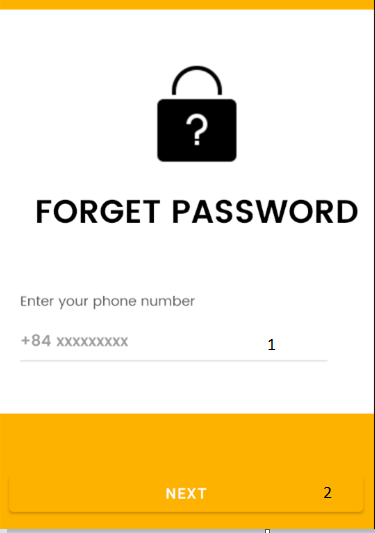
* Giao diện Login:



Hình ‑: Màn hình Đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | edtUsername | EditText | Dùng để nhập Username |
| 2 | edtpassword | EditText | Dùng để nhập Password |
| 3 | txtFPass | TextView | Khi người dùng quên mật khẩu sẽ nhấn vào để chạy tới layout quên mật khẩu |
| 4 | imgViewLogin | ImageView | Dùng để thực hiện đăng nhập |

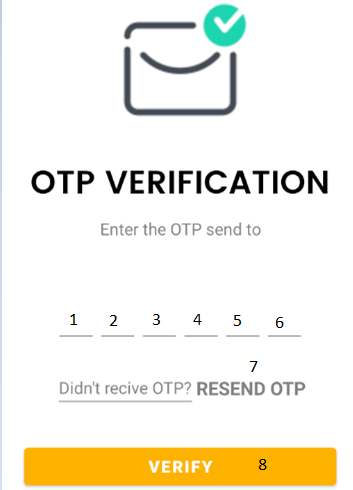
Bảng biểu ‑: Thiết kế Login



Hình ‑: Màn hình quên mật khẩu

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | phoneEt | EditText | Dùng để nhập số điện thoại |
| 2 | ForgotPass\_btn | Button | Dùng để xác nhận số điện thoại, nếu số điện thoại tồn tại sẽ cho qua form xác nhận |

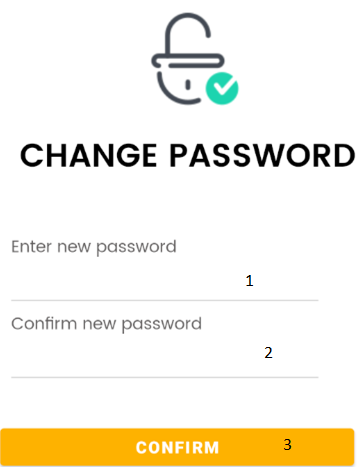
Bảng biểu ‑: Thiết kế ForgetPass



Hình ‑: Màn hình xác nhận mã OTP

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | inputCode1 | EditText | Dùng để nhập một giá trị Code OTP |
| 2 | inputCode2 | EditText | Dùng để nhập một giá trị Code OTP |
| 3 | inputCode3 | EditText | Dùng để nhập một giá trị Code OTP |
| 4 | inputCode4 | EditText | Dùng để nhập một giá trị Code OTP |
| 5 | inputCode5 | EditText | Dùng để nhập một giá trị Code OTP |
| 6 | inputCode6 | EditText | Dùng để nhập một giá trị Code OTP |
| 7 | tvResend | TextView | Dùng để gửi lại mã OTP |
| 8 | btnVerifyOTP | Button | Dùng để xác nhận mã OTP tiến tới layout thay đổi mật khẩu |

Bảng biểu ‑: Thiết kế Verify OTP

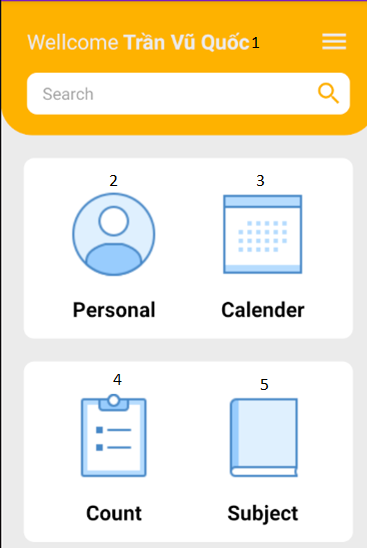


Hình ‑: Màn hình đổi mật khẩu

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | txtNewPass | EditText | Dùng để nhập mật khẩu mới |
| 2 | txtConfirmPass | EditText | Dùng để xác nhận lại mật khẩu |
| 3 | btnChangePass | Button | Nhấp vào để xác nhận thay đổi mật khẩu, nếu thành công sẽ về trang login |

Bảng biểu ‑: Thiết kế Change Pass

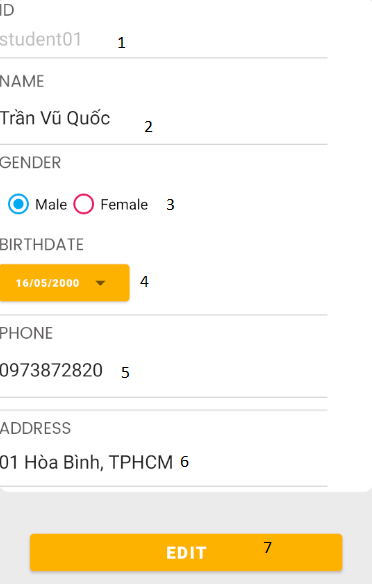
* Giao diện sinh viên:



Hình ‑: Màn hình chính của sinh viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | StudentName | TextView | Hiển thị tên người đăng nhập |
| 2 | Personal | ImageView | Nhấp vào để sang layout thông tin cá nhân |
| 3 | Calender | ImageView | Nhấp vào để sang layout thông tin lịch học |
| 4 | Count | ImageView | Nhấp vào để sang layout thông tin điểm số |
| 5 | Subject | ImageView | Nhấp vào để sang layout thông tin môn học |

Bảng biểu ‑: Thiết kế Màn hình chính của Sinh viên



Hình ‑: Màn hình thông tin sinh viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | Id\_student | EditText | Hiển thị mã sinh viên |
| 2 | name | EditText | Hiển thị tên sinh viên |
| 3 | rdbMale-rdbFemale | RadioButton | Hiển thị giới tính sinh viên |
| 4 | btnBirthday | Button | Hiển thị ngày tháng năm sinh |
| 5 | Phone | EditText | Hiển thị số điện thoại sinh viên |
| 6 | address | EditText | Hiển thị địa chỉ sinh viên |
| 7 | btnedit | Button | Nhấp vào để sang trang sửa thông tin sinh viên |

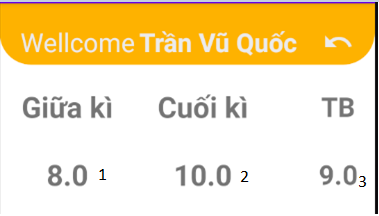
Bảng biểu ‑: Thiết kế Personal Sinh viên



Hình ‑: Màn hình môn học của sinh viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | txtSubjectName | TextView | Hiển thị tên môn của sinh viên đang theo học |
| 2 | txtNoteSubject | TextView | Hiển thị mã lớp học của sinh viên |

Bảng biểu ‑: Thiết kế Hiển thị môn học của sinh viên

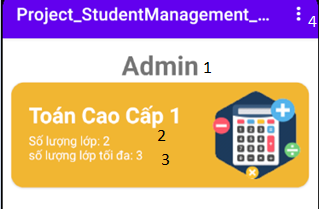


Hình ‑: Màn hình điểm của sinh viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | dbMid | TextView | Hiển thị điểm giữa kì của sinh viên |
| 2 | dbEnd | TextView | Hiển thị điểm cuối kì của sinh viên |
| 3 | dbAvg | TextView | Hiển thị điểm trung bình kì của sinh viên |

Bảng biểu ‑: Thiết kế hiển thị điểm của sinh viên

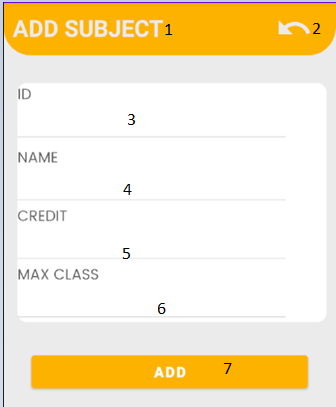
* Giao diện giảng viên:



Hình ‑: Màn hình trang chủ admin

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | txtAdminHomeTittle | TextView | Hiển thị tên Admin đăng nhập |
| 2 | txtSubjectName | TextView | Hiển thị tên môn |
| 3 | txtNoteSubject | TextView | Hiển thị thông tin chi tiết của môn |
| 4 | AdminHomePageMenu | Menu | Xuất ra đường dẫn đến các chức năng của Admin |

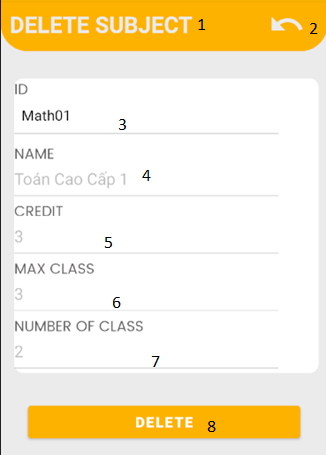
Bảng biểu ‑: Màn hình chính của Admin



Hình ‑: Màn hình thêm môn học của giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | layoutHeader | TextView | Hiển thị tên layout |
| 2 | imgReturn | ImageView | Trở về Layout màn hình của giáo viên |
| 3 | editIdSubject | EditText | Nhập mã môn học |
| 4 | edtNameSubject | EditText | Nhập tên môn học |
| 5 | edtCreditsSubject | EditText | Nhập số tín chỉ |
| 6 | edtMaxClassSubject | EditText | Nhập số lớp tối đa |
| 7 | btnAddSubject | Button | Ấn vào button sẽ thêm dữ liệu vào Cơ sở dữ liệu |

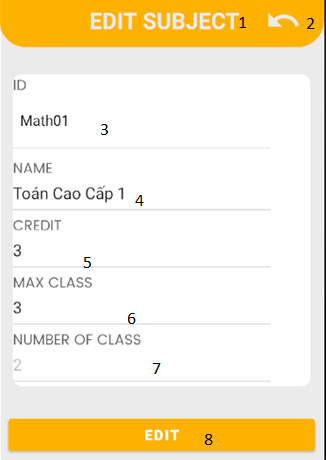
Bảng biểu ‑: Thiết kế Thêm môn học của giáo viên



Hình ‑: Màn hình xoá môn của giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | layoutHeader | TextView | Hiển thị tên Layout |
| 2 | imgReturn | ImageView | Trở về Layout màn hình của giáo viên |
| 3 | editIdSubject | EditText | Hiển thị mã lớp |
| 4 | edtNameSubject | EditText | Hiển thị tên lớp |
| 5 | edtCreditsSubject | EditText | Hiển thị tín chỉ |
| 6 | edtMaxClassSubject | EditText | Hiển thị Tối đa lớp |
| 7 | edtNumberOfClass | EditText | Hiển thị Số lượng lớp hiện có |
| 8 | btnAddSubject | Button | Ấn vào button sẽ xóa dữ liệu vào cơ sở dữ liệu |

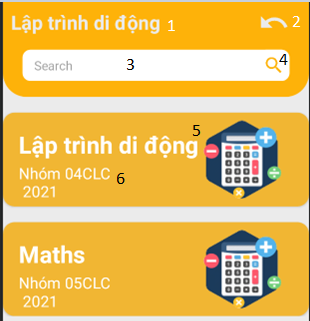
Bảng biểu ‑: Thiết kế Xoá môn học của giáo viên



Hình ‑: Màn hình sửa môn của giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | layoutHeader | TextView | Hiển thị tên Layout |
| 2 | imgReturn | ImageView | Trở về Layout màn hình của giáo viên |
| 3 | editIdSubject | EditText | Nhập mã môn học |
| 4 | edtNameSubject | EditText | Nhập tên môn học |
| 5 | edtCreditsSubject | EditText | Nhập số tín chỉ |
| 6 | edtMaxClassSubject | EditText | Nhập Số lớp tối đa |
| 7 | edtNumberOfClass | EditText | Nhập Số lượng lớp hiện có |
| 8 | btnAddSubject | Button | Ấn vào button sẽ cập nhật dữ liệu vào DB |

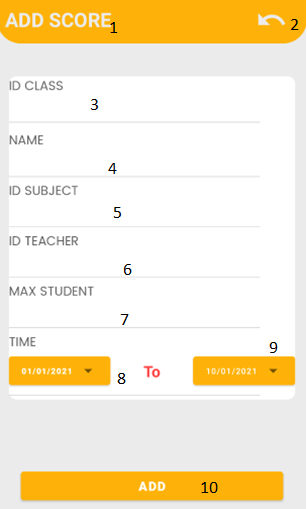
Bảng biểu ‑: Thiết kế Sửa môn học của giáo viên



Hình ‑: Màn hình quản lý lớp của giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | tvSubjectName | TextView | Hiển thị tên môn học |
| 2 | imgReturn | ImageView | Trở về Layout màn hình của giáo viên |
| 3 | editTSearchClass | EditText | Nhập tên lớp cần tìm kiếm |
| 4 | imgSearchClass | imgReturn | Thực hiện việc tìm kiếm |
| 5 | txtClassName | TextView | Hiển thị tên lớp |
| 6 | txtNoteClass | TextView | Hiển thị thông tin lớp |

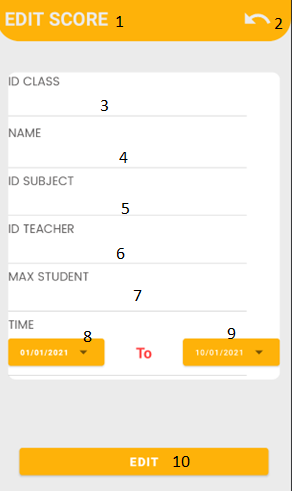
Bảng biểu ‑: Thiết kế quản lý lớp



Hình ‑: Màn hình thêm lớp học của giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | layoutHeader | TextView | Hiển thị tên Layout |
| 2 | imgReturn | ImageView | Trở về Layout màn hình của giáo viên |
| 3 | editIdClass | EditText | Nhập mã lớp học |
| 4 | edtName | EditText | Nhập tên lớp học |
| 5 | editsubject | EditText | Nhập mã môn học |
| 6 | editidteacher | EditText | Nhập mã giáo viên |
| 7 | edtMaxStudent | EditText | Nhập số lượng tối đa sinh viên |
| 8 | btntimestart | Button | Chọn ngày bắt đầu lớp học |
| 9 | btntimeend | Button | Chọn ngày kết thúc lớp học |
| 10 | btnAddScore | Button | Ấn vào button sẽ thêm dữ liệu vào cơ sở dữ liệu |

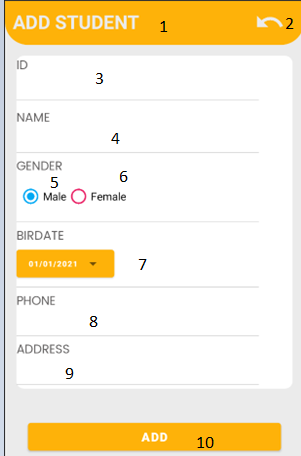
Bảng biểu ‑: Thiết kế Thêm lớp của giáo viên



Hình ‑: Màn hình sửa lớp học của giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | layoutHeader | TextView | Hiển thị tên Layout |
| 2 | imgReturn | ImageView | Trở về Layout màn hình của giáo viên |
| 3 | editIdClass | EditText | Nhập mã lớp |
| 4 | edtName | EditText | Hiển thị và nhập tên lớp |
| 5 | editsubject | EditText | Hiển thị và nhập mã môn học |
| 6 | editidteacher | EditText | Hiển thị và nhập mã giáo viên |
| 7 | edtMaxStudent | EditText | Hiển thị và nhập Số học sinh tối đa |
| 8 | btntimestart | Button | Hiển thị và chọn ngày bắt đầu lớp học |
| 9 | btntimeend | Button | Hiển thị và chọn ngày kết thúc lớp học |
| 10 | btnEditScore | Button | Ấn vào button sẽ cập nhật dữ liệu vào cơ sở dữ liệu |

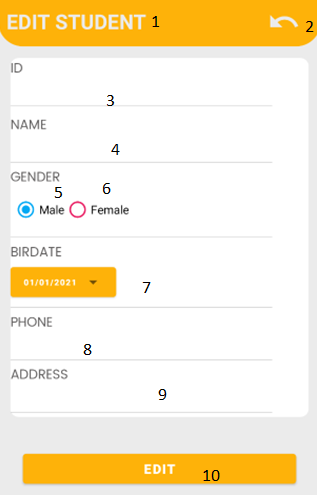
Bảng biểu ‑: Thiết kế Sửa lớp của giáo viên



Hình ‑: Màn hình thêm sinh viên của giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | layoutHeader | TextView | Hiển thị tên Layout |
| 2 | imgReturn | ImageView | Trở về Layout màn hình của giáo viên |
| 3 | editIdStudent | EditText | Nhập mã của sinh viên |
| 4 | edtName | EditText | Nhập tên sinh viên |
| 5 | radiobuttonmale | Radiobutton | Chọn giới tính nam |
| 6 | radiobuttonfemale | Radiobutton | Chọn giới tính nữ |
| 7 | btntime | Button | Chọn ngày sinh của sinh viên |
| 7 | editphone | EditText | Nhập số điện thoại sinh viên |
| 9 | editAdrress | EditText | Nhập địa chỉ của sinh viên |
| 10 | btnAddStudent | Button | Ấn vào button sẽ thêm dữ liệu vào cơ sở dữ liệu |

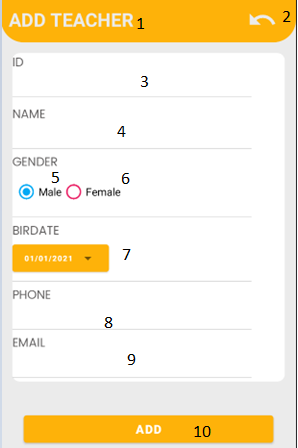
Bảng biểu ‑: Thiết kế Thêm sinh viên của Giáo viên



Hình ‑: Màn hình sửa thông tin sinh viên của giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | layoutHeader | TextView | Hiển thị tên Layout |
| 2 | imgReturn | ImageView | Trở về Layout màn hình của giáo viên |
| 3 | editIdStudent | EditText | Nhập mã của sinh viên |
| 4 | edtName | EditText | Hiển thị và nhập tên sinh viên |
| 5 | radiobuttonmale | Radiobutton | Chọn giới tính nam |
| 6 | radiobuttonfemale | Radiobutton | Chọn giới tính nữ |
| 7 | btntime | Button | Chọn ngày sinh của sinh viên |
| 7 | editphone | EditText | Hiển thị và nhập số điện thoại sinh viên |
| 9 | editAdrress | EditText | Hiển thị và nhập địa chỉ của sinh viên |
| 10 | btnEditStudent | Button | Ấn vào button sẽ cập nhật dữ liệu vào cơ sở dữ liệu |

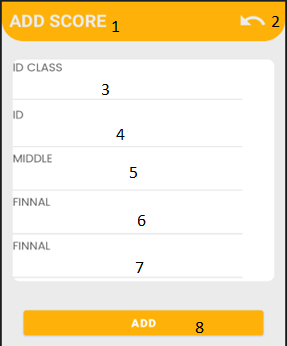
Bảng biểu ‑: Thiết kế sửa thông tin sinh viên của giáo viên



Hình ‑: Màn hình thêm giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | layoutHeader | TextView | Hiển thị tên Layout |
| 2 | imgReturn | ImageView | Trở về Layout màn hình của giáo viên |
| 3 | editIdTeacher | EditText | Nhập mã của giáo viên |
| 4 | edtName | EditText | Nhập tên giáo viên |
| 5 | radiobuttonmale | Radiobutton | Chọn giới tính nam |
| 6 | radiobuttonfemale | Radiobutton | Chọn giới tính nữ |
| 7 | btntime | Button | Chọn ngày sinh của giáo viên |
| 7 | editphone | EditText | Nhập số điện thoại giáo viên |
| 9 | editAdrress | EditText | Nhập địa chỉ của giáo viên |
| 10 | btnAddTeacher | Button | Ấn vào button sẽ thêm dữ liệu vào cơ sở dữ liệu |

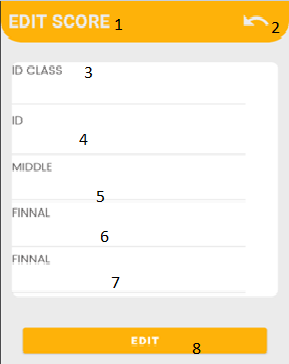
Bảng biểu ‑: Thiết kế Thêm giáo viên



Hình ‑: Màn hình nhập điểm của giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | layoutHeader | TextView | Hiển thị tên form |
| 2 | imgReturn | ImageView | Về form chính của teacher |
| 3 | editIdClass | EditText | Nhập ID lớp học |
| 4 | editIdStudent | EditText | Nhập id của sinh viên |
| 5 | editmiddlescore | EditText | Nhập middlescore |
| 6 | editFinnal | EditText | Nhập điểm cuối kì |
| 7 | editAverage | EditText | Hiển thị điểm trung bình |
| 8 | btnAddScore | Button | Ấn vào button sẽ thêm dữ liệu vào cơ sở dữ liệu |

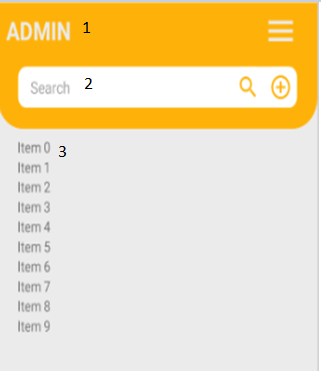
Bảng biểu ‑: Thiết kế nhập điểm của giáo viên



Hình ‑: Màn hình sửa điểm của giáo viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Đối tượng | Kiểu dữ liệu | Ghi chú |
| 1 | layoutHeader | TextView | Hiển thị tên form |
| 2 | imgReturn | ImageView | Về form chính của teacher |
| 3 | editIdClass | EditText | Nhập ID lớp học |
| 4 | editIdStudent | EditText | Nhập id của sinh viên |
| 5 | editmiddlescore | EditText | Hiển thị và nhập điểm giữa kì |
| 6 | editFinnal | EditText | Hiển thị và nhập điểm giữa kì |
| 7 | editAverage | EditText | Hiển thị và nhập điểm giữa kì |
| 8 | btnEditScore | Button | Ấn vào button sẽ cập nhật dữ liệu vào cơ sở dữ liệu |

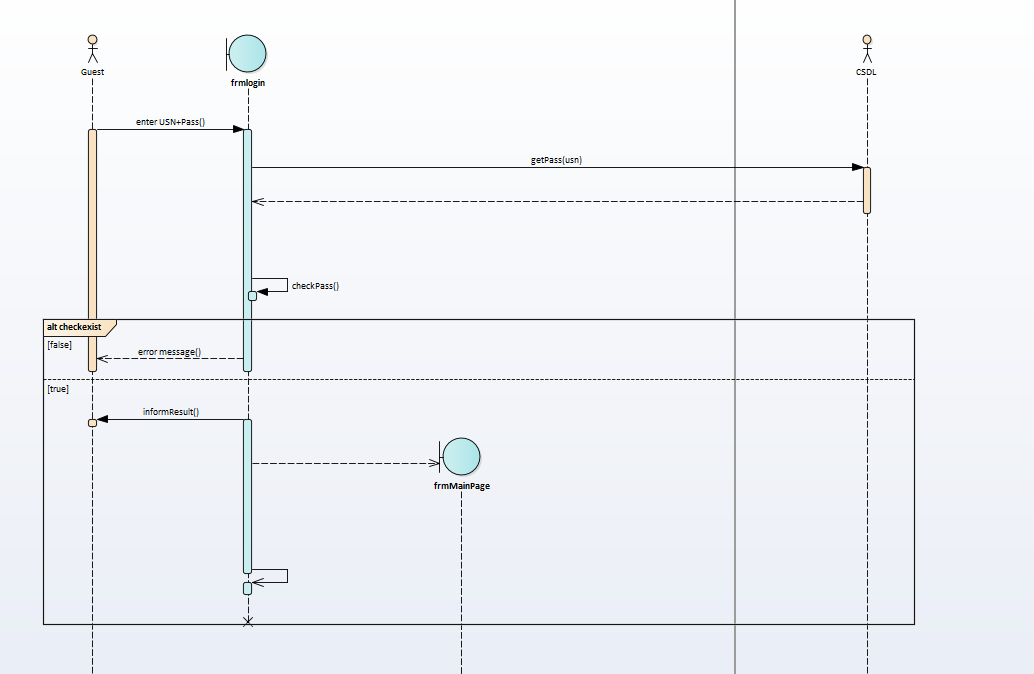
Bảng biểu ‑: Thiết kế sửa điểm của giáo viên



Hình ‑: Màn hình quản lý tài khoản của giáo viên

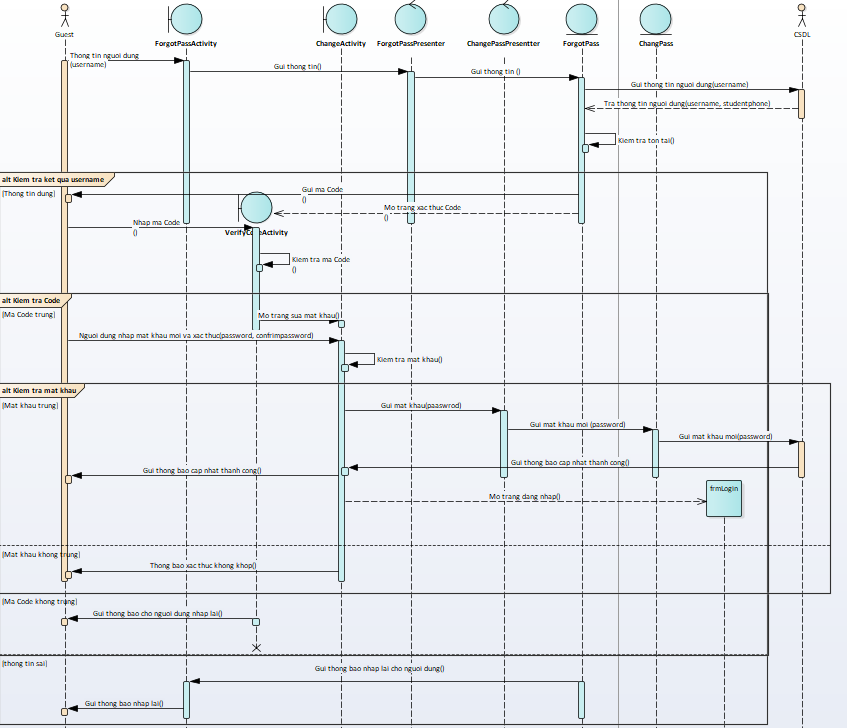
1. Thiết kế xử lý
   1. *Sequence Diagram*

* Login



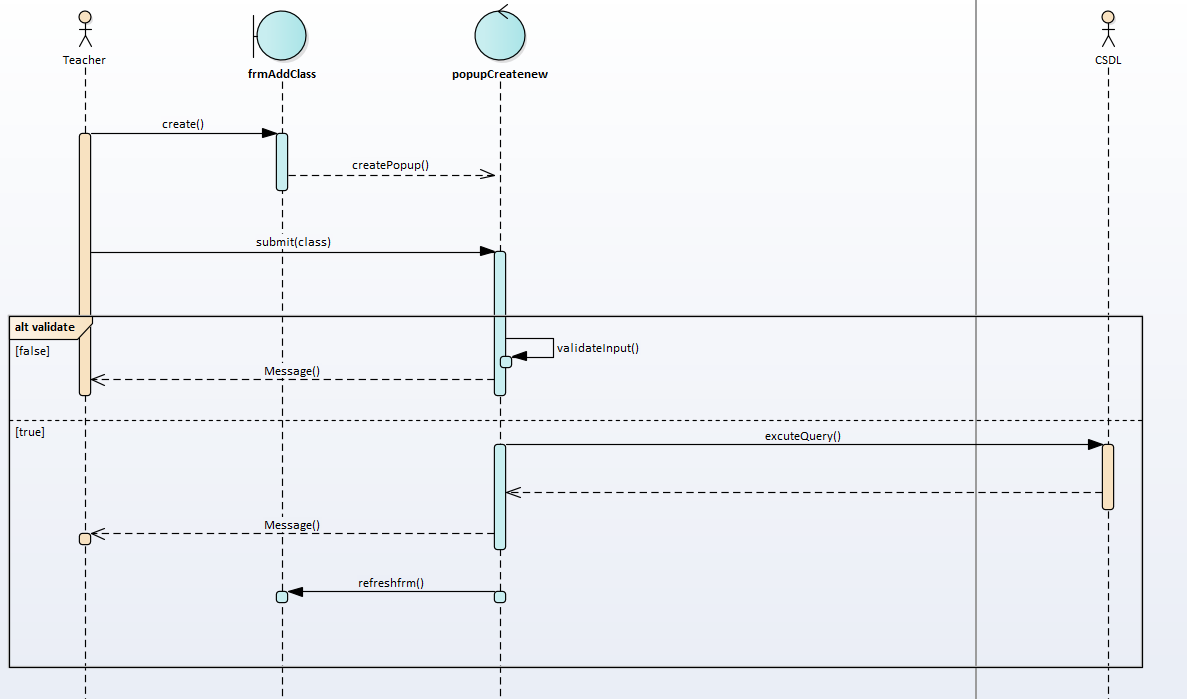
Hình ‑: Sequence Diagram của Login

* Forgetpass

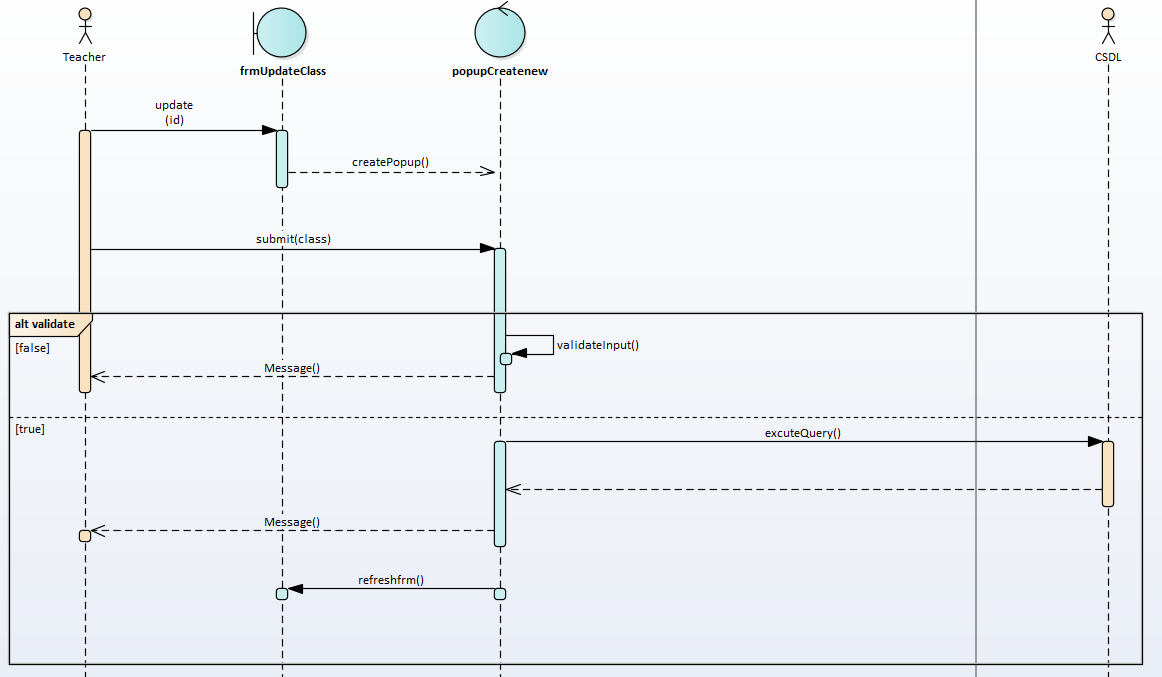


Hình ‑: Sequence Diagram của Forgetpass

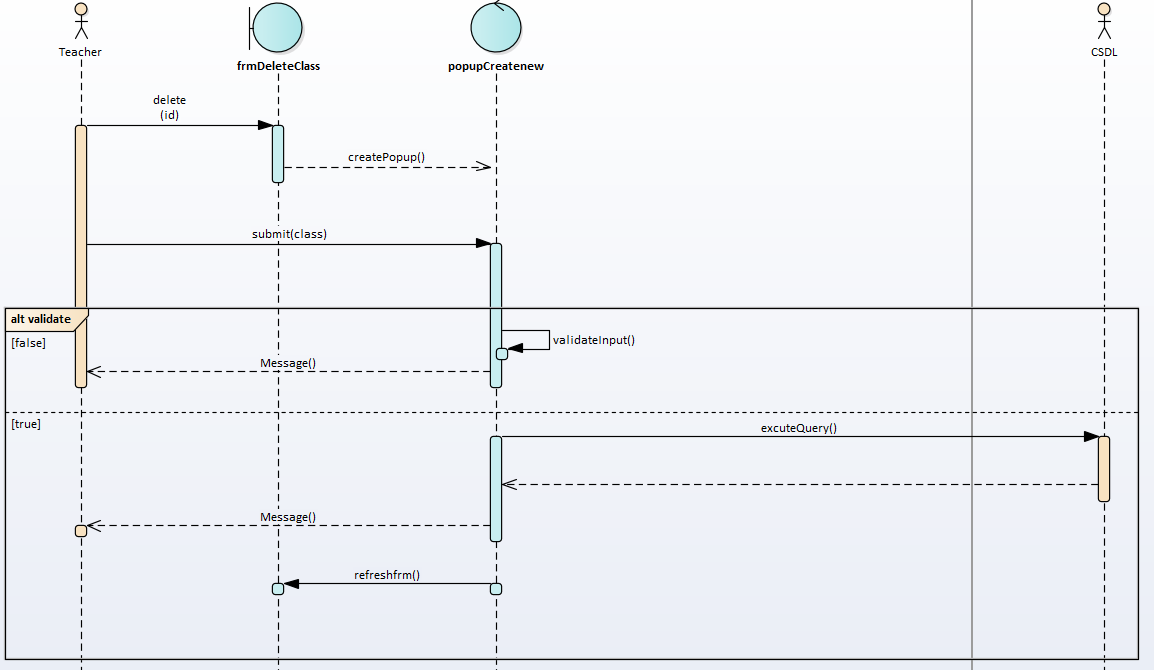
* Add, edit, delete class(teacher)



Hình ‑: : Sequence Diagram của Add Class

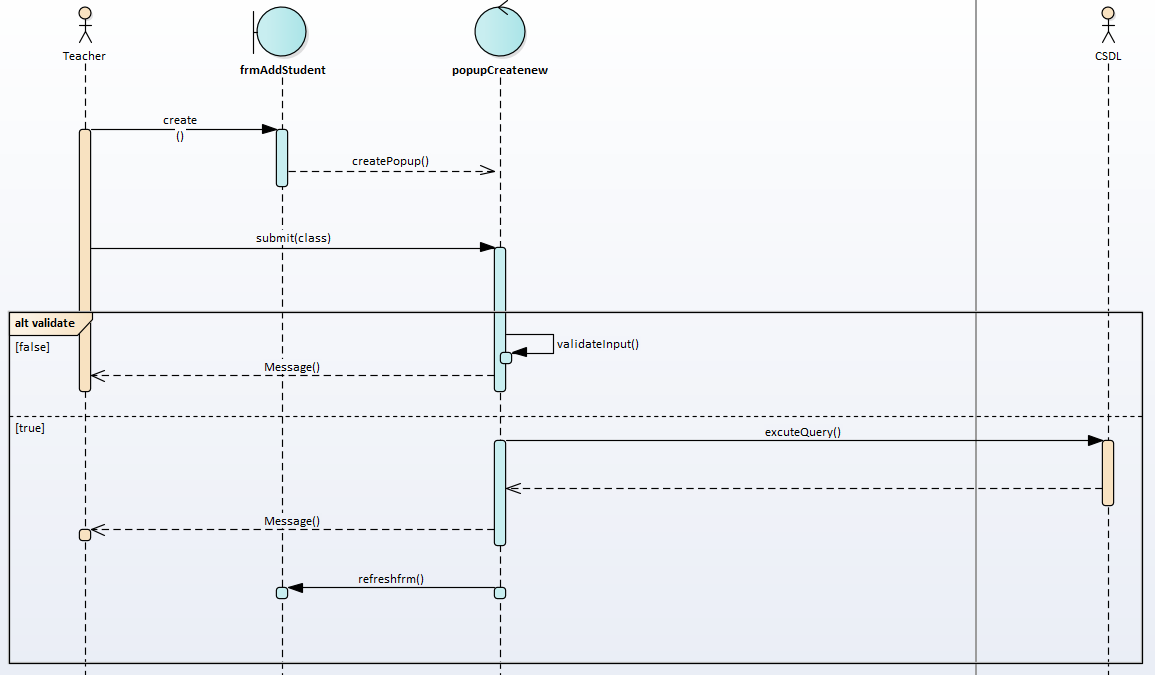


Hình ‑: : Sequence Diagram của Edit Class

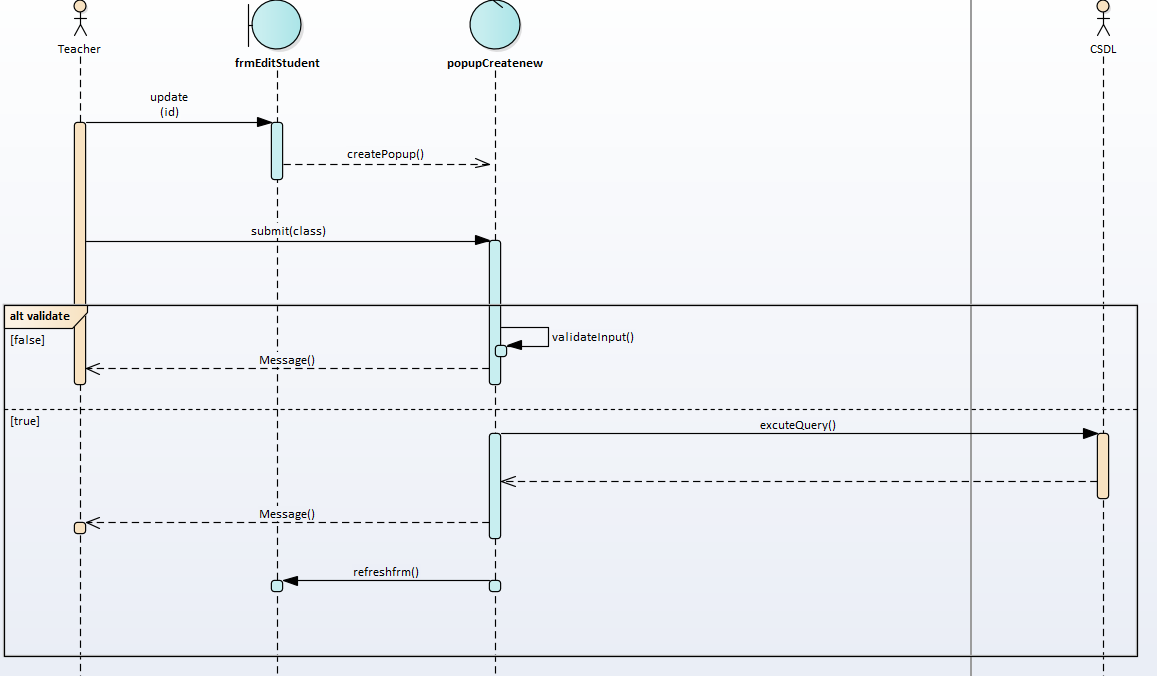


Hình ‑: Sequence Diagram của Delete Class

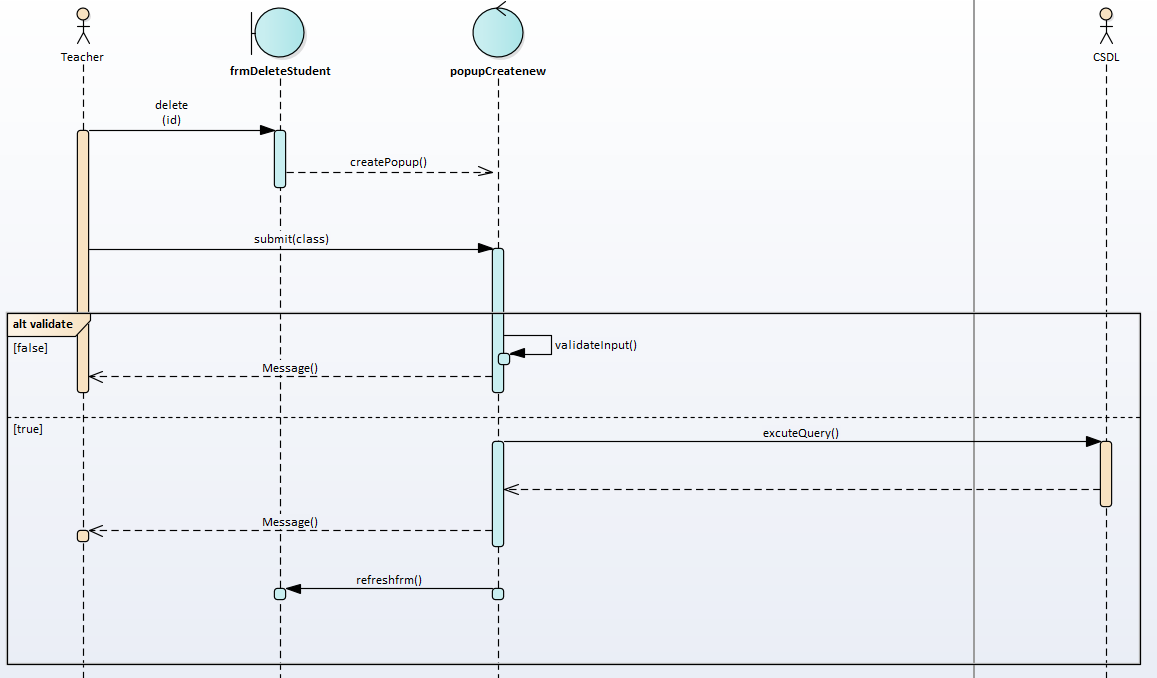
* Add,edit,delete student(teacher)



Hình ‑: Sequence Diagram của Add Student

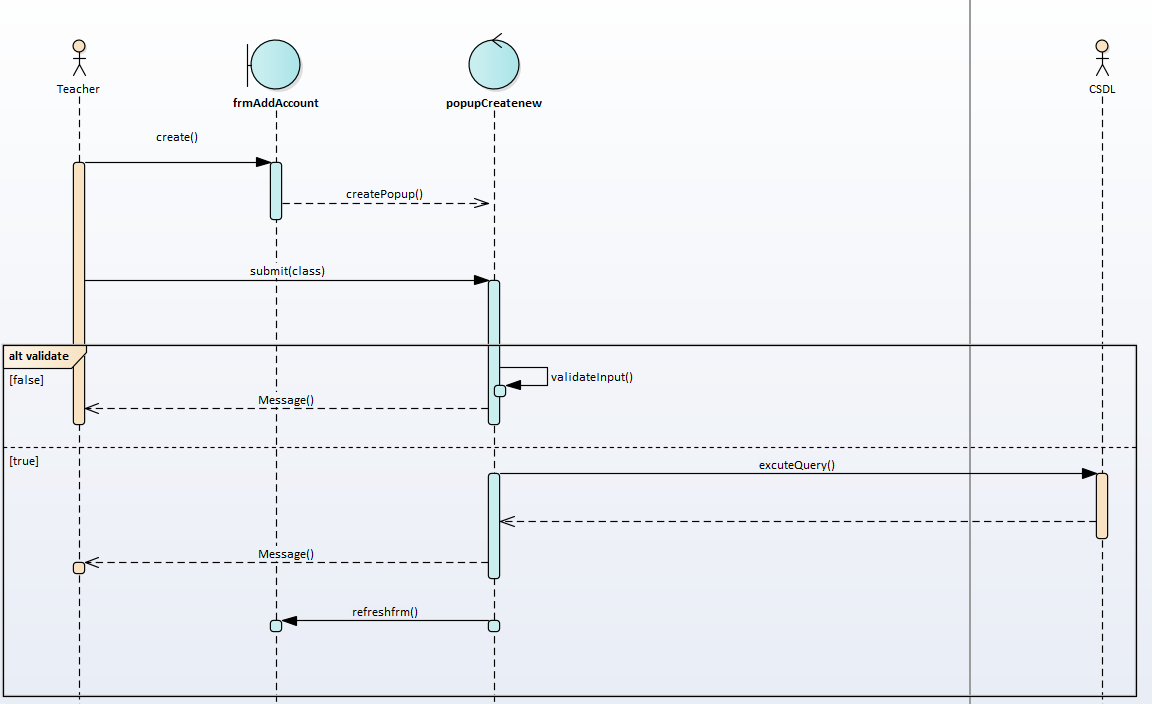


Hình ‑: Sequence Diagram của Edit Student

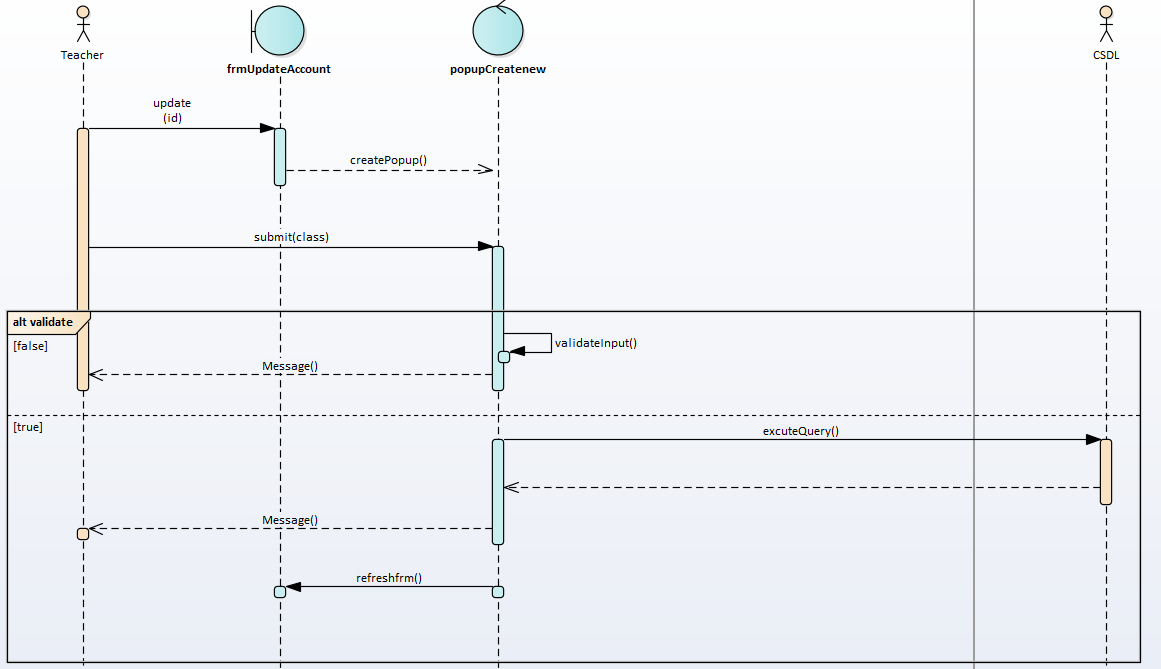


Hình ‑: Sequence Diagram của Delete Student

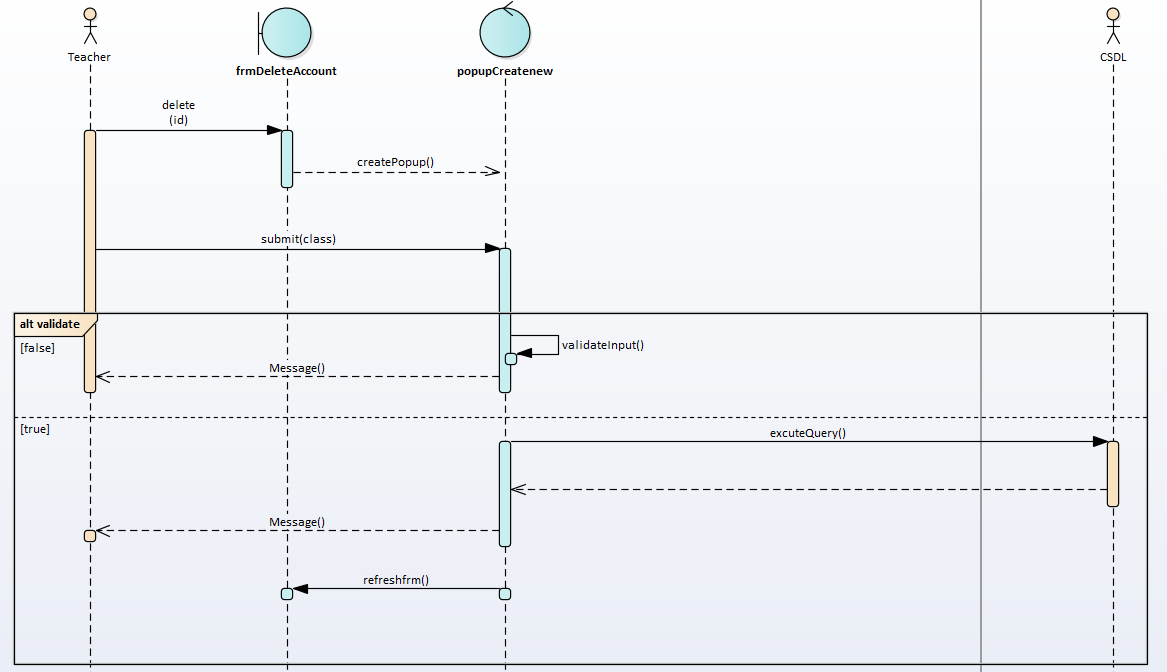
* Add,edit,delete account( teacher)



Hình ‑: Sequence Diagram của Add Account

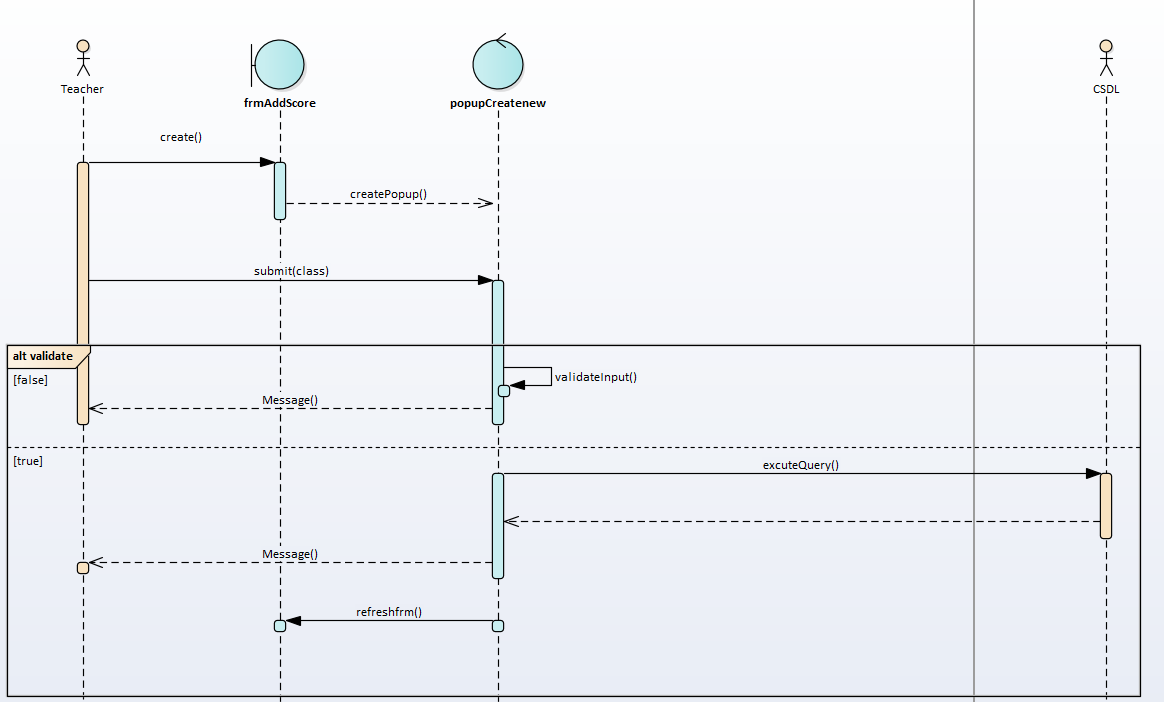


Hình ‑: Sequence Diagram của Edit Account

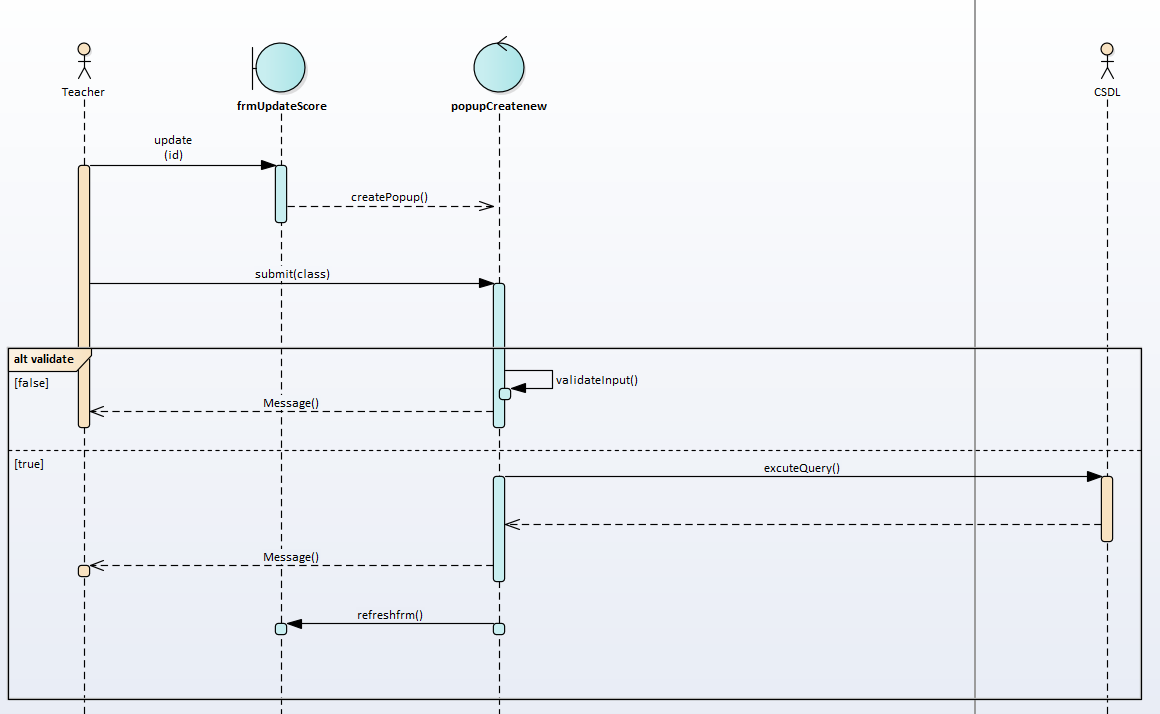


Hình ‑: Sequence Diagram của Delete Account

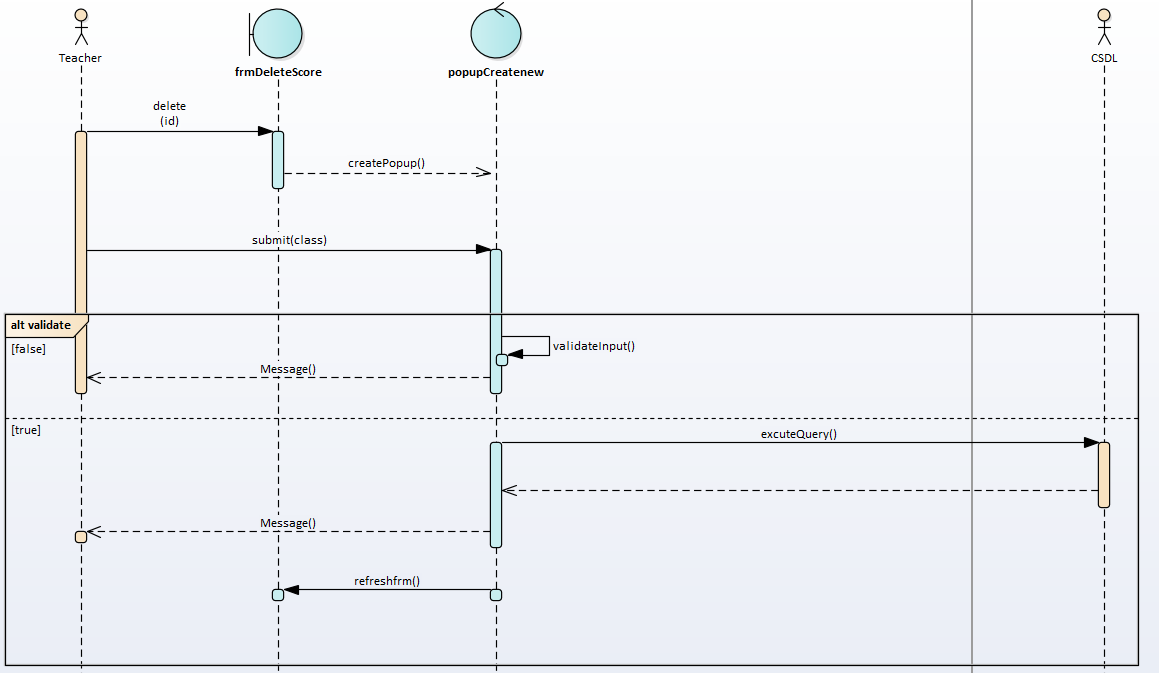
* Add, delete, edit score(teacher)



Hình ‑: Sequence Diagram của Add Score

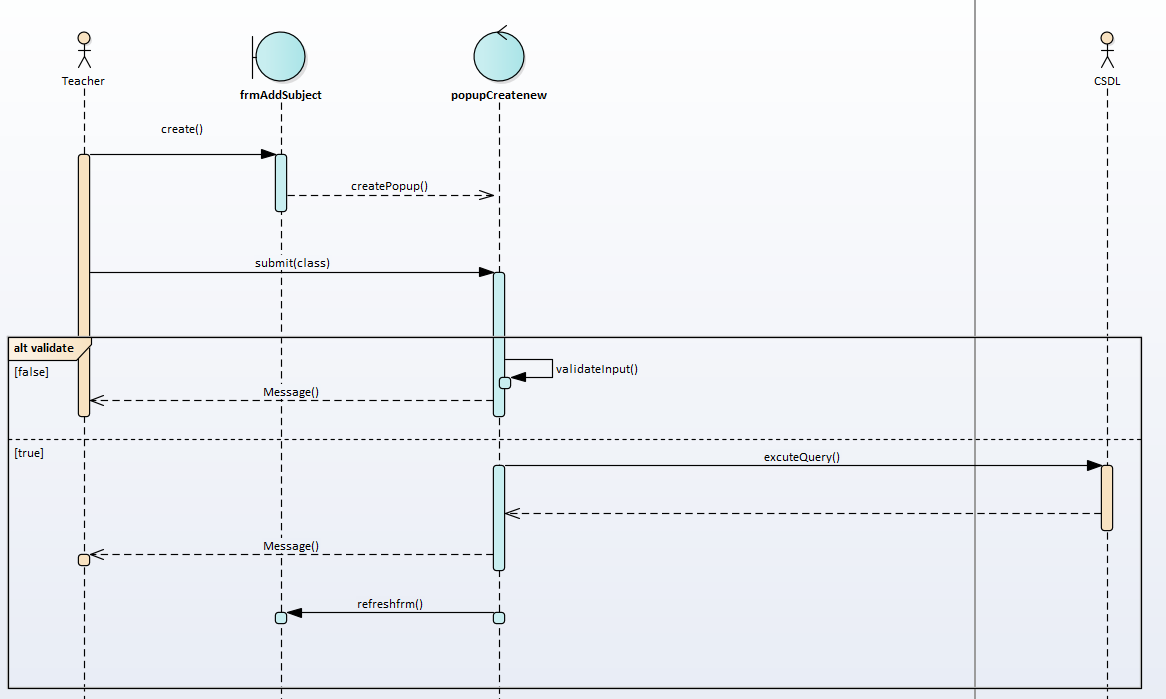


Hình ‑: Sequence Diagram của Edit Score

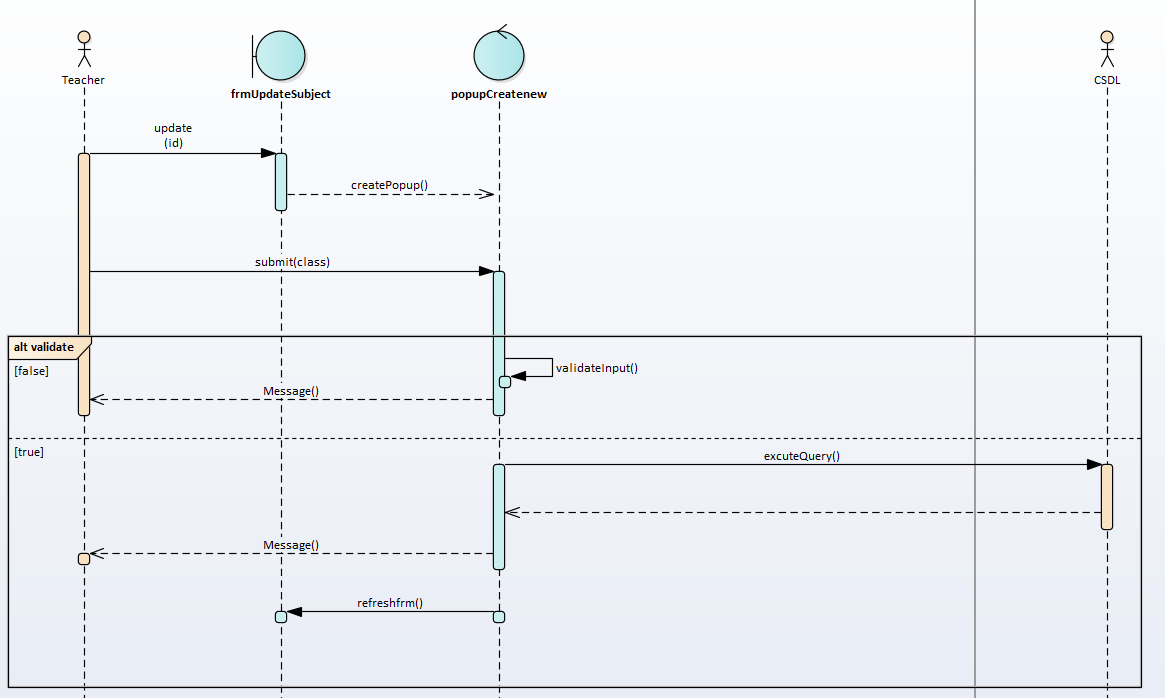


Hình ‑: Sequence Diagram của Delete Score

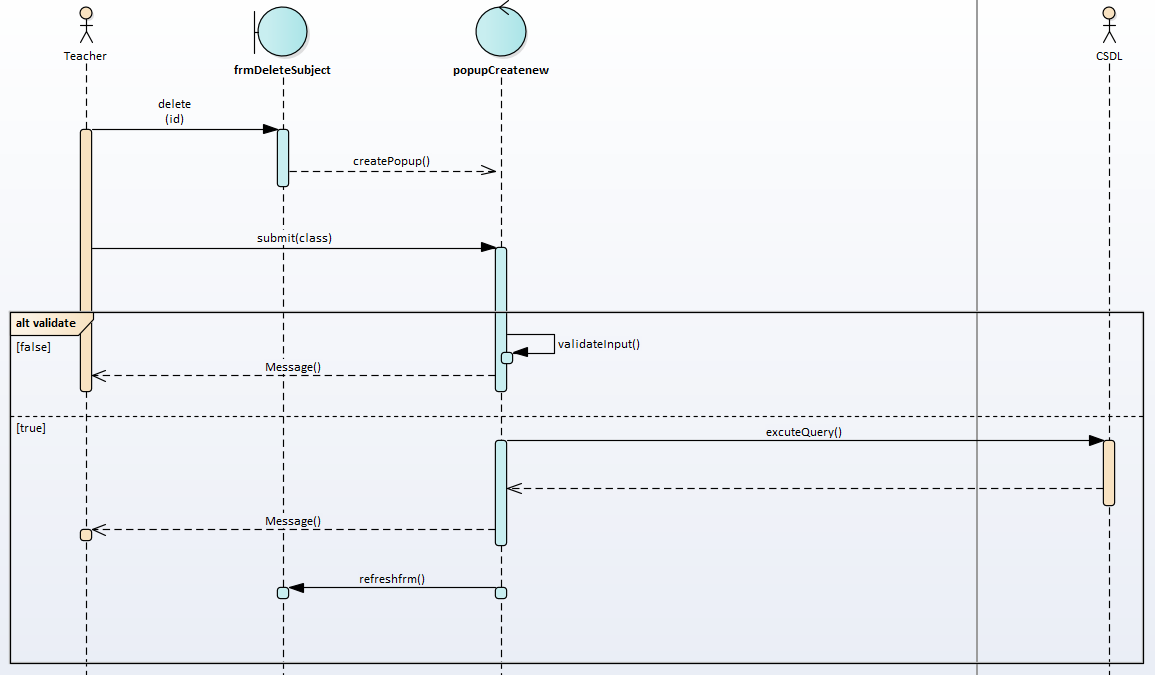
* Add, delete, edit subject(teacher)



Hình ‑: Sequence Diagram của Add Subject

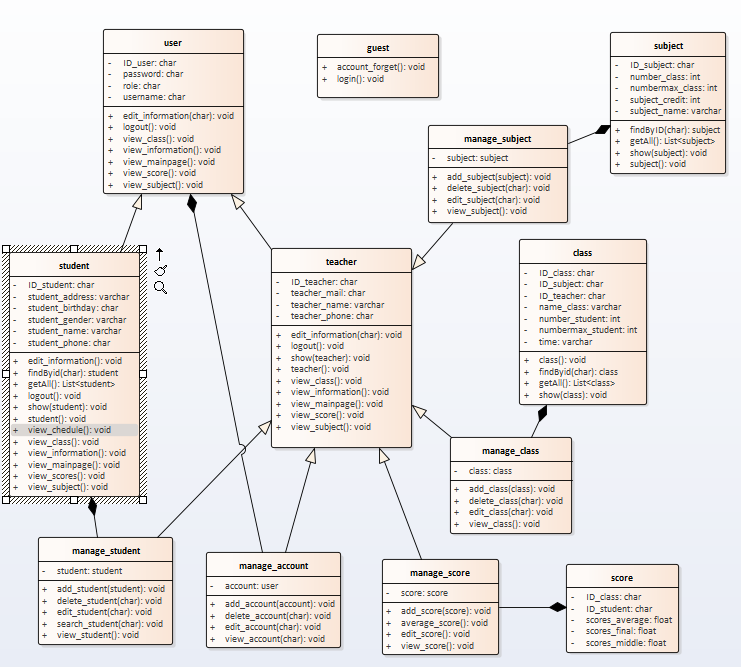


Hình ‑: Sequence Diagram của Edit Subject



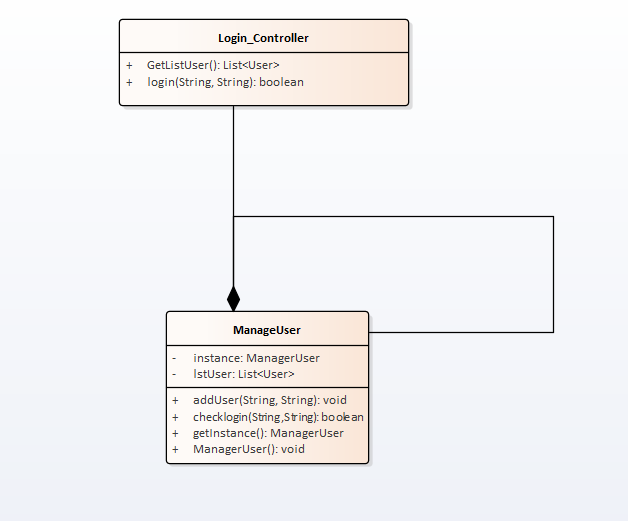
Hình ‑: Sequence Diagram của Delete Subject

* 1. *Class Diagram*
* Class Diagram



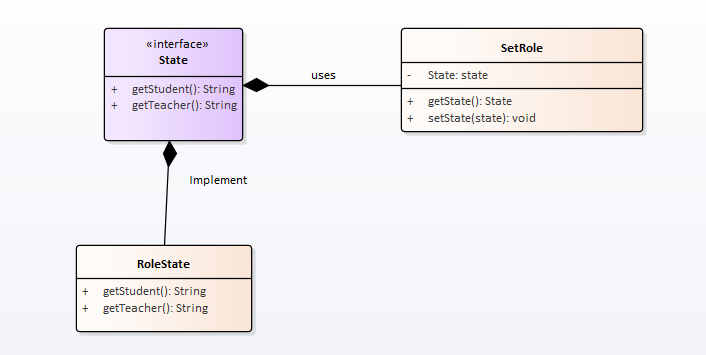
Hình ‑: Class Diagram

* Class Diagram của Singleton Pattern



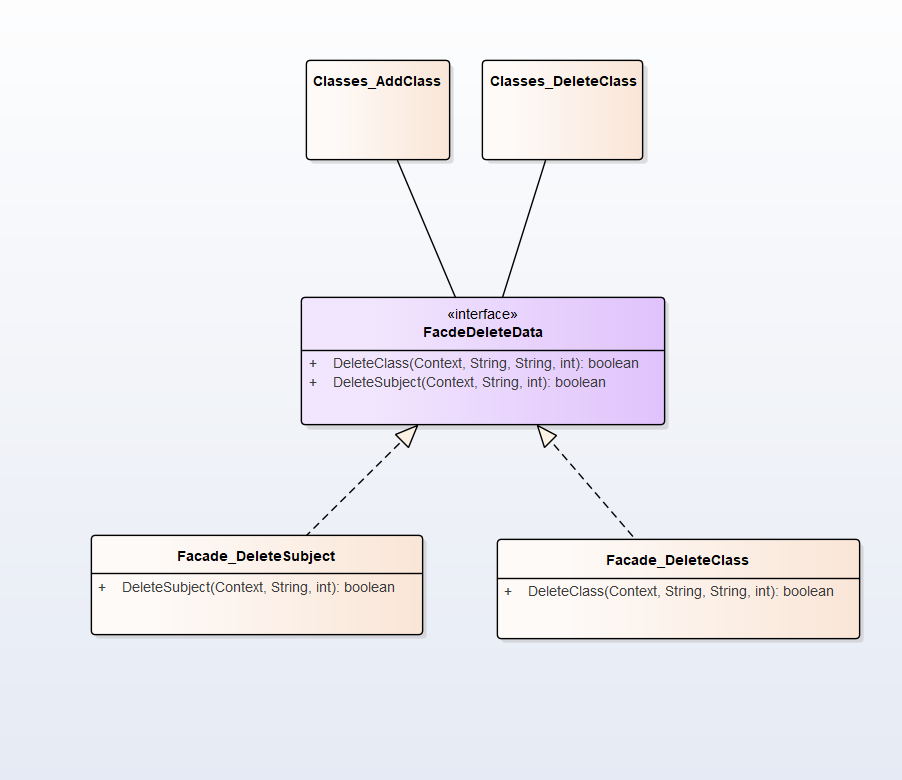
Hình ‑: Class Diagram của Singleton Pattern

* Class Diagram của State Pattern



Hình ‑: Class Diagram của State Pattern

* Class Diagram của Façade Pattern

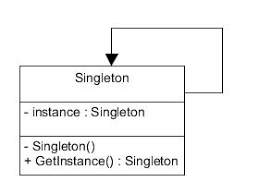


Hình ‑: Class Diagram của Façade Pattern

# ÁP DỤNG CÁC MẪU THIẾT KẾ

1. Singleton Pattern
   1. *Định nghĩa*

Singleton là một trong những design pattern của nhóm Creational Design Pattern. Mẫu thiết kế Singleton Pattern được dùng để đảm bảo chỉ duy nhất một instance trong một class, và class đó sẽ cung cấp phương thức toàn cục để truy cập đến thực thể đó. Singleton Pattern được dùng khi bạn tạo ra một class mà bạn chỉ muốn chỉ có duy nhất một thực thể là instance của class đó và bạn có thể truy cập đến nó ở bất kỳ nơi đâu khi bạn muốn.



Hình ‑: Singleton Pattern

Ưu điểm cơ bản:

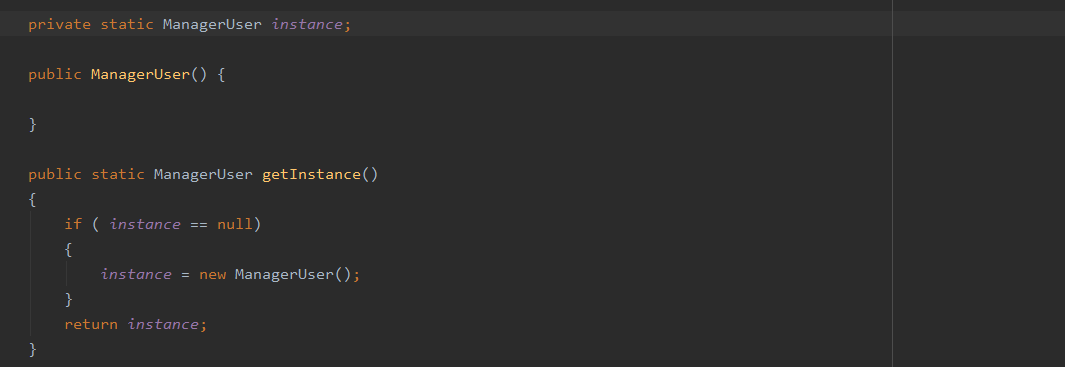
* Chỉ tồn tại duy nhất 1 Instance của class Singleton được tạo ra trong suốt chương trình ( chỉ có duy nhất một instance trong một class.)
* Ẩn constructor của class. ( Các constructor ẩn đảm bảo rằng các singleton class không bao giờ có thể được instantiated từ bên ngoài )
* Ý tưởng chính trong mô hình này là làm cho lớp đó chịu trách nhiệm kiểm soát sự khởi tạo của nó (nó chỉ được khởi tạo một lần).
* Cung cấp việc truy cập vào Object global : Loại bỏ các biến ko cần thiết, chỉ quản lý các biến ứng dụng cần dùng tới; Singleton sẻ chỉ sử dụng một số ít tài nguyên, còn biến toàn cầu thì thường được tạo từ nhều ngôn ngữ / kiểu dữ liệu phức tạp nên tiêu tốn tài nguyên hơn

Nhược điểm cơ bản :

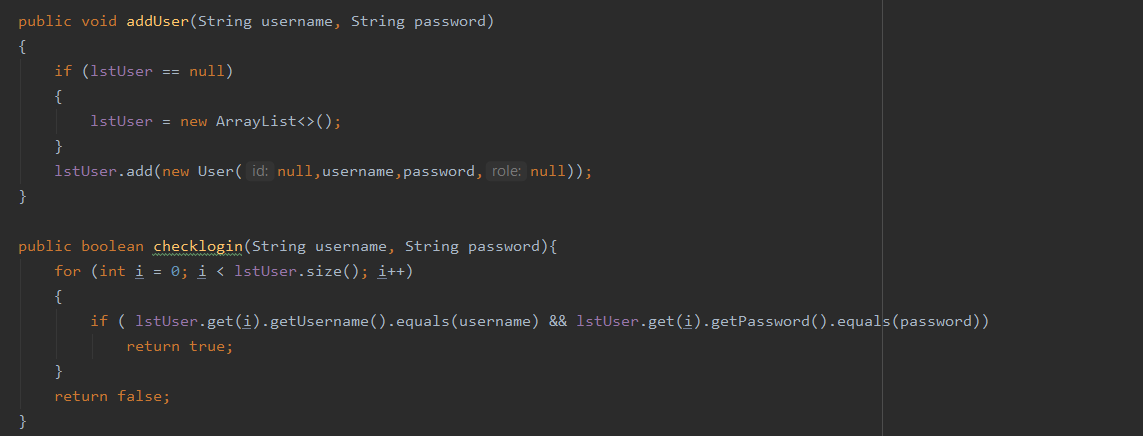
* Khó triển khai 1 cách hiệu quả để đảm bảo rằng chỉ 1 class chỉ có 1 đối tượng.
* Khó khăn trong việc kiểm soát sự hiện thân của nó.
* Hạn chế số lượng các instance của một class.
* Khi chỉ cho phép một trường hợp hoặc một số trường hợp cụ thể của một lớp. Các đối tượng Facade thường là những Singleton bởi vì ứng dụng luôn chỉ cần một đối tượng Façade
  1. *Lý do sử dụng*

Singleton giải quyết vấn đề làm sao để đảm bảo rằng 1 class chỉ có 1 instance duy nhất và nó luôn sẵn sàng để sử dụng ở bất kỳ thời điểm hoặc vị trí nào trong phần mềm của chúng ta.

* 1. *Áp dụng trong ứng dụng*
* Sử dụng trong chức năng login vào phần mềm:



* Tạo ra danh sách và kiểm tra đăng nhập:

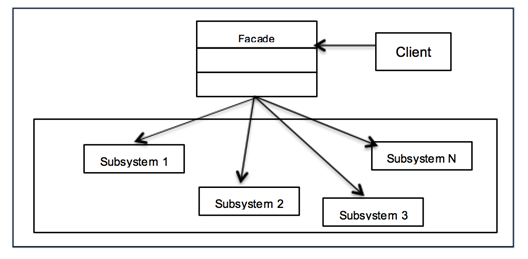


* Gọi hàm singleton để thêm dữ liệu vào danh sách và tiến thành kiểm tra. Nếu login thành công trả về true và ngược lại sẽ trả về false



1. Façade Pattern
   1. *Định nghĩa*

Facade Pattern là một trong những Pattern thuộc nhóm Structural Pattern. Mẫu thiết kế này cung cấp một giao diện chung đơn giản thay cho một nhóm các giao diện có trong một hệ thống phức tạp. Hệ thống phức tạp này có thể là 1 thư viện (library), 1 framework, hay tập hợp các class phức tạp. Facade Pattern định nghĩa một giao diện ở một cấp độ cao hơn để giúp cho người dùng có thể dễ dàng sử dụng hệ thống này. Mục đích của Facade Pattern là giấu sự phức tạp của hệ thống con đằng sau.



Hình ‑: Façade Pattern

Ưu điểm cơ bản :

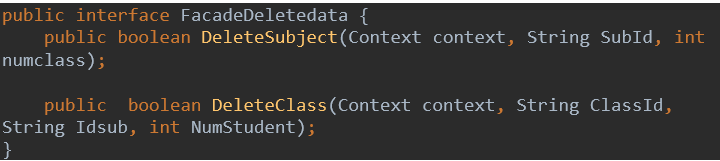
* Ta có thể tách mã nguồn của mình ra khỏi sự phức tạp của hệ thống con
* Hệ thống tích hợp thông qua Facade sẽ đơn giản hơn vì chỉ cần tương tác với Facade thay vì hàng loạt đối tượng khác.
* Có thể đóng gói nhiều được thiết kế không tốt bằng 1 hàm có thiết kế tốt hơn

Nhược điểm cơ bản: Class Facade của bạn có thể trở lên quá lớn, làm quá nhiều nhiệm vụ với nhiều hàm chức năng trong nó. Điều này sẽ dễ bị phá vỡ các quy tắc trong SOLID – nguyên lý lập trình hướng đối tượng.

* 1. *Lý do sử dụng*

Nhằm che giấu đi sự phức tạp của hệ thống xử lý logic bên dưới, giúp người dùng dễ sử dụng hệ thống bằng cách chỉ gọi tên hàm để sử dụng mà không cần quan tâm đến phần xử lý bên trong. Làm cho code súc tích, ngắn gọn và tránh lặp code cũng như dễ sửa chữa, mở rộng khi cần thiết. Khi phần xử lý của một tác vụ nào đó phức tạp và được sử dụng nhiều lần trong phần mềm.

* 1. *Áp dụng trong ứng dụng*
* Tạo Interface đặt tên FacadeDeletedata chứa các phương thức xóa lớp và môn



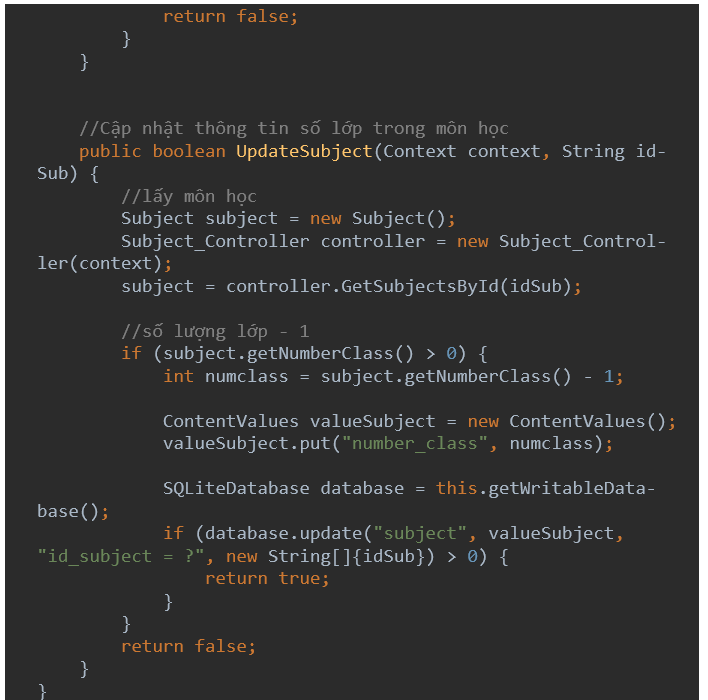
* Triển khai các phương thức ở các lớp con.
* Triển khai xóa môn ở lớp con Façade\_DeleteSubject



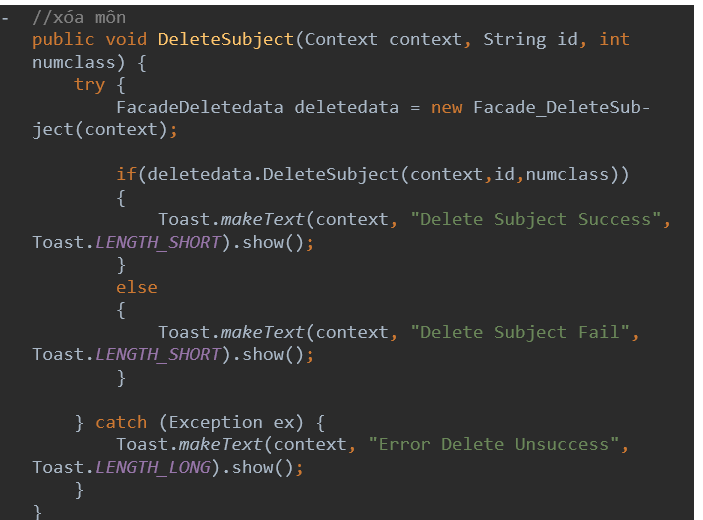


* Triển khai xóa lớp học ở class Facde\_DeleteClass.

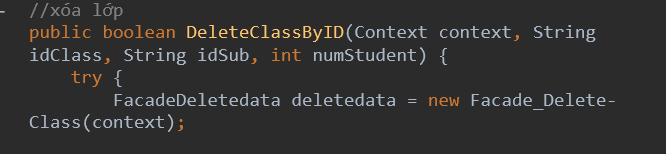


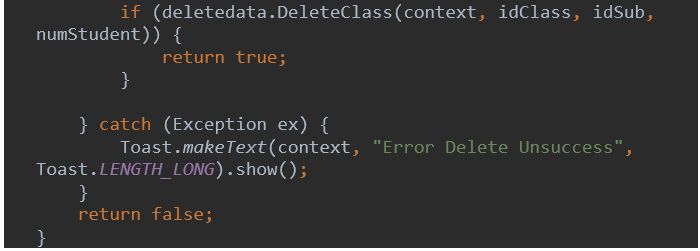


* Ta lấy ra sử dụng ở controller của chương trình
* Xóa môn được gọi ra ở Subject\_controller



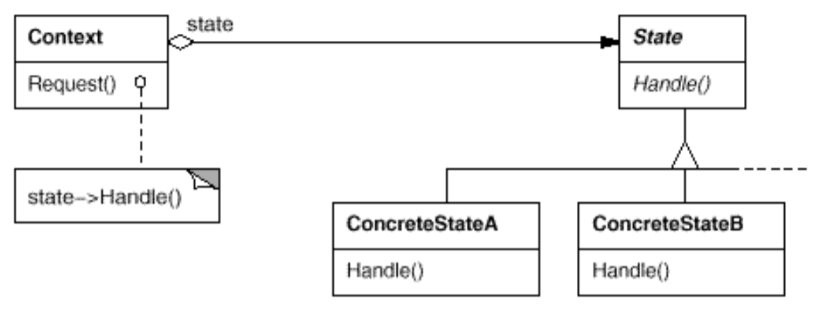
* Xóa lớp gọi ra ở controller Classes\_Controller





1. State Pattern
   1. *Định nghĩa*

State Pattern là một trong những mẫu thiết kế thuộc nhóm behavioral cho phép một object có thể biến đổi hành vi của nó khi có những sự thay đổi trạng thái nội bộ. Mẫu thiết kế này có thể được hiểu gần giống như Strategy, nó có thể chuyển đổi các chiến lược thông qua các phương thức được định nghĩa trong interface.



Hình ‑: State Pattern

Ưu điểm cơ bản :

* Mỗi trạng thái của đối tượng sẽ được lưu trong một lớp state riêng nên khi thêm một state mới chỉ cần thêm 1 lớp đối tượng mới
* Các state hoạt động độc lập nên dễ dàng sửa chữa, bảo trì
* Mỗi State bao gồm những thuộc tính và hành vi của đối tượng. Các state khác nhau sẽ có từng hành động khác nhau nên đối tượng sẽ phản ứng lại tác động bên ngoài khác nhau tuỳ trạng thái
* Hệ thống tích hợp thông qua Facade sẽ đơn giản hơn vì chỉ cần tương tác với Facade thay vì hàng loạt đối tượng khác.
* Có thể đóng gói nhiều được thiết kế không tốt bằng 1 hàm có thiết kế tốt hơn

Nhược điểm cơ bản: Vì mỗi lớp state mang đậm tính riêng biệt mà mỗi state thuộc về nên không thể tái dùng trạng thái. Chương trình có thể chậm một chút.

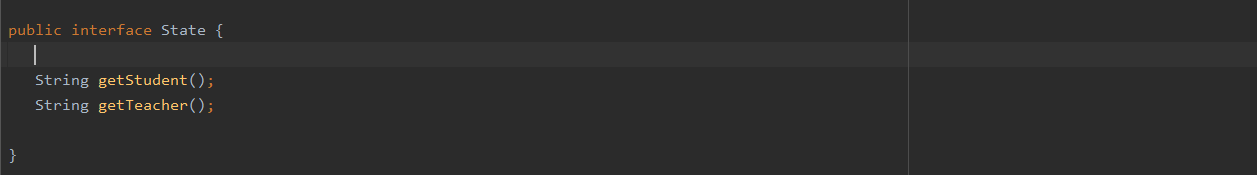
* 1. *Lý do sử dụng*

Singleton cho phép chúng ta trích xuất logic để phân tách các lớp và để đối tượng context của chúng ta ủy thác hành vi cho các phương thức được triển khai trong lớp trạng thái. Bên cạnh đó, chúng ta có thể tận dụng sự chuyển đổi giữa các trạng thái, trong đó một trạng thái có thể thay đổi trạng thái của context.

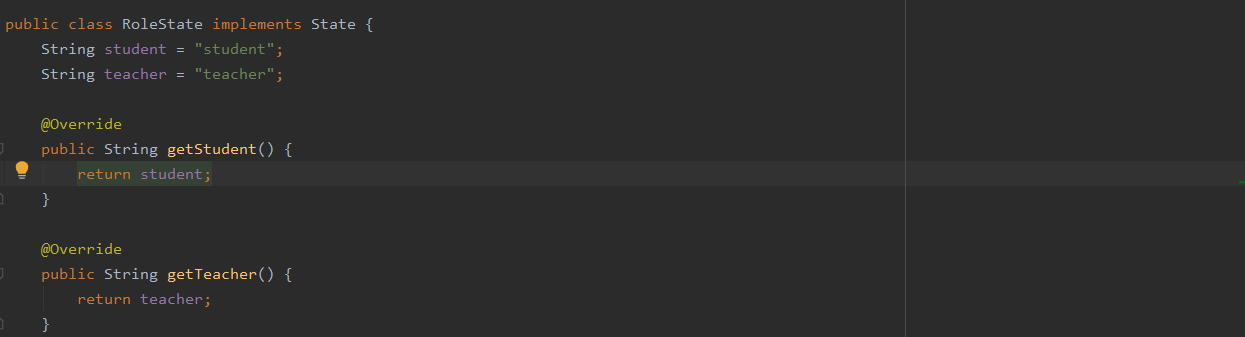
* 1. *Áp dụng trong ứng dụng*

State Design Pattern sử dụng trong chức năng xét quyền truy cập ứng dụng. Vì hành vi của đối tượng phụ thuộc vào trạng thái đăng nhập và có khả năng thay đổi giữa “student”, “teacher” trong mỗi lần run-time.

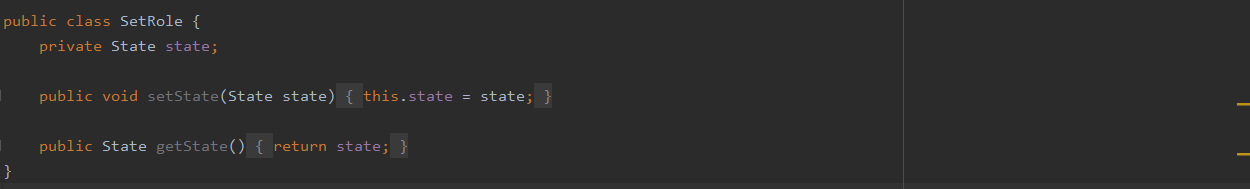
* State.java



* RoleState.java



* SetRole.java



# KẾT QUẢ

1. Ưu điểm phần mềm

* Chương trình chạy tương thích trên hệ Android.
* Đáp ứng được những nhu cầu cơ bản của việc quản lý chương trình như thêm, xoá, sửa, tìm kiếm, hiển thị.
* Phân quyền hệ thống cho giảng viên và sinh viên
* Phần mềm tự thiết kế.
* Thiết kế bắt mắt dễ sử dụng
* Giao diện trực quan, sinh động.
* Xử lý dữ liệu nhanh.

1. Nhược điểm phần mềm

* Một số chức năng chưa thực sự hoàn thiện
* Đa phần là chức năng cơ bản
* Dữ liệu chưa thực sự bảo mật
* Yêu cầu kết nối mạng ổn định
* Sử dụng trong môi trường đại học kín.

1. Khó khăn gặp phải

Hiện nay đối với tình hình dịch phức tạp, một vài thành viên nhóm lại nằm trong diện phải đi cách ly nên quá trình thực hiện xây dựng ứng dụng khá khó. Thực tế thì điều này ảnh hưởng tới tất cả mọi người làm cho quá trình học tập bị gián đoạn giai đoạn then chốt nên không tránh khỏi sai sót. Tuy nhiên do thực hiện đồ án cũng từ giai đoạn trước dịch một thời gian nên nhóm không quá khó khăn trong thời điểm nhiều môn cùng kết thúc

Quan điểm thành viên nhóm khó thống nhất do một vài thành viên có những ý kiến thực hiện riêng không thông qua nhóm trưởng. Tuy nhiên vấn đề đã được giải quyết xong.

1. Hướng phát triển đề tài

Chương tình có thể mở rộng độ hiệu quả bằng việc quản lý sang những thông tin lớn hơn như học phí, chuyên ngành… Ngoài ra có thể chia thêm phần quyền cho phụ huynh, cựu học sinh truy cập. Cải thiện thêm về UI/ UX, tăng cường bảo mật dữ liệu bằng việc sử dụng host server trên mạng. Tăng lượng truy cập người dùng và mở rộng dần đối tượng và phạm vi. Có thể cập nhật sự kiện từ trang chủ trường, cho phép tham dự hoặc đăng ký sự kiện dễ dàng hơn.

1. Phân công công việc.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Võ Phạm Hoàng Quân** | **Trần**  **Vũ Quốc** | **Trần Liễu Nhựt Anh** | **Hoàng Ngọc Doanh** |
| 1 | Ứng dụng Design Pattern | 30% | 50% | 10% | 10% |
| 2 | Thiết kế Class Diagram | 20% | 10% | 60% | 10% |
| 3 | Thiết kế Usecase diagram và mô tả chi tiết Usecase | 40% | 10% | 10% | 40% |
| 4 | Thiết kế Sequence diagram |  |  | 100% |  |
| 5 | Thiết kế cơ sở dữ liệu | 60% |  |  | 40% |
| 6 | Thiết kế giao diện | 25% | 25% | 25% | 25% |
| 7 | Tổng hợp bài |  |  |  | 100% |
| 8 | Thực hiện báo cáo | 10% | 10% | 30% | 50% |

Bảng biểu ‑: Bảng phân công công việc

# LỜI KẾT

Với việc áp dụng được những công nghệ của thời kì hiện nay, chúng ta có thể quản lý mọi thứ trên máy tính thông qua những chương trình được tạo ra, giải quyết những vấn đề mà giấy tờ không thể quán xuyến một cách nhanh gọn và hiệu quả bằng. Đây là lý do nhóm thực hiện đề tài “Xây dựng ứng dụng quản lý sinh viên”. Sau thời gian tìm hiểu và làm việc cùng nhau, nhóm đã rút ra nhiều kinh nghiệm khi áp dụng kiến thức và thông tin môn học để giải quyết vấn đề thực tế, từ đó đúc kết ra một số bài học cho chặng đường còn dài phía trước

Những lợi thế nhóm tác giả có cũng giúp cho công việc tiến triển tốt đẹp, nhưng những hạn chế tay nghề và kiến thức khiến việc thực hiện không tránh khỏi sai sót. Vấn đề thời gian chưa được phân bố ổn định để thực hiện nhiều đề tài cùng lúc. Những chức năng tạo dựng đa phần cơ bản. Nhưng nhờ sự giúp đỡ từ nhà trường, giáo viên hướng dẫn và bạn bè xung quanh đã tạo động lực lớn để nhóm tác giả có thể thực hiện hoàn thành tốt đề tài này. Nhóm tác giả sẵn sàng đón nhận những ý kiến đóng góp dù nhỏ nhất để cải thiện đồ án được phát triển hơn.

*Xin gửi lời cảm ơn chân thành nhất đến những người luôn hỗ trợ nhóm tác giả!*

# TÀI LIỆU THAM KHẢO

1. [Top 7 Phần mềm quản lý học sinh - sinh viên tốt nhất hiện nay (e-pro.vn)](http://e-pro.vn/top-7-phan-mem-quan-ly-hoc-sinh-hoc/)*.*
2. Tài liệu thiết kế phần mềm hướng đối tượng – Ths. Nguyễn Trần Thi Văn
3. [Singleton Design Pattern - Chuyên Nghiệp Hơn Trong Lập Trình (codelearn.io)](https://codelearn.io/sharing/singleton-design-pattern-developer?fbclid=IwAR0k3hKoZ376e6AP8FKFwF8k9WJmHTI-shJ7S_oGd5GZ8_FsK2b7mGv1qs0)
4. [Design Pattern: Singleton pattern (viblo.asia)](https://viblo.asia/p/design-pattern-singleton-pattern-4dbZNgn8lYM)
5. [[Phần 3] Facade pattern (viblo.asia)](https://viblo.asia/p/phan-3-facade-pattern-57rVRqXJv4bP?fbclid=IwAR0JGv6mWF-JiRr8v_mFMzd_xAXaXvObNpVboPdtUh0LmcKR6r4XGCgB0Zw)
6. [Facade Pattern là gì? Hướng dẫn Facade sử dụng trong PHP (codetutam.com)](http://codetutam.com/facade-pattern-la-gi/)
7. [State Design Pattern (viblo.asia)](https://viblo.asia/p/state-design-pattern-07LKXjPDlV4)
8. [(ADO.NET) SqlCommand truy vấn và cập nhật dữ liệu C# SQL Server (xuanthulab.net)](https://xuanthulab.net/ado-net-sqlcommand-truy-van-va-cap-nhat-du-lieu-c-sql-server.html?fbclid=IwAR1Qqti24fbWgRDSqTZqPEZtQcW7VaR5tYFqgvO01wMBxPlvdwKQQGLOgeQ)
9. [Login & Register Android App Using Firebase | Android Studio Authentication Tutorials | Part 1/4 - YouTube](https://www.youtube.com/watch?v=tbh9YaWPKKs)
10. [Login and Register Form using SQLite Database in Android Studio | login registration android studio - YouTube](https://www.youtube.com/watch?v=8obgNNlj3Eo&t=532s)

**SOURCE CODE của chương trình :** [**TKPMHDT - Google Drive**](https://drive.google.com/drive/folders/1NDYegfNx7nObaoDWSid-A41_SYoqlom5?fbclid=IwAR2YJgVnF3Z97KfinhEP1iMcUi1spAaTJK7zhbzuWDI5Mt_z-0-V0if7PTQ)