

TRƯỜNG ĐẠI HỌC BÁCH KHOA THÀNH PHỐ HỒ CHÍ MINH
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



Hệ thống hỗ trợ học tập cho sinh viên

Phiên bản: 1.1.0.050419

GVHD: Nguyễn Hữu Vũ

Nhóm: 33

<i>SV:</i>	Trình Thị Thu Thảo	1613232
	Vũ Hoàng Văn	1614063
	Nguyễn Lê Quốc Việt	1614096
	Phạm Quang Nghĩa	1612221
	Cao Thanh Tùng	1613989

TP. HỒ CHÍ MINH, THÁNG 4/2019

Lịch sử thay đổi tài liệu

Ngày	Phiên bản	Thay đổi	Người thay đổi
8/3/2019	1.0.0.080319	Tạo mới đặc tả yêu cầu hệ thống	Vũ Hoàng Văn
17/3/2019	1.1.0.170319	Bổ sung mô tả chức năng quản lý thời khóa biểu Bổ sung mô tả chức năng quản lý lịch làm việc và deadlines Bổ sung mô tả chức năng quản lý công việc nhóm Bổ sung mô tả chức năng đăng nhập, đăng kí, quản lý thông tin cá nhân và thống kê cá nhân Bổ sung mô tả sơ bộ về cơ sở dữ liệu	Cao Thanh Tùng Nguyễn Lê Quốc Việt Phạm Quang Nghĩa Vũ Hoàng Văn Trình Thị Thu Thảo
23/3/2019	1.1.0.230319	Bổ sung User story và Mockup cho chức năng tạo lịch học trên trường. Bổ sung User Story và Mockup cho trang đăng ký đăng nhập, khung thông báo, khung và hộp tin nhắn. Bổ sung mô tả chức năng quản lý công việc nhóm Bổ sung User Story và Mockup cho trang chủ và thông tin cá nhân Bổ sung User Story và Mockup cho chức năng tạo những công việc cần làm	Cao Thanh Tùng Nguyễn Lê Quốc Việt Phạm Quang Nghĩa Vũ Hoàng Văn Trình Thị Thu Thảo
30/3/2019	1.1.0.300319	Bổ sung kiến trúc phần mềm cho chức năng quản lý lịch học trên trường. Bổ sung kiến trúc phần mềm cho trang đăng ký đăng nhập, khung thông báo, khung và hộp tin nhắn. Bổ sung kiến trúc phần mềm cho chức năng quản lý công việc nhóm Bổ sung kiến trúc phần mềm cho trang chủ và thông tin cá nhân Bổ sung kiến trúc phần mềm cho chức năng tạo những công việc cần làm	Cao Thanh Tùng Nguyễn Lê Quốc Việt Phạm Quang Nghĩa Vũ Hoàng Văn Trình Thị Thu Thảo
05/4/2019	1.1.0.050419	Bổ sung class diagram cho chức năng quản lý lịch học trên trường. Bổ sung class diagram và đặc tả phương thức cho trang đăng ký đăng nhập, khung thông báo, khung và hộp tin nhắn. Bổ sung class diagram và đặc tả phương thức cho chức năng quản lý công việc nhóm Bổ sung class diagram và đặc tả phương thức cho trang chủ và thông tin cá nhân Bổ sung class diagram và đặc tả phương thức cho chức năng tạo những công việc cần làm	Cao Thanh Tùng Nguyễn Lê Quốc Việt Phạm Quang Nghĩa Vũ Hoàng Văn Trình Thị Thu Thảo

Mục lục

1	Giới thiệu	11
2	Mục tiêu	11
3	Đặc tả chức năng	11
3.1	Nhóm chức năng quản lý thời khóa biểu	11
3.1.1	Mô tả chung	11
3.1.2	Use case diagram	12
3.1.3	Chi tiết chức năng	13
3.2	Nhóm chức năng quản lý công việc nhóm	16
3.2.1	Mô tả chung	16
3.2.2	Use case diagram	17
3.2.3	Chi tiết chức năng	18
3.3	Nhóm chức năng quản lý công việc cá nhân	19
3.3.1	Mô tả chung	19
3.3.2	Use case diagram	21
3.3.3	Chi tiết chức năng	24
3.4	Nhóm chức năng quản lý thông tin cá nhân	26
3.4.1	Mô tả chung	26
3.4.2	Use case diagram	26
3.4.3	Chi tiết chức năng	27
4	Thiết kế sơ bộ cơ sở dữ liệu	31
5	Các yêu cầu phi chức năng	32

6	User Story và Mockup	32
6.1	Trang đăng ký đăng nhập	32
6.1.1	User Story	32
6.1.2	Mockup	33
6.2	Trang chủ	34
6.2.1	User Story	34
6.2.2	Mockup	34
6.3	Khung thông báo	35
6.3.1	User Story	35
6.3.2	Mockup	36
6.4	Khung và hộp tin nhắn	37
6.4.1	User Story	37
6.4.2	Mockup	38
6.5	Khung thông tin cá nhân	39
6.5.1	User Story	39
6.5.2	Mockup	40
6.6	Trang lịch học	41
6.6.1	User Story	41
6.6.2	Mockup	41
6.7	Trang công việc cá nhân	45
6.7.1	User Story	45
6.7.2	Mockup	45
6.8	Trang Công việc nhóm	47
6.8.1	Trang project chính	47

6.8.2	Tạo project mới	48
6.8.3	Project detail	50
7	Thiết kế kiến trúc phần mềm cho hệ thống	51
7.1	Mô hình chung của tất cả các chức năng	51
7.2	Kiến trúc phần mềm cho chức năng quản lý lịch học trên trường	51
7.3	Kiến trúc phần mềm cho chức năng cho trang đăng ký đăng nhập, khung thông báo, khung và hộp tin nhắn	52
7.4	Kiến trúc phần mềm cho chức năng quản lý công việc nhóm	53
7.5	Kiến trúc phần mềm cho trang chủ và thông tin cá nhân	54
7.6	Kiến trúc phần mềm cho chức năng tạo những công việc cần làm	55
8	Class diagram cho hệ thống	56
8.1	Class diagram cho chức năng quản lý lịch học trên trường	56
8.1.1	Class diagram	56
8.1.2	Đặc tả các phương thức cho chức năng quản lý lịch học	57
8.1.2.a	Views	57
8.1.2.b	Controllers	57
8.1.2.c	Model	57
8.2	Class diagram cho chức năng cho trang đăng ký đăng nhập, khung thông báo, khung và hộp tin nhắn	58
8.2.1	Class diagram cho chức năng trang đăng ký đăng nhập	58
8.2.2	Class diagram cho chức năng khung thông báo	62
8.2.3	Class diagram cho chức năng khung và hộp tin nhắn	66
8.3	Class diagram và đặc tả phương thức cho chức năng quản lý công việc nhóm	75
8.4	Class diagram cho trang chủ và thông tin cá nhân	78

8.4.1	Class diagram	78
8.4.2	Đặc tả phương thức	78
8.5	Class diagram cho chức năng tạo những công việc cần làm	86
9	Kế hoạch phát triển hệ thống	92

Danh sách hình vẽ

3.1	Use case cho nhóm chức năng quản lý thời khóa biểu	12
3.2	Use case cho nhóm chức năng quản lý nhóm	17
3.3	Use case cho nhóm chức năng todo-list	21
3.4	Use case cho nhóm chức năng quản lý deadlines	22
3.5	Use case cho nhóm chức năng quản lý thông tin cá nhân	26
4.6	Thiết kế cơ sở dữ liệu sơ bộ	31
6.7	Mockup đăng ký đăng nhập	33
6.8	Mockup trang chủ của hệ thống	34
6.9	Mockup khung thông báo	36
6.10	Mockup khung và hộp tin nhắn	38
6.11	Mockup khung thông tin cá nhân	40
6.12	Mockup trang lịch học	41
6.13	Mockup cho chức năng hiển thị chi tiết môn học tại trường.	42
6.14	Mockup cho chức năng thêm lịch học tại trường.	44
6.15	Mockup trang công việc cá nhân	45
6.16	Giao diện trang công việc cá nhân với tác vụ thêm một công việc mới cần làm	46
7.17	MVC - Quản lý lịch học	51
7.18	MVC - Nhóm chức năng đăng nhập	52
7.19	MVC - Nhóm chức năng cá nhân	52
7.20	MVC - Quản lý công việc nhóm	53
7.21	MVC - Chức năng tin nhắn	54
7.22	MVC - Chức năng thông báo	54
7.23	MVC - Todolist	55

8.24 Class diagram cho chức năng quản lý lịch học.	56
8.25 Class diagram - Quản lý công việc nhóm	75
8.26 Class diagram cho nhóm chức năng thông tin cá nhân	78

Danh sách bảng

1	Chức năng tạo và thêm lịch làm việc	13
2	Chức năng tìm kiếm công việc theo tiến độ	14
3	Chức năng hiển thị thời khóa biểu hiện tại	15
4	Chức năng tạo project private	18
5	Chức năng xóa project private	18
6	Chức năng tham gia	19
7	Chức năng rời project	19
8	Chức năng Todo-list	24
9	Chức năng quản lý Deadlines	25
10	Chức năng đăng kí tài khoản	27
11	Chức năng đăng nhập	28
12	Chức năng chỉnh sửa thông tin cá nhân	29
13	Chức năng thống kê cá nhân	30
14	Bảng các thành phần trang đăng ký đăng nhập	33
15	Bảng các thành phần trang chủ	35
16	Bảng các thành phần khung thông báo	37
17	Bảng các thành phần khung và hộp tin nhắn	39
18	Bảng các thành phần khung thông tin cá nhân	40
19	Bảng các thành phần trang lịch học	42
20	Bảng các thành phần chức năng hiển thị chi tiết môn học tại trường	43
21	Bảng các thành phần chức năng thêm lịch học tại trường	44
22	Bảng các thành phần khung và hộp tin nhắn	47
23	Method render()	79

24	Method submitEditing()	79
25	Method submitChangePassword()	80
26	Method render()	80
27	Method render()	81
28	Method submitRegisterRequest()	81
29	Method calProperties()	82
30	Method updateInfor()	82
31	Method changePassword()	83
32	Method checkLoginRequest()	83
33	Method checkLoginRequest()	84
34	Method getUser(username:String)	84
35	Method update()	85
36	Method createUser()	85

1 Giới thiệu

Tài liệu này bước đầu đặc tả dự án Hệ thống hỗ trợ học tập cho sinh viên nói chung và sinh viên Bách Khoa nói riêng. Dự án được thực hiện bởi nhóm 33, môn Thực tập Công nghệ phần mềm ở học kì 182, trường Đại học Bách Khoa thành phố Hồ Chí Minh. Dự án này nếu được hoàn thành sẽ giúp đỡ sinh viên rất nhiều bằng việc ứng dụng các tiện ích công nghệ để hỗ trợ sinh viên về quản lý thời gian biểu cũng như tiến độ học tập.

2 Mục tiêu

Phần lớn sinh viên hiện nay đang lâm vào tình cảnh chạy đôn đáo vì lượng công việc quá nặng nhọc kèm một chế độ làm việc không hợp lý. Điều này dẫn đến giảm hiệu suất công việc, kèm theo đó là stress và hàng loạt hệ lụy khác. Hệ thống ra đời nhằm hỗ trợ sinh viên sinh hoạt, làm việc và quản lý thời gian học tập một cách hiệu quả hơn, kiểm soát công việc cũng như tận dụng thời gian tốt hơn.

3 Đặc tả chức năng

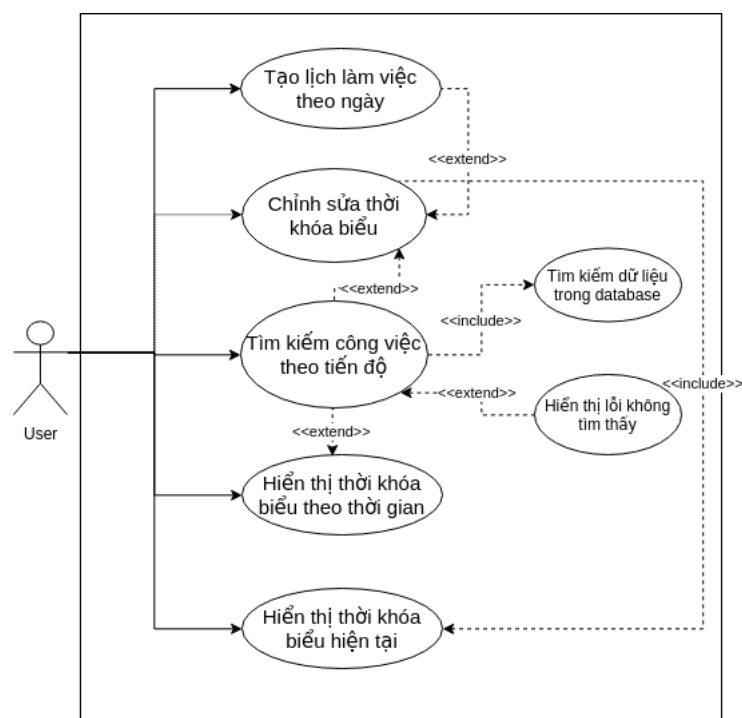
3.1 Nhóm chức năng quản lý thời khóa biểu

3.1.1 Mô tả chung

User có thể tạo cho mình nhiều loại thời khóa biểu ở trang hiển thị lịch làm việc bao gồm time table, todo-list, assignment,... Sau đây là mô tả chức năng quản lý time table gồm một số công việc sau:

- Tạo bản ghi công việc
- Chỉnh sửa chi tiết cho công việc trong time table (update, remove)
- Xem toàn bộ thời khóa biểu hiện tại
- Xem thời khóa biểu theo ngày, tuần, tháng
- Tìm kiếm công việc theo tiến độ

3.1.2 Use case diagram



Hình 3.1: Use case cho nhóm chức năng quản lý thời khóa biểu

3.1.3 Chi tiết chức năng

Use-case name	Tạo lịch làm việc cho một ngày
Actor	User
Description	Tác nhân thêm lịch làm việc của mình vào trong thời khóa biểu
Preconditions	Tác nhân đang ở trang xem thời khóa biểu của mình
Normal Flow	<ol style="list-style-type: none">1. Tác nhân chọn "Create a daily schedule" ở trang chính2. Hệ thống đưa ra một trang với thông tin là lịch.3. Tác nhân chọn vào ngày muốn thêm lịch làm việc.4. Hệ thống đưa ra một pop-up để điền chi tiết của lịch làm việc ví dụ mô tả công việc, tiến độ...5. Tác nhân chọn thời gian bắt đầu, kết thúc.6. Tác nhân chọn "Add to timetables."7. Hệ thống thêm lịch làm việc vào timetable của user và cập nhật database.
Exceptions	Exception 1 : tại bước 5. 5a Nếu tác nhân không chọn thời gian bắt đầu và kết thúc, hệ thống sẽ mở thông báo nhắc nếu bạn không đồng ý thì đặt thời gian là Null.
Alternative Flows	Alternative 1: tại bước 6: 6a. Tác nhân chọn "Quay về" tiếp tục ở bước 2

Bảng 1: Chức năng tạo và thêm lịch làm việc

Use-case name	Tìm kiếm công việc theo tiến độ
Actor	User
Description	Tác nhân tìm kiếm công việc của mình đặt trong thời khóa biểu
Preconditions	Tác nhân đang ở trang xem thời khóa biểu của mình
Normal Flow	<ol style="list-style-type: none">1. Tác nhân chọn "Search" ở trang chính2. Hệ thống đưa ra một pop-up về một số thông số như : tên ghi chú, thời gian hoàn thành, ngày(tháng), tiến độ công việc,..3. Tác nhân điền thông tin mình muốn tìm kiếm trong pop-up.4. Tác nhân nhấn vào nút search.5. Tác nhân chọn tác vụ kết thúc6. Hệ thống quay về trang chính.
Exceptions	<p>Exception 1 : tại bước 4. 4a Nếu người dùng không điền vào bất cứ hàng nào thì hệ thống hiển thị thông báo yêu cầu điền thông tin..</p> <p>Exception 2 : tại bước 5: 5a Nếu hệ thống không tìm thấy dữ liệu trong database. Hệ thống mở bảng thông báo yêu cầu người dùng nhập lại hoặc tạo một ghi chú mới. 5b Quay lại bước 2.</p>
Alternative Flows	<p>Alternative 1: tại bước 4: 4.a Tác nhân không chọn tác vụ search ->chọn tác vụ cancel. Hệ thống đóng bảng tìm kiếm. 4.b Quay lại bước 1</p> <p>Alternative 2: tại bước 6: 6.a Tác nhân chọn tác vụ Edit Record ->hệ thống gọi chức năng cập nhập lịch làm việc cho người dùng.</p>

Bảng 2: Chức năng tìm kiếm công việc theo tiến độ

Use-case name	Hiển thị thời gian biểu hiện tại của user
Actor	User
Description	Tác nhân hiển thị thời khóa biểu
Preconditions	Tác nhân đang ở trang xem thời khóa biểu của mình
Normal Flow	<ol style="list-style-type: none">1. Tác nhân chọn "Display timetable" ở trang chính2. Hệ thống đọc dữ liệu trong database.3. Hệ thống hiển thị dữ liệu các ghi chú theo lịch.4. Tác nhân chọn nút "Back"5. Hệ thống hiển thị trang chính.
Exceptions	Exception 1 : tại bước 3. 3.a Nếu không có bản record nào được lưu trong database, hệ thống sẽ hiển thị lịch trống cho phép người dùng thêm bản ghi.
Alternative Flows	Alternative 1: tại bước 4: 4.a Tác nhân không chọn tác vụ back ->chọn tác vụ Edit. 4.b Hệ thống mở bản pop-up cho phép tác nhân sửa thông tin record. 4.c Tác nhân bấm nút Save. 4.d Hệ thống cập nhập vào database. 4.e Quay lại bước 1. Alternative 2: tại bước 4: 4.f Tác nhân không chọn tác vụ back ->chọn tác vụ Add record. 4.g Hệ thống mở bản pop-up cho phép tác nhân sửa thông tin record. 4.h Tác nhân bấm nút Save. 4.p Hệ thống cập nhập vào database. 4.q Quay lại bước 1.

Bảng 3: Chức năng hiển thị thời khóa biểu hiện tại

3.2 Nhóm chức năng quản lý công việc nhóm

3.2.1 Mô tả chung

User có thể tự đăng kí cho mình một tài khoản để đăng nhập vào hệ thống. Khi đăng nhập vào hệ thống, user có thể chỉnh sửa thông tin cá nhân của mình cũng như theo dõi các thông kê cá nhân.

- Tạo project: Mỗi người dùng (user) đều có quyền tạo ra project của riêng mình, project có 2 chế độ công khai (public) và bí mật (private). Người dùng (user) khác có thể đề nghị tham gia các project công khai, còn với project bí mật thì chỉ có chủ project(owner) mới có thể mời người dùng (user) khác vào tham gia.
- Xóa project: Chủ project (owner) có khả năng xóa bỏ project.
- Tìm, tham gia project: Người dùng (user) có thể tìm được tất cả các project ở chế độ công khai (public) và tham gia vào project với sự cho phép (permission) của chủ project (owner).
- Rời project: Người dùng (user) đang ở trong project có thể rời project bất kỳ lúc nào.
- Mời một người dùng khác tham gia project: Chủ project (owner) có thể mời những người dùng (user) khác tham gia vào project của mình. Những người dùng này sẽ nhận được thông báo và có quyền chấp nhận (accept) hay từ chối (refuse) yêu cầu này.
- Lên thời khóa biểu công việc cho project: Tương tự như với thời khóa biểu cá nhân, chủ project (owner) có thể tạo thời khóa biểu và phân công công việc cho từng thành viên của project.
- Xem, hiệu chỉnh công việc, deadline: Mỗi thành viên có thể xem công việc và deadline của cả group. Với công việc (task) của mỗi cá nhân, thành viên có thể hiệu chỉnh trạng thái công việc (như việc đã hoàn thành hay progress đến đâu). Chủ project (owner) có thêm khả năng extend deadline.

Hình 3.2: Use case cho nhóm chức năng quản lý nhóm

3.2.3 Chi tiết chức năng

Use-case name	Create private project
Actor	User
Description	Tạo một private project
Preconditions	Người dùng đã đăng nhập vào hệ thống
Normal Flow	<ol style="list-style-type: none">1. Người dùng chọn tính năng 'Create new project'2. Người dùng đặt tên project và xác nhận muốn tạo project3. Nếu thành công, hệ thống cập nhật thông tin người dùng mới trên database4. Người dùng tiến hành mời các user khác cùng tham gia project
Exceptions	3. Hệ thống báo tên project bị trùng, cần đặt lại tên
Alternative Flows	Không

Bảng 4: Chức năng tạo project private

Use-case name	Remove private project
Actor	User
Description	Xóa một private project
Preconditions	Người dùng đã đăng nhập vào hệ thống
Normal Flow	<ol style="list-style-type: none">1. Người dùng chọn tính năng 'Show my owned project'2. Người dùng tiến hành chọn project muốn xóa3. Người dùng xác nhận muốn xóa project
Exceptions	Không
Alternative Flows	Không

Bảng 5: Chức năng xóa project private

Use-case name	Join public project
Actor	User
Description	Tham gia một project công khai và được chấp nhận
Preconditions	Người dùng đã đăng nhập vào hệ thống
Normal Flow	<ol style="list-style-type: none">1. Người dùng chọn tính năng 'Find public projects'2. Người dùng tiến hành tìm project bằng tên3. Người dùng chọn project muốn tham gia4. Người dùng được owner của của project đó chấp thuận
Exceptions	Không
Alternative Flows	Không

Bảng 6: Chức năng tham gia

Use-case name	Leave project
Actor	User
Description	Rời một project đang tham gia
Preconditions	Người dùng đã đăng nhập vào hệ thống
Normal Flow	<ol style="list-style-type: none">1. Người dùng chọn tính năng 'Find all projects'2. Người dùng tiến hành chọn một project3. Người dùng chọn tính năng "Leave project"4. Người dùng xác nhận muốn rời project
Exceptions	Không
Alternative Flows	Không

Bảng 7: Chức năng rời project

3.3 Nhóm chức năng quản lý công việc cá nhân

3.3.1 Mô tả chung

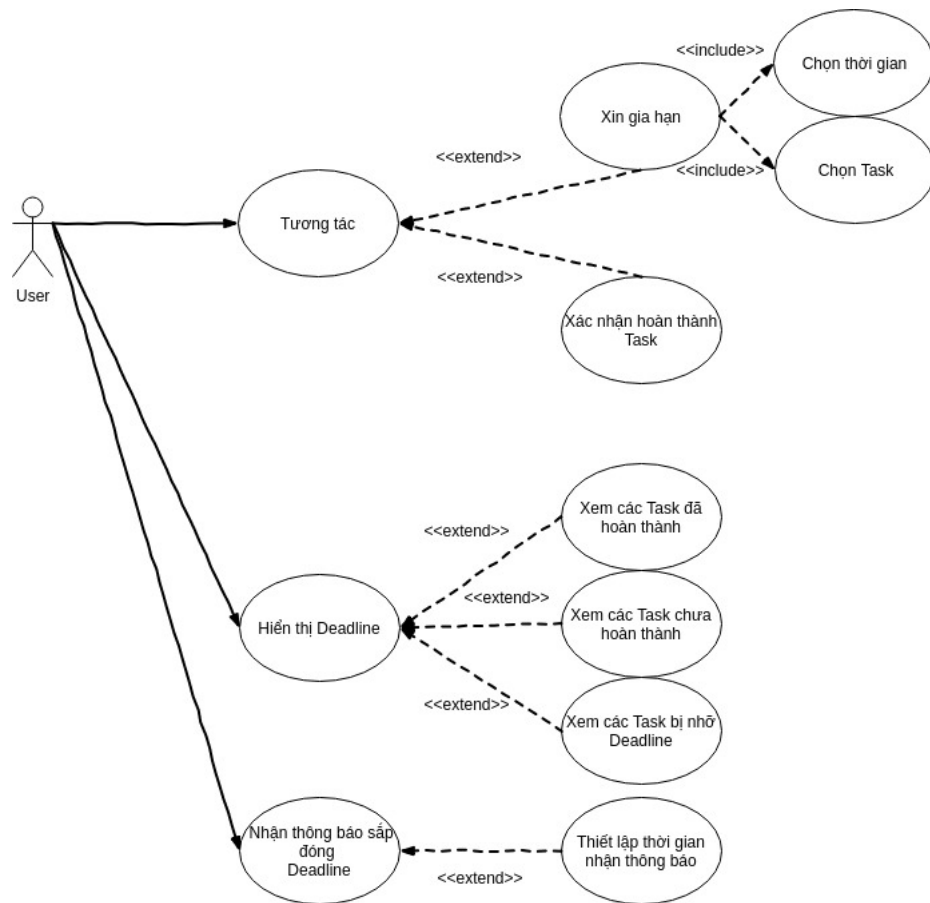
Người dùng có thể quản lý các việc cần làm, các deadlines

Todo-list

- Tùy chỉnh Todolist: Mỗi người dùng có quyền tùy chỉnh cá nhân các Task được phân công trong nhóm. Tùy chỉnh bao gồm thêm Tag để phân loại các Task hoặc tùy chỉnh Task được phân công: gắn Tag, thêm ghi chú hoặc mô tả, tùy chỉnh thời gian thực thi, thêm Priority, tùy chỉnh Due Date và thêm các thiết lập tính năng khác.
- Hiển thị Todolist: Người dùng có thể xem Todolist các Task và đánh dấu hoàn thành trên các Task hoặc chọn hủy bỏ Task.
- Nhận thông báo Task sắp tới: Người dùng có thể nhận được thông báo của các Task sắp đến hạn chót phải hoàn thành.

Quản lý deadlines

- Tương tác: Mỗi người dùng có quyền xin gia hạn Deadline cho Task của bản thân - yêu cầu chỉ được thông qua khi có sự chấp nhận của Owner tạo Project. Khi đã hoàn thành xong Task, người dùng phải gửi file và xác nhận hoàn thành trước Deadline.
- Hiển thị Deadline: Người dùng có thể xem Deadline các Task chưa được hoàn thành của bản thân cũng như có thể xem lại lịch sử các Task mà bản thân đã hoàn thành hoặc bị nhỡ Deadline.
- Nhận thông báo sắp đóng Deadline: Khi sắp đến Deadline thời điểm mà Owner đã thiết lập, người dùng sẽ nhận thông báo qua Gmail.



Hình 3.4: Use case cho nhóm chức năng quản lý deadlines

3.3.3 Chi tiết chức năng

Use-case name	Quản lý Todolist.
Actor	Thành viên trong nhóm (User).
Description	User quản lý việc thực hiện các Task được chia của bản thân, xem TodoList hằng ngày và nhận thông báo nếu Task sắp hết thời gian thực thi.
Preconditions	User là thành viên trong Project và được nhóm trưởng (Owner) chia Task.
Normal Flow	<ol style="list-style-type: none">1. User đăng nhập vào trang chủ cá nhân.2. User chọn tính năng Todolist.3. 3a. Người dùng chọn tùy chỉnh Todolist4. 4a.a. Người dùng chọn thêm Tag5. 4a.b. Người dùng chọn điều chỉnh Task6. 5a.a. Hiện Form, người dùng điền thông tin trên Form sau đó bấm chọn Create7. 5a.b. Hiện Form, người dùng điền thông tin trên Form sau đó bấm chọn.8. 6a. Hệ thống lấy thông tin từ Form, cập nhật database, cuối cùng hiện thông báo hoàn thành.9. 3b. Người dùng xem Todolist.10. 4b. Người dùng đánh dấu hoàn thành hoặc hủy bỏ Task.11. 5b. Hệ thống cập nhật lại database theo tương tác của người dùng.12. 3c. Người dùng nhận được thông báo Task hoặc chọn thiết lập thời gian nhận thông báo.13. 4c. Người dùng xác nhận đã xem thông báo hoặc tùy chỉnh lại thời gian nhận thông báo.
Exceptions	<ol style="list-style-type: none">1. 6a. Hệ thống báo Form không hợp lệ hoặc database cập nhật không thành công.2. 5b. Hệ thống báo cập nhật database không thành công.
Alternative Flows	Không có.

Bảng 8: Chức năng Todo-list

Use-case name	Quản lý Deadline.
Actor	Thành viên trong nhóm (User).
Description	User xem Deadline các Task của bản thân, tương tác với Owner thay đổi hoặc xác nhận hoàn thành Deadline và nhận thông báo nếu Task nào sắp đến Deadline.
Preconditions	User là thành viên trong Project và được nhóm trưởng (Owner) chia Task.
Normal Flow	<ol style="list-style-type: none">1. User đăng nhập vào trang chủ cá nhân.2. User chọn tính năng Deadline.3. 3a. Người dùng chọn tương tác.4. 4a.a. Người dùng chọn xin gia hạn Deadline.5. 4a.b. Người dùng chọn xác nhận hoàn thành Task.6. 5a.a. Hiện Form, người dùng điền thông tin trên Form sau đó bấm chọn Create7. 5a.b. Hệ thống cập nhật lại database và gửi thông báo đến cho Owner.8. 6a.a. Hệ thống lấy thông tin từ Form, gửi đến cho Owner, nếu được Owner thông qua thì hệ thống tự động cập nhật database và hiện thông báo thành công cho người dùng.9. 4b. Người dùng chọn xem Task đã hoàn thành, Task chưa hoàn thành hoặc Task bị nhỡ Deadline.10. 5b. Hệ thống lọc thông tin và hiển thị cho người dùng.11. 5b. Hệ thống cập nhật lại database theo tương tác của người dùng.12. 3c. Người dùng nhận được thông báo sắp đến Deadline hoặc chọn thiết lập lại thời gian nhận thông báo.13. 4c. Người dùng xác nhận đã xem thông báo hoặc tùy chỉnh lại thời gian nhận thông báo.
Exceptions	<ol style="list-style-type: none">1. 5a.b. Hệ thống không gửi được thông báo cho Owner.2. 6a.a. Hệ thống báo Form không hợp lệ hoặc hệ thống không gửi được Form cho Owner hoặc database cập nhật không thành công.
Alternative Flows	Không có.

Bảng 9: Chức năng quản lý Deadlines

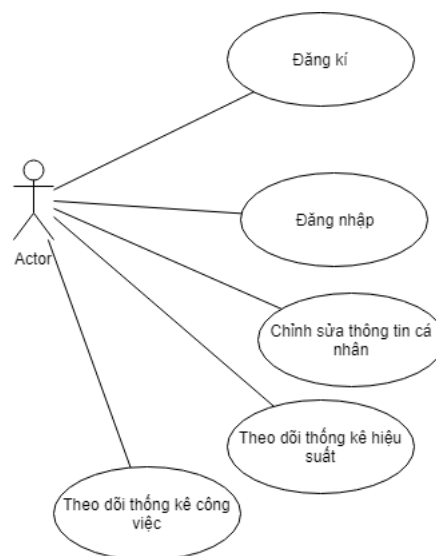
3.4 Nhóm chức năng quản lý thông tin cá nhân

3.4.1 Mô tả chung

User có thể tự đăng kí cho mình một tài khoản để đăng nhập vào hệ thống. Khi đăng nhập vào hệ thống, user có thể chỉnh sửa thông tin cá nhân của mình cũng như theo dõi các thông kê cá nhân.

- Đăng kí tài khoản
- Đăng nhập
- Chỉnh sửa thông tin cá nhân
- Thống kê cá nhân

3.4.2 Use case diagram



Hình 3.5: Use case cho nhóm chức năng quản lý thông tin cá nhân

3.4.3 Chi tiết chức năng

Use-case name	Đăng kí tài khoản
Actor	User
Description	Người dùng đăng kí tài khoản
Preconditions	Người dùng khởi động hệ thống, truy cập mục đăng kí
Normal Flow	<ol style="list-style-type: none">1. Người dùng nhập thông tin cá nhân bao gồm: Tên đăng nhập, mật khẩu, họ tên, ngày tháng năm sinh, số điện thoại và email, đồng thời nhấn nút đồng ý điều khoản sử dụng2. Hệ thống kiểm tra yêu cầu đăng kí3. Nếu thành công, hệ thống cập nhật thông tin người dùng mới trên database4. Trả về thông báo thành công và chuyển vào trang chủ
Exceptions	<p>Exception 1: tại bước 1 1a: Người dùng bấm chọn hủy</p> <p>Exception 2 : tại bước 2. 2a: Người dùng nhập tên đăng nhập đã tồn tại, nhập một trường chứa định dạng không hợp lệ hoặc bỏ trống, không chọn đồng ý điều khoản</p>
Alternative Flows	<p>Alternative 1: tại bước 1: 1a. Quay trở về trang đăng nhập</p> <p>Alternative 2: tại bước 2 2a: trả về thông báo thông tin đăng nhập không hợp lệ và trường không hợp lệ</p>

Bảng 10: Chức năng đăng kí tài khoản

Use-case name	Đăng nhập
Actor	User
Description	Người dùng đăng nhập vào hệ thống để thực hiện các chức năng khác
Preconditions	Người dùng đang ở trang đăng nhập khi mới tải trang web
Normal Flow	<ol style="list-style-type: none">1. Người dùng nhập thông tin đăng nhập gồm tên đăng nhập và mật khẩu2. Người dùng nhấn enter hoặc nút đăng nhập3. Hệ thống kiểm tra thông tin đăng nhập4. Đăng nhập thành công, người dùng được đưa đến trang chủ
Exceptions	Exception 1: tại bước 1 1a: Người dùng nhập sai thông tin
Alternative Flows	Alternative 1: tại bước 1: 1a. Thông báo "Sai tên đăng nhập hoặc mật khẩu"

Bảng 11: Chức năng đăng nhập

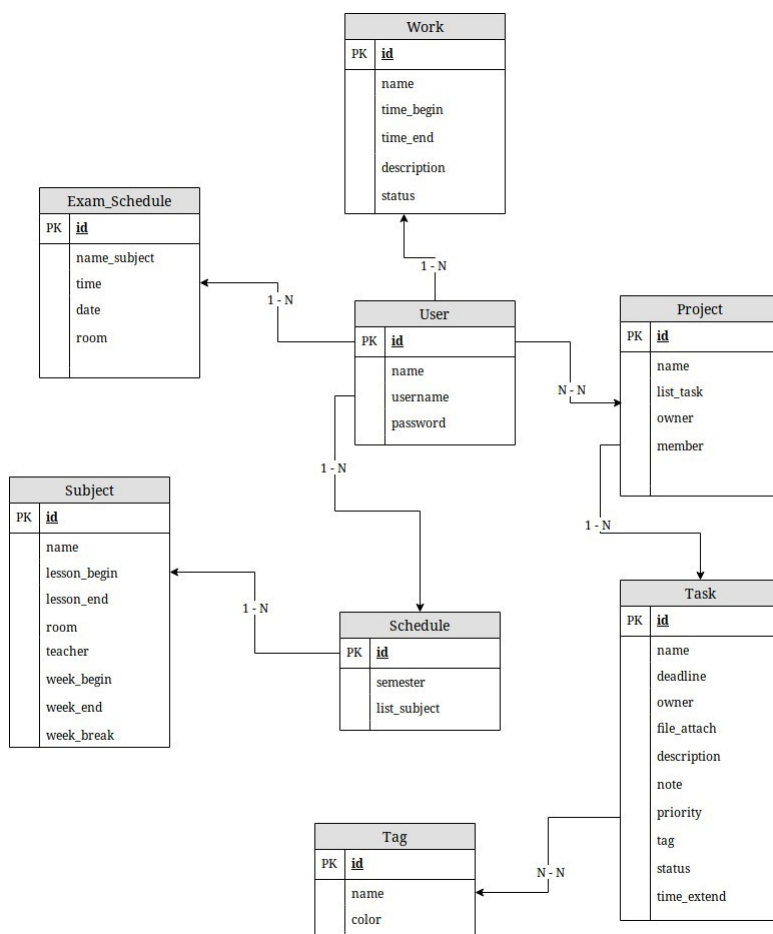
Use-case name	Chỉnh sửa thông tin cá nhân
Actor	User
Description	Người dùng chỉnh sửa thông tin cá nhân của bản thân
Preconditions	Người dùng đã đăng nhập vào hệ thống
Normal Flow	<ol style="list-style-type: none">1. Người dùng chọn mục " cá nhân", chọn " chỉnh sửa"2. Giao diện chỉnh sửa hiện lên với các mục chứa thông tin hiện tại. Người dùng thay đổi thông tin mong muốn. Riêng với mật khẩu, người dùng cần nhập mật khẩu cũ một lần và mật khẩu mới hai lần để xác nhận. Mật khẩu sẽ không hiển thị3. Người dùng nhấn nút xác nhận4. Hệ thống kiểm tra và cập nhật thông tin người dùng5. Thay đổi thông tin thành công, trở về trang cá nhân
Exceptions	Exception 2: tại bước 1 1a:Người dùng nhấn hủy Exception 1: tại bước 4 4a: Người dùng nhập thông tin không hợp lệ ở một trường nào đó
Alternative Flows	Alternative 1: tại bước 1: 1a. Trở về trang cá nhân Alternative 2: tại bước 4: 1a. Thông báo trường thông tin không hợp lệ, quay lại bước 1

Bảng 12: Chức năng chỉnh sửa thông tin cá nhân

Use-case name	Thống kê cá nhân
Actor	User
Description	Người dùng vào thống kê cá nhân để xem thống kê về công việc của bản thân
Preconditions	Người dùng đã đăng nhập vào hệ thống
Normal Flow	<ol style="list-style-type: none">1. Người dùng chọn mục " cá nhân", chọn "thống kê"2. Hệ thống nhận thông tin từ database, thống kê các thông tin của người dùng và hiển thị lên màn hình
Exceptions	Không
Alternative Flows	Không

Bảng 13: Chức năng thống kê cá nhân

4 Thiết kế sơ bộ cơ sở dữ liệu



Hình 4.6: Thiết kế cơ sở dữ liệu sơ bộ

5 Các yêu cầu phi chức năng

- Hệ thống tương tác với người dùng qua nền Web, hoạt động tốt trên các nền tảng hỗ trợ HTML5
- Hệ thống phải đảm bảo trực tuyến với xác suất 99% cho một giờ truy cập liên tục
- Nếu server sập, cần khởi động lại trong vòng ít nhất 30 phút
- Giao diện web hỗ trợ đầy đủ các loại độ phân giải màn hình
- Giao diện web hỗ trợ đa thiết bị bao gồm Desktop, Tablet và Mobile
- Với tốc độ internet trên 8Mbps, thời gian toàn phần để tải trang không quá 3 giây
- Hệ thống có thể lưu trữ dữ liệu cho ít nhất 5000 người dùng mà không gặp lỗi tràn bộ nhớ hay đáp ứng chậm
-

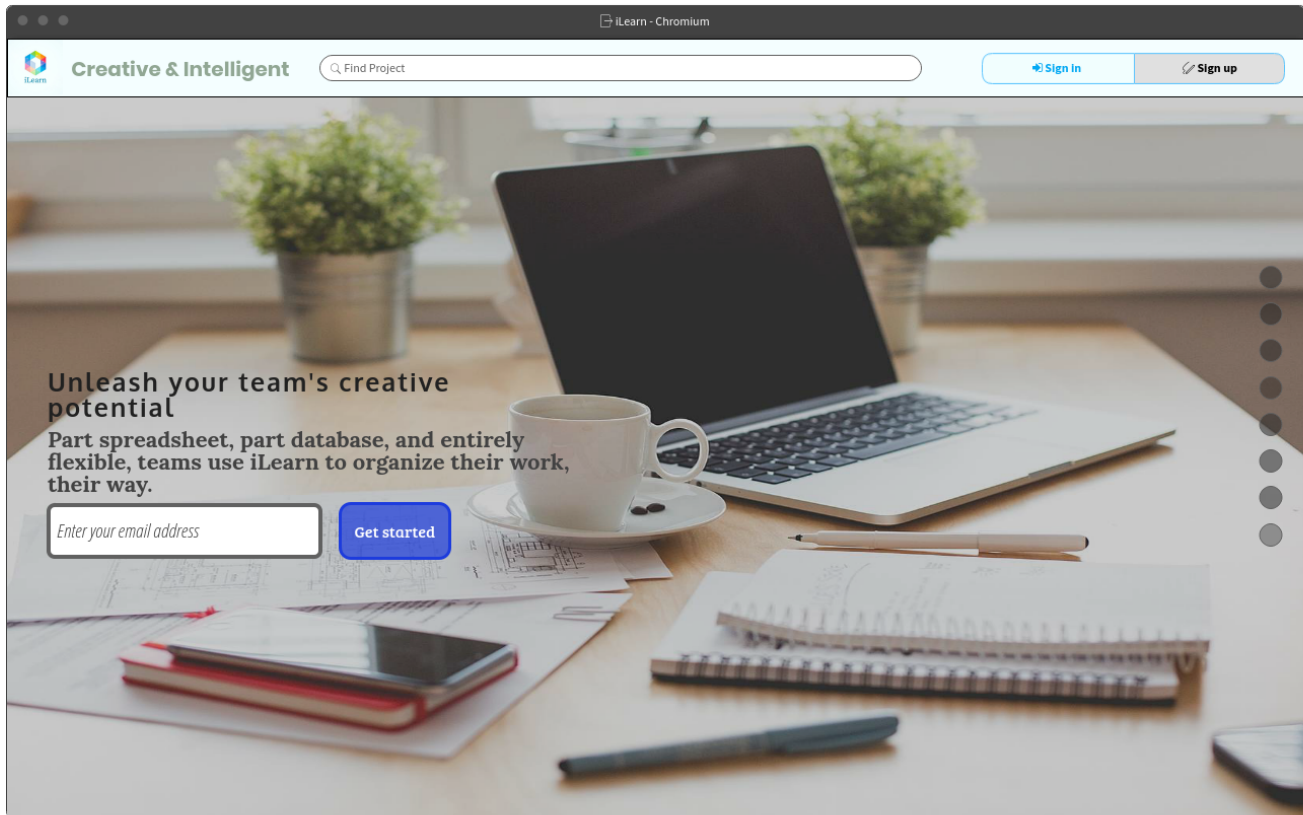
6 User Story và Mockup

6.1 Trang đăng ký đăng nhập

6.1.1 User Story

Khi người dùng truy cập trang web mà chưa có account, đây sẽ là giao diện mà người dùng sẽ tương tác đăng ký và đăng nhập.

6.1.2 Mockup



Hình 6.7: Mockup đăng ký đăng nhập

Các thành phần của giao diện được mô tả như sau:

Đối tượng	Mô tả
Biểu tượng iLearn	Logo của trang web, đồng thời cũng là nút kích hoạt animation cho logo
Nút Sign In	Nhấn nút để hiển thị khung tên và mật khẩu đăng nhập.
Nút Sign Up	Nhấn nút để hiển thị khung điền thông tin đăng ký tài khoản
Nút Get started	Nhấn nút để cho phép đăng ký nhanh bằng gmail
Khung tìm kiếm Project	Tìm kiếm và hiển thị tên các Project sẵn có.
Các điểm chuyển hình ảnh	Chuyển hình ảnh nền giới thiệu về thông tin và cách sử dụng ứng dụng hiệu quả.

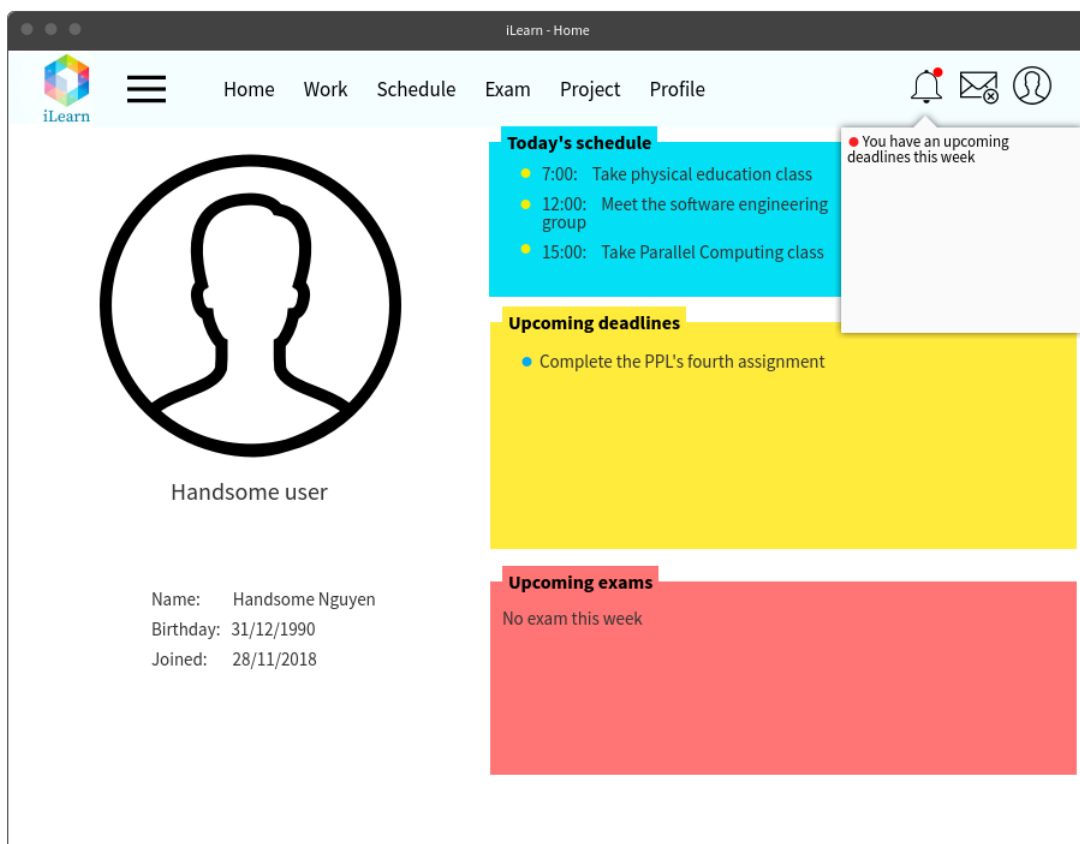
Bảng 14: Bảng các thành phần trang đăng ký đăng nhập

6.2 Trang chủ

6.2.1 User Story

Sau khi người dùng đăng nhập vào hệ thống, trang chủ hiển thị lên cho người dùng cái nhìn tổng quát cho toàn hệ thống:

6.2.2 Mockup



Hình 6.8: Mockup trang chủ của hệ thống

Các thành phần của giao diện được mô tả như sau:

Đối tượng	Mô tả
Biểu tượng iLearn	Tên của trang web, đồng thời là nút nhấn nhanh để quay lại trang chủ. Biểu tượng này hiển thị cố định ở góc trên bên trái màn ảnh ở mọi subpages.
Nút dashboard	Nhấn nút dashboard để hiển thị thanh dashboard, bao gồm các nút nhấn để chuyển chức năng.
Thanh chức năng	Bao gồm các nút chức năng điều hướng tới các chức năng của hệ thống như "Trang chủ", "Lịch", ...
Các cửa sổ chức năng con	Hiển thị khái quát về các thông báo thuộc về chức năng đó trong phạm vi một tuần tới
Profile và thông tin	Hiển thị tên người dùng, ảnh đại diện và một số thông tin cơ bản.

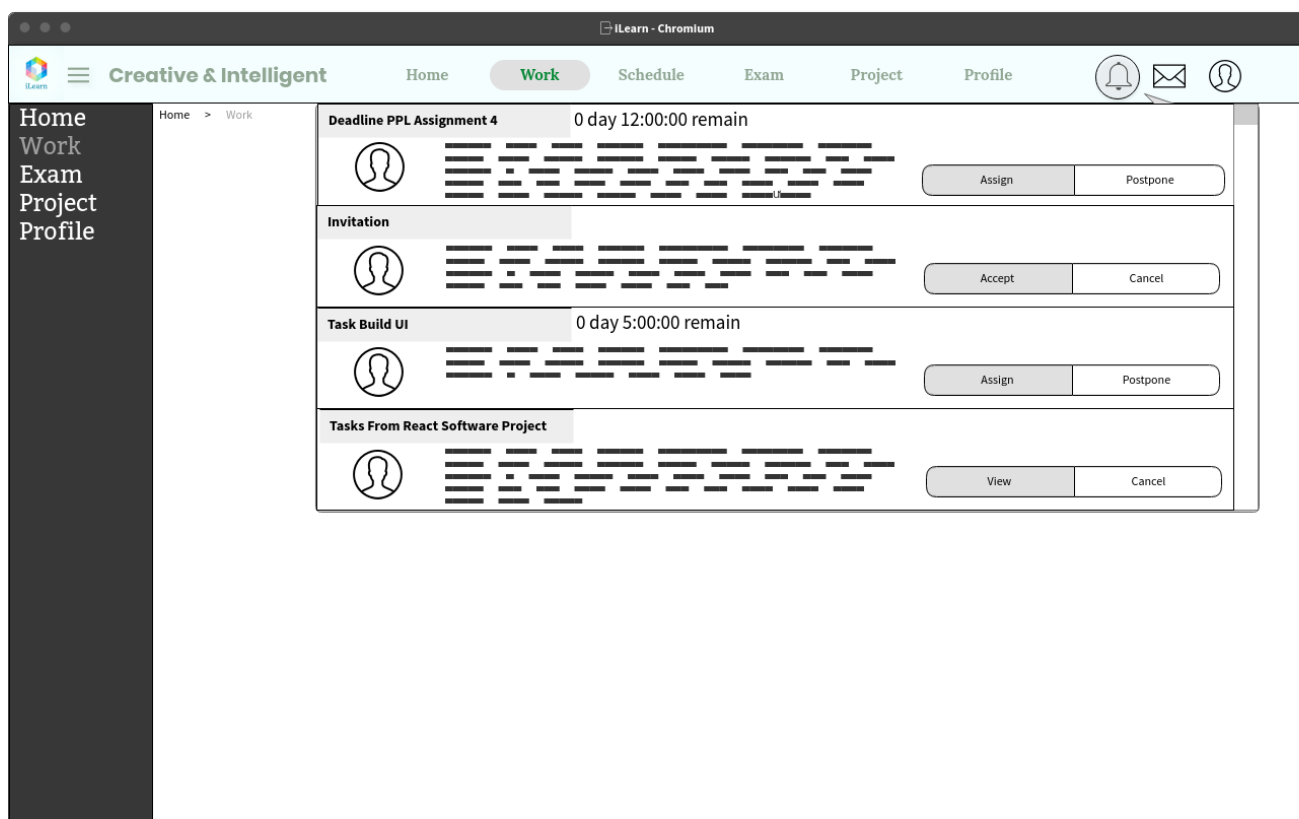
Bảng 15: Bảng các thành phần trang chủ

6.3 Khung thông báo

6.3.1 User Story

Khi người dùng đã đăng nhập thành công, người dùng có thể nhấn chọn icon hiển thị thông báo để chọn xem các thông báo của cá nhân.

6.3.2 Mockup



Hình 6.9: Mockup khung thông báo

Các thành phần của giao diện được mô tả như sau:

Đối tượng	Mô tả
Nút icon thông báo	Kích hoạt hiển thị hay thu gọn khung thông báo.
Các khung con trong thông báo	Chọn để chuyển nhanh đến trang chứa thông tin chi tiết.
Nút Postpone	Có trong khung thông báo nộp bài trước Deadline, hiển thị tùy chọn dời thời gian hoàn thành hoặc xin phép Owner gia hạn Deadline.
Nút Assign	Hiển thị tùy chọn xác thực nộp bài và hoàn thành Task cho owner Project.
Nút Accept	Có trong khung thông báo lời mời tham gia Project, xác nhận tham gia.
Nút Cancell	Có trong khung thông báo lời mời tham gia Project hoặc yêu cầu nhận Task từ Owner, xác nhận không tham gia hoặc không nhận Task.
Nút View	Có trong khung thông báo yêu cầu nhận Task từ Owner, xác nhận nhận Task và xem thông tin chi tiết các Task.

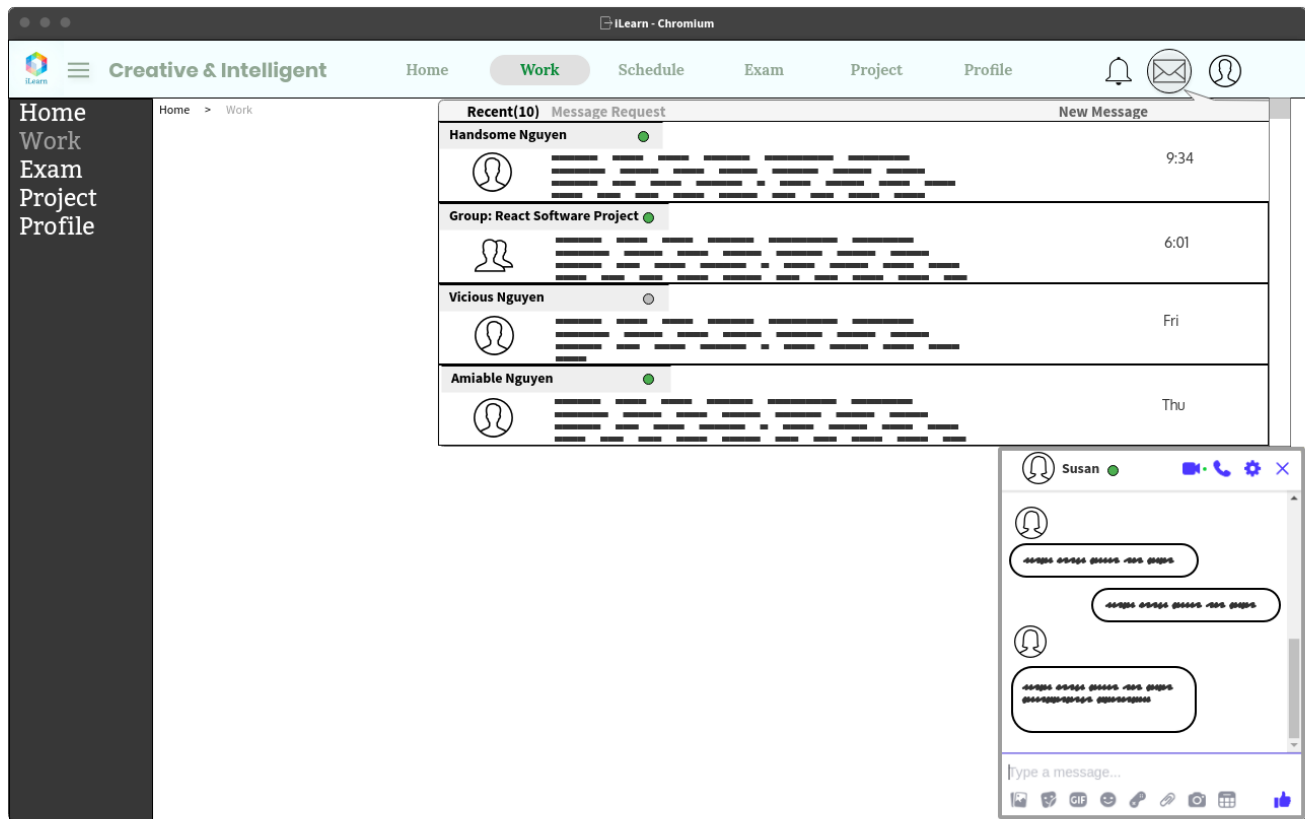
Bảng 16: Bảng các thành phần khung thông báo

6.4 Khung và hộp tin nhắn

6.4.1 User Story

Khi người dùng đã đăng nhập thành công, người dùng có thể nhấn chọn icon hiển thị tin để chọn xem các hộp tin nhắn của cá nhân với các người dùng khác. Trong hộp tin nhắn người dùng có thể thực thi các tương tác trao đổi thông tin với nhau.

6.4.2 Mockup



Hình 6.10: Mockup khung và hộp tin nhắn

Các thành phần của giao diện được mô tả như sau:

Đối tượng	Mô tả
Nút Message Request	Hiển thị các khung tin nhắn yêu cầu tương tác từ người lạ.
Nút New Message	Tạo khung tin nhắn tương tác với một cá nhân mới.
Các khung con trong tin nhắn	Chọn để mở hộp tin nhắn.
Biểu tượng hình cá nhân trong hộp tin nhắn	Dùng để xem trang cá nhân của người dùng đó.
Biểu tượng máy quay video trong hộp tin nhắn	Dùng để kích hoạt video call với người dùng đó.
Biểu tượng điện thoại trong hộp tin nhắn	Dùng để kích hoạt gọi điện.
Biểu tượng cài đặt trong hộp tin nhắn	Dùng để thiết lập cài đặt hộp tin nhắn.
Biểu tượng cross chép trong hộp tin nhắn	Dùng để đóng hộp tin nhắn.
Biểu tượng hình ảnh trong hộp tin nhắn	Dùng để gửi ảnh từ ảnh trong máy tính.
Biểu tượng gim trong hộp tin nhắn	Dùng để gửi file từ file trong máy tính.
Các biểu tượng icon trong hộp tin nhắn	Dùng để gửi các loại icon.
Các biểu tượng lịch trong hộp tin nhắn	Dùng để thiết lập thời gian biểu.

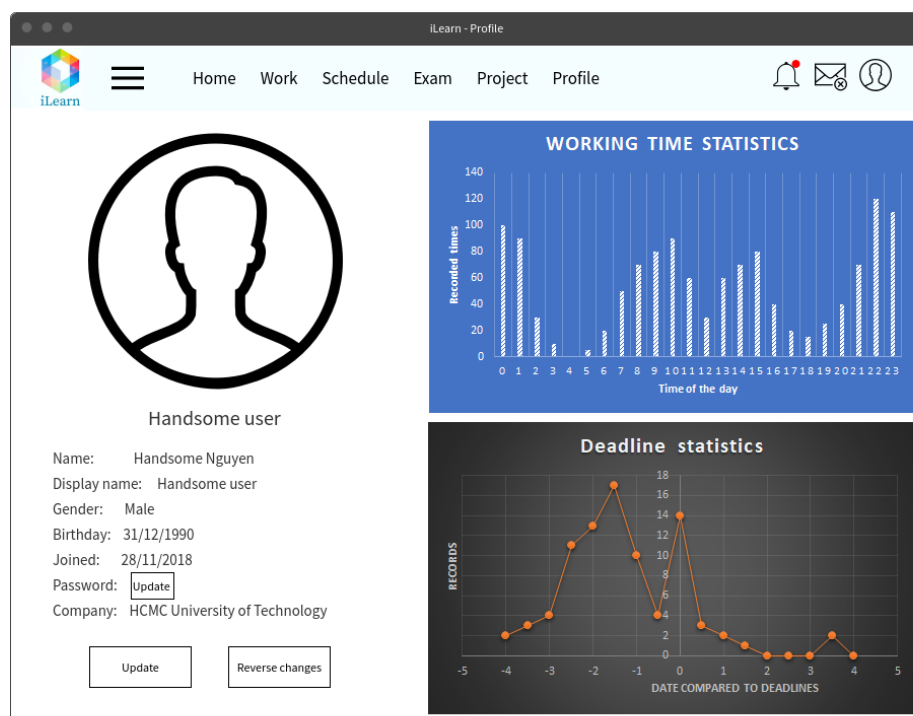
Bảng 17: Bảng các thành phần khung và hộp tin nhắn

6.5 Khung thông tin cá nhân

6.5.1 User Story

Khi người dùng đã đăng nhập thành công, người dùng có thể nhấn chọn icon hiển thị tin để chọn xem các hộp tin nhắn của cá nhân với các người dùng khác. Trong hộp tin nhắn người dùng có thể thực thi các tương tác trao đổi thông tin với nhau.

6.5.2 Mockup



Hình 6.11: Mockup khung thông tin cá nhân

Các thành phần của giao diện được mô tả như sau:

Đối tượng	Mô tả
Các trường thông tin	Hiển thị tên người dùng, các thông tin và cho phép chỉnh sửa chúng.
Biểu đồ thời gian hoạt động	Biểu đồ thống kê thời gian hoạt động trong ngày của người dùng.
Biểu đồ thống kê deadlines	Biểu đồ thống kê thời gian hoàn thành deadlines.

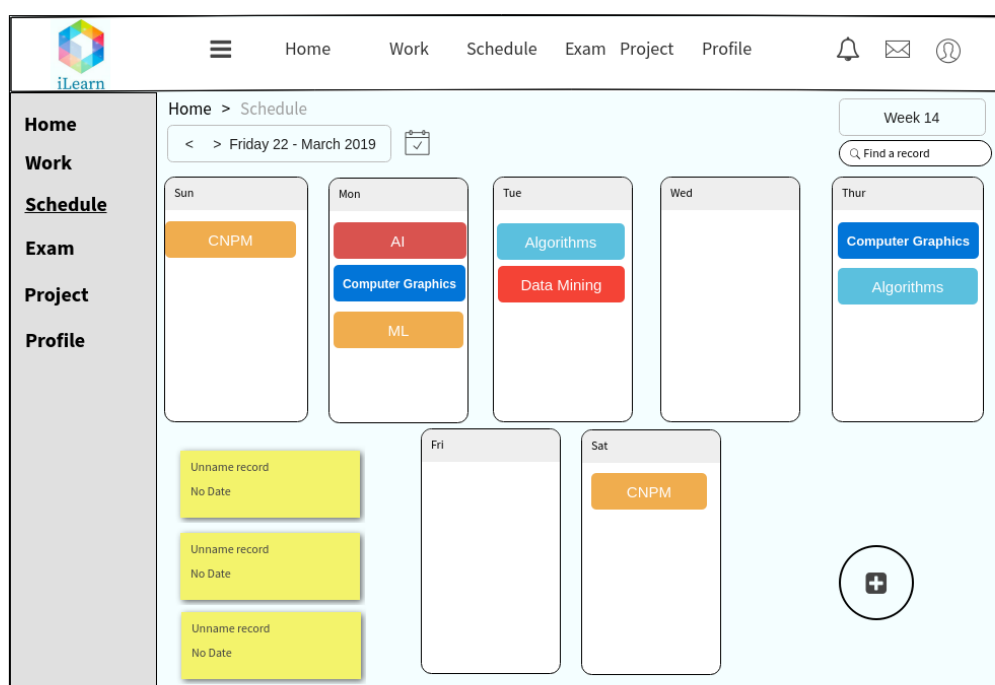
Bảng 18: Bảng các thành phần khung thông tin cá nhân

6.6 Trang lịch học

6.6.1 User Story

Người dùng nhấn vào button Schedule ở thanh sidebar hoặc ở thanh header để vào trang của chức năng lịch học tại trường.

6.6.2 Mockup



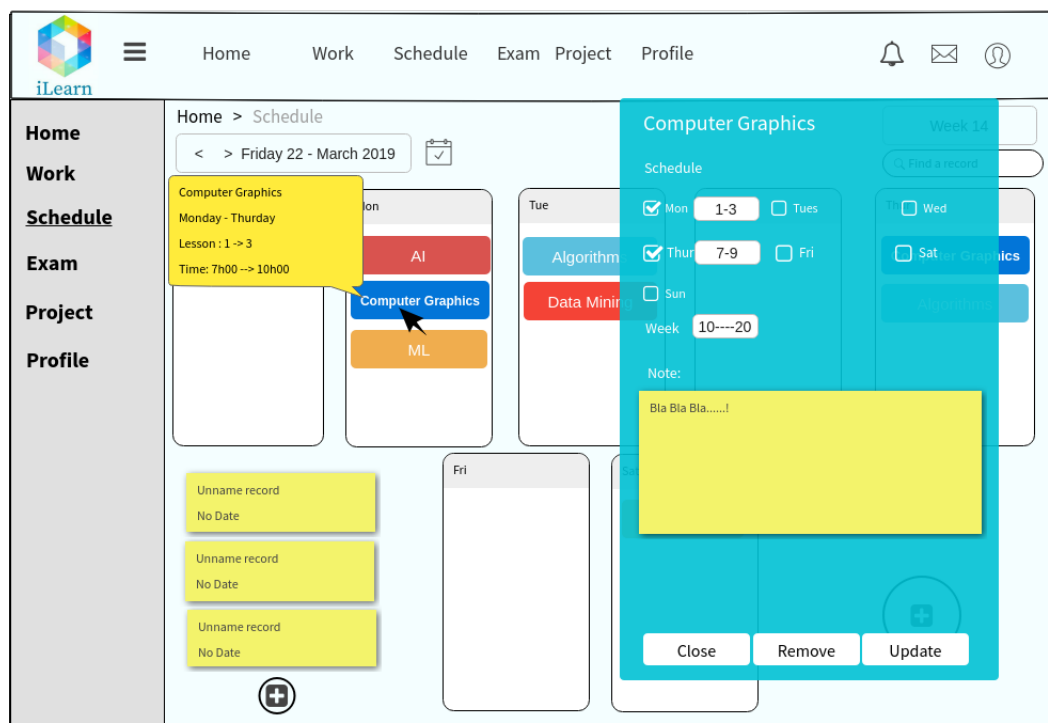
Hình 6.12: Mockup trang lịch học

Các thành phần của giao diện được mô tả như sau:

Đối tượng	Hoạt động	Chức năng
Nút (+) ở góc phải màn hình	Click	Thêm vào lịch học của một môn vào thời khóa biểu.
Biểu tượng lịch bên cạnh ngày	Click	Hiện thị lịch của tháng này
Thanh tìm kiếm	Gõ	Tìm kiếm lịch học của một môn học cụ thể
Tag môn học trên lịch	Click	Hiện thị chi tiết lịch của môn này và cho phép cập nhập thông tin

Bảng 19: Bảng các thành phần trang lịch học

- Xem lịch thời khóa biểu hiện tại được hiển thị từ thứ 2 đến chủ nhật.
- Trong 1 bảng đại diện cho một ngày vd : thứ 2 hiển thị những tag đại diện cho môn học.
- Đưa chuột vào tag môn học để biết chi tiết môn học đó được bắt đầu vào giờ nào có chú thích gì. Một pop-up được hiển thị như hình sau:



Hình 6.13: Mockup cho chức năng hiển thị chi tiết môn học tại trường.

Các thành phần của giao diện được mô tả như sau:

Đối tượng	Hoạt động	Chức năng
Checkbox tại ngày	Đánh dấu	Xác định lịch học và giờ học
Ô nhập week	Gõ	Cập nhật tuần bắt đầu và kết thúc môn.
Ô Note	Gõ	Cập nhật note của môn này.
Nút Update	Click	Update thông tin môn học
Nút Remove	Click	Xóa lịch của môn này
Nút Close	Click	Đóng bảng chọn

Bảng 20: Bảng các thành phần chức năng hiển thị chi tiết môn học tại trường

- Click vào icon (+) ở góc trái màn hình để add lịch học vào thời khóa biểu.

1. Điền thông tin các trường Name of Subject, Semestric, Week start, week end, Day, Lesson,..
2. Nhấn add để add lịch mon này vào thời khóa biểu.

The screenshot displays the iLearn web application interface. The main content area is titled 'Add to the course schedule of subject'. It contains several input fields and checkboxes: 'Name of Subject' (filled with 'Artificial Intelligence'), 'Day - Lesson' (with checkboxes for Monday 1-3, Tuesday, Wednesday, Thursday, Friday, Saturday 7-9, and Sunday), 'Semestric' (filled with '182'), and 'Period' (with 'Week 5' and 'Week 21' connected by a double arrow). Below these is a 'Note' section with a text area containing 'Midterm : 30%', 'Assignment : 20%', 'Final : 50%', and 'Absent for more 2 sessions -> score = 10.0'. An 'Add' button is at the bottom right of the form. On the left, a sidebar lists navigation options: Home, Work, Schedule (highlighted), Exam, Project, and Profile. On the right, there's a 'Week 14' selector, a search bar 'Find a record', and a list of subjects: 'Computer Graphics' and 'Algorithms'. A circular icon with a plus sign is located at the bottom right of the interface.

Hình 6.14: Mockup cho chức năng thêm lịch học tại trường.

Đối tượng	Hoạt động	Chức năng
Checkbox tại ngày	Đánh dấu	Xác định lịch học và giờ học
Ô nhập week	Gõ	Cập nhật tuần bắt đầu và kết thúc môn.
Ô Note	Gõ	Cập nhật note của môn này.
Nút Add	Click	Thêm môn học vào lịch

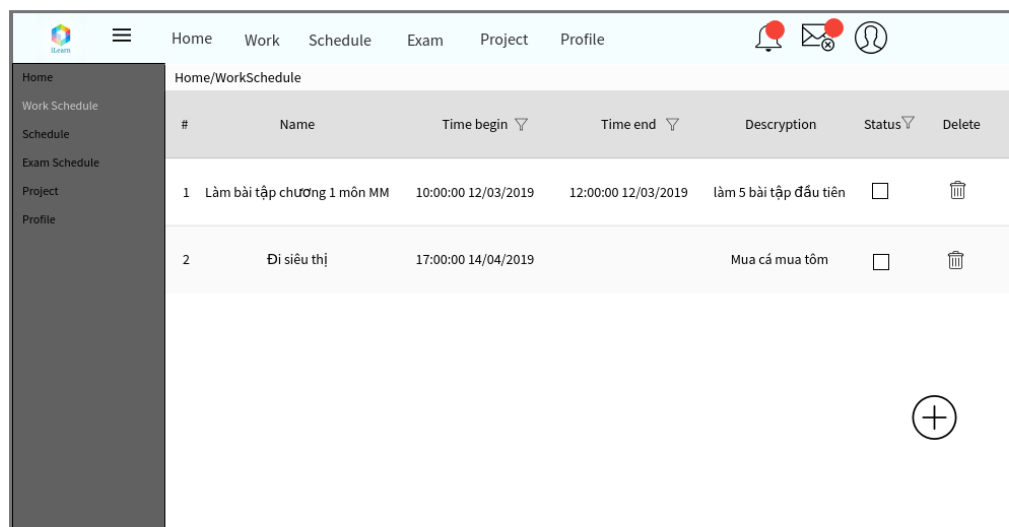
Bảng 21: Bảng các thành phần chức năng thêm lịch học tại trường

6.7 Trang công việc cá nhân

6.7.1 User Story

Người dùng nhấn vào button Work Schedule ở thanh sidebar hoặc Work ở thanh header để chuyển tới trang Work Schedule. Như hình 1

6.7.2 Mockup



Hình 6.15: Mockup trang công việc cá nhân

Các thành phần của giao diện được mô tả như sau:

1. Xem các công việc cần làm được hiển thị ra trong bảng
2. Đánh dấu vào nút ở cột Status để cập nhật trạng thái đã hoàn thành của công việc
3. Click vào icon thùng rác ở cột Delete để xóa một công việc
4. Nhấn vào dấu cộng ở nút gốc bên phải để tạo công việc cần làm mới
5. Một popup được hiển thị ra như hình 2.

(a) Điền thông tin các trường như name, chọn ngày bắt đầu và ngày kết thúc, thêm mô tả.

- (b) Nhấn Save để lưu lại
- (c) Nhấn Reset để xóa các trường đã điền
- (d) Nhấn dấu X ở góc trên bên phải để tắt và không lưu

Giao diện

The screenshot displays a web application interface for managing a work schedule. On the left, a sidebar menu lists navigation options: Home, Work Schedule, Schedule, Exam Schedule, Project, and Profile. The main content area is titled 'Home/WorkSchedule' and features a table with columns for '#', 'Name', and 'Description'. Two entries are visible in the table: '1 Làm bài tập chương 1 n' and '2 Đi siêu thị'. A modal window titled 'Create a work_schedule' is overlaid on the main content, providing a form to add a new task. The form includes fields for 'Name', 'Time begin' (with a time picker set to 11:00:00 and a date picker set to 12 May 2016), 'Time end' (with a time picker set to 11:00:00 and a date picker set to 12 May 2016), and a 'Description' text area. At the bottom of the modal, there are 'Reset' and 'Save' buttons. The background interface also shows a top navigation bar with 'Home', 'Work', and 'Schedule' tabs, and a right sidebar with a notification bell, an email icon, and a user profile icon.

Hình 6.16: Giao diện trang công việc cá nhân với tác vụ thêm một công việc mới cần làm

Phần tử	Hoạt động	Miêu tả
Nút ở cột Status	Đánh dấu	Check để đánh dấu hoàn thành
Icon cái thùng rác	Click	Xóa một công việc để làm
Dấu cộng góc phải	Click	Mở popup để thêm một công việc cần làm mới
Input của Name	Gõ	Điền thông tin của tên công việc
Input của Time begin	Chọn	Chọn giờ phút giây
Input của Time end	Chọn	Chọn ngày tháng năm
Input của Description	Gõ	Điền mô tả cho công việc
Button Save	Click	Lưu thông tin mới
Button Reset	Click	Xóa các thông tin của các trường đã điền

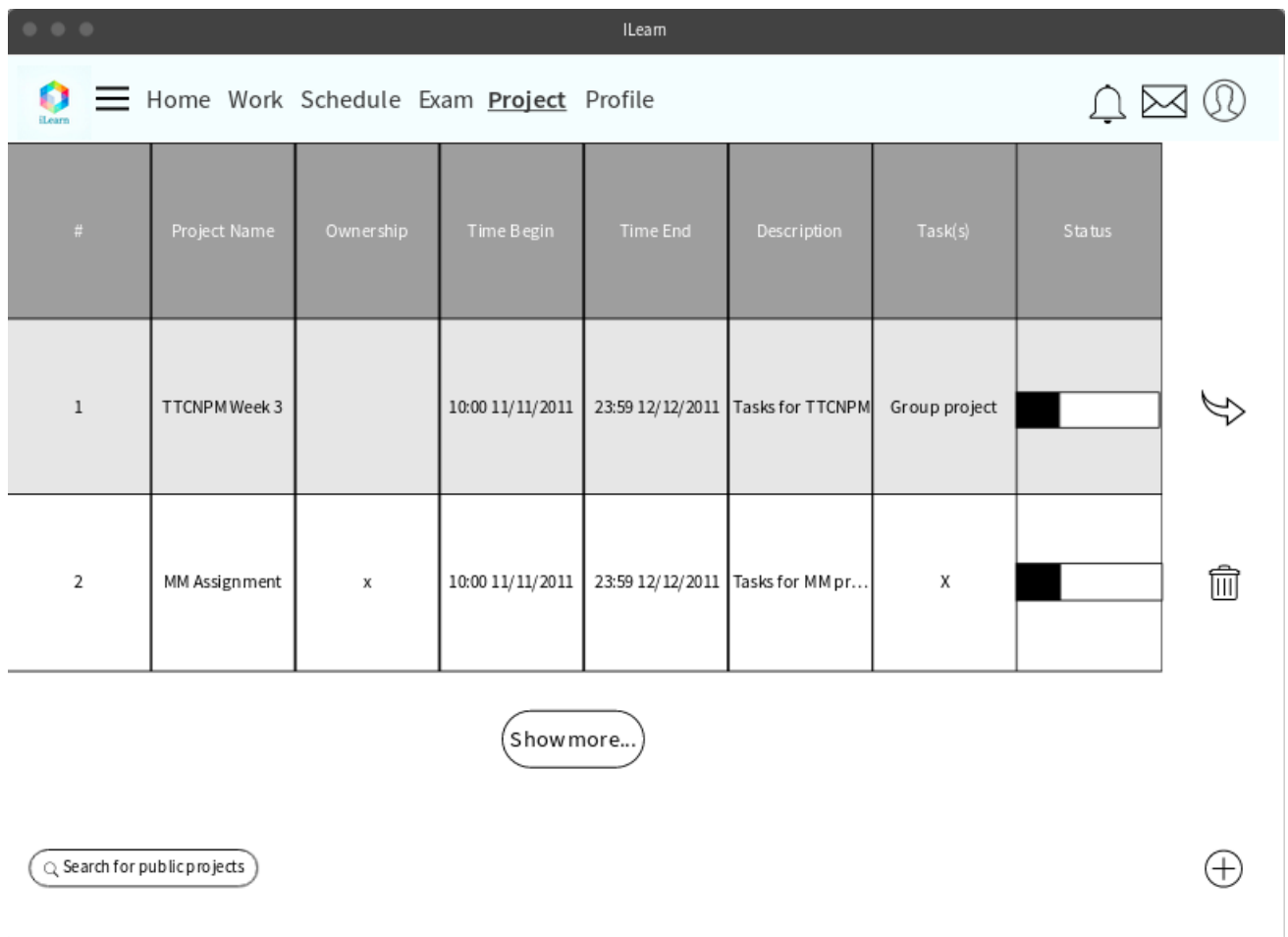
Bảng 22: Bảng các thành phần khung và hộp tin nhắn

6.8 Trang Công việc nhóm

6.8.1 Trang project chính

User Story liên quan: Là một user, Tôi muốn đến trang Project để có thể xem thông tin về các project của tôi

f Người dùng nhấn Project trên thanh header để chuyển tới trang Project như hình:

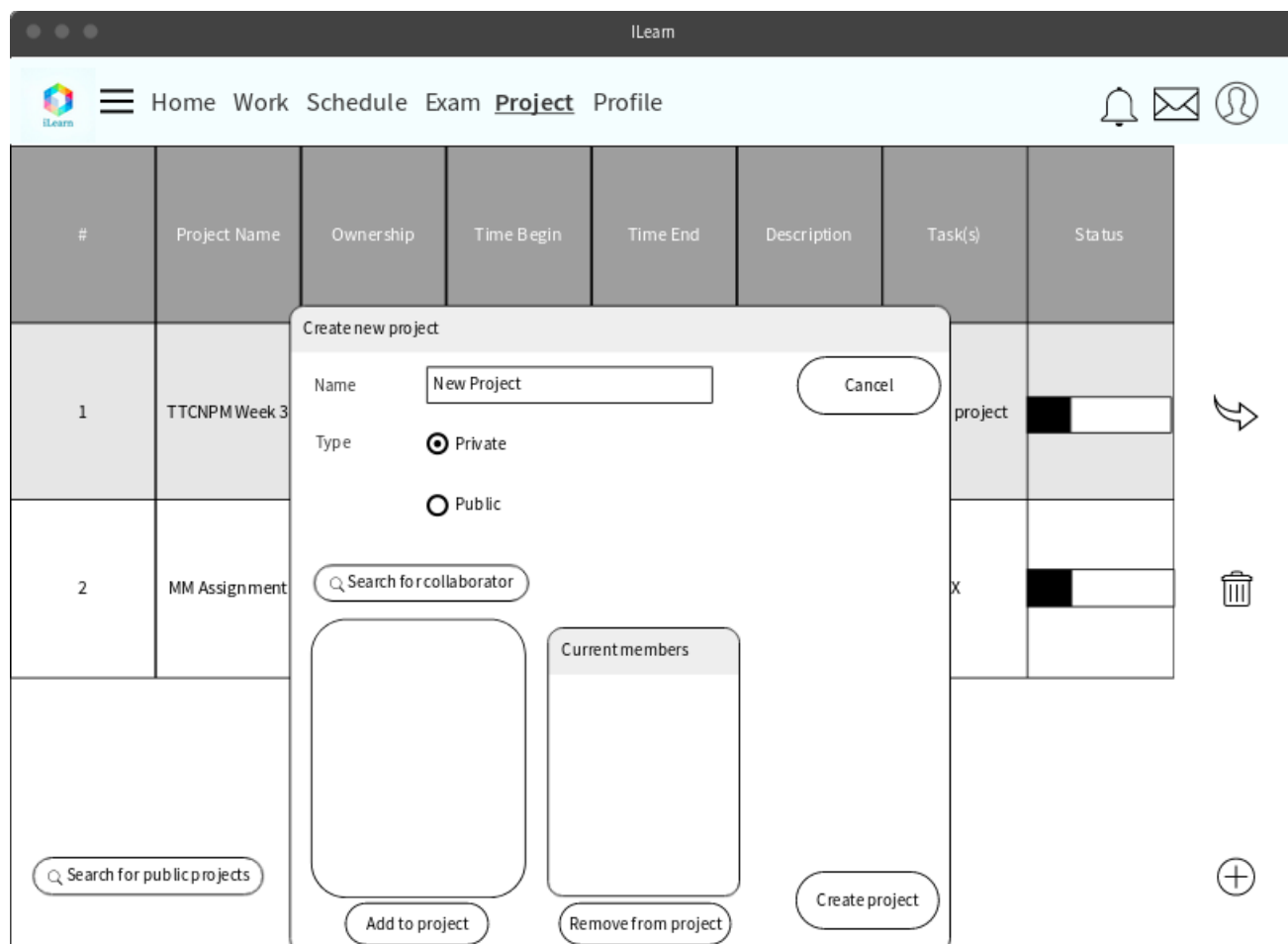


Các thành phần giao diện được mô tả như sau:

Đối tượng	Mô tả
Nút dấu cộng góc dưới phải	Mở pop-up tạo project mới
Nút mũi tên hướng sang phải	Rời project
Nút thùng rác	Xóa project
Nút 'show more'	Expand thêm list các project mà user tham gia
Project search bar góc dưới trái	Cho phép tìm public project thông qua tên

6.8.2 Tạo project mới

User Story liên quan: Là một user, Tôi muốn đến tạo một project mới để làm việc
Người dùng nhấn nút dấu cộng ở góc dưới phải trên trang project chính để tạo pop-up như hình:



Các thành phần giao diện được mô tả như sau:

Dối tượng	Mô tả
Input của Name	Đặt tên cho project mới
Radio check box Type	Chọn type cho project
Nút 'show more'	Expand thêm list các project mà user tham gia
Collaborator search bar	Cho phép owner mời các user khác tham gia project
Current member panel	Hiển thị các member hiện tại của project
Button Add to project	Add user từ phần hiển thị bên trên vào current member
Button Remove from project	Remove user từ panel current member khỏi project
Button Cancel	Thoát khỏi pop-up create new project
Button Create project	Tạo project

6.8.3 Project detail

User Story liên quan: Là một user, Tôi muốn xem chi tiết một project để biết tiến độ làm việc
Người dùng nhấn vào row chứa project muốn đến trên trang project chính để đến trang chứa project đó như hình:

ILearn

Home Work Schedule Exam **Project** Profile

Back to project main page ProjectDemo

#	User	Task	Time Begin	Time End	Description	Status
1	abc	Task 1	10:00 11/11/2011	23:59 12/12/2011	abc	<div><div></div></div>
2	xyz	Task 2	10:00 11/11/2011	23:59 12/12/2011	asd	<div><div></div></div>

Search for collaborators

Các thành phần giao diện được mô tả như sau:

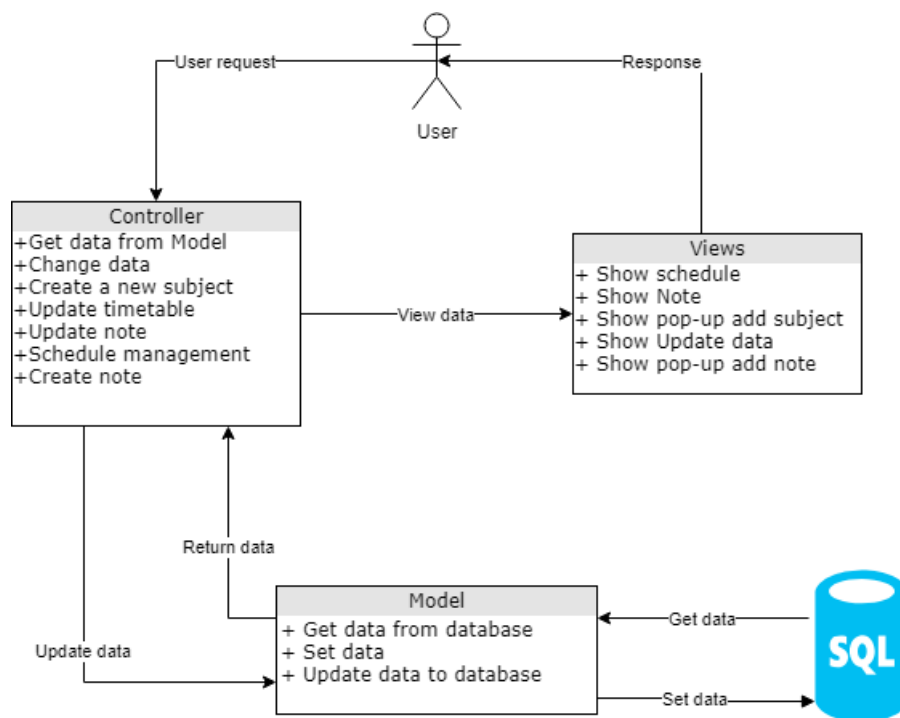
Đối tượng	Mô tả
Nút mũi tên hướng sang trái ở đầu trang	Về lại project main page
Nút thùng rác góc dưới phải	Xóa project
Nút hình cây viết	Edit thuộc tính của từng row
Collaborator search bar góc dưới trái	Cho phép tìm và mời user khác gia nhập project
Icon hình người cạnh username	Đánh dấu owner của project

7 Thiết kế kiến trúc phần mềm cho hệ thống

7.1 Mô hình chung của tất cả các chức năng

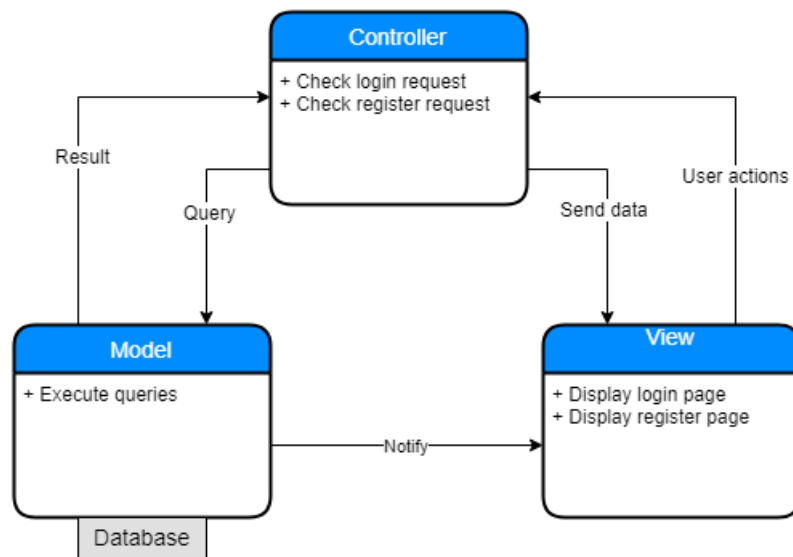
Kiến trúc phần mềm hệ thống được xây dựng theo mô hình MVC - Model Views Controller

7.2 Kiến trúc phần mềm cho chức năng quản lí lịch học trên trường

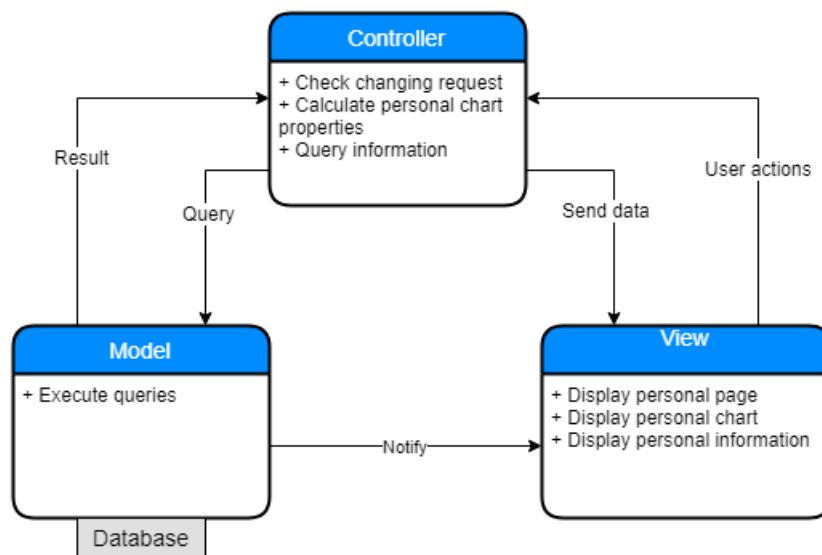


Hình 7.17: MVC - Quản lý lịch học

7.3 Kiến trúc phần mềm cho chức năng cho trang đăng ký đăng nhập, khung thông báo, khung và hộp tin nhắn

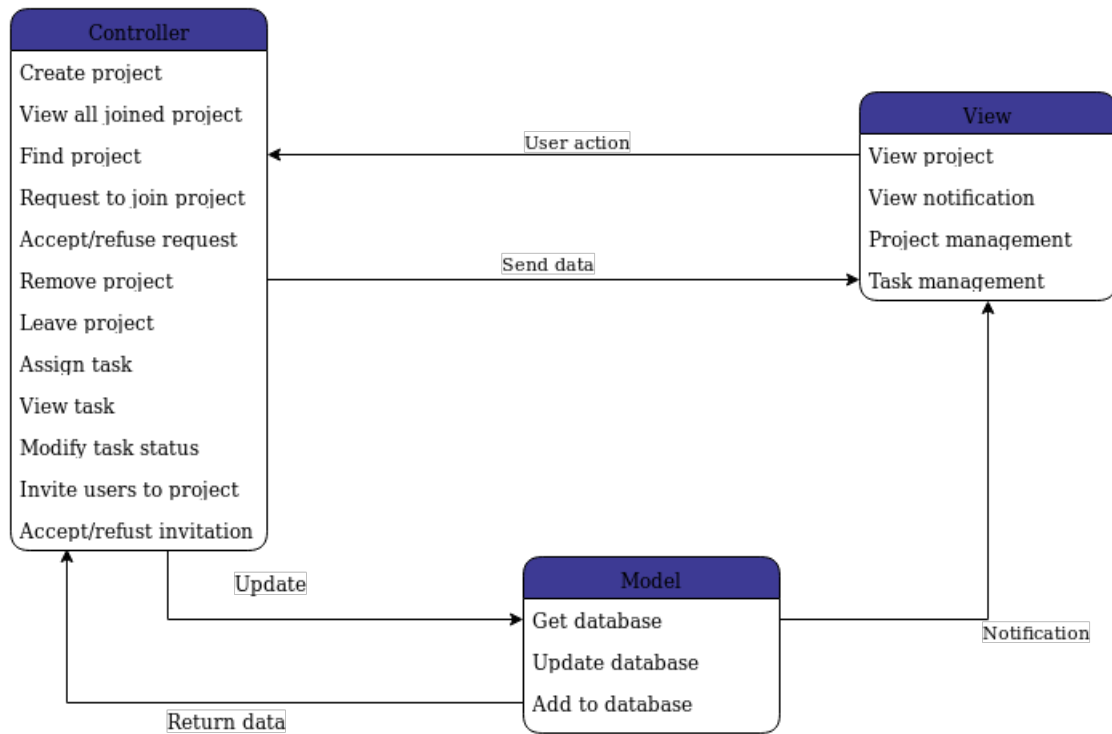


Hình 7.18: MVC - Nhóm chức năng đăng nhập



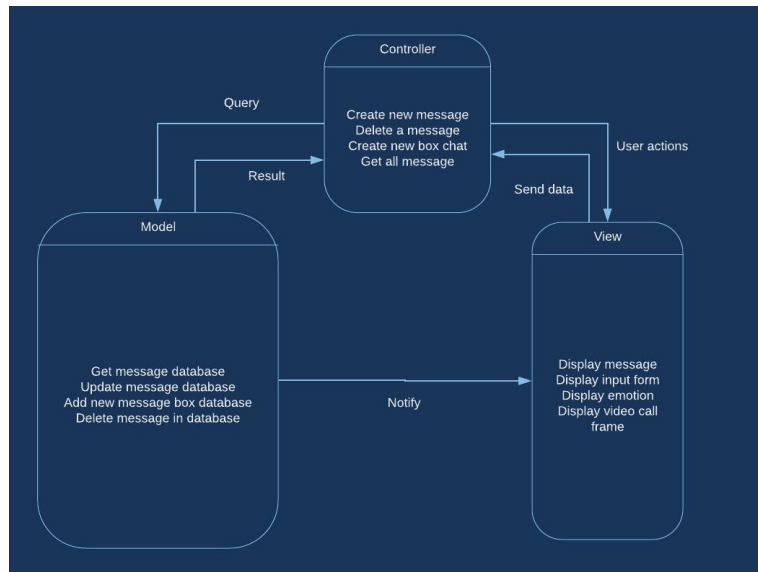
Hình 7.19: MVC - Nhóm chức năng cá nhân

7.4 Kiến trúc phần mềm cho chức năng quản lý công việc nhóm

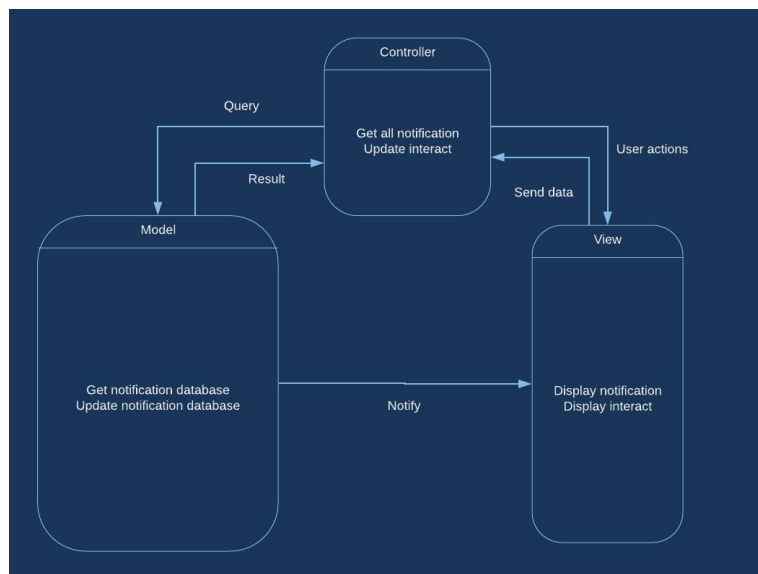


Hình 7.20: MVC - Quản lý công việc nhóm

7.5 Kiến trúc phần mềm cho trang chủ và thông tin cá nhân

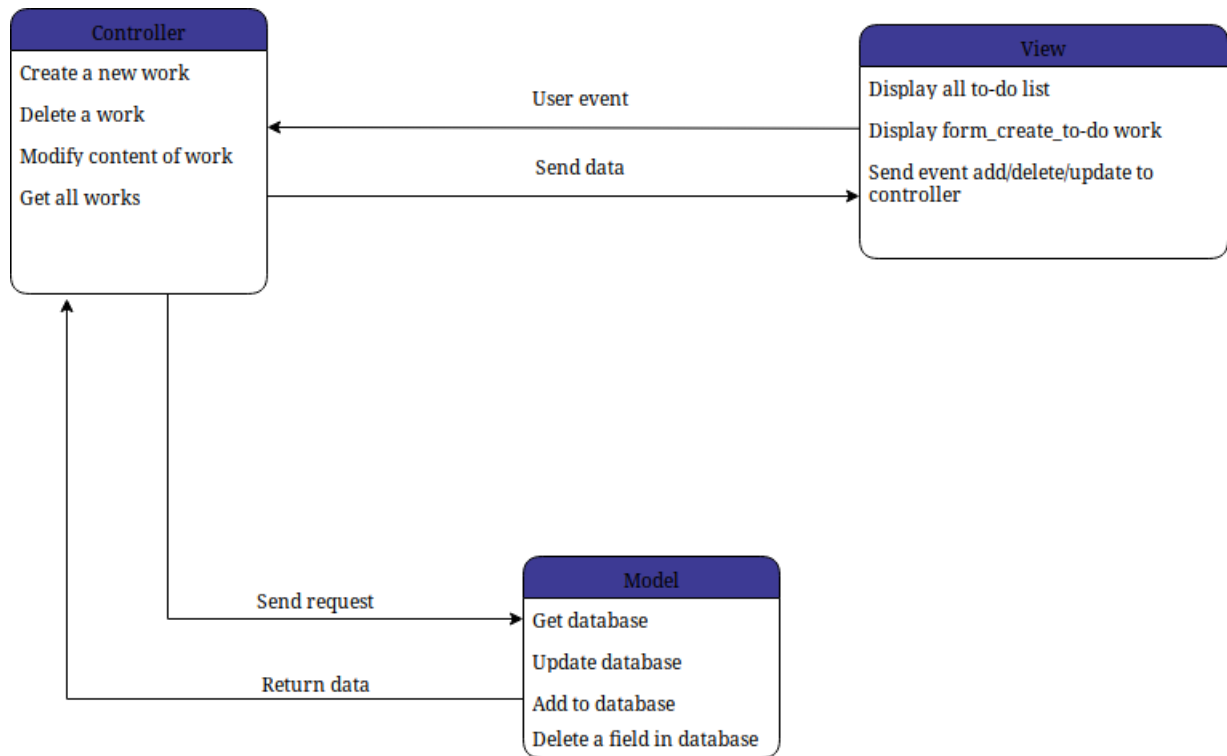


Hình 7.21: MVC - Chức năng tin nhắn



Hình 7.22: MVC - Chức năng thông báo

7.6 Kiến trúc phần mềm cho chức năng tạo những công việc cần làm

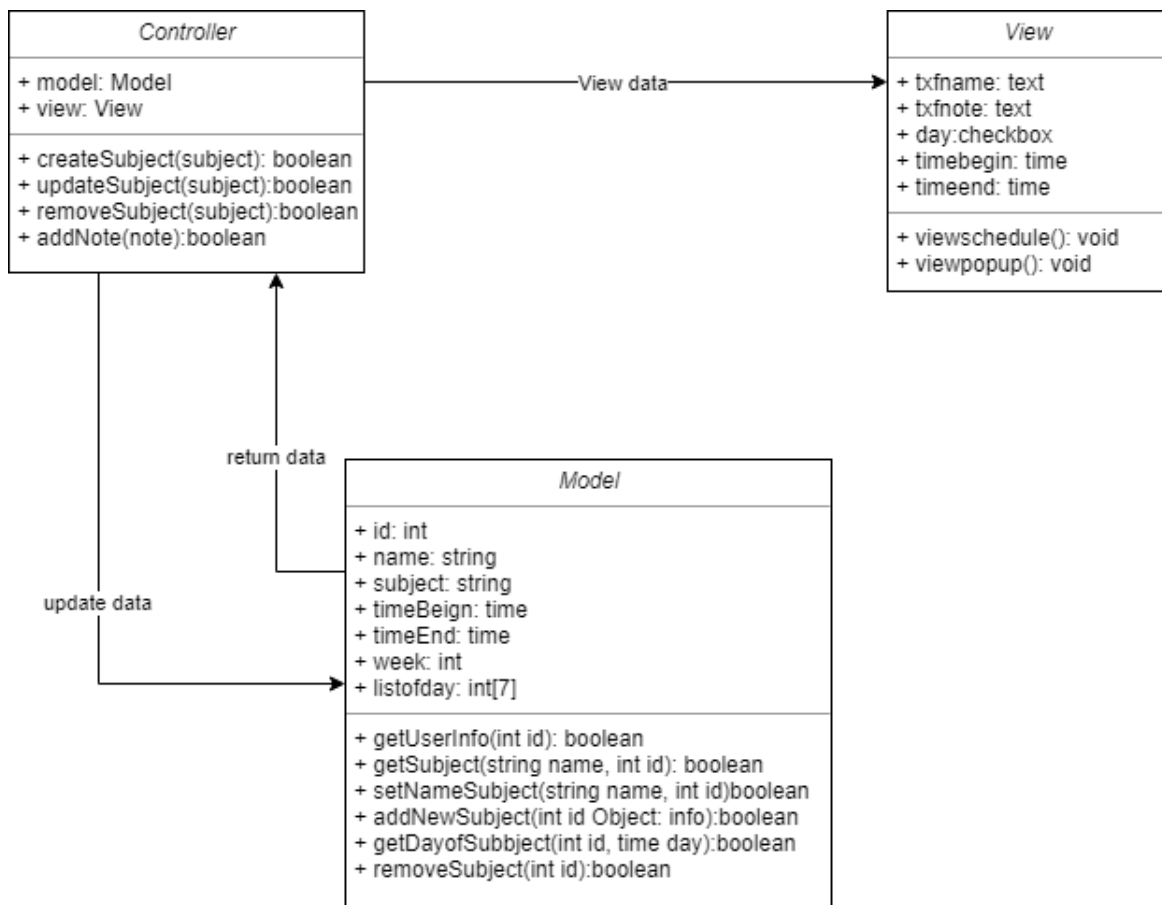


Hình 7.23: MVC - Todolist

8 Class diagram cho hệ thống

8.1 Class diagram cho chức năng quản lý lịch học trên trường

8.1.1 Class diagram



Hình 8.24: Class diagram cho chức năng quản lý lịch học.

8.1.2 Đặc tả các phương thức cho chức năng quản lý lịch học

8.1.2.a Views

- `viewschedule()`: Hiển thị thông tin lịch học hiện tại
- `viewpopup()`: Hiển thị thông tin pop up tạo và cập nhập môn học.

8.1.2.b Controllers

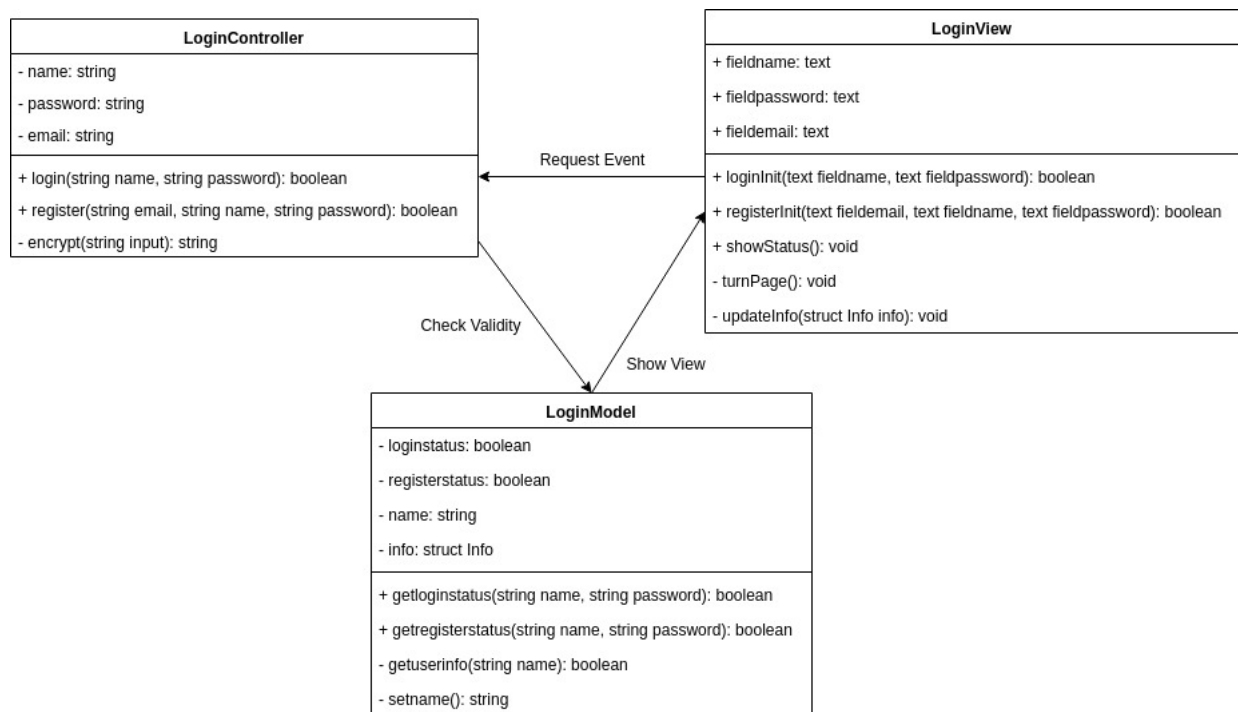
- `loadView()`: Lấy thông tin từ database và gửi event sang views.
- `createSubject()`: Thực hiện tạo một môn học mới rồi gửi thông tin xuống model để cập nhập
- `updateSubject()`: Thực hiện thay đổi thông tin của lịch học đã lưu và gửi đến model.
- `removeSubject()`: Gửi event xóa bản ghi trong database về môn học muốn xóa
- `addNote()`: Thực hiện tạo ghi chú và gửi thông tin xuống database.

8.1.2.c Model

- `getUserInfo()`: Lấy thông tin user từ database sau khi đăng nhập.
- `getSubject()`: Lấy thông tin môn học của user.
- `getNameSubject()`: Lấy thông tin môn học thông qua tên môn học hoặc id.
- `addNewSubject()`: Tạo một bản record mới trong database về môn học.
- `getDayofSubject()`: Lấy thông tin môn học trong database theo ngày.
- `removeSubject()`: Cập nhập database.
- `updateSubject()`: Thay đổi dữ liệu trong database theo môn học.

8.2 Class diagram cho chức năng cho trang đăng ký đăng nhập, khung thông báo, khung và hộp tin nhắn

8.2.1 Class diagram cho chức năng trang đăng ký đăng nhập



Class LoginView:

. loginInit(text fieldname, text fieldpassword): boolean

Arguments: Trường tên đăng nhập và mật khẩu

Return: Boolean

Description: Kích hoạt đăng nhập vào tài khoản

Pre-condition: Hai trường tên đăng nhập và mật khẩu được điền đầy đủ

Post-condition: Tài khoản được đăng nhập thành công

Exceptions: Không.

Flow of Events

- Nhập thông tin các trường tên đăng nhập và mật khẩu.
- Kích hoạt kiểm tra các trường tên đăng nhập và mật khẩu.
- Kích hoạt chuyển giao diện nếu đăng nhập thành công.

. registerInit(text fieldmail, text fieldpassword): boolean

Arguments: Trường email và mật khẩu

Return: Boolean

Description: Kích hoạt đăng kí tài khoản

Pre-condition: Hai trường email và mật khẩu được điền đầy đủ

Post-condition: Tài khoản được đăng kí thành công

Exceptions: Không.

Flow of Events

- (a) Nhập thông tin các trường email và mật khẩu.
- (b) Kích hoạt kiểm tra xác nhận các trường email và mật khẩu.
- (c) Kích hoạt chuyển giao diện nếu đăng kí thành công.

. showStatus(): void

Arguments: Đã kích hoạt đăng kí tài khoản mới

Return: Void

Description: Kiểm tra trạng thái gmail đã được xác nhận hay chưa.

Pre-condition: Trạng thái kiểm tra xác nhận gmail được kích hoạt

Post-condition: Kiểm tra gmail được xác nhận hay chưa.

Exceptions: Không.

Flow of Events

- (a) Kích hoạt chức năng xác nhận gmail.
- (b) Tiến hành kiểm tra xác nhận gmail..
- (c) Kích hoạt tài khoản nếu gmail đã xác nhận thành công.

. turnPage(): void

Arguments: Đăng kí tài khoản thành công

Return: Void

Description: Chuyển đổi giao diện sang trang mới.

Pre-condition: Kích hoạt thay đổi giao diện và xác nhận có thông tin người dùng từ cơ sở dữ liệu.

Post-condition: Giao diện được thay đổi thành công.

Exceptions: Không.

Flow of Events

- (a) Xác nhận đã có thông tin người dùng từ cơ sở dữ liệu.
- (b) Kích hoạt chuyển đổi giao diện.
- (c) Tiến hành chuyển đổi giao diện.

. updateInfo(struct Info info): void

Arguments: Thông tin người dùng từ cơ sở dữ liệu

Return: Void

Description: Trả về thông tin người dùng từ cơ sở dữ liệu sau khi đã đăng nhập/đăng kí thành công.

Pre-condition: Đăng nhập/đăng kí thành công.

Post-condition: Trả về thông tin người dùng từ cơ sở dữ liệu.

Exceptions: Không.

Flow of Events

- (a) Xác nhận đăng nhập thành công.
- (b) Tiến hành lấy dữ liệu người dùng từ cơ sở dữ liệu.

Class LoginControl:

. login(string name, string password): boolean

Arguments: Tên đăng nhập và mật khẩu.

Return: Boolean

Description: Tiến hành xác nhận đăng nhập vào tài khoản

Pre-condition: Tên đăng nhập và mật khẩu được có đầy đủ

Post-condition: Trả về kết quả tìm kiếm xác nhận đăng nhập thành công hay không

Exceptions: Không.

Flow of Events

- (a) Lấy thông tin các biến tên đăng nhập và mật khẩu.
- (b) Tiến hành tìm kiếm các trường tên đăng nhập và mật khẩu trong cơ sở dữ liệu.
- (c) Xác nhận kết quả.

. register(string email, string name, string password): boolean

Arguments: Email, tên đăng nhập và mật khẩu.

Return: Boolean

Description: Tiến hành xác nhận đăng kí tài khoản

Pre-condition: Email, tên đăng nhập và mật khẩu được có đầy đủ

Post-condition: Trả về kết quả tìm kiếm xác nhận đăng kí thành công hay không

Exceptions: Không.

Flow of Events

- (a) Lấy thông tin các biến email, tên đăng nhập và mật khẩu.
- (b) Tiến hành tìm kiếm các trường tên đăng nhập và mật khẩu trong cơ sở dữ liệu và xác nhận email.
- (c) Xác nhận kết quả.

. encrypt(string input): string

Arguments: Dữ liệu lưu trong các biến tên đăng nhập, mật khẩu và email.

Return: Boolean

Description: Tiến hành mã hóa các trường thông tin

Pre-condition: Email, tên đăng nhập và mật khẩu được có đầy đủ

Post-condition: Trả về chuỗi giá trị sau khi đã được mã hóa

Exceptions: Không.

Flow of Events

- (a) Lấy thông tin chuỗi đầu vào.
- (b) Tiến hành mã hóa chuỗi.
- (c) Xác nhận và trả lại kết quả.

Class LoginModel:

. getloginstatus(string name, string password): boolean

Arguments: Tên đăng nhập và mật khẩu.

Return: Boolean

Description: Tiến hành xác nhận trạng thái đăng nhập vào tài khoản

Pre-condition: Tên đăng nhập và mật khẩu được có đầy đủ và đã được mã hóa

Post-condition: Tìm kiếm xác nhận đăng nhập thành công hay không

Exceptions: Không.

Flow of Events

- (a) Tiến hành tìm kiếm các trường tên đăng nhập và mật khẩu trong cơ sở dữ liệu.
- (b) Xác nhận kết quả.

. getregisterstatus(string name, string password): boolean

Arguments: Trường tên đăng nhập và mật khẩu

Return: Boolean

Description: Kích hoạt đăng kí tài khoản và xác nhận tên đăng nhập đã tồn tại hay chưa

Pre-condition: Hai trường tên đăng nhập và mật khẩu được điền đầy đủ

Post-condition: Tìm kiếm và xác nhận tài khoản được đăng kí thành công

Exceptions: Không.

Flow of Events

- (a) Kích hoạt kiểm tra xác nhận các trường tên đăng nhập và mật khẩu.
- (b) Xác nhận kết quả đăng kí thành công.

. getuserinfo(string name): boolean

Arguments: Sau khi đã đăng kí/ đăng nhập thành công

Return: Boolean

Description: Kích hoạt lấy thông tin user tương ứng

Pre-condition: Trường tên đăng nhập đã được xác định

Post-condition: Tìm kiếm và trả về thông tin người dùng

Exceptions: Không.

Flow of Events

- (a) Kích hoạt kiểm tra và lấy thông tin người dùng tương ứng.
- (b) Xác nhận lấy thông tin thành công.

. setname(): string

Arguments: Sau khi đã đăng kí thành công

Return: Boolean

Description: Đặt tên đăng nhập

Pre-condition: Trường tên đăng nhập đã được nhập

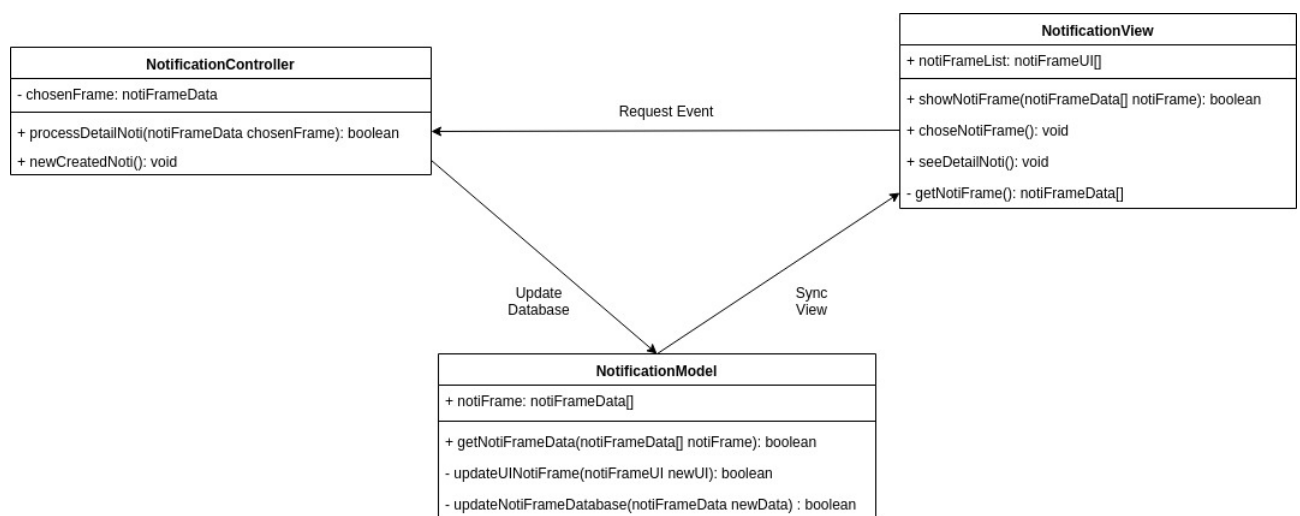
Post-condition: Thêm người dùng với tên đăng nhập mới và mật khẩu tương ứng

Exceptions: Không.

Flow of Events

- (a) Kích hoạt kiểm tra và thêm mới tên đăng nhập.
- (b) Xác nhận thêm thông tin thành công.

8.2.2 Class diagram cho chức năng khung thông báo



Class NotificationView:

. showNotiFrame(notiFrameData[] notiFrame): boolean

Arguments: Chọn xem khung thông báo

Return: Boolean

Description: Hiển thị các nội dung trong khung thông báo

Pre-condition: Hiển thị nội dung trong khung thông báo được chọn

Post-condition: Hiển thị toàn bộ nội dung thông báo

Exceptions: Không.

Flow of Events

- (a) Kích hoạt hiển thị các nội dung trong khung thông báo.
- (b) Hiển thị các trường thông tin lên khung thông báo.

(c) Xác nhận hiển thị thành công

. choseNotiFrame(): void

Arguments: Chọn xem thông tin chi tiết một thông báo

Return: Void

Description: Kích hoạt hiển thị các nội dung chi tiết trong một thông báo

Pre-condition: Hiển thị chi tiết nội dung trong một thông báo được chọn

Post-condition: Kích hoạt hiển thị chi tiết nội dung trong thông báo được chọn

Exceptions: Không.

Flow of Events

(a) Kích hoạt chọn hiển thị chi tiết một nội dung trong khung thông báo.

(b) Kích hoạt hiển thị các trường thông tin liên quan.

. seeDetailNoti(): void

Arguments: Xem thông tin chi tiết một thông báo được kích hoạt

Return: Void

Description: Hiển thị các nội dung chi tiết trong một thông báo

Pre-condition: Một khung thông báo cụ thể được chọn

Post-condition: Hiển thị chi tiết nội dung trong thông báo

Exceptions: Không.

Flow of Events

(a) Gọi các tiến trình xử lý tác vụ xem chi tiết thông báo.

(b) Hiển thị các trường thông tin liên quan.

. getNotiFrame(): notiFrameData[]

Arguments: Xem khung thông báo được kích hoạt

Return: notiFrameData[]

Description: Cập nhật các nội dung trong khung thông báo

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Cập nhật các nội dung trong khung thông báo

Exceptions: Không.

Flow of Events

(a) Lấy các trường thông tin trong cơ sở dữ liệu.

(b) Thêm mới hoặc tái hiển thị lại các trường thông tin liên quan.

Class NotificationController:

. processDetailNoti(notiFrameData chosenFrame): boolean

Arguments: Xem thông tin chi tiết một khung thông báo đã được chọn

Return: Void

Description: Xử lý việc hiển thị các nội dung chi tiết trong thông báo được chọn

Pre-condition: Một khung thông báo cụ thể được chọn

Post-condition: Xử lý việc hiển thị chi tiết nội dung trong thông báo

Exceptions: Không.

Flow of Events

- (a) Kiểm tra tiến trình xử lý tác vụ xem chi tiết thông báo được gọi.
- (b) Xử lý các trường thông tin liên quan trong cơ sở dữ liệu.

. newCreateNoti(): void

Arguments: Khi có một sự kiện liên quan đến người dùng xảy ra

Return: Void

Description: Tạo thông báo cho sự kiện liên quan đến những người dùng cụ thể

Pre-condition: Một sự kiện liên quan đến người dùng được kích hoạt

Post-condition: Thông báo cho các người dùng cụ thể được khởi tạo

Exceptions: Không.

Flow of Events

- (a) Gọi các tiến trình xử lý tác vụ tạo mới thông báo.
- (b) Cập nhật thông báo mới trong cơ sở dữ liệu người dùng liên quan.

Class NotificationModel:

. getNotiFrameData(notiFrameData[] notiFrame): boolean

Arguments: Người dùng đăng nhập vào tài khoản, việc lấy thông tin được kích hoạt

Return: notiFrameData[]

Description: Tải lên ứng dụng các nội dung trong khung thông báo từ cơ sở dữ liệu

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Lấy các trường nội dung trong khung thông báo

Exceptions: Không.

Flow of Events

- (a) Lấy các trường thông tin trong cơ sở dữ liệu.
- (b) Gọi các tiến trình giao diện cập nhật lại thông tin.

. updateUINotiFrame(notiFrameUI newUI): boolean

Arguments: Người dùng đăng nhập vào tài khoản, việc lấy thông tin được kích hoạt

Return: Boolean

Description: Tải lên ứng dụng các nội dung trong khung thông báo từ cơ sở dữ liệu

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Lấy các trường nội dung trong khung thông báo

Exceptions: Không.

Flow of Events

- (a) Lấy các trường thông tin trong cơ sở dữ liệu.
- (b) Gọi các tiến trình giao diện cập nhật lại thông tin.

. updateNotiFrameDataBase(notiFrameData newData): boolean

Arguments: Hệ thống cập nhật thêm các thông báo mới

Return: Boolean

Description: Thêm mới các thông báo vào cơ sở dữ liệu

Pre-condition: Thông báo cho các người dùng cụ thể đã được khởi tạo

Post-condition: Cập nhật các thông báo mới được tạo vào cơ sở dữ liệu

Exceptions: Không.

Flow of Events

- (a) Cập nhật các thông báo mới được tạo vào cơ sở dữ liệu.
- (b) Kiểm tra và tiến hành xác nhận để đảm bảo tính toàn vẹn dữ liệu.

8.2.3 Class diagram cho chức năng khung và hộp tin nhắn

Class **MessageView**:

. **showMessageFrame**(messFrameList[] messFrameUI): boolean **Arguments:** Chọn xem khung tin nhắn

Return: Boolean

Description: Hiển thị các nội dung trong khung tin nhắn

Pre-condition: Hiển thị nội dung trong khung tin nhắn được kích hoạt

Post-condition: Hiển thị toàn bộ nội dung trong khung tin nhắn

Exceptions: Không.

Flow of Events

- (a) Kích hoạt hiển thị các nội dung trong khung tin nhắn.
- (b) Hiển thị các trường thông tin của các hộp tin nhắn lên khung tin nhắn.
- (c) Xác nhận hiển thị thành công

. **showMessageBox**(messBoxList[] messBoxUI): boolean **Arguments:** Chọn xem chi tiết hộp tin nhắn

Return: Boolean

Description: Hiển thị các nội dung trong các hộp tin nhắn được chọn

Pre-condition: Hiển thị nội dung trong hộp tin nhắn được kích hoạt

Post-condition: Hiển thị toàn bộ nội dung trong hộp tin nhắn

Exceptions: Không.

Flow of Events

- (a) Kích hoạt hiển thị các nội dung trong các hộp tin nhắn.
- (b) Hiển thị các trường thông tin của các tin nhắn lên hộp tin nhắn.
- (c) Xác nhận hiển thị thành công

. **choseMessageFrame**(): void **Arguments:** Chọn xem chi tiết hộp tin nhắn

Return: Void

Description: Khi một thành phần trong khung tin nhắn được chọn thì tiến hành mở hộp tin nhắn tương ứng

Pre-condition: Một thành phần trong khung tin nhắn được chọn

Post-condition: Hiển thị hộp tin nhắn tương ứng

Exceptions: Không.

Flow of Events

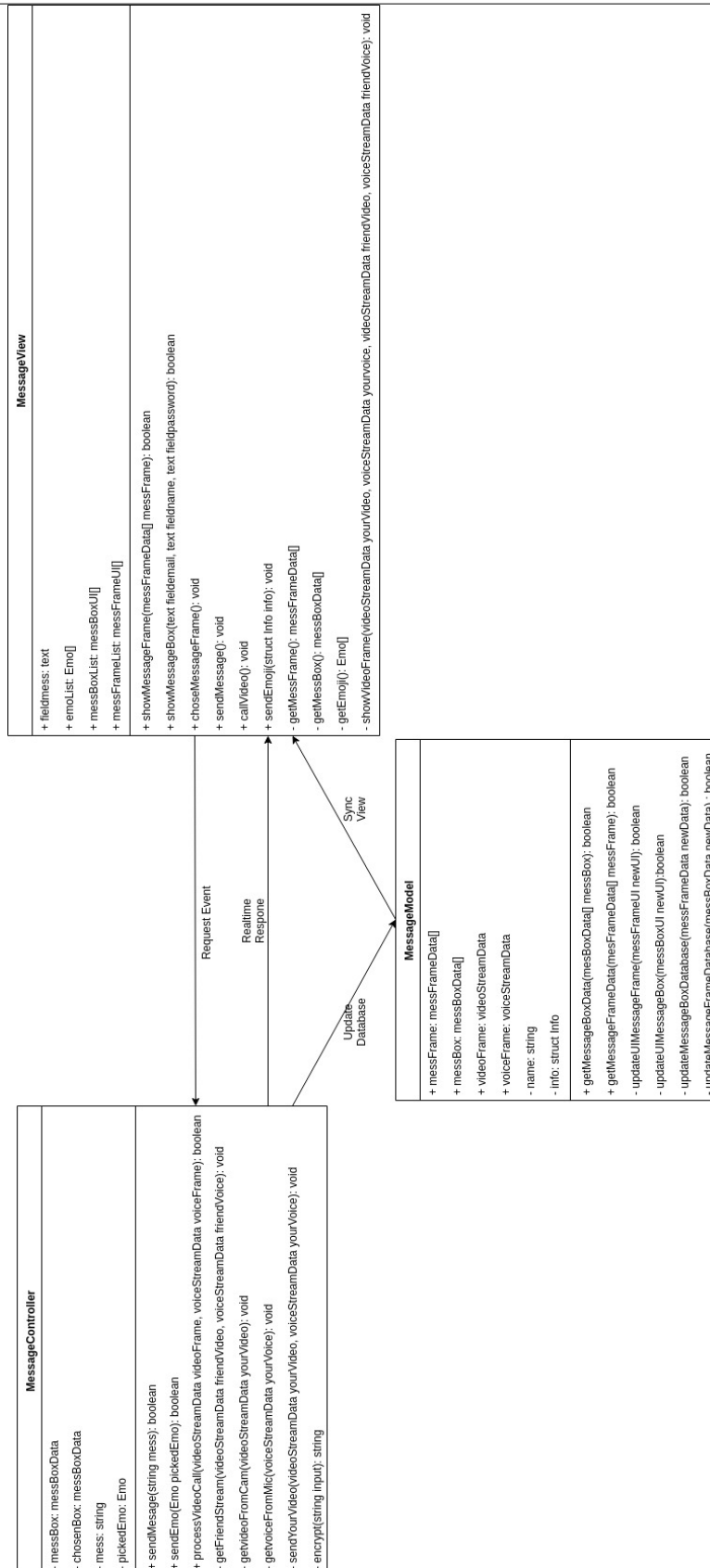
- (a) Hiển thị hộp tin nhắn tương ứng với khung tin nhắn được chọn.

. **sendMessage**(): void **Arguments:** Truyền tải tin nhắn

Return: Void

Description: Gửi tin nhắn

Pre-condition: Trường tin nhắn được điền



Post-condition: Tin nhắn được gửi

Exceptions: Không.

Flow of Events

(a) Kiểm tra trường tin nhắn được điền. Tiến hành gửi tin nhắn.

. callVideo(): void **Arguments:** Truyền tải video

Return: Void

Description: Gửi và nhận video

Pre-condition: Tính năng đàm thoại video được kích hoạt

Post-condition: Tiến hành gửi và nhận video

Exceptions: Không.

Flow of Events

(a) Kiểm tra thành phần webcam và mic. Tiến hành truyền tải video.

. sendEmoji(struct Info info): void **Arguments:** Truyền tải Emoji

Return: Void

Description: Gửi emoji

Pre-condition: Emoji đã được chọn

Post-condition: Emoji được gửi

Exceptions: Không.

Flow of Events

(a) Kiểm tra emoji được chọn. Tiến hành gửi emoji.

. getMessFrame(): messFrameData[] **Arguments:** Xem khung tin nhắn được kích hoạt

Return: notiFrameUI[]

Description: Cập nhật các nội dung trong khung tin nhắn

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Cập nhật các nội dung trong khung tin nhắn

Exceptions: Không.

Flow of Events

(a) Lấy các trường thông tin trong cơ sở dữ liệu.

(b) Thêm mới hoặc tái hiển thị lại các trường thông tin liên quan.

. getMessBox(): messBoxUI[] **Arguments:** Xem hộp tin nhắn được kích hoạt

Return: notiFrameUI[]

Description: Cập nhật các nội dung trong hộp tin nhắn

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Cập nhật các nội dung trong hộp tin nhắn

Exceptions: Không.

Flow of Events

- (a) Lấy các trường thông tin trong cơ sở dữ liệu.
- (b) Thêm mới hoặc tái hiển thị lại các trường thông tin liên quan.

. getEmoji(): Emo[] **Arguments:** Xem hộp tin nhắn được kích hoạt
Return: Emo[]

Description: Cập nhật các loại emoji

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Cập nhật các loại emoji trong hộp tin nhắn

Exceptions: Không.

Flow of Events

- (a) Lấy các trường thông tin trong cơ sở dữ liệu.
- (b) Thêm mới hoặc tái hiển thị lại các trường thông tin liên quan.

. showVideoFrame(videoStreamData yourVideo, voiceStreamData yourvoice, videoStreamData friendVideo, voiceStreamData friendVoice): void **Arguments:** Truyền tải videocall kích hoạt
Return: Void

Description: Hiển thị và phát các video và voice được truyền tải.

Pre-condition: Truyền tải videocall được kích hoạt

Post-condition: Hiển thị và phát các video và voice được truyền tải.

Exceptions: Không.

Flow of Events

- (a) Truyền tải videocall được kích hoạt.
- (b) Hiển thị và phát các video và voice được truyền tải.

Class MessageController:

. sendMessage(string mess): boolean **Arguments:** Xử lý truyền tải tin nhắn
Return: Boolean

Description: Xử lý đồng bộ gửi tin nhắn

Pre-condition: Trường tin nhắn có giá trị

Post-condition: Tin nhắn được gửi

Exceptions: Không.

Flow of Events

- (a) Kiểm tra trường tin nhắn có giá trị. Tiến hành đồng bộ gửi tin nhắn.
- (b) Xác nhận thành công

. sendEmo(Emo pickedEmo): boolean **Arguments:** Xử lý truyền tải Emoji
Return: Boolean

Description: Xử lý đồng bộ gửi emoji

Pre-condition: Trường emoji có giá trị

Post-condition: Emoji được gửi

Exceptions: Không.

Flow of Events

- (a) Kiểm tra emoji có giá trị. Tiến hành đồng bộ gửi emoji.
- (b) Xác nhận thành công

. processVideoCall(videoStreamData videoFrame, voiceStreamData voiceFrame): boolean **Arguments:** Xử lý truyền tải Video Call

Return: Boolean

Description: Xử lý đồng bộ truyền tải Video Call

Pre-condition: Nhận được dữ liệu âm thanh và hình ảnh

Post-condition: Dữ liệu được truyền tải

Exceptions: Không.

Flow of Events

- (a) Kiểm tra dữ liệu âm thanh và hình ảnh. Tiến hành đồng bộ truyền tải.
- (b) Xác nhận thành công

. getFriendStream(videoStreamData friendVideo, voiceStreamData friendVoice): void **Arguments:** Lấy dữ liệu video voice từ bạn chat

Return: Void

Description: Lấy dữ liệu video voice từ bạn chat

Pre-condition: Lấy dữ liệu video voice từ bạn chat được kích hoạt

Post-condition: Nhận được dữ liệu âm thanh và hình ảnh

Exceptions: Không.

Flow of Events

- (a) Kiểm tra kết nối. Tiến hành lấy dữ liệu.

. getvideoFromCam(videoStreamData yourVideo): void **Arguments:** Lấy dữ liệu video từ webcam

Return: Void

Description: Lấy dữ liệu video từ webcam

Pre-condition: Lấy dữ liệu video từ webcam được kích hoạt

Post-condition: Nhận được dữ liệu hình ảnh

Exceptions: Không.

Flow of Events

- (a) Kiểm tra kết nối webcam. Tiến hành lấy dữ liệu.

. getvoiceFromMic(voiceStreamData yourVoice): void **Arguments:** Lấy dữ liệu voice từ mic

Return: Void

Description: Lấy dữ liệu voice từ mic

Pre-condition: Lấy dữ liệu voice từ mic được kích hoạt

Post-condition: Nhận được dữ liệu âm thanh

Exceptions: Không.

Flow of Events

(a) Kiểm tra kết nối mic. Tiến hành lấy dữ liệu.

. sendYourVideo(videoStreamData yourVideo, voiceStreamData yourVoice): void **Arguments:** Gửi dữ liệu video voice cho bạn chat

Return: Void

Description: Gửi dữ liệu video voice cho bạn chat

Pre-condition: Gửi dữ liệu video voice cho bạn chat được kích hoạt

Post-condition: Dữ liệu âm thanh và hình ảnh được gửi

Exceptions: Không.

Flow of Events

(a) Kiểm tra kết nối. Tiến hành gửi dữ liệu.

. encrypt(string input): string **Arguments:** Dữ liệu lưu trong các biến tên đăng nhập, mật khẩu và email.

Return: Boolean

Description: Tiến hành mã hóa các trường thông tin

Pre-condition: Email, tên đăng nhập và mật khẩu được có đầy đủ

Post-condition: Trả về chuỗi giá trị sau khi đã được mã hóa

Exceptions: Không.

Flow of Events

(a) Lấy thông tin chuỗi đầu vào.

(b) Tiến hành mã hóa chuỗi.

(c) Xác nhận và trả lại kết quả.

Class MessageModel:

. getMessageBoxData(mesBoxData[] messBox): boolean **Arguments:** Người dùng đăng nhập vào tài khoản, việc lấy thông tin được kích hoạt

Return: boolean

Description: Tải lên ứng dụng các nội dung trong hộp tin nhắn từ cơ sở dữ liệu

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Lấy các trường nội dung trong hộp tin nhắn

Exceptions: Không.

Flow of Events

(a) Lấy các trường thông tin trong cơ sở dữ liệu.

(b) Gọi các tiến trình giao diện cập nhật lại thông tin.

. getMessageFrameData(mesFrameData[] messFrame): boolean **Arguments:** Người dùng đăng nhập vào tài khoản, việc lấy thông tin được kích hoạt

Return: boolean

Description: Tải lên ứng dụng các nội dung trong khung tin nhắn từ cơ sở dữ liệu

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Lấy các trường nội dung trong khung tin nhắn

Exceptions: Không.

Flow of Events

- (a) Lấy các trường thông tin trong cơ sở dữ liệu.
- (b) Gọi các tiến trình giao diện cập nhật lại thông tin.

. updateUIMessageFrame(messFrameUI newUI): boolean **Arguments:** Cập nhật hiển thị các trường thông tin trong khung tin nhắn

Return: Boolean

Description: Cập nhật hiển thị các nội dung trong khung tin nhắn từ cơ sở dữ liệu.

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Cập nhật hiển thị các trường nội dung trong khung tin nhắn

Exceptions: Không.

Flow of Events

- (a) Lấy các trường thông tin trong cơ sở dữ liệu.
- (b) Gọi các tiến trình giao diện cập nhật lại thông tin.

. updateUIMessageBox(messBoxUI newUI):boolean **Arguments:** Cập nhật hiển thị các trường thông tin trong hộp tin nhắn

Return: Boolean

Description: Cập nhật hiển thị các nội dung trong hộp tin nhắn từ cơ sở dữ liệu.

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Cập nhật hiển thị các trường nội dung trong hộp tin nhắn

Exceptions: Không.

Flow of Events

- (a) Lấy các trường thông tin trong cơ sở dữ liệu.
- (b) Gọi các tiến trình giao diện cập nhật lại thông tin.

. updateMessageBoxDatabase(messFrameData newData): boolean **Arguments:** Cập nhật các trường thông tin trong hộp tin nhắn trong cơ sở dữ liệu

Return: Boolean

Description: Cập nhật các nội dung trong hộp tin nhắn trong cơ sở dữ liệu.

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Cập nhật các trường nội dung trong hộp tin nhắn trong cơ sở dữ liệu

Exceptions: Không.

Flow of Events

- (a) Kiểm tra các trường thông tin.

- (b) Gọi các tiến trình cập nhật lại thông tin trong cơ sở dữ liệu.
- (c) Kiểm tra tính toàn vẹn dữ liệu

. updateMessageFrameDatabase(messBoxData newData) : boolean **Arguments:** Cập nhật các trường thông tin trong khung tin nhắn trong cơ sở dữ liệu

Return: Boolean

Description: Cập nhật các nội dung trong khung tin nhắn trong cơ sở dữ liệu.

Pre-condition: Người dùng trong trạng thái đã đăng nhập

Post-condition: Cập nhật các trường nội dung trong khung tin nhắn trong cơ sở dữ liệu

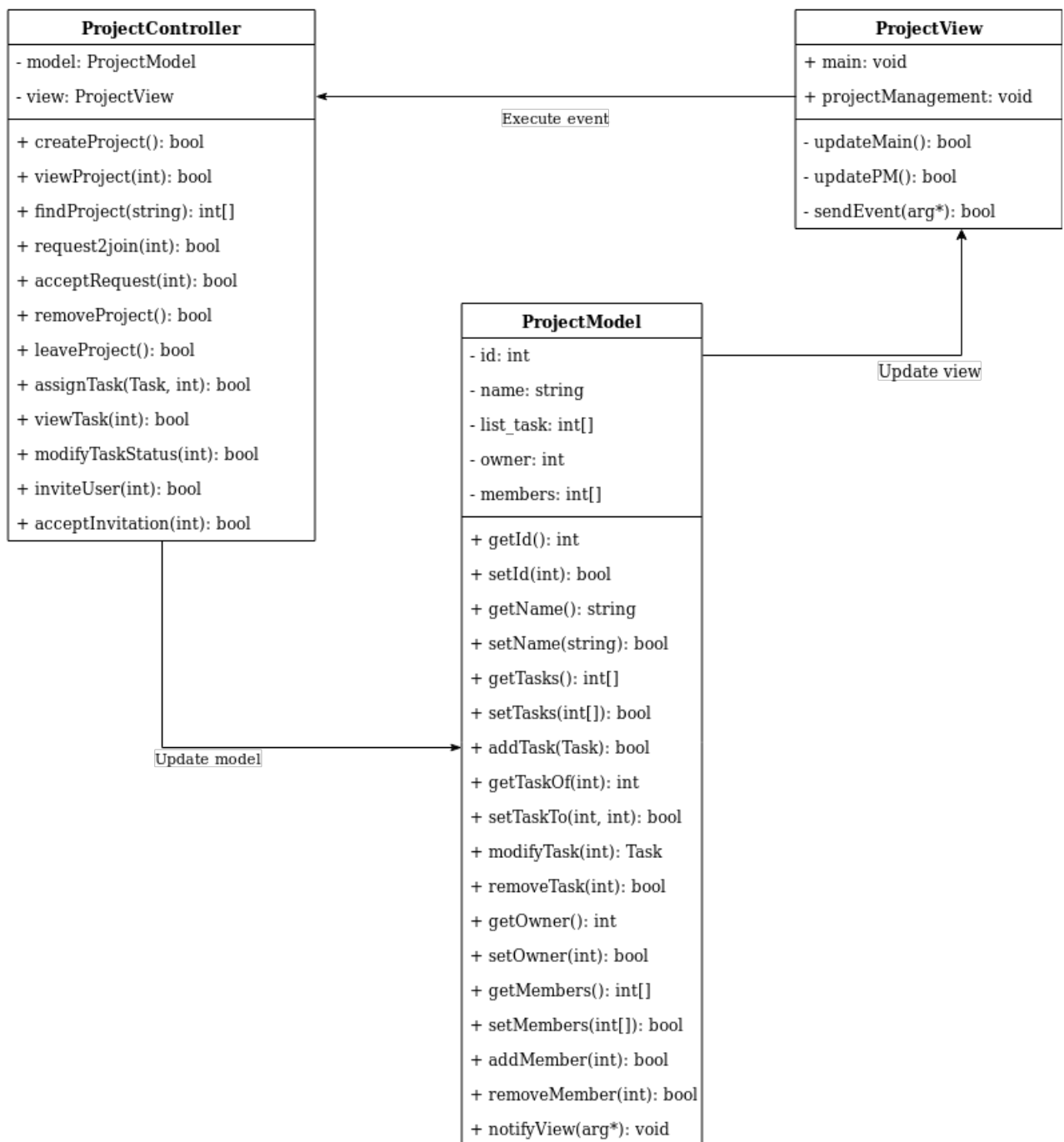
Exceptions: Không.

Flow of Events

- (a) Kiểm tra các trường thông tin.
- (b) Gọi các tiến trình cập nhật lại thông tin trong cơ sở dữ liệu.
- (c) Kiểm tra tính toàn vẹn dữ liệu

8.3 Class diagram và đặc tả phương thức cho chức năng quản lý công việc nhóm

Class diagram cho chức năng Công việc nhóm



Hình 8.25: Class diagram - Quản lý công việc nhóm

Đặc tả các phương thức cho chức năng Công việc nhóm

ProjectView

- updateMain(): Update frame chính của tab công việc nhóm
- updatePM(): Update frame quản lý công việc của tab công việc nhóm
- sendEvent(arg*): Gửi user event tới ProjectController để xử lý

ProjectModel

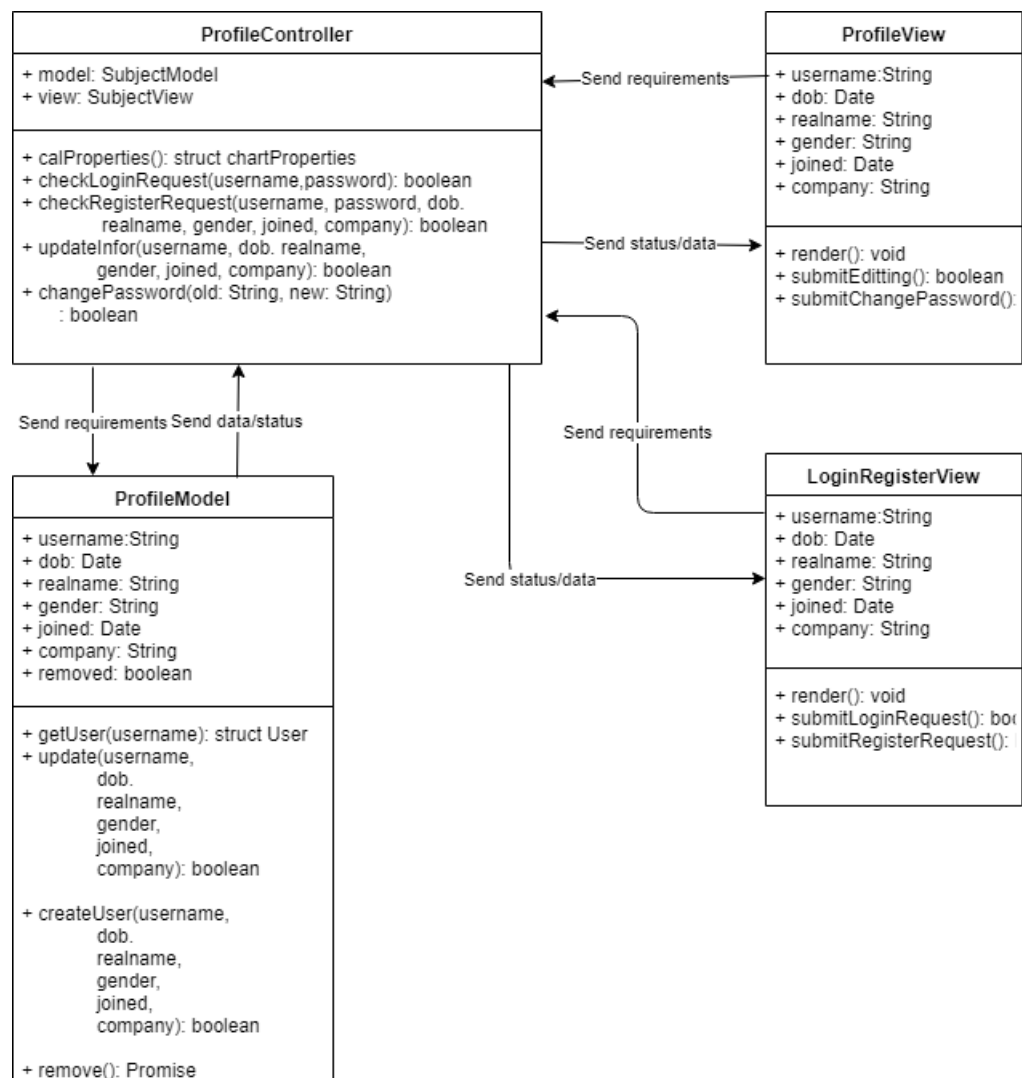
- getId(): Trả về id của project
- setId(): Set id cho project
- getName(): Trả về name của project
- setName(): Set name cho project
- getTasks(): Trả về danh sách id các task của project
- setTasks(): Set các task với id tương ứng cho project
- addTask(): Thêm task cho project với input là một Task object
- getTaskOf(): Trả về task của một member tương ứng với id input
- setTaskTo(): Set task cho member với id input tương ứng
- modifyTask(): Sửa đổi một Task object
- removeTask(): Xóa task khỏi project
- getOwner(): Trả về owner của project
- setOwner(): Set owner cho project
- getMembers(): Trả về danh sách id các member của project
- setMembers(): Set các member với id tương ứng cho project
- addMember(): Thêm member cho project với input là một id của user
- removeMember(): Xóa member có id tương ứng input khỏi project
- notifyView(): Thông báo đến View các thay đổi tương ứng với action

ProjectController

- `createProject()`: Tạo một project mới
- `viewProject()`: Xem chi tiết 1 project từ list các project đã tham gia hoặc đang tìm kiếm
- `findProject()`: Tìm và trả về danh sách các id các project public theo từ khóa tìm kiếm input
- `request2join()`: Yêu cầu tham gia 1 project public
- `acceptRequest()`: Chấp nhận yêu cầu tham gia project của 1 user khác
- `removeProject()`: Xóa project
- `leaveProject()`: Rời project
- `assignTask()`: Phân công 1 task cho một member với Task object và user id tương ứng
- `viewTask()`: Xem 1 task nào đó
- `modifyTaskStatus()`: Chính sửa trạng thái của 1 task trong quyền của user
- `inviteUser()`: Mời một user nào đó cộng tác trong project
- `acceptInvitation()`: Chấp nhận lời mời cộng tác của 1 user khác

8.4 Class diagram cho trang chủ và thông tin cá nhân

8.4.1 Class diagram



Hình 8.26: Class diagram cho nhóm chức năng thông tin cá nhân

8.4.2 Đặc tả phương thức

Class ProfileView

Arguments:	None
Return:	Void
Description:	Hiển thị các thành phần giao diện trang cá nhân, các thông tin
Pre-condition:	None
Post-condition:	Các thành phần hiển thị chính xác
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Hiển thị dữ liệu mỗi lần load trang2. Vẽ lại biểu đồ

Bảng 23: Method render()

render():

Arguments:	None
Return:	Boolean
Description:	Gửi yêu cầu thay đổi thông tin từ phía người dùng
Pre-condition:	None
Post-condition:	Yêu cầu được gửi lên server
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Lấy thông tin nhập vào từ giao diện người dùng2. Đóng gói các thông tin vào data struct3. Gửi gói yêu cầu đến controller4. Bắt đầu chờ tin trả về từ controller

Bảng 24: Method submitEditing()

submitEditing():

Arguments:	None
Return:	Boolean
Description:	Gửi yêu cầu thay đổi mật khẩu
Pre-condition:	None
Post-condition:	Yêu cầu được gửi lên server
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Lấy mật khẩu nhập vào từ giao diện người dùng2. Gửi gói yêu cầu đến controller3. Bắt đầu chờ tin trả về từ controller

Bảng 25: Method submitChangePassword()

submitChangePassword(): Class LoginRegisterView

Arguments:	None
Return:	Void
Description:	Hiển thị các thành phần giao diện trang cá nhân, các thông tin
Pre-condition:	None
Post-condition:	Các thành phần hiển thị chính xác
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Hiển thị dữ liệu mỗi lần load trang

Bảng 26: Method render()

render():

Arguments:	None
Return:	true nếu gửi thành công, ngược lại false
Description:	Gửi yêu cầu đăng nhập tới máy chủ
Pre-condition:	None
Post-condition:	Yêu cầu được gửi chính xác và bảo mật
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Lấy dữ liệu từ giao diện2. Mã hóa mật khẩu3. Gửi gói yêu cầu đến Controller

Bảng 27: Method render()

submitLoginRequest():

Arguments:	None
Return:	true nếu gửi thành công, ngược lại false
Description:	Gửi yêu cầu đăng kí tới máy chủ
Pre-condition:	None
Post-condition:	Yêu cầu được gửi chính xác và bảo mật
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Lấy dữ liệu từ giao diện2. Mã hóa mật khẩu3. Gửi gói yêu cầu đến Controller

Bảng 28: Method submitRegisterRequest()

submitRegisterRequest(): Class ProfileController

Arguments:	None
Return:	struct chartProperties chứa gói các thông số để vẽ biểu đồ
Description:	Tính toán các thông số để vẽ biểu đồ
Pre-condition:	None
Post-condition:	Các thông số được tính toán và trả về
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Gửi yêu cầu dữ liệu đến model2. Tính toán thông số3. Trả về thông số

Bảng 29: Method calProperties()

calProperties():

Arguments:	username:String,dob: Date,realname: String, gender: String, joined: Date, company: String
Return:	True nếu yêu cầu là hợp lệ, ngược lại là false
Description:	Kiểm tra điều kiện update dữ liệu
Pre-condition:	None
Post-condition:	Các điều kiện được kiểm tra đầy đủ và chính xác
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Kiểm tra các điều kiện trên thông số2. Nếu thành công, gửi yêu cầu cập nhật tới Model3. Thông báo kết quả cho View

Bảng 30: Method updateInfor()

updateInfor():

Arguments:	old:String, new:String
Return:	True nếu yêu cầu là hợp lệ, ngược lại là false
Description:	Kiểm tra điều kiện thay đổi mật khẩu
Pre-condition:	None
Post-condition:	Các điều kiện được kiểm tra đầy đủ và chính xác
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Kiểm tra các điều kiện trên mật khẩu2. Nếu thành công, gửi yêu cầu cập nhật tới Model3. Thông báo kết quả cho View

Bảng 31: Method changePassword()

changePassword():

Arguments:	username:String,password:String
Return:	True nếu yêu cầu là hợp lệ, ngược lại là false
Description:	Kiểm tra điều kiện đăng nhập
Pre-condition:	None
Post-condition:	Các điều kiện được kiểm tra đầy đủ và chính xác
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Kiểm tra các điều kiện trên thông số2. Gửi yêu cầu lấy thông tin User đến model3. Trả về kết quả yêu cầu đăng nhập có hợp lệ hay không

Bảng 32: Method checkLoginRequest()

checkLoginRequest():

Arguments:	username, password, dob, realname, gender, joined, company
Return:	True nếu yêu cầu là hợp lệ, ngược lại là false
Description:	Kiểm tra yêu cầu đăng kí
Pre-condition:	None
Post-condition:	Các điều kiện được kiểm tra đầy đủ và chính xác
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Kiểm tra các điều kiện trên thông tin đăng kí2. Gửi yêu cầu lấy đăng kí đến model3. Trả về kết quả yêu cầu đăng kí có hợp lệ hay không

Bảng 33: Method checkLoginRequest()

checkRegisterRequest(): class ProfileModel

Arguments:	username:String
Return:	struct pData chứa gói các thông tin về người dùng
Description:	Đọc và trả về thông tin của một người dùng với tên xác định
Pre-condition:	None
Post-condition:	Thông tin người dùng được tra cứu và trả về
Exception:	Tên người dùng không xác định, trả về NULL
Flows of Events:	<ol style="list-style-type: none">1. Kiểm tra tên người dùng trong hệ thống2. Đọc thông tin của người dùng và gắn vào một struct pData3. Trả kết quả

Bảng 34: Method getUser(username:String)

getUser(username:String):

Arguments:	username:String,dob: Date,realname: String, gender: String, joined: Date, company: String
Return:	True nếu yêu cầu thành công, ngược lại là false
Description:	Cập nhật dữ liệu người dùng
Pre-condition:	Dữ liệu đã được kiểm tra bởi Controller
Post-condition:	Dữ liệu được cập nhật đầy đủ và chính xác
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Cập nhật các thông số vào database2. Trả về kết quả cập nhật

Bảng 35: Method update()

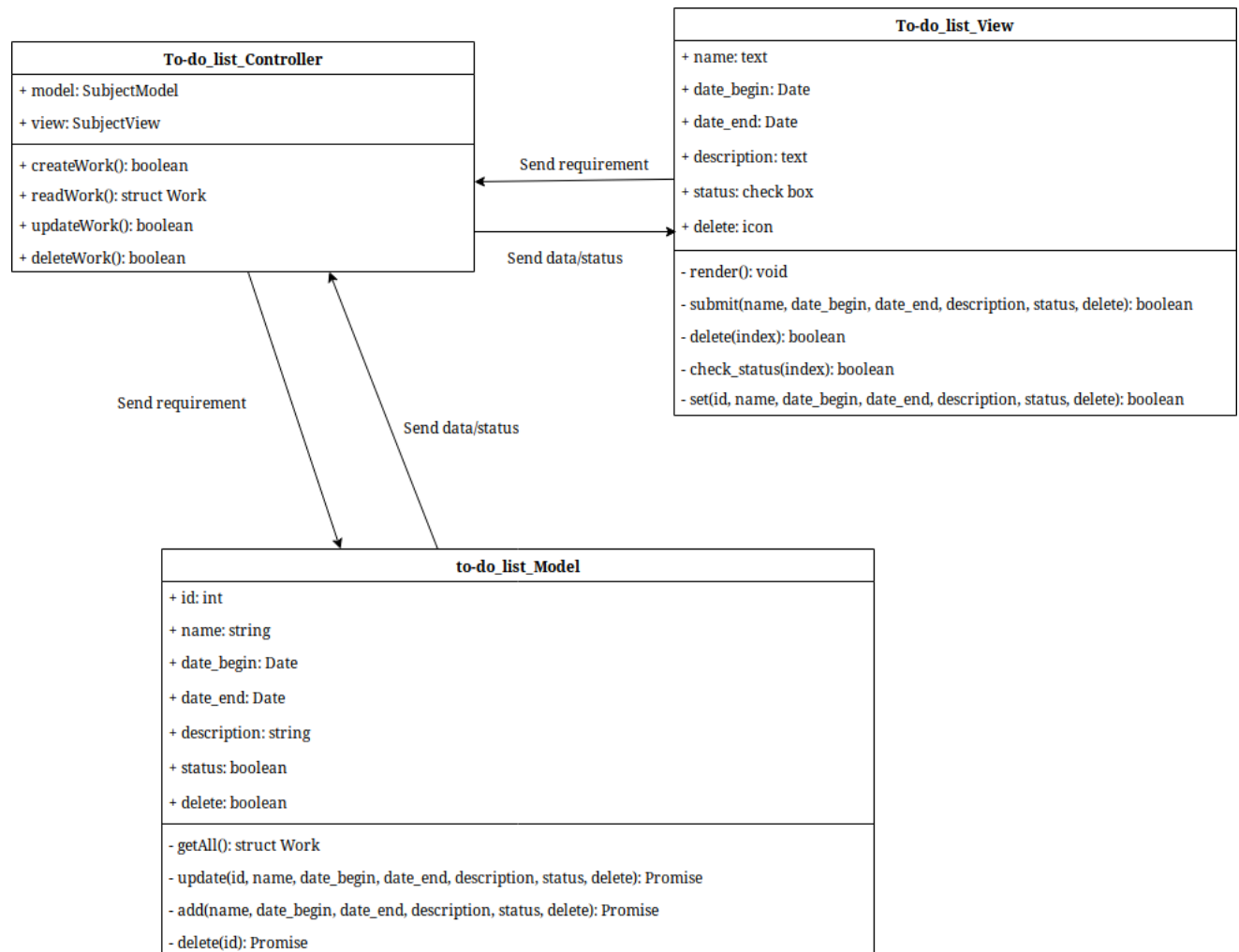
update():

Arguments:	username:String,dob: Date,realname: String, gender: String, joined: Date, company: String
Return:	True nếu yêu cầu thành công, ngược lại là false
Description:	Tạo người dùng mới
Pre-condition:	Dữ liệu đã được kiểm tra bởi Controller
Post-condition:	Dữ liệu được cập nhật đầy đủ và chính xác
Exception:	
Flows of Events:	<ol style="list-style-type: none">1. Cập nhật các thông số vào database2. Trả về kết quả cập nhật

Bảng 36: Method createUser()

createUser():

8.5 Class diagram cho chức năng tạo những công việc cần làm



Class todo list View:

. `render(): void`

Arguments:

Return: `void`

Description: Hiển thị danh sách các lịch các công việc đã tạo

Pre-condition:

Post-condition: Giao diện hiển thị dữ liệu chính xác

Exceptions: Không.

Flow of Events

(a) Hiển thị dữ liệu mỗi lần reload trang

. submit(name, date_begin, date_end, description, status, delete): boolean

Arguments: Tên của công việc, ngày bắt đầu công việc, ngày kết thúc công việc, mô tả, trạng thái hoàn thành, trạng thái xóa công việc

Return: boolean

Description: Gửi yêu cầu để thêm một công việc cần làm mới với những trường được lấy từ form

Pre-condition: Yêu cầu nhập tên của công việc là bắt buộc

Post-condition: Giao diện hiển thị giao diện mới với công việc được thêm hoặc hiển thị lỗi nếu không thêm được công việc

Exceptions: Không.

Flow of Events

(a) Lấy thông tin công việc từ form

(b) Gửi yêu cầu và dữ liệu đến controller,

(c) Hiển thị giao diện mới được cập nhật

. delete(index): boolean

Arguments: Hàng cần xóa

Return: boolean

Description: Xóa 1 công việc trong bảng danh sách các công việc cần làm

Pre-condition: Lấy đúng index của hàng cần xóa với id tương ứng tại hàng đó

Post-condition: Giao diện hiển thị giao diện mới với công việc cần xóa đã được xóa hoặc hiển thị lỗi nếu không xóa được công việc

Exceptions: Không.

Flow of Events

(a) Lấy index và lấy được id của công việc tại index đó

(b) Gửi yêu cầu xóa và id cho controller

(c) Hiển thị giao diện mới được cập nhật

. check_status(): boolean

Arguments:

Return: boolean

Description: Đánh dấu tick vào ô status trong bảng(thiết lập lại giá trị cho ô đó)

Pre-condition:

Post-condition: Ô được check sẽ hiển thị có dấu check hoặc không có dấu check

Exceptions: Không.

Flow of Events

(a) Lấy index và lấy được id của công việc tại index đó

(b) Gửi yêu cầu cập nhật trạng thái và id cho controller

(c) Hiển thị giao diện mới được cập nhật

. set(id, name, date_begin, date_end, description, status, delete): boolean

Arguments: Id công việc, tên của công việc, ngày bắt đầu công việc, ngày kết thúc công việc, mô tả, trạng thái hoàn thành, trạng thái xóa công việc

Return: boolean

Description: Gửi yêu cầu sửa lại thông tin của một công việc theo id với những thông tin được sửa

Pre-condition: Trường tên công việc bắt buộc phải có

Post-condition: Dữ liệu mới được cập nhật và hiển thị mới

Exceptions: Không.

Flow of Events

- (a) Gửi yêu cầu cập nhật trạng thái và dữ liệu cho controller
- (b) Hiển thị giao diện mới được cập nhật

Class todo list Controller:

. createWork(): boolean

Arguments:

Return: boolean

Description: Gửi yêu cầu tạo công việc mới đến Model và trả kết quả về cho View khi thực hiện xong

Pre-condition:

Post-condition: Gửi yêu cầu đến Model và trả kết quả về cho View

Exceptions: Không.

Flow of Events

- (a) Nhận yêu cầu tạo và dữ liệu từ View
- (b) Gửi yêu cầu tạo và dữ liệu đến Model
- (c) Nhận phản hồi từ Model
- (d) Gửi phản hồi đến View

. readWork(): boolean

Arguments:

Return: boolean

Description: Gửi yêu cầu lấy dữ liệu từ Model và trả kết quả về cho View khi thực hiện xong

Pre-condition:

Post-condition: Gửi yêu cầu đến Model thành công và trả kết quả về cho View

Exceptions: Không.

Flow of Events

- (a) Nhận yêu cầu get dữ liệu từ View
- (b) Gửi yêu cầu để lấy dữ liệu từ Model
- (c) Nhận phản hồi và dữ liệu trả về từ Model

(d) Gửi phản hồi đến View

. updateWork(): boolean

Arguments:

Return: boolean

Description: Gửi yêu cầu cập nhật dữ liệu đến Model và trả kết quả về cho View khi thực hiện xong

Pre-condition:

Post-condition: Gửi yêu cầu cập nhật công việc thành công đến Model và trả kết quả về cho View khi thực hiện xong

Exceptions: Không.

Flow of Events

(a) Nhận yêu cầu cập nhật dữ liệu và dữ liệu từ View

(b) Gửi yêu cầu cập nhật dữ liệu và dữ liệu từ đến Model

(c) Nhận phản hồi và dữ liệu trả về từ Model

(d) Gửi phản hồi đến View

. deleteWork(): boolean

Arguments:

Return: boolean

Description: Gửi yêu cầu xóa một công việc đến Model và trả kết quả về cho View khi thực hiện xong

Pre-condition:

Post-condition: Gửi yêu đến Model thành công và trả kết quả về cho View

Exceptions: Không.

Flow of Events

(a) Nhận yêu cầu xóa dữ liệu từ View

(b) Gửi yêu cầu xóa dữ liệu đến Model

(c) Nhận phản hồi từ Model

(d) Gửi phản hồi đến View

Class todo list Model:

. getAll(): struct Work

Arguments:

Return: struct Work

Description: Lấy tất cả dữ liệu từ database lên trả về một danh sách chi tiết các công việc

Pre-condition:

Post-condition: Trả dữ liệu thành công về cho controller

Exceptions: Không.

Flow of Events

- (a) Nhận yêu cầu lấy dữ liệu từ controller
- (b) Lấy dữ liệu từ database
- (c) Trả dữ liệu về cho controller

. update(id, name, date_begin, date_end, description, status, delete): Promise

Arguments: Id công việc, Tên của công việc, ngày bắt đầu công việc, ngày kết thúc công việc, mô tả, trạng thái hoàn thành, trạng thái xóa công việc

Return: Promise

Description: Cập nhật dữ liệu của một công việc, trả về một Promise là trạng thái thành công hay thất bại kèm với dữ liệu

Pre-condition: Id và tên công việc là bắt buộc

Post-condition: Trả trạng thái thành công về cho controller

Exceptions: Không.

Flow of Events

- (a) Nhận yêu cầu cập nhật dữ liệu và dữ liệu từ controller
- (b) Cập nhật dữ liệu vào database
- (c) Trả dữ liệu và kết quả về cho controller

. add(name, date_begin, date_end, description, status, delete): Promise

Arguments: Tên của công việc, ngày bắt đầu công việc, ngày kết thúc công việc, mô tả, trạng thái hoàn thành, trạng thái xóa công việc

Return: Promise

Description: Thêm một công việc mới, trả về một Promise là trạng thái thành công hay thất bại kèm với dữ liệu

Pre-condition: Tên của công việc là bắt buộc

Post-condition: Trả trạng thái thành công về cho controller

Exceptions: Không.

Flow of Events

- (a) Nhận yêu cầu thêm dữ liệu và dữ liệu từ controller
- (b) Thêm dữ liệu vào database
- (c) Trả dữ liệu và kết quả về cho controller

. delete(id): Promise

Arguments: id của công việc cần xóa

Return: Promise

Description: Xóa một công việc trong database, trả về một Promise là trạng thái thành công hay thất bại kèm với dữ liệu

Pre-condition: Id của công việc là bắt buộc

Post-condition: Trả trạng thái thành công về cho controller

Exceptions: Không.

Flow of Events

- (a) Nhận yêu cầu xóa dữ liệu từ controller
- (b) Xóa dữ liệu với id tương ứng trong database
- (c) Trả dữ liệu và kết quả về cho controller

9 Kế hoạch phát triển hệ thống

Tài liệu này đã đặc tả sơ lược về hệ thống và chức năng của hệ thống hỗ trợ học tập cho sinh viên. Hệ thống dự định sẽ được hoàn thành trong vòng 10 tuần, bao gồm các công đoạn thiết kế các tài liệu SRS, SDD, hiện thực hệ thống và kiểm thử. Trong tương lai gần, nhóm mong muốn mang dự án ra áp dụng vào thực tế nếu dự án thành công trong phạm vi môn học.

Tài liệu