

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

# **ĐỒ ÁN TỐT NGHIỆP**

**Ứng dụng quản lý đồ án trên nền tảng Android**

**ĐINH THUYẾT HÀ**

ha.dt173086@sis.hust.edu.vn

**Ngành: Công nghệ thông tin**

**Chuyên ngành: Kỹ thuật máy tính**

**Giảng viên hướng dẫn:** ThS.Lê Bá Vui

---

Chữ ký GV

**Ngành:** Kỹ thuật máy tính

**Trường:** Công nghệ thông tin và truyền thông

**Hà Nội, 06/2022**

## **Lời cảm ơn**

Đồ án tốt nghiệp là minh chứng kết quả học tập tích lũy của em trong suốt 5 năm học tập tại Trường Đại học bách khoa Hà Nội. Ngoài sự nỗ lực của bản thân, đồ án tốt nghiệp của em sẽ không thể hoàn thành tốt, chẵn chu nếu không có sự chỉ bảo và hướng dẫn tận tình của thầy trong suốt thời gian qua.

Trước hết em xin bày tỏ lòng biết ơn sâu sắc đến thầy hướng dẫn là ThS.Lê Bá Vui người luôn đồng hành, đưa ra lời khuyên giúp em có thể hoàn thiện đồ án. Em xin chân thành cảm ơn các thầy, cô Trường Công nghệ Thông tin và Truyền thông - ĐH Bách khoa Hà Nội đã truyền đạt những kiến thức quý báu của mình cho em cũng như các bạn sinh viên khác, có thể nói đây là hành trang là những kiến thức nền tảng quý báu trước khi bước ra ngoài xã hội.

Cuối cùng em xin gửi lời cảm tới tới gia đình bạn bè luôn động viên, tin tưởng em trong thời gian qua. Trong quá trình làm đồ án, do thời gian có hạn và kinh nghiệm làm các ứng dụng thực tế của em còn ít nên không thể tránh khỏi các sai sót, em mong nhận được sự đóng góp của các thầy cô, bạn bè để đồ án của em có thể hoàn thiện hơn.

## Tóm tắt nội dung đề án

Bộ não của con người không thể nhớ được hết các công việc cần làm. Đây là điều mà ai cũng từng mắc phải thế nhưng từ khi các phần mềm, ứng dụng quản lý công việc ra đời đã giúp người dùng tối ưu được hiệu suất làm việc, quản lý thời gian hiệu quả, hoàn thành công việc đúng thời hạn. Các ứng dụng này mang đến nhiều lợi ích thiết thực chính vì vậy em quyết định lựa chọn đề tài "Quản lý đồ án trên nền tảng Android" dựa trên các ứng dụng quản lý công việc phục vụ cho học tập.

Ứng dụng quản lý đồ án được xây dựng nhằm mục đích giúp cho sinh viên, giảng viên có thể quản lý đồ án một cách hiệu quả, có thể lên danh sách các công việc cần thực hiện trong suốt quá trình làm đồ án, thông báo cho sinh viên biết các công việc sắp đến hạn để sinh viên có cách xử lý. Ngoài ra ứng dụng cũng có các tính năng tương tự tính năng của "eHust" như: sinh viên có thể xem các tin tức, thông báo đến từ trường, xem danh sách, thông tin chi tiết của các bạn trong lớp cũng như một bạn bất kì thông qua mã số sinh viên hay tên đầy đủ; xem lịch học, lịch gặp giáo viên hướng dẫn đồ án, tạo công việc; giảng viên cũng có thể tạo công việc cho sinh viên thực hiện, nhận được thông báo khi sinh viên cập nhật công việc, thống kê được số lượng đồ án đang hướng dẫn, tạo lịch gặp với sinh viên sau khi đã có sự thống nhất về thời gian với sinh viên đó; admin có thể phân công sinh viên cho giảng viên.

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Lí do chọn đề tài.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Bố cục đồ án .....	1
<b>CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....</b>	<b>3</b>
2.1 Khảo sát .....	3
2.2 Tổng quan chức năng .....	4
2.2.1 Biểu đồ use case tổng quát .....	4
2.2.2 Phân rã use case Xem chi tiết công việc .....	6
2.2.3 Phân rã use case Tìm kiếm người dùng.....	6
2.2.4 Phân rã use case Bình luận vào công việc .....	7
2.2.5 Phân rã use case Xem danh sách phân công đồ án .....	7
2.2.6 Phân rã use case Xem file đính kèm.....	7
2.2.7 Phân rã use case Xem danh sách đồ án hướng dẫn theo kì .....	8
2.3 Các quy trình nghiệp vụ .....	8
2.3.1 Nghiệp vụ Đăng nhập.....	8
2.3.2 Nghiệp vụ Tạo công việc .....	9
2.3.3 Nghiệp vụ Chỉnh sửa công việc .....	9
2.3.4 Nghiệp vụ Bình luận công việc.....	10
2.3.5 Nghiệp vụ Xoá thông báo đồ án đã đọc .....	10
2.3.6 Nghiệp vụ Tạo lịch gặp sinh viên.....	11
2.3.7 Nghiệp vụ Phân công hướng dẫn đồ án .....	11
2.3.8 Nghiệp vụ Lọc danh sách phân công đồ án .....	12
2.3.9 Nghiệp vụ Xoá phân công hướng dẫn đồ án .....	12

2.3.10 Nghiệp vụ Xem chi tiết đề tài .....	13
2.3.11 Nghiệp vụ Xem danh sách đề án đang hướng dẫn.....	13
2.3.12 Nghiệp vụ Đăng xuất.....	13
2.4 Đặc tả ca chức năng .....	14
2.4.1 Đặc tả use case Đăng nhập.....	14
2.4.2 Đặc tả use case Tạo công việc .....	15
2.4.3 Đặc tả use case Chỉnh sửa công việc .....	17
2.4.4 Đặc tả use case Bình luận công việc.....	19
2.4.5 Đặc tả use case Xem file đính kèm .....	20
2.4.6 Đặc tả use case Xóa thông báo đề án đã đọc .....	21
2.4.7 Đặc tả use case Tạo lịch gặp sinh viên.....	22
2.4.8 Đặc tả use case Xem danh sách đề tài đang hướng dẫn.....	24
2.4.9 Đặc tả use case Xem chi tiết đề tài .....	25
2.4.10 Đặc tả use case Phân công hướng dẫn đề án.....	26
2.4.11 Đặc tả use case Lọc danh sách phân công đề án .....	27
2.4.12 Đặc tả use case Xóa phân công hướng dẫn đề án" .....	28
2.4.13 Đặc tả use case Đăng xuất.....	29
2.5 Các yêu cầu phi chức năng.....	29
2.5.1 Yêu cầu về bảo mật .....	29
2.5.2 Yêu cầu về hiệu năng.....	29
2.5.3 Yêu cầu về giao diện.....	29
<b>CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....</b>	<b>30</b>
3.1 Ngôn ngữ lập trình Kotlin.....	30
3.2 Hệ quản trị cơ sở dữ liệu MySQL.....	30
3.3 Spring Boot framework .....	31
3.4 Bộ công cụ Jetpack Compose.....	31

3.5 Kotlin Coroutines .....	32
3.6 Dagger Hilt .....	32
3.7 Retrofit .....	33
3.8 Json Web Token (JWT) .....	33
3.9 Docker.....	34
3.10 Minio Object Storage .....	34
3.11 Azure Cloud.....	34
<b>CHƯƠNG 4. PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG.....</b>	<b>36</b>
4.1 Thiết kế kiến trúc.....	36
4.1.1 Lựa chọn kiến trúc phần mềm .....	36
4.1.2 Thiết kế tổng quan.....	39
4.1.3 Thiết kế chi tiết gói Task.....	41
4.1.4 Thiết kế chi tiết gói Admin.....	42
4.1.5 Thiết kế chi tiết gói Projects .....	43
4.2 Thiết kế chi tiết.....	44
4.2.1 Thiết kế giao diện .....	44
4.2.2 Thiết kế lớp .....	44
4.2.3 Thiết kế cơ sở dữ liệu .....	48
4.3 Xây dựng ứng dụng.....	56
4.3.1 Thư viện và công cụ sử dụng.....	56
4.3.2 Kết quả đạt được .....	58
4.3.3 Minh hoạ các chức năng chính .....	58
4.4 Kiểm thử.....	64
4.5 Triển khai .....	67
<b>CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>71</b>
5.1 Kết luận.....	71

5.2 Hướng phát triển của đồ án.....	71
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>73</b>

## DANH MỤC HÌNH VẼ

Hình 2.1	Sơ đồ use case tổng quan . . . . .	5
Hình 2.2	Phân rã use case Xem chi tiết công việc . . . . .	6
Hình 2.3	Phân rã use case Tìm kiếm người dùng . . . . .	6
Hình 2.4	Phân rã use case Bình luận vào công việc . . . . .	7
Hình 2.5	Phân rã use case Xem danh sách phân công đồ án . . . . .	7
Hình 2.6	Phân rã use case Xem file đính kèm . . . . .	7
Hình 2.7	Phân rã use case Xem danh sách đồ án hướng dẫn theo kì . . . . .	8
Hình 2.8	Sơ đồ hoạt động quy trình nghiệp vụ Đăng nhập . . . . .	8
Hình 2.9	Sơ đồ hoạt động quy trình nghiệp vụ Tạo công việc . . . . .	9
Hình 2.10	Sơ đồ hoạt động quy trình nghiệp vụ Chỉnh sửa công việc . . . . .	9
Hình 2.11	Sơ đồ hoạt động quy trình nghiệp vụ Bình luận công việc . . . . .	10
Hình 2.12	Sơ đồ hoạt động quy trình Xóa thông báo đồ án đã đọc . . . . .	10
Hình 2.13	Sơ đồ hoạt động quy trình nghiệp vụ Tạo lịch gặp sinh viên . . . . .	11
Hình 2.14	Sơ đồ hoạt động quy trình nghiệp vụ Phân công hướng dẫn đồ án . . . . .	11
Hình 2.15	Sơ đồ hoạt động quy trình nghiệp vụ Lọc danh sách phân công đồ án . . . . .	12
Hình 2.16	Sơ đồ hoạt động quy trình nghiệp vụ Xóa phân công hướng dẫn đồ án . . . . .	12
Hình 2.17	Sơ đồ hoạt động quy trình nghiệp vụ Xem chi tiết đề tài . . . . .	13
Hình 2.18	Sơ đồ hoạt động quy trình nghiệp vụ Xem danh sách đồ án đang hướng dẫn . . . . .	13
Hình 2.19	Sơ đồ hoạt động quy trình nghiệp vụ Đăng xuất . . . . .	13
Hình 4.1	Kiến trúc tổng quan của hệ thống . . . . .	36
Hình 4.2	Luồng hoạt động của Backend theo mô hình kết hợp giữa mô hình ba lớp và MVC . . . . .	37
Hình 4.3	Luồng hoạt động của Android theo mô hình MVVP . . . . .	38
Hình 4.4	Thiết kế gói tổng quan . . . . .	39
Hình 4.5	Thiết kế chi tiết gói Task . . . . .	41
Hình 4.6	Thiết kế chi tiết gói Admin . . . . .	42
Hình 4.7	Thiết kế chi tiết gói Projects . . . . .	43
Hình 4.8	Thiết kế chi tiết lớp Task . . . . .	45
Hình 4.9	Thiết kế chi tiết lớp Admin . . . . .	47
Hình 4.10	Sơ đồ thiết kế cơ sở dữ liệu . . . . .	48



Hình 4.11	Màn hình đăng nhập . . . . .	59
Hình 4.12	Màn hình danh sách công việc . . . . .	59
Hình 4.13	Màn hình xem chi tiết đề tài . . . . .	59
Hình 4.14	Màn hình tạo công việc . . . . .	59
Hình 4.15	Màn hình chi tiết công việc . . . . .	60
Hình 4.16	Màn hình chỉnh sửa công việc . . . . .	60
Hình 4.17	Màn hình bình luận công việc . . . . .	60
Hình 4.18	Màn hình xem file đính kèm . . . . .	60
Hình 4.19	Màn hình trang chủ của sinh viên . . . . .	61
Hình 4.20	Màn hình danh sách đồ án đang làm . . . . .	61
Hình 4.21	Màn hình trang chủ của giảng viên . . . . .	61
Hình 4.22	Màn hình danh sách đồ án hướng theo học kì . . . . .	61
Hình 4.23	Màn hình danh sách đề tài hướng dẫn của Đồ án kỹ sư . . . . .	62
Hình 4.24	Màn hình danh sách thông báo về đồ án . . . . .	62
Hình 4.25	Màn hình tạo lịch gặp sinh viên . . . . .	62
Hình 4.26	Màn hình trang chủ của quản trị viên . . . . .	63
Hình 4.27	Màn hình quản lý phân công đồ án . . . . .	63
Hình 4.28	Màn hình chọn xoá phân công đồ án để xoá . . . . .	63
Hình 4.29	Màn hình thống kê thông tin . . . . .	63
Hình 4.30	Cấu hình nginx . . . . .	69

## DANH MỤC BẢNG BIỂU

Bảng 2.1	Bảng đặc tả use case Đăng nhập . . . . .	14
Bảng 2.2	Bảng dữ liệu đầu vào use case Đăng nhập. . . . .	15
Bảng 2.3	Bảng đặc tả use case Tạo công việc. . . . .	15
Bảng 2.4	Bảng dữ liệu đầu vào use case Tạo công việc. . . . .	16
Bảng 2.5	Bảng đặc tả use case Chỉnh sửa công việc. . . . .	17
Bảng 2.6	Bảng dữ liệu đầu vào use case Chỉnh sửa công việc. . . . .	18
Bảng 2.7	Bảng đặc tả use case Bình luận công việc. . . . .	19
Bảng 2.8	Bảng dữ liệu đầu vào use case Bình luận công việc. . . . .	19
Bảng 2.9	Bảng đặc tả use case Xem file đính kèm. . . . .	20
Bảng 2.10	Bảng đặc tả use case Xóa thông báo đồ án đã đọc. . . . .	21
Bảng 2.11	Bảng đặc tả use case Tạo lịch gặp sinh viên. . . . .	22
Bảng 2.12	Bảng dữ liệu đầu vào use case Tạo lịch gặp sinh viên. . . . .	23
Bảng 2.13	Bảng đặc tả use case Xem danh sách đề tài đang hướng dẫn. . . . .	24
Bảng 2.14	Bảng đặc tả use case Xem chi tiết đề tài. . . . .	25
Bảng 2.15	Bảng đặc tả use case Phân công hướng dẫn đồ án. . . . .	26
Bảng 2.16	Bảng đặc tả use case Lọc danh sách phân công đồ án. . . . .	27
Bảng 2.17	Bảng đặc tả use case Xóa danh sách phân công đồ án. . . . .	28
Bảng 2.18	Bảng đặc tả use case Đăng xuất. . . . .	29
Bảng 4.1	Bảng dữ liệu User . . . . .	49
Bảng 4.2	Bảng dữ liệu Class . . . . .	50
Bảng 4.3	Bảng dữ liệu User_class . . . . .	51
Bảng 4.4	Bảng dữ liệu News . . . . .	51
Bảng 4.5	Bảng dữ liệu User_subject . . . . .	52
Bảng 4.6	Bảng dữ liệu Task . . . . .	52
Bảng 4.7	Bảng dữ liệu Topic . . . . .	53
Bảng 4.8	Bảng dữ liệu Comments . . . . .	53
Bảng 4.9	Bảng dữ liệu Meeting . . . . .	54
Bảng 4.10	Bảng dữ liệu Attachments . . . . .	54
Bảng 4.11	Bảng dữ liệu More_information_topic . . . . .	55
Bảng 4.12	Bảng dữ liệu Pairing_teacher_with_student . . . . .	56
Bảng 4.13	Bảng dữ liệu Subject . . . . .	56
Bảng 4.14	Danh sách thư viện và công cụ sử dụng . . . . .	57
Bảng 4.15	Thông tin kết quả đạt được . . . . .	58
Bảng 4.16	Bảng kết quả kiểm thử . . . . .	67

## DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
AWS S3	Dịch vụ lưu trữ mà Amazon cung cấp (Amazon Simple Storage Service)
DI	Design pattern giúp giảm sự phụ thuộc giữa các module (Dependency Injection)
JWT	Thư viện lập trình (JSON WEB TOKEN)

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Lí do chọn đề tài

Trí nhớ con người không phải là một kho lưu trữ vô hạn vì thế không thể nhớ được mình đã làm những gì, tiến độ đến đâu, còn điều gì chưa thực hiện, việc theo dõi và kiểm soát trở nên ngày càng khó khăn. Bên cạnh đó những biện pháp quản lý truyền thống không thể hỗ trợ con người thực hiện công việc một cách hiệu quả. Chính vì thế mọi người tìm đến một công cụ thông minh, tiện lợi hơn đó chính là các phần mềm, ứng dụng quản lý công việc.

Các ứng dụng quản lý công việc ra đời mang lại nhiều lợi ích cho con người trong mọi lĩnh vực: cải thiện hiệu quả, năng suất công việc, các công việc được sắp xếp theo thứ tự ưu tiên nhờ đó mọi người sẽ biết mình cần phải làm việc gì trước việc gì sau, nhắc nhở người dùng hoàn thành công việc trước deadline, các ứng dụng quản lý công việc còn giúp người dùng tránh sai sót, bỏ sót công việc. Có thể nói các ứng dụng quản lý công việc để trở thành một công cụ đắc lực giúp con người kiểm soát công việc, trở thành một phần không thể thiếu.

Nhằm tìm hiểu quá trình phát triển cũng như có thể tự xây dựng một ứng dụng quản lý công việc trong khuôn khổ phù hợp với sinh viên, giảng viên em đã quyết định lựa chọn đề tài xây dựng "Ứng dụng quản lý đồ án trên nền tảng Android".

## 1.2 Mục tiêu và phạm vi đề tài

Mục tiêu và phạm vi của đồ án là thiết kế và xây dựng một ứng dụng quản lý đồ án trên nền tảng Android, có thể đáp ứng được các chức năng cơ bản của sinh viên và giảng viên như lên danh sách các công việc cần làm khi thực hiện đồ án, thường xuyên cập nhật tiến độ thực hiện công việc để giảng viên có thể kịp thời nắm bắt tình hình. Ngoài ra sinh viên cũng có thể xem lịch học của mình, xem danh sách lớp sinh viên, tìm kiếm một sinh viên hoặc giảng viên bất kỳ, tìm kiếm lớp, giảng viên có thể tạo lịch gặp mặt với sinh viên sau khi hai bên đã thống nhất về mặt thời gian. Không chỉ vậy ứng dụng cũng có không gian riêng dành cho quản trị viên có thể thống kê được số sinh viên, giảng viên, số môn đồ án, phân công hướng dẫn đồ án.

## 1.3 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về khảo sát và phân tích yêu cầu, phạm vi của đề tài. Trong chương này trình bày về ưu và nhược điểm của các ứng dụng đã thực hiện khảo sát. Từ đó đưa ra các chức năng cần phát triển cho ứng dụng của mình các đối tượng sử

dụng ứng dụng, nêu ra các chức năng của ứng dụng, trình bày tổng quan, phân tích làm rõ các quy trình nghiệp vụ, đặc tả chi tiết các use case, các yêu cầu phi chức năng. Mỗi chức năng sẽ được mô tả thông qua biểu đồ use case phân rã, quy trình nghiệp vụ, đặc tả chi tiết cho từng use case.

Trong Chương 3 sẽ trình bày về các công nghệ, cơ sở lý thuyết của các công nghệ, ưu nhược điểm của nó và cách áp dụng nó vào đề án này như thế nào.

Chương 4 trình bày chi tiết về cách phát triển và triển khai ứng dụng, từ việc thiết kế tổng quan đến việc đi sâu vào thiết kế chi tiết từng gói từng lớp, thiết kế cơ sở dữ liệu, thiết kế giao diện,... Thêm vào đó chương này cũng sẽ trình bày các thư viện và công cụ sử dụng trong quá trình phát triển ứng dụng. Sau cùng là minh hoạ một số kết quả đạt được sau khi xây dựng thành công ứng dụng quản lý đề án và tiến hành kiểm thử.

Chương 5 đưa ra kết quả mà đề án đạt được. Các ưu điểm và nhược điểm của ứng dụng. Chỉ ra được hướng phát triển của hệ thống trong tương lai.

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

### 2.1 Khảo sát

Để có cái nhìn khách quan cũng như có trải nghiệm thực tế để xây dựng một ứng dụng quản lý đồ án Android em đã khảo sát qua một số ứng dụng: Trello, Asana, Any.Do, eHUST.

Sau khi trải nghiệm các ứng dụng thì em nhận thấy nhìn chung tất cả các ứng dụng đều có giao diện rất thân thiện với người dùng. Về Trello thông qua khảo sát em thấy ứng dụng này hướng đến sự đơn giản, linh hoạt, miễn phí, nó thu hút được khoảng 4.6 triệu người đăng kí sử dụng, tuy nhiên em nhận thấy nó có một nhược điểm cho là đối với các thiết bị như smart phone thì màn hình của điện thoại không được rộng nên việc sử dụng tay để kéo thả các task là không hợp lí, nó chỉ hợp lí và đẹp trên các thiết bị có màn hình rộng hơn như ipad hay dùng với bản web, các tính năng của Trello rất hữu ích nó là một công cụ quản lý công việc trực quan hỗ trợ các nhóm lên ý tưởng, lập kế hoạch một cách hiệu quả, có tổ chức.

Khác với Trello thì Asana không trực quan hoá luồng công việc mà nó trông giống với một danh sách các công việc cần phải làm, nơi tất cả mọi người có thể tạo nhiệm vụ và giao việc cho người khác, giúp người quản lí có thể theo dõi tổng quan được dự án, các công việc được sắp xếp theo độ ưu tiên giúp cho người thực hiện dễ dàng có thể nhận biết các công việc nào cần thực hiện trước, sau.

Về Any.Do là phần mềm mang lại trải nghiệm khá tuyệt vời cho em. Nó có giao diện tinh tế, hiện đại, có thể sắp xếp công việc theo độ ưu tiên, có thể thực hiện ghi chú bằng âm thanh hoặc hình ảnh rất sống động, tuy nhiên Any.Do lại không hỗ trợ với ứng dụng thứ 3 như Gmail.

eHUST là một phần mềm dành cho cả sinh viên, giảng viên, cán bộ toàn trường. Với ứng dụng này người dùng có khả năng truy cập và khai thác các thông tin một cách hiệu quả: các thông tin về lớp học, thời khoá biểu, tra cứu sinh viên, giảng viên, nhắc lịch học. Tuy nhiên phần mềm này còn khá là hạn chế và đôi lúc xảy ra sai sót trong quá trình nhắc lịch học cho các bạn sinh viên.

Thông qua việc khảo sát 4 ứng dụng trên em đã quyết định phát triển ứng dụng "Quản lý đồ án trên nền tảng Android" giúp cho sinh viên, giảng viên có thể quản lý công việc một cách khoa học, giảng viên có thể sát sao hơn với quá trình làm đồ án của sinh viên. Ứng dụng cho phép người dùng có thể tạo công việc, cập nhật công việc, gửi thông báo đến giảng viên khi có bất cứ một sự cập nhật công việc nào đến từ sinh viên, giảng viên cũng có thể tạo lịch gặp mặt để trao đổi trực tiếp

với sinh viên, quản trị viên có thể phân công hướng dẫn đồ án. Bên cạnh các chức năng chính trên, ứng dụng cũng cho phép sinh viên, giảng viên xem các tin tức, thông báo quan trọng từ trường, xem thời khoá biểu, tra cứu sinh viên, giảng viên, lớp học.

## **2.2 Tổng quan chức năng**

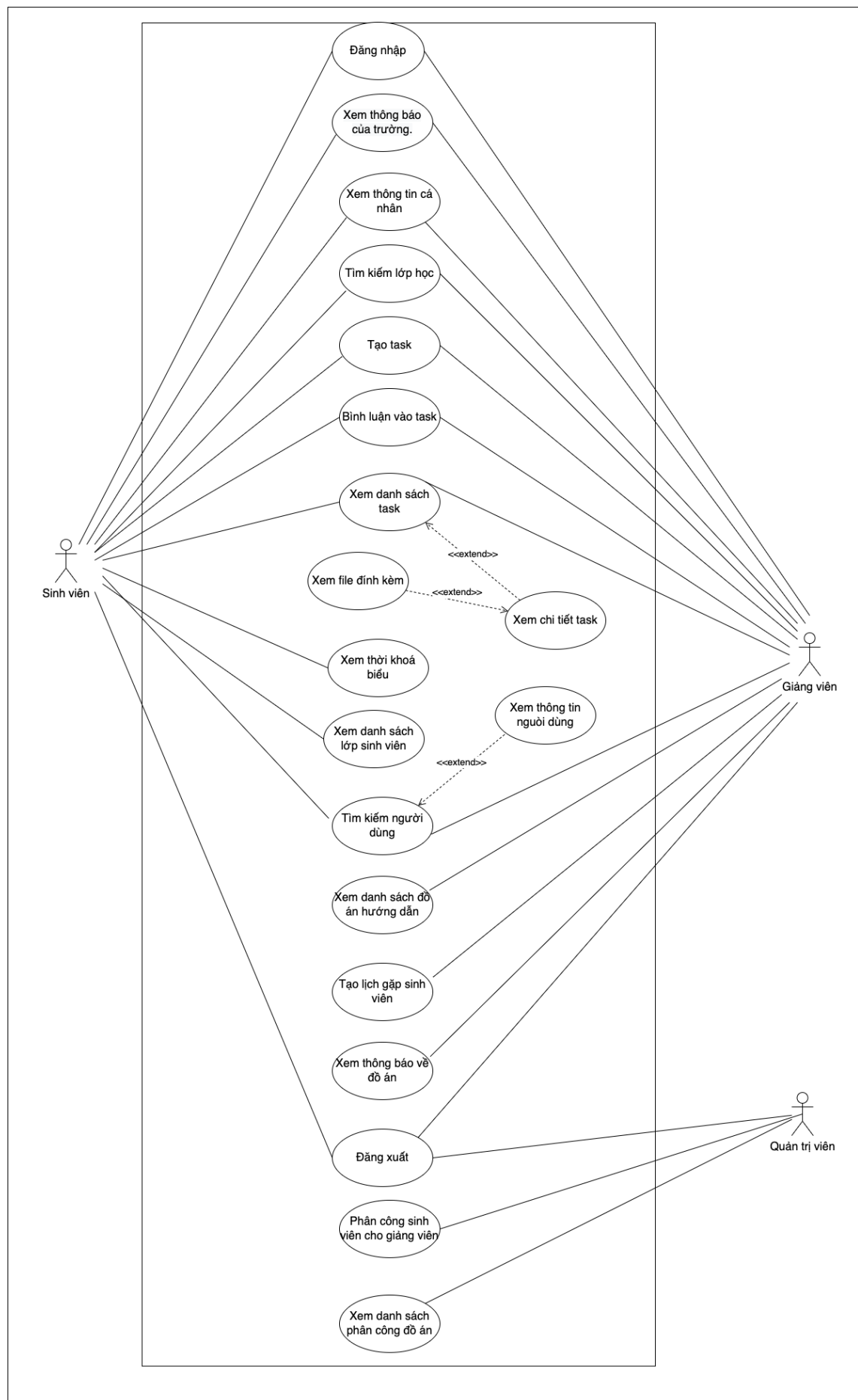
### **2.2.1 Biểu đồ use case tổng quát**

Hệ thống bao gồm 3 tác nhân:

**Sinh viên:** Là đối tượng đã có tài khoản và đăng nhập thành công vào hệ thống.

**Giảng viên:** Là đối tượng đã có tài khoản đăng nhập thành công trên hệ thống, có các chức năng xem danh sách sinh viên hướng dẫn, quản lí đồ án, xem danh sách lớp trong kì...

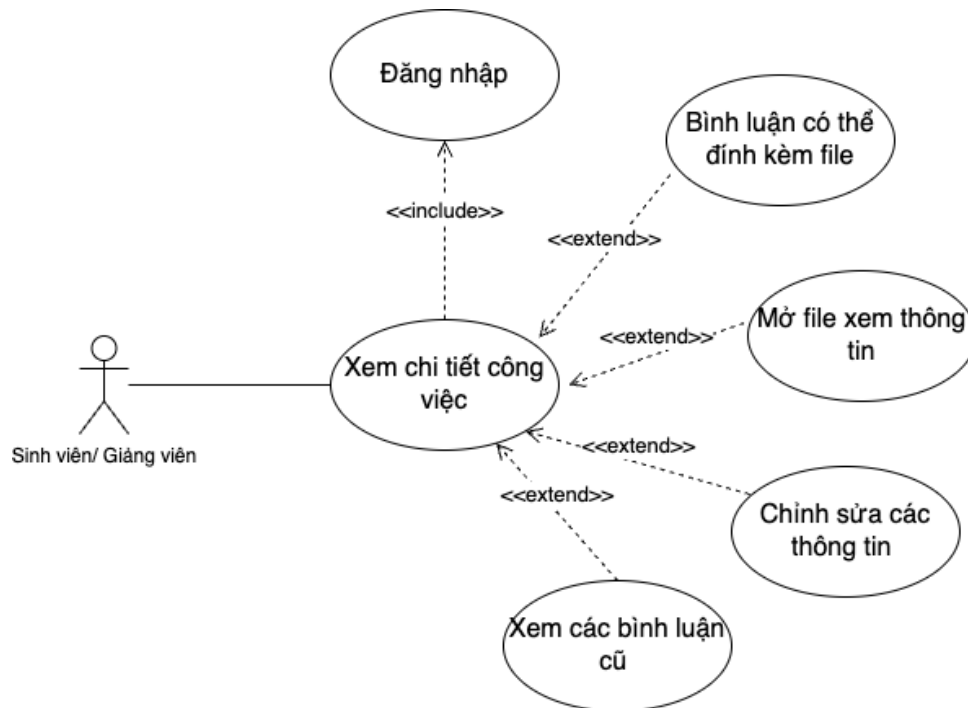
**Quản trị viên:** Là đối tượng đã đăng nhập thành công vào hệ thống với vai trò quản trị có chức năng phân công sinh viên cho giảng viên bất kì.



**Hình 2.1:** Sơ đồ use case tổng quan

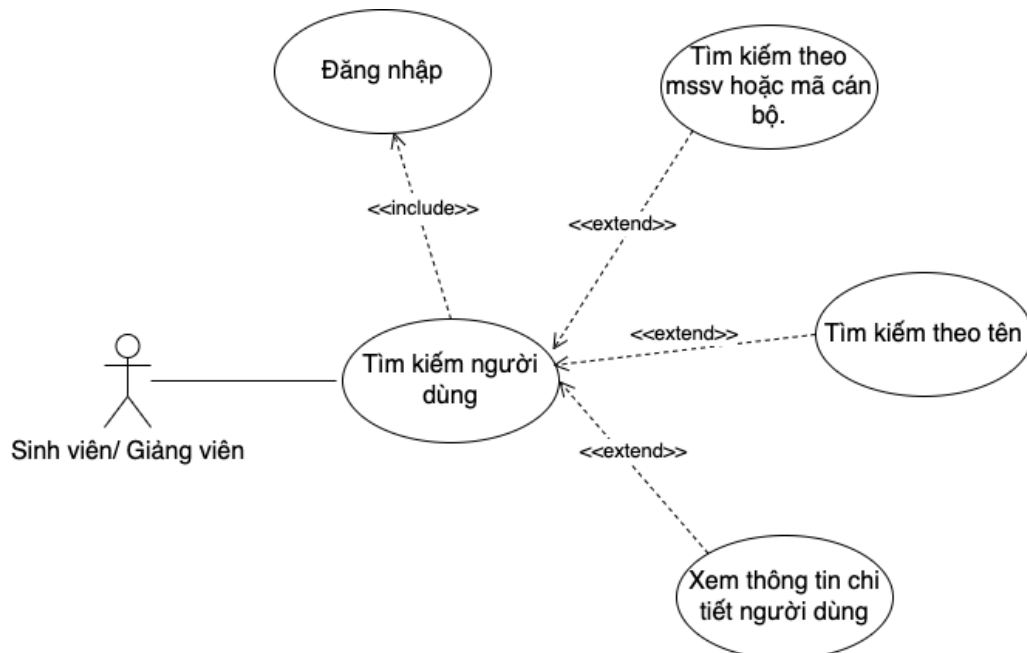


### 2.2.2 Phân rã use case Xem chi tiết công việc



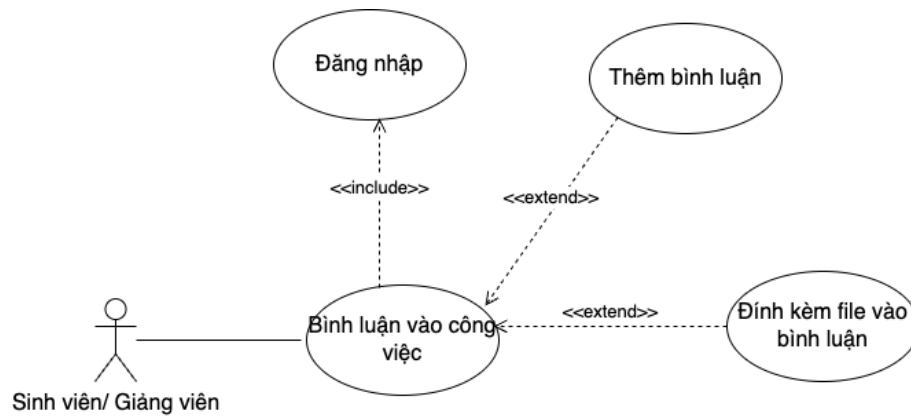
**Hình 2.2:** Phân rã use case Xem chi tiết công việc

### 2.2.3 Phân rã use case Tìm kiếm người dùng



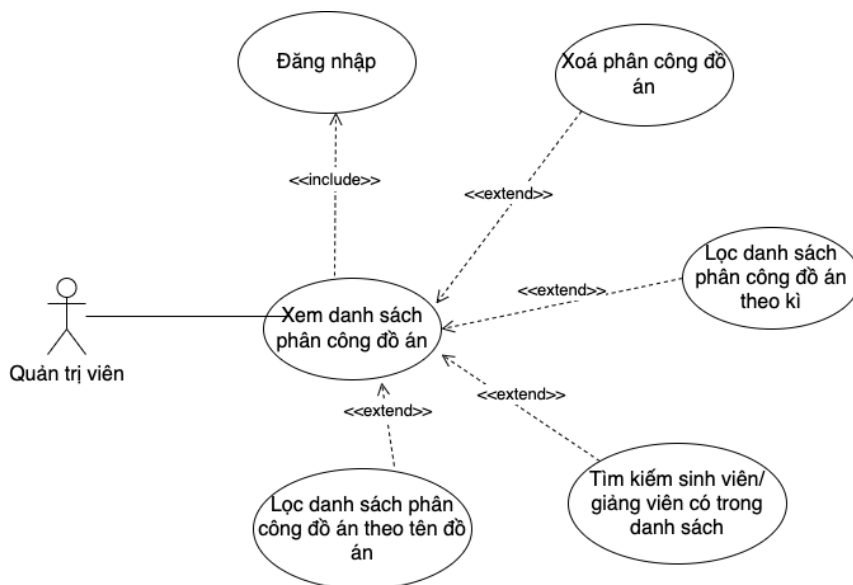
**Hình 2.3:** Phân rã use case Tìm kiếm người dùng

### 2.2.4 Phân rã use case Bình luận vào công việc



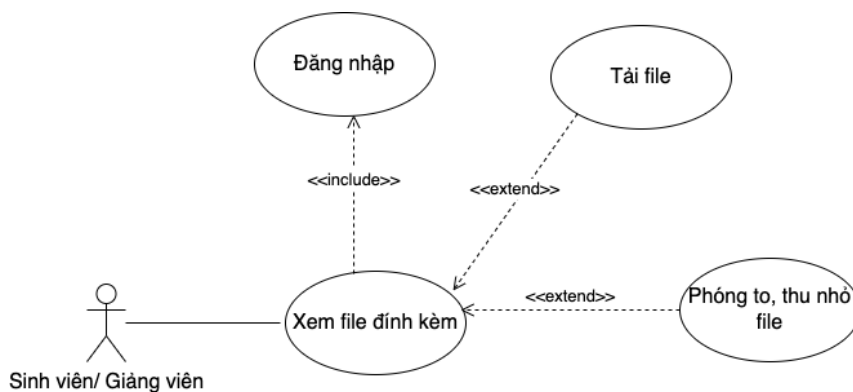
**Hình 2.4:** Phân rã use case Bình luận vào công việc

### 2.2.5 Phân rã use case Xem danh sách phân công đồ án



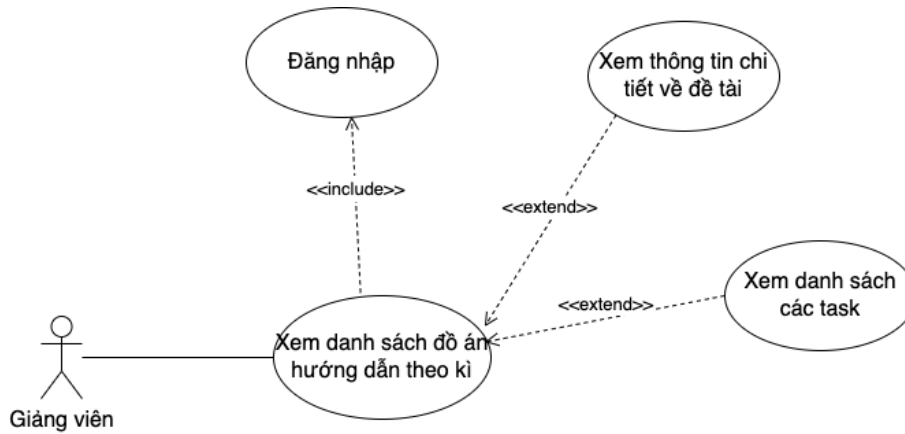
**Hình 2.5:** Phân rã use case Xem danh sách phân công đồ án

### 2.2.6 Phân rã use case Xem file đính kèm



**Hình 2.6:** Phân rã use case Xem file đính kèm

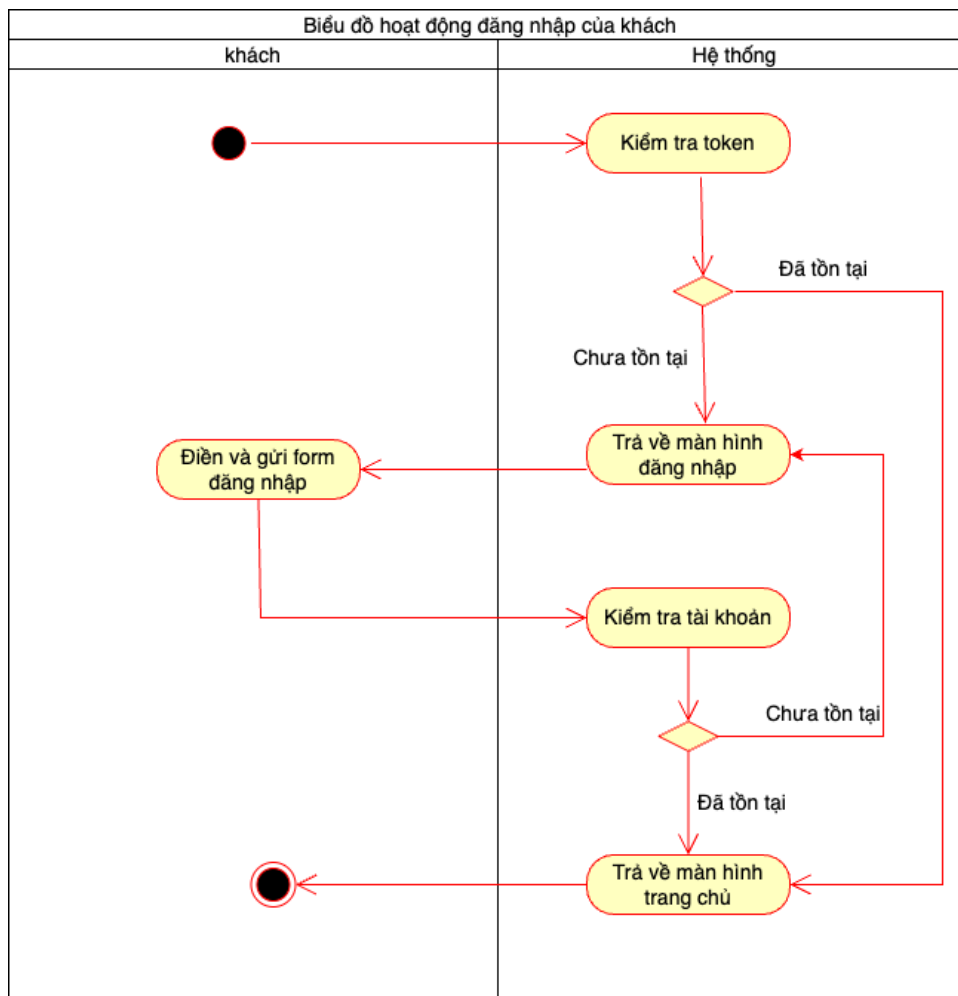
### 2.2.7 Phân rã use case Xem danh sách đề án hướng dẫn theo kì



**Hình 2.7:** Phân rã use case Xem danh sách đề án hướng dẫn theo kì

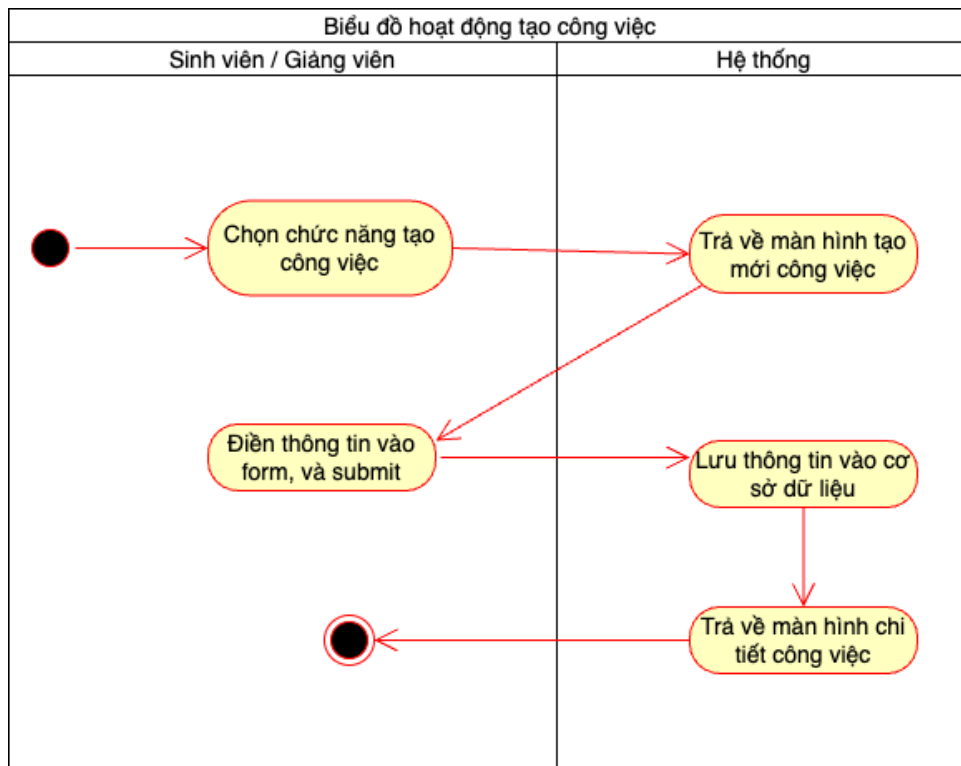
## 2.3 Các quy trình nghiệp vụ

### 2.3.1 Nghiệp vụ Đăng nhập



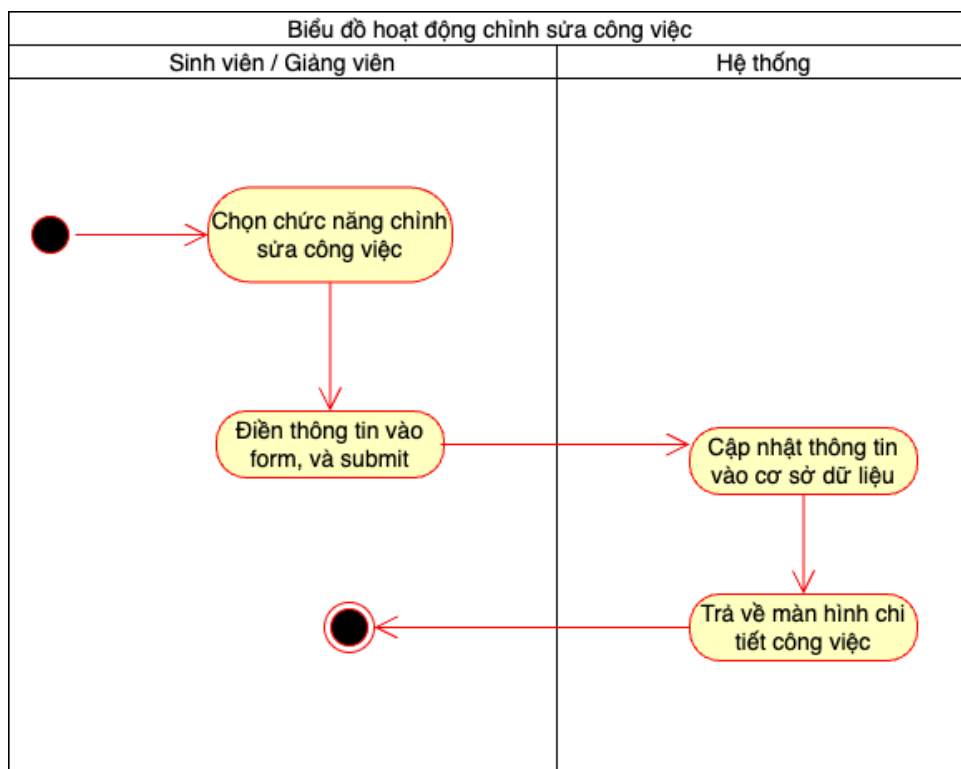
**Hình 2.8:** Sơ đồ hoạt động quy trình nghiệp vụ Đăng nhập

### 2.3.2 Nghiệp vụ Tạo công việc



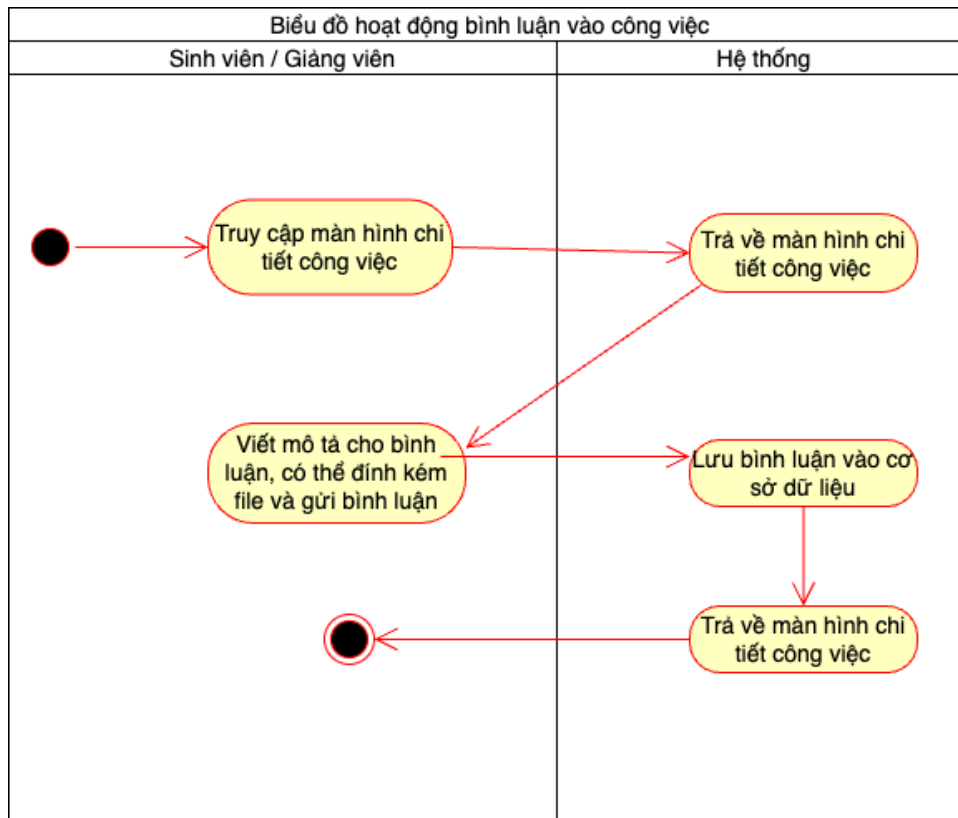
**Hình 2.9:** Sơ đồ hoạt động quy trình nghiệp vụ Tạo công việc

### 2.3.3 Nghiệp vụ Chỉnh sửa công việc



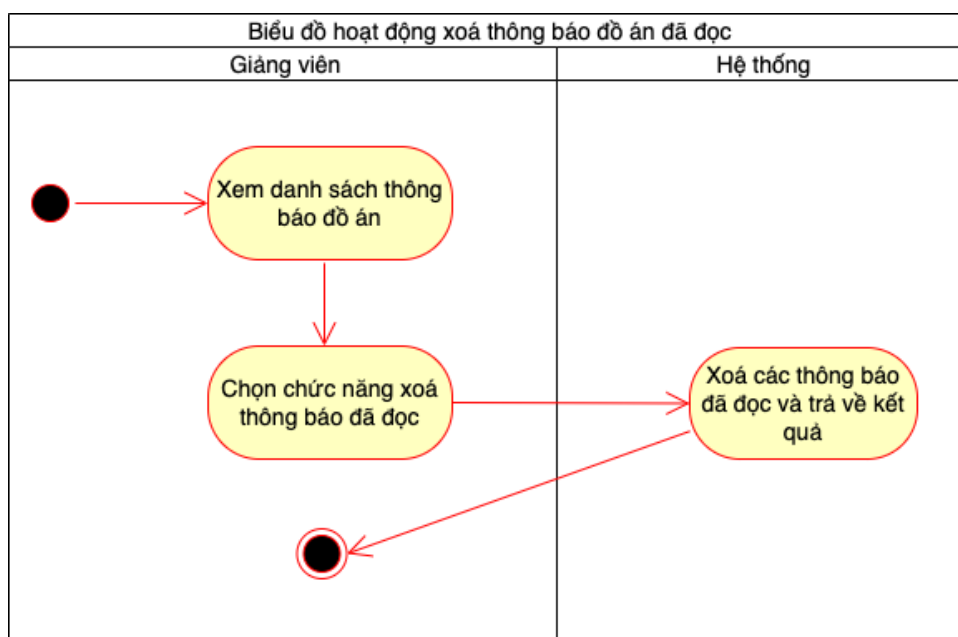
**Hình 2.10:** Sơ đồ hoạt động quy trình nghiệp vụ Chỉnh sửa công việc

### 2.3.4 Nghiệp vụ Bình luận công việc



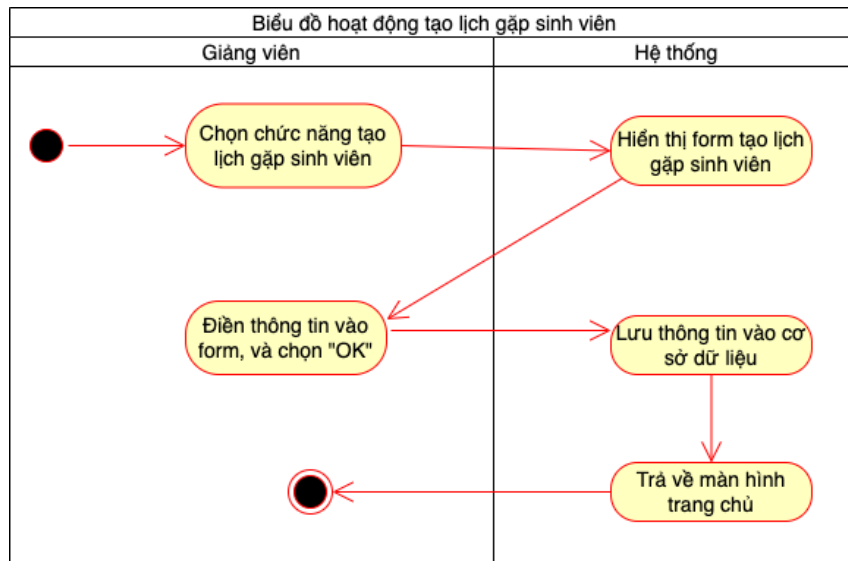
**Hình 2.11:** Sơ đồ hoạt động quy trình nghiệp vụ Bình luận công việc

### 2.3.5 Nghiệp vụ Xóa thông báo đồ án đã đọc



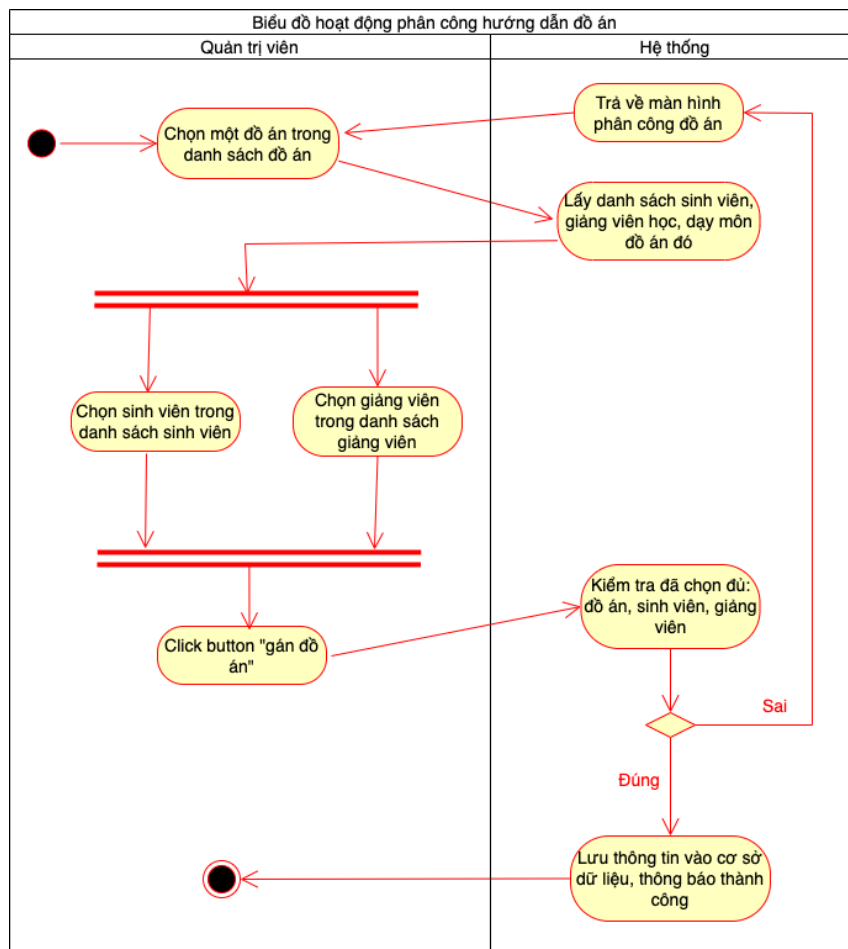
**Hình 2.12:** Sơ đồ hoạt động quy trình Xóa thông báo đồ án đã đọc

### 2.3.6 Nghiệp vụ Tạo lịch gặp sinh viên



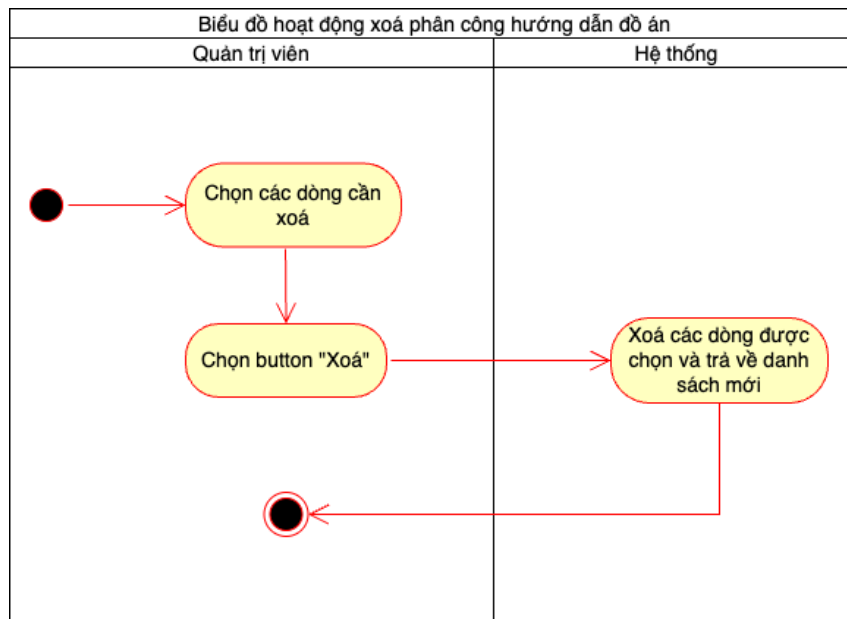
**Hình 2.13:** Sơ đồ hoạt động quy trình nghiệp vụ Tạo lịch gặp sinh viên

### 2.3.7 Nghiệp vụ Phân công hướng dẫn đồ án



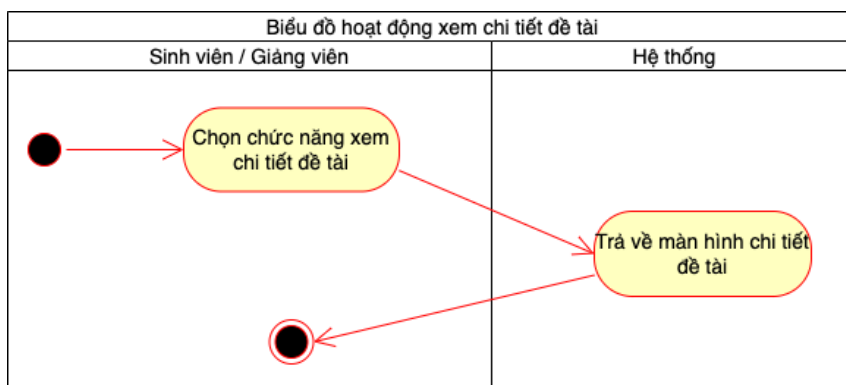
**Hình 2.14:** Sơ đồ hoạt động quy trình nghiệp vụ Phân công hướng dẫn đồ án

### 2.3.9 Nghiệp vụ Xóa phân công hướng dẫn đồ án



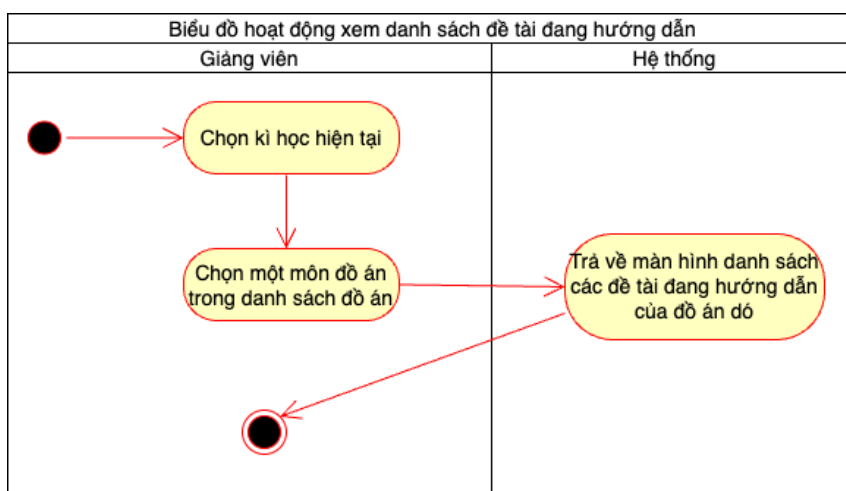
**Hình 2.16:** Sơ đồ hoạt động quy trình nghiệp vụ Xoá phân công hướng dẫn đồ án

### 2.3.10 Nghiệp vụ Xem chi tiết đề tài



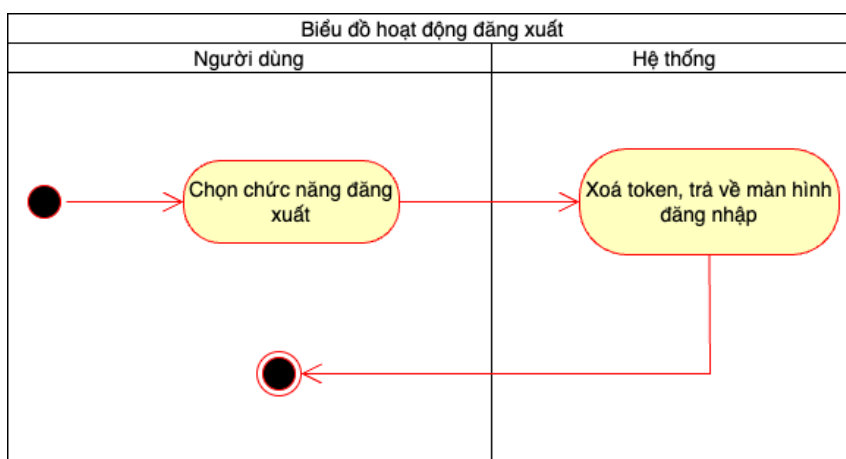
**Hình 2.17:** Sơ đồ hoạt động quy trình nghiệp vụ Xem chi tiết đề tài

### 2.3.11 Nghiệp vụ Xem danh sách đề án đang hướng dẫn



**Hình 2.18:** Sơ đồ hoạt động quy trình nghiệp vụ Xem danh sách đề án đang hướng dẫn

### 2.3.12 Nghiệp vụ Đăng xuất



**Hình 2.19:** Sơ đồ hoạt động quy trình nghiệp vụ Đăng xuất



## 2.4 Đặc tả ca chức năng

### 2.4.1 Đặc tả use case Đăng nhập

Mã use case	UC01	Tên use case	Đăng nhập
Tác nhân	Sinh viên / giảng viên / quản trị viên		
Tiền điều kiện	Đã có tài khoản		
Mục đích sử dụng	Cho phép đăng nhập vào hệ thống và sử dụng các chức năng tương ứng với từng vai trò		
Sự kiện kích hoạt	Không		
Luồng hoạt động chính	Thứ tự thao tác	Được thực hiện bởi	Hành động
	1.	Hệ thống	Kiểm tra token đã tồn tại hay chưa
	2.	Người dùng	Điền thông tin đăng nhập
	3.	Người dùng	Gửi yêu cầu đăng nhập
	4.	Hệ thống	Kiểm tra dữ liệu đầu vào
	5.	Hệ thống	Kiểm tra tài khoản có tồn tại hay không, kiểm tra quyền là sinh viên hay giảng viên hay quản trị viên
	6.	Hệ thống	Chuyển hướng đến trang chủ ứng có giao diện phù hợp với quyền của người dùng
Sự kiện thay thế	1a.	Hệ thống	Trả về màn hình trang chủ
	4a.	Hệ thống	Thông báo lỗi: Mời người dùng nhập đầy đủ các trường
	5a.	Hệ thống	Thông báo lỗi: Đăng nhập không thành công
Hậu điều kiện	Trả về màn hình trang chủ		

**Bảng 2.1:** Bảng đặc tả use case Đăng nhập

Danh sách các trường dữ liệu đầu vào form đăng nhập:

Thứ tự	Trường dữ liệu	Mô tả	Bắt buộc	Yêu cầu hợp lệ	Ví dụ
1.	Id	Id của người dùng	Có		20173086
2.	Password	Mật khẩu	Có		hadinh

**Bảng 2.2:** Bảng dữ liệu đầu vào use case Đăng nhập.

### 2.4.2 Đặc tả use case Tạo công việc

<b>Mã use case</b>	UC02	<b>Tên use case</b>	Tạo công việc
<b>Tác nhân</b>	Sinh viên / giảng viên		
<b>Tiền điều kiện</b>	Đã đăng nhập hệ thống		
<b>Mục đích sử dụng</b>	Cho phép người dùng tạo công việc		
<b>Sự kiện kích hoạt</b>	Người dùng click vào floating button add		
<b>Luồng hoạt động chính</b>	<b>Thứ tự thao tác</b>	<b>Được thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Người dùng click vào floating button add
	2.	Hệ thống	Chuyển đến trang tạo mới công việc
	3.	Người dùng	Điền các thông tin trong trang tạo mới
	4.	Người dùng	Click vào icon done
	5.	Hệ thống	Lưu lại thông tin công việc
<b>Sự kiện thay thế</b>	Không		
<b>Hậu điều kiện</b>	Thêm một công việc mới và cập nhật lại danh sách các công việc trong trang các công việc		

**Bảng 2.3:** Bảng đặc tả use case Tạo công việc.

Danh sách các trường dữ liệu đầu vào form "Tạo công việc"

Thứ tự	Trường dữ liệu	Mô tả	Bắt buộc	Yêu cầu hợp lệ	Ví dụ
1	Title	Tên công việc	Có		Chỉnh sửa layout login
2	Description	Mô tả công việc	Không		Chỉnh lại font chữ, ảnh
3	Start_date	Thời gian bắt đầu thực hiện công việc	Có		2022-04-01
4	Due_date	Thời gian kết thúc công việc	Có		2022-04-08
5	Esti-mate_time	Thời gian ước tính hoàn thành	Có		16
6	Progress	Tiến độ thực hiện công việc	Có		0.8
7	User_post	Người tạo công việc	Có		Đinh Thuý Hà
8	Status	Trạng thái công việc	Có		Hủy

**Bảng 2.4:** Bảng dữ liệu đầu vào use case Tạo công việc.

### 2.4.3 Đặc tả use case Chỉnh sửa công việc

Mã use case	UC03	Tên use case	Chỉnh sửa công việc
Tác nhân	Sinh viên / giảng viên		
Tiền điều kiện	Đã đăng nhập hệ thống		
Mục đích sử dụng	Cho phép người dùng chỉnh sửa công việc		
Sự kiện kích hoạt	Người dùng click vào icon edit		
Luồng hoạt động chính	Thứ tự thao tác	Được thực hiện bởi	Hành động
	1.	Người dùng	Người dùng click vào icon edit
	2.	Hệ thống	Chuyển sang chế độ cho phép chỉnh sửa công việc
	3.	Người dùng	Điền các thông tin cần chỉnh sửa.
	4.	Người dùng	Click vào icon done
	5.	Hệ thống	Cập nhật lại thông tin công việc
Sự kiện thay thế	Không		
Hậu điều kiện	Người dùng cập nhật lại công việc thành công.		

**Bảng 2.5:** Bảng đặc tả use case Chỉnh sửa công việc.

Danh sách các trường dữ liệu đầu vào form "Chỉnh sửa công việc"

Thứ tự	Trường dữ liệu	Mô tả	Bắt buộc	Yêu cầu hợp lệ	Ví dụ
1	Title	Tên công việc	Có		Chỉnh sửa layout login
2	Description	Mô tả công việc	Không		Chỉnh lại font chữ, ảnh
3	Start_date	Thời gian bắt đầu thực hiện công việc	Có		2022-04-01
4	Due_date	Thời gian kết thúc công việc	Có		2022-04-08
5	Estimate_time	Thời gian ước tính hoàn thành	Có		16
6	Progress	Tiến độ thực hiện công việc	Có		0.8
7	User_post	Người tạo công việc	Có		Đinh Thuý Hà
8	Status	Trạng thái công việc	Có		Hủy

**Bảng 2.6:** Bảng dữ liệu đầu vào use case Chỉnh sửa công việc.

**2.4.4 Đặc tả use case Bình luận công việc**

<b>Mã use case</b>	UC04	<b>Tên use case</b>	Bình luận công việc
<b>Tác nhân</b>	Sinh viên / giảng viên		
<b>Tiền điều kiện</b>	Đã đăng nhập hệ thống		
<b>Mục đích sử dụng</b>	Cho phép người dùng chỉnh sửa công việc		
<b>Sự kiện kích hoạt</b>	Người dùng click vào icon edit		
<b>Luồng hoạt động chính</b>	<b>Thứ tự thao tác</b>	<b>Được thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Người dùng click vào icon edit
	2.	Hệ thống	Chuyển sang chế độ cho phép chỉnh sửa công việc
	3.	Người dùng	Điền các thông tin cần chỉnh sửa.
	4.	Người dùng	Click vào icon done
	5.	Hệ thống	Cập nhật lại thông tin công việc
<b>Sự kiện thay thế</b>	Không		
<b>Hậu điều kiện</b>	Người dùng cập nhật lại công việc thành công.		

**Bảng 2.7:** Bảng đặc tả use case Bình luận công việc.

Danh sách các trường dữ liệu đầu vào 1 bình luận

<b>Thứ tự</b>	<b>Trường dữ liệu</b>	<b>Mô tả</b>	<b>Bắt buộc</b>	<b>Yêu cầu hợp lệ</b>	<b>Ví dụ</b>
1.	content	Nội dung bình luận	Có		hadinh
2.	file_name	Tên file đính kèm	Không		hadinh
3.	file_path	Đường dẫn file đính kèm	Không		hadinh

**Bảng 2.8:** Bảng dữ liệu đầu vào use case Bình luận công việc.

**2.4.5 Đặc tả use case Xem file đính kèm**

<b>Mã use case</b>	UC05	<b>Tên use case</b>	Xem file đính kèm
<b>Tác nhân</b>	Sinh viên / giảng viên		
<b>Tiền điều kiện</b>	Đã đăng nhập hệ thống		
<b>Mục đích sử dụng</b>	Cho phép người dùng xem file đính kèm và thao tác với file		
<b>Sự kiện kích hoạt</b>	Người dùng click một file trong bình luận		
<b>Luồng hoạt động chính</b>	<b>Thứ tự thao tác</b>	<b>Được thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Người dùng click một file trong phần bình luận
	2.	Hệ thống	Trả về màn hình xem nội dung file
	3.	Người dùng	Có thể thực hiện các thao tác: phóng to, thu nhỏ nội dung, tải file
	4.	Hệ thống	Kiểm tra hành động của người dùng có phải tải file hay không
	5.	Hệ thống	Trả về nội dung được phóng to hoặc thu nhỏ
<b>Sự kiện thay thế</b>	4a.	Hệ thống	Thông báo: Tải file thành công khi file đã được tải về máy
<b>Hậu điều kiện</b>	Người dùng xem file và thao tác với file thành công.		

**Bảng 2.9:** Bảng đặc tả use case Xem file đính kèm.

**2.4.6 Đặc tả use case Xoá thông báo đồ án đã đọc**

<b>Mã use case</b>	UC06	<b>Tên use case</b>	Xoá thông báo đồ án đã đọc
<b>Tác nhân</b>	Giảng viên		
<b>Tiền điều kiện</b>	Đã đăng nhập vào hệ thống		
<b>Mục đích sử dụng</b>	Cho phép người dùng xoá danh sách thông báo về đồ án đã đọc		
<b>Sự kiện kích hoạt</b>	Người dùng chọn chức năng xoá thông báo đồ án đã đọc		
<b>Luồng hoạt động chính</b>	<b>Thứ tự thao tác</b>	<b>Được thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Click vào icon xoá
	2.	Hệ thống	Lấy danh sách thông báo đồ án người dùng đã đọc . Xoá danh sách đây trong cơ sở dữ liệu
<b>Sự kiện thay thế</b>	Không		
<b>Hậu điều kiện</b>	Xoá thành công danh sách đồ án đã đọc, trả về danh sách các thông báo đồ án chưa đọc		

**Bảng 2.10:** Bảng đặc tả use case Xoá thông báo đồ án đã đọc.



**2.4.7 Đặc tả use case Tạo lịch gặp sinh viên**

<b>Mã use case</b>	UC07	<b>Tên use case</b>	Tạo lịch gặp sinh viên
<b>Tác nhân</b>	Giảng viên		
<b>Tiền điều kiện</b>	Đã đăng nhập vào hệ thống		
<b>Mục đích sử dụng</b>	Cho phép người dùng tạo lịch gặp mặt sinh viên		
<b>Sự kiện kích hoạt</b>	Người dùng chọn chức tạo lịch gặp sinh viên		
<b>Luồng hoạt động chính</b>	<b>Thứ tự thao tác</b>	<b>Được thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Click vào icon add trong thẻ lịch hôm nay
	2.	Hệ thống	Hiện thị form tạo lịch gặp sinh viên
	3.	Người dùng	Điền đầy đủ thông tin vào form, chọn button "OK"
	4.	Hệ thống	Lưu thông tin vào cơ sở dữ liệu,
	5.	Hệ thống	Quay lại màn hình trang chủ. Thêm lịch gặp sinh viên vào danh sách lịch hôm nay
<b>Sự kiện thay thế</b>	Không		
<b>Hậu điều kiện</b>	Tạo thành công lịch gặp sinh viên và hiển thị trong danh sách lịch hôm nay		

**Bảng 2.11:** Bảng đặc tả use case Tạo lịch gặp sinh viên.

Danh sách đầu vào form "Tạo lịch gặp sinh viên"

Thứ tự	Trường dữ liệu	Mô tả	Bắt buộc	Yêu cầu hợp lệ	Ví dụ
1	Title	Tên sự kiện	Có		Hẹn gặp trao đổi Project 1
2	Name_student	Tên sinh viên cần gặp	Có		Giáp Ngọc Hiếu
3	Date	Thời gian gặp sinh viên	Có		2022-04-01
4	Start_time	Giờ bắt đầu trao đổi với sinh viên	Có		09:00
5	End_time	Giờ kết thúc quá trình trao đổi	Có		09:30

**Bảng 2.12:** Bảng dữ liệu đầu vào use case Tạo lịch gặp sinh viên.

**2.4.8 Đặc tả use case Xem danh sách đề tài đang hướng dẫn**

<b>Mã use case</b>	UC08	<b>Tên use case</b>	Xem danh sách đề tài đang hướng dẫn
<b>Tác nhân</b>	Giảng viên		
<b>Tiền điều kiện</b>	Đã đăng nhập vào hệ thống		
<b>Mục đích sử dụng</b>	Cho phép người dùng xem danh sách đề tài đang hướng dẫn và có thể xem chi tiết đề tài		
<b>Sự kiện kích hoạt</b>	Chọn kì học hiện tại		
<b>Luồng hoạt động chính</b>	<b>Thứ tự thao tác</b>	<b>Được thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Click chọn kì học hiện tại
	2.	Hệ thống	Trả về danh sách đồ án đang hướng dẫn trong kì học đã chọn
	3.	Người dùng	Chọn một môn đồ án bất kì trong danh sách
	4.	Hệ thống	Chuyển hướng đến màn hình danh sách các đề tài đang hướng dẫn của đồ án đó
<b>Sự kiện thay thế</b>	Không		
<b>Hậu điều kiện</b>	Hệ thống hiển thị danh sách các đề tài đang hướng dẫn của đồ án		

**Bảng 2.13:** Bảng đặc tả use case Xem danh sách đề tài đang hướng dẫn.

**2.4.9 Đặc tả use case Xem chi tiết đề tài**

<b>Mã use case</b>	UC09	<b>Tên use case</b>	Xem chi tiết đề tài
<b>Tác nhân</b>	Sinh viên / Giảng viên		
<b>Tiền điều kiện</b>	Đã đăng nhập vào hệ thống		
<b>Mục đích sử dụng</b>	Cho phép người dùng xem thông tin chi tiết về đề tài như : tên đề tài, mô tả đề tài, thời gian thực hiện đề tài, thông tin giảng viên hướng dẫn, thông tin sinh viên thực hiện.		
<b>Sự kiện kích hoạt</b>	Người dùng click vào button chi tiết trong thẻ đề tài		
<b>Luồng hoạt động chính</b>	<b>Thứ tự thao tác</b>	<b>Được thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Click vào button chi tiết trong thẻ đề tài
	2.	Hệ thống	Trả về màn hình chi tiết đề tài với các thông tin của đề tài
<b>Sự kiện thay thế</b>	Không		
<b>Hậu điều kiện</b>	Hệ thống hiển thị danh sách các đề tài đang hướng dẫn của đồ án		

**Bảng 2.14:** Bảng đặc tả use case Xem chi tiết đề tài.

**2.4.10 Đặc tả use case Phân công hướng dẫn đồ án**

Mã use case	UC10	Tên use case	Phân công hướng dẫn đồ án
Tác nhân	Quản trị viên		
Tiền điều kiện	Đã đăng nhập vào hệ thống		
Mục đích sử dụng	Cho phép người dùng phân công hướng dẫn đồ án		
Sự kiện kích hoạt	Không		
Luồng hoạt động chính	Thứ tự thao tác	Được thực hiện bởi	Hành động
	1.	Người dùng	Chọn một môn đồ án trong danh sách đồ án
	2.	Hệ thống	Lấy danh sách sinh viên học môn đồ án, giảng viên dạy môn đồ án
	3.	Người dùng	Chọn ra sinh viên trong danh sách sinh viên, giảng viên trong danh sách giảng viên
	4.	Người dùng	Chọn button "Gán dự án"
	5.	Hệ thống	Kiểm tra đã chọn đủ: đồ án, sinh viên, giảng viên
	6	Hệ thống	Lưu thông tin vào cơ sở dữ liệu, thông báo thành công
Sự kiện thay thế	5a	Hệ thống	Thông báo: Mời bạn chọn đầy đủ 3 trường: đồ án, sinh viên, giảng viên
Hậu điều kiện	Hệ thống hiển thị danh sách các đề tài đang hướng dẫn của đồ án		

**Bảng 2.15:** Bảng đặc tả use case Phân công hướng dẫn đồ án.

**2.4.11 Đặc tả use case Lọc danh sách phân công đồ án**

<b>Mã use case</b>	UC11	<b>Tên use case</b>	Lọc danh sách phân công đồ án
<b>Tác nhân</b>	Quản trị viên		
<b>Tiền điều kiện</b>	Đã đăng nhập vào hệ thống		
<b>Mục đích sử dụng</b>	Cho phép người dùng lọc danh sách phân công đồ án theo tên đồ án, học kì, tên sinh viên, tên giảng viên		
<b>Sự kiện kích hoạt</b>	Chọn chức năng quản lí		
Luồng hoạt động chính	<b>Thứ tự thao tác</b>	<b>Được thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Chọn chức năng quản lí
	2.	Hệ thống	Trả về màn hình quản lí
	3.	Người dùng	Có thể thực hiện các hành động: chọn kì học, chọn hoặc điền tên sinh viên, chọn hoặc điền tên giảng viên, chọn hoặc điền tên đồ án
	4.	Hệ thống	Trả về danh sách phân công đồ án mới
<b>Sự kiện thay thế</b>	Không		
<b>Hậu điều kiện</b>	Hệ thống hiển thị danh sách phân công đồ án tùy theo hành động của người dùng		

**Bảng 2.16:** Bảng đặc tả use case Lọc danh sách phân công đồ án.

**2.4.12 Đặc tả use case Xoá phân công hướng dẫn đồ án"**

<b>Mã use case</b>	UC12	<b>Tên use case</b>	Xoá phân công hướng dẫn đồ án
<b>Tác nhân</b>	Quản trị viên		
<b>Tiền điều kiện</b>	Đã đăng nhập vào hệ thống		
<b>Mục đích sử dụng</b>	Cho phép người dùng xoá phân công đồ án		
<b>Sự kiện kích hoạt</b>	Người dùng tích vào các ô checkbox ở từng hàng		
<b>Luồng hoạt động chính</b>	<b>Thứ tự thao tác</b>	<b>Được thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Tích vào ô checkbox
	2.	Người dùng	Click button "Xoá"
	3.	Hệ thống	Xoá các dòng được chọn, hiển thị danh sách mới sau khi xoá thành công.
<b>Sự kiện thay thế</b>	Không		
<b>Hậu điều kiện</b>	Hệ thống cập nhật danh sách phân công đồ sau khi xoá một số phân công đồ án		

**Bảng 2.17:** Bảng đặc tả use case Xoá danh sách phân công đồ án.

**2.4.13 Đặc tả use case Đăng xuất**

Mã use case	UC13	Tên use case	Đăng xuất
<b>Tác nhân</b>	Sinh viên/ giảng viên/ quản trị viên		
<b>Tiền điều kiện</b>	Đã đăng nhập vào hệ thống		
<b>Mục đích sử dụng</b>	Cho phép người dùng đăng xuất khỏi hệ thống		
<b>Sự kiện kích hoạt</b>	Người dùng click vào button "Đăng xuất"		
<b>Luồng hoạt động chính</b>	<b>Thứ tự thao tác</b>	<b>Được thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Click button "Đăng xuất"
	2.	Hệ thống	Xoá token, trả về màn hình đăng nhập
<b>Sự kiện thay thế</b>	Không		
<b>Hậu điều kiện</b>	Đăng xuất người dùng khỏi hệ thống		

**Bảng 2.18:** Bảng đặc tả use case Đăng xuất.**2.5 Các yêu cầu phi chức năng****2.5.1 Yêu cầu về bảo mật**

Mật khẩu của người dùng phải được mã hoá. Các request lên server phải được đính kèm token.

Sử dụng HTTPS để giao tiếp giữa server và client cho phép trao đổi thông tin một cách an toàn trên internet.

**2.5.2 Yêu cầu về hiệu năng**

Thời gian phản hồi chấp nhận được.

**2.5.3 Yêu cầu về giao diện**

Ngôn ngữ sử dụng là Tiếng Việt, giao diện thân thiện với người dùng.



## CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

### 3.1 Ngôn ngữ lập trình Kotlin

Kotlin là một ngôn ngữ lập trình được giới thiệu vào năm 2011 bởi JetBrains[1]. Từ lúc được giới thiệu cho đến khi ra mắt phiên bản 1.0, JetBrains luôn chú trọng đến tính tương hợp của Kotlin với Java. Sau này Google đã tích hợp trực tiếp ngôn ngữ Kotlin vào Android Studio phiên bản 3.0. Với những ưu điểm vượt trội của mình Kotlin ngày càng được nhiều người sử dụng. Phiên bản Kotlin mới nhất tính đến hiện tại là 1.7.

Một trong những thế mạnh lớn nhất của Kotlin như là một ứng viên để thay thế cho Java là khả năng tương tác rất tốt giữa Java-Kotlin, thậm chí có thể dùng cả hai ngôn ngữ Java và Kotlin trong cùng một dự án, tất cả mọi thứ vẫn sẽ được biên dịch như bình thường. Vì Kotlin là hoàn toàn tương thích với Java nên cũng có thể sử dụng phần lớn các thư viện của Java vào trong dự án Kotlin. Kotlin được thiết kế để các nhà phát triển Java có thể dễ dàng tiếp cận. Những người đã học qua code Java sẽ cảm thấy rằng các cú pháp của Kotlin đều rất quen thuộc. Nếu so sánh Kotlin với Java hay bất kỳ một ngôn ngữ nào khác bạn sẽ thấy Kotlin sẽ gọn gàng hơn nhiều, Kotlin đặc biệt tốt trong việc giảm số lượng code mà bạn cần phải viết. Ngoài ra, Kotlin được thiết kế để có thể thực thi trên đa nền tảng (multi-platform) từ back-end đến front-end. Điều này cho phép các nhà phát triển chỉ cần học Kotlin là có thể xây dựng các ứng dụng full-stack.

Với tất cả những ưu điểm trên, em đã chọn Kotlin làm ngôn ngữ lập trình chính để xây dựng ứng dụng Android cho front-end và Spring Boot cho back-end của đề tài này.

### 3.2 Hệ quản trị cơ sở dữ liệu MySQL

MySQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở được ưa chuộng hàng đầu, được phát triển, phân phối và hỗ trợ bởi tập đoàn Oracle. MySQL được đánh giá là một hệ quản trị cơ sở dữ liệu có tốc độ cao, ổn định, dễ dùng, có khả năng thay đổi mô hình phù hợp với điều kiện công việc, độ bảo mật cao, đa tính năng, có khả năng mở rộng và mạnh mẽ.

Về nguyên tắc MySQL hoạt động dựa trên mô hình client-server. Cốt lõi của MySQL là máy chủ MySQL, xử lý tất cả các hướng dẫn cơ sở dữ liệu hoặc các lệnh. Máy chủ MySQL có sẵn như một chương trình riêng biệt để sử dụng trong môi trường client-server. MySQL hoạt động cùng một số chương trình tiện ích hỗ trợ quản trị cơ sở dữ liệu MySQL. Các lệnh được gửi để MySQL server thông qua

máy khách được cài đặt trên máy tính.

Với những ưu điểm vượt trội sự kết hợp của MySQL với ngôn ngữ lập trình Kotlin là giải pháp lựa chọn hàng đầu trong việc xây dựng back-end.

### **3.3 Spring Boot framework**

Spring là một framework giúp cho các nhà phát triển có thể xây dựng hệ thống và chạy trên máy ảo JVM một cách thuận tiện, đơn giản, nhanh chóng nhất. Spring là một mã nguồn mở có số lượng người dùng rất cao [2].

Spring boot là một mô-đun nằm trong Spring nó cung cấp các tính năng phát triển ứng dụng nhanh để tạo và phát triển nhanh chóng các dự án độc lập với Spring [3]. Spring Boot loại bỏ các cấu hình phức tạp của Spring. Nó giúp lập trình viên đơn giản hoá quá trình lập ứng dụng và chỉ cần tập trung vào việc phát triển business cho ứng dụng. Sử dụng Spring Boot sẽ mang đến nhiều lợi ích nổi bật: hội tụ đầy đủ các tính năng của Spring, đơn giản hoá cấu hình, xây dựng các ứng dụng độc lập hỗ trợ java-jar với trình khởi động phụ thuộc, cung cấp nhiều plugin, giảm thiểu thời gian phát triển, tăng thời gian phát triển chung cho dự án, cung cấp công cụ Command Line Interface cho việc phát triển và test ứng dụng nhanh chóng từ Command Line, dễ dàng triển khai vì các ứng dụng máy chủ được nhúng trực tiếp vào ứng dụng để tránh khó triển khai đến phiên bản sản xuất mà không cần tải xuống file WAR.

Với những ưu điểm trên em đã lựa chọn xây dựng Backend của hệ thống dựa trên Spring boot.

### **3.4 Bộ công cụ Jetpack Compose**

Jetpack Compose là bộ công cụ hiện đại để xây dựng giao diện người dùng Android [4]. Công cụ này đơn giản hoá và tăng tốc độ phát triển với ít mã hơn. Việc này sẽ ảnh hưởng đến tất cả các giai đoạn phát triển : có ít mã để kiểm tra với ít bug hơn; đối với người đọc thì có ít mã để đọc, hiểu, nhận xét, và bảo trì. Compose sử dụng API khai báo (declarative), điều đó có nghĩa rằng tất cả những gì nhà phát triển làm là mô tả giao diện người dùng. Compose sẽ xử lý phần còn lại. API rất trực quan - dễ tìm hiểu và sử dụng. Với Compose, nhà phát triển sẽ tạo các thành phần nhỏ được xem như các thành phần (component) không gắn cụ thể với bất kỳ một dữ liệu hay hành động nào (stateless). Điều đó giúp dễ dàng có thể sử dụng lại, mở rộng, tùy biến. Compose có khả năng tương thích ngược với các ứng dụng Android đã phát triển theo công nghệ cũ. Nhà phát triển có thể gọi Compose từ Views hay Views từ Compose. Hầu hết các thư viện như Navigation, ViewModel, Kotlin Coroutines đều tích hợp tốt với Compose. Android Studio cũng hỗ trợ xem trước giao diện người dùng được code bằng Compose. Một điều làm cho

các nhà phát triển ưu tiên dùng dùng Compose để xây dựng giao diện là Compose mặc định sử dụng thư viện Material Design để thiết kế giao diện giúp các nhà phát triển tạo nên một ứng dụng có giao diện đẹp, tinh tế.

Ứng dụng sẽ sử dụng Jetpack Compose để xây dựng giao diện người dùng.

### 3.5 Kotlin Coroutines

Coroutines không phải là một khái niệm mới và được sử dụng trong rất nhiều ngôn ngữ khác nhau.

Coroutines [5] cho phép dừng một tính toán mà không cần chiếm giữ (blocking) luồng (thread). Coroutines về cơ bản có thể hiểu nó như một "light-weight" thread nhưng nó không phải thread mà chỉ hoạt động như một thread. Một sự khác biệt quan trọng là tại một thời điểm số thread sẽ là hạn chế do chúng rất tốn kém để duy trì, được kiểm soát bởi hệ thống, còn đối với coroutines sẽ là vô hạn, hàng nghìn coroutines có thể được bắt đầu cùng lúc. Coroutines dễ sử dụng hơn, nhưng có một vài quy tắc. Xuất phát từ ý tưởng cơ bản là các đoạn code có thể bị hoãn lại mà không chặn thread. Sự khác biệt là việc chặn một thread là không thể làm bất cứ điều gì khác, trong khi hoãn lại có thể làm những việc khác trong khi chờ đợi sự hoàn thành đoạn code bị hoãn lại. Coroutines được triển khai ở mức thấp nhất có thể. Đồng thời, một cấu trúc coroutines cấp cao hơn được cung cấp trong thư viện coroutines Kotlin.

Trong đồ án của mình em sử dụng Coroutines để xử lý bất đồng bộ ở các tác vụ đọc/ ghi, gọi API lấy dữ liệu từ back-end mà không làm ảnh hưởng tới luồng chính (main thread).

### 3.6 Dagger Hilt

Trong quá trình xây dựng một ứng dụng Android có rất nhiều thành phần, lớp phức tạp thì việc quản lý các phụ thuộc (dependencies) cho những thành phần này trở thành một bài toán cần được giải quyết. Từ bài toán này, các nhà phát triển đã nghĩ ra một kỹ thuật được gọi là Dependency Injection (DI). Tuy nhiên, để sử dụng kỹ thuật này một cách dễ dàng và hiệu quả thì không phải lập trình viên nào cũng có thể làm tốt. Nắm bắt được điều này, Google đã cho ra đời một Dependency Injection Framework với tên gọi là Dagger. Dagger đối với người mới bắt đầu để tìm hiểu thì không dễ dàng, lượng code sinh ra sau khi compile quá lớn. Dagger-Hilt [6] được phát hành gần đây trong gói Jetpack và được Google khuyến nghị sử dụng để xây dựng ứng dụng Android. Hilt cho phép sử dụng DI bằng cách cung cấp các container cho tất cả các class Android và tự động quản lý vòng đời của chúng. Được xây dựng dựa trên thư viện DI phổ biến Dagger Hilt được hưởng rất nhiều lợi ích từ độ chính xác của thời gian biên dịch, hiệu suất thời gian chạy, khả năng mở

rộng. Hilt đã trở thành một phần không thể thiếu khi xây dựng một ứng dụng, nó giúp chúng ta: khiến cho code dagger trở nên dễ dàng hơn và đơn giản hơn cho các nhà phát triển, cung cấp bộ ràng buộc khác nhau, chỉ cần quan tâm đến nơi inject dependencies và phần còn lại của tất cả code generations xảy ra bởi chính dagger bằng cách sử dụng các chú thích (annotation) và do đó loại bỏ tất cả các đoạn mã dựng sẵn.

### 3.7 Retrofit

Trước đây để thực hiện các tác vụ trao đổi thông tin qua mạng, các thư viện thường được sử dụng là Volley, AsyncTask, KSOAP,... Tuy nhiên từ khi Retrofit ra đời các thư viện khác như bị lãng quên bởi thời gian thực thi và hiệu năng đáng kinh ngạc. Retrofit [7] được định nghĩa là một type-safe HTTP client cho Java, Android, Kotlin và được phát triển bởi Square. Retrofit thậm chí còn tốt hơn vì nó siêu nhanh, cung cấp các chức năng tốt hơn, cú pháp đơn giản hơn. Retrofit hỗ trợ các nhà lập trình chuyển đổi API thành Java Interface để dễ dàng kết nối đến một địa chỉ REST trên web, dễ dàng xử lý dữ liệu JSON hoặc XML và sau đó sẽ được phân tích cú pháp thành Plain Old Java Objects (POJOs) một cách tự động.

Retrofit được xây dựng dựa trên các thư viện và công cụ mạnh mẽ và làm cho quá trình nhận, gửi, tạo các yêu cầu, phản hồi HTTP trở nên đơn giản hơn. Ngoài ra nếu muốn sử dụng các bộ chuyển đổi từ JSON thành các đối tượng Java thì cần thêm các thư viện chuyển đổi vào trong dự án.

### 3.8 Json Web Token (JWT)

Json Web Token [8] là một chuẩn mở (RFC 7519) định nghĩa cách để gói gọn và trao đổi thông tin một cách an toàn giữa các bên dưới dạng một đối tượng một đối tượng JSON. Thông tin này có thể được xác minh và tin cậy vì nó có chứa chữ ký số. JWT có thể được ký bằng một mật mã (với thuật toán HMAC) hoặc bằng một cặp public/private key với thuật toán RSA hoặc ECDSA. Một JWT có cấu trúc 3 phần được mã hóa dạng chuỗi Base64 và nối với nhau bằng một ký tự chấm “.”.

- Header: chứa các thông tin về thuộc tính của JWT ở dạng JSON (thuật toán ký, định dạng, ...).
- Payload: chứa các thông tin cần trao đổi giữa các bên ở dạng JSON.
- Signature: chữ ký số của JWT.

Trong quá trình thực hiện đồ án, JWT được sử dụng để xác thực và định quyền người dùng. Mỗi người dùng được cấp một JWT chứa thông tin định danh sau khi đăng nhập thành công. JWT này sẽ được đính kèm vào các truy vấn API, back-end sẽ xác thực JWT và sử dụng thông tin bên trong để định danh được người dùng

đang truy vấn. Từ đó có thể định quyền của người dùng đối với truy vấn đang thực hiện và phản hồi một cách an toàn và chính xác.

### 3.9 Docker

Docker [9] nền tảng phần mềm cho phép dựng, kiểm thử và triển khai ứng dụng một cách nhanh chóng. Có thể coi Docker như một máy ảo cho phép cài đặt môi trường, cấu hình hệ thống, mọi thứ cần thiết để có thể chạy chương trình.

Có 2 khái niệm chính trong Docker:

- Image: Là một ảnh đóng gói của môi trường, chứa tất cả các thành phần (tệp, thư viện, phụ thuộc, cấu hình,...) để có thể khởi chạy ứng dụng.
- Container: Là một thực thể của Image (được tạo ra khi chạy Image). Các container hoạt động độc lập nhau và với hệ điều hành chủ. Có thể khởi tạo hay gỡ bỏ một container rất dễ dàng.

Một ưu điểm lớn của Docker là khả năng chia sẻ. Nhà phát triển có thể dễ dàng tìm thấy các Image được chia sẻ bởi cộng đồng trên Docker Hub. Điều này giúp giảm bớt thời gian tạo Image so với cách thông thường, tăng tốc độ phát triển phần mềm.

Trong đồ án của mình em dùng Docker thiết lập môi trường và chạy các ứng dụng back-end: Spring Boot, MySQL, Minio.

### 3.10 Minio Object Storage

Minio [10] là một máy chủ lưu trữ phân tán hiệu năng cao, được thiết kế cho cơ sở hạ tầng đám mây riêng với quy mô lớn. Minio được thiết kế tương tự như AWS S3 nhưng thay vì sử dụng hạ tầng của nhà cung cấp dịch vụ thì các lập trình viên có thể lựa chọn tự triển khai và cấu hình máy chủ Minio. Minio được sử dụng để lưu trữ các loại dữ liệu như ảnh, video, tệp tin,... với khả năng phân quyền và lưu trữ quy mô lớn. Việc thiết lập máy chủ Minio khá đơn giản, đặc biệt là khi sử dụng Docker Image của Minio.

Với những khả năng nêu trên, em đã sử dụng Minio làm server lưu trữ và truy xuất dữ liệu người dùng.

### 3.11 Azure Cloud

Microsoft Azure [11] là nền tảng tính toán đám mây được xây dựng bởi Microsoft. Azure cung cấp các giải pháp tích hợp toàn diện cho việc xây dựng, triển khai ứng dụng đám mây. Tuy ra đời muộn hơn các nền tảng đám mây khác như AWS nhưng Azure đang dần chiếm lĩnh thị phần, khẳng định vị thế “ông lớn” công nghệ. Ngoài ra với các gói ưu đãi cho sinh viên trải nghiệm dịch vụ miễn phí cũng

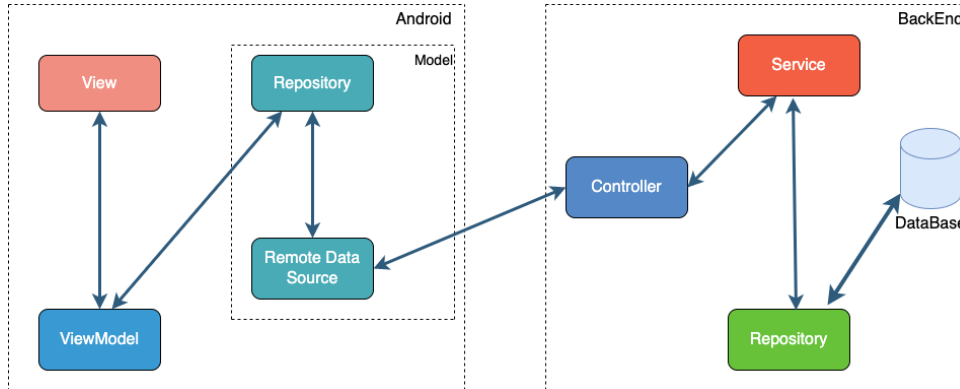
là một điểm cộng lớn cho nền tảng này.

Trong đồ án của em, toàn bộ hệ thống back-end được triển khai trên dịch vụ Virtual Machine của Azure.

## CHƯƠNG 4. PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG

### 4.1 Thiết kế kiến trúc

#### 4.1.1 Lựa chọn kiến trúc phần mềm



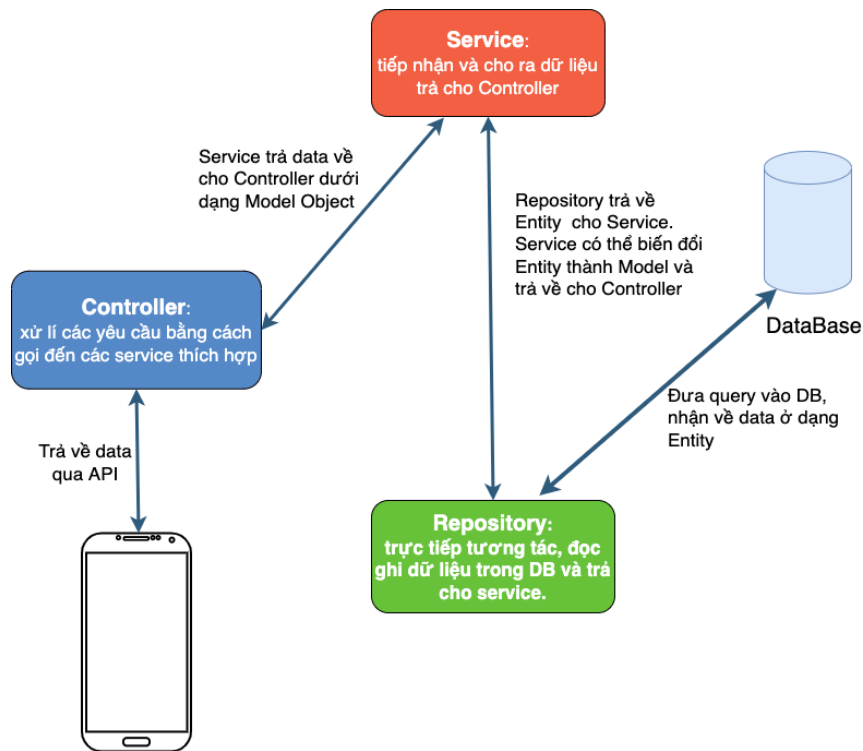
**Hình 4.1:** Kiến trúc tổng quan của hệ thống

Phần Backend của hệ thống được xây dựng bằng Spring boot, dựa trên sự kết hợp của hai mô hình MVC và mô hình ba lớp. Phần Android được xây dựng dựa trên kiến trúc MVVM.

Ưu điểm khi kết hợp hai mô hình MVC và mô hình ba lớp được sử dụng trong Backend:

- Phát triển backend nhanh, đơn giản, dễ lập trình, dễ bảo trì.
- Dễ dàng kiểm tra: dễ dàng hơn trong việc kiểm tra, rà soát lỗi do được phân tách một cách độc lập controller, service, repository.

Luồng hoạt động của mô hình: Khi người dùng gửi yêu cầu đến server, thì Controller sẽ nhận được request và bắt đầu đi hỏi Service. Service nhận được yêu cầu từ Controller đối với các tính toán đơn giản thì có thể trả về luôn, nhưng đối với các thao tác liên quan đến Database thì Service thì sẽ liên lạc với Repository để truy cập vào Database truy vấn dữ liệu trả về. Giờ sẽ là cách Server trả ngược lại dữ liệu cho người dùng: Service nhận Entity do Repository trả về biến đổi nó cho phù hợp và cuối cùng các Entity được chuyển thành Model và trả về cho Controller. Controller nhận được Model trả về cho App thông qua API.



**Hình 4.2:** Luồng hoạt động của Backend theo mô hình kết hợp giữa mô hình ba lớp và MVC

Mô hình MVVM được phát triển dựa trên cả MVC và MVP. MVVM được phát triển để tách UI ra khỏi logic nghiệp vụ. Một số ưu điểm của MVVM:

- Thực hiện Unit test một cách dễ dàng và không phụ thuộc vào View.
- Khi test không cần phải tạo mockup như MVP và chỉ cần chỉ cần xác nhận biến observable thích hợp.
- Phát triển ứng dụng nhanh, đơn giản, dễ bảo trì.

MVVM là một mô hình trong thiết kế xây dựng phần mềm. Ứng dụng được xây dựng dựa trên mô hình MVVM được chia làm 3 phần: Model, View, ViewModel. Các thành phần độc lập với nhau và có một vai trò riêng:

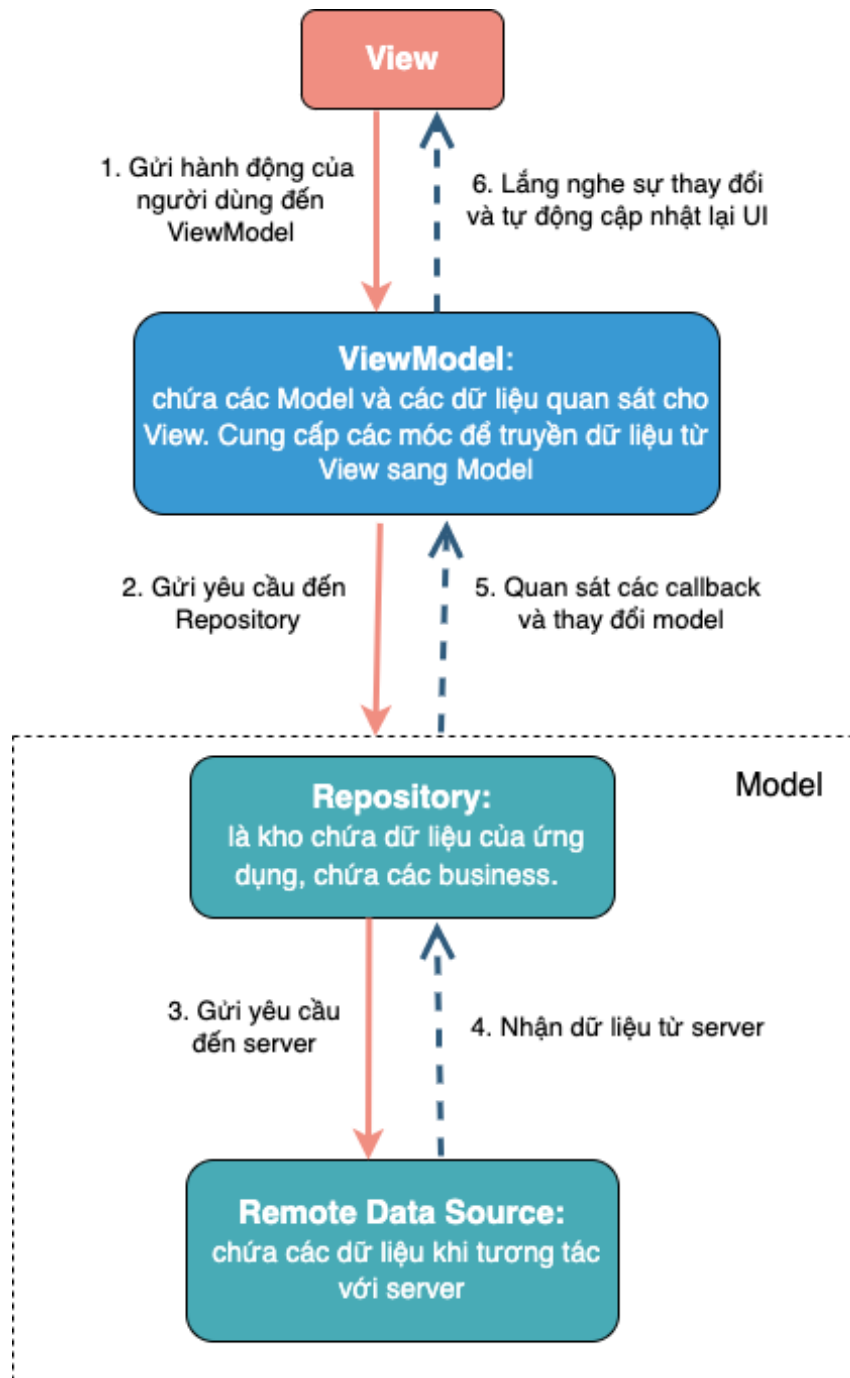
**Model:** tầng làm việc với dữ liệu cung cấp cho ứng dụng, tại đây chứa các business logic, được chia làm 2 loại: Local, Remote. Local là nguồn dữ liệu được lấy từ chính thiết bị như cơ sở dữ liệu, bộ nhớ trong, bộ nhớ ngoài, tệp, dữ liệu hệ thống. Remote là các dữ liệu được lấy từ xa như Server, Firebase RealTime database.

**View:** Tầng thực hiện nhiệm vụ hiển thị giao diện và tương tác với người dùng. View không chứa dữ liệu mà chỉ cung cấp các phương thức, thuộc tính có tính hiển thị dữ liệu. View chỉ giao tiếp với ViewModel.

**ViewModel:** có nhiệm vụ xử lý logic của ứng dụng. Giao tiếp với tầng Model

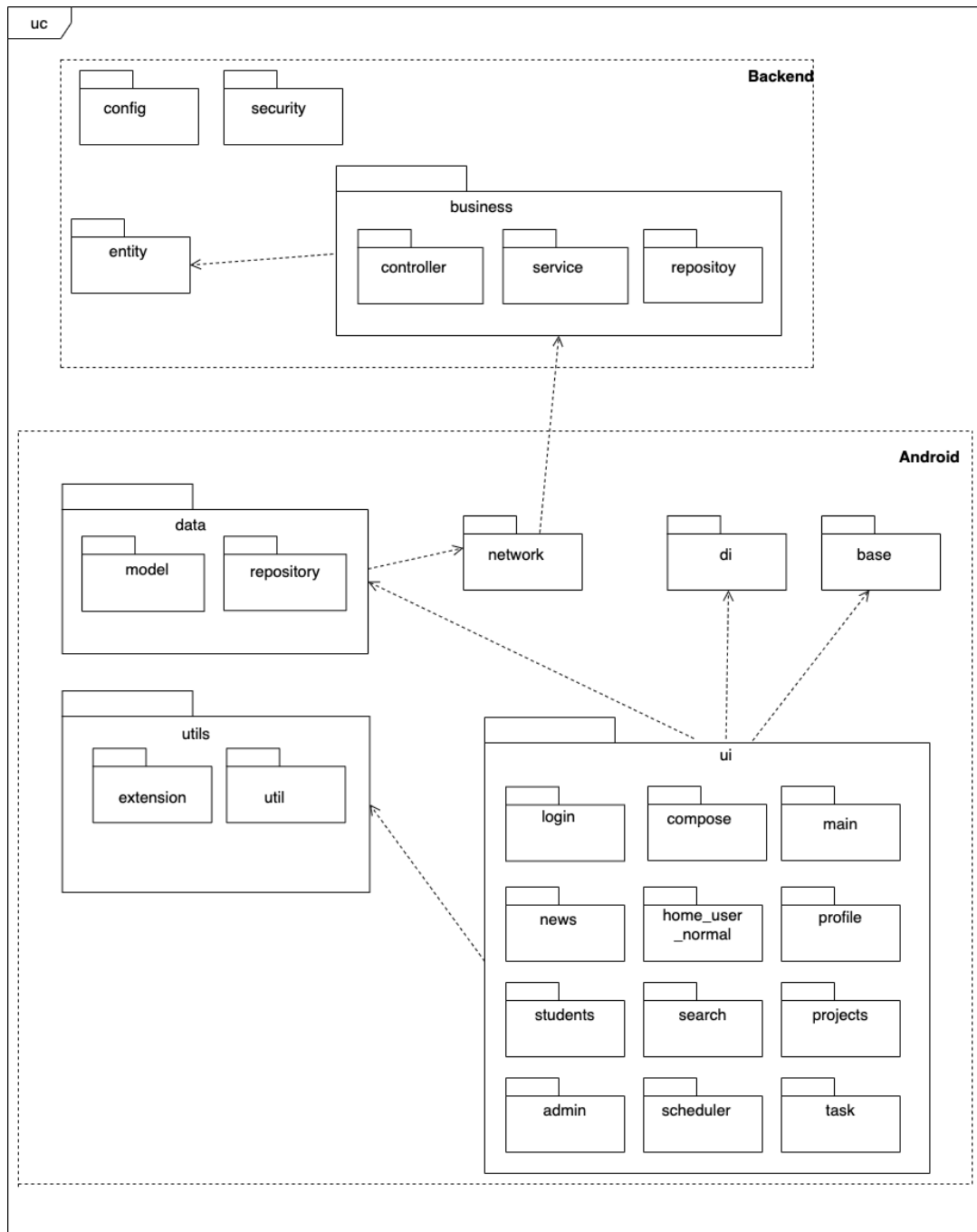


nhận dữ liệu tử đây và biến đổi trả về cho View. ViewModel lưu giữ trạng thái của View và điều khiển việc hiển thị dữ liệu.



**Hình 4.3:** Luồng hoạt động của Android theo mô hình MVVP

### 4.1.2 Thiết kế tổng quan



**Hình 4.4:** Thiết kế gói tổng quan

Hệ thống được chia làm 2 phần chính: Backend và Android.

Về phía Backend bao gồm 5 gói chính: (i) config, (ii) security, (iii) entity, (iv) business:

**Gói config:** Gói chứa các cấu hình cho spring.

**Gói security:** Gói chứa các cấu hình liên quan đến Jwt, phân quyền.

**Gói entity:** Gói entity tương ứng với các bảng trong cơ sở dữ liệu.

**Gói business:** Gói chứa các logic của server, nơi xử lý các yêu cầu và trả về kết quả cho ứng dụng.

Về phía Android bao gồm 6 gói chính: (i) di, (ii) base, (iii) data, (iv) utils, (v) network, (vi) ui:

**Gói di (Dependency Injection):** Gói chứa các lớp hỗ trợ việc dependency injection giảm thiểu sự phụ thuộc các lớp với nhau.

**Gói base:** Gói chứa các lớp cơ bản dùng chung cho nhiều màn hình, được các lớp khác kế thừa và sử dụng.

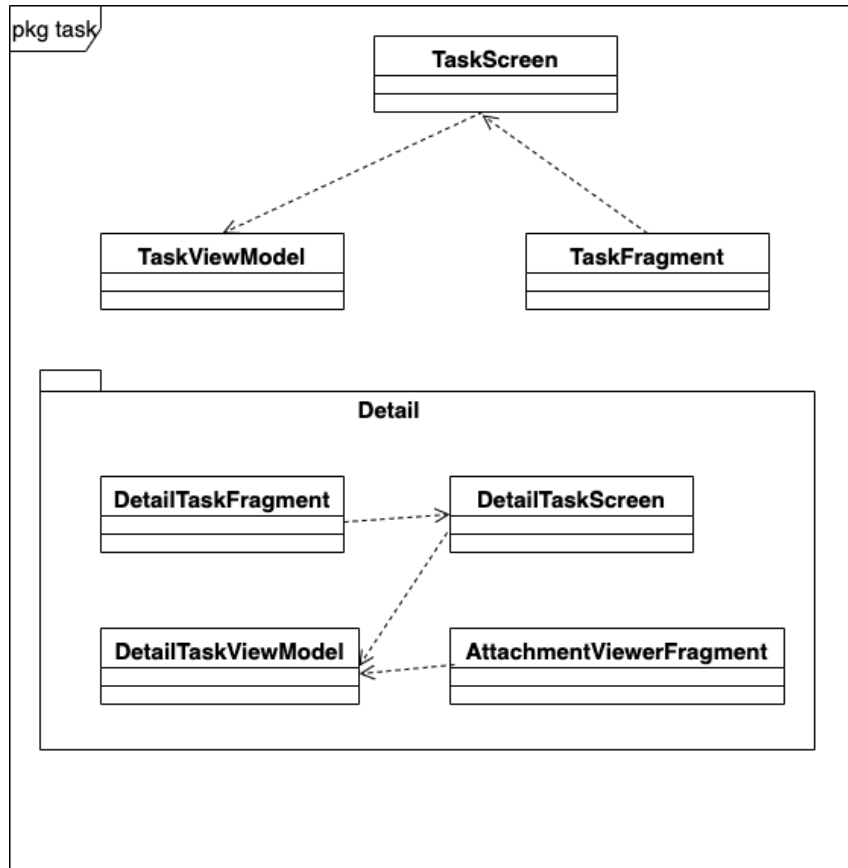
**Gói data:** Đại diện cho Model trong kiến trúc MVVM. Cung cấp dữ liệu cho ứng dụng.

**Gói network:** Gói chứa các class thực hiện việc giao tiếp với server.

**Gói utils:** Gói chứa các lớp, các chức năng được tái sử dụng nhiều lần trong hệ thống.

**Gói ui (User Interface):** Gói chứa các gói thành phần đại diện cho tầng View trong kiến trúc MVVM. Gói có nhiệm vụ chính là hiển thị dữ liệu và tương tác trực tiếp với người dùng. Bên trong gói ui là các gói thành phần, mỗi gói thành phần bao gồm View, ViewModel.

### 4.1.3 Thiết kế chi tiết gói Task

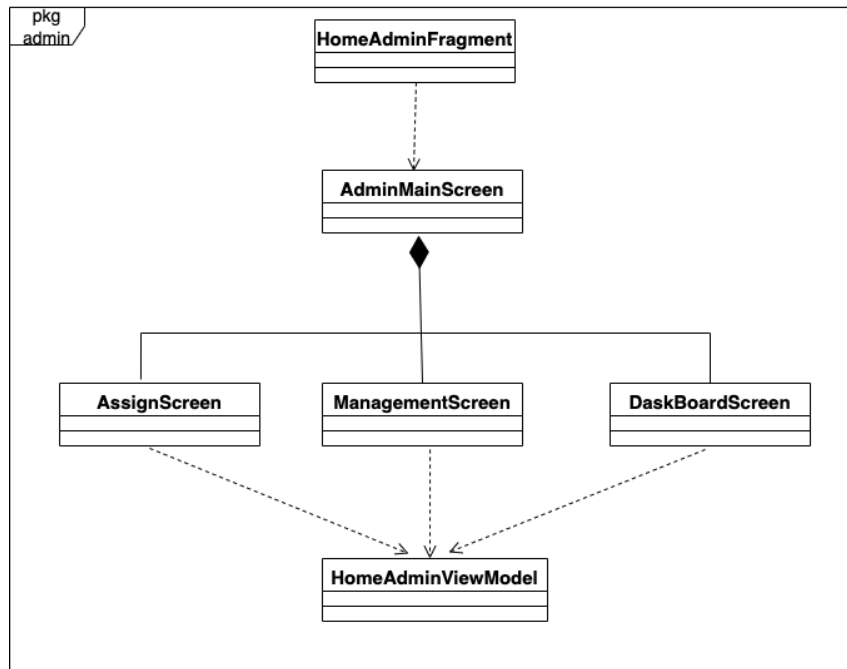


**Hình 4.5:** Thiết kế chi tiết gói Task

Hình 4.5 mô tả chi tiết gói Task. Bao gồm các lớp : (i) TaskScreen, (ii) TaskViewModel, (iii) TaskFragment, (iv) DetailTaskFragment, (v) DetailScreen, (vi) DetailTaskViewModel, (viii) AttachmentViewerFragment. Trong đó từng lớp có nhiệm vụ sau:

- TaskScreen: Nơi thiết kế giao diện của màn hình danh sách công việc.
- TaskViewModel: Nơi nắm giữ dữ liệu về danh sách công việc và thông tin chi tiết về đề tài để lớp TaskFragment hiển thị.
- TaskFragment: Màn hình hiển thị danh sách công việc, tên đề tài.
- DetailTaskFragment: Màn hình hiển thị chi tiết công việc.
- DetailScreen: Nơi thiết kế giao diện cho màn hình chi tiết công việc.
- DetailTaskViewModel: Nơi nắm giữ thông tin chi tiết của từng công việc để lớp DetailTaskFragment hiển thị.
- AttachmentViewerFragment: Màn hình hiển thị nội dung file đính kèm.

#### 4.1.4 Thiết kế chi tiết gói Admin

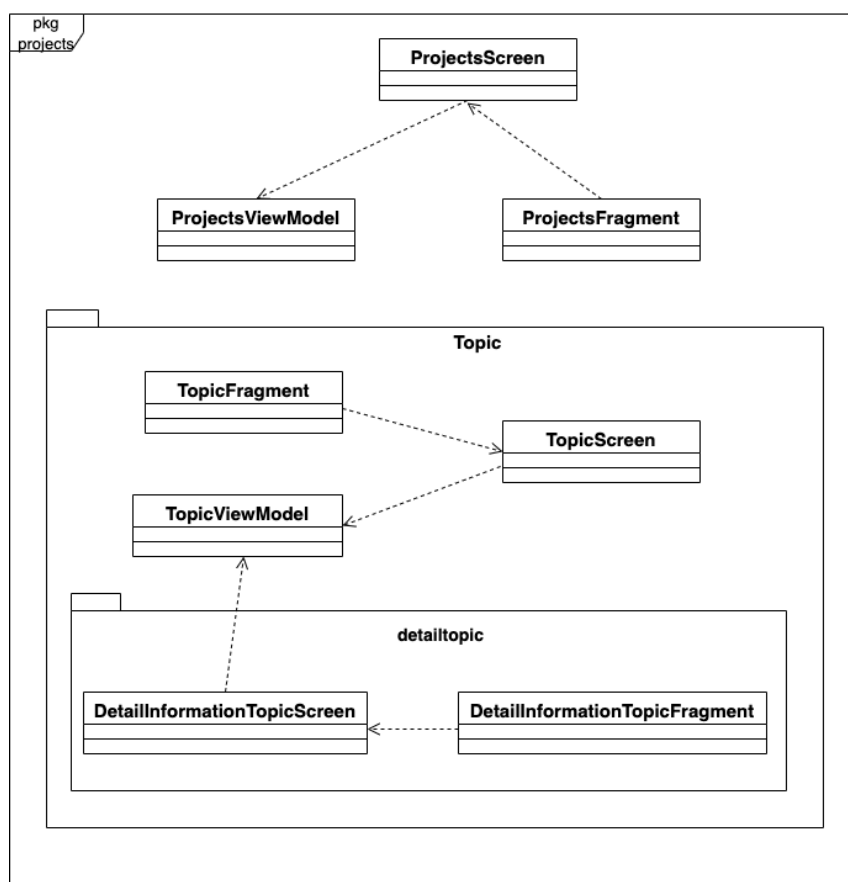


**Hình 4.6:** Thiết kế chi tiết gói Admin

Hình 4.6 mô tả chi tiết gói Admin. Bao gồm các lớp: (i) AdminMainScreen, (ii) AssignScreen, (iii) ManagementScreen, (iv) DashBoardScreen, (v) HomeAdminViewModel, (vi) HomeAdminFragment. Trong đó nhiệm vụ của từng lớp như sau:

- AdminScreen: Nơi thiết kế giao diện cho Admin. Màn hình cho admin gồm có 3 tablayout: AssignScreen, ManagementScreen, DaskBoardScreen.
- AssignScreen: Nơi thiết kế màn hình phân công hướng dẫn đồ án.
- ManagementScreen: Nơi thiết kế giao diện cho màn hình quản lý danh sách phân công đồ án.
- DashBoardScreen: Nơi thiết kế màn hình thống kê tổng số giảng viên đang được quản lý, tổng số sinh viên đăng ký đồ án.
- HomeAdminViewModel: Nơi nắm giữ dữ liệu về danh sách đồ án, danh sách sinh viên đăng ký từng môn đồ án, danh sách giảng viên, con số thống kê cần có để hiển thị lên màn hình DashBoard.
- HomeAdminFragment: Màn hình cha chứa 3 màn hình con: màn hình phân công hướng dẫn đồ án, màn hình quản lý danh sách đồ án, màn hình thống kê.

### 4.1.5 Thiết kế chi tiết gói Projects



**Hình 4.7:** Thiết kế chi tiết gói Projects

Hình 4.7 mô tả chi tiết gói Projects. Bao gồm các lớp: (i) ProjectsScreen, (ii) ProjectViewModel, (iii) ProjectFragment, (iv) TopicFragment, (v) TopicScreen, (vi) TopicViewModel, (vii) DetailTopicInformationScreen, (viii) DetailTopicInformationFragment. Trong đó nhiệm vụ của từng lớp như sau:

- ProjectsScreen: Nơi thiết kế giao diện cho màn hình danh sách đề án.
- ProjectViewModel: Nơi nắm giữ dữ liệu danh sách đề án, danh sách sinh viên, danh sách các kì học, danh sách giảng viên.
- ProjectFragment: Màn hình hiển thị danh sách đề án hướng dẫn theo kì học.
- TopicFragment: Màn hình hiển thị danh sách đề tài của một đề án.
- TopicScreen: Nơi thiết kế giao diện cho màn hình danh sách đề tài.
- TopicViewModel: Nơi nắm giữ danh sách đề tài của một đề án A để lớp TopicFragment hiển thị.
- DetailTopicInformationScreen: Nơi thiết kế giao diện cho màn hình thông tin chi tiết về đề tài: tên đề tài, mô tả đề tài, thông tin giảng viên hướng dẫn, thông

tin sinh viên thực hiện, thời gian thực hiện.

- `DetailTopicInformationFragment`: Màn hình hiển thị thông tin chi tiết về đề tài.

## 4.2 Thiết kế chi tiết

### 4.2.1 Thiết kế giao diện

Đối với việc phát triển ứng dụng thì thiết kế giao diện là đặc biệt quan trọng. Thiết kế giao diện quyết định đến hiệu năng tương tác người dùng và dựa trên một số yếu tố sau:

**Kích thước màn hình:** ứng dụng được thiết kế phù hợp với tất cả các loại điện thoại thông minh và phổ biến hiện nay. Không thực hiện fix cứng các thông số trên màn hình. Giao diện được thiết kế trên điện thoại Galaxy S10+(1440 x 3040).

**Màu sắc:** Ứng dụng có sự thống nhất về màu sắc, chỉ sử dụng hai màu chủ đạo là màu đỏ và trắng.

**Độ phân giải:** Hỗ trợ tốt nhất với các ứng dụng có độ phân giải HD trở lên.

**Thiết kế nút:** Sử dụng các icon tượng hình. Các icon được cung cấp bởi Google và một số icon được lấy từ trang Icon8. Tất cả các icon đều tuân theo chuẩn Material Design cả về màu sắc lẫn kích thước.

### 4.2.2 Thiết kế lớp

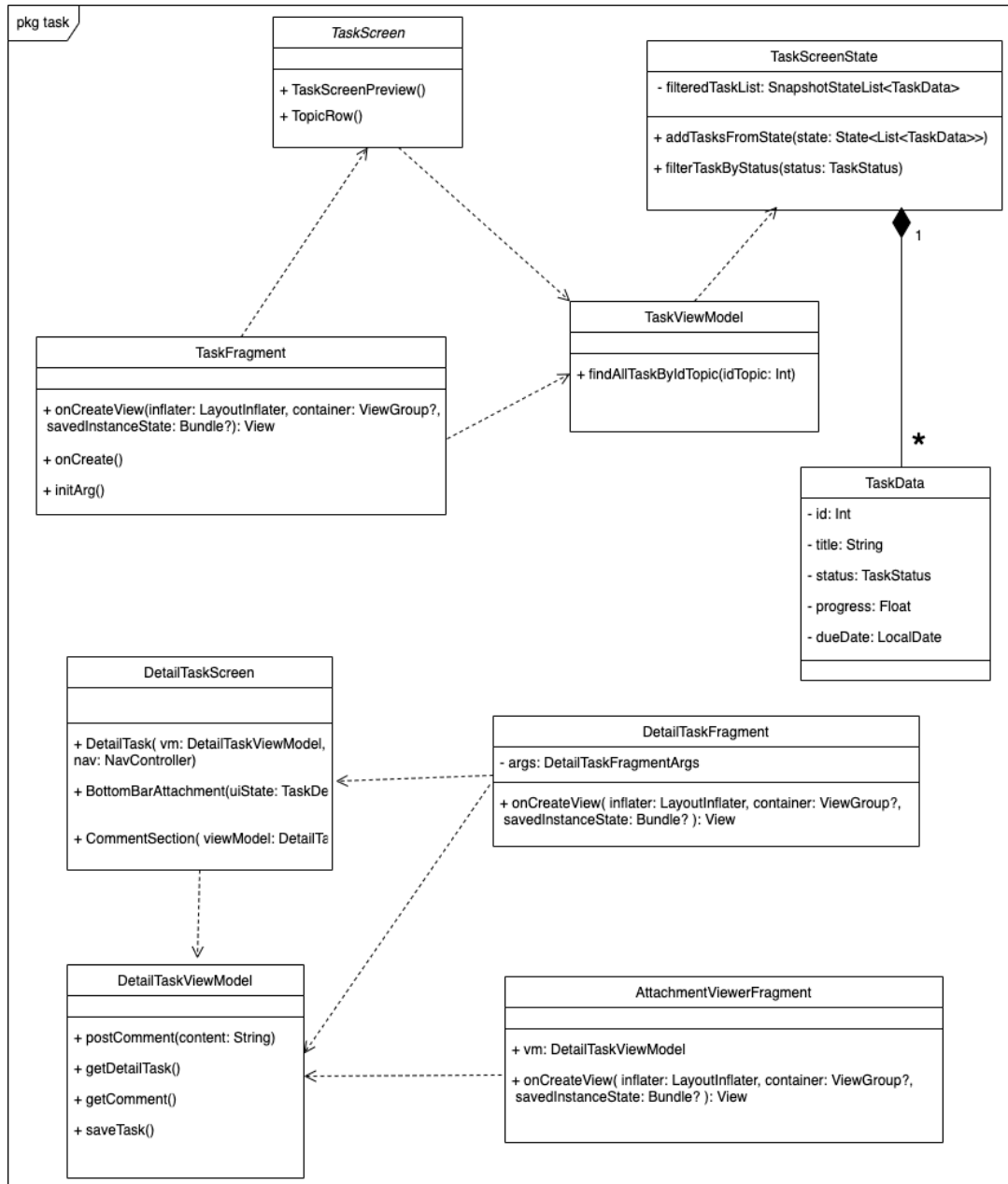
#### a, Thiết kế chi tiết lớp Task

Thiết kế chi tiết lớp Task được miêu tả như hình 4.8. Ở đây lớp `TaskViewModel` phụ thuộc vào `TaskScreenState`. `TaskScreenState` chứa danh sách `TaskData` đại diện cho công việc cần hiển thị và danh sách các công việc khi thực hiện lọc theo status thông qua hàm `filterTaskByStatus()`. Lớp `TaskViewModel` cần thực hiện công việc lấy ra danh sách công việc của topic qua phương thức `findAllTaskByIdTopic()`.

Lớp `TaskScreen` cần các tham số đầu vào là `TaskViewModel` để bind dữ liệu vào trong các `Compose`. Và sau đó để hiển thị danh sách công việc lên màn hình, `TaskScreen` sẽ được `TaskFragment` gọi qua hàm `onCreateView()` để vẽ UI lên màn hình. Hàm `initArg()` trong `TaskFragment` để lấy ra dữ liệu khi chuyển từ một màn hình A đến màn hình `TaskFragment`.

Lớp `DetailTaskScreen` chứa các `Compose` để vẽ nên giao diện hoàn chỉnh cho màn hình `DetailTaskFragment` như `DetailTask()`, `BottomBarAttachment()`, `CommentSection()`. Lớp `DetailTaskViewModel` chứa phương thức `getDetailTask()` để lấy thông tin chi tiết của công việc hiển thị lên màn hình chi tiết công việc và phương thức `getComment()` để lấy về toàn bộ bình luận có trong màn hình này.

Khi tạo mới một công việc hay chỉnh sửa một công việc thì các công việc sẽ được lưu lại thông qua hàm `saveTask()` ở `ViewModel`. Khi có bất cứ sự thay đổi nào về thông tin chi tiết của công việc thì ngay lập tức `DetailTaskFragment` sẽ được cập nhật và người dùng nhận được thông tin đã cập nhật trong tức thì. Lớp `AttachmentViewerFragment` gọi đến `ViewModel` và lấy ra tên file đính kèm, sau đó sử dụng `WebViewContainer` một `Compose Android` cung cấp sẵn để hiển thị file.



**Hình 4.8:** Thiết kế chi tiết lớp Task

### b, Thiết kế chi tiết lớp Admin

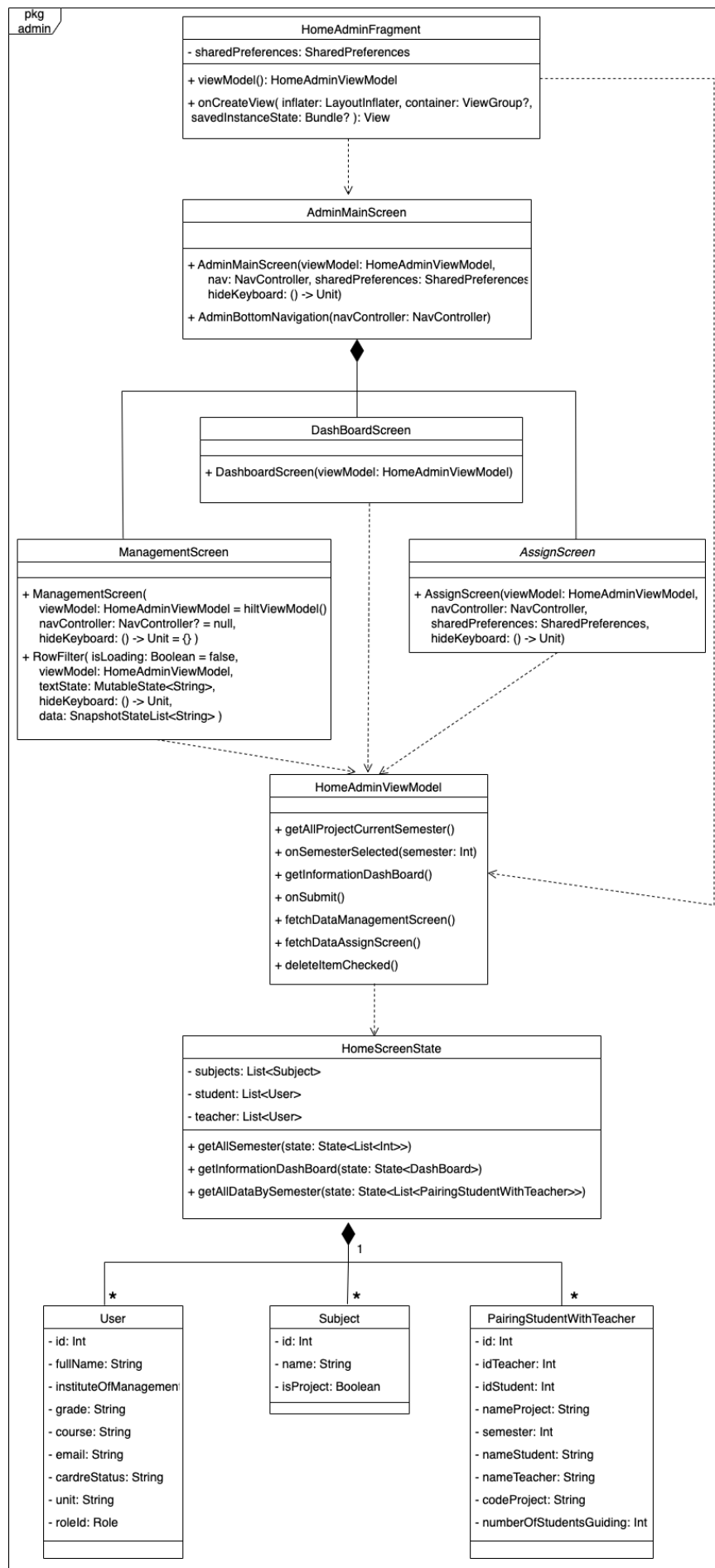
Thiết kế chi tiết lớp Task được miêu tả như hình 4.9. Ở đây lớp `AdminMainScreen` là nơi thiết kế giao diện cho admin, quy định khi nào chuyển đến các lớp `DashBoardScreen`, `ManagementScreen`, `AssignScreen`. Các lớp `DashBoardScreen`,



ManagementScreen, AssignScreen phụ thuộc vào HomeAdminViewModel. HomeAdminViewModel có chứa HomeScreenState khi state thay đổi thì nội dung trong DashboardScreen, ManagementScreen, AssignScreen tự động cập nhật theo.

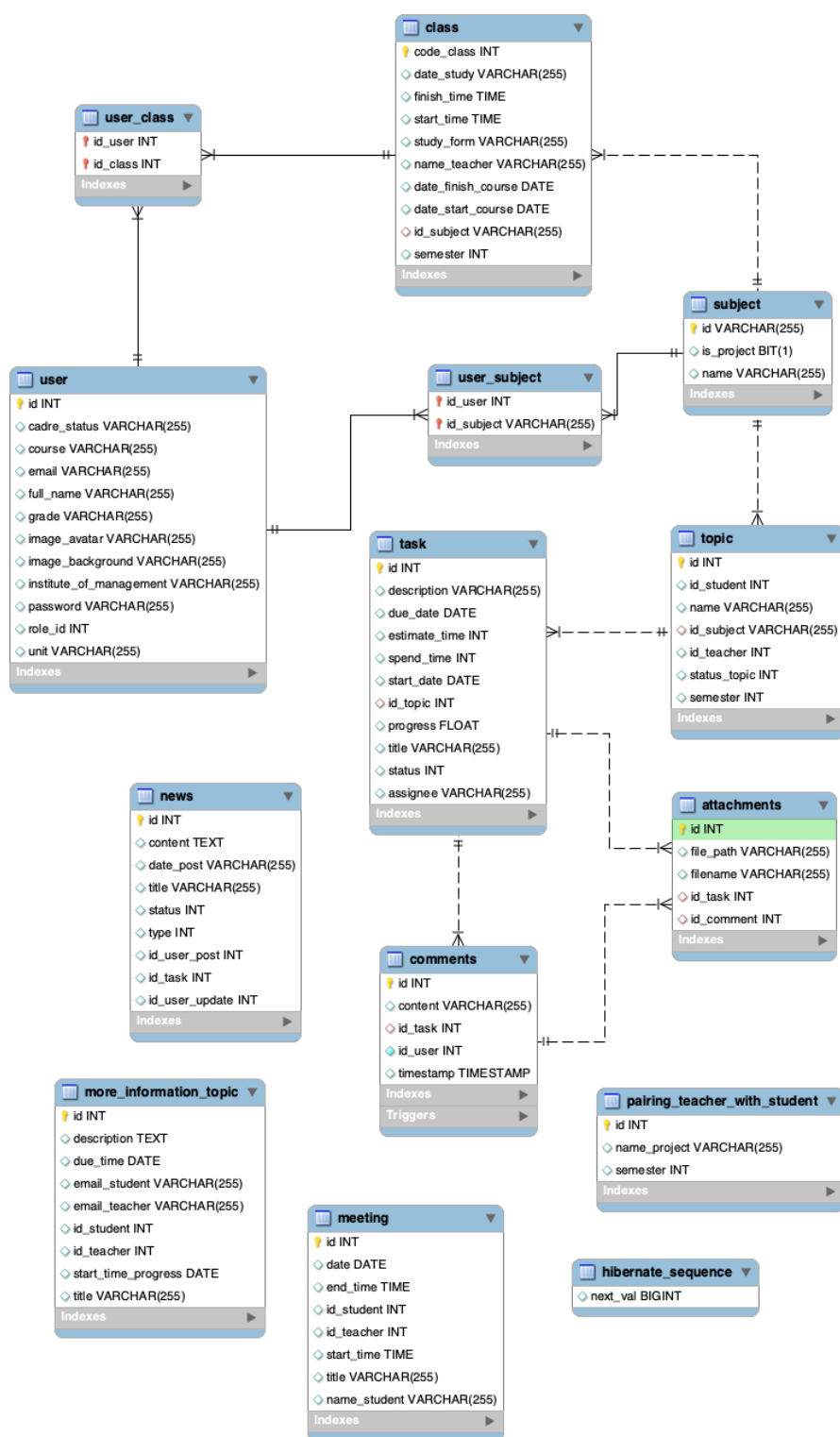
HomeViewModel chứa các hàm getAllCurrentSemester() để lấy danh sách các kì học, hàm getInformationDashboard() để lấy thông tin cho DashboardScreen, hàm fetchDataManagementScreen() để đổ dữ liệu vào cho lớp ManagementScreen, hàm fetchDataAssignScreen() để đổ dữ liệu vào cho lớp AssignScreen, hàm onSubmit() thực hiện phân công hướng dẫn đồ án. Hàm deleteItemChecked() thực hiện xóa danh sách phân công được chọn.

Lớp HomeAdminFragment gọi đến lớp AdminMainScreen trong hàm onCreateView() để vẽ UI lên màn hình. Token được lưu trong SharedPreferences sẽ được xóa khi người dùng đăng xuất ra khỏi hệ thống.



Hình 4.9: Thiết kế chi tiết lớp Admin

### 4.2.3 Thiết kế cơ sở dữ liệu



Hình 4.10: Sơ đồ thiết kế cơ sở dữ liệu

**Bảng dữ liệu User**

Mục đích: Lưu thông tin của người dùng.

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id	Int	Id của người dùng	có	Khóa chính
password	Varchar(255)	Mật khẩu của người dùng	có	
full_name	Varchar(255)	Tên đầy đủ của người dùng	có	
email	Varchar(255)	Email của người dùng	có	Duy nhất
course	Varchar(255)	Nếu là sinh viên thì thuộc khóa bao nhiêu	Không	
grade	Varchar(255)	Lớp	không	
cadre_status	Varchar(255)	Nếu là giảng viên thì trạng thái cán bộ là gì	Không	
institute_of_management	Varchar(255)	Khoa/ viện quản lí	Có	
image_avatar	Varchar(255)	Tên file làm ảnh đại diện	Không	
image_background	Varchar(255)	Tên file làm ảnh bìa	Không	
unit	Varchar(255)	Chương trình học	Có	
role_id	Int	Role_id = 1 ứng với phân quyền của quản trị viên. Role_id = 2 ứng với phân quyền của giảng viên. Role_id = 3 ứng với phân quyền của sinh viên.	Có	

**Bảng 4.1:** Bảng dữ liệu User

**Bảng dữ liệu Class**

Mục đích: Lưu thông tin lớp học.

Danh sách thuộc tính:

<b>Tên</b>	<b>Kiểu dữ liệu</b>	<b>Mô tả</b>	<b>Có bắt buộc</b>	<b>Có Ràng buộc</b>
code_class	Int	Mã môn học	Có	Khóa chính
date_study	Varchar(255)	Ngày học môn học này trong tuần: Thứ hai, hay ba...	Có	
finish_time	Time	Giờ kết thúc môn học	Có	
start_time	Time	Giờ bắt đầu môn học	Có	
study_form	Varchar(255)	Hình thức học: online hay offline	Không	
name_teacher	Varchar(255)	Tên giảng viên giảng dạy	Không	
date_finish_course	Date	Ngày kết thúc môn học	Không	
date_start_course	Date	Ngày bắt đầu môn học	Có	
id_subject	Varchar(255)	Mã môn học	Có	Khoá ngoại
semester	Int	Học kì	Có	

**Bảng 4.2:** Bảng dữ liệu Class

**Bảng dữ liệu User\_class**

Mục đích: Bảng trung gian để bề mối quan hệ nhiều nhiều giữa User, Class thành hai quan hệ một nhiều.

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id_user	Int	Id của người dùng	Có	Khóa chính
id_class	Int	Id lớp học	Có	Khoá chính

**Bảng 4.3:** Bảng dữ liệu User\_class

**Bảng dữ liệu News**

Mục đích: Lưu thông tin thông báo

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id	Int	Id thông báo	Có	Khóa chính
title	Text	Tên thông báo	Có	
content	Varchar(255)	Nội dung thông báo	Có	
date_post	Varchar(255)	Ngày đăng thông báo	Có	
status	Int	Trạng thái thông báo: 0: Chưa đọc 1: Đã đọc	Có	
type	Int	Loại thông báo: 0: Thông báo thường 1: Thông báo về đồ án	Không	
id_task	Int	Id công việc	Không	
id_user_update	Int	Id người cập nhật công việc	Không	

**Bảng 4.4:** Bảng dữ liệu News

**Bảng dữ liệu User\_subject**

Mục đích: Bảng trung gian để bề mối quan hệ nhiều nhiều giữa User, Subject thành hai quan hệ một nhiều.

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id_user	Int	Id của người dùng	Có	Khóa chính
id_subject	Varchar(255)	Mã môn học	Có	Khoá chính

**Bảng 4.5:** Bảng dữ liệu User\_subject

**Bảng dữ liệu Task**

Mục đích: Lưu thông tin công việc.

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id	Int	Mã môn học	Có	Khóa chính
title	Varchar(255)	Tên công việc	Có	
description	Varchar(255)	Mô tả chi tiết công việc	Không	
due_date	Date	Hạn hoàn thành công việc	Có	
start_date	Date	Thời gian có thể bắt đầu công việc	Có	
estimate_time	Int	Số giờ dự đoán hoàn thành	Có	
progress	Float	Tiến độ công việc	Có	
status	Int	Trạng thái công việc	Có	
name_user_create	Varchar(255)	Tên người tạo công việc	Có	
id_topic	Int	Id đề tài	Có	Khoá ngoại

**Bảng 4.6:** Bảng dữ liệu Task

**Bảng dữ liệu Topic**

Mục đích: Lưu thông tin của đề tài.

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id	Int	Id đề tài	Có	Khóa chính
id_student	Int	Id của sinh viên	Có	
name	Varchar(255)	Tên đề tài	Có	
id_subject	Varchar(255)	Mã môn học	Có	Khoá ngoại
id_teacher	Int	Thời gian có thể bắt đầu công việc	Có	
status_topic	Int	Trạng thái của đề tài( Đã được chấp nhận, hay đang yêu cầu)	Có	
semester	Int	Học kì	Có	

**Bảng 4.7:** Bảng dữ liệu Topic

**Bảng dữ liệu Comments**

Mục đích: Lưu thông tin của bình luận.

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id	Int	Id đề tài	Có	Khóa chính
content	Varchar(255)	Nội dung hình luận	Có	
id_task	Int	Id công việc	Có	Khoá ngoại
id_user	Int	Id người bình luận	Có	
timestamp	TIMESTAMP	Thời gian bình luận	Có	

**Bảng 4.8:** Bảng dữ liệu Comments



**Bảng dữ liệu Meeting**

Mục đích: Lưu thông tin chi tiết về lịch gặp mặt của sinh viên và giảng viên.

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id	Int	Id cuộc gặp giữa sinh viên và giảng viên	Có	Khóa chính
date	Date	Ngày gặp	Có	
title	Varchar(255)	Nội dung gặp mặt	Có	
name_student	Varchar(255)	Tên sinh viên gặp	Có	
id_teacher	Timestamp	Id giảng viên	Có	
id_student	Int	Id sinh viên	Có	
start_time	Time	Thời gian bắt đầu	Có	
end_time	Time	Thời gian kết thúc	Có	

**Bảng 4.9:** Bảng dữ liệu Meeting

**Bảng dữ liệu Attachments**

Mục đích: Lưu thông tin file đính kèm

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id	Int	Id file đính kèm	Có	Khóa chính
file_path	Varchar(255)	UUID của file	Có	
file_name	Varchar(255)	Tên file	Có	
id_task	Int	Id task	Có	Khoá ngoại
id_comment	Int	Id bình luận	Có	Khoá ngoại

**Bảng 4.10:** Bảng dữ liệu Attachments

**Bảng dữ liệu More\_information\_topic**

Mục đích: Lưu thông tin chi tiết của đề tài.

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id	Int	Id của đề tài	Có	Khóa chính
description	Varchar(255)	Mô tả chi tiết đề tài	Có	
title	Varchar(255)	Tên đề tài	Có	
start_time _progress	Date	Thời gian bắt đầu thực hiện đồ án	Có	
due_time	Date	Thời gian kết thúc đồ án	Có	
id_student	Int	Id sinh viên	Có	
id_teacher	Int	Id giảng viên	Có	
email_student	Varchar(255)	Email sinh viên	Có	
process_score	Float	Điểm quá trình của đồ án môn học	Không	
end_score	Float	Điểm cuối kì của đồ án môn học	Không	
state_process	Int	Trạng thái của đồ án tốt nghiệp là Đang thực hiện hay đã hoàn thành, hay chưa hoàn thành	Không	

**Bảng 4.11:** Bảng dữ liệu More\_information\_topic

**Bảng dữ liệu Pairing\_teacher\_with\_student**

Mục đích: Lưu thông tin phân hướng dẫn công đồ án.

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id	Int	Id ghép đôi giữa giảng viên và sinh viên	Có	Khóa chính
name_project	Varchar(255)	Tên đồ án	Có	
semester	Int	Học kì	Có	
id_student	Int	Id sinh viên	Có	
id_teacher	Int	Id giảng viên	Có	

**Bảng 4.12:** Bảng dữ liệu Pairing\_teacher\_with\_student

**Bảng dữ liệu Subject**

Mục đích: Lưu thông tin môn học.

Danh sách thuộc tính:

Tên	Kiểu dữ liệu	Mô tả	Có bắt buộc	Có Ràng buộc
id	Varchar(255)	Mã môn học	Có	Khóa chính
type	Int	Có phải môn đồ án hay không. Type = 0 ứng với môn học bình thường. Type = 1 ứng với đồ án môn học. Type = 2 ứng với đồ án tốt nghiệp	Có	
name	Varchar(255)	Tên môn học	Có	

**Bảng 4.13:** Bảng dữ liệu Subject

### 4.3 Xây dựng ứng dụng

#### 4.3.1 Thư viện và công cụ sử dụng

Các thư viện và công cụ sử dụng để xây dựng "Ứng dụng quản lý đồ án trên nền tảng Android":

Công cụ	Mục đích	Địa chỉ URL
Android Studio	IDE lập trình	<a href="https://developer.android.com/studio">https://developer.android.com/studio</a>
IntelliJ	IDE lập trình	<a href="https://www.jetbrains.com/idea">https://www.jetbrains.com/idea</a>
MYSQL	Hệ quản trị cơ sở dữ liệu	<a href="https://www.mysql.com">https://www.mysql.com</a>
Kotlin	Ngôn ngữ lập trình server và Android	<a href="https://kotlinlang.org/">https://kotlinlang.org/</a>
Spring boot	Framework lập trình phía server	<a href="https://spring.io/">https://spring.io/</a>
Kotlin compose	Xây dựng giao diện	<a href="https://developer.android.com/jetpack/compose">https://developer.android.com/jetpack/compose</a>
Coroutines	Thư viện hỗ trợ lập trình bất đồng bộ	<a href="https://developer.android.com/kotlin/coroutines">https://developer.android.com/kotlin/coroutines</a>
Dagger Hilt	Thư viện hỗ trợ lập trình	<a href="https://developer.android.com/training/dependency-injection/hilt-android">https://developer.android.com/training/dependency-injection/hilt-android</a>
Json web token	Xác thực và phân quyền	<a href="https://jwt.io/introduction">https://jwt.io/introduction</a>
Docker	Tạo môi trường chạy spring boot	<a href="https://www.docker.com/101-tutorial/">https://www.docker.com/101-tutorial/</a>
Minio Object storage	Server lưu trữ các file đính kèm	<a href="https://min.io">https://min.io</a>
Azure cloud	Dịch vụ cloud deploy backend	

**Bảng 4.14:** Danh sách thư viện và công cụ sử dụng

### 4.3.2 Kết quả đạt được

Mô tả	Thông tin
Số dòng code	19821
Số lượng lớp	185
Số giao diện	22
Dung lượng mã nguồn và tài nguyên	318MB
Công cụ kiểm thử	JUnit
Môi trường lập trình	MacOS

**Bảng 4.15:** Thông tin kết quả đạt được

### 4.3.3 Minh hoạ các chức năng chính

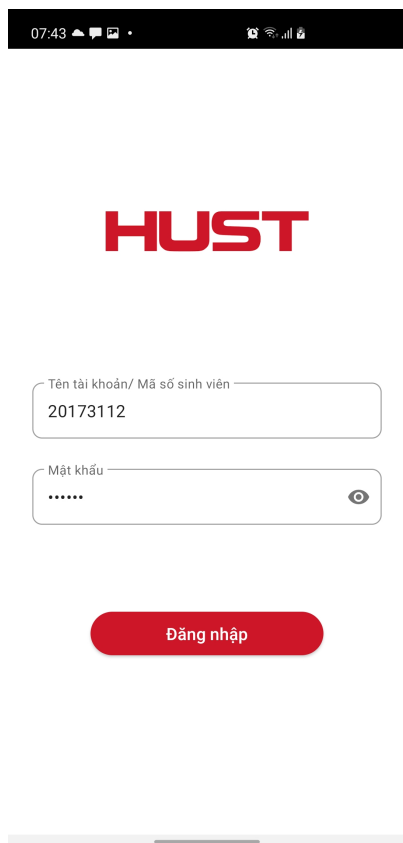
Kết quả đạt được là xây dựng thành công một ứng dụng quản lý đồ án trên nền tảng Android với các chức năng cơ bản:

Sinh viên và giảng viên sau khi đăng nhập có thể xem lịch ngày hôm nay, xem thông báo tin tức quan trọng từ trường, tạo công việc, chỉnh sửa công việc, bình luận công việc. Ngoài ra sinh viên còn có thể tra cứu thông tin sinh viên, giảng viên, lớp học, danh sách lớp, xem thời khoá biểu; giảng viên sẽ nhận được thông báo ngay sau khi sinh viên cập nhật công việc, có thể tạo lịch gặp với sinh viên, xem danh sách đồ án hướng dẫn theo kì và xem thông tin chi tiết các đề tài.

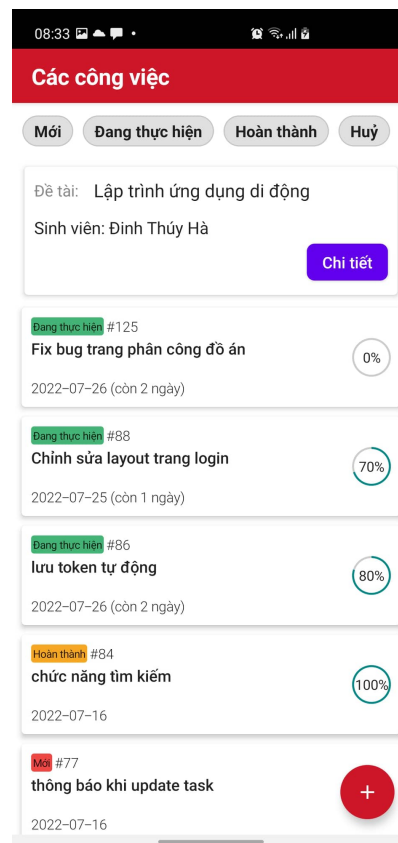
Với quản trị viên, quản trị viên có thể phân công hướng dẫn đồ án, xoá một phân công hướng dẫn bất kì nếu thấy không hợp lí, lọc danh sách phân công đồ án theo kì học, tên giảng viên, tên sinh viên, tên đồ án, thống kê tổng số sinh viên đăng kí học các môn đồ án, số giảng viên đang được quản lý tại học kì hiện tại.

Hình ảnh minh hoạ sử dụng ứng dụng:

**\*Đối với người dùng là sinh viên hoặc giảng viên:**



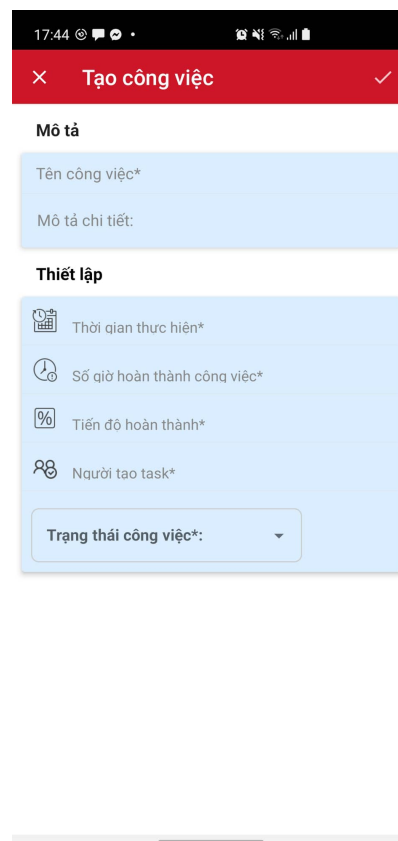
Hình 4.11: Màn hình đăng nhập



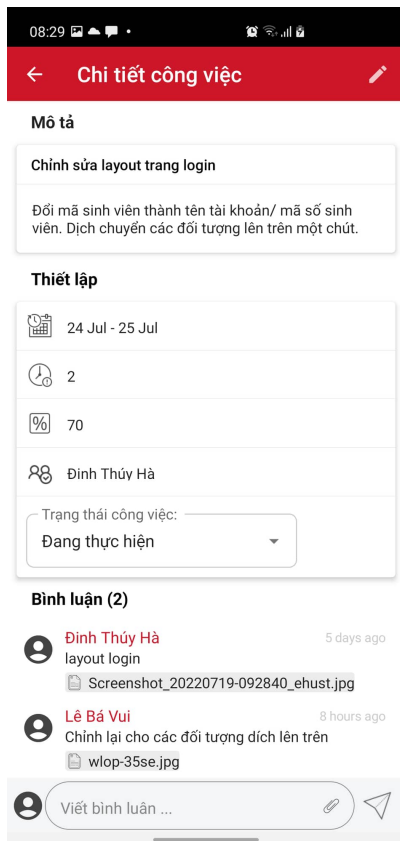
Hình 4.12: Màn hình danh sách công việc



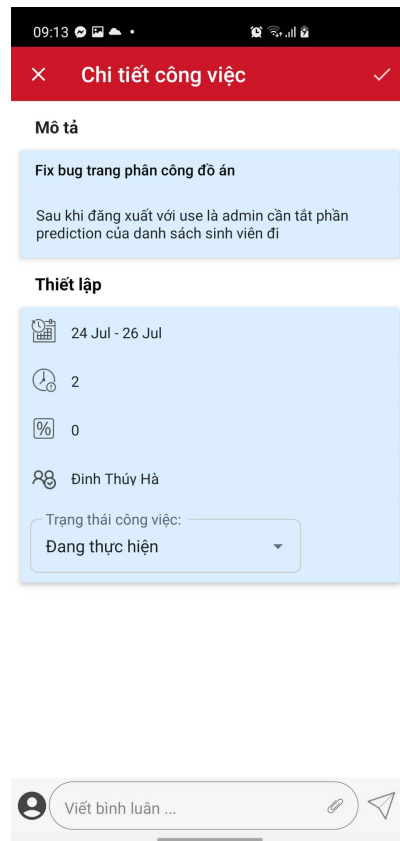
Hình 4.13: Màn hình xem chi tiết đề tài



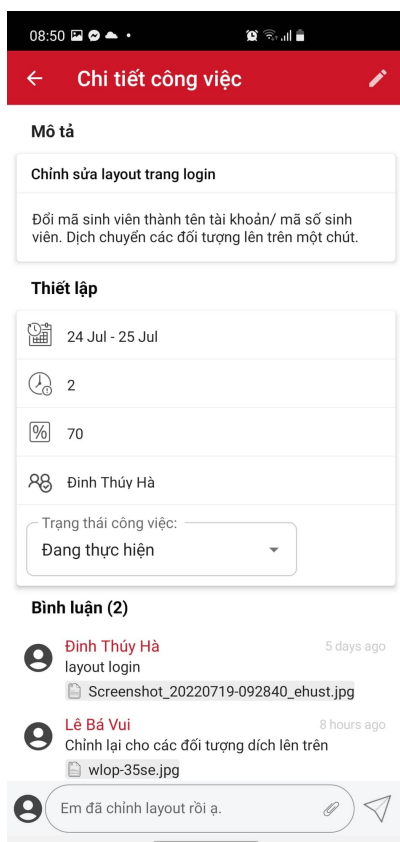
Hình 4.14: Màn hình tạo công việc



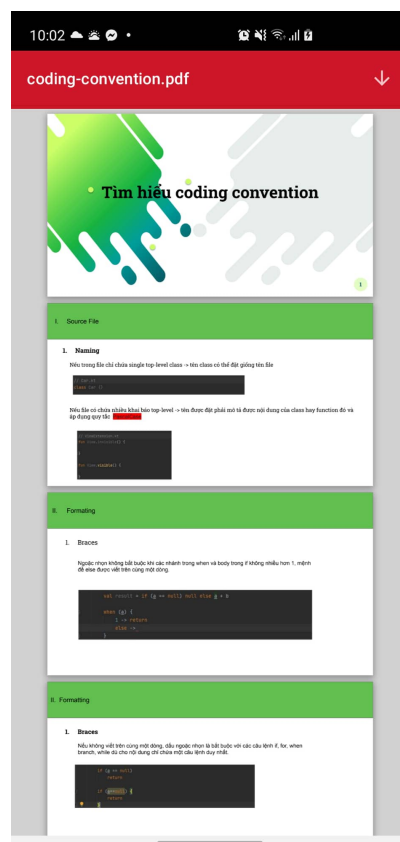
Hình 4.15: Màn hình chi tiết công việc



Hình 4.16: Màn hình chỉnh sửa công việc

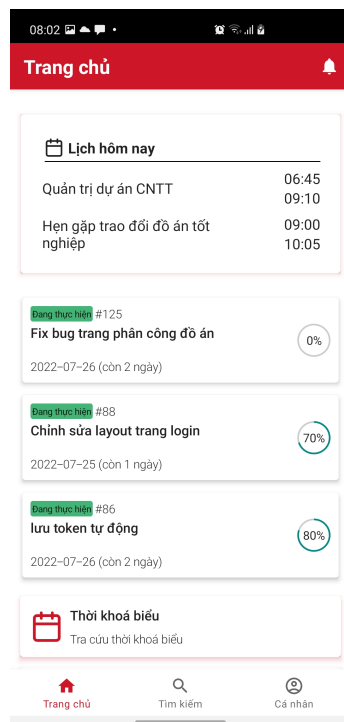


Hình 4.17: Màn hình bình luận công việc

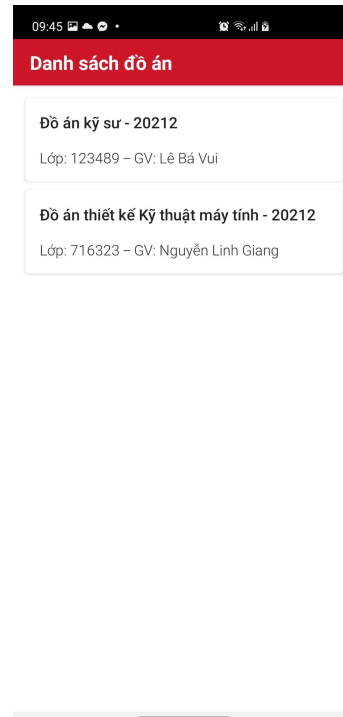


Hình 4.18: Màn hình xem file đính kèm

**\*Đối với người dùng là sinh viên:**

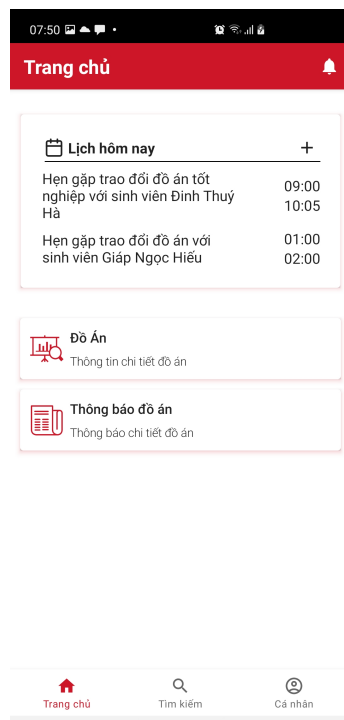


**Hình 4.19:** Màn hình trang chủ của sinh viên

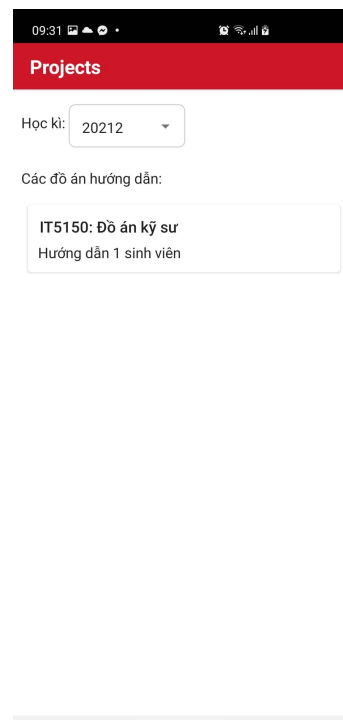


**Hình 4.20:** Màn hình danh sách đồ án đang làm

**\*Đối với người dùng là giảng viên:**



**Hình 4.21:** Màn hình trang chủ của giảng viên



**Hình 4.22:** Màn hình danh sách đồ án hướng theo học kì

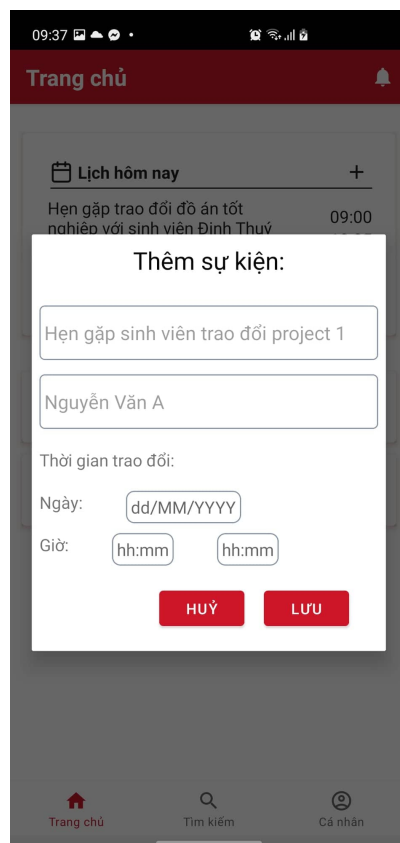




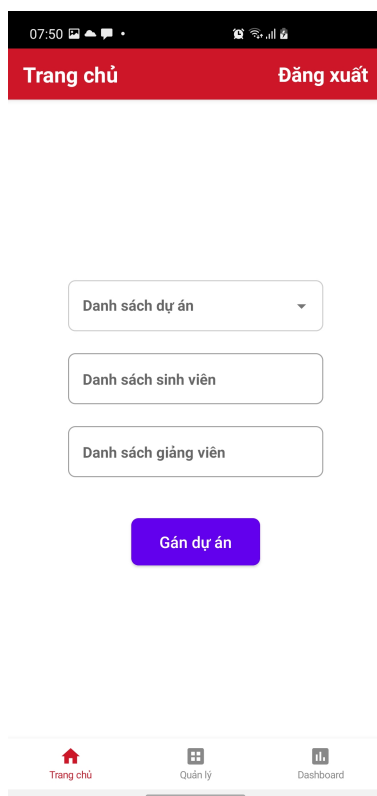
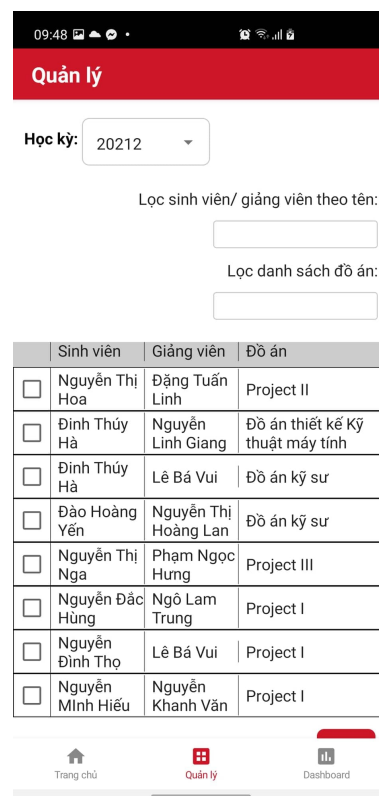
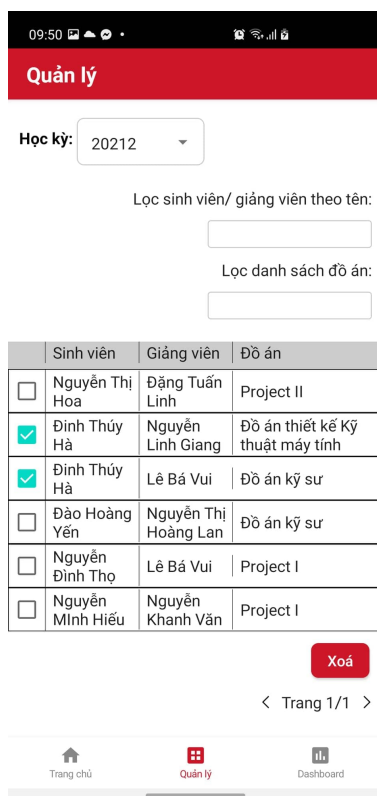
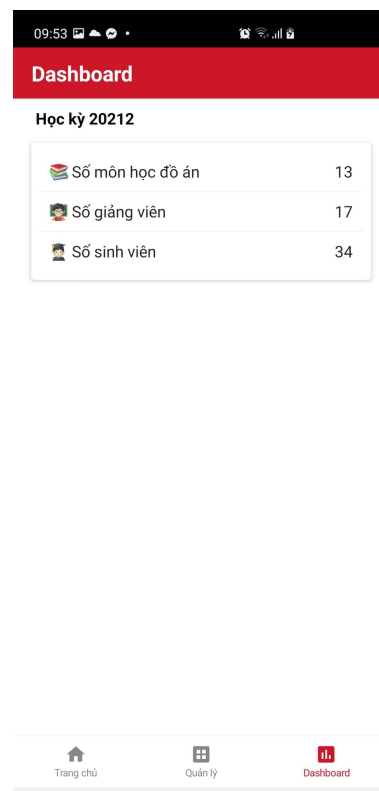
**Hình 4.23:** Màn hình danh sách đề tài hướng dẫn của Đồ án kỹ sư



**Hình 4.24:** Màn hình danh sách thông báo về đồ án



**Hình 4.25:** Màn hình tạo lịch gặp sinh viên

**\*Đối với người dùng là quản trị viên:****Hình 4.26:** Màn hình trang chủ của quản trị viên**Hình 4.27:** Màn hình quản lý phân công đồ án**Hình 4.28:** Màn hình chọn xóa phân công đồ án để xóa**Hình 4.29:** Màn hình thống kê thông tin

#### 4.4 Kiểm thử

STT	Chức năng	Đầu vào	Đầu ra	Kết quả
1	Đăng nhập	Điền đầy đủ, hợp lệ thông tin và tài khoản trùng khớp với trong cơ sở dữ liệu với role_id = 0	Đăng nhập thành công và chuyển đến trang chủ của quản trị viên	Đạt
		Điền đầy đủ, hợp lệ thông tin và tài khoản trùng khớp với trong cơ sở dữ liệu với role_id = 1	Đăng nhập thành công và chuyển đến trang chủ của giảng viên	
		Điền đầy đủ, hợp lệ thông tin và tài khoản trùng khớp với trong cơ sở dữ liệu với role_id = 2	Đăng nhập thành công và chuyển đến trang chủ của sinh viên	
		Điền thiếu một hoặc nhiều trường trong form	Hiển thị thông báo lỗi	
		Nhập sai thông tin tài khoản hoặc mật khẩu	Hiển thị thông báo lỗi	
2	Đăng xuất	Chọn Đăng xuất	Đăng xuất ra khỏi hệ thống và chuyển đến màn hình đăng nhập	Đạt
3	Xem chi tiết công việc	Chọn một công việc	Chuyển đến màn hình chi tiết công việc, có đầy đủ các thông tin của công việc	Đạt
4	Tạo công việc	Chọn button cập nhật ảnh bìa và tải ảnh lên	Hiển thị bản xem trước của ảnh bìa khi người dùng tải ảnh lên sau đó cập nhật lại ảnh bìa và hiển thị cập nhật đó ra trang cá nhân	Đạt
5	Chỉnh sửa công việc	Điền thông tin vào các trường cần cập nhật	Cập nhật thông tin thành công và hiển thị thông tin đó lên màn hình	Đạt

STT	Chức năng	Đầu vào	Đầu ra	Kết quả
6	Bình luận công việc	Điền nội dung bình luận, có thể đính kèm file	Hiển thị bình luận vừa được thêm trên màn hình	Đạt
7	Xem file đính kèm	Chọn file cần xem nội dung	Chuyển đến màn hình xem nội dung file	Đạt
		Chọn tải file	Hiển thị thông báo tải file thành công sau khi đã lưu được file về máy	
		Phóng to file	Nội dung file được phóng to	
		Thu nhỏ file	Nội dung file được thu nhỏ	
8	Xem danh sách thông báo từ trường	Click vào icon thông báo	Hiển thị danh sách thông báo từ trường	Đạt
9	Xem danh sách thông báo đồ án	Chọn thẻ thông báo đồ án	Hiển thị danh sách thông báo đồ án bao gồm các thông báo đã đọc và chưa đọc	Đạt
10	Xoá danh sách thông báo đồ án đã đọc	Chọn icon xoá trên thành actionBar	Danh sách thông báo đồ án chỉ hiển thị các thông báo chưa đọc	Đạt
11	Tạo lịch gặp sinh viên	Chọn icon tạo lịch gặp sinh viên	Mở ra dialog để giảng viên điền thông tin	Đạt
		Điền thông tin đầy đủ vào các trường trong dialog, chọn button "OK"	Ẩn dialog, thông báo tạo lịch gặp thành công	
		Điền thiếu một trong các trường, chọn button "OK"	Thông báo bắt người dùng điền đầy đủ các trường	
12	Xem danh sách đồ án hướng dẫn theo kì	Chọn kì học	Hiển thị danh sách đồ án hướng dẫn theo kì	Đạt

STT	Chức năng	Đầu vào	Đầu ra	Kết quả
13	Xem chi tiết đề tài	Click button "Chi tiết" trong thẻ đề tài trong màn hình các công việc	Chuyển đến màn hình thông tin chi tiết đề tài	Đạt
14	Phân công hướng dẫn đồ án	Chọn đầy đủ các trường trong màn hình phân công đồ án, click button "Phân công"	Thông báo thành công, reset các trường về trạng thái ban đầu	Đạt
		Chọn thiếu một trong ba trường, click button "Phân công"	Thông báo mời người dùng phải chọn đầy đủ cả ba trường	
15	Lọc danh sách phân công đồ án	Chọn kì học trong danh sách kì học	Trả về danh sách phân công theo kì	Đạt
		Chọn một sinh viên trong danh sách sinh viên	Trả về danh sách phân công lọc theo sinh viên	
		Chọn một giảng viên trong danh sách giảng viên	Trả về danh sách phân công lọc theo giảng viên	
		Chọn đồ án trong danh sách đồ án	Trả về danh sách phân công lọc theo tên đồ án	
16	Xoá danh sách phân công đồ án	Chọn phân công cần xoá, chọn button "Xoá"	Trả về danh sách phân công không có các phân công đã chọn xoá	Đạt
17	Xem thống kê tổng số sinh viên đang đăng kí các môn đồ án, tổng số giảng viên đang quản lí	Chọn tab thống kê	Trả về màn hình thống kê tổng số sinh viên đang đăng kí các môn đồ án, tổng số giảng viên đang quản lí	Đạt
18	Tìm kiếm người dùng	Điền id của người dùng.	Trả về thông tin cơ bản của người dùng: tên, id. hoặc trả về thông báo "Không tìm thấy kết quả phù hợp"	Đạt

STT	Chức năng	Đầu vào	Đầu ra	Kết quả
19	Tìm kiếm lớp	Điền mã lớp	Trả về thông tin lớp: mã lớp, tên môn học, hình thức học, mã lớp, kì học. Hoặc trả về thông báo "Không tìm thấy kết quả phù hợp"	Đạt
20	Xem thời khoá biểu	Chọn thẻ thời khoá biểu	Chuyển sang màn hình xem thời khoá biểu	Đạt

**Bảng 4.16:** Bảng kết quả kiểm thử

## 4.5 Triển khai

Backend được triển khai trên Azure Virtual Machine Ubuntu 18.04.6 LTS, backend có thể được sử dụng ở mọi nơi, không cần mất công sức để chạy backend mỗi lần muốn gọi API. Dưới đây là các bước cài đặt và triển khai:

### Cài đặt:

Bước 1: Cài đặt Docker Sau khi đăng nhập vào máy ảo Ubuntu qua SSH, thực hiện các lệnh sau trên cửa sổ lệnh.

Cập nhật gói: `sudo apt update` `sudo apt upgrade`

Cài đặt các gói hỗ trợ: `sudo apt-get install curl apt-transport-https ca-certificates software-properties-common`

Thêm kho ứng dụng của Docker: `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`

`sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"`

`sudo apt update`

`apt-cache policy docker-ce`

Cài đặt Docker `sudo apt install docker-ce`

Kiểm tra trạng thái dịch vụ Docker đã chạy hay chưa `sudo systemctl status docker`

Nếu chưa, có thể khởi động dịch vụ Docker bằng lệnh `sudo systemctl start docker`

Bước 2: Cài đặt Nginx Sau khi đăng nhập vào máy ảo Ubuntu qua SSH, thực

hiện các lệnh sau trên cửa sổ lệnh.

```
sudo apt update sudo apt install nginx
```

Kiểm tra trạng thái dịch vụ Nginx đã chạy hay chưa `sudo systemctl status nginx`

Nếu chưa, có thể khởi động dịch vụ Nginx bằng lệnh `sudo systemctl start nginx`

Bước 3: Cấu hình HTTPS:

Cài đặt công cụ certbot theo hướng dẫn

<https://certbot.eff.org/instructions?ws=nginx&os=ubuntubionic>

Chạy lệnh sau để yêu cầu chứng thư HTTPS và cấu hình với Nginx: `sudo certbot --nginx --register-unsafely-without-email`

Nhập tên miền của máy ảo: `ehust-vip-pr01.southeastasia.cloudapp.azure.com` và thực hiện theo hướng dẫn.

Bước 4: Cấu hình nginx Sau khi đăng nhập vào máy ảo Ubuntu qua SSH, thực hiện các lệnh sau trên cửa sổ lệnh.

```
cd /etc/nginx/site-enabled
```

Chỉnh sửa hoặc tạo mới tệp cấu hình ehust `sudo nano ehust`

Chỉnh sửa nội dung tệp cấu hình như sau:

```
server {
    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/ehust-vip-pr01.southeastasia.cloudapp.azure.com/fullchain.pem;
    # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/ehust-vip-pr01.southeastasia.cloudapp.azure.com/privkey.pem;
    # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

    ignore_invalid_headers off;
    client_max_body_size 0;
    proxy_buffering off;

    # tên miền của máy ảo
    server_name ehust-vip-pr01.southeastasia.cloudapp.azure.com;

    location /api {
        proxy_pass http://localhost:8081/api;
    }

    location / {
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Host $http_host;

        proxy_connect_timeout 300;
        # Default is HTTP/1, keepalive is only enabled in HTTP/1.1
        proxy_http_version 1.1;
        proxy_set_header Connection "";
        chunked_transfer_encoding off;
        proxy_pass http://localhost:9000;
    }
}
```

**Hình 4.30:** Cấu hình nginx

Lưu tệp cấu hình, sau đó chạy lệnh sau để khởi động lại Nginx `sudo systemctl reload nginx`

### **Triển khai:**

Bước 1: Khởi chạy các dịch vụ bằng Docker

Sau khi đăng nhập vào máy ảo Ubuntu qua SSH, thực hiện các lệnh sau trên cửa sổ lệnh.

Chạy container MySQL với lệnh sau:

```
docker run -d -e MYSQL_ROOT_PASSWORD=[*****] -p 127.0.0.1:3306:3306
--name mysql-ins mysql:8
```

Chạy container Minio với lệnh sau:

```
docker run -d -p 9000:9000 -p 9001:9001 -e "MINIO_ROOT_USER=root" -
e "MINIO_ROOT_PASSWORD=[*****]" --name minio-ins -v /minio-data:/data
minio/minio server /data --console-address ":9001"
```



Tạo dải địa chỉ mạng trong Docker và kết nối với container chạy MySQL:

```
docker network create ehust-net
```

```
docker network connect ehust-net mysql-ins
```

Bước 2: Tạo tệp thực thi Spring Boot: Tại máy local, thực hiện lệnh sau trên cửa sổ lệnh tại thư mục gốc của project:

Trên macOS, Linux: `./gradlew build`

Trên Windows: `gradlew.bat build`

Sau khi hoàn tất, tệp thực thi sẽ được lưu tại đường dẫn: `./build/output/ehust-0.0.1-SNAPSHOT.jar`

Bước 3: Tạo Docker Image và khởi chạy Spring Boot

Sao chép tệp thực thi ở bước 2 vào máy ảo Ubuntu, tạo tệp định nghĩa Dockerfile với nội dung như sau:

```
FROM azul/zulu-openjdk-alpine:11-jre
```

```
WORKDIR /opt/app
```

```
COPY ehust-0.0.1-SNAPSHOT.jar app.jar
```

```
CMD ["java", "-jar", "/opt/app/app.jar"]
```

Tạo Docker Image và chạy container với lệnh sau tại thư mục chứa tệp thực thi và tệp Dockerfile:

```
docker build -t ehust-app .
```

```
docker run -it --name ehust-ins -e SPRING_PROFILES_ACTIVE='azure' -d -p 8081:8081 --network ehust-net ehust-app
```

Để chạy thử nghiệm Ứng dụng quản lý đồ án chỉ cần cài đặt thêm APK của ứng dụng.

## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Kết luận

Xuất phát từ quá trình muốn tìm hiểu và phát triển một ứng dụng liên quan đến quản lý công việc với quy mô trường đại học để quản lý đồ án, em đã thực hiện đồ án xây dựng Ứng dụng quản lý đồ án trên nền tảng Android này. Đồ án đã đáp ứng được các yêu cầu đề ra ban đầu. Cụ thể em đã xây dựng thành công một ứng dụng chạy trên nền tảng Android để quản lý đồ án. Thông qua việc khảo sát các ứng dụng quản lý công việc và ứng dụng EHUST từ đó em đã phân tích ra được các tính năng mà ứng dụng cần có: Người dùng là sinh viên, giảng viên có thể thực hiện các chức năng: tạo mới công việc, xem chi tiết và chỉnh sửa công việc, bình luận có thể đính kèm file vào trong công việc, có thể tải file đính kèm về, xem chi tiết đề tài( có các thông tin liên quan đến đề tài, giảng viên hướng dẫn, sinh viên thực hiện, thời gian thực hiện), ngoài ra sinh viên, giảng viên còn có thể tìm kiếm sinh viên hoặc giảng viên, tìm kiếm lớp học, xem lịch hôm nay, nhận được thông báo từ trường. Đối với người dùng là sinh viên sẽ xem được thời khoá biểu và người dùng là giảng viên có thể tạo lịch gặp với sinh viên sau khi đã có sự trao đổi về mặt thời gian giữa hai bên, nhận được các thông báo về đồ án khi sinh viên đang hướng dẫn cập nhật công việc. Người dùng là quản trị viên thực hiện chức năng phân công đồ án, quản lý danh sách phân công đồ án.

Đồ án tuy đã đạt được mục tiêu đề ra tuy nhiên do kiến thức bản thân còn hạn chế và thời gian làm đồ án là có hạn nên không thể tránh khỏi các sai sót nhất định: mỗi đề tài chỉ có một sinh viên thực hiện chưa xử lý trường hợp hai sinh viên cùng làm chung trong một đề tài, chưa thêm được thành viên vào trong đề tài. Những hạn chế của đồ án sẽ được em khắc phục và hoàn thiện trong tương lai để mang đến trải nghiệm tốt nhất với người dùng.

### 5.2 Hướng phát triển của đồ án

Ứng dụng quản lý đồ án trên nền tảng Android sẽ tiếp tục xây dựng để mang lại trải nghiệm tốt hơn cho sinh viên và cán bộ trong trường với các tính năng trong tương lai đó là: cho phép giảng viên đánh giá kết quả của sinh viên cuối mỗi kì đồ án thông qua điểm số, hoàn thiện quá trình từ khi sinh viên yêu cầu làm đề tài đến khi được giảng viên chấp nhận, cho phép nhiều sinh viên làm cùng một đề tài và có thể thêm sinh viên đầy vào trong đồ án, cải thiện giao diện tinh tế phù hợp hơn với người dùng, quét mã qrcode để lấy được mã số sinh viên hay mã cán bộ, cho phép người quản trị viên tạo thông báo, tạo lớp học, cho phép sinh viên xem danh sách đề tài của các giảng viên và yêu cầu lên hệ thống khi đó giảng viên sẽ nhận

được yêu cầu của sinh viên và đưa ra quyết định xem có cho phép sinh viên làm đề tài đấy hay không, quản trị viên dựa vào quyết định của giảng viên để đưa ra quyết định cuối về phân công giảng viên hướng dẫn cho sinh viên đó.

## TÀI LIỆU THAM KHẢO

- [1] Lê Văn Tuấn, *Tìm hiểu về kotlin*. [Online]. Available: <https://viblo.asia/p/tim-hieu-ve-kotlin-ByEZkLoqlQ0> (visited on 07/25/2022).
- [2] *Tổng quan về spring framework*. [Online]. Available: <https://viblo.asia/p/tong-quan-ve-spring-framework-YWOZryEyKQ0> (visited on 07/26/2022).
- [3] *Lợi ích và hạn chế của việc sử dụng spring boot*. [Online]. Available: <https://hocspringboot.net/2021/09/25/loi-ich-va-han-che-cua-viec-su-dung-spring-boot> (visited on 07/25/2022).
- [4] L. V. Tuấn, *Why adopt compose*. [Online]. Available: <https://developer.android.com/jetpack/compose/why-adopt> (visited on 07/25/2022).
- [5] Coroutines, *Improve app performance with kotlin coroutines*. [Online]. Available: <https://developer.android.com/kotlin/coroutines/coroutines-adv> (visited on 07/25/2022).
- [6] V. Thanh, *Dependency injection với hilt*. [Online]. Available: <https://2kvn.com/dependency-injection-voi-hilt-p5f31303233> (visited on 07/25/2022).
- [7] N. T. Anh, *[android] có thể bạn chưa biết - retrofit*. [Online]. Available: [https://viblo.asia/p/android-co-the-ban-chua-biet-retrofit-9241JxG0KPM#\\_retrofit-2](https://viblo.asia/p/android-co-the-ban-chua-biet-retrofit-9241JxG0KPM#_retrofit-2) (visited on 07/25/2022).
- [8] T. V. Minh, *Tìm hiểu về json web token (jwt)*. [Online]. Available: <https://viblo.asia/p/tim-hieu-ve-json-web-token-jwt-7rVRqp73v4bP> (visited on 07/25/2022).
- [9] *Tìm hiểu về docker, các ưu điểm và kiến trúc của docker*. [Online]. Available: <https://stanford.com.vn/dao-tao/tin-chi-tiet/cagId/27/id/24595/tim-hieu-ve-docker-cac-uu-diem-va-kien-truc-cua-docker> (visited on 07/25/2022).
- [10] V. Nam, *Multi-cloud object storage*. [Online]. Available: <https://min.io/> (visited on 07/26/2022).
- [11] *Azure là gì? lập trình viên khai thác được gì từ azure*. [Online]. Available: <https://itnavi.com.vn/blog/microsoft-azure-giai-phap-dien-toan-dam-may-toi-uu-nhat-cho-doanh-nghiep-chuyen-doi-so> (visited on 07/26/2022).