

TRƯỜNG ĐẠI HỌC BÀ RỊA - VŨNG TÀU
KHOA KỸ THUẬT-CÔNG NGHỆ



ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI:
XÂY DỰNG ỨNG DỤNG QUẢN LÝ
CÔNG VIỆC CÁ NHÂN – DAILY GOALS

Trình độ đào tạo:	Đại học
Hệ đào tạo:	Chính quy
Niên khóa:	2021-2025
Năm học:	Học kỳ 1 năm 2024-2025
Ngành đào tạo:	Công nghệ thông tin
Chuyên ngành:	Lập trình ứng dụng di động và game
Giảng viên hướng dẫn:	ThS. Võ Trương Hoàng Oanh
Sinh viên thực hiện:	Võ Xuân Nghi
Mã SV:	21030904
Lớp:	DH21LT

Bà Rịa-Vũng Tàu, tháng 01 năm 2025

ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

....., ngày tháng ... năm 20...

Giảng viên hướng dẫn

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn sâu sắc đến quý thầy cô trong trường đại học Bà Rịa – Vũng Tàu, những người đã không quản ngại khó khăn, tận tình giảng dạy, truyền đạt những kiến thức, kinh nghiệm và những lời động viên quý giá cho em trong suốt quá trình học tập.

Em xin gửi lời tri ân đến quý thầy cô trong khoa Kỹ Thuật – Công Nghệ, đặc biệt là ThS. Võ Trương Hoàng Oanh, người đã tận tình hướng dẫn, góp ý và truyền đạt những kiến thức, kinh nghiệm quý báu, là nguồn cảm hứng giúp em luôn giữ vững tinh thần học hỏi, không ngừng hoàn thiện bản thân và kiên trì theo đuổi mục tiêu.

Sự động viên, góp ý cùng sự hướng dẫn tận tình của thầy cô đã giúp em vượt qua những trở ngại, hoàn thành đồ án với tất cả nỗ lực và tâm huyết. Nhờ đó, em không chỉ tiếp thu được kiến thức chuyên môn mà còn học hỏi thêm được nhiều kỹ năng quan trọng trong nghiên cứu và làm việc thực tế.

Mặc dù đã dành nhiều thời gian và cố gắng hết sức để hoàn thiện đồ án, song do kiến thức và kinh nghiệm còn hạn chế nên chắc chắn vẫn không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý quý báu từ quý thầy cô và các bạn để đồ án được hoàn thiện hơn nữa. Một lần nữa, em xin chân thành cảm ơn và kính chúc quý thầy cô dồi dào sức khỏe, luôn thành công trong sự nghiệp trồng người.

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	2
1.1. Lý do chọn đề tài.....	2
1.2. Mục tiêu nghiên cứu	2
1.3. Phạm vi nghiên cứu.....	2
1.4. Ý nghĩa thực tiễn.....	3
1.5 Công nghệ và nền tảng hỗ trợ	3
1.5.1 Android Studio	3
1.5.2 Firebase	6
1.5.3 SQLite	8
1.5.4 Cách kết hợp giữa Firebase và Sqlite	8
1.5.5 Các công cụ hỗ trợ khác	8
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	9
2.1 Xác định người dùng.....	9
2.2 Xác định chức năng.....	9
2.3 Sơ đồ usecase	10
2.4 Đặc tả usecase	11
2.4.1 Đặc tả bằng bảng	11
2.4.2 Đặc tả bằng sơ đồ tuần tự	24
2.5 Sơ đồ thực thể liên kết	29
2.6 Lược đồ CSDL	29
2.7 Các bảng dữ liệu	30
2.7.1 Bảng User	30
2.7.2 Bảng Task.....	31

CHƯƠNG 3: XÂY DỰNG HỆ THỐNG	32
3.1 Kiến trúc hệ thống	32
3.2 Xây dựng các chức năng chính	32
3.3 Thiết kế các màn hình giao diện	35
3.4 Kết nối cơ sở dữ liệu	69
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	71
4.1 Kết quả đạt được	71
4.2 Hạn chế còn tồn đọng	71
4.3 Hướng phát triển trong tương lai	71
TÀI LIỆU THAM KHẢO	72

DANH MỤC HÌNH ẢNH

Hình 2.3.1: Sơ đồ usecase tổng quát	10
Hình 2.4.2.1: Sơ đồ tuần tự usecase đăng xuất	24
Hình 2.4.2.2: Sơ đồ tuần tự usecase đặt lại mật khẩu	25
Hình 2.4.2.3: Sơ đồ tuần tự usecase đăng nhập	25
Hình 2.4.2.4: Sơ đồ tuần tự usecase đăng nhập	26
Hình 2.4.2.5: Sơ đồ tuần tự usecase tạo mới công việc	26
Hình 2.4.2.6: Sơ đồ tuần tự usecase chỉnh sửa công việc	27
Hình 2.4.2.7: Sơ đồ tuần tự usecase xóa công việc.....	27
Hình 2.4.2.8: Sơ đồ tuần tự usecase chỉnh sửa trạng thái công việc.....	28
Hình 2.4.2.9: Sơ đồ tuần tự usecase xem thống kê công việc.....	28
Hình 2.4.2.10: Sơ đồ tuần tự usecase xem thông báo công việc.....	28
Hình 3.3.1: Logo app	35
Hình 3.3.2: Màn hình đăng nhập.....	36
Hình 3.3.3: Màn hình đăng ký	37
Hình 3.3.4: Màn hình nhập email để đặt lại mật khẩu	38
Hình 3.3.5: Màn hình khi đăng ký thất bại	39
Hình 3.3.6: Màn hình khi đăng ký thành công.....	40
Hình 3.3.7: Màn hình khi đăng nhập thành công.....	41
Hình 3.3.8: Màn hình khi nhập email khôi phục mật khẩu thành công.....	42
Hình 3.3.9: Màn hình thông báo khi nhập email khôi phục mật khẩu thành công...43	
Hình 3.3.10: Màn hình thông báo trong gmail khi nhập email khôi phục mật khẩu thành công.....	44
Hình 3.3.11: Màn hình khi đã đăng nhập thành công và chưa tạo công việc	45
Hình 3.3.12: Màn hình khi đã đăng nhập thành công và đã tạo công việc	46
Hình 3.3.13: Màn hình tạo mới công việc	47

Hình 3.3.14: Màn hình tạo mới công việc – thêm deadline	48
Hình 3.3.15: Màn hình tạo mới công việc – thêm trạng thái	49
Hình 3.3.16: Màn hình tạo mới công việc – khi nhập thông tin	50
Hình 3.3.17: Màn hình tạo mới công việc thành công.....	51
Hình 3.3.18: Màn hình công việc vừa tạo.....	52
Hình 3.3.19: Màn hình khi nhấn vào nút và cập nhật trạng thái công việc	53
Hình 3.3.20: Màn hình thông báo khi công việc vừa được tạo, công việc còn 1 giờ nửa đến hạn, và danh sách công việc được gửi cố định vào 8 giờ sáng (nếu có công việc có deadline trong ngày)	54
Hình 3.3.21: Màn hình cập nhật công việc	55
Hình 3.3.22: Màn hình khi cập nhật công việc thành công	56
Hình 3.3.23: Màn hình khi ấn icon thùng rác để xóa tất cả	57
Hình 3.3.24: Màn hình khi ấn nút Delete để xóa 1 công việc.....	58
Hình 3.3.25: Màn hình thông báo lỗi khi chưa điền đầy đủ thông tin	59
Hình 3.3.26: Màn hình thông báo lỗi khi nhập hạn chót là ngày trong quá khứ	60
Hình 3.3.27: Màn hình thông báo lỗi khi nhập loại công việc quá 10 ký tự - a	61
Hình 3.3.28: Màn hình thông báo lỗi khi nhập loại công việc quá 10 ký tự - b	62
Hình 3.3.29: Màn hình thông báo lỗi khi nhập trạng thái công việc không hợp lệ ..	63
Hình 3.3.30: Màn hình xác nhận đăng xuất	64
Hình 3.3.31: Màn hình thống kê công việc khi chưa lựa chọn thời gian.....	65
Hình 3.3.32: Màn hình thống kê công việc theo ngày	66
Hình 3.3.33: Màn hình thống kê công việc theo tuần	67
Hình 3.3.34: Màn hình thống kê công việc theo tháng	68
Hình 3.4.1: Kết nối Firebase	69
Hình 3.4.2: Dữ liệu user được tải vào SQLite từ Firebase.....	69
Hình 3.4.3: Dữ liệu task được tạo bởi user	70

LỜI MỞ ĐẦU

Trong thời đại công nghệ phát triển mạnh mẽ, việc ứng dụng các giải pháp công nghệ vào quản lý công việc cá nhân đã trở thành xu hướng phổ biến. Những công cụ này không chỉ giúp con người tiết kiệm thời gian mà còn nâng cao hiệu suất và khả năng tổ chức công việc một cách khoa học.

Đề tài “Xây dựng ứng dụng quản lý công việc cá nhân – Daily Goals” được thực hiện với mục tiêu tạo ra một giải pháp tối ưu, hỗ trợ người dùng quản lý công việc hàng ngày, đặt mục tiêu và theo dõi tiến độ thực hiện. Ứng dụng không chỉ dừng lại ở việc ghi chép công việc mà còn mang đến các tính năng thông minh như nhắc nhở, đánh giá hiệu suất và tùy chỉnh theo nhu cầu của từng cá nhân.

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1. Lý do chọn đề tài

- Trong bối cảnh xã hội hiện đại, việc quản lý thời gian và công việc cá nhân trở thành yếu tố quan trọng để tăng cường hiệu quả công việc và cân bằng cuộc sống. Tuy nhiên, nhiều cá nhân gặp khó khăn trong việc lập kế hoạch, theo dõi tiến độ và đánh giá hiệu suất công việc hàng ngày. Các công cụ hỗ trợ hiện tại đa phần phức tạp hoặc không đáp ứng đầy đủ nhu cầu cá nhân hóa.
- Việc phát triển một ứng dụng quản lý công việc cá nhân như Daily Goals giúp giải quyết vấn đề này bằng cách cung cấp các tính năng đơn giản, dễ sử dụng nhưng vẫn hiệu quả trong việc lập kế hoạch, nhắc nhở và theo dõi công việc.
- Lý do cá nhân chọn đề tài này xuất phát từ thực tế bản thân thường gặp khó khăn trong việc hoàn thành công việc đúng thời hạn và mong muốn tạo ra một giải pháp hữu ích không chỉ cho bản thân mà còn cho cộng đồng.

1.2. Mục tiêu nghiên cứu

- Xây dựng một ứng dụng di động giúp người dùng quản lý công việc cá nhân, hỗ trợ lập kế hoạch, nhắc nhở và theo dõi tiến độ.
- Mục tiêu phụ:
- Tăng cường hiệu quả trong việc quản lý thời gian thông qua các chức năng thông báo nhắc nhở và thống kê công việc.
- Đảm bảo ứng dụng có giao diện thân thiện và dễ sử dụng, phù hợp với mọi đối tượng người dùng.
- Cung cấp khả năng cá nhân hóa để đáp ứng nhu cầu đặc thù của từng người dùng.

1.3. Phạm vi nghiên cứu

- Ứng dụng sẽ được thiết kế và triển khai trên nền tảng di động Android, hướng đến người dùng cá nhân.

1.4. Ý nghĩa thực tiễn

- Góp phần vào sự phát triển của các ứng dụng hỗ trợ quản lý thời gian và công việc cá nhân trên nền tảng Android.
- Hỗ trợ người dùng cải thiện khả năng lập kế hoạch và quản lý công việc hàng ngày, góp phần nâng cao hiệu suất làm việc.
- Cung cấp một công cụ đơn giản nhưng hiệu quả, phù hợp với đối tượng người dùng từ học sinh, sinh viên đến người đi làm.
- Nếu ứng dụng đạt hiệu quả tốt, có thể mở rộng và tích hợp thêm các tính năng như đồng bộ hóa dữ liệu đa nền tảng hoặc chia sẻ kế hoạch với người khác.

1.5 Công nghệ và nền tảng hỗ trợ

Android Studio để phát triển ứng dụng.

Firebase và SQLite để lưu trữ dữ liệu người dùng.

Figma và Photoshop để thiết kế giao diện.

1.5.1 Android Studio

Android Studio là môi trường phát triển ứng dụng (IDE) chính thức và mạnh mẽ nhất để phát triển ứng dụng cho nền tảng Android. Được phát triển bởi Google, Android Studio cung cấp một bộ công cụ toàn diện, giúp các lập trình viên xây dựng, kiểm tra, debug và triển khai các ứng dụng Android chuyên nghiệp trên nhiều thiết bị di động khác nhau.

Ưu điểm

- **Môi trường phát triển tích hợp (IDE) mạnh mẽ với nhiều công cụ hỗ trợ lập trình:**

Android Studio cung cấp một môi trường lập trình tích hợp đầy đủ các tính năng cần thiết cho quá trình phát triển ứng dụng Android. Trình soạn thảo mã nguồn với đèn gợi ý, tự động điền mã, thực thi mã, gỡ lỗi (debug) và các tính năng tiện ích khác giúp lập trình viên viết mã nhanh chóng và hiệu quả hơn. Ngoài ra, Android Studio còn cung cấp công cụ kiểm tra và phân tích mã nguồn, công cụ quản lý phiên bản và nhiều tiện ích khác hỗ trợ quá trình phát triển.

- **Trình thiết kế giao diện trực quan, giúp dễ dàng tạo và chỉnh sửa giao diện người dùng:**

Một trong những tính năng nổi bật của Android Studio là trình thiết kế giao diện người dùng (UI) trực quan. Lập trình viên có thể kéo thả các thành phần giao diện, xem trước và chỉnh sửa giao diện ứng dụng một cách dễ dàng mà không cần viết mã. Điều này giúp tiết kiệm thời gian và nâng cao hiệu quả trong việc thiết kế giao diện người dùng đẹp mắt và thân thiện.

- **Hệ thống xây dựng Gradle linh hoạt, quản lý thư viện và tự động hóa quy trình xây dựng:**

Android Studio sử dụng hệ thống xây dựng Gradle, một công cụ xây dựng linh hoạt và mạnh mẽ. Gradle giúp quản lý các thư viện và phụ thuộc, tự động hóa các tác vụ xây dựng, tối ưu hóa mã và thực hiện các kiểm tra chất lượng mã nguồn. Điều này giúp lập trình viên dễ dàng quản lý các phần phụ thuộc của dự án và tự động hóa các quy trình xây dựng, triển khai ứng dụng.

- **Trình giả lập Android (Emulator) cho phép kiểm tra ứng dụng trên nhiều thiết bị ảo:**

Android Studio bao gồm một trình giả lập Android mạnh mẽ, cho phép lập trình viên tạo và chạy ứng dụng trên các thiết bị ảo khác nhau với các kích thước màn hình, cấu hình phần cứng và phiên bản Android khác nhau. Điều này giúp kiểm tra và đảm bảo ứng dụng hoạt động tốt trên nhiều thiết bị khác nhau mà không cần sử dụng thiết bị vật lý.

- **Tích hợp với các dịch vụ và công cụ của Google như Firebase, Google Play Services, v.v.:**

Android Studio tích hợp sẵn với các dịch vụ và công cụ của Google như Firebase (cơ sở dữ liệu đám mây, xác thực người dùng, lưu trữ tệp, v.v.), Google Play Services (Bản đồ, Thanh toán, Xác thực người dùng, v.v.) và nhiều dịch vụ khác. Điều này giúp lập trình viên dễ dàng tích hợp các tính năng của Google vào ứng dụng một cách trơn tru.

- **Hỗ trợ nhiều ngôn ngữ lập trình như Java, Kotlin và C/C++:**

Android Studio hỗ trợ nhiều ngôn ngữ lập trình phổ biến trong phát triển ứng dụng Android, bao gồm Java, Kotlin và C/C++. Java và Kotlin là hai ngôn ngữ chính để phát triển ứng dụng Android, trong đó Kotlin ngày càng được ưa chuộng hơn nhờ cú pháp hiện đại và các tính năng an toàn hơn. Ngoài ra, lập trình viên cũng có thể sử dụng C/C++ để viết các thành phần ứng dụng yêu cầu hiệu năng cao.

Hạn chế

- **Yêu cầu phần cứng tương đối mạnh:** Android Studio đòi hỏi một máy tính có cấu hình tương đối mạnh để chạy mượt mà. Điều này có thể làm giảm hiệu suất và tốn nhiều tài nguyên hệ thống.
- **Kích thước lớn:** Android Studio có kích thước tải về khá lớn, đòi hỏi không gian lưu trữ đáng kể trên máy tính. Điều này có thể khiến việc cài đặt và cập nhật trở nên chậm chạp và tốn tài nguyên.

1.5.2 Firebase

Firebase là một nền tảng giúp phát triển các ứng dụng di động trong web. Bên cạnh đó, Firebase còn được hiểu là một dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây cloud với hệ thống máy chủ mạnh mẽ của Google.

Firebase chứa cơ sở dữ liệu mang đến khả năng code nhanh và thuận tiện hơn. Lập trình viên có thể dễ dàng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu sẵn có.

Ưu điểm

- **Sử dụng miễn phí và thuận tiện**

Firebase cung cấp những tính năng cơ bản miễn phí thông qua gói dịch vụ Spark. Với những tính năng cao hơn, cần phải trả phí khá cao. Bên cạnh đó, cách đăng ký tài khoản Firebase cũng rất đơn giản, có thể đăng ký tài khoản Firebase thông qua tài khoản Google.

- **Dễ sử dụng và tích hợp**

Firebase có giao diện trực quan và thân thiện. Sau khi đăng ký tài khoản, chỉ cần đăng nhập vào là đã có thể sử dụng các tính năng của Firebase. Ngoài ra, Firebase có một kho backend với giao diện đa dạng, phong phú giúp developer thoải mái lựa chọn.

Firebase còn có chức năng hỗ trợ tạo lập các tiêu chí nhằm giúp nâng cao thứ hạng của ứng dụng. Đồng thời, điều này còn giúp tăng khả năng khách hàng có thể tiếp cận được sản phẩm và tăng traffic.

- **Đáp ứng nhu cầu của người dùng**

Firebase chính là một phần của Google. Vì vậy, Google đã không ngừng khai thác và phát triển triệt để những điểm mạnh và các tính năng của Firebase nhằm tạo ra được những dịch vụ đáp ứng nhu cầu của người dùng.

Theo Gartner, có tới khoảng 30% doanh nghiệp sử dụng máy học (ML - Machine Learning) trong quy trình của họ. Nhận thấy được điều này, Firebase đã cung cấp cho

các lập trình viên máy học với các API có sẵn cho các tính năng khác nhau của nền tảng di động. Điều này sẽ giúp cho họ phát triển ứng dụng một cách tốt nhất.

- **Cập nhật liên tục và đa nền tảng**

Firebase liên tục cập nhật và có tốc độ phát triển rất nhanh. Nhờ đó mà lập trình viên tiết kiệm được thời gian phát triển và tiếp thị ứng dụng. Bên cạnh đó, Firebase là một nền tảng đa dịch vụ. Điều này được thể hiện thông qua việc Firebase cung cấp đa dạng các dịch vụ với mục đích phát triển website. Qua đó, nhà phát triển có thể dễ dàng lựa chọn được database Firestore hoặc Realtime theo mong muốn.

Hạn chế

- **Giới hạn về quy mô ứng dụng**

Firebase là một phần Subdomain của Google. Nền tảng này hiện đang bị chặn ở một số quốc gia, trong đó có Trung Quốc. Bên cạnh đó, tất cả cơ sở hạ tầng của nền tảng đều hoạt động trên Google Cloud. Điều này đồng nghĩa với việc người dùng không thể chạy ứng dụng Firebase trên những đơn vị cung cấp đám mây khác.

- **Khả năng tùy chỉnh hạn chế**

Firebase chỉ có cơ sở dữ liệu được lưu dưới dạng JSON và không có SQL. Điều này sẽ gây khó khăn trong việc di chuyển và xử lý dữ liệu. Hơn nữa, Firebase không phải là một mã nguồn mở. Do đó, lập trình viên không thể thay đổi được mã nguồn của Firebase, từ đó gây trở ngại cho việc cải tiến sản phẩm.

Bên cạnh đó, nhà phát triển không có quyền truy cập mã nguồn. Vì vậy, đối với những ứng dụng lớn, việc chuyển đổi sang các nhà cung cấp khác không phải là một nhiệm vụ dễ dàng. Để thực hiện được việc này thì toàn bộ backend sẽ cần phải xây dựng lại từ đầu.

- **Bảo mật còn yếu**

Nhược điểm bảo mật còn yếu của Firebase có thể được thể hiện thông qua việc tất cả cơ sở dữ liệu được lưu trữ dưới dạng JSON. JSON là một định dạng nhẹ, không phụ thuộc vào ngôn ngữ và dễ sử dụng. Tuy nhiên, trong một số trường hợp, JSON được đánh giá là độc hại và sẽ xảy ra các vấn đề liên quan đến bảo mật.

1.5.3 SQLite

SQLite là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) nhẹ, được sử dụng phổ biến vì tính đơn giản và dễ dàng tích hợp vào các ứng dụng. Khác với các hệ quản trị cơ sở dữ liệu khác, SQLite không yêu cầu cài đặt server độc lập mà lưu trữ trực tiếp trên file hệ thống.

SQLite: Lưu trữ dữ liệu cục bộ trên thiết bị, phù hợp cho các ứng dụng không yêu cầu kết nối internet hoặc để tối ưu hóa truy cập dữ liệu offline.

Ưu điểm

- Dung lượng rất nhỏ, chỉ khoảng 400 KiB, phù hợp với những ứng dụng yêu cầu bộ nhớ hạn chế.
- Không cần cài đặt hay cấu hình phức tạp, có thể sử dụng ngay sau khi tải về.
- SQLite tuân thủ tiêu chuẩn ACID, đảm bảo tính toàn vẹn của dữ liệu.
- Hỗ trợ trên nhiều nền tảng như iOS, Android, Windows và Linux.
- SQLite là mã nguồn mở và miễn phí, dễ dàng tích hợp vào các ứng dụng thương mại và phi thương mại.
- Được sử dụng rộng rãi trong các ứng dụng di động, thiết bị IoT và trình duyệt web.

Hạn chế

SQLite chỉ phù hợp với dữ liệu nhỏ, không tối ưu khi lưu trữ hoặc đồng bộ dữ liệu lớn.

1.5.4 Cách kết hợp giữa Firebase và Sqlite

Firebase: Xử lý các tính năng liên quan đến dữ liệu online, xác thực người dùng, thông báo nhắc nhở hoặc sao lưu dữ liệu lên cloud.

SQLite: Lưu trữ dữ liệu cục bộ (như danh sách công việc), đảm bảo ứng dụng vẫn hoạt động mượt mà khi không có kết nối internet.

1.5.5 Các công cụ hỗ trợ khác

Thiết kế giao diện: Figma, Photoshop.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

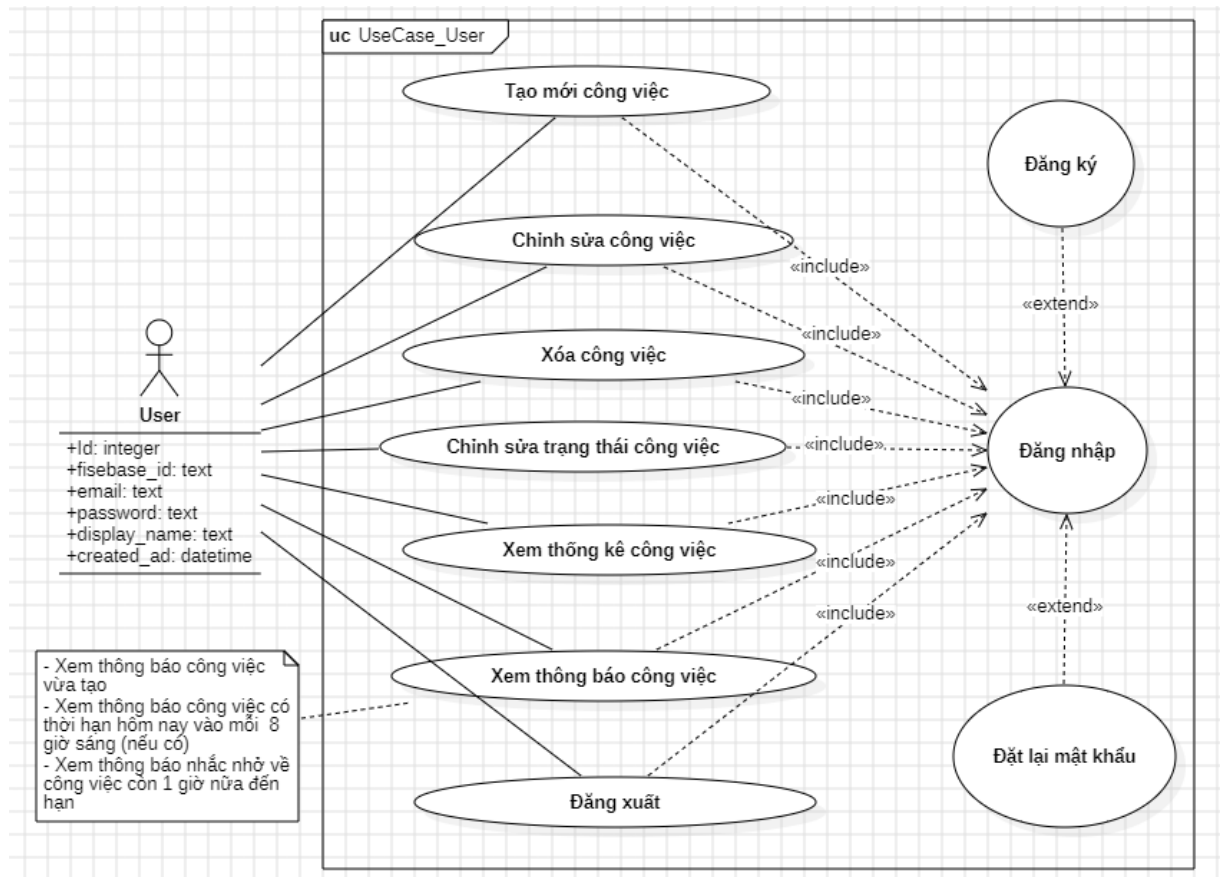
2.1 Xác định người dùng

- **Đối tượng chính:** Người dùng cá nhân có nhu cầu quản lý công việc và thời gian hàng ngày.
- **Mô tả chi tiết:**
 - Sinh viên, nhân viên văn phòng hoặc người làm tự do (freelancer).
 - Cần công cụ đơn giản, tiện lợi, hỗ trợ việc theo dõi và nhắc nhở công việc.
- **Yêu cầu từ người dùng:**
 - Giao diện thân thiện, dễ sử dụng.
 - Tính năng nhắc nhở chính xác.
 - Khả năng lưu trữ và truy cập dữ liệu nhanh chóng.

2.2 Xác định chức năng

- Tạo, sửa, xóa công việc.
- Chỉnh sửa trạng thái công việc
- Xem thống kê công việc theo ngày/tuần/tháng.
- Xem thông báo nhắc nhở công việc
- Đăng ký
- Đăng nhập
- Đặt lại mật khẩu
- Đăng xuất

2.3 Sơ đồ usecase



Hình 2.3.1: Sơ đồ usecase tổng quát

2.4 Đặc tả usecase

2.4.1 Đặc tả bằng bảng

2.4.1.1 Đặc tả usecase đăng nhập

Tóm tắt	
Tên Usecase	Đăng nhập
Mô tả chung	Use Case này cho phép người dùng cá nhân đăng nhập vào ứng dụng Daily Goals để sử dụng các chức năng chính như tạo, theo dõi, chỉnh sửa và quản lý công việc cá nhân hằng ngày. Việc đăng nhập giúp hệ thống đồng bộ dữ liệu cá nhân và lưu trữ thông tin người dùng trên nhiều thiết bị, đồng thời bảo vệ dữ liệu riêng tư.
Tác nhân	User
Độ ưu tiên	Phải có / Tiên quyết
Kích hoạt	Nhấn nút “Đăng nhập” trong app Daily Goals
Tiền điều kiện	Kết nối Internet ổn định Người dùng đã đăng ký tài khoản
Hậu điều kiện	Người dùng đăng nhập thành công Truy xuất các thông tin để hiển thị cho màn hình chính của app Daily Goals
Dòng sự kiện:	

Dòng sự kiện chính	1. Người dùng truy cập app 2. Website kiểm tra nếu đã đăng nhập thì chuyển đến Bước 5, ngược lại chuyển đến màn hình đăng nhập 3. Người dùng nhập email và mật khẩu, chọn nút “Đăng nhập” 4. Hệ thống xác minh email và mật khẩu có chính xác hay không 5. App chuyển tới màn hình chính
Dòng sự kiện ngoại lệ	4.1. Người dùng nhập thiếu thông tin, thông báo lỗi tới người dùng và yêu cầu thao tác lại 4.2. Thông tin người dùng nhập không chính xác, thông báo lỗi tới người dùng và yêu cầu thao tác lại

2.4.1.2 Đặc tả usecase đăng ký

Tóm tắt	
Tên Usecase	Đăng ký
Mô tả chung	Cho phép người dùng tạo tài khoản mới để bắt đầu sử dụng ứng dụng Daily Goals. Sau khi đăng ký, người dùng có thể đăng nhập và đồng bộ dữ liệu cá nhân trên các thiết bị.
Tác nhân	User
Độ ưu tiên	Phải có / Tiên quyết

Kích hoạt	Người dùng nhấn nút “Đăng ký” trên màn hình đăng nhập
Tiền điều kiện	Kết nối Internet ổn định Người dùng chưa có tài khoản
Hậu điều kiện	Tài khoản được tạo thành công Người dùng được chuyển đến màn hình đăng nhập sau khi đăng ký thành công
Dòng sự kiện:	
Dòng sự kiện chính	1. Người dùng truy cập màn hình đăng ký. 2. Nhập đầy đủ các thông tin như email, mật khẩu, và xác nhận mật khẩu. 3. Nhấn nút “Đăng ký”. 4. Hệ thống kiểm tra thông tin hợp lệ và tạo tài khoản mới. 5. Hiện thị thông báo thành công và chuyển tới màn hình đăng nhập
Dòng sự kiện ngoại lệ	3.1 Thông tin người dùng nhập thiếu hoặc không hợp lệ (email sai định dạng, mật khẩu quá ngắn), thông báo lỗi và yêu cầu người dùng nhập lại. 3.2 Email đã tồn tại trong hệ thống, thông báo lỗi và yêu cầu người dùng sử dụng email khác.

2.4.1.3 Đặc tả usecase đặt lại mật khẩu

Tóm tắt	
Tên Usecase	Đặt lại mật khẩu
Mô tả chung	Cho phép người dùng đặt lại mật khẩu trong trường hợp quên mật khẩu. Một liên kết đặt lại mật khẩu sẽ được gửi qua email.
Tác nhân	User
Độ ưu tiên	Phải có / Quan trọng
Kích hoạt	Người dùng nhấn “Quên mật khẩu” trên màn hình đăng nhập
Tiền điều kiện	Kết nối Internet ổn định Người dùng có tài khoản hợp lệ
Hậu điều kiện	Email đặt lại mật khẩu được gửi thành công Người dùng đặt lại mật khẩu thông qua liên kết trong email
Dòng sự kiện:	

Dòng sự kiện chính	1. Người dùng nhấn nút “Quên mật khẩu”. 2. Nhập email đã đăng ký. 3. Nhấn nút “Xác nhận”. 4. Hệ thống kiểm tra email và gửi liên kết đặt lại mật khẩu. 5. Người dùng nhận email và đặt lại mật khẩu mới thông qua liên kết.
Dòng sự kiện ngoại lệ	3.1. Email không tồn tại trong hệ thống, thông báo lỗi và yêu cầu người dùng kiểm tra lại. 3.2. Kết nối Internet bị gián đoạn, thông báo lỗi và yêu cầu thử lại sau.

2.4.1.4 Đặc tả usecase tạo mới công việc

Tóm tắt	
Tên Usecase	Tạo mới công việc
Mô tả chung	Người dùng có thể tạo mới một công việc cần làm với các thông tin như tiêu đề, mô tả, thời gian hoàn thành, loại công việc, và trạng thái.
Tác nhân	User
Độ ưu tiên	Phải có / Quan trọng
Kích hoạt	Người dùng nhấn nút “+” trên màn hình chính
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống Kết nối Internet ổn định

Hậu điều kiện	Công việc mới được lưu vào hệ thống và hiển thị trong danh sách công việc
Dòng sự kiện:	
Dòng sự kiện chính	1. Người dùng nhấn nút “Tạo công việc mới”. 2. Nhập các thông tin cần thiết (tiêu đề, mô tả, thời gian hoàn thành, loại công việc). 3. Nhấn nút “Lưu”. 4. Hệ thống lưu công việc vào cơ sở dữ liệu và hiển thị trong danh sách công việc.
Dòng sự kiện ngoại lệ	3.1 Người dùng nhập thiếu thông tin bắt buộc, thông báo lỗi và yêu cầu người dùng nhập đủ thông tin.

2.4.1.1 Đặc tả usecase chỉnh sửa công việc

Tóm tắt	
Tên Usecase	Chỉnh sửa công việc
Mô tả chung	Cho phép người dùng chỉnh sửa thông tin của một công việc đã tạo.
Tác nhân	User
Độ ưu tiên	Quan trọng
Kích hoạt	Người dùng nhấn vào một công việc trong danh sách

Tiền điều kiện	Công việc cần chỉnh sửa tồn tại trong danh sách Người dùng đã đăng nhập
Hậu điều kiện	Thông tin công việc được cập nhật và hiển thị trong danh sách
Dòng sự kiện:	
Dòng sự kiện chính	1. Người dùng chọn công việc cần chỉnh sửa. 2. Nhập các thông tin mong muốn. 3. Nhấn nút “Lưu”. 4. Hệ thống cập nhật thông tin công việc và hiển thị thông tin đã chỉnh sửa.
Dòng sự kiện ngoại lệ	3.1 Người dùng nhập thiếu thông tin bắt buộc, thông báo lỗi và yêu cầu người dùng nhập đủ thông tin. 4.1 Không thể cập nhật trạng thái do lỗi kết nối hoặc lỗi hệ thống, thông báo lỗi và yêu cầu người dùng thử lại.

2.4.1.6 Đặc tả usecase xóa công việc

Tóm tắt	
Tên Usecase	Xóa công việc

Mô tả chung	Cho phép người dùng xóa công việc không cần thiết khỏi danh sách công việc.
Tác nhân	User
Độ ưu tiên	Phải có
Kích hoạt	Nhấn nút “Xóa” khi xem chi tiết công việc
Tiền điều kiện	Người dùng đã đăng nhập và có công việc cần xóa
Hậu điều kiện	Công việc được xóa khỏi danh sách.
Dòng sự kiện:	
Dòng sự kiện chính	1. Người dùng chọn công việc cần xóa và nhấn nút “Xóa”. 2. Hệ thống hiển thị thông báo xác nhận. 3. Người dùng nhấn “Đồng ý”. 4. Hệ thống xóa công việc khỏi danh sách và thông báo thành công.
Dòng sự kiện ngoại lệ	2.1 Người dùng nhấn “Hủy” khi hệ thống hỏi xác nhận, hệ thống hủy thao tác xóa. 4.1 Lỗi hệ thống, thông báo lỗi và yêu cầu người dùng thử lại sau

2.4.1.1 Đặc tả usecase chỉnh sửa trạng thái công việc

Tóm tắt	
Tên Usecase	Chỉnh sửa trạng thái công việc
Mô tả chung	Cho phép người dùng thay đổi trạng thái của công việc từ “chờ xử lý” sang “đang làm” hoặc “hoàn tất” hoặc “quá hạn”. Tính năng này giúp người dùng theo dõi tiến độ hoàn thành công việc.
Tác nhân	User
Độ ưu tiên	Quan trọng
Kích hoạt	Người dùng nhấn vào công việc và chọn nút chỉnh sửa trạng thái
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống Công việc tồn tại trong danh sách
Hậu điều kiện	Trạng thái công việc được cập nhật thành công và hiển thị trên danh sách công việc
Dòng sự kiện:	
Dòng sự kiện chính	1. Người dùng nhấn nút “sỏ xuống” trong một công việc trong danh sách. 2. Chọn trạng thái mới cho công việc. 3. Hệ thống tự động cập nhật trạng thái công việc và hiển thị trạng thái mới trên danh sách công việc.

Dòng sự kiện ngoại lệ	3.1 Hệ thống không thể cập nhật trạng thái do lỗi kết nối hoặc lỗi hệ thống, thông báo lỗi và yêu cầu người dùng thử lại.
-----------------------	---

2.4.1.1 Đặc tả usecase xem thống kê công việc

Tóm tắt	
Tên Usecase	Xem thống kê công việc
Mô tả chung	Cho phép người dùng xem thống kê công việc đã hoàn thành và chưa hoàn thành theo ngày, tuần hoặc tháng. Tính năng này giúp người dùng theo dõi hiệu suất công việc.
Tác nhân	User
Độ ưu tiên	Quan trọng
Kích hoạt	Người dùng nhấn vào mục “Thống kê” trên menu
Tiền điều kiện	Người dùng đã đăng nhập Có ít nhất một công việc đã tạo trong hệ thống
Hậu điều kiện	Hiển thị bảng thống kê công việc theo ngày, tuần hoặc tháng
Dòng sự kiện:	

Dòng sự kiện chính	1. Người dùng nhấn vào mục “Thống kê” trên menu. 2. Chọn khoảng thời gian cần thống kê (ngày, tuần, tháng). 3. Hệ thống lấy dữ liệu từ cơ sở dữ liệu và hiển thị số công việc đã hoàn thành và chưa hoàn thành. 4. Người dùng xem thống kê công việc.
Dòng sự kiện ngoại lệ	3.1 Không có công việc nào trong khoảng thời gian đã chọn, thông báo “Không có dữ liệu công việc cho khoảng thời gian này” và yêu cầu người dùng chọn khoảng thời gian khác. 3.2 Lỗi kết nối khiến hệ thống không thể lấy dữ liệu, thông báo lỗi và yêu cầu thử lại sau.

2.4.1.9 Đặc tả usecase xem thông báo công việc

Tóm tắt	
Tên Usecase	Xem thông báo công việc
Mô tả chung	Cho phép người dùng xem danh sách các thông báo nhắc nhở về công việc cần hoàn thành. Tính năng này giúp người dùng không bỏ sót các công việc quan trọng.
Tác nhân	User
Độ ưu tiên	Quan trọng

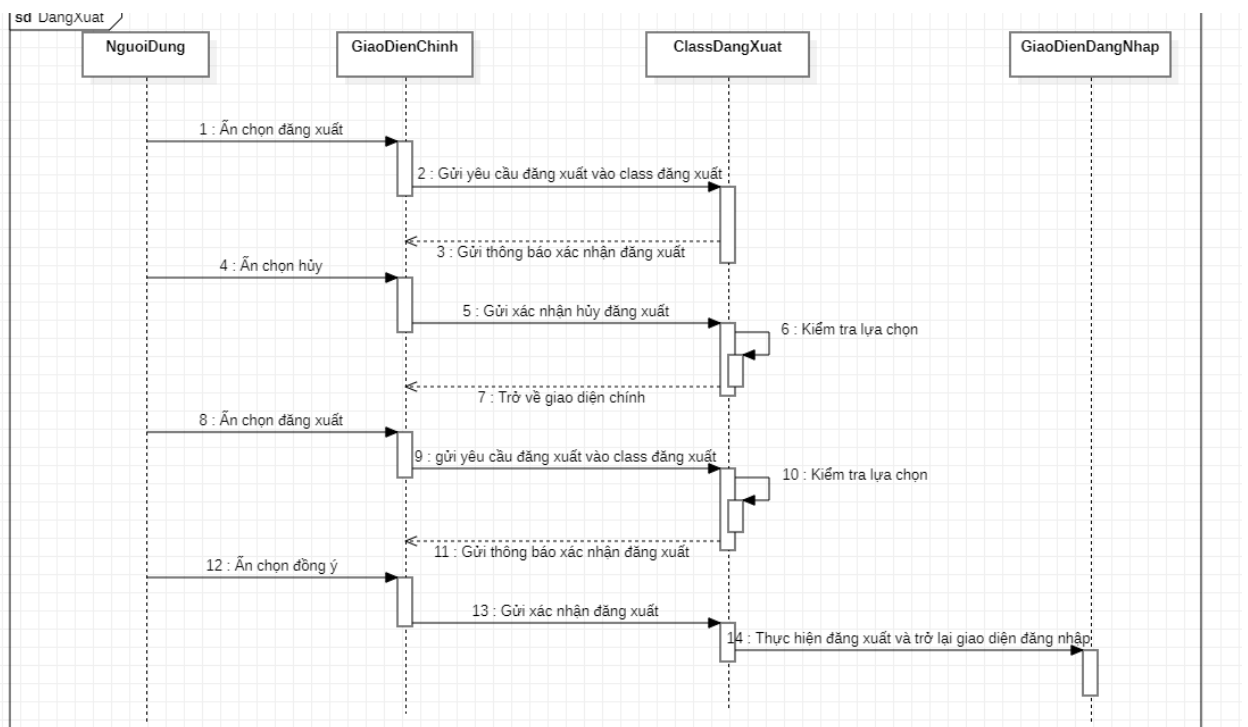
Kích hoạt	Người dùng tạo thành công 1 công việc mới, khi công việc đến 1 giờ trước deadline, và vào 8 giờ sáng mỗi ngày khi có công việc có hạn vào hôm đó
Tiền điều kiện	Người dùng đã đăng nhập Có ít nhất một công việc đã tạo trong hệ thống
Hậu điều kiện	Danh sách thông báo được hiển thị thành công
Dòng sự kiện:	
Dòng sự kiện chính	1. Người dùng tạo thành công 1 công việc mới 2. Hệ thống lấy danh sách thông báo từ cơ sở dữ liệu. 3. Hiển thị danh sách thông báo nhắc nhở công việc theo thứ tự thời gian hoặc khi đến mốc thời gian cố định đã thiết lập. 4. Người dùng có thể nhấn vào từng thông báo để xem chi tiết công việc liên quan.
Dòng sự kiện ngoại lệ	2.1 Lỗi kết nối khiến hệ thống không thể lấy dữ liệu thông báo, thông báo lỗi và yêu cầu người dùng thử lại sau.

2.4.1.10 Đặc tả usecase đăng xuất

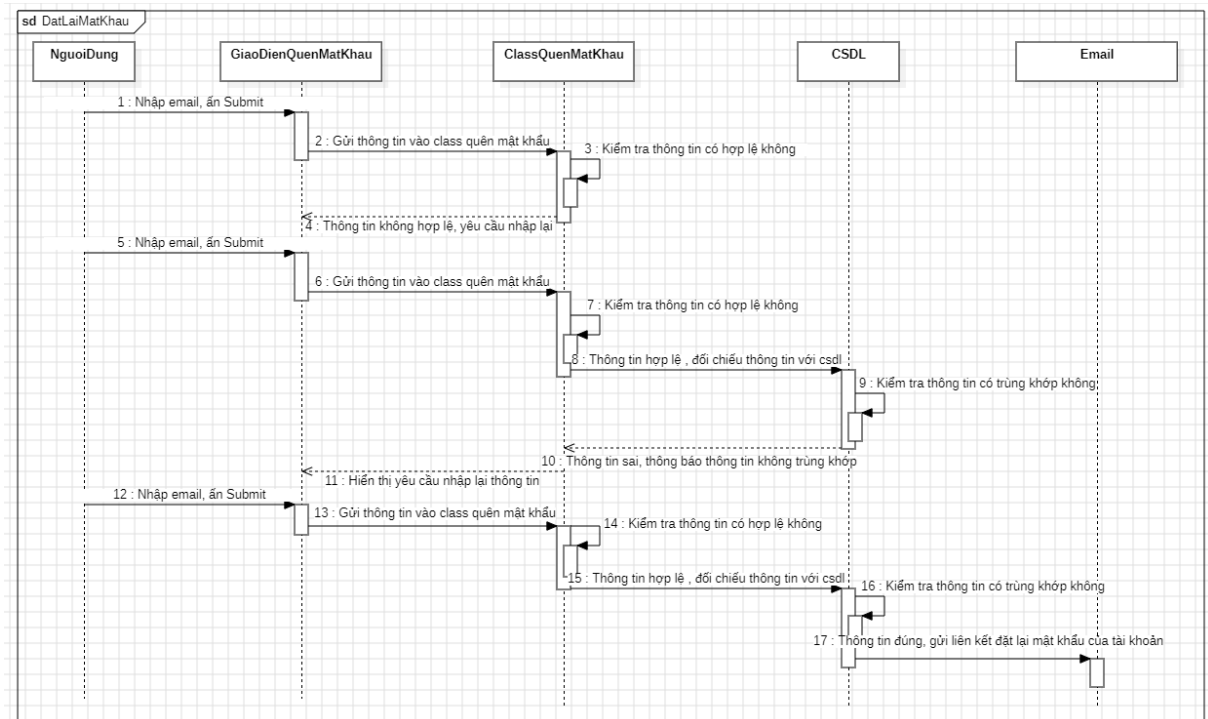
Tóm tắt	
Tên Usecase	Đăng xuất
Mô tả chung	Cho phép người dùng đăng xuất khỏi ứng dụng. Tính năng này giúp đảm bảo tính bảo mật và cho phép người dùng khác đăng nhập trên cùng thiết bị.
Tác nhân	User
Độ ưu tiên	Phải có / Tiên quyết
Kích hoạt	Người dùng nhấn vào nút “Đăng xuất” trên menu cài đặt hoặc thanh điều hướng
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Hậu điều kiện	Phiên đăng nhập của người dùng kết thúc Ứng dụng chuyển về màn hình đăng nhập
Dòng sự kiện:	
Dòng sự kiện chính	1. Người dùng nhấn vào mục “Đăng xuất” trên menu cài đặt hoặc thanh điều hướng. 2. Hệ thống hiển thị thông báo xác nhận đăng xuất. 3. Người dùng chọn “Đồng ý”. 4. Hệ thống kết thúc phiên đăng nhập và chuyển về màn hình đăng nhập.

Dòng sự kiện ngoại lệ	2.1 Người dùng chọn “Hủy” khi hiển thị thông báo xác nhận đăng xuất, hệ thống hủy bỏ yêu cầu đăng xuất và giữ nguyên trạng thái đăng nhập. 3.1 Lỗi hệ thống hoặc kết nối khiến hệ thống không thể đăng xuất đúng cách, thông báo lỗi và yêu cầu thử lại sau.
-----------------------	--

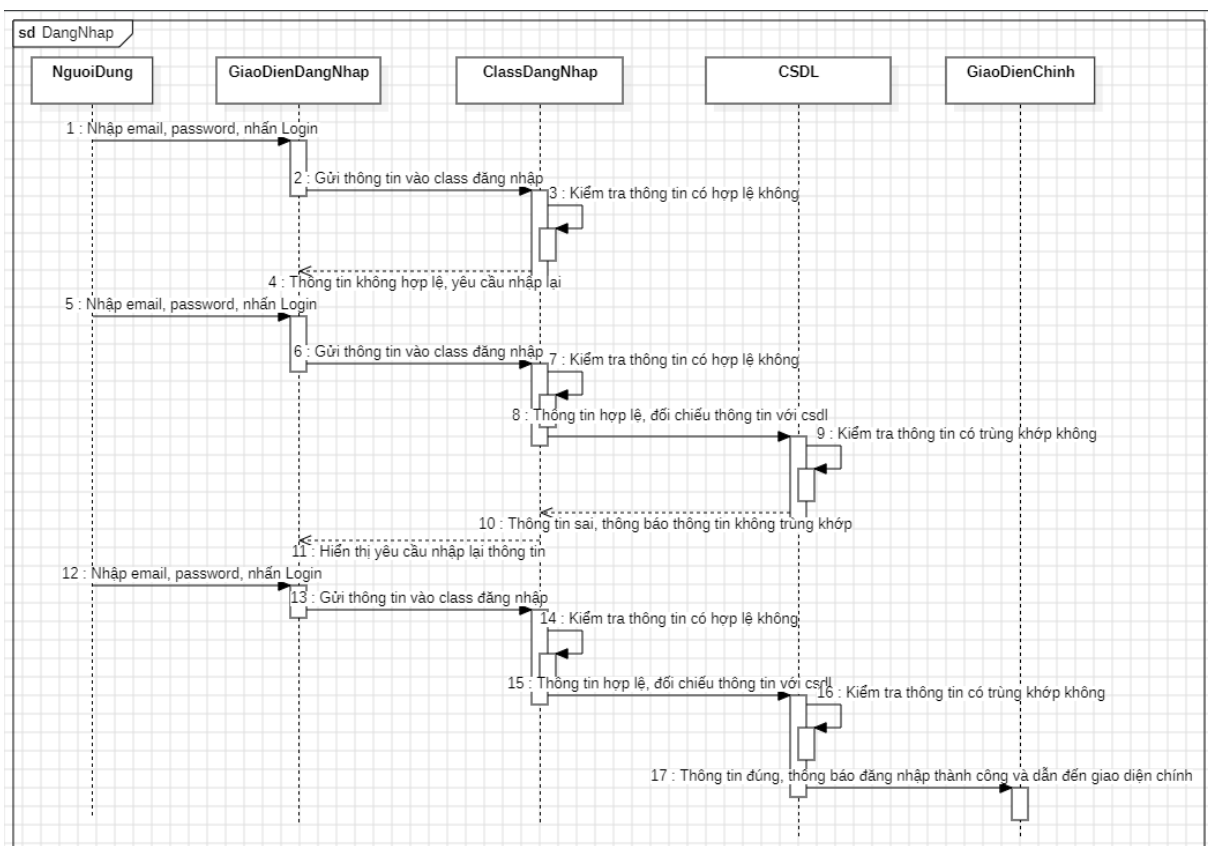
2.4.2 Đặc tả bằng sơ đồ tuần tự



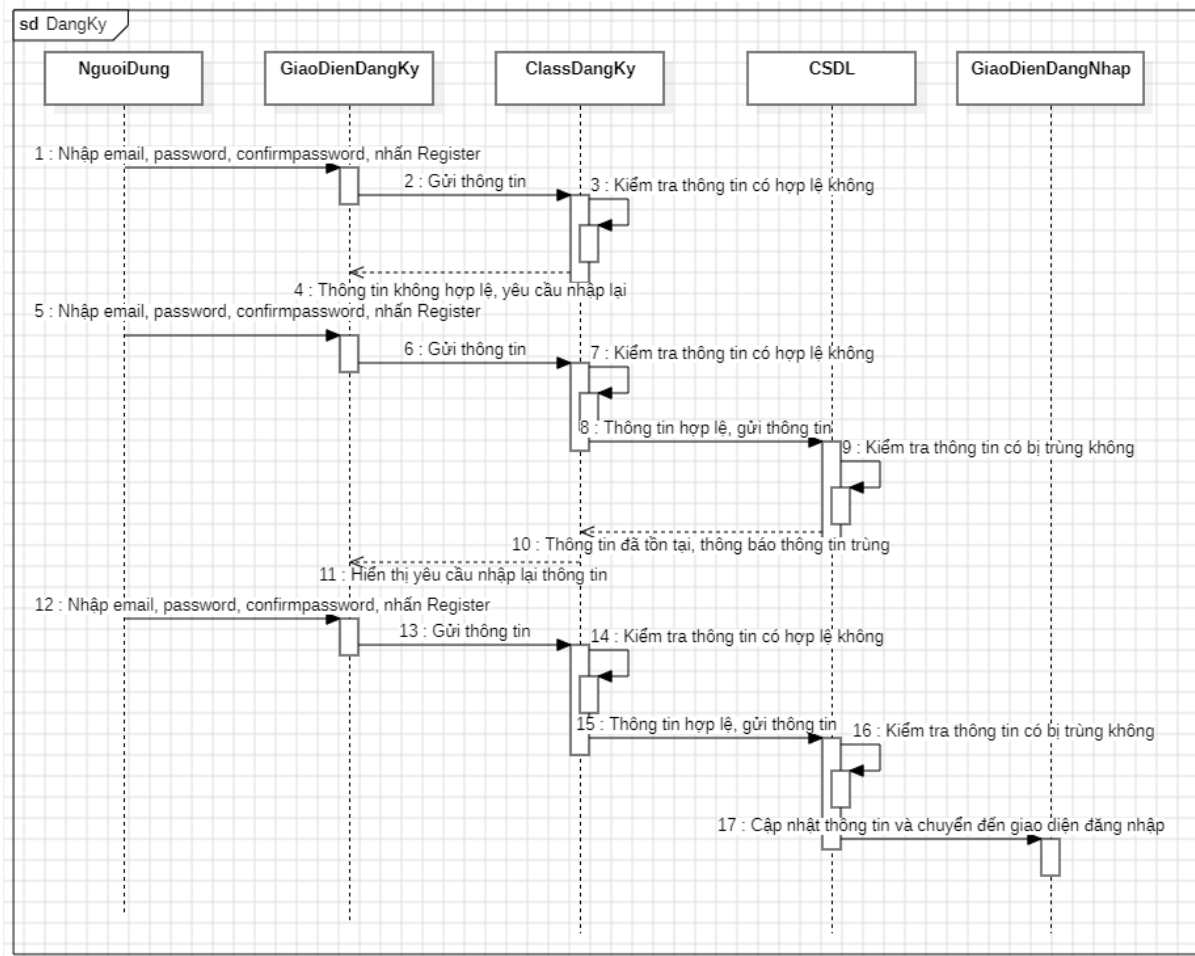
Hình 2.4.2.1: Sơ đồ tuần tự usecase đăng xuất



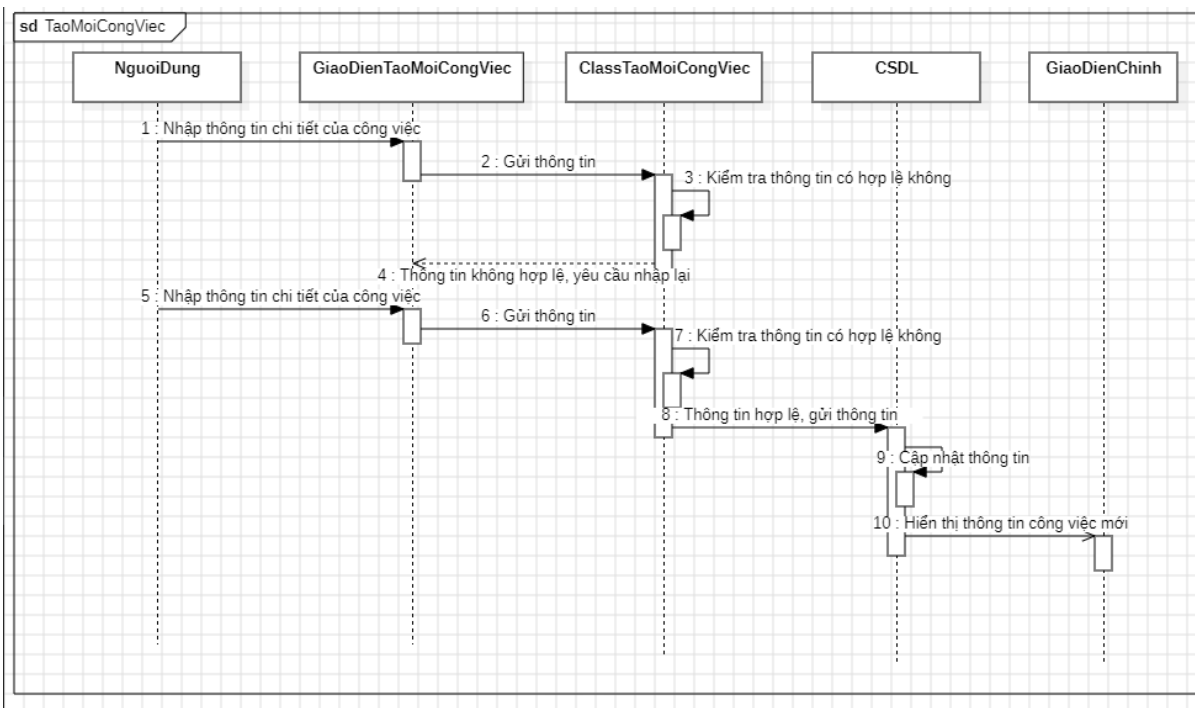
Hình 2.4.2.2: Sơ đồ tuần tự usecase đặt lại mật khẩu



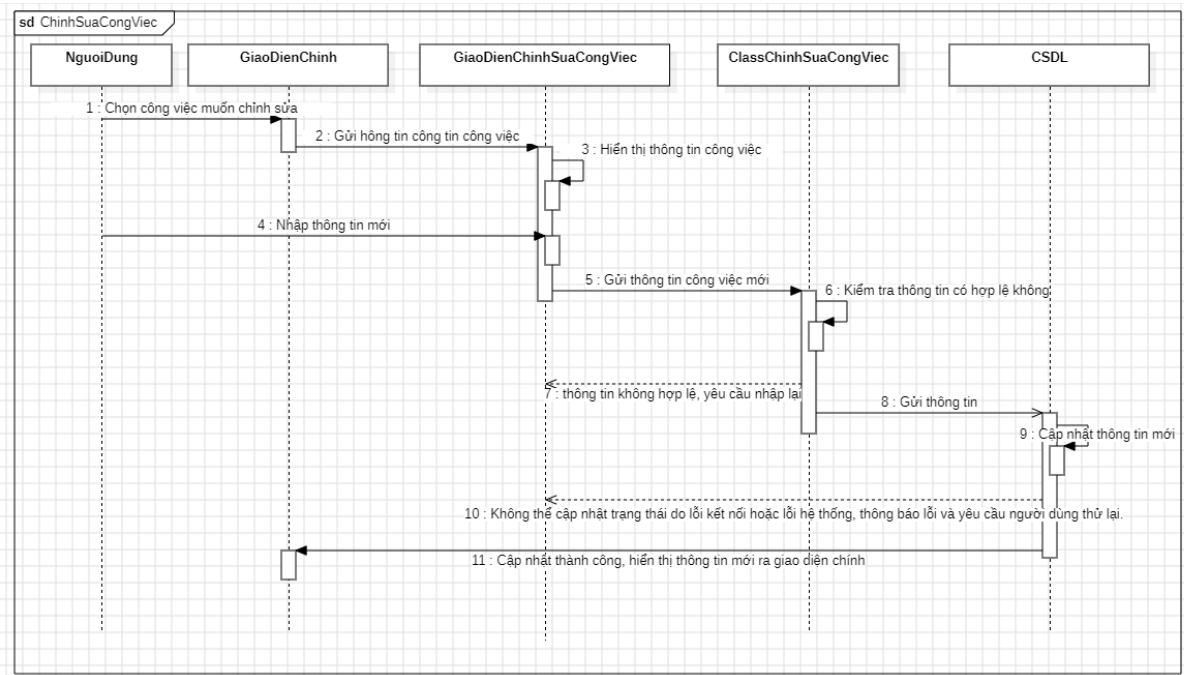
Hình 2.4.2.3: Sơ đồ tuần tự usecase đăng nhập



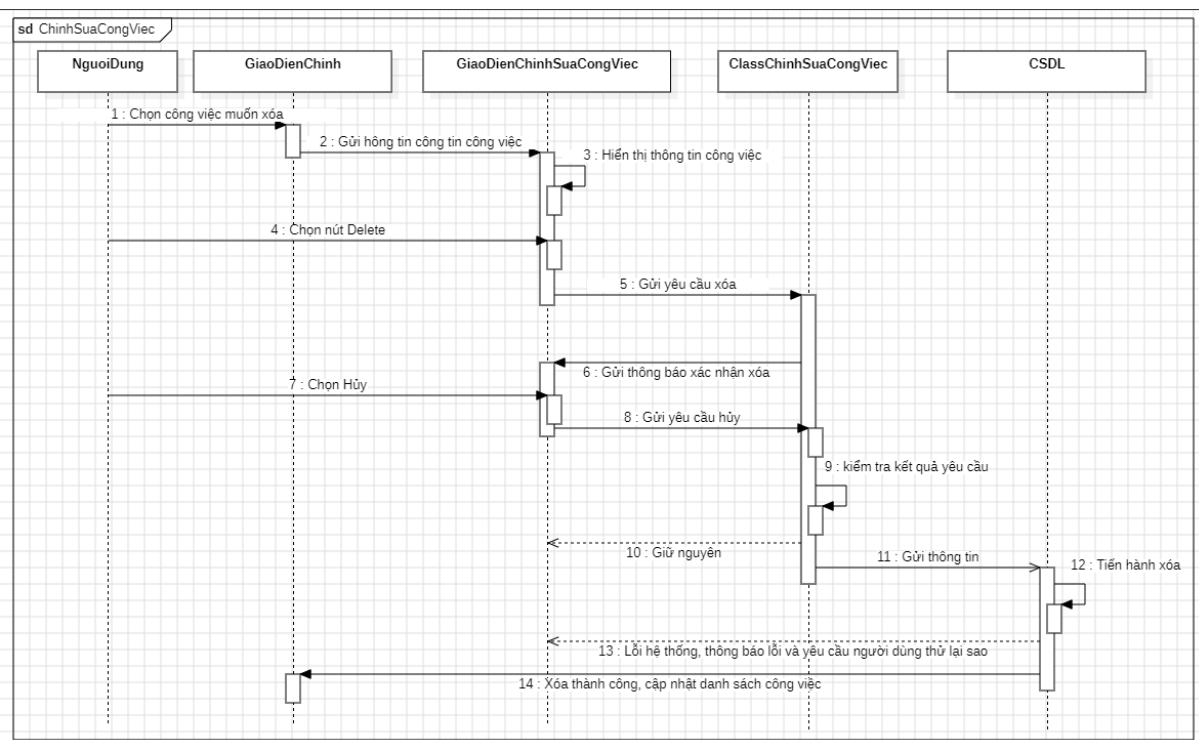
Hình 2.4.2.4: Sơ đồ tuần tự usecase đăng nhập



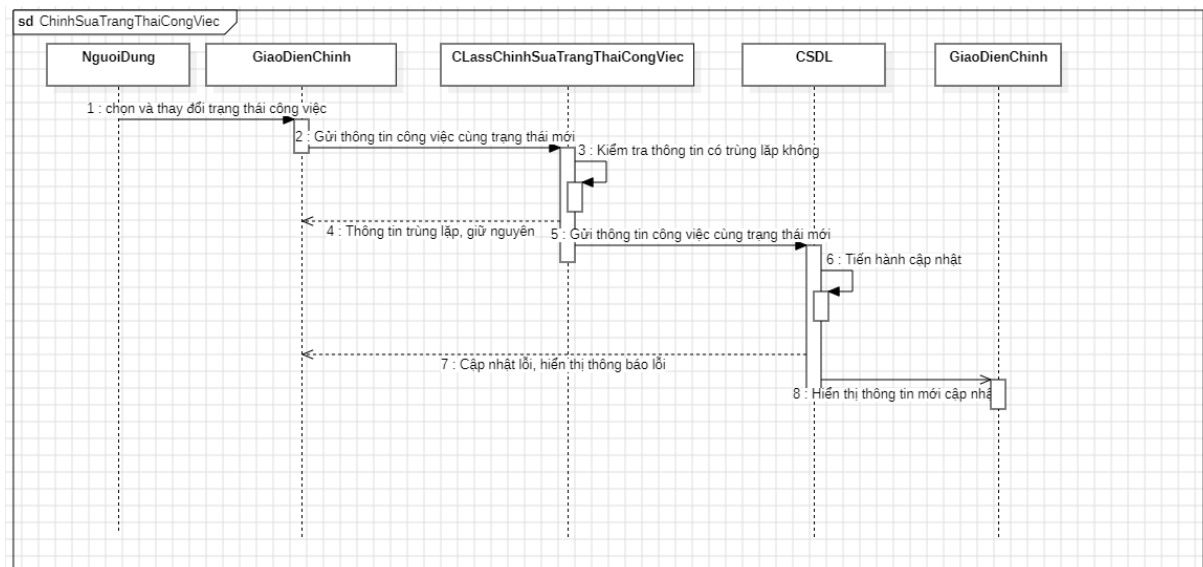
Hình 2.4.2.5: Sơ đồ tuần tự usecase tạo mới công việc



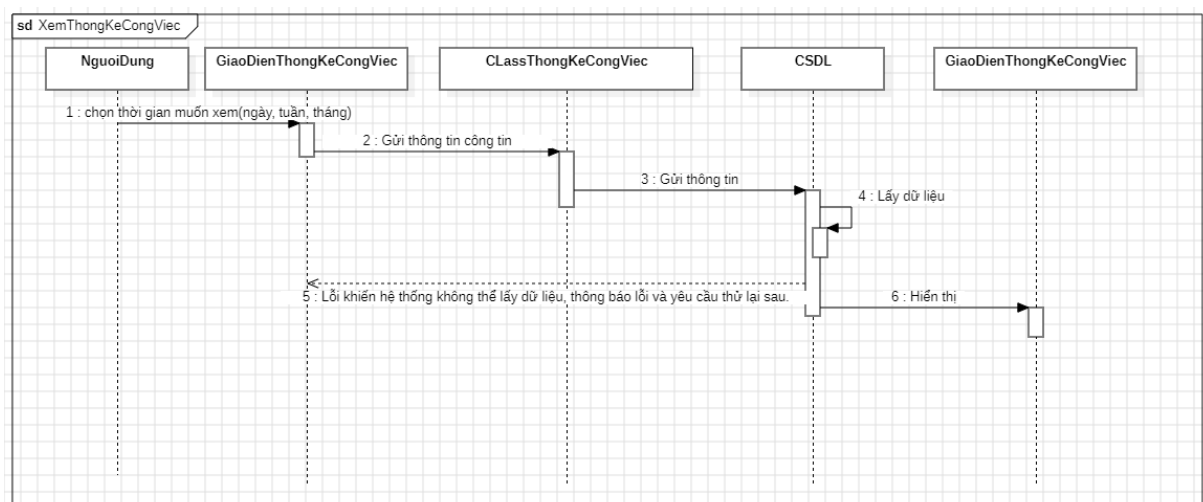
Hình 2.4.2.6: Sơ đồ tuần tự usecase chỉnh sửa công việc



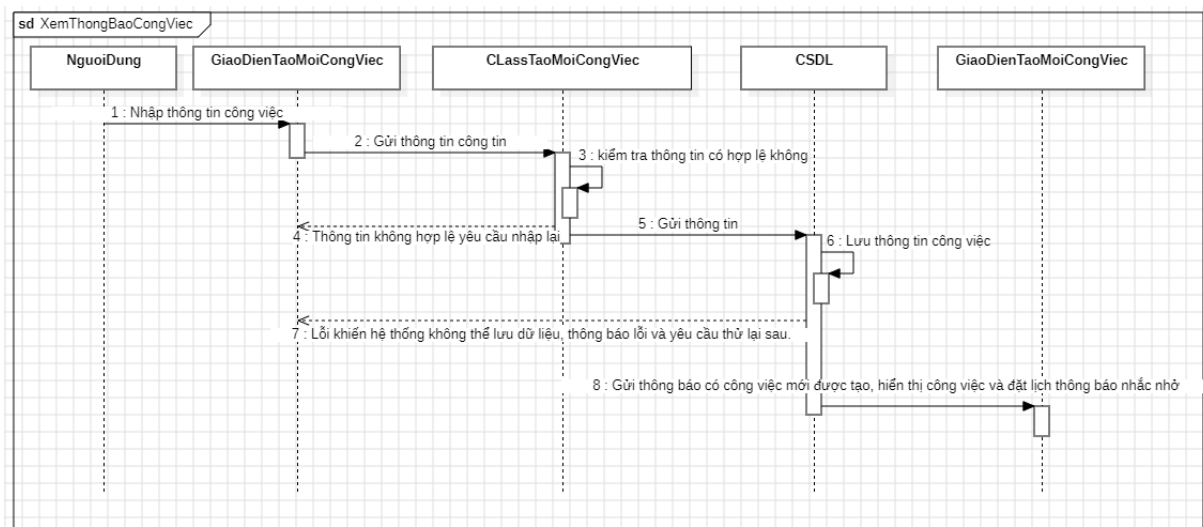
Hình 2.4.2.7: Sơ đồ tuần tự usecase xóa công việc



Hình 2.4.2.8: Sơ đồ tuần tự usecase chỉnh sửa trạng thái công việc

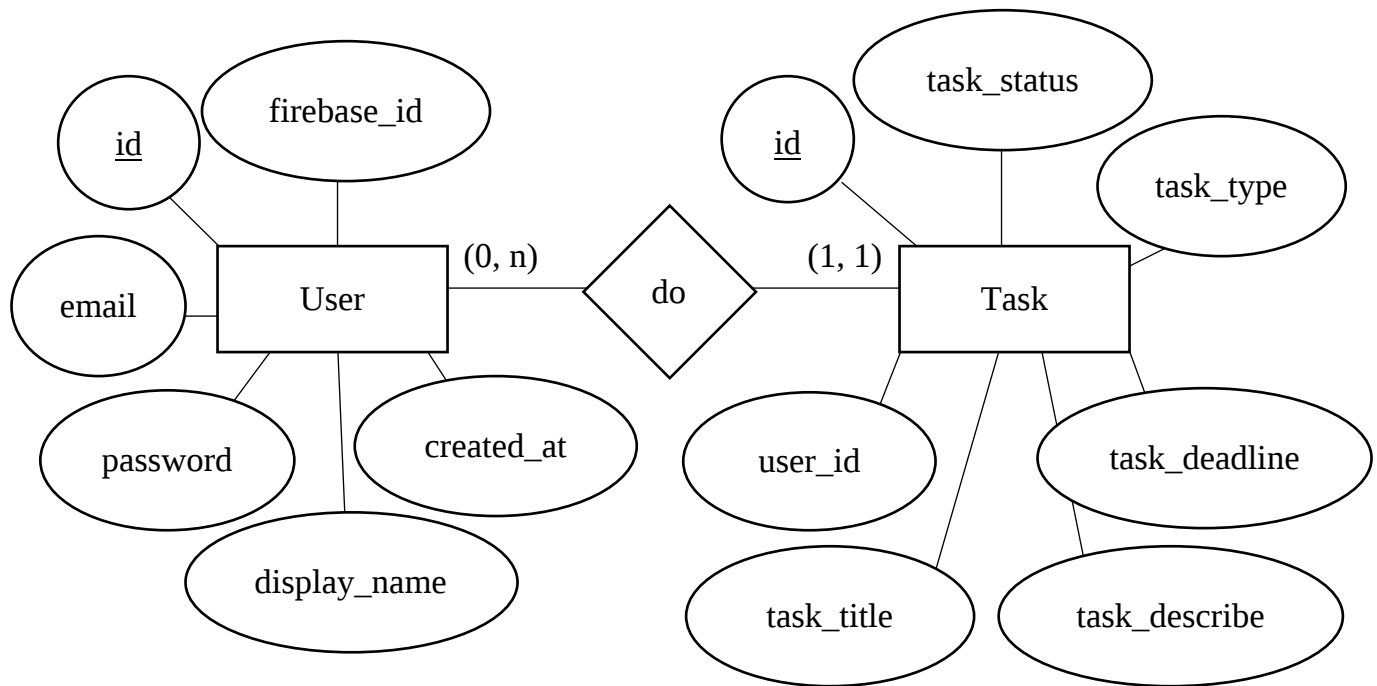


Hình 2.4.2.9: Sơ đồ tuần tự usecase xem thống kê công việc



Hình 2.4.2.10: Sơ đồ tuần tự usecase xem thông báo công việc

2.5 Sơ đồ thực thể liên kết



Hình 2.12: Sơ đồ thực thể liên kết

2.6 Lược đồ CSDL

User (id, firebase_id, email, password, display_name, created_at)

Task (id, user_id, task_title, task_describe, task_deadline, task_type, task_status)

2.7 Các bảng dữ liệu

2.7.1 Bảng User

User			
Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ghi chú
id	INTEGER	PRIMARY KEY AUTOINCREMENT	Mã định danh duy nhất cho mỗi người dùng
firebase_id	TEXT	UNIQUE	Mã định danh duy nhất do Firebase tạo khi người dùng đăng ký thông qua Firebase Authentication.
email	TEXT	NOT NULL UNIQUE	Địa chỉ email duy nhất, không được để trống.
password	TEXT		Mật khẩu của người dùng, có thể lưu dưới dạng hash (băm).
display_name	TEXT		Tên hiển thị của người dùng (tùy chọn).
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Thời điểm tài khoản được tạo, mặc định là thời gian hiện tại khi đăng ký.

2.7.2 Bảng Task

Task			
Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ghi chú
id	INTEGER	PRIMARY KEY AUTOINCREMENT	Mã định danh duy nhất cho mỗi công việc.
user_id	INTEGER	FOREIGN KEY REFERENCES User(id)	Khóa ngoại liên kết đến id trong bảng User, chỉ định người tạo công việc.
task_title	TEXT		Tiêu đề của công việc
task_describe	TEXT		Mô tả chi tiết nội dung công việc.
task_deadline	TEXT		Thời hạn hoàn thành công việc.
task_type	TEXT		Loại công việc (ví dụ: công việc cá nhân, công việc công ty).
task_status	TEXT		Trạng thái công việc

CHƯƠNG 3: XÂY DỰNG HỆ THỐNG

3.1 Kiến trúc hệ thống

Luồng hoạt động của hệ thống: Từ người dùng đến xử lý logic và lưu trữ trong cơ sở dữ liệu.

Cấu hình hệ thống:

Phiên bản Android tối thiểu và khuyến nghị: Android 8 (API 26) trở lên.

3.2 Xây dựng các chức năng chính

Chức năng 1: Tạo công việc

Quy trình từ giao diện người dùng (UI) đến lưu dữ liệu vào SQLite

Giao diện người dùng (UI):

Người dùng điền thông tin công việc vào form:

- Tên công việc (TextField).
- Mô tả công việc (Multiline TextField).
- Thời gian thực hiện (DatePicker/TimePicker).
- Mức độ ưu tiên (Spinner hoặc RadioButton).

Nút "Lưu công việc" để gửi thông tin.

Xử lý sự kiện từ UI:

Khi người dùng nhấn nút "Lưu công việc":

- Dữ liệu từ các trường nhập liệu được kiểm tra (validation) để đảm bảo không bỏ sót trường quan trọng.
- Nếu hợp lệ, dữ liệu được đóng gói thành một đối tượng (Task).

Kết nối với SQLite:

Sử dụng SQLiteOpenHelper hoặc Room Database để lưu dữ liệu:

- Tạo một bảng Tasks trong cơ sở dữ liệu SQLite với các cột: id, name, description, time, priority, status.

- Thực hiện câu lệnh SQL để chèn dữ liệu
- Xử lý ngoại lệ nếu có lỗi khi lưu dữ liệu.

Phản hồi cho người dùng:

Hiện thị thông báo (Toast hoặc Snackbar) báo thành công hoặc thất bại.

Reset các trường nhập liệu sau khi lưu thành công.

Chức năng 2: Hiện thị danh sách công việc

Sử dụng RecyclerView để hiện thị danh sách

Giao diện người dùng:

RecyclerView được đặt trong màn hình chính để hiện thị danh sách công việc.

Item layout của RecyclerView bao gồm:

- Tên công việc.
- Thời gian thực hiện.
- Trạng thái (hoàn thành/chưa hoàn thành).

Kết nối dữ liệu từ SQLite:

Sử dụng Cursor để lấy dữ liệu từ bảng Tasks.

Viết query để lấy danh sách công việc

Dữ liệu được chuyển đổi thành danh sách đối tượng (List<Task>).

Cập nhật giao diện:

Adapter được tạo riêng để kết nối dữ liệu với RecyclerView

Mỗi item trong RecyclerView sẽ được cập nhật với thông tin từ danh sách công việc.

Khi có thay đổi dữ liệu (thêm/sửa/xóa), RecyclerView tự động làm mới giao diện

Chức năng 3: Nhắc nhở công việc

Thiết lập nhắc nhở với AlarmManager

Tạo nhắc nhở:

Khi người dùng tạo công việc và thiết lập thời gian nhắc nhở, sử dụng AlarmManager để đặt lịch:

triggerTime là thời điểm nhắc nhở được thiết lập.

Xử lý sự kiện nhắc nhở (BroadcastReceiver):

Khi đến thời gian nhắc nhở, BroadcastReceiver sẽ xử lý sự kiện:

Thông báo đẩy (Notification):

Sử dụng NotificationManager để hiển thị thông báo:

Chức năng 4: Thống kê công việc

Lấy dữ liệu từ SQLite và hiển thị thống kê

Thu thập dữ liệu:

Viết các query để lấy số liệu thống kê từ bảng Tasks:

- Số công việc hoàn thành:

```
SELECT COUNT(*) FROM Tasks WHERE status = 'Completed';
```

- Số công việc chưa hoàn thành:

```
SELECT COUNT(*) FROM Tasks WHERE status = 'Pending';
```

- Thống kê công việc theo mức độ ưu tiên:

```
SELECT priority, COUNT(*) FROM Tasks GROUP BY priority;
```

Xử lý và hiển thị:

Dữ liệu thống kê được chuyển thành biểu đồ (Pie Chart hoặc Bar Chart) hoặc bảng tóm tắt.

Sử dụng thư viện như MPAndroidChart để vẽ biểu đồ:

```
PieChart chart = findViewById(R.id.pieChart);
```

```
PieData data = new PieData(dataset);
```

```
chart.setData(data);
```

chart.invalidate();

Giao diện người dùng:

Hiển thị số liệu tổng quan:

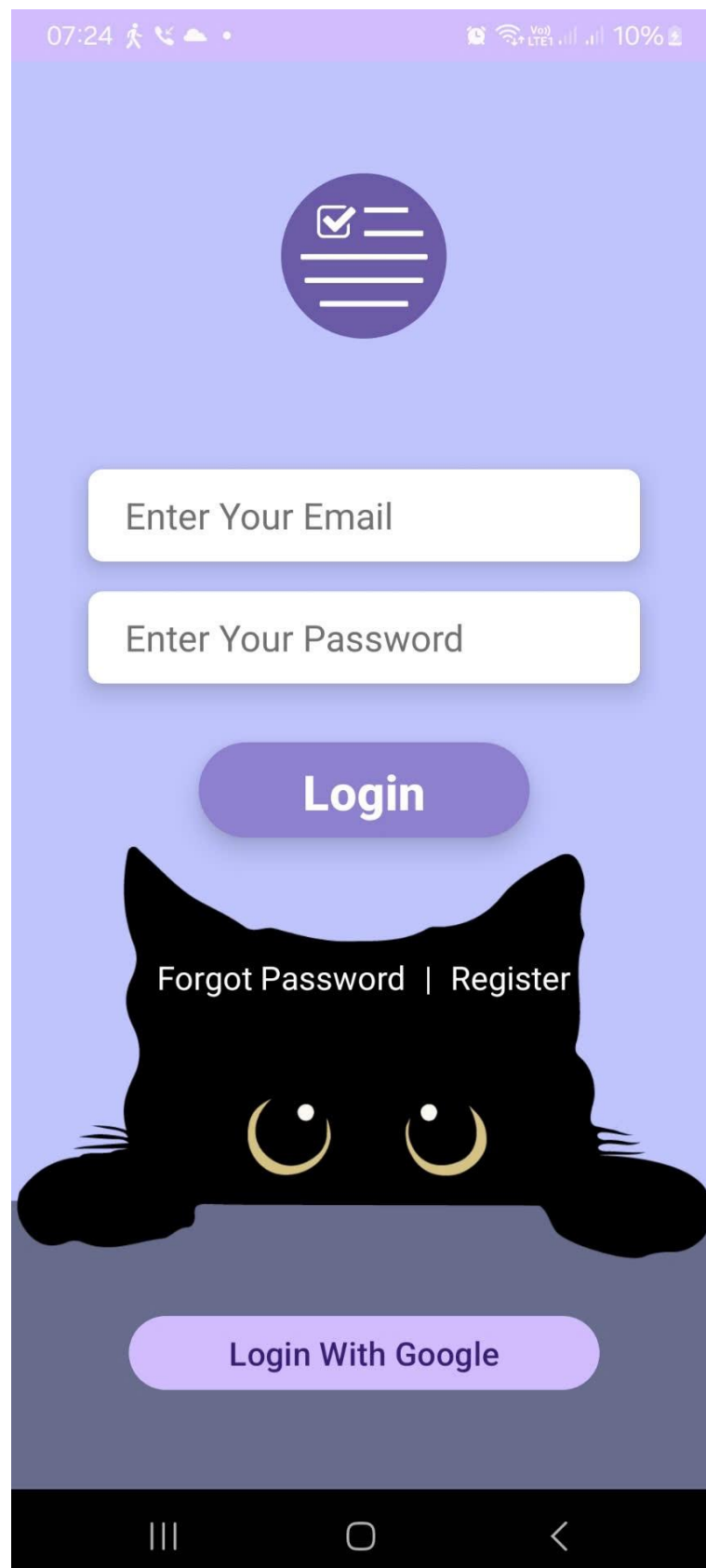
- Tổng số công việc.
- Phần trăm công việc hoàn thành.

Biểu đồ trực quan để người dùng dễ theo dõi tiến độ.

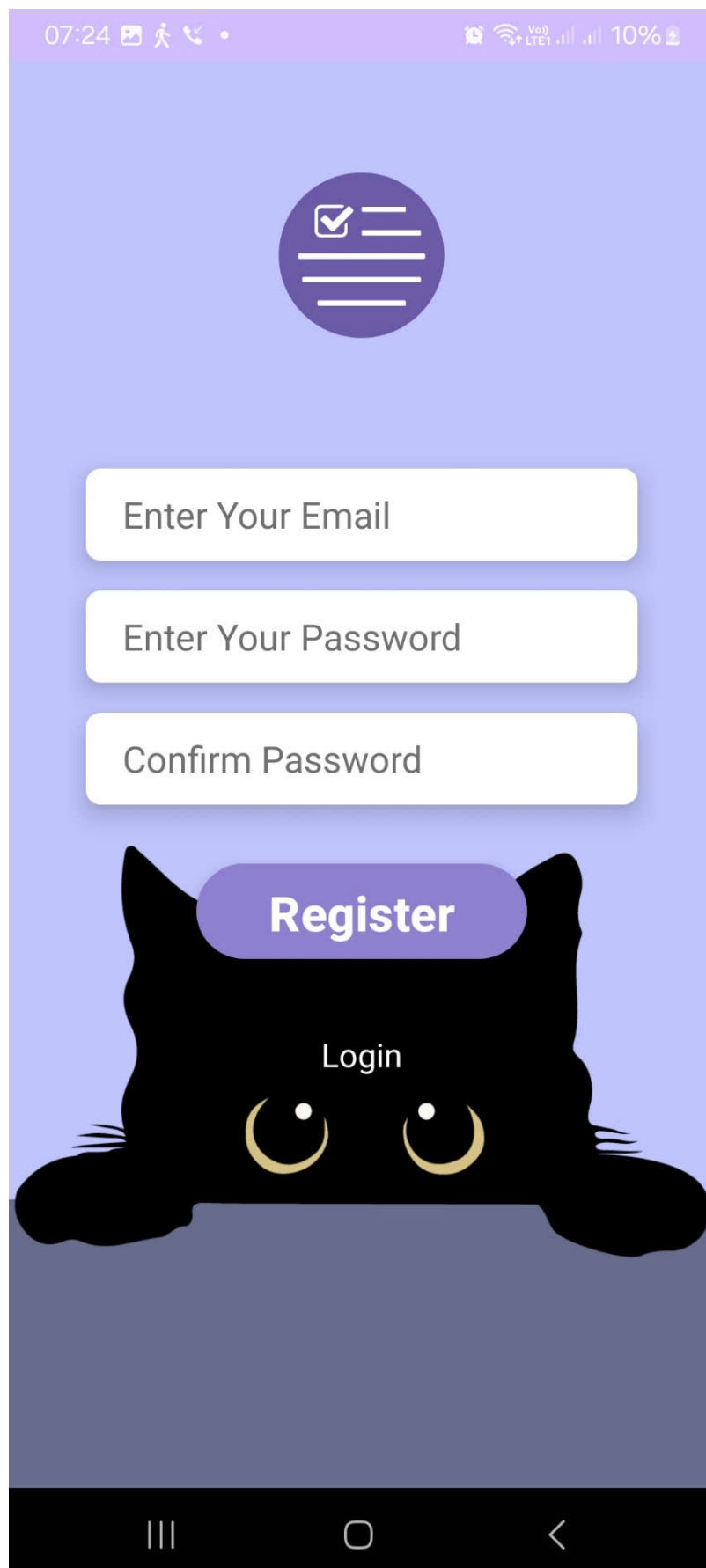
3.3 Thiết kế các màn hình giao diện



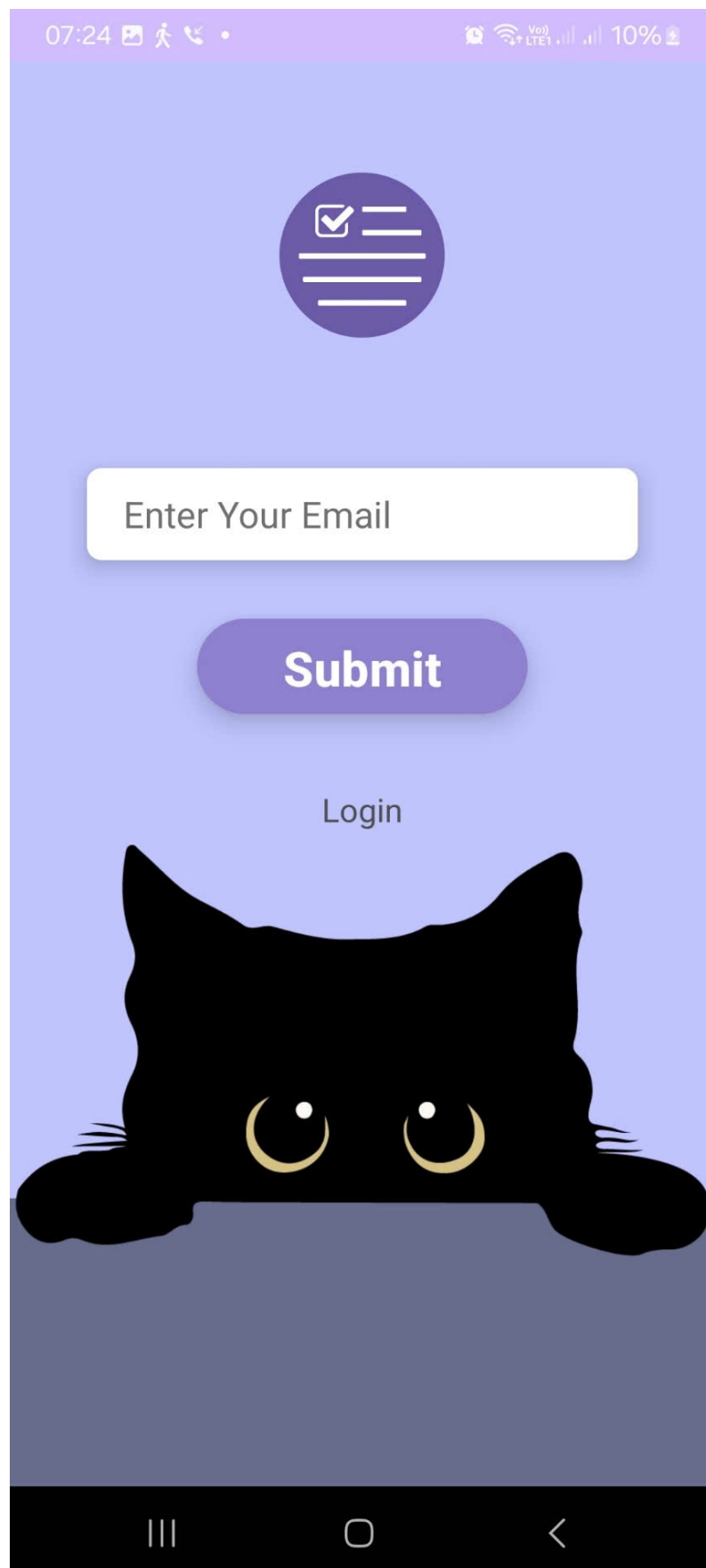
Hình 3.3.1: Logo app



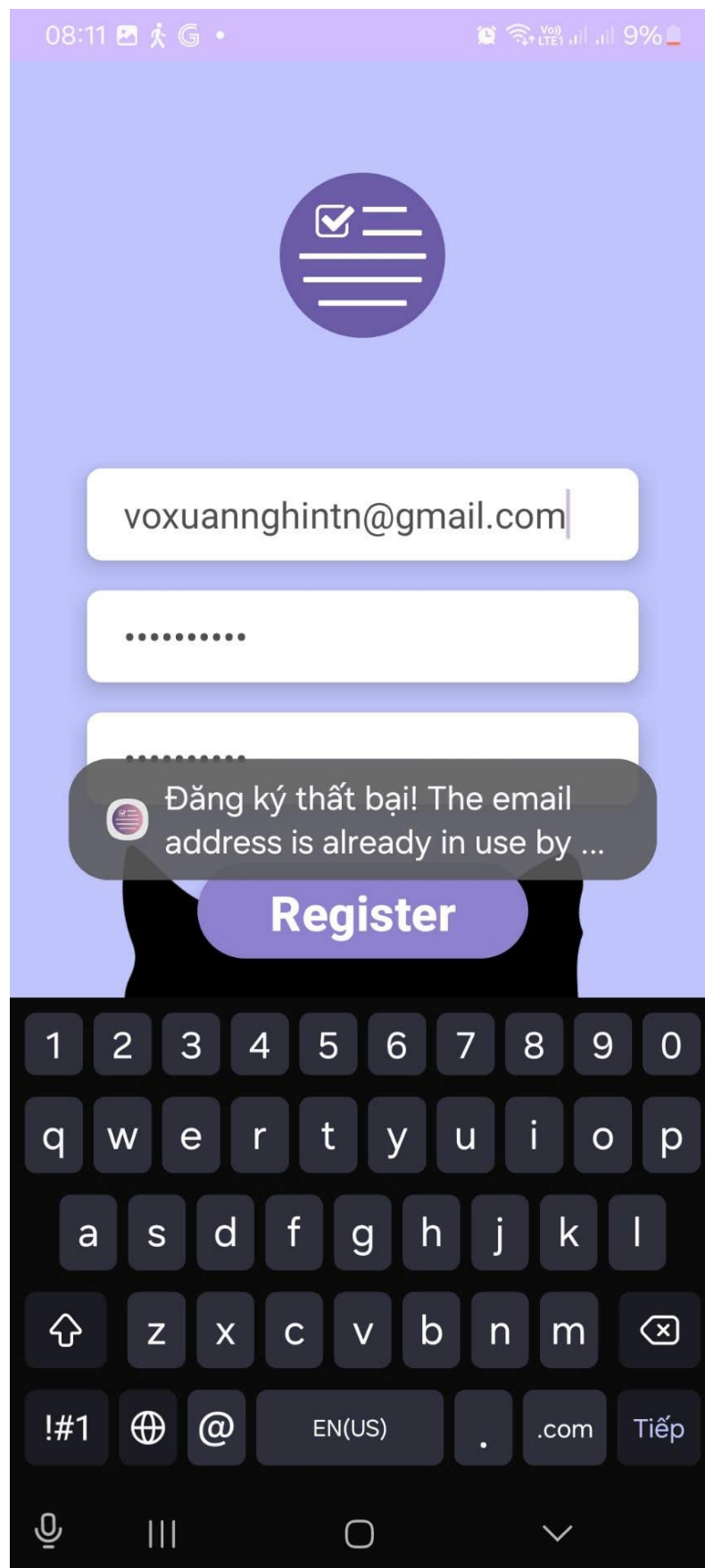
Hình 3.3.2: Màn hình đăng nhập



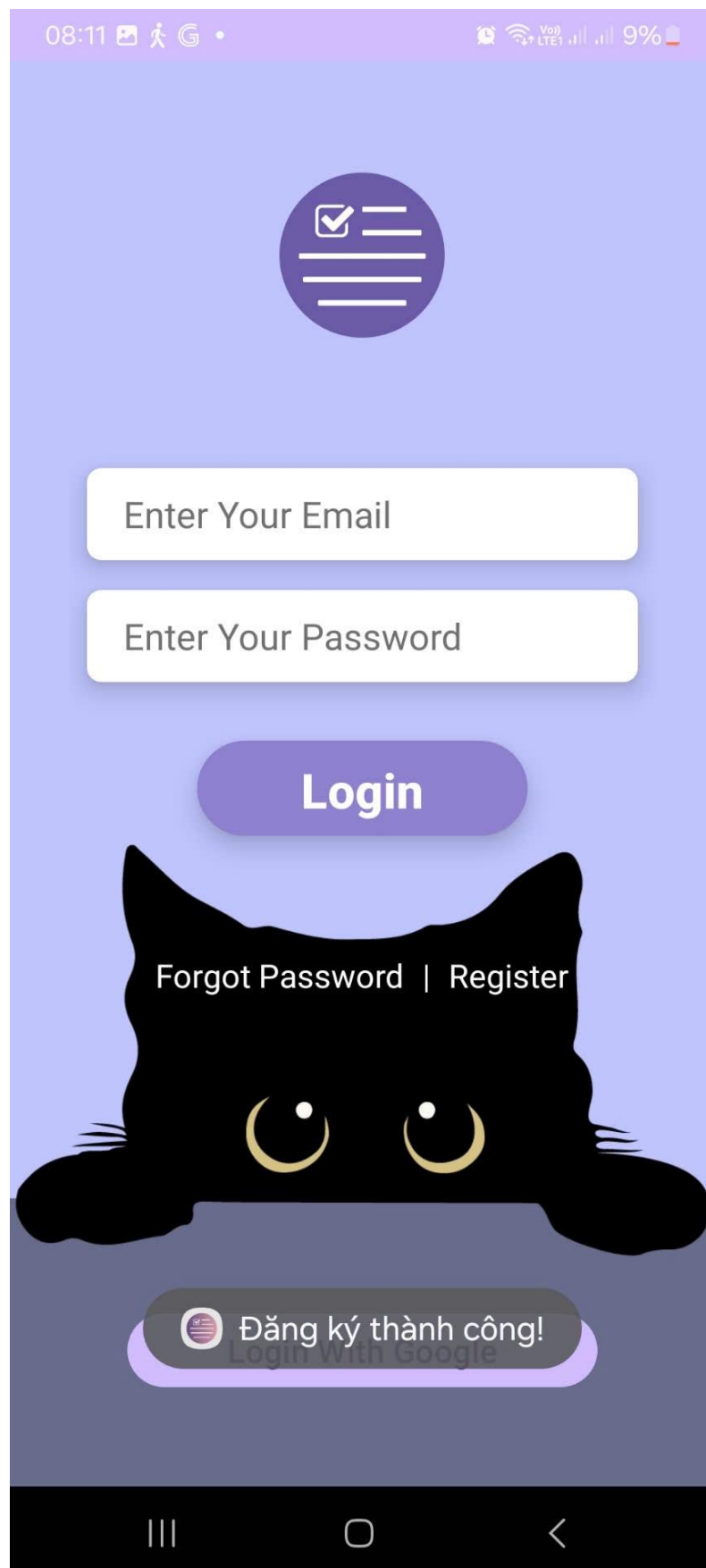
Hình 3.3.3: Màn hình đăng ký



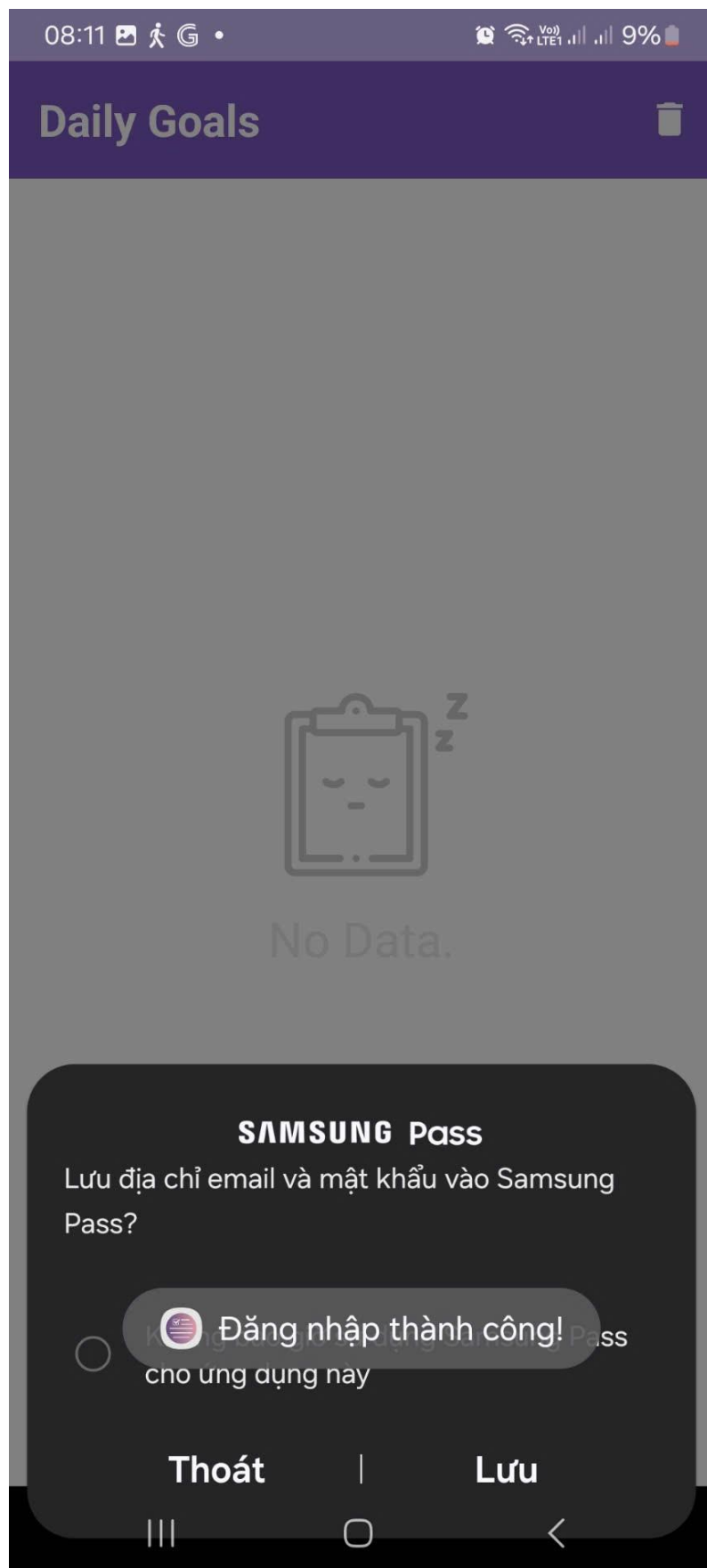
Hình 3.3.4: Màn hình nhập email để đặt lại mật khẩu



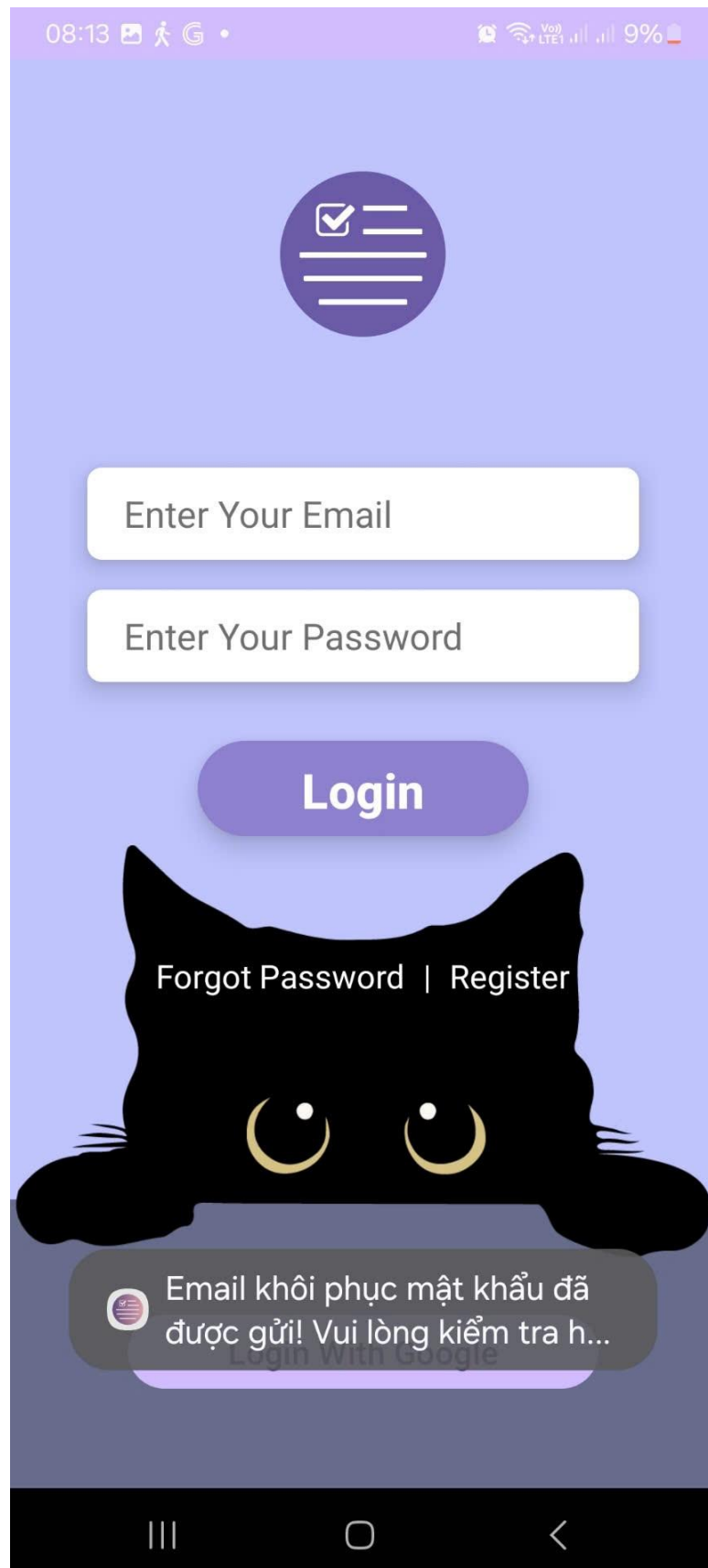
Hình 3.3.5: Màn hình khi đăng ký thất bại



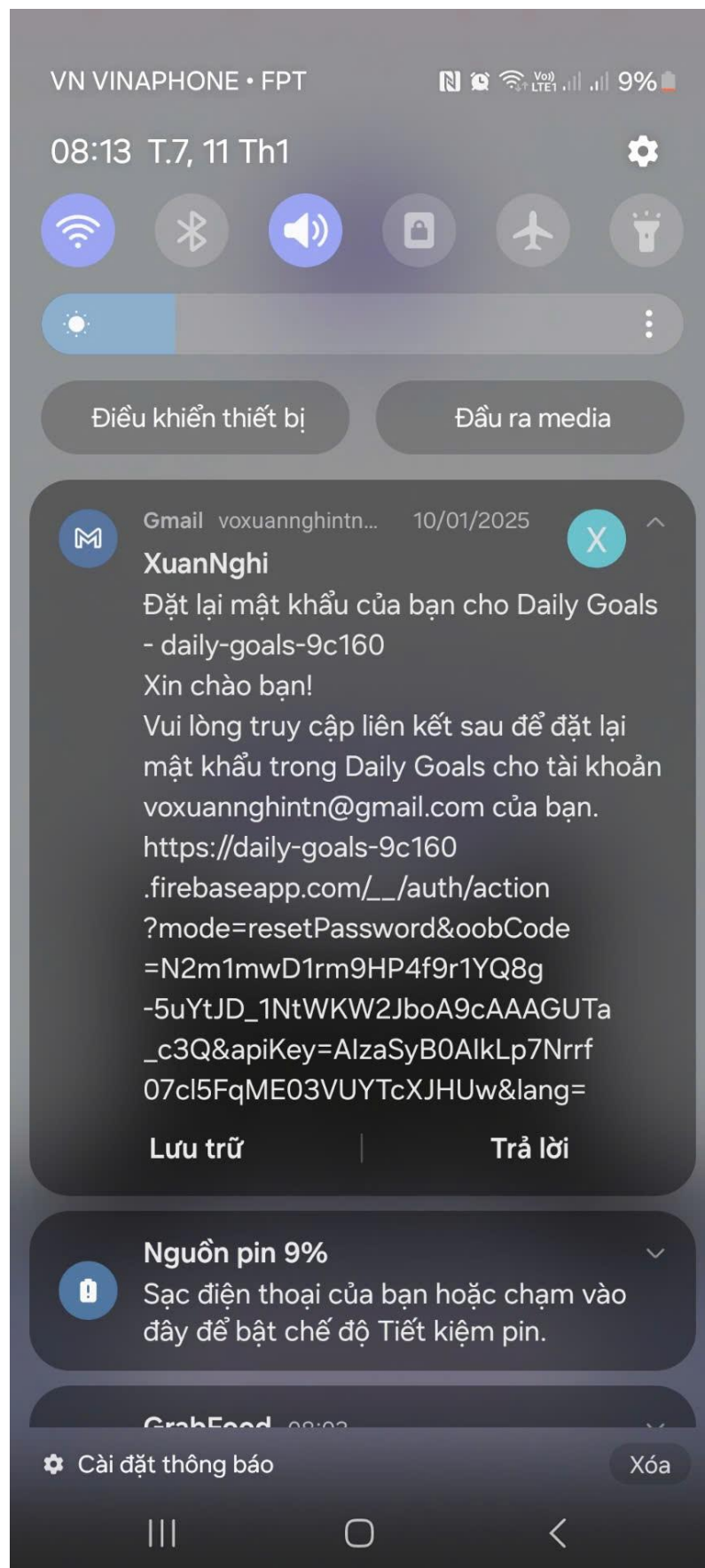
Hình 3.3.6: Màn hình khi đăng ký thành công



Hình 3.3.7: Màn hình khi đăng nhập thành công



Hình 3.3.8: Màn hình khi nhập email khôi phục mật khẩu thành công



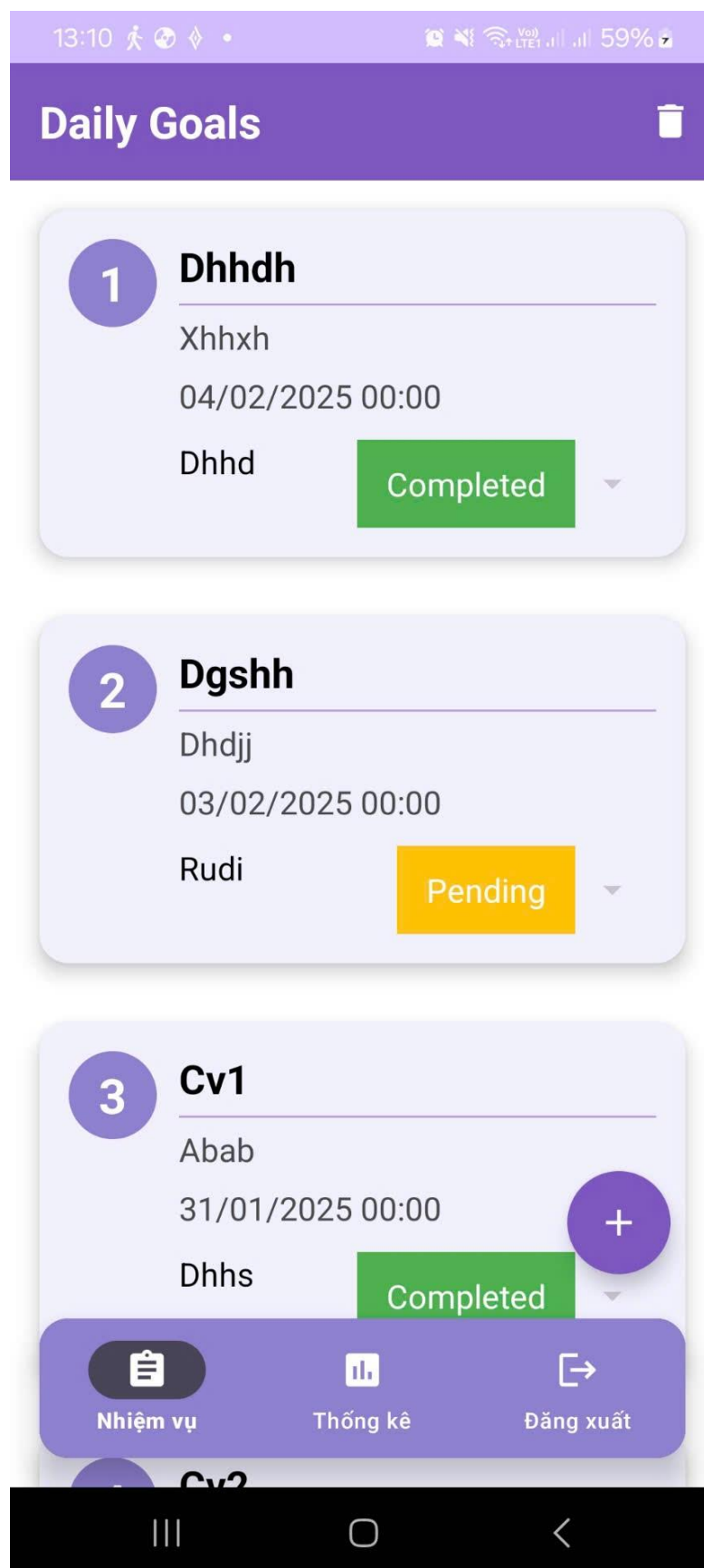
Hình 3.3.9: Màn hình thông báo khi nhập email khôi phục mật khẩu thành công



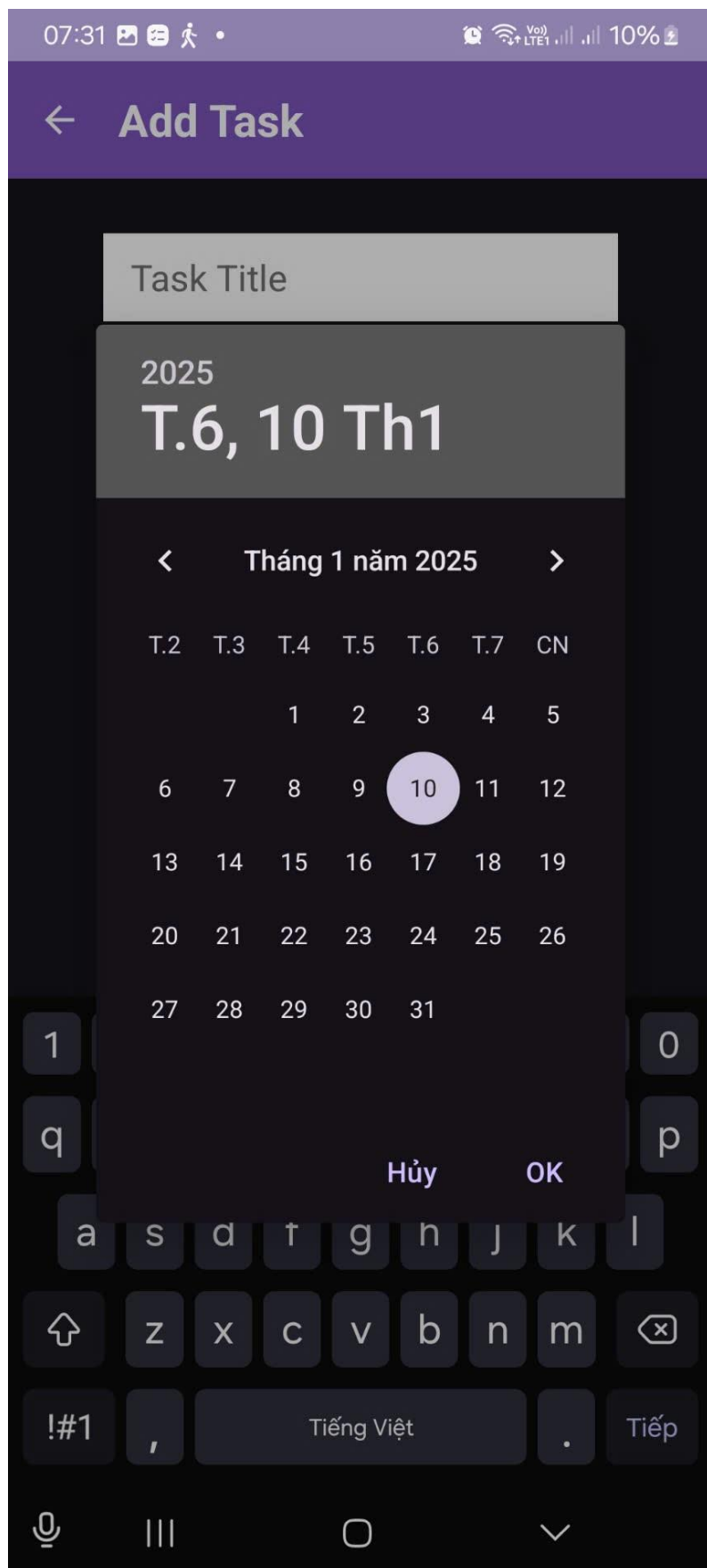
Hình 3.3.10: Màn hình thông báo trong gmail khi nhập email khôi phục mật khẩu thành công



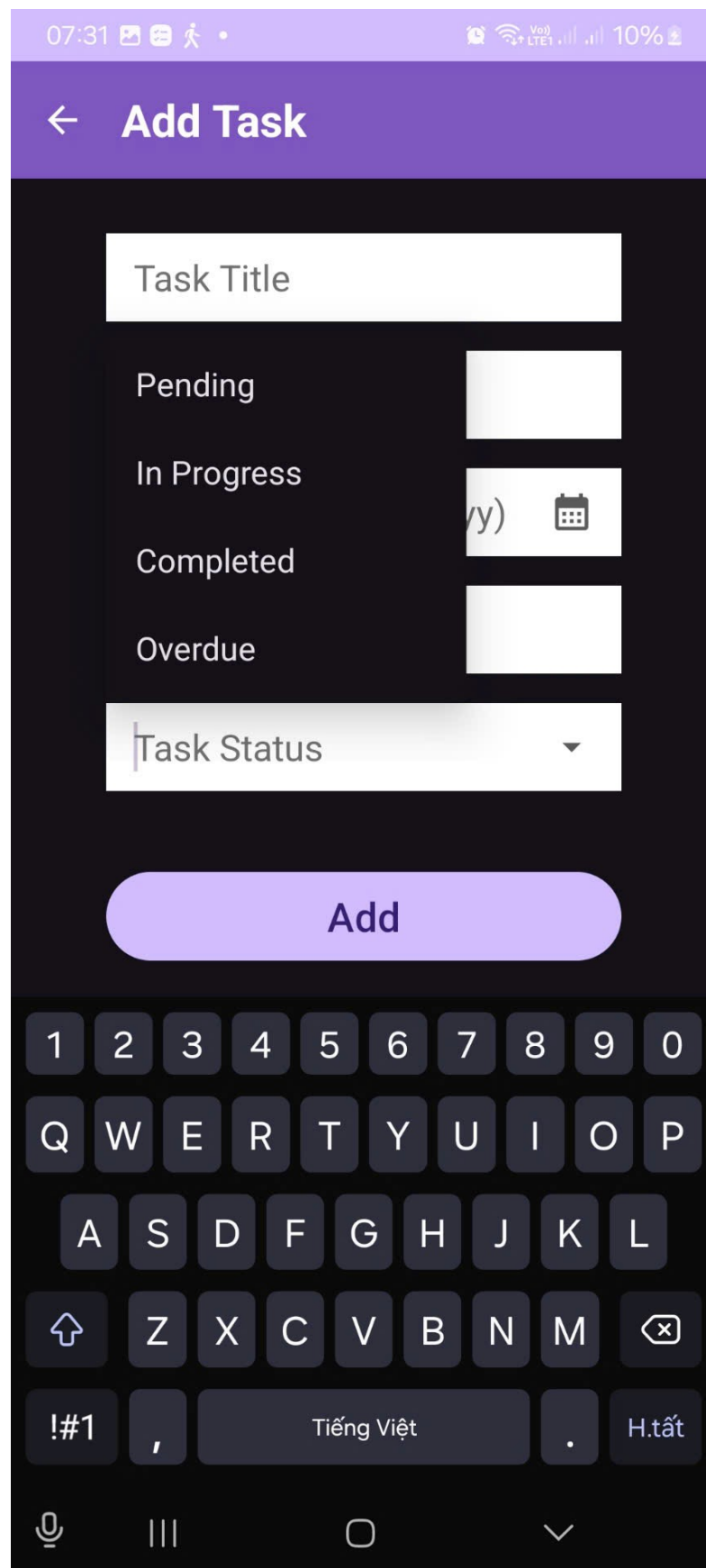
Hình 3.3.11: Màn hình khi đã đăng nhập thành công và chưa tạo công việc

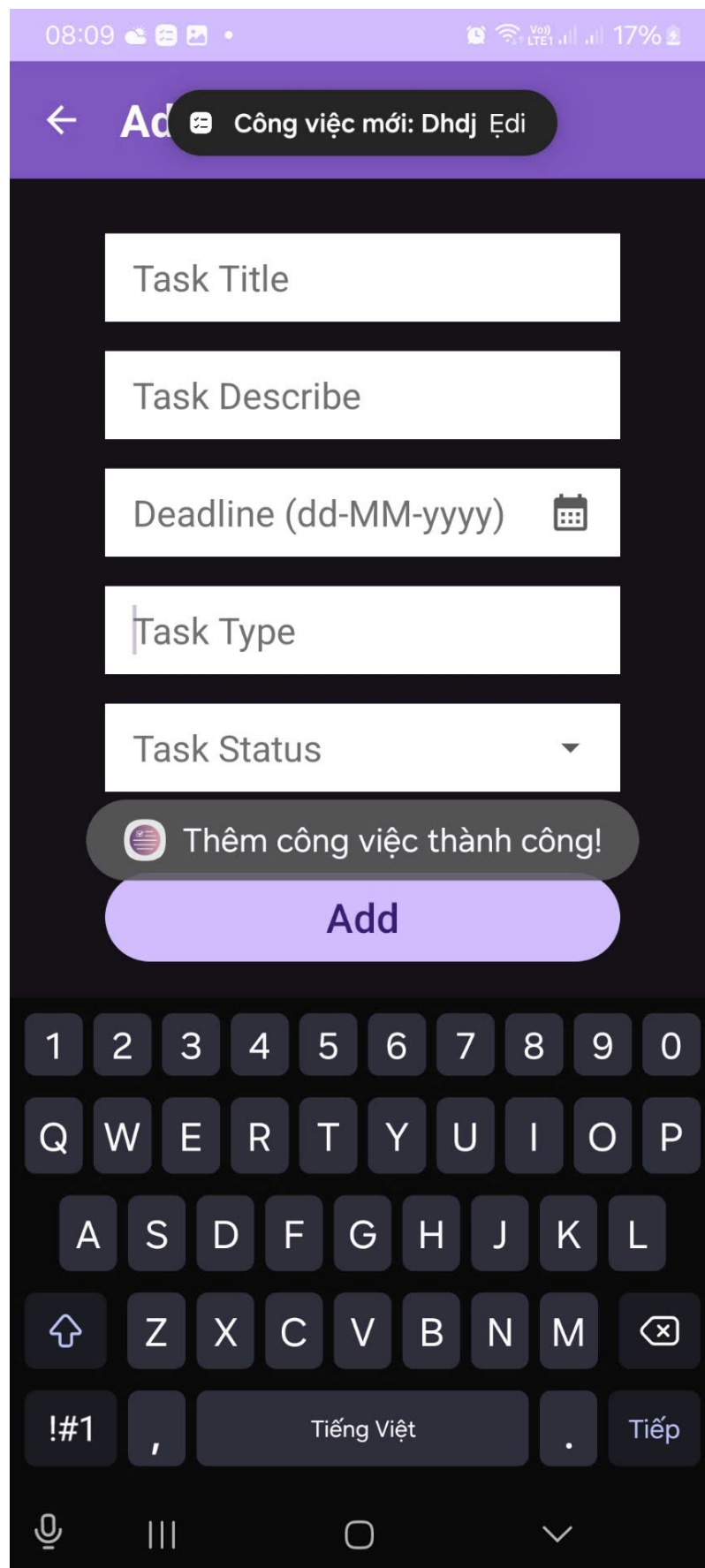


Hình 3.3.12: Màn hình khi đã đăng nhập thành công và đã tạo công việc

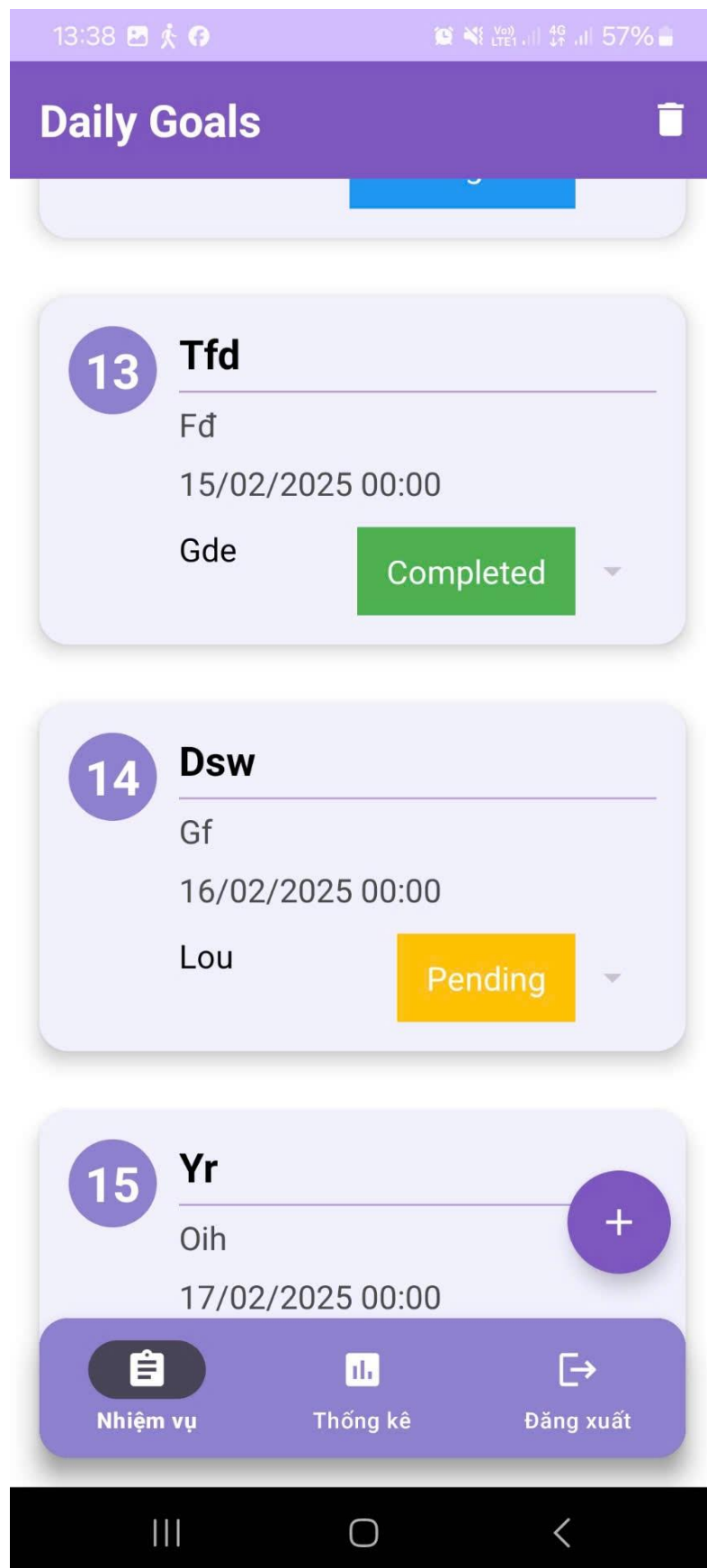


Hình 3.3.14: Màn hình tạo mới công việc – thêm deadline

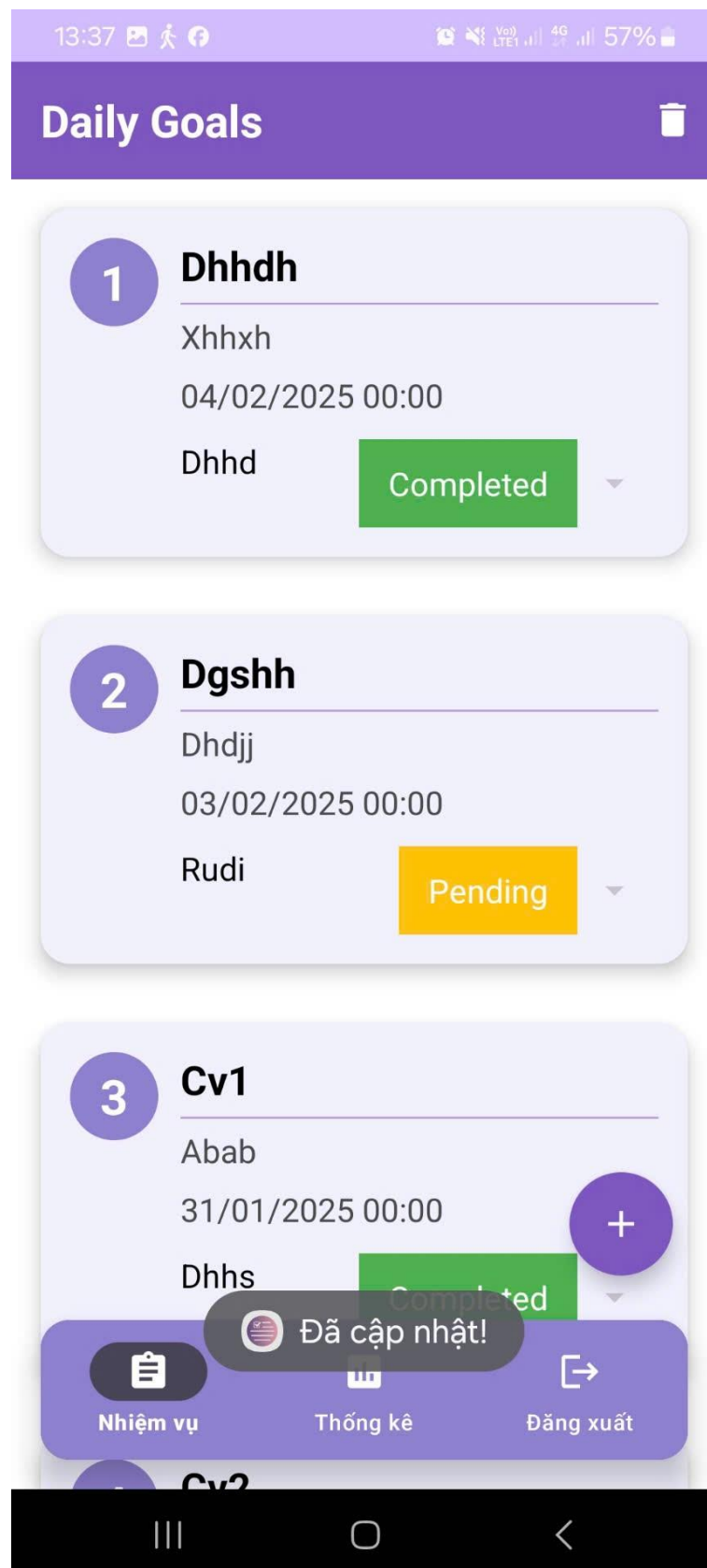




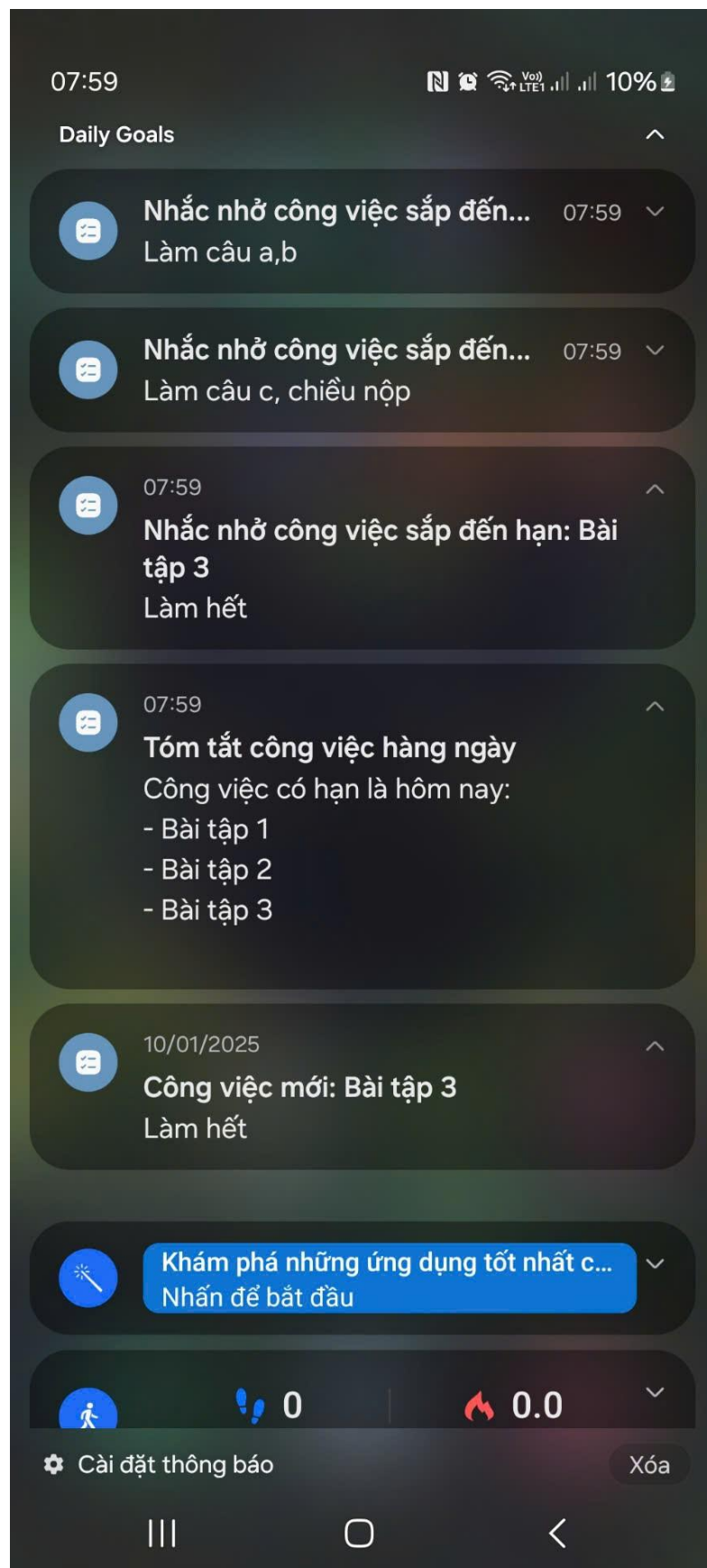
Hình 3.3.17: Màn hình tạo mới công việc thành công



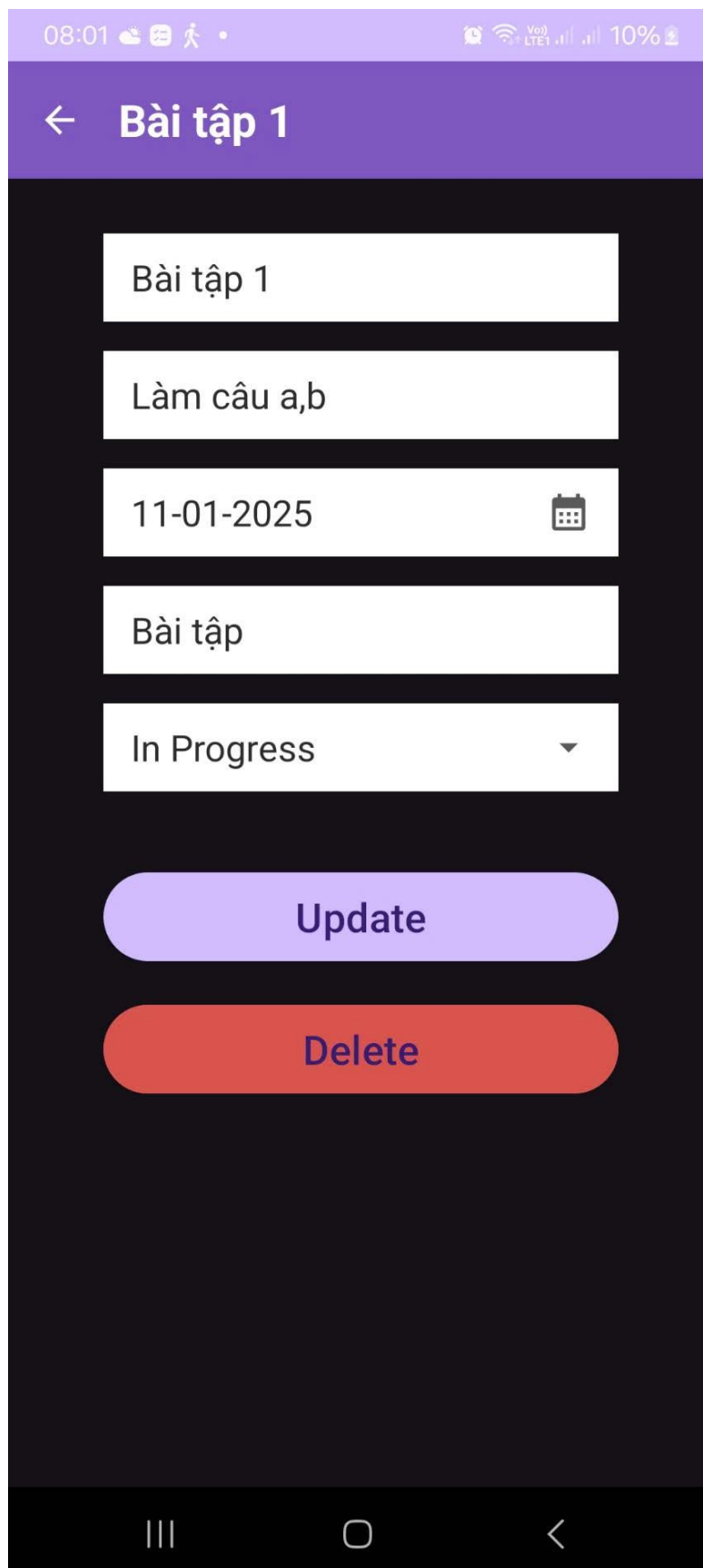
Hình 3.3.18: Màn hình công việc vừa tạo



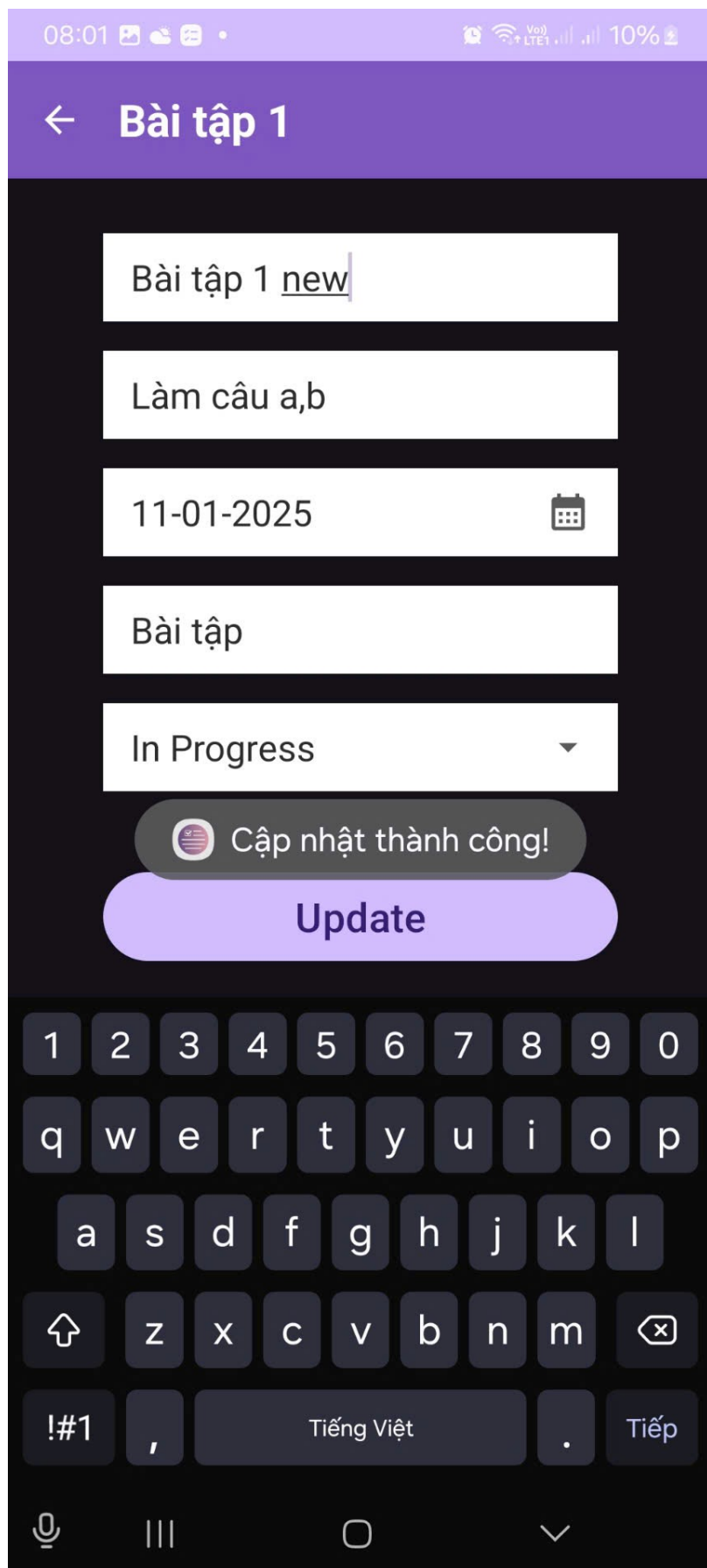
Hình 3.3.19: Màn hình khi nhấn vào nút và cập nhật trạng thái công việc



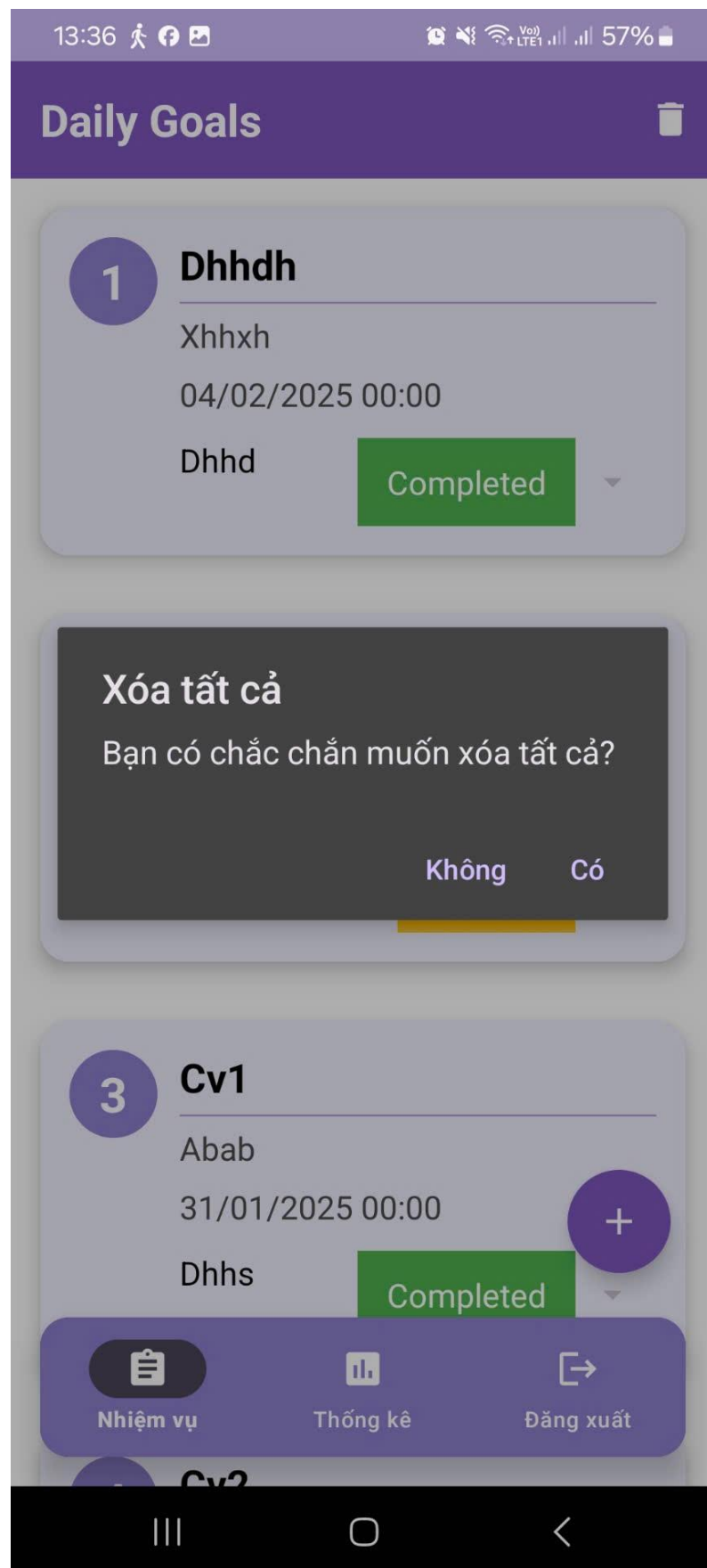
Hình 3.3.20: Màn hình thông báo khi công việc vừa được tạo, công việc còn 1 giờ nữa đến hạn, và danh sách công việc được gửi cố định vào 8 giờ sáng (nếu có công việc có deadline trong ngày)



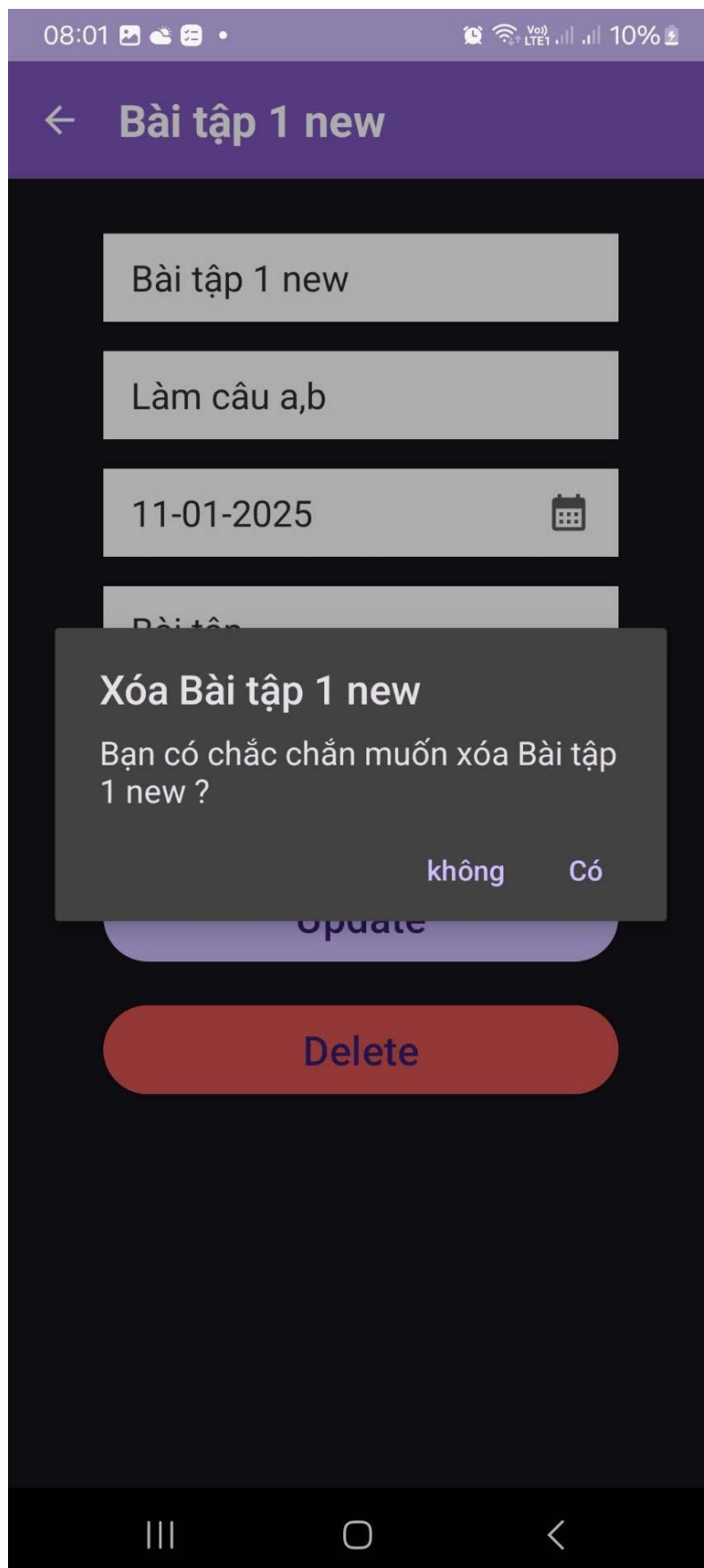
Hình 3.3.21: Màn hình cập nhật công việc



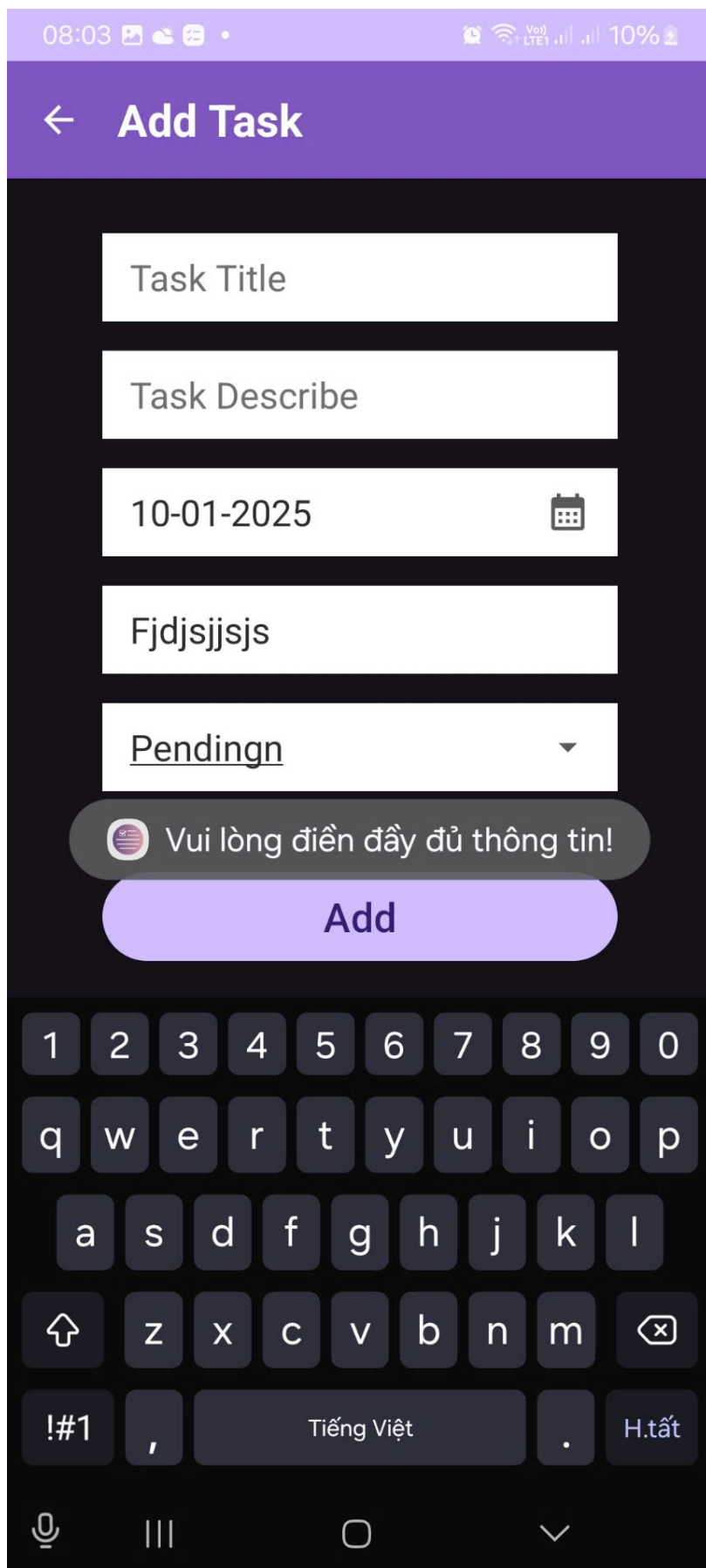
Hình 3.3.22: Màn hình khi cập nhật công việc thành công



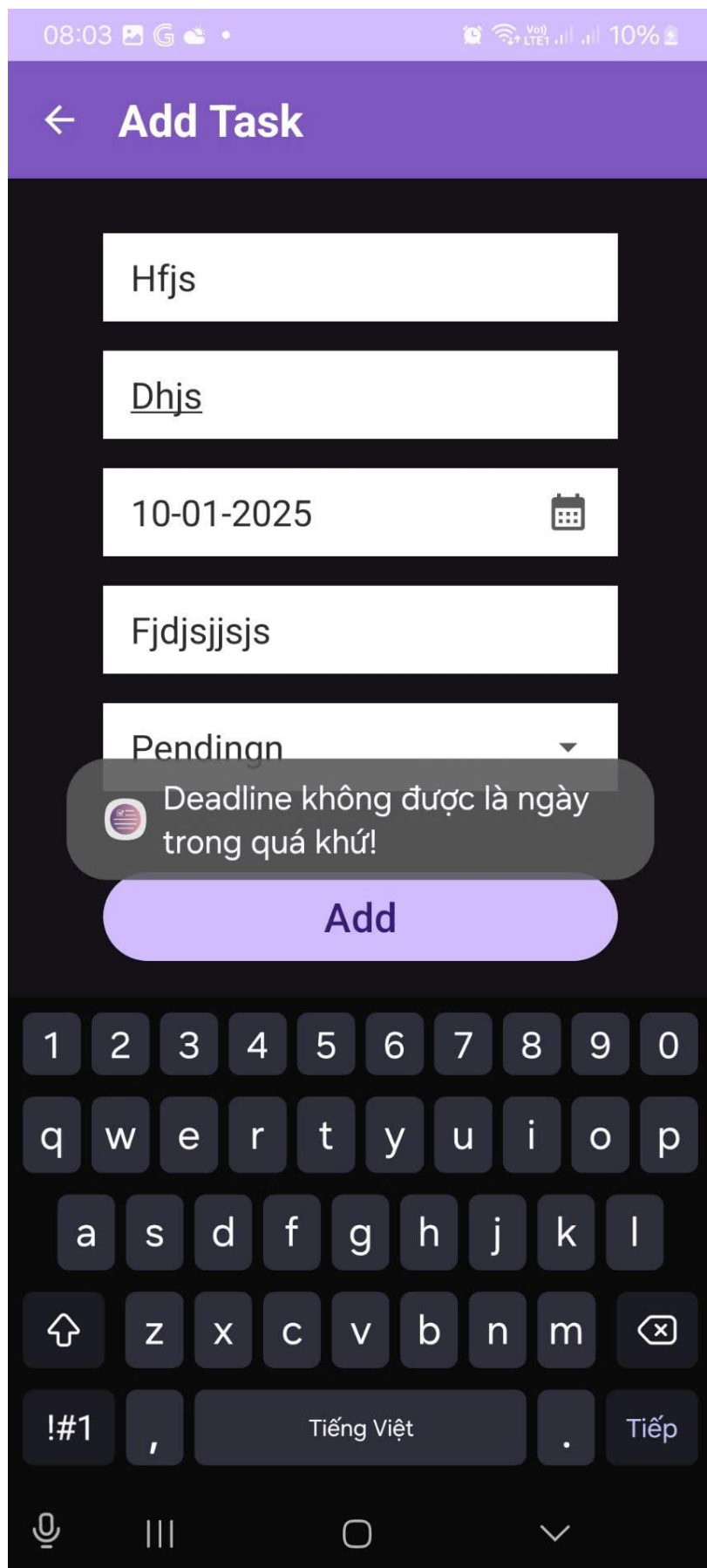
Hình 3.3.23: Màn hình khi ấn icon thùng rác để xóa tất cả



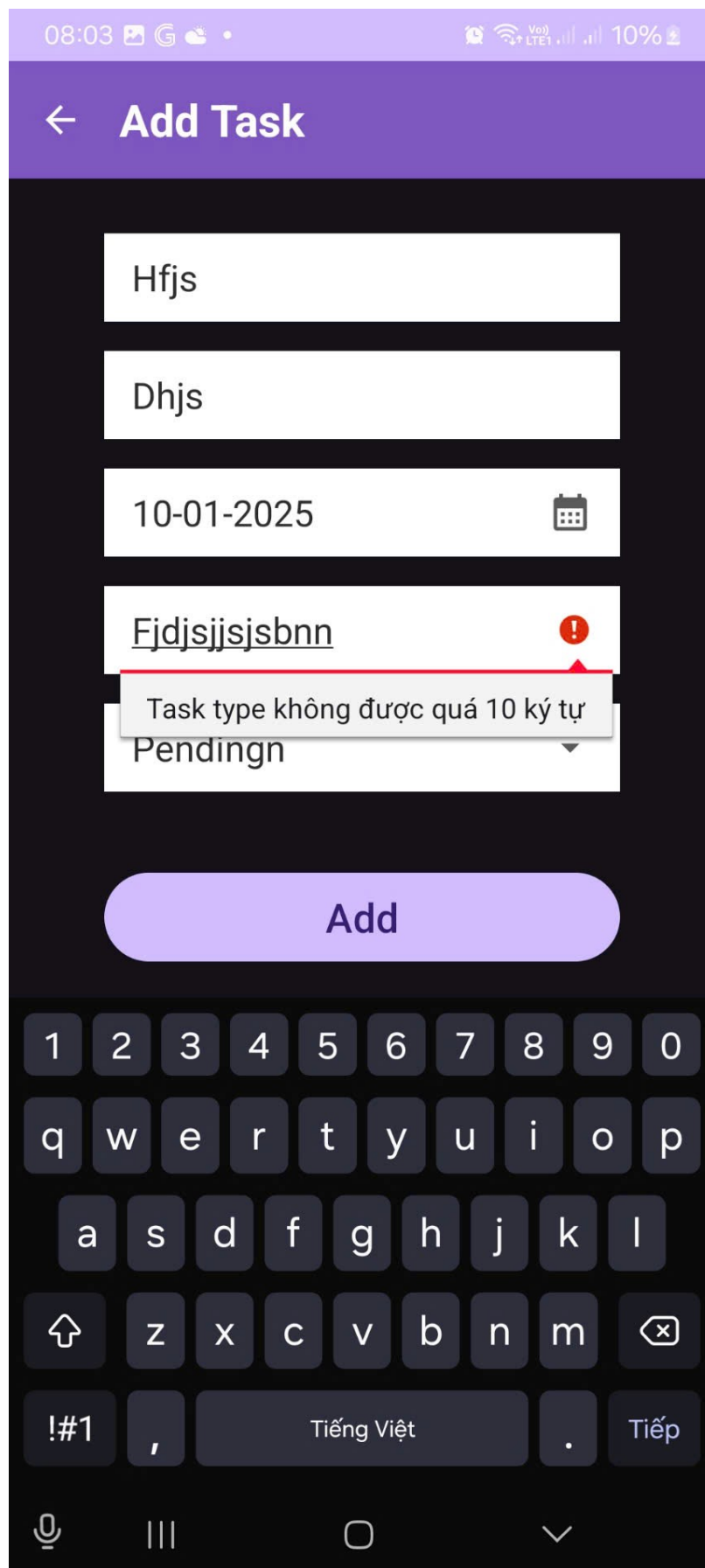
Hình 3.3.24: Màn hình khi ấn nút Delete để xóa 1 công việc



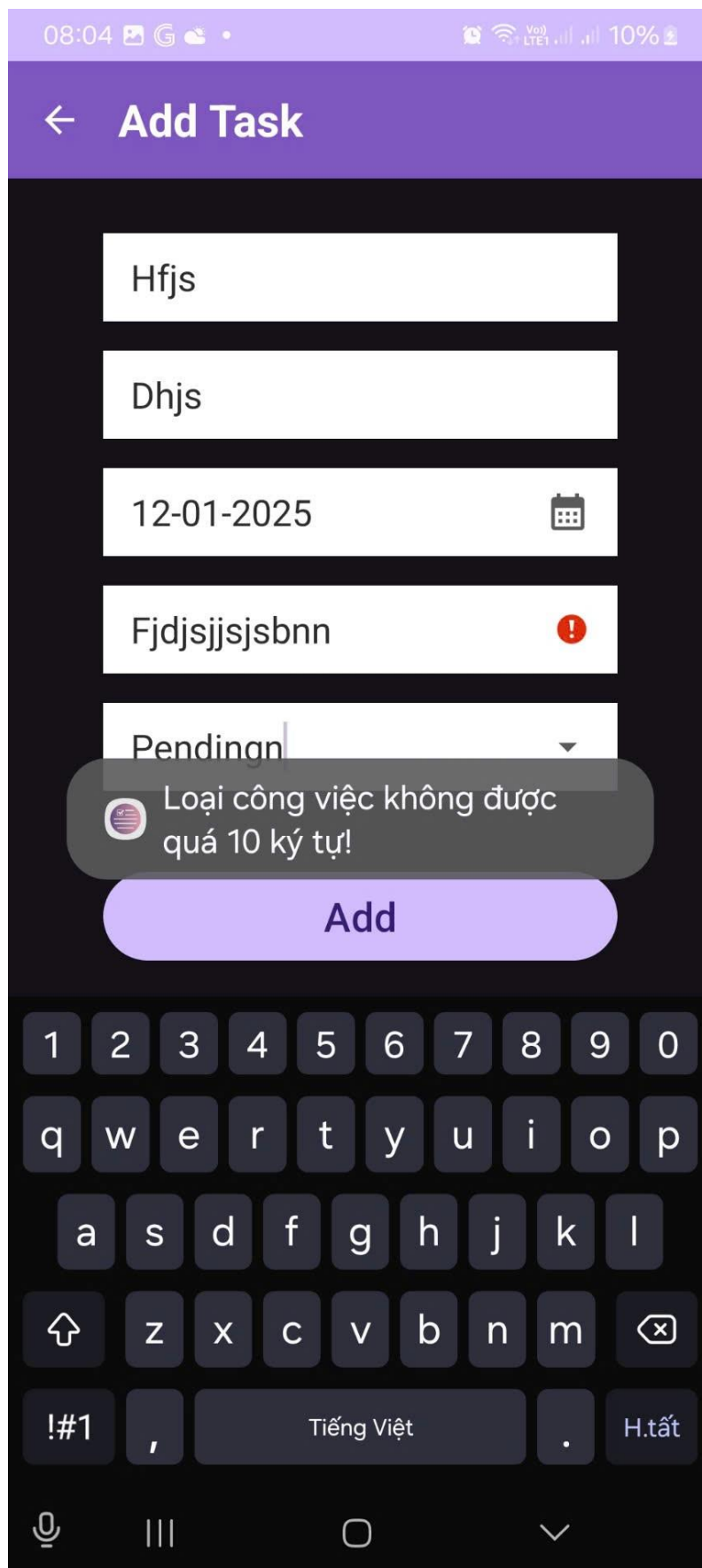
Hình 3.3.25: Màn hình thông báo lỗi khi chưa điền đầy đủ thông tin



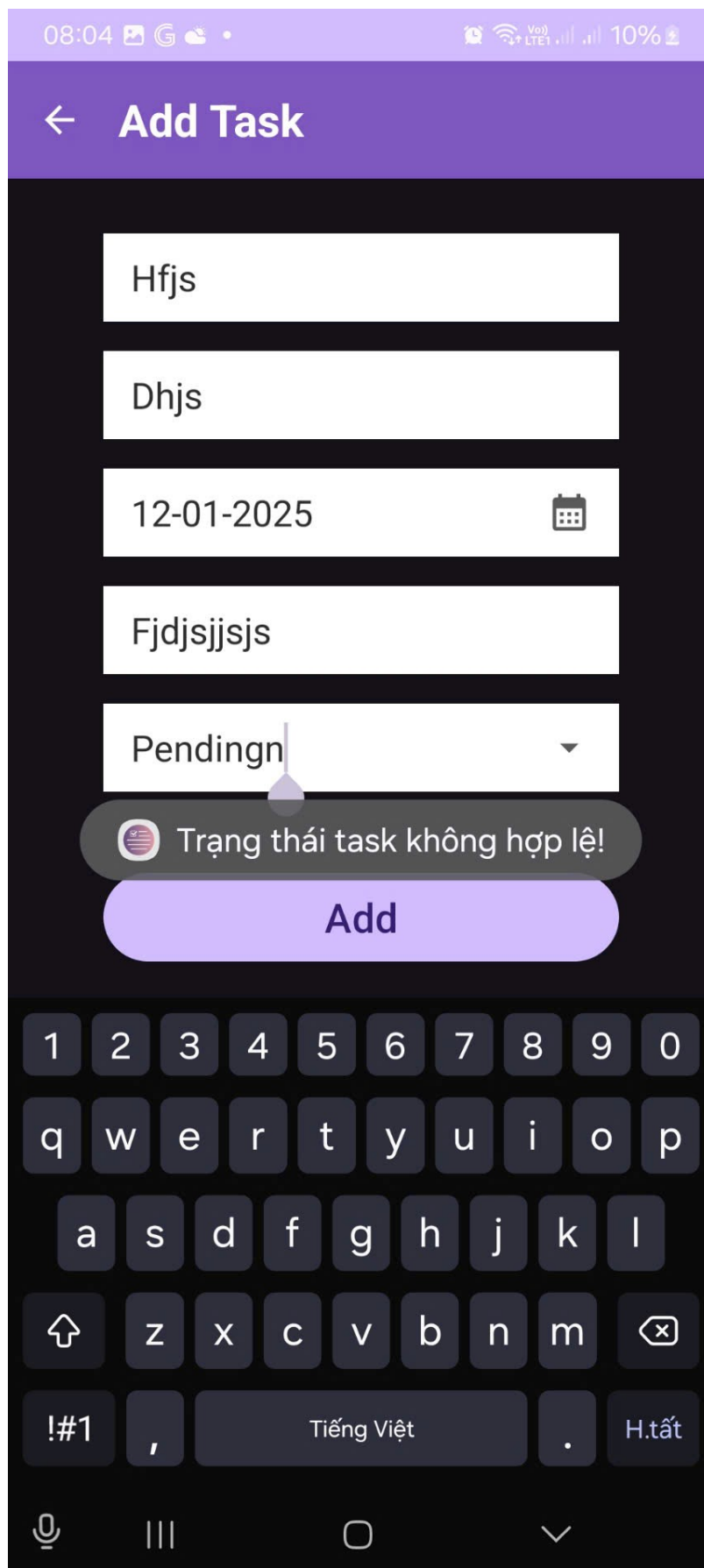
Hình 3.3.26: Màn hình thông báo lỗi khi nhập hạn chót là ngày trong quá khứ



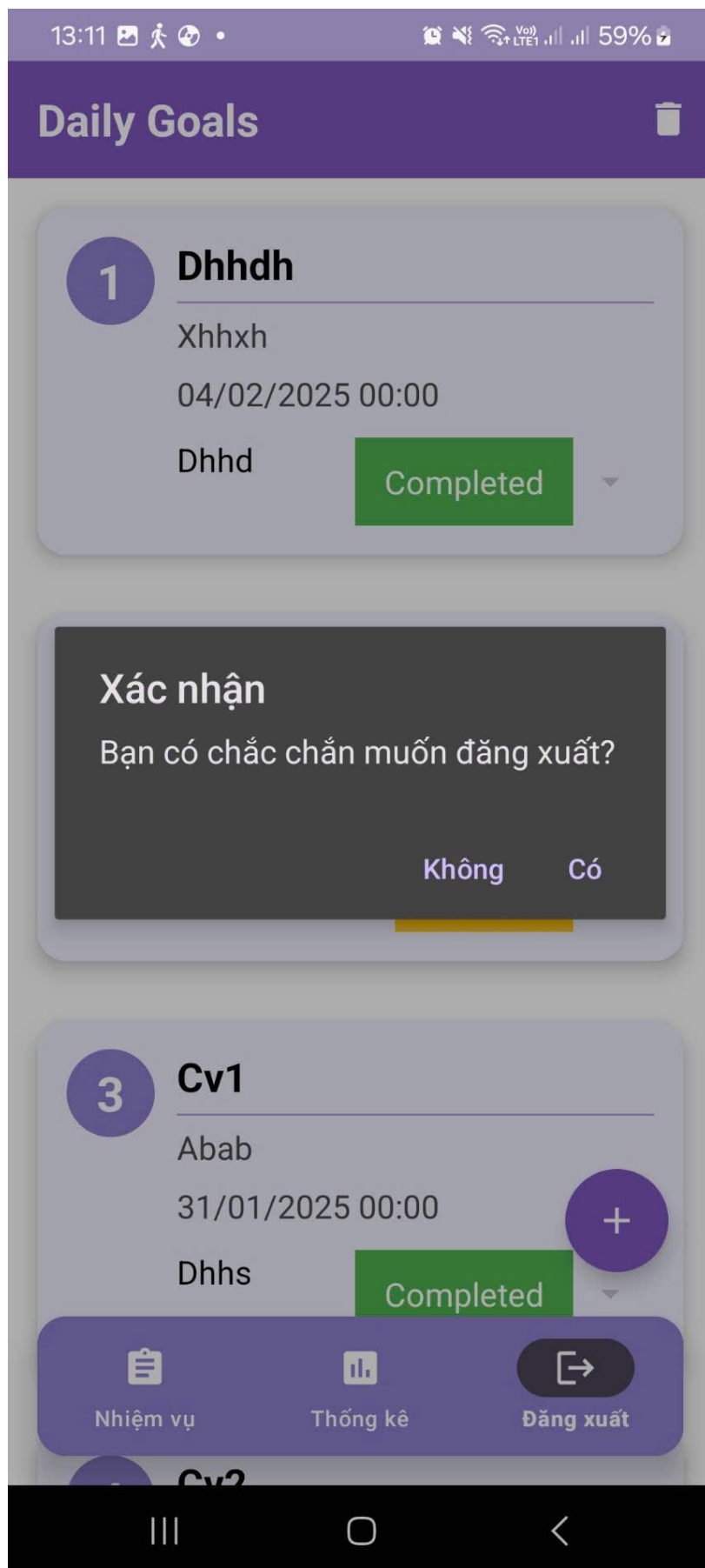
Hình 3.3.27: Màn hình thông báo lỗi khi nhập loại công việc quá 10 ký tự - a



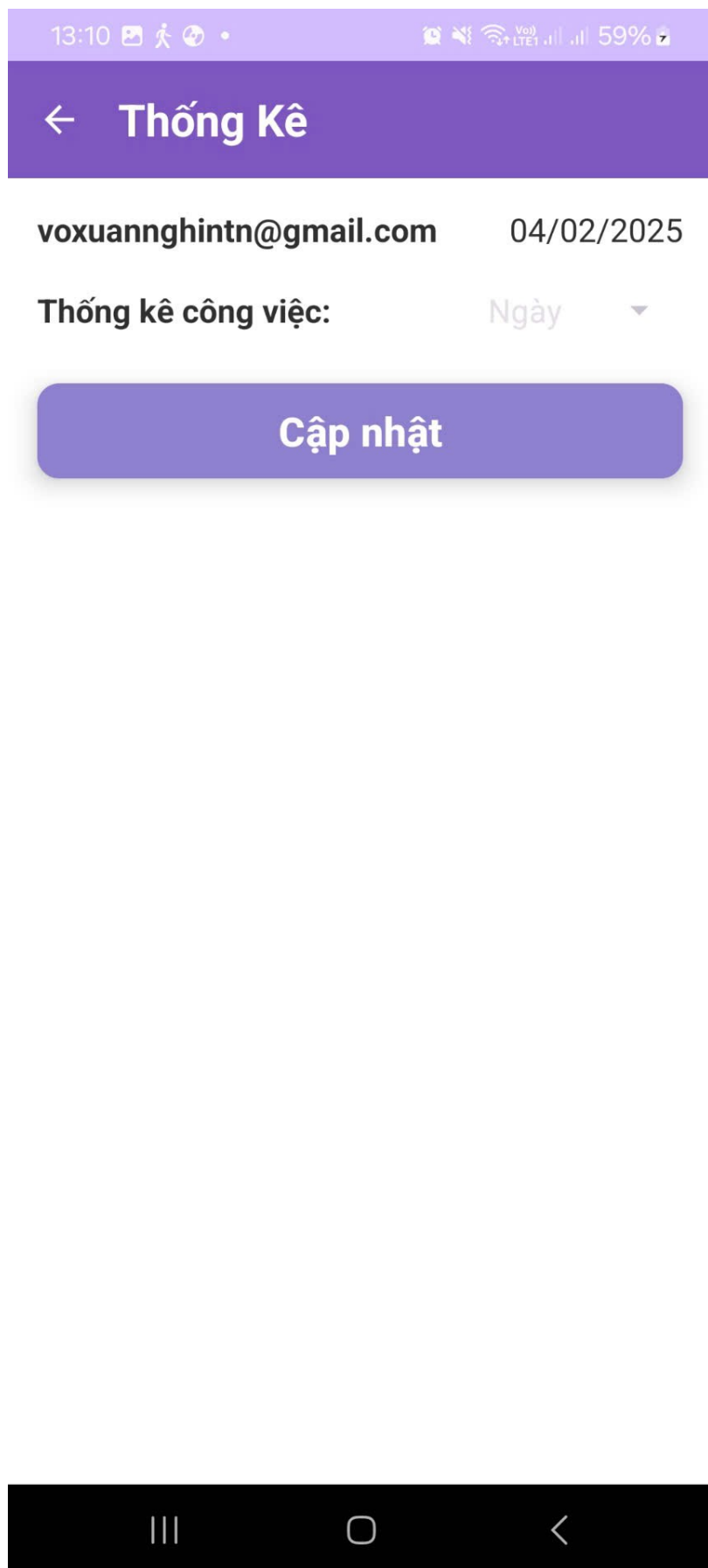
Hình 3.3.28: Màn hình thông báo lỗi khi nhập loại công việc quá 10 ký tự - b



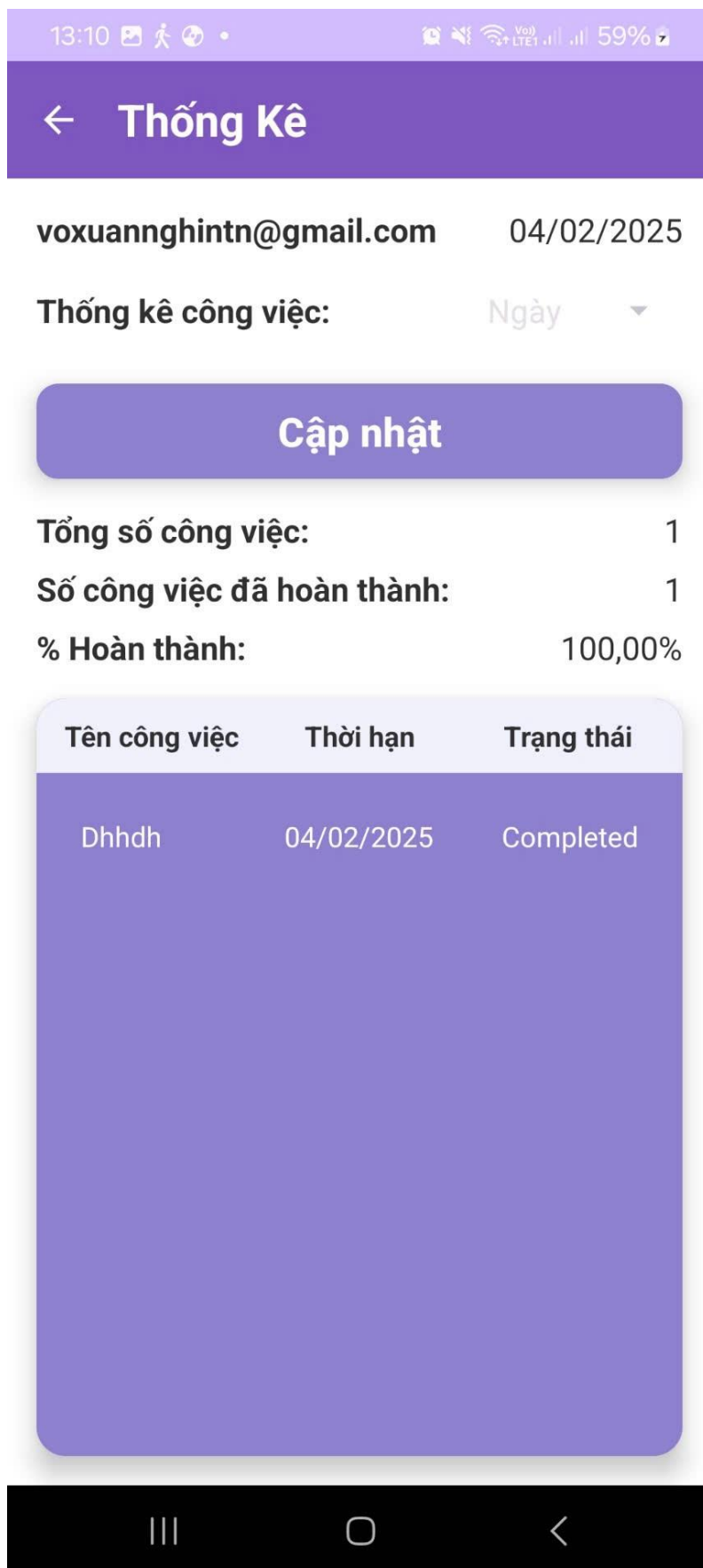
Hình 3.3.29: Màn hình thông báo lỗi khi nhập trạng thái công việc không hợp lệ



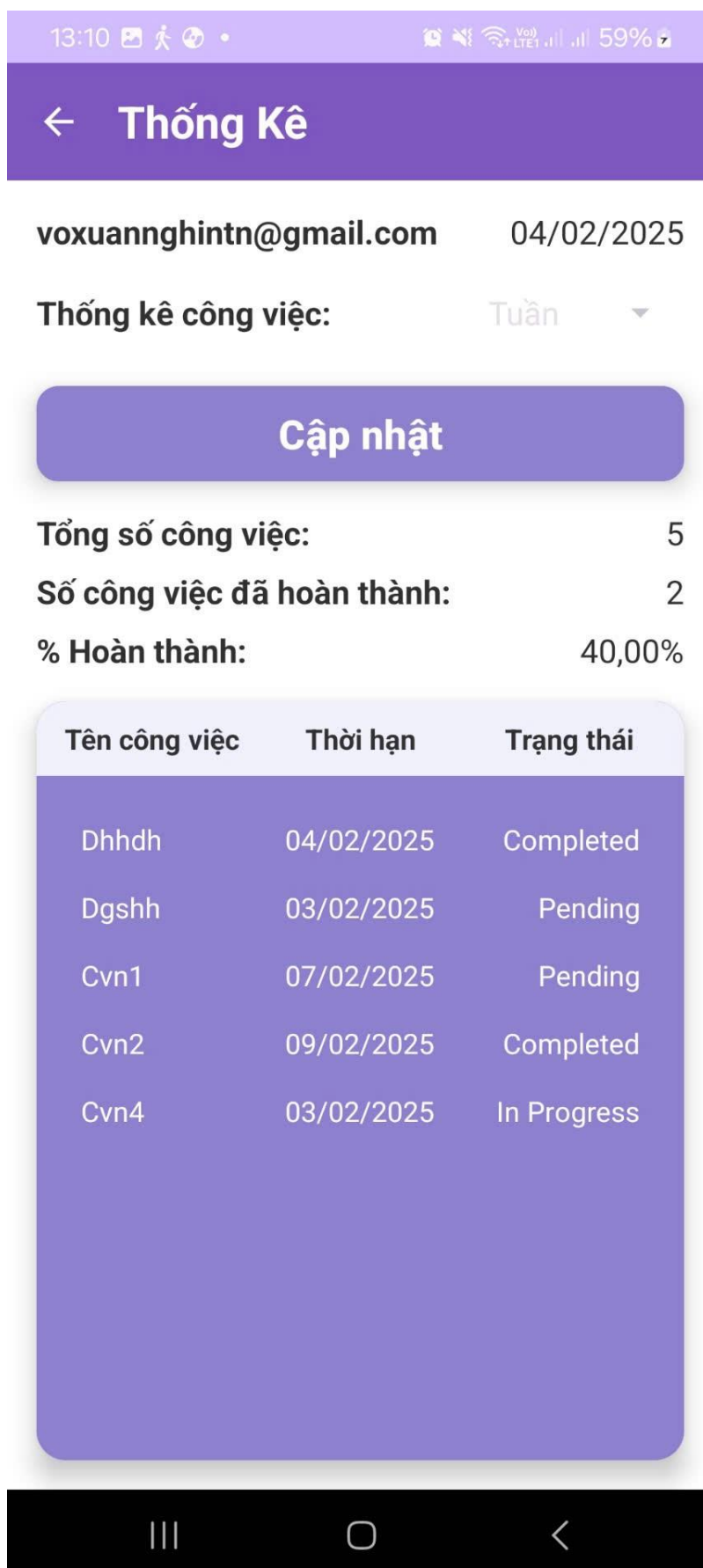
Hình 3.3.30: Màn hình xác nhận đăng xuất



Hình 3.3.31: Màn hình thống kê công việc khi chưa lựa chọn thời gian



Hình 3.3.32: Màn hình thống kê công việc theo ngày



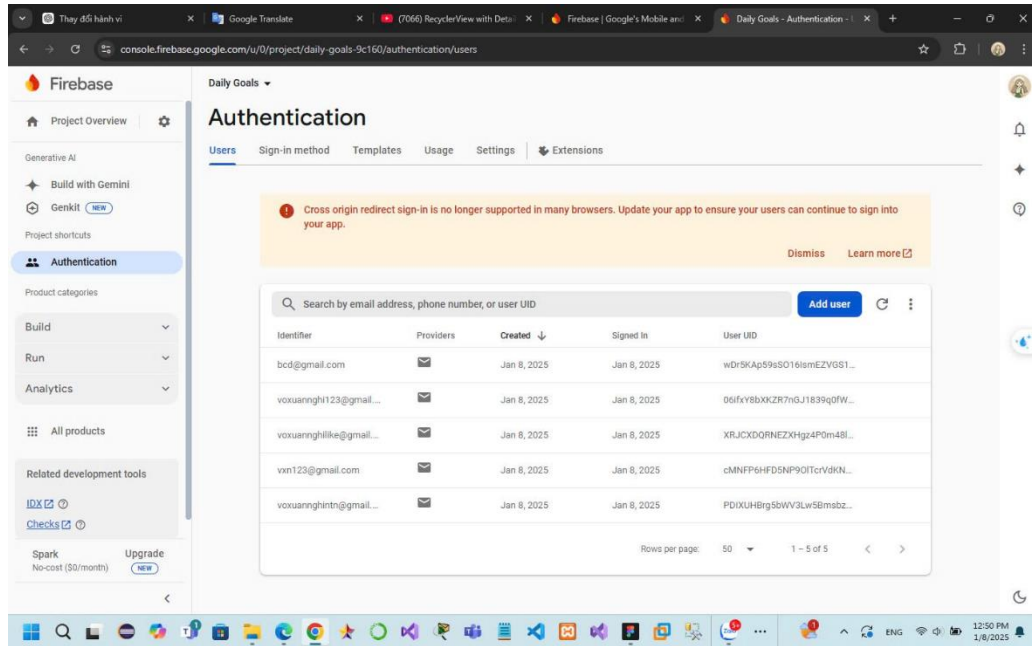
Hình 3.3.33: Màn hình thống kê công việc theo tuần



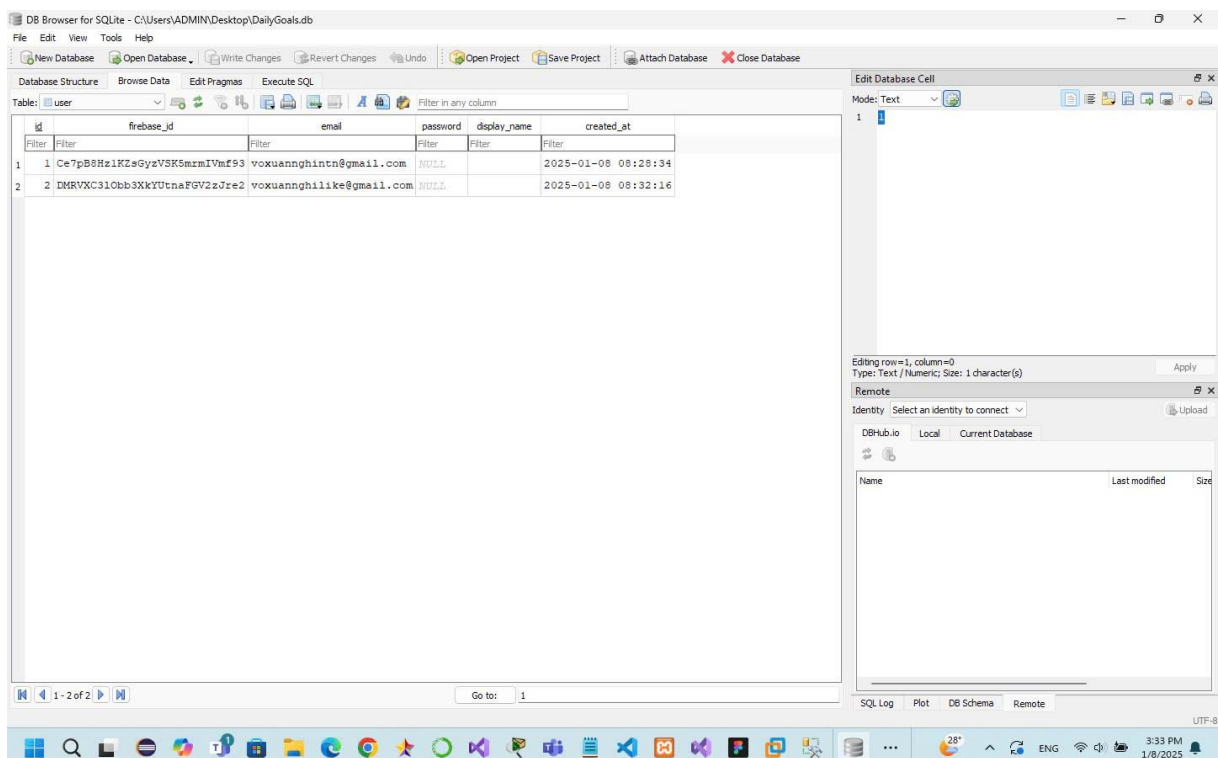
Hình 3.3.34: Màn hình thống kê công việc theo tháng

3.4 Kết nối cơ sở dữ liệu

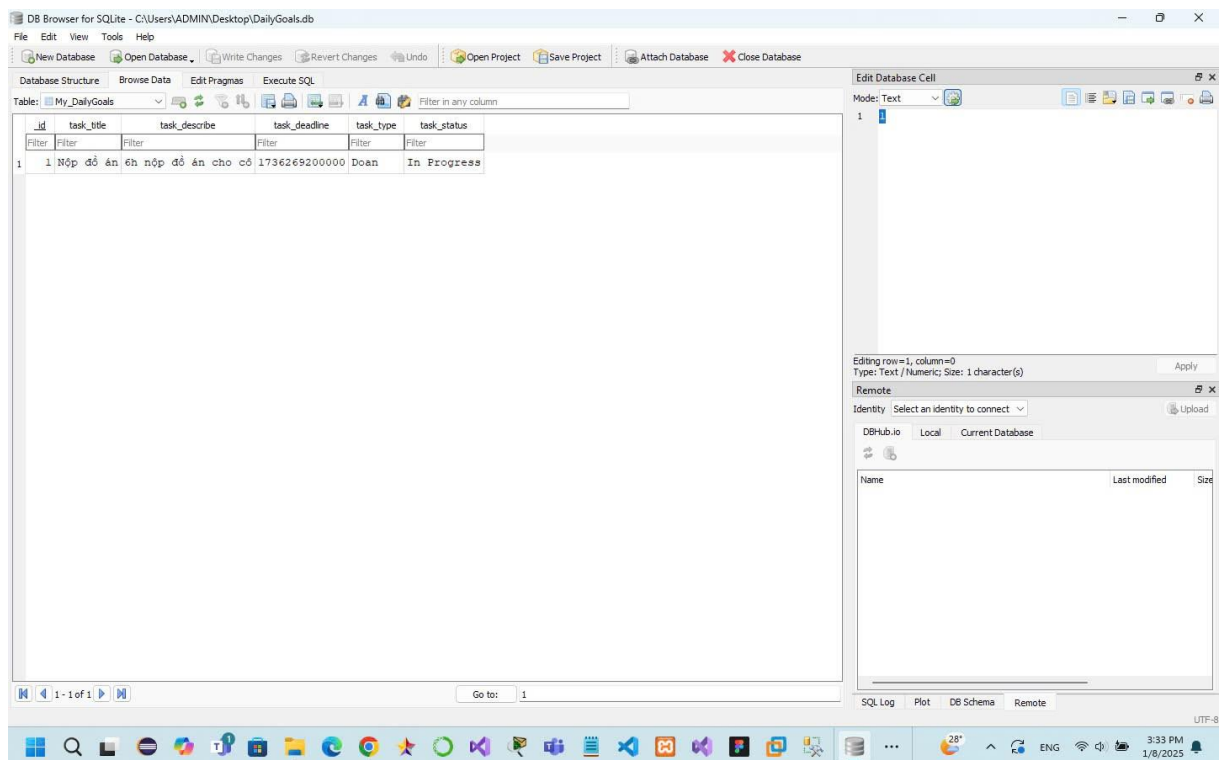
Kết nối dữ liệu Firebase với ứng dụng:



Hình 3.4.1: Kết nối Firebase



Hình 3.4.2: Dữ liệu user được tải vào SQLite từ Firebase



Hình 3.4.3: Dữ liệu task được tạo bởi user

Mô tả cách sử dụng class SQLiteOpenHelper để xử lý dữ liệu:

1. Tạo lớp kế thừa SQLiteOpenHelper.
2. Ghi đè phương thức onCreate() để tạo bảng.
3. Ghi đè phương thức onUpgrade() để xử lý khi nâng cấp cơ sở dữ liệu.
4. Thêm dữ liệu bằng phương thức insert().
5. Đọc dữ liệu bằng phương thức query() hoặc rawQuery().
6. Cập nhật dữ liệu bằng phương thức update().
7. Xóa dữ liệu bằng phương thức delete().

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết quả đạt được

- **Về chức năng:**

- Xây dựng được ứng dụng với đầy đủ các chức năng: quản lý công việc, nhắc nhở, thống kê.
- Giao diện thân thiện, dễ sử dụng

- **Về kỹ thuật:**

- Ứng dụng SQLite, firebase để lưu trữ dữ liệu và xác thực người dùng.
- Sử dụng AlarmManager để thiết lập nhắc nhở.
- Đảm bảo ứng dụng hoạt động ổn định trên Android 14.

- **Về mục tiêu:**

- Hoàn thành mục tiêu nghiên cứu ban đầu: hỗ trợ người dùng quản lý công việc cá nhân hiệu quả.

4.2 Hạn chế còn tồn đọng

- Chưa hỗ trợ đồng bộ dữ liệu với đám mây.
- Giới hạn ở một số tính năng cơ bản, chưa có trí tuệ nhân tạo để gợi ý lịch làm việc.
- Thiết kế chưa đa dạng, chưa tối ưu cho nhiều thiết bị.
- Khả năng xử lý khi có lượng lớn công việc chưa được kiểm thử sâu.

4.3 Hướng phát triển trong tương lai

- **Phát triển tính năng:**

Tích hợp trí tuệ nhân tạo để gợi ý công việc ưu tiên.

Đồng bộ hóa dữ liệu với các nền tảng đám mây như Google Drive.

- **Cải thiện giao diện:**

Thiết kế giao diện đa ngôn ngữ, hỗ trợ nhiều chủ đề hơn.

- **Tối ưu hiệu năng:**

Sử dụng Room Database thay cho SQLite để cải thiện khả năng truy xuất dữ liệu.

Kiểm tra và cải thiện hiệu suất cho các thiết bị cấu hình thấp.

TÀI LIỆU THAM KHẢO

1. <https://www.veed.io/edit/289a5e89-7e70-48f7-a2d1-a72413dfad2d/media-audio>
2. <https://convertio.co/vn/download/eab6905546342200edba7b717d1d93ff859249/>
3. <https://pixabay.com/vi/music/search/%c4%91%e1%bb%99ng%20l%e1%bb%b1c/>
4. <https://www.youtube.com/watch?v=hJPk50p7xwA&list=PLOV4JdZKVeR6m6elcOSQnOEaR22APrmuC&index=52>
5. <https://www.youtube.com/watch?v=KUNXaftGuM8&list=PLOV4JdZKVeR6m6elcOSQnOEaR22APrmuC&index=82>
6. <https://developer.android.com/develop/ui/views/notifications/navigation?hl=vi>
7. <https://developer.android.com/reference/androidx/core/app/ActivityCompat>
8. <https://developer.android.com/develop/ui/views/notifications/channels?hl=vi>
9. <https://viblo.asia/p/tim-hieu-sqlite-trong-android-rEBRAaQrR8Zj>
10. <https://howkteam.vn/course/khoa-hoc-lap-trinh-android-co-ban/luu-tru-du-lieu-voi-sqlite-1215>
11. <https://www.youtube.com/watch?v=TStttJRAPhE>
12. <https://topdev.vn/blog/android-studio-la-gi/#android-studio-la-gi>
13. https://fptcloud.com/firebase-la-gi/#1_Firebase_la_gi
14. <https://rdsic.edu.vn/blog/blog-3/sqlite-la-gi-vi-cb.html>