# MỤC LỤC

[MỤC LỤC I](#_Toc153525744)

[BẢNG PHÂN CHIA CÔNG VIỆC II](#_Toc153525745)

[DANH MỤC CÁC HÌNH VẼ III](#_Toc153525746)

[MỞ ĐẦU 1](#_Toc153525747)

[CHƯƠNG 1 : TỔNG QUAN VỀ LẬP TRÌNH DI ĐỘNG ,DART & FRAMEWORK FLUTTER 2](#_Toc153525748)

[1.1 Lập trình di động là gì? 2](#_Toc153525749)

[1.2 Dart là gì? 2](#_Toc153525750)

[1.3 Tại sao lại là Flutter? 3](#_Toc153525751)

[CHƯƠNG 2 : QUY TRÌNH PHÁT TRIỂN 5](#_Toc153525752)

[2.1 Giai đoạn 1: Lập kế hoạch và nghiên cứu 5](#_Toc153525753)

[2.2 Giai đoạn 2: Thiết kế hệ thống 5](#_Toc153525754)

[2.3 Giai đoạn 3: Phát triển và kiểm thử. 5](#_Toc153525755)

[2.4 Giai đoạn 4: Triển khai và bảo trì. 5](#_Toc153525756)

[CHƯƠNG 3 : CÁC CHỨC NĂNG CƠ BẢN VỀ ĐỀ TÀI – TASK MANAGEMENT 6](#_Toc153525757)

[3.1 Chức năng đăng ký, đăng nhập, quên mật khẩu. 6](#_Toc153525758)

[3.1.1 Chức năng đăng ký. 6](#_Toc153525759)

[3.1.2 Chức năng quên mật khẩu 9](#_Toc153525760)

[3.1.3 Chức năng đăng nhập 16](#_Toc153525761)

[3.2 Màn hình chính 18](#_Toc153525762)

[3.3 Màn hình chi tiết 20](#_Toc153525763)

[3.3.1 Màn hình Tasks Screen 20](#_Toc153525764)

[3.3.2 Màn hình Add Tasks Screen 21](#_Toc153525765)

[3.3.3 Menu 26](#_Toc153525766)

[3.3.4 Màn hình thông tin cá nhân 28](#_Toc153525767)

[3.3.5 Màn hình thống kê tiến độ 29](#_Toc153525768)

[CHƯƠNG 4 PHÂN TÍCH THIẾT KẾ HỆ THỐNG 30](#_Toc153525769)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 31](#_Toc153525770)

# BẢNG PHÂN CHIA CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| Họ và tên | Mã sinh viên | Nhiệm vụ |
| Phạm Bá Thọ (C) | 2021050819 |  |
| Phạm Tuấn Lực | 2021050417 |  |
| Phạm Xuân Trường | 2021050901 |  |
| Nguyễn Hải Đăng | 2021050848 |  |

# DANH MỤC CÁC HÌNH VẼ

[Hình 3.1.1‑1 Màn hình đăng ký tài khoản. 7](#_Toc153525325)

[Hình 3.1.1‑2 Ví dụ nhập liệu màn hình đăng ký 8](#_Toc153525326)

[Hình 3.1.1‑3 Nơi lưu trữ thông tin đăng ký tài khoản 9](#_Toc153525327)

[Hình 3.1.2‑1 aleart quên mật khẩu – nhập tài khoản cần lấy lại mật khẩu 10](#_Toc153525328)

[Hình 3.1.2‑2 Ví dụ về nhập liệu tài khoản cần lấy lại mật khẩu 11](#_Toc153525329)

[Hình 3.1.2‑3 aleart quên mật khẩu – nhập liệu mã code gửi về email đã yêu cầu 12](#_Toc153525330)

[Hình 3.1.2‑4 Ví dụ về nhập liệu mã code 13](#_Toc153525331)

[Hình 3.1.2‑5 aleart quên mật khẩu – nhập mật khẩu mới 14](#_Toc153525332)

[Hình 3.1.2‑6 Ví dụ về nhập mật khẩu mới 15](#_Toc153525333)

[Hình 3.1.3‑1 Màn hình đăng nhập 16](#_Toc153525334)

[Hình 3.1.3‑2 Ví dụ về màn hình đăng nhập 17](#_Toc153525335)

[Hình 3.1.3‑1 Giao diện màn hình chính sau khi đăng nhập 18](#_Toc153525336)

[Hình 3.1.3‑2 Giao diện Complete 19](#_Toc153525337)

[Hình 3.3.1‑1 Giao diện màn hình Tasks Screen 20](#_Toc153525338)

[Hình 3.3.2‑1 Màn hình Add Tasks Screen 22](#_Toc153525339)

[Hình 3.3.2‑2 Màn hình Select a Date 23](#_Toc153525340)

[Hình 3.3.2‑3 Màn hình Time Start 24](#_Toc153525341)

[Hình 3.3.2‑4 Màn hình Time End 25](#_Toc153525342)

[Hình 3.3.3‑1 Menu dạng pop-up 26](#_Toc153525343)

[Hình 3.3.3‑2 Menu dạng danh sách 27](#_Toc153525344)

[Hình 3.3.4‑1 Trang thông tin cá nhân người dùng 28](#_Toc153525345)

[Hình 3.3.5‑1 Màn hình thống kê tiến độ 29](#_Toc153525346)

# MỞ ĐẦU

Trong thời đại hiện đại ngày nay, cuộc sống nhanh chóng và bận rộn đặt ra nhiều thách thức trong việc quản lý công việc hàng ngày. Sự phát triển của công nghệ di động, đặc biệt là thông qua smartphone, đã mang lại những cơ hội mới và cách tiếp cận hiệu quả hơn đối với việc quản lý nhiệm vụ và công việc cá nhân.

Việc có thể quản lý công việc mọi nơi và mọi lúc thông qua ứng dụng di động không chỉ giúp tiết kiệm thời gian mà còn tăng cường tính linh hoạt. Người dùng có thể dễ dàng theo dõi tiến độ công việc, đặt ưu tiên, và giao việc từ xa, không bị ràng buộc bởi vị trí địa lý hay thời gian. Điều này đặc biệt quan trọng trong bối cảnh môi trường làm việc ngày càng toàn cầu hóa và đòi hỏi tính linh hoạt cao.

Sự tích hợp của trí tuệ nhân tạo và học máy trong các ứng dụng quản lý công việc cũng đem lại những lợi ích to lớn. Khả năng dự đoán thời gian hoàn thành, gợi ý thông minh, và phân loại công việc theo độ ưu tiên giúp người dùng quản lý công việc một cách hiệu quả và khoa học hơn.

Từ những phân tích trên, ý tưởng cho một ứng dụng quản lý hoạt động trên di động hình thành nhằm mục đích giúp cho người sử dụng có thể tối ưu hóa được thời gian và sắp xếp được những việc của mình phù hợp nhất.

Trong báo cáo này sẽ tập trung trình bày những nghiên cứu kiến thức cơ bản về lập trình di động đa nền tảng bằng ngôn ngữ lập trình Dart – framework Flutter. Triển khai các bước thiết kế, xây dựng ứng dụng giúp người dùng có thể đăng nhập,đăng ký tài khoản,.....

Đây là ứng dụng thực tế khá hay trên điện thoại, tuy không phải là mới nhưng do kinh nghiệm của nhóm em chưa nhiều và còn nhiều hạn chế cũng như sai sót mong thầy và các bạn đóng góp ý kiến để đề tài được hoàn thiện hơn.

# : TỔNG QUAN VỀ LẬP TRÌNH DI ĐỘNG ,DART & FRAMEWORK FLUTTER

## Lập trình di động là gì?

Lập trình di động hiện nay không chỉ là một lĩnh vực kỹ thuật mà còn là nguồn động viên mạnh mẽ đằng sau sự phát triển của nền công nghiệp công nghệ. Với sự đa dạng ngôn ngữ lập trình như Swift, Objective-C, Java, Kotlin, Dart, và JavaScript, nhà phát triển có nhiều lựa chọn để tạo ra ứng dụng cho cả hai hệ điều hành chính là iOS và Android.

Các framework đa nền tảng như Flutter và React Native mang lại khả năng phát triển ứng dụng một lần và chạy trên nhiều nền tảng, giúp giảm chi phí và thời gian phát triển. Tính tương tác và hiệu suất của ứng dụng ngày càng được đặt lên hàng đầu, với việc chú trọng đến giao diện người dùng đáp ứng và sử dụng các tính năng đặc biệt như camera, GPS, và cảm biến để tạo ra trải nghiệm người dùng độc đáo.

Công cụ phát triển như Xcode, Android Studio, Visual Studio, và Visual Studio Code đang ngày càng được cập nhật để hỗ trợ quá trình phát triển hiệu quả. Bảo mật dữ liệu và quản lý người dùng là những ưu tiên quan trọng, đặc biệt trong bối cảnh nhu cầu ngày càng cao về sự an toàn và quyền riêng tư.

Xu hướng thiết kế đang ngày càng chú trọng đến sự đơn giản, tối giản, và màu sắc tươi tắn. Cộng đồng lập trình di động ngày càng mạnh mẽ, với sự chia sẻ kiến thức và tài nguyên thông qua nhiều nền tảng trực tuyến.

Cuối cùng, lập trình di động không chỉ là việc tạo ra các ứng dụng hữu ích mà còn là quá trình sáng tạo và thú vị, đưa người ta vào thế giới không ngừng biến đổi và đổi mới của công nghệ di động.

## Dart là gì?

Dart là một ngôn ngữ lập trình được Google phát triển, được thiết kế với mục tiêu hỗ trợ việc xây dựng ứng dụng di động và web hiệu quả và linh hoạt. Được giới thiệu vào năm 2011, Dart đã trở thành ngôn ngữ chính trong một số dự án quan trọng của Google, đặc biệt là khi kết hợp với framework Flutter.

Đặc điểm nổi bật của Dart bao gồm tính đồng bộ và bất đồng bộ, cung cấp sự linh hoạt trong xử lý luồng và tương tác với các tác vụ không đồng bộ. Một ưu điểm quan trọng của Dart là khả năng sử dụng Hot Reload, cho phép nhà phát triển xem ngay lập tức kết quả của các thay đổi mã nguồn mà không cần phải khởi động lại ứng dụng, giúp tăng độ nhanh chóng trong quá trình phát triển.

Dart thường được ứng dụng mạnh mẽ trong việc phát triển ứng dụng di động thông qua framework Flutter, nơi mà sự kết hợp giữa Dart và Flutter mang lại khả năng xây dựng ứng dụng đa nền tảng (iOS và Android) từ một mã nguồn duy nhất. Đồng thời, Dart cũng có sẵn framework AngularDart cho việc phát triển ứng dụng web, đưa ra lựa chọn linh hoạt cho những dự án phát triển đa dạng trên nền tảng web và di động.

## Tại sao lại là Flutter?

Như đã trình bày ở trên, Flutter là một framework mã nguồn mở của Google, đang ngày càng trở thành một lựa chọn phổ biến trong việc phát triển ứng dụng di động đa nền tảng. Lý do nhiều nhà phát triển chọn sử dụng Flutter có thể được giải thích thông qua những ưu điểm đặc sắc mà nó mang lại.

Đầu tiên, Flutter cung cấp khả năng phát triển ứng dụng một lần và chạy trên cả hai hệ điều hành chính là iOS và Android. Điều này giúp giảm đau đầu về quản lý mã nguồn và chi phí phát triển, đồng thời tạo ra trải nghiệm người dùng đồng nhất trên mọi nền tảng.

Một ưu điểm quan trọng khác là Hot Reload, tính năng cho phép nhà phát triển xem ngay lập tức kết quả của các thay đổi mã nguồn mà không cần phải khởi động lại ứng dụng. Điều này giúp tăng tốc quá trình phát triển, làm cho việc thử nghiệm, debug và điều chỉnh trở nên linh hoạt và hiệu quả.

Flutter còn nổi tiếng với Widget-based UI, với mọi thành phần trong ứng dụng đều là một widget. Điều này giúp tạo ra giao diện người dùng đẹp mắt và dễ bảo trì. Hơn nữa, Flutter hỗ trợ nhiều chủ đề thiết kế, cho phép tùy chỉnh giao diện theo yêu cầu cụ thể của dự án.

Cuối cùng, sự tích hợp sẵn sàng với Dart, ngôn ngữ lập trình hiện đại và linh hoạt, làm cho Flutter trở thành một công cụ mạnh mẽ cho cả những nhà phát triển mới và có kinh nghiệm. Những lợi ích này đồng tỏi đưa Flutter trở thành một lựa chọn hấp dẫn để xây dựng ứng dụng di động hiện đại và đa nền tảng.

* Tóm lại, sự kết hợp giữa lập trình di động, Dart và Flutter đang định hình một tương lai đầy triển vọng và đa dạng cho ngành công nghiệp công nghệ di động.

# : QUY TRÌNH PHÁT TRIỂN

Quy trình phát triển ứng dụng Task Management được chia làm 4 giai đoạn:

## Giai đoạn 1: Lập kế hoạch và nghiên cứu

* Đề tài: Task Management
* Lý do chọn đề tài: Như những lý do đã đề cập ở trên, ngoài ra còn mang lại hiểu biết sâu sắc về cách tổ chức và quản lý công việc, giá trị thực tế và áp dụng rộng rãi trong nhiều tình huống cuộc sống và làm việc.
* Công cụ thực hiện: VS Code, Framework Flutter, FireBase, GitHub,..
* Phân chia công việc cho các thành viên trong nhóm.

## Giai đoạn 2: Thiết kế hệ thống

* Thiết kế giao diện người dùng: Figma.
* Cơ sở dữ liệu: FireBase.
* Sơ đồ UseCase (chưa làm được).

## Giai đoạn 3: Phát triển và kiểm thử.

* Phát triển ứng dụng: đề tài được chia đều cho các thành viên trong nhóm, đảm bảo mỗi thành viên có thể tự tạo ra được 1 screen.
* Kiểm thử hệ thống:
* Thực hiện kiểm thử đơn vị cho từng phần của hệ thống.
* Thực hiện kiểm thử tích hợp để đảm bảo các thành phần hoạt động chính xác khi kết hợp.
* Kiểm thử giao diện người dùng: Kiểm tra trải nghiệm người dùng và đảm bảo giao diện hoạt động mượt mà và đáp ứng.

## Giai đoạn 4: Triển khai và bảo trì.

* Triển khai: Kiểm tra tính ổn định của ứng dụng, sửa lỗi phát sinh từ giai đoạn 3.
* Hỗ trợ và bảo trì: Trình bày về cách ứng dụng hoạt động.

# : CÁC CHỨC NĂNG CƠ BẢN VỀ ĐỀ TÀI – TASK MANAGEMENT

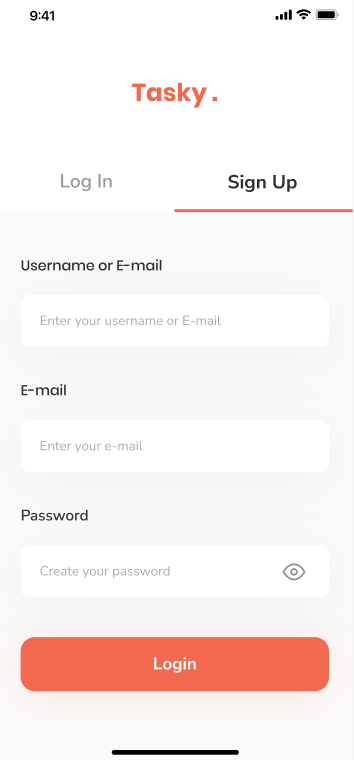
Task Management (Quản lý công việc) là một quy trình hoặc hệ thống để xác định, theo dõi, và quản lý công việc và nhiệm vụ trong một dự án hoặc tổ chức. Mục tiêu chính của quản lý công việc là tối ưu hóa sự hiệu quả và hiệu suất của cá nhân hoặc nhóm làm việc.

## Chức năng đăng ký, đăng nhập, quên mật khẩu.

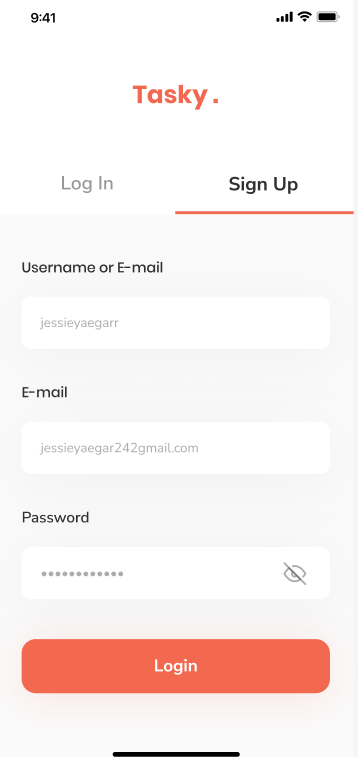
### Chức năng đăng ký.

* Cho phép người dùng tạo tài khoản mới bằng cách nhập thông tin: username, email, password
* Xác nhận tính hợp lệ của địa chỉ email (chưa làm được)

Hình 3.1.1‑1 Màn hình đăng ký tài khoản.

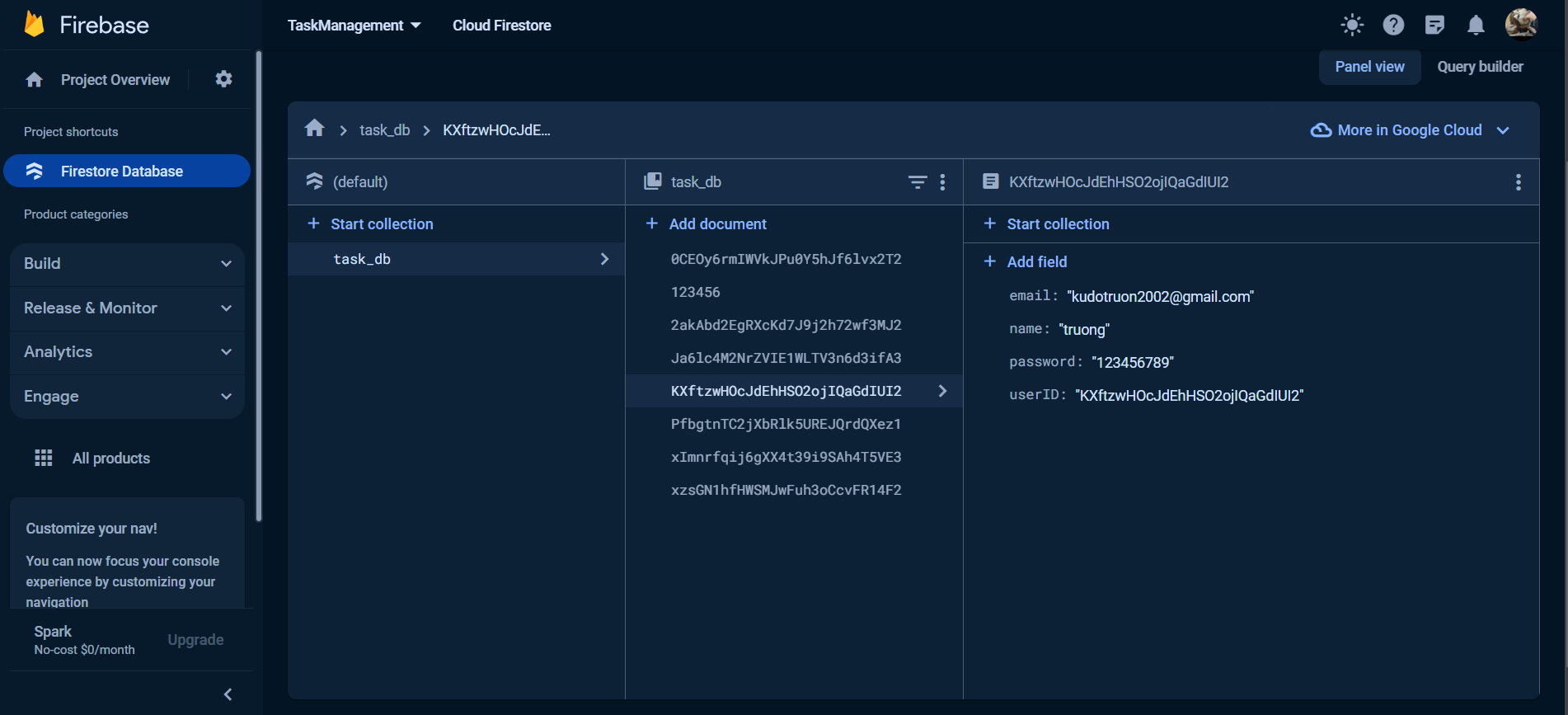


Hình 3.1.1‑2 Ví dụ nhập liệu màn hình đăng ký



Sau khi đăng ký thành công, thông tin tài khoảng sẽ được lưu vào cơ sở dữ liệu (ở đây nhóm em dùng firebase để quản lý)

Hình 3.1.1‑3 Nơi lưu trữ thông tin đăng ký tài khoản

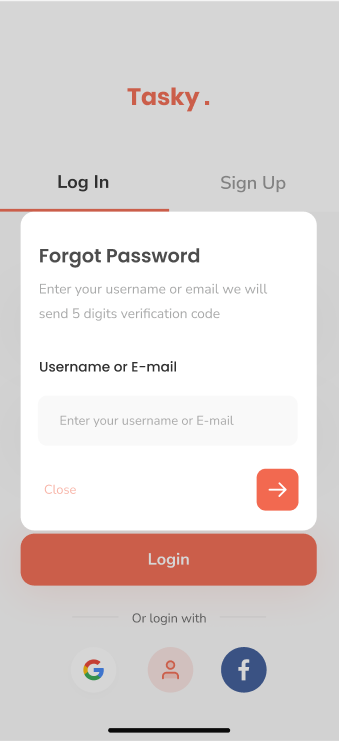


### Chức năng quên mật khẩu

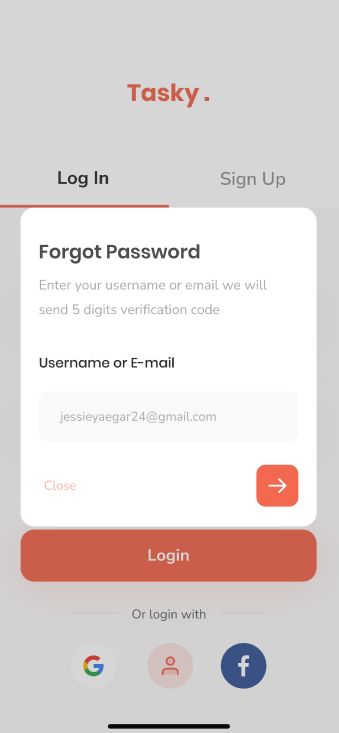
* Cung cấp tính năng khôi phục mật khẩu trong trường hợp người dùng quên mật khẩu.
* Gửi về email mã số để xác minh cho phép người dùng cập nhật mật khẩu mới.

Khi người dùng Click vào ‘Forgot Password’ aleart quên mật khẩu sẽ hiện lên để người dùng thao tác.

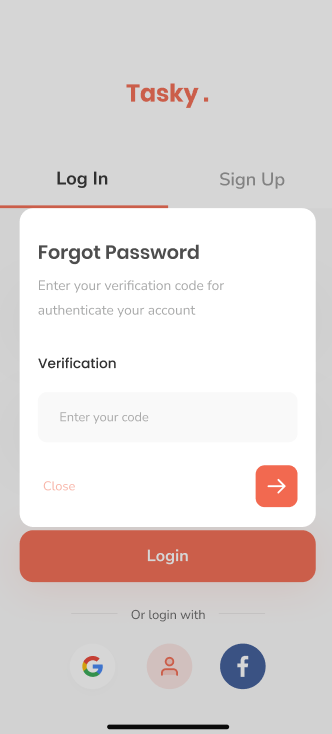
Hình 3.1.2‑1 aleart quên mật khẩu – nhập tài khoản cần lấy lại mật khẩu



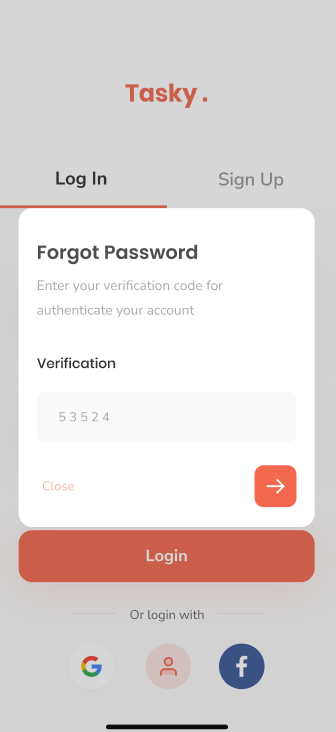
Hình 3.1.2‑2 Ví dụ về nhập liệu tài khoản cần lấy lại mật khẩu



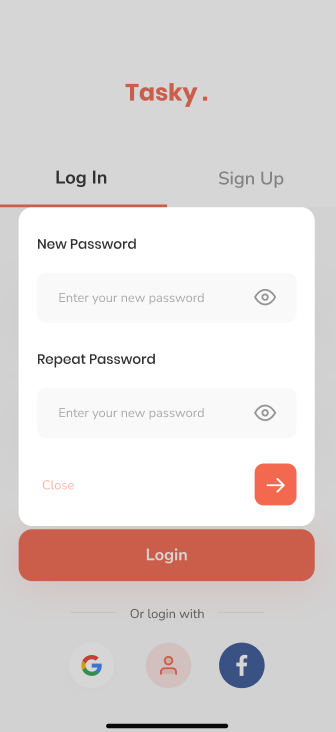
Hình 3.1.2‑3 aleart quên mật khẩu – nhập liệu mã code gửi về email đã yêu cầu



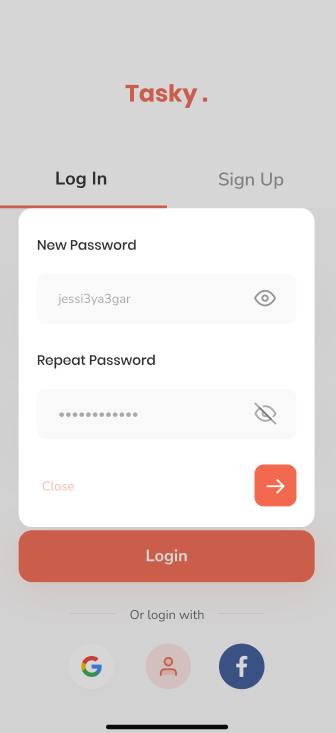
Hình 3.1.2‑4 Ví dụ về nhập liệu mã code



Hình 3.1.2‑5 aleart quên mật khẩu – nhập mật khẩu mới



Hình 3.1.2‑6 Ví dụ về nhập mật khẩu mới

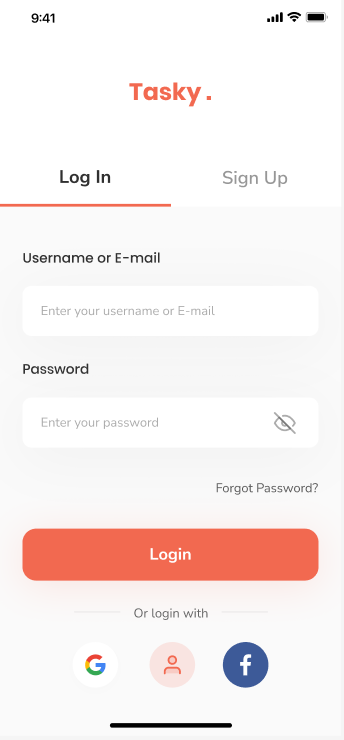


Sau khi lấy lại được mật khẩu thành công, người dùng có thể đăng nhập vào ứng dụng bằng tên đăng nhập và mật khẩu mới.

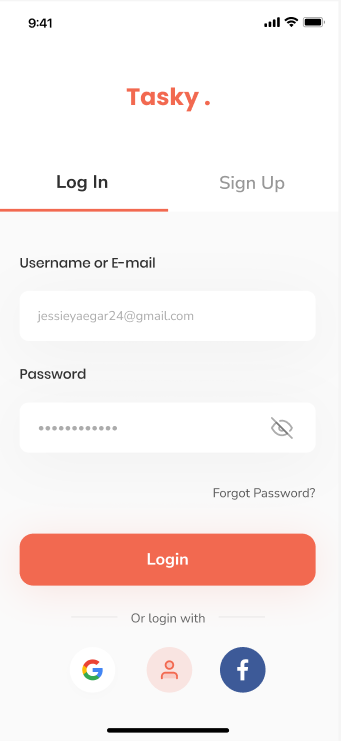
### Chức năng đăng nhập

* Khi đã có tài khoản trước đó hoặc đăng ký tài khoản thành công, người dùng có thể đăng nhập vào hệ thống bằng cách cung cấp thông tin đăng nhập email, password.
* Sử dụng cơ chết xác thực an toàn để bảo vệ thông tin người dùng ( chưa làm được)

Hình 3.1.3‑1 Màn hình đăng nhập



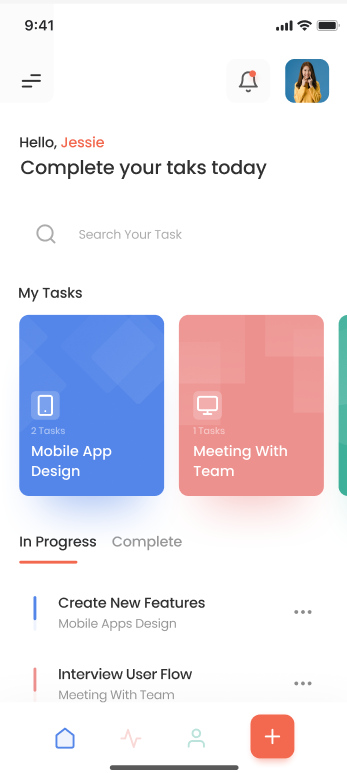
Hình 3.1.3‑2 Ví dụ về màn hình đăng nhập



Sau khi nhập đầy đủ và đúng tên đăng nhập và mật khẩu, ứng dụng sẽ điều hướng người dùng vào màn chính (màn home) để hiển thị các chức năng trên giao diện màn hình

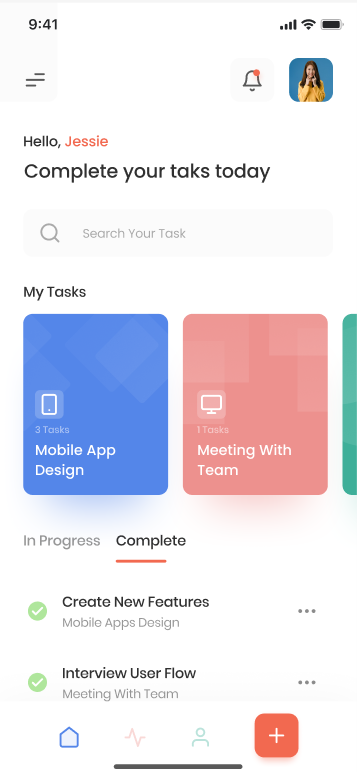
## Màn hình chính

Hình 3.1.3‑1 Giao diện màn hình chính sau khi đăng nhập



* **In Progress:** tình trạng hiện tại của công việc – đang thực hiện
* **Complete:** tình trạng hiện tại của công việc – đã hoàn thành

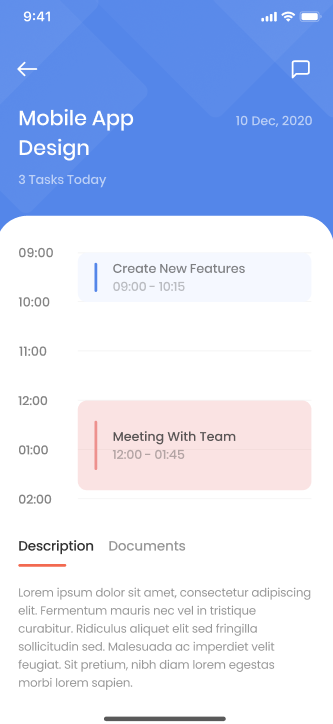
Hình 3.1.3‑2 Giao diện Complete



## Màn hình chi tiết

### Màn hình Tasks Screen

Hình 3.3.1‑1 Giao diện màn hình Tasks Screen



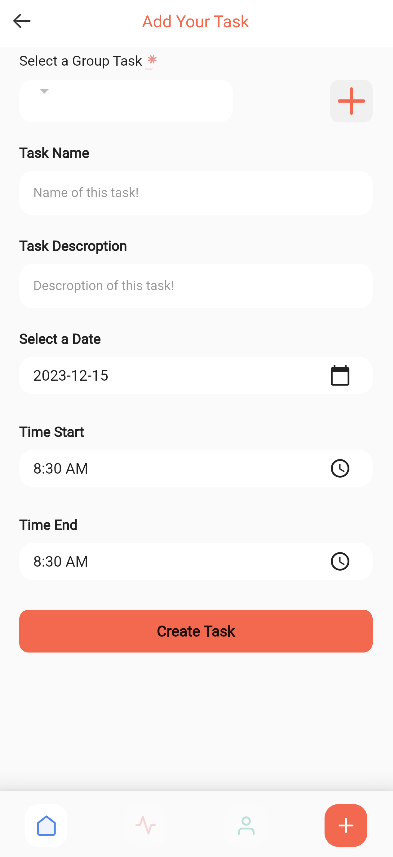
Màn hình này xuất hiện khi người dùng nhấn và ‘Mobile App Design’ trên màn hình chính.

* Description: Mô tả công việc.
* Document: Các tài liệu đính kèm của công việc.

### Màn hình Add Tasks Screen

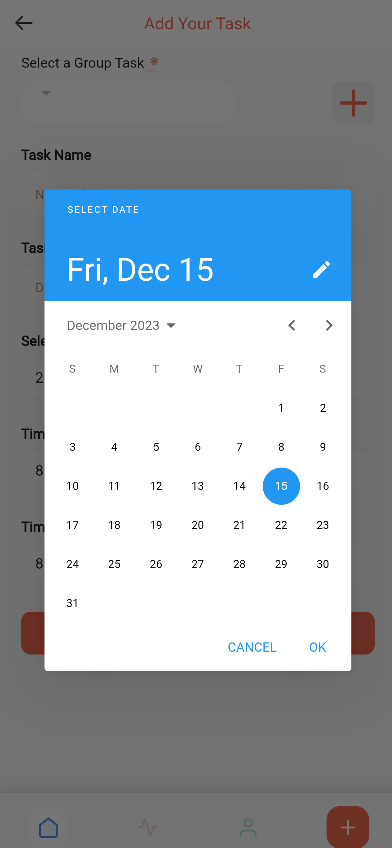
Màn hình này được hiện lên qua việc nhấn vào biểu tượng ‘+’ ở bottom Navbar

Hình 3.3.2‑1 Màn hình Add Tasks Screen



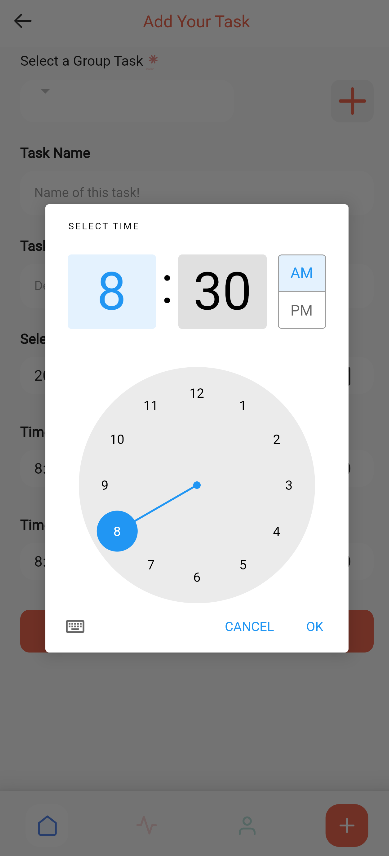
* Task Name: Tên của nhiệm vụ.
* Task Descroption: Mô tả nhiệm vụ
* Select a Date: chọn ngày làm nhiệm vụ.

Hình 3.3.2‑2 Màn hình Select a Date



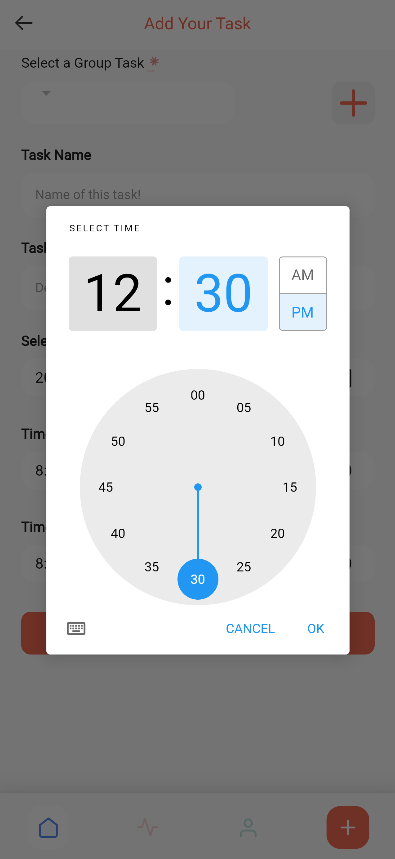
* Time Start: Thời gian bắt đầu.

Hình 3.3.2‑3 Màn hình Time Start



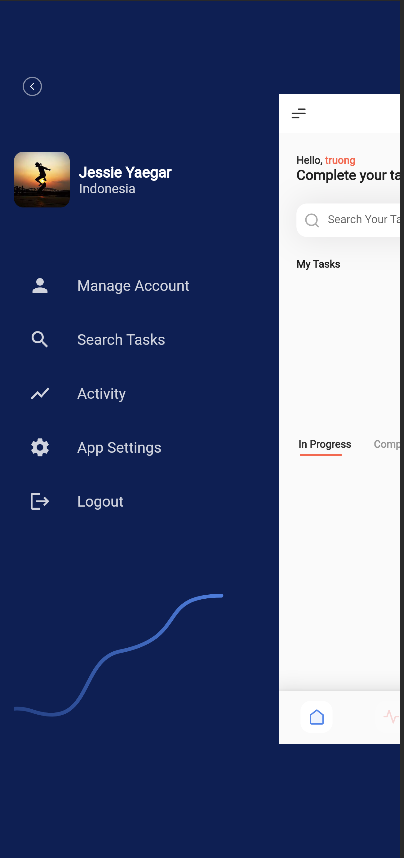
* Time End: Thời gian kết thúc.

Hình 3.3.2‑4 Màn hình Time End

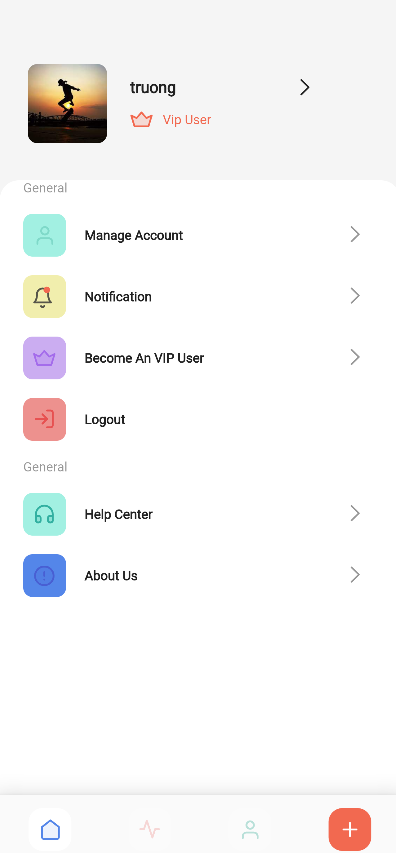


### Menu

Hình 3.3.3‑1 Menu dạng pop-up



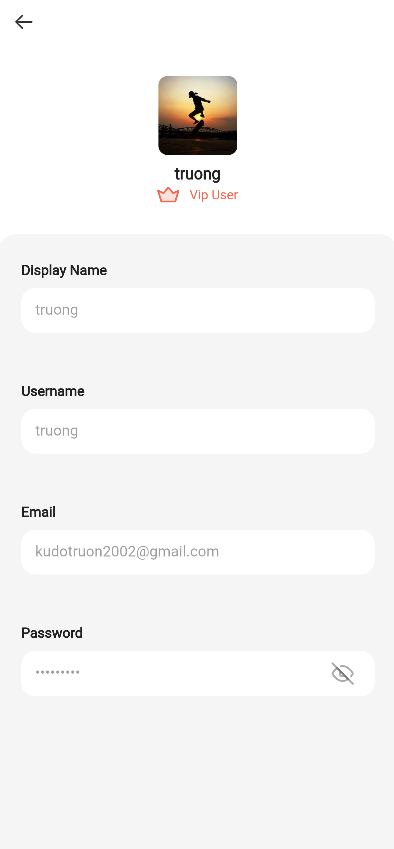
Hình 3.3.3‑2 Menu dạng danh sách



### Màn hình thông tin cá nhân

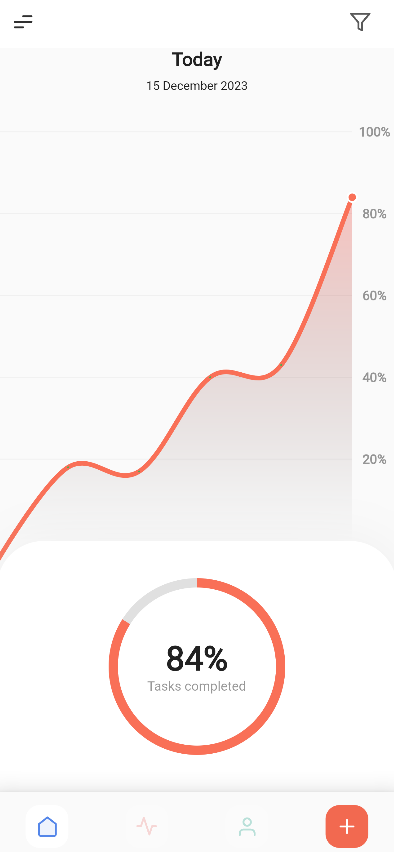
Hiển thị tất cả các thông tin của người dùng.

Hình 3.3.4‑1 Trang thông tin cá nhân người dùng



### Màn hình thống kê tiến độ

Hình 3.3.5‑1 Màn hình thống kê tiến độ



# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Sau quá trình làm việc nhóm về đề tài ứng dụng Task Management, nhóm chúng em nhận thấy đây là một đề tài thú vị để sinh viên có thể chao dồi kiến thức về lập trình đa nền tảng (ở đây là về mobile). Với đề tài này, nhóm em bước đầu đã rất khó khăn trong việc định hình công việc. Tuy nhiên, nhờ sự tận tâm của anh Hiệp (người trực tiếp giảng dạy) thì nhóm đã có những bước thực hiện ổn định hơn và có thể làm các phần cơ bản. Đây có thể chưa là kết quả mà nhóm em mong muốn, nhưng là bản demo tốt nhất nhóm em có thể tạo ra trong thời hạn và nhóm em vui vì đã có thể cùng nhau làm lên ứng dụng này.

Vì xuất phát từ mảng lập trình website nên khi tiếp xúc với các câu lệnh của Flutter, nhóm em thấy khá bối rối và khó đọc. Sau 1 thời gian tìm hiểu sâu thì đã dần quen với các cú pháp và câu lệnh. Tuy nhiên, nhóm em cũng dừng ở mức đọc hiểu một phần đoạn code. Lý do có thể là nhóm em chưa chủ động tìm hiểu sâu hơn về hướng lập trình mobile này.

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | https://vi.wikipedia.org/wiki/Flutter\_(ph%E1%BA%A7n\_m%E1%BB%81m) |