**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG QUẢN LÝ CÔNG VIỆC BẰNG FLUTTER VÀ FIREBASE**

Giảng viên hướng dẫn: **TH.S TRẦN PHONG NHÃ**

Sinh viên thực hiện: **LÊ CÔNG ĐOAN**

Lớp :  **CÔNG NGHỆ THÔNG TIN K58**

Khoá : **58**

Tp. Hồ Chí Minh, năm 2025

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG QUẢN LÝ CÔNG VIỆC BẰNG FLUTTER VÀ FIREBASE**

Giảng viên hướng dẫn: **TH.S TRẦN PHONG NHÃ**

Sinh viên thực hiện: **LÊ CÔNG ĐOAN**

Lớp : **CÔNG NGHỆ THÔNG TIN**

Khoá : **58**

Tp. Hồ Chí Minh, năm 2025

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI **CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** Độc lập – Tự do – Hạnh phúc

# NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

**Mã sinh viên:**5851071018 **Họ tên SV:** Lê Công Đoan

**Khóa:** 58 **Lớp:** CQ.58.CNTT

1. **Tên đề tài:**

* Xây dựng ứng dụng quản lý công việc bằng Flutter và Firebase

1. **Mục đích, yêu cầu:**
2. **Mục đích:**

- Xây dựng ứng dụng hổ trợ quản lý công việc, giúp cho cá nhân và tổ chức quy mô nhỏ có thể dễ dàng phân công, theo dõi và cập nhật tiến độ công việc.

- Thông qua đề tài này, em muốn vận dụng về Flutter, Dart và Firestore để tạo ra sản phẩm hoàn chỉnh hơn, rèn luyện kỹ năng làm việc với dự án thực tế, chuẩn bị cho công việc sau này.

1. **Yêu cầu:**

* Ứng dụng có các vai trò: Admin, Quản lý và Nhân viên với mỗi vai trò có quyền hạn riêng.
* Admin tạo tài khoản và phân quyền cho người dùng.
* Tạo project, phân công task cho nhân viên và theo dõi tiến độ.
* Nhân viên nhận công việc, cập nhật trạng thái task.
* Các task có thể kéo thả để thay đổi trạng thái.
* Ghi lại lịch sử thay đổi.
* Gửi thông báo đến người có liên quan khi có thay đổi công việc.

1. **Nội dung và phạm vi đề tài:**
2. **Nội dung:**

* Tìm hiểu ngôn ngữ lập trình Flutter và Dart, sử dụng Firestore lưu cơ sở dữ liệu.
* Ứng dụng vào để xây dựng ứng dụng quản lý đồ án tốt nghiệp.

1. **Phạm vi đề tài:**

* Xây dựng ứng dụng quản lý công việc

1. **Công nghệ, công cụ và ngôn ngữ lập trình**
2. **Công nghệ:**

* Flutter
* Firebase: Authentication, **Cloud Firestore,** Cloud Messaging (FCM)

1. **Công cụ:**

* **Android Studio**
* GitHub

1. **Ngôn ngữ lập trình:**

* Dart cùng với Flutter

1. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng**

* Hoàn chỉnh cuốn báo cáo đề tài.
* Xây dựng hoàn chỉnh đề tài tốt nghiệp

1. **Giáo viên và cán bộ hướng dẫn**

Họ tên: ThS Trần Phong Nhã

Đơn vị công tác: Bộ môn Công Nghệ Thông Tin – Trường Đại Học Giao Thông Vận Tải phân hiệu tại Thành Phố Hồ Chí Minh

Điện thoại: 0906761014 Email: tpnha@utc2.edu.vn

|  |  |
| --- | --- |
| **Ngày tháng 03 năm 2025**  **Trưởng BM Công nghệ Thông tin** | **Đã giao nhiệm vụ TKTN**  **Giáo viên hướng dẫn** |
| **ThS. Trần Phong Nhã** | **ThS. Trần Phong Nhã** |

Đã nhận nhiệm vụ TKTN

Sinh viên: Lê Công Đoan Ký tên:

Điện thoại: 0364602670 Email: 5851071018@st.utc2.edu.vn

**LỜI CẢM ƠN**

Trong suốt quá trình thực hiện đồ án tốt nghiệp em đã nhận được rất nhiều sự giúp đỡ và chỉ bảo tận tình, chi tiết tới từ thầy cô, gia đình và bạn bè.

Trước hết em xin gửi lời cảm ơn chân thành tới thầy Nguyễn Lê Minh, giảng viên Bộ Môn Công Nghệ Thông Tin, Trường Đại Học Giao Thông Vận Tải phân hiệu tại Thành Phố Hồ Chí Minh là người đã trực tiếp hướng dẫn và góp ý để em hoàn thiện tốt đồ án tốt nghiệp của mình.

Đồng thời em cũng chân thành cảm ơn các thầy cô Trường Đại Học Giao Thông Vận Tải phân hiệu tại Thành Phố Hồ Chí Minh nói chung, thầy cô Bộ môn Công nghệ Thông tin nói riêng đã chỉ dạy cho em những kiến thức vững vàng từ cơ sở cho tới chuyên sâu là nền tảng để em học tập tốt và phấn đấu trở thành người có ích cho xã hội.

Cuối cùng em gửi lời cảm ơn sâu sắc tới gia đình, người thân đã luôn ở bên động viên, chăm sóc, giúp đỡ và quan tâm để em hoàn thành tốt quá trình học tập tại trường.

Em xin chân thành cảm ơn !

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

|  |
| --- |
| ***Tp. Hồ Chí Minh, ngày ….… tháng ….… năm ….…***  **Giáo viên hướng dẫn**  **Trần Phong Nhã** |

**MỤC LỤC**

CHƯƠNG 1: TỔNG QUAN ………………………………………..…….………… 1

1.1. Tổng quan về nợ công…………………………………………………..……....... 1

1.1.1. Nợ công Việt Nam …………………………………………………………..… 2

**DANH MỤC CHỮ VIẾT TẮT**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mô tả** | **Ý nghĩa** | **Ghi chú** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**BẢNG BIỂU, SƠ ĐỒ, HÌNH VẼ (size 15, bold)**

*(Cách 1 tab, Time newRoman, 20, mỗi nội dung trình bày bắt đầu từ 1 trang mới)*

Bảng 1.1: (size 13)…………………………………………………………………........

Bảng 1.2: ..........................................................................................................................

Sơ đồ 1.1:………………………………………………………………………………..

Hình 1.1: ………………………………………………………………………………..

**Ghi chú:**

* + Xếp sau trang Mục lục
  + Chữ số thứ nhất chỉ tên chương
  + Chữ số thứ hai chỉ thứ tự bảng biểu, sơ đồ, hình,…trong mỗi chương
  + Ở cuối mỗi bảng biểu, sơ đồ, hình,…trong mỗi chương phải có ghi chú, giải thích, nêu rõ nguồn trích hoặc sao chụp,…

**LỜI MỞ ĐẦU**

Trong thời đại công nghệ phát triển không ngừng như hiện nay, việc tổ chức, theo dõi và quản lý công việc ngày càng đóng vai trò quan trọng, đặc biệt là trong các doanh nghiệp, tổ chức hay thậm chí là trong những nhóm làm việc nhỏ. Nếu trước đây việc giao việc hay báo cáo tiến độ chủ yếu được thực hiện thủ công, gây ra nhiều bất cập như thiếu minh bạch, khó kiểm soát tiến độ hay thông tin bị đứt gãy, thì nay, với sự hỗ trợ của công nghệ, những vấn đề đó hoàn toàn có thể được giải quyết một cách hiệu quả.

Xuất phát từ nhu cầu thực tế đó, em lựa chọn đề tài “Xây dựng ứng dụng quản lý công việc trên nền tảng di động sử dụng Flutter và Firebase” làm đồ án tốt nghiệp. Ứng dụng được xây dựng nhằm hỗ trợ người dùng trong việc phân công, theo dõi tiến độ công việc cũng như phối hợp làm việc nhóm một cách linh hoạt, mọi lúc mọi nơi.

Công nghệ được sử dụng trong đồ án là Flutter – một bộ công cụ phát triển giao diện đa nền tảng do Google phát triển, kết hợp với Firebase – nền tảng backend mạnh mẽ hỗ trợ các tính năng như xác thực người dùng, cơ sở dữ liệu thời gian thực, lưu trữ và gửi thông báo. Việc kết hợp hai công nghệ này không chỉ giúp rút ngắn thời gian phát triển mà còn đảm bảo hiệu năng và khả năng mở rộng của hệ thống.

Thông qua đồ án, em mong muốn áp dụng những kiến thức đã học vào thực tế, đồng thời rèn luyện kỹ năng lập trình, thiết kế hệ thống, giải quyết vấn đề và làm việc độc lập. Đây cũng là cơ hội để em có thêm trải nghiệm trong việc xây dựng một sản phẩm phần mềm hoàn chỉnh, từ khâu lên ý tưởng, thiết kế giao diện, xây dựng chức năng cho đến thử nghiệm và triển khai.

Nội dung đề tài gồm ác chương:

**Chương 1:** Giới thiệu

**Chương 2:** Cơ sở lí thuyết

**Chương 3:** Nội dung thực hiện

**CHƯƠNG 1. GIỚI THIỆU**

* 1. **Lí do chọn đề tài**

Trong thời đại công nghệ số hiện nay, việc quản lý công việc hiệu quả đóng vai trò quan trọng trong hoạt động của các cá nhân, nhóm làm việc và doanh nghiệp. Tuy nhiên, nhiều tổ chức và cá nhân vẫn gặp khó khăn trong việc theo dõi tiến độ, phân công nhiệm vụ và đánh giá hiệu quả làm việc.

Bên cạnh đó, xu hướng phát triển ứng dụng đa nền tảng (cross-platform) đang ngày càng phổ biến vì tính linh hoạt và tiết kiệm chi phí. Flutter, với khả năng viết một lần và chạy trên nhiều nền tảng (Android, iOS, Web), cùng với Firebase – một nền tảng Backend-as-a-Service mạnh mẽ – là một sự kết hợp lý tưởng để xây dựng các ứng dụng quản lý hiệu quả.

Chính vì những lí do nêu trên em đã quyết định chọn đề tài “Xây dựng ứng dụng quản lý công việc” để làm báo cáo tốt nghiệp giúp công việc quản lí đạt hiệu quả cao.

Thông qua đề tài này, em mong muốn tích lũy kinh nghiệm phát triển ứng dụng thực tế, từ thiết kế giao diện đến tích hợp hệ thống backend, đồng thời thể hiện năng lực tư duy và tổ chức hệ thống phần mềm.

* 1. **Mục tiêu của đề tài**

- Xây dựng ứng dụng quản lý công việc.

- Triển khai và cài đặt chương trình thực tế.

* 1. **Giới hạn và phạm vi đề tài**

**-** Tìm hiểu về công tác quản lí đồ án tốt nghiệp trong thực tế.

**-** Ứng dụng được xây dựng bằng phần mềm ANDROID STUDIO, sử dụng ngôn ngữ lập trình Flutter và cơ sở dữ liệu được xây dựng bằng FIREBASE.

* 1. **Kết quả dự kiến đạt được**

- Hoàn thành báo cáo chi tiết đồ án tốt nghiệp.

- Hoàn thành cài đặt chương trình quản lí trong thực tế**.**

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

**2.1 Lý thuyết về Flutter**

**2.1.1 Khái niệm**

Flutter là một nền tảng hỗ trợ phát triển  cho các ứng dụng đa nền cho hệ iOS và Android do tập đoàn Google phát triển. Nó được sử dụng  vô cùng phổ biến cho nhiệm vụ tạo ra các ứng dụng gốc dành cho Google.

Flutter thường bao gồm 2 thành phần chính quan trọng như sau:

- Một SDK (Software Development Kit): Đây là một bộ sưu tập bao gồm các công cụ có thể hỗ trợ cho người dùng có thể  phát triển được các ứng dụng nền của mình.

- Một Framework (UI Library based on widgets): Mỗi một tập hợp những thành phần giao diện của người dùng đều có thể thực hiện tái sử dụng vô cùng dễ dàng nên người sử dụng có thể cá nhân hóa tùy theo nhu cầu riêng của bản thân mình.

**2.1.2 Ưu điểm**

* Phát triển đa nền tảng: Chỉ cần một codebase để chạy trên Android, iOS, Web và Desktop, giúp tiết kiệm thời gian và công sức.
* Giao diện đẹp và nhất quán: Flutter sử dụng engine đồ họa Skia để vẽ toàn bộ giao diện, đảm bảo đồng nhất và mượt mà trên mọi nền tảng.
* Hiệu năng cao: Code Dart được biên dịch trực tiếp sang mã máy (AOT), giúp ứng dụng chạy nhanh và ổn định.
* Hot Reload: Cho phép cập nhật giao diện và logic tức thì mà không cần chạy lại ứng dụng, tăng tốc độ phát triển và thử nghiệm.
* Thư viện phong phú, cộng đồng mạnh: Có nhiều plugin hỗ trợ tích hợp với Firebase, camera, bản đồ,... và được hỗ trợ tốt từ cộng đồng.
* Dễ học và tiếp cận: Cú pháp Dart thân thiện, widget-based nên dễ xây dựng và tùy chỉnh giao diện.

**2.1.3** **Nhược điểm**

* Dung lượng ứng dụng lớn: Các ứng dụng Flutter thường có kích thước khởi tạo cao hơn so với ứng dụng viết bằng ngôn ngữ native, gây ảnh hưởng đến trải nghiệm người dùng trên thiết bị lưu trữ hạn chế.
* Hiệu năng trên Web và Desktop chưa tối ưu: Dù hỗ trợ đa nền tảng, nhưng hiệu suất và tính ổn định trên nền tảng Web và Desktop vẫn đang trong quá trình hoàn thiện.
* Thiếu một số plugin chuyên sâu: Với các tính năng đặc thù như Bluetooth, NFC hoặc tích hợp hệ thống phần cứng, đôi khi lập trình viên phải viết thêm mã native.
* Khó tích hợp vào hệ thống cũ: Đối với các dự án đã có nền tảng native từ trước, việc tích hợp Flutter có thể phức tạp do sự khác biệt về công nghệ và cấu trúc dự án.

**2.1.4 Tính chất**

* Mã nguồn mở (Open Source): Flutter là một dự án mã nguồn mở do Google phát triển, hoàn toàn miễn phí và được cộng đồng đóng góp mạnh mẽ.
* Đa nền tảng (Cross-platform): Cho phép phát triển ứng dụng trên nhiều nền tảng (Android, iOS, Web, Desktop) chỉ với một codebase duy nhất.
* Widget-based: Mọi thành phần trong Flutter đều là widget, giúp dễ dàng xây dựng, tùy biến và tái sử dụng giao diện.
* Hiệu năng cao (High Performance): Flutter sử dụng engine đồ họa Skia để vẽ UI trực tiếp, không phụ thuộc vào thành phần native của hệ điều hành, giúp ứng dụng mượt và ổn định.
* Tương tác thời gian thực (Real-time Interaction): Tính năng Hot Reload giúp cập nhật nhanh giao diện và logic mà không cần chạy lại toàn bộ ứng dụng.
* Dễ tích hợp với backend hiện đại: Flutter hoạt động tốt với các dịch vụ backend như Firebase, REST API, GraphQL,...
  1. **Ngôn ngữ lập trình Dart**

**2.2.1 Khái niệm**

Dart là một ngôn ngữ lập trình hiện đại, do Google phát triển, được thiết kế nhằm xây dựng các ứng dụng đa nền tảng với hiệu suất cao và dễ phát triển. Dart hỗ trợ lập trình hướng đối tượng, có kiểu dữ liệu tĩnh tùy chọn và hỗ trợ đầy đủ các tính năng giúp phát triển giao diện người dùng nhanh chóng và hiệu quả, đặc biệt khi kết hợp với framework Flutter. Dart có thể biên dịch thành mã máy gốc (native code) hoặc mã JavaScript để chạy trên trình duyệt.

**2.2.2 Tính chất**

* + Hướng đối tượng: Dart là ngôn ngữ hoàn toàn hướng đối tượng, mọi thứ đều là đối tượng kể cả kiểu dữ liệu nguyên thủy.
  + Hỗ trợ kiểu tĩnh (optional static typing): Dart cho phép sử dụng kiểu tĩnh để kiểm tra lỗi ngay khi biên dịch nhưng vẫn hỗ trợ kiểu động linh hoạt.
  + Null safety: Dart hỗ trợ tính năng an toàn null (null safety), giúp giảm lỗi phát sinh do biến null.
  + Đa nền tảng: Dart có thể chạy trên nhiều nền tảng khác nhau như mobile (Android/iOS), web, desktop (Windows, macOS, Linux) và server.
  + Hỗ trợ lập trình bất đồng bộ: Dart có các tính năng như Future, async/await để xử lý bất đồng bộ dễ dàng.
  + Biên dịch linh hoạt: Dart hỗ trợ biên dịch Ahead-of-Time (AOT) cho ứng dụng hiệu suất cao và Just-In-Time (JIT) giúp phát triển nhanh nhờ hot reload.

**2.2.3 Ưu điểm**

* + Phát triển nhanh với hot reload: Giúp lập trình viên thấy ngay kết quả thay đổi mà không phải khởi động lại ứng dụng.
  + Cú pháp rõ ràng, dễ học: Cú pháp gần gũi với Java, C# nên dễ tiếp cận cho nhiều lập trình viên.
  + Tối ưu hiệu suất: Biên dịch AOT giúp ứng dụng chạy mượt mà và nhanh.
  + Tính linh hoạt trong kiểu dữ liệu: Hỗ trợ cả kiểu tĩnh và kiểu động, thuận tiện cho cả phát triển nhanh và phát triển quy mô lớn.
  + Thư viện chuẩn phong phú: Cung cấp nhiều thư viện sẵn có hỗ trợ đa dạng nhu cầu như UI, mạng, xử lý dữ liệu,...
  + Hỗ trợ lập trình bất đồng bộ tốt: Giúp xử lý các tác vụ cần chờ như truy xuất dữ liệu, gọi API dễ dàng và hiệu quả.

**2.2.4 Nhược điểm**

* + Cộng đồng nhỏ hơn so với JavaScript hay Java: Dart chưa phổ biến rộng rãi như JavaScript hay Java, dẫn đến nguồn tài nguyên, thư viện bên thứ ba ít hơn.
  + Hạn chế khi phát triển web: Dù có thể biên dịch ra JavaScript, nhưng Dart chưa thực sự chiếm lĩnh thị trường web như JavaScript.
  + Thường gắn liền với Flutter: Dart chủ yếu được biết đến khi phát triển Flutter, ít dùng độc lập ở các lĩnh vực khác.
  + Cần học thêm khi chuyển từ các ngôn ngữ khác: Mặc dù dễ học nhưng vẫn cần thời gian làm quen với các đặc điểm riêng của Dart và Flutter.
  1. **Lý thuyết về Firebase**
     1. **Giới thiệu về Firebase**

Firebase là nền tảng phát triển ứng dụng được Google cung cấp, hỗ trợ xây dựng và vận hành các ứng dụng web và di động một cách nhanh chóng, hiệu quả. Firebase cung cấp các dịch vụ phía backend như cơ sở dữ liệu, xác thực người dùng, lưu trữ dữ liệu, hosting và các dịch vụ serverless mà không cần lập trình viên phải xây dựng hệ thống server phức tạp.

Firebase giúp rút ngắn thời gian phát triển, giảm chi phí vận hành và dễ dàng mở rộng quy mô ứng dụng.

**2.3.2 Kiến trúc và thành phần chính của Firebase**

**a.Firebase Authentication**

Firebase Authentication là dịch vụ xác thực người dùng hỗ trợ nhiều phương thức như:

* Email/Password
* Google, Facebook, GitHub
* Số điện thoại (OTP)
* Anonymous (ẩn danh)

Dịch vụ này giúp bảo mật ứng dụng và phân quyền người dùng hiệu quả. Firebase cung cấp SDK và giao diện quản lý người dùng trực quan trên Firebase Console.

b. Cloud Firestore

Cloud Firestore là cơ sở dữ liệu NoSQL dạng document, lưu trữ dữ liệu theo mô hình: Collections → Documents → Fields/Subcollections.

Đặc điểm:

* Hỗ trợ thời gian thực (Realtime).
* Cho phép truy vấn linh hoạt, lọc, phân trang.
* Dễ mở rộng và tích hợp bảo mật thông qua Security Rules.
* Có thể sử dụng offline (đặc biệt hữu ích trên mobile).

**c. Firebase Storage**

Firebase Storage hỗ trợ lưu trữ và truy xuất các file như hình ảnh, video, tài liệu,... trên nền tảng Google Cloud Storage.

Tính năng nổi bật:

* Phân quyền truy cập dựa trên Authentication.
* Tốc độ truy xuất nhanh.
* Hỗ trợ upload file từ client (mobile/web) với trạng thái tiến trình.

**d. Firebase Cloud Functions**

Cloud Functions cho phép viết mã chạy trên server mà không cần quản lý máy chủ. Các hàm có thể được gọi bởi:

* Sự kiện thay đổi trong Firestore.
* Sự kiện xác thực người dùng.
* Các HTTP Request.
* Trigger từ Firebase Storage.

Đây là công cụ hữu ích để xử lý logic nghiệp vụ, gửi email, log hệ thống, hoặc gửi thông báo đẩy (push notification).

**e. Firebase Hosting**

Dịch vụ hosting cho phép triển khai các ứng dụng web tĩnh với hiệu suất cao, bảo mật SSL miễn phí và tên miền tùy chỉnh.

**2.3.3 Ưu điểm của Firebase**

* Realtime: Cập nhật dữ liệu tức thời giữa nhiều thiết bị.
* Tích hợp dễ dàng: SDK cho Android, iOS, Flutter, Web,...
* Không cần quản lý backend: Giúp tập trung vào phát triển tính năng.
* Hệ sinh thái đầy đủ: Authentication, Database, Storage, Cloud Function,...
* Bảo mật cao: Tích hợp cơ chế phân quyền, xác thực và quy tắc truy cập.
* Dễ dàng mở rộng: Phù hợp với cả ứng dụng nhỏ và hệ thống lớn.

**2.3.4 Nhược điểm của Firebase**

* Truy vấn hạn chế: Không hỗ trợ truy vấn phức tạp như join, group by như trong SQL.
* Chi phí cao khi mở rộng: Firebase tính phí theo số lần đọc/ghi, có thể phát sinh chi phí lớn nếu không tối ưu.
* Phụ thuộc vào Google: Khó chuyển đổi sang nền tảng khác nếu muốn thay đổi công nghệ backend.
* Hạn chế xử lý dữ liệu lớn: Không phù hợp cho các hệ thống big data hoặc cần tính toán phức tạp.
* Vùng máy chủ giới hạn: Ít tùy chọn vùng địa lý, có thể ảnh hưởng đến hiệu năng với người dùng toàn cầu.

**CHƯƠNG 3: NỘI DUNG THỰC HIỆN**

**3.1 Khảo sát hệ thống**

**3.1.1 Khảo sát quy trình quản lý công việc**

Trong môi trường làm việc hiện nay, việc quản lý và phân chia công việc đóng vai trò quan trọng nhằm đảm bảo tiến độ và hiệu quả công việc. Tuy nhiên, nhiều doanh nghiệp nhỏ hoặc nhóm làm việc vẫn đang quản lý công việc qua các công cụ truyền thống như Excel, Zalo, hoặc trao đổi trực tiếp, dẫn đến:

* Khó theo dõi tiến độ công việc.
* Dễ bỏ sót nhiệm vụ.
* Thiếu tính đồng bộ giữa các thành viên.
* Không có hệ thống phân quyền và ghi nhật ký thay đổi.

Dựa trên khảo sát, quy trình làm việc tại một số nhóm làm việc thường được thực hiện như sau:

* Người quản lý thông báo công việc cho từng nhân viên qua tin nhắn hoặc gọi điện.
* Nhân viên tự theo dõi và cập nhật tiến độ, thường không được lưu trữ trên hệ thống chung.
* Các thay đổi về công việc hoặc tiến độ được thông báo miệng hoặc nhắn lại trong nhóm chat.
* Việc tổng hợp tiến độ hoặc kiểm tra hiệu quả công việc được thực hiện thủ công, mất nhiều thời gian và thiếu chính xác.

Từ thực trạng nêu trên, nhận thấy nhu cầu cấp thiết cần có một hệ thống quản lý công việc chuyên nghiệp, hỗ trợ đa nền tảng, có khả năng đồng bộ dữ liệu theo thời gian thực, phân quyền người dùng rõ ràng và có tính linh hoạt cao trong sử dụng, nên em xây dựng ứng dụng quản lý công việc

**3.1.2 Phân tích bài toán**

**3.1.2.1 Bài toán đặt ra**

Xây dựng một hệ thống quản lý công việc trên nền tảng di động và web, hỗ trợ người dùng có thể tạo, phân chia, theo dõi và cập nhật tiến độ công việc theo từng dự án, đảm bảo thông tin được đồng bộ hóa theo thời gian thực và được phân quyền rõ ràng theo vai trò người dùng. Hệ thống cần có giao diện trực quan, dễ sử dụng và phù hợp với quy mô của nhóm làm việc hoặc doanh nghiệp nhỏ.

**3.1.2.2 Mục tiêu bài toán**

* Hỗ trợ người dùng đăng nhập và phân quyền (Admin, Quản lý, Nhân viên).
* Quản lý danh sách project, mỗi project gồm nhiều task và subtask.
* Giao diện trực quan theo dạng bảng Kanban, cho phép theo dõi tiến độ dễ dàng.
* Cho phép người dùng thay đổi trạng thái công việc bằng cách kéo-thả giữa các cột.
* Ghi lại nhật ký thay đổi công việc, giúp truy vết người chỉnh sửa.
* Gửi thông báo (notification) đến người liên quan khi có sự thay đổi.
* Đồng bộ dữ liệu real-time giữa các thiết bị.
* Đảm bảo tính bảo mật và phân quyền truy cập.

**3.1.2.3 Giao diện người dùng**

* Đơn giản – trực quan: Giao diện được thiết kế tối giản, tập trung vào các chức năng chính.
* Phân quyền rõ ràng: Giao diện hiển thị theo vai trò người dùng (Admin, Quản lý, Nhân viên).
* Hiển thị theo Kanban: Công việc được trình bày theo bảng Kanban trực quan giúp dễ dàng quản lý.
* Phản hồi nhanh và realtime: Cập nhật trạng thái, thông báo thay đổi ngay tức thì.
* Đồng bộ đa nền tảng: Thiết kế giao diện phù hợp cho cả điện thoại và web.

**3.1.2.4 Các yêu cầu phi chức năng**

* Hiệu năng: Phản hồi nhanh, dữ liệu đồng bộ theo thời gian thực với Firebase Firestore.
* Bảo mật: Xác thực người dùng bằng Firebase Authentication, phân quyền rõ ràng, kiểm soát truy cập bằng Firebase Security Rules.
* Khả dụng: Hoạt động ổn định 24/7 khi có kết nối mạng, hỗ trợ nhiều thiết bị.
* Dễ sử dụng: Giao diện trực quan, thao tác đơn giản, phù hợp cho người dùng không chuyên.
* Mở rộng: Dễ tích hợp thêm tính năng (báo cáo, lịch, nhắc nhở…), đáp ứng tốt số lượng người dùng và dự án tăng cao.
* Tương thích: Hỗ trợ đa nền tảng (Android, iOS, Web) nhờ sử dụng Flutter.
* Bảo trì: Mã nguồn rõ ràng, dễ nâng cấp và sửa lỗi trong tương lai.

**3.1.2.5 Phân tích và đặc tả yêu cầu hệ thống**

Hệ thống cần đáp ứng các chức năng cơ bản sau:

* Đăng nhập, đăng xuất: Cho phép người dùng đăng nhập vào hệ thống thông qua tài khoản xác thực Firebase Authentication.
* Phân quyền người dùng: Hệ thống hỗ trợ ba vai trò: Admin, Quản lý, và Nhân viên. Mỗi vai trò có quyền hạn tương ứng.
* Quản lý tài khoản: Admin có quyền tạo mới người dùng, gán vai trò và ẩn (xóa mềm) tài khoản người dùng.
* Quản lý dự án: Admin và Quản lý có thể tạo, chỉnh sửa, và xem danh sách các dự án.
* Quản lý công việc (Task): Quản lý và Nhân viên có thể tạo công việc, cập nhật thông tin và thay đổi trạng thái công việc thông qua bảng Kanban.
* Quản lý công việc con (Subtask): Nhân viên có thể tạo subtask và cập nhật trạng thái của từng subtask.
* Ghi nhận lịch sử chỉnh sửa: Hệ thống ghi lại lịch sử chỉnh sửa của task, bao gồm: thời gian, người chỉnh sửa, và nội dung thay đổi.
* Thông báo: Khi có thay đổi liên quan đến người dùng, hệ thống gửi thông báo thông qua Firebase Cloud Messaging.
* Giao diện Kanban: Các công việc trong dự án được hiển thị dưới dạng bảng Kanban với các cột trạng thái: To Do, Doing, Done, Complete.
* Tìm kiếm, lọc dữ liệu: Người dùng có thể tìm kiếm và lọc công việc theo trạng thái, người thực hiện hoặc dự án.

**3.1.3 Biểu đồ phân rã chức năng**

A diagram of a company

AI-generated content may be incorrect.

**Hình 3.1 Biểu đồ phân rã chức năng của hệ thống**

**3.1.4 Biểu đồ Use Case**

**3.1.4.1 Biểu đồ Use Case tổng quát**

**3.1.4.2 Biểu đồ Use Case đăng nhập**

**A screen shot of a computer screen

AI-generated content may be incorrect.**

**Hình 3.3 Biểu đồ Use Case đăng nhập**

* Mô tả: Use Case này mô tả quá trình người dùng đăng nhập vào hệ thống bằng tài khoản đã được cấp. Sau khi xác thực thông tin, người dùng sẽ được chuyển đến giao diện phù hợp với vai trò của mình (Admin, Quản lý, Nhân viên).
* Actor: Người dùng
* Tiền điều kiện: Người dùng đã được tạo tài khoản và có thông tin đăng nhập hợp lệ.
* Luồng chính:
  + Người dùng mở ứng dụng và chọn chức năng đăng nhập.
  + Nhập số điện thoại và mật khẩu.
  + Hệ thống kiểm tra thông tin đăng nhập qua Firebase Authentication.
  + Nếu thông tin hợp lệ, người dùng được chuyển đến màn hình chính.
* Luồng phụ:
  + Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.
* Hậu điều kiện: Người dùng được đăng nhập thành công vào hệ thống và sử dụng các chức năng theo phân quyền.

**3.1.4.3 Biểu đồ Use Case quản lý dự án**

Tác nhân: Admin  
Mô tả: Use Case cho phép xem, thêm, sửa, xóa người dùng và quản lý người thực hiện trong các dự án.  
Điều kiện trước: Admin đã đăng nhập vào hệ thống.

Dòng sự kiện chính:  
Người sử dụng chọn kiểu tác động: thêm, sửa, xóa dự án.

* Thêm dự án: Chọn “Thêm dự án”, hệ thống hiển thị giao diện nhập thông tin và danh sách dự án. Người dùng nhập thông tin và chọn người thực hiện. Nếu thành công, hệ thống thông báo và lưu dữ liệu. Nếu sai, thực hiện dòng rẽ nhánh A1.
* Sửa dự án: Hệ thống hiển thị danh sách dự án. Chọn dự án cần sửa, nhập thông tin cần thay đổi. Nếu thành công, hệ thống thông báo và lưu dữ liệu. Nếu sai, thực hiện dòng rẽ nhánh A1.
* Xóa dự án: Hệ thống hiển thị danh sách dự án. Chọn dự án cần xóa. Nếu thành công, hệ thống thông báo và cập nhật trạng thái xóa mềm. Nếu sai, thực hiện dòng rẽ nhánh A1.

Dòng rẽ nhánh A1: Hệ thống thông báo dữ liệu không hợp lệ, yêu cầu nhập lại, quay lại bước 1 dòng chính.

Hậu điều kiện: Thông tin dự án được cập nhật trong cơ sở dữ liệu.

**3.1.4.4 Biểu đồ Use Case quản lý công việc (task)**

Tác nhân: Quản lý

Mô tả: Use Case cho phép xem, thêm, sửa, xóa, phân công và cập nhật trạng thái công việc trong hệ thống.

Điều kiện trước: Quản lý đã đăng nhập vào hệ thống và đã được phân công vào dự án.

Dòng sự kiện chính:

Người sử dụng chọn kiểu tác động: thêm, sửa, xóa, phân công, cập nhật trạng thái công việc.

* Thêm công việc: Chọn “Thêm công việc”, hệ thống hiển thị giao diện nhập thông tin và danh sách công việc. Người dùng nhập thông tin. Nếu thành công, hệ thống thông báo và lưu dữ liệu. Nếu sai, thực hiện dòng rẽ nhánh A1.
* Sửa công việc: Hệ thống hiển thị danh sách công việc. Chọn công việc cần sửa, nhập thông tin cần thay đổi. Nếu thành công, hệ thống thông báo và lưu dữ liệu. Nếu sai, thực hiện dòng rẽ nhánh A1.
* Xóa công việc: Hệ thống hiển thị danh sách công việc. Chọn công việc cần xóa. Nếu thành công, hệ thống thông báo và cập nhật trạng thái xóa mềm. Nếu sai, thực hiện dòng rẽ nhánh A1.
* Phân công công việc: Chọn công việc, chọn nhân viên từ danh sách, gán cho công việc. Nếu thành công, hệ thống thông báo và lưu dữ liệu. Nếu sai, thực hiện dòng rẽ nhánh A1.
* Cập nhật trạng thái: Chọn công việc cần cập nhật, thay đổi trạng thái (To Do, Doing, Done, Complete). Nếu thành công, hệ thống thông báo và lưu dữ liệu. Nếu sai, thực hiện dòng rẽ nhánh A1.

Dòng rẽ nhánh A1:

Hệ thống thông báo dữ liệu không hợp lệ, yêu cầu nhập lại, quay lại bước 1 dòng chính.

Hậu điều kiện:

Thông tin công việc được cập nhật trong cơ sở dữ liệu, trạng thái và phân công phản ánh đúng trên hệ thống.

**3.1.4.5 Biểu đồ Use Case quản lý subtask**

Tác nhân: Nhân viên

Mô tả: Use Case cho phép xem, tạo và cập nhật trạng thái subtask thuộc công việc được phân công.

Điều kiện trước: Nhân viên đã đăng nhập và được phân công vào công việc.

Dòng sự kiện chính:

Người sử dụng chọn kiểu tác động: tạo mới, xem chi tiết, cập nhật trạng thái subtask.

* Tạo subtask: Chọn công việc đang làm, chọn “Tạo subtask”, hệ thống hiển thị giao diện nhập thông tin. Người dùng nhập tên subtask, mô tả. Trạng thái mặc định là “Bàn giao”. Nếu thành công, hệ thống thông báo và lưu dữ liệu. Nếu sai, thực hiện dòng rẽ nhánh A1.
* Xem subtask: Hệ thống hiển thị danh sách các subtask thuộc công việc. Chọn subtask để xem thông tin chi tiết.
* Cập nhật trạng thái subtask: Chọn subtask, chọn trạng thái mới từ danh sách có sẵn (Đang làm, Hoàn thành, Tạm dừng, Hủy bỏ). Nếu thành công, hệ thống thông báo và lưu dữ liệu. Nếu sai, thực hiện dòng rẽ nhánh A1.

Dòng rẽ nhánh A1: Hệ thống thông báo dữ liệu không hợp lệ, yêu cầu nhập lại, quay lại bước 1 dòng chính.

Hậu điều kiện:Thông tin subtask được cập nhật trong cơ sở dữ liệu; trạng thái hiển thị đúng với tiến độ thực tế.

**3.1.4.6 Biểu đồ Use Case quản lý thông báo và lịch sử**

* Tác nhân: Quản lý / Nhân viên
* Mô tả: Use Case cho phép người dùng xem danh sách thông báo và lịch sử chỉnh sửa liên quan đến các công việc được phân công.
* Điều kiện trước: Người dùng đã đăng nhập vào hệ thống.

Dòng sự kiện chính:

Người dùng chọn tab “Thông báo” hoặc “Lịch sử”.

* Xem thông báo:  
  Hệ thống hiển thị danh sách thông báo (theo assigneeId), gồm:
  + Nội dung thông báo
  + Thời gian
  + Trạng thái (chưa đọc / đã đọc)

Người dùng có thể đánh dấu là đã đọc hoặc xóa thông báo. Nếu thành công, hệ thống cập nhật trạng thái. Nếu lỗi, thực hiện dòng rẽ nhánh A1.

* Xem lịch sử chỉnh sửa:  
  Hệ thống hiển thị log các chỉnh sửa công việc liên quan, bao gồm:
  + Tên công việc
  + Người chỉnh sửa
  + Thời gian
  + Nội dung thay đổi
  + Trạng thái trước và sau

Dòng rẽ nhánh A1:

Nếu không thể tải dữ liệu, hệ thống hiển thị lỗi và cho phép người dùng thử lại.

Hậu điều kiện:

Người dùng đã nắm được các thông báo mới và lịch sử thay đổi liên quan đến công việc của họ.

**3.1.4.7 Biểu đồ Use Case giao diện người dùng**

* Tác nhân: Admin
* Mô tả: Use Case cho phép xem, thêm, sửa, xóa (ẩn), và gán vai trò cho người dùng trong hệ thống.
* Điều kiện trước: Admin đã đăng nhập vào hệ thống.

Dòng sự kiện chính:

Người sử dụng chọn kiểu tác động: thêm, sửa, xóa (ẩn), gán vai trò người dùng.

* Thêm người dùng: Chọn “Thêm người dùng”, hệ thống hiển thị giao diện nhập thông tin và danh sách người dùng. Người sử dụng nhập thông tin (họ tên, số điện thoại, email, v.v.). Nếu thành công, hệ thống thông báo và lưu dữ liệu. Nếu sai, thực hiện dòng rẽ nhánh A1.
* Sửa người dùng: Hệ thống hiển thị danh sách người dùng. Chọn người cần sửa, nhập các thông tin cần thay đổi. Nếu thành công, hệ thống thông báo và lưu dữ liệu. Nếu sai, thực hiện dòng rẽ nhánh A1.
* Xóa (ẩn) người dùng: Hệ thống hiển thị danh sách người dùng. Chọn người cần ẩn, hệ thống đánh dấu là “đã xóa” (xóa mềm). Nếu thành công, hệ thống thông báo. Nếu sai, thực hiện dòng rẽ nhánh A1.
* Gán vai trò: Chọn người dùng, chọn vai trò (Quản lý / Nhân viên). Nếu gán thành công, hệ thống thông báo và cập nhật dữ liệu. Nếu sai, thực hiện dòng rẽ nhánh A1.

Dòng rẽ nhánh A1:

Hệ thống thông báo dữ liệu không hợp lệ, yêu cầu nhập lại, quay lại bước 1 dòng chính.

Hậu điều kiện:

Thông tin người dùng được cập nhật trong cơ sở dữ liệu và hiển thị chính xác trên hệ thống.

**3.1.5 Biểu đồ tuần tự**

**3.1.5.1 Biểu đồ tuần tự chức năng đăng nhập**

**A diagram of a diagram

AI-generated content may be incorrect.**

**Hình Biểu đồ tuần tự chức năng đăng nhập**

**3.1.5.2 biểu đồ tuần tự chức năng quản lý dự án**

**3.1.5.3 Biểu đồ tuần tự chức năng quản lý task**

**3.1.5.4 Biểu đồ tuần tự chức năng quản lý subtask**

**3.1.5.5 Biểu đồ tuần tự chức năng quản lý thông báo và lịch sử**

**3.1.6 Thiết kế cơ sở dữ liệu**

**3.1.6.1**

**3.2 Thiết kế và xây dựng ứng dụng**

**3.3 Kết luận**

**3.3.1 Nhận xét**

Trong quá trình thực hiện đồ án, em đã có cơ hội tiếp cận và áp dụng các công nghệ hiện đại như Flutter và Firebase để xây dựng một ứng dụng quản lý công việc đa nền tảng. Hệ thống được thiết kế với các chức năng phân quyền rõ ràng (Admin, Quản lý, Nhân viên), cho phép theo dõi, phân chia và cập nhật tiến độ công việc một cách trực quan và tiện lợi.

Bên cạnh việc trau dồi kỹ năng lập trình, đồ án cũng giúp em hiểu rõ hơn về quy trình xây dựng một ứng dụng thực tế: từ thiết kế giao diện, tổ chức dữ liệu trong Firestore, cho đến xử lý logic nghiệp vụ và triển khai các chức năng đồng bộ, realtime.

Tuy còn một số điểm cần tối ưu và hoàn thiện thêm, nhưng qua đồ án này, em đã rút ra được nhiều kinh nghiệm quan trọng trong việc phát triển ứng dụng mobile/web, cũng như khả năng tư duy hệ thống và làm việc theo hướng sản phẩm.

**3.3.2 Ưu điểm**

Sau khi xây dựng được ứng dụng quản lí công việc em đã đạt được một số ưu điểm sau:

* Ứng dụng hỗ trợ realtime, đa nền tảng (mobile, web) nhờ Flutter và Firebase.
* Phân quyền rõ ràng giữa Admin, Quản lý và Nhân viên.
* Giao diện trực quan, dễ sử dụng, có thể kéo-thả task giữa các trạng thái.
* Dữ liệu được đồng bộ tức thời và bảo mật qua Firebase Rules.
* Hệ thống thông báo và log giúp theo dõi thay đổi công việc hiệu quả.

**3.3.3 Hạn chế**

Bên cạnh những kết quả khả quan đã đạt được, em nhận thấy còn tồn tại một số hạn chế như sau:

* Chưa hỗ trợ hoạt động tốt khi mất kết nối mạng (offline).
* UI còn đơn giản, thiếu tuỳ biến (chưa có dark mode, đa ngôn ngữ).
* Chưa có kiểm thử tự động và triển khai CI/CD.
* Một số thao tác xử lý logic vẫn phụ thuộc phía client.
* Khả năng mở rộng lớn cần tối ưu thêm nếu triển khai thực tế.

**3.3.4 Hướng phát triển**

Sau khi hoàn thành đề tài và xây dựng được “ứng dụng quản lý công việc” em sẽ tiếp tục nghiên cứu và phát triển ứng dụng này nhằm tăng các tính năng và tối ưu hóa tốc độ xử lý để đem lại hiệu quả cao hơn.

* Cải tiến chức năng cũ
* Thêm một số tính năng mới như thông báo đẩy khi có task mới, bị giao việc hoặc cập nhật trạng thái ngay cả khi không mở ứng dụng; đính kèm file tài liệu, hình ảnh; hiển thị công việc dạng lịch, giúp quản lý theo ngày/tuần/tháng.

**PHỤ LỤC**

* + Link sourcecode: https://github.com/congdoan99/DATN-3-2025
  + Tên WebSite: https://lecongdoan.io.vn/
  + Tài khoản admin: Tên đăng nhập “doan@gmail.com”, mật khẩu “123456”
  + Tài khoản người dùng: Tên đăng nhập “poro@gmail.com”, mật khẩu “654321”

**TÀI LIỆU THAM KHẢO**

1. Google. Flutter Documentation. Truy cập tại: https://docs.flutter.dev
2. Google. Firebase Documentation. Truy cập tại: https://firebase.google.com/docs
3. FlutterFire. Official Firebase plugins for Flutter. Truy cập tại: https://firebase.flutter.dev
4. Reso Coder. Flutter Firebase tutorials (YouTube). Truy cập tại: <https://www.youtube.com/c/ResoCoder>
5. Medium. Building Real-Time Apps with Flutter & Firestore. Truy cập tại: <https://medium.com> (từ khóa: Flutter Firestore real-time)
6. Stack Overflow. Hỏi đáp về Flutter và Firebase. Truy cập từ: <https://stackoverflow.com>