

**ĐỀ ÁN CUỐI KÌ**

MÔN

**LẬP TRÌNH THIẾT BỊ DI ĐỘNG**

Đề tài:

**TaskCV – Quản lý công việc cá nhân**

A picture containing sign

Description automatically generated

GVHD: **Th.s Lê Huỳnh Phước**

Sinh viên thực hiện: **Lê Trần Bảo Sương**

Lớp: **11DHPM**

MSSV: **1731103052**

1. Danh sách nhiệm vụ

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Nhiệm vụ |
| 1 | Mai Trúc Lâm | 1731103030 | Xây dựng cơ sở dữ liệu  Word, PowerPoint  Thuyết trình |
| 2 | Lê Trần Bảo Sương | 1731103052 | Thiết kế, xây dựng, lập trình ứng dụng Android. |

1. Lời mở đầu

Hiện tại có một thực trạng rất đáng quan ngại, đó chính là các bạn sinh viên và đôi khi là những người đi làm mắc phải. Đó chính là vấn đề quản lý thời gian và công việc. Dễ thấy nhất là ở những bạn sinh viên, vốn thời gian của các bạn nhàng rỗi rất lớn, vì thế mà nhiều bạn chọn cách đi làm thêm. Và việc học của các bạn đôi khi còn bị gác qua một bên để nhường cho thời gian đi làm, cho đến khi các bài kiểm tra ập tới, thi giữa kì, cuối kì. Thì các bạn lại chạy “deadline” trong sự hỗn loạn và vô vọng, từ đó dẫn đến việc sản phẩm các bạn làm ra thường có chất lượng thấp vậy nên điểm số của các bạn sẽ rất thấp. Còn ở người đi làm, khi họ không thể sắp xếp thời gian và công việc thì mọi thứ còn trở nên tồi tệ hơn, khi một dự án lớn bị một vài cá nhân làm kéo dài thời gian, thì chi phí khi bồi thường hợp đồng do trễ hạn giao sản phẩm là không hề nhỏ.

Vậy nên một giải pháp mang tính kịp thời đáp ứng được những nhu cầu đó là vô cùng cần thiết. Và cũng chính từ điều đó mà nhóm chúng tôi đã thực hiện dự án TaskCV – Ứng dụng quản lý công việc.

Một ứng dụng đa nền tảng có thể giúp các cá nhân có thể ghi chú công việc ngay trên điện thoại và sắp xếp mức độ quan trọng của công việc để hiện thị trên màn hình; giúp thiết lập thời gian của công việc, đính kèm thêm các tài liệu âm thanh, tệp, … và đến thời gian nhất định, khi người dùng yêu cầu thì phần mềm sẽ xuất ra báo cáo về tiến độ hoàn thành các công việc để giúp họ có thể lên kế hoạch một cách chính xác hơn.

Kế hoạch của chúng tôi hướng đến trong tương lai là phát triển cả một hệ thống giúp các công ty vừa và nhỏ có thể quản lý dự án của họ một cách dễ dàng hơn, và giúp cả những người quản lý có thể dễ dàng nắm bắt và theo sát được tình hình của dự án. Và xuất ra những báo cáo để những lãnh đạo đưa ra những quyết định kịp thời khi có công việc bị tắt ngẽn tiến độ, và xử lý nó.

1. Cơ sở lý thuyết
2. Lập trình Android là gì?

Lập trình Android là nghề sử dụng các ngôn ngữ lập trình để viết và phát triển các phần mềm. Nhằm gia tăng tiện ích cho thiết bị di động, đặc biệt là di động sử dụng hệ điều hành.

Với sự phát triển của di động thông mình như hiện nay thì lập trình Android quả là một ngành rất hot. Theo thống kê, hiện nay Android chính là hệ điều hành được sử dụng nhiều nhất. Và nếu như bạn đang lo lắng về đối thủ lớn nhất của Android là iOS thì bạn đừng lo lắng. Bởi mỗi nền tảng đem lại cho người dung những trải nghiệm riêng. Nếu như iOS đem lại cho người quá nhiều khó khăn với độ bảo mật cao. Thì Android lại đặc biệt được người dùng yêu thích sử dụng vì tính năng dễ tùy biến và đa dạng hơn. Chính vì vậy mà hiện nay nhu cầu tuyển dụng lập trình viên mảng Android tăng cao.

1. Ngôn ngữ sử dụng

* Flutter được phát triển nhằm giải quyết bài toán thường gặp trong mobile là Fast Development và Native Performance. Nếu như React Native chỉ đảm bảo Fast Development và code native thuần chỉ đảm bảo Native Performance thì Flutter làm được cả 2 điều trên.
* Sử dụng DART. Một ngôn ngữ lập trình hướng đối tượng do Google phát triển. DART là một static type language nên nó là AOT (Ahead of Time), compile xong hết rồi mới chạy. Trong khi đó nó cũng là JIT (Just in Time) giống như các dynamic type language. Khi dev thì nó sử dụng JIT để hỗ trợ Hot Load và build release thì dùng AOT để tối ưu hiệu năng như một native code bình thường
* Ngoài ra DART cũng hướng tới việc trở thành một ngôn ngữ có thể chạy trên nhiều platform khác nhau, nó cũng có máy ảo (VM) làm nhiệm vụ dịch source code sang bytecode như Java. Hiện tại DART có thể transpile ra các ngôn ngữ khác như JS để chạy trên trình duyệt. Cái này không liên quan mobile lắm tuy nhiên đây cũng là 1 ưu điểm lớn của ngôn ngữ này nên không thể bỏ qua.
* Vậy còn native module ? Khác với JS Bridge, Flutter “nói chuyện” với các native module bằng chính các native interface. Mặc dù vẫn được gọi là “bridge”, tuy nhiên nó nhanh hơn rất nhiều và gân như không bị “thắc cổ chai” như React Native. Ngoài ra các module này được kiến trúc theo “plugin”, các module viết cho Flutter phải tuân thủ các rule trong này. Cá nhân mình thấy viết native module cho Flutter rất tự nhiên, không cần phải học các syntax macro C/C++ như RN.

1. Cấu trúc của một chương trình dùng Flutter
2. Một chương trình Flutter gồm các phần sau:

* Khai báo Namespace
* Một class
* Phương thức của lớp
* Thuộc tính của lớp
* Một phương thức Main
* Lệnh và biểu thức
* Comment

1. Cấu trúc của lớp

**Constructors** – Hàm thiết lập của lớp

* Tự động thực hiện khi tạo đối tượng của lớp
* Có quyền public
* Có tên trùng với tên lớp
* Không có giá trị trả về
* Có thể định nghĩa các tham số
* Có thể định nghĩa nhiều hơn một hàm thiết lập trong lớp
* Một hàm thiết lập có thể gọi một hàm thiết lập khác với
* từ khoá this.
* Tạo đối tượng dựa trên hàm thiết lập của lớp

**Hàm thiết lập ngầm định**

* Nếu một lớp không định nghĩa hàm thiết lập, chương
* trình dịch tự sinh ra ham thiết lập ngầm định
* Nếu có một hàm thiết lập được khai báo, hàm thiết lập
* ngầm định không được sinh ra.

**Destructors** - hàm huỷ bỏ

* Tự động thực hiện khi giải phóng đối tượng khỏi bộ nhớ
* Không có từ khoá public hoặc private
* Tên hàm bắt đầu bằng dấu ~ và tiếp đến là tên lớp
* Mỗi lớp chỉ có một hàm huỷ bỏ

1. Một số khái niệm

**Đối tượng (object)**

Đối tượng là sự đóng gói của dữ liệu và phương thức (Object = Data + Method)

Dữ liệu: Mô tả đối tượng

Phương thức: Các hàm xử lý dữ liệu của đối

tượng

Truy nhập các thành phần của đối tượng

public: Truy nhập được ở mọi nơi.

private: Chỉ truy nhập được trong định nghĩa lớp

Đối tượng phải được tạo bằng từ khoá new

1. Giới thiệu Android Studio
2. **Android Studio** là cung cấp giao diện để tạo các ứng dụng và xử lý phần lớn các công cụ quản lý file phức tạp đằng sau hậu trường. Ngôn ngữ lập trình được sử dụng ở đây là Java và được cài đặt riêng trên thiết bị của bạn. Android Studio rất đơn giản, bạn chỉ cần viết, chỉnh sửa và lưu các dự án của mình và các file trong dự án đó. Đồng thời, Android Studio sẽ cấp quyền truy cập vào Android SDK.
3. Giới thiệu các thành phần chính trong giao diện:

* **XML** là viết tắt của Ngôn ngữ đánh dấu có thể mở rộng. XML là một ngôn ngữ đánh dấu giống như HTML được sử dụng để mô tả dữ liệu. Các thẻ XML không được xác định trước trong XML. Chúng ta phải xác định các Thẻ của riêng mình. Bản thân Xml cũng có thể đọc được bằng cả con người và máy móc. Ngoài ra, nó có thể mở rộng và đơn giản để phát triển. Trong Android, sử dụng xml để thiết kế bố cục của mình vì xml là ngôn ngữ nhẹ nên nó không làm cho bố cục nặng nề.
* **Navigation:** Một tài liệu được mở trong một tab; Dùng để tiết kiệm không gian khi lập trình
* **Menu:** Chứa các lệnh dùng để xây dựng và thực hiện chương trình
* **ToolBar:** Chứa các biểu tượng dùng để thực hiện các lệnh

1. Về cơ sở dữ liệu Firebase:

1. Firebase Authentication:

- Hầu hết các ứng dụng cần biết danh tính của người dùng. Việc biết danh tính của người dùng cho phép ứng dụng lưu dữ liệu người dùng một cách an toàn trên đám mây và cung cấp trải nghiệm được cá nhân hóa giống nhau trên tất cả các thiết bị của người dùng.

- Xác thực Firebase cung cấp các dịch vụ phụ trợ, SDK dễ sử dụng và thư viện giao diện người dùng được tạo sẵn để xác thực người dùng với ứng dụng của bạn. Nó hỗ trợ xác thực bằng mật khẩu, số điện thoại, các nhà cung cấp danh tính liên hợp phổ biến như Google, Facebook và Twitter, v.v.

- Phần mềm sử dụng Email và mật khẩu để Xác thực người dùng bằng địa chỉ email và mật khẩu của họ. SDK xác thực Firebase cung cấp các phương pháp để tạo và quản lý người dùng sử dụng địa chỉ email và mật khẩu của họ để đăng nhập. Xác thực Firebase cũng xử lý việc gửi email đặt lại mật khẩu.

2. Firebase Realtime Database:

- Lưu trữ và đồng bộ dữ liệu với cơ sở dữ liệu đám mây NoSQL. Dữ liệu được đồng bộ hóa trên tất cả các ứng dụng khách trong thời gian thực và vẫn có sẵn khi ứng dụng của bạn ngoại tuyến.

- Cơ sở dữ liệu thời gian thực của Firebase là cơ sở dữ liệu được lưu trữ trên đám mây. Dữ liệu được lưu trữ dưới dạng JSON và được đồng bộ hóa trong thời gian thực cho mọi máy khách được kết nối. Khi bạn tạo ứng dụng đa nền tảng với SDK iOS, Android và JavaScript của chúng tôi, tất cả khách hàng của bạn đều chia sẻ một phiên bản Cơ sở dữ liệu thời gian thực và tự động nhận các bản cập nhật với dữ liệu mới nhất.

- Thay vì các yêu cầu HTTP thông thường, Cơ sở dữ liệu thời gian thực của Firebase sử dụng đồng bộ hóa dữ liệu — mỗi khi dữ liệu thay đổi, mọi thiết bị được kết nối sẽ nhận được bản cập nhật đó trong vòng mili giây. Cung cấp trải nghiệm hợp tác và nhập vai mà không cần suy nghĩ về mã mạng.

3. Firebase Cloud Storage:

- Cloud Storage cho Firebase là dịch vụ lưu trữ đối tượng mạnh mẽ, đơn giản và tiết kiệm chi phí được xây dựng cho quy mô của Google. SDK Firebase cho Bộ nhớ đám mây bổ sung tính năng bảo mật của Google vào các tệp tải lên và tải xuống cho ứng dụng Firebase của bạn, bất kể chất lượng mạng. Có thể sử dụng để lưu trữ hình ảnh, âm thanh, video hoặc nội dung khác do người dùng tạo. Trên máy chủ, bạn có thể sử dụng Google Cloud Storage, để truy cập các tệp giống nhau.

1. Phân tích hệ thống
2. Mục đích phát triển phần mềm

Phần mềm sẽ giúp cho doanh nghiệp có thể chủ động hơn, dễ dàng thao tác và kiểm tra tiến độ dự án. Hạn chế tình trạng doanh nghiệp bị đình trệ chỉ vì một vài khâu trong một dự án bị chậm trễ tiến độ.

Giúp cho nhân viên dễ dàng làm việc hơn dù ở bất cứ nơi đâu, cũng như hạn chế tiếp xúc trong những trường hợp nhất định.

1. Mục tiêu

Hoàn thành việc xây dựng ứng dụng trên Android và IOS.

Cải thiện quy trình đánh giá tiến độ từ đó doanh nghiệp có thể đưa ra được những mục tiêu, tiến độ deadline phù hợp hơn.

Cải thiện được năng suất làm việc của nhân viên, mọi hoạt động đều dựa trên sự chủ động của mỗi cá nhân, vì thế mà tình trạng đùn đẩy, ùn ứ sẽ được giảm bớt.

1. Yêu cầu hệ thống:
2. Yêu cầu chung:

TaskCV trên tất cả các nền tảng cần hoạt động trơn tru và ít lỗi nhất có thể. Đặc biệt hạn chế các lỗi liên quan đến bảo mật thông tin.

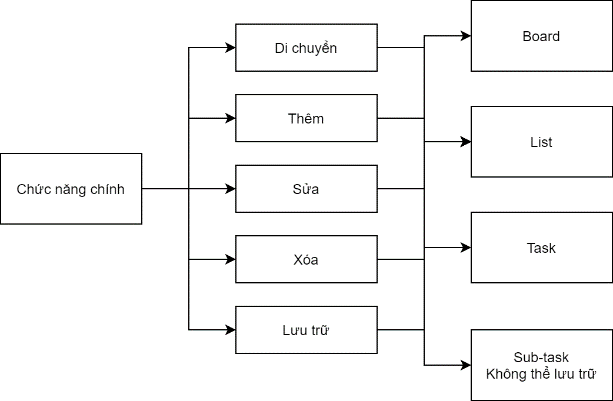
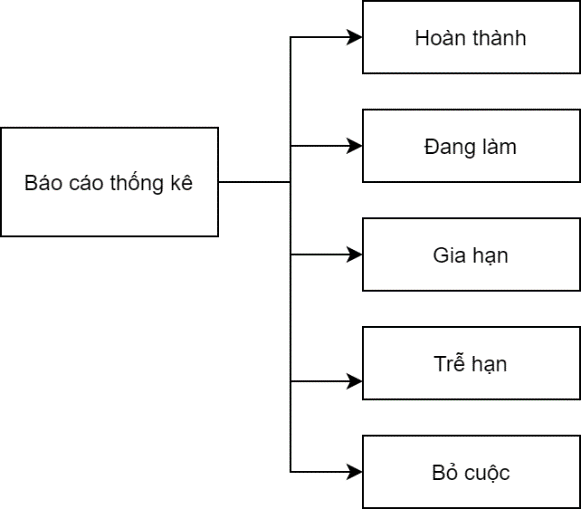
Thao tác và giao diện dễ dàng, đơn giản nhất để một người mới lần đầu sử dụng có thể thành thạo những thao tác cơ bản nhất trong vòng một giờ đồng hồ.

Chất lượng hình ảnh ổn định khi sử dụng TaskCV Meet.

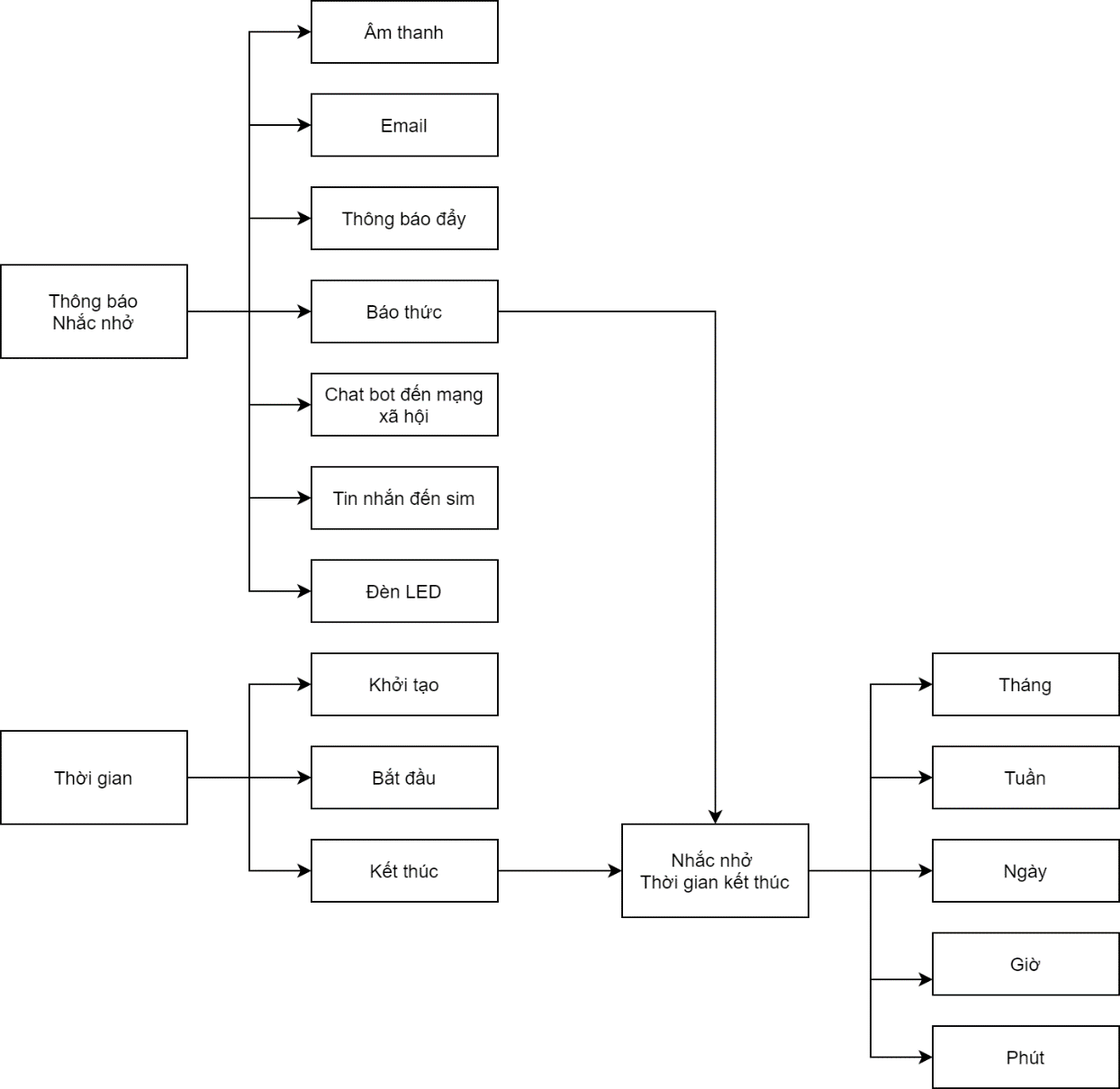
1. Yêu cầu tối thiểu:

* Có thể thao tác nhập dữ liệu trên phần mềm.
* Có hiển thị thông báo theo cài đặt của người dùng.
* Có thể xuất báo cáo theo yêu cầu.
* Quá trình vận hành tạo ra ít lỗi nhất, ít treo máy.

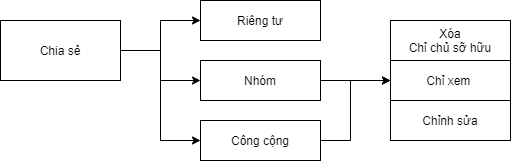
1. Sơ đồ chức năng hệ thống
2. Chức năng thao tác cơ bản với dữ liệu



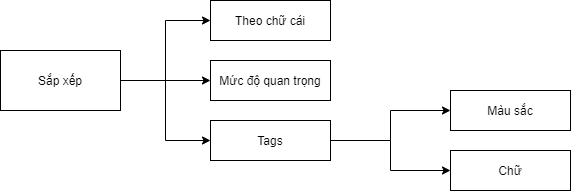
1. Thông báo



1. Xuất báo cáo
2. Chia sẻ



1. Sắp xếp



1. Chi tiết đề tài
2. Class Diagram

A screenshot of a cell phone

Description automatically generated

1. Use Case Diagram

A close up of a map

Description automatically generated

1. Giao diện ứng dụng



Figure 1Logo icon

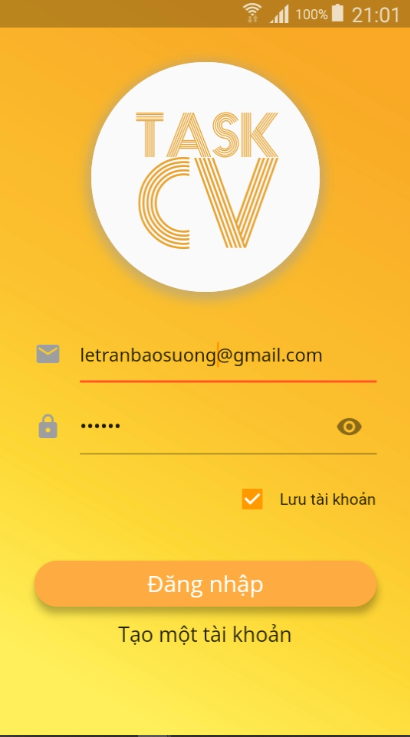


Figure 2Màn hình đăng nhập

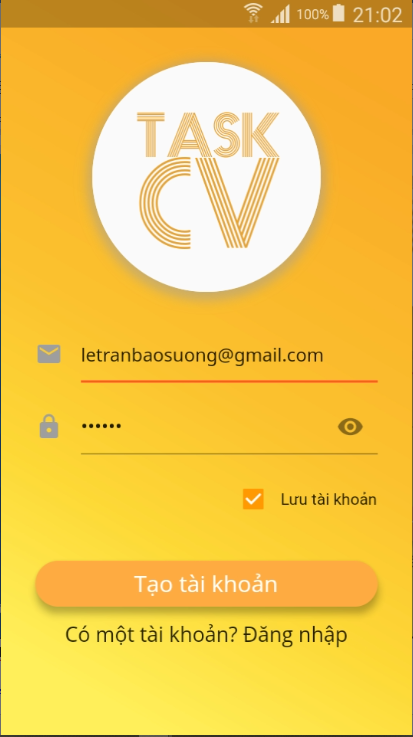


Figure 3Màn hình đăng ký

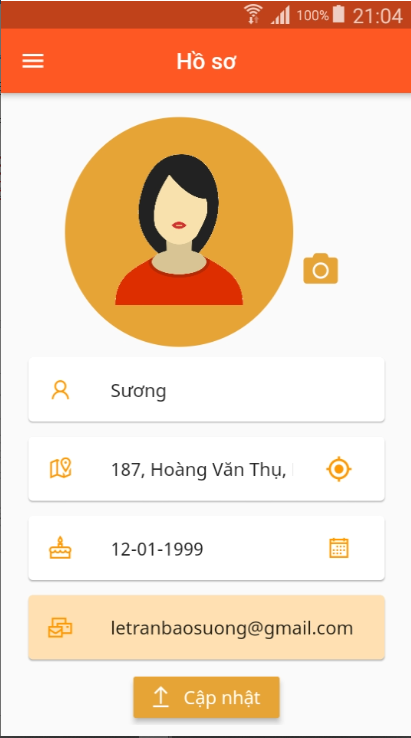


Figure 4Màn hình hồ sơ



Figure 5Drawer

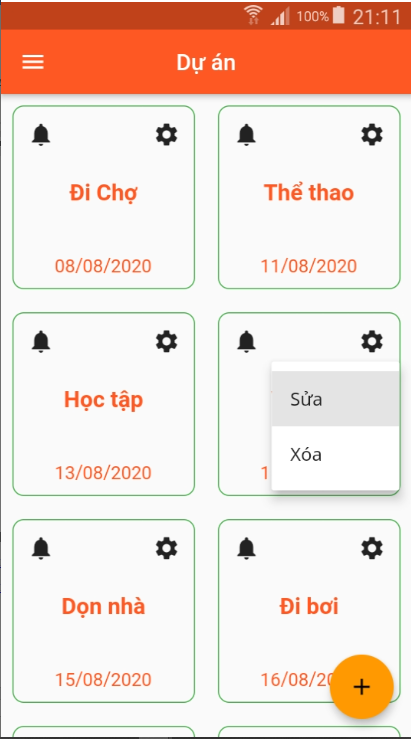


Figure 6Màn hình dự án

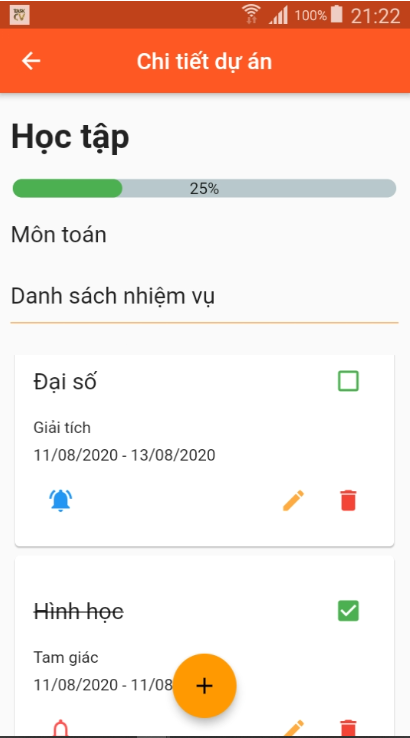


Figure 7Màn hình chi tiết dự án

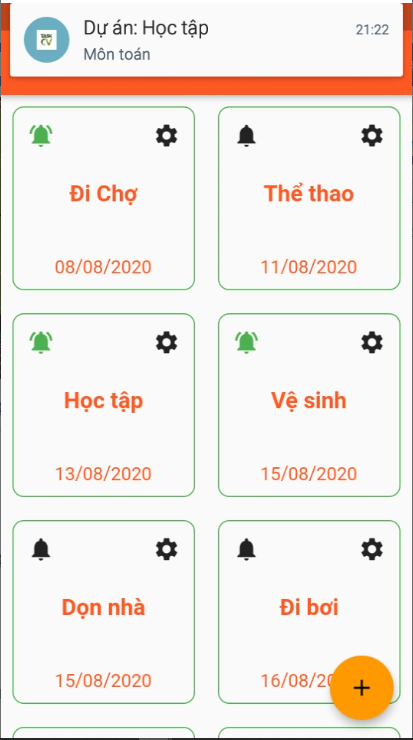


Figure 8Notification



Figure 9Notifications trên thanh trạng thái

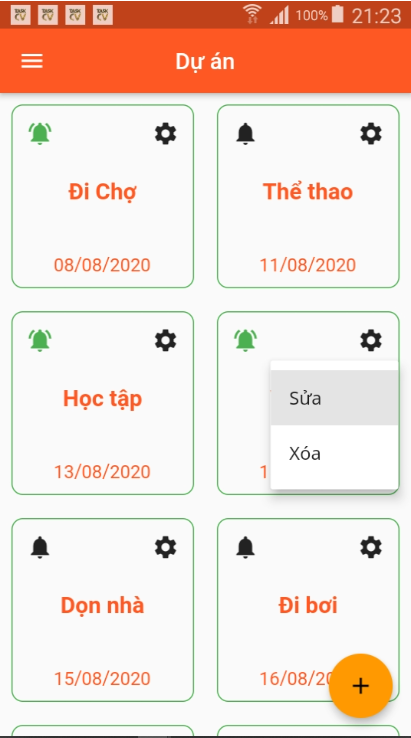


Figure 10Chức năng màn hình dự án

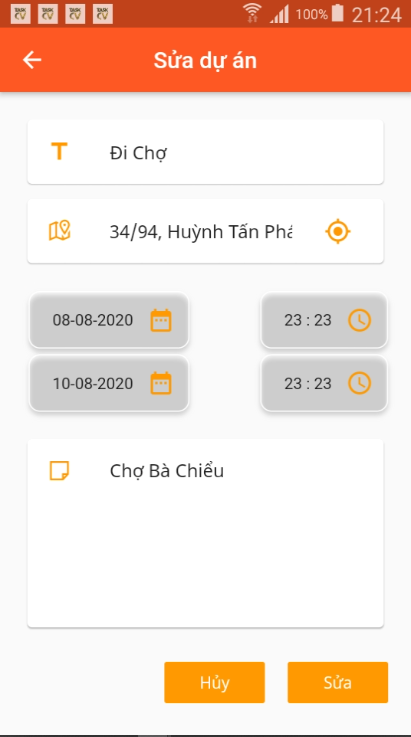


Figure 11Màn hình chỉnh sửa dự án

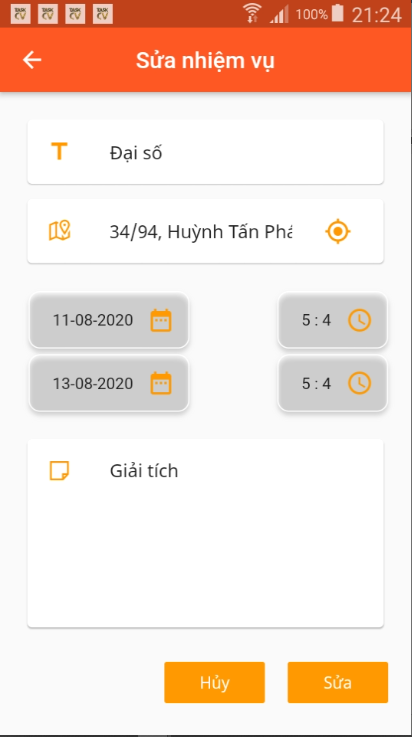


Figure 12Màn hình chỉnh sửa nhiệm vụ

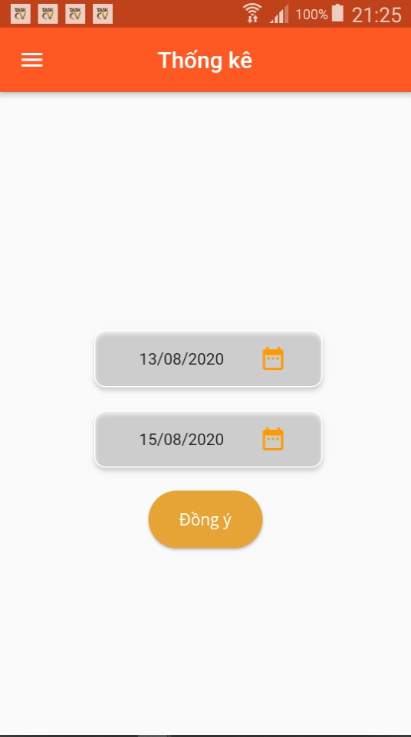


Figure 13Thống kê theo thời gian

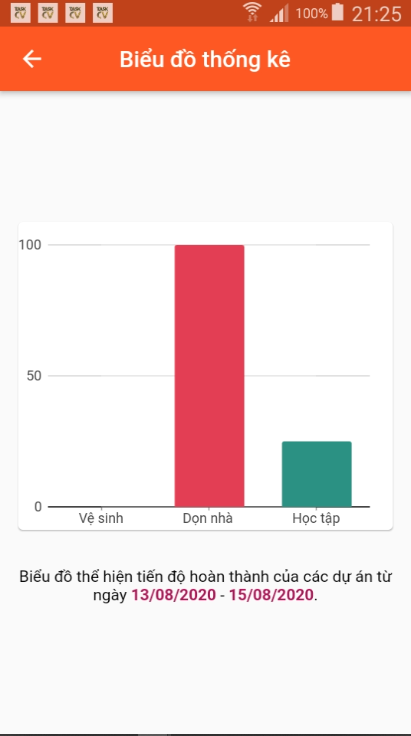


Figure 14Biểu đồ thống kê theo thời gian



Figure 15Màn hình thông tin trợ giúp

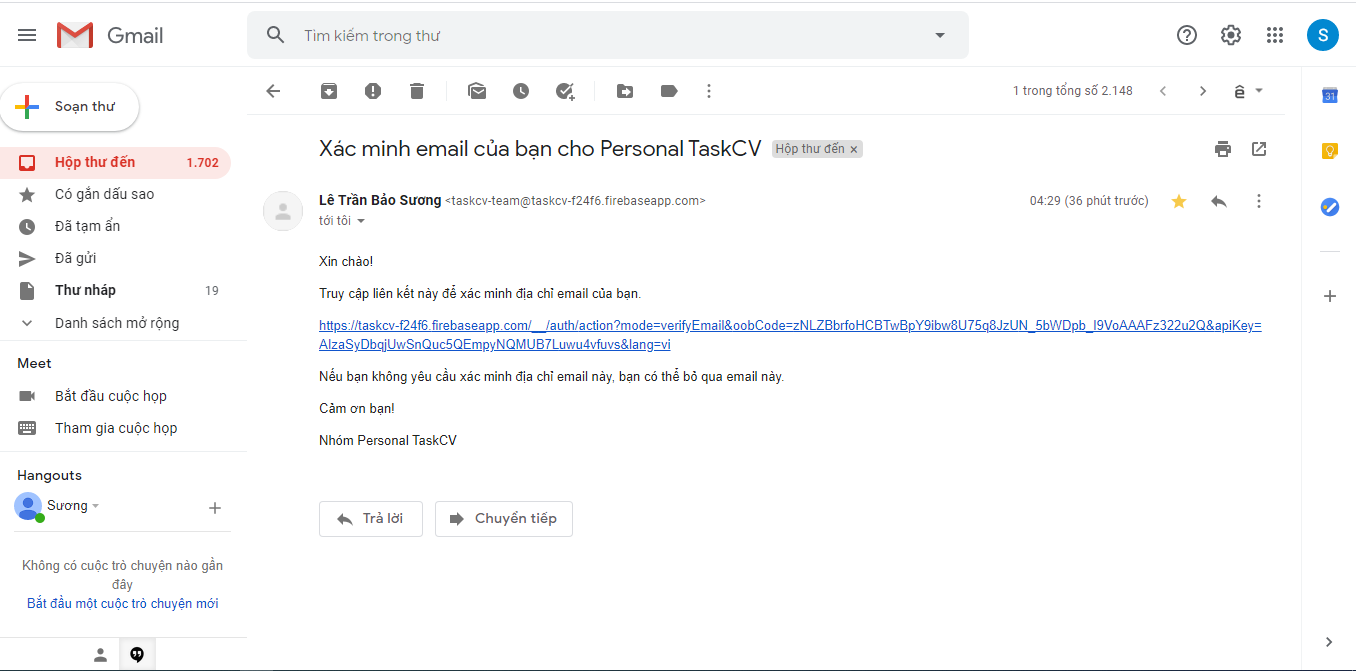


Figure Xác thực tài khoản

1. Tài liệu tham khảo

<https://topdev.vn/blog/nhung-tinh-nang-co-ban-cua-firebase/>

<https://vi.wikipedia.org/wiki/Android_Studio>

<https://google.fandom.com/wiki/Flutter>

<https://codegym.vn/blog/2018/12/22/lap-trinh-android-la-gi/#:~:text=L%E1%BA%ADp%20tr%C3%ACnh%20Android%20l%C3%A0%20ngh%E1%BB%81,s%E1%BB%AD%20d%E1%BB%A5ng%20h%E1%BB%87%20%C4%91i%E1%BB%81u%20h%C3%A0nh>.