

**LUẬN VĂN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**XÂY DỰNG ỨNG DỤNG DI ĐỘNG SCANNER TRÊN FRAMEWORK IONIC**

**Sinh viên thực hiện: Nguyễn Phan Hoàng Oanh**

**MSSV: B1401075**

**Khóa: 40**



**LUẬN VĂN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**XÂY DỰNG ỨNG DỤNG DI ĐỘNG SCANNER TRÊN FRAMEWORK IONIC**

**Giáo viên hướng dẫn: Sinh viên thực hiện:**

**Ts. Ngô Bá Hùng Nguyễn Phan Hoàng Oanh**

**MSSV: B1401075**

**Khóa: 40**

**NHẬN XÉT CỦA GIẢNG VIÊN**

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

..................................................................................................................................

**Tên giảng viên nhận xét**

# LỜI CẢM ƠN

Trong suốt thời gian theo học tại Trường Đại Học Cần Thơ em đã được các thầy cô chỉ bảo tận tình, nhận được nhiều sự giúp đỡ từ các anh chị khóa trước cũng như các bạn cùng lớp. Đó không chỉ là những kiến thức chuyên môn mà còn là những chia sẻ về kinh nghiệm, kỹ năng học tập và kỹ năng sống. Tất cả những điều đó sẽ là hành trang quý báu, là nền tảng góp phần giúp em sẵn sàng đón nhận những thử thách trong môi trường mới đầy cạnh tranh, trong công việc và cuộc sống ngoài xã hội sắp tới.

Em xin gửi lời cảm ơn chân thành đến thầy Ngô Bá Hùng, người đã tận tình hướng dẫn, giúp đỡ em trong quá trình thực hiện đề tài.

Em xin cám ơn quý thầy cô Khoa Công Nghệ Thông Tin và Truyền Thông trường Đại Học Cần Thơ đã sẵn sàng giúp đỡ em trong suốt thời gian em theo học tại trường.

Xin gửi lời cám ơn đến bạn bè, các anh, các chị đã tận tình giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài.

Trong thời gian qua, mặc dù đã em rất cố gắng để hoàn thành đề tài luận văn, nhưng chắc chắn sẽ không thể tránh khỏi những thiếu sót. Vì vậy em rất mong nhận được sự chỉ bảo, ý kiến từ quý thầy cô và các bạn.

Cần Thơ, ngày 30 tháng 10 năm 2017

Chân thành cảm ơn

Nguyễn Phan Hoàng Oanh

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc529694081)

[MỤC LỤC 2](#_Toc529694082)

[DANH MỤC BẢNG 4](#_Toc529694083)

[DANH MỤC HÌNH 5](#_Toc529694084)

[DANH MỤC TỪ VIẾT TẮT 6](#_Toc529694085)

[TÓM TẮT 7](#_Toc529694086)

[ABSTRACT 8](#_Toc529694087)

[CHƯƠNG 1: TỔNG QUAN 9](#_Toc529694088)

[**1.1** **Đặt vấn đề** 9](#_Toc529694089)

[**1.2** **Phạm vi đề tài** 9](#_Toc529694090)

[**1.3** **Phương pháp nghiên cứu** 9](#_Toc529694091)

[**1.4** **Bố cục luận văn** 10](#_Toc529694092)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 11](#_Toc529694093)

[**2.1** **ANGULARJS** 11](#_Toc529694095)

[**2.2** **TypeScript** 11](#_Toc529694096)

[**2.3** **NODE.JS** 12](#_Toc529694097)

[**2.4** **CORDOVA** 13](#_Toc529694098)

[**2.5** **FRAMEWORK IONIC** 14](#_Toc529694099)

[**2.6** **FIREBASE** 20](#_Toc529694100)

[**2.7** **OPENCV** 26](#_Toc529694101)

[**2.8** **Git và Github** 32](#_Toc529694102)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 36](#_Toc529694103)

[**3.1** **Các yêu cầu và chức năng cần có** 36](#_Toc529694105)

[**3.2** **Mô hình chức năng, đặc tả chức năng** 36](#_Toc529694106)

[**3.3** **Mô hình dữ liệu** 58](#_Toc529694107)

[CHƯƠNG 4: MÔ TẢ GIAO DIỆN 59](#_Toc529694108)

[**4.1** **Giao diện** 59](#_Toc529694109)

[**4.2** **Giao diện** 59](#_Toc529694110)

[**4.3** **Giao diện** 59](#_Toc529694111)

[CHƯƠNG 5: TỔNG KẾT 60](#_Toc529694112)

[THAM KHẢO 61](#_Toc529694113)

# DANH MỤC BẢNG

[Bảng 1. Cơ sở dữ liệu 58](#_Toc529777759)

[Bảng 2. Thư mục 59](#_Toc529777760)

[Bảng 3. Hình ảnh 59](#_Toc529777761)

# DANH MỤC HÌNH

Hình 1. Tải Git cho Windows 18

Hình 2. Menu Item của Git sau khi cài đặt xong 18

Hình 3. Tải Node.js phiên bản LTS 22

Hình 4. Command Prompt của Node.js 22

Hình 5. Cài đặt Ionic và Cordova CLI 22

Hình 6. Khởi tạo ứng dụng Ionic 23

Hình 7. Chạy thử ứng dụng trên trình duyệt 24

# DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| HyperText Markup Language | HTML |
| JavaScript | JS |
| Cascading Style Sheets | CSS |
| Cơ sở dữ liệu | CSDL |
| Node Package Manager | NPM |
| Open Computer Vision | OpenCV |
| User Interface | UI |
| Application Programming Interface | API |
| Global Positioning System | GPS |
| Command Line Interface | CLI |
| Distributed Version Control System | DVCS |
| Software development kit | SDK |
| Java Development Kit | JDK |
| Graphical User Interface | GUI |

# TÓM TẮT

Ứng dụng Scanner là một ứng dụng được xây dựng cho điện thoại di động dựa trên framework Ionic dùng để quét ảnh chứng minh thư, hộ chiếu hoặc tài liệu. Ứng dụng có chức năng chụp ảnh cần quét, sau đó ảnh sẽ được trải qua các bước xử lý như cắt, xoay trái, xoay phải, chỉnh sáng bằng thư viện xử lý ảnh OpenCV để thu được một bức ảnh như khi dùng máy quét và lưu vào bộ nhớ của điện thoại. Từ kết quả trên, người dùng có thể sử dụng ứng dụng để chia sẻ tệp dưới định dạng ảnh hoặc PDF.

# ABSTRACT

The Scanner application is an application built for mobile phones based on the Ionic framework used to scan ID cards, passports or documents. The application has the function to capture the image to be processed, then the image will be undergo the processing steps such as crop, rotate left, rotate right, adjust the brightness by using the image processing library OpenCV to obtain an image as when using the scanner and save it to phone’s memory. From the above results, users can use the application to share tệps in image or PDF format.

# CHƯƠNG 1: TỔNG QUAN

## Đặt vấn đề

Đồng hành với sự phát triển không ngừng của thế giới, số lượng các loại tài liệu, văn bản cũng tăng lên nhanh chóng với một tốc độ đáng kể. Để tiện cho việc quản lý, chỉnh sửa, lưu trữ cũng như chia sẻ tài liệu một cách dễ dàng nhất không cần tốn nhiều thời gian của người sử dụng, chúng ta sử dụng scanner (máy quét) để chuyển các loại tài liệu từ dạng vật lý sang dữ liệu số. Nhưng scanner vẫn còn hạn chế ở điểm chỉ có thể sử dụng được tại một địa điểm nhất định, nếu đầu tư nhiều scanner để thuận tiện cho việc sử dụng thì chi phí bỏ ra lại quá cao.

Ở thời đại công nghệ phát triển như hiện nay, các thiết bị công nghệ di động nói chung và điển hình là điện thoại di động đang ngày càng được ưa chuộng bởi những lợi ích mà nó mang đến. Vậy tại sao chúng ta không tạo ra một ứng dụng scanner có thể sử dụng cho trên điện thoại di động?

Với một chiếc điện thoại di động được cài đặt ứng dụng scanner, việc quét ảnh ở bất kỳ đâu và vào bất kỳ lúc nào chưa bao giờ là chuyện dễ dàng đến vậy. Không những thế, sự ra đời của ứng dụng scanner trên điện thoại cũng giúp tiết kiệm chi phí hơn khi phải đầu tư vào những máy scanner.

Từ đó, ý tưởng xây dựng một ứng dụng di động scanner đã được hình thành và phát triển thành kết quả của luận văn này.

## Phạm vi đề tài

Ứng dụng di động Scanner có khả năng phục vụ cho tất cả mọi người có nhu cầu quét, xử lý, quản lý và lưu trữ tài liệu ở bất cứ đâu và bất kỳ thời điểm nào chỉ cần bạn sở hữu một chiếc điện thoại di động trong tay. Khi điện thoại được kết nối mạng, ứng dụng còn có khả năng chia sẻ ảnh đến những mạng xã hội hoặc bất kỳ ứng dụng nào dùng để liên lạc đang có trong điện thoại.

## Phương pháp nghiên cứu

### Về lý thuyết

* + - Tìm hiểu phân tích, thiết kế hệ thống thông tin.
    - Phương pháp phân tích thiết kế cơ sỡ dữ liệu.
    - Nắm vững các ngôn ngữ HTML, CSS, JavaScript.
    - Tìm hiểu về các kiến thức TypeScript, Angular, Cordova, NodeJS.
    - Các kiến thức nền tảng cho lập trình web và ứng dụng di động lai.
    - Nắm vững kiến thức về Ionic Framework.

### Về kỹ thuật

* + - Cài đặt Node để sử dụng Cordova và Ionic Framework.
    - Sử dụng Ionic Framework để tạo cấu trúc và phát triển ứng dụng.
    - Sử dụng và phát triển ứng dụng di động với Ionic Framework dựa trên ngôn ngữ lập trình web và các Component sẵn có của nó.
    - Tạo các chức năng cần thiết từ các API và Native của Ionic: Camera, Tệp, SQLite, SocialSharing,...

## Bố cục luận văn

Chương 1 - Tổng quan: Giới thiệu chung về đề tài, lý do chọn đề tài, phạm vi của đề tài và phương pháp, kĩ thuật cần tìm hiểu để xây dựng được ứng dụng.

Chương 2 - Cơ sở lý thuyết: Giới thiệu các thuật ngữ, kỹ thuật, Framework, API và Native được sử dụng trong ứng dụng.

Chương 3 - Phân tích thiết kế hệ thống: Kết quả của quá trình phân tích, thiết kế hệ thống bao gồm: mô hình use case, đặc tả chức năng, mô hình CDM, PDM và mô tả các bảng dữ liệu.

Chương 4 – Mô tả giao diện: Giới thiệu và mô tả các giao diện chính của ứng dụng khi hoạt động.

Chương 5 - Tổng kết: Kết quả đạt được, các kinh nghiệm, ưu, nhược điểm và hướng phát triển của đề tài.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



## ANGULARJS

AngularJS là một một dự án mã nguồn mở được phát triển đầu tiên bởi Miško Hevery, một nhân viên của Google. Hevery bắt đầu nghiên cứu và phát triển dự án vào năm 2009 và phiên bản 1.0 được cho ra mắt vào năm 2012. Do sự hữu ích của dự án này nên Google quyết định trở thành công ty chính thức đứng đằng sau hỗ trợ cho sự phát triển của AngularJS. Phiên bản mới nhất của AngularJS tính đến thời điểm này là 7.0.3.

AngularJS là một framework được viết bằng JavaScript có cấu trúc cho các ứng dụng web động. Nó cho phép sử dụng HTML như là ngôn ngữ mẫu và cho phép mở rộng cú pháp của HTML để diễn đạt các thành phần ứng dụng một cách rõ ràng và súc tích. Hai tính năng cốt lõi: Data binding và Dependency injection của AngularJS loại bỏ phần lớn code thường phải viết. Nó xuất hiện trong tất cả các trình duyệt, làm cho nó trở thành đối tác lý tưởng của bất kỳ công nghệ Server nào.

Ngoài ra, AngularJs còn có các thành phần bổ sung cho Cordova, framework thường được sử dụng để viết các ứng dụng di động. Nó nhằm mục đích để đơn giản hóa cả việc phát triển và [thử nghiệm](https://vi.wikipedia.org/wiki/Ki%E1%BB%83m_th%E1%BB%AD_ph%E1%BA%A7n_m%E1%BB%81m) của các ứng dụng bằng cách cung cấp một framework với kiến trúc cho client side đó là MVC (Model View Controller) và MVVM (Model – View – ViewModel).

Ưu điểm của AngularJS:

* + - Cho phép tạo ra các ứng dụng một cách đơn giản, code sạch.
    - Sử dụng data binding giống .NET với tính năng liên kết với HTML nên giúp người dùng cảm thấy dễ sử dụng.
    - Có thể chạy trên hầu hết các trình duyệt điện thoại thông minh.
    - Cung cấp khả năng tái sử dụng các component.
    - Code dễ dàng khi unit test (kiểm tra từng đơn vị nhỏ như thuộc tính, sự kiện, thủ tục và hàm).

## TypeScript

Typescript là một dự án mã nguồn mở được Microsoft phát triển,  nó có thể được coi là một phiên bản nâng cao của JavaScript bởi việc bổ sung tùy chọn kiểu tĩnh và lớp hướng đối tượng mà điều này không có ở JavaScript. TypeScript có thể sử dụng để phát triển các ứng dụng chạy ở client-side (Angular2) và server-side (NodeJS). TypeScript là một ngôn ngữ giúp cung cấp quy mô lớn hơn so với JavaScript.

Typescript đang được sử dụng ở các framework front-end phổ biến như Angular 2, Ionic... cũng như nền tảng cho back-end như Node.js bởi những ưu điểm của nó.

Ưu điểm của Typescript:

* + - Dễ dàng hơn trong việc phát triển các dự án lớn, được hỗ trợ bởi các Framework JavaScript lớn.
    - Hỗ trợ hướng đối tượng mạnh: Hầu hết các cú pháp hướng đối tượng đều được hỗ trợ bởi Typescript như kế thừa, đóng gói, constructor, abstract, interface, implement, override...v.v
    - Cách tổ chức code rõ ràng hơn bởi được hỗ trợ các kỹ thuật mới nhất và hỗ trợ lập trình hướng đối tượng: Hỗ trợ cơ chế giúp kiến trúc hệ thống code hướng module, hỗ trợ namespace, giúp xây dựng các hệ thống lớn nơi mà nhiều lập trình viên có thể làm việc cùng nhau một cách dễ dàng hơn.
    - Hỗ trợ các tính năng mới nhất của JavaScript.
    - Một lợi thế của Typescript nữa là mã nguồn mở vì vậy nó miễn phí và có cộng đồng hỗ trợ rất lớn.

## NODE.JS

### Giới thiệu Node.js

Node.js được tạo bởi Ryan Dahl từ năm 2009 và phát triển dưới sự bảo trợ của [Joyent](https://vi.wikipedia.org/w/index.php?title=Joyent&action=edit&redlink=1). Mục tiêu ban đầu của Dahl là làm cho trang web có [khả năng push](https://vi.wikipedia.org/w/index.php?title=C%C3%B4ng_ngh%E1%BB%87_push&action=edit&redlink=1) như trong một số ứng dụng web như [Gmail](https://vi.wikipedia.org/wiki/Gmail). Sau khi thử với vài ngôn ngữ Dahl chọn JavaScript vì một API Nhập/Xuất không đầy đủ. Điều này cho phép anh có thể định nghĩa một quy ước Nhập/Xuất điểu khiển theo sự kiện, non-blocking.

Node.js là một [hệ thống phần mềm](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng_ph%E1%BA%A7n_m%E1%BB%81m) được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là cho phép lập trình JavaScript phía server. Trước đây JavaScript chỉ được coi là một ngôn ngữ đơn giản tạo hiệu ứng hoạt hình, tương tác phía trình duyệt. Cho đến khi Google Chrome sử dụng V8 engine - một bộ dịch – để thực thi mã JavaScript nhanh nhất thế giới tính đến hiện nay. V8 được viết bằng C/C++, mã nguồn mở, cho phép bên thứ 3 toàn quyền sử dụng, triển khai trên các hệ điều hành khác nhau: Windows, Linux, Unix, MacOSX. Node.js được xây dựng trên V8 và thư viện LibUV cho phép lệnh thực thi hiện tại không ngăn các lệnh kế tiếp chạy luôn: cơ chế Non-Blocking I/O.

### Giới thiệu NPM

NPM (Node Package Manager) là một công cụ (chương trình) quản lý các thư viện lập trình JavaScript cho Node.js, được tích hợp sẵn trong Node.js, vì vậy khi cài đặt xong Node.js chúng ta sẽ có thể sử dụng được cả NPM.

## CORDOVA

Apache Cordova (trước kia được gọi là PhoneGap) là một framework mã nguồn mở phát triển ứng dụng di động nổi tiếng được tạo ra bởi Nitobi. Adobe Systems đã mua Nitobi vào năm 2011, đổi tên nó thành PhoneGap, và sau đó đã phát hành một phiên bản mã nguồn mở được gọi là Apache Cordova. Cordova là một nền tảng để xây dựng những ứng dụng di động lai (hybrid mobile applications) sử dụng HTML, CSS và JavaScript. Những ứng dụng này có thể chạy trên nhiều nền tảng khác nhau như Android, iOS, Windows Phone…. Ứng dụng lai đem lại nhiều lợi thế như khả năng hiển thị nội dung trên tất cả các thiết bị di động, cũng như tận dụng tối đa các tính năng khác của thiết bị di động như GPS, camera, danh sách liên lạc,… Bên cạnh đó, thời gian và chi phí dùng để tạo nên một ứng dụng lai cũng thấp hơn so với các ứng dụng di động thông thường.

Cordova bao gồm một tập hợp các API thiết bị cho phép người lập trình di động truy cập, sử dụng các chức năng native của thiết bị như là camera hay cảm biến gia tốc bằng JavaScript. Kết hợp với một bộ khung phát triển giao diện như jQuery Mobile, Dojo Mobile hoặc Ionic, nó cho phép có thể phát triển ứng dụng di động chỉ dựa trên HTML, CSS và JavaScript. Sử dụng Cordova sẽ nhanh hơn rất nhiều so với phát triển các ứng dụng thuần Android hay iOS. Chỉ cần sử dụng JavaScript khi làm việc với Cordova nên ta không cần thiết phải biết ngôn ngữ lập trình của mỗi loại hệ điều hành.

Khi sử dụng Cordova API, một ứng dụng có thể được xây dựng mà không phải sử dụng bất kỳ một đoạn code native nào. Thay vào đó, công nghệ web sẽ được sử dụng, và chúng sẽ được tổ chức trên chính ứng dụng đấy chứ không cần thông qua bất kỳ server nào. Và bởi vì những API JavaScript này là thống nhất trên tất cả các nền tảng thiết bị và được xây dựng trên chuẩn web nên những ứng dụng được viết trên nền tảng này có thể được sử dụng trên các nền tảng khác mà không cần có bất cứ sự thay đổi nào.

Các tính năng của Cordova:

* + - Command Line Interface (Cordova CLI): đây là một công cụ dùng để khởi tạo một dự án , xây dựng ứng dụng trên nhiều nên tảng khác nhau cũng như là thêm nhiều plugin có ích để giúp việc phát triển trở nên dễ dàng hơn.
    - Cordova Core Components: Cordova đưa ra một tập các thành phần mà mọi ứng dụng di động cần đến.
    - Cordova Plugins: Cordova đưa ra các API để sử dụng các chức năng của thiết bị di động như cảm biến, camera, GPS…

## FRAMEWORK IONIC

### Giới thiệu

Framework Ionic là một SDK (bộ công cụ phát triển phần mềm) mã nguồn mở cho phép các nhà phát triển xây dựng các ứng dụng di động lai chất lượng cao, hiệu quả bằng các công nghệ web quen thuộc (HTML, CSS và JavaScript). Sau khi phát hành phiên bản alpha của Ionic vào tháng 11 năm 2013, chúng tôi đã phát hành bản beta 1.0 vào tháng 3 năm 2014 và trận chung kết 1.0 vào tháng 5 năm 2015. Phiên bản mới nhất của Ionic tính đến thời điểm hiện tại là 4.3.1.

Có thể coi Ionic như là một bộ khung front-end giúp người dùng kiểm soát hình ảnh và trải nghiệm trên ứng dụng. Giống như “Bootstrap for Native”, nhưng với sự hỗ trợ của một lượng lớn các thành phần di động, hiệu ứng chuyển động mượt mà và thiết kế đẹp mắt. Không như các framework khác, Ionic mang đến cho chúng ta những thành phần giao diện và cách bố trí mang phong cách rất giống với phong cách thiết kế mặc định trên thiết bị di động.

Vì Ionic là một framework HTML5 nên nó cần bộ đóng gói (wrapper) như Cordova hay PhoneGap để có thể chạy được như một ứng dụng di động. Cordova là bộ đóng gói mặc định trong Ionic framework.

Ionic sử dụng AngularJS để tạo ra một framework tốt nhất cho việc xây dựng các ứng dụng di động đa nền tảng, nó cung cấp một bộ giao diện người dùng (UI) mã nguồn mở miễn phí đi cùng với các tính năng của AngularJS. Việc xây dựng ứng dụng dựa trên AngularJS đòi hỏi mã nguồn phải có khả năng mở rộng cao để bổ sung các tính năng mới. Tuy nhiên với Ionic, người ta có thể tái sử dụng các chức năng trong ứng dụng trên các nền tảng khác nhau đồng thời vẫn có thể tùy chỉnh giao diện người dùng cho mỗi nền tảng riêng biệt. Các thành phần trong Ionic như danh sách, slide,.. chính là các directive (các thuộc tính của thẻ HTML dùng trong Angular) của AngularJS. Đó là lí do khiến cho Ionic và AngularJS kết hợp rất tốt với nhau.

Ưu điểm:

* + - Các ứng dụng lai có nhiều ưu điểm như có thể hiển thị cho nhiều nền tảng.
    - Tận dụng các tính năng sẵn có của thiết bị di động như GPS, camera....
    - Thời gian và chi phí dùng để phát triển ứng dụng thấp hơn nhiều so với native.
    - Chỉ cần biết JavaScript mà không cần phải biết từng ngôn ngữ lập trình.
    - Có thể sử dụng các kỹ năng từ lập trình web.
    - Đa nền tảng.
    - Dễ dàng thiết kế giao diện cho các thiết bị có kích cỡ khác nhau.
    - Việc sử dụng AngularJS làm core giúp phần xử lý UI linh động hơn so với JavaScript hay thư viện jQuery, đồn thời cũng mang lại lợi thế lớn so với các framework cho ứng dụng lai khác.
    - Ionic cung cấp đầy đủ các thành phần trong giao diện người dùng như Pull-to-Refresh, Infinite-loader, tabs, ..

Nhược điểm:

* + - Tốc độ chậm với một số tính năng chuyển trang trên di động.
    - Các plugin có thể không tương thích với một số thiết bị và nền tảng. Một số API cũng chưa được hỗ trợ để giao tiếp với thiết bị.
    - Vẫn còn trong giai đoạn phát triển.
    - Hiệu năng vẫn chưa cao và ổn định.
    - Cộng đồng phát triển ứng dụng vẫn còn chưa đông.

### Các thành phần cơ bản

* + - UI components (Thành phần giao diện người dùng): Ứng dụng Ionic được tạo thành từ các khối xây dựng cao cấp được gọi là các Component. Component cho phép bạn xây dựng giao diện cho ứng dụng một cách nhanh chóng. Component gồm có Action Sheets, Alert, Badges, Button,...
    - API (Giao diện lập trình ứng dụng): Gồm hai loại Component và Service
      * API Component: bao gồm các lớp như Checkbox, Toggle hoặc Item. Component chỉ cho bạn thấy cách sử dụng chúng, ngoài ra còn liệt kê các bộ chọn, các thuộc tính và sự kiện có sẵn của chúng.
      * API Service: bao gồm các lớp như MenuController, Config hoặc Platform. Đây là những dịch vụ được cung cấp bởi Ionic có thể được tiêm vào các lớp của bạn.
    - Native: Là một trình bao bọc TypeScript cho các plugin Cordova/PhoneGap giúp bổ sung thêm bất kỳ chức năng gốc nào bạn cần cho ứng dụng di động Ionic của mình một cách dễ dàng như tương tác với camera, GPS,...
    - Storage: là một cách dễ dàng để lưu trữ các cặp key/value và các đối tượng JSON. Storage sử dụng nhiều công cụ lưu trữ bên dưới điện thoại, chọn công cụ lưu trữ tốt nhất có sẵn tùy thuộc vào nền tảng. Khi chạy trong ngữ cảnh ứng dụng Native, Storage sẽ ưu tiên sử dụng SQLite, vì nó là một trong những CSDL dựa trên tệp ổn định, được sử dụng rộng rãi nhất và tránh được một số cạm bẫy của những thứ như localstorage và IndexedDB, chẳng hạn như hệ điều hành quyết định xóa dữ liệu đó trong các tình huống không gian đĩa thấp.
    - CLI (Giao diện dòng lệnh Ionic): là công cụ giúp phát triển ứng dụng Ionic.

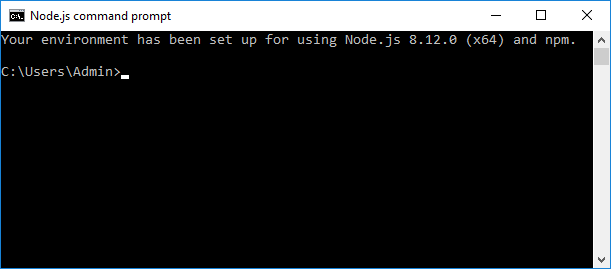
### Cài đặt và sử dụng

Ứng dụng Ionic được xây dựng và phát triển chủ yếu thông qua giao diện dòng lệnh Ionic (“CLI”) và sử dụng Cordova để xây dựng/triển khai dưới dạng ứng dụng Native. Điều này có nghĩa là chúng ta cần phải cài đặt một vài tiện ích để phát triển.

2.5.3.1 Cài đặt Node.js và NPM

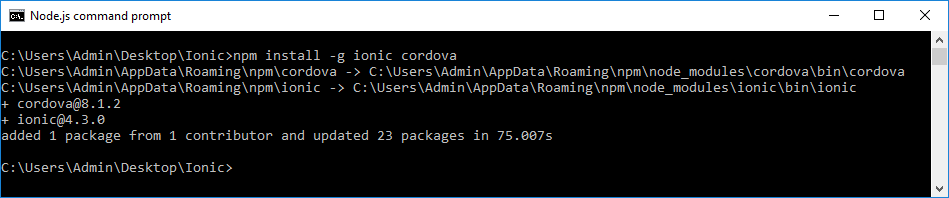
Hầu hết các công cụ trong CLI đều dựa trên Node và được quản lý thông qua NPM. Cách nhanh nhất để cài đặt Node và NPM là thông qua trình cài đặt NodeJS. Tải Node.js phiên bản LTS mới nhất tại trang chủ về máy và cài đặt.

Sau khi cài đặt hoàn tất, chúng ta sẽ có Command Prompt của Node.js



2.5.3.2 Ionic CLI và Cordova

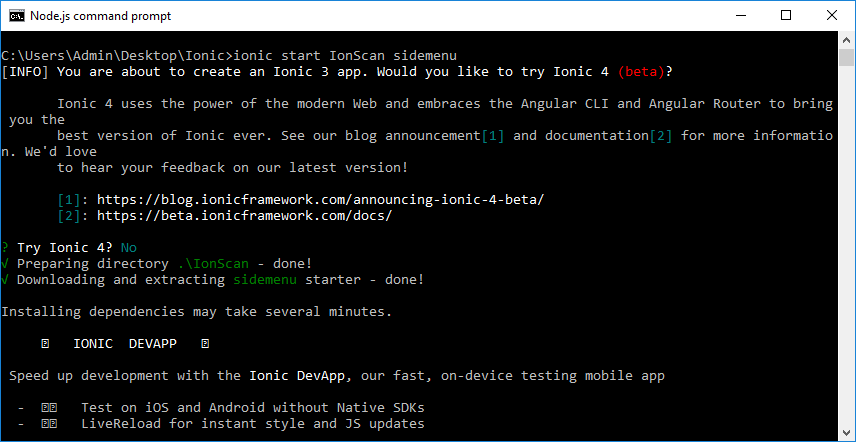
Với Command Prompt đang có, tiếp tục cài đặt Ionic và Cordova CLI bằng câu lệnh sau: $ npm install -g ionic cordova



2.5.3.3 Khởi tạo một ứng dụng

Khi đã xong, bắt đầu tạo ứng dụng Ionic, với IonScan là tên ứng dụng và sidemenu là template. Ionic có 3 template để lựa chọn: blank, sidemenu và tabs

$ ionic start IonScan sidemenu



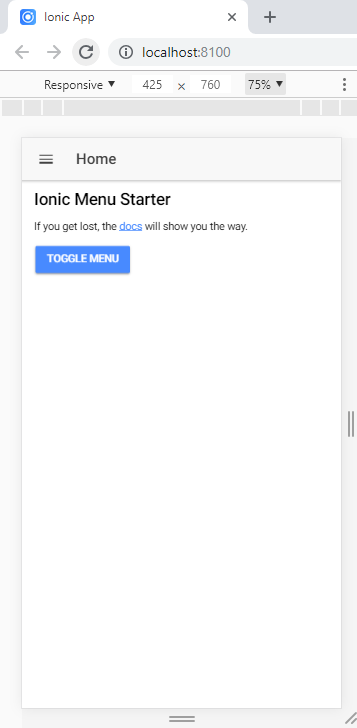
Trong lúc cài đặt, Ionic sẽ cho phép liên kết dự án với Ionic Pro SDK hoặc Git để lưu trữ dự án. Nếu bỏ qua bước này, chúng ta vẫn có thể liên kết trong quá trình phát triển ứng dụng.

2.5.3.4 Chạy ứng dụng

Để chạy ứng dụng, di chuyển vào thư mục đã được tạo bằng câu lệnh cd và sau đó chạy lệnh ion serve để kiểm tra ứng dụng ngay trên trình duyệt:

$ cd IonScan

$ ionic serve



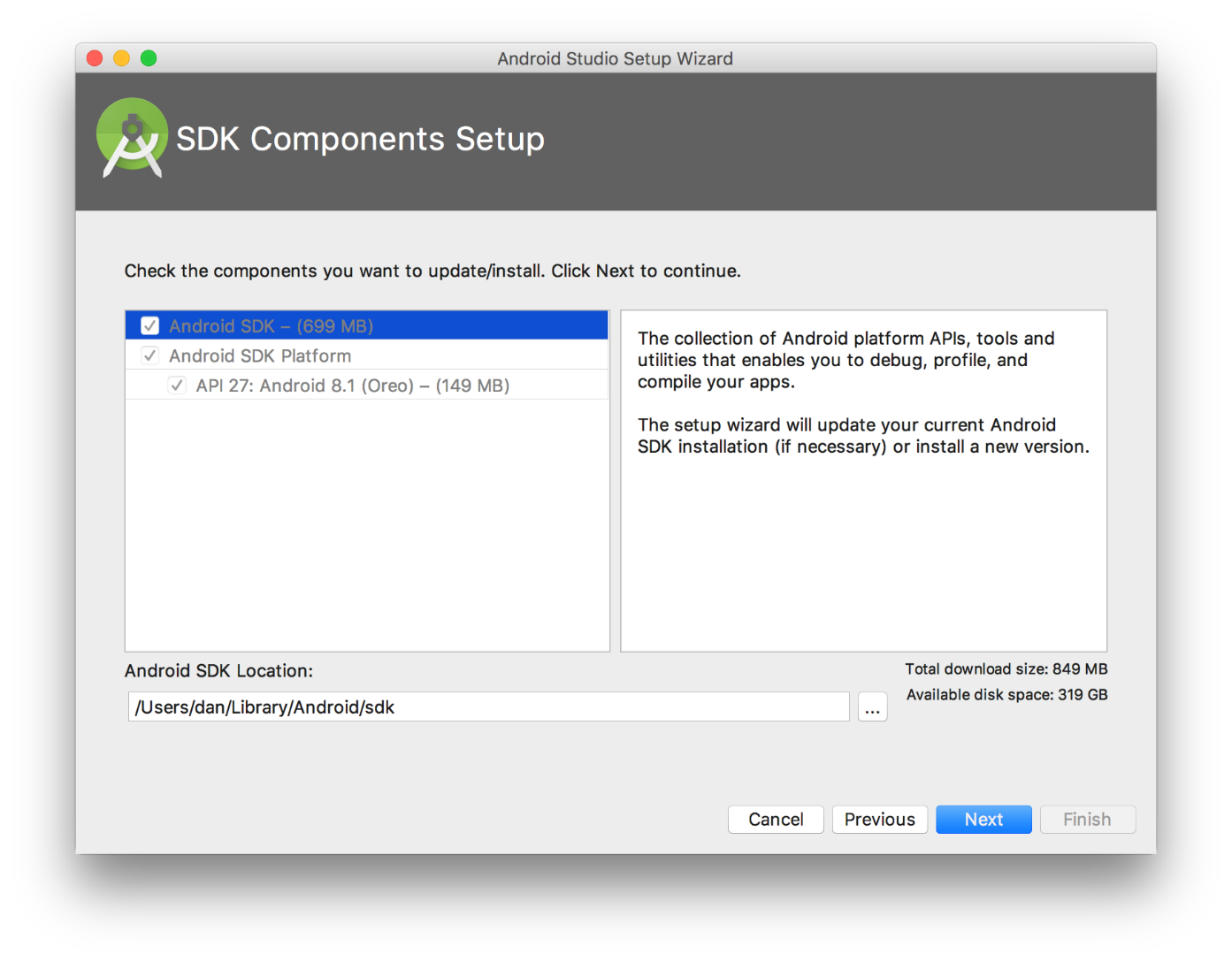
2.5.3.5 Cài đặt Java

Các ứng dụng Android Native được biên dịch bằng ngôn ngữ lập trình Java. Để tạo được ứng dụng Android Native chúng ta cần tải JDK8 (bộ công cụ phát triển Java) từ trang Java xuống, cài đặt và cấu hình.

2.5.3.6 Cài đặt Android Studio

Android Studio là IDE để tạo ứng dụng Android Native. Nó bao gồm Android SDK, cần được cấu hình để sử dụng trong dòng lệnh. Android Studio cũng được sử dụng để tạo thiết bị Android ảo, được yêu cầu cho trình giả lập Android. Ứng dụng Ionic cũng có thể được chạy trên một thiết bị.

Tải Android Studio về máy và cài đặt. Sau đó mở Android Studio lên, IDE sẽ phát hiện ra cần phải cài Android SDK. Trong màn hình SDK Components Setup, hoàn tất việc cài đặt SDK. Hãy lưu ý đến vị trí của Android SDK. Sau khi Android SDK được cài đặt xong, có thể đóng Android Studio lại.



2.5.3.7 Tạo tệp apk

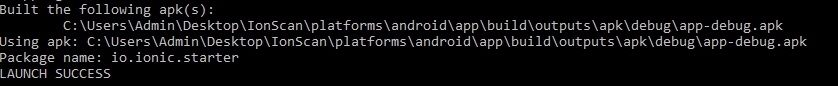
Kích hoạt nền tảng mà chúng ta muốn xây dựng (Android/iOS):

$ ionic cordova platform add android

Xây dựng dự án Ionic cho nền tảng vừa tạo:

$ ionic cordova build android

Sau khi xây dựng thành công, Ionic sẽ trả về đường dẫn chứa tệp apk



2.5.3.8 Cài đặt Native plugin hoặc NPM

Để sử dụng các Native plugin của Ionic hoặc NPM, chúng ta cần cài đặt Plugin như sau (ví dụ như Plugin Camera):

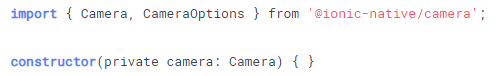
$ ionic cordova plugin add cordova-plugin-camera

$ npm install --save @ionic-native/camera

Tiếp theo, thêm nó vào NgModule trong app/app.module.ts:



Sau đó khai báo Plugin trong tệp TypeScript mà chúng ta cần sử dụng:



## FIREBASE

### Giới thiệu Firebase

Firebase là một dịch vụ CSDL thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với CSDL.

Các chức năng chính của Firebase:

* + - Realtime Database – Cơ sở dữ liệu thời gian thực: Firebase lưu trữ CSDL dưới dạng JSON và thực hiện đồng bộ CSDL tới tất cả các client theo thời gian thực. Điều đó có nghĩa là khi tất cả các client cùng sử dụng chung một CSDL đến từ Firebase, ứng dụng có thể tự động cập nhật mỗi khi dữ liệu trong CSDL được thêm mới hoặc sửa đổi. Ngoài ra Firebase còn cho phép phân quyền một cách đơn giản bằng cú pháp tương tự như javascript.
    - Firebase Hosting: Người dùng có thể triển khai một ứng dụng nền web chỉ với vài giây với hệ thống Firebase, và các dữ liệu sẽ được lưu trữ đám mây đồng thời được bảo mật thông qua giao thức truy cập SSL. Các ứng dụng sẽ được cấp một tên miền dạng \*.firebaseapp.com hoặc người dùng có thể trả tiền để sử dụng tên miền của riêng mình.
    - Firebase Authentication – Hệ thống xác thực người dùng.

Ưu điểm của Firebase:

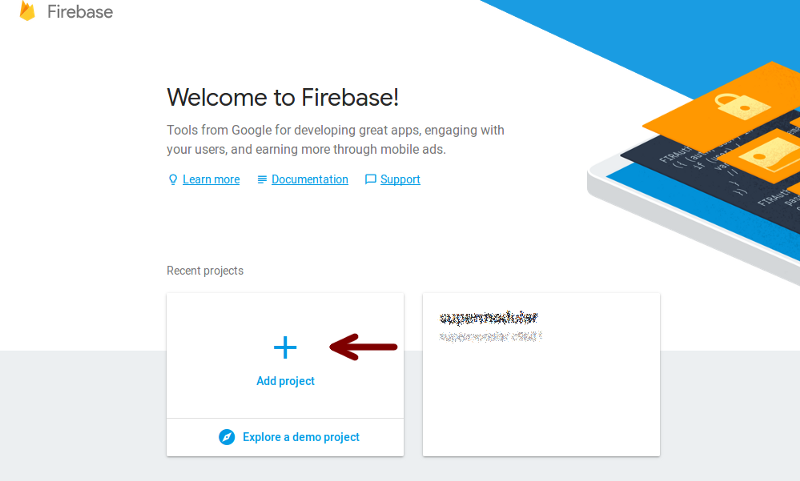
* + - Triển khai ứng dụng cực nhanh: Tiết kiệm thời gian quản lý và đồng bộ CSDL và xây dựng ứng dụng đa nền tảng. Không chỉ vậy, Firebase còn giúp đơn giản hóa quá trình đăng kí và đăng nhập vào ứng dụng.
    - Bảo mật: Firebase hoạt động dựa trên nền tảng cloud và thực hiện kết nối thông qua giao thức bảo mật SSL, người dùng có thể yên tâm về việc bảo mật của dữ liệu cũng như đường truyền giữa client và server. Không chỉ vậy, việc cho phép phân quyền người dùng CSDL bằng cú pháp javascipt cũng nâng cao độ bảo mật cho ứng dụng hơn, bởi vì chỉ những người dùng được cho phép mới có thể có quyền chỉnh sửa cơ sở dữ liệu.
    - Tính linh hoạt và khả năng mở rộng: Sử dụng Firebase sẽ giúp người dễ dàng nâng cấp hoặc mở rộng dịch vụ. Ngoài ra firebase còn cho phép người dùng tự xây dựng server của riêng mình để thuận tiện cho việc quản lý hơn. Việc Firebase sử dụng NoSQL, giúp cho CSDL của người dùng sẽ không bị bó buộc trong các bảng và các trường mà có thể tùy ý xây dựng CSDL theo cấu trúc riêng.
    - Sự ổn định: Firebase hoạt động dựa trên nền tảng đám mây đến từ Google vì vậy gần như không phải lo lắng về việc sập server, tấn công mạng như DDOS, tốc độ kết nối lúc nhanh lúc chậm,…, bởi vì đơn giản là Firebase hoạt động trên hệ thống server của Google. Hơn nữa nhờ hoạt động trên nền tảng đám mây nên việc nâng cấp, bảo trì server cũng diễn ra rất đơn giản  mà không cần phải dừng server để nâng cấp như truyền thống.
    - Giá thành: Google Firebase có rất nhiều gói dịch vụ với các mức dung lượng lưu trữ cũng như băng thông khác nhau với mức giá dao động từ miễn phí đến $1500 đủ để đáp ứng được nhu cầu của tất cả các đối tượng. Điều này giúp bạn tới ưu hóa được vốn đầu tư và vận hành tùy theo số lượng người sử dụng. Ngoài ra người dùng còn không mất chi phí để bảo trì, nâng cấp, khắc phục các sự cố bởi vì những điều này đã có Firebase lo.

### Giới thiệu Firebase Authentication

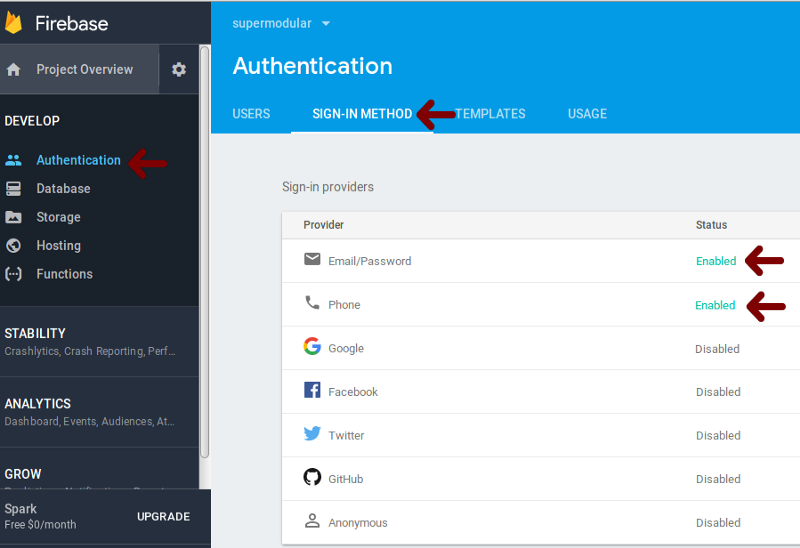
Hầu hết các ứng dụng đều cần biết danh tính của người dùng. Việc biết danh tính của người dùng cho phép ứng dụng lưu dữ liệu người dùng trên đám mây một cách an toàn và cung cấp trải nghiệm được cá nhân hóa giống nhau trên tất cả các thiết bị của người dùng. Firebase Authentication cung cấp dịch vụ backend, SDK dễ sử dụng và thư viện giao diện người dùng được tạo sẵn để xác thực người dùng cho ứng dụng của bạn. Firebase hỗ trợ xác thực bằng mật khẩu, số điện thoại, tích hợp các công nghệ xác thực của các nhà cung cấp nhận dạng liên kết phổ biến như Google, Facebook và Twitter, v.v.

### Sử dụng Firebase Authentication trong dự án Ionic

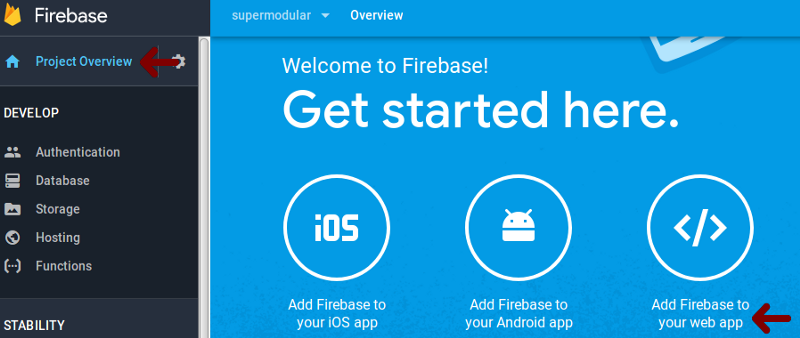
Truy cập <https://console.firebase.google.com/> và tạo dự án mới.

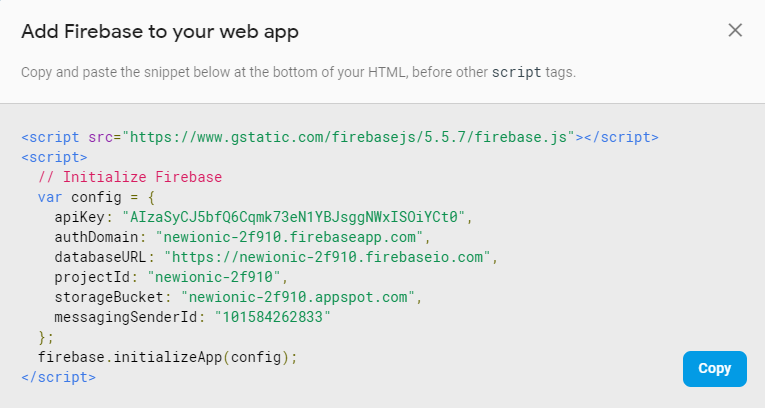


Sau đó, bật phương thức xác thực cần dùng lên, trong dự án này là Email và Phone.

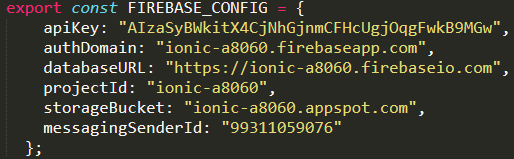


Truy cập vào Project Overview và chọn Add Firebase to your web app, chúng ta sẽ thấy được đoạn code bên dưới:





Tạo tệp app.firebase.config.ts trong src/app và copy đoạn code ở bên trên vào:



Tiếp theo chúng ta cài đặt AngularFire2 thông qua CLI

$ npm install firebase angularfire2 --save

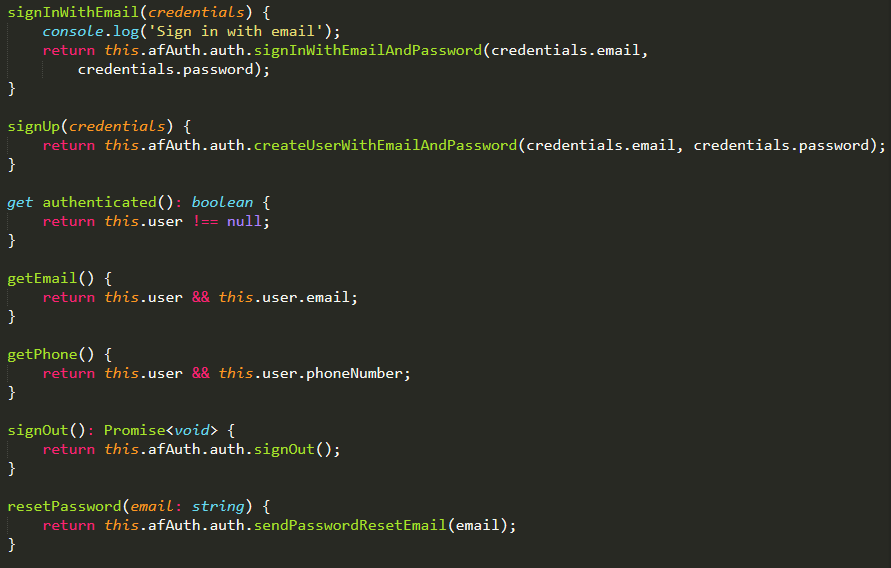
AngularFire2 là thư viện AngularJS được hỗ trợ chính thức cho Firebase. Nó cho phép chúng ta liên kết các tham chiếu Firebase với các mô hình Angular để chúng được đồng bộ với cơ sở dữ liệu và với tất cả các ứng dụng khách khác hiện đang sử dụng ứng dụng.

Sau đó chúng ta khai báo chúng trong tệp app.module.ts





Tạo một folder service chứa tệp auth.service.ts để sử dụng các lệnh đăng nhập, đăng ký của AngularFireAuth:



Sau đó, chúng ta chỉ cần khai báo tệp vừa tạo trong tệp TypeScript cần dùng.

## OPENCV

### Lịch sử

* + - Năm 1999, OpenCV được tạo ra ở Intel vào bởi Gary Bradski.
    - Năm 2000, bản phát hành đầu tiên ra mắt. Vadim Pisarevsky gia nhập Gary Bradski để quản lý đội OpenCV phần mềm của Intel.
    - Năm 2005, OpenCV đã được sử dụng trên Stanley; chiếc xe giành được giải thưởng DARPA Grand Challenge năm 2005.
    - Sau đó, sự phát triển tích cực của nó được tiếp tục dưới sự hỗ trợ của Willow Garage, với Gary Bradski và Vadim Pisarevsky dẫn đầu dự án.
    - OpenCV hiện hỗ trợ vô số các thuật toán liên quan đến Computer Vision và Machine Learning và đang mở rộng từng ngày.

### Giới thiệu

OpenCV (Open Computer Vision) là một thư viện được cấp phép BSD mã nguồn mở hàng đầu với hàng trăm thuật toán đã được tối ưu cho xử lý về thị giác máy tính, máy học và xử lý ảnh. OpenCV đươc viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. OpenCV hỗ trợ nhiều ngôn ngữ lập trình khác nhau như C ++, Python và Java. OpenCV có sẵn trên các nền tảng khác nhau bao gồm Windows, Linux, MacOS, Android và iOS. Giao diện cho các hoạt động GPU tốc độ cao dựa trên CUDA và OpenCL cũng đang được phát triển tích cực. OpenCV.js mang OpenCV đến nền tảng web mở và tạo cơ hội cho lập trình viên JavaScript sử dụng.

### OpenCV.js

Web là nền tảng điện toán mở phổ biến nhất. Với các tiêu chuẩn HTML5 được triển khai trong mọi trình duyệt, các ứng dụng web có thể hiển thị video trực tuyến bằng thẻ video HTML5, quay video webcam qua API WebRTC và truy cập từng pixel của khung video thông qua API canvas. Với sự phong phú của nội dung đa phương tiện có sẵn, các nhà phát triển web đang cần một loạt các thuật toán xử lý hình ảnh và hình ảnh trong JavaScript để xây dựng các ứng dụng sáng tạo.

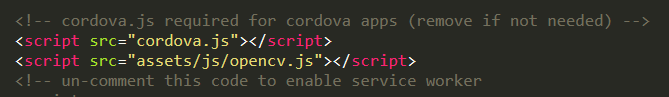
OpenCV.js là một ràng buộc JavaScript cho tập hợp con của các hàm OpenCV được chọn cho nền tảng web. Nó cho phép các ứng dụng web mới nổi với xử lý đa phương tiện hưởng lợi từ nhiều chức năng nhìn khác nhau có sẵn trong OpenCV. OpenCV.js sử dụng Emscripten để biên dịch các hàm OpenCV thành các mục tiêu asm.js hoặc WebAssembly, và cung cấp một API JavaScript cho ứng dụng web để truy cập chúng.

### Sử dụng OpenCV.js trong dự án Ionic

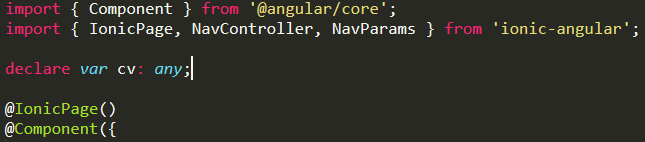
Tải tệp OpenCV.js tại trang <https://github.com/shkrwnd/openCV-js>

Chuyển tệp OpenCV.js vừa tải vào đường dẫn sau: src/assets/js.

Khai báo tệp OpenCV.js trong tệp index.html nằm trong thư mục src:



Như vậy, khi cần sử dụng OpenCV, chúng ta chỉ cần khai báo biến cv trong mỗi tệp TypeScript cần dùng:



### Các tính năng giao diện người dùng của OpenCV.js

* + - Hình ảnh: Tải một ảnh và hiển thị nó
      * Đọc ảnh[**cv.imread**](https://docs.opencv.org/3.4/d4/da8/group__imgcodecs.html#ga288b8b3da0892bd651fce07b3bbd3a56)**(imageSource):** đọc ảnh từ phần tử HTML canvas hoặc img. Trong đó imageSource có thể là phần tử id hoặc element của img hoặc canvas. Và kết quả trả về sẽ là một ma trận (mat) với các kênh được lưu trữ theo thứ tự RGBA.
      * Hiển thị ảnh[**cv.imshow**](https://docs.opencv.org/3.4/d7/dfc/group__highgui.html#ga453d42fe4cb60e5723281a89973ee563)**(canvasSource, mat):** hàm có thể chia tỉ lệ mat, tùy thuộc vào độ sâu của nó (8, 16, 32 bit).
    - Video: quay video từ Camera và phát.
    - Thêm một Trackbar vào ứng dụng: Tạo thanh trượt để kiểm soát các thông số nhất định.

### Các thao tác chính của OpenCV.js

**Thao tác cơ bản trên hình ảnh:** Tìm hiểu cách đọc và chỉnh sửa các giá trị pixel, làm việc với ROI hình ảnh và các hoạt động cơ bản khác:

* + - Truy cập thuộc tính hình ảnh: Thuộc tính hình ảnh bao gồm số hàng, cột và kích thước, độ sâu, kênh, loại dữ liệu hình ảnh.
    - Xây dựng Mat:
      * Có 4 hàm khởi tạo cơ bản:
        + Cơ bản: let mat = new cv.Mat();
        + Mảng hai chiều theo kích thước và loại:  
          let mat = new cv.Mat(size, type);
        + Mảng hai chiều theo hàng, cột và loại:  
          let mat = new cv.Mat(rows, cols, type);
        + Mảng hai chiều theo hàng, cột và loại có giá trị khởi tạo:  
          let mat = new cv.Mat(rows, cols, type, new cv.Scalar());
      * Có 3 hàm tĩnh:
        + Mat toàn số 0: let mat = cv.Mat.zeros(rows, cols, type);
        + Mat toàn số 1: let mat = cv.Mat.ones(rows, cols, type);
        + Mat là một ma trận nhận dạng:   
          let mat = cv.Mat.eye(rows, cols, type);
      * Có 2 hàm factory:
        + Sử dụng mảng JS để xây dựng một mat:   
          let mat = cv.matFromArray(rows, cols, type, array);
        + Sử dụng imgData để xây dựng một mat:  
          let mat = cv.matFromImageData(imgData);
    - Sao chép Mat:
      * Sao chép: let dst = src.clone();
      * Sao chép đến một mask được chỉ định: src.copyTo(dst, mask);
    - Chuyển đổi loại của Mat: src.convertTo(dst, rtype);   
      Trong đó rtype là loại ma trận đầu ra mong muốn hoặc, đúng hơn là độ sâu vì số lượng kênh giống như đầu vào có.
    - Truy cập và sửa đổi các giá trị pixel.
    - Cắt ảnh: roi (rect) với rect là hình chữ nhật cần cắt.

**Thao tác số học trên hình ảnh:** Thực hiện các phép toán số học trên ảnh:

* + - Cộng hai ảnh bằng hàm cv.add(src1, src2, dst, mask, dtype);
    - Trừ hai ảnh bằng hàm cv.subtract(src1, src2, dst, mask, dtype);  
      Cả hai hình ảnh phải có cùng độ sâu và loại.
    - Phép toán thao tác bit AND, OR, NOT và XOR. Rất có ích trong khi trích xuất bất kỳ phần nào của hình ảnh, xác định và làm việc với ROI không phải hình chữ nhật.

Một số cấu trúc dữ liệu

* + - Điểm: let point = new cv.Point(x, y); Trong đó x, y là tọa độ x, y của điểm (mặc định  là góc trên cùng bên trái của hình ảnh).
    - Vô hướng: let scalar = new cv.Scalar(R, G, B, Alpha);   
      Trong đó R, G, B, Alpha lần lượt là giá trị pixel của kênh màu đỏ, xanh lá, xanh và alpha.
    - Kích thước: let size = new cv.Size(width, height);   
      Trong đó width là chiều rộng và height là chiều dài.
    - Hình tròn: let circle = new cv.Circle(center, radius);   
      Trong đó center là điểm giữa và radius là bánh kính của hình tròn.
    - Hình chữ nhật: let rect = new cv.Rect(x, y, width, height);
    - Xoay ảnh: let rotatedRect = new cv.RotatedRect(center, size, angle);  
      Trong đó center là điểm giữa, size là chiều rộng và chiều dài của hình chữ nhật, góc quay theo chiều kim đồng hồ.

### Xử lý ảnh

Thay đổi không gian màu

* + - Thay đổi màu: [cv.cvtColor](https://docs.opencv.org/3.4/d8/d01/group__imgproc__color__conversions.html#ga397ae87e1288a81d2363b61574eb8cab) (src, dst, code, dstCn)

|  |  |
| --- | --- |
| src | hình ảnh đầu vào. |
| dst | hình ảnh đầu ra có cùng kích thước và chiều sâu như src |
| code | mã chuyển đổi không gian màu |
| dstCn | số kênh trong hình ảnh đích; nếu tham số là 0, số kênh được bắt nguồn tự động từ src và code. |

* + - Kiểm tra xem các phần tử mảng có nằm giữa các phần tử của hai mảng khác hay không: [cv.inRange](https://docs.opencv.org/3.4/d2/de8/group__core__array.html#ga48af0ab51e36436c5d04340e036ce981) (src, lowerb, upperb, dst)

|  |  |
| --- | --- |
| lowerb | bao gồm ranh giới thấp hơn Mat có cùng kích thước với src. |
| upperb | bao gồm ranh giới cao hơn Mat có cùng kích thước với src. |

Chuyển đổi hình học của hình ảnh

* + - Thay đổi kích thước: [cv.resize](https://docs.opencv.org/3.4/da/d54/group__imgproc__transform.html#ga47a974309e9102f5f08231edc7e7529d) (src, dst, dsize, fx = 0, fy = 0, interpolation)

|  |  |
| --- | --- |
| dsize | kích thước hình ảnh đầu ra |
| fx | hệ số tỷ lệ dọc theo trục hoành |
| fy | hệ số tỷ lệ dọc theo trục tung |
| interpolation | phương pháp nội suy |

* + - Dịch chuyển vị trí của đối tượng: [cv.warpAffine](https://docs.opencv.org/3.4/da/d54/group__imgproc__transform.html#ga0203d9ee5fcd28d40dbc4a1ea4451983) (src, dst, M, dsize, flags, borderMode, borderValue)

|  |  |
| --- | --- |
| Mat | Ma trận chuyển đổi 2 × 3 (loại cv.CV\_64FC1) |
| flags | kết hợp các phương thức nội và cờ tùy chọn WARP\_INVERSE\_MAP có nghĩa M là phép biến đổi nghịch đảo |
| borderMode | phương pháp ngoại suy pixel |
| borderValue | giá trị được sử dụng trong trường hợp đường viền không đổi |

* + - Xoay ảnh: [cv.getRotationMatrix2D](https://docs.opencv.org/3.4/da/d54/group__imgproc__transform.html#gafbbc470ce83812914a70abfb604f4326) (center, angle, scale)
    - Tịnh tiến ảnh (biến đổi affin): [cv.getAffineTransform](https://docs.opencv.org/3.4/da/d54/group__imgproc__transform.html#ga8f6d378f9f8eebb5cb55cd3ae295a999) (src, dst)

**Phân ngưỡng ảnh:** Nếu giá trị pixel lớn hơn giá trị ngưỡng, nó được gán một giá trị (có thể là màu trắng), nếu không giá trị pixel được gán một giá trị khác (có thể là màu đen): [cv.threshold](https://docs.opencv.org/3.4/d7/d1b/group__imgproc__misc.html#gae8a4a146d1ca78c626a53577199e9c57) (src, dst, thresh, maxval, type)

|  |  |
| --- | --- |
| thresh | giá trị ngưỡng |
| maxval | giá trị tối đa để sử dụng với các loại ngưỡng cv.THRESH\_BINARY và cv.THRESH\_BINARY\_INV |
| type | Lọại ngưỡng |

Làm mịn ảnh

* + - Lọc ảnh: [cv.filter2D](https://docs.opencv.org/3.4/d4/d86/group__imgproc__filter.html#ga27c049795ce870216ddfb366086b5a04) (src, dst, ddepth, kernel, anchor, delta, borderType)

|  |  |
| --- | --- |
| ddepth | độ sâu mong muốn của hình ảnh đích. |
| kernel | hạt nhân chập (hoặc đúng hơn là một hạt nhân tương quan) |
| anchor | neo của hạt nhân chỉ ra vị trí tương đối của một điểm lọc trong hạt nhân |
| delta | giá trị tùy chọn được thêm vào các pixel đã lọc trước khi lưu chúng trong dst. |
| borderType | phương pháp ngoại suy pixel |

* + - Làm mờ ảnh: bằng cách liên kết hình ảnh với hạt nhân lọc low-pass
      * Trung bình: lấy trung bình của tất cả các điểm ảnh trong vùng nhân và thay thế phần tử trung tâm.  
        [cv.blur](https://docs.opencv.org/3.4/d4/d86/group__imgproc__filter.html#ga8c45db9afe636703801b0b2e440fce37) (src, dst, ksize, anchor = new [cv.Point](https://docs.opencv.org/3.4/dc/d84/group__core__basic.html#ga1e83eafb2d26b3c93f09e8338bcab192)(-1, -1), borderType = [cv.BORDER\_DEFAULT](https://docs.opencv.org/3.4/d2/de8/group__core__array.html#gga209f2f4869e304c82d07739337eae7c5afe14c13a4ea8b8e3b3ef399013dbae01)

|  |  |
| --- | --- |
| ksize | làm mờ kích thước hạt nhân |

* + - * Làm mờ Gaussian: sử dụng hạt nhân Gaussian để làm mờ.  
        [cv.GaussianBlur](https://docs.opencv.org/3.4/d4/d86/group__imgproc__filter.html#gaabe8c836e97159a9193fb0b11ac52cf1) (src, dst, ksize, sigmaX, sigmaY = 0, borderType = [cv.BORDER\_DEFAULT](https://docs.opencv.org/3.4/d2/de8/group__core__array.html#gga209f2f4869e304c82d07739337eae7c5afe14c13a4ea8b8e3b3ef399013dbae01))

|  |  |
| --- | --- |
| sigmaX | Độ lệch chuẩn của hạt nhân Gaussian theo hướng X. |
| sigmaY | Độ lệch chuẩn của hạt nhân Gaussian theo hướng Y |

* + - * Làm mờ trung bình: lấy trung bình của tất cả các điểm ảnh trong vùng nhân và phần tử trung tâm được thay thế bằng giá trị trung bình này.  
        [cv.medianBlur](https://docs.opencv.org/3.4/d4/d86/group__imgproc__filter.html#ga564869aa33e58769b4469101aac458f9) (src, dst, ksize)
      * Lọc song phương: có hiệu quả cao trong việc loại bỏ nhiễu trong khi vẫn giữ các cạnh sắc nét.  
        [cv.bilateralFilter](https://docs.opencv.org/3.4/d4/d86/group__imgproc__filter.html#ga9d7064d478c95d60003cf839430737ed) (src, dst, d, sigmaColor, sigmaSpace, borderType)

|  |  |
| --- | --- |
| d | đường kính của mỗi vùng lân cận pixel được sử dụng trong quá trình lọc. |
| sigmaColor | lọc sigma trong không gian màu. |
| sigmaSpace | lọc sigma trong không gian tọa độ. |

## Git và Github

### Giới thiệu Git

Git là Hệ thống quản lý phiên bản phân tán (Distributed Version Control System *–*DVCS) mã nguồn mở được phát triển bởi [Linus Torvalds](https://vi.wikipedia.org/wiki/Linus_Torvalds) vào năm [2005](https://vi.wikipedia.org/wiki/2005), ban đầu dành cho việc phát triển [nhân Linux](https://vi.wikipedia.org/wiki/H%E1%BA%A1t_nh%C3%A2n_Linux). DVCS là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (commit) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia.

Hiện nay, Git trở thành một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất. Trên Git, ta có thể lưu trạng thái của tệp dưới dạng lịch sử cập nhật. Vì thế, có thể đưa tệp đã chỉnh sửa một lần về trạng thái cũ hay có thể biết được tệp đã được chỉnh sửa chỗ nào. Nó có thể giúp các lập trình viên quản lý mã nguồn tốt hơn với cơ chế quản lý phiên bản.

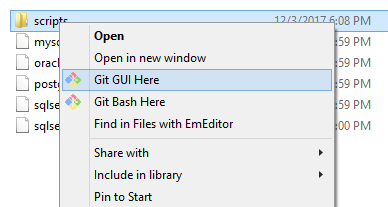
### Giới thiệu Github

GitHub là một nền tảng lưu trữ mã nguồn trên web để kiểm soát phiên bản và cộng tác dành cho các dự án có sử dụng hệ thống kiểm soát Git revision. Nó cho phép bạn và những người khác làm việc cùng nhau trên các dự án từ bất cứ đâu.

### Sử dụng Git và GitHub trong dự án Ionic

Truy cập vào trang <https://git-scm.com/download/win> để tải Git cho Windows.

Sau khi cài đặt xong, nhấn phải chuột vào một thư mục bất kỳ, một Context-Menu sẽ hiển thị, chúng ta có thể nhìn thấy các Menu-Item của Git, điều này chứng tỏ rằng chúng ta đã cài đặt Git thành công.

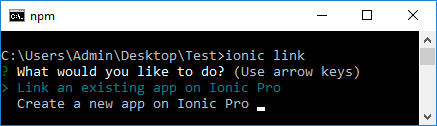


Hình 1. Menu Item của Git sau khi cài đặt xong

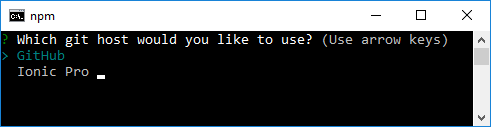
Tiếp đó, chúng ta truy cập vào trang chủ của Github <https://github.com/>, đăng ký tài khoản và tạo Repository để lưu trữ mã nguồn. Qua đó, chúng ta có thể push code từ dự án lên Repository và clone về để sử dụng thông qua giao diện của Git.

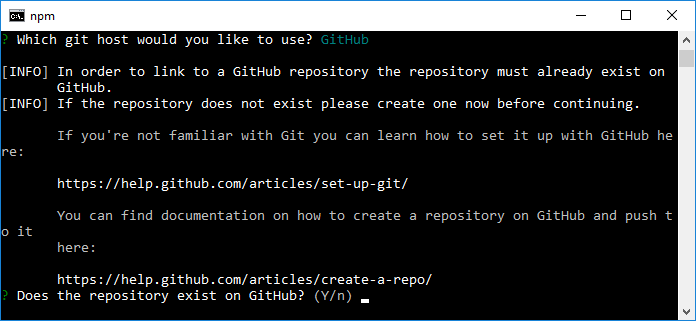
Đối với dự án Ionic, để sử dụng Git, chúng ta phải liên kết dự án Ionic, Ionic sẽ cho chúng ta 2 lựa chọn là kết nối với ứng dụng đã tồn tại trên Ionic Pro hoặc tạo một ứng dụng mới

$ ionic link

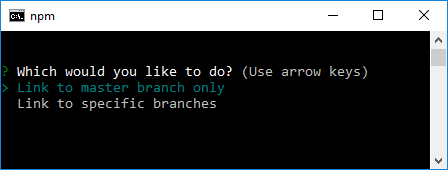


Sau đó Ionic sẽ hỏi chúng ta chọn git host của Ionic Pro hay Github, ứng dụng scanner này sử dụng Github.

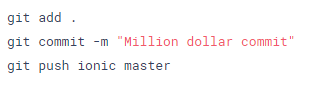


Tiếp theo Ionic sẽ hỏi Repository đó đã tồn tại hay chưa:

Nếu chưa Ionic sẽ thoát khỏi quá trình liên kết. Nếu có Ionic sẽ hiển thị danh sách Repository có trên Github. Sau khi chọn Repository, chúng ta tiếp tục bước chọn brach:



Như vậy quá trình liên kết với Github đã hoàn tất. Tiếp theo chúng ta có thể thao tác trực tiếp với Github trong Command Prompt của Node.js bằng các câu lệnh như:



# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Chương này trình bày các kết quả, mô hình sau quá trình phân tích và thiết kế chức năng, dữ liệu của hệ thống. Bao gồm: giới thiệu các chức năng có trong hệ thống, đặt tả chức năng, mô hình dữ liệu và mô tả các bảng dữ liệu.



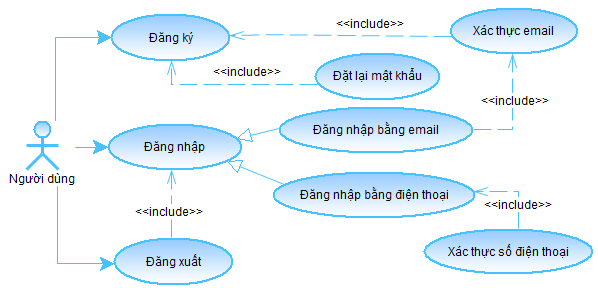
## Các yêu cầu và chức năng cần có

* + - Đăng nhập:
      * Đăng nhập bằng email (cần xác thực email).
      * Đăng nhập bằng điện thoại (cần xác thực số điện thoại).
    - Đăng ký.
    - Đặt lại mật khẩu.
    - Đăng xuất.
    - Quản lý thư mục (thêm, sửa, xóa).
    - Chụp ảnh (chứng minh thư, hộ chiếu, tài liệu).
    - Sử dụng ảnh có sẵn trong điện thoại.
    - Xử lý ảnh (cắt ảnh, thay đổi màu ảnh, xoay trái, xoay phải).
    - Quản lý ảnh (thêm, sửa, xóa).
    - Chia sẻ (dạng ảnh, dạng PDF).
    - Tải ảnh lên server.

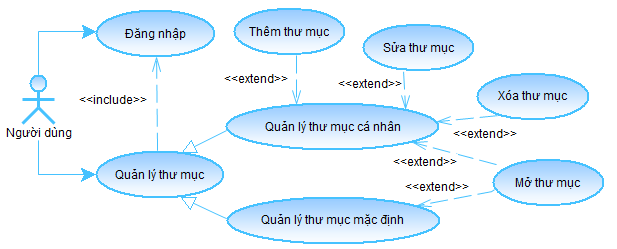
## Mô hình chức năng, đặc tả chức năng

Dựa vào các yêu cầu chức năng ở phần trên, mô hình Use Case của ứng dụng và đặc tả các Use Case được thể hiện như sau:

### Mô hình Use Case người dùng



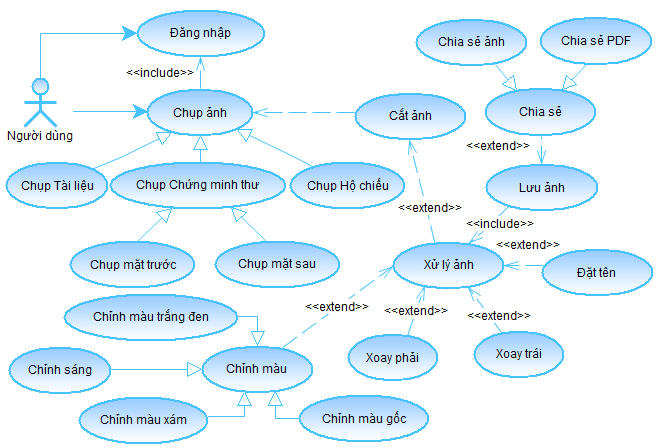
Mô hình Use Case người dùng với chức năng Đăng nhập



Mô hình Use Case người dùng với chức năng Quản lý thư mục



Mô hình Use Case người dùng với chức năng Quản lý ảnh



Mô hình Use Case người dùng với chức năng Chụp ảnh

### Đăng nhập

|  |
| --- |
| **Mã số: UC\_01** |
| Tên Use Case: Đăng nhập. |
| Mô tả: Cung cấp các chức năng đăng nhập bằng email hoặc số điện thoại. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Không có. |
| Cách kích hoạt: Mở ứng dụng. |
| Các luồng xử lý:   1. Hiển thị 2 tab là đăng nhập bằng email hoặc đăng nhập bằng số điện thoại, mặc định là tab ‘Email’, nút “Đăng ký” và nút “Quên mật khẩu”. 2. Nếu chọn tab ‘Email’ thì thực hiện [UC\_02](#_Đăng_nhập_bằng). 3. Nếu chọn tab “Điện thoại” thì thực hiện [UC\_03](#_Đăng_nhập_bằng_1). 4. Nếu nhấn nút “Đăng ký” thì thực hiện [UC\_06](#_Đăng_ký). Kết thúc chức năng. 5. Nếu nhấn nút “Quên mật khẩu” thì thực hiện [UC\_07](#_Đặt_lại_mật). Kết thúc chức năng. |

### Đăng nhập bằng email

|  |
| --- |
| **Mã số: UC\_02** |
| Tên Use Case: Đăng nhập bằng email. |
| Mô tả: Cung cấp chức năng đăng nhập bằng email. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Xác thực email. |
| Cách kích hoạt: Nhấn tab “Email” trong trang “Đăng nhập”. |
| Các luồng xử lý:   1. Hiển thị khung nhập tài khoản email, mật khẩu và nút đăng nhập. 2. Nếu điền xong tài khoản, mật khẩu và nhấn nút “Đăng nhập” thì thực hiện [UC\_09](#_Quản_lý_thư). Kết thúc chức năng.   Luồng phụ:   1. Nếu khung email hoặc mật khẩu trống thì xuất hiện thông báo lỗi. 2. Nếu nhập sai định dạng email thì xuất hiện thông báo lỗi. 3. Nếu email chưa được đăng ký thì xuất hiện thông báo lỗi. 4. Nếu mật khẩu không đúng với tài khoản đã đăng ký thì xuất hiện thông báo lỗi. 5. Nếu tài khoản chưa được xác thực thì xuất hiện thông báo lỗi. |

### Đăng nhập bằng số điện thoại

|  |
| --- |
| **Mã số: UC\_03** |
| Tên Use Case: Đăng nhập bằng số điện thoại. |
| Mô tả: Cung cấp chức năng đăng nhập bằng số điện thoại. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Không có. |
| Cách kích hoạt: Nhấn tab “Điện thoại” trong trang “Đăng nhập”. |
| Các luồng xử lý:   1. Hiển thị khung nhập tài khoản số điện thoại và nút đăng nhập. 2. Nếu nhập xong khung số điện thoại và nhấn nút “Đăng nhập” thì thực hiện [UC\_05](#_Xác_thực_số). Kết thúc chức năng.   Luồng phụ:   1. Nếu nhập sai định dạng số điện thoại thì xuất hiện thông báo lỗi. |

### Xác thực email

|  |
| --- |
| **Mã số: UC\_04** |
| Tên Use Case: Xác thực email. |
| Mô tả: Cung cấp chức năng xác thực email cho tài khoản đã đăng ký. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng ký tài khoản email. |
| Cách kích hoạt: Nhấn nút “Đăng ký” trong trang “Đăng ký”. |
| Các luồng xử lý:   1. Hiển thị thông báo “Đã gửi email xác thực” và trở về trang “Đăng nhập”. 2. Mở email vừa nhận được và nhấn vào đường dẫn đính kèm để xác thực. Kết thúc chức năng. |

### Xác thực số điện thoại

|  |
| --- |
| **Mã số: UC\_05** |
| Tên Use Case: Xác thực số điện thoại. |
| Mô tả: Cung cấp chức năng xác thực số điện thoại. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng nhập bằng số điện thoại. |
| Cách kích hoạt: Nhấn nút “Đăng nhập” trong trang “Đăng ký”. |
| Các luồng xử lý:   1. Hiển thị thông báo “Nhập mã xác thực” gồm một khung nhập mã và hai nút “Gửi”, “Hủy”: 2. Nếu nhấn nút “Gửi” thì thực hiện [UC\_09](#_Quản_lý_thư). Kết thúc chức năng. 3. Nếu nhấn nút “Hủy” thì thông báo biến mất. Kết thúc chức năng.   Luồng phụ:   1. Nếu nhập sai mã xác thực thì xuất hiện thông báo lỗi. |

### Đăng ký

|  |
| --- |
| **Mã số: UC\_06** |
| Tên Use Case: Đăng ký. |
| Mô tả: Cung cấp chức năng đăng ký tài khoản email. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Không có. |
| Cách kích hoạt: Nhấn vào nút “Đăng ký” trong trang “Đăng nhập”. |
| Các luồng xử lý:   1. Hiển thị khung nhập email, mật khẩu, nút đăng ký và nút “←”. 2. Nếu nhấn nút “←” thì trở về trang “Đăng nhập”. Kết thúc chức năng. 3. Nếu nhấn nút “Đăng ký” thì thực hiện [UC\_04](#_Xác_thực_email). Kết thúc chức năng.   Luồng phụ:   1. Nếu khung đăng ký hoặc mật khẩu trống thì xuất hiện thông báo lỗi. 2. Nếu sai định dạng email thì xuất hiện thông báo lỗi. 3. Nếu mật khẩu không đủ 6 kí tự thì xuất hiện thông báo lỗi. |

### Đặt lại mật khẩu

|  |
| --- |
| **Mã số: UC\_07** |
| Tên Use Case: Đặt lại mật khẩu. |
| Mô tả: Cung cấp chức năng đặt lại mật khẩu cho người dùng email. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng ký tài khoản email. |
| Cách kích hoạt: Nhấn vào nút “Quên mật khẩu” trong trang đăng nhập. |
| Các luồng xử lý:   1. Hiển thị khung nhập email, nút “Đặt lại mật khẩu” và nút “←”. 2. Nếu nhấn nút “←” thì trở về trang “Đăng nhập”. Kết thúc chức năng. 3. Nếu nhấn nút “Đặt lại mật khẩu” sẽ nhận được thông báo đã gửi email để đặt lại mật khẩu và trở về trang “Đăng nhập”. 4. Mở email nhấn vào đường dẫn đính kèm để tạo mật khẩu mới. Kết thúc chức năng.   Luồng phụ:   1. Nếu khung đặt lại mật khẩu trống thì xuất hiện thông báo lỗi. 2. Nếu sai định dạng email thì xuất hiện thông báo lỗi. |

### Đăng xuất

|  |
| --- |
| **Mã số: UC\_08** |
| Tên Use Case: Đăng xuất. |
| Mô tả: Cung cấp chức năng đăng xuất khỏi ứng dụng. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng nhập vào ứng dụng. |
| Cách kích hoạt: Nhấn vào nút “Đăng xuất” trong thanh menu bên trái. |
| Các luồng xử lý:   1. Mở thanh menu trái bằng cách nhấn vào nút “≡” hoặc vuốt màn hình sang phải. 2. Nếu nhấn nút “Đăng xuất” thì tài khoản sẽ đăng xuất khỏi ứng dụng. 3. Trở về trang “Đăng nhập”. Kết thúc chức năng. |

### Quản lý thư mục

|  |
| --- |
| **Mã số: UC\_09** |
| Tên Use Case: Quản lý thư mục. |
| Mô tả: Cung cấp các chức năng quản lý thư mục cá nhân và thư mục mặc định. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng nhập vào ứng dụng. |
| Cách kích hoạt: Tự động kích hoạt sau khi đăng nhập. |
| Các luồng xử lý:   1. Hiển thị tên trang “Lưu trữ”, tổng số thư mục, nút “≡”, danh sách các thư mục đang có gồm những thư mục cá nhân do người dùng tự tạo và 3 thư mục mặc định: “Chứng minh thư”, “Hộ chiếu” và “Tài liệu”. 2. Nếu nhấn nút “≡” hoặc vuốt màn hình sang phải sẽ hiển thị thanh menu trái gồm các nút “Lưu trữ”, “Chứng minh thư”, “Hộ chiếu”, “Tài liệu”, “Về chúng tôi” và “Đăng xuất”: 3. Nếu nhấn nút “Chứng minh thư” thì thực hiện [UC\_40](#_Chụp_chứng_minh). Kết thúc chức năng. 4. Nếu nhấn nút “Hộ chiếu” thì thực hiện [UC\_43](#_Chụp_hộ_chiếu). Kết thúc chức năng. 5. Nếu nhấn nút “Tài liệu” thì thực hiện [UC\_44](#_Chụp_tài_liệu). Kết thúc chức năng. 6. Nếu nhấn nút “Về chúng tôi” thì chuyển sang trang giới thiệu về ứng dụng. Kết thúc chức năng. 7. Nếu nhấn nút “Đăng xuất” thì thực hiện [UC\_08](#_3.2.5_Đăng_xuất). Kết thúc chức năng. 8. Đối với thư mục cá nhân, thực hiện [UC\_10](#_3.2.7_Thêm_thư). 9. Đối với thư mục mặc định, thực hiện [UC\_11](#_Quản_lý_thư_1). |

### Quản lý thư mục cá nhân

|  |
| --- |
| **Mã số: UC\_10** |
| Tên Use Case: Quản lý thư mục cá nhân. |
| Mô tả: Cung cấp các chức năng quản lý thư mục cá nhân như hiển thị, mở, thêm, sửa và xóa thư mục. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng nhập vào ứng dụng. |
| Cách kích hoạt: Tự động kích hoạt sau khi đăng nhập. |
| Các luồng xử lý:   1. Hiển thị nút “+” và danh sách các thư mục cá nhân đang có. 2. Nếu nhấn vào thư mục thì thực hiện [UC\_12](#_Mở_thư_mục). Kết thúc chức năng. 3. Nếu nhấn nút “+” thì thực hiện [UC\_13](#_Thêm_thư_mục). Kết thúc chức năng. 4. Nếu vuốt thư mục cá nhân sang trái thì xuất hiện nút vàng và nút đỏ: 5. Nếu nhấn nút vàng thì thực hiện [UC\_14](#_Sửa_thư_mục). Kết thúc chức năng. 6. Nếu nhấn nút đỏ thì thực hiện [UC\_15](#_Xóa_thư_mục). Kết thúc chức năng. |

### Quản lý thư mục mặc định

|  |
| --- |
| **Mã số: UC\_11** |
| Tên Use Case: Quản lý thư mục mặc định. |
| Mô tả: Cung cấp các chức năng quản lý thư mục mặc định như hiển thị và mở thư mục. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng nhập vào ứng dụng. |
| Cách kích hoạt: Tự động kích hoạt sau khi đăng nhập. |
| Các luồng xử lý:   1. Hiển thị danh sách các thư mục mặc định đang có. 2. Nếu nhấn vào thư mục thì thực hiện [UC\_12](#_Mở_thư_mục). Kết thúc chức năng. |

### Mở thư mục

|  |
| --- |
| **Mã số: UC\_12** |
| Tên Use Case: Mở thư mục. |
| Mô tả: Cung cấp chức năng mở thư mục. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng nhập vào ứng dụng. |
| Cách kích hoạt: Nhấn vào một thư mục trong trang “Lưu trữ”. |
| Các luồng xử lý:   1. Nhấn vào một thư mục trong danh sách những thư mục đang có thì thực hiện [UC\_16](#_Quản_lý_ảnh). Kết thúc chức năng. |

### Thêm thư mục

|  |
| --- |
| **Mã số: UC\_13** |
| Tên Use Case: Thêm thư mục. |
| Mô tả: Cung cấp chức năng thêm thư mục cá nhân. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng nhập vào ứng dụng. |
| Cách kích hoạt: Nhấn nút “+” nằm ở góc phải trên cùng trong trang “Lưu trữ”. |
| Các luồng xử lý:   1. Chuyển sang trang “Thêm thư mục”. Hiển thị khung nhập tên và khung chọn loại thư mục, nút “Lưu thư mục” và nút “←”. 2. Nếu vuốt màn hình sang phải sẽ hiển thị thanh menu trái gồm các nút “Lưu trữ”, “Chứng minh thư”, “Hộ chiếu”, “Tài liệu”, “Về chúng tôi” và “Đăng xuất”: 3. Nếu nhấn nút “Chứng minh thư” thì thực hiện [UC\_40](#_Chụp_chứng_minh). Kết thúc chức năng. 4. Nếu nhấn nút “Hộ chiếu” thì thực hiện [UC\_43](#_Chụp_hộ_chiếu). Kết thúc chức năng. 5. Nếu nhấn nút “Tài liệu” thì thực hiện [UC\_44](#_Chụp_tài_liệu). Kết thúc chức năng. 6. Nếu nhấn nút “Về chúng tôi” thì chuyển sang trang giới thiệu về ứng dụng. Kết thúc chức năng. 7. Nếu nhấn nút “Đăng xuất” thì thực hiện [UC\_08](#_3.2.5_Đăng_xuất). Kết thúc chức năng. 8. Nếu nhấn nút “←” thì quay lại trang “Lưu trữ”. Kết thúc chức năng. 9. Nếu nhập tên, chọn loại thưc mục và nhấn nút “Lưu thư mục” thì thư mục mới sẽ được tạo ra và trở về trang “Lưu trữ”. Kết thúc chức năng.   Luồng phụ:   1. Nếu tên thư mục trống thì xuất hiện thông báo lỗi. |

### Sửa thư mục

|  |
| --- |
| **Mã số: UC\_14** |
| Tên Use Case: Sửa thư mục. |
| Mô tả: Cung cấp chức năng sửa thư mục cá nhân. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng nhập vào ứng dụng. |
| Cách kích hoạt: Nhấn nút vàng sau khi vuốt thư mục trong trang “Lưu trữ” sang trái. |
| Các luồng xử lý:   1. Chuyển sang trang “Chỉnh sửa thư mục”. Hiển thị khung nhập tên với thông tin của thư mục được chọn, nút “Chỉnh sửa thư mục” và nút “←”. 2. Nếu vuốt màn hình sang phải sẽ hiển thị thanh menu trái gồm các nút “Lưu trữ”, “Chứng minh thư”, “Hộ chiếu”, “Tài liệu”, “Về chúng tôi” và “Đăng xuất”: 3. Nếu nhấn nút “Chứng minh thư” thì thực hiện [UC\_40](#_Chụp_chứng_minh). Kết thúc chức năng. 4. Nếu nhấn nút “Hộ chiếu” thì thực hiện [UC\_43](#_Chụp_hộ_chiếu). Kết thúc chức năng. 5. Nếu nhấn nút “Tài liệu” thì thực hiện [UC\_44](#_Chụp_tài_liệu). Kết thúc chức năng. 6. Nếu nhấn nút “Về chúng tôi” thì chuyển sang trang giới thiệu về ứng dụng. Kết thúc chức năng. 7. Nếu nhấn nút “Đăng xuất” thì thực hiện [UC\_08](#_3.2.5_Đăng_xuất). Kết thúc chức năng. 8. Nếu nhấn nút “←” thì quay lại trang “Lưu trữ”. Kết thúc chức năng. 9. Nếu chỉnh sửa tên thư mục xong và nhấn nút “Chỉnh sửa thư mục” thì thông tin sẽ được thay đổi và trở về trang “Lưu trữ”. Kết thúc chức năng.   Luồng phụ:   1. Nếu tên thư mục trống thì xuất hiện thông báo lỗi. |

### Xóa thư mục

|  |
| --- |
| **Mã số: UC\_15** |
| Tên Use Case: Xóa thư mục. |
| Mô tả: Cung cấp chức năng xóa thư mục cá nhân. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng nhập vào ứng dụng. |
| Cách kích hoạt: Nhấn nút đỏ sau khi vuốt thư mục trong trang “Lưu trữ” sang trái. |
| Các luồng xử lý:   1. Thư mục được xóa và cập nhật lại trang “Lưu trữ”. Kết thúc chức năng. |

### Quản lý ảnh

|  |
| --- |
| **Mã số: UC\_16** |
| Tên Use Case: Quản lý ảnh. |
| Mô tả: Cung cấp các chức năng quản lý ảnh trong ứng dụng như thêm, sửa, xóa, tải và chia sẻ ảnh. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng mở thư mục. |
| Cách kích hoạt: Tự động kích hoạt sau khi mở thư mục. |
| Các luồng xử lý:   1. Hiển thị nút “←”, tên thư mục hiện hành, tổng số ảnh, danh sách các ảnh đang có trong thư mục và nút “+” (chỉ có trong thư mục cá nhân). 2. Nếu vuốt màn hình sang phải sẽ hiển thị thanh menu trái gồm các nút “Lưu trữ”, “Chứng minh thư”, “Hộ chiếu”, “Tài liệu”, “Về chúng tôi” và “Đăng xuất”: 3. Nếu nhấn nút “Chứng minh thư” thì thực hiện [UC\_40](#_Chụp_chứng_minh). Kết thúc chức năng. 4. Nếu nhấn nút “Hộ chiếu” thì thực hiện [UC\_43](#_Chụp_hộ_chiếu). Kết thúc chức năng. 5. Nếu nhấn nút “Tài liệu” thì thực hiện [UC\_44](#_Chụp_tài_liệu). Kết thúc chức năng. 6. Nếu nhấn nút “Về chúng tôi” thì chuyển sang trang giới thiệu về ứng dụng. Kết thúc chức năng. 7. Nếu nhấn nút “Đăng xuất” thì thực hiện [UC\_08](#_3.2.5_Đăng_xuất). Kết thúc chức năng. 8. Nếu nhấn nút “←” nằm ở góc trái bên trên thì trở về trang “Lưu trữ”. Kết thúc chức năng. 9. Nếu nhấn nút “+” nằm ở góc phải bên dưới thì thực hiện [UC\_17](#_Thêm_ảnh). Kết thúc chức năng. 10. Đối với thư mục cá nhân, vuốt ảnh từ phải sang trái sẽ hiển thị 3 nút xanh, vàng và đỏ: 11. Nếu nhấn nút xanh thì thực hiện [UC\_24](#_Tải_ảnh_lên). 12. Nếu nhấn nút vàng thì thực hiện [UC\_25](#_Chia_sẻ). Kết thúc chức năng. 13. Nếu nhấn nút đỏ thì thực hiện [UC\_23](#_Xóa_ảnh). 14. Đối với thư mục mặc định, , vuốt ảnh từ phải sang trái sẽ hiển thị nút đỏ: 15. Nếu nhấn nút đỏ thì thực hiện [UC\_23](#_Xóa_ảnh). 16. Nếu nhấn vào ảnh thì thực hiện [UC\_22](#_Sửa_ảnh). Kết thúc chức năng. |

### Thêm ảnh

|  |
| --- |
| **Mã số: UC\_17** |
| Tên Use Case: Thêm ảnh. |
| Mô tả: Cung cấp chức năng thêm ảnh. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng quản lý ảnh. |
| Cách kích hoạt: Nhấn nút “+” nằm ở góc phải bên dưới. |
| Các luồng xử lý:   1. Hiển thị nút vàng và nút đỏ: 2. Nếu nhấn nút “vàng” thì thực hiện [UC\_18](#_Thêm_ảnh_trong). Kết thúc chức năng. 3. Nếu nhấn nút “đỏ” thì thực hiện [UC\_19](#_Thêm_ảnh_ngoài). Kết thúc chức năng. |

### Thêm ảnh trong ứng dụng

|  |
| --- |
| **Mã số: UC\_18** |
| Tên Use Case: Thêm ảnh trong ứng dụng. |
| Mô tả: Cung cấp chức năng thêm ảnh từ thư mục mặc định trong ứng dụng. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng thêm ảnh. |
| Cách kích hoạt: Nhấn nút vàng xuất hiện sau khi nhấn nút “+” nằm ở góc phải bên dưới trong chức năng thêm ảnh. |
| Các luồng xử lý:   1. Chuyển sang thư mục mặc định cùng loại với loại thư mục cá nhân. 2. Hiển thị nút “←”, tên thư mục, tổng số thư mục và danh sách các ảnh đang có trong thư mục, nút “+” sau mỗi ảnh. 3. Nếu nhấn nút “←” nằm ở góc trên cùng bên trái thì trở về trang quản lý ảnh. Kết thúc chức năng. 4. Nếu nhấn nút “+” thì ảnh sẽ được thêm vào thư mục hiện hành và quay về trang quản lý ảnh. Kết thúc chức năng. |

### Thêm ảnh ngoài ứng dụng

|  |
| --- |
| **Mã số: UC\_19** |
| Tên Use Case: Thêm ảnh ngoài ứng dụng. |
| Mô tả: Cung cấp chức năng thêm ảnh từ ngoài ứng dụng. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng thêm ảnh. |
| Cách kích hoạt: Nhấn nút đỏ xuất hiện sau khi nhấn nút “+” nằm ở góc phải bên dưới trong chức năng thêm ảnh. |
| Các luồng xử lý:   1. Hiển thị bảng thông báo “Get Picture” với hai lựa chọn “Drive” và “Bộ sưu tập”. 2. Nếu chọn “Drive” thì thực hiện[UC\_20](#_Thêm_ảnh_từ)**.** Kết thúc chức năng. 3. Nếu chọn “Bộ sưu tập” thì thực hiện [UC\_21](#_Thêm_ảnh_từ_1)**.** Kết thúc chức năng. |

### Thêm ảnh từ Drive

|  |
| --- |
| **Mã số: UC\_20** |
| Tên Use Case: Thêm ảnh từ Drive. |
| Mô tả: Cung cấp chức năng thêm ảnh từ tài khoản Gmail. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng thêm ảnh ngoài ứng dụng. |
| Cách kích hoạt: Nhấn nút “Drive” trong bảng thông báo “Get Picture”. |
| Các luồng xử lý:   1. Hiển thị bảng thông báo “Chọn tài khoản” gồm một danh sách các tài khoản Gmail hiện có trong điện thoại, nút “OK” và nút “Hủy”. 2. Nếu nhấn nút “Hủy” thì sẽ tắt bảng thông báo. Kết thúc chức năng. 3. Nếu chọn tài khoản và nhấn nút “OK” thì truy cập vào Drive, hiển thị danh sách thư mục và tệp có trong tài khoản, nút “Chọn” và nút “Hủy”: 4. Nếu nhấn nút “Hủy” thì quay lại trang quản lý ảnh. Kết thúc chức năng. 5. Nếu chọn tệp hoặc di chuyển đến thư mục cần tìm và chọn tệp rồi nhấn nút “Chọn” thì tải ảnh về và thực hiện [UC\_28](#_Cắt_ảnh). Kết thúc chức năng. |

### Thêm ảnh từ Bộ sưu tập

|  |
| --- |
| **Mã số: UC\_21** |
| Tên Use Case: Thêm ảnh từ Bộ sưu tập. |
| Mô tả: Cung cấp chức năng thêm ảnh từ Bộ sưu tập trong điện thoại. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng thêm ảnh ngoài ứng dụng. |
| Cách kích hoạt: Nhấn nút “Bộ sưu tập” trong bảng thông báo “Get Picture”. |
| Các luồng xử lý:   1. Truy cập vào Bộ sưu tập của điện thoại. 2. Nhấn chọn ảnh cần thêm. 3. Ảnh được chuyển sang thực hiện [UC\_28](#_Cắt_ảnh).Kết thúc chức năng. |

### Sửa ảnh

|  |
| --- |
| **Mã số: UC\_22** |
| Tên Use Case: Sửa ảnh. |
| Mô tả: Cung cấp chức năng sửa ảnh. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng quản lý ảnh. |
| Cách kích hoạt: Nhấn trực tiếp vào ảnh cần sửa. |
| Các luồng xử lý:   1. Nhấn vào ảnh và thực hiện [UC\_28](#_Cắt_ảnh). Kết thúc chức năng. |

### Xóa ảnh

|  |
| --- |
| **Mã số: UC\_23** |
| Tên Use Case: Xóa ảnh. |
| Mô tả: Cung cấp chức năng xóa ảnh. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng quản lý ảnh. |
| Cách kích hoạt: Nhấn nút đỏ sau khi vuốt ảnh sang trái. |
| Các luồng xử lý:   1. Xóa ảnh được chọn**.** Kết thúc chức năng. |

### Tải ảnh lên server

|  |
| --- |
| **Mã số: UC\_24** |
| Tên Use Case: Tải ảnh lên server. |
| Mô tả: Cung cấp chức năng tải ảnh lên server. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng quản lý ảnh. |
| Cách kích hoạt: Nhấn nút xanh sau khi vuốt ảnh sang trái. |
| Các luồng xử lý:   1. Tải ảnh được chọn lên server**.** 2. Nếu tải ảnh thành công sẽ hiện dấu check sau ảnh. Kết thúc chức năng. 3. Nếu tải ảnh không thành công sẽ xuất hiện thông báo lỗi. Kết thúc chức năng. |

### Chia sẻ

|  |
| --- |
| **Mã số: UC\_25** |
| Tên Use Case: Chia sẻ. |
| Mô tả: Cung cấp chức năng chia sẻ ảnh dưới định dạng ảnh hoặc PDF. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng quản lý ảnh hoặc đã lưu ảnh trong chức năng chụp ảnh. |
| Cách kích hoạt: Nhấn nút vàng sau khi vuốt ảnh sang trái trong trang quản lý ảnh hoặc nhấn nút “Xong” trong trang xử lý ảnh. |
| Các luồng xử lý:   1. Hiển thịnút “←”, tên trang, ảnh, nút “Chia sẻ ảnh” và nút “Chia sẻ PDF”. 2. Nếu vuốt màn hình sang phải sẽ hiển thị thanh menu trái gồm các nút “Lưu trữ”, “Chứng minh thư”, “Hộ chiếu”, “Tài liệu”, “Về chúng tôi” và “Đăng xuất”: 3. Nếu nhấn nút “Chứng minh thư” thì thực hiện [UC\_40](#_Chụp_chứng_minh). Kết thúc chức năng. 4. Nếu nhấn nút “Hộ chiếu” thì thực hiện [UC\_43](#_Chụp_hộ_chiếu). Kết thúc chức năng. 5. Nếu nhấn nút “Tài liệu” thì thực hiện [UC\_44](#_Chụp_tài_liệu). Kết thúc chức năng. 6. Nếu nhấn nút “Về chúng tôi” thì chuyển sang trang giới thiệu về ứng dụng. Kết thúc chức năng. 7. Nếu nhấn nút “Đăng xuất” thì thực hiện [UC\_08](#_3.2.5_Đăng_xuất). Kết thúc chức năng. 8. Nếu nhấn nút“Chia sẻ ảnh” thì thực hiện [UC\_26](#_Chia_sẻ_ảnh). Kết thúc chức năng. 9. Nếu nhấn nút“Chia sẻ PDF” thì thực hiện [UC\_27](#_Chia_sẻ_PDF). Kết thúc chức năng. 10. Nếu nhấn nút “←” thì quay về trang trước. Kết thúc chức năng. |

### Chia sẻ ảnh

|  |
| --- |
| **Mã số: UC\_26** |
| Tên Use Case: Chia sẻ ảnh. |
| Mô tả: Cung cấp chức năng chia sẻ ảnh dưới định dạng ảnh. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng chia sẻ. |
| Cách kích hoạt: Nhấn nút “Chia sẻ ảnh”. |
| Các luồng xử lý:   1. Hiển thịbảng chọn ứng dụng để chia sẻ gồm những ứng dụng có chức năng chia sẻ có trong điện thoại. 2. Chọn ứng dụng để chia sẻ và được chuyển sang chức năng chia sẻ trên ứng dụng đó với nội dung chia sẻ là ảnh đã chọn: 3. Nếu thực hiện chia sẻ thì chia sẻ hình ảnh hoàn tất. Kết thúc chức năng. 4. Nếu nhấn nút “←” thì quay về trang “Chia sẻ”. Kết thúc chức năng. |

### Chia sẻ PDF

|  |
| --- |
| **Mã số: UC\_27** |
| Tên Use Case: Chia sẻ PDF. |
| Mô tả: Cung cấp chức năng chia sẻ ảnh dưới định dạng PDF. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng chia sẻ. |
| Cách kích hoạt: Nhấn nút “Chia sẻ PDF”. |
| Các luồng xử lý:   1. Hiển thịtệp bằng ứng dụng đọc tệp PDF của điện thoại, nút “←”, nút “Tìm kiếm”, nút “Lưu vào Drive” và nút “Tùy chọn”. 2. Nếu nhấn nút “←” thì quay về trang “Chia sẻ”. Kết thúc chức năng. 3. Nếu nhấn nút “Tìm kiếm” thì hiển thị khung nhập nội dung cần tìm, nút “←”: 4. Nếu nhập nội dung vào khung thì thông báo lỗi không tìm thấy kết quả. 5. Nếu nhấn nút “←” thì thoát khỏi khung tìm kiếm. 6. Nếu nhấn nút “Lưu vào Drive” thì hiển thị khung nhập tên tài liệu, khung chọn tài khoản Drive, thư mục sẽ lưu và hai nút “Hủy”, “Lưu”: 7. Nếu nhập xong thông tin để lưu tệp và nhấn nút “Lưu” thì lưu ảnh thành công và quay lại trang “Chia sẻ PDF”. 8. Nếu nhấn nút “Hủy” thì quay lại trang “Chia sẻ PDF”. 9. Nếu nhấn nút “Tùy chọn” thì hiển thị năm lựa chọn là “Gởi tệp”, “Mở tệp với”, “Tải”, “In” và “Báo cáo vấn đề”: 10. Nếu nhấn nút “Gởi tệp” thì hiển thị các ứng dụng có thể gởi tệp PDF có trong điện thoại:  * Nếu chọn một ứng dụng thì sẽ được chuyển sang ứng dụng đó và thực hiện chức năng chia sẻ của ứng dụng. Sau khi chia sẻ thành công sẽ quay về ứng dụng. * Nếu không chia sẻ thì thoát khỏi ứng dụng được chuyển và mở lại ứng dụng scanner.  1. Nếu nhấn nút “Mở tệp với” thì tệp PDF được mở bằng ứng dụng đọc PDF của điện thoại. 2. Nếu nhấn nút “Tải” thì tệp PDF được tải về điện thoại. 3. Nếu nhấn nút “In” thì tệp PDF được hiển thị để chuẩn bị in tệp. 4. Nếu nhấn nút “Báo cáo vấn đề” thì chuyển sang trang “Phản hồi” gồm khung chọn tài khoản Drive, khung nhập phản hồi, ảnh chụp tệp PDF và các điều khoản, nút “Gửi”, nút “←”:  * Nếu nhấn nút “Gửi” thì thực hiện gửi phản hồi. * Nếu nhấn nút “←” thì quay lại trang “Chia sẻ PDF”. |

### Cắt ảnh

|  |
| --- |
| **Mã số: UC\_28** |
| Tên Use Case: Cắt ảnh. |
| Mô tả: Cung cấp chức năng cắt ảnh. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đang thực hiện sửa ảnh hoặc thêm ảnh ngoài ứng dụng hoặc chụp ảnh. |
| Cách kích hoạt: Nhấn trực tiếp vào ảnh đối với chức năng sửa ảnh hoặc nhấn chọn ảnh đối với chức năng thêm ảnh hoặc nhấn nút chụp đối với chức năng chụp ảnh. |
| Các luồng xử lý:   1. Hiển thị nút “←” , tên trang, ảnh cần cắt, đường viền bao quanh đối tượng sẽ cắt nếu nhận dạng được hình chữ nhật, nút “Cắt” và nút “Xong. 2. Nếu vuốt màn hình sang phải sẽ hiển thị thanh menu trái gồm các nút “Lưu trữ”, “Chứng minh thư”, “Hộ chiếu”, “Tài liệu”, “Về chúng tôi” và “Đăng xuất”: 3. Nếu nhấn nút “Chứng minh thư” thì thực hiện [UC\_40](#_Chụp_chứng_minh). Kết thúc chức năng. 4. Nếu nhấn nút “Hộ chiếu” thì thực hiện [UC\_43](#_Chụp_hộ_chiếu). Kết thúc chức năng. 5. Nếu nhấn nút “Tài liệu” thì thực hiện [UC\_44](#_Chụp_tài_liệu). Kết thúc chức năng. 6. Nếu nhấn nút “Về chúng tôi” thì chuyển sang trang giới thiệu về ứng dụng. Kết thúc chức năng. 7. Nếu nhấn nút “Đăng xuất” thì thực hiện [UC\_08](#_3.2.5_Đăng_xuất). Kết thúc chức năng. 8. Nếu nhấn nút “←” thì quay về trang trước. Kết thúc chức năng. 9. Nếu nhấn nút “Cắt” thì cắt ảnh theo đường viền đã vẽ và trả về ảnh đã cắt. 10. Nếu nhấn nút “Xong” thì thực hiện [UC\_29](#_Xử_lý_ảnh). Kết thúc chức năng. |

### Xử lý ảnh

|  |
| --- |
| **Mã số: UC\_29** |
| Tên Use Case: Xử lý ảnh. |
| Mô tả: Cung cấp các chức năng xử lý ảnh như lọc ảnh, xoay trái, xoay phải, đặt tên ảnh và lưu ảnh. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện cắt ảnh. |
| Cách kích hoạt: Nhấn nút “Xong” trong trang “Cắt ảnh”. |
| Các luồng xử lý:   1. Hiển thị nút “←” , khung nhập tên ảnh với giá trị mặc định là thời gian hiện tại, ảnh cần xử lý, nút “Lọc”, nút “Trái”, nút “Phải” và nút “Xong. 2. Nếu vuốt màn hình sang phải sẽ hiển thị thanh menu trái gồm các nút “Lưu trữ”, “Chứng minh thư”, “Hộ chiếu”, “Tài liệu”, “Về chúng tôi” và “Đăng xuất”: 3. Nếu nhấn nút “Chứng minh thư” thì thực hiện [UC\_40](#_Chụp_chứng_minh). Kết thúc chức năng. 4. Nếu nhấn nút “Hộ chiếu” thì thực hiện [UC\_43](#_Chụp_hộ_chiếu). Kết thúc chức năng. 5. Nếu nhấn nút “Tài liệu” thì thực hiện [UC\_44](#_Chụp_tài_liệu). Kết thúc chức năng. 6. Nếu nhấn nút “Về chúng tôi” thì chuyển sang trang giới thiệu về ứng dụng. Kết thúc chức năng. 7. Nếu nhấn nút “←” thì quay về trang trước. Kết thúc chức năng. 8. Nếu nhập tên ảnh thì thực hiện [UC\_30](#_Đặt_tên). 9. Nếu nhấn nút “Lọc” thì thực hiện [UC\_34](#_Chỉnh_màu). 10. Nếu nhấn nút “Trái” thì thực hiện [UC\_32](#_Xoay_trái). 11. Nếu nhấn nút “Phải” thì thực hiện [UC\_33](#_Xoay_phải). 12. Nếu nhấn nút “Xong” thì thực hiện [UC\_31](#_Lưu_ảnh). Kết thúc chức năng. |

### Đặt tên

|  |
| --- |
| **Mã số: UC\_30** |
| Tên Use Case: Đặt tên. |
| Mô tả: Cung cấp chức năng đặt tên cho ảnh. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện xử lý ảnh. |
| Cách kích hoạt: Nhấn vào khung tên ảnh trong trang “Xử lý ảnh”. |
| Các luồng xử lý:   1. Nếu giữ nguyên thì tên được lưu là thời gian hiện tại. 2. Nếu nhập vào khung tên ảnh thì tên mới được lưu. |

### Lưu ảnh

|  |
| --- |
| **Mã số: UC\_31** |
| Tên Use Case: Lưu ảnh. |
| Mô tả: Cung cấp chức năng lưu ảnh vào ứng dụng. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện xử lý ảnh. |
| Cách kích hoạt: Nhấn nút “Xong” trong trang “Xử lý ảnh”. |
| Các luồng xử lý:   1. Đối với chức năng “Sửa ảnh” và “Thêm ảnh ngoài ứng dụng”thì thực hiện [UC\_16](#_Quản_lý_ảnh). Kết thúc chức năng. 2. Đối với chức năng “Chụp ảnh” thì thực hiện [UC\_25](#_Chia_sẻ). Kết thúc chức năng. |

### Xoay trái

|  |
| --- |
| **Mã số: UC\_32** |
| Tên Use Case: Xoay trái. |
| Mô tả: Cung cấp chức năng xoay ảnh sang trái. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện xử lý ảnh. |
| Cách kích hoạt: Nhấn nút “Trái” trong trang “Xử lý ảnh”. |
| Các luồng xử lý:   1. Ảnh đang hiển thị sẽ được xoay sang trái. Kết thúc chức năng. |

### Xoay phải

|  |
| --- |
| **Mã số: UC\_33** |
| Tên Use Case: Xoay phải. |
| Mô tả: Cung cấp chức năng xoay ảnh sang phải. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện xử lý ảnh. |
| Cách kích hoạt: Nhấn nút “Phải” trong trang “Xử lý ảnh”. |
| Các luồng xử lý:   1. Ảnh đang hiển thị sẽ được xoay sang phải. Kết thúc chức năng. |

### Chỉnh màu

|  |
| --- |
| **Mã số: UC\_34** |
| Tên Use Case: Chỉnh màu. |
| Mô tả: Cung cấp các chức năng chỉnh màu ảnh như chỉnh màu gốc, chỉnh sáng, chỉnh màu xám, chỉnh màu trắng đen. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện xử lý ảnh. |
| Cách kích hoạt: Nhấn nút “Lọc” trong trang “Xử lý ảnh”. |
| Các luồng xử lý:   1. Hiển thị khung với bốn ảnh ví dụ về bốn màu ảnh sẽ được chỉnh, nút “Gốc”, nút “Sáng”, nút “Xám” và nút “Trắng đen” với màu ảnh mặc định là màu gốc. 2. Nếu nhấn nút “Gốc” thì ảnh sẽ hiển thị màu gốc. 3. Nếu nhấn nút “Sáng” thì ảnh sẽ được làm sáng. 4. Nếu nhấn nút “Xám” thì ảnh sẽ chuyển sang màu xám. 5. Nếu nhấn nút “Trắng đen” thì ảnh sẽ chuyển sang màu trắng đen. 6. Nếu nhấn giữ nút “Lọc” thì khung chỉnh màu sẽ được đóng lại. Kết thúc chức năng. |

### Chỉnh màu gốc

|  |
| --- |
| **Mã số: UC\_35** |
| Tên Use Case: Chỉnh màu gốc. |
| Mô tả: Cung cấp chức năng chỉnh màu ảnh thành màu gốc. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chỉnh màu. |
| Cách kích hoạt: Nhấn nút “Gốc” trong khung chỉnh màu. |
| Các luồng xử lý:   1. Ảnh sẽ được hiển thị với màu ảnh gốc. Kết thúc chức năng. |

### Chỉnh sáng

|  |
| --- |
| **Mã số: UC\_36** |
| Tên Use Case: Chỉnh sáng. |
| Mô tả: Cung cấp chức năng chỉnh sáng ảnh. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chỉnh màu. |
| Cách kích hoạt: Nhấn nút “Sáng” trong khung chỉnh màu. |
| Các luồng xử lý:   1. Ảnh sẽ được nâng độ sáng lên. Kết thúc chức năng. |

### Chỉnh màu xám

|  |
| --- |
| **Mã số: UC\_37** |
| Tên Use Case: Chỉnh màu xám. |
| Mô tả: Cung cấp chức năng chỉnh ảnh sang màu xám. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chỉnh màu. |
| Cách kích hoạt: Nhấn nút “Xám” trong khung chỉnh màu. |
| Các luồng xử lý:   1. Ảnh sẽ được chuyển sang màu xám. Kết thúc chức năng. |

### Chỉnh màu trắng đen

|  |
| --- |
| **Mã số: UC\_38** |
| Tên Use Case: Chỉnh màu trắng đen. |
| Mô tả: Cung cấp chức năng chỉnh ảnh sang màu trắng đen. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chỉnh màu. |
| Cách kích hoạt: Nhấn nút “Trắng đen” trong khung chỉnh màu. |
| Các luồng xử lý:   1. Ảnh sẽ được chuyển sang màu trắng đen. Kết thúc chức năng. |

### Chụp ảnh

|  |
| --- |
| **Mã số: UC\_39** |
| Tên Use Case: Chụp ảnh. |
| Mô tả: Cung cấp các chức năng chụp ảnh như chụp “Chứng minh thư”, “Hộ chiếu” và “Tài liệu”. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã đăng nhập vào ứng dụng. |
| Cách kích hoạt: Nhấn nút “Chứng minh thư” hoặc “Hộ chiếu” hoặc “Tài liệu” trong thanh menu trái. |
| Các luồng xử lý:   1. Nếu nhấn nút “Chứng minh thư” thì thực hiện [UC\_40](#_Chụp_chứng_minh). Kết thúc chức năng. 2. Nếu nhấn nút “Hộ chiếu” thì thực hiện [UC\_43](#_Chụp_hộ_chiếu). Kết thúc chức năng. 3. Nếu nhấn nút “Tài liệu” thì thực hiện [UC\_44](#_Chụp_tài_liệu). Kết thúc chức năng. |

### Chụp chứng minh thư

|  |
| --- |
| **Mã số: UC\_40** |
| Tên Use Case: Chụp chứng minh thư. |
| Mô tả: Cung cấp chức năng chụp ảnh mặt trước và mặt sau của chứng minh thư. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng “Chụp ảnh”. |
| Cách kích hoạt: Nhấn nút “Chứng minh thư” trong thanh menu trái. |
| Các luồng xử lý:   1. Hiển thị nút “≡”, tên trang (“Mặt trước” hoặc “Mặt sau”), nút flash, khung chụp, khung mô tả và nút chụp. 2. Nếu nhấn nút “≡” hoặc vuốt màn hình sang phải sẽ hiển thị thanh menu trái gồm các nút “Lưu trữ”, “Chứng minh thư”, “Hộ chiếu”, “Tài liệu”, “Về chúng tôi” và “Đăng xuất”: 3. Nếu nhấn nút “Chứng minh thư” thì thực hiện [UC\_40](#_Chụp_chứng_minh). Kết thúc chức năng. 4. Nếu nhấn nút “Hộ chiếu” thì thực hiện [UC\_43](#_Chụp_hộ_chiếu). Kết thúc chức năng. 5. Nếu nhấn nút “Tài liệu” thì thực hiện [UC\_44](#_Chụp_tài_liệu). Kết thúc chức năng. 6. Nếu nhấn nút “Về chúng tôi” thì chuyển sang trang giới thiệu về ứng dụng. Kết thúc chức năng. 7. Nếu nhấn nút flash thì bật flash, nhấn thêm lần nữa để tắt flash. 8. Nếu nhấn nút chụp thì thực hiện [UC\_41](#_Chụp_mặt_trước). Kết thúc chức năng. |

### Chụp mặt trước

|  |
| --- |
| **Mã số: UC\_41** |
| Tên Use Case: Chụp mặt trước. |
| Mô tả: Cung cấp chức năng chụp mặt trước của chứng minh thư. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng “Chụp chứng minh thư”. |
| Cách kích hoạt: Nhấn nút “Chụp” trong trang “Mặt trước”. |
| Các luồng xử lý:   1. Chụp ảnh hiển thị trong khung và thực hiện [UC\_28](#_Cắt_ảnh). Kết thúc chức năng. |

### Chụp mặt sau

|  |
| --- |
| **Mã số: UC\_42** |
| Tên Use Case: Chụp mặt sau. |
| Mô tả: Cung cấp chức năng chụp mặt sau của chứng minh thư. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng “Chụp mặt trước”. |
| Cách kích hoạt: Nhấn nút “Chụp” trong trang “Mặt sau”. |
| Các luồng xử lý:   1. Chụp ảnh hiển thị trong khung và thực hiện [UC\_28](#_Cắt_ảnh). Kết thúc chức năng. |

### Chụp hộ chiếu

|  |
| --- |
| **Mã số: UC\_43** |
| Tên Use Case: Chụp hộ chiếu. |
| Mô tả: Cung cấp chức năng chụp hộ chiếu. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng “Chụp ảnh”. |
| Cách kích hoạt: Nhấn nút “Chứng minh thư” trong thanh menu trái. |
| Các luồng xử lý:   1. Hiển thị nút “≡”, tên trang “Hộ chiếu”, nút flash, khung chụp, khung mô tả và nút chụp. 2. Nếu nhấn nút “≡” hoặc vuốt màn hình sang phải sẽ hiển thị thanh menu trái gồm các nút “Lưu trữ”, “Chứng minh thư”, “Hộ chiếu”, “Tài liệu”, “Về chúng tôi” và “Đăng xuất”: 3. Nếu nhấn nút “Chứng minh thư” thì thực hiện [UC\_40](#_Chụp_chứng_minh). Kết thúc chức năng. 4. Nếu nhấn nút “Hộ chiếu” thì thực hiện [UC\_43](#_Chụp_hộ_chiếu). Kết thúc chức năng. 5. Nếu nhấn nút “Tài liệu” thì thực hiện [UC\_44](#_Chụp_tài_liệu). Kết thúc chức năng. 6. Nếu nhấn nút “Về chúng tôi” thì chuyển sang trang giới thiệu về ứng dụng. Kết thúc chức năng. 7. Nếu nhấn nút flash thì bật flash, nhấn thêm lần nữa để tắt flash. 8. Nếu nhấn nút chụp thì chụp ảnh hiển thị trong khung và thực hiện [UC\_28](#_Cắt_ảnh). Kết thúc chức năng. |

### Chụp tài liệu

|  |
| --- |
| **Mã số: UC\_44** |
| Tên Use Case: Chụp tài liệu. |
| Mô tả: Cung cấp chức năng chụp tài liệu. |
| Actor chính: Người dùng. |
| Tiền điều kiện: Đã thực hiện chức năng “Chụp ảnh”. |
| Cách kích hoạt: Nhấn nút “Tài liệu” trong thanh menu trái. |
| Các luồng xử lý:   1. Hiển thị nút “≡”, tên trang “Hộ chiếu”, nút flash, khung chụp và nút chụp. 2. Nếu nhấn nút “≡” hoặc vuốt màn hình sang phải sẽ hiển thị thanh menu trái gồm các nút “Lưu trữ”, “Chứng minh thư”, “Hộ chiếu”, “Tài liệu”, “Về chúng tôi” và “Đăng xuất”: 3. Nếu nhấn nút “Chứng minh thư” thì thực hiện [UC\_40](#_Chụp_chứng_minh). Kết thúc chức năng. 4. Nếu nhấn nút “Hộ chiếu” thì thực hiện [UC\_43](#_Chụp_hộ_chiếu). Kết thúc chức năng. 5. Nếu nhấn nút “Tài liệu” thì thực hiện [UC\_44](#_Chụp_tài_liệu). Kết thúc chức năng. 6. Nếu nhấn nút “Về chúng tôi” thì chuyển sang trang giới thiệu về ứng dụng. Kết thúc chức năng. 7. Nếu nhấn nút flash thì bật flash, nhấn thêm lần nữa để tắt flash. 8. Nếu nhấn nút chụp thì chụp ảnh hiển thị trong khung và thực hiện [UC\_28](#_Cắt_ảnh). Kết thúc chức năng. |

## Mô hình dữ liệu



Mô hình CDM



Mô hình PDM

## Mô tả các bảng dữ liệu

### Cơ sở dữ liệu

Lưu tên CSDL là tài khoản email hoặc số điện thoại dùng để đăng nhập vào ứng dụng được lưu trên Firebase. Mỗi CSDL lấy tên CSDL làm khóa chính.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **COSODULIEU** | | | | | | | |
| **STT** | Tên trường | Kiểu | Kích  thước | Khóa  chính | Khóa  ngoại | Not  Null | Mô tả |
| **1** | TenCSDL | int |  | x |  | x | Tên cơ sở dữ liệu. |

Bảng 1. Cơ sở dữ liệu

### Thư mục

Lưu trữ các thông tin của thư mục được tạo ra trong ứng dụng như: tên thư mục, ngày tạo, loại thư mục và trạng thái hiển thị. Mỗi thư mục lấy một mã thư mục làm khóa chính, đồng thời tham chiếu đến tên CSDL trong bảng CSDL.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **THUMUC** | | | | | | | |
| **STT** | Tên trường | Kiểu | Kích  thước | Khóa  chính | Khóa  ngoại | Not  Null | Mô tả |
| **1** | MaThuMuc | int |  | x |  | x | Mã thư mục. |
| **2** | TenCSDL | int |  |  | x | x | Tên cơ sở dữ liệu. |
| **3** | TenThuMuc | text |  |  |  | x | Tên thư mục. |
| **4** | NgayTaoThuMuc | text |  |  |  |  | Ngày tạo thư mục. |
| **5** | LoaiThuMuc | text |  |  |  |  | Loại thư mục. |
| **6** | HienThi | smallint |  |  |  |  | Hiển thị. |

Bảng 2. Thư mục

### Hình ảnh

Lưu trữ các thông tin của hình ảnh được tạo ra trong mỗi thư mục như: tên hình ảnh, ngày tạo, đường dẫn lưu ảnh, hình ảnh đã được mã hóa thành chuỗi base64, loại hình ảnh và trạng thái tải ảnh lên server. Mỗi hình ảnh lấy một mã hình ảnh làm khóa chính, đồng thời tham chiếu đến mã thư mục trong bảng thư mục và tên CSDL trong bảng CSDL.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **HINHANH** | | | | | | | |
| **STT** | Tên trường | Kiểu | Kích  thước | Khóa  chính | Khóa  ngoại | Not  Null | Mô tả |
| **1** | MaHinhAnh | int |  | x |  | x | Mã hình ảnh. |
| **2** | MaThuMuc | int |  |  | x | x | Mã thư mục. |
| **3** | TenCSDL | int |  |  | x | x | Tên cơ sở dữ liệu. |
| **4** | TenHinhAnh | text |  |  |  | x | Tên hình ảnh. |
| **5** | NgayTaoHinhAnh | text |  |  |  |  | Ngày tạo hình ảnh. |
| **6** | DuongDan | text |  |  |  |  | Đường dẫn. |
| **7** | Base64 | text |  |  |  |  | Base64. |
| **8** | LoaiHinhAnh | text |  |  |  |  | Loại hình ảnh. |
| **9** | TaiLen | smallint |  |  |  |  | Tải lên. |

Bảng 3. Hình ảnh

# CHƯƠNG 4: MÔ TẢ GIAO DIỆN

Chương này sẽ mô tả bố cục các giao diện chính của hệ thống cũng như thể hiện quá trình thực thi các chức năng chính lên các giao diện này.

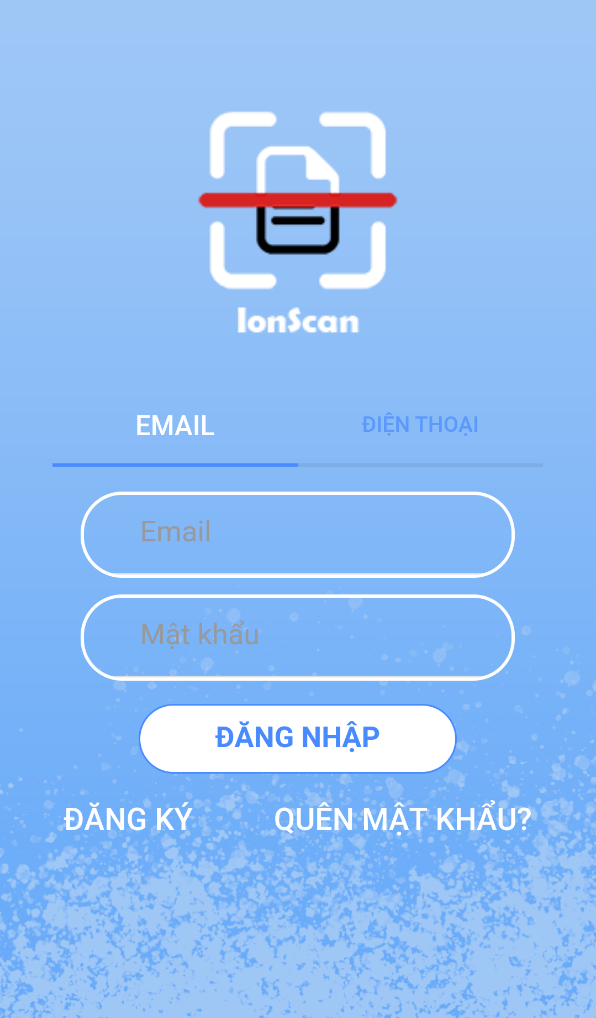


## Giao diện đăng nhập

Giao diện đăng nhập bao gồm các chức năng: Đăng nhập bằng email, số điện thoại, đăng ký tài khoản hoặc đặt lại mật khẩu.

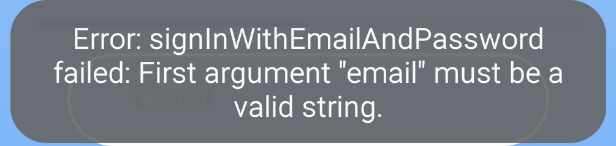
### Đăng nhập bằng email

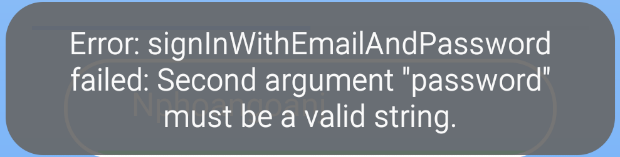
Sau khi mở ứng dụng, tab đăng nhập bằng email sẽ được hiển thị mặc định.

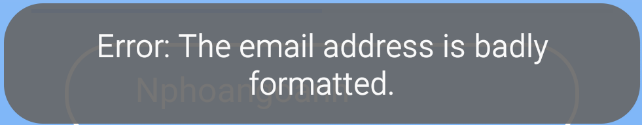


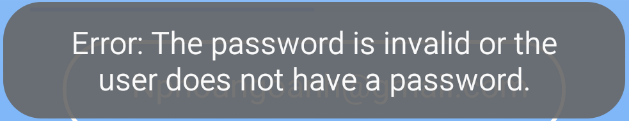
Giao diện đăng nhập bằng email

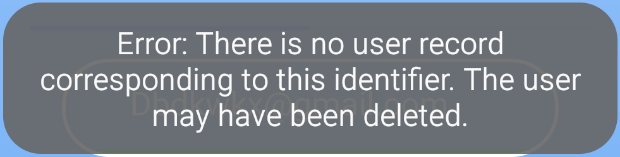
Nếu nhập sai sẽ xuất hiện những thông báo lỗi như sau:

 Lỗi khi chưa nhập email

 Lỗi khi chưa nhập mật khẩu

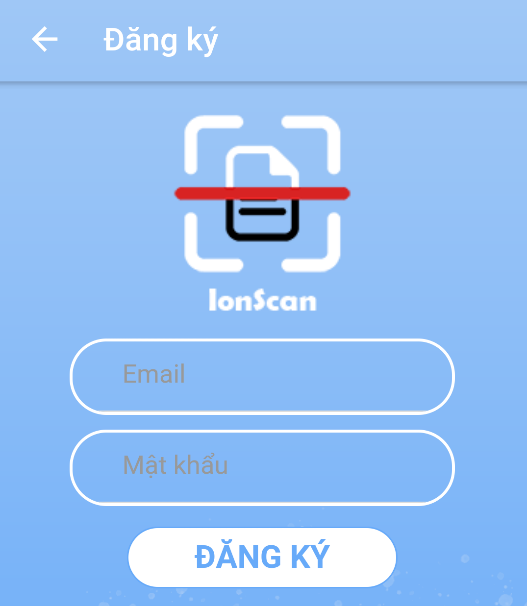
 Lỗi khi nhập sai định dạng email

 Lỗi khi nhập sai mật khẩu đã đăng ký

 Lỗi khi nhập email chưa đăng ký

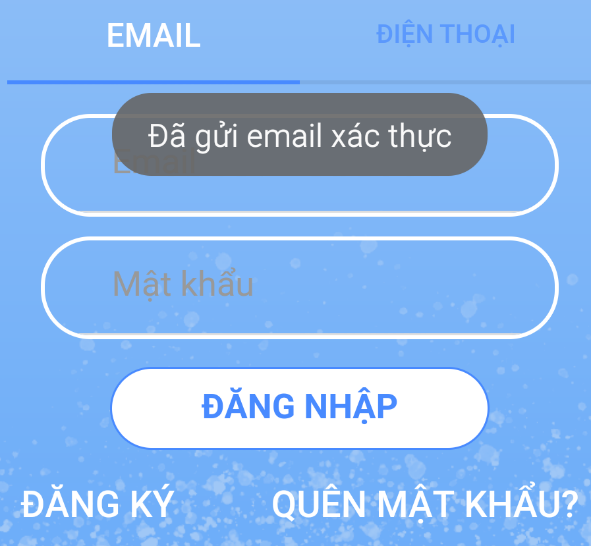
### Đăng ký

Nhấn vào nút “Đăng ký” để đăng ký tài khoản email, người dùng sẽ được chuyển sang trang đăng ký.



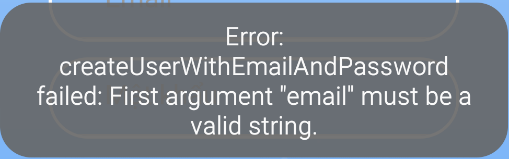
Giao diện đăng ký

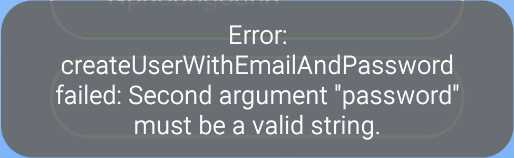
Nếu nhập đúng định dạng email và mật khẩu thì sẽ được trả về thông báo “Đã gửi email xác thực” và trở về trang đăng nhập.

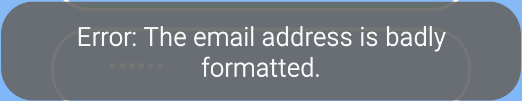


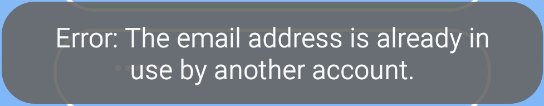
Thông báo đã gửi email xác thực

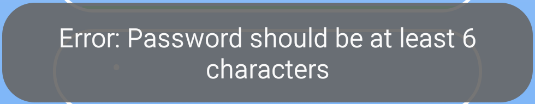
Nếu nhập sai định dạng email và mật khẩu thì xuất hiện thông báo lỗi.

 *Lỗi khi chưa nhập email*

 *Lỗi khi chưa nhập mật khẩu*

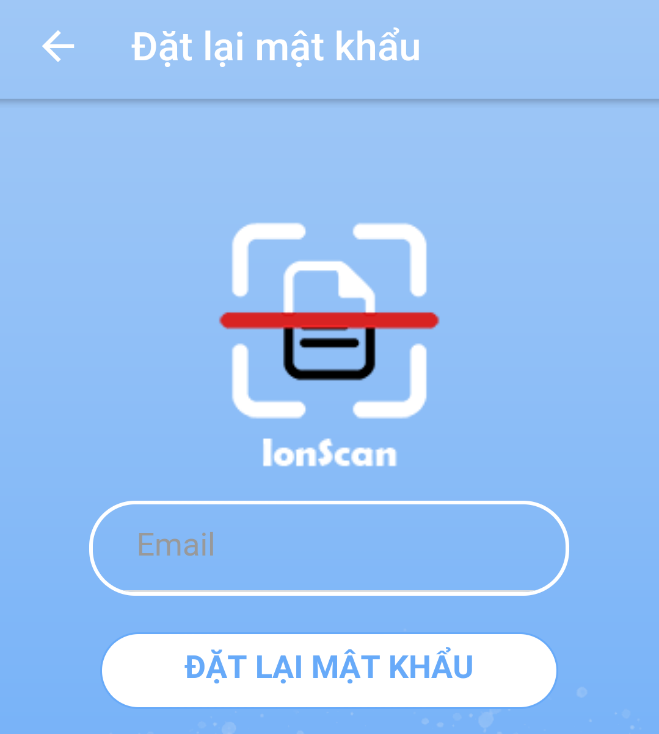
 *Lỗi khi nhập sai định dạng email*

 *Lỗi khi nhập email đã đăng ký*

 *Lỗi khi nhập email ít hơn 6 kí tự*

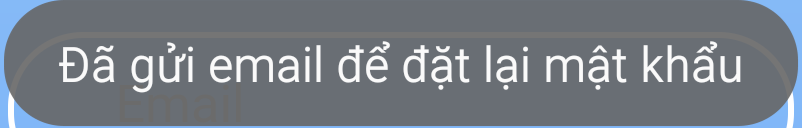
### Đặt lại mật khẩu

Nhấn vào nút “Quên mật khẩu” để đặt lại mật khẩu, người dùng sẽ được chuyển sang trang đặt lại mật khẩu.



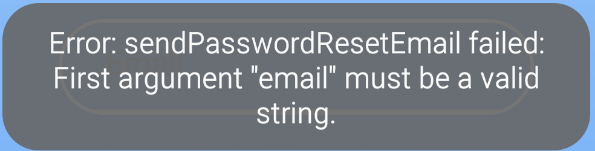
Giao diện đặt lại mật khẩu

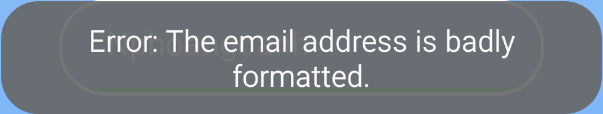
Nếu nhập đúng email đã đăng ký thì sẽ được trả về thông báo “Đã gửi email để đặt lại mật khẩu” và trở về trang đăng nhập.

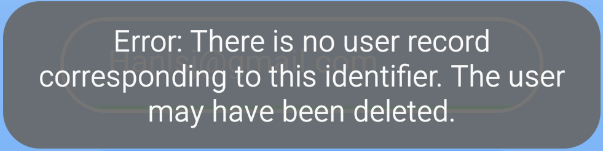


Thông báo đã gửi email để đặt lại mật khẩu

Nếu nhập sai định dạng email và mật khẩu thì xuất hiện thông báo lỗi:

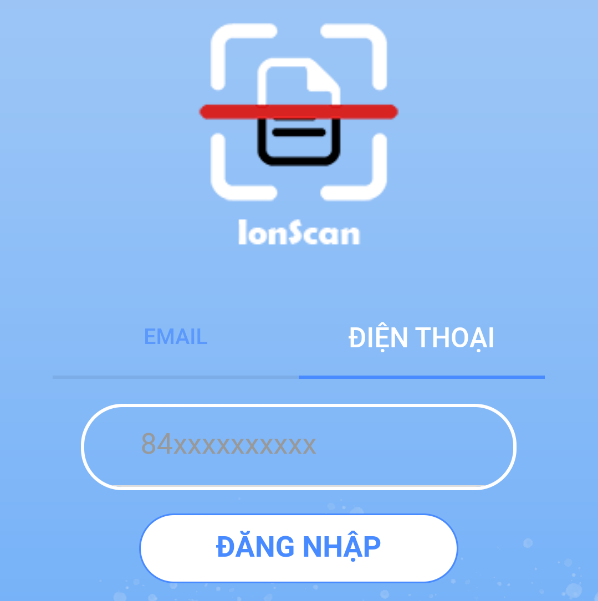
 *Lỗi khi chưa nhập email*

 *Lỗi khi nhập email sai định dạng*

 *Lỗi khi nhập email chưa được đăng ký*

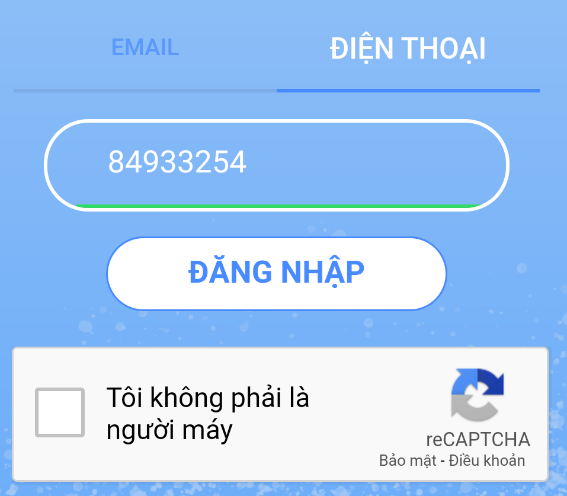
### Đăng nhập bằng số điện thoại

Nhấn vào tab Điện thoại để thực hiện đăng nhập bằng số điện thoại.



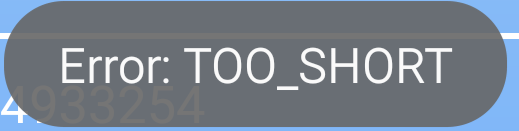
Giao diện đăng nhập bằng số điện thoại

Nếu nhập xong số điện thoại và nhấn nút “Đăng nhập” sẽ xuất hiện khung xác thực recaptcha.

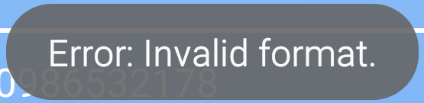


Khung xác thực recaptcha

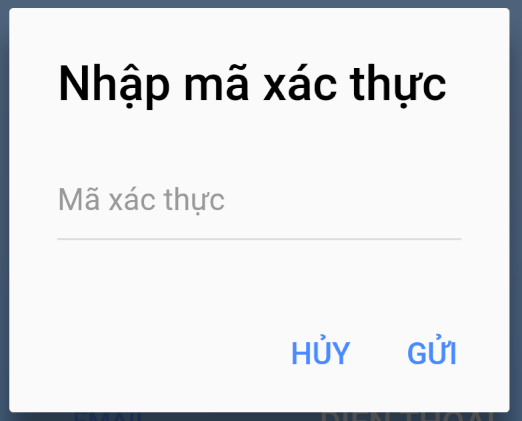
Nếu số điện thoại sai định dạng, quá dài hoặc quá ngắn thì sẽ xuất hiện những thông báo lỗi sau:

 Lỗi số điện thoại quá dài

 Lỗi số điện thoại quá ngắn

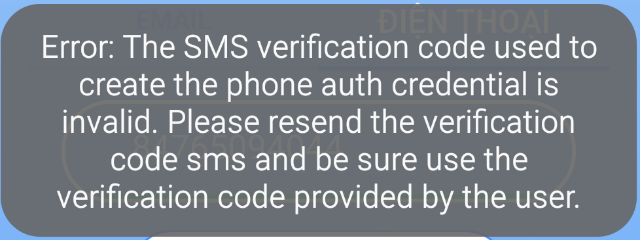
 Lỗi sai định dạng số điện thoại

Nếu nhập đúng thì sẽ xuất hiện thông báo nhập mã xác thực.



Thông báo nhập mã xác thực

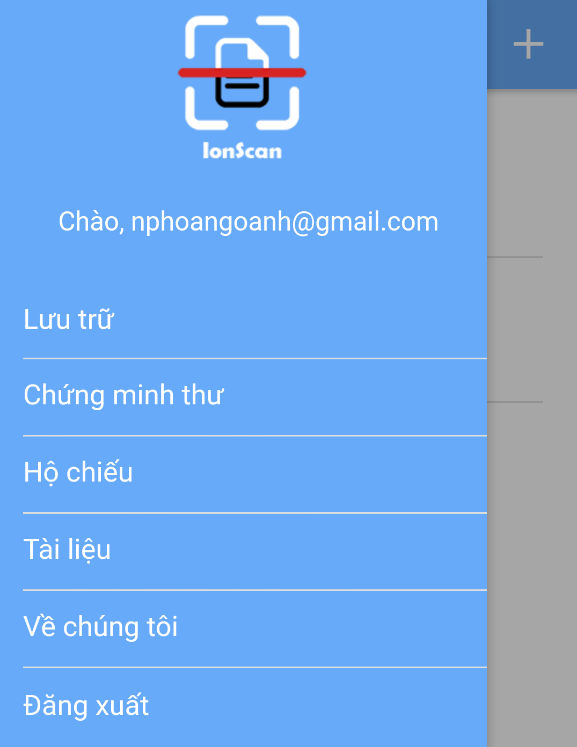
Nếu nhập sai mã xác thực sẽ xuất hiện thông báo lỗi.



Lỗi khi nhập sai mã xác thực

## Giao diện thanh menu trái

Giao diện thanh menu trái bao gồm các chức năng: Hiển thị tài khoản người dùng, trang quản lý thư mục, trang chụp các loại ảnh như chứng minh thư, hộ chiếu, tài liệu, trang giới thiệu về ứng dụng và nút đăng xuất.



Giao diện thanh menu trái

## Giao diện quản lý thư mục

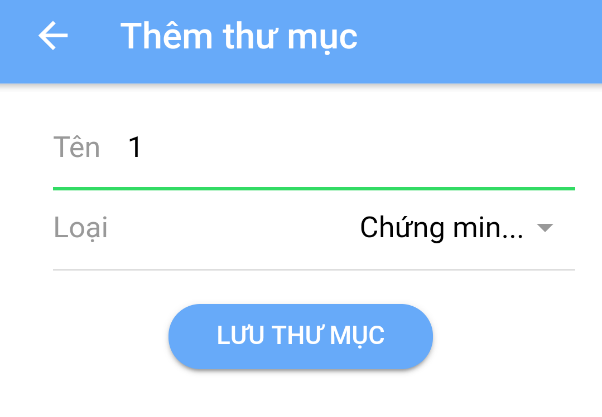
Giao diện quản lý bao gồm các chức năng: Hiển thị và mở thư mục mặc định và cá nhân, thêm, sửa và xóa thư mục cá nhân.



Giao diện quản lý thư mục

### Thêm thư mục

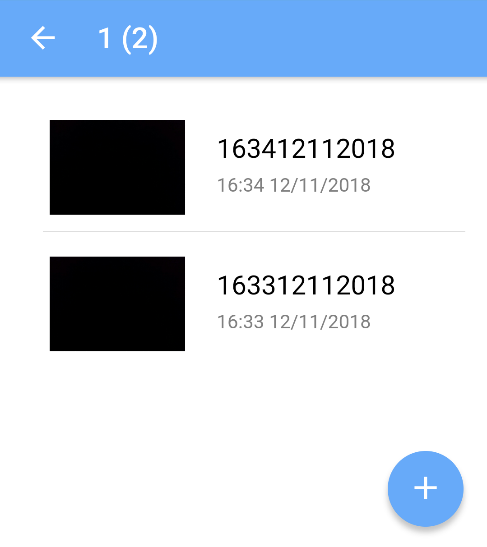
Để thêm thư mục, người dùng nhấn nút “+” và sẽ được chuyển sang trang thêm thư mục.



Thêm thư mục

### Mở thư mục

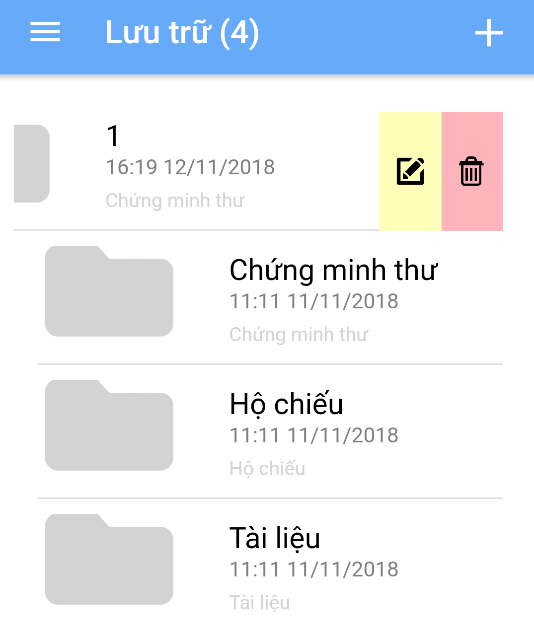
Để mở thư mục, người dùng nhấn trực tiếp vào thư mục cần mở thì sẽ được chuyển vào thư mục đó.



Mở thư mục

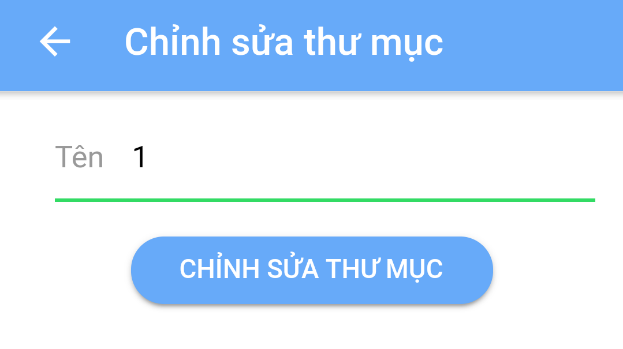
### Sửa và xóa thư mục

Để sửa và xóa thư mục, người dùng vuốt thư mục sang trái. Nhấn nút đỏ để xóa thư mục.



Xóa thư mục

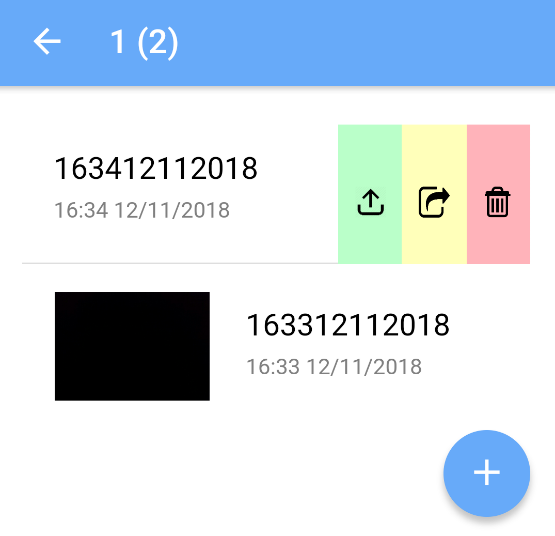
Để sửa thư mục, người dùng nhấn nút vàng và sẽ được chuyển sang trang chỉnh sửa thư mục.



Sửa thư mục

## Giao diện quản lý ảnh

Để mở giao diện quản lý ảnh, thực hiện mở thư mục. Giao diện quản lý ảnh gồm hiển thị ảnh, thêm, xóa, chia sẻ và tải ảnh lên server.



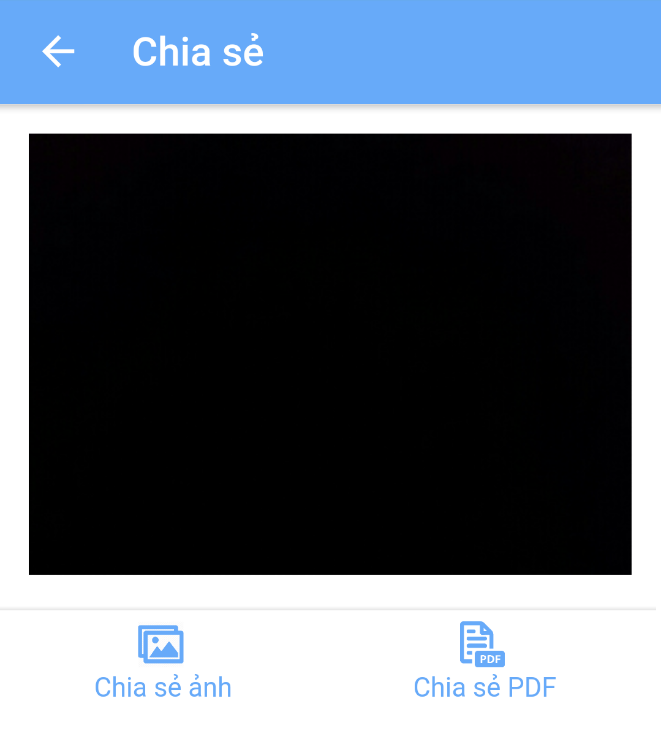
Giao diện quản lý ảnh

### Sửa, xóa, chia sẻ và tải ảnh lên server

Nhấn trực tiếp vào ảnh để sửa ảnh.

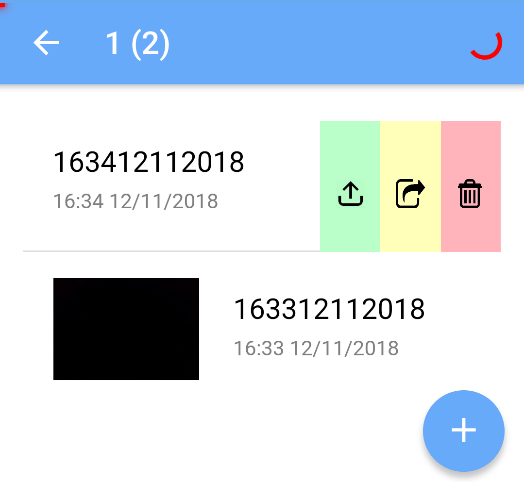
Nhấn nút đỏ để thực hiện xóa ảnh.

Nhấn nút vàng để chia sẻ ảnh dưới dạng ảnh hoặc PDF.

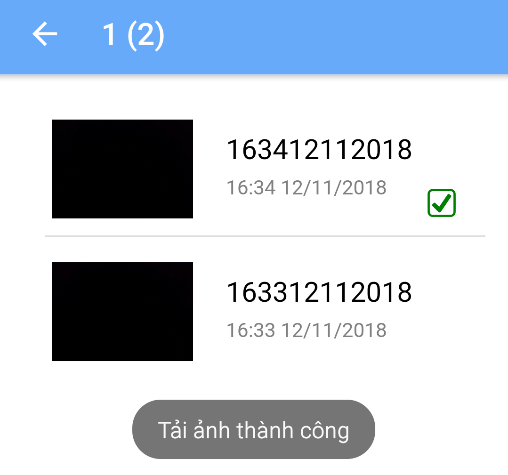


Chia sẻ

Nhấn nút xanh để tải ảnh lên server. Sau khi tải thành công sẽ xuất hiện dấu check.



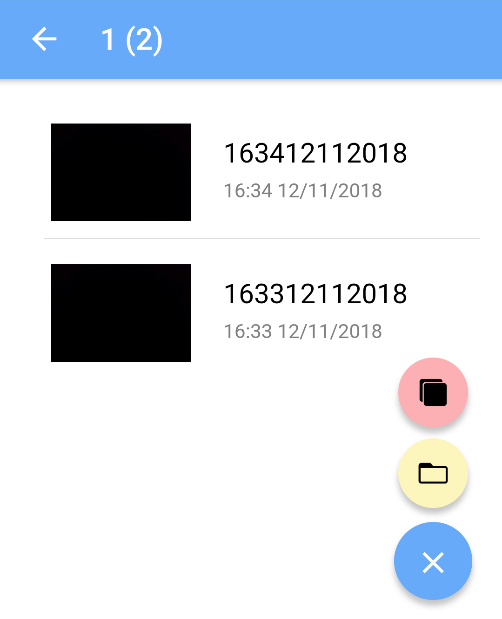
Tải ảnh lên server



Tải ảnh lên server thành công

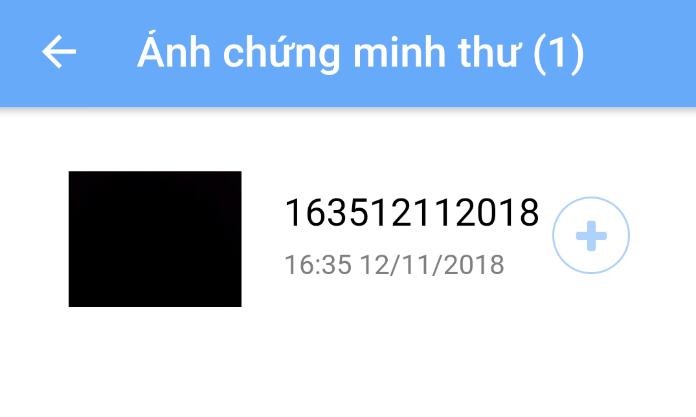
### Thêm ảnh

Mở giao diện thêm ảnh bằng cách nhấn vào nút “+”. Giao diện thêm ảnh gồm các chức năng thêm ảnh từ trong hoặc ngoài ứng dụng.



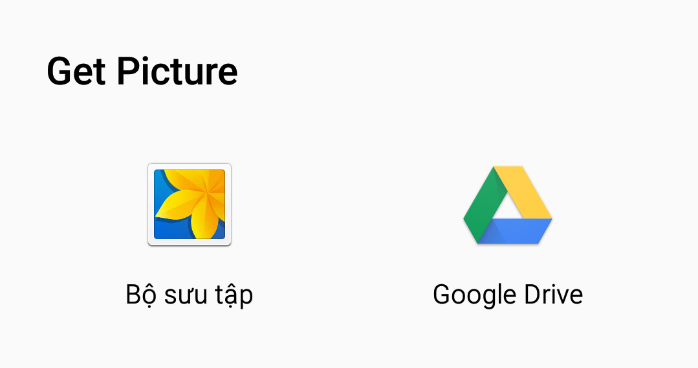
Thêm ảnh

Thêm ảnh trong ứng dụng: Nhấn vào nút vàng để được chuyển đến thư mục mặc định và chọn ảnh bằng cách nhấn nút “+” sau mỗi ảnh.



Thêm ảnh trong ứng dụng

Thêm ảnh ngoài ứng dụng: Nhấn vào nút đỏ để hiển thị chức năng thêm ảnh ngoài ứng dụng gồm thêm ảnh từ Bộ sưu tập hoặc Drive.



Thêm ảnh ngoài ứng dụng

## Giao diện chụp chứng minh thư

Để mở giao diện chụp chứng minh thư, nhấn nút “Chứng minh thư” trong thanh menu trái. Giao diện chụp chứng minh thư gồm giao diện chụp mặt trước và sau của chứng minh thư.

## Giao diện chụp hộ chiếu

## Giao diện chụp tài liệu

## Giao diện cắt ảnh

## Giao diện xử lý ảnh

# CHƯƠNG 5: TỔNG KẾT

* 1. **Kết quả đạt được**
  2. **Hạn chế**
  3. **Hướng phát triển**

# THAM KHẢO

[1] AngularJS: <https://docs.angularjs.org/guide/introduction>

[2] Node.js: <https://www.tutorialspoint.com/nodejs/nodejs_introduction.html>

[3] OpenCV: <https://opencv.org/>

[4] OpenCV.js: <https://docs.opencv.org/3.4/df/d0a/tutorial_js_intro.html>

[5] Apache Cordova: <https://cordova.apache.org/docs/en/latest/guide/overview/>

[6] Firebase: <https://firebase.google.com/docs/auth/>

[7] Ionic: <https://ionicframework.com/docs/intro/concepts>

[8] TypeScript: [https://www.typescriptlang.org/docs/handbook/declaration-tệps/introduction.html](https://www.typescriptlang.org/docs/handbook/declaration-files/introduction.html)

[9] Git: [https://vi.wikipedia.org/wiki/Git\_(phần mềm)](https://vi.wikipedia.org/wiki/Git_(ph%E1%BA%A7n_m%E1%BB%81m))

[10] Github: <https://guides.github.com/activities/hello-world/>