**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI:**

**PHÁT TRIỂN ỨNG DỤNG TÌM KIẾM PHÒNG TRỌ**

**TẠI THÀNH PHỐ HỒ CHÍ MINH**

Giảng viên hướng dẫn : THS. TRẦN PHONG NHÃ

Sinh viên thực hiện : VÕ TẤN ĐÀO

Mã sinh viên : 5951071011

Lớp : CÔNG NGHỆ THÔNG TIN

Khóa : 59

Tp. Hồ Chí Minh, tháng 07 năm 2022

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI:**

**PHÁT TRIỂN ỨNG DỤNG TÌM KIẾM PHÒNG TRỌ**

**TẠI THÀNH PHỐ HỒ CHÍ MINH**

Giảng viên hướng dẫn : THS. TRẦN PHONG NHÃ

Sinh viên thực hiện : VÕ TẤN ĐÀO

Mã sinh viên : 5951071011

Lớp : CÔNG NGHỆ THÔNG TIN

Khóa : 59

Tp. Hồ Chí Minh, tháng 07 năm 202

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

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

# **NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP**

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

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

**Mã sinh viên:** 5951071011 **Họ tên SV:** Võ Tấn Đào

**Khóa:** 59 **Lớp:** CQ.59.CNTT

1. **Tên đề tài**
2. **Mục đích, yêu cầu**
3. ***Mục đích:*** 
   * + Tìm hiểu về công nghệ mới trong lập trình di động Flutter và ngôn ngữ lập trình dart
     + Tìm hiểu ngôn ngữ lập trình node js.
     + Xây dựng app tìm kiếm phòng trọ cho người dân thành phố Hồ Chí Minh sử dụng
     + Sử dụng google map api.
4. ***Yêu cầu***

* Phân tích xây dựng hệ thống
* Xây dựng cơ sở dữ liệu
* Hiểu và sử dụng được ngôn ngữ Node js xây dựng back end
* Hiểu được ngôn ngữ lập trình dart và công nghệ code platform trong flutter
* Áp dụng và triển khai ứng dụng.

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

* Tổng quan về Flutter và ngôn ngữ lập trình Dart
* Tổng quan ngôn ngữ Node js
* Tổng qua về MongoDB
* Tổng quan và áp dụng Google Map Api
  + Phạm vi đề tài:

Phát triển ứng dụng tìm kiếm phòng trọ tại thành phố Hồ Chí Minh.

1. **Công nghệ, công cụ và ngôn ngữ lập trình** 
   * Công cụ lập trình: visual studio code
   * Công cụ xây dựng cơ sở dữ liệu: MongoDB
   * Công nghệ sử dụng: Flutter
   * Ngôn ngữ lập trình: Dart, Node js.
2. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng**

* Quyển báo cáo đề tài tốt nghiệp
* Hiểu được quy trình để xây dựng phát triển một app ứng dụng
* Hiểu được ngôn ngữ lập trình dart và công nghệ flutter.
* Biết được cách thiết kế, phân tích dữ liệu, phân tích thiết kế hệ thống
* Áp dụng để xây dựng, và kiểm thử một app

**6. Giáo viên và cán bộ hướng dẫn**

Họ tên: Ths.Trần Phong Nhã

Đơn vị công tác: Đại học giao thông vận tải phân hiệu tại thành phố Hồ Chí Minh

Điện thoại: Email:

|  |  |
| --- | --- |
| **Ngày tháng 03 năm 2022**  **Trưởng BM Công nghệ Thông tin** | **Đã giao nhiệm vụ TKTN**  **Giáo viên hướng dẫn** |
| **ThS. Trần Phong Nhã** | **Ths. Trần Phong Nhã** |

Đã nhận nhiệm vụ TKTN

Sinh viên: ` Ký tên:

Điện thoại: Email:

**LỜI CẢM ƠN**

Lời nói đầu tiên, em xin gửi tới Quý Thầy Cô Bộ môn Công nghệ Thông tin Trường Đại học Giao thông vận tải phân hiệu tại thành phố Hồ Chí Minh lời chúc sức khỏe và lòng biết hơn sâu sắc.

Em xin chân thành cảm ơn quý thầy cô đã giúp đỡ tạo điều kiện để em hoàn thành đồ án với đề tài “**PHÁT TRIỂN ỨNG DỤNG TÌM KIẾM PHÒNG TRỌ TẠI THÀNH PHỐ HỒ CHÍ MINH**”. Hơn hết em xin cảm ơn thầy Trần Phong Nhã đã nhiệt tình chỉ dạy, tạo điều kiện để em có thể hoàn thành tốt đồ án tốt nghiệp.

Trong thời gian thực hiện đề tài em đã tìm hiểu kiến thức nghiên cứu công nghệ mới để có thể xây dựng ứng dụng, tuy nhiên do thời gian và khả năng còn hạn chế, vì vậy sẽ không thể tránh khỏi các thiếu sót. Rất mong nhận được sự đánh giá và góp ý của quý thầy cô để đề tài của em có thể hoàn thiện hơn.

Lời sau cùng, em xin gửi lời chúc tới Quý Thầy Cô Bộ môn Công nghệ thông tin và hơn hết là thầy Trần Phong Nhã có thật nhiều sức khỏe, có nhiều thành công trong công việc.

Em xin chân thành cảm ơn!

**Tp. Hồ Chí Minh, ngày….tháng 07 năm 2022**

**Sinh viên thực hiên**

**Võ Tấn Đào**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

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

MỤC LỤC

[NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP i](#_Toc106978241)

[DANH MỤC CHỮ VIẾT TẮT vii](#_Toc106978242)

[CHƯƠNG 1 CƠ SỞ LÝ THUYẾT 1](#_Toc106978243)

[1.1 Tổng quan về Ngôn Ngữ Lập Trình Dart 1](#_Toc106978244)

[1.1.1 Giới thiệu về ngôn ngữ 1](#_Toc106978245)

[1.1.2 Lý do chọn ngôn ngữ lập trình Dart 1](#_Toc106978246)

[1.1.3 Tính hướng đối tượng trong Dart 1](#_Toc106978247)

[1.1.4 Ưu điểm của Dart 2](#_Toc106978248)

[1.1.5 Nhược Điểm của Dart 3](#_Toc106978249)

[1.2 Tổng quan về Flutter 3](#_Toc106978250)

[1.2.1 Khái niệm về ngôn ngữ 3](#_Toc106978251)

[1.2.2 Lý do nên sử dụng Flutter 4](#_Toc106978252)

[1.2.2 Tính năng của Flutter 5](#_Toc106978253)

[1.2.3 Ưu Điểm Flutter 5](#_Toc106978254)

[1.2.4 Nhược Điểm Flutter 6](#_Toc106978255)

[1.3 Tổng quan về NodeJS 8](#_Toc106978256)

[1.3.1 Khái niệm 8](#_Toc106978257)

[1.3.2 Ứng dụng của Nodejs 9](#_Toc106978258)

[1.3.3 Ưu điểm 9](#_Toc106978259)

[1.3.4 Nhược điểm 10](#_Toc106978260)

[1.4 Tổng quan về Google Map Api 10](#_Toc106978261)

[1.4.1 Khái niệm 10](#_Toc106978262)

[1.4.2 Một số ứng dụng của Google Map Api. 11](#_Toc106978263)

[1.4.3 Thuật toán tìm đường đi ngắn nhất trong Google Map API 11](#_Toc106978264)

[1.5 Tổng quan MongoDB 14](#_Toc106978265)

[1.5.1 Tìm hiểu MongoDB 14](#_Toc106978266)

[1.5.2 Nguyên tắc hoạt động của MongoDB 17](#_Toc106978267)

[1.5.3 Ưu điểm và hạn chế của MongoDB 18](#_Toc106978268)

[CHƯƠNG 2 PHÂN TÍCH YÊU CẦU CỦA ỨNG DỤNG 20](#_Toc106978269)

[2.1 Mô tả bài toán 20](#_Toc106978270)

[2.2 Phân tích và thiết kế hệ thống 20](#_Toc106978271)

[2.2.1 Sơ đồ Use case 20](#_Toc106978272)

[2.2.2 Sơ đồ phân rã chức năng 22](#_Toc106978273)

[2.2.3 Đặc tả chức năng 22](#_Toc106978274)

[2.2.3 Thiết kế cơ sở dữ liệu 30](#_Toc106978275)

[2.3 Danh sách API 32](#_Toc106978276)

[CHƯƠNG 3 THỬ NGHIỆM VÀ ĐÁNH GIÁ 34](#_Toc106978277)

[3.1 Môi trường thử nghiệm 34](#_Toc106978278)

[3.2 Giao diện chương trình 34](#_Toc106978279)

[KẾT LUẬN VÀ KIẾN NGHỊ 56](#_Toc106978280)

[PHỤ LỤC 58](#_Toc106978281)

[TÀI LIỆU THAM KHẢO 59](#_Toc106978282)

# **DANH MỤC CHỮ VIẾT TẮT**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mô tả** | **Ý nghĩa** | **Ghi chú** |
| 1 | API | Application Programming Interface |  |
| 2 | UI | User Interface |  |
| 3 | IOS | Internetwork Operating System |  |
| 4 | OOP | Object Oriented Programming |  |
| 5 | SDK | Software Development Kit |  |
| 6 | HTTP | Hypertext Transfer Protocol |  |
| 7 | IIS | Microsoft Internet Information Services |  |
| 8 | CSDL | Cơ sở dữ liệu |  |
| 9 | RDBMS | Relational Database Management System |  |

**DANH MỤC BẢNG**

[Bảng 2. 1 Đặc tả usecase”Tìm kiếm nhà trọ 23](#_Toc106966315)

[Bảng 2. 2 Đặc tả usecase “Đăng bài viết” 25](#_Toc106966316)

[Bảng 2. 3 Đặc tả usecase “Đăng kí tài khoản” 26](#_Toc106966317)

[Bảng 2. 4 Đặc tả usecase “Đăng nhập” 26](#_Toc106966318)

[Bảng 2. 5 Đặc tả usecase ”xem thông tin phòng” 27](#_Toc106966319)

[Bảng 2. 6 Đặc tả usecase ”Đăng phòng” 28](#_Toc106966320)

[Bảng 2. 7 Đặc tả usecase “Lưu phòng yêu thích” 29](#_Toc106966321)

[Bảng 2. 8 Mô tả chi tiết bảng Room 31](#_Toc106966322)

[Bảng 2. 9 Mô tả chi tiết bảng User 32](#_Toc106966323)

[Bảng 2. 10 Mô tả chi tiết bảng District 32](#_Toc106966324)

**DANH MỤC HÌNH ẢNH**

[Hình 1. 1 Logo Dart 1](#_Toc106966347)

[Hình 1. 2 Logo Flutter 4](#_Toc106966348)

[Hình 1. 3 Kiến trúc Flutter 7](#_Toc106966349)

[Hình 1. 4 Logo NodeJS 8](#_Toc106966350)

[Hình 1. 5 Minh hoạ thuật toán Dijkstra bước 1 12](#_Toc106966351)

[Hình 1. 6 Minh hoạ thuật toán Dijkstra bước 2 13](#_Toc106966352)

[Hình 1. 7 Hình minh hoạ thuật toán Dijkstra bước 3 13](#_Toc106966353)

[Hình 1. 8 Hình minh hoạ thuật toán Dijkstra bước 4 14](#_Toc106966354)

[Hình 1. 9 Hình mongoDB 14](#_Toc106966355)

[Hình 1. 10 Kiến trúc của MongoDB 17](#_Toc106966356)

[Hình 2. 1 Sơ đồ use case 19](#_Toc106140130)

[Hình 2. 2 Hình sơ đồ phân rã chức năng 20](#_Toc106140131)

[Hình 3. 1 Màn hình splash 35](#_Toc107014254)

[Hình 3. 2 Giao diện đăng nhập 36](#_Toc107014255)

[Hình 3. 3 Giao diện đăng kí tài khoản 37](#_Toc107014256)

[Hình 3. 4 Giao diện trang chủ 38](#_Toc107014257)

[Hình 3. 5 Giao diện xem chi tiết phòng trọ 39](#_Toc107014258)

[Hình 3. 6 Giao diện thông báo lưu phòng yêu thích 40](#_Toc107014259)

[Hình 3. 7 Giao diện filter-lọc theo tính năng 41](#_Toc107014260)

[Hình 3. 8 Giao diện hiển thị phòng trọ lên Map và chỉ dẫn đường đi 42](#_Toc107014261)

[Hình 3. 9 Giao diện xem phòng trên Map 43](#_Toc107014262)

[Hình 3. 10 Giao diện đăng bài cho thuê phòng-Nhập thông tin địa chỉ 44](#_Toc107014263)

[Hình 3. 11 Giao diện nhập thông tin phòng Hình 3. 12 Mô tả thông tin phòng 45](#_Toc107014264)

[Hình 3. 13 Giao diện danh sách bài đăng của bạn 46](#_Toc107014265)

[Hình 3. 14 Màn hình profile 47](#_Toc107014266)

[Hình 3. 15 Màn hình đăng xuất 48](#_Toc107014267)

**TỔNG QUAN ĐỀ TÀI**

1. **Tổng quan đề tài**

Có thể nói công nghệ thông tin đang ngày càng chiếm ưu thế trong xã hội, nhờ có công nghệ thông tin chất lượng đời sống của con người ngày càng được cải thiện. Hầu hết các lĩnh vực đều cần đến công nghệ thông tin từ trường học, quán ăn, hay ngay cả trong tìm kiếm phòng trọ…

Bên cạnh đó con người sử dụng smart phone như một công cụ không thể thiếu, vì vậy việc tạo các ứng dụng trên smart phone đang là xu thế ngày càng phát triển vì tính tiện lợi, phù hợp với nhu cầu sử dụng của con người. Vì vậy em muốn xây dựng một ứng tìm kiếm phòng trọ. Nếu trước đây việc tìm kiếm phòng trọ phải thủ công, người cho thuê trọ chỉ có thể gán bảng “còn phòng cho thuê” và người tìm trọ phải trực tiếp đi tìm, việc đó gây mất thời gian, và mệt mỏi cho người tìm kiếm. Đặc biệt khu vực thành phố Hồ Chí Minh có rất nhiều bạn sinh viên ở các tỉnh vào để học tập, vì vậy nhu cầu tìm phòng trọ là vô cùng cần thiết.

Hiểu được những khó khăn đó, em muốn xây dựng một app tìm kiếm phòng trọ trên nền tảng di động, đặc biệt có thể chạy trên cả android và ios. App sẽ hỗ trợ chủ nhà có thể đăng thông tin về phòng cần cho thuê. Người tìm kiếm phòng có thể xem thông tin về các phòng, so sánh giữa các phòng và tìm kiếm cho mình được một phòng thuê ưng ý, đặc biệt ứng dụng lấy địa chỉ xác trên google map, người tìm kiếm phòng có thể theo vị trí trên google map để đi tới địa điểm. Những hạn chế, khó khăn so với cách tìm phòng trọ thủ công sẽ được giải quyết

1. **Mục tiêu nghiên cứu**

* Nghiên cứu sử dụng ngôn ngữ dart và công nghệ flutter
* Nghiên cứu quy trình triển khai xây dựng ứng dụng
* Nghiên cứu và áp dụng ứng dụng Google Map Api
* Cài đặt và thử nghiệm ứng dụng

1. **Phạm vi**

- Phát triển ứng dụng tìm kiếm phòng trọ tại thành phố Hồ Chí Minh.

1. **Cấu trúc báo cáo thực tập tốt nghiệp**

Chương 1: Cơ sở lý thuyết

Chương 2: Phân tích yêu cầu của ứng dụng

Chương 3: Thử nghiệm và đánh giá

Kết luận và kiến nghị

Phụ lục

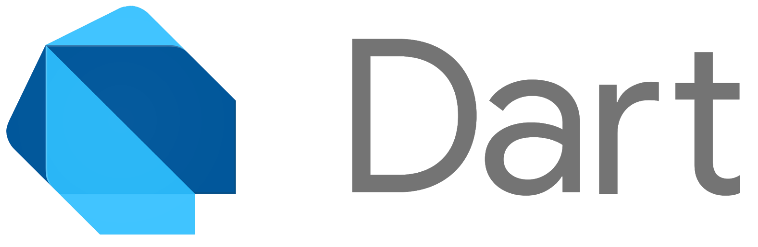
Tài liệu tham khảo

# **CHƯƠNG 1 CƠ SỞ LÝ THUYẾT**

## 1.1 Tổng quan về Ngôn Ngữ Lập Trình Dart

### 1.1.1 Giới thiệu về ngôn ngữ

Dart là ngôn ngữ lập trình đa mục đích ban đầu được phát triển bởi Google và sau đó được Ecma (ECMA-408) phê chuẩn làm tiêu chuẩn. Nó được sử dụng để xây dựng các ứng dụng web, server, máy tính để bàn và thiết bị di động. Dart là một ngôn ngữ hướng đối tượng, được xác định theo lớp, với cơ chế garbage-collected, sử dụng cú pháp kiểu C để dịch mã tùy ý sang JavaScript. Nó hỗ trợ interface, mixin, abstract, generic, static typing và sound type (2 cái cuối có thể hiểu là type-safe). Dart là ngôn ngữ mã nguồn mở và miễn phí, được phát triển trên [GitHub.](https://github.com/dart-lang)



Hình 1. 1 Logo Dart

### 1.1.2 Lý do chọn ngôn ngữ lập trình Dart

Các nhà phát triển tại Google và các nơi khác sử dụng Dart để tạo các ứng dụng chất lượng cao, quan trọng cho iOS, Android và web. Với các tính năng nhắm đến sự phát triển phía khách hàng, Dart rất phù hợp cho cả ứng dụng di động và web. Dart giúp bạn tạo ra những trải nghiệm đẹp, chất lượng cao trên tất cả các màn hình, với:

* Một ngôn ngữ được tối ưu hóa cho client
* Framework mạnh mẽ
* Công cụ linh hoạt

### 1.1.3 Tính hướng đối tượng trong Dart

**Tính trừu tượng:**

Tính trừu tượng thể hiện ở việc lựa chọn các thuộc tính và hành vi của đối tượng mà không phải liệt kê hết tất cả các thuộc tính và hành vi của đối tượng đó. Ví dụ: Để mô tả một người có rất nhiều thuộc tính và hành vi. Nhưng chúng ta chỉ sử dụng các thuộc tính như: tên , năm sinh, quê quán và thuộc tính như: đi, chạy mà không cần liệt kê hết tất cả các thuộc tính và hành vi khác như : tình trạng hôn nhân , lái xe, đá , đấm…

**Tính đóng gói:**

Thể hiện tính che dấu trong đối tượng với mục đích bảo vệ dữ liệu và tăng khả năng mở rộng.

**Tính kế thừa:**

Trong một phần mềm hay chương trình, được cấu tạo bởi nhiều lớp khác nhau cùng các thành phần khác. Mối quan hệ giữa các lớp, có mối quan hệ kế thừa, gồm lớp cha ( super class )và các lớp con(sub class), Các lớp con đó lại có thể là lớp cha của các lớp khác.

**Tính đa hình:**

Tính đa hình trong ngôn ngữ Dart cũng có ý nghĩa giống như trong các ngôn ngữ khác. Cùng biểu diễn một hành vi nhưng từng lớp có cách biểu diễn khác nhau. Như vậy, Cùng một hành vi là Chở, nhưng với các loại xe là các công dụng lại khác nhau. Đó là tính đa hình

Kết luận: Vậy là Báo Flutter đã giới thiệu đầy đủ về các đặc tính của hướng đối tượng trong ngôn ngữ Dart: tính đóng gói, tính trừu tượng , tính kế thừa và tính đa hình.

### 1.1.4 Ưu điểm của Dart

Năng suất Cú pháp Dart rõ ràng và súc tích, công cụ của nó đơn giản nhưng mạnh mẽ. Type-safe giúp bạn xác định sớm các lỗi tinh tế. Dart có các thư viện cốt lõi và một hệ sinh thái gồm hàng ngàn package.

Nhanh Dart cung cấp tối ưu hóa việc biên dịch trước thời hạn để có được dự đoán hiệu suất cao và khởi động nhanh trên các thiết bị di động và web.

Di động Dart biên dịch thành mã ARM và x86, để các ứng dụng di động của Dart có thể chạy tự nhiên trên iOS, Android và hơn thế nữa. Đối với các ứng dụng web, chuyển mã từ Dart sang JavaScript.

Dễ gần Dart quen thuộc với nhiều nhà phát triển hiện có, nhờ vào cú pháp và định hướng đối tượng không gây ngạc nhiên của nó. Nếu bạn đã biết C ++, C # hoặc Java, bạn có thể làm việc hiệu quả với Dart chỉ sau vài ngày.

Reactive Dart rất phù hợp với lập trình Reactive, với sự hỗ trợ để quản lý các đối tượng tồn tại trong thời gian ngắn, chẳng hạn như các widget UI, thông qua phân

bổ đối tượng nhanh và GC. Dart hỗ trợ lập trình không đồng bộ thông qua các tính năng ngôn ngữ và API sử dụng các đối tượng Future và Stream.

Dart cung cấp bí quyết biên dịch:

* Ahead Of time (AOT): Với AOT thì trình biên dịch chuyển ngôn ngữ Dart thẳng sang Native Code giúp hiệu suất tối ưu có khả năng (tức là khi chạy chương trình nó sẽ biên dịch từ đầu đến cuối)
* Just In Time (JIT): Còn với JIT cho phép hot reloading công việc, giúp phát triển mặt hàng nhanh và tiện lợi hơn (được hiểu như việc debug trong ngôn ngứ lập trình khác là debug hàm nào chạy hàm đấy thì ở đây nó sẽ viết đến đây thì sẽ biên dịch đến đấy)

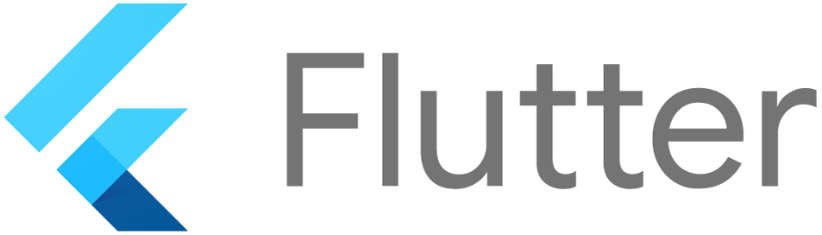
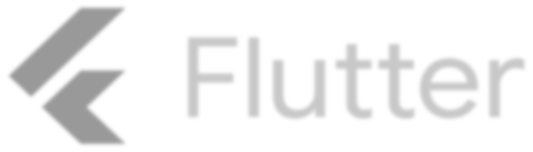
### Nhược Điểm của Dart

* Dart là một ngôn ngữ thuần OOP (hướng đối tượng)
* Cần nắm rõ hướng đối tượng trước khi bắt đầu làm việc với ngôn ngữ Dart
* Hơi khó tiếp cận cho người mới bắt đầu.

## 1.2 Tổng quan về Flutter

### Khái niệm về ngôn ngữ

Flutter được Google giới thiệu là một người mới trong thế giới ứng dụng di động. Là một SDK mới của Google dành cho các thiết bị di động giúp developers và designers xây dựng nhanh chóng ứng dụng dành cho các thiết bị di động (Android, iOS). Flutter là dự án mã nguồn mở đang trong giai đoạn thử nghiệm. Flutter bao gồm Reactive framework và công nghệ hiển thị 2D (2D rendering engine)và các công cụ phát trển(development tool). Các thành phần này làm việc cùng nhau giúp ta thiết kế, xây dựng, test, debug ứng dụng. Không có gì ngạc nhiên khi Flutter giúp các nhà phát triển tạo ra các ứng dụng native đẹp mắt và giúp họ phát triển các ứng dụng đa nền tảng một cách dễ dàng.



Hình 1. 2 Logo Flutter

Flutter gồm 2 thành phần chính quan trọng như sau:

Một SDK: Đây là một bộ sưu tập bao gồm các công cụ có thể hỗ trợ cho người dùng có thể phát triển được các ứng dụng

nền của mình. Những điều này thường bao gồm các công cụ có trình để biên dịch mã thành các mã gốc dành riêng cho hệ điều hành iOS và Android.

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

### 1.2.2 Lý do nên sử dụng Flutter

**Tính năng phát triển nhanh chóng các ứng dụng:** Hầu hết, mọi tính năng hot reload của Flutter giúp cho người dùng có thể sử dụng thử nghiệm nhanh chóng và dễ dàng hơn rất nhiều. Với khả năng xây dựng giao diện cho người dùng cộng thêm các tính năng về sửa lỗi nhanh chóng nên Flutter đang thu hút không ít người sử dụng lựa chọn. Ngoài ra, các trải nghiệm về thực hiện tải lại lần thứ hai đều rất dễ dàng mà không làm mất đi trạng thái ở trên emulator, simulator và device cho cả iOS và Android.

**UI đẹp mắt và có tính biểu cảm:** Flutter thỏa mãn người sử dụng nhờ các widget built-in vô cùng đẹp mắt dựa theo Material Design và Cupertino (iOS-flavor), thì các API sẽ thực hiện chuyển động theo nhiều hướng phong phú, scroll tự nhiên và mượt mà nên có thể tự động nhận thức được các nền tảng cần thiết.

**Quá trình truy cập với nhiều tính năng và SDK native:** Nhờ vậy, các ứng dụng của bạn sẽ trở nên sống động hơn rất nhiều nhờ vào API của platform, SDK của các bên thứ ba và native code. Từ đó, nó sẽ cho phép lập trình viên sử dụng lại được mã Java, Swift hiện tại của mình. Nhờ vậy, các truy cập sẽ thực hiện được mọi tính năng mà SDK native dựa trên IOS và Android.

**Có khả năng phát triển các ứng dụng thống nhất:** Nhờ sở hữu các công cụ cũng như thư viện nên người sử dụng có thể dễ dàng đưa ra được ý tưởng của mình vào chính trong cuộc sống trên hệ điều hành iOS và Android. Chính vì thế, nếu bạn là người chưa có nhiều kinh nghiệm cho mục đích phát triển các thiết bị di động thì việc lựa chọn Flutter được xem là một phương pháp dễ dàng và rất nhanh chóng trong việc xây dựng ra một ứng dụng di động tuyệt đẹp

### Tính năng của Flutter

* Flutter trong mắt của các chuyên gia đã và đang là một react framework.
* Flutter sử dụng một dạng ngôn ngữ lập trình Dart vô cùng đơn giản và dễ học.
* Flutter có thể phát triển nhanh chóng các ứng dụng một cách dễ dàng.
* Các giao diện của người dùng thường rất đẹp và linh hoạt.
* Nó có thể thực hiện hỗ trợ được rất nhiều widget khác nhau.
* Flutter sẽ được thể hiện ngay cùng trên một nền tảng hoàn toàn khác.
* Các ứng dụng sở hữu hiệu năng làm việc cao và nhanh chóng.

### Ưu Điểm Flutter

Flutter được đánh giá cao nhờ khả năng mạnh về hiệu ứng cũng như hiệu suất ứng dụng cao.

Sở hữu khả năng giao tiếp gần như được xem là trực tiếp với hệ thống. Là dạng ngôn ngữ kiểu tĩnh với các cú pháp hiện đại tương tự như JS, Python, Java, … Ngoài ra, compiler còn được đánh giá là linh động ngay khi dùng AOT (dành cho các sản phẩm cuối) và JIT (dành cho các quá trình này phát triển với các hot reload)

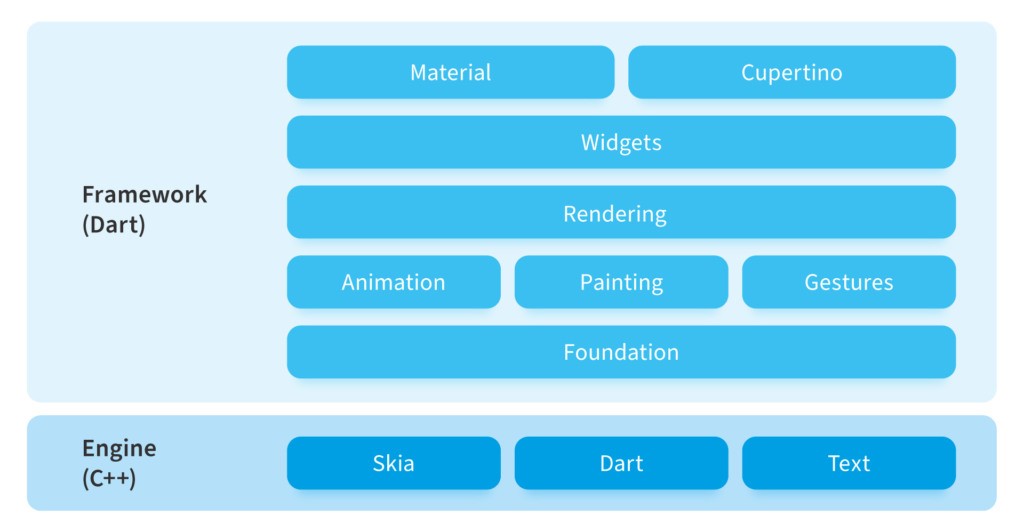
Flutter có thể chạy được trên các giả lập mobile ngay trên trang web có thể tiện cho việc phát triển. Các bộ đo lường thường chỉ các hiệu suất được hỗ trợ giúp cho lập trình viên có thể kiểm soát tốt hơn các hiệu suất của ứng dụng. Ngoài ra, nó còn thể sử dụng để xây dựng được các nền tảng gắn với ứng dụng native để có thể gia tăng hiệu suất.

### Nhược Điểm Flutter

Bộ render UI đã được nhóm phát triển viết lại hầu hết nên thường không còn liên quan tới UI đã có sẵn trong UI Framework native. Từ đó, dẫn đến việc memory sẽ sử dụng tương đối nhiều. Ngoài ra, các UI sẽ không còn đi chung với OS mà chủ yếu chỉ được phát triển riêng và chúng có thể được xem cùng một phiên bản Futter ngay khi tạo ra được ứng dụng dành riêng cho iOS thì iOS 8.x -> 12.x đều sẽ tương tự nhau và tương tự như với Android. Tuy nhiên, các UI của Android đương nhiên sẽ khác hơn rất nhiều so với iOS.

Bổ sung thường xuyên ngôn ngữ DART: Theo điều tra thì số lượng lập trình viên biết về DART là không lớn và có nhiều rủi ro rằng khi học xong DART thì Developer có thể sẽ dính liền luôn với DART ở công việc phát triển cho các ứng dụng mobile. Chính vì vậy, vấn đề làm việc uyển chuyển JS hay Python hay có thể linh động qua lại giữa front, back hay AI…sẽ không còn khả năng thực hiện.

Một mô hình dữ liệu hoàn toàn mới: Khi hướng đến phát triển React Native thì bạn sẽ mất rất nhiều thời gian cho việc học thêm các mô hình dữ liệu trong Flutter, đây là một công việc không bị đánh giá khó. Chính vì Flutter là con cưng được sản xuất bởi Google nên sẽ dính phải nhiều phốt là điều vô cùng dễ dàng, chính vì vậy bạn cần cân nhắc khi sử dụng. Tuy nhiên, vẫn nhiều chuyên gia khẳng định rằng Flutter rất tốt và được cho là tốt hơn rất nhiều lần so với Angular mà Google đã từng làm ra trước đây.

* + 1. **Kiến trúc của Flutter**

Hình 1. 3 Kiến trúc Flutter

Với Flutter thì tất cả đều sẽ được quy về các widget và mỗi một widget phức hợp khác thường sẽ bao gồm những widget khác ở bên trong. Nhờ vào sự kết hợp linh hoạt trên mà người sử dụng có thể tạo ra được bất kỳ ứng dụng phức tạp nào khác.

Mọi tính năng tương tác của Flutter đều sẽ được tích hợp vào bất kỳ thời điểm nào nhờ vào GestureDetector widget. Tất cả các trạng thái của widget đều sẽ được các quản lý cập nhật bởi StatefulWidget Widget.

Flutter có khả năng cung cấp các thiết kế class để bất kỳ các lớp đều sẽ được lập trình sẵn tùy thuộc vào các mức độ phức tạp riêng của từng tác vụ.

**1.2.7 Tóm tắt những lí do nên sử dụng Flutter:**

Có rất nhiều nguyên nhân thuyết phục người dùng mở rộng học Flutter như sau:

**Là ngôn ngữ dễ dàng và dễ dùng:** Flutter có khả năng tạo ra một ứng dụng gốc mà không cần đến code quá nhiều. Chính vì vậy, nếu người dùng đã rất thành thạo Java, Swift hoặc React Native thì có thể sử dụng Flutter một cách đơn giản và dễ dàng hơn rất nhiều.

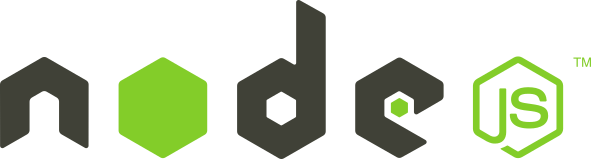
**Hiệu suất được tối ưu hóa, khả năng biên dịch nhanh chóng:** Người sử dụng có thể thay đổi code của mình một cách nhanh chóng và có thể dễ dàng xem được kết quả ngay trong thời gian thực. Điều này còn được gọi là Hot-Reload vì bạn sẽ chỉ cần mất một khoảng thời gian ngắn sau khi cập nhật ứng dụng.

**Hiện thực hóa các ý tưởng của các Startup nhanh chóng:** Nếu như bạn muốn giới thiệu các sản phẩm của mình đến với nhà đầu tư thì lựa chọn Flutter là một nước đi vô cùng hoàn hảo. Khi phát triển được các ứng dụng di động với Flutter sẽ rẻ hơn vì bạn không cần phải thực hiện cải tạo duy trì hai phiên bản dành cho IOS hay Android nữa.

**Cộng đồng phát triển lớn mạnh:** Flutter đang sở hữu một trong những cộng đồng phát triển vô cùng mạnh mẽ và nó được đánh giá cao nhờ khởi đầu bằng sự chất lượng tuyệt đối: bạn có thể trao đổi mọi thắc mắc của mình về Flutter trên rất nhiều website nó sở hữu.

## 1.3 Tổng quan về NodeJS

### Khái niệm



Hình 1. 4 Logo NodeJS

Node.js là một mã nguồn mở, một môi trường cho các máy chủ và ứng dụng mạng.Node.js sử dụng Google V8 JavaScript engine để thực thi mã, và một tỷ lệ lớn các mô-đun cơ bản được viết bằng JavaScript. Các ứng dụng node.js thì được viết bằn JavaScript.

Node.js chứa một thư viện built-in cho phép các ứng dụng hoạt động như một Webserver mà không cần phần mềm như Nginx, Apache HTTP Server hoặc IIS.

Node.js cung cấp kiến trúc hướng sự kiện (event-driven) và non-blocking I/O API, tối ưu hóa thông lượng của ứng dụng và có khả năng mở rộng cao

Mọi hàm trong Node.js là không đồng bộ (asynchronous). Do đó, các tác vụ đều được xử lý và thực thi ở chế độ nền (background processing).

### Ứng dụng của Nodejs

* Xây dựng websocket server (Chat server)
* Hệ thống Notification (Giống như facebook hayTwitter)
* Ứng dụng upload file trên client
* Các máy chủ quảng cáo
* Các ứng dụng dữ liệu thời gian thực khác.

### Ưu điểm

Đặc điểm nổi bật của Node.js là nó nhận và xử lý nhiều kết nối chỉ với một singlethread. Điều này giúp hệ thống tốn ít RAM nhất và chạy nhanh nhất khi không phải tạo thread mới cho mỗi truy vấn giống PHP. Ngoài ra, tận dụng ưu điểm nonblocking I/O của Javascript mà Node.js tận dụng tối đa tài nguyên của server mà không tạo ra độ trễ như PHP.

JSON APIs Với cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tuyệt vời cho các dịch vụ Webs làm bằng JSON.

Ứng dụng trên 1 trang( Single page Application) Nếu bạn định viết 1 ứng dụng thể hiện trên 1 trang (Gmail?) NodeJS rất phù hợp để làm. Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh. Các ứng dụng bạn định viết không muốn nó tải lại trang, gồm rất nhiều request từ người dùng cần sự hoạt động nhanh để thể hiện sự chuyên nghiệp thì NodeJS sẽ là sự lựa chọn của bạn.

Shelling tools unix NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu xuất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.

Streamming Data (Luồng dữ liệu) Các web thông thường gửi HTTP request và nhận phản hồi lại (Luồng dữ liệu). Giả xử sẽ cần xử lý 1 luồng giữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.

Ứng dụng Web thời gian thực Với sự ra đời của các ứng dụng di động & HTML 5 nên Node.js rất hiệu quả khi xây dựng những ứng dụng thời gian thực (real-time applications) như ứng dụng chat, các dịch vụ mạng xã hội như Facebook, Twitter,…

### Nhược điểm

Ứng dụng nặng tốn tài nguyên Nếu bạn cần xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding encryption… hoặc các ứng dụng tương tự như vậy thì không nên dùng NodeJS (Lý do: NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm 1 trình biên dịch của NodeJS sẽ lâu hơn 1 chút ). Trường hợp này bạn hãy viết 1 Addon C++ để tích hợp với NodeJS để tăng hiệu suất tối đa !

NodeJS và ngôn ngữ khác NodeJS, PHP, Ruby, Python .NET …thì việc cuối cùng là phát triển các App Web. NodeJS mới sơ khai như các ngôn ngữ lập trình khác. Vậy nên bạn đừng hi vọng NodeJS sẽ không hơn PHP,Ruby,Python… ở thời điểm này. Nhưng với NodeJS bạn có thể có 1 ứng dụng như mong đợi, điều đó là chắc chắn !

## 1.4 Tổng quan về Google Map Api

### 1.4.1 Khái niệm

Google Map là một dịch vụ ứng dụng vào công nghệ bản đồ trực tuyến trên web miễn phí được cung cấp bởi Google, hỗ trợ nhiều dịch vụ khác của Google đặc biệt là dò đường và chỉ đường; hiển thị bản đồ đường sá, các tuyến đường tối ưu cho từng loại phương tiện, cách bắt xe và chuyển tuyến cho các loại phương tiện công cộng (xe bus, xe khách ...), và những địa điểm (kinh doanh, trường học, bệnh viện, cây

ATM...) trong khu vực cũng như khắp nơi trên thế giới.

**Map API là gì?**

Là một phương thức cho phép 1 website B sử dụng dịch vụ bản đồ của website A (gọi là Map API) và nhúng vào website của mình (site B). Site A ở đây là google map, site B là các website cá nhân hoặc tổ chức muốn sử dụng dịch vụ của google (di chuột, room, đánh dấu trên bản đồ…)

Các ứng dụng xây dựng trên maps được nhúng vào trang web cá nhân thông qua các thẻ javascripts do vậy việc sử dụng API google rất dễ dàng.

Google Map API đã được nâng cấp lên phiên bản v3 không chỉ hỗ trợ cho các máy để bàn truyền thống mà cho cả các thiết bị di động; các ứng dụng nhanh hơn và nhiều hơn.

Các dịch vụ hoàn toàn miễn phí với việc xây dựng một ứng dụng nhỏ. Trả phí nếu đó là việc sử dụng cho mục đích kinh doanh, doanh nghiệp.

### Một số ứng dụng của Google Map Api.

Đánh dấu các địa điểm trên bản đồ kèm theo thông tin cho địa điểm đó : khu vui chơi giải trí, nhà hàng khách sạn, cây ATM, bệnh viện, trường học,… bất cứ địa điểm nào bạn muốn.

Chỉ dẫn đường đến các địa điểm cần tìm(đường tối ưu và nhiều option khác),chỉ dẫn đường giao thông công cộng, có thể là các địa điểm cung cấp như trên. Ở đây sử dụng các service google cung cấp. Khoanh vùng khu vực: các trung tâm kinh tế, khu đô thị, khu ô nhiễm. Tình trạng giao thông các khu vực từ đó đưa ra các giải pháp…

### Thuật toán tìm đường đi ngắn nhất trong Google Map API

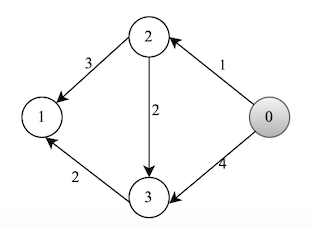
Dijkstra là thuật toán đường đi ngắn nhất. Thuật toán Dijkstra được sử dụng để tìm khoảng cách ngắn nhất của tất cả các nút từ nút bắt đầu đã cho. Nó tạo ra cây đường đi ngắn nhất từ ​​một nút nguồn một cách hợp lý, bằng cách tiếp tục thêm các nút một cách tham lam sao cho tại mọi điểm mỗi nút trong cây có khoảng cách tối thiểu từ nút bắt đầu đã cho.

**Ý tưởng:**

Giống như thuật toán Bellman-Ford, thuật toán Dijkstra cũng tối ưu hóa đường đi bằng cách xét các cạnh (𝑢,𝑣), so sánh hai đường đi 𝑆→𝑣 sẵn có với đường đi 𝑆→𝑢→𝑣. Thuật toán hoạt động bằng cách duy trì một tập hợp các đỉnh trong đó ta đã biết chắc chắn đường đi ngắn nhất. Mỗi bước, thuật toán sẽ chọn ra một đỉnh 𝑢 mà chắc chắn sẽ không thể tối ưu hơn nữa, sau đó tiến hành tối ưu các đỉnh 𝑣 khác dựa trên các cạnh (𝑢,𝑣) đi ra từ đỉnh 𝑢. Sau 𝑁 bước, tất cả các đỉnh đều sẽ được chọn, và mọi đường đi tìm được sẽ là ngắn nhất. Cụ thể hơn, thuật toán sẽ duy trì đường đi ngắn nhất đến tất cả các đỉnh. Ở mỗi bước, chọn đường đi 𝑆→𝑢 có tổng trọng số nhỏ nhất trong tất cả các đường đi đang được duy trì. Sau đó tiến hành tối ưu các đường đi 𝑆→𝑣 bằng cách thử kéo dài thành 𝑆→𝑢→𝑣 như đã mô tả trên.

**Minh hoạ thuật toán:**

Ta sẽ minh họa thuật toán bằng một đồ thị như hình.



Hình 1. 5 Minh hoạ thuật toán Dijkstra bước 1

* Định nghĩa: là đường đi ngắn nhất từ đỉnh nguồn đên đỉnh u đã tìm được.
* nhận hai giá trị 𝑡𝑟𝑢𝑒, 𝑓𝑎𝑙𝑠𝑒 cho biết đỉnh đã được chọn để tối ưu chưa.

Ban đầu, 𝐷=[0,∞,∞,∞], 𝑃=[𝑓𝑎𝑙𝑠𝑒,𝑓𝑎𝑙𝑠𝑒,𝑓𝑎𝑙𝑠𝑒,𝑓𝑎𝑙𝑠𝑒]

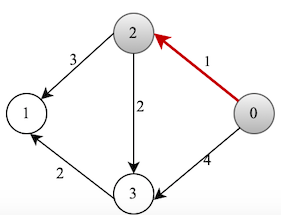
* Bước 1: Thuật toán sẽ chọn đỉnh 0, vì =0 là nhỏ nhất thỏa mãn =𝑓𝑎𝑙𝑠𝑒.

Tiến hành tối ưu các cạnh đi ra:

* Cạnh (0,2): cập nhật =𝑚𝑖𝑛(, + )=𝑚𝑖𝑛(∞,0+1)=1
* Cạnh (0,3): cập nhật =𝑚𝑖𝑛(,𝐷0+)=𝑚𝑖𝑛(∞,0+4)=4

Sau bước này, 𝐷=[0,∞,1,4], 𝑃=[𝑡𝑟𝑢𝑒,𝑓𝑎𝑙𝑠𝑒,𝑓𝑎𝑙𝑠𝑒,𝑓𝑎𝑙𝑠𝑒]

* Bước 2: thuật toán sẽ chọn ra đỉnh 2, có =1 là nhỏ nhất thỏa mãn =𝑓𝑎𝑙𝑠𝑒.



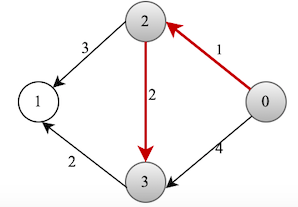
Hình 1. 6 Minh hoạ thuật toán Dijkstra bước 2

Tiến hành tối ưu các cạnh đi ra:

* Cạnh (2,1): cập nhật =𝑚𝑖𝑛(,+)=𝑚𝑖𝑛(∞,1+3)=4
* Cạnh (2,3): cập nhật 𝐷3=𝑚𝑖𝑛(,+)=𝑚𝑖𝑛(4,1+2)=3

Sau bước này, 𝐷=[0,4,1,3], 𝑃=[𝑡𝑟𝑢𝑒,𝑓𝑎𝑙𝑠𝑒,𝑡𝑟𝑢𝑒,𝑓𝑎𝑙𝑠𝑒]

* Bước 3: thuật toán sẽ chọn ra đỉnh 3, có =1 là nhỏ nhất thỏa mãn 𝑃3=𝑓𝑎𝑙𝑠𝑒.



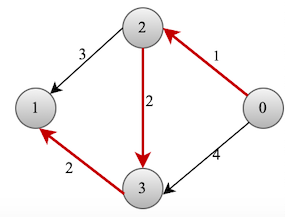
Hình 1. 7 Hình minh hoạ thuật toán Dijkstra bước 3

Tiến hành tối ưu các cạnh đi ra:

* Cạnh (3,1): cập nhật =𝑚𝑖𝑛(,+)=𝑚𝑖𝑛(4,3+2)=4

Sau bước này, 𝐷=[0,4,1,3], 𝑃=[𝑡𝑟𝑢𝑒,𝑓𝑎𝑙𝑠𝑒,𝑡𝑟𝑢𝑒,𝑡𝑟𝑢𝑒]

* Bước 4: thuật toán sẽ chọn đỉnh 1. Không có cạnh nào đi ra. Đến đây, tất cả các đỉnh đều đã được đánh dấu. Thuật toán kết thúc. Đường đi ngắn nhất tìm được từ đỉnh 0 là 𝐷=[0,4,1,3].



Hình 1. 8 Hình minh hoạ thuật toán Dijkstra bước 4

## 1.5 Tổng quan MongoDB



Hình 1. 9 Hình mongoDB

### 1.5.1 Tìm hiểu MongoDB

**1.5.1.1 Khái Niệm**

MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++. Nó có thể hỗ trợ trên nhiều nền tảng khác nhau và được thiết kế với mục đích hướng đến đối tượng. MongoDB hoạt động dựa vào các khái niệm Collection và Document. Đồng thời, nó có hiệu suất cao cùng với tính khả dụng tốt và dễ dàng mở rộng.

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng. Với sự ra đời của nó đã khắc phục được những nhược điểm của dữ liệu theo mô hình quan hệ RDBMS nhằm cải thiện tốc độ vận hành, chức năng, khả năng mở rộng mô hình, bộ nhớ memory cache…

So với RDBMS thì trong MongoDB **collection** ứng với **table**, còn **document** sẽ ứng với **row** , MongoDB sẽ dùng các document thay cho row trong RDBMS.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB

Một chút thông tin bên lề về MongoDB: [MongoDB](https://www.mongodb.com/) được ra đời bởi MongoDB Inc. (một công ty phần mềm của Mỹ). Công ty được thành lập lần đầu tiên vào năm 2007 với tên gọi 10gen. Nó cũng là một PaaS (Platform as a Service) tương tự như Windows Azure và Google App Engine, sau đó đã được chuyển thành nguồn mở từ năm 2009.

MongoDB đã trở thành một trong những NoSQL database nổi trội nhất bấy giờ, được dùng làm backend cho rất nhiều website như eBay, SourceForge và The New York Times.

**1.5.1.2 Các thuật ngữ hay sử dụng trong MongoDB**

**\_id** – Là trường bắt buộc có trong mỗi document. Trường \_id đại diện cho một giá trị duy nhất trong document MongoDB. Trường \_id cũng có thể được hiểu là khóa chính trong document. Nếu bạn thêm mới một document thì MongoDB sẽ tự động sinh ra một \_id đại diện cho document đó và là duy nhất trong cơ sở dữ liệu MongoDB.

**Collection** – Là nhóm của nhiều document trong MongoDB. Collection có thể được hiểu là một bảng tương ứng trong cơ sở dữ liệu RDBMS (Relational Database Management System). Collection nằm trong một cơ sở dữ liệu duy nhất. Các collection không phải định nghĩa các cột, các hàng hay kiểu dữ liệu trước.

**Cursor** – Đây là một con trỏ đến tập kết quả của một truy vấn. Máy khách có thể lặp qua một con trỏ để lấy kết quả.

**Database** – Nơi chứa các Collection, giống với cơ sở dữ liệu RDMS chúng chứa các bảng. Mỗi Database có một tập tin riêng lưu trữ trên bộ nhớ vật lý. Một mấy chủ MongoDB có thể chứa nhiều Database.

**Document** – Một bản ghi thuộc một Collection thì được gọi là một Document. Các Document lần lượt bao gồm các trường tên và giá trị.

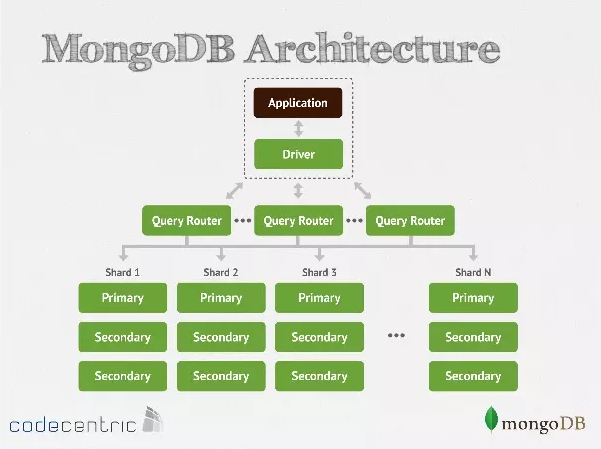
**Field** – Là một cặp name – value trong một document. Một document có thể có không hoặc nhiều trường. Các trường giống các cột ở cơ sở dữ liệu quan hệ.

**JSON** – Viết tắt của JavaScript Object Notation. Con người có thể đọc được ở định dạng văn bản đơn giản thể hiện cho các dữ liệu có cấu trúc. Hiện tại JSON đang hỗ trợ rất nhiều ngôn ngữ lập trình.

**Index** – Là những cấu trúc dữ liệu đặc biệt, dùng để chứa một phần nhỏ của các tập dữ liệu một cách dễ dàng để quét. Chỉ số lưu trữ giá trị của một fields cụ thể hoặc thiết lập các fields, sắp xếp theo giá trị của các fields này. Index hỗ trợ độ phân tích một cách hiệu quả các truy vấn. Nếu không có chỉ mục, MongoDB sẽ phải quét tất cả các documents của collection để chọn ra những document phù hợp với câu truy vấn. Quá trình quét này là không hiệu quả và yêu cầu MongoDB để xử lý một khối lượng lớn dữ liệu.

Hãy lưu ý sự khác biệt của các trường và \_id trong một document. Một \_id được dùng để đại diện cho một document và chúng được sinh ra khi thêm một Document vào Collection.

### 1.5.2 Nguyên tắc hoạt động của MongoDB



Hình 1. 10 Kiến trúc của MongoDB

MongoDB hoạt động dưới một tiến trình ngầm service, luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.

Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau. Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.

Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.

### 1.5.3 Ưu điểm và hạn chế của MongoDB

**a. Ưu điểm**

Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên bạn muốn gì thì cứ insert vào thoải mái.

Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.

MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster:

Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

Khi có một truy vấn dữ liệu, bản ghi được cached lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng

Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.

Đội ngũ hỗ trợ chuyên nghiệp: Khi gặp phải bất kỳ sự cố nào chúng ta có thể liên hệ trực tiếp đến hệ thống hỗ trợ để xử lý kịp thời.

**Tính khả dụng cao**: MongoDB không chỉ có các tính năng về sao chép mà có thể sử dụng gridFS (có thể lưu trữ và truy xuất các tệp vượt quá kích thước 16MB). Các tính năng giúp tăng tính khả dụng và đạt hiệu suất cao.

**b. Hạn chế**

Một ưu điểm của MongoDB cũng chính là nhược điểm của nó. MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận.

Tốn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dữ thừa dữ liệu.

Khi insert/update/remove bản ghi, MongoDB sẽ chưa cập nhật ngay xuống ổ cứng, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng điêù này sẽ là nhược điểm vì sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện...

# **CHƯƠNG 2 PHÂN TÍCH YÊU CẦU CỦA ỨNG DỤNG**

## 2.1 Mô tả bài toán

Ngày nay xã hội càng phát triển, con người có xu hướng đến những nơi hiện đại hơn để học tập, làm việc, tìm kiếm những cơ hội để phát triển bản thân, trong đó có thể nói đến TP.HCM nói chung và TP. Thủ Đức nói riêng, đây là nơi tập trung nhiều khu công nghiệp, nhiều trường đại học, cao đẳng. Do đó với mong muốn giải quyết vấn đề trên em đã thực hiện đề tài “Phát triển ứng dụng tìm kiếm phòng trọ tại thành phố Hồ Chí Minh”.

Ứng dụng cho phép người dùng đăng kí tài khoản bằng tên đăng nhập và mật khẩu. Nếu đăng kí thành công thì người dùng có thể đăng nhập dựa tên tài khoản đã đăng kí. Đăng nhập thành công thì người dùng vào trang chủ ứng dụng sẽ hiển thị thông tin của các bài đăng phòng trọ. Ứng dụng này có các chức năng chính sau: hiển thị danh sách các phòng trọ, hiển thị các phòng lên google map, lưu danh sách các phòng yêu thích, tìm kiếm/lọc các yêu cầu mà người dùng mong muốn. Người dùng có thể xem chi tiết phòng bao gồm các thuộc tính như: hình ảnh, mô tả, giá phòng, tiện ích, địa chỉ…Khi người dùng chọn được phòng mong muốn thì có thể xem đường đi đến địa điểm đó. Trong quá trình tìm kiếm phòng nếu người dùng thích một phòng bất kì có thể lưu lại thông tin của phòng đó. Người dùng có thể đăng bài cho thuê phòng trọ bao gồm các thuộc tính như: địa chỉ, giá phòng, kích thước, mô tả…

Những tiện ích của việc xây dựng ứng dụng tìm phòng trọ sẽ giúp cả người cho thuê và người thuê tiết kiệm được thời gian, công sức. Mang đến sự hài lòng, bên cạnh đó mỗi người dung sẽ được tạo một tài khoản để truy cập vào ứng dụng.

## 2.2 Phân tích và thiết kế hệ thống

### 2.2.1 Sơ đồ Use case

Hệ thống có 2 tác nhân chính: người dùng và khách, các usecase chính trong hệ thống bao gồm:

(i). Quản lý thông tin cá nhân

(ii). Đăng nhập hệ thống

(iii). Đăng kí hệ thống

(iv). Quản lý bài đăng cá nhân

(v). Tìm kiếm nhà trọ

(vi). Xem thông tin nhà trọ

(vii). Xem bản đồ

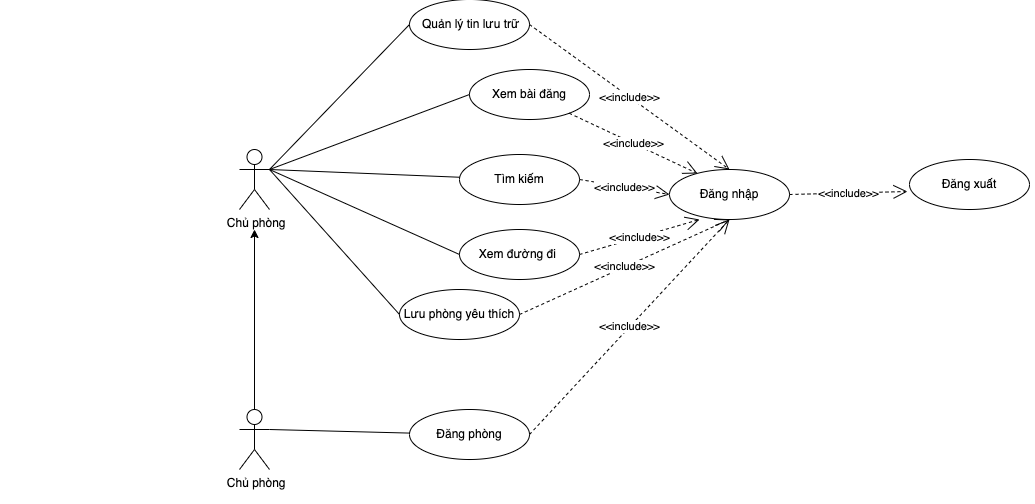
(viii). Lưu trữ tin yêu thích

(ix). Quản lý tin yêu thích

(x). Đăng bài cho thuê phòng

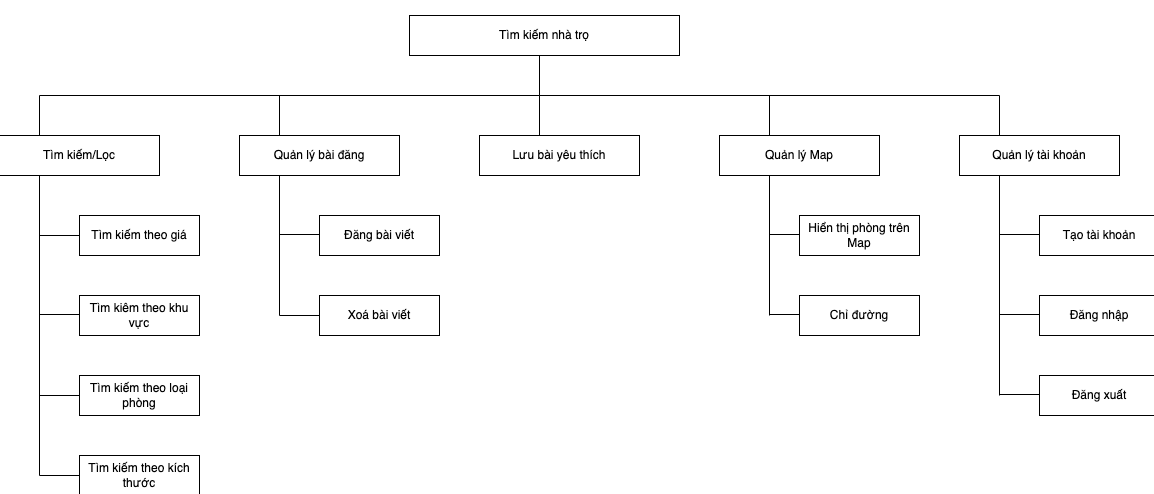
Trong đó:

1. Quản lý thông tin cá nhân người dùng bao gồm: tên, hình ảnh…
2. Cho phép người dùng đăng nhập vào hệ thống thông qua username và password.
3. Người dùng đăng kí thông tin tài khoản thông qua username và pasword.
4. Sau khi người dùng đã đăng tin thì các bài đăng sẽ được lưu trữ lại.
5. Người dùng có thể tìm kiếm thông tin phòng trọ thông qua loại phòng, giá cả, kích thước phòng.
6. Cho phép người dùng xem chi tiết thông tin của nhà trọ bao gồm: mô tả, địa chỉ, kích thước phòng, giá cả…
7. Cho phép người dùng có thể xem trực tiếp trên bản đồ giúp người dùng có thể tiện theo dõi hơn vị trí các phòng trọ.
8. Người dùng có thể lưu tin mà mình muốn sau này có thể xem lại.
9. Các bài đăng mà người dùng đã lưu trữ sẽ được hiển thị ở phần này.
10. Người dùng có thể đăng thông tin phòng trọ mà mình muốn cho thuê.



Hình 2. 1 Sơ đồ use case

### 2.2.2 Sơ đồ phân rã chức năng



Hình 2. 2 Hình sơ đồ phân rã chức năng

### 2.2.3 Đặc tả chức năng

Các usecase chính của hệ thống bao gồm use case “Tìm kiếm nhà trọ”, use case “Đăng bài viết”.

**2.2.3.1 Đặc tả usecase”Tìm kiếm nhà trọ”**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mã use case** | **UC1** | | | | |
| **Tên use case** | Tìm kiếm nhà trọ | | | | |
| **Tác nhân** | Người dùng, hệ thống | | | | |
| **Tiền điều kiện** | Thiết bị di động được kết nối với Internet | | | | |
| **Luồng sự kiện chính** |  | **STT** | **Người thực hiện** | **Hành động** |  | |
| 1 | Người dùng | Mở ứng dụng, chọn chức năng  lọc |
| 2 | Hệ thống | Hiển thị màn hình chức năng lọc. |
| 3 | Người dùng | Nhập các thông tin |
|  |  | 4 | Hệ thống | Kiểm tra định dạng thông tin nhập vào, gọi api tìm kiếm nhà  trọ |  | |
| 5 | Server | Lấy dữ liệu từ cơ sở dữ liệu, lọc theo các thông tin người  dùng đã nhập, trả về cho Client |
| 6 | Client | Nhận kết quả từ Server, hiển thị kết quả ở màn hình Bản đồ |
| **Luồng sự kiện thay thế** |  | **STT** | **Người thực hiện** | **Hành động** |  | |
| 1 | Người dùng | Chọn lọc thông tin theo loại phòng |
| 2 | Người dùng | Chọn kích thước phòng |
| 3 | Người dùng | Chọn lọc thông tin theo giá |

Bảng 2. 1 Đặc tả usecase”Tìm kiếm nhà trọ

**2.2.3.2 Đặc tả usecase”Đăng bài viết”**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mã use case** | **UC2** | | | | | |
| **Tên use case** | Đăng bài viết | | | | | |
| **Tác nhân** | User | | | | | |
| **Tiền điều kiện** | Thiết bị di động được kết nối với Internet, đăng nhập vào hệ thống | | | | | |
| **Luồng sự kiện chính** |  | **STT** | **Người thực hiện** | | **Hành động** |  |
| 1 | Người dùng | | Mở ứng dụng, vào tính năng đăng tin. |
| 2 | Hệ thống |  | Hiển thị màn hình đăng bài viết. |
| 3 | Người dùng |  | Nhập các thông tin bao gồm: Quận, phường/xã, tên đường, số nhà, số điện thoại, giá, loại phòng, kích thước phòng, hình ảnh, tiêu đề bài đăng, mô tả. |
|  | Hệ thống |  | Kiểm tra định dạng thông tin nhập vào đúng chưa, gọi api để đăng bài viết. |
| 7 | Server |  | Lưu bài viết vào cơ sỡ dữ liệu sau đó trả lại thông tin bài viết cho người dùng. |
| 8 | Người dùng |  | Nhận kết quả trả về từ Server, hiển thị kết quả ở trang chủ |
| **Luồng sự kiện thay thế** |  | **STT** | **Người thực hiện** | **Hành động** | |  |
| 1 | Hệ thống | Thông báo: Nhập tiêu đề | |
| 2 | Hệ thống | Thông báo: Nhập giá phòng | |
| 3 | Hệ thống | Thông báo: Nhập diện tích phòng | |
| 4 | Hệ thống | Thông báo: Nhập số điện thoại liên hệ | |
| 6 | Hệ thống | Thông báo: Nhập địa chỉ phòng | |
| 6 | Hệ thống | Thông báo: Nhập mô tả phòng | |
| 7 | Hệ thống | Thông báo: Cần chọn ảnh | |
|  |  | 8 | Hệ thống | Thông báo: Cần chọn loại phòng | |  |

Bảng 2. 2 Đặc tả usecase “Đăng bài viết”

**2.2.3.3 Đặc tả usecase “Đăng kí tài khoản”**

|  |  |
| --- | --- |
| **Tên Usecase** | Đăng ký tài khoản |
| **Tác nhân** | User |
| **Tiền điều kiện** | Thiết bị di động được kết nối với Internet, đăng nhập vào hệ thống |
| **Luồng sự kiện**  **Chính** | * Người dùng chọn mục đăng ký tài khoản. * Form đăng ký tài khoản hiển thị. * Khách hàng xem nhập thông tin vào. * Nhấn nút đăng ký. * Ứng dụng sẽ báo kết quả quá trình nhập thông tin cá nhân. Nếu  thông tin nhập không chính xác thì thực hiện nguồn rẽ nhánh. Còn  nếu nhập chính xác thì hệ thống sẽ cập nhật thông tin khách vào danh sách người dùng. |
| **Luồng sự kiện thay thế** | * Quá trình nhập thông tin không chính xác thì ứng dụng sẽ hiển thị thông báo và người dùng đăng kí lại. |
| **Hậu điều kiện** | * Người dùng chưa có tài khoản |

Bảng 2. 3 Đặc tả usecase “Đăng kí tài khoản”

**2.2.3.4 Đặc tả usecase “Đăng nhập”**

|  |  |
| --- | --- |
| **Tên Usecase** | Đăng nhập |
| **Tác nhân** | Người dùng |
| **Tiền điều kiện** | Thiết bị di động được kết nối với Internet, người dùng chưa đăng nhập vào ứng dụng |
| **Luồng sự kiện chính** | * Người dùng chọn chức năng đăng nhập. * Form đăng nhập hiển thị. Nhập username và password vào form đăng nhập. * Server kiểm tra username và password của người dùng. Nếu đăng  nhập thành công thì vào trang chủ và tiếp tục, nếu nhập sai thì  thực hiện điều kiện rẽ nhánh. |
| **Luồng sự kiện rẽ**  **Nhánh** | * Đăng nhập không thành công thì ứng dụng sẽ báo quá trình  đăng nhập không thành công. Chọn đăng nhập lại thực hiện  luồng trên. |
| **Hậu điều kiện** | Thành viên đã đăng nhập thành công và có thể sử dụng các  chức năng mà hệ thống cung cấp. |

Bảng 2. 4 Đặc tả usecase “Đăng nhập”

**2.2.3.5 Đặc tả usecase “Xem thông tin phòng”**

|  |  |
| --- | --- |
| **Tên Usecase** | Xem thông tin phòng |
| **Tác nhân** | Người dùng |
| **Tiền điều kiện** | Thiết bị di động được kết nối với Internet, đăng nhập vào hệ thống |
| **Luồng sự kiện chính** | * Người dùng chọn phòng cần xem. * Màn hình xem thông tin chi tiết phòng xuất hiện,  hệ thống hiển thị thông tin phòng đã  chọn. * Người dùng xem thông  tin chi tiết về sản phẩm được hiển thị. |
| **Luồng sự kiện rẽ nhánh** | Không có |
| **Hậu điều kiện** | Không có |

Bảng 2. 5 Đặc tả usecase ”xem thông tin phòng”

**2.2.3.6 Đặc tả usecase “Đăng phòng”**

|  |  |
| --- | --- |
| **Tên Usecase** | Đăng phòng |
| **Tác nhân** | Người dùng |
| **Tiền điều kiện** | Thiết bị di động được kết nối với Internet, đăng nhập vào hệ thống |
| **Luồng sự kiện chính** | * Người dùng chọn chức năng đăng phòng. * Màn hình xem đăng phòng sẽ cho người dùng nhập thông tin gồm 3 bước:  1. Nhập thông tin địa chỉ(Quận/Huyện, Phường/Xã, Tên đường, số nhà), số điện thoại. 2. Nhập thông tin loại phòng, kích thước, giá tiền, hình ảnh 3. Nhập tiêu đề, mô tả của bài đăng  * Sau khi nhập đầy đủ các thông tin trên thì bấm nút “Đăng phòng”, nếu đăng thành công thì ứng dụng sẽ hiển thị thông báo thành công. |
| **Luồng sự kiện rẽ nhánh** | Không có |
| **Hậu điều kiện** | Không có |

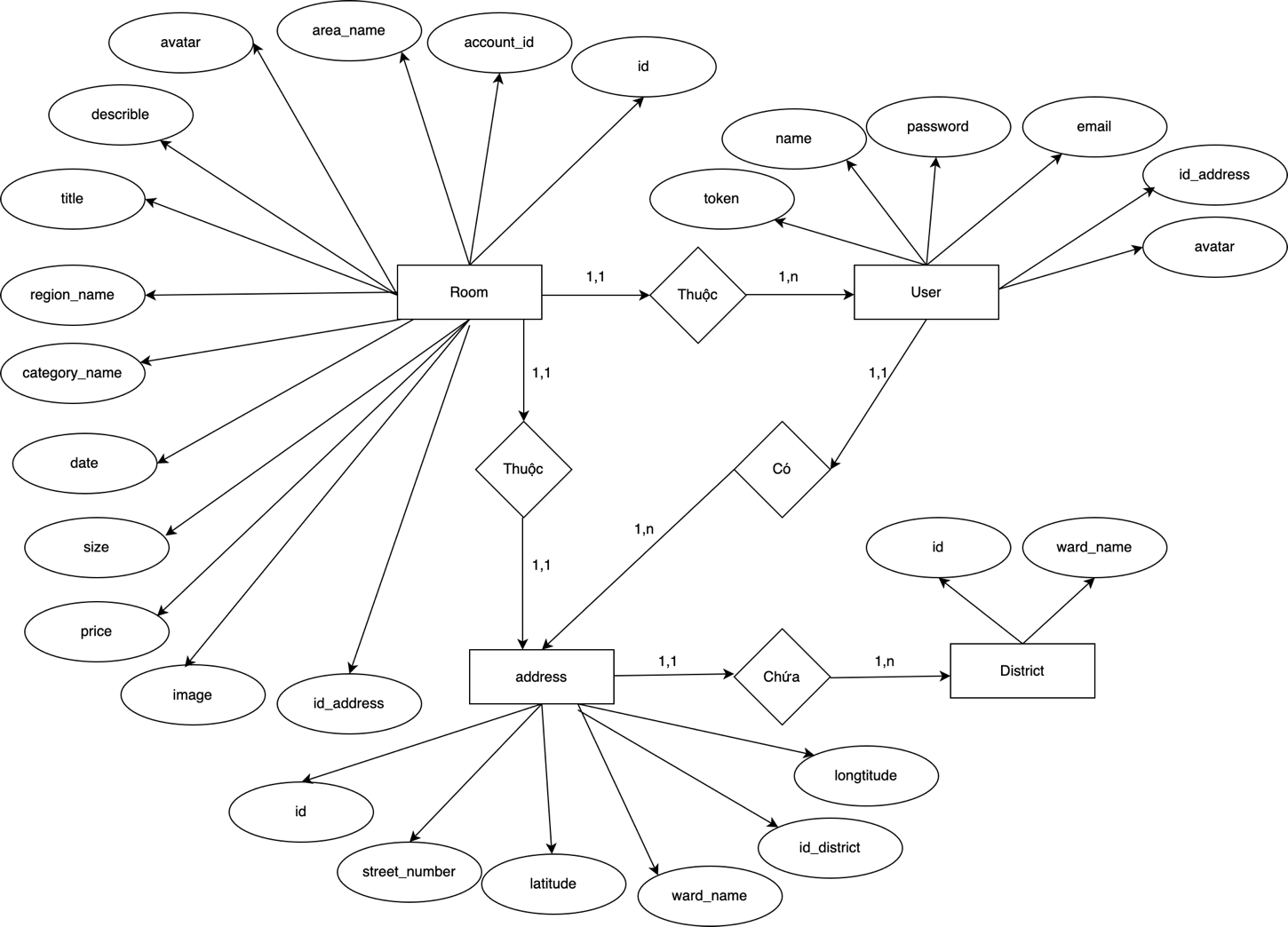
Bảng 2. 6 Đặc tả usecase ”Đăng phòng”

**2.2.3.6 Đặc tả usecase “Lưu phòng yêu thích”**

|  |  |
| --- | --- |
| **Tên Usecase** | Lưu phòng yêu thích |
| **Tác nhân** | Người dùng |
| **Tiền điều kiện** | Thiết bị di động được kết nối với Internet, đăng nhập vào hệ thống |
| **Luồng sự kiện chính** | * Người dùng chọn phòng cần xem. * Màn hình xem thông tin chi tiết phòng xuất hiện,  hệ thống hiển thị thông tin phòng đã  chọn. * Người dùng bấm lưu phòng yêu thích, nếu lưu thành công thì ứng dụng thông báo lưu thành công |
| **Luồng sự kiện rẽ nhánh** | Không có |
| **Hậu điều kiện** | Không có |

Bảng 2. 7 Đặc tả usecase “Lưu phòng yêu thích”

### 2.2.3 Thiết kế cơ sở dữ liệu



Hình 2. 3 Sơ đồ thực thể liên kết

Bảng 2.3 mô tả các thuộc tính của phòng

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | id | VARCHAR(255) | Khóa chính của bảng Room |
| 2 | account\_id | VARCHAR(255) | Id của người dùng |
| 3 | area\_name | VARCHAR(255) | Tên khu vực |
| 4 | avatar | VARCHAR(255) | Avatar người dùng |
| 5 | describle | VARCHAR(255) | Mô tả phòng |
| 6 | title | VARCHAR(255) | Tiêu đề bài đăng |
| 8 | category\_name | VARCHAR(255) | Tên loại phòng phòng |
| 9 | date | DateTime | Ngày đăng |
| 10 | size | INT | Kích thước phòng |
| 11 | price | LONG | Giá phòng |
| 12 | image | VARCHAR(255) | Đường dẫn ảnh |
| 13 | Id\_address | VARCHAR(255) | Khoá ngoại |

Bảng 2. 8 Mô tả chi tiết bảng Room

Bảng 2.4 mô tả chi tiết các thuộc tính của User, ở đây có thuộc tính token khi người dùng đăng nhập vào thì server sẽ trả về một chuỗi kí tự(token).

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | id | VARCHAR(255) | Khóa chính của bảng User |
| 2 | name | VARCHAR(255) | Tên của người dùng |
| 3 | password | VARCHAR(255) | Mật khẩu |
| 4 | avatar | VARCHAR(255) | Avatar người dùng |
| 5 | email | VARCHAR(255) | Email |
| 6 | token | VARCHAR(255) | Token để xác thực người dùng |
| 7 | Id\_address | VARCHAR(255) | Khoá ngoại |

Bảng 2. 9 Mô tả chi tiết bảng User

Bảng 2.5 mô tả các thuộc tính của District

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | id | VARCHAR(255) | Khóa chính của bảng Adress |
| 2 | street\_name | VARCHAR(255) | Số nhà, tên đường |
| 3 | latitude | VARCHAR(255) | Vĩ độ |
| 4 | longitude | VARCHAR(255) | Kinh độ |
| 5 | Id\_district | VARCHAR(255) | Khoá ngoại |

Bảng 2. 10 Mô tả chi tiết bảng District

## 

## 2.3 Danh sách API

|  |  |  |
| --- | --- | --- |
| API | **Kiểu** | Chi tiết |
| <https://findroomapi.herokuapp.com/findroom/room> | GET | Lấy về danh sách các phòng trọ đã được đăng |
| <https://findroomapi.herokuapp.com/findroom/room> | POST | Thực hiện đăng phòng |
| <https://findroomapi.herokuapp.com/findroom/favoriteRoom/>id | POST | Thực hiện lưu phòng yêu thích theo user id |
| <https://findroomapi.herokuapp.com/findroom/favoriteRoom/>id | GET | Lấy về danh sách phòng trọ yêu thích |
| <https://findroomapi.herokuapp.com/findroom/adduser> | POST | Đăng kí tài khoản và mã hoá mật khẩu khi người dùng đăng kí thành công |
| <https://findroomapi.herokuapp.com/findroom/authenticate> | GET | Xác thực tài khoản đăng nhập và trả về cho người dùng token |
| [https://findroomapi.herokuapp.com /findroom/getinfo](https://findroomapi.herokuapp.com/findroom/getinfo) | GET | Lấy về thông tin user |
| [https://findroomapi.herokuapp.com /findroom/favoriteRoom/delete/](https://findroomapi.herokuapp.com/findroom/favoriteRoom/delete/)id\_room | DELETE | Xoá phòng yêu thích đã lưu trữ dựa vào id\_room |
| [https://findroomapi.herokuapp.com /findroom/room/user/](https://findroomapi.herokuapp.com/findroom/room/user/)id\_user | GET | Lấy về danh sách phòng đã đăng |

# **CHƯƠNG 3 THỬ NGHIỆM VÀ ĐÁNH GIÁ**

## Môi trường thử nghiệm

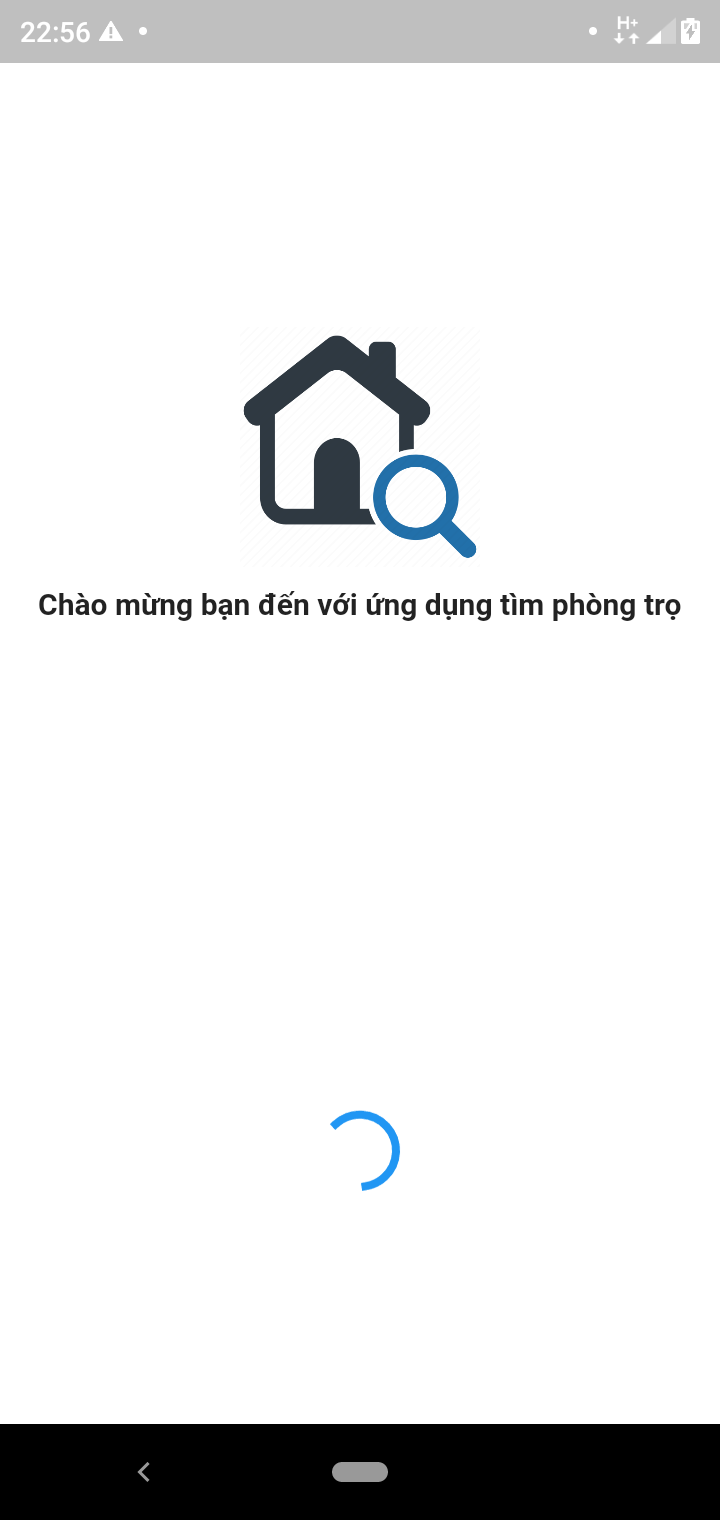
* Công nghệ: Flutter framework 2.0.2
* Ngôn ngữ lập trình: Dart
* Công cụ: Visual Studio Code, Android Studio
* Giả lập: Emulator

## Giao diện chương trình

* + 1. **Giao diện màn hình splash**

Splash screen thường là màn hình xuất hiện đầu tiên khi mở app. Màn hình này thường chỉ chứa branding logo của ứng dựng. Thông thường, các bạn muốn add splash screen vào app vào mục đích:

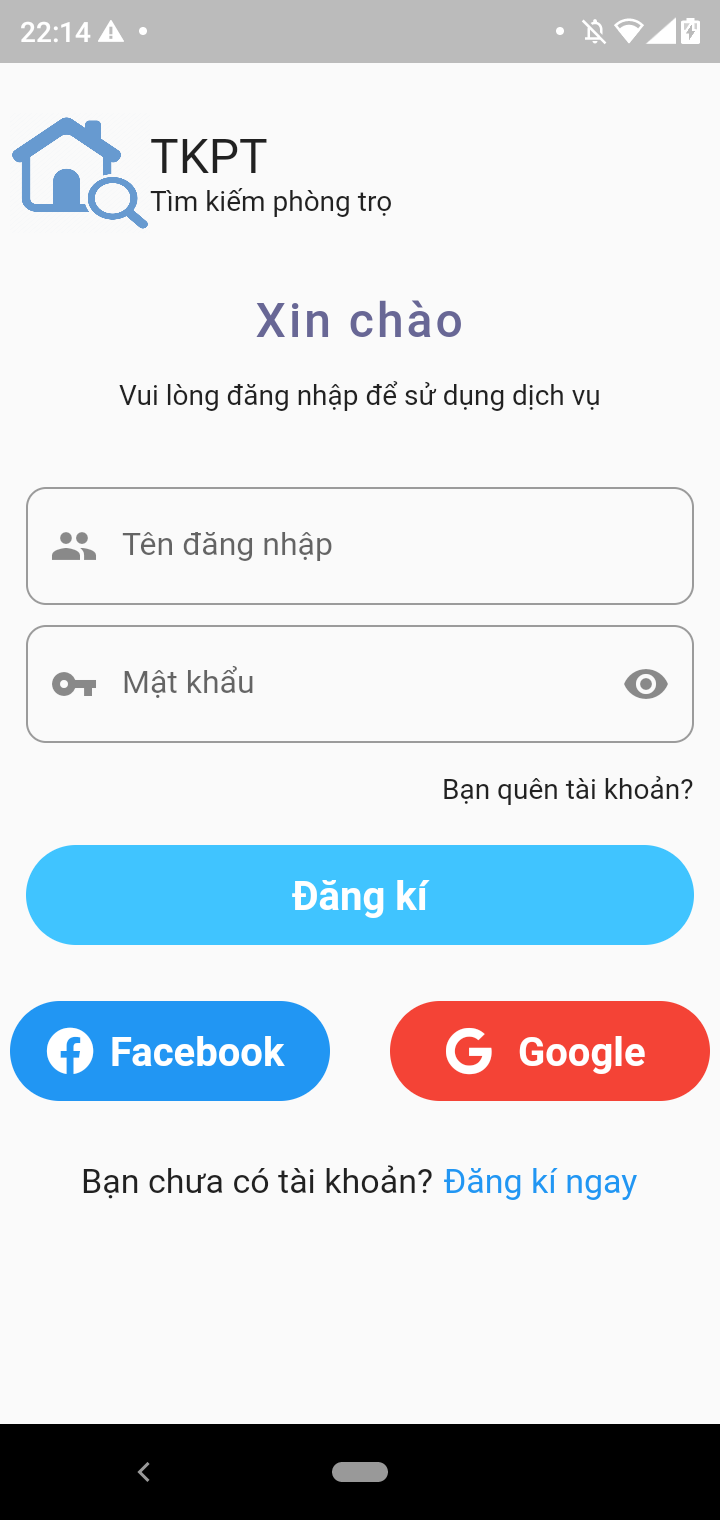
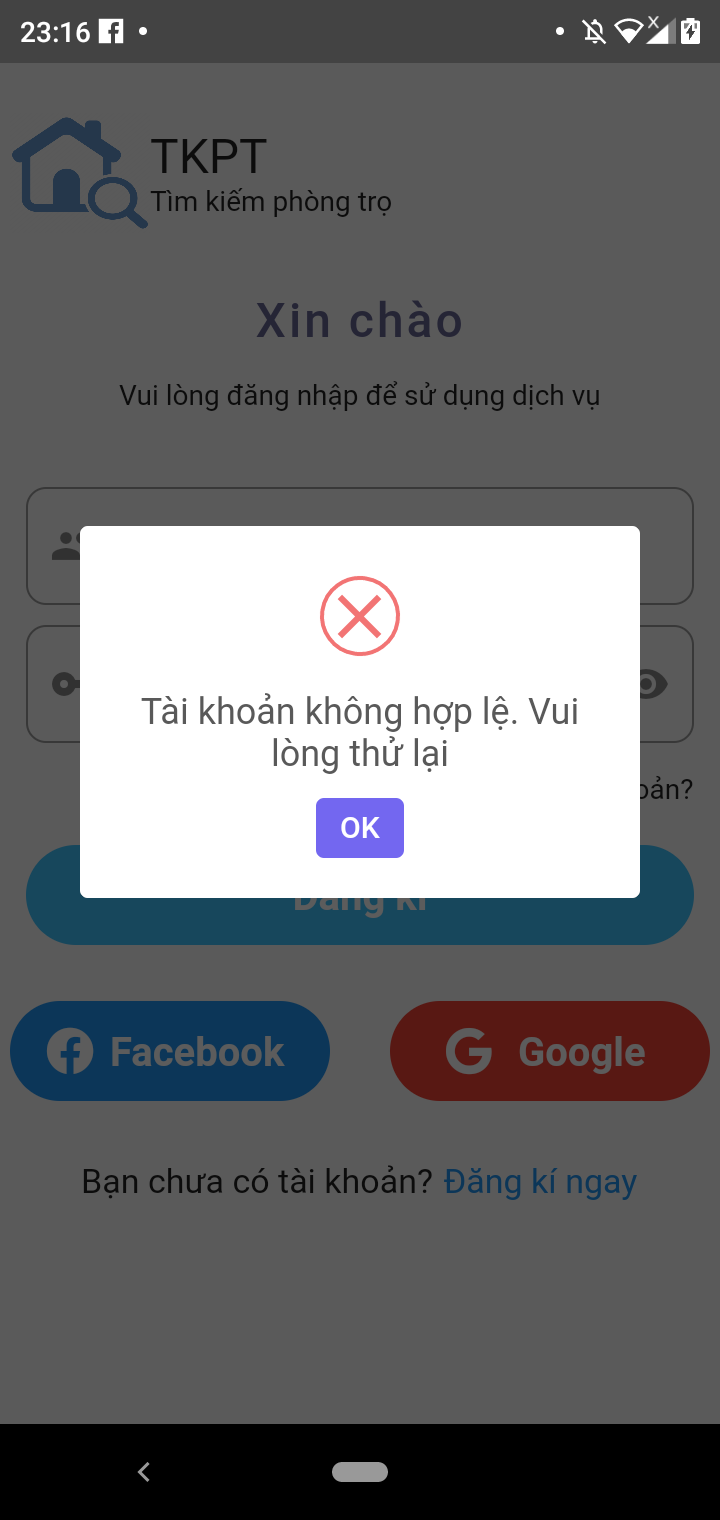
* Tận dụng thời gian chạy các task nặng khi app lần đầu được khởi động. Thời gian app khởi động là thời gian có nhiều task cần phải xử lý nhất: load storage, authenticate tài khoản, check update, load config,... Nếu bạn tận dụng được đúng cách splash screen, bạn có thể tăng tốc app và cải thiện UX của app rất nhiều.



Hình 3. 1 Màn hình splash

**3.2.2. Màn hình đăng nhập**

Cũng như bao ứng dụng khác đầu tiên chúng ta phải tiến hành đăng nhập để sử dụng ứng dụng. Người dùng đăng nhâp bằng username và password đã đăng kí trước đó. Sau đó ứng dụng sẽ xác thực.

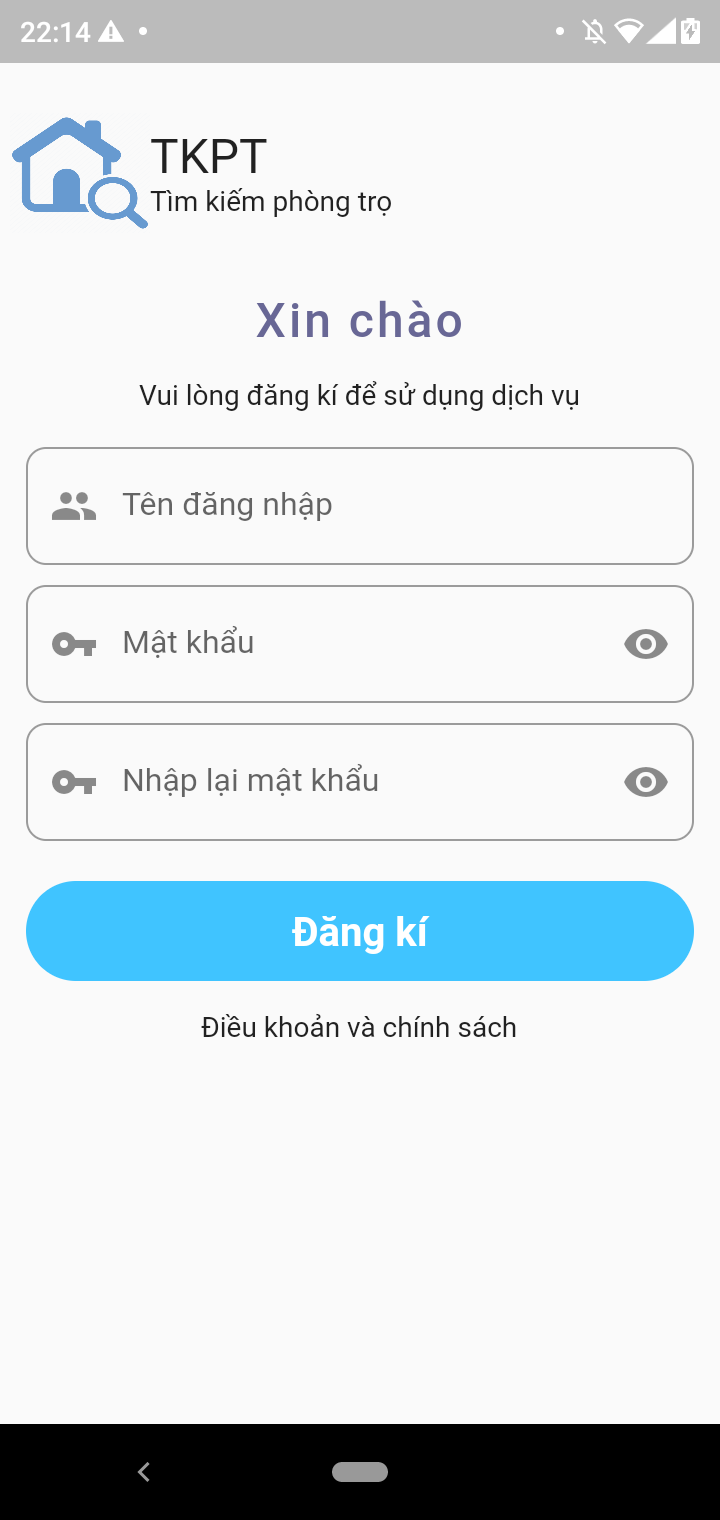
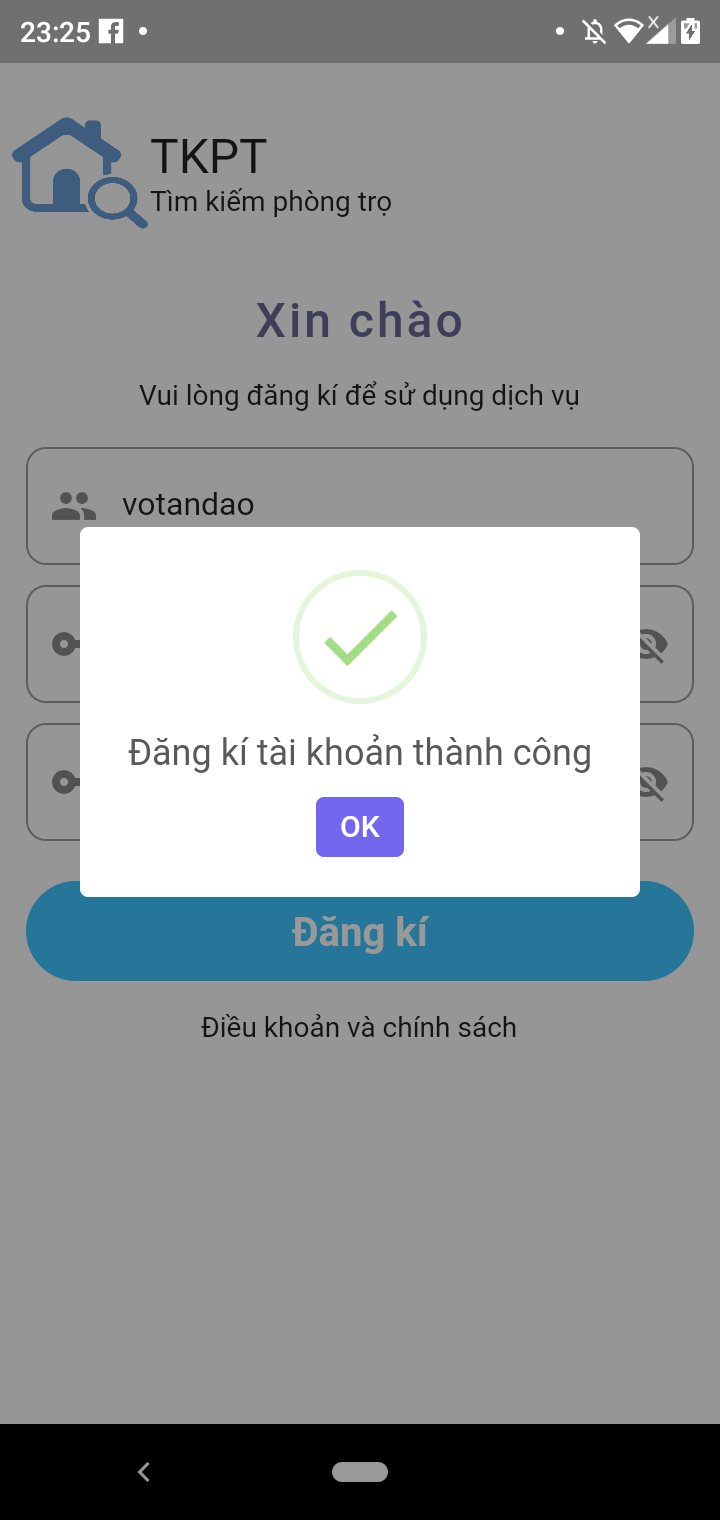


Hình 3. 2 Giao diện đăng nhập

Nếu xác thực tài khoản thất bại thì ứng dụng sẽ hiển thị lên màn hình 1 thông báo ”Tài khoản không hợp lệ. Vui lòng thử lại”. Ngược lại, nếu ứng dụng xác thực tài khoản thành công thì sẽ vào màn hình trang chủ.

* + 1. **Màn hình đăng kí tài khoản**

Nếu người dùng chưa có tài khoản thì tiến hành vào trang đăng kí để đăng kí tài khoản bao gồm username và password. Lưu ý người dùng phải xác nhận lại đúng mật khẩu thì mới đăng kí thành công.

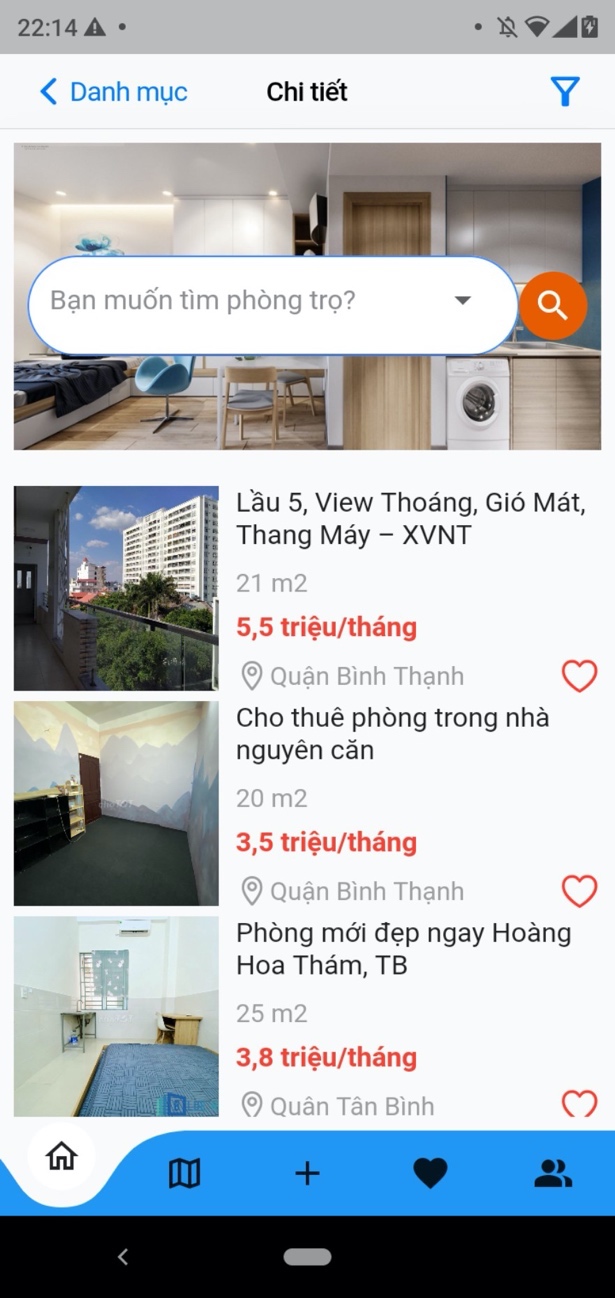


Hình 3. 3 Giao diện đăng kí tài khoản

Nếu đăng kí tài khoản thành công thì ứng dụng sẽ thông báo cho người dùng”Đăng kí tài khoản thành công”. Sau đó người dùng có thể tiến hành đăng nhập.

* + 1. **Giao diện trang chủ**

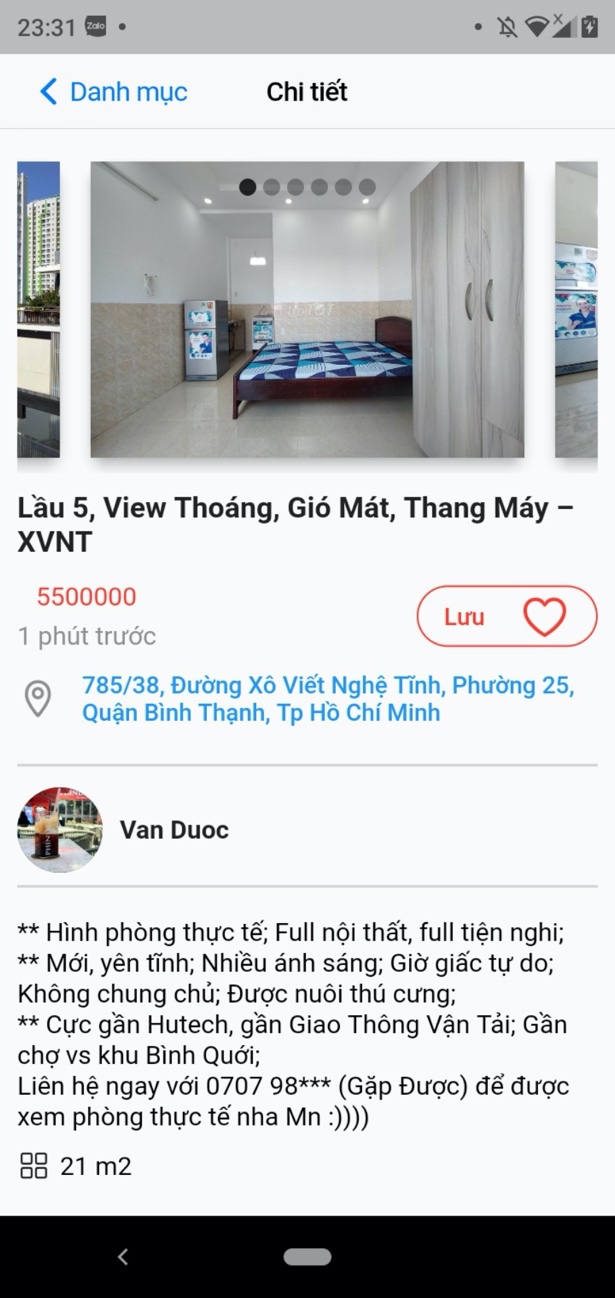
Khiđăng nhập thành công ứng dụng sẽ vào trang chủ, ở đây sẽ hiển thị tất cả danh sách phòng cho người dùng xem, bạn có thể tìm kiếm theo quận mà mình mong muốn.



Hình 3. 4 Giao diện trang chủ

* + 1. **Giao diện chi tiết phòng trọ**

Khi ở màn hình trang chủ nếu người dùng muốn xem chi tiết nội dung của phòng trọ đó thì nhấn vào thì sẽ hiển thị hình ảnh, nội dung, giá, địa chỉ…Ở màn hình chi tiết phòng trọ, nếu người dùng muốn lưu lại tin đó để sau này có thể xem thì nhấn vào nút “Lưu”.

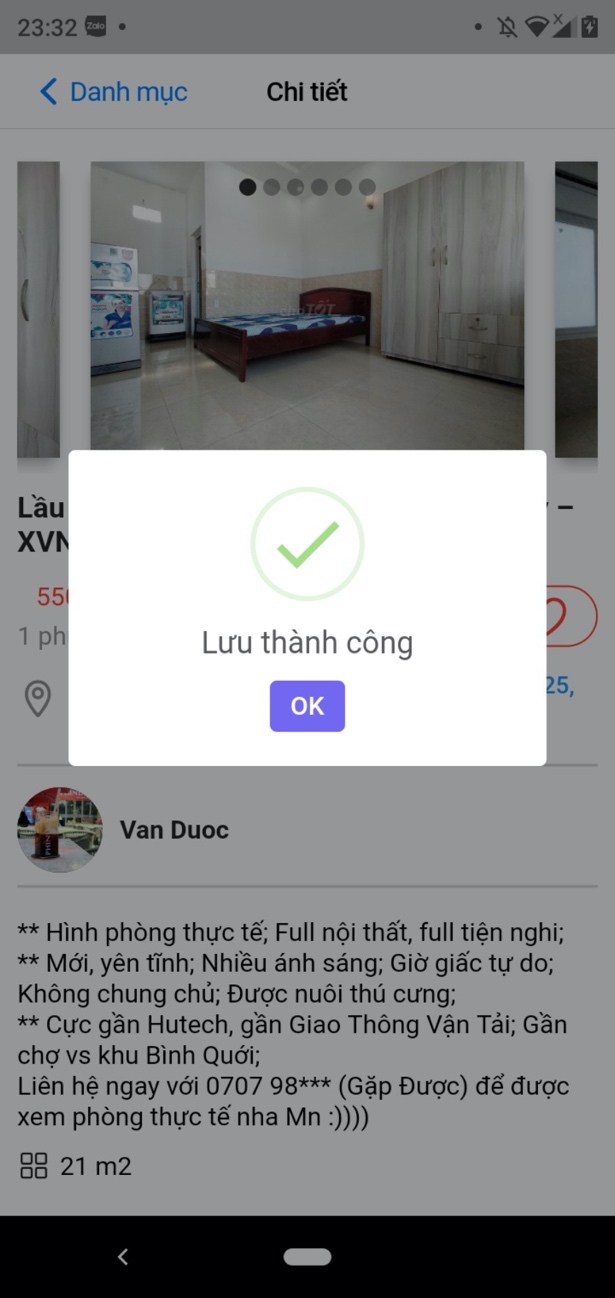
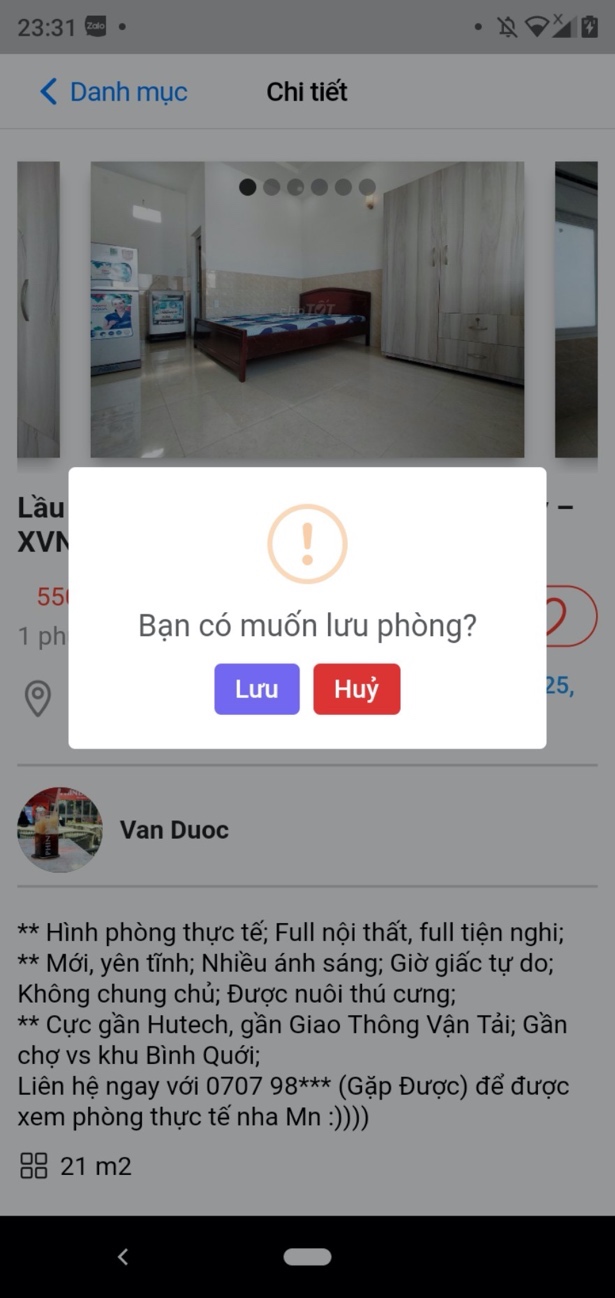


Hình 3. 5 Giao diện xem chi tiết phòng trọ

* + 1. **Giao diện lưu phòng yêu thích**

Khi người dùng muốn lưu lạiphòng để có thể tiện xem lại thì người dùng click vào nút lưu,ứng dụng sẽ hiển thị thông báo”Bạn có muốn lưu phòng?”.

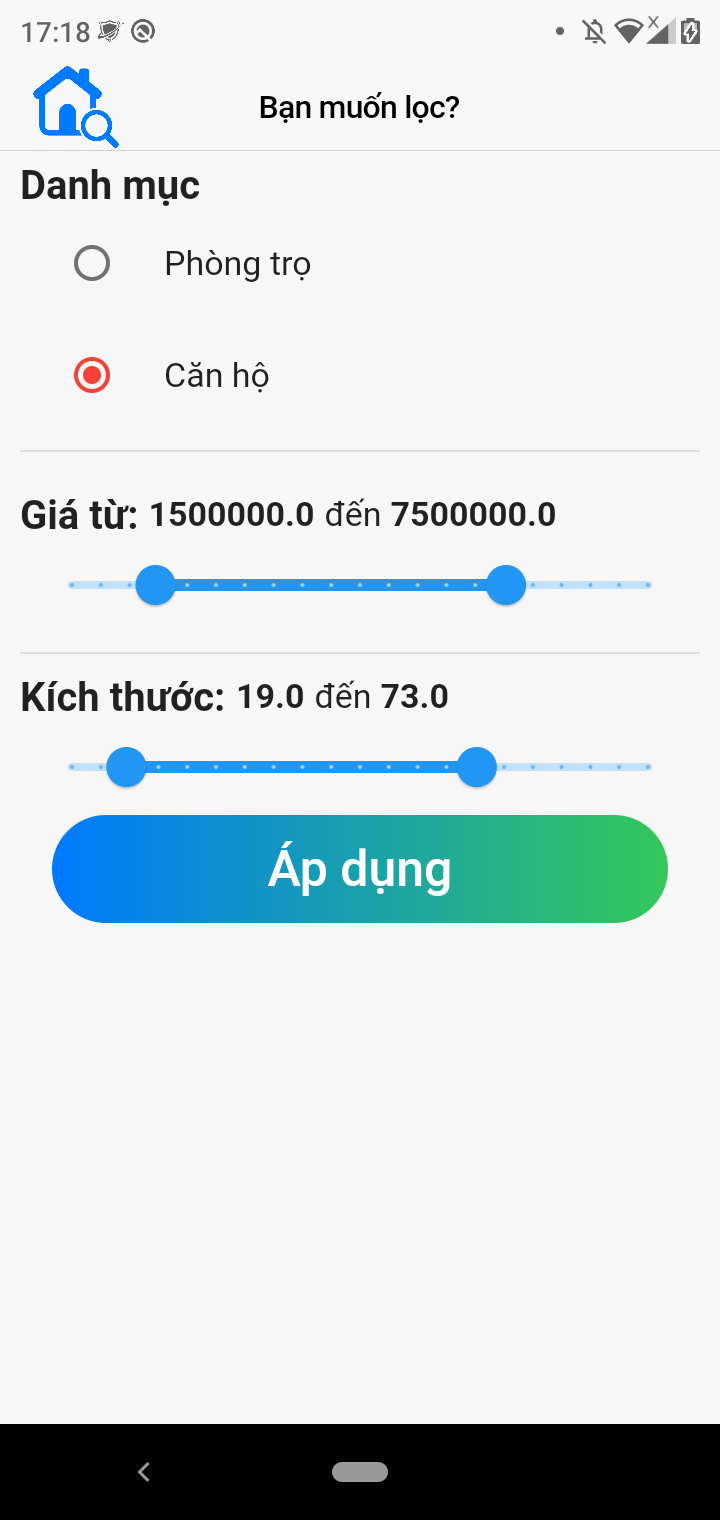
Nếu người dùng bấm ”Lưu” thì phòng đó sẽ được lưu lại và thông báo cho người dùng biết ”Lưu thành công”.

****

Hình 3. 6 Giao diện thông báo lưu phòng yêu thích

* + 1. **Giao diện filter**

Để thuận tiện cho người dùng tìm kiếm theo mong muốn của mình thì ở màn hình này người dùng có thể lọc những thông tin như: loại phòng(Phòng trọ, căn hộ), giá(từ 0 đồng đến 10 triệu đồng), kích thước(từ 10 m2 đến 100 m2).



Hình 3. 7 Giao diện filter-lọc theo tính năng

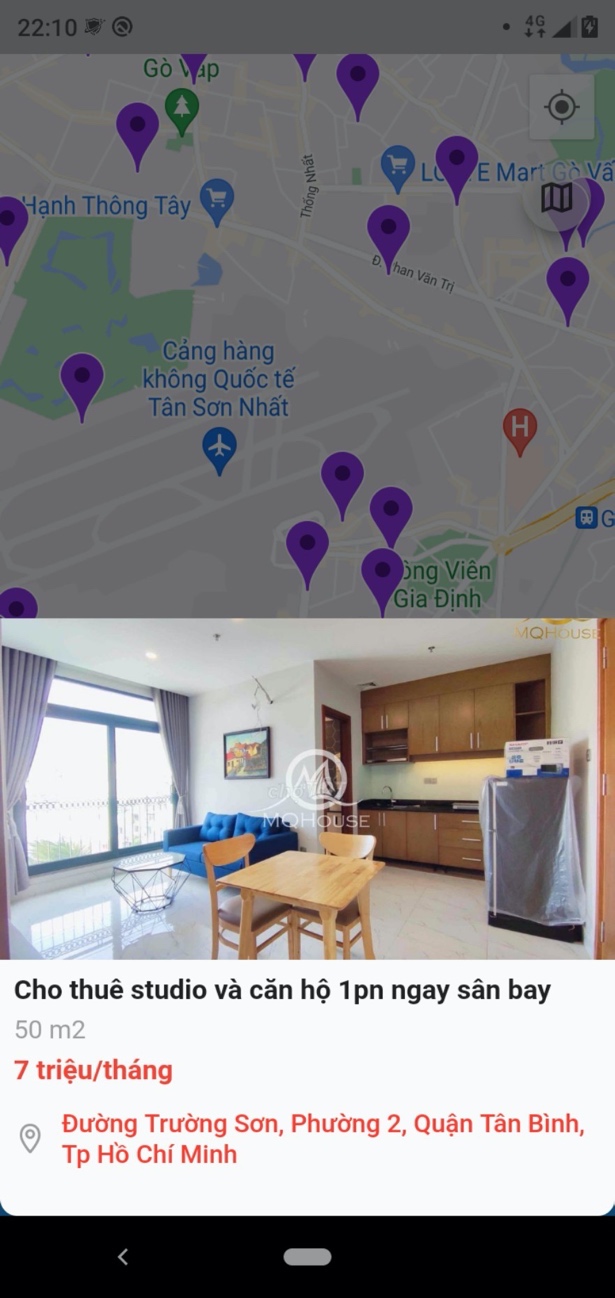
* + 1. **Giao diện hiển thị phòng trên Map**

Hình 3. 8 Giao diện hiển thị phòng trọ lên Map và chỉ dẫn đường đi

Sử dụng google map API để hiển thị danh sách các phòng trong khu vực qua đó giúp người dùng có thể dễ dàng nhìn thấy tổng quát vị trí của tùng phòng theo khu vực, từ đó có thể tìm kiếm dễ dàng và phù hợp với nhu cầu của bản thân.

Khi người dùng chọn vào điểm đã đánh dấu trên bản đồ, người dùng có thể xem đường đi từ vị trí hiện tại của mình so với địa điểm phòng trọ.

Khi người dùng click vào điểm được đảnh dấu trên bản đồ thì ứng dụng sẽ hiển thị một vài thông tin cơ bản của phòng trọ cho người dùng biết, nếu người dùng muốn xem chi tiết thì click vào phòng trọ đó để có thể xem được chi tiết hơn.

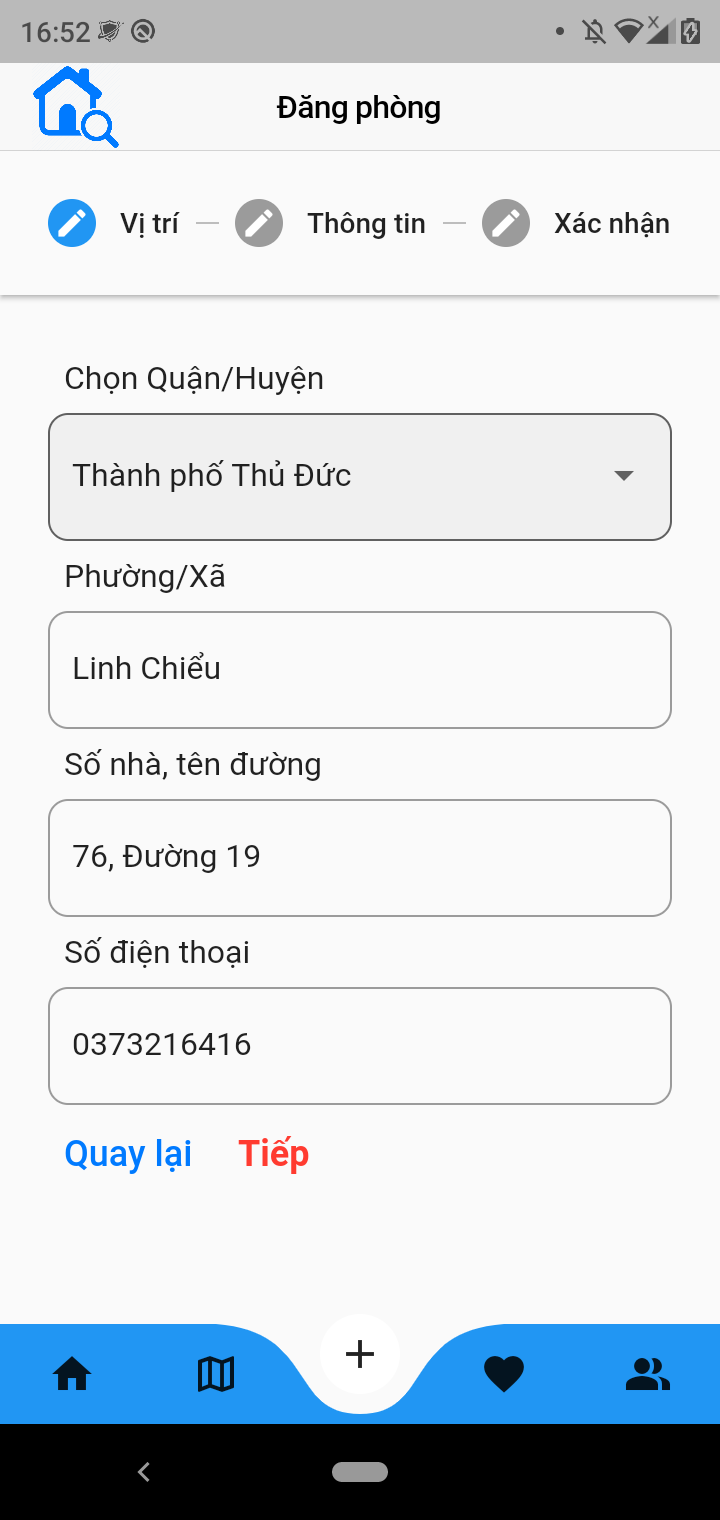


Hình 3. 9 Giao diện xem phòng trên Map

* + 1. **Giao diện đăng phòng cho thuê**

Người dùng có thể sử dụng ứng dụng để đăng phòng muốn cho người khác thuê, người dùng cần nhập ba thông tin bao gồm: vị trí, thông tin phòng(loại phòng cho thuê, giá, kích thước, hình ảnh), và xác nhận(tiêu đề, mô tả).

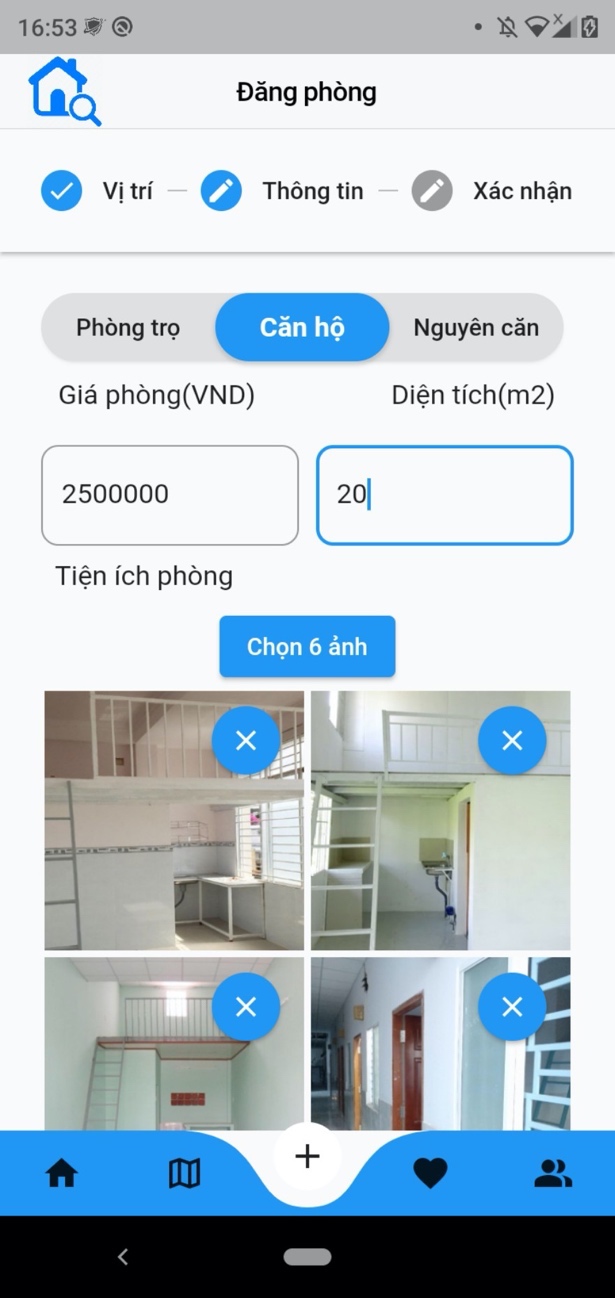
Bước 1: Người dùng nhập thông tin địa chỉ phòng bao gồm(Quận/Huyện, Phường/Xã, Số nhà, tên đường, số điện thoại.



Hình 3. 10 Giao diện đăng bài cho thuê phòng-Nhập thông tin địa chỉ

Bước 2: Người dùng nhập thông tin của phòng: chọn loại phòng(có 3 loại: phòng trọ, căn hộ và nhà nguyên căn), nhập giá phòng, diện tích, và hình ảnh.

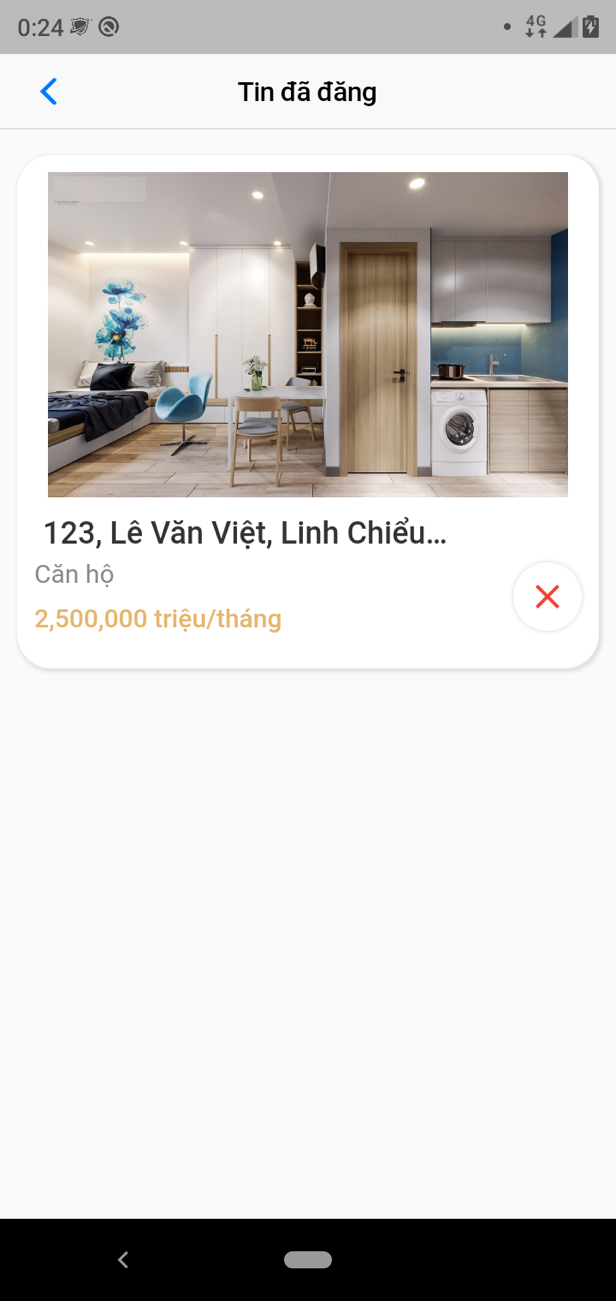
Bước 3: Nhập tiêu đề bài đăng và mô tả chi tiết cho phòng trọ bạn muốn cho thuê.



Hình 3. 11 Giao diện nhập thông tin phòng Hình 3. 12 Mô tả thông tin phòng

* + 1. **Giao diện tin bạn đã đăng**

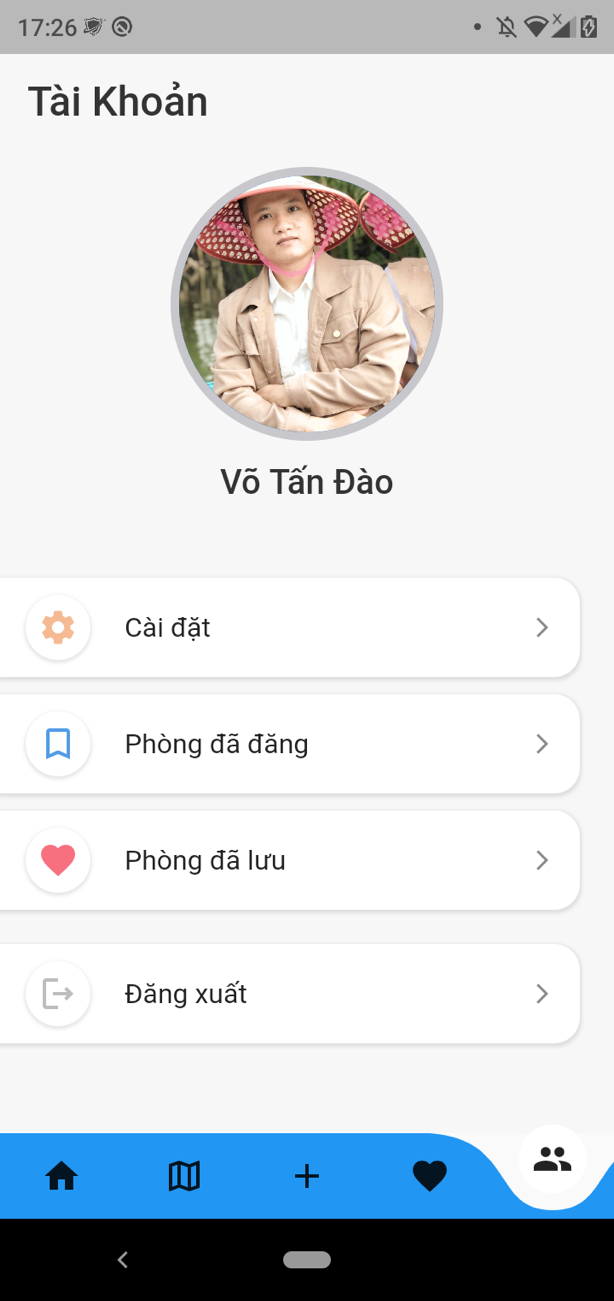
Sau khi đăng nhập tin cho thuê thành công thì ứng dụng sẽ lưu lại thông tin bài đăng và hiển thị ở trang chủ, trong phần Profile của người dùng thì có chức năng “Phòng đã lưu”. Ở đây người dùng có thể xem lại phòng mình đã đăng và có thể xoá tin đó đi nếu phòng đã được cho thuê.



Hình 3. 13 Giao diện danh sách bài đăng của bạn

* + 1. **Giao diện Profile**

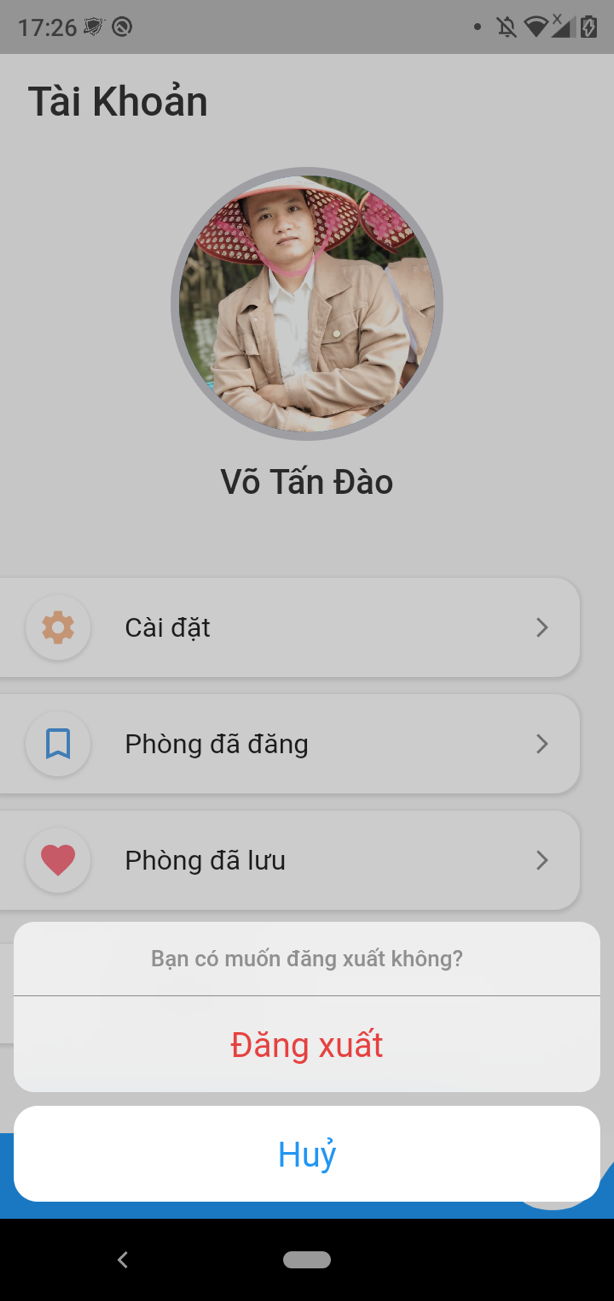
Sau khi đăng nhập thành công thì ứng dụng sẽ lưu lại thông tin người dùng trên CSDL và hiển thị trên màn hình này. Ở màn hình này còn có các chức năng như: “Phòng đã đăng”, “Phòng đã lưu”, “Đăng xuất”



Hình 3. 14 Màn hình profile

* + 1. **Màn hình đăng xuất**

Ở màn hình này khi người dùng muốn đăng xuất khỏi ứng dụng thì ứng dụng sẽ hiển thị thông báo cho người dùng”Bạn có muốn đăng xuất không”. Nếu người dùng bấm”Đăng xuất” thì tài khoản sẽ đăng xuất và quay về màn hình đăng nhập.

****

Hình 3. 15 Màn hình đăng xuất

# **KẾT LUẬN VÀ KIẾN NGHỊ**

1. **Kết quả đạt được**

Trong quá trình nghiên cứu và hoàn thành đồ án tốt nghiệp với đề tài “**PHÁT TRIỂN** **ỨNG DỤNG TÌM KIẾM PHÒNG TRỌ TẠI THÀNH PHỐ HỒ CHÍ MINH**” em đã đạt được các kết quả sau:

**Về mặt kiến thức:**

* Hiểu được và sử dụng được ngôn ngữ NodeJS để xây dựng back end.
* Hiểu và sử dụng được công nghệ mới trong xây dựng ứng dụng di động chạy trên hai nền tảng android và ios. Cụ thể hiểu được ngôn ngữ dart, một ngôn ngữ hướng đối tượng, cũng như cách cách xây dựng layout, xử lý luồng dữ liệu.
* Hiểu được cách thức để xây dựng app từ việc xây dựng ý tưởng cho các chức năng, thiết kế cơ sở dữ liệu phù hợp.

**Về ứng dụng:**

Từ những kết quả đã được ở mặt kiến thức em đã xây dựng thành công ứng dụng tìm kiếm phòng trọ ở khu vực TP. Hồ Chí Minh. Ứng dụng có các chức năng đăng bài, người thuê trọ có thể xem các thông tin về phòng, lưu lại thông tin về phòng yêu thích, tìm kiếm, xem địa chỉ và cách đi đến địa chỉ trên google map, … từ đó có thể tiết kiệm thời gian cho cả bên cho thuê và bên thuê.

**Về mặt con người**

Trong quá trình làm đồ án mặc dù thời gian làm còn hạn chế, tuy nhiên cũng đã cho em được rất nhiều kiến thức, em có thể biết thêm về công nghệ mới, vận dụng các kiến thức lập trình xây dựng đồ án cũng là cách em cũng cố và nâng cao kĩ năng. Vì đề tài này chỉ làm cá nhân nhưng việc phải làm full-stack cũng khiến em gặp nhiều khó khăn, tuy nhiên không vì thế mà có thể làm khó được mình, em đã nổ lực và hoàn thành đề tài được giao

**2. Tồn tại**

Bên cạnh những điểm đã đạt được, thì đồ án của em vẫn còn tồn tại nhiều hạn chế có thể kể đến như: chức năng còn ở mức độ cơ bản, giao diện chưa tối ưu, chưa có thông báo cho người dùng.

**3. Hướng phát triển**

Trong tương lai nếu như có điều kiện đồ án của em sẽ được điều chỉnh và phát triển theo các hướng sau:

* Xây dựng hệ cơ sở dữ liệu đầy đủ hơn
* Khảo sát thực tế đối với người sử dụng để có những yêu cầu chính xác hơn
* Xây dựng chức năng ưu tiên hiển thị trên bảng tin thông qua việc thanh toán online, tích hợp banking,…

Với lý do thời gian thực hiện có thời hạn, cũng như kiến thức của bản thân còn nhiều hạn chế do vậy trong quá trình và kết quả hoàn thành đồ án còn nhiều hạn chế và thiếu sót, rất mong nhận được sự góp ý của quý Thầy Cô để đồ án của em có thể hoàn thiện và phát triển hơn trong tương lai.

# **PHỤ LỤC**

**Phụ lục 1: hướng dẫn cài đặt**

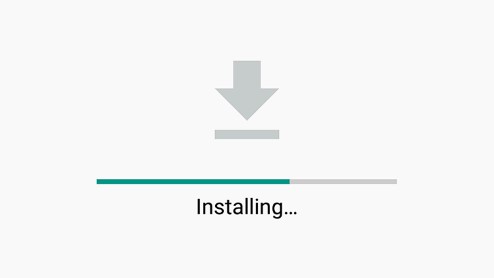
1. **Cách 1:** Quét mã scan dưới đây để tải ứng dụng.



1. **Cách 2:** 
   * + [Download file APK](https://apkfab.com/find-room/com.example.clean_achitecture/apk?h=c9b2f7cbb9ace45cfe94e86ae7e449572c31cd2202843afe87b7eff6223e838d)



* + - Cài đặt file APK.



# **TÀI LIỆU THAM KHẢO**

*[1]. Book Quick Start Guide to Dart Program,* tác giả *Sanjib Sinha, 30/11/2019*

*[2]. Book Web development with node and express: Leveraging the javascript Stack,* tác giả *Ethan Brown, 07/2014*

[3]. *MongoDB: The Definitive Guide,* tác giả *Kristina Chodorow, Michael Dirolf, 08/09/2018*

[4]. *Learn NodeJS in 1 Day: Complete Node JS Guide with Examples,* tác giả *Krishna Rungta, 20/12/2016*

[5]. *REST API Development with Node.js: Manage and Understand the Full Capabilities of Successful REST Development, tác giả Fernando Doglio, 19/07/2018*

[6]. <https://flutter.dev/>, “*Tài liệu chính thức của google để tìm hiểu về Flutter*” , truy cập vào lúc 16/05/2022

[7]. <https://dart.dev/>, “*Tài liệu chính thức của google để tìm hiểu về ngôn ngữ Dart*”, truy cập vào lúc 19/05/2022

[8]. [https://nodejs.org/api/,](https://nodejs.org/api/) “*Tài liệu chính thức để tìm hiểu về Nodejs*”, truy cập vào lúc 12/05/2022