

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN**

ĐÀO NHÂM PHÚC

PHẠM HOÀNG SANG

ĐỒ ÁN MÔN HỌC

Công nghệ lập trình đa nền tảng cho ứng dụng di động

TP. HỒ CHÍ MINH, 12 - 2023

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN**

ĐỒ ÁN MÔN HỌC

Công nghệ lập trình đa nền tảng cho ứng dụng di động

SINH VIÊN THỰC HIỆN

**ĐÀO NHÂM PHÚC MSSV: 22550017
PHẠM HOÀNG SANG MSSV: 22550019**

GIẢNG VIÊN HƯỚNG DẪN

Thầy Võ Ngọc Tân

TP. HỒ CHÍ MINH, 12 - 2023

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
TRƯỜNG ĐẠI HỌC ĐỘC LẬP - Tự Do - Hạnh Phúc
CÔNG NGHỆ THÔNG TIN

TP. HCM, ngày 19 tháng 12 năm 2023

NHẬN XÉT ĐỒ ÁN MÔN HỌC

Tên đồ án:

Ứng dụng quản lý phòng trọ bằng React Native Framework

Nhóm SV thực hiện:

Họ và tên: Đào Nhâm Phúc

MSSV: 22550017

Họ và tên: Phạm Hoàng Sang

MSSV: 22550019

Giảng viên phụ trách:

Thầy Võ Ngọc Tân

Đánh giá Đồ án:

1. Về cuốn báo cáo:

Số trang:

Số bảng số liệu:

Số tài liệu tham khảo:

Số chương:

Số hình vẽ:

Sản phẩm:

Một số nhận xét về hình thức cuốn báo cáo:

2. Về nội dung nghiên cứu:

3. Về thái độ làm việc của sinh viên:

Đánh giá chung:

Điểm sinh viên:

Đào Nhân Phúc:...../10

Phạm Hoàng Sang:...../10

Người nhận xét
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Trải qua thời gian học tập và rèn luyện cùng Thầy và các bạn, bản thân chúng em đã tiếp thu được rất nhiều những kiến thức.

Chúng em xin chân thành cảm ơn Thầy Võ Ngọc Tân đã tận tình giảng dạy, hướng dẫn và giúp đỡ chúng em trong suốt quá trình học tập.

Cuối cùng, chúng em chúc Thầy công tác tốt và luôn dồi dào sức khỏe để có thể cống hiến hết mình cho những thế hệ sinh viên tiếp theo.

Chúng em xin chân thành cảm ơn!

Sinh viên thực hiện

Đào Nhâm Phúc

Phạm Hoàng Sang

ĐỀ CƯƠNG CHI TIẾT

TÊN ĐỒ ÁN: Ứng dụng quản lý phòng trọ bằng React Native Framework	
Cán bộ hướng dẫn: Thầy Võ Ngọc Tân	
Thời gian thực hiện: Từ ngày 13-12-2023 đến ngày 27-12-2023	
Sinh viên thực hiện: Đào Nhâm Phúc – 22550017 Phạm Hoàng Sang – 22550019	
Nội dung đồ án: <ul style="list-style-type: none">- Giới thiệu về ứng dụng quản lý phòng trọ- Xây dựng các giao diện bằng React Native Framework- Call API và thao tác dữ liệu với MongoDB- Demo phần mềm- Kết luận và hướng phát triển	
Kế hoạch thực hiện: Thiết kế giao diện ứng dụng người dùng và kết nối cơ sở dữ liệu	
Xác nhận của Giảng viên (Ký tên và ghi rõ họ tên)	TP. HCM, ngày 27 tháng 12 năm 2023 Sinh viên (Ký tên và ghi rõ họ tên) Đào Nhâm Phúc, Phạm Hoàng Sang

MỤC LỤC

I. Giới thiệu về React Native Framework và ứng dụng	1
1. React Native là gì?	1
2. React Native hoạt động như thế nào?	1
3. Ưu điểm và nhược điểm của React Native	1
3.1. Ưu điểm	1
3.2. Nhược điểm	2
4. Ứng dụng.....	2
II. Giới thiệu về MongoDB & Call API	2
1. Tổng quan về MongoDB.....	2
1.1. Khái niệm	2
1.2. Một số đặc điểm của MongoDB	3
1.3. Lợi thế của MongoDB so với các cơ sở dữ liệu dạng quan hệ(RDBMS).....	3
1.4. Hoạt động của MongoDB là gì	3
1.5. Hệ cơ sở dữ liệu MongoDB có tính năng gì?	4
2. Call API là gì?.....	5
3. Kết nối với MongoDB thông qua một API bằng Node.js như thế nào?	5
3.1. Tạo RESTful API	5
3.2. Cấu hình Node.js với Express framework	6
3.3. Tương tác với MongoDB sử dụng Mongoose.....	7
3.4. Tạo RESTful API và xử lý logic hành động	8
III. Giới thiệu về đề tài quản lý phòng trọ	9
1. Giới thiệu	9
2. Công nghệ sử dụng	9
3. Đối tượng sử dụng.....	9
4. Chức năng của ứng dụng.....	9
IV. Giao diện ứng dụng và chức năng	10
1. Giao diện Đăng nhập – Đăng ký	10
2. Giao diện Trang chủ.....	11
3. Giao diện Dịch vụ	12
4. Giao diện giao diện Phòng.....	13
5. Giao diện giao diện Người thuê	14
6. Giao diện giao diện Sự cố	15
7. Giao diện giao diện Hướng dẫn	16
8. Giao diện giao diện Cài đặt.....	17
8.1. Danh sách tài khoản	17

8.2. Chính sách.....	18
8.3. Điều khoản sử dụng.....	19
9. Các giao diện còn đang trong quá trình Cập nhật	20
V. Kết luận và hướng phát triển.....	21
1. Kết luận	21
2. Hướng phát triển	21
TÀI LIỆU THAM KHẢO.....	22

DANH MỤC CHỮ VIẾT TẮT

STT	Ký hiệu chữ viết tắt	Chữ viết đầy đủ
1	CSDL	Cơ sở dữ liệu
2	IOS	Hệ điều hành (Viết tắt của iPhone Operating System) là một hệ điều hành của iPhone.
3	Android	Hệ điều hành có mã nguồn mở dựa trên nền tảng Linux do chính Google phát hành.
4	API	Application Programming Interface
5	HTTP	HyperText Transfer Protocol (giao thức truyền tải siêu văn bản)
6	HTTPS	(Hypertext Transfer Protocol Secure) là giao thức truyền tải siêu văn bản an toàn
7	URL	Uniform Resource (Hệ thống định vị tài nguyên thống nhất; được gọi một cách thông thường là một địa chỉ web)

I. Giới thiệu về React Native Framework và ứng dụng

1. React Native là gì?

React Native là một framework mã nguồn mở được phát triển bởi Facebook. Framework này giúp phát triển app mobile trên cả 2 hệ sinh thái IOS và Android một cách dễ dàng. Việc lập trình sẽ diễn ra nhanh chóng, cắt giảm chi phí và sửa chữa trực quan. Sự ra đời của React Native là lời giải đáp cho bài toán về hiệu năng và tối ưu hóa ngôn ngữ cho mọi nền tảng di động.



2. React Native hoạt động như thế nào?

React Native hoạt động bằng cách tích hợp cho ứng dụng di động 2 thread là JS thread và Main thread.

Main thread giữ vai trò cập nhật giao diện người dùng UI và xử lý tương tác người dùng ngay sau đó. Còn JS thread sẽ thực thi và xử lý các code javascript. Hai thread này hoạt động hoàn toàn độc lập với nhau. Vì thế phải sử dụng một cầu nối bridge để hai thread này có thể tương tác, giao tiếp được với nhau nhưng không phụ thuộc lẫn nhau. Nó chỉ là sự vận hành và chuyển đổi dữ liệu từ thread này sang thread khác.

3. Ưu điểm và nhược điểm của React Native

3.1. Ưu điểm

Trước tiên là React Native sở hữu các ưu điểm sau:

- **Tiết kiệm thời gian học:** Việc học từng loại ngôn ngữ cho từng nền tảng thường rất khó và mất nhiều thời gian. Tuy nhiên, với React Native, chỉ cần học duy nhất một bộ công cụ.

- Tái sử dụng code: Trong lập trình phần mềm, React Native là công cụ tái sử dụng code hiệu quả nhất mang lại các lợi thế như duy trì ít code, tận dụng tốt nguồn nhân lực,...
- Cộng đồng lớn: React Native đang ngày càng phổ biến và nhận được sự đóng góp của nhiều lập trình viên. Đặc biệt nó được xây dựng và hỗ trợ cho Facebook với lượng lớn cộng tác viên hoạt động năng nổ.
- Hot reloading: Với React Native, sẽ không cần phải tốn quá nhiều thời gian để tổng hợp app mỗi khi có sự thay đổi. Chỉ cần làm mới app trong thiết bị, emulator hoặc simulator.

3.2. Nhược điểm

Vì React Native là một framework mới nên nó cũng tồn tại những nhược điểm như:

- React Native vẫn còn thiếu khá nhiều component quan trọng..
- React Native không xây dựng được các app IOS trên Linux và Window.
- Không sử dụng để viết các chương trình game có cách chơi phức tạp và đòi hỏi tính năng đồ họa cao.
- React Native không có khả năng xây dựng được các app quá phức tạp nếu như bạn không có thêm sự hiểu biết về các ngôn ngữ lập trình khác như Objective-C, Swift, Java,...

4. Ứng dụng

React Native đã được sử dụng rộng rãi để xây dựng nhiều ứng dụng nổi tiếng trên cả hai nền tảng di động là iOS và Android. Dưới đây là một số ứng dụng nổi tiếng đã được phát triển bằng React Native:

- Facebook: Một phần của ứng dụng di động của Facebook được xây dựng bằng React Native, bao gồm một số tính năng như Marketplace.
- Instagram: Một số thành phần trong ứng dụng Instagram, như trình xem stories, được xây dựng bằng React Native.
- Skype: Skype đã sử dụng React Native để tái thiết kế ứng dụng di động của họ trên nền tảng iOS và Android.

II. Giới thiệu về MongoDB & Call API

1. Tổng quan về MongoDB

1.1. Khái niệm

MongoDB là một chương trình cơ sở dữ liệu mã nguồn mở được thiết kế theo kiểu hướng đối tượng trong đó các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt

này nên MongoDB có thể được dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data).

1.2. Một số đặc điểm của MongoDB

- Kho lưu định hướng Document: Dữ liệu được lưu trong các tài liệu kiểu JSON.
- Lập chỉ mục trên bất kỳ thuộc tính nào.
- Các truy vấn đa dạng.
- Cập nhật nhanh hơn.

1.3. Lợi thế của MongoDB so với các cơ sở dữ liệu dạng quan hệ(RDBMS)

- Ít Schema hơn: MongoDB là một cơ sở dữ liệu dựa trên Document, trong đó một Collection giữ các Document khác nhau.
- Số trường, nội dung và kích cỡ của Document này có thể khác với Document khác.
- Cấu trúc của một đối tượng là rõ ràng.
- Không có các Join phức tạp.
- Khả năng truy vấn sâu hơn.
- MongoDB hỗ trợ các truy vấn động trên các Document bởi sử dụng một ngôn ngữ truy vấn dựa trên Document mà mạnh mẽ như SQL.
- MongoDB dễ dàng để mở rộng.
- Việc chuyển đổi/ánh xạ của các đối tượng ứng dụng đến các đối tượng cơ sở dữ liệu là không cần thiết.
- Sử dụng bộ nhớ nội tại để lưu giữ phần công việc, giúp truy cập dữ liệu nhanh hơn.

1.4. Hoạt động của MongoDB là gì

Nguyên tắc hoạt động của MongoDB là dưới một tiến trình dịch vụ ngầm và mở một cổng (mặc định là cổng 27017), để có thể tiếp nhận các yêu cầu truy vấn, thao tác; sau đó tiến hành xử lý.

Mỗi bản ghi của MongoDB (document) được gán một trường có tên “_id” nhằm xác định tính duy nhất của bản ghi. Có thể hiểu id này như tên gọi của một bản ghi và dùng phân biệt chúng với các bản ghi khác. Đồng thời, nó còn được sử dụng cho mục đích truy vấn hoặc tìm kiếm thông tin. Trường dữ liệu “_id” được tự động đánh chỉ mục (index) để đảm bảo tốc độ truy vấn đạt hiệu suất tối ưu.

Mỗi truy vấn dữ liệu đều được ghi đệm lên bộ nhớ RAM nên các truy vấn sau đó sẽ diễn ra nhanh hơn. Bởi nó không cần đọc dữ liệu từ ổ cứng.

Khi thực hiện thêm, xóa hay sửa bản ghi thì MongoDB đều mất 60s để ghi các dữ liệu được thay đổi từ RAM xuống ổ cứng. Điều này nhằm mục đích đảm bảo hiệu suất mặc định của chương trình.

1.5. Hệ cơ sở dữ liệu MongoDB có tính năng gì?

MongoDB có những tính năng cơ bản được nhấn mạnh như sau:

➤ Truy vấn ad hoc:

Truy vấn ad hoc được xem là tính năng tốt nhất, có khả năng hỗ trợ các trường, truy vấn các phạm vi và tìm kiếm các biểu thức. Từ đó trả về các trường tài liệu cụ thể bao gồm các hàm JavaScript do người dùng xác định hoặc truy vấn được cấu hình trả về mẫu có kích thước nhất định ngẫu nhiên. Ngoài ra, các trường trong MongoDB có thể được dùng để lập các mục chính, chỉ mục phụ.

➤ Nhân rộng:

Đây là tính năng Replica set bao gồm hai hoặc nhiều bản sao dữ liệu, mỗi bản sao có thể giữ vai trò chính hoặc phụ. Trong quá trình nhân rộng, những dữ liệu ghi và đọc thực hiện trên bản sao chính, bản sao thứ cấp sẽ dùng để tích hợp để duy trì dữ liệu từ bản sao chính. Với trường hợp bản sao chính thất bại, Replica set chọn bản sao thứ cấp để làm bản sao chính tiếp theo.

➤ Cân bằng tải:

Với cách sử dụng các Sharding, MongoDB sẽ chia theo tỷ lệ chiều ngang để người dùng có thể chọn một Shard key. Nhìn chung, dữ liệu được chia thành các phạm vi sau đó sẽ phân phối đồng đều qua các Shard key

➤ Lưu trữ tệp:

Với tính năng này, MongoDB sử dụng như một hệ thống tệp GridFS để cân bằng tải, sao chép dữ liệu đến nhiều máy tính. GridFS sẽ chia các tệp thành nhiều phần, nhiều đoạn để lưu trữ thành các tài liệu riêng biệt.

➤ Tập hợp:

Tính năng này, chương trình cung cấp 3 cách chính để tập hợp là Aggregation Pipeline, Single-purpose Aggregation và Mapreduce. Trong đó, Aggregation Pipeline được ghi nhận có hiệu suất tốt hơn hầu hết những hoạt động tổng hợp.

➤ Thực thi Javascript:

JavaScript thường thực thi trong truy vấn, các hàm tổng hợp được và được gửi thủ tục đến cơ sở dữ liệu.

➤ Giới hạn kích thước:

MongoDB hỗ trợ collection có kích thước cố định, từ đó làm tăng các hiệu suất các hoạt động khác có liên quan đến dữ liệu. Nếu như dữ liệu vượt qua mức giới hạn, các tài liệu cũ sẽ bị xóa mà không cần thiết nhập bất cứ dòng lệnh nào.

2. Call API là gì?

"Call API" đề cập đến việc gửi yêu cầu từ một ứng dụng hoặc một phần mềm để truy cập dữ liệu hoặc các chức năng từ một dịch vụ hoặc một hệ thống khác thông qua một giao thức cụ thể được định nghĩa trong API.

API (Application Programming Interface) là một tập hợp các quy tắc, giao thức và công cụ mà các ứng dụng sử dụng để tương tác với nhau. API định nghĩa cách mà các phần mềm khác nhau có thể giao tiếp với nhau, cho phép chúng truyền thông và chia sẻ dữ liệu một cách an toàn và hiệu quả.

Khi "call API," chính là đang gửi một yêu cầu từ một ứng dụng hoặc một phần mềm đến một API cụ thể. Yêu cầu này có thể là một trong những phương thức chung như GET, POST, PUT, DELETE, PATCH, vv., đóng vai trò quy định loại hoạt động bạn muốn thực hiện với dữ liệu hoặc tài nguyên được quản lý bởi API.

Khi "call" một API, bạn đang sử dụng một giao thức cụ thể (như HTTP hoặc HTTPS) để gửi yêu cầu đến một địa chỉ cụ thể (URL) của API, sau đó API sẽ xử lý yêu cầu và trả về kết quả, thông tin hoặc thực hiện hành động được yêu cầu trở lại cho ứng dụng để xử lý tiếp theo.

3. Kết nối với MongoDB thông qua một API bằng Node.js như thế nào?

3.1. Tạo RESTful API

Tạo RESTful API là một kiểu thiết kế ứng dụng trong đó quy định định dạng của địa chỉ URL và phương thức HTTP sử dụng để quản lý tài nguyên trên server.

Tài nguyên trên Server trong bài viết này mình giả sử là danh sách person trong Database.

Việc thiết kế REST cũng khá đơn giản, chúng ta sẽ định nghĩa 4 hàm xử lý 4 actions:

- Tạo mới tài nguyên (ví dụ thêm một room).
- Chỉnh sửa tài nguyên có sẵn (ví dụ cập nhập thông tin của một room).
- Lấy về dữ liệu của tài nguyên (xem thông tin của một room /hoặc nhiều room).
- Xóa tài nguyên (Xóa thông tin của một room).

3.2. Cấu hình Node.js với Express framework

Để bắt đầu, chúng ta sẽ tạo mới một dự án với các dependencies cần thiết. Từ cửa sổ lệnh, chúng ta thực hiện các lệnh sau:

```
npm init -y  
npm install express body-parser mongoose --save
```

Kết thúc những lệnh trên, chúng ta sẽ tạo một package.json, và cài đặt ExpressJS, Mongoose và một package hỗ trợ chuyển dữ liệu trong cách request từ client thành JSON data.

Tạo tệp index.js tại thư mục gốc của dự án:

```
const express = require('express');  
const mongoose = require('mongoose');  
  
const app = express();  
app.use(express.json());  
const routes = require('./routes/routes');  
app.use('/api', routes)  
require('dotenv').config();  
const mongoString = process.env.DATABASE_URL  
mongoose.connect(mongoString);  
  
const database = mongoose.connection;  
app.use('/images', express.static('images'));  
database.on('error', (error)=>{  
  console.log(error)  
})  
database.once('connected', ()=>{  
  console.log('Database Connected');  
})  
app.listen(3000, () => {  
  console.log(` Server Started at ${3000}`)
```

```
})
```

Đầu tiên, chúng ta cần require những module cần thiết như: express, mongoose,... và định nghĩa API,... Cuối cùng, chúng ta khai báo ứng dụng sẽ lắng nghe port 3000.

3.3. Tương tác với MongoDB sử dụng Mongoose

Để có thể connect được với MongoDB, bạn cần khai báo đúng đường dẫn trong file env:

```
DATABASE_URL=mongodb://0.0.0.0:27017/QLPhongTro
```

Sau khi đã cấu hình kết nối xong, mình sẽ tiến hành tạo model. Trong khuôn khổ ví dụ của bài viết này, chúng ta chỉ cần tạo một model “user” như sau:

```
const mongoose = require('mongoose');
const dataSchema = new mongoose.Schema({
  username: {
    require: true,
    type: String
  },
  password: {
    require: true,
    type: String
  },
  email: {
    require: true,
    type: String
  },
})
module.exports = mongoose.model('user', dataSchema)
```

Model này sẽ tạo một user collection (tương đương với tạo một table trong SQL vậy). Với document này sẽ có 3 thông tin: username, password và email.

Tiếp theo chúng ta ta tiến hành xử lý logic cho mỗi API.

3.4. Tạo RESTful API và xử lý logic hành động

Ví dụ chúng ta sẽ viết code xử lý phần tạo mới một user:

```
router.post('/DangKy', async (req, res) => {  
  const data = new user({  
    username: req.body.username,  
    password: req.body.password,  
    email: req.body.email })  
  try {  
    const dataToSave = await data.save();  
    res.status(200).json(dataToSave) }  
  catch (error){  
    res.status(400).json({ message: error.message }) } })
```

Khi client thực hiện một POST request tới ứng dụng của chúng ta. Chúng ta sẽ sử dụng user và JSON payload để lưu thông tin vào cơ sở dữ liệu.

Có một điểm lưu ý, tất cả những công việc tương tác với cơ sở dữ liệu đều có thể tốn thời gian và bạn cần phải sử dụng cơ chế xử lý bất đồng bộ. Trước kia bạn có thể dùng Promise, và giờ thì hiện đại hơn, dùng kỹ thuật mới async và await (hỗ trợ từ javascript ES7).

Tiếp theo, khi đã có dữ liệu, chúng ta có thể lấy dữ liệu từ database:

```
router.get('/getAllUser', async (req, res) => {  
  try {  
    const data = await user.find();  
    res.json(data)  
  }  
  catch (error){  
    res.status(500).json({ message: error.message })  
  }  
})
```

Có hai kiểu lấy dữ liệu: Một là lấy toàn bộ danh sách user có trong database, hai là lấy cụ thể thông tin của một người nào đó. Trong kịch bản API này, khi client không gửi bất kì điều kiện nào thì sẽ trả về một danh sách các user. Và còn các phương thức khác như put và delete.

III. Giới thiệu về đề tài quản lý phòng trọ

1. Giới thiệu

Trong thời đại công nghệ phát triển mạnh mẽ như hiện nay, việc sử dụng ứng dụng công nghệ để quản lý và tối ưu hóa các hoạt động kinh doanh, đặc biệt là quản lý phòng trọ, đã trở thành xu hướng không thể tránh khỏi. Đề án này giới thiệu ứng dụng quản lý phòng trọ hiệu quả, giúp tối ưu hóa quá trình quản lý và cung cấp trải nghiệm cho thuê phòng tốt nhất dành cho người dùng.

2. Công nghệ sử dụng

Ứng dụng được phát triển dựa trên React Native Framework & kết nối CSDL với MongoDB bằng nodejs.

3. Đối tượng sử dụng

Đối tượng là các cá nhân hay doanh nghiệp hiện đang kinh doanh trong lĩnh vực cho thuê phòng trọ. Phần mềm quản lý phòng trọ sau đây được thiết kế đơn giản nên dễ sử dụng đối với mọi lứa tuổi.

4. Chức năng của ứng dụng

Khi có khách đến thuê trọ, ứng dụng quản lý phòng trọ sẽ giúp người dùng lưu trữ thông tin cá nhân của khách đến thuê trọ. Những thông tin bao gồm: Họ tên, số điện thoại, ngày tháng năm sinh, cccd,...

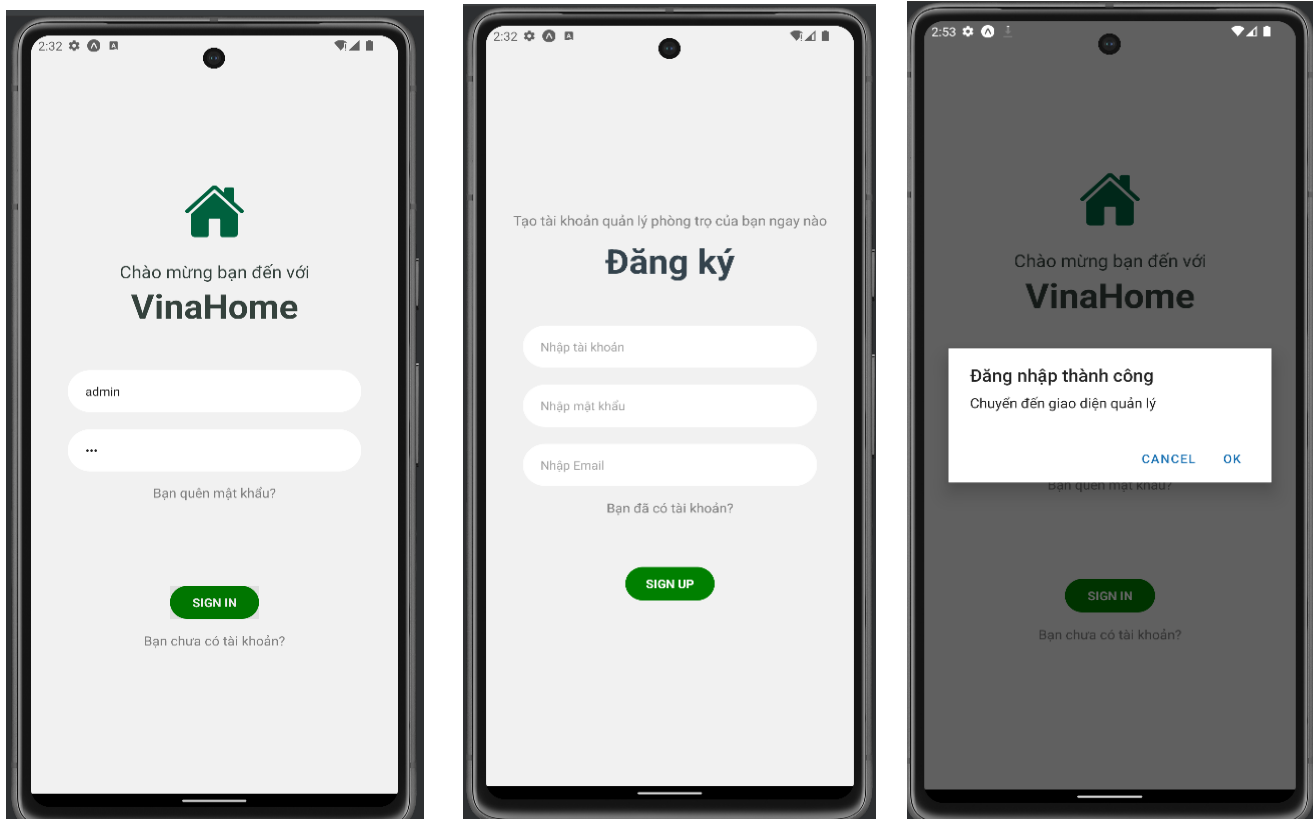
Đồng thời lưu giữ phòng, các dịch vụ mà khách thuê. Ứng dụng giúp kiểm tra tình trạng các phòng của nhà trọ, xem đang có phòng trống, phòng nào có người ở, danh sách khách thuê, các sự cố đang xảy ra.

Hiện tại ứng dụng gồm 4 chức năng chính:

- Quản lý dịch vụ.
- Quản lý phòng.
- Quản lý người thuê.
- Quản lý sự cố.

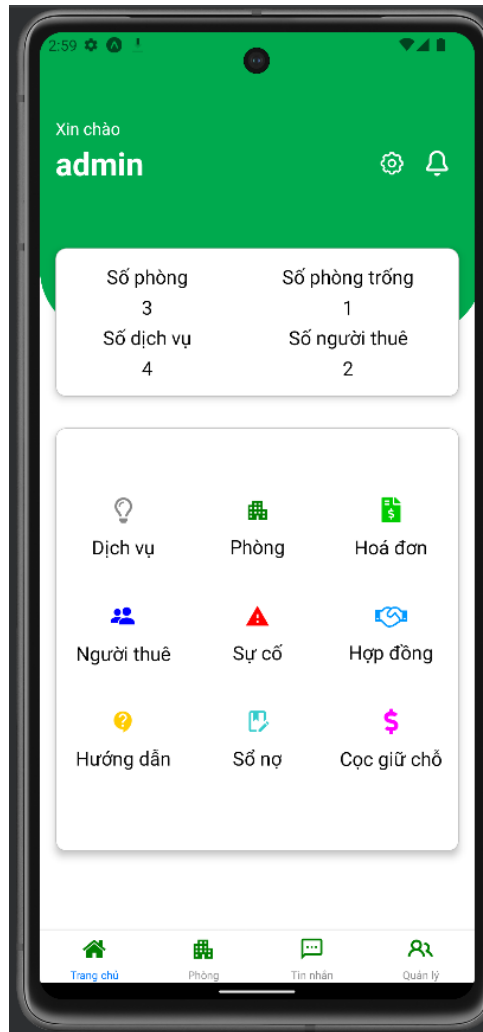
IV. Giao diện ứng dụng và chức năng

1. Giao diện Đăng nhập – Đăng ký



Sau khi tạo tài khoản, dữ liệu sẽ được lưu lại trên MongoDB, người dùng sẽ dùng tài khoản vừa đăng kí để đăng nhập, nếu tài khoản và mật khẩu trùng với thông tin có trong phần mềm sẽ được chuyển đến trang chủ.

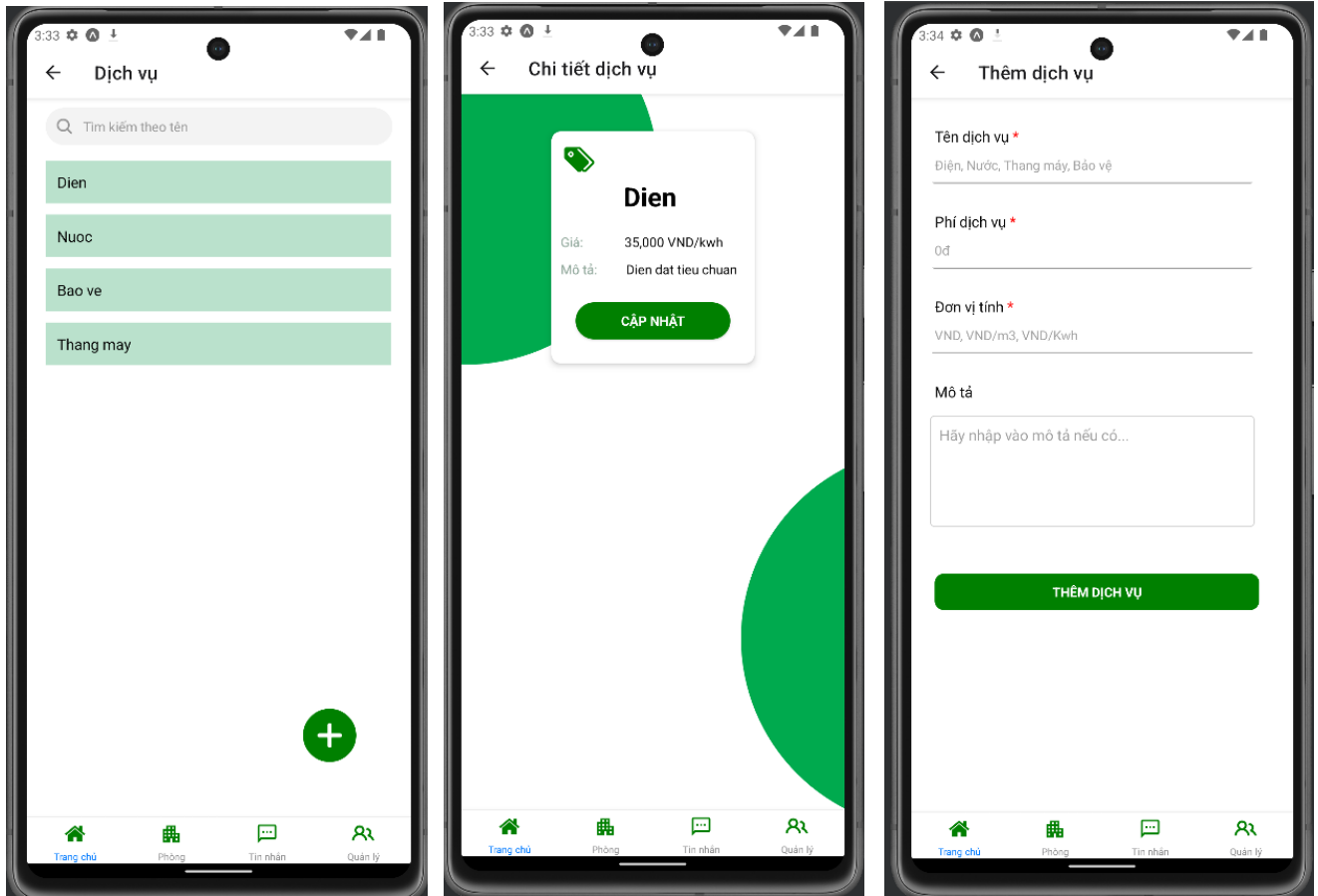
2. Giao diện Trang chủ



Trang chủ cung cấp thông tin, chức năng tổng quan về ứng dụng, số lượng phòng, dịch vụ, ... và các thông tin cần thiết khác.

Các màu sắc, font chữ, icon đều thể hiện sự thân thiện với người dùng

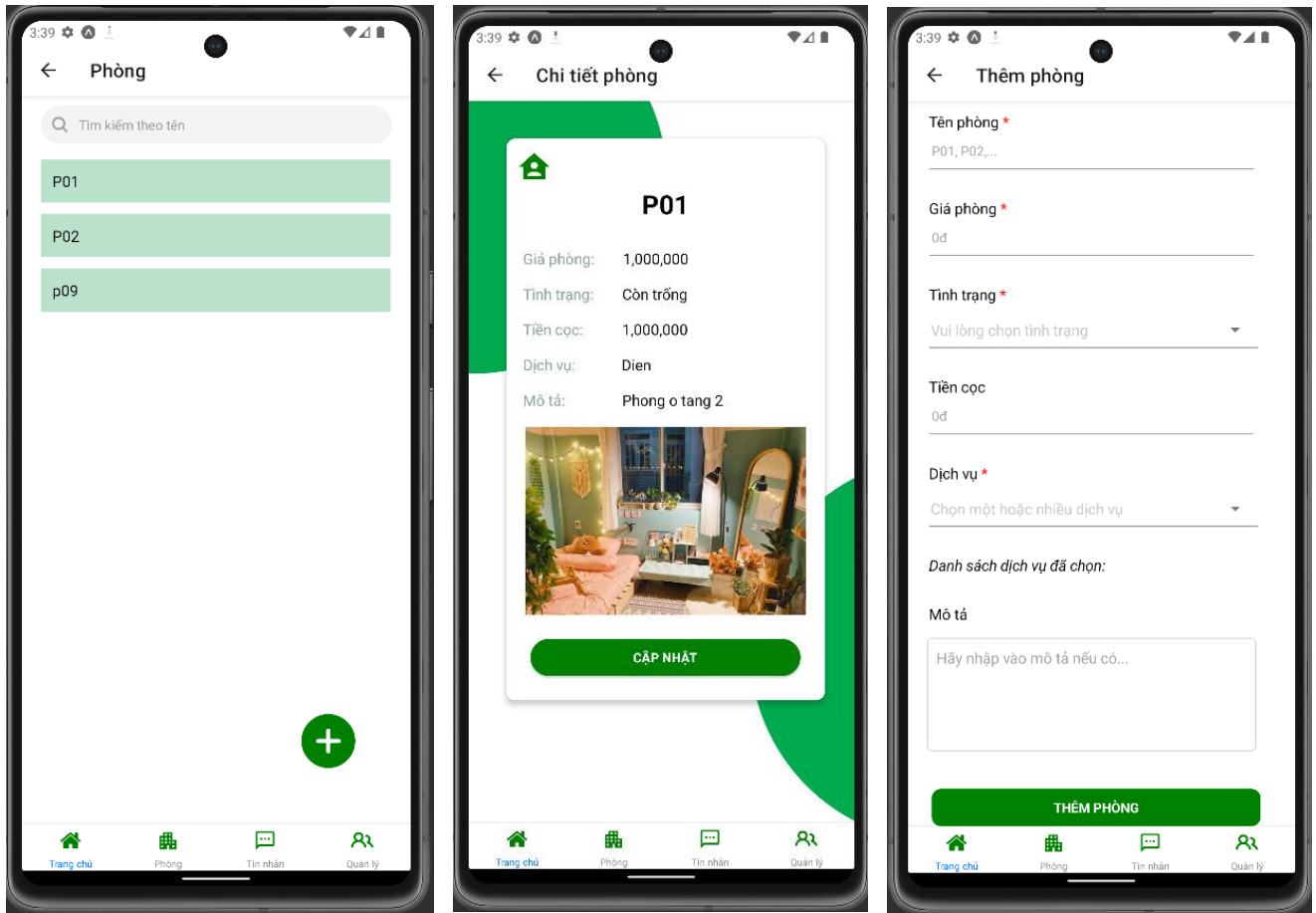
3. Giao diện Dịch vụ



Giao diện tính năng Dịch Vụ sẽ giúp người dùng dễ dàng quản lý các dịch vụ mà Phòng trọ cho thuê.

Ở đây người dùng có thể Thêm dịch vụ, Tìm kiếm các dịch vụ, Xem được danh sách dịch vụ và Chi tiết dịch vụ.

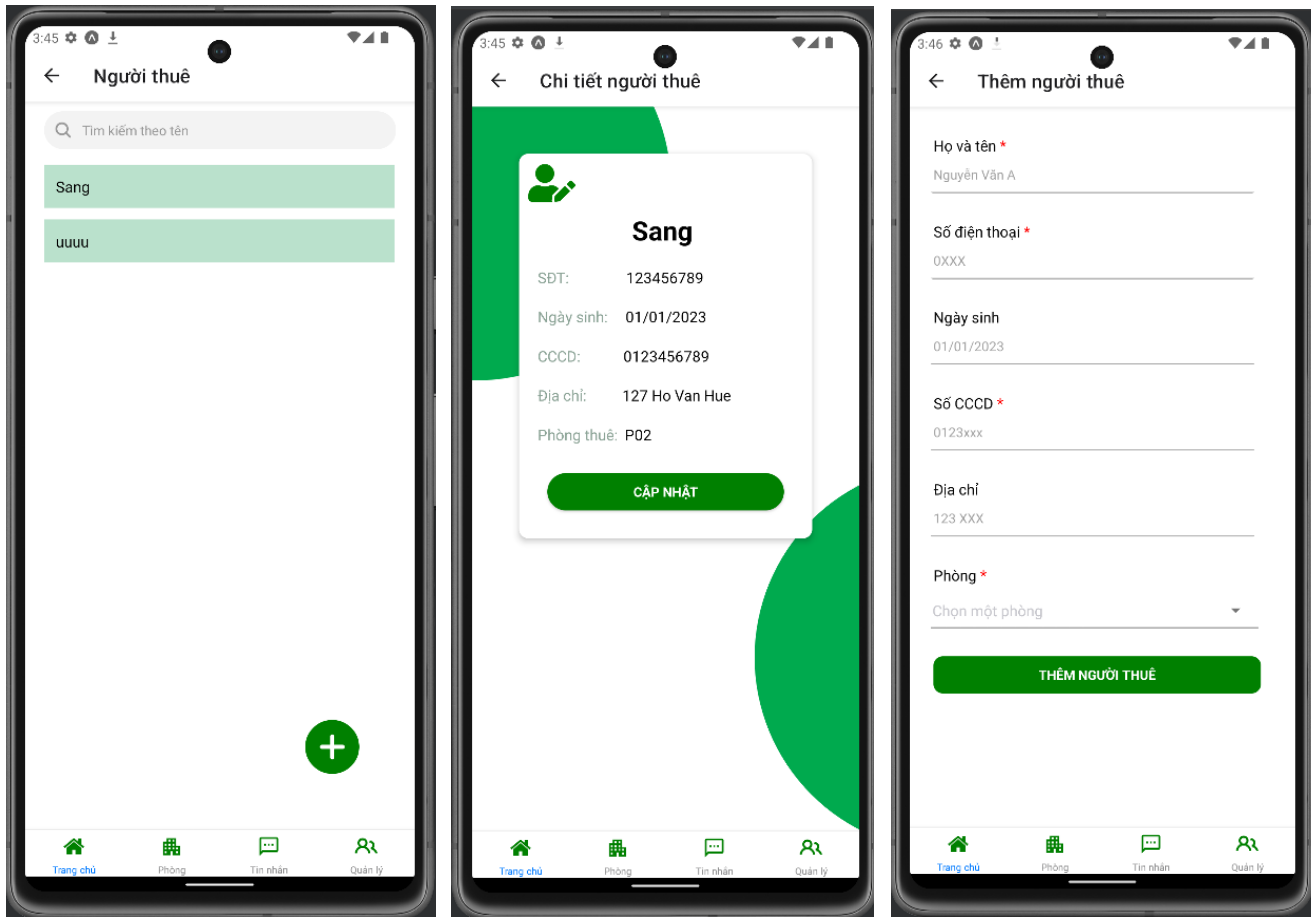
4. Giao diện giao diện Phòng



Giao diện tính năng Phòng sẽ giúp người dùng dễ dàng quản lý tất cả các Phòng và tìm kiếm khi số lượng nhiều.

Ở đây người dùng có thể Thêm Phòng, Tìm kiếm các Phòng và xem chi tiết của từng Phòng để có thể dễ dàng quản lý.

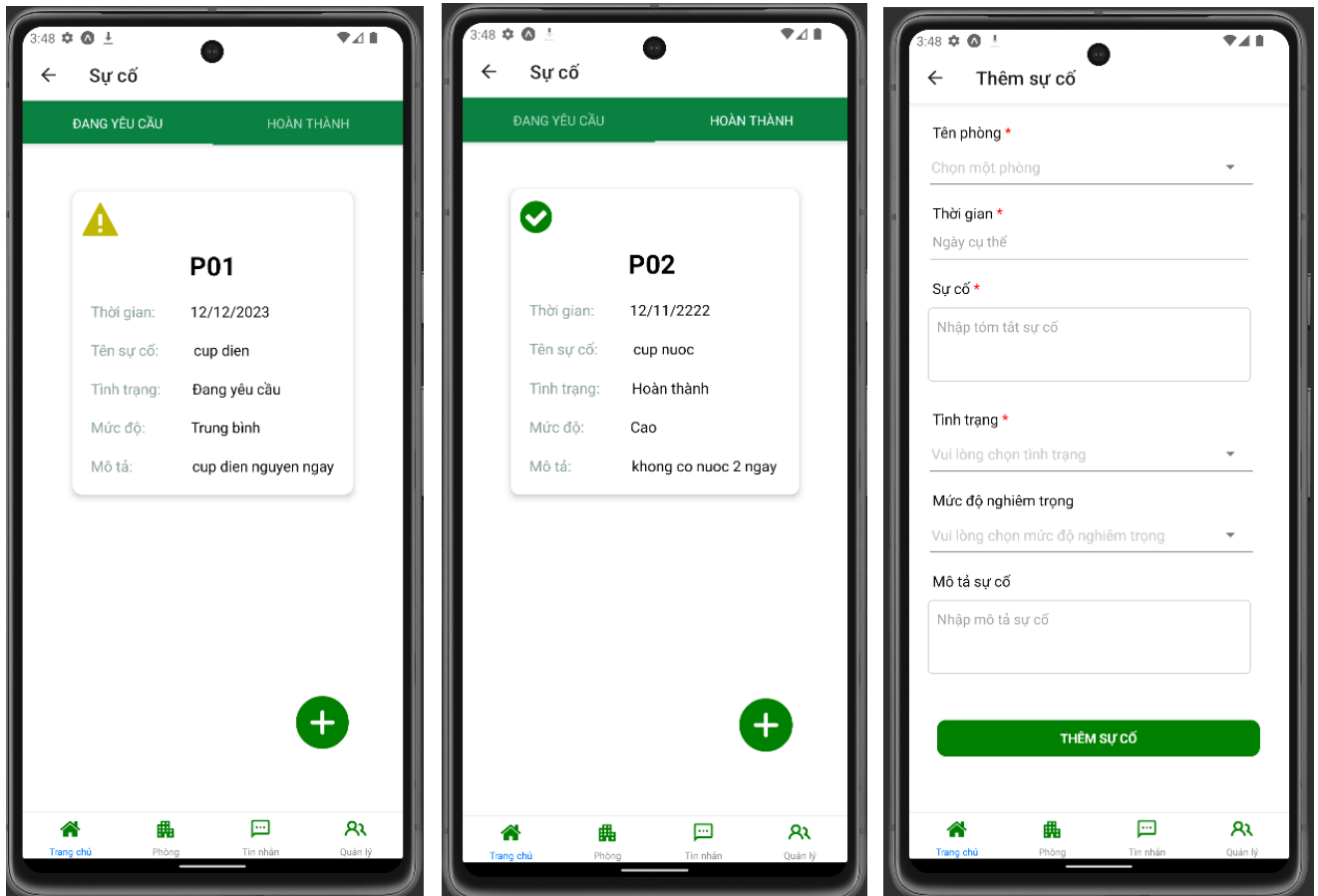
5. Giao diện giao diện Người thuê



Giao diện tính năng Người thuê sẽ giúp người dùng dễ dàng quản lý tất cả các Người thuê và tìm kiếm Người thuê đó nếu cần thiết.

Ở đây người dùng có thể Thêm Người thuê, Tìm kiếm Người thuê và xem chi tiết thông tin của Người thuê.

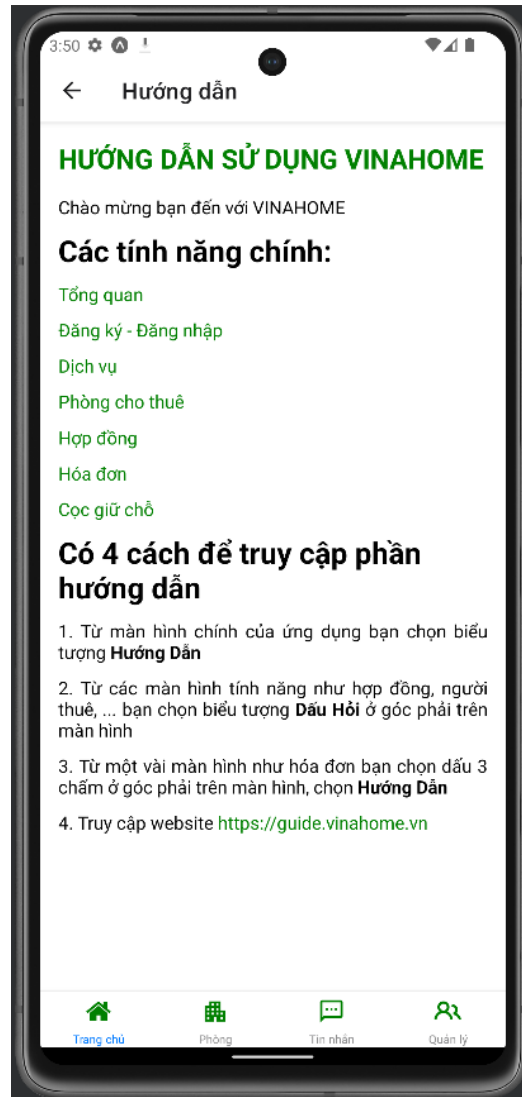
6. Giao diện giao diện Sự cố



Giao diện tính năng Sự cố sẽ giúp người dùng dễ dàng quản lý tất cả các Sự cố đang xảy ra và đã hỗ trợ xong ở các Phòng trọ của mình .

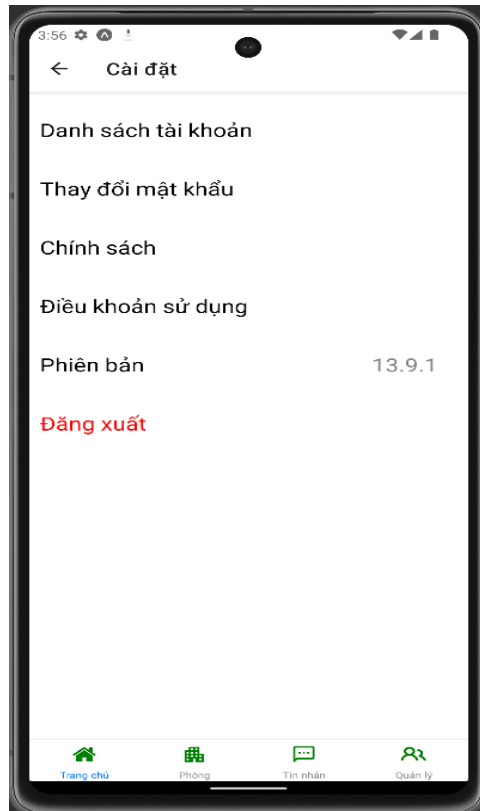
Ở đây người dùng có thể Thêm Sự cố, và xem chi tiết thông tin của Người thuê.

7. Giao diện giao diện Hướng dẫn

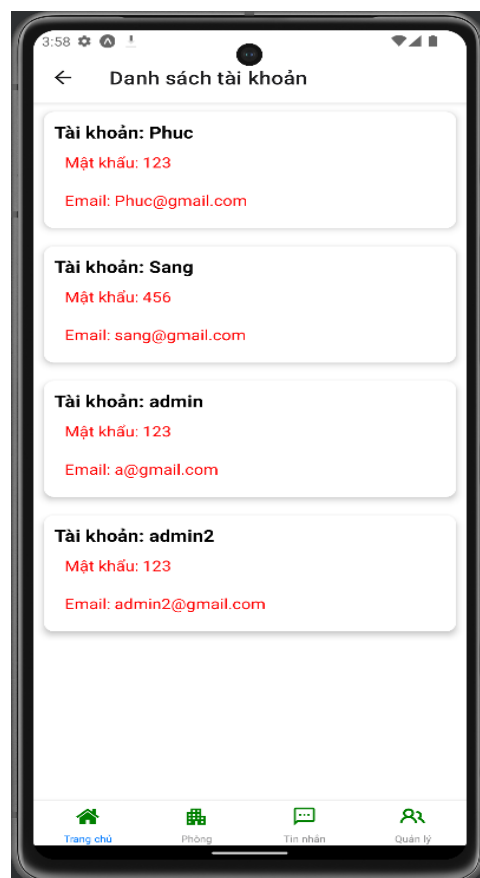


Ở đây, trang hướng dẫn sẽ giới thiệu các tính năng của phần mềm và các đề truy phần mềm.

8. Giao diện giao diện Cài đặt

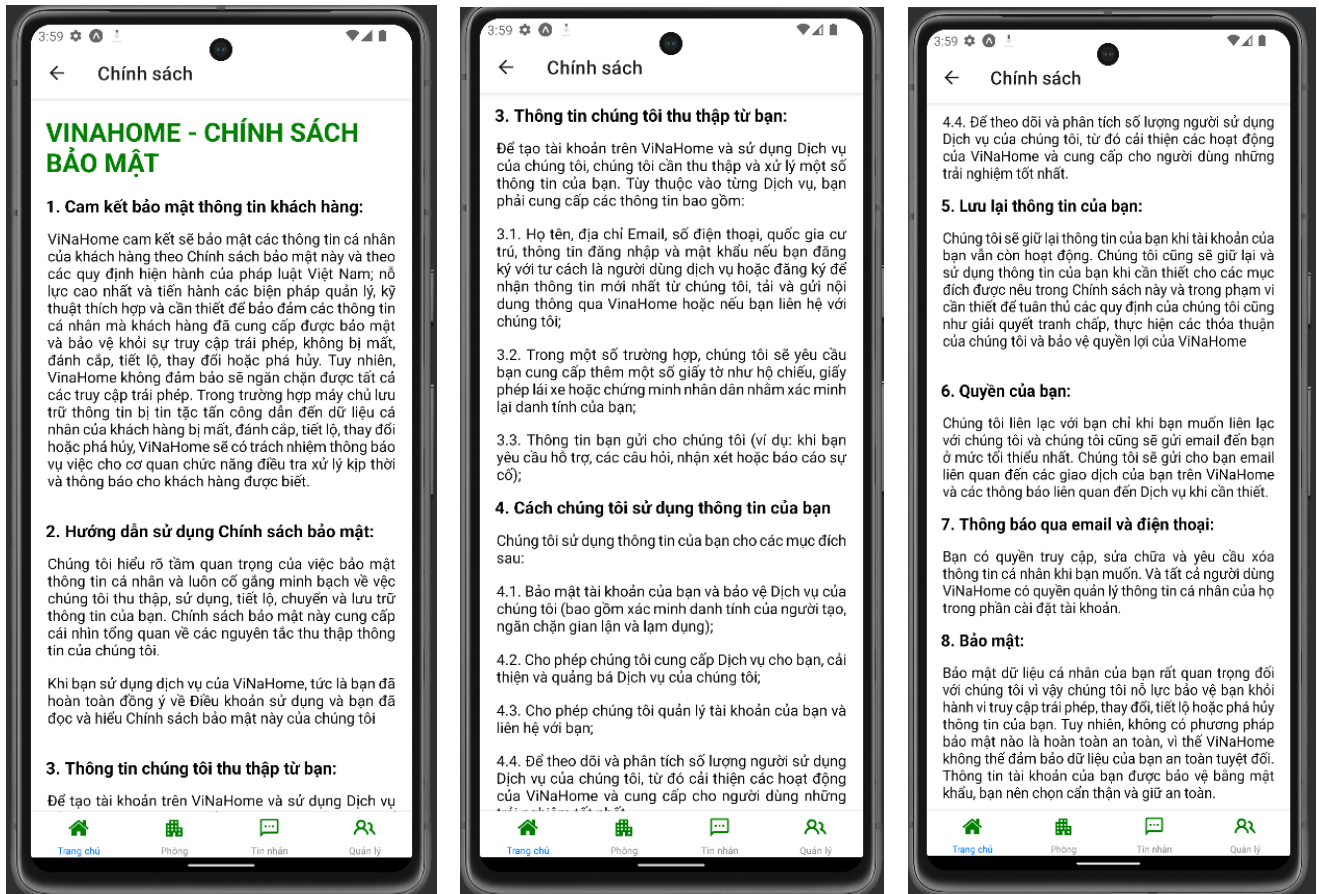


8.1. Danh sách tài khoản



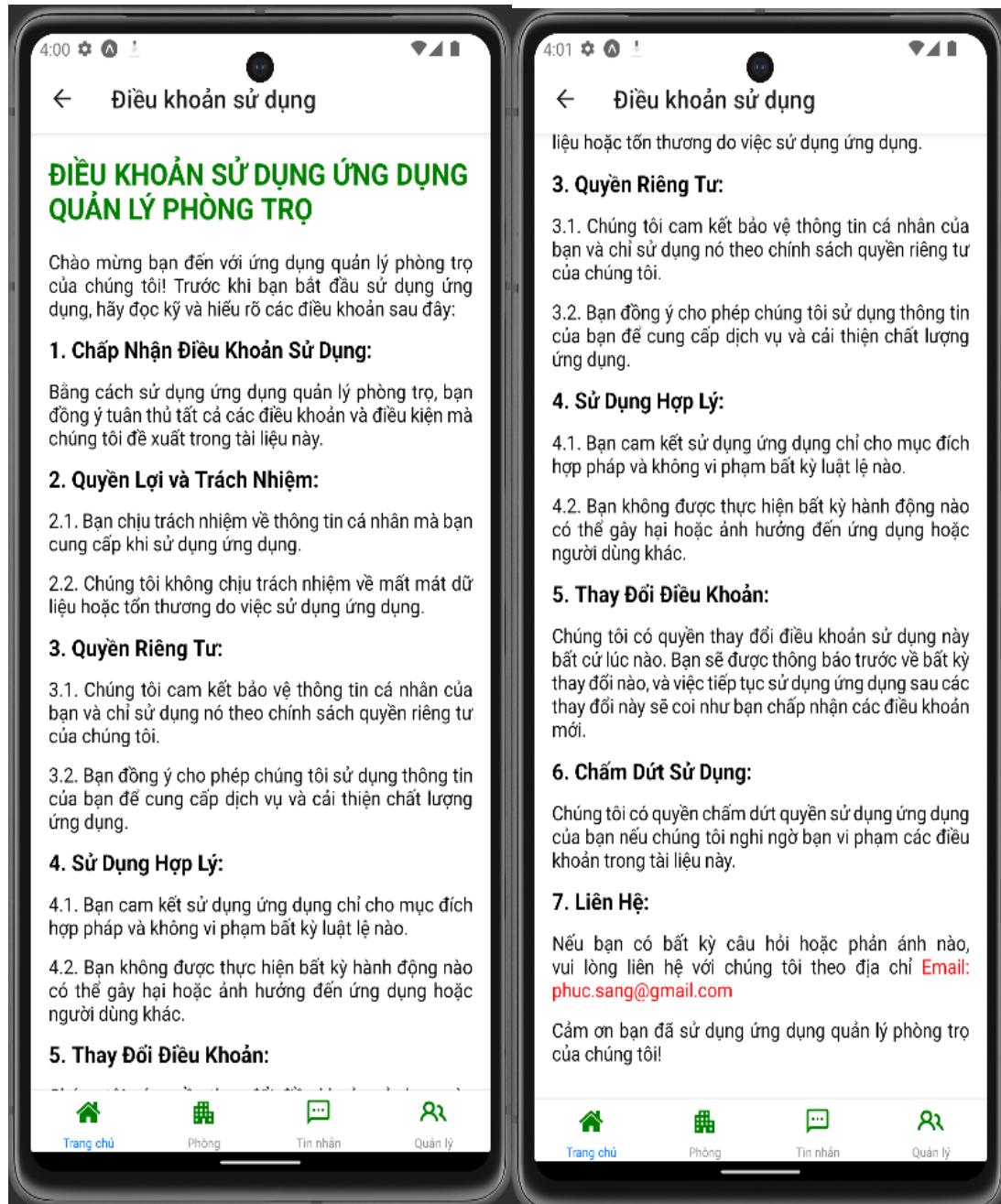
Giao diện các tài khoản đã được đăng ký thành công.

8.2. Chính sách



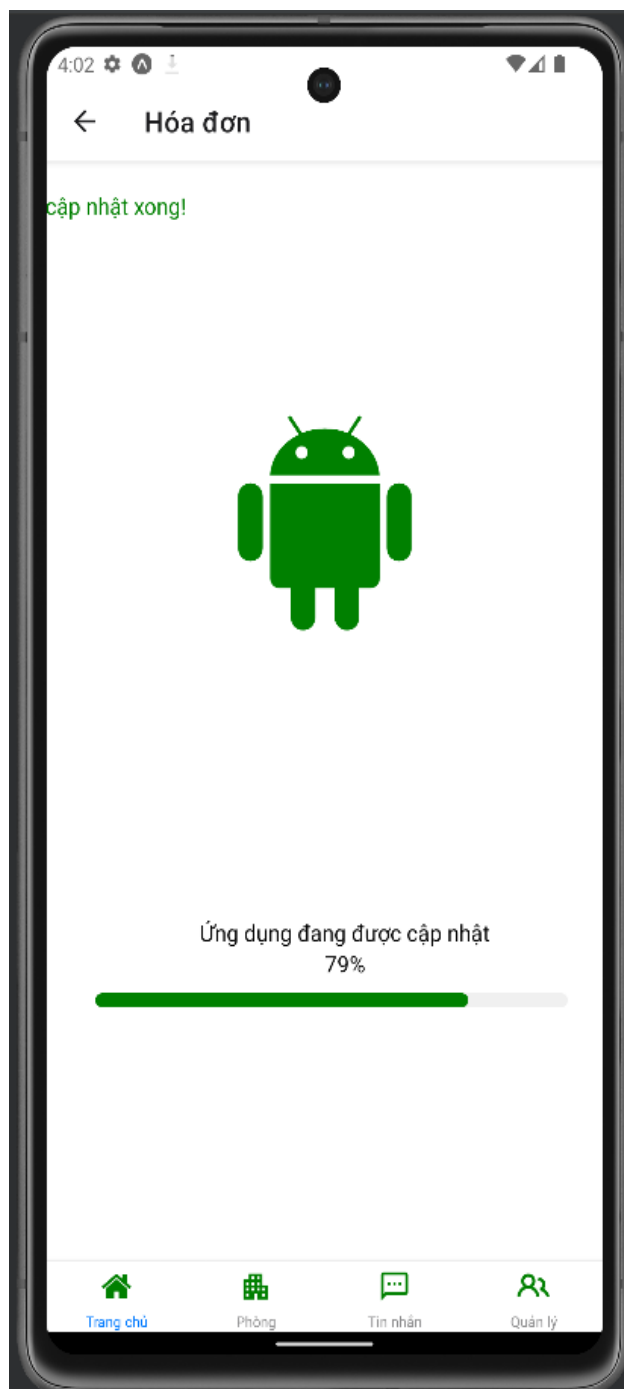
Ở đây người dùng sẽ được thông báo về các chính sách bảo mật.

8.3. Điều khoản sử dụng



Ở đây người dùng sẽ đọc được các điều khoản khi sử dụng ứng dụng.

9. Các giao diện còn đang trong quá trình Cập nhật



Các giao diện như Hóa đơn, Hợp đồng, ... còn đang trong quá trình cập nhật sẽ được hiển thị thông báo “Vui lòng quay lại sau khi ứng dụng được cập nhật xong!”.

V. Kết luận và hướng phát triển

1. Kết luận

Sau một thời gian nghiên cứu thực hiện đề tài thì chúng em cũng hoàn thành đề tài với kết quả khả quan. Tuy nhiên do các điều kiện khách quan cũng như chủ quan và đặc biệt là do công việc học hành và công việc đang đi làm của chúng em nên bài nghiên cứu còn vài thiếu sót mong Thầy thông cảm.

Nhìn chung, chương trình thực hiện cơ bản đầy đủ các chức năng của yêu cầu trong công việc quản lý phòng trọ. Làm bước đệm để nâng cấp thêm các tính năng chuyên sâu của hệ thống để hoàn thiện hơn và thêm nhiều tác vụ để dễ quản lý.

2. Hướng phát triển

- Nâng cấp thêm các tính năng chuyên sâu để quản lý thành một hệ thống nhiều dãy trọ.
- Nâng cấp thêm tính năng bảo trì cho các thiết bị trong phòng trọ.
- Tạo tiền đề để phát triển thêm các hệ thống quản lý khác như: Ký túc xá, khu dân cư,...

TÀI LIỆU THAM KHẢO

<https://reactnative.dev/docs/components-and-apis>

<https://icons.expo.fyi/Index>

<https://aboutreact.com/react-native-app-intro-slider/>

<https://aboutreact.com/react-native-axios/>