

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO TỔNG KẾT

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN
NĂM 2021**

**Nghiên cứu xây dựng ứng dụng quản lý
đồ dùng trong gia đình**

Sinh viên thực hiện:

Khuất Văn Chung

Trịnh Thị Hồng

Nguyễn Huy Sơn

Trần Quốc Nam

Nguyễn Đức Dương

Lớp: CNTT2K59

Lớp: CNTT2K59

Lớp: CNTT2K59

Lớp: CNTT2K59

Lớp: CNTT2K59

Khoa: CNTT

Khoa: CNTT

Khoa: CNTT

Khoa: CNTT

Khoa: CNTT

Người hướng dẫn: ThS. Nguyễn Đức Dư

Hà Nội, Ngày ... Tháng ... Năm 2021

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN
NĂM 2021

**Nghiên cứu xây dựng ứng dụng quản lý
đồ dùng trong gia đình**

Sinh viên thực hiện:

Khuất Văn Chung **Nam, Nữ: Nam** **Dân tộc: Kinh**

Lớp: CNTT2-K59 **Khoa: CNTT** **Năm thứ: 3/4**

Ngành học: Công nghệ thông tin

Nguyễn Huy Sơn **Nam, Nữ: Nam** **Dân tộc: Kinh**

Lớp: CNTT2-K59 **Khoa: CNTT** **Năm thứ: 3/4**

Ngành học: Công nghệ thông tin

Trịnh Thị Hồng **Nam, Nữ: Nữ** **Dân tộc: Kinh**

Lớp: CNTT2-K59 **Khoa: CNTT** **Năm thứ: 3/4**

Ngành học: Công nghệ thông tin

Trần Quốc Nam **Nam, Nữ: Nam** **Dân tộc: Kinh**

Lớp: CNTT2-K59 **Khoa: CNTT** **Năm thứ: 3/4**

Ngành học: Công nghệ thông tin

Nguyễn Đức Dương **Nam, Nữ: Nam** **Dân tộc: Kinh**

Lớp: CNTT2-K59 **Khoa: CNTT** **Năm thứ: 3/4**

Ngành học: Công nghệ thông tin

Người hướng dẫn: ThS. Nguyễn Đức Dư

MỤC LỤC

| | |
|-----------------------------------------------------|----|
| LỜI CẢM ƠN..... | 4 |
| MỞ ĐẦU | 5 |
| 1. Tổng quan lĩnh vực nghiên cứu thuộc đề tài..... | 5 |
| 2. Lí do lựa chọn đề tài..... | 5 |
| 3. Mục tiêu đề tài | 6 |
| 4. Tính mới và sáng tạo:..... | 6 |
| CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI..... | 7 |
| 1. JAVA | 7 |
| 2. Android | 9 |
| 3. C#..... | 10 |
| 4. ASP.NET CORE..... | 11 |
| 5. Firebase | 12 |
| 6. Lập trình ESP8266..... | 13 |
| 7. Websocket | 14 |
| 8. SQL Server | 15 |
| CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ..... | 17 |
| I. Phân tích:..... | 17 |
| 1. Sơ đồ use case các chức năng của người dùng..... | 17 |
| 2. Phân tích các chức năng của hệ thống..... | 18 |
| 3. Biểu đồ hoạt động..... | 23 |
| II. Thiết kế..... | 26 |
| 1. Thiết kế giao diện | 26 |
| 2. Đặc tả giao diện. | 36 |
| 3. Thiết kế cơ sở dữ liệu | 39 |
| CHƯƠNG 3: KẾT QUẢ NGHIÊN CỨU ĐẠT ĐƯỢC | 40 |
| 1. Một số giao diện kết quả | 40 |
| 2. Đánh giá kết quả..... | 47 |
| CHƯƠNG 4: KẾT LUẬN | 48 |
| TÀI LIỆU THAM KHẢO | 49 |

LỜI CẢM ƠN

“Có một nghề bụi phấn bám đầy tay
Người ta bảo đó là nghề trong sạch nhất
Có một nghề không trông cậy vào đất
Mà mang lại cho Đời đầy “trái ngọt hoa tươi””

Đó chính là nghề giáo viên cao quý, một nghề đã ban tặng cho chúng em những người thầy, người cô đáng quý, đáng trân trọng. Với tình cảm chân thành và lòng biết ơn sâu sắc, chúng em xin được gửi đến quý thầy cô ở Khoa Công nghệ Thông tin trường Đại học Giao thông Vận tải đã tận tâm chỉ dạy và truyền đạt cho chúng em vô vàn những kiến thức quý báu từ những ngày đầu tiên chúng em bước chân vào cánh cổng trường đại học.

Trong suốt quá trình thực hiện đề tài này, ngoài những cố gắng của bản thân, chúng em sẽ không thể hoàn thành tốt được đề tài này nếu không có sự giúp đỡ, quan tâm và hướng dẫn tận tình của thầy Nguyễn Đức Dư. Chúng em xin gửi lời cảm ơn sâu sắc nhất đến thầy.

Trong quá trình nghiên cứu đề tài, chúng em đã cố gắng hết sức mình. Xong, một phần vì thời gian có hạn, một phần là vì chúng em bước đầu đi vào thực tế còn hạn chế và có nhiều bỡ ngỡ nên không tránh khỏi những sai sót. Chúng em rất mong nhận được những ý kiến đóng góp của quý thầy cô để đề tài này của chúng em được hoàn thiện hơn. Đồng thời giúp chúng em trau dồi, bổ sung thêm nhiều kiến thức trong lĩnh vực này.

Chúng em xin chân thành cảm ơn!

Hà Nội, ngày tháng năm 2021
Nhóm sinh viên thực hiện đề tài

MỞ ĐẦU

1. Tổng quan lĩnh vực nghiên cứu thuộc đề tài.

Ngày nay, Điện thoại thông minh – thiết bị công nghệ đã cách mạng xã hội trong chưa đầy một thập kỷ. Có thể dễ dàng nhận ra một chiếc điện thoại thông minh nhờ khả năng tích hợp email, lịch, trình tổ chức và các ứng dụng mạnh mẽ được trình bày một cách đơn giản và trực quan. Bằng cái nhìn thực tế ta cũng có thể thấy tác động mạnh mẽ của smartphone đến mọi mặt. Từ công việc, đời sống, giải trí, giáo dục... con người giờ đây không còn xem điện thoại là một thiết bị thông thường mà nó đã trở thành một “vật bất ly thân”.

Điện thoại thay thế đồng hồ báo thức truyền thống để đánh dấu một ngày mới tràn đầy năng lượng. Điện thoại với tính năng ghi chú, nhắc nhở... thay thế những danh sách kế hoạch dài dằng dặc cho một ngày... Mọi thứ mà trước đây chỉ có thể làm được với một chiếc máy tính cồng kềnh nay đã có thể hoàn thành tốt đẹp chỉ với smartphone có thể gửi email....Và cũng chỉ với một chiếc điện thoại chúng ta cũng hoàn toàn có thể quản lý được những đồ dùng trong gia đình mình, thời hạn sửa chữa, bảo dưỡng,...và điện thoại cũng hoàn toàn có thể gửi đến chúng ta thông báo khi chúng ta quên mất lịch sửa chữa, bảo dưỡng định kì.

2. Lí do lựa chọn đề tài.

Trong thời đại công nghệ phát triển mạnh mẽ trên toàn cầu, ứng dụng Công nghệ được áp dụng vào mọi lĩnh vực: kinh tế, văn hóa, y tế, giáo dục, đời sống xã hội... mang lại sự thay đổi cục diện mạnh mẽ trong quá trình phát triển của loài người. Ở đâu có sử dụng công nghệ, ở đó sẽ có sự hiện đại, tiện ích nhanh chóng và độ chính xác cao. Trong đời sống thực tế hiện nay, chúng em nhận thấy hiện nay mọi người trong gia đình thường không thể quản lý chính xác được số tiền đã dùng, số lượng các đồ dùng đã mua nhưng không nhớ gây đến việc mua lại và trở nên thừa không dùng đến. Hơn nữa, việc sửa chữa và bảo hành không được thực hiện đúng thời gian thường dễ gây ra hỏng các thiết bị, đồ dùng trong gia đình gây ra những thiệt hại lãng phí về kinh tế không đáng có.

Chính vì những vấn đề trên nên chúng em quyết định xây dựng một ứng dụng quản lý đồ dùng cá nhân và gia đình để có thể tiện quản lý và số hóa những

thông tin của hóa đơn mua, sửa chữa và bảo dưỡng giúp việc quản lý chi tiêu trong gia đình trở nên dễ dàng, chính xác hơn.

a) Quy trình mua đồ:

- Khi người dùng mua đồ chỉ cần lưu lại thông tin khi mua đồ và các hình ảnh đi kèm

b) Quy trình sửa chữa, bảo dưỡng:

- Khi đi bảo dưỡng lưu lại thông tin cần thiết và đặt lịch cho lần bảo dưỡng tiếp theo khi cần thiết

3. Mục tiêu đề tài

- Nắm được quy trình xây dựng app
- Nắm được công nghệ Android, Web Api, SQL Server, Firebase
- Nâng cao khả năng lập trình
- Xây dựng app trên điện thoại quản lý đồ dung trong gia đình giúp cho người dung quản lý đồ đạc trong gia đình như lịch sử mua bán, sửa chữa, bảo dưỡng...

4. Tính mới và sáng tạo:

- Xây dựng app quản lý đồ dung trong gia đình giúp người dung quản lý đồ đạc trong gia đình như lịch sửa mua, sửa chữa, bảo dưỡng thay thế việc lưu trữ hóa đơn giấy tờ bảo dưỡng, sửa chữa...
- Quản lý thông tin đồ dùng tăng khả năng kiểm soát các đồ dung cá nhân của riêng người dung hoặc những đồ dung chung trong một hoặc nhiều môi trường khác nhau.
- Cung cấp thông tin cho người dung trong việc mua bán, sửa chữa, bảo dưỡng...
- Không cần phải giữ lại các hóa đơn về việc mua bán, sửa chữa, bảo dưỡng
- Tạo thông báo nhắc nhở cho cá nhân, thành viên
- Có thể chia các đồ dung cá nhân theo địa điểm như: ở nhà, phòng trọ...

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

Chương này sẽ giới thiệu cơ bản về những môi trường dung để phát triển đề tài.

1. JAVA

1.1. Tổng quan về Java

- Java là một ngôn ngữ lập trình dạng lập trình hướng đối tượng (OOP). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy. Bằng cách này, một ứng dụng Java có thể chạy trên nhiều nền tảng khác nhau.

- Cú pháp Java được vay mượn nhiều từ C & C++ nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn.

1.1.2 Những đặc điểm của ngôn ngữ Java

- **Máy ảo Java (JVM - Java Virtual Machine):** Tất cả các chương trình muốn thực thi được thì phải được biên dịch ra mã máy. Mã máy của từng kiến trúc CPU của mỗi máy tính là khác nhau (tập lệnh mã máy của CPU Intel, CPU Solarix, CPU Macintosh ... là khác nhau), vì vậy trước đây một chương trình sau khi được biên dịch xong chỉ có thể chạy được trên một kiến trúc CPU cụ thể nào đó. Đối với CPU Intel chúng ta có thể chạy các hệ điều hành như Microsoft Windows, Unix, Linux, OS/2, ... Chương trình thực thi được trên Windows được biên dịch dưới dạng file có đuôi .EXE còn trên Linux thì được biên dịch dưới dạng file có đuôi ELF, vì vậy trước đây một chương trình chạy được trên Windows muốn chạy được trên hệ điều hành khác như Linux chẳng hạn thì phải chỉnh sửa và biên dịch lại. Ngôn ngữ lập trình Java ra đời, nhờ vào máy ảo Java mà khó khăn nêu trên đã được khắc phục. Một chương trình viết bằng ngôn ngữ lập trình Java sẽ được biên dịch ra mã của máy ảo java (mã java bytecode). Sau đó máy ảo Java chịu trách nhiệm chuyển mã java bytecode thành mã máy tương ứng. Sun Microsystem chịu trách nhiệm phát triển các máy ảo Java chạy trên các hệ điều hành trên các kiến trúc CPU khác nhau.

- **Đơn giản:** Java được phát triển trên nền tảng C++ nên khá quen thuộc với lập trình viên C++. Tuy nhiên Java lại đơn giản hơn nhờ loại bỏ các đặc điểm phức tạp, dễ gây nhầm lẫn của C++ như nạp chồng toán tử, thao tác con trỏ, đa kế thừa...

- **Hướng đối tượng:** Hướng đối tượng trong Java tương tự như C++ nhưng Java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Tất cả mọi thứ đề cập đến trong Java đều liên quan đến các đối tượng được định nghĩa trước, thậm chí hàm

chính của một chương trình viết bằng Java (đó là hàm main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính đa kế thừa (multi inheritance) như trong C++ mà thay vào đó Java đưa ra khái niệm interface để hỗ trợ tính đa kế thừa.

- **Phân tán:** Java là ngôn ngữ thông dụng trong việc xây dựng các ứng dụng phân tán (ứng dụng mạng, web...). Java có thư viện mở rộng dùng cho lập trình phân tán như net, rmi, idl, CORBA... Các ứng dụng Java có thể truy xuất các đối tượng liên mạng dùng địa chỉ URL tương tự như truy xuất file cục bộ.

- **Mạnh mẽ:** Java có một cơ chế định kiểu mạnh, tường minh, kiểm tra lúc biên dịch và kiểm tra khi thông dịch trước khi thực thi nên giới hạn được lỗi; kiểm tra truy xuất phần tử của mảng, chuỗi lúc thực thi, kiểm tra ép kiểu run-time. Java cũng có mô hình quản lý bộ nhớ hiệu quả, tự động thu hồi bộ nhớ bằng trình gom rác – garbage collection.

- **An toàn:** Java cung cấp môi trường thực thi có kiểm tra chặt chẽ. Không bao giờ giả định mã thực thi là an toàn. Có cơ chế kiểm tra an ninh hệ thống đa tầng.

- **Khả chuyển:** Chương trình ứng dụng viết bằng ngôn ngữ Java chỉ cần chạy được trên máy ảo Java là có thể chạy được trên bất kỳ máy tính, hệ điều hành nào có máy ảo Java, đúng như thông điệp của Java: “Viết một lần, chạy mọi nơi” (Write Once, Run Anywhere).

- **Biên dịch và thông dịch:** Java là một ngôn ngữ lập trình vừa biên dịch vừa thông dịch. Chương trình nguồn viết bằng ngôn ngữ lập trình Java có đuôi *.java đầu tiên được biên dịch thành tập tin bytecode có đuôi *.class và sau đó sẽ được trình thông dịch thông dịch thành mã máy.

- **Độc lập nền:** Một chương trình viết bằng ngôn ngữ Java có thể chạy trên nhiều máy tính có hệ điều hành khác nhau (Windows, Unix, Linux, ...) miễn sao ở đó có cài đặt máy ảo java (Java Virtual Machine). Viết một lần chạy mọi nơi (write once run anywhere).

- **Đa nhiệm - đa luồng:** Java hỗ trợ lập trình đa nhiệm, đa luồng cho phép nhiều tiến trình, tiểu trình có thể chạy song song cùng một thời điểm và tương tác với nhau.

- **Hỗ trợ mạnh cho việc phát triển ứng dụng:** Công nghệ Java phát triển mạnh mẽ nhờ vào Sun Microsystem và nay là Oracle đã cung cấp nhiều công cụ, thư viện lập trình phong phú hỗ trợ cho việc phát triển nhiều loại hình ứng dụng khác nhau cụ thể như: J2SE (Java 2 Standard Edition) hỗ trợ phát triển những ứng dụng đơn, ứng dụng client-server; J2EE (Java 2 Enterprise Edition) hỗ trợ phát triển các

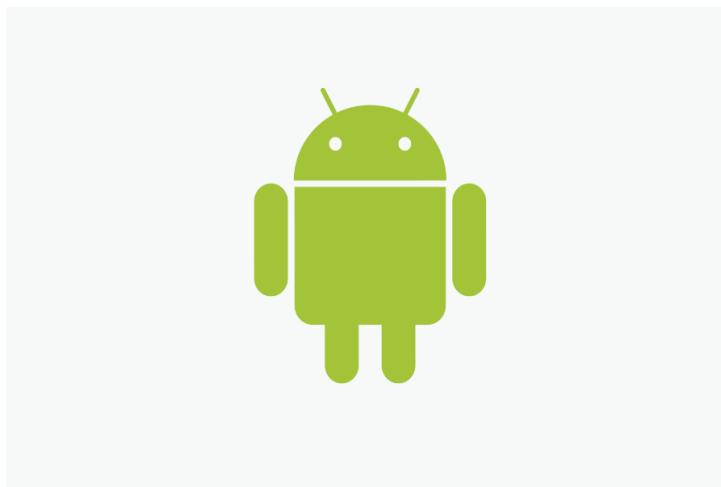
ứng dụng thương mại, J2ME (Java 2 Micro Edition) hỗ trợ phát triển các ứng dụng trên các thiết bị di động, không dây, ...

2. Android

Android là một hệ điều hành có mã nguồn mở dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng.

Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google, sau này được chính Google mua lại vào năm 2005 và hệ điều hành Android đã ra mắt vào năm 2007. Chiếc điện thoại đầu tiên chạy Android là HTC Dream được bán vào ngày 22 tháng 10 năm 2008.

Chính việc là một hệ điều hành mã nguồn mở cùng với giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới. Tại Hội nghị Google I/O 2019, Stephanie Cuthbertson - Giám đốc cấp cao của Android đã công bố kỷ lục mới mà hệ điều hành Android vừa thiết lập: "Chúng ta ở đây cùng nhau kỷ niệm một cột mốc mới, với 2.5 tỷ thiết bị Android đang hoạt động."



Hình 1: Logo Android

Hiện tại phiên bản mới nhất của Android là Android Q (10.0) với rất nhiều thay đổi và cập nhật cho người dùng. Không những vậy, các gói, các thư viện dành cho nhà phát triển và các lập trình viên được Google đưa ra và cập nhật liên tục, giúp cho lập trình viên có thể tiếp cận với các công nghệ mới nhất.

Chính nhờ môi trường phát triển năng động, cộng đồng nhà phát triển đông đảo và có tính hỗ trợ cao mà ta có thể phát triển ứng dụng của bản thân và đưa đến tay người dùng một cách dễ dàng.

3. C#

Lập trình C# đang là cái tên nhận được rất nhiều sự quan tâm từ cộng đồng lập trình. Bởi C# là ngôn ngữ lập trình phổ biến thứ 4 (sau Java, PHP, Python), với khoảng 31% các nhà phát triển sử dụng nó thường xuyên. Đây cũng là cộng đồng lớn thứ 3 trên StackOverflow với hơn 1,1 triệu chủ đề. Sự phổ biến này giúp thị trường nhân sự C# phát triển với hơn 17.000 công việc C# được quảng cáo mỗi tháng với mức lương trung bình là hơn 72.000 đô la. Nó đã tạo ra những cơ hội rất lớn cho những nhà lập trình viên

C# thường được đọc là C thăng hoặc “See Sharp” hay “C-sharp”. Nó là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Tập đoàn Microsoft. Ngôn ngữ này được xem là ngôn ngữ lập trình hướng đối tượng trong sáng và thuần nhất. Nó hiện thực hầu hết các tính chất tốt của mô hình hướng đối tượng giống như ngôn ngữ lập trình Java. C# là ngôn ngữ được Microsoft phát triển dựa trên 2 ngôn ngữ huyền thoại đó là C++ và Java. Và nó cũng được miêu tả là loại ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

C# làm việc chủ yếu trên bộ khung .NET (.NET framework). Ngôn ngữ lập trình này có khả năng tạo ra nhiều ứng dụng mạnh mẽ và an toàn cho nền tảng Windows. Các thành phần máy chủ, dịch vụ web, ứng dụng di động và nhiều khả năng khác nữa.

Các ưu điểm của lập trình C#:

C# là ngôn ngữ gần gũi với các ngôn ngữ lập trình thông dụng như C++, Java hay Pascal. Bởi nó được xây dựng trên nền tảng của những ngôn ngữ này. Chính bởi vậy mà nó sẽ được kế thừa những tinh hoa của những ngôn ngữ thông dụng trên. Khi C# ra đời nó đã giải quyết được các khiếm khuyết của C/C++ như con trỏ, các hiệu ứng phụ, ... Vì vậy nó dễ tiếp cận và dễ được phát triển hơn. Đây cũng là một lợi thế cho những ai đã học và quen thuộc với C++ hay Java khi mà học ngôn ngữ C#.

C# được hỗ trợ bởi một cộng đồng các nhà phát triển lớn. Chính cộng đồng sử dụng này đã khiến cho C# trở thành ngôn ngữ đáng tin cậy và hoạt động hiệu quả hơn. Cộng đồng sử dụng đóng một vai trò rất quan trọng trong một ngôn ngữ lập trình. Nếu cộng đồng của một ngôn ngữ lớn hơn các mục tiêu hướng đến của các công cụ thì ngôn ngữ đó sẽ xuất hiện thường xuyên trên thị trường. Cộng đồng lớn cũng có nghĩa là có rất nhiều trợ giúp và thảo luận có ích cho những người mới chân ướt chân ráo bước vào nghề.

C# có thể được sử dụng để tạo ra hầu hết mọi ứng dụng, Nhưng nó mạnh mẽ nhất là trong việc xây dựng ứng dụng và game trên desktop của Windows. C# cũng ngày càng trở nên phổ biến cho phát triển điện thoại di động và các ứng dụng

web. Một số ví dụ về phát triển Game sử dụng bộ khung C# và .net là Unity, Mono Game và CryEngine.

4. ASP.NET CORE

ASP.NET CORE là một framework mới của Microsoft. Nó được thiết kế lại từ đầu để trở nên nhanh chóng, linh hoạt và chạy trên được nhiều nền tảng khác nhau. Trong bài viết này, ASP.NET Core là một framework có thể được sử dụng để phát triển web với .NET. Nếu bạn có kinh nghiệm làm việc với MVC hoặc Web API trong một vài năm thì bạn sẽ nhận thấy những tính năng quen thuộc. Kết thúc



khóa học ASP.NET Core của NET CORE VN. Bạn sẽ có những thứ cần thiết để bắt đầu làm việc được với ASP.NET Core và viết 1 ứng dụng có thể thêm, sửa hay xem dữ liệu từ database như các môn lập trình khác bạn đã từng học.

Hình 2: ASP.NET CORE

4.1 Sơ lược về lịch sử của ASP.NET Core

ASP.NET đã được sử dụng từ nhiều năm để phát triển các ứng dụng web. Kể từ đó, framework này đã trải qua một sự thay đổi tiến hóa đều đặn và cuối cùng đã đưa chúng ta đến với ASP.NET Core 1.0 hậu duệ mới gần đây nhất của nó.

- ASP.NET Core không phải là phiên bản tiếp theo của ASP.NET 4.6
- Đó là một framework hoàn toàn mới, may mắn rằng nó một dự án side-by-side tương tự với mọi thứ mà chúng ta biết.
- Nó thực ra được viết lại trên framework ASP.NET 4.6 hiện tại những kích thước nhỏ hơn và nhiều modular hơn.

4.2 ASP.NET Core là gì?

ASP.NET Core là một web framework mã nguồn mở và được tối ưu hóa cho cloud để phát triển các ứng dụng web chạy trên nhiều nền tảng như Windows, Linux và Mac. Hiện tại, nó bao gồm MVC framework được kết hợp các tính năng của MVC và Web API thành một web framework duy nhất.

- Các ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên .NET Framework hoàn chỉnh.
- Nó đã được thiết kế để cung cấp một framework tối ưu cho các ứng dụng để triển khai tới cloud hoặc chạy on-premises.
- Nó bao gồm những modular với các thành phần tối thiểu, do đó bạn giữ được tính linh hoạt trong quá trình xây dựng các giải pháp của mình.
- Bạn có thể phát triển và chạy các ứng dụng đa nền tảng từ ASP.NET Core trên Windows, Mac và Linux.

4.3 Ưu điểm của ASP.NET Core

ASP.NET Core đi kèm với những ưu điểm sau:

- ASP.NET Core có một số thay đổi kiến trúc dẫn đến modular framework nhỏ hơn.
- ASP.NET Core không còn dựa trên System.Web.dll. Nó dựa trên một tập hợp nhiều yếu tố của Nuget packages.
- Điều này cho phép bạn tối ưu ứng dụng của mình chỉ cần những NuGet packages cần thiết.
- Lợi ích của diện tích bề mặt ứng dụng nhỏ hơn thì bảo mật chặt chẽ hơn, giảm dịch vụ, cải thiện hiệu suất và giảm chi phí.

4.4 Với ASP.NET Core, bạn có thể nhận được các cải tiến sau:

- Xây dựng và chạy các ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.
- Được xây dựng trên **.NET Core**, hỗ trợ side-by-side app versioning.
- Công cụ mới giúp đơn giản hóa việc phát triển web hiện đại.
- Liên kết đơn các web stack như Web UI và API Web.
- Cấu hình dựa trên môi trường đám mây sẵn có.
- Được xây dựng dựa trên cho DI (Dependency Injection).
- Tag Helpers làm cho các Razor makup trở nên tự nhiên hơn với HTML.
- Có khả năng host trên IIS hoặc self-host.

5. Firebase

5.1 Firebase Cloud Messaging là gì?

Firebase là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính là giúp người dùng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

Cụ thể là những giao diện lập trình ứng dụng API đơn giản. Mục đích nhằm tăng số lượng người dùng và thu lại nhiều lợi nhuận hơn.

Đặc biệt, còn là dịch vụ đa năng và bảo mật cực tốt. Firebase hỗ trợ cả hai nền tảng Android và IOS. Không có gì khó hiểu khi nhiều lập trình viên chọn Firebase làm nền tảng đầu tiên để xây dựng ứng dụng cho hàng triệu người dùng trên toàn thế giới.

5.2 Firebase Cloud Messages

Firebase cloud messaging là một dịch vụ gửi thông báo, tin nhắn đa nền tảng được cung cấp bởi Google, cho phép bạn gửi tin nhắn, thông báo một cách đáng tin cậy và hoàn toàn miễn phí tới các thiết bị đã được đăng ký với FCM.

Nguyên tắc hoạt động: Các thiết bị client sẽ đăng ký device_token lên cho FCM. Các thông báo, tin nhắn được soạn và gửi từ một website, từ Notification composer của firebase cung cấp, FCM sẽ nhận những thông báo này và xử lý gửi về các thiết bị đã đăng ký với FCM từ trước. Khi các thiết bị có kết nối mạng thì thông báo sẽ được gửi về ứng dụng thành công.

6. Lập trình ESP8266

6.1. Internet of things (IoT)

Internet of Things (IoT) hay Mạng lưới vạn vật kết nối Internet Là một kịch bản của thế giới, khi mà mỗi đồ vật, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính. IoT đã phát triển từ sự hội tụ của công nghệ không dây, công nghệ vi cơ điện tử và Internet [1]. Nói đơn giản là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet và với thế giới bên ngoài để thực hiện một công việc nào đó.

6.2. ESP8266

ESP8266 là dòng chip tích hợp Wi-Fi 2.4Ghz có thể lập trình được, rẻ tiền được sản xuất bởi một công ty bán dẫn Trung Quốc: Espressif Systems.

Được phát hành đầu tiên vào tháng 8 năm 2014, đóng gói đưa ra thị trường dạng Mô đun ESP-01, được sản xuất bởi bên thứ 3: AI-Thinker. Có khả năng kết nối Internet qua mạng Wi-Fi một cách nhanh chóng và sử dụng rất ít linh kiện đi kèm.

Với giá cả có thể nói là rất rẻ so với tính năng và khả năng ESP8266 có thể làm được.

ESP8266 có một cộng đồng các nhà phát triển trên thế giới rất lớn, cung cấp nhiều Module lập trình mã mở giúp nhiều người có thể tiếp cận và xây dựng ứng dụng rất nhanh.

Hiện nay tất cả các dòng chip ESP8266 trên thị trường đều mang nhãn ESP8266EX, là phiên bản nâng cấp của ESP8266

7. Websocket

WebSoket là công nghệ hỗ trợ giao tiếp hai chiều giữa client và server bằng cách sử dụng một TCP socket để tạo một kết nối hiệu quả và ít tốn kém. Mặc dù được thiết kế để chuyên sử dụng cho các ứng dụng web, lập trình viên vẫn có thể đưa chúng vào bất kì loại ứng dụng nào.

- WebSockets mới xuất hiện trong HTML5, là một kỹ thuật Reverse Ajax. WebSockets cho phép các kênh giao tiếp song song hai chiều và hiện đã được hỗ trợ trong nhiều trình duyệt (Firefox, Google Chrome và Safari). Kết nối được mở thông qua một HTTP request (yêu cầu HTTP), được gọi là liên kết WebSockets với những header đặc biệt. Kết nối được duy trì để bạn có thể viết và nhận dữ liệu bằng JavaScript như khi bạn đang sử dụng một TCP socket đơn thuần.
- Dữ liệu truyền tải thông qua giao thức HTTP (thường dùng với kỹ thuật Ajax) chứa nhiều dữ liệu không cần thiết trong phần header. Một header request/response của HTTP có kích thước khoảng 871 byte, trong khi với WebSocket, kích thước này chỉ là 2 byte (sau khi đã kết nối).

Ưu điểm

- WebSockets cung cấp khả năng giao tiếp hai chiều mạnh mẽ, có độ trễ thấp và dễ xử lý lỗi. Không cần phải có nhiều kết nối như phương pháp Comet long-polling và cũng không có những nhược điểm như Comet streaming.
- API cũng rất dễ sử dụng trực tiếp mà không cần bất kỳ các tầng bổ sung nào, so với Comet, thường đòi hỏi một thư viện tốt để xử lý kết nối lại, thời gian chờ timeout, các Ajax request (yêu cầu Ajax), các tin báo nhận và các dạng truyền tải tùy chọn khác nhau (Ajax long polling và jsonp polling).

Nhược điểm

Những nhược điểm của WebSockets gồm có:

- Nó là một đặc tả mới của HTML5, nên nó vẫn chưa được tất cả các trình duyệt hỗ trợ.
- Không có phạm vi yêu cầu nào. Do WebSocket là một TCP socket chứ không phải là HTTP request, nên không dễ sử dụng các dịch vụ có phạm vi-yêu cầu,

như SessionInViewFilter của Hibernate. Hibernate là một framework kinh điển cung cấp một bộ lọc xung quanh một HTTP request. Khi bắt đầu một request, nó sẽ thiết lập một contest (chứa các transaction và liên kết JDBC) được ràng buộc với luồng request. Khi request đó kết thúc, bộ lọc hủy bỏ contest này.

8. SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS)) sử dụng câu lệnh SQL (Transact-SQL) để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS. SQL Server được phát triển và tiếp thị bởi Microsoft.



Hình 3:SQL Server

Tại sao lại là SQL Server ?

- SQL Server Database Engine: công cụ này kiểm soát việc lưu trữ, xử lý và bảo mật dữ liệu. Thành phần này bao gồm một công cụ quan hệ có chức năng xử lý các lệnh và truy vấn, một công cụ lưu trữ quản lý các tệp, bảng, trang, chỉ mục, bộ đệm dữ liệu và giao dịch cơ sở dữ liệu. Các nhiệm vụ, trigger, trình xem và các đối tượng dữ liệu lưu trữ khác cũng được Database Engine khởi tạo và xử lý.
- Lớp phía dưới Database Engine là Hệ điều hành SQL Server – viết tắt SQLOS. Hệ điều hành xử lý các chức năng ở cấp độ thấp hơn như quản lý bộ nhớ và I/O, lên lịch nhiệm vụ và khóa dữ liệu để tránh các xung đột xảy ra khi update. Một lớp giao diện mạng nằm trên lớp Database Engine và sử dụng một giao thức gọi là Tabular Data Stream của Microsoft để các yêu cầu và phản hồi

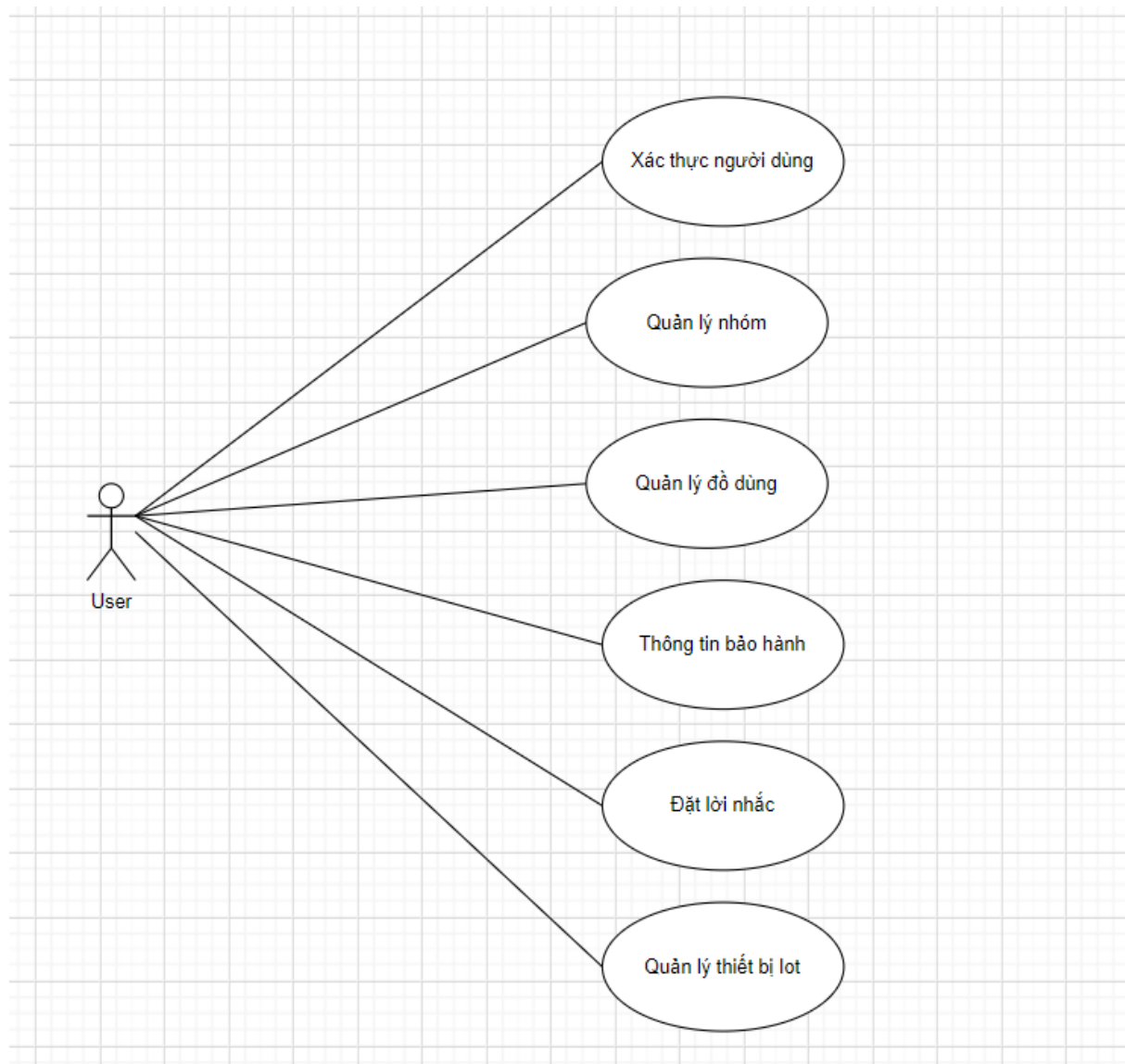
tương tác với máy chủ cơ sở dữ liệu thuận tiện hơn. Ở cấp độ user, SQL Server DBAs và developers viết các câu lệnh T-SQL để xây dựng và sửa đổi cấu trúc cơ sở dữ liệu, thao tác, thiết lập các bảo vệ, sao lưu cơ sở dữ liệu, cùng với nhiều nhiệm vụ khác.

- Công cụ và dịch vụ của SQL Server: Microsoft cung cấp cả công cụ và dịch vụ quản lý dữ liệu và kinh doanh (BI) cùng với SQL Server.
- Tính năng doanh nghiệp chuyên biệt: Microsoft hiện nay đã có thêm các tính năng quản lý dữ liệu đa dạng, tính năng dành cho doanh nghiệp (business intelligence – BI), và các công cụ phân tích SQL Server. Bên cạnh các dịch vụ Machine Learning mới được tích hợp lần đầu tiên trong phiên bản SQL Server 2016, các dịch vụ phân tích dữ liệu bao gồm SQL Server Analysis Services, công cụ phân tích xử lý dữ liệu sử dụng trong BI, các ứng dụng trực quan hóa dữ liệu và các dịch vụ SQL Server Reporting, hỗ trợ tạo và phân phối các báo cáo BI.
- Tính năng quản lý: Về phương diện quản trị, Microsoft SQL Server gồm có các dịch vụ tích hợp SQL Server, dịch vụ SQL Server Data Quality và dịch vụ SQL Server Master. Hai bộ công cụ dành riêng cho DBAs và developer: SQL Server Data Tools, sử dụng trong việc phát triển cơ sở dữ liệu và SQL Server Management Studio, nhằm mục đích triển khai, giám sát và quản lý cơ sở dữ liệu.
- SQL server có nhiều ấn bản, trong đó bản Express: SQL Server Express dễ sử dụng và quản trị cơ sở dữ liệu đơn giản. Được tích hợp với Microsoft Visual Studio, nên dễ dàng để phát triển các ứng dụng dữ liệu, an toàn trong lưu trữ, và nhanh chóng triển khai. SQL Server Express là phiên bản miễn phí, không giới hạn về số cơ sở dữ liệu hoặc người sử dụng, nhưng nó chỉ dùng cho 1 bộ vi xử lý với 1 GB bộ nhớ và 10 GB file cơ sở dữ liệu. SQL Server Express là lựa chọn tốt cho những người dùng chỉ cần một phiên bản SQL Server 2005 nhỏ gọn, dùng trên máy chủ có cấu hình thấp, những nhà phát triển ứng dụng không chuyên hay những người yêu thích xây dựng các ứng dụng nhỏ.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ

I. Phân tích:

1. Sơ đồ use case các chức năng của người dùng



Hình 1: Sơ đồ use case các chức năng của người dùng

| Mục | Tên chức năng | Mô tả |
|-----|-----------------------------|-------------------------------------------------------------------------------------------|
| 1 | Xác thực | Người dùng đăng nhập vào hệ thống với tài khoản đã đăng ký hoặc đăng ký một tài khoản mới |
| 2 | Nhóm chức năng quản lý nhóm | Quản lý thông tin nhóm, thành viên |
| 3 | Quản lý đồ dùng | Quản lý thông tin đồ dùng |
| 4 | Thông tin bảo hành | Quản lý các hóa đơn, thông tin sửa chữa, bảo hành của sản phẩm |

| | | |
|---|----------------------|-----------------------------------------------------|
| 5 | Đặt lời nhắc | Tạo lời nhắc, thông báo cho người dùng |
| 6 | Quản lý thiết bị IOT | Thao tác với các thiết bị hỗ trợ IOT của người dùng |

2. Phân tích các chức năng của hệ thống

2.2.1 Chức năng đăng nhập

2.2.1.1. Đặc tả chức năng

| | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-----------------------------------------------------------|
| Mô tả | | Chức năng này cho phép người dùng đăng nhập vào hệ thống. |
| Tác nhân | Chính | Người dùng |
| | Phụ | Hệ thống |
| Tiền điều kiện | | Người dùng có tài khoản đăng nhập vào hệ thống. |
| Hậu điều kiện | Thành công | Đăng nhập thành công, chuyển đến giao diện trang chủ. |
| | Lỗi | Đăng nhập không thành công |
| ĐẶC TẢ CHỨC NĂNG | | |
| Dòng sự kiện | | |
| <p><i>Dòng sự kiện chính:</i></p> <p>Chức năng này bắt đầu khi người dùng mở app:</p> <ol style="list-style-type: none"> 1. Người dùng mở app. 2. Hệ thống hiển thị form đăng nhập. 3. Khi người dùng nhập đầy đủ thông tin, hệ thống kiểm tra: <ul style="list-style-type: none"> -Dữ liệu nhập đã đầy đủ và đúng hay chưa. -Tài khoản có trong hệ thống và đúng mật khẩu không <p><i>Dòng sự kiện rẽ nhánh:</i></p> <p>Khi người dùng không đăng nhập thành công:</p> <ol style="list-style-type: none"> 1. Hệ thống sẽ báo đăng nhập không thành công do sai tên đăng nhập hoặc mật khẩu 2. Người dùng nhập lại thông tin yêu cầu 3. Quay lại các bước của luồng sự kiện chính hoặc hủy bỏ việc đăng nhập , kết thúc. | | |

2.2.2 Chức năng quản lý nhóm

2.2.2.1. Đặc tả chức năng

| | | |
|--------------|--------------|--------------------------------------------------|
| Mô tả | | Chức năng này cho phép người dùng thêm, xóa nhóm |
| | Chính | Người dùng |

| | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------------------------------------------------------|
| Tác nhân | Phụ | Hệ thống |
| Tiền điều kiện | | Người dùng được phân quyền để thêm xóa thành viên khỏi nhóm |
| Hậu điều kiện | Thành công | Thêm hoặc xóa thành công sẽ quay về giao diện danh sách nhóm |
| kiện | Lỗi | Thêm xóa không thành công |
| ĐẶC TẢ CHỨC NĂNG | | |
| Dòng sự kiện | | |
| <p><i>Dòng sự kiện chính:</i></p> <p>Chức năng này bắt đầu khi người dùng mở giao diện danh sách nhóm:</p> <ol style="list-style-type: none"> 1. Bấm vào nút thêm nhóm. 2. Hệ thống hiển thị màn hình giao diện thêm nhóm 3. Khi bấm đồng ý tạo nhóm hệ thống sẽ kiểm tra <ul style="list-style-type: none"> -Người dùng có nhập đúng không <p><i>Dòng sự kiện rẽ nhánh:</i></p> <p>Khi người dùng không thêm, xóa nhóm được:</p> <ol style="list-style-type: none"> 1. Hệ thống sẽ báo thêm, xóa không thành công vì chưa nhập đủ thông tin 2. Quay lại các bước của luồng sự kiện chính hoặc hủy bỏ việc thêm xóa, kết thúc. | | |

2.2.3 Chức năng quản lý đồ dùng

2.2.3.1. Đặc tả chức năng

| | | |
|-----------------------------------|-------------------|---------------------------------------------------------------|
| Mô tả | | Chức năng này cho phép người dùng quản lý đồ dùng cần quản lý |
| Tác nhân | Chính | Người dùng |
| | Phụ | Hệ thống |
| Tiền điều kiện | | Người dùng đã tham gia hoặc tạo một nhóm |
| Hậu điều kiện | Thành công | Trở về màn hình danh sách đồ dùng |
| kiện | Lỗi | Thông báo tạo thất bại |
| ĐẶC TẢ CHỨC NĂNG | | |
| Dòng sự kiện | | |
| <i>Dòng sự kiện chính:</i> | | |

Chức năng này bắt đầu khi người click vào nút thêm mới đồ dùng trong giao diện chi tiết của nhóm:

1. Người dùng nhập đầy đủ thông tin của đồ dùng
2. Khi người dùng nhập đầy đủ bấm đồng ý hệ thống lưu lại thông tin đồ dùng

Dòng sự kiện rẽ nhánh:

Khi người dùng không thống kê thành công thành công:

1. Hệ thống sẽ báo thất bại
2. Người dùng nhập lại thông tin yêu cầu
3. Quay lại các bước của luồng sự kiện chính hoặc hủy bỏ việc thống kê, kết thúc.

2.2.4 Chức năng quản lý thông tin bảo hành

2.2.4.1. Đặc tả chức năng thêm thông tin bảo hành

| | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|----------------------------------------------------------------------------------------------------|
| Mô tả | | Chức năng này cho phép người dùng lưu lại thông tin sửa chữa, bảo hành của đồ dùng |
| Tác nhân | Chính | Người dùng |
| | Phụ | Hệ thống |
| Tiền điều kiện | | Người dùng đã được thêm vào một nhóm hoặc đã tạo một nhóm Chọn đồ dùng mà đi bảo hành, sửa chữa |
| Hậu điều kiện | Thành công | Có thông báo là tạo thành công |
| | Lỗi | Có thông báo là tạo không thành công |
| ĐẶC TẢ CHỨC NĂNG | | |
| Dòng sự kiện | | |
| <i>Dòng sự kiện chính:</i> | | |
| Chức năng này bắt đầu khi người dùng click vào nút thêm thông tin bảo hành trong chi tiết đồ dùng: | | |
| <ol style="list-style-type: none"> 1. Người dùng click vào nút thêm bảo hành. 2. Hệ thống hiển thị form nhập thông tin bảo hành 3. Đặt lời nhắc cho lần bảo hành, sửa chữa sắp tới | | |
| <ol style="list-style-type: none"> 3. Khi người thực hiện các thao tác thêm lời nhắc hệ thống sẽ: <ul style="list-style-type: none"> -Kiểm tra xem các trường dữ liệu đã được nhập đúng hay chưa | | |
| <i>Dòng sự kiện rẽ nhánh:</i> | | |
| Khi người dùng không thêm, xóa hoặc đặt biệt danh được: | | |
| <ol style="list-style-type: none"> 1. Hệ thống sẽ báo các trường dữ liệu nhập chưa đúng | | |

2. Người dùng tiến hành nhập lại
3. Quay lại các bước của luồng sự kiện chính hoặc hủy bỏ việc thêm, kết thúc.

2.2.5 Chức năng quản lý thông báo, lịch hẹn

2.2.5.1. Đặc tả chức năng thêm thông báo, lịch hẹn

| | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---------------------------------------------------------------------------------------|
| Mô tả | | Chức năng này cho phép người dùng tạo lời nhắc cho cá nhân hoặc thành viên trong nhóm |
| Tác nhân | Chính | Người dùng |
| | Phụ | Hệ thống |
| Tiền điều kiện | | Không có |
| Hậu điều kiện | Thành công | Có thông báo về máy là tạo thành công thông báo |
| | Lỗi | Không thành công |
| ĐẶC TẢ CHỨC NĂNG | | |
| Dòng sự kiện | | |
| <p><i>Dòng sự kiện chính:</i></p> <p>Chức năng này bắt đầu khi người dùng click vào nút tạo lịch hẹn tại giao diện danh mục nhóm</p> <ol style="list-style-type: none"> 1. Người dùng click vào nút thêm mới. 2. Hệ thống hiển thị form tạo lời nhắc 3. Khi người thực hiện các thao tác thêm lời nhắc hệ thống sẽ: <ul style="list-style-type: none"> -Kiểm tra xem các trường dữ liệu đã được nhập đúng hay chưa <p><i>Dòng sự kiện rẽ nhánh:</i></p> <p>Khi người dùng không thêm được:</p> <ol style="list-style-type: none"> 1. Hệ thống sẽ báo các trường dữ liệu nhập chưa đúng 2. Người dùng tiến hành nhập lại 3. Quay lại các bước của luồng sự kiện chính hoặc hủy bỏ việc thêm, kết thúc. | | |

2.2.6 Chức năng quản lý thiết bị Iot

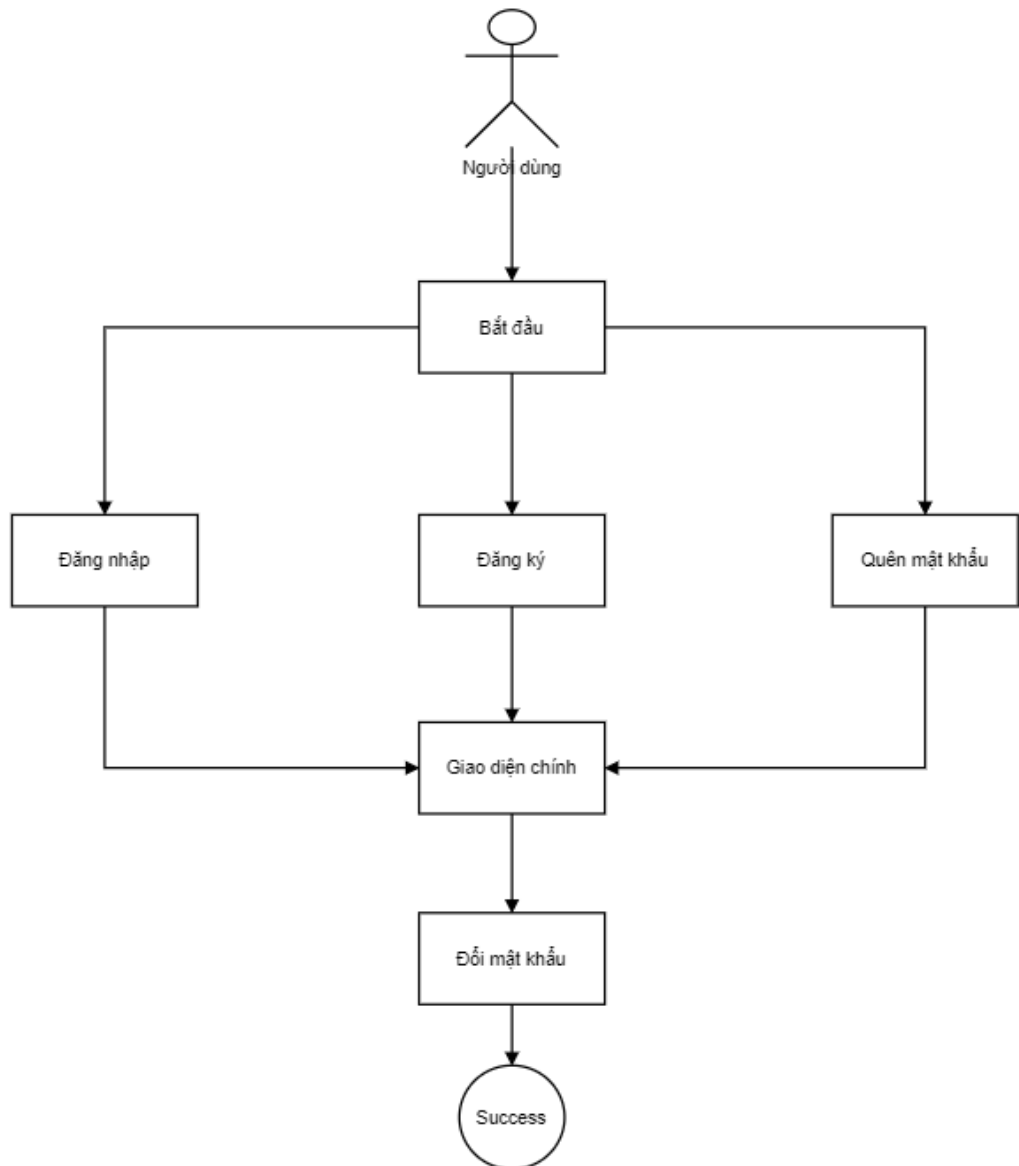
2.2.6.1. Đặc tả chức năng thêm mới Iot

| | | |
|--------------|--------------|---------------------------------------------------------|
| Mô tả | | Chức năng này cho phép người dùng thêm mới thiết bị Iot |
| | Chính | Người dùng |

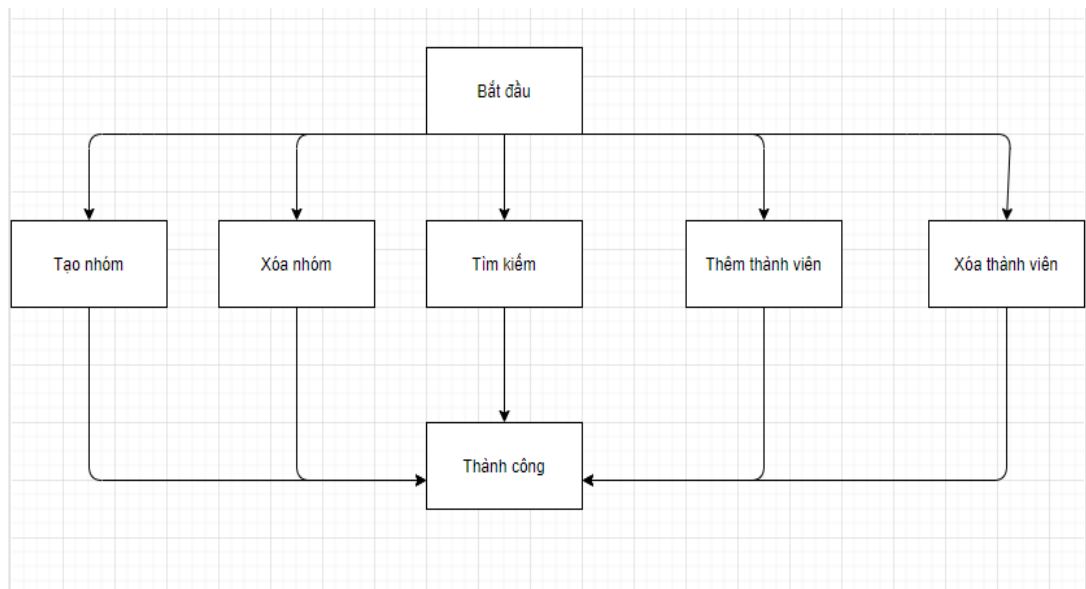
| | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------------------------|
| Tác nhân | Phụ | Hệ thống |
| Tiền điều kiện | | Không có |
| Hậu điều kiện | Thành công | Hiển thị ra danh mục thông báo |
| kiện | Lỗi | Không thành công |
| ĐẶC TẢ CHỨC NĂNG | | |
| Dòng sự kiện | | |
| <p><i>Dòng sự kiện chính:</i></p> <p>Chức năng này bắt đầu khi người dung click nút thêm mới thiết bị Iot.</p> <ol style="list-style-type: none"> 1. Nhập Ip của thiết bị 3. Khi người thực hiện các thao tác thêm hệ thống sẽ: <ul style="list-style-type: none"> -Kiểm tra xem các trường dữ liệu đã được nhập đúng hay chưa <p><i>Dòng sự kiện rẽ nhánh:</i></p> <p>Khi người dung không thêm được:</p> <ol style="list-style-type: none"> 1. Hệ thống sẽ báo các trường dữ liệu nhập chưa đúng 2. Người dung tiến hành nhập lại <p>Quay lại các bước của luồng sự kiện chính hoặc hủy bỏ việc thêm, kết thúc.</p> | | |

3. Biểu đồ hoạt động

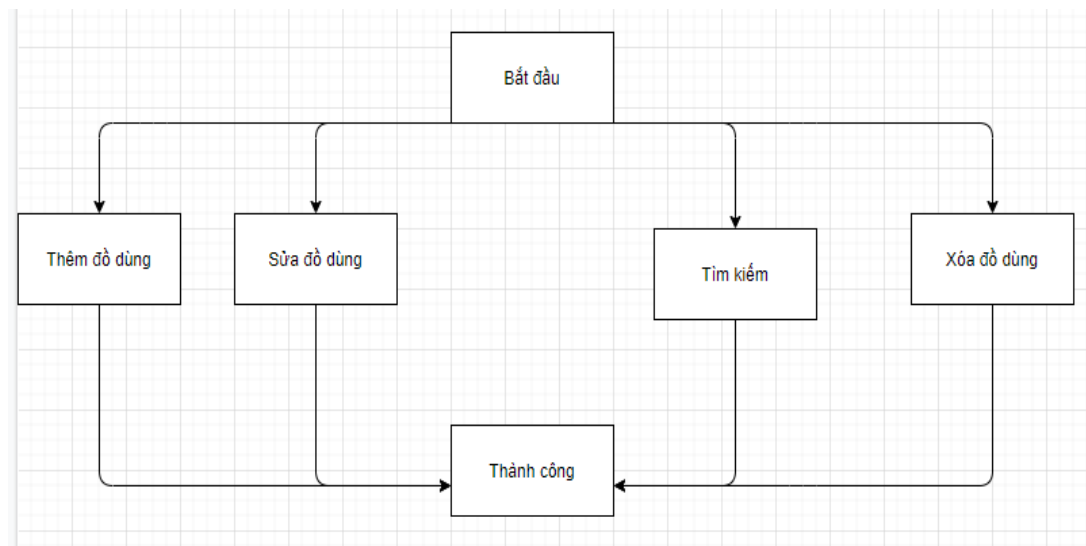
Dựa vào các quy trình đã phân tích ở trên, ta có thể xây dựng các biểu đồ hoạt động như sau:



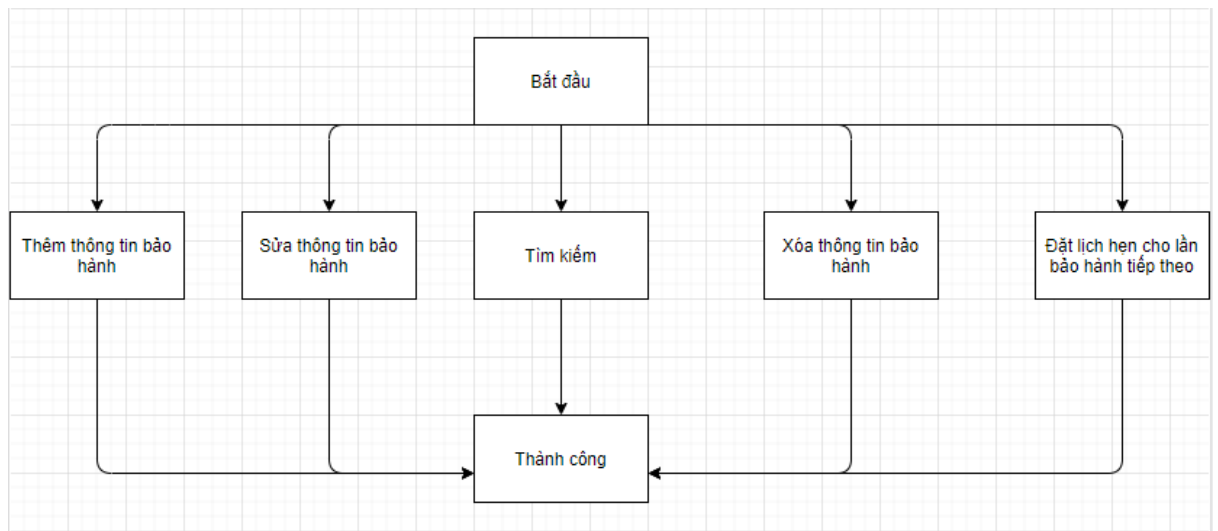
Hình 1: Biểu thị quản lý tài khoản



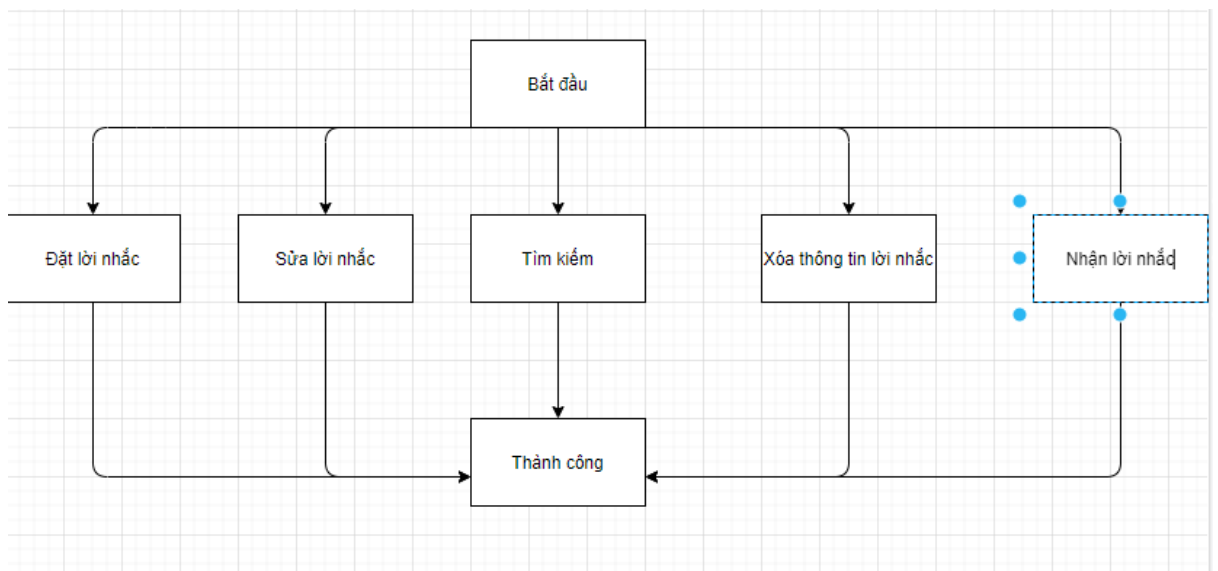
Hình 2: Biểu thị quản lý nhóm



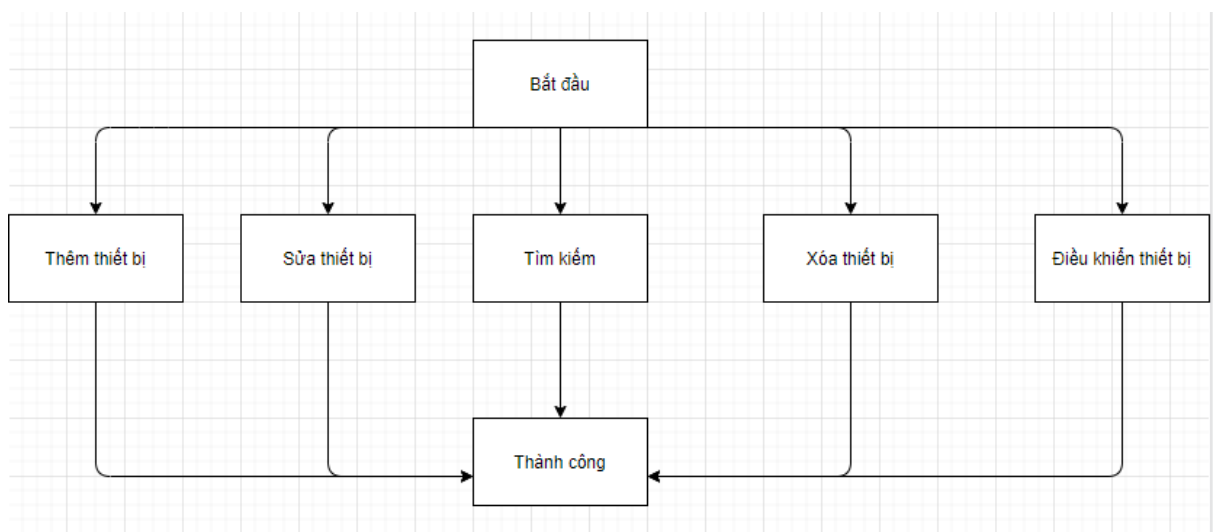
Hình 3: Biểu thị quản lý đồ dùng



Hình 4: Biểu thị quản lý thông tin bảo hành



Hình 5: Biểu thị quản lý lời nhắc



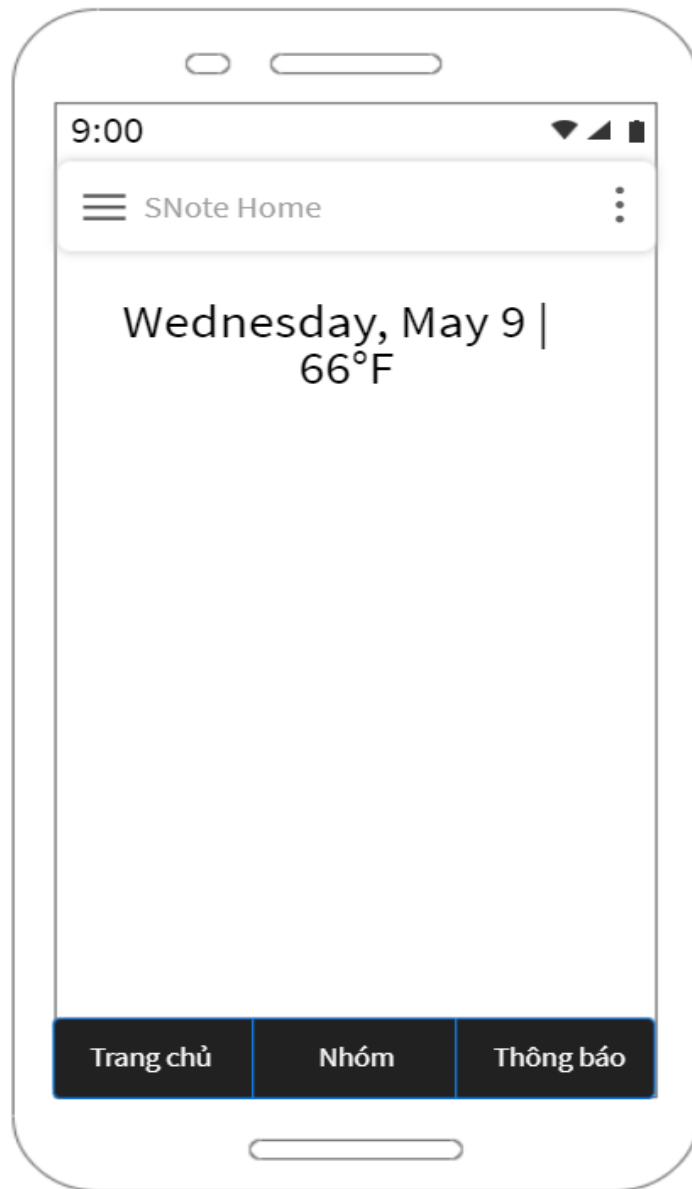
Hình 6: Biểu thị quản lý thiết bị Iot

II. Thiết kế.

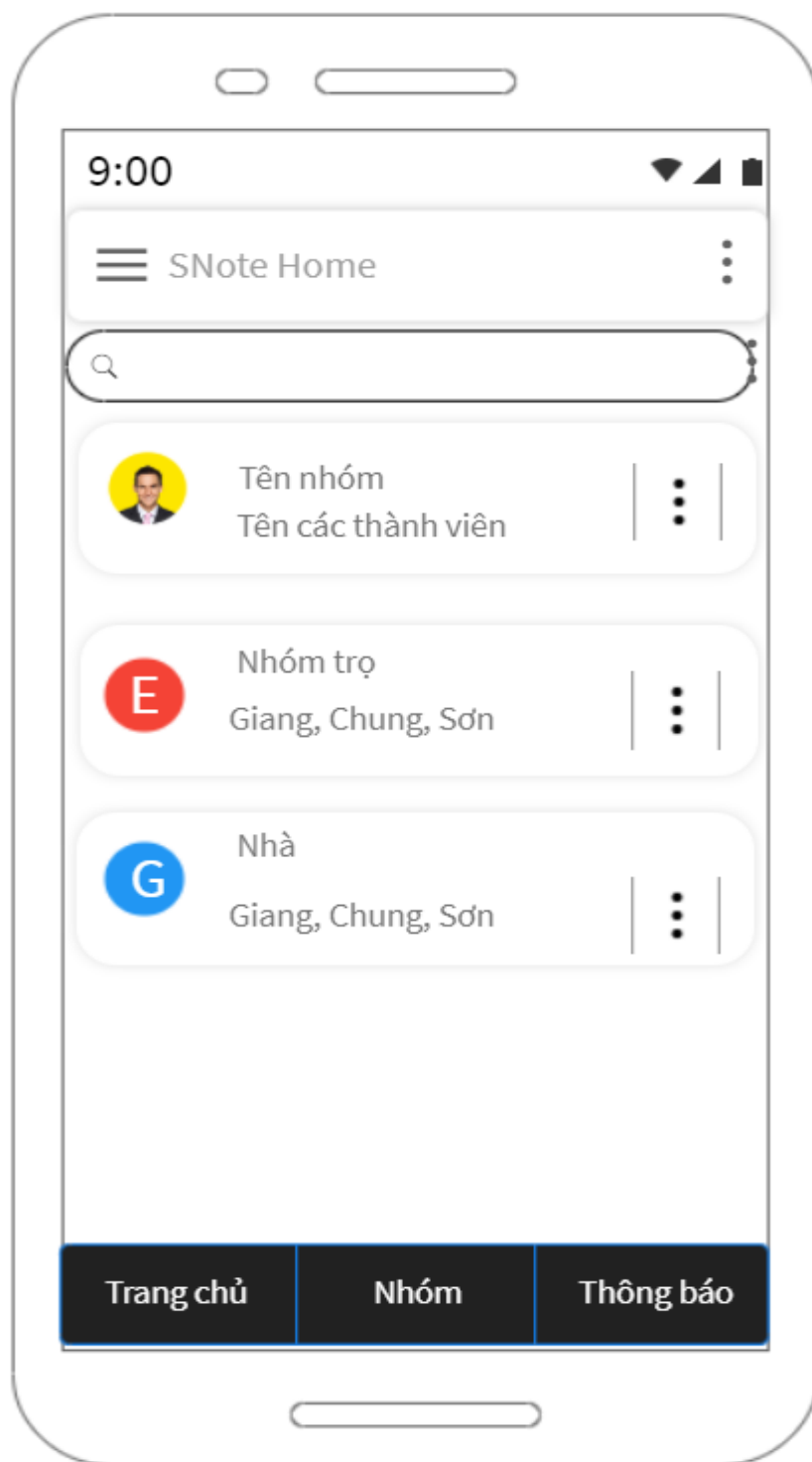
1. Thiết kế giao diện



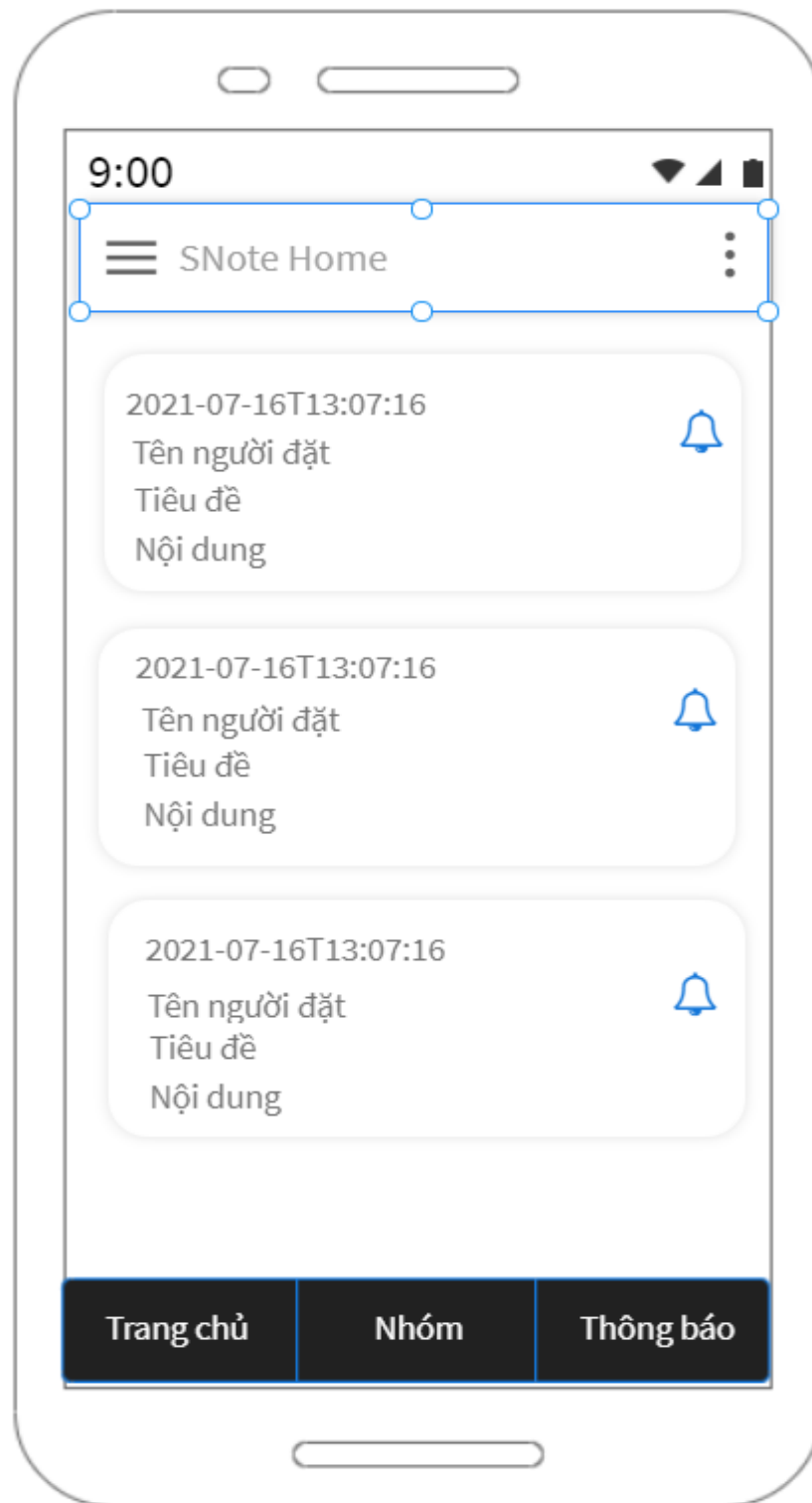
Hình 1:Giao diện đăng nhập.



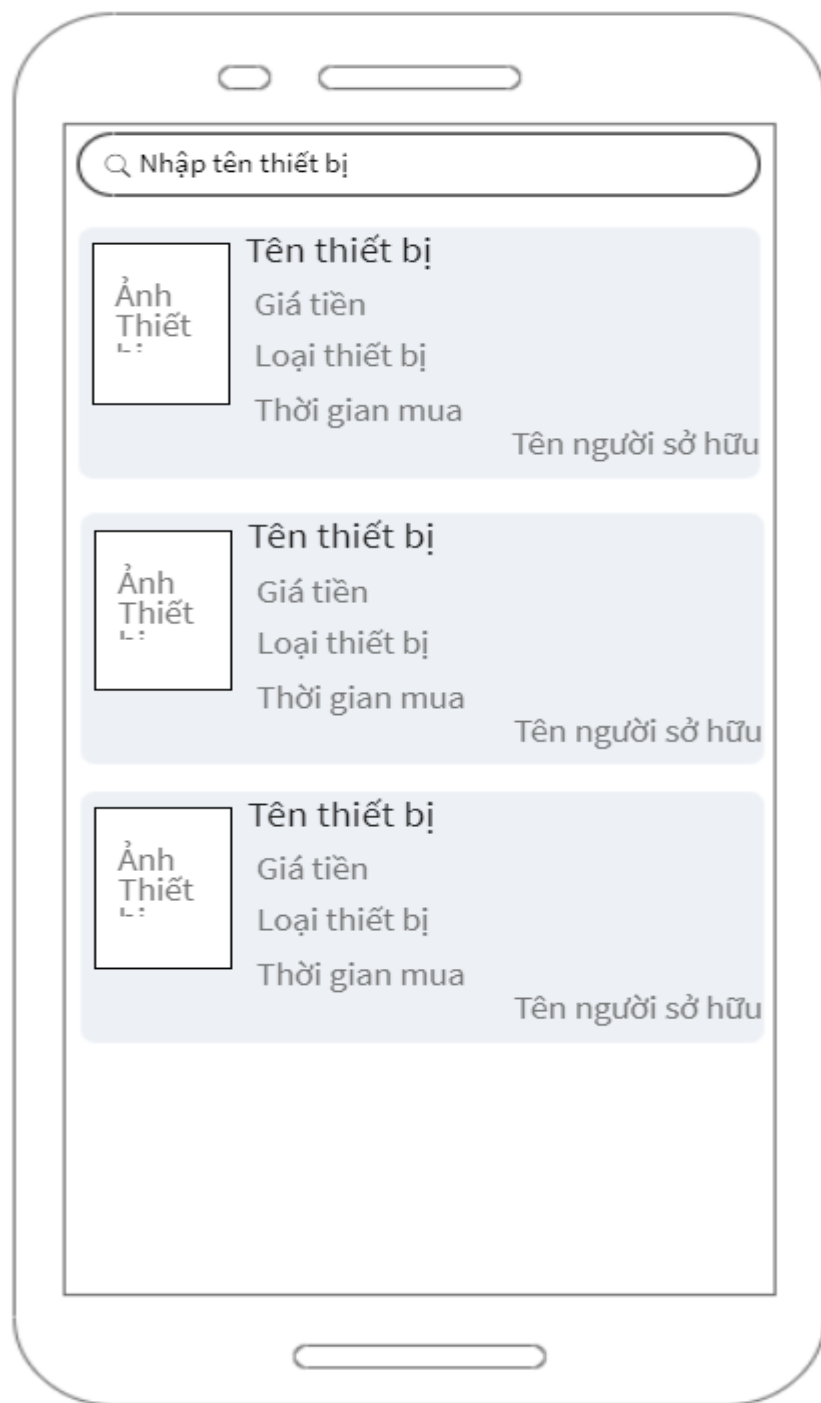
Hình 2: Giao diện trang chủ



Hình 3: Giao diện nhóm



Hình 4: Giao diện thông báo



Hình 5: Giao diện các thiết bị

9:00

← Thông tin chi tiết sản phẩm Sửa

Ảnh

Ảnh

Chi tiết sản phẩm

Thông số sản phẩm

Tên sản phẩm

Giá sản phẩm

Loại

Ngày mua sản phẩm

Ghi chú

Tên người sở hữu

Thông tin liên hệ

SốĐT người bán

Email người bán

Địa chỉ người bán

Thông tin bảo hành (Thêm mới)

Hình 6.1: Giao diện thông tin chi tiết thiết bị

9:00

← Thông tin chi tiết sản phẩm Sửa

Chi tiết sản phẩm

Thông số sản phẩm

Tên sản phẩm

Giá sản phẩm Tên người sở hữu

Loại

Ngày mua sản phẩm

Ghi chú

Thông tin liên hệ

SĐT người bán

Email người bán

Địa chỉ người bán

Thông tin bảo hành (Thêm mới)

| Ngày giờ | Nội dung | Giá tiền | Địa chỉ |
|----------|----------|----------|---------|
| | | | |

Hình 6.2: Giao diện thông tin chi tiết thiết bị

The image shows a mobile application interface for adding a new product. At the top, the status bar displays the time 9:00 and signal icons. Below the status bar, there is a navigation bar with a back arrow and the text "Thêm mới sản phẩm" (Add new product), and a "Thêm mới" (Add new) button. The main form consists of several input fields: "Tên sản phẩm" (Product name), "Giá sản phẩm" (Product price), "Loại" (Type), "Thời gian mua" (Purchase time), "Số dt nơi mua" (Phone number of the place of purchase), "Email nơi mua" (Email of the place of purchase), "Địa chỉ nơi mua" (Address of the place of purchase), and "Ghi chú" (Remarks). At the bottom of the form, there is a "Thêm ảnh" (Add image) button.

9:00

← Thêm mới sản phẩm [Thêm mới](#)

Tên sản phẩm

Giá sản phẩm

Loại

Thời gian mua

Số dt nơi mua

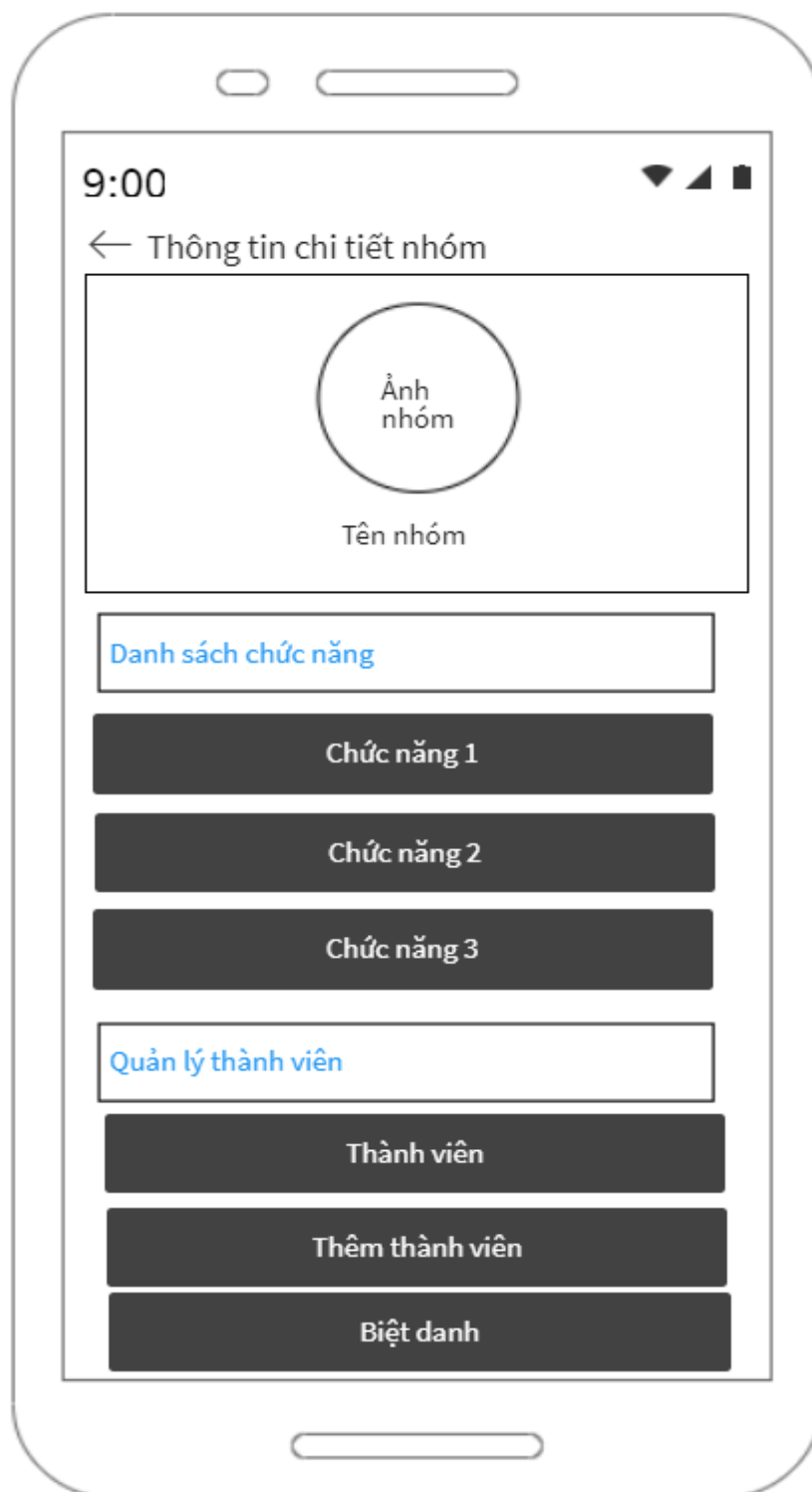
Email nơi mua

Địa chỉ nơi mua

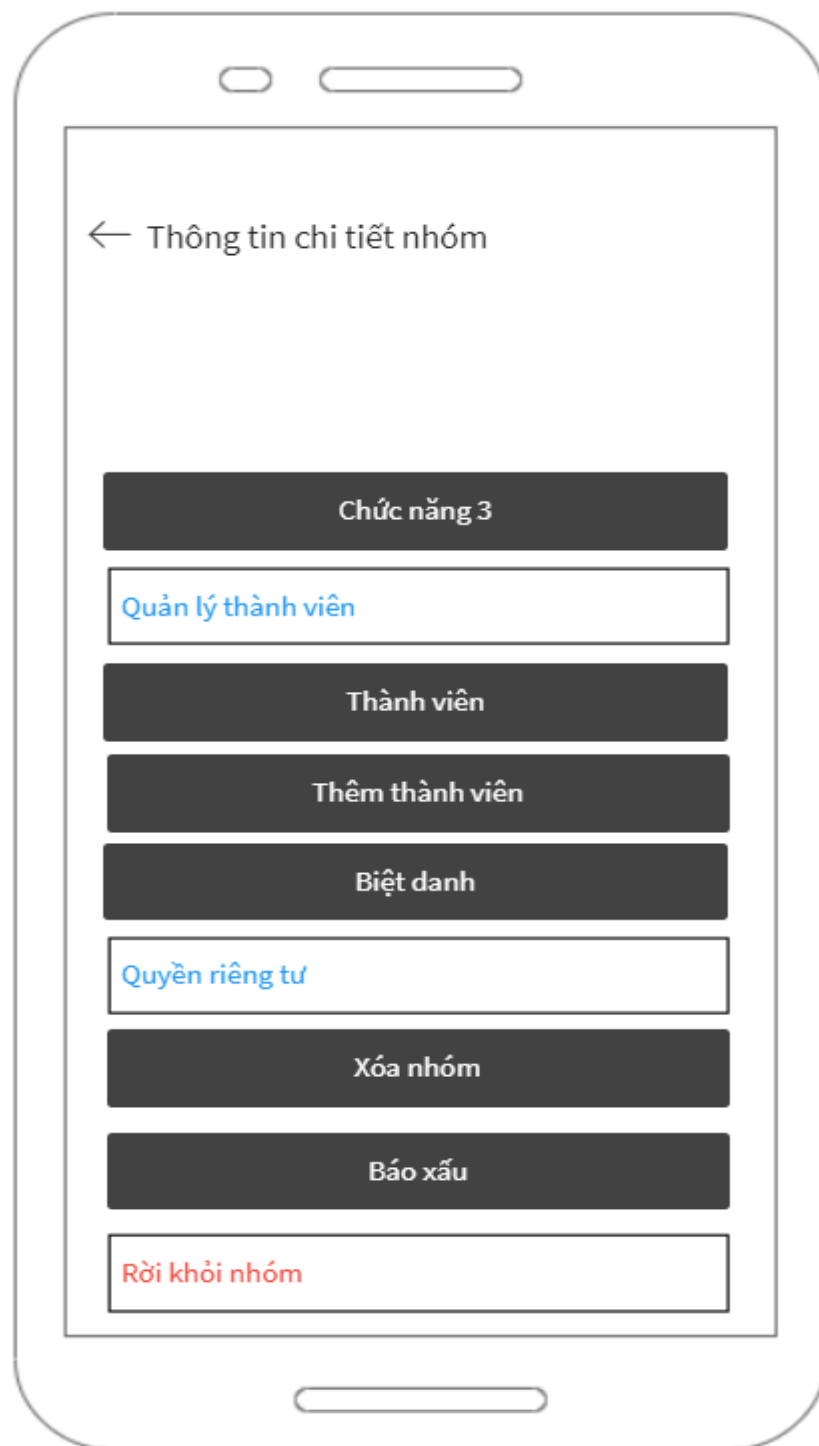
Ghi chú

[Thêm ảnh](#)

Hình 7: Giao diện thêm mới thiết bị



Hình 8: Giao diện chi tiết nhóm



Hình 8.2 Giao diện chi tiết nhóm


2. Đặc tả giao diện.

- **Giao diện đăng nhập**

| Đặc tả | Mô tả |
|------------------|-------------------------------------------------------------------|
| Tên | Giao diện đăng nhập |
| Mô tả sơ lược | Giao diện hiển thị các chức năng như đăng nhập, đăng ký tài khoản |
| Thao tác | Người dùng nhập tài khoản mật khẩu để đăng nhập vào hệ thống |
| Độ phức tạp | Bình Thường |
| Trường nhập liệu | Tài khoản, mật khẩu |
| Hiển thị | Hiển thị màn hình đăng nhập |

Bảng 1. Giao diện đăng nhập

- **Giao diện trang chủ**

| Đặc tả | Mô tả |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tên | Giao diện trang chủ |
| Mô tả sơ lược | Giao diện hiển thị Tab menu có thể chuyển sang giao diện nhóm hoặc giao diện thông báo, tại giao diện trang chủ ở chính giữa sẽ hiện số liệu về chất lượng không khí trong nhà của bạn, khi click vào nút  sẽ hiển thị ra cửa sổ quản lý tài khoản. |
| Thao tác | Người dùng click |
| Độ phức tạp | Bình Thường |
| Trường nhập liệu | Không |
| Hiển thị | Hiển thị màn hình trang chủ |

Bảng 2. Giao diện trang chủ

- **Giao diện nhóm**

| Đặc tả | Mô tả |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tên | Giao diện nhóm |
| Mô tả sơ lược | Giao diện hiển thị các nhóm người dùng đã tham gia hoặc đã tạo, tại đây người dùng có thể tìm kiếm thông qua thanh search, click vào ‘dòng’ của từng nhóm để xem những thiết bị của mình trong nhóm |
| Thao tác | Người dùng Click |
| Độ phức tạp | Bình Thường |

| | |
|------------------|------------------------|
| Trường nhập liệu | Tìm kiếm |
| Hiển thị | Hiển thị màn hình nhóm |

Bảng 3. Giao diện nhóm

- **Giao diện thông báo**

| Đặc tả | Mô tả |
|------------------|--------------------------------------------------------------------|
| Tên | Giao diện thông báo |
| Mô tả sơ lược | Giao diện hiển thị các thông báo mà người dùng đã và sẽ nhận được, |
| Thao tác | Người dùng Click |
| Độ phức tạp | Bình Thường |
| Trường nhập liệu | Không có |
| Hiển thị | Hiển thị màn hình thông báo |

Bảng 4. Giao diện thông báo

- **Giao diện các thiết bị**

| Đặc tả | Mô tả |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tên | Giao diện các thiết bị |
| Mô tả sơ lược | Giao diện hiển thị các thiết bị của người dùng theo nhóm, người dùng có thể tìm kiếm theo tên thiết bị, khi ấn vào thiết bị sẽ hiển thị ra giao diện thông tin chi tiết của thiết bị |
| Thao tác | Người dùng Click |
| Độ phức tạp | Bình Thường |
| Trường nhập liệu | Không có |
| Hiển thị | Hiển thị giao diện các thiết bị |

Bảng 5. Giao diện các thiết bị

- **Giao diện thông tin chi tiết thiết bị**

| Đặc tả | Mô tả |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tên | Giao diện thông tin chi tiết thiết bị |
| Mô tả sơ lược | Giao diện hiển thị các thông tin chi tiết về thiết bị như tên, ảnh, ngày mua, lịch sử bảo hành, thông tin người bán, khi bấm vào nút sửa sẽ hiển thị ra 1 activity mới cho phép người dùng sửa thông tin của thiết bị đó, khi bấm vào chữ thêm mới sẽ hiển thị ra 1 activity mới để thêm mới hóa đơn, khi ấn vào dòng hóa đơn bên dưới sẽ hiển thị ra dialog sửa hoặc xóa, |

| | |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | khi nhấn vào mỗi ảnh trong list ảnh sẽ hiện ra cửa sổ chọn ảnh trong máy để thay thế và khi nhấn vào icon thùng rác bên cạnh sẽ hiện ra dialog xác nhận xóa ảnh. |
| Thao tác | Người dùng Click |
| Độ phức tạp | Bình Thường |
| Trường nhập liệu | Không có |
| Hiển thị | Hiển thị giao diện thông tin chi tiết của sản phẩm muốn xem |

Bảng 6. Giao diện chi tiết thiết bị

- **Giao diện thêm mới thiết bị**

| Đặc tả | Mô tả |
|------------------|------------------------------------------------------------------------------------|
| Tên | Giao diện thêm mới thiết bị |
| Mô tả sơ lược | Giao diện dùng để nhập liệu những thông tin cần thiết của thiết bị |
| Thao tác | Người dùng nhập những thông tin của thiết bị |
| Độ phức tạp | Bình Thường |
| Trường nhập liệu | Tên thiết bị, giá thiết bị, loại, thời gian mua, thông tin của người bán, ảnh,.... |
| Hiển thị | Hiển thị giao diện thêm mới thiết bị |

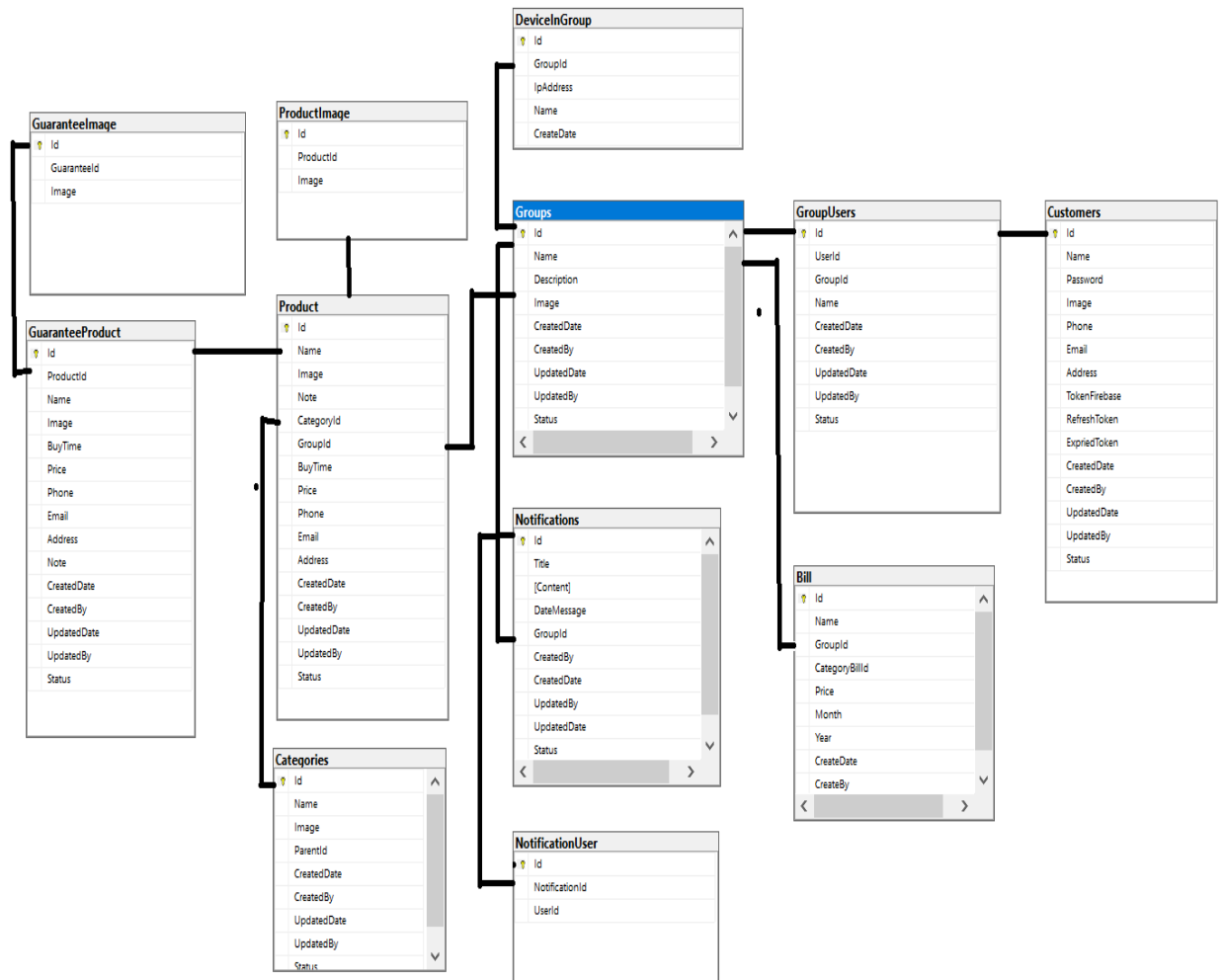
Bảng 7. Giao diện thêm mới thiết bị

- **Giao diện chi tiết nhóm**

| Đặc tả | Mô tả |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Tên | Giao diện chi tiết nhóm |
| Mô tả sơ lược | Giao diện hiển thị tên, ảnh nhóm, các chức năng của nhóm ví dụ như thống kê tiền theo tháng, năm, tạo lịch hẹn, quản lý thành viên, xóa nhóm ,... |
| Thao tác | Người dùng click |
| Độ phức tạp | Bình Thường |
| Trường nhập liệu | Không có |
| Hiển thị | Hiển thị giao diện chi tiết nhóm |

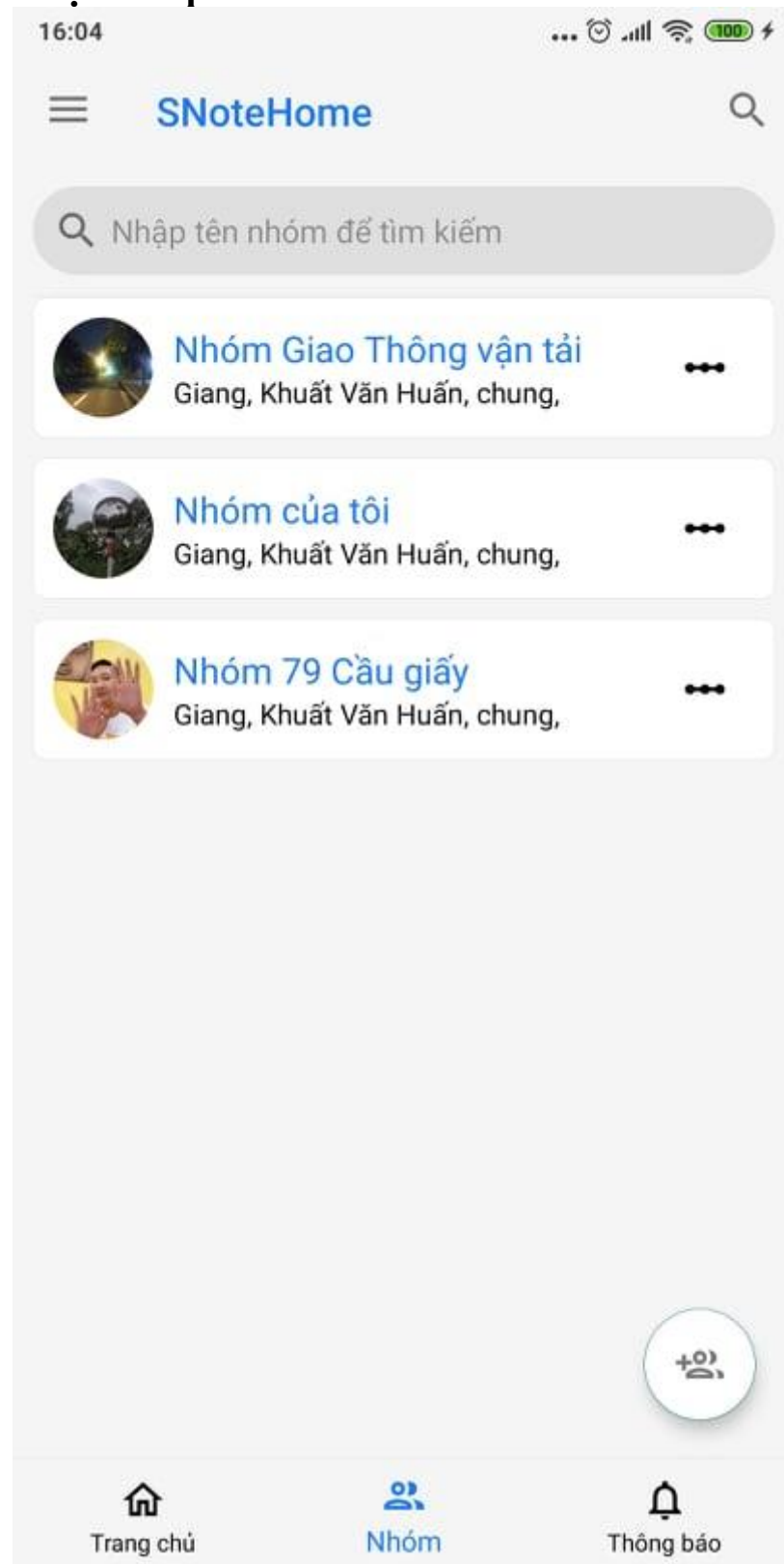
Bảng 8. Giao diện chi tiết nhóm

3. Thiết kế cơ sở dữ liệu

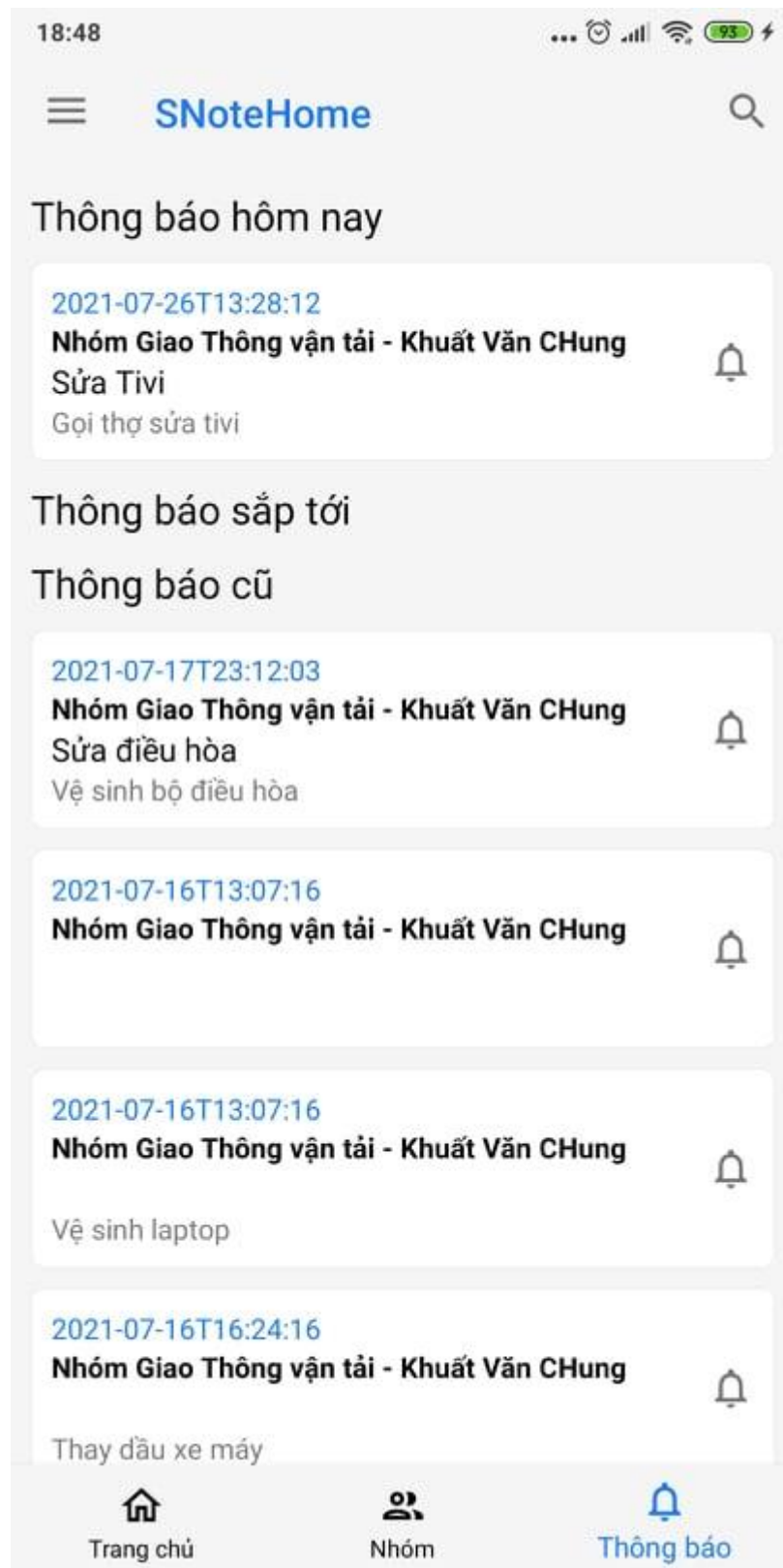


CHƯƠNG 3: KẾT QUẢ NGHIÊN CỨU ĐẠT ĐƯỢC

1. Một số giao diện kết quả



Hình 1: Giao diện nhóm



Hình 2: Giao diện thông báo

10:45

65

<

Thêm mới sản phẩm

Thêm mới

Tên sản phẩm

Giá sản phẩm

Điện tử

Thời gian mua

Số điện thoại nơi mua

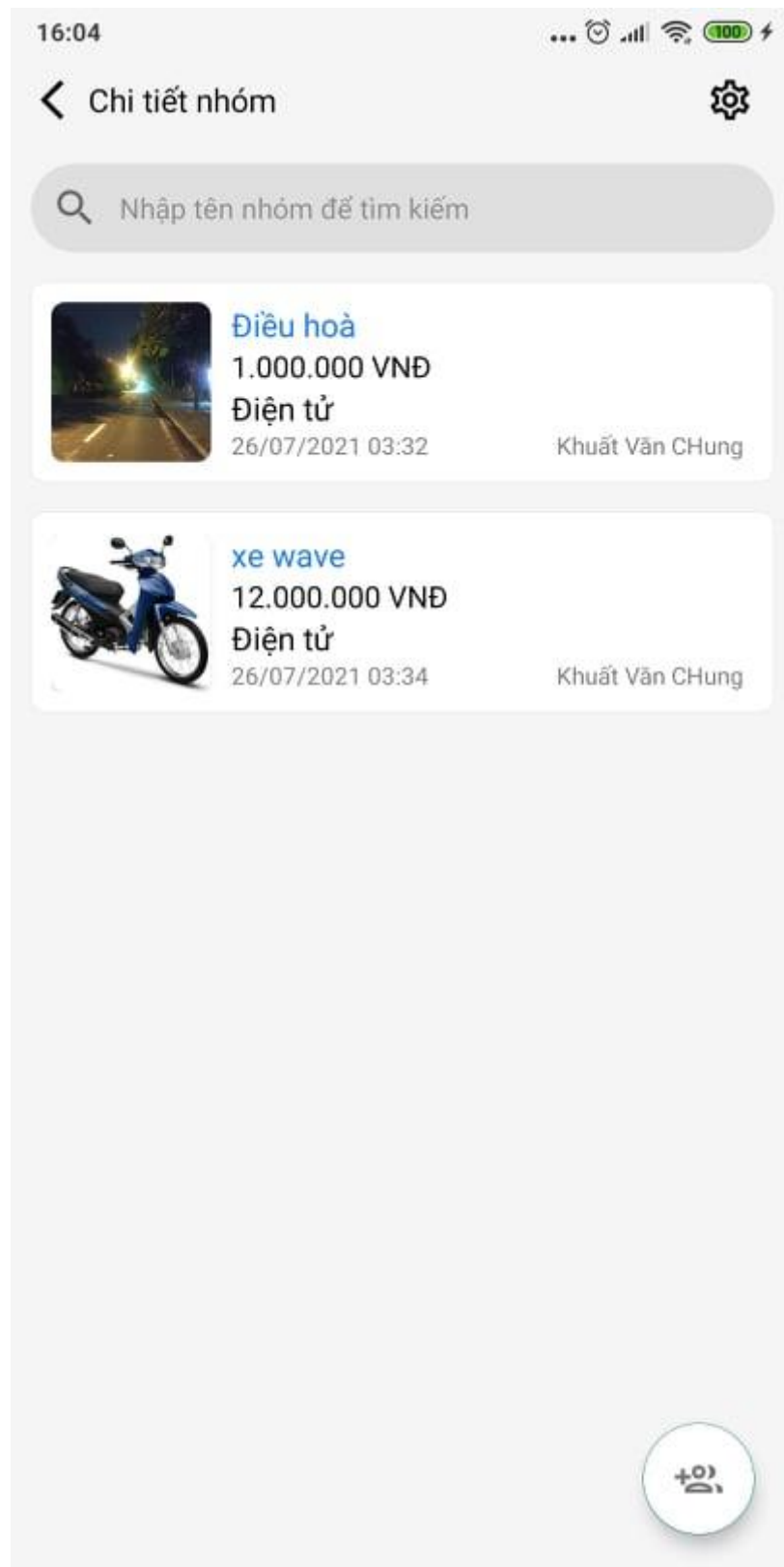
Email nơi mua

Địa chỉ nơi mua

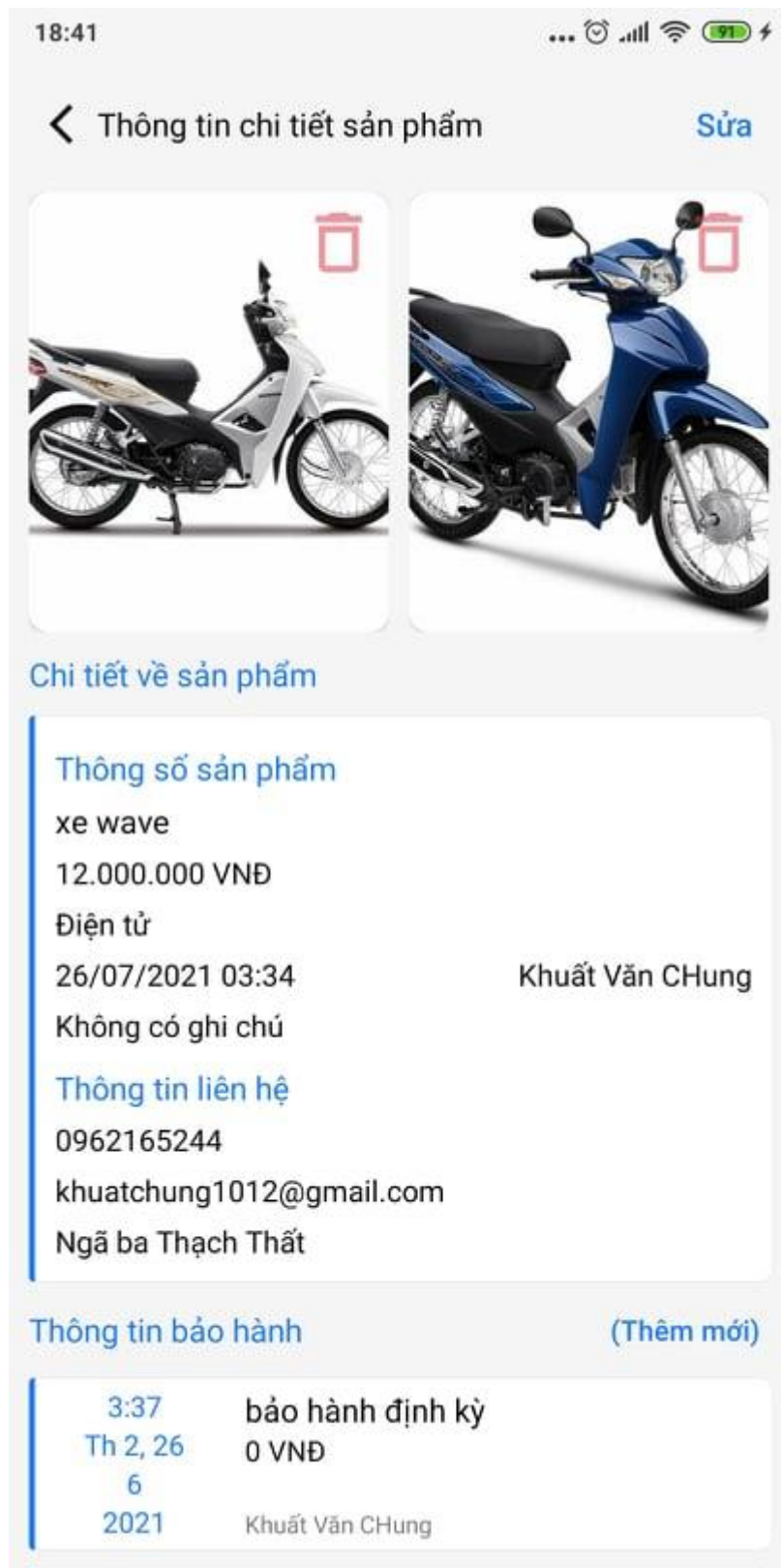
Ghi chú

Thêm ảnh

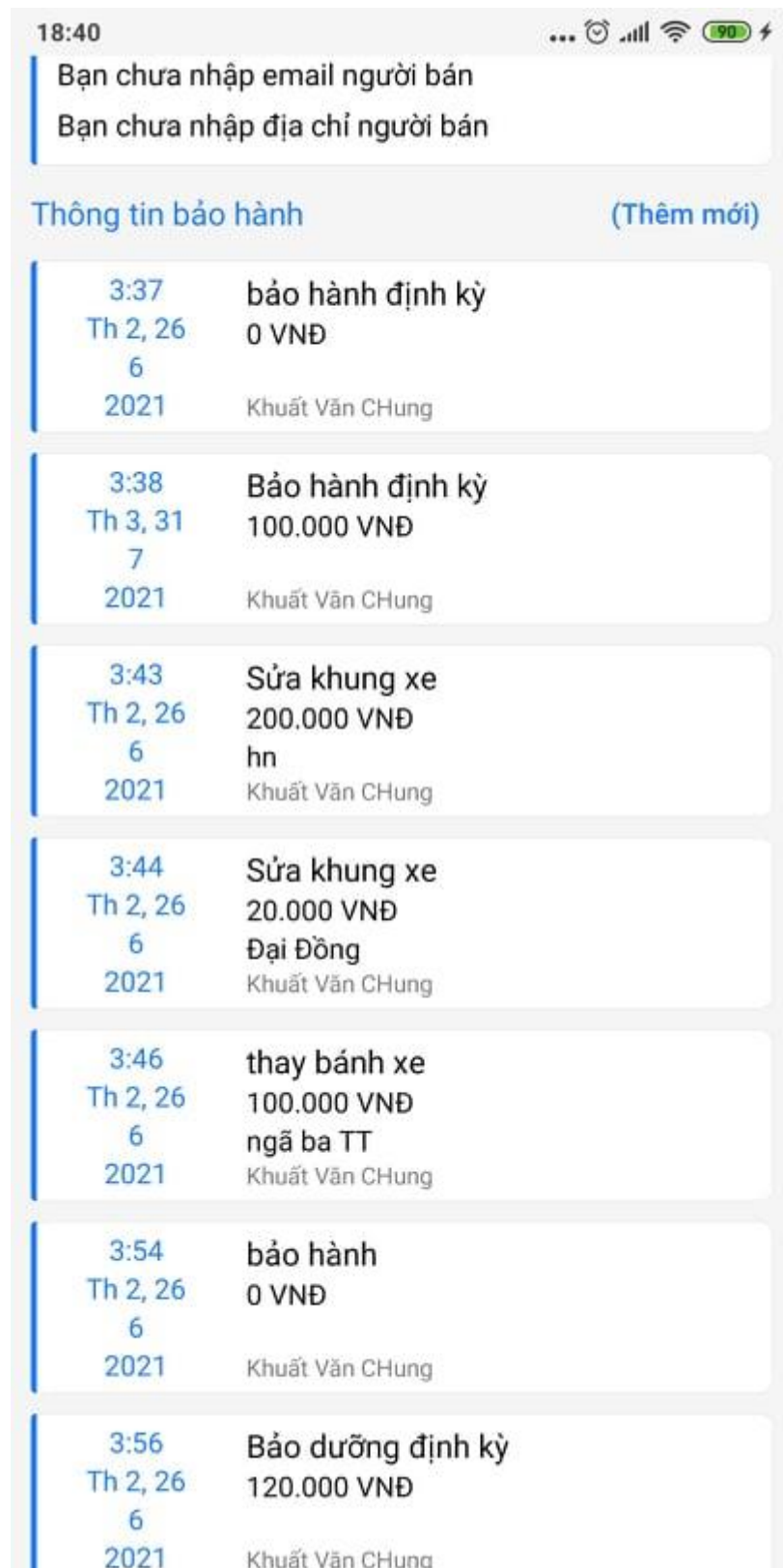
Hình 3: Giao diện thêm mới sản phẩm



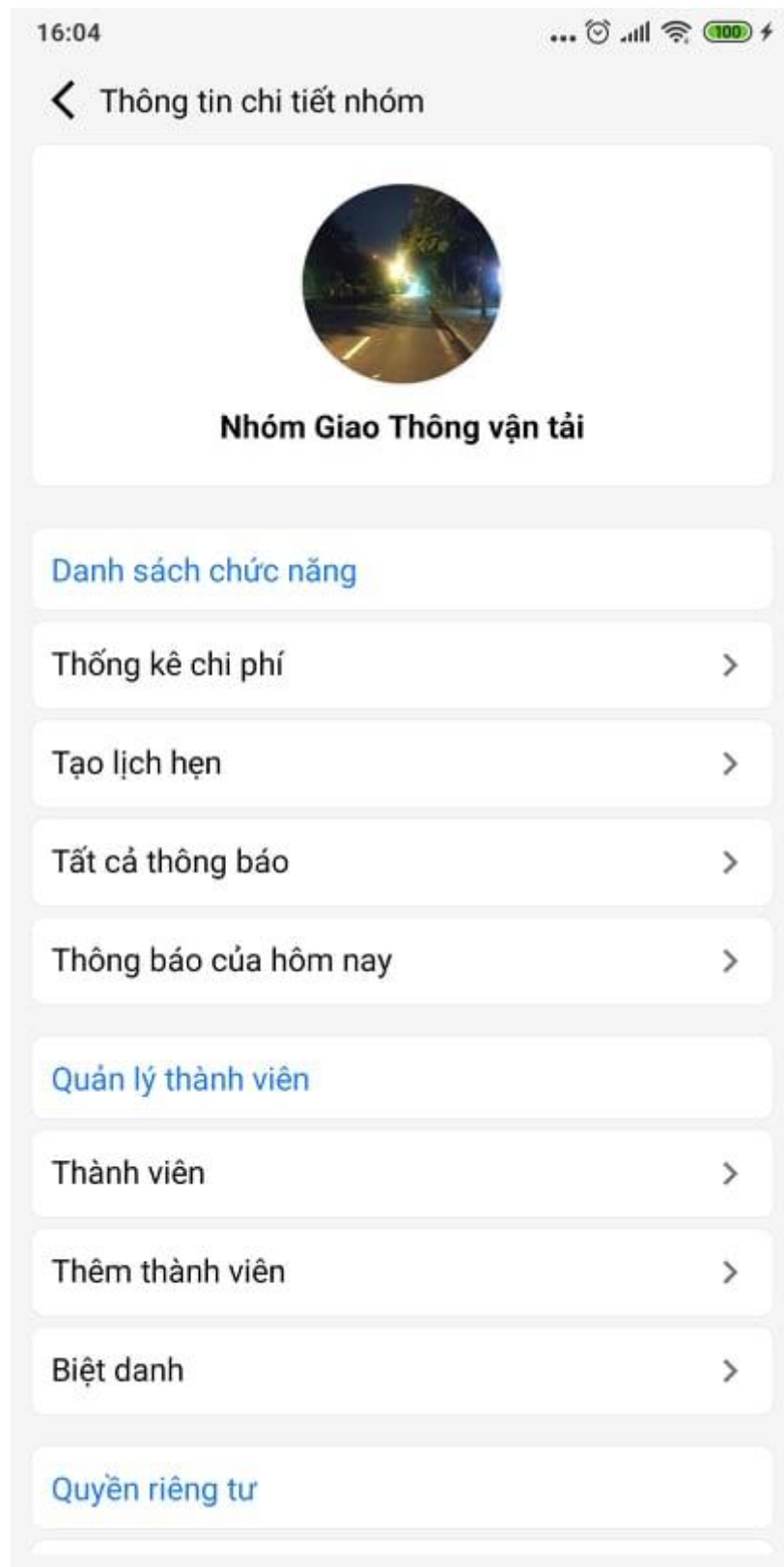
Hình 4: Giao diện danh sách sản phẩm



Hình 5: Giao diện chi tiết sản phẩm



Hình 6: Giao diện chi tiết sản phẩm



Hình 7: Giao diện chi tiết nhóm

2. Đánh giá kết quả

2.1 Ưu điểm

- Ý tưởng ứng dụng mới.
- Phù hợp với thực tiễn.
- Ứng dụng sử dụng tốt.
- Giao diện thân thiện với người dùng.

2.2 Nhược điểm

- Ứng dụng chưa hoàn thiện tốt đôi khi còn gặp một số lỗi.
- Mới chỉ chạy trên nền tảng Android.

2.3 Hướng phát triển

- Thêm một số chức năng như scan ảnh để người dùng lưu thông tin hóa đơn, thiết bị nhanh hơn.
- Tích hợp thêm nhiều ứng dụng IOT.
- Quản lý được nhiều loại hóa đơn khác nhau.
- Quản lý cả đồ mua đồ bán.
- Phát triển ứng dụng trên nền tảng web và IOS.

CHƯƠNG 4: KẾT LUẬN

Trong quá trình thực hiện đề tài, do phải tiếp cận với nhiều lĩnh vực mới nên chúng em đã gặp phải không ít khó khăn, tuy nhiên với sự giúp đỡ nhiệt tình của thầy cô và những người đi trước cùng với nỗ lực của bản thân của mỗi thành viên, đề tài đã được hoàn thành đúng hạn và thu được kết quả khả quan. Bên cạnh đó, do thời gian thực hiện có hạn nên một vài phần của đề tài có kết quả chưa được hoàn toàn như mong đợi. Trong thời gian tới, đề tài sẽ tiếp tục được chỉnh sửa, mở rộng để hoàn thiện hơn.

Việc tạo ra một app để mọi người có thể thống kê chi tiêu và theo dõi các đồ dùng đã tạo nên động lực rất lớn lao cho chúng em. Nó tạo ra thói quen cho người dùng về cách chi tiêu và quản lý đồ dùng một cách hợp lý và hiệu quả

Qua quá trình thực hiện đề tài, chúng em đã thu nhận được những kinh nghiệm quý báu về phát triển ứng dụng trên nền tảng, kinh nghiệm làm việc nhóm và phân công công việc hiệu quả. Những kiến thức này sẽ là những hành trang quý báu trong công việc tương lai của bản thân mỗi thành viên.

Hà Nội, ngày ... tháng ... năm 2020

Nhóm sinh viên thực hiện đề tài

TÀI LIỆU THAM KHẢO

1. <https://www.google.com.vn/>
2. <https://stackoverflow.com/>
3. <https://firebase.google.com/docs/android/setup>
4. <https://firebase.google.com/docs/web/setup>
5. https://viblo.asia/p/gioi-thieu-ve-java-63vKjD8Vl2R?fbclid=IwAR32PjkwuMFWuYDKG9WheRIYxpk7EFeeriAJjpEDqSsh2Hy9u4p4v5bqp_4
6. <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet-core>
7. <https://docs.microsoft.com/en-us/dotnet/csharp/>
8. <https://firebase.google.com/>
9. <https://viblo.asia/p/websocket-la-gi-Ljy5VxkbZra>