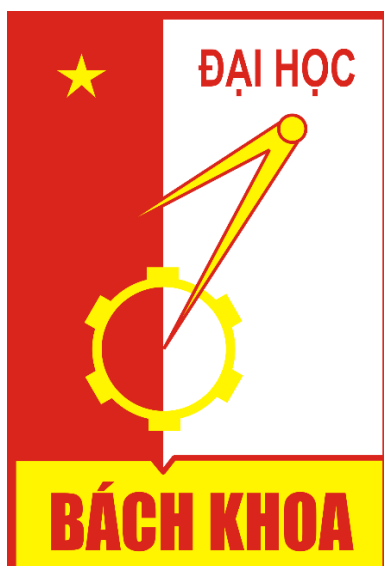


TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO MÔN HỌC: PROJECT 2

XÂY DỰNG PHẦN MỀM QUẢN LÝ MÁY LỌC NƯỚC.

Sinh viên: **ĐỒNG QUANG LINH**

Lớp : **CNTT1.01 -K61**

GVHD : **PGS-TS TRỊNH VĂN LOAN**

HÀ NỘI 06-2019

1. Thông tin về sinh viên.

Họ và tên sinh viên: Đồng Quang Linh.

Điện thoại liên lạc: 0976268575.

Email: linh.dq162385@sis.hust.edu.vn

Lớp: CNTT1.01-K61

Hệ đào tạo: đại học

Thời gian làm đồ án: Từ ngày 25/01/2019 đến ngày 30/05/2019

2. Mục đích nội dung của đồ án.

- Xây dựng phần mềm quản lý máy lọc nước nhằm thông báo thời gian thay bộ lọc đúng hạn cho các sản phẩm.

3. Các nhiệm vụ cụ thể của ĐATN.

- Tìm hiểu về các loại máy lọc nước trên thị trường.
- Tiến hành xây dựng app ứng dụng cho điện thoại Android giúp hiển thị các thông tin cần thiết về sản phẩm và thời gian thay bộ lọc đúng hạn.

4. Lời cam đoan của sinh viên:

Tôi -Đồng Quang Linh- cam kết đồ án là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của giáo viên. Các kết quả nêu trong đồ án là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Link source code : <https://github.com/LinhIThust/Project2>

<https://github.com/LinhIThust/QuanLiMayLocNuoc-pr2->

	<i>Hà Nội, ngày 30 tháng 05 năm 2019</i> Tác giả đồ án Linh Đồng Quang Linh
--	--

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của đồ án.

	<i>Hà Nội, ngày ... tháng ... năm</i> Giáo viên hướng dẫn
--	--

Mục Lục

PHẦN I: ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP	4
1.1. Tổng quan.	4
1.1.1. Các vấn đề.....	4
1.1.2. Mục tiêu cần đạt được.....	4
1.1.3. Lựa chọn và định hướng thiết kế	5
1.2. Các công nghệ kỹ thuật liên quan.....	5
1.3. Các framework/thư viện lập trình	6
PHẦN II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	8
2.1. Các thành phần chức năng.	8
2.2. Thiết kế giao diện/layout	11
PHẦN III: CÀI ĐẶT VÀ TRIỂN KHAI GIẢI PHÁP	16
3.1. Mô hình dữ liệu.....	16
3.2. Đăng kí, Đăng nhập, Đăng xuất.....	16
3.3. Hiện thị các sản phẩm đã mua	19
3.4. Hiện thị thời gian cần thay các bộ lọc.....	20
3.5. Thông báo thời gian thay bộ lọc	20
PHẦN IV: KẾT LUẬN	22
TÀI LIỆU THAM KHẢO	23

DANH MỤC HÌNH VẼ

Hình 1:Mô hình User-case khách hàng.....	8
Hình 2:Mô hình User-Case của chủ cửa hàng	9
Hình 3:Mô hình User-case của Firebase	9
Hình 4:Mô hình User-case của System	10
Hình 5:Màn hình đăng kí,đăng nhập.	12
Hình 6: Màn hình chính của giao diện.	13
Hình 7:Danh sách các sản phẩm đã mua.	13
Hình 8:Thời gian nên chi tiết của từng lõi lọc.....	14
Hình 9:Các sản phẩm bày bán.	15
Hình 10:Trang web hỗ trợ.	15
Hình 11:Mô hình dữ liệu	16
Hình 12: Thông tin đăng nhập	17
Hình 13:Thông tin khách hàng.....	18
Hình 14:Thông tin sản phẩm,hóa đơn.....	19
Hình 15 :Lịch sử thay các bộ lọc.....	20

PHẦN I: ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP

1.1. Tổng quan.

1.1.1. Các vấn đề.

- Trong hoàn cảnh hiện nay,khi khí hậu ngày càng biến đổi,môi trường ảnh hưởng nhiều đến chất lượng cuộc sống của mỗi gia đình hiện nay.Ô nhiễm nguồn nước cũng là 1 vấn đề mà ở Việt Nam ta đang phải đối mặt.Vì thế mà,những chiếc máy lọc nước ngày càng trở nên thân thuộc với mỗi gia đình.Tuy nhiên,không ít người quên rằng việc sử dụng máy lọc nước cần phải thay đổi các bộ lọc theo đúng thời gian sử dụng,điều đó khiến cho những chiếc máy lọc nước mất đi tính an toàn của nó.Chính vì vậy mà em đã nghĩ ra ý tưởng tạo 1 app Android giúp người dung có thể biết khi nào cần thay bộ lọc.
- Tìm hiểu các loại máy lọc nước trên thị trường hiện nay.
- Nắm được số bộ lọc và chức năng của từng lõi lọc trong các sản phẩm.Đồng thời biết được thời gian sử dụng tối ưu của chúng.
- Tìm hiểu về lập trình Android cơ bản và các vấn đề liên quan để xây dựng phát triển ứng dụng.

1.1.2. Mục tiêu cần đạt được.

- Hiện thị thời gian để nên thay lõi lọc trong sản phẩm.

- Thông báo qua điện thoại cá nhân của người dùng về thời gian nên thay lõi lọc sản phẩm của khách hàng.

1.1.3. Lựa chọn và định hướng thiết kế

- Sử dụng Android Studio để xây dựng giao diện hiển thị, xây dựng các tính năng như yêu cầu.
- Quản lý code của dự án với Git và kho lưu trữ GitHub.
- Sử dụng dịch vụ lưu trữ Real-time của Firebase Google để lưu dữ liệu khách hàng.

1.2. Các công nghệ kỹ thuật liên quan

- **Android Studio**
 - Có nhiều công cụ để phát triển Android nhưng đến nay công cụ chính thức và mạnh mẽ nhất là Android Studio. Đây là IDE (Môi trường phát triển tích hợp) chính thức cho nền tảng Android, được phát triển bởi Google và được sử dụng để tạo phần lớn các ứng dụng mà bạn có thể sử dụng hàng ngày.
 - Android Studio lần đầu tiên được công bố tại hội nghị Google I/O vào năm 2013 và được phát hành cho công chúng vào năm 2014 sau nhiều phiên bản beta khác nhau. Trước khi được phát hành, các nhà phát triển Android thường sử dụng các công cụ như Eclipse IDE, một IDE Java chung cũng hỗ trợ nhiều ngôn ngữ lập trình khác.
 - Android Studio khiến việc tạo ứng dụng trở nên dễ dàng hơn đáng kể so với phần mềm không chuyên dụng. Đối với người mới bắt đầu, có rất nhiều thứ để học và nhiều thông tin có sẵn, thậm chí thông qua các kênh chính thức nhưng chúng có thể đã lỗi thời hoặc quá nhiều thông tin khiến họ cảm thấy choáng ngợp.
- **Git và GitHub**
 - Git là tên gọi của một Hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. DVCS nghĩa là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (commit) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc

clone lại một tập hợp các thay đổi mới nhất trên máy tính kia. Trong Git, thư mục làm việc trên máy tính gọi là Working Tree.

- GitHub là một dịch vụ cung cấp kho lưu trữ mã nguồn Git dựa trên nền web cho các dự án phát triển phần mềm. GitHub cung cấp cả phiên bản trả tiền lẫn miễn phí cho các tài khoản. Các dự án mã nguồn mở sẽ được cung cấp kho lưu trữ miễn phí. Nó trở thành máy chủ chứa mã nguồn lớn trên thế giới. Github đã trở thành một yếu tố có sức ảnh hưởng trong cộng đồng phát triển mã nguồn mở. Thậm chí nhiều nhà phát triển đã bắt đầu xem nó là một sự thay thế cho sơ yếu lý lịch và một số nhà tuyển dụng yêu cầu các ứng viên cung cấp một liên kết đến tài khoản Github để đánh giá ứng viên.

1.3. Các framework/thư viện lập trình

- **Google Firebase**

- Đó là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.
- Các dịch vụ sử dụng là :Real-time database và xác nhận người dùng.
- Lợi ích của việc sử dụng Google Firebase là : Triển khai nhanh, ổn định và tính bảo mật cao.
- Cài đặt.
 - Trước hết cần tạo 1 tài khoản Google sau đó đăng nhập vào trang <https://firebase.google.com/> để tạo project Và làm theo hướng dẫn.
 - Để cài ta thêm 2 phần sau dòng trên trong file build.gradle(app).

```
dependencies {  
    implementation 'com.google.firebase:firebase-database:16.0.1'  
    implementation 'com.google.firebase:firebase-auth:16.0.3'  
    ...  
    apply plugin : 'com.google.gms.google-services'
```

- Và thêm 1 phần sau trong build.gradle(project).

```
dependencies {  
    classpath 'com.google.gms:google-services:4.2.0'  
    ...  
}
```

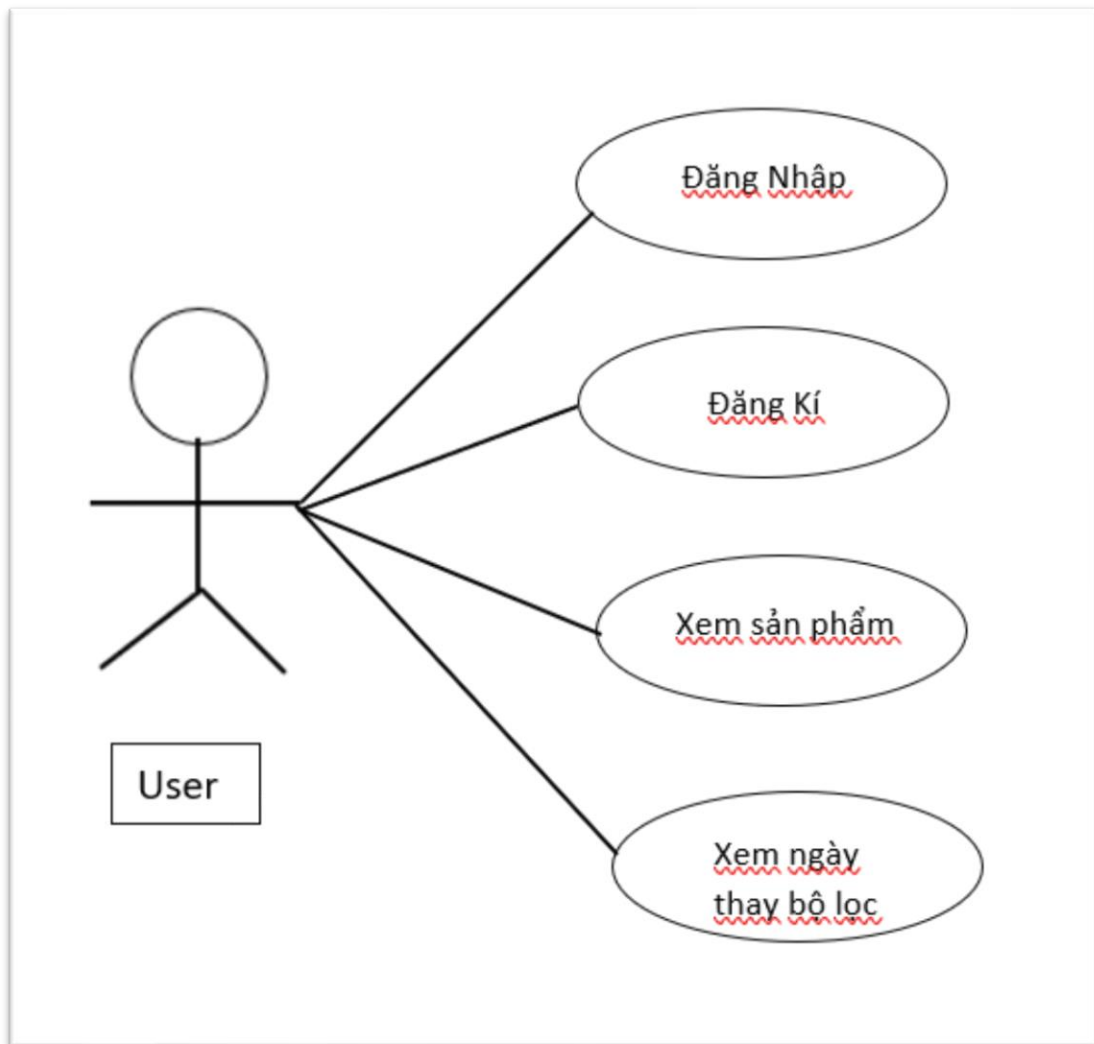
-
- **Picasso**

- Picasso là một thư viện Android mã nguồn mở rất phổ biến, dùng để load hoặc hiển thị hình ảnh trong ứng dụng Android.
- Picasso có những ưu điểm sau:
 - Đơn giản hóa code, chính là ưu điểm hàng đầu của Picasso, thay vì viết nhiều dòng bạn chỉ cần viết có 1 dòng là có thể load được ảnh.
 - Tối ưu hóa bộ nhớ sử dụng của ứng dụng.
 - Tự động sử dụng RAM cache và Disk cache để tối ưu hóa lưu lượng mạng và tăng tốc độ hiển thị ảnh.
 - Tự động transforms bitmap và cancel download ngữ cảnh.
- Cài đặt.
 - Để cài đặt ta thêm dòng trên trong file build.gradle(app).

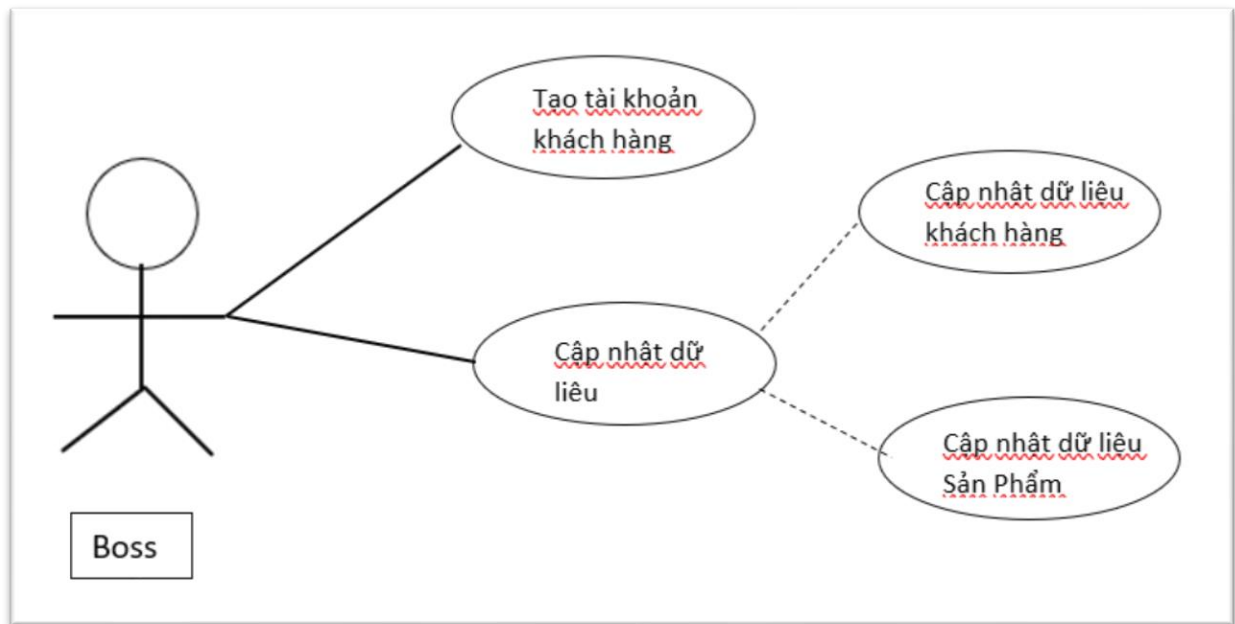
```
dependencies {  
    implementation 'com.squareup.picasso:picasso:2.71828'  
}
```

PHẦN II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

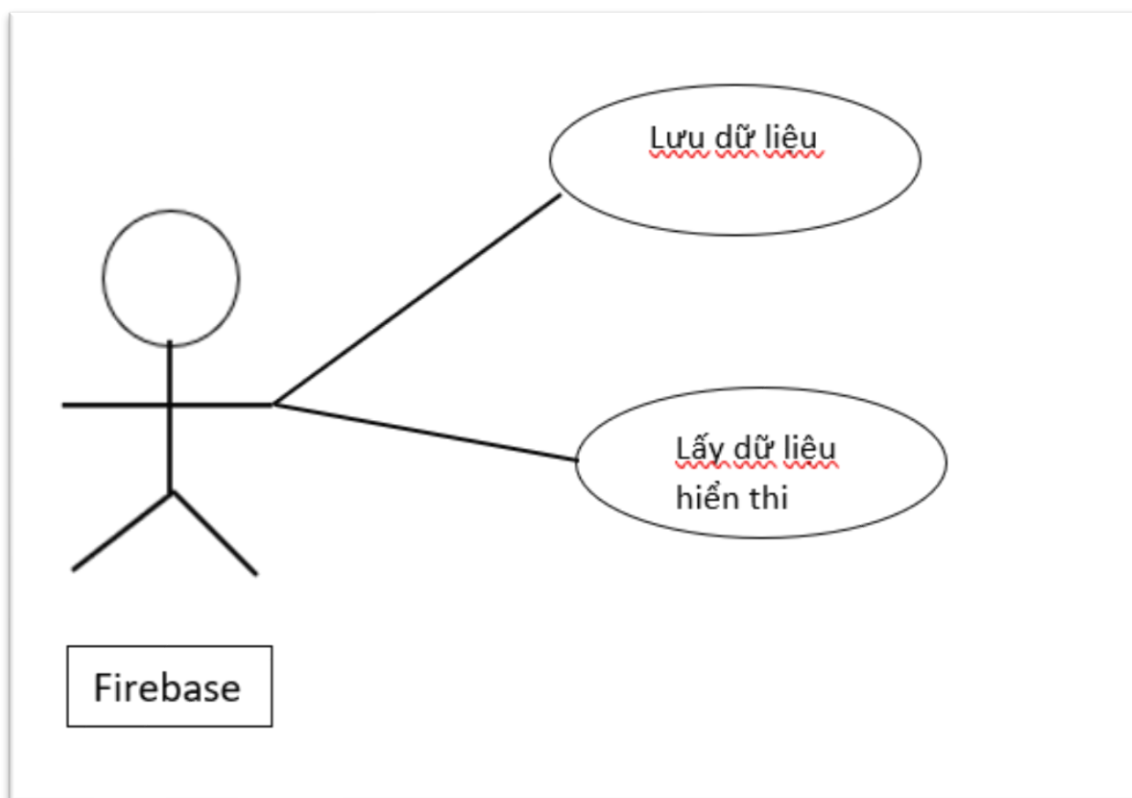
2.1. Các thành phần chức năng.



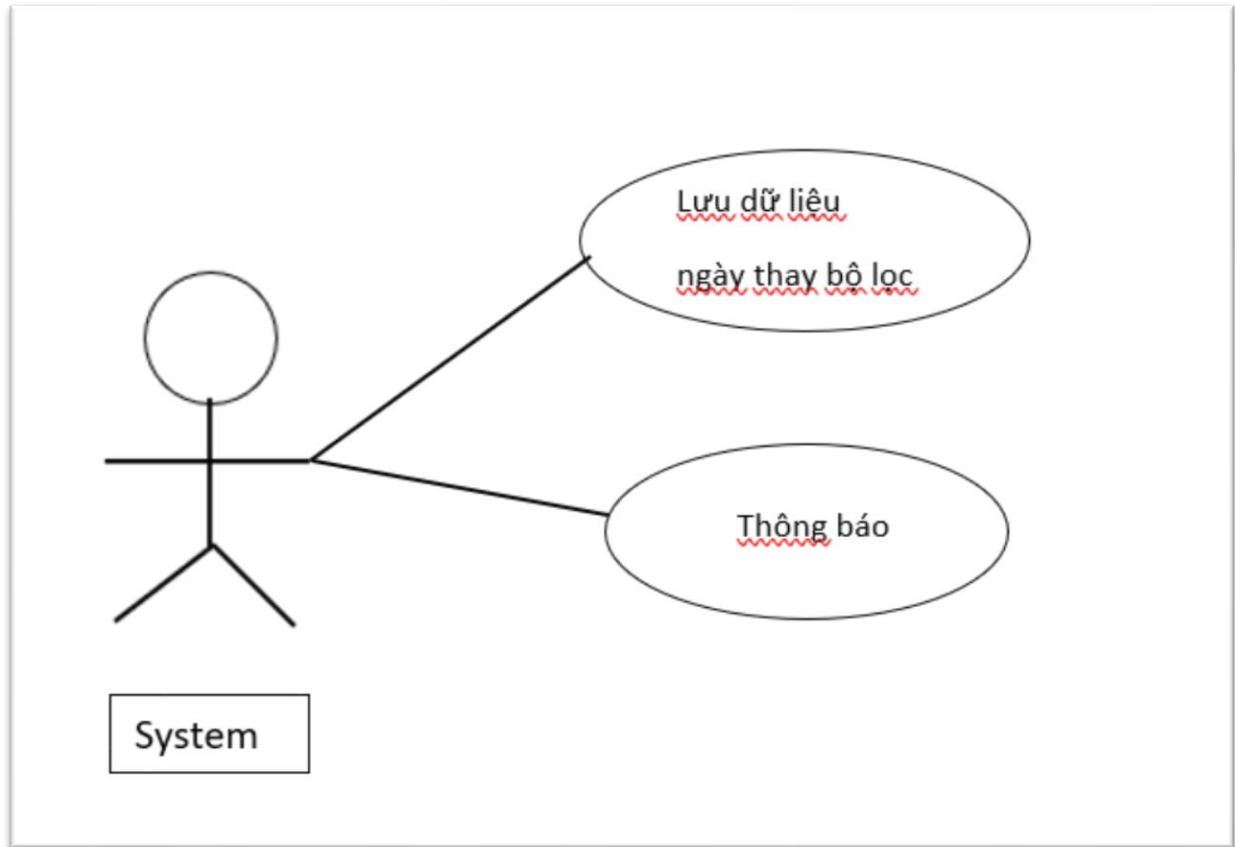
Hình 1: Mô hình User-case khách hàng



Hình 2: Mô hình User-Case của chủ cửa hàng



Hình 3: Mô hình User-case của Firebase



Hình 4: Mô hình User-case của System

2.1.1. Đăng nhập, đăng kí, đăng xuất

- Đăng kí.
 - Khi khách hàng đến mua hàng, thì người chủ sẽ tạo cho người đó một tài khoản mặc định là số điện thoại của người đó để đăng nhập và sử dụng app.
 - Với những người không mua hàng ở cửa hàng đó vẫn có thể tạo 1 tài khoản để truy cập các chức năng khác.
- Đăng nhập.
 - Khách hàng đăng nhập vào app, phía server sẽ kiểm tra tính đúng đắn của tài khoản. Khi đăng nhập vào thì chức năng thông báo cũng sẽ được kích hoạt.
- Đăng xuất.
 - Giúp người dùng có thể sử dụng với nhiều tài khoản khác nhau.
 - Đồng thời, chức năng thông báo của tài khoản cũ cũng được kết thúc.

2.1.2. Hiện thị thông tin các sản phẩm đã mua.

- Khi đăng nhập vào hệ thống.App sẽ hiển thị các sản phẩm mà user đó đã mua trong cửa hàng cũng các thông tin liên quan đến khách hàng.

2.1.3. Hiển thị thời gian cần thay các bộ lọc khi cần

- Với mỗi sản phẩm ta đều có thể xem số bộ lọc của chúng,cũng thời gian thay gần nhất,và ngày cần thay.
- Khi mua thì ngày mua được tính là ngày đầu tiên sử dụng.

2.1.4. Thông báo thay bộ lọc khi đến hạn

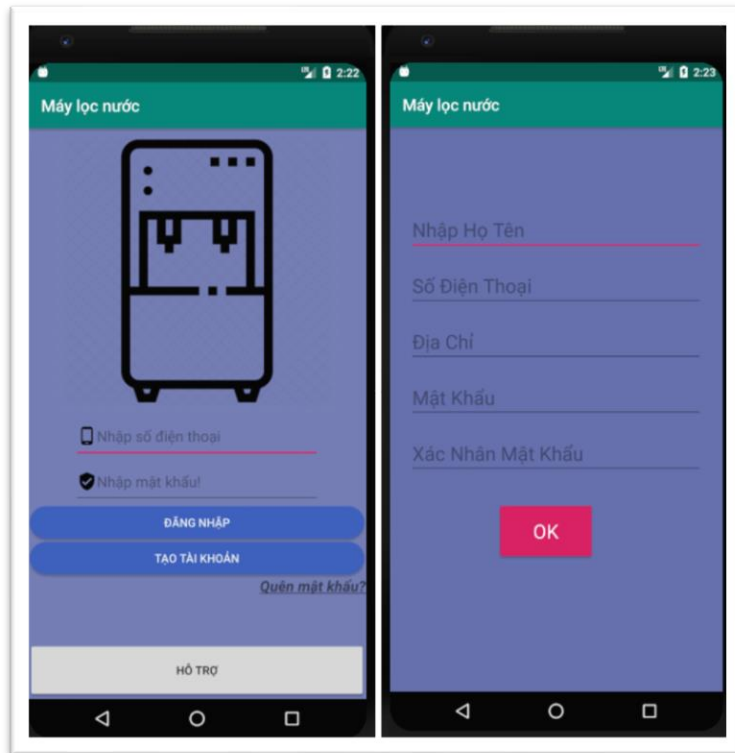
- Với ngày hiện thị cần thay của mỗi bộ lọc thì người dùng có thể biết được ngày cần thay.Tuy nhiên không phải ai cũng để ý,vì vậy cần với chức năng thông báo.
- Thông báo sẽ được hiển thị trên thanh trạng thái của app,với tần suất 2 lần một ngày.Thông báo sẽ tự tắt nếu không có bộ lọc nào cần thay.

2.1.5. Hỗ trợ người dùng,Hiển thị các sản phẩm của cửa hàng

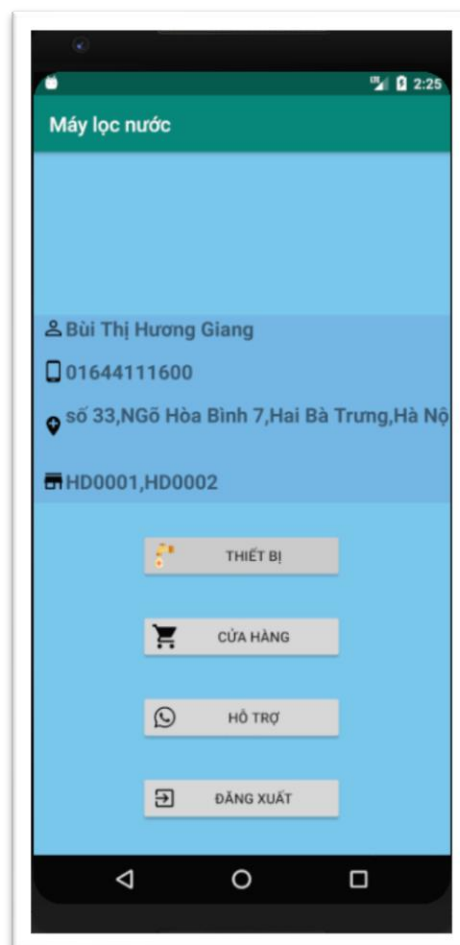
- App có 1 số chức năng phụ.Trong đó có hỗ trợ người dùng.Khi gặp vấn đề người dùng có thể liên hệ với hotline trong app,hay website của cửa hàng.
- Ngoài ra,app còn hiển thị thông tin các sản phẩm đang bày bán ở cửa hàng để người dùng có thể tham khảo.

2.2. Thiết kế giao diện/layout

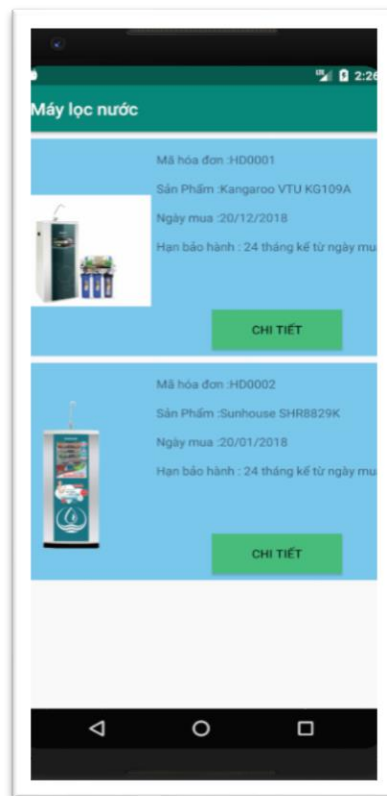
- Giao diện đăng kí đăng nhập :Cho phép người dùng đăng nhập,hoặc tạo tài khoản (nếu chưa có).
- Màn hình chính:Hiển thị các chức năng và thông tin khách hàng.
- Danh sách các sản phẩm đã mua.
- Thời gian thay lõi lọc của sản phẩm.
- Sản phẩm đang bán ở cửa hàng.
- Trang web hỗ trợ của cửa hàng.



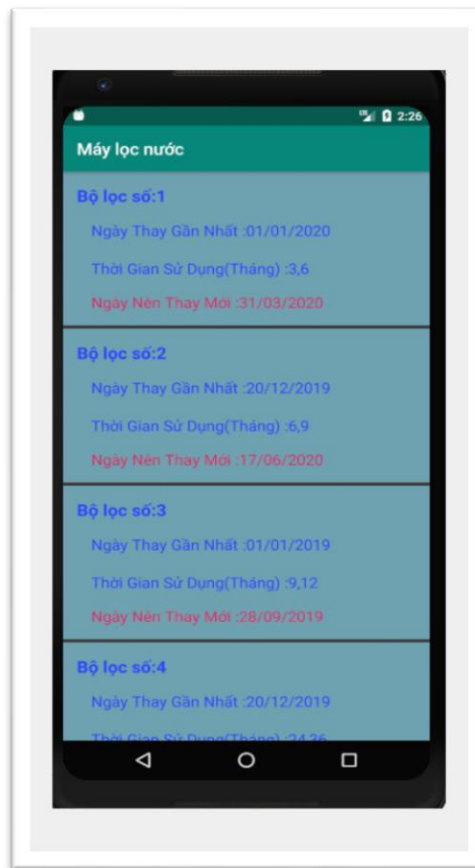
Hình 5: Màn hình đăng kí, đăng nhập.



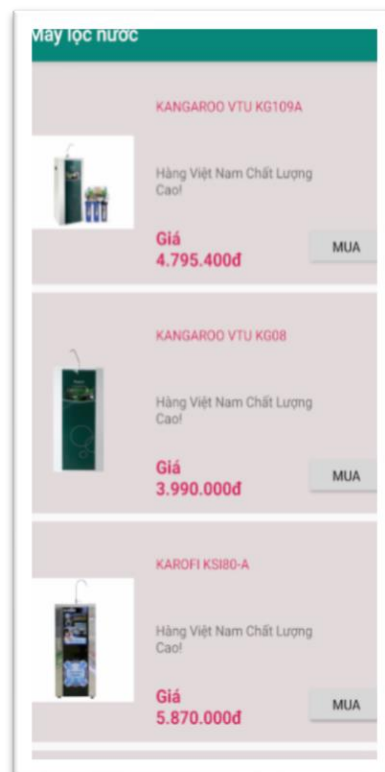
Hình 6: Màn hình chính của giao diện.



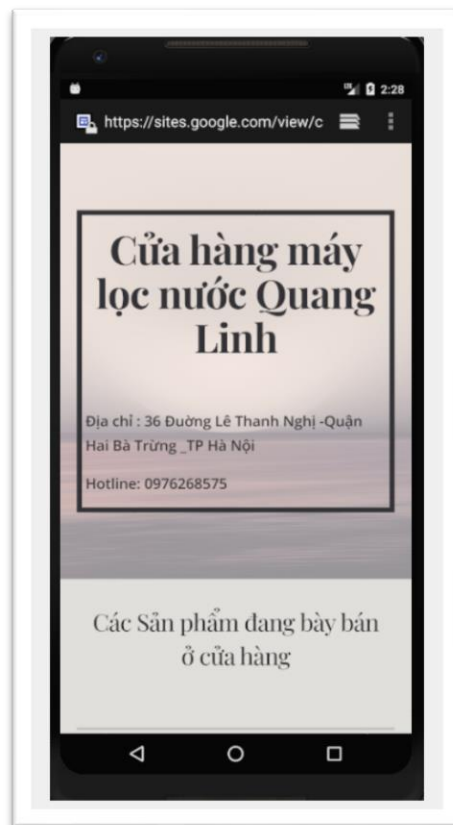
Hình 7: Danh sách các sản phẩm đã mua.



Hình 8: Thời gian nên chi tiết của từng lõi lọc.



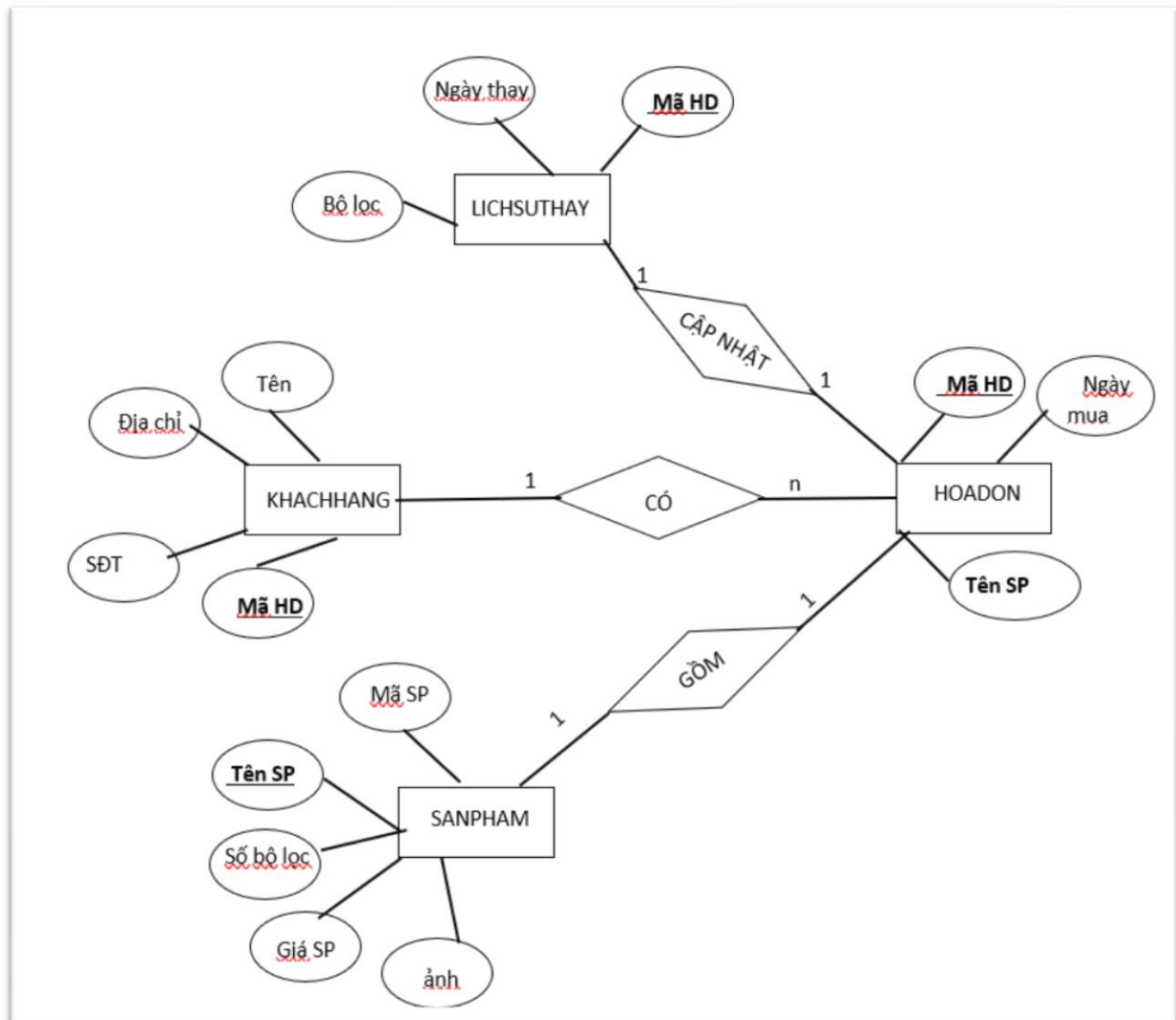
Hình 9: Các sản phẩm bày bán.



Hình 10: Trang web hỗ trợ.

PHẦN III: CÀI ĐẶT VÀ TRIỂN KHAI GIẢI PHÁP

3.1. Mô hình dữ liệu.



Hình 11: Mô hình dữ liệu

3.2. Đăng kí, Đăng nhập, Đăng xuất

- Đăng kí : Có 2 Cách
 - Cách 1 : Với những người mua hàng ở cửa hàng thì người chủ sẽ tạo cho người dùng 1 tài khoản bằng số điện thoại của khách (Sử dụng phần mềm riêng của chủ cửa hàng) kèm theo hóa đơn mà người đó đã mua sản phẩm.
 - Cách 2: Đăng kí trên app. Người dùng phải điền đầy đủ các thông tin liên quan đến tên, số điện thoại, địa chỉ, mật khẩu.

Cả 2 cách thì dữ liệu đều được lưu và đẩy lên lưu trên firebase. Nếu đăng kí thành công thì sẽ chuyển đến màn hình đăng nhập.


```
AUFIREFBASE.createUserWithEmailAndPassword(phone, pass)
    .addOnCompleteListener(SignUp.this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            PROGRESSDIALOG.hide();
            if (!task.isSuccessful()) {
                Toast.makeText(SignUp.this, "Không Thành Công", Toast.LENGTH_SHORT).show();
            } else {
                AUFIREFBASE.signOut();
                databaseKhachHang.push().setValue(new KhachHang(
                    name, address, phone, "Chưa mua sản phẩm nào"
                ));
                startActivity(new Intent(SignUp.this, Login.class));
            }
        }
    });
```

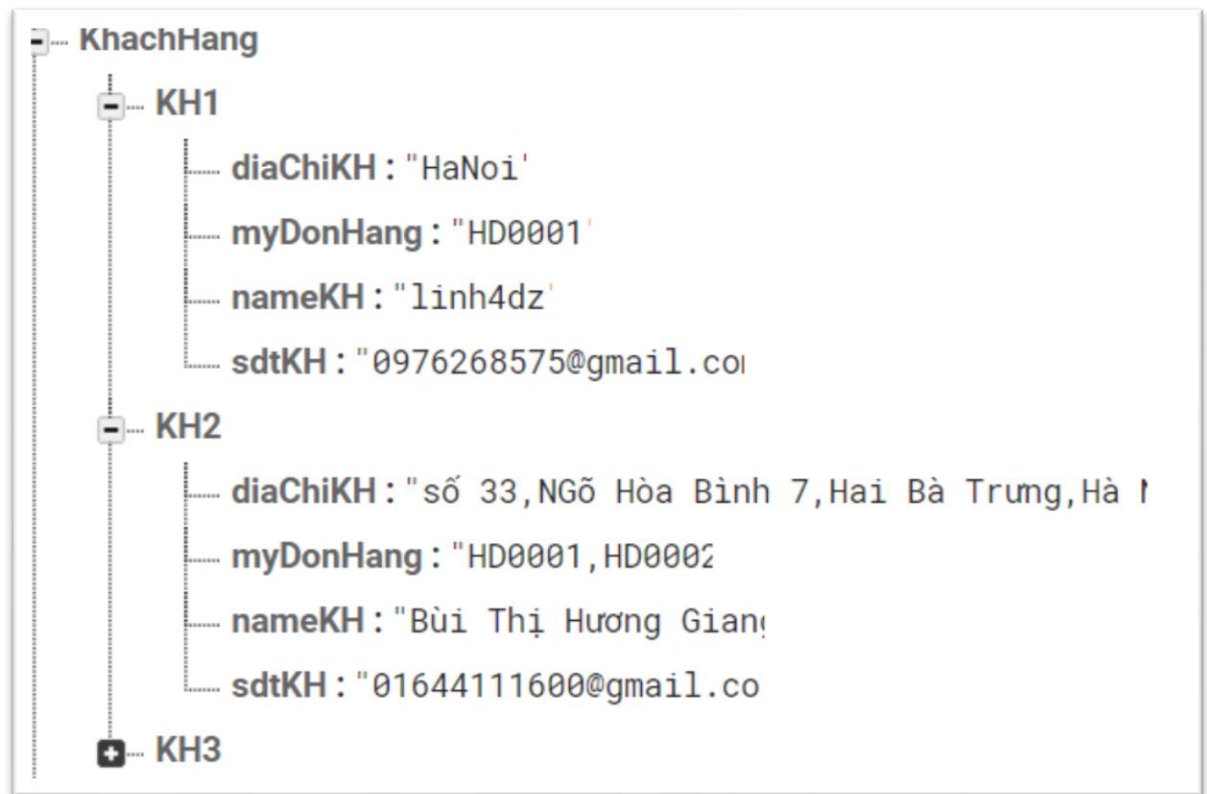
Khi đăng kí thành công dữ liệu sẽ được lưu như sau:

Thông tin về đăng nhập sẽ được lưu trong mục **Authentication** của Firebase.

login	suppliers	Creation date	last connection	user ID ↑
0976268575@gmail.com	✉	March 2, 2019	Apr 28, 2019	YP2O6hACVAhVkwf9KyVgPA2ijR2
01644111600@gmail.com	✉	March 4, 2019	May 11, 2019	kxxUmTgGIHeBILQTmNbvYEDvQm...
0923604382@gmail.com	✉	March 27, 2019	March 27, 2019	tkyrQ15co0M1Or7yLfX9fsbw7tE3

Hình 12: Thông tin đăng nhập

Còn thông tin về khách hàng được lưu trên Realtime_database.



Hình 13: Thông tin khách hàng

- Đăng nhập.

Người dùng điền mật khẩu và tài khoản của mình. AUFIREBASE sẽ tìm trên các tài khoản đang có trên Authentication. Nếu thành công ứng dụng chuyển sang màn hình chính của app. Nếu sai thì app sẽ thông báo và người dùng phải nhập lại.

Người dùng chỉ cần phải đăng nhập cho lần đầu tiên sử dụng app. Người dùng chỉ phải đăng nhập lại nếu đã đăng xuất.

```
• AUFIREBASE.signInWithEmailAndPassword(phone, pass)
    .addOnCompleteListener(Login.this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (!task.isSuccessful()) {
                Toast.makeText(Login.this, "Số điện thoại hoặc mật khẩu sai!",
                    Toast.LENGTH_SHORT).show();
            } else {
                startActivity(new Intent(Login.this, Home.class));
                finish();
            }
        }
    });
```

- Đăng xuất

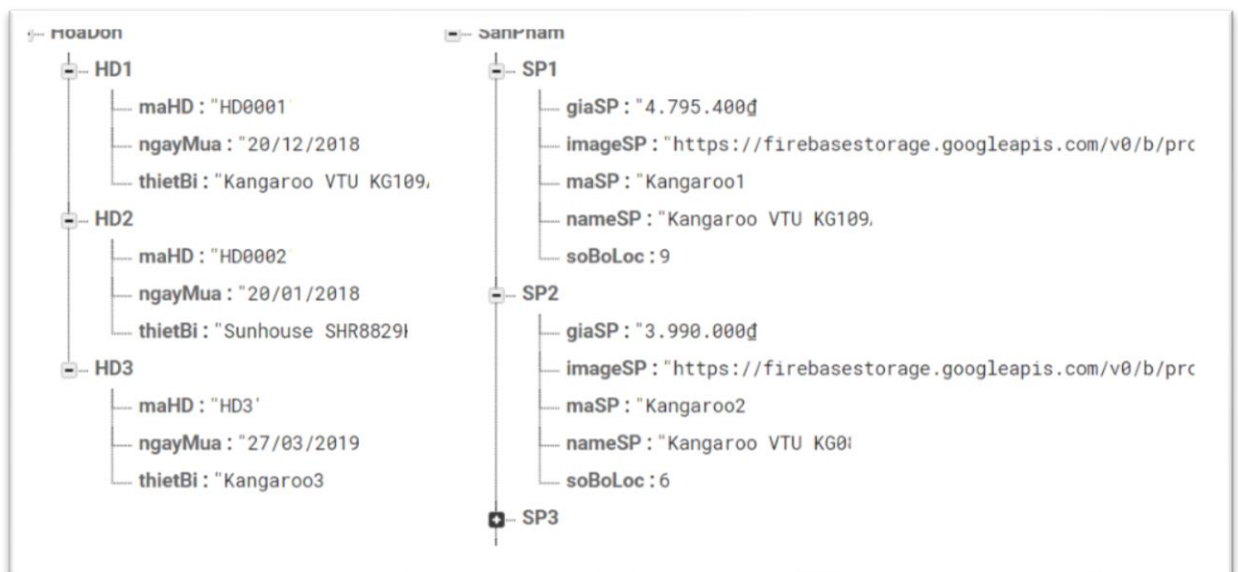
Ngoài nhiệm vụ thông thường là để tắt sử dụng tài khoản hiện tại, có thể sử dụng tài khoản khác. Đăng xuất sẽ làm cho app tắt thông báo cho các sản phẩm trước đó.

```
checkLogOut = true;
ngayThayBoLoc.clear();
AUFIREBASE.signOut();
startActivity(new Intent(Home.this, Login.class));
```

3.3. Hiện thị các sản phẩm đã mua

Để hiển thị các sản phẩm. Tôi dùng RecyclerView trong Android.

- Với mỗi khách hàng đến mua hàng thì mỗi sản phẩm sẽ được lưu dưới dạng 1 hóa đơn có cấu trúc : Mã Hóa đơn, Ngày mua, Tên thiết bị đã mua.
- Với thông tin khách hàng đã trình bày ở trên đều có thông tin về hóa đơn. Khi đó, khi đăng nhập màn hình sẽ hiển thị các sản phẩm mà người dùng có trong các hóa đơn mua hàng.



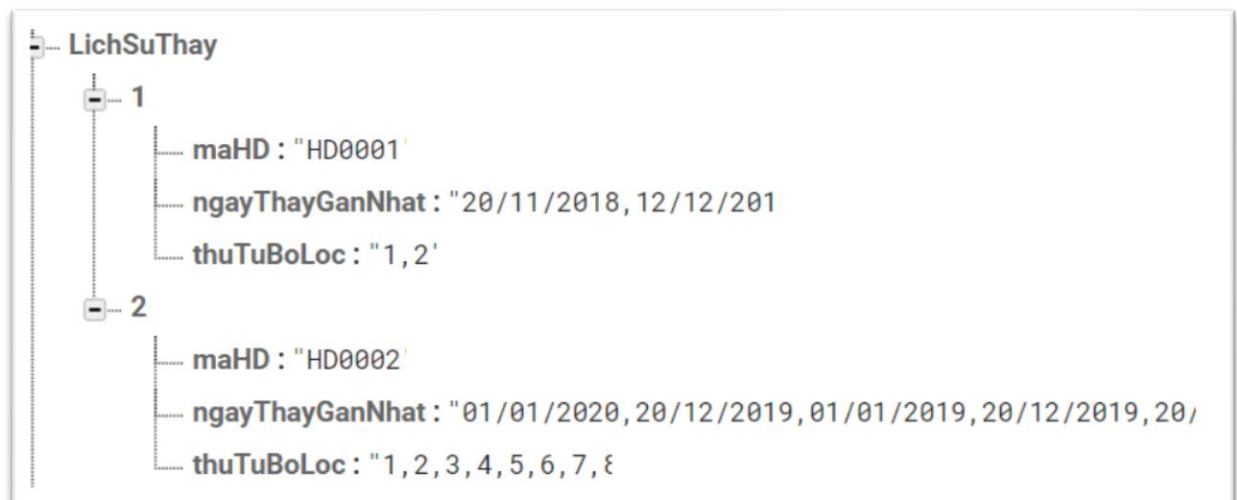
Hình 14: Thông tin sản phẩm, hóa đơn

```
Intent intent = getIntent();
final String idHoaDon = intent.getStringExtra("mahoadon");
listDH = new ArrayList<>();
for(int i = 0; i < donHangList.size(); i++){
    if(idHoaDon.contains(donHangList.get(i).getMaHD())){
        listDH.add(donHangList.get(i));
    }
}
recyclerView = findViewById(R.id.rv_thiet_bi);
thietBiAdapter = new ThietBiAdapter(listDH, this);
```

```
GridLayoutManager gridLayoutManager = new GridLayoutManager(this, 1);
recyclerView.setLayoutManager(gridLayoutManager);
recyclerView.setAdapter(thietBiAdapter);
```

3.4. Hiển thị thời gian cần thay các bộ lọc

Cũng giống như cách hiển thị ở trên. Cách làm cũng tương tự. Với những bộ lọc chưa thay bao giờ thì ngày bắt đầu sử dụng là ngày mua sản phẩm. Sau mỗi lần thay thì thời gian đó được cập nhật lại.



Hình 15 :Lịch sử thay các bộ lọc

3.5. Thông báo thời gian thay bộ lọc

Đây là chức năng chính và cũng là mục đích chính của app.

Để sử dụng Service trong Android. Trong file AndroidManifest thêm khai báo.

```
<service
    android:name=".Notification"
    android:enabled="true"
    android:exported="true" />
```

Trước hết ở tính năng hiển thị thời gian cần thay bộ lọc. Tôi đã lấy được danh sách thời gian cần thiết. Thông báo được hiện trên màn hình chờ khi kiểm tra thấy thời gian hiện tại lớn hơn 1 thời gian bất kỳ nào đó trong danh sách đó.

```
AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
Intent intent = new Intent(Home.this, Notification.class);
PendingIntent pendingIntent = PendingIntent.getService(
    Home.this,
    0,
    intent,
    PendingIntent.FLAG_UPDATE_CURRENT);

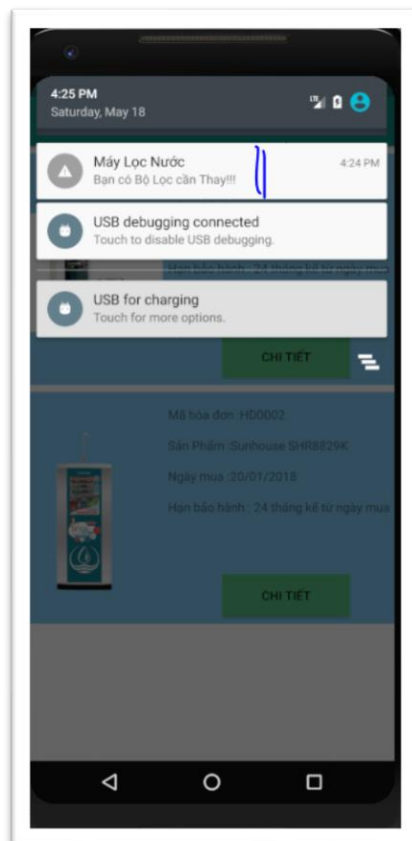
alarmManager.setRepeating(
    AlarmManager.RTC_WAKEUP,
    Calendar.getInstance().getTimelnMillis(),
    AlarmManager.INTERVAL_DAY,
    pendingIntent
);
```

Ở đây, Tôi dùng 1 con AlarmManager có sẵn của Android. Với tần suất thông báo 2 lần 1 ngày (`AlarmManager.INTERVAL_DAY`).

Cài đặt thông báo

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(this)
    .setContentTitle("Máy Lọc Nước")
    .setSmallIcon(R.drawable.ic_warning_black_24dp)
    .setContentText("Bạn có Bộ Lọc cần Thay!!!")
    .setAutoCancel(true)
    .setVisibility(android.app.Notification.VISIBILITY_PUBLIC)
    .setSound(RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION))
    .setContentIntent(pendingIntent);
```

Khí đó màn hình thông báo sẽ như sau:



PHẦN IV: KẾT LUẬN

Phần mềm “Quản lý máy lọc nước” về cơ bản đã hoàn thành với các chức năng cơ bản và đầy đủ theo yêu cầu, đáp ứng được nhu cầu cần thiết của người dùng. Vì máy lọc nước là sản phẩm phổ biến trên khắp mọi miền, mọi đối tượng, vì vậy em đã thiết kế một cách đơn giản nhất, để những người ít tiếp xúc với smart phone hay người lớn tuổi có thể sử dụng 1 cách đơn giản nhất. Tuy đã đáp ứng được phần nào nhu cầu người dùng, App vẫn còn 1 số khiếm khuyết về giao diện chưa được bắt mắt, một số chức năng còn sơ sài, đây cũng là động lực để em hoàn thành và phát triển sản phẩm trong tương lai với các version sau. Em xin được gửi lời cảm ơn đến PGS-TS Trình Văn Loan đã giúp em hoàn thành sản phẩm và môn học trên!

TÀI LIỆU THAM KHẢO

1. Quy định về đồ án tốt nghiệp của Viện CNTT-TT
<https://soict.hust.edu.vn/index.php/2017/03/16/quy-dinh-ve-do-an-tot-nghiep/>
2. Git <http://dce.hust.edu.vn/huong-dan-su-dung-git/>
3. Branch: <https://youtu.be/JtIX3HJKwfo>
4. Picasso : <https://square.github.io/picasso/>
5. Firebase: <https://firebase.google.com/docs/android/setup>
6. AlarmsService : <https://developer.android.com/training/scheduling/alarms>