TRƯỜNG ĐẠI HỌC BÀ RỊA VŨNG TÀU

----------

KHOA KỸ THUẬT - CÔNG NGHỆ

Logo, company name

Description automatically generated

**---------------------------------**

**BÁO CÁO MÔN**

**LẬP TRÌNH ỨNG DỤNG DI ĐỘNG GAME 2**

**ĐỀ TÀI: TRANG TRÍ NỘI THẤT**

**VỚI CÔNG NGHỆ AR**

|  |  |  |
| --- | --- | --- |
| Trình độ đào tạo | : | Đại Học |
| Tên sinh viên | : | Nguyễn Văn Sơn |
| Mã sinh viên | : | 20030016 |
| Lớp | : | DH20LT |
| Chuyên Ngành | : | Lập trình ứng dụng di động  và game |
| Giảng viên hướng dẫn | : | Nguyễn Thị Minh Nương |

BÀ RỊA – VŨNG TÀU, NĂM 2022

MỤC LỤC

[**MỞ ĐẦU** 4](#_Toc123121386)

[**LỜI CÁM ƠN** 6](#_Toc123121387)

[**Chương I: TỔNG QUAN** 9](#_Toc123121388)

[**1.1 Mục đích và ý nghĩa của đề tài** 9](#_Toc123121389)

[1.1.1 Mục đích 9](#_Toc123121390)

[1.1.2 Ý nghĩa của đề tài 9](#_Toc123121391)

[**1.2 Nhiệm vụ của đề tài** 10](#_Toc123121392)

[1.2.1 Về lý thuyết 10](#_Toc123121393)

[1.2.2 Về chương trình 11](#_Toc123121394)

[**1.3 Phương pháp triển khai** 11](#_Toc123121395)

[**Chương II: PHÂN TÍCH THIẾT KẾ** 12](#_Toc123121396)

[**3.1 Xác định yêu cầu** 12](#_Toc123121397)

[3.1.1 Chức năng chính 12](#_Toc123121398)

[a) Chức năng của app 12](#_Toc123121399)

[b) Chức năng của Firebase 12](#_Toc123121400)

[**3.2 Cơ sở dữ liệu** 12](#_Toc123121401)

[**3.4 Sơ đồ User-case** 15](#_Toc123121402)

[**Chương 4: XÂY DỰNG ỨNG DỤNG** 16](#_Toc123121403)

[**4.1 Ứng dụng** 16](#_Toc123121404)

[4.1.1 Danh sách các màn hình. 16](#_Toc123121405)

[4.1.2 Màn hình khi mở ứng dụng 17](#_Toc123121406)

[4.1.3 Màn hình OTPVerify 19](#_Toc123121407)

[4.1.4 Màn hình CreateProfile 24](#_Toc123121408)

[4.1.5 Màn hình home và HomeFragment 28](#_Toc123121409)

[4.1.6 Màn hình Profile 33](#_Toc123121410)

[4.1.7 Màn hình Favorite 39](#_Toc123121411)

[4.1.8 Màn hình product 42](#_Toc123121412)

[4.1.9 Màn hình Main 44](#_Toc123121413)

[4.1.10 File Build gradle 54](#_Toc123121414)

[4.1.11 File Android Manifest 55](#_Toc123121415)

[**Chương III: TỔNG KẾT** 59](#_Toc123121416)

**MỞ ĐẦU**

Vài năm trở lại đây, Pokemon Go là cái tên “gây sốt” trên thị trường game điện tử tại Việt Nam và cả nước ngoài. Được ứng dụng công nghệ thực tế ảo tăng cường (AR), Pokemon Go đã phủ sóng khắp các quốc gia và trở thành trò chơi “quốc dân” được đại đa số người dùng ưa chuộng**.**

Thực tế ảo tăng cường (Augmented Reality – AR) là một xu hướng công nghệ mới được phát triển trên nền tảng công nghệ thực tế ảo (Virtual Reality). Công nghệ này cho phép những sự vật/hiện tượng mô phỏng theo [dạng 3D](https://tino.org/vi/in-3d-la-gi/) xuất hiện trong môi trường thật, không gian thật qua màn hình máy tính hoặc điện thoại. Ngoài ra, thực tế ảo tăng cường còn hỗ trợ thêm âm thanh, đồ họa, video… nhằm mang đến cảm giác chân thật và sinh động cho người dùng. Hiện nay, công nghệ này đã trở nên cực kỳ phổ biến vì chúng có mặt trên tất cả các thiết bị điện tử, đặc biệt là smartphone – “vật bất ly thân” của con người thời 4.0.

Nhận thấy tiềm năng của công nghệ này, tuy nhiên ở Việt Nam vẫn còn khá ít sản phẩm áp dụng. Vì vậy em đã quyết định chọn đề tài “Trang trí nội thất bằng công nghệ AR” để ôn luyện lại những kiến thức ở các học phần Game 1 và Game 2 cũng như tìm hiểu thêm về AR.

### **LỜI CÁM ƠN**

Đầu tiên cho phép em được gửi lời cảm ơn chân thành và sâu sắc nhất đến quý thầy cô giáo Đại Học Bà Rịa – Vũng Tàu, đặc biệt là quý thầy cô Khoa Công Nghệ Thông Tin đã tận tình chỉ dạy và quan tâm trong suốt quá trình học tập và rèn luyện tại trường.

Chúng em trân trọng cám ơn cô Nguyễn Thị Minh Nương đã tận tình hướng dẫn, chỉ bảo, góp ý cho em trong suốt môn học, qua đó giúp em có những kiến thức cần thiết để thực hiện đề tài này.

Đây là một chủ đề khá khó đối với một người mới, vì vậy đề tài của em không thể tránh khỏi những thiếu sót nên em rất mong nhận được những ý kiến đóng góp từ quý thầy cô và các bạn.

* Vũng Tàu, 28 tháng 12 năm 2022

**NHẬN XÉT**

**(Của giảng viên hướng dẫn)**

**NHẬN XÉT**

**(Của giảng viên phản biện)**

### **Chương I: TỔNG QUAN**

### **1.1 Mục đích và ý nghĩa của đề tài**

### 1.1.1 Mục đích

Ứng dụng trang trí nội thất chạy trên nền tảng Android được thực hiện dựa trên hai mục đích chính sau :

Thứ nhất là nghiên cứu công nghệ Argument Facility(Thực tế tăng cường) và áp dụng vào việc phát triển ứng dụng.

Thứ hai là mở ra một hướng đi mới cho ngành thương mai điện tử nói chung và ngành nội thất nói riêng, giúp những người có nhu cầu mua sắm nội thất có thể thử nó ngay tại nhà của mình trên chiếc điện thoại di động.

### 1.1.2 Ý nghĩa của đề tài

Trước hết, đề tài “Trang trí nội thất bằng công nghệ AR” là một ứng dụng được xây dựng như một đề tài thể hiện việc áp dụng những kiến thức quý báu đã được các thầy cô của trường Đại học Bà Rịa – Vũng Tàu nhiệt tình truyền thụ lại cho chúng em. Và đặc biệt là sự theo dõi và quan tâm giúp đỡ của cô Nguyễn Thị Minh Nương trong suốt thời gian em thực hiện đề tài này.

Thứ hai là những kinh nghiệm quý báu mà em có được trong quá trình thực hiện đề tài sẽ là hành trang tuyệt vời giúp ích rất nhiều cho công việc sau này của mình.

Thứ ba, trong thời đại công nghệ thông tin phát triển như vũ bão ngày nay, công nghệ không ngừng được cải tiến, những công nghệ mới ra đời đòi hỏi người lập trình viên phải cố gắng nghiên cứu và tìm hiểu để áp dụng vào thực tiễn. Điều này cũng được thể hiện qua sự cố gắng em khi quyết định thực hiện đề tài này với các công nghệ được coi là mới nhất hay và được dự báo sẽ phát triển mạnh mẽ trong tương lai.

Do vậy, em rất hy vọng sẽ tạo nên một sản phẩm hoàn chỉnh nhất có thể để đáp lại sự nhiệt tình giúp đỡ từ thầy cô và cả tâm huyết của em khi quyết định thực hiện đề tài này.

### **1.2 Nhiệm vụ của đề tài**

### 1.2.1 Về lý thuyết

Đề tài “Trang trí nội thất với công nghệ AR” là một ứng dụng chạy trên điện thoại sử dụng hệ điều hành Android đã khá quen thuộc. Tuy nhiên ở đây sử dụng một công nghệ khá mới mẻ đó là Argument Facility.

* Thực tế ảo tăng cường (Augmented Reality – AG) là một xu hướng công nghệ mới được phát triển trên nền tảng công nghệ thực tế ảo (Virtual Reality). Hiện nay, công nghệ này đã trở nên cực kỳ phổ biến vì chúng có mặt trên tất cả các thiết bị điện tử, đặc biệt là smartphone.
* Công nghệ này cho phép những sự vật/hiện tượng mô phỏng theo dạng 3D xuất hiện trong môi trường thật, không gian thật qua màn hình máy tính hoặc điện thoại. Ngoài ra, thực tế ảo tăng cường còn hỗ trợ thêm âm thanh, đồ họa, video,…, nhằm mang đến cảm giác chân thật và sinh động cho người dùng.
* Thực tế ảo tăng cường (AR) chỉ thật sự bùng nổ vào năm 2016 khi trò chơi Pokemon Go ra mắt người dùng, nhưng trên thực tế, công nghệ này xuất hiện từ rất lâu. Trước đây, thực tế ảo tăng cường chỉ được ứng dụng trong các lĩnh vực công nghệ phức tạp hoặc nghiên cứu khoa học.Giờ đây, công nghệ hiện đại này đã dần góp mặt trong đời sống thường nhật, bất kỳ ai cũng có thể tiếp cận với thực tế ảo tăng cường chỉ bằng một chiếc smartphone nhỏ gọn. Thực tế ảo tăng cường được ứng dụng phổ biến trong 5 hoạt động sau: giáo dục và đào tạo, giải trí, bán hàng, y tế
* Thư viện được em sử dụng lần này để tích hợp công nghệ AR vào ứng dụng đó SceneView. SceneView được phát triển dựa trên thư viện SceneForm của Google.

[Link Github SceneView](https://github.com/SceneView/sceneform-android)

### 1.2.2 Về chương trình

Chương trình thực hiện được các yêu cầu đề ra của đề tài cuối kỳ

Ứng dụng Furniture đã phần nào xây dựng và đáp ứng được một số chức năng chính:

* Đăng nhập bằng Authencation của Firebase
* Xem thông tin nội thất
* Cho phép tìm kiếm nội thất.
* Giao diện thân thiện, dễ sử dụng.
* Cho phép ngươi dùng thử nội thất.
* Cập nhật thông tin người dùng.
* Cho phép người dùng lưu và chia sẻ hình ảnh thử nội thất.

### **1.3 Phương pháp triển khai**

Công nghệ AR đang còn mới mẻ và ít được sử dụng tại Việt Nam *cần rất nhiều thời gian, tâm huyết và cả công* *sức để nghiên cứu.*

* *Khảo sát lấy yêu cầu người sử dụng điện thoại về các ứng dụng bán hàng, tham khảo các ứng dụng tương tự như IKEA Mobile App.*
* Tiếp đến, em sẽ nghiên cứu công nghệ phát triển ứng dụng chạy trên hệ điều hành Android. Xây dựng một app có thể cài đặt trên điện thoại chạy Android đáp ứng các yêu cầu cơ bản của hệ thống.
* Sau khi hoàn thành phần ứng dụng chạy trên điện thoại, em sẽ nghiên cứ*u* *đến việc lấy dữ liệu từ server về sau đó đưa lên ứng dụng. Ở đây, em sử dụng Firebase đã thêm dữ liệu cũng như lấy dữ liệu trả về ứng dụng.*

Sau đây là danh sách các công nghệ sử dụng để xây dựng ứng dụng :

* Android SDK
* Firebase
* Argurment Facility(SceneView)

### **Chương II: PHÂN TÍCH THIẾT KẾ**

### **3.1 Xác định yêu cầu**

### 3.1.1 Chức năng chính

### a) Chức năng của app

* Đăng nhập đơn giản chỉ cần nhập số điện thoại.
* Tìm kiếm, xem thông tin nội thất, thêm nội thất vào mục ưa thích.
* Thử nội thất bằng công nghệ AR.
* Chụp lại hình ảnh thử nội thất, lưu vào thiết bị, bộ sưu tập trên tài khoản người dùng hoặc chia sẻ trên facebook.
* Cập nhật profile người dùng.

### b) Chức năng của Firebase

* Quản lý người dùng (Authencation).
* Quản lý cơ sở dữ liệu (Realtime Database).
* Quản lý các tập tin như hình ảnh, models 3D phục vụ cho AR (Cloud Storage).

### **3.2 Cơ sở dữ liệu**

Cấu trúc Realtime Database:

Graphical user interface, text, application, email

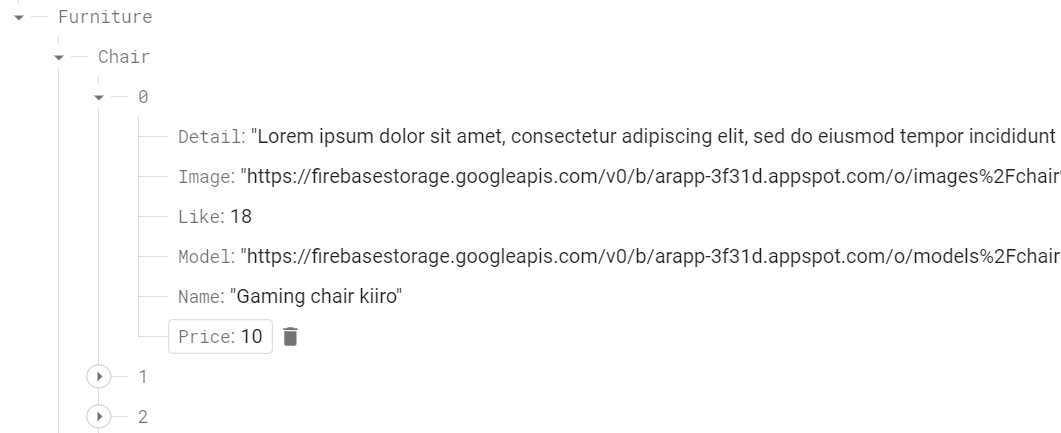
Description automatically generated

Cấu trúc nhánh Furniture:

Background pattern

Description automatically generated with low confidence

Trong nhánh Furniture bao gồm 5 nhánh nhỏ để phân loại nội thất.Trong mỗi nhánh nhỏ sẽ bao gồm danh sách nội thất.



Một đối tượng nội thất sẽ bao gồm các thông tin:

|  |  |
| --- | --- |
| Name | Tên nội thất |
| Detail | Mô tả nội thất |
| Price | Giá của nội thất |
| Like | Số lượt yêu thích của nội thất |
| Image | URL hình ảnh của nội thất |
| Model | URL model của nội thất |

Cấu trúc nhánh Collection:

Graphical user interface, text, application

Description automatically generated

Mỗi một đối tượng collection sẽ có key là uid của người dùng đó. Trong đó sẽ bao gồm danh sách các ảnh đã được người dùng lưu lại. Mỗi đối tượng ảnh sẽ bao gồm các thuộc tính:

|  |  |
| --- | --- |
| Link | URL của hình ảnh |
| isCover | Kiểm tra xem ảnh đó người dùng có đặt làm ảnh bìa hay không |

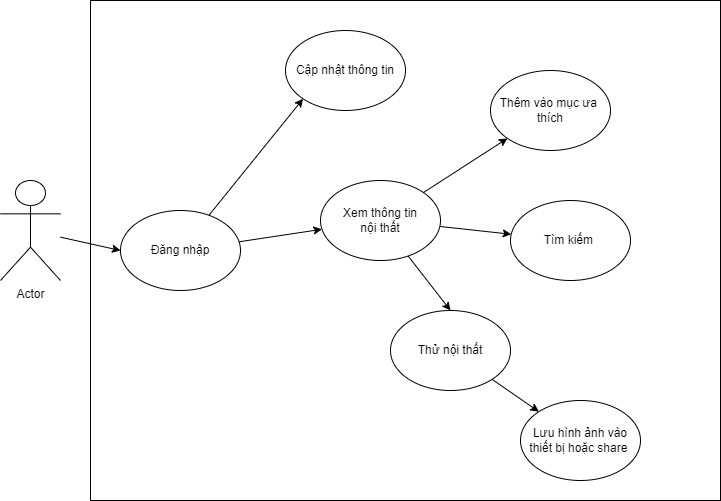
Cấu trúc nhánh Favorite:

Text

Description automatically generated

Tương tự như nhánh Collection, mỗi đối tượng Favorite sẽ có key là uid của người dùng. Trong đó sẽ bao gồm danh sách tên các nội thất đã được người dùng “thả tim”.

### **3.4 Sơ đồ User-case**

****

### **Chương 4: XÂY DỰNG ỨNG DỤNG**

### **4.1 Ứng dụng**

* Khách hàng mở ứng dụng và đăng nhập bằng số điện thoại.
* Giao diện thiết kế đơn giản dễ sử dụng phù hợp với độ tuổi người dùng(người trung niên hoặc cao tuổi)
* Kết nỗi dữ liệu bằng internet.
* Màn hình chờ ứng dụng bắt mắt.
* Icon của ứng Dụng :



### 4.1.1 Danh sách các màn hình.

*Bảng 4.1 Danh sách các màn hình client*

|  |  |  |
| --- | --- | --- |
| STT | Tên màn hình | Ý nghĩa |
| 1 | SplashActivity | Màn hình bắt đầu vào ứng dụng, hiển thị màn hình chờ |
| 2 | home | Bao gồm một cái ViewPager để chuyển các màn hình trong tarbar |
| 3 | HomeFragment | Để chưa một fragment chính là fragment home |
| 4 | ProfileFragment | Để chưa một fragment hiển thị thông tin người dùng |
| 5 | login | Màn hình đăng nhập |
| 6 | OTPVerifyActivity | Màn hình nhập mã xác nhận |
| 7 | product | Màn hình hiển thị chi tiết nội thất. |
| 8 | Favorite | Màn hình hiển thị sản phẩm yêu thích |
| 9 | CreateProfile | Màn hình cập nhật thông tin khi mới tạo tài khoản |
| 10 | MainActivity | Màn hình thử nội thất |

### 4.1.2 Màn hình khi mở ứng dụng

|  |  |
| --- | --- |
|  |  |

*Hình 4.1 Giao diện khi mở ứng dụng*

Khi lần đầu tiên người dùng nhấn vào biểu tượng của ứng dụng thì ứng dụng hiển thị màn hình đầu tiên đó chính là màn hình chờ bao gồm 1 logo và sau đó sẽ chuyển sang màn hình đăng nhập(nếu đã đăng nhập trước đó thì chuyển sang màn hình home).

* Code xử lý các sự kiện màn hình splash:

|  |
| --- |
| public class SplashScreen extends AppCompatActivity {   private static int *SPLASH\_SCREEN* = 3200;  ImageView imgLogo;  Animation topAnim;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  getWindow().setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*, WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);  setContentView(R.layout.*activity\_splash\_screen*);   topAnim = AnimationUtils.*loadAnimation*(this, R.anim.*top\_anim*);   imgLogo = findViewById(R.id.*img\_logo*);   imgLogo.setAnimation(topAnim);    new Handler().postDelayed(new Runnable() {  @Override  public void run() {  isLogin();  }  }, *SPLASH\_SCREEN*);  }   */\*\*  \* Hàm định nghĩa phương thức kiểm tra người dùng đã đăng nhập hay chưa  \* Nếu Đã đăng nhập thì sẽ chuển qua màn hình main  \* Nếu chưa đăng nhập sẽ chuyển qua màn hình login  \*/* private void isLogin() { // Lấy user hiện tại trên Firebase  FirebaseUser user = FirebaseAuth.*getInstance*().getCurrentUser();   Pair[] pair = new Pair[1];  pair[0] = new Pair<View, String>(imgLogo,"logo\_trans");   if(user == null){  Intent intent = new Intent(SplashScreen.this, login.class);   startActivity(intent);  finish();  }  else{  Intent intent = new Intent(SplashScreen.this, home.class);   startActivity(intent);  finish();  }  } } |

### 4.1.3 Màn hình OTPVerify

Graphical user interface, application

Description automatically generated

Nếu mã được gửi về số điện thoại người dùng thì ứng dụng sẽ tự chuyển qua màn hình OTPVerify. Người dùng nhập mã để đăng nhập.

public class OtpVerifyActivity extends AppCompatActivity {  
  
   
 private String verificationId;  
 TextView tvMobile;  
 TextView tvResendBtn;  
 Button btnVerify;  
 ProgressBar progressBarVerify;  
 EditText etC1, etC2, etC3, etC4, etC5, etC6;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_otp\_verify*);  
  
 tvMobile = findViewById(R.id.*tvMobile*);  
 tvResendBtn = findViewById(R.id.*tvResendBtn*);  
 btnVerify = findViewById(R.id.*btnVerify*);  
 progressBarVerify = findViewById(R.id.*progressBarVerify*);  
  
 etC1 = findViewById(R.id.*etC1*);  
 etC2 = findViewById(R.id.*etC2*);  
 etC3 = findViewById(R.id.*etC3*);  
 etC4 = findViewById(R.id.*etC4*);  
 etC5 = findViewById(R.id.*etC5*);  
 etC6 = findViewById(R.id.*etC6*);  
  
 editTextInput();  
  
 tvMobile.setText(String.*format*(  
 "+84-%s", getIntent().getStringExtra("phone")  
 ));  
  
 verificationId = getIntent().getStringExtra("verificationId");  
  
 tvResendBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Toast.*makeText*(OtpVerifyActivity.this, "OTP Send Successfully.", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 btnVerify.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 progressBarVerify.setVisibility(View.*VISIBLE*);  
 btnVerify.setVisibility(View.*INVISIBLE*);  
 if (etC1.getText().toString().trim().isEmpty() ||  
 etC2.getText().toString().trim().isEmpty() ||  
 etC3.getText().toString().trim().isEmpty() ||  
 etC4.getText().toString().trim().isEmpty() ||  
 etC5.getText().toString().trim().isEmpty() ||  
 etC6.getText().toString().trim().isEmpty()) {  
 Toast.*makeText*(OtpVerifyActivity.this, "OTP is not Valid!", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 if (verificationId != null) {  
 String code = etC1.getText().toString().trim() +  
 etC2.getText().toString().trim() +  
 etC3.getText().toString().trim() +  
 etC4.getText().toString().trim() +  
 etC5.getText().toString().trim() +  
 etC6.getText().toString().trim();  
  
 PhoneAuthCredential credential = PhoneAuthProvider.*getCredential*(verificationId, code);  
 FirebaseAuth  
 .*getInstance*()  
 .signInWithCredential(credential)  
 .addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull @NotNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 progressBarVerify.setVisibility(View.*VISIBLE*);  
 btnVerify.setVisibility(View.*INVISIBLE*);  
 Toast.*makeText*(OtpVerifyActivity.this, "Welcome...", Toast.*LENGTH\_SHORT*).show();  
 FirebaseUser user = FirebaseAuth.*getInstance*().getCurrentUser();  
 if (user != null) {  
 // Name, email address, and profile photo Url  
 String name = user.getDisplayName();  
 if(user.getDisplayName() == null) {  
 Intent intent = new Intent(OtpVerifyActivity.this, CreateProfile.class);  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);  
 }else{  
 Intent intent = new Intent(OtpVerifyActivity.this, home.class);  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);  
 }  
 }  
  
 } else {  
 progressBarVerify.setVisibility(View.*GONE*);  
 btnVerify.setVisibility(View.*VISIBLE*);  
 Toast.*makeText*(OtpVerifyActivity.this, "OTP is not Valid!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 }  
 }  
 });  
 }  
  
 private void editTextInput() {  
 etC1.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
  
 }  
  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
 etC2.requestFocus();  
 }  
  
 @Override  
 public void afterTextChanged(Editable s) {  
  
 }  
 });  
 etC2.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
  
 }  
  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
 etC3.requestFocus();  
 }  
  
 @Override  
 public void afterTextChanged(Editable s) {  
  
 }  
 });  
 etC3.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
  
 }  
  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
 etC4.requestFocus();  
 }  
  
 @Override  
 public void afterTextChanged(Editable s) {  
  
 }  
 });  
 etC4.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
  
 }  
  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
 etC5.requestFocus();  
 }  
  
 @Override  
 public void afterTextChanged(Editable s) {  
  
 }  
 });  
 etC5.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
  
 }  
  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
 etC6.requestFocus();  
 }  
  
 @Override  
 public void afterTextChanged(Editable s) {  
  
 }  
 });  
 }  
}

### 4.1.4 Màn hình CreateProfile

Graphical user interface, text, application, chat or text message

Description automatically generated

Nếu tài khoản người dùng mới tạo lần đầu thì sẽ được đưa đến màn hình này. Người dùng nhập tên và chọn ảnh đại diện để tiếp tục.

public class CreateProfile extends AppCompatActivity {  
  
 CircleImageView avatar;  
 String realpath = "";  
 Button btnSend;  
 EditText name;  
 FirebaseStorage storage;  
 StorageReference storageRef;  
 ProgressBar progressBar;  
 Uri avatarUri;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_create\_profile*);  
  
 avatar = findViewById(R.id.*avatar*);  
 name = findViewById(R.id.*name*);  
 btnSend = findViewById(R.id.*btnSend*);  
 progressBar = findViewById(R.id.*progressBar*);  
 storage = FirebaseStorage.*getInstance*();  
 storageRef = storage.getReference();  
  
 avatar.setOnClickListener(v ->{  
 ActivityCompat.*requestPermissions*(CreateProfile.this,  
 new String[]{Manifest.permission.*READ\_EXTERNAL\_STORAGE*},  
 0);  
 });  
 btnSend.setOnClickListener(v ->{  
 if(name.getText().toString().trim().isEmpty()){  
 name.setError("Please enter your name");  
 return;  
 }  
 progressBar.setVisibility(View.*VISIBLE*);  
 btnSend.setVisibility(View.*INVISIBLE*);  
 UploadAvatar();  
  
 });  
  
 }  
  
 private void CreateProfile(){  
 FirebaseUser user = FirebaseAuth.*getInstance*().getCurrentUser();  
  
 UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()  
 .setDisplayName(name.getText().toString())  
 .setPhotoUri(avatarUri)  
 .build();  
  
 user.updateProfile(profileUpdates)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if (task.isSuccessful()) {  
 Toast.*makeText*(CreateProfile.this, "Create profile success", Toast.*LENGTH\_SHORT*).show();  
 progressBar.setVisibility(View.*GONE*);  
 btnSend.setVisibility(View.*VISIBLE* );  
 startActivity(new Intent(CreateProfile.this, home.class));  
 }  
 }  
 });  
  
 }  
  
 private void UploadAvatar() {  
 if(realpath == ""){  
 avatarUri = Uri.*parse*("https://firebasestorage.googleapis.com/v0/b/arapp-3f31d.appspot.com/o/avatar%2Fimages.png?alt=media&token=a0a0f128-656a-4f32-af55-f969755e104e");  
 CreateProfile();  
 }else{  
 Uri file = Uri.*fromFile*(new File(realpath));  
 StorageReference riversRef = storageRef.child("avatar/"+file.getLastPathSegment());  
 UploadTask uploadTask = riversRef.putFile(file);  
  
// Register observers to listen for when the download is done or if it fails  
 uploadTask.addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception exception) {  
 Toast.*makeText*(CreateProfile.this, "Upload Failed: " + exception.getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 }).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  
 riversRef.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 avatarUri = uri;  
 CreateProfile();  
 }  
 });  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 if (requestCode == 0){  
 if (grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*){  
 Intent intent = new Intent(Intent.*ACTION\_PICK*);  
 intent.setType("image/\*");  
 someActivityResultLauncher.launch(intent);  
 }else{  
 Toast.*makeText*(this, "Permission Denied", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
  
 ActivityResultLauncher<Intent> someActivityResultLauncher = registerForActivityResult(  
 new ActivityResultContracts.StartActivityForResult(),  
 new ActivityResultCallback<ActivityResult>() {  
 @Override  
 public void onActivityResult(ActivityResult result) {  
 if (result.getResultCode() == Activity.*RESULT\_OK*) {  
 // There are no request codes  
 Intent data = result.getData();  
 Uri uri = data.getData();  
 realpath = getRealPathFromURI(uri);  
  
 try {  
 InputStream inputStream = getContentResolver().openInputStream(uri);  
 Bitmap bitmap = BitmapFactory.*decodeStream*(inputStream);  
 avatar.setImageBitmap(bitmap);  
 }catch (Exception e){  
 e.printStackTrace();  
 }  
  
 }  
 }  
 });  
  
 public String getRealPathFromURI (Uri contentUri) {  
 String path = null;  
 String[] proj = { MediaStore.MediaColumns.*DATA* };  
 Cursor cursor = getContentResolver().query(contentUri, proj, null, null, null);  
 if (cursor.moveToFirst()) {  
 int column\_index = cursor.getColumnIndexOrThrow(MediaStore.MediaColumns.*DATA*);  
 path = cursor.getString(column\_index);  
 }  
 cursor.close();  
 return path;  
 }  
}

### 4.1.5 Màn hình home và HomeFragment

Graphical user interface

Description automatically generated

Màn hình home bao gồm view pager để chuyển giữa 2 fragment home và profile.

Code xử lý màn hình home:

public class home extends AppCompatActivity{  
  
  
 BottomNavigationView bottomMenu;  
 ViewPager viewPager;  
 FloatingActionButton favorite;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*home*);  
  
 setUpViewPager();  
  
 bottomMenu = findViewById(R.id.*bottommenu*);  
 favorite = findViewById(R.id.*favorite*);  
  
 favorite.setOnClickListener(v -> {  
 Intent intent = new Intent(home.this, favorite.class);  
 intent.putExtra("category", "favorite");  
 startActivity(intent);  
 });  
  
 bottomMenu.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
 switch (item.getItemId()){  
 case R.id.*homemenu*:  
 viewPager.setCurrentItem(0);  
 break;  
 case R.id.*feedmenu*:  
 viewPager.setCurrentItem(1);  
 break;  
 default:  
 viewPager.setCurrentItem(1);  
 break;  
 }  
 return true;  
 }  
 });  
  
  
// LoadData();  
  
 }  
  
 private void setUpViewPager() {  
 viewPager = findViewById(R.id.*viewPager*);  
 ViewPagerAdapter viewPagerAdapter = new ViewPagerAdapter(getSupportFragmentManager(), FragmentStatePagerAdapter.*BEHAVIOR\_RESUME\_ONLY\_CURRENT\_FRAGMENT*);  
 viewPager.setAdapter(viewPagerAdapter);  
  
 viewPager.setOnPageChangeListener(new ViewPager.OnPageChangeListener() {  
 @Override  
 public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {  
 }  
  
 @Override  
 public void onPageSelected(int position) {  
 switch (position){  
 case 0:  
 bottomMenu.setSelectedItemId(R.id.*homemenu*);  
 break;  
 case 1:  
 bottomMenu.setSelectedItemId(R.id.*feedmenu*);  
 break;  
 default:  
 bottomMenu.setSelectedItemId(R.id.*homemenu*);  
 break;  
 }  
 }  
  
 @Override  
 public void onPageScrollStateChanged(int state) {  
 }  
 });  
 }  
  
}

Code xử lý FragmentHome:

public class HomeFragment extends Fragment implements OnClickFurniture {  
  
 FirebaseStorage storage;  
 StorageReference storageRef;  
 String mUri;  
 RecyclerView hotProduct,newProduct;  
 FurnitureHomeAdapter adapter,adapter2;  
 DatabaseReference databaseReference;  
 FirebaseAuth mAuth;  
 List<Furniture> furnitureList;  
 List<Furniture> furnitureList2;  
 View view;  
 ImageView loading1,loading2;  
 CardView plant,lamp,chair,table,other;  
 Furniture f;  
  
 @Nullable  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 view = inflater.inflate(R.layout.*fragment\_home*, container, false);  
 return view;  
 }  
  
 @Override  
 public void onStart() {  
 super.onStart();  
 init();  
 LoadData("Table");  
 LoadData("Chair");  
 LoadData("Lamp");  
 LoadData("Plant");  
 LoadData2("Other");  
 }  
  
 private void init() {  
 storage = FirebaseStorage.*getInstance*();  
 storageRef = storage.getReference();  
 hotProduct = view.findViewById(R.id.*HotProduct*);  
 newProduct = view.findViewById(R.id.*NewProduct*);  
 loading1 = view.findViewById(R.id.*loading1*);  
 loading2 = view.findViewById(R.id.*loading2*);  
 plant = view.findViewById(R.id.*plant\_category*);  
 lamp = view.findViewById(R.id.*lamp\_category*);  
 chair = view.findViewById(R.id.*chair\_category*);  
 table = view.findViewById(R.id.*table\_category*);  
 other = view.findViewById(R.id.*other\_category*);  
 Glide.*with*(getContext()).load("https://upload.wikimedia.org/wikipedia/commons/c/c7/Loading\_2.gif").into(loading1);  
 Glide.*with*(getContext()).load("https://upload.wikimedia.org/wikipedia/commons/c/c7/Loading\_2.gif").into(loading2);  
 plant.setOnClickListener(v->{  
 Intent intent = new Intent(getContext(),favorite.class);  
 intent.putExtra("category","Plant");  
 startActivity(intent);  
 });  
 lamp.setOnClickListener(v->{  
 Intent intent = new Intent(getContext(),favorite.class);  
 intent.putExtra("category","Lamp");  
 startActivity(intent);  
 });  
 chair.setOnClickListener(v->{  
 Intent intent = new Intent(getContext(),favorite.class);  
 intent.putExtra("category","Chair");  
 startActivity(intent);  
 });  
 table.setOnClickListener(v->{  
 Intent intent = new Intent(getContext(),favorite.class);  
 intent.putExtra("category","Table");  
 startActivity(intent);  
 });  
 other.setOnClickListener(v->{  
 Intent intent = new Intent(getContext(),favorite.class);  
 intent.putExtra("category","Other");  
 startActivity(intent);  
 });  
 }  
  
 private void LoadData(String category) {  
 int maxLike = 0;  
 f = new Furniture();  
 furnitureList = new ArrayList<Furniture>();  
 adapter = new FurnitureHomeAdapter(furnitureList,this);  
 hotProduct.setAdapter(adapter);  
 databaseReference = FirebaseDatabase.*getInstance*().getReference("Furniture").child(category);  
 databaseReference.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 for (DataSnapshot data : snapshot.getChildren()) {  
 Furniture furniture = data.getValue(Furniture.class);  
 if(furniture.getLike() > maxLike){  
 f = furniture;  
 }  
 }  
 furnitureList.add(f);  
 adapter.notifyDataSetChanged();  
 loading1.setVisibility(View.*GONE*);  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
 loading1.setImageResource(R.drawable.*error*);  
  
 }  
  
 });  
 }  
  
 private void LoadData2(String category) {  
 furnitureList2 = new ArrayList<Furniture>();  
 adapter2 = new FurnitureHomeAdapter(furnitureList2,this);  
 newProduct.setAdapter(adapter2);  
  
 databaseReference = FirebaseDatabase.*getInstance*().getReference("Furniture").child(category);  
 databaseReference.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 for (DataSnapshot data : snapshot.getChildren()) {  
 Furniture furniture = data.getValue(Furniture.class);  
 furnitureList2.add(furniture);  
 }  
  
 adapter2.notifyDataSetChanged();  
  
 loading2.setVisibility(View.*GONE*);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 loading2.setImageResource(R.drawable.*error*);  
 }  
  
 });  
 }  
  
  
 @Override  
 public void onFurnitureClicked(Furniture furniture) {  
 Intent intent = new Intent(getContext(), product.class);  
 intent.putExtra("Furniture", furniture);  
 startActivity(intent);  
 }  
}

### 4.1.6 Màn hình Profile

A picture containing graphical user interface

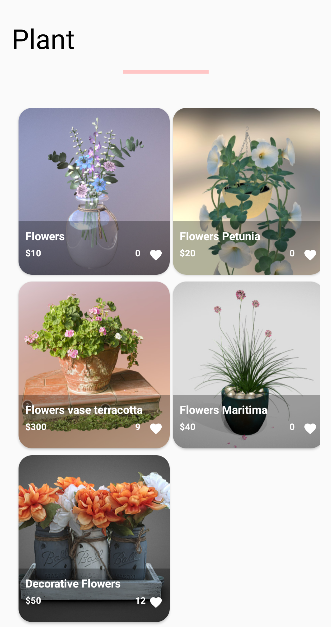
Description automatically generated

Màn hình bao gồm thông tin người dùng và bộ sưu tập ảnh của người dùng.

Code xử lý:

public class ProfileFragment extends Fragment {  
 ImageView imageProfile,cover;  
 TextView name, phone;  
 ImageView exit;  
 DatabaseReference databaseReference;  
 RecyclerView recyclerView;  
 CoverAdapter coverAdapter;  
 List<Cover> coverList;  
 Dialog dialog;  
 Map<String,Cover> coverMap;  
 View slide1,slide2;  
 LinearLayout cl,infor,lycl,lyinfor;  
 TextView txtcl,txtinfor;  
 EditText edtname;  
 Button btnrename;  
  
 @Nullable  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 return inflater.inflate(R.layout.*fragment\_profile*, container, false);  
 }  
  
 @Override  
 public void onStart() {  
 name = getView().findViewById(R.id.*name*);  
 phone = getView().findViewById(R.id.*phone*);  
 cover = getView().findViewById(R.id.*cover*);  
 imageProfile = getView().findViewById(R.id.*profile\_image*);  
 recyclerView = getView().findViewById(R.id.*recycler\_view\_collection*);  
 coverList = new ArrayList<Cover>();  
 coverMap = new HashMap<String,Cover>();  
 coverAdapter = new CoverAdapter(coverList, new OnClickCover() {  
 @Override  
 public void onClickCover(Cover cover, View view) {  
 showImage(cover);  
 }  
  
 @Override  
 public void onLongClickCover(Cover cover, View view) {  
 ShowPopUpMenu(cover,view);  
 }  
 });  
 recyclerView.setAdapter(coverAdapter);  
  
  
 FirebaseUser user = FirebaseAuth.*getInstance*().getCurrentUser();  
 if (user != null) {  
 Glide.*with*(this).load(user.getPhotoUrl().toString()).into(imageProfile);  
 name.setText(user.getDisplayName());  
 phone.setText(user.getPhoneNumber());  
 }  
 databaseReference = FirebaseDatabase.*getInstance*().getReference("Collection").child(user.getUid());  
 databaseReference.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 coverMap.clear();  
 coverList.clear();  
 for (DataSnapshot dataSnapshot: snapshot.getChildren()  
 ) {  
 Cover coverImage = dataSnapshot.getValue(Cover.class);  
 coverMap.put(dataSnapshot.getKey(),coverImage);  
  
 if (coverImage != null && coverImage.isIscover() == true) {  
 Glide.*with*(ProfileFragment.this).load(coverImage.getLink()).into(cover);  
 }  
 coverList.add(coverImage);  
 coverAdapter.notifyDataSetChanged();  
  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 imageProfile.setOnClickListener(v->{  
  
 });  
  
 cl = getView().findViewById(R.id.*cl*);  
 infor = getView().findViewById(R.id.*infor*);  
 lycl = getView().findViewById(R.id.*lycl*);  
 lyinfor = getView().findViewById(R.id.*lyinfor*);  
 slide1 = getView().findViewById(R.id.*slide1*);  
 slide2 = getView().findViewById(R.id.*slide2*);  
 txtcl = getView().findViewById(R.id.*txtcl*);  
 txtinfor = getView().findViewById(R.id.*txtinfor*);  
 edtname = getView().findViewById(R.id.*edtname*);  
 btnrename = getView().findViewById(R.id.*btnrename*);  
 exit = getView().findViewById(R.id.*exit*);  
 exit.setOnClickListener(v->{  
 FirebaseAuth.*getInstance*().signOut();  
 Intent intent = new Intent(getActivity(), login.class);  
 startActivity(intent);  
 });  
  
 cl.setOnClickListener(v->{  
 lycl.setVisibility(View.*VISIBLE*);  
 lyinfor.setVisibility(View.*GONE*);  
 slide1.setVisibility(View.*VISIBLE*);  
 slide2.setVisibility(View.*GONE*);  
 txtcl.setTextColor(Color.*parseColor*("#000000"));  
 txtinfor.setTextColor(Color.*parseColor*("#b3b3b3"));  
 });  
 infor.setOnClickListener(v->{  
 lycl.setVisibility(View.*GONE*);  
 lyinfor.setVisibility(View.*VISIBLE*);  
 slide1.setVisibility(View.*GONE*);  
 slide2.setVisibility(View.*VISIBLE*);  
 txtcl.setTextColor(Color.*parseColor*("#b3b3b3"));  
 txtinfor.setTextColor(Color.*parseColor*("#000000"));  
  
 });  
 edtname.setText(user.getDisplayName());  
 btnrename.setOnClickListener(v->{  
 UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()  
 .setDisplayName(edtname.getText().toString())  
 .build();  
  
 user.updateProfile(profileUpdates)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if (task.isSuccessful()) {  
 Toast.*makeText*(getContext(), "Rename Success", Toast.*LENGTH\_SHORT*).show();  
 name.setText(user.getDisplayName());  
 }  
 }  
 });  
 });  
  
  
 super.onStart();  
 }  
  
 private void ShowPopUpMenu(Cover cover,View view) {  
 PopupMenu popupMenu = new PopupMenu(getContext(),view);  
 popupMenu.getMenuInflater().inflate(R.menu.*collectionmenu*,popupMenu.getMenu());  
 popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {  
 @Override  
 public boolean onMenuItemClick(MenuItem menuItem) {  
 switch (menuItem.getItemId()){  
 case R.id.*menucover*:  
 {  
 for (Map.Entry<String,Cover> pair: coverMap.entrySet()  
 ) {  
 if(pair.getValue().isIscover() == true){  
 databaseReference.child(pair.getKey()).child("iscover").setValue(false);  
 }  
 if (pair.getValue().getLink() == cover.getLink()){  
 databaseReference.child(pair.getKey()).child("iscover").setValue(true);  
 }  
 Toast.*makeText*(getContext(), "Update Cover Success", Toast.*LENGTH\_SHORT*).show();  
 }  
 break;  
 }  
 case R.id.*menudel*:  
 {  
 for (Map.Entry<String,Cover> pair: coverMap.entrySet()  
 ) {  
 if (pair.getValue().getLink() == cover.getLink()){  
 databaseReference.child(pair.getKey()).removeValue();  
 Toast.*makeText*(getContext(), "Delete Success", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
 break;  
 }  
 case R.id.*menuavatar*:  
 {  
 FirebaseUser user = FirebaseAuth.*getInstance*().getCurrentUser();  
  
 UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()  
 .setPhotoUri(Uri.*parse*(cover.getLink()))  
 .build();  
  
 user.updateProfile(profileUpdates)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if (task.isSuccessful()) {  
 Toast.*makeText*(getContext(), "Update avatar success", Toast.*LENGTH\_SHORT*).show();  
 Glide.*with*(ProfileFragment.this).load(user.getPhotoUrl().toString()).into(imageProfile);  
 }  
 }  
 });  
 break;  
 }  
 }  
 return false;  
 }  
 });  
 popupMenu.show();  
  
 }  
  
 public void showImage(Cover cover) {  
 CreateDialog(R.layout.*dialog\_view*);  
 dialog.show();  
 ImageView imageView = dialog.findViewById(R.id.*imageView*);  
 Glide.*with*(this).load(cover.getLink()).into(imageView);  
 }  
  
 public void CreateDialog(int layout){  
 dialog = new Dialog(getContext());  
 dialog.requestWindowFeature(Window.*FEATURE\_NO\_TITLE*);  
 dialog.setContentView(layout);  
  
 final Window window = dialog.getWindow();  
 window.setLayout(AbsListView.LayoutParams.*MATCH\_PARENT*, AbsListView.LayoutParams.*WRAP\_CONTENT*);  
 dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*TRANSPARENT*));  
  
 WindowManager.LayoutParams wlp = window.getAttributes();  
 wlp.gravity = Gravity.*CENTER*;  
 window.addFlags(WindowManager.LayoutParams.*FLAG\_DIM\_BEHIND*);  
 window.setDimAmount(0.5f);  
 window.setAttributes(wlp);  
  
 ((ViewGroup)dialog.getWindow().getDecorView())  
 .getChildAt(0).startAnimation(AnimationUtils.*loadAnimation*(  
 getContext(),android.R.anim.*slide\_in\_left*));  
  
 }  
  
}

### 4.1.7 Màn hình Favorite



Màn hình này bao gồm danh sách nội thất theo loại hoặc danh sách nội thất yêu thích của người dùng.

public class favorite extends AppCompatActivity implements OnClickFurniture {  
  
 ImageView loading\_favorite;  
 RecyclerView favorite\_recycler;  
 TextView txt\_favorite;  
 List<Furniture> furnitureList;  
 List<Furniture> furnitureList2;  
 FurnitureHomeAdapter adapter;  
 DatabaseReference databaseReference;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*favourite*);  
 getWindow().getDecorView().setSystemUiVisibility( View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* | View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN*);  
  
 Init();  
 Intent intent = getIntent();  
 String category = intent.getStringExtra("category");  
 Furniture furniture = (Furniture) intent.getSerializableExtra("furniture");  
 if(!category.equals(null)){  
 txt\_favorite.setText(category);  
 LoadData(category);  
 }  
 if (category.equals("favorite")){  
 txt\_favorite.setText("Favorite");  
 LoadData("Table");  
 LoadData("Chair");  
 LoadData("Lamp");  
 LoadData("Plant");  
 LoadData("Other");  
 LoadFavorite();  
 }  
  
  
  
 }  
  
 private void LoadFavorite() {  
 furnitureList2 = new ArrayList<Furniture>();  
 adapter = new FurnitureHomeAdapter(furnitureList2,this);  
 favorite\_recycler.setAdapter(adapter);  
 databaseReference = FirebaseDatabase.*getInstance*().getReference("Favorite").child(FirebaseAuth.*getInstance*().getCurrentUser().getUid());  
 databaseReference.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 for (DataSnapshot data : snapshot.getChildren()) {  
 String name = (String) data.getValue();  
 for (Furniture f: furnitureList  
 ) {  
 if (f.getName().equals(name)){  
 furnitureList2.add(f);  
 }  
 }  
 }  
 adapter.notifyDataSetChanged();  
 loading\_favorite.setVisibility(View.*GONE*);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
 loading\_favorite.setImageResource(R.drawable.*error*);  
 }  
  
 });  
 }  
  
 private void LoadData(String category) {  
 furnitureList = new ArrayList<Furniture>();  
 adapter = new FurnitureHomeAdapter(furnitureList,this);  
 favorite\_recycler.setAdapter(adapter);  
 databaseReference = FirebaseDatabase.*getInstance*().getReference("Furniture").child(category);  
 databaseReference.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 for (DataSnapshot data : snapshot.getChildren()) {  
 Furniture furniture = data.getValue(Furniture.class);  
 furnitureList.add(furniture);  
 }  
 adapter.notifyDataSetChanged();  
 loading\_favorite.setVisibility(View.*GONE*);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
 loading\_favorite.setImageResource(R.drawable.*error*);  
 }  
  
 });  
 }  
  
 private void Init() {  
 loading\_favorite = findViewById(R.id.*loading\_favorite*);  
 favorite\_recycler = findViewById(R.id.*rv\_favorite*);  
 txt\_favorite = findViewById(R.id.*txt\_favorite*);  
 Glide.*with*(this).load("https://upload.wikimedia.org/wikipedia/commons/c/c7/Loading\_2.gif").into(loading\_favorite);  
 }  
  
 @Override  
 public void onFurnitureClicked(Furniture furniture) {  
 Intent intent = new Intent(getApplicationContext(), product.class);  
 intent.putExtra("Furniture", furniture);  
 startActivity(intent);  
 }  
}

### 4.1.8 Màn hình product

Graphical user interface, website

Description automatically generated

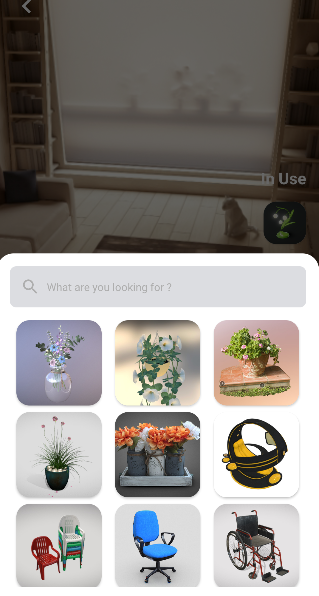
Màn hình hiển thị chi tiết sản phẩm

public class product extends AppCompatActivity {  
  
 ImageView image\_product;  
 TextView name\_product,detail\_product,price\_product;  
 Button btnViewAR;  
 ImageView back;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*product*);  
 getWindow().getDecorView().setSystemUiVisibility( View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* | View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN*);  
  
 Init();  
 Intent intent = getIntent();  
 Furniture furniture = (Furniture) intent.getSerializableExtra("Furniture");  
 Glide.*with*(this).load(furniture.getImage()).into(image\_product);  
 name\_product.setText(furniture.getName());  
 detail\_product.setText(furniture.getDetail());  
 price\_product.setText("$"+furniture.getPrice());  
 btnViewAR.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(product.this,MainActivity.class);  
 intent.putExtra("Furniture",furniture);  
 startActivity(intent);  
 }  
 });  
 }  
  
 private void Init() {  
 image\_product = findViewById(R.id.*image\_product*);  
 name\_product = findViewById(R.id.*name\_product*);  
 detail\_product = findViewById(R.id.*detail\_product*);  
 price\_product = findViewById(R.id.*price\_product*);  
 btnViewAR = findViewById(R.id.*btnViewAR*);  
 back = findViewById(R.id.*back1*);  
  
 back.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Toast.*makeText*(product.this, "wi", Toast.*LENGTH\_SHORT*).show();  
 finish();  
 }  
 });  
 }  
  
  
}

### 4.1.9 Màn hình Main



Màn hình cho phép người dùng thử nội thất, bao gồm nút chụp hình, thanh dialog và danh sách các model đang được sử dụng.



Dialog tìm kiếm cho phép người dùng thêm model.

A picture containing text, indoor, screenshot

Description automatically generated

Dialog cho phép người dùng lưu ảnh hoặc share lên fb.

Code xử lý

public class MainActivity extends AppCompatActivity implements  
 FragmentOnAttachListener,  
 BaseArFragment.OnTapArPlaneListener,  
 BaseArFragment.OnSessionConfigurationListener,  
 ArFragment.OnViewCreatedListener,  
 OnClickFurniture {  
  
 private ArFragment arFragment;  
 private Renderable model;  
 private ViewRenderable viewRenderable;  
 CardView cardView;  
 String name;  
 private boolean isRecording = false;  
 private HandlerThread videoProcessingThread;  
 private Handler videoProcessingHandler;  
 private int messages = 60;  
 List<Furniture> furnitureList;  
 List<Furniture> furnitureListUse;  
 DatabaseReference databaseReference;  
 SearchAdapter searchAdapter;  
 UseAdapter furnitureUseAdapter;  
 BottomSheetDialog bottomSheetDialog;  
 RecyclerView recyclerViewUse;  
 ImageView capture,back;  
 Bitmap bitmap;  
 FirebaseStorage storage;  
 StorageReference storageRef;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 setContentView(R.layout.*activity\_main*);  
 getSupportFragmentManager().addFragmentOnAttachListener(this);  
  
  
 if (savedInstanceState == null) {  
 if (Sceneform.*isSupported*(this)) {  
 getSupportFragmentManager().beginTransaction()  
 .add(R.id.*arFragment*, ArFragment.class, null)  
 .commit();  
 }  
 }  
  
 storage = FirebaseStorage.*getInstance*();  
 storageRef = storage.getReference();  
 cardView = findViewById(R.id.*cardView*);  
 furnitureList = new ArrayList<Furniture>();  
 furnitureListUse = new ArrayList<Furniture>();  
 recyclerViewUse = findViewById(R.id.*recyclerViewUse*);  
 capture = findViewById(R.id.*capture*);  
 back = findViewById(R.id.*back2*);  
 back.setOnClickListener(v ->{  
 finishActivity(0);  
 });  
  
 searchAdapter = new SearchAdapter(furnitureList, new OnClickFurniture() {  
 @Override  
 public void onFurnitureClicked(Furniture furniture) {  
 for (Furniture furniture1: furnitureListUse  
 ) {  
 if (furniture1.getName().equals(furniture.getName())){  
 Toast.*makeText*(MainActivity.this, "Furniture in use", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 }  
  
 loadModels(furniture.getModel());  
 bottomSheetDialog.dismiss();  
 furnitureListUse.add(furniture);  
 furnitureUseAdapter.notifyDataSetChanged();  
  
  
 }  
 });  
 furnitureUseAdapter = new UseAdapter(furnitureListUse, new OnClickFurniture() {  
 @Override  
 public void onFurnitureClicked(Furniture furniture) {  
 loadModels(furniture.getModel());  
 }  
 });  
 recyclerViewUse.setAdapter(furnitureUseAdapter);  
 LoadData("Plant");  
 LoadData("Chair");  
 LoadData("Lamp");  
 LoadData("Table");  
 LoadData("Other");  
  
  
 cardView.setOnClickListener(v -> {  
  
 OpenBottomDialog(R.layout.*bottom\_dialog\_layout*, R.id.*container*);  
  
 });  
  
 capture.setOnClickListener(v->{  
 Bitmap.Config conf = Bitmap.Config.*ARGB\_8888*; // see other conf types  
 bitmap = Bitmap.*createBitmap*(arFragment.getArSceneView().getWidth(),  
 arFragment.getArSceneView().getHeight(), conf);  
 PixelCopy.*request*(arFragment.getArSceneView(), bitmap, new PixelCopy.OnPixelCopyFinishedListener() {  
 @Override  
 public void onPixelCopyFinished(int i) {  
 OpenBottomDialog(R.layout.*activity\_share*, R.id.*sharecontainer*);  
 }  
 }, new Handler());  
  
  
 });  
  
  
 Intent intent = getIntent();  
 Furniture furniture = (Furniture) intent.getSerializableExtra("Furniture");  
 name = furniture.getName();  
 loadModels(furniture.getModel());  
 furnitureListUse.add(furniture);  
 furnitureUseAdapter.notifyDataSetChanged();  
  
 }  
  
  
  
 private void LoadData(String category) {  
 databaseReference = FirebaseDatabase.*getInstance*().getReference("Furniture").child(category);  
 databaseReference.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 for (DataSnapshot data : snapshot.getChildren()) {  
 Furniture furniture = data.getValue(Furniture.class);  
 furnitureList.add(furniture);  
 }  
 searchAdapter.notifyDataSetChanged();  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
  
 });  
 }  
  
  
 void OpenBottomDialog(int layout, int container) {  
 bottomSheetDialog = new BottomSheetDialog(MainActivity.this, R.style.*BottomSheetDialogTheme*);  
 View bottomSheetView = LayoutInflater.*from*(getApplicationContext()).inflate(layout, (ConstraintLayout) findViewById(container));  
 bottomSheetDialog.setContentView(bottomSheetView);  
 if(layout == R.layout.*bottom\_dialog\_layout*){  
 RecyclerView recyclerView = bottomSheetView.findViewById(R.id.*recycler\_view*);  
 recyclerView.setAdapter(searchAdapter);  
 EditText search = bottomSheetView.findViewById(R.id.*search*);  
 search.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
  
 }  
  
 @Override  
 public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
  
 }  
  
 @Override  
 public void afterTextChanged(Editable editable) {  
 filter(editable.toString());  
 }  
 });  
 }else{  
 ImageView imageView = bottomSheetView.findViewById(R.id.*imageView*);  
 imageView.setImageBitmap(bitmap);  
 Button button = bottomSheetView.findViewById(R.id.*buttonSave1*);  
 Button button1 = bottomSheetView.findViewById(R.id.*buttonSave2*);  
 Button button2 = bottomSheetView.findViewById(R.id.*buttonShare*);  
 button.setOnClickListener(v->{  
 getImageFromBitmapAndSave();  
 });  
 button1.setOnClickListener(v->{  
 saveToCollection();  
 });  
 button2.setOnClickListener(v->{  
 shareWithFB(bitmap);  
 });  
 }  
  
 bottomSheetDialog.show();  
 }  
  
 private void saveToCollection() {  
 ByteArrayOutputStream baos = new ByteArrayOutputStream();  
 bitmap.compress(Bitmap.CompressFormat.*JPEG*, 100, baos);  
 byte[] data = baos.toByteArray();  
 storageRef = storageRef.child("collection/"+System.*currentTimeMillis*()+".jpg");  
  
 UploadTask uploadTask = storageRef.putBytes(data);  
 uploadTask.addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception exception) {  
 // Handle unsuccessful uploads  
 }  
 }).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  
 storageRef.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {  
 @Override  
 public void onSuccess(Uri uri) {  
 databaseReference = FirebaseDatabase.*getInstance*().getReference("Collection");  
 databaseReference.child(FirebaseAuth.*getInstance*().getCurrentUser().getUid()).child(System.*currentTimeMillis*()+"").setValue(new Cover(uri.toString(),false));  
 Toast.*makeText*(MainActivity.this, "Saved to collection", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
 });  
 }  
  
 void getImageFromBitmapAndSave(){  
 try {  
 File file = new File(Environment.*getExternalStorageDirectory*() + "/Pictures/" + System.*currentTimeMillis*() + ".jpg");  
 FileOutputStream fOut = new FileOutputStream(file);  
 bitmap.compress(Bitmap.CompressFormat.*JPEG*, 100, fOut);  
 fOut.flush();  
 fOut.close();  
 Toast.*makeText*(MainActivity.this, "Screenshot saved", Toast.*LENGTH\_SHORT*).show();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 void shareWithFB(Bitmap bitmap){  
 SharePhoto sharePhoto1 = new SharePhoto.Builder()  
 .setBitmap(bitmap)  
 .build();  
  
  
 ShareContent shareContent = new ShareMediaContent.Builder()  
 .addMedium(sharePhoto1)  
 .build();  
  
 ShareDialog shareDialog = new ShareDialog(this);  
 shareDialog.show(shareContent, ShareDialog.Mode.*AUTOMATIC*);  
 }  
  
 void filter(String text) {  
 List<Furniture> filteredList = new ArrayList<>();  
 for (Furniture item : furnitureList) {  
 if (item.getName().toLowerCase().contains(text.toLowerCase())) {  
 filteredList.add(item);  
 }  
 }  
 searchAdapter.filterList(filteredList);  
 }  
  
  
  
 @Override  
 public void onAttachFragment(@NonNull FragmentManager fragmentManager, @NonNull Fragment fragment) {  
 if (fragment.getId() == R.id.*arFragment*) {  
 arFragment = (ArFragment) fragment;  
 arFragment.setOnSessionConfigurationListener(this);  
 arFragment.setOnViewCreatedListener(this);  
 arFragment.setOnTapArPlaneListener(this);  
 }  
 }  
  
 @Override  
 public void onSessionConfiguration(Session session, Config config) {  
 if (session.isDepthModeSupported(Config.DepthMode.*AUTOMATIC*)) {  
 config.setDepthMode(Config.DepthMode.*AUTOMATIC*);  
 }  
 }  
  
 @Override  
 public void onViewCreated(ArSceneView arSceneView) {  
 arFragment.setOnViewCreatedListener(null);  
 // Fine adjust the maximum frame rate  
 arSceneView.setFrameRateFactor(SceneView.FrameRate.*FULL*);  
 }  
  
 public void loadModels(String uri) {  
 WeakReference<MainActivity> weakActivity = new WeakReference<>(this);  
 ModelRenderable.*builder*()  
 .setSource(this, Uri.*parse*(uri))  
 .setIsFilamentGltf(true)  
 .setAsyncLoadEnabled(false)  
 .build()  
 .thenAccept(model -> {  
 MainActivity activity = weakActivity.get();  
 if (activity != null) {  
 activity.model = model;  
 }  
 })  
 .exceptionally(throwable -> {  
 Toast.*makeText*(  
 this, "Unable to load model", Toast.*LENGTH\_LONG*).show();  
 return null;  
 });  
  
  
 ViewRenderable.*builder*()  
 .setView(this, R.layout.*view\_model\_title*)  
 .build()  
 .thenAccept(viewRenderable -> {  
 MainActivity activity = weakActivity.get();  
 if (activity != null) {  
 activity.viewRenderable = viewRenderable;  
 }  
 })  
 .exceptionally(throwable -> {  
 Toast.*makeText*(this, "Unable to load model", Toast.*LENGTH\_LONG*).show();  
 return null;  
 });  
 }  
  
 @Override  
 public void onTapPlane(HitResult hitResult, Plane plane, MotionEvent motionEvent) {  
 if (model == null || viewRenderable == null) {  
 Toast.*makeText*(this, "Loading...", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 // Create the Anchor.  
 Anchor anchor = hitResult.createAnchor();  
 AnchorNode anchorNode = new AnchorNode(anchor);  
 anchorNode.setParent(arFragment.getArSceneView().getScene());  
  
 // Create the transformable model and add it to the anchor.  
 TransformableNode model = new TransformableNode(arFragment.getTransformationSystem());  
 model.setParent(anchorNode);  
 model.setRenderable(this.model)  
 .animate(true).start();  
 model.select();  
  
  
 Node titleNode = new Node();  
 titleNode.setParent(model);  
 titleNode.setEnabled(false);  
 titleNode.setLocalPosition(new Vector3(0.0f, 1.0f, 0.0f));  
 titleNode.setRenderable(viewRenderable);  
 titleNode.setEnabled(true);  
  
 titleNode.setOnTouchListener(new Node.OnTouchListener() {  
 @Override  
 public boolean onTouch(HitTestResult hitTestResult, MotionEvent motionEvent) {  
 model.setRenderable(null);  
 model.removeChild(titleNode);  
 return false;  
 }  
 });  
  
 }  
  
 @Override  
 public void onFurnitureClicked(Furniture furniture) {  
  
 }  
}

### 4.1.10 File Build gradle

plugins **{** id 'com.android.application'  
 id 'com.google.gms.google-services'  
  
**}**android **{** compileSdk 33  
  
 signingConfigs **{** myconfig **{** storeFile file('C:\\Workspace\\arsha.jks')  
 storePassword '123456'  
 keyAlias 'arsha'  
 keyPassword '123456'  
 **}  
 }** defaultConfig **{** applicationId "com.google.ar.sceneform.samples.gltf"  
  
 // Sceneform requires minSdk >= 24.  
 minSdk rootProject.ext.minSdk  
 targetSdk 33  
 versionCode 1  
 versionName "1.0"  
 **}** compileOptions **{** sourceCompatibility JavaVersion.*VERSION\_1\_8* targetCompatibility JavaVersion.*VERSION\_1\_8* **}** aaptOptions **{** noCompress 'filamat', 'ktx'  
 **}** buildTypes **{** release **{** signingConfig signingConfigs.myconfig  
 **}  
 }  
}**dependencies **{** implementation "androidx.appcompat:appcompat:$appcompat\_version"  
 implementation "androidx.fragment:fragment:$fragment\_version"  
  
 implementation 'com.google.android.material:material:1.7.0'  
 implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
 implementation 'com.google.firebase:firebase-database:20.1.0'  
 implementation 'com.google.firebase:firebase-auth'  
 implementation 'com.google.firebase:firebase-storage'  
 implementation platform('com.google.firebase:firebase-bom:31.1.1')  
 testImplementation 'junit:junit:4.13.2'  
 androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
 androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
 implementation 'de.hdodenhof:circleimageview:3.0.0'  
 implementation 'androidx.browser:browser:1.2.0'  
 implementation 'com.squareup.picasso:picasso:2.8'  
 implementation 'com.firebaseui:firebase-ui-storage:7.2.0'  
  
 implementation 'com.intuit.sdp:sdp-android:1.0.6'  
 implementation 'com.intuit.ssp:ssp-android:1.0.6'  
  
 implementation "org.jcodec:jcodec:0.2.2"  
 implementation "org.jcodec:jcodec-android:0.2.2"  
 implementation 'com.facebook.android:facebook-android-sdk:latest.release'  
  
 releaseImplementation "com.gorisse.thomas.sceneform:sceneform:1.21.0"  
 debugApi project(":sceneform")  
  
  
  
  
**}**

### 4.1.11 File Android Manifest

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="com.google.ar.sceneform.samples.gltf">  
 <!-- Always needed for AR. -->  
 <uses-permission android:name="android.permission.CAMERA" /> <!-- Needed to load gltf from network. -->  
 <uses-permission android:name="android.permission.INTERNET" />  
 <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />  
 <uses-permission  
 android:name="com.google.android.gms.permission.AD\_ID"  
 tools:node="remove" /> <!-- Sceneform requires OpenGLES 3.0 or later. -->  
 <uses-feature  
 android:glEsVersion="0x00030000"  
 android:required="true" />  
 <!--  
 Indicates that this app requires Google Play Services for AR ("AR Required") and results in  
 the app only being visible in the Google Play Store on devices that support ARCore.  
 For an "AR Optional" app, remove this tag.  
 -->  
 <uses-feature  
 android:name="android.hardware.camera.ar"  
 android:required="true" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@drawable/ic\_launcher"  
 android:label="@string/app\_name"  
 android:requestLegacyExternalStorage="true"  
 android:theme="@style/AppTheme"  
 android:supportsRtl="true"  
 android:usesCleartextTraffic="true">  
 <activity  
 android:name=".ShareActivity"  
 android:exported="false">  
 <meta-data  
 android:name="android.app.lib\_name"  
 android:value="" />  
 </activity>  
 <activity  
 android:name=".CreateProfile"  
 android:exported="true">  
 <meta-data  
 android:name="android.app.lib\_name"  
 android:value="" />  
 </activity>  
  
 <meta-data  
 android:name="android.app.lib\_name"  
 android:value="" />  
 <!--  
 Indicates that this app requires Google Play Services for AR ("AR Required") and causes  
 the Google Play Store to download and intall Google Play Services for AR along with the app.  
 For an "AR Optional" app, specify "optional" instead of "required".  
 -->  
 <meta-data  
 android:name="com.google.ar.core"  
 android:value="required" />  
  
 <activity  
 android:name=".MainActivity"  
 android:configChanges="orientation|screenSize"  
 android:exported="false"  
 android:noHistory="false"  
 android:label="@string/app\_name"  
 android:theme="@style/AppTheme"></activity>  
 <activity  
 android:name=".product"  
 android:exported="false"  
 android:label=""  
 android:noHistory="false"  
 android:screenOrientation="portrait"  
 android:theme="@style/MyAppTheme"  
 android:windowSoftInputMode="stateHidden|adjustPan" />  
 <activity  
 android:name=".favorite"  
 android:exported="false"  
 android:label=""  
 android:screenOrientation="portrait"  
 android:theme="@style/AppTheme.NoActionBar"  
 android:windowSoftInputMode="stateHidden|adjustPan" />  
 <activity  
 android:name=".profile"  
 android:exported="false"  
 android:label=""  
 android:screenOrientation="portrait"  
 android:theme="@style/AppTheme.NoActionBar"  
 android:windowSoftInputMode="stateHidden|adjustPan" />  
 <activity  
 android:name=".home"  
 android:exported="false"  
 android:label=""  
 android:noHistory="false"  
 android:screenOrientation="portrait"  
 android:theme="@style/AppTheme.NoActionBar"  
 android:windowSoftInputMode="stateHidden|adjustPan" />  
 <activity  
 android:name=".OtpVerifyActivity"  
 android:exported="false"  
 android:label=""  
 android:screenOrientation="portrait"  
 android:theme="@style/AppTheme.NoActionBar"  
 android:windowSoftInputMode="stateHidden|adjustPan" />  
 <activity  
 android:name=".SplashScreen"  
 android:exported="true"  
 android:label=""  
 android:screenOrientation="portrait"  
 android:theme="@style/AppTheme.NoActionBar"  
 android:windowSoftInputMode="stateHidden|adjustPan">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <activity  
 android:name=".login"  
 android:exported="false"  
 android:label=""  
 android:screenOrientation="portrait"  
 android:theme="@style/MyAppTheme"  
 android:windowSoftInputMode="stateHidden|adjustPan" />  
  
 <provider  
 android:name="com.facebook.FacebookContentProvider"  
 android:authorities="com.facebook.app.FacebookContentProvider5852391044772958"  
 android:exported="true" />  
  
 <meta-data  
 android:name="com.facebook.sdk.ApplicationId"  
 android:value="@string/facebook\_app\_id" />  
 <meta-data  
 android:name="com.facebook.sdk.ClientToken"  
 android:value="@string/facebook\_client\_token" />  
 </application>  
  
</manifest>

### **Chương III: TỔNG KẾT**

Nhờ sự chỉ dẫn của giảng viên và qua tìm hiểu cũng như nỗ lực nghiên cứu và thực hiện thì em đã xây dựng thành công ứng dụng trang trí nội thất trên nền tảng Android. Mong là ứng dụng sẽ được ứng dủng rộng rãi và phát triển hơn trong tương lai và em sẽ cố gắng khắc phục sai sót và thêm nhiều tính năng mới nữa. Em xin cám ơn và gửi lời chào đến quý giảng viên cùng thầy cô và các bạn đã xem bài báo cáo.