

Clover - Ứng dụng mua sắm thời trang

Nhóm 09

Thành viên

- 20424008 - Dương Mạnh Cường (nhóm trưởng)
- 19424015 - Dương Trọng Đức
- 20424013 - Phạm Nguyễn Mỹ Diễm



*Đại học Khoa Học Tự Nhiên
ĐHQG Thành phố Hồ Chí Minh*

Nội dung

- ① Giới thiệu về ứng dụng Clover
- ② Các package + dịch vụ bên thứ ba
 - Các package
 - Các dịch vụ bên thứ ba
- ③ Các màn hình và chức năng kèm theo
- ④ Chạy thử ứng dụng
- ⑤ Tài nguyên và tham khảo
- ⑥ Hỏi đáp và kết thúc

Giới thiệu về ứng dụng Clover



Hình 1: Màn hình chính của ứng dụng sau khi đăng nhập

- Clover **không phải** là một e-commerce mobile application.
- Clover là một **fashion shopping application** phục vụ cho các nhãn hàng thời trang mà họ không muốn kinh doanh sản phẩm của mình trên các trang thương mại điện tử.
- Các fashion shopping application khác mà Clover có tính năng tương tự: [CoupleTX](#), [Mr.Simple](#), [Peek & Cloppenburg](#), [Farfetch](#),...
- Hạn chế của ứng dụng đến hiện tại:
 - Chưa có server.
 - Chưa có các web service.
 - Một vài chức năng còn sơ sài và chưa khớp với thực tế.
 - ...

Các package + dịch vụ bên thứ ba (1)

Các package

- **RoundedImageView** - Tạo ra các ImageView bo tròn ở các góc, import vào gradle:

```
implementation 'com.makeramen:roundedimageview:2.3.0'
```

- **Glide** - Load ảnh từ internet vào ImageView, import vào gradle:

```
implementation 'com.github.bumptech.glide:glide:4.12.0'
```

```
annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'
```

- Và một vài package khác.

Các package + dịch vụ bên thứ ba (2)

Các dịch vụ bên thứ ba

- **Firebase Firestore** - CSDL NoSQL do Google phát triển.
- **Firebase FireStorage** - Dùng để lưu trữ tệp tin, thư mục trên cloud, nhóm không tìm thấy document do Google công bố cho dịch vụ này, nhưng để sử dụng thì nhóm tìm hiểu trong cuốn **Mastering Firebase for Android Development**.
- **Firebase Authentication** - Dịch vụ xác nhận danh tính, đăng ký tài khoản người dùng do Google phát triển.

Các màn hình và chức năng kèm theo (1)

Nội dung

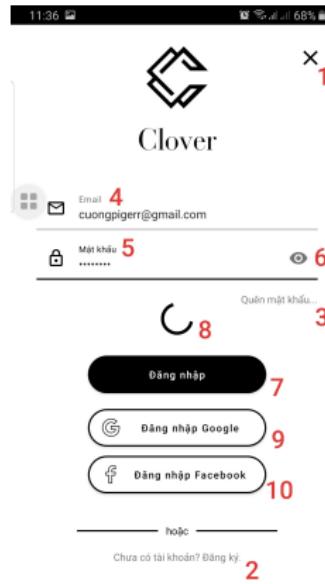
Các màn hình chính và chức năng quan trọng được trình bày:

- ① Màn hình đăng nhập.
- ② Màn hình chi tiết sản phẩm.
- ③ Màn hình chính.
- ④ Chức năng tìm kiếm.

Ngoài ra còn có các màn hình khác như **Giỏ hàng**, **Lịch sử mua hàng**, **Tài khoản người dùng**,... nhưng đơn giản và đa phần được kế thừa từ các màn hình trên nên nhóm xin rút gọn phần trình bày cho chúng.

Các màn hình và chức năng kèm theo (2)

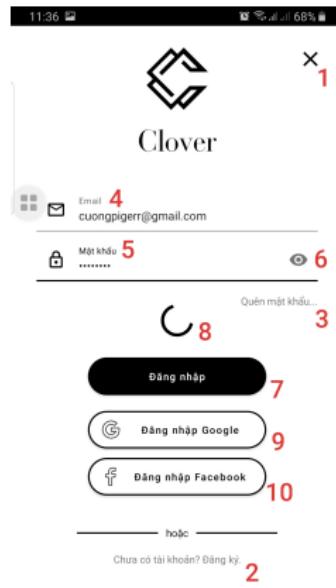
Màn hình đăng nhập



Các thành phần và chức năng kèm theo trên màn hình:

- 1: Image button - khi nhấn vào button này người dùng sẽ được chuyển thẳng đến màn hình chính của ứng dụng mà **không cần đăng nhập**.
- 2: Text button - khi nhấn vào sẽ đưa đến fragment đăng ký tài khoản mới cho người dùng.
- 3: Text button - khi nhấn vào đưa người dùng đến fragment đặt lại mật khẩu.
- 4: Text input layout - nhập thông tin email của người dùng.

Hình 2: Màn hình đăng nhập



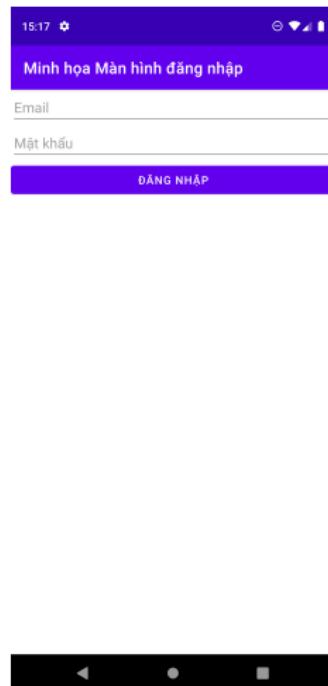
Các thành phần và chức năng kèm theo trên màn hình:

- **5:** Text input layout - nhập mật khẩu của người dùng.
- **6:** Icon - khi nhấn vào sẽ hiển thị mật khẩu ở **5** ra dưới dạng text.
- **7:** Button - nhấn vào để đăng nhập.
- **8:** Progress Circle - luôn ẩn đi, chỉ hiển thị khi người dùng nhấn vào **7** để cho người dùng biết rằng hệ thống vẫn chạy và đang trong quá trình kiểm tra đăng nhập.

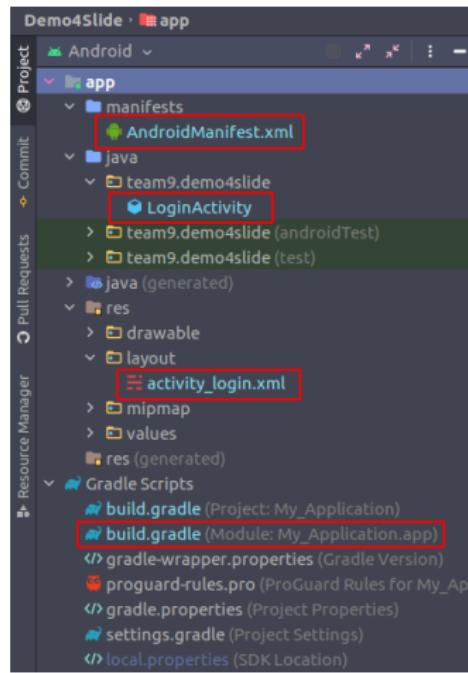
Hình 2: Màn hình đăng nhập

Các màn hình và chức năng kèm theo (3)

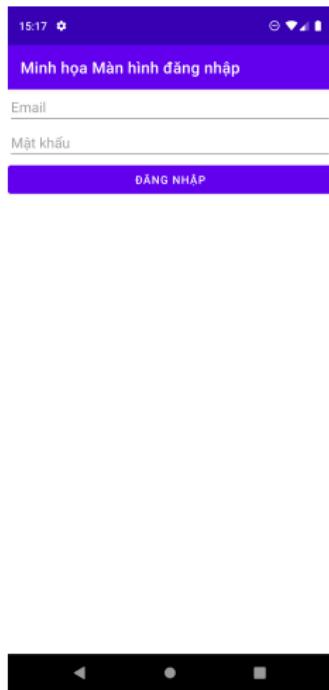
Màn hình đăng nhập - Mã nguồn minh họa



Cấu trúc cây thư mục:



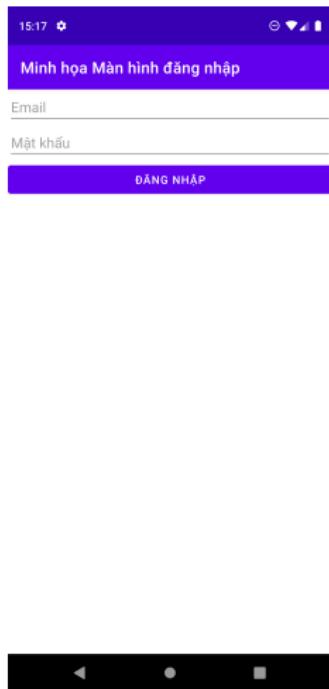
Hình 3: Minh họa màn hình đăng nhập



AndroidManifest.xml

```
AndroidManifest.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="team9.demo4slide">
4          <uses-permission android:name="android.permission.INTERNET" />
5
6          <application
7              android:allowBackup="true"
8              android:icon="@mipmap/ic_launcher"
9              android:label="Minh họa Màn hình đăng nhập"
10             android:roundIcon="@mipmap/ic_launcher_round"
11             android:supportsRtl="true"
12             android:theme="@style/Theme.MyApplication">
13                 <activity
14                     android:name=".LoginActivity"
15                     android:exported="true">
16                     <intent-filter>
17                         <action android:name="android.intent.action.MAIN" />
18
19                         <category android:name="android.intent.category.LAUNCHER" />
20                     </intent-filter>
21                 </activity>
22             </application>
23         </manifest>
```

Hình 3: Minh họa màn hình đăng nhập



build.gradle

A screenshot of the build.gradle file in an Android Studio project. The code is as follows:

```
30     }
31     compileSdkVersion 31
32     buildToolsVersion '30.0.2'
33 }
34
35 dependencies {
36
37     implementation 'androidx.appcompat:appcompat:1.4.0'
38     implementation 'com.google.android.material:material:1.4.0'
39     implementation 'androidx.constraintlayout:constraintlayout:2.1.2'
40     implementation 'com.google.firebaseio:firebase-analytics:20.0.2'
41     implementation 'com.google.firebaseio:firebase-auth:21.0.1' // Line 41 highlighted with a red box
42     testImplementation 'junit:junit:4.+'
43     androidTestImplementation 'androidx.test.ext:junit:1.1.3'
44     androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
45 }
```

Hình 3: Minh họa màn hình đăng nhập

activity_login.xml

Minh họa Màn hình đăng nhập

Email

Mật khẩu

DĂNG NHẬP

```
<?xml version="1.0" encoding="Utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".LoginActivity">

    <EditText
        android:id="@+id/etEmail"
        android:hint="Email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

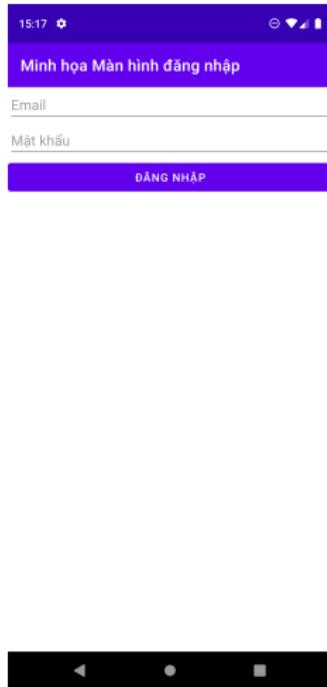
    <EditText
        android:id="@+id/etPassword"
        android:hint="Mật khẩu"
        android:inputType="textPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/bSubmit"
        android:text="Đăng nhập"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

Hình 3: Minh họa màn hình đăng nhập

LoginActivity.java

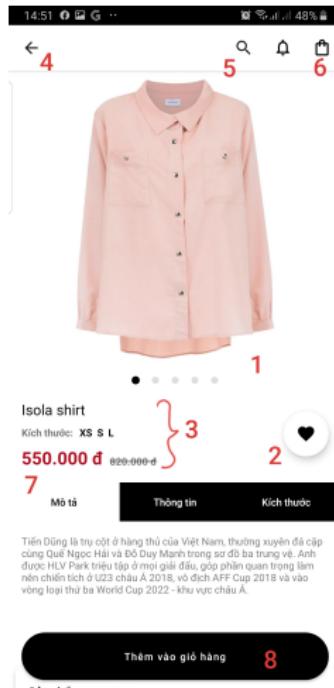


```
1  package team9.demo4slide;
2
3  import androidx.appcompat.app.AppCompatActivity;
4  import android.os.Bundle;
5  import android.widget.Button;
6  import android.widget.EditText;
7  import com.google.firebase.auth.FirebaseAuth;
8
9  public class LoginActivity extends AppCompatActivity {
10     EditText mEmail, mPassword;
11     Button mSubmit;
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_login);
17
18         mEmail = findViewById(R.id.etEmail); mPassword = findViewById(R.id.etPassword);
19         mSubmit = findViewById(R.id.bSubmit);
20         mSubmit.setOnClickListener(e -> {
21             FirebaseAuth.getInstance().signInWithEmailAndPassword(
22                 mEmail.getText().toString(), mPassword.getText().toString()
23             ).addOnCompleteListener(task -> {
24                 if (task.isSuccessful()) {
25                     // đăng nhập thành công, do something
26                 } else {
27                     // đăng nhập thất bại, do something
28                 }
29             });
30         });
31     }
32 }
```

Hình 3: Minh họa màn hình đăng nhập

Các màn hình và chức năng kèm theo (4)

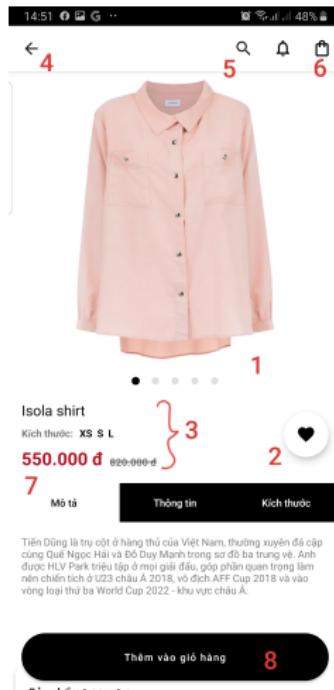
Màn hình chi tiết sản phẩm



Các thành phần và chức năng kèm theo trên màn hình:

- 1: View Pager - hiển thị các hình ảnh của sản phẩm.
- 2: Floating Action Button - cho phép người dùng nhấn vào để thêm sản phẩm vào mục yêu thích.
- 3: Các text view - hiển thị thông tin của sản phẩm.
- 4: As home indicator - quay về fragment trước đó được lưu trong backstack.

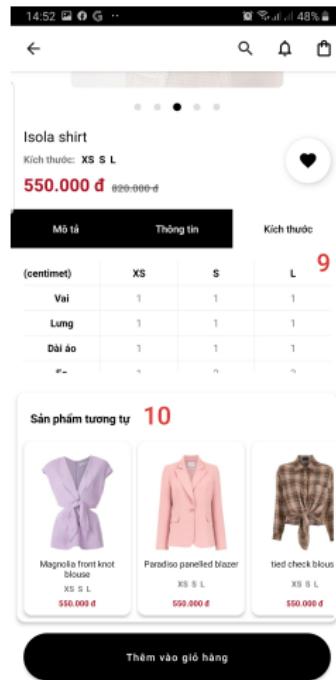
Hình 4: Màn hình chi tiết sản phẩm 1



Các thành phần và chức năng kèm theo trên màn hình:

- 5: Menu item Search - nhấn vào để thực hiện chức năng tìm kiếm.
- 6: Menu item Cart - nhấn vào để đến fragment giỏ hàng.
- 7: TabLayout + View Pager 2 - bên trong tab "Mô tả" là một Text View.
- 8: Button - nhấn vào sẽ hiện một Dialog để người dùng có thể chọn size và số lượng sản phẩm để thêm vào giỏ hàng.

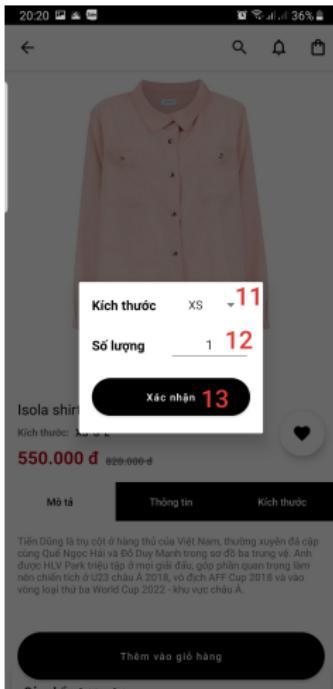
Hình 4: Màn hình chi tiết sản phẩm 1



Các thành phần và chức năng kèm theo trên màn hình:

- **9:** TabLayout + View Pager 2 - bên trong tab "Thông tin" và "Kích thước" là các GridView.
- **10:** Horizontal Recycler View - hiển thị tám sản phẩm nằm cùng danh mục với sản phẩm hiện tại.

Hình 5: Màn hình chi tiết sản phẩm 2



Các thành phần và chức năng kèm theo trên màn hình:

- **11:** Spinner - cho phép người dùng chọn các size của sản phẩm.
- **12:** Text input - để người dùng nhập số lượng sản phẩm cần mua cho size tương ứng.
- **13:** Button - khi click vào sản phẩm sẽ được thêm vào giỏ hàng cùng size và số lượng đã chọn.

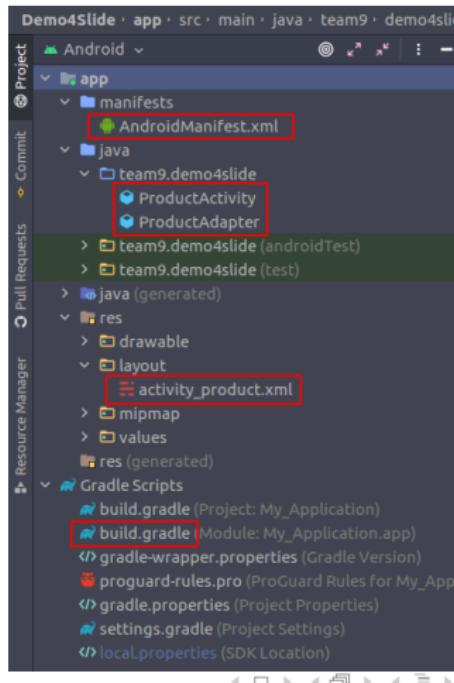
Hình 6: Màn hình chi tiết sản phẩm 3

Các màn hình và chức năng kèm theo (5)

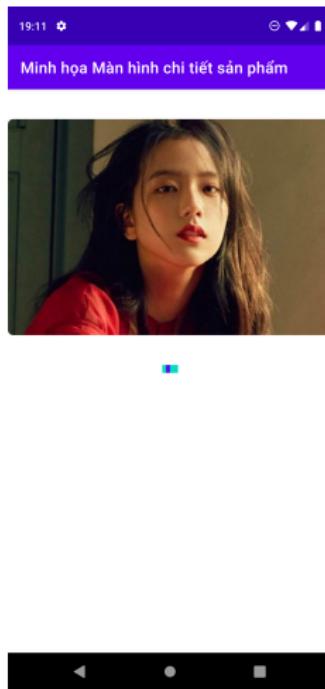
Màn hình chi tiết sản phẩm - Mã nguồn minh họa



Cấu trúc cây thư mục:



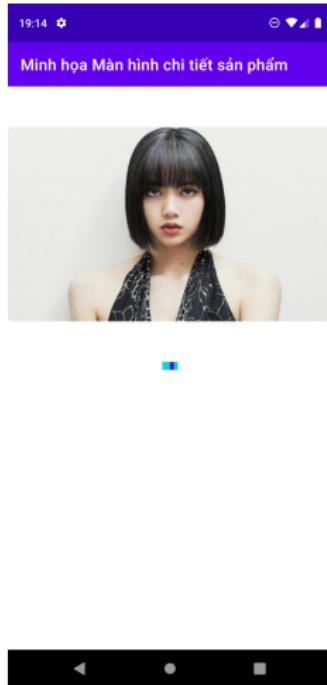
Hình 7: Minh họa màn hình chi tiết sản phẩm 1



AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="team9.demo4slide">
4          <uses-permission android:name="android.permission.INTERNET" />
5
6      <application
7          android:allowBackup="true"
8          android:icon="@mipmap/ic_launcher"
9          android:label="Minh họa Màn hình chi tiết sản phẩm"
10         android:roundIcon="@mipmap/ic_launcher_round"
11         android:supportsRtl="true"
12         android:theme="@style/Theme.MyApplication">
13             <activity
14                 android:name=".ProductActivity"
15                 android:exported="true">
16                 <intent-filter>
17                     <action android:name="android.intent.action.MAIN" />
18
19                     <category android:name="android.intent.category.LAUNCHER" />
20                 </intent-filter>
21             </activity>
22         </application>
23     </manifest>
```

Hình 8: Minh họa màn hình chi tiết sản phẩm 2



build.gradle

```
build.gradle (:app) x
You can use the Project Structure dialog to view and edit your project configuration

33    }
34
35    dependencies {
36
37        implementation 'androidx.appcompat:appcompat:1.4.0'
38        implementation 'com.google.android.material:material:1.4.0'
39        implementation 'androidx.constraintlayout:constraintlayout:2.1.2'
40        testImplementation 'junit:junit:4.+'
41        androidTestImplementation 'androidx.test.ext:junit:1.1.3'
42        androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
43        implementation 'com.makeramen:roundedimageview:2.3.0' // Line 43
44        implementation 'com.github.bumptech.glide:glide:4.12.0' // Line 44
45        annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0' // Line 45
46    }
```

Hình 9: Minh họa màn hình chi tiết sản phẩm 3



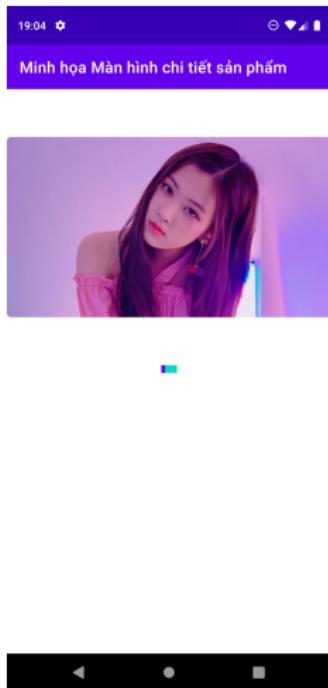
activity_product.xml

activity_product.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      xmlns:app="http://schemas.android.com/apk/res-auto"
7      android:orientation="vertical"
8      tools:context=".ProductActivity">
9
10 <androidx.viewpager.widget.ViewPager
11     android:id="@+id/vpContainer" ←
12     android:layout_width="match_parent"
13     android:layout_height="350dp"/>
14
15 <com.google.android.material.tabs.TabLayout
16     android:id="@+id/tlIndicator" ←
17     android:layout_width="20dp"
18     android:layout_height="10dp"
19     android:layout_gravity="center"
20     app:tabBackground="@color/teal_200"
21     app:tabIndicatorHeight="10dp" />
22 </LinearLayout>
```

Hình 10: Minh họa màn hình chi tiết sản phẩm 4

ProductAdapter.java



Hình 7: Minh họa màn hình chi tiết sản phẩm 1

```
ProductAdapter.java ×
1 package team9.demo4slide;
2
3 import android.view.View;
4 import android.view.ViewGroup;
5 import androidx.annotation.NonNull;
6 import androidx.viewpager.widget.PagerAdapter;
7 import com.bumptech.glide.Glide;
8 import com.makeramen.roundedimageview.RoundedImageView;
9 import java.util.List;
10
11 public class ProductAdapter extends PagerAdapter {
12     List<String> images;
13     public ProductAdapter(List<String> images) { this.images = images; }
14
15     @NonNull
16     @Override
17     public Object instantiateItem(@NonNull ViewGroup viewGroup, int position) {
18         RoundedImageView image = new RoundedImageView(viewGroup.getContext());
19         image.setCornerRadius((float) 30);
20         Glide.with(viewGroup.getContext()).load(images.get(position)).into(image);
21         viewGroup.addView(image, index: 0);
22         return image;
23     }
24
25     @Override
26     public int getCount() { return images.size(); }
27
28     @Override
29     public boolean isViewFromObject(@NonNull View view, @NonNull Object object) {
30         return view == object;
31     }
32
33     @Override
34     public void destroyItem(@NonNull ViewGroup container, int position, @NonNull Object object) {
35         container.removeView((RoundedImageView) object);
36     }
37 }
```

ProductActivity.java



Hình 7: Minh họa màn hình chi tiết sản phẩm 1

```
1 package team9.demo4slide;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.viewpager.widget.ViewPager;
5 import android.os.Bundle;
6 import com.google.android.material.tabs.TabLayout;
7 import java.util.ArrayList;
8 import java.util.List;
9
10 public class ProductActivity extends AppCompatActivity {
11     ViewPager mContainer;
12     TabLayout mIndicator;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_product);
18         mContainer = findViewById(R.id.vpContainer);
19         mIndicator = findViewById(R.id.tlIndicator);
20
21         List<String> images = new ArrayList<>();
22         images.add("https://channel-korea.com/wp-content/uploads/2018/02/00180668.jpg");
23         images.add("https://media-cdn.laodong.vn/storage/newsportal/2021/3/29" +
24             "/893926/Blackpink.jpg?w=414&h=276&crop=auto&scale=both");
25         images.add("https://vcdn-vnexpress.vnecdn.net/2021/09/24/lisa-6874-1632475092.jpg");
26         images.add("https://media-cdn.laodong.vn/storage/newsportal" +
27             "/2020/10/2/841202/20200617123919-E0c6.jpg");
28
29         mContainer.setAdapter(new ProductAdapter(images));
30         mIndicator.setupWithViewPager(mContainer, autoRefresh: true);
31     }
32 }
```

Các màn hình và chức năng kèm theo (6)

Màn hình chính



Các thành phần và chức năng kèm theo trên màn hình:

- 1: Action bar - chứa các thành phần từ trái qua phải như sau:
 - Hamburger toggle - nhấn vào để mở navigation view.
 - Actionbar logo - hiển thị logo của ứng dụng.
 - Menu item Search - chức năng tìm kiếm.
 - Menu item Notification - chức năng thông báo (chưa hỗ trợ do ứng dụng chưa có web service + server xử lý).
 - Menu item Cart - bấm vào để đi đến fragment giỏ hàng.

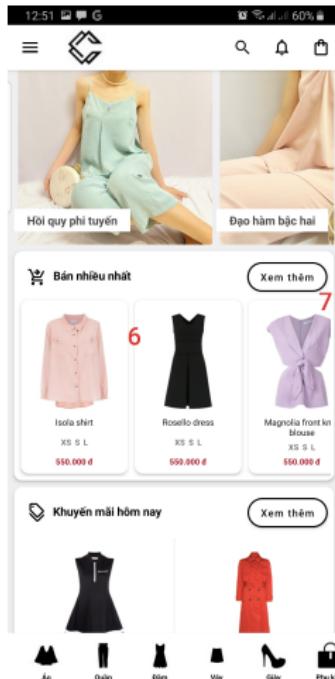
Hình 11: Màn hình chính của ứng dụng 1



Các thành phần và chức năng kèm theo trên màn hình:

- 2: Horizontal Recycler View - hiển thị các sản phẩm theo danh mục.
- 3: Carousel (View Pager) - hiển thị các chương trình, khuyến mãi của cửa hàng. Sau ba giây sẽ tự chuyển.
- 4: Banner (Image View) - dùng để chèn quảng cáo.
- 5: Slider (Horizontal Recycler View) - hiển thị các sản phẩm mới nhất.

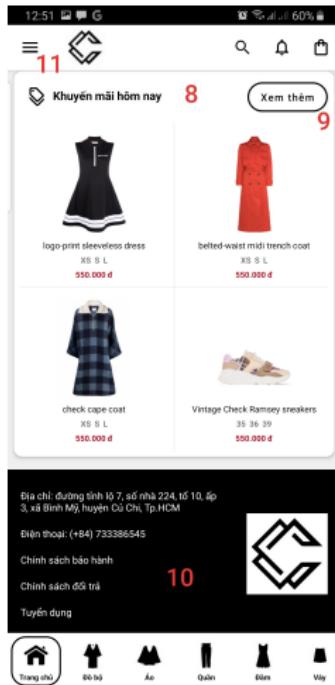
Hình 11: Màn hình chính của ứng dụng 1



Các thành phần và chức năng kèm theo trên màn hình:

- 6: Horizontal Recycler View - hiển thị tám các sản phẩm bán chạy hiện tại. Nhấn vào item sẽ đưa đến trang chi tiết sản phẩm tương ứng.
- 7: Button - đưa đến một fragment khác hiển thị nhiều hơn tám sản phẩm bán chạy dưới dạng GridView.

Hình 12: Màn hình chính của ứng dụng 2



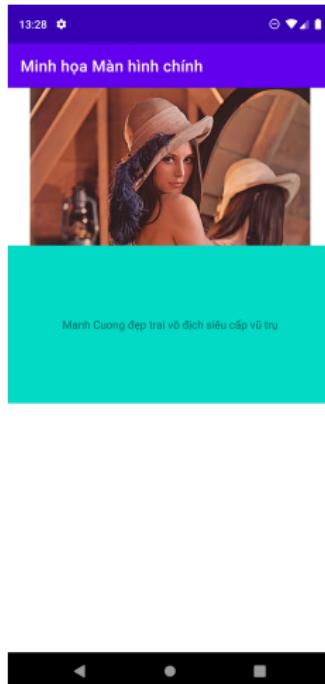
Các thành phần và chức năng kèm theo trên màn hình:

- **8:** GridView - hiển thị bốn sản phẩm được khuyến mãi hôm nay. Nhấn vào item sẽ đưa đến trang chi tiết sản phẩm tương ứng.
- **9:** Button - nhấn vào sẽ đưa đến một fragment khác hiển thị nhiều sản phẩm khuyến mãi hơn dưới dạng GridView.
- **10:** Footer - thể hiện địa chỉ, số điện thoại, các chính sách đổi trả, bảo hành của nhãn hàng.
- **11:** Hamburger toggle - nhấn vào để mở navigation view.

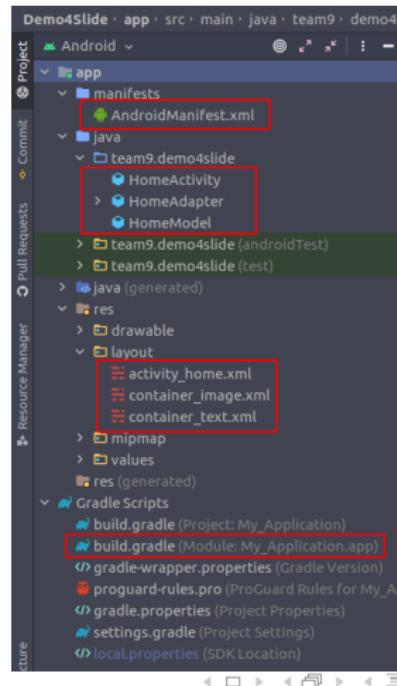
Hình 13: Màn hình chính của ứng dụng 3

Các màn hình và chức năng kèm theo (7)

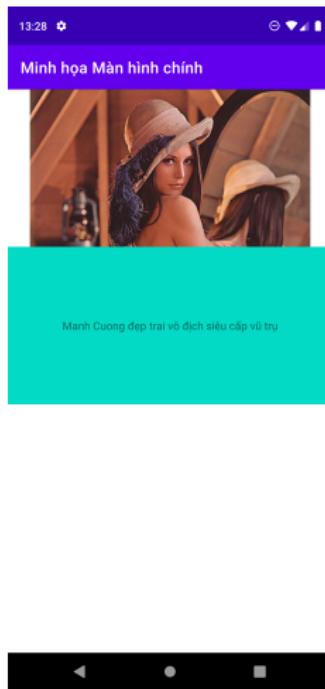
Màn hình chính - Mã nguồn minh họa



Cấu trúc cây thư mục:



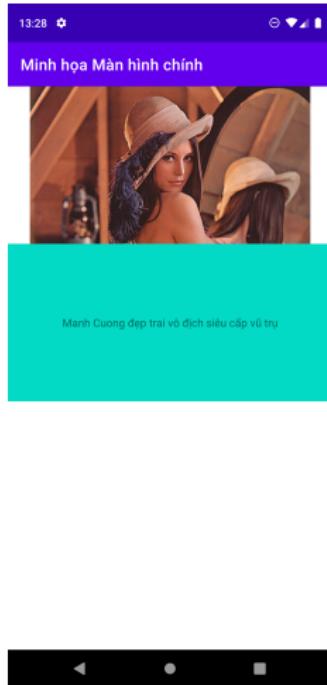
Hình 14: Minh họa màn hình chính
của ứng dụng



AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="team9.demo4slide">
4          <uses-permission android:name="android.permission.INTERNET" />
5
6      <application
7          android:allowBackup="true"
8          android:icon="@mipmap/ic_launcher"
9          android:label="Minh họa Màn hình chính"
10         android:roundIcon="@mipmap/ic_launcher_round"
11         android:supportsRtl="true"
12         android:theme="@style/Theme.MyApplication">
13         <activity
14             android:name=".HomeActivity"
15             android:exported="true">
16             <intent-filter>
17                 <action android:name="android.intent.action.MAIN" />
18
19                 <category android:name="android.intent.category.LAUNCHER" />
20             </intent-filter>
21         </activity>
22     </application>
23 </manifest>
```

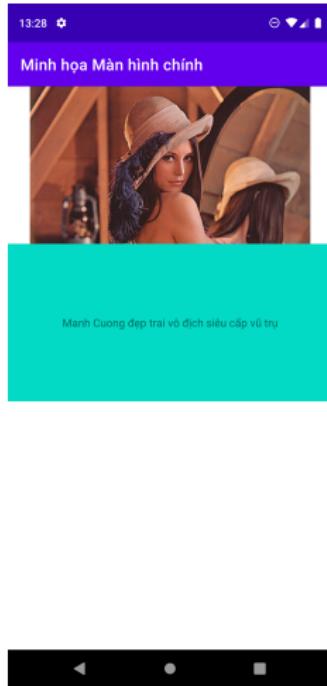
Hình 14: Minh họa màn hình chính
của ứng dụng



build.gradle

```
build.gradle (:app) ×
You can use the Project Structure dialog to view and edit your project configuration
26     targetCompatibility JavaVersion.VERSION_1_8
27 }
28 }
29
30 dependencies {
31
32     implementation 'androidx.appcompat:appcompat:1.4.0'
33     implementation 'com.google.android.material:material:1.4.0'
34     implementation 'androidx.constraintlayout:constraintlayout:2.1.2'
35     testImplementation 'junit:junit:4.+'
36     androidTestImplementation 'androidx.test.ext:junit:1.1.3'
37     androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
38     implementation 'com.github.bumptech.glide:glide:4.12.0' // This line is highlighted by a red box
39     annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'
40 }
```

Hình 14: Minh họa màn hình chính
của ứng dụng

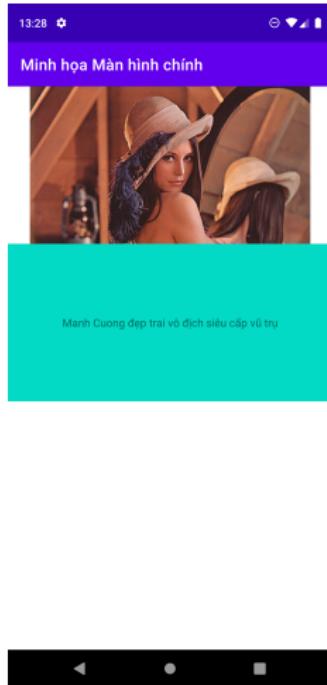


activity_home.xml

```
activity_home.xml x

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4         xmlns:tools="http://schemas.android.com/tools"
5         android:layout_width="match_parent"
6         android:layout_height="match_parent"
7         tools:context=".HomeActivity">
8
9     <androidx.recyclerview.widget.RecyclerView
10        android:id="@+id/rvContainer" ←
11        android:layout_width="match_parent"
12        android:layout_height="match_parent"/>
13 </androidx.constraintlayout.widget.ConstraintLayout>
```

Hình 14: Minh họa màn hình chính
của ứng dụng

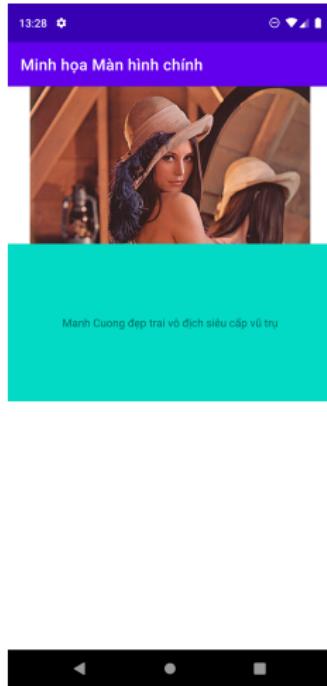


container_image.xml

```
container_image.xml x

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4         xmlns:tools="http://schemas.android.com/tools"
5             android:layout_width="match_parent"
6             android:layout_height="wrap_content">
7
8     <ImageView
9         android:id="@+id/ivImage"
10        android:layout_width="match_parent"
11        android:layout_height="200dp"
12        android:scaleType="fitCenter"
13        tools:ignore="MissingConstraints" />
14
15 </androidx.constraintlayout.widget.ConstraintLayout>
```

Hình 14: Minh họa màn hình chính
của ứng dụng



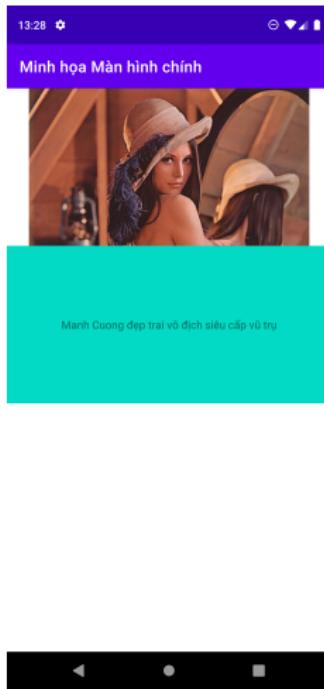
container_text.xml

```
container_text.xml x

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4         android:background="@color/teal_200"
5         android:layout_width="match_parent"
6         android:layout_height="200dp">
7
8     <TextView
9         android:id="@+id/tvText"
10        android:gravity="center"
11        android:layout_width="match_parent"
12        android:layout_height="match_parent" />
13
14 </androidx.constraintlayout.widget.ConstraintLayout>
```

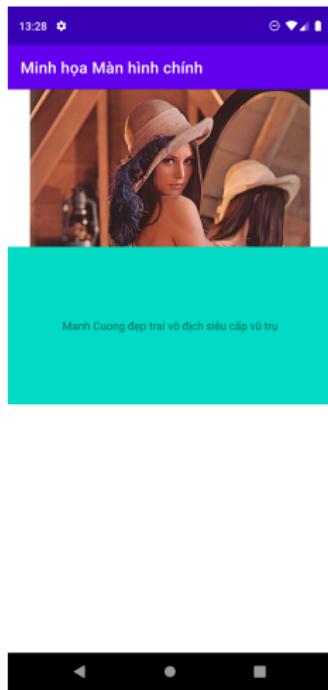
Hình 14: Minh họa màn hình chính
của ứng dụng

HomeModel.java



```
1 package team9.demo4slide;
2
3 public class HomeModel {
4     public static final int IMAGE_CONTAINER = 0;
5     public static final int TEXT_CONTAINER = 1;
6
7     int type;
8     String value;
9
10    public HomeModel(int type, String value) {
11        this.type = type;
12        this.value = value;
13    }
14
15    public int getType() { return type; }
16
17    public void setType(int type) { this.type = type; }
18
19    public String getValue() { return value; }
20
21    public void setValue(String value) { this.value = value; }
22}
```

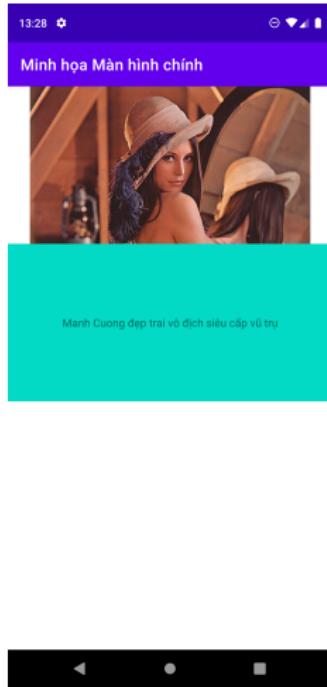
Hình 14: Minh họa màn hình chính
của ứng dụng



Hình 14: Minh họa màn hình chính
của ứng dụng

HomeAdapter.java (1)

```
❸ HomeAdapter.java ×
 4 import android.view.LayoutInflater;
 5 import android.view.View;
 6 import android.view.ViewGroup;
 7 import android.widget.ImageView;
 8 import android.widget.TextView;
 9
10 import androidx.annotation.NonNull;
11 import androidx.recyclerview.widget.RecyclerView;
12
13 import com.bumptech.glide.Glide;
14
15 import java.util.Comparator;
16 import java.util.List;
17
18 public class HomeAdapter extends RecyclerView.Adapter {
19
20     List<HomeModel> containers;
21
22     public HomeAdapter(List<HomeModel> containers) {
23         this.containers = containers;
24
25         // sắp xếp các thành phần tăng dần dựa theo @type của chúng
26         this.containers.sort(Comparator.comparing(HomeModel::getType));
27     }
}
```

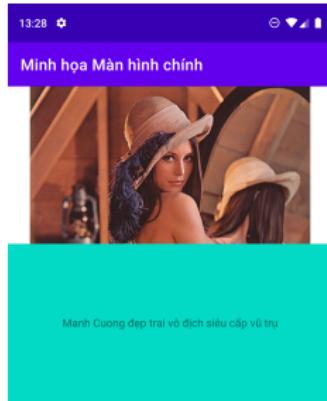


HomeAdapter.java (2)

```
29     * @NotNull
30     * @Override
31     public RecyclerView.ViewHolder onCreateViewHolder(@NotNull ViewGroup parent, int viewType) {
32         switch(viewType) {
33             case HomeModel.IMAGE_CONTAINER:
34                 return new ImageViewHolder(
35                     LayoutInflater.from(parent.getContext()).inflate(
36                         R.layout.container_image, parent, false));
37             case HomeModel.TEXT_CONTAINER:
38                 return new TextViewHolder(
39                     LayoutInflater.from(parent.getContext()).inflate(
40                         R.layout.container_text, parent, false));
41             default: return null;
42         }
43     }
44
45
46     * @Override
47     public int getItemViewType(int position) {
48         switch (containers.get(position).getType()) {
49             case HomeModel.IMAGE_CONTAINER:
50                 return HomeModel.IMAGE_CONTAINER;
51
52             case HomeModel.TEXT_CONTAINER:
53                 return HomeModel.TEXT_CONTAINER;
54
55             default: return -1;
56         }
57     }

```

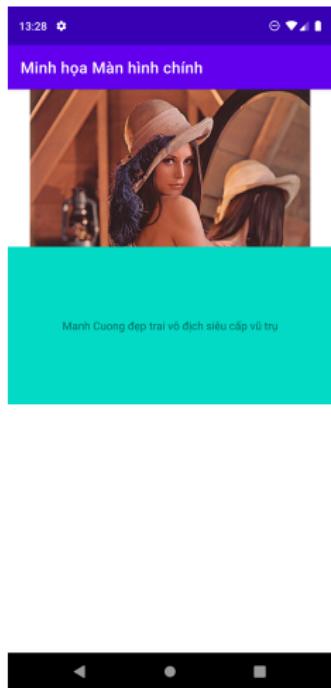
Hình 14: Minh họa màn hình chính
của ứng dụng



HomeAdapter.java (3)

```
59     @Override
60     public void onBindViewHolder(@NotNull RecyclerView.ViewHolder holder, int position) {
61         switch (containers.get(position).getType()) {
62             case HomeModel.IMAGE_CONTAINER:
63                 ((ImageViewHolder) holder).set(containers.get(position).getValue()); break;
64
65             case HomeModel.TEXT_CONTAINER:
66                 ((TextViewHolder) holder).set(containers.get(position).getValue()); break;
67
68             default: return;
69         }
70     }
71
72     @Override
73     public int getItemCount() { return containers.size(); }
```

Hình 14: Minh họa màn hình chính
của ứng dụng



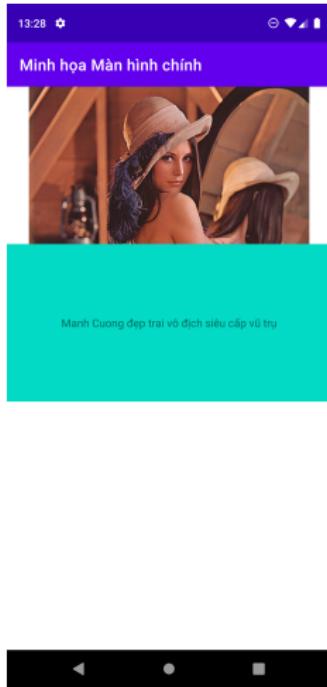
HomeAdapter.java (4)

```
13:28 * 13:28 * 
Minh họa Màn hình chính

77     public class ImageViewHolder extends RecyclerView.ViewHolder {
78         ImageView mImage;
79
80         public ImageViewHolder(@NonNull View itemView) {
81             super(itemView);
82             mImage = itemView.findViewById(R.id.ivImage);
83         }
84
85         private void set(String image) {
86             Glide.with(itemView.getContext()).load(image).into(mImage);
87         }
88     }
89
90     public class TextViewHolder extends RecyclerView.ViewHolder {
91         TextView mText;
92
93         public TextViewHolder(@NonNull View itemView) {
94             super(itemView);
95             mText = itemView.findViewById(R.id.tvText);
96         }
97
98         private void set(String text) { mText.setText(text); }
99     }
100
101 }
```

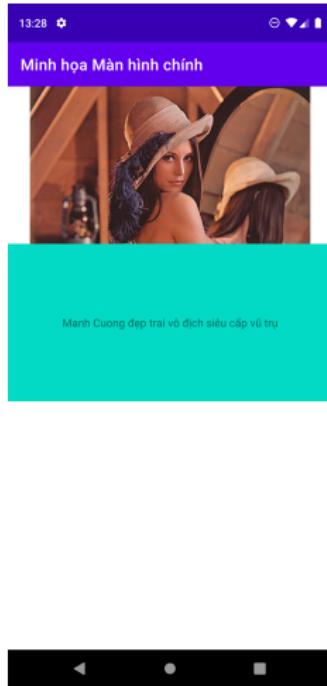
Hình 14: Minh họa màn hình chính
của ứng dụng

HomeActivity.java (1)



```
1 package team9.demo4slide;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.recyclerview.widget.LinearLayoutManager;
5 import androidx.recyclerview.widget.RecyclerView;
6
7 import android.os.Bundle;
8
9 import java.util.ArrayList;
10 import java.util.List;
11
12 public class HomeActivity extends AppCompatActivity {
13
14     RecyclerView mRvContainer;
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_home);
20
21         // tham chiếu đến @rvContainer trên activity_home.xml
22         mRvContainer = findViewById(R.id.rvContainer);
23     }
}
```

Hình 14: Minh họa màn hình chính
của ứng dụng



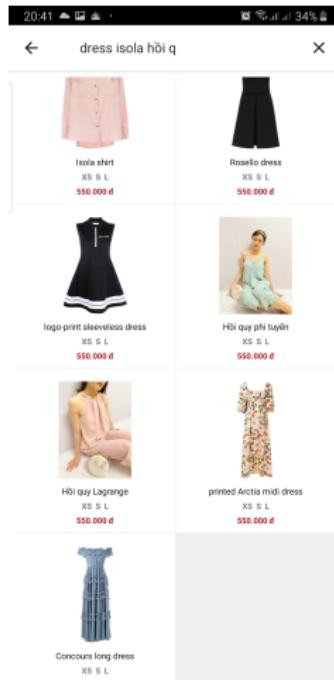
HomeActivity.java (2)

```
24     // chuẩn bị dữ liệu cho adapter
25     List<HomeModel> containers = new ArrayList<>();
26     containers.add(new HomeModel(HomeModel.IMAGE_CONTAINER,
27         value: "https://raw.githubusercontent.com/nkmk/python-snippets/master/notebook/data/src/lena.jpg"));
28     containers.add(new HomeModel(HomeModel.TEXT_CONTAINER,
29         value: "Mạnh Cường đẹp trai vô địch siêu cấp vũ trụ"));
30
31
32     // thiết lập adapter và layout-manager
33     HomeAdapter adapter = new HomeAdapter(containers);
34     LinearLayoutManager layoutManager = new LinearLayoutManager(context: HomeActivity.this);
35     layoutManager.setOrientation(LinearLayoutManager.VERTICAL);
36
37     mRvContainer.setLayoutManager(layoutManager);
38     mRvContainer.setAdapter(adapter);
39
40     adapter.notifyDataSetChanged();
41 }
42 }
```

Hình 14: Minh họa màn hình chính
của ứng dụng

Các màn hình và chức năng kèm theo (8)

Chức năng tìm kiếm



Các thành phần và chức năng kèm theo trên màn hình:

- Khi người dùng nhấn vào bất kỳ một menu item search nào, một search box như hình trên sẽ hiển thị ra, đồng thời người dùng sẽ được đưa đến một fragment khác, nơi các sản phẩm được tìm kiếm sẽ hiển thị dưới dạng một GridView.
- Ở đây, nhóm tìm kiếm các sản phẩm dựa vào ba từ khóa là **dress**, **isola** và **hồi**, thì toàn bộ các sản phẩm được tìm ra bên dưới đều có tên chứa ít nhất một từ mà người dùng nhập vào.

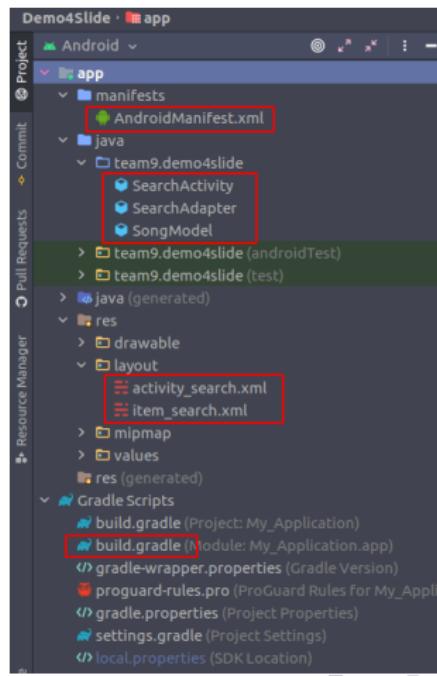
Hình 15: Chức năng tìm kiếm sản phẩm

Các màn hình và chức năng kèm theo (9)

Chức năng tìm kiếm - Mã nguồn minh họa



Cấu trúc cây thư mục:



Hình 16: Minh họa chức năng tìm kiếm sản phẩm



AndroidManifest.xml

```
AndroidManifest.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="team9.demo4slide">
4          <uses-permission android:name="android.permission.INTERNET" />
5
6      <application
7          android:allowBackup="true"
8          android:icon="@mipmap/ic_launcher"
9          android:label="Minh họa Chức năng tìm kiếm"
10         android:roundIcon="@mipmap/ic_launcher_round"
11         android:supportsRtl="true"
12         android:theme="@style/Theme.MyApplication">
13             <activity
14                 android:name=".SearchActivity"
15                 android:exported="true">
16                 <intent-filter>
17                     <action android:name="android.intent.action.MAIN" />
18
19                     <category android:name="android.intent.category.LAUNCHER" />
20                 </intent-filter>
21             </activity>
22         </application>
23     </manifest>
```

Hình 16: Minh họa chức năng tìm kiếm sản phẩm



build.gradle

```
You can use the Project Structure dialog to view and edit your project configuration

30     }
31     compileSdkVersion 31
32     buildToolsVersion '30.0.2'
33 }
34
35 dependencies {
36
37     implementation 'androidx.appcompat:appcompat:1.4.0'
38     implementation 'com.google.android.material:material:1.4.0'
39     implementation 'androidx.constraintlayout:constraintlayout:2.1.2'
40     implementation 'com.google.firebaseio:firebase-firebase:24.0.0'implementation 'com.google.firebaseio:firebase-firebase:24.0.0'
41     implementation 'com.google.firebaseio:firebase-analytics:20.0.2'
42     testImplementation 'junit:junit:4.+'
43     androidTestImplementation 'androidx.test.ext:junit:1.1.3'
44     androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
45 }
```

A screenshot of the build.gradle file in an IDE. The file content is shown with line numbers from 30 to 45. Line 40 contains the code "implementation 'com.google.firebaseio:firebase-firebase:24.0.0'", which is highlighted with a red rectangular box. The rest of the file includes standard Android dependencies like androidx.appcompat and com.google.android.material.

Hình 16: Minh họa chức năng tìm kiếm sản phẩm



activity_search.xml

```
activity_search.xml x

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:orientation="vertical"
7      tools:context=".SearchActivity">
8
9      <EditText
10         android:id="@+id/etSearch" ↑
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:hint="Tim kiếm" />
14
15      <GridView
16         android:id="@+id/gvContainer" ↑
17         android:layout_width="match_parent"
18         android:layout_height="wrap_content"
19         android:numColumns="2"/>
20  </LinearLayout>
```

Hình 16: Minh họa chức năng tìm kiếm sản phẩm



item_search.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="100dp">

    <TextView
        android:id="@+id/tvTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:ignore="MissingConstraints" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

A screenshot of an Android Studio code editor showing the XML layout file "item_search.xml". The XML code defines a ConstraintLayout with a single TextView inside. The TextView has an id of "@+id/tvTitle". A red box highlights the entire TextView element, and a red arrow points to the "tvTitle" part of the id attribute.

Hình 16: Minh họa chức năng tìm kiếm sản phẩm



SongModel.java

```
SongModel.java ×
1 package team9.demo4slide;
2
3 import java.util.ArrayList;
4
5 public class SongModel {
6     String title;
7     ArrayList<String> split;
8
9     // bắt buộc
10    public SongModel() {}
11
12    public String getTitle() { return title; }
13
14    public void setTitle(String title) { this.title = title; }
15
16    public ArrayList<String> getSplit() { return split; }
17
18    public void setSplit(ArrayList<String> split) { this.split = split; }
19 }
```

Hình 16: Minh họa chức năng tìm kiếm sản phẩm



SongModel trong Firestore

A screenshot of the Google Cloud Firestore console. The left sidebar shows a collection named "nuoc-soi" and a document named "SongModel". The main view shows a document with the ID "3A7Dfd9b36nVVqspfqXj". The document contains several fields: "how": "How", "you": "you", "like": "like", and "that": "that". A red box highlights the "split" field, which contains the array [0: "how", 1: "you", 2: "like", 3: "that"]. The title of the document is "How you like that".

Start collection	Add document	Start collection
SongModel	3A7Dfd9b36nVVqspfqXj	Add field
	A2UZkDmUn5Kz741fs7WK HcL3xolii9p2my07xDcq aCekm1VHt40cgwywq9rGc dOK3IQikIVVoNrQuC01C	split 0 "how" 1 "you" 2 "like" 3 "that"

Hình 16: Minh họa chức năng tìm kiếm sản phẩm



SearchAdapter.java (1)

```
● SearchAdapter.java ×
1  package team9.demo4slide;
2
3  import android.annotation.SuppressLint;
4  import android.view.LayoutInflater;
5  import android.view.View;
6  import android.view.ViewGroup;
7  import android.widget.BaseAdapter;
8  import android.widget.TextView;
9
10 import java.util.ArrayList;
11 import java.util.List;
12
13 public class SearchAdapter extends BaseAdapter {
14
15     List<SongModel> songs;
16
17     public SearchAdapter() { songs = new ArrayList<>(); }
18
19     public List<SongModel> getSongs() { return songs; }
20
21     public void setSongs(List<SongModel> songs) { this.songs = songs; }
```

Hình 16: Minh họa chức năng tìm kiếm sản phẩm



SearchAdapter.java (2)

```
25     @Override
26     public int getCount() { return songs.size(); }
27
28     @Override
29     public Object getItem(int i) { return null; }
30
31     @Override
32     public long getItemId(int i) { return 0; }
33
34     @Override
35     @SuppressLint({"ViewHolder", "InflateParams"})
36     public View getView(int position, View contentView, ViewGroup viewGroup) {
37         View view = LayoutInflater.from(viewGroup.getContext()).inflate(
38             R.layout.item_search, root: null);
39         TextView mTitle = view.findViewById(R.id.tvTitle);
40         mTitle.setText(songs.get(position).getTitle());
41
42         return view;
43     }
44 }
```

Hình 16: Minh họa chức năng tìm kiếm sản phẩm

SearchActivity.java (1)



```
21:27 ⓘ Minh họa Chức năng tìm kiếm
love du like new | How you like that Kill this love

1 package team9.demo4slide;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.text.Editable;
6 import android.text.TextWatcher;
7 import android.widget.EditText;
8 import android.widget.GridView;
9 import com.google.firebaseio.firebaseio.CollectionReference;
10 import com.google.firebaseio.firebaseio.FirebaseFirestore;
11 import com.google.firebaseio.firebaseio.QueryDocumentSnapshot;
12 import java.util.ArrayList;
13 import java.util.Arrays;
14 import java.util.List;
15
16 <�新 public class SearchActivity extends AppCompatActivity {
17     EditText mSearch;
18     GridView mContainer;
19     CollectionReference songRef;
20     SearchAdapter adapter;
21     boolean canSearch = true;
22
23     @Override
24     保护了 void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         setContentView(R.layout.activity_search);
27
28         songRef = FirebaseFirestore.getInstance().collection("SongModel");
29         adapter = new SearchAdapter();
30
31         mSearch = findViewById(R.id.etSearch);
32         mContainer = findViewById(R.id.gvContainer);
33         mContainer.setAdapter(adapter);
```

Hình 16: Minh họa chức năng tìm kiếm sản phẩm



SearchActivity.java (2)

```
35     mSearch.addTextChangedListener(new TextWatcher() {  
36         @Override  
37         public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {}  
38  
39         @Override  
40         public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
41             String keyword = charSequence.toString().trim();  
42  
43             if (canSearch && !keyword.isEmpty()) {  
44                 canSearch = false;  
45                 songRef.whereArrayContainsAny("field", "split",  
46                     Arrays.asList(keyword.split(" "))).get()  
47                     .addOnCompleteListener(task -> {  
48                         if (task.isSuccessful()) {  
49                             List<SongModel> foundedSongs = new ArrayList<>();  
50                             for (QueryDocumentSnapshot snapshot : task.getResult())  
51                                 foundedSongs.add(snapshot.toObject(SongModel.class));  
52                             adapter.setSongs(foundedSongs);  
53                             adapter.notifyDataSetChanged();  
54                         }  
55                     });  
56             }  
57         }  
58     }  
59  
60     @Override  
61     public void afterTextChanged(Editable editable) {}  
62  
63 }  
64 }
```

Hình 16: Minh họa chức năng tìm kiếm sản phẩm

Chạy thử ứng dụng

Link Youtube: <https://youtu.be/yRglZ4Segkk>



Tài nguyên và tham khảo

- Logo ứng dụng và tên Clover: được nhóm xin phép từ nhãn hàng Clover và đã được sự đồng ý từ chị chủ nhãn hàng, trang bán hàng của nhãn hàng tại <https://www.facebook.com/TiemmaynhaAn>.
- Về hình ảnh người mẫu Việt Nam ở mục sản phẩm mới: là bạn Phan Hữu Lộc - cựu sinh viên K17 khoa Sinh học của trường Đại Học Khoa Học Tự Nhiên TP.HCM và là người mẫu của nhãn hàng Clover phía trên.
- Các hình ảnh sản phẩm khác được nhóm lấy về (chưa xin phép) từ trang bán hàng Farfetch tại <https://www.farfetch.com>.
- Về các kỹ thuật code trong đồ án ngoài slide của Thầy lý thuyết nhóm còn tham khảo thêm trong cuốn **Android Programming for Beginners**, link sách tại <https://www.amazon.com/Android-Programming-Beginners-depth-full-featured/dp/1800563434>.

Hỏi đáp và kết thúc



HẾT

Cảm ơn Thầy và các bạn đã chú ý lắng nghe.