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 mobile 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 mobile 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

Hình 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.

Các màn hình và chức năng kèm theo (2)

Màn hình đăng nhập - Mã nguồn minh họa

Hình 2: Màn hình đăng nhập

Cấu trúc cây thư mục:

Các màn hình và chức năng kèm theo (3)

Màn hình chi tiết sản phẩm

Hình 3: Màn hình chi tiết sản phẩm 1

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 3: 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 2

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 mới sản phẩm hiện tại.

Hình 5: Màn hình chi tiết sản phẩm 3

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.

Các màn hình và chức năng kèm theo (4)

Màn hình chính

Hình 6: Màn hình chính của ứng dụng 1

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 lí).
  - Menu item Cart - bấm vào để đi đến fragment giỏ hàng.

Hình 6: 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 (Horizontal Recycler View) - 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 7: Màn hình chính của ứng dụng 2

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 8: Màn hình chính của ứng dụng 3

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 đứa 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.

Các màn hình và chức năng kèm theo (4)

Màn hình chính - Mã nguồn minh họa

Hình 6: Màn hình chính của ứng dụng 1

Cấu trúc cây thư mục:

Hình 6: Màn hình chính của ứng dụng 1

AndroidManifest.xml

AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="team9.demo4slide">
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Minh họa Màn hình chính"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication">
        <activity
            android:name=".HomeActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Hình 6: Màn hình chính của ứng dụng 1

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'
39        annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'
40    }
```

Hình 6: Màn hình chính của ứng dụng 1

activity_home.xml

activity_home.xml

```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="match_parent"
    tools:context=".HomeActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvContainer"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

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

Hình 6: Màn hình chính của ứng dụng 1

container_image.xml

container_image.xml

```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="wrap_content">

    <ImageView
        android:id="@+id/ivImage"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:scaleType="fitCenter"
        tools:ignore="MissingConstraints" />

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

Hình 6: Màn hình chính của ứng dụng 1

container_text.xml

container_text.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
    android:background="@color/teal_200"
    android:layout_width="match_parent"
    android:layout_height="200dp">

    <TextView
        android:id="@+id/tvText"
        android:gravity="center"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

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

HomeModel.java

Hình 6: Màn hình chính của ứng dụng 1

HomeModel.java

```java
package team9.demo4slide;

public class HomeModel {
    public static final int IMAGE_CONTAINER = 0;
    public static final int TEXT_CONTAINER = 1;

    int type;
    String value;

    public HomeModel(int type, String value) {
        this.type = type;
        this.value = value;
    }

    public int getType() { return type; }

    public void setType(int type) { this.type = type; }

    public String getValue() { return value; }

    public void setValue(String value) { this.value = value; }
}
```

HomeAdapter.java (1)

Hình 6: Màn hình chính của ứng dụng 1

HomeAdapter.java

```java
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;

import java.util.Comparator;
import java.util.List;

public class HomeAdapter extends RecyclerView.Adapter {

    List<HomeModel> containers;

    public HomeAdapter(List<HomeModel> containers) {
        this.containers = containers;

        // sắp xếp các thành phần tăng dần dựa theo @type của chúng
        this.containers.sort(Comparator.comparing(HomeModel::getType));
    }
```

Hình 6: Màn hình chính của ứng dụng 1

HomeAdapter.java (2)

```java
29  @NonNull
30  @Override
31  public RecyclerView.ViewHolder onCreateViewHolder(@NonNull 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, attachToRoot: false));
37          case HomeModel.TEXT_CONTAINER:
38              return new TextViewHolder(
39                      LayoutInflater.from(parent.getContext()).inflate(
40                              R.layout.container_text, parent, attachToRoot: false));
41
42          default: return null;
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 6: Màn hình chính của ứng dụng 1

HomeAdapter.java (3)

```java
59      @Override
60      public void onBindViewHolder(@NonNull 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 6: Màn hình chính của ứng dụng 1

HomeAdapter.java (4)

```
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); }
101     }
102 }
```

HomeActivity.java (1)

Hình 6: Màn hình chính của ứng dụng 1

HomeActivity.java

```java
package team9.demo4slide;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;

import java.util.ArrayList;
import java.util.List;

public class HomeActivity extends AppCompatActivity {

    RecyclerView mRvContainer;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);

        // tham chiếu đến @rvContainer trên activity_home.xml
        mRvContainer = findViewById(R.id.rvContainer);

```

Hình 6: Màn hình chính của ứng dụng 1

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/" +
28                          "nkmk/python-snippets/master/notebook/data/src/lena.jpg"));
29          containers.add(new HomeModel(HomeModel.TEXT_CONTAINER,
30                  value: "Manh Cuong đẹp trai vô địch siêu cấp vũ trụ"));
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  }
```

Các màn hình và chức năng kèm theo (5)

Chức năng tìm kiếm

Hình 9: Chức năng tìm kiếm sản phẩm

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

- Khi người dùng nhấn vào bất kì 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.

Chạy thử ứng dụng

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 kĩ 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.