

TRƯỜNG CAO ĐẲNG FPT POLYTECHNIC



**FPT POLYTECHNIC**

BÁO CÁO DỰ ÁN THỰC TẬP TỐT NGHIỆP



**XÂY DỰNG ỨNG DỤNG BÁN HÀNG THỜI TRANG**

**--- FashionStoreApp---**

Họ và tên :

Phùng Văn Quý- PH27252

Chuyên ngành:

Lập trình Mobile

Kỳ:

Fall2023

Hà Nội – 19/11/2023

**SINH VIÊN THỰC HIỆN**

STT	Họ và tên	MSSV	Tỉ lệ đóng góp
1	Phùng Văn Quý	PH27252	100%

**NHẬN XÉT CỦA DOANH NGHIỆP:**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Hà nội, ngày..., tháng 11, năm 2023

## MỤC LỤC

<b>PHẦN 1: MỞ ĐẦU</b>	<b>1</b>
<b>PHẦN 2: CƠ SỞ LÝ THUYẾT THỰC HIỆN PROJECT</b>	<b>2</b>
<b>PHẦN 3: PHÂN TÍCH - THIẾT KẾ CƠ SỞ DỮ LIỆU</b>	<b>3</b>
<b>PHẦN 4: USECASE VÀ LUỒNG CHẠY HỆ THỐNG</b>	<b>4</b>
4.1. Usecase hệ thống	4
4.2. Usecase luồng chạy của hệ thống	5
<b>PHẦN 5: GIAO DIỆN HỆ THỐNG</b>	<b>6</b>
5.1. Giao diện đăng ký	6
5.2. Giao diện đăng nhập	9
5.3. Giao diện đăng nhập Google	12
5.4. Giao diện quên mật khẩu	15
5.5. Giao diện trang chính	19
5.6. Giao diện sản phẩm theo danh mục	22
5.7. Giao diện tìm kiếm sản phẩm	24
5.8. Giao diện giỏ hàng	26
5.9. Giao diện đặt hàng	29
5.10. Giao diện đổi địa chỉ	33
5.11. Giao diện liệt kê đơn hàng	35
5.12. Giao diện liệt kê đơn hàng thanh toán tiền mặt	37
5.13. Giao diện liệt kê đơn hàng thanh toán bằng zalo pay	39
5.14. Giao diện đặt hàng thành công	41
5.15. Giao diện Profile	43
5.16. Giao diện edit profile	46
5.17. Giao diện thay đổi mật khẩu	50
<b>CÁC GIAO DIỆN ADMIN ĐƯỢC LÀM TỪ PAGEVIEW:</b>	<b>53</b>
5.17. Giao diện đăng nhập tài khoản admin	54

<b>5.18. Giao diện chính của admin</b>	<b>55</b>
<b>5.19. Giao diện danh sách tất cả các đơn hàng đã order</b>	<b>56</b>
<b>5.20. Giao diện gửi mail đến đơn hàng của người dùng</b>	<b>57</b>
<b>5.21. Giao diện chi tiết hoá đơn của một đơn hàng</b>	<b>57</b>
<b>5.22. Giao diện Wallet</b>	<b>58</b>
<b>PHẦN 6: KẾT LUẬN</b>	<b>60</b>
<b>TÀI LIỆU THAM KHẢO</b>	<b>61</b>

## **PHẦN 1: MỞ ĐẦU**

### **1.1. Phát biểu bài toán**

Dự án xây dựng app thương mại điện tử shop thời trang là một dự án nhằm xây dựng một ứng dụng thương mại điện tử cho phép người dùng mua sắm các sản phẩm thời trang trực tuyến. Bài toán đặt ra là làm thế nào để phát triển một ứng dụng thương mại điện tử đáp ứng được nhu cầu mua sắm của khách hàng, đồng thời đảm bảo tính bảo mật, an toàn và tiện lợi cho người dùng.

### **1.2. Mục đích**

Mục đích của dự án là xây dựng một ứng dụng thương mại điện tử cho phép khách hàng mua sắm các sản phẩm thời trang trực tuyến một cách dễ dàng và thuận tiện. Ứng dụng sẽ cung cấp cho khách hàng một trải nghiệm mua sắm trực tuyến tốt hơn, giúp tiết kiệm thời gian và chi phí đi lại để mua sắm. Đồng thời, ứng dụng cũng cần đảm bảo tính bảo mật và an toàn thông tin của khách hàng khi mua sắm trực tuyến.

### **1.3. Phạm vi và đối tượng**

Phạm vi của dự án là xây dựng một ứng dụng thương mại điện tử trên nền tảng di động, cho phép người dùng mua sắm các sản phẩm thời trang trực tuyến. Ứng dụng sẽ cung cấp cho người dùng các tính năng như tìm kiếm sản phẩm, đặt hàng, thanh toán, quản lý đơn hàng, đánh giá sản phẩm.

Đối tượng của dự án là các khách hàng có nhu cầu mua sắm các sản phẩm thời trang trực tuyến trên nền tảng di động. Ngoài ra, dự án cũng có thể hướng đến các nhà cung cấp sản phẩm thời trang muốn mở rộng kênh bán hàng trực tuyến và kết nối với khách hàng thông qua ứng dụng thương mại điện tử.

## PHẦN 2: CƠ SỞ LÝ THUYẾT THỰC HIỆN PROJECT

### 2.1. Công cụ - môi trường để lập trình và ngôn ngữ lập trình

- Môi trường lập trình : IntelliJ IDEA, Android Studio
- Ngôn ngữ lập trình : Java.



### 2.2. Các công nghệ được sử dụng

- Spring Framework
- Hibernate
- Restful API

### 2.3. Công cụ lưu trữ dữ liệu MySQL

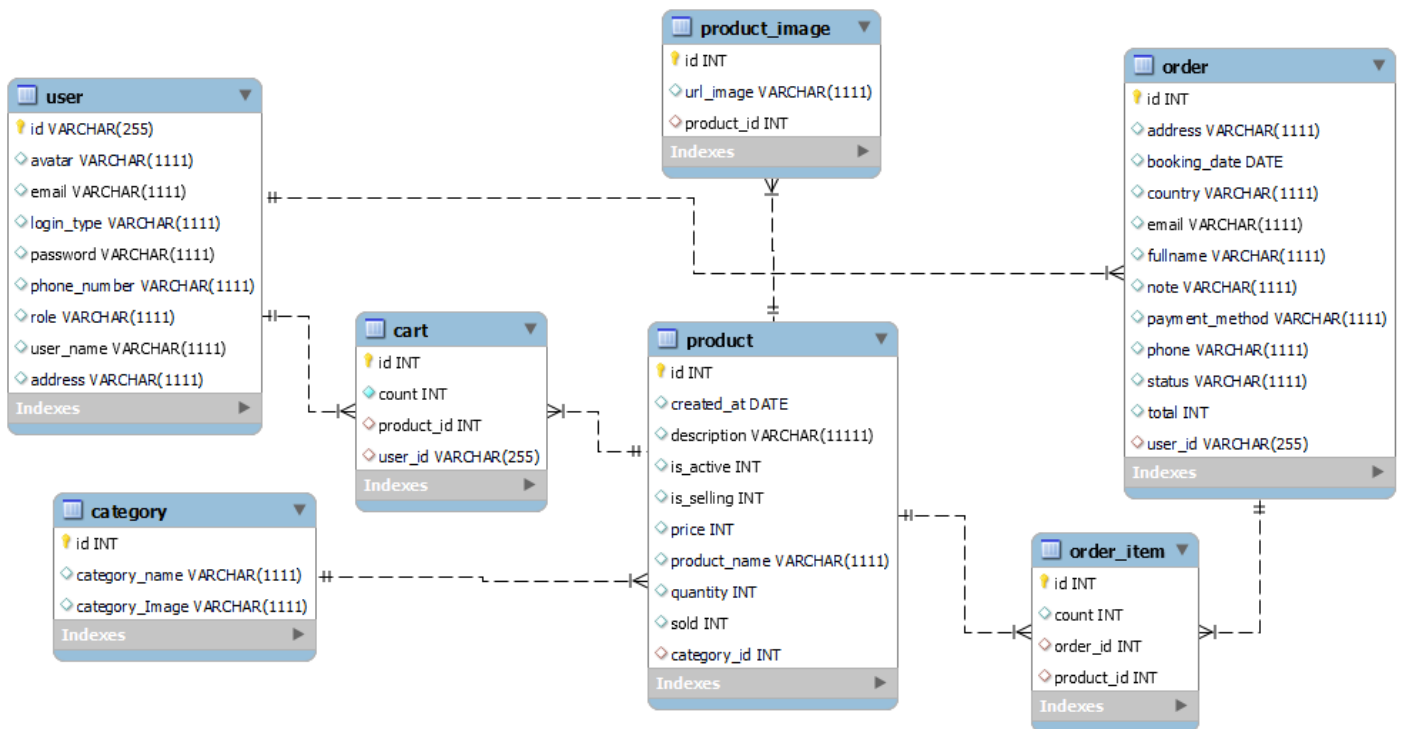


## PHẦN 3: PHÂN TÍCH - THIẾT KẾ CƠ SỞ DỮ LIỆU

### 3.1. Ý tưởng

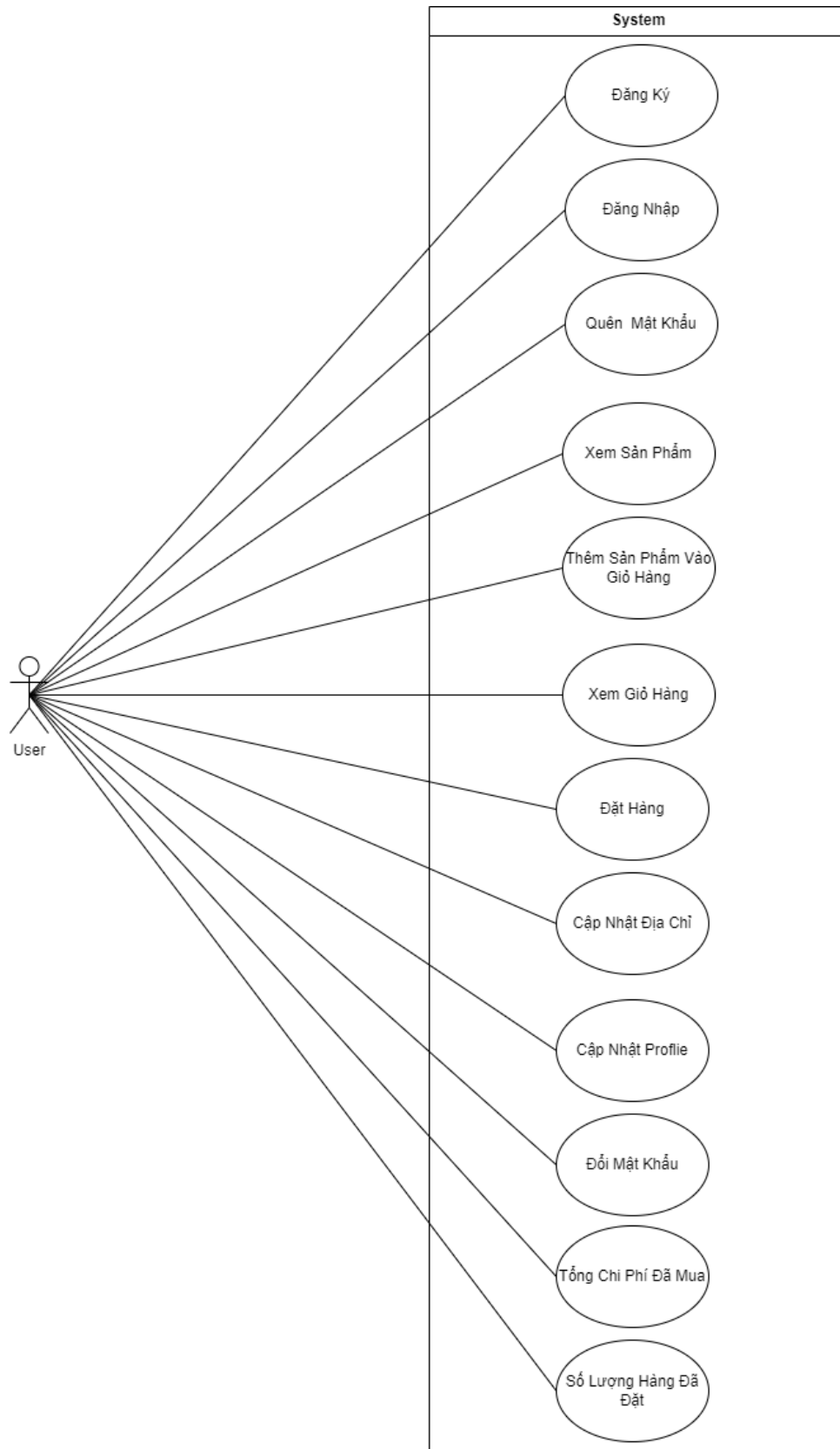
- Thiết kế giao diện người dùng thu hút: Giao diện người dùng là điểm đầu tiên mà người dùng sẽ nhìn thấy khi truy cập vào ứng dụng. Vì vậy, cần thiết kế một giao diện đẹp mắt, dễ sử dụng và thu hút khách hàng đến với ứng dụng.
- Hỗ trợ thanh toán trực tuyến: Tính năng thanh toán trực tuyến là một phần quan trọng của một ứng dụng thương mại điện tử. Ứng dụng hỗ trợ thanh toán qua Zalo Pay.
- Hỗ trợ đặt hàng và vận chuyển hàng hóa: Để trở thành một ứng dụng thương mại điện tử hoàn chỉnh, cần cung cấp cho người dùng khả năng đặt hàng và vận chuyển hàng hóa. Ứng dụng của bạn cần phải hỗ trợ nhiều phương thức vận chuyển và cung cấp cho người dùng các tùy chọn đặt hàng.
- Đăng nhập và bảo mật: Ứng dụng phải có chế độ đăng nhập tài khoản dễ dàng và nhanh chóng, hỗ trợ thêm đăng nhập bằng tài khoản google.

### 3.2. Thiết kế cơ sở dữ liệu



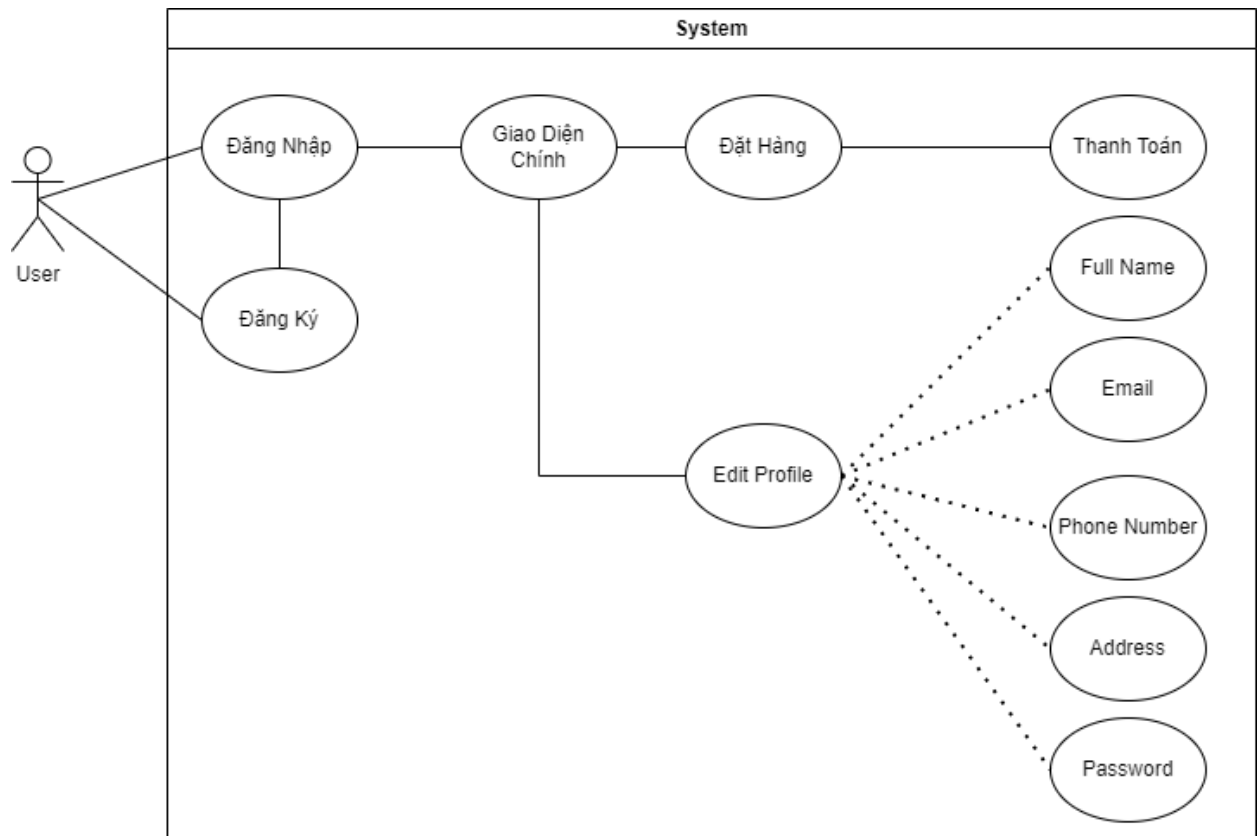
## PHẦN 4: USECASE VÀ LUỒNG CHẠY HỆ THỐNG

### 4.1. Usecase hệ thống





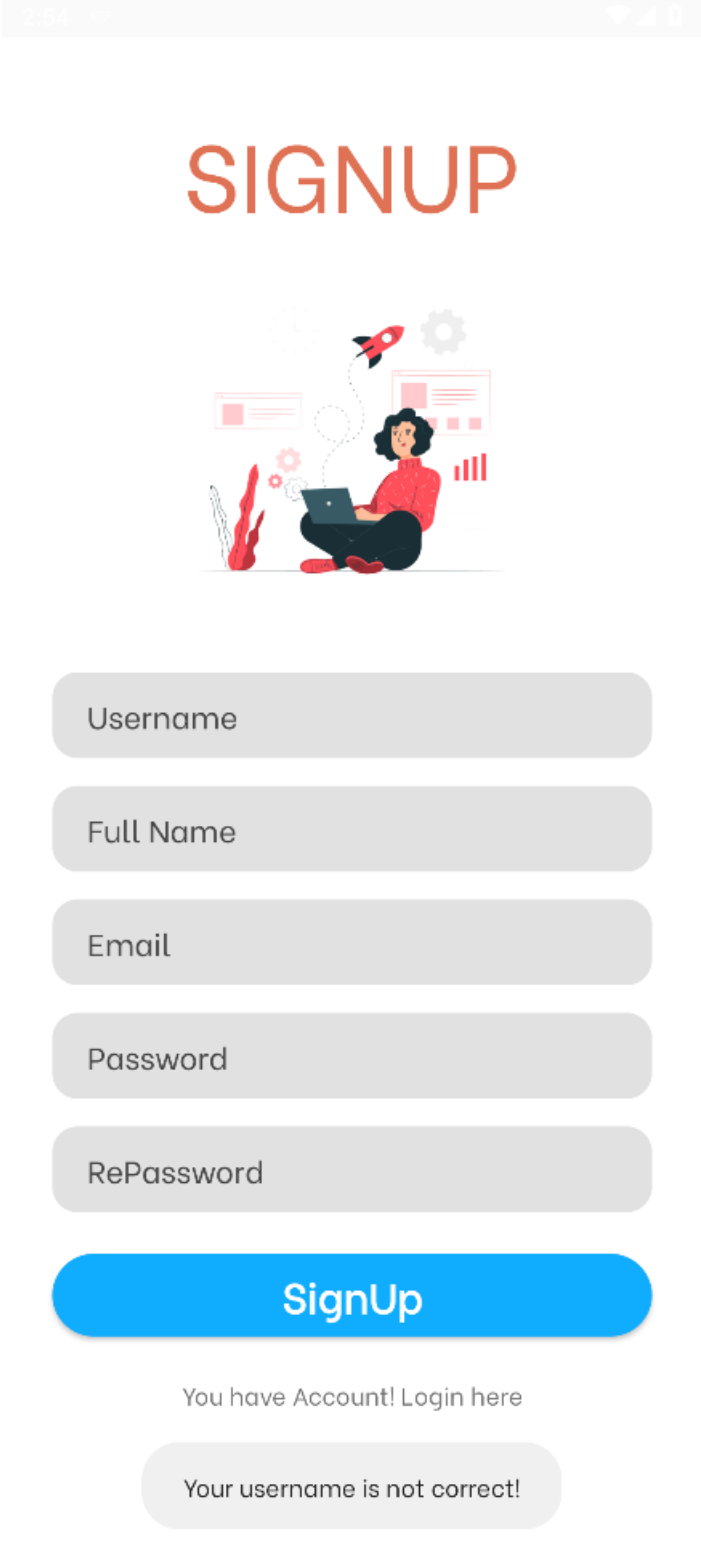
## 4.2. Usecase luồng chạy của hệ thống



## PHẦN 5: GIAO DIỆN HỆ THỐNG

### 5.1. Giao diện đăng ký


Chức năng: Đăng ký bằng Username, fullname, email, password



A mobile application interface for a signup screen. At the top, there is a status bar showing the time 2:50 and battery level. The main heading "SIGNUP" is displayed in large, bold, orange letters. Below the heading is an illustration of a person sitting on the floor, working on a laptop, surrounded by various icons representing technology and business, such as a rocket, gears, charts, and a lightbulb. The form consists of five input fields stacked vertically, each with a light gray background and rounded corners: "Username", "Full Name", "Email", "Password", and "RePassword". Below these fields is a prominent blue button with the text "SignUp" in white. Underneath the button, there is a link that says "You have Account! Login here". At the bottom, a light gray rounded rectangle contains the error message "Your username is not correct!".

2:50

# SIGNUP



Username

Full Name

Email

Password

RePassword

**SignUp**

You have Account! Login here

Your username is not correct!

## Phần code chính thực hiện:

```
@FormUrlEncoded
@POST("/signup")
Call<User> SignUp(@Field("username") String username, @Field("fullname")
String fullname, @Field("email") String email, @Field("password")String
password);
```

```
private void btnSignUpClick() {
    btnSignUp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            etUserName = findViewById(R.id.etUserName);
            etFullName = findViewById(R.id.etNewPass);
            etEmail = findViewById(R.id.etEmail);
            etPassword = findViewById(R.id.etPassword);
            etRePassword = findViewById(R.id.etRePassword);

            if (TextUtils.isEmpty(etUserName.getText().toString())){
                etUserName.setError("Please enter your username");
                etUserName.requestFocus();
                return;
            }

            if (TextUtils.isEmpty(etFullName.getText().toString())){
                etFullName.setError("Please enter your full name");
                etFullName.requestFocus();
                return;
            }

            if (TextUtils.isEmpty(etEmail.getText().toString())){
                etEmail.setError("Please enter your email");
                etEmail.requestFocus();
                return;
            }

            if (TextUtils.isEmpty(etPassword.getText().toString())){
                etPassword.setError("Please enter your username");
                etPassword.requestFocus();
                return;
            }

            if (TextUtils.isEmpty(etRePassword.getText().toString())){
                etRePassword.setError("Please enter your password");
                etRePassword.requestFocus();
                return;
            }

            String username = etUserName.getText().toString();
            String fullname = etFullName.getText().toString();
            String email = etEmail.getText().toString();
            String password = etPassword.getText().toString();
            String repassword = etRePassword.getText().toString();

            if (!password.equals(repassword)){
                etRePassword.setError("Password and RePassword not match");
                etRePassword.requestFocus();
                return;
            }
            else {

                UserAPI.userApi.SignUp(username, fullname, email,
```

```

password).enqueue(new Callback<User>() {
    @Override
    public void onResponse(Call<User> call, Response<User>
response) {
        User user = response.body();
        Log.e("ffff", "Thành công");
        Log.e("ffff", user.toString());
        startActivity(new Intent(SignUpActivity.this,
LoginActivity.class));
    }
    @Override
    public void onFailure(Call<User> call, Throwable t) {
        Log.e("ffff", "Kết nối API thất bại");
    }
});
    }
}
});
}
}

```

## 5.2. Giao diện đăng nhập

Chức năng: Đăng nhập bằng username và password



# LOGIN



[Forgot Your Password?](#)

Login

[Are you new user? Register](#)



Login with Google

---

Phần code chính thực hiện:

```
@GET("/login")
Call<User> Login(@Query("id") String id, @Query("password") String password);
private void btnLoginClick() {
    btnLogin.setOnClickListener(new View.OnClickListener() {
        @Override
```

```

        public void onClick(View v) {
            Login();
        }
    });
}

private void Login() {
    etPassword = findViewById(R.id.etPassword);
    etUserName = findViewById(R.id.etUserName);
    if (TextUtils.isEmpty(etUserName.getText().toString())) {
        etUserName.setError("Please enter your username");
        etUserName.requestFocus();
        return;
    }

    if (TextUtils.isEmpty(etPassword.getText().toString())) {
        etPassword.setError("Please enter your password");
        etPassword.requestFocus();
        return;
    }

    String username = etUserName.getText().toString();
    String password = etPassword.getText().toString();
    // Log.e("ffff", "1===== "+username);
    // Log.e("ffff", "2===== "+password);
    UserAPI.userApi.Login(username,password).enqueue(new
    retrofit2.Callback<User>() {
        @Override
        public void onResponse(Call<User> call, Response<User> response)
        {
            // User user = new User();
            // user.setUser_Name("dmm");
            user = response.body();
            if (user!=null){
                Toast.makeText(LoginActivity.this,"Login Successfully",
                Toast.LENGTH_LONG).show();
            // Log.e("ffff", user.toString());

            ObjectSharedPreferences.saveObjectToSharedPreference(LoginActivity.this,
            "User", "MODE_PRIVATE", user);
            if(user.getAddress()!=null &&
            user.getPhone_Number()!=null){
                Address address = new Address(user.getUser_Name(),
                user.getPhone_Number(), user.getAddress());
                ObjectSharedPreferences.saveObjectToSharedPreference(LoginActivity.this,
                "address", "MODE_PRIVATE", address);
            }

            Intent intent= new Intent(LoginActivity.this,
            MainActivity.class);
            intent.putExtra("object", user);
            startActivity(intent);
            finish();
        }
        else{
            Toast.makeText(LoginActivity.this,"Incorrect UserName or
            Password", Toast.LENGTH_LONG).show();
        }
        Log.e("ffff", "Đăng nhập thành công");
    }

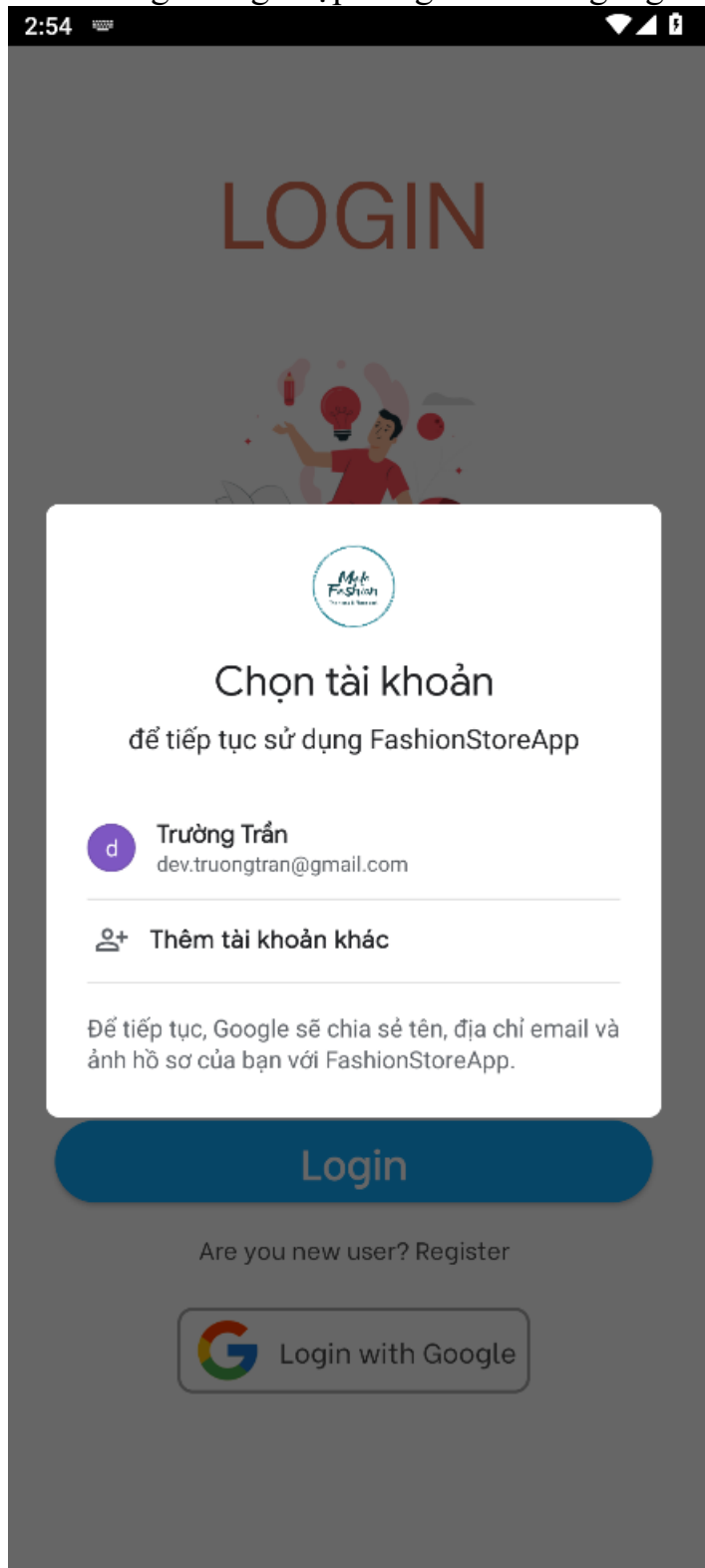
    @Override

```

```
        public void onFailure(Call<User> call, Throwable t) {
            Toast.makeText(LoginActivity.this, "Failed to connect, try
again later", Toast.LENGTH_LONG).show();
            Log.e("ffff", "Kết nối API Login thất bại");
            Log.e("TAG", t.toString());
        }
    });
}
```

### 5.3. Giao diện đăng nhập Google

Chức năng: Đăng nhập bằng tài khoản google



Phần code chính thực hiện:

```
@FormUrlEncoded
@POST("/google")
Call<User> LoginWitGoogle(@Field("id") String userId, @Field("fullname")
String fullName, @Field("email") String email, @Field("avatar") String
avatar);
```



```

private void clGoogleClick() {
    clGoogle.setOnClickListener(v -> {
        Intent signInIntent = googleSignInClient.getSignInIntent();
        startActivityForResult(signInIntent,1000);
    });
}
@Override
protected void onActivityResult(int requestCode, int resultCode,Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode == 1000){
        Task<GoogleSignInAccount> task =
        GoogleSignIn.getSignedInAccountFromIntent(data);

        try {
            task.getResult(ApiException.class);
            navigateToSecondActivity();
        } catch (ApiException e) {
            Toast.makeText(getApplicationContext(), "Something went wrong",
            Toast.LENGTH_SHORT).show();
        }
    }
}

void navigateToSecondActivity(){
    GoogleSignInAccount acct = GoogleSignIn.getLastSignedInAccount(this);
    if(acct!=null){
        String id = acct.getId();
        String name = acct.getDisplayName();
        String email = acct.getEmail();
        String avatar = String.valueOf(acct.getPhotoUrl());
        UserAPI.userApi.LoginWitGoogle(id, name, email, avatar).enqueue(new
        Callback<User>() {
            @Override
            public void onResponse(Call<User> call, Response<User> response)
            {
                user = response.body();
                if (user!=null){
                    Toast.makeText(LoginActivity.this,"Login Successfully",
                    Toast.LENGTH_LONG).show();

                    ObjectSharedPreferences.saveObjectToSharedPreference(LoginActivity.this,
                    "User", "MODE_PRIVATE", user);
                    if(user.getAddress() !=null &&
                    user.getPhone_Number() !=null){
                        Address address = new Address(user.getUser_Name(),
                        user.getPhone_Number(), user.getAddress());
                        ObjectSharedPreferences.saveObjectToSharedPreference(LoginActivity.this,
                        "address", "MODE_PRIVATE", address);
                    }

                    Intent intent= new Intent(LoginActivity.this,
                    MainActivity.class);
                    intent.putExtra("object", user);
                    startActivity(intent);
                    finish();
                }
                else{
                    Toast.makeText(LoginActivity.this,"Incorrect UserName or
                    Password", Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}

```

```

        }
    }

    @Override
    public void onFailure(Call<User> call, Throwable t) {

    }

});

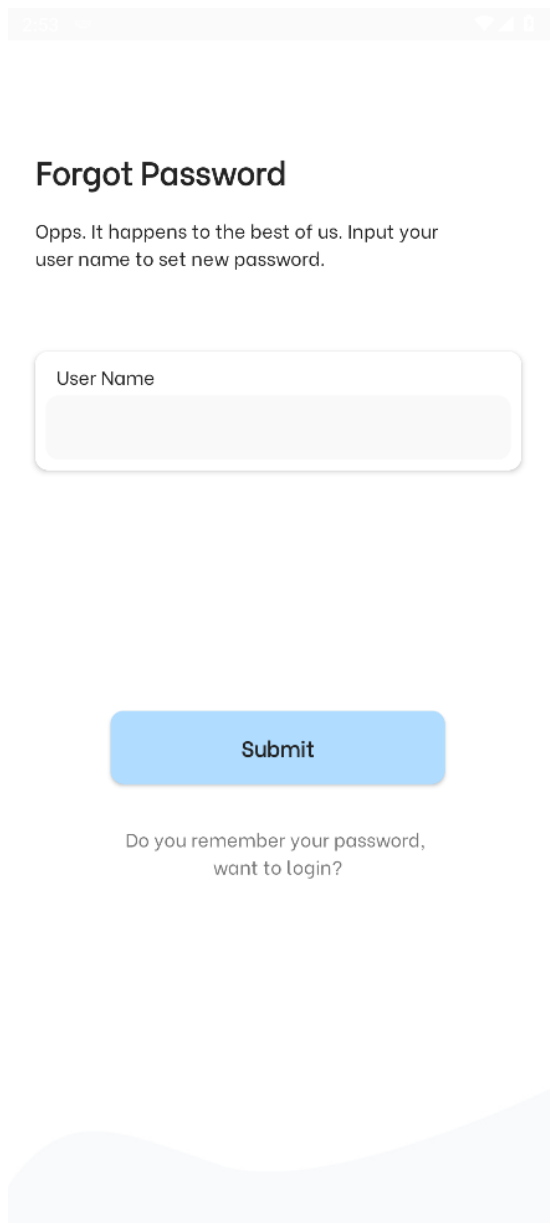
}

private void tvForgotPasswordClick() {
    tvForgotPassword.setOnClickListener(v -> {
        startActivity(new Intent(LoginActivity.this,
ForgotPasswordActivity.class));
    });
}
}

```

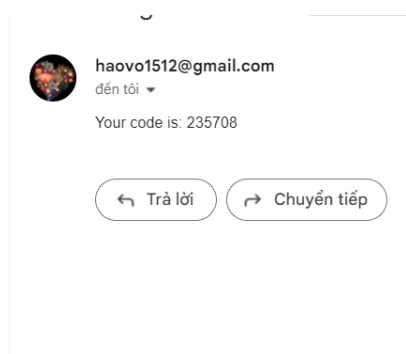
## 5.4. Giao diện quên mật khẩu

Chức năng: Tìm lại mật khẩu đã mất



The image shows a mobile application interface for a 'Forgot Password' feature. At the top, there's a status bar with the time 12:51 and signal/battery icons. The title 'Forgot Password' is prominently displayed. Below it, a message states: 'Oops. It happens to the best of us. Input your user name to set new password.' There is a text input field labeled 'User Name'. Below the input field is a blue 'Submit' button. At the bottom, there is a link that says 'Do you remember your password, want to login?'. The background is a light gray with a subtle wave pattern at the bottom.

⇒ Kết quả:



The image shows a mobile application interface for email verification. At the top, there's a status bar with the time 12:51 and signal/battery icons. Below the status bar, there's a header with a circular profile picture and the email address 'haovo1512@gmail.com'. Below the email address, there's a dropdown menu with the text 'đến tôi'. Below the dropdown menu, there's a message that says 'Your code is: 235708'. At the bottom, there are two buttons: 'Trả lời' (Reply) and 'Chuyển tiếp' (Forward).

## Phần code chính thực hiện:

```
@FormUrlEncoded
@POST("/forgotnewpass")
Call<String> forgotNewPass(@Field("id") String userId, @Field("code") String
code, @Field("password") String password);
    private void btnSubmitClick() {
        btnSubmit.setOnClickListener(view -> {
            String user_id = etUserName.getText().toString();
            if (user_id.isEmpty()) {
                Toast.makeText(this, "Please enter your user name!",
Toast.LENGTH_SHORT).show();
                etUserName.requestFocus();
                return;
            }
            ProgressDialog progressDialog = new
ProgressDialog(ForgotPasswordActivity.this);
            progressDialog.setMessage("Checking..."); // Setting Message
            progressDialog.setTitle("Forgot Password"); // Setting Title
            progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER); //
Progress Dialog Style Spinner
            progressDialog.show(); // Display Progress Dialog
            progressDialog.setCancelable(false);
            UserAPI.userApi.forgotPassword(user_id).enqueue(new
Callback<String>() {
                @Override
                public void onResponse(Call<String> call, Response<String>
response) {
                    String codeForgot = response.body();
                    // Log.e("===", codeForgot);
                    progressDialog.dismiss();
                    if(response.isSuccessful()){
                        Log.e("===", codeForgot);
                        clForgotPassword.setVisibility(View.GONE);
                        clVerification.setVisibility(View.VISIBLE);
                        // smsConfirmationView.requestFocus();
                        smsConfirmationView.setOnChangeListener((code,
isComplete) -> {
                            if(isComplete==true){
                                if(code.equals(codeForgot)){
                                    clVerification.setVisibility(View.GONE);

clSetNewPassword.setVisibility(View.VISIBLE);
                                    btnSubmitPassword.setOnClickListener(v ->
{
                                        String newPass =
etNewPass.getText().toString();
                                        String reNewPass =
etReNewPass.getText().toString();
                                        if (newPass.equals(reNewPass) &&
!newPass.isEmpty()) {
                                            UserAPI.userApi.forgotNewPass(user_id, codeForgot, newPass).enqueue(new
                                            Callback<String>() {
                                                @Override
                                                public void
onResponse(Call<String> call, Response<String> response) {
                                                    if
                                                    (response.isSuccessful()){
                                                        Dialog dialog = new
Dialog(ForgotPasswordActivity.this);
                                                        dialog.setContentView(R.layout.dialog_forgot_password_success);
```

```

dialog.getWindow().setLayout(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.WRAP_CONTENT);

dialog.getWindow().getAttributes().windowAnimations = R.style.animation;

dialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));

dialog.getWindow().setGravity(Gravity.BOTTOM);

dialog.setCancelable(false);
//
Intent (ForgotPasswordActivity.this, LoginActivity.class));
startActivity(new
dialog.show();
Button btnLogin =

dialog.findViewById(R.id.btnChangePassword);

btnLogin.setOnClickListener(v1 -> {
startActivity(new
Intent (ForgotPasswordActivity.this, LoginActivity.class));
});
}
@Override
public void
onFailure(Call<String> call, Throwable t) {
}
});
}
else{

tvPasswordNotMatch.setVisibility(View.VISIBLE);

Toast.makeText(ForgotPasswordActivity.this, "Password and confirm password do
not match!", Toast.LENGTH_SHORT).show();
}
});
tvLogin3.setOnClickListener(v -> {
startActivity(new
Intent (ForgotPasswordActivity.this, LoginActivity.class));
});
}
else{

tvCodeNotCorrect.setVisibility(View.VISIBLE);

Toast.makeText(ForgotPasswordActivity.this, "Your code is not correct!",
Toast.LENGTH_SHORT).show();
}
}
});
tvLogin2.setOnClickListener(v -> {
startActivity(new
Intent (ForgotPasswordActivity.this, LoginActivity.class));
});
ivBack.setOnClickListener(v -> {
tvUserNameNotCorrect.setVisibility(View.GONE);
etUserName.setText(null);
clVerification.setVisibility(View.GONE);

```

```

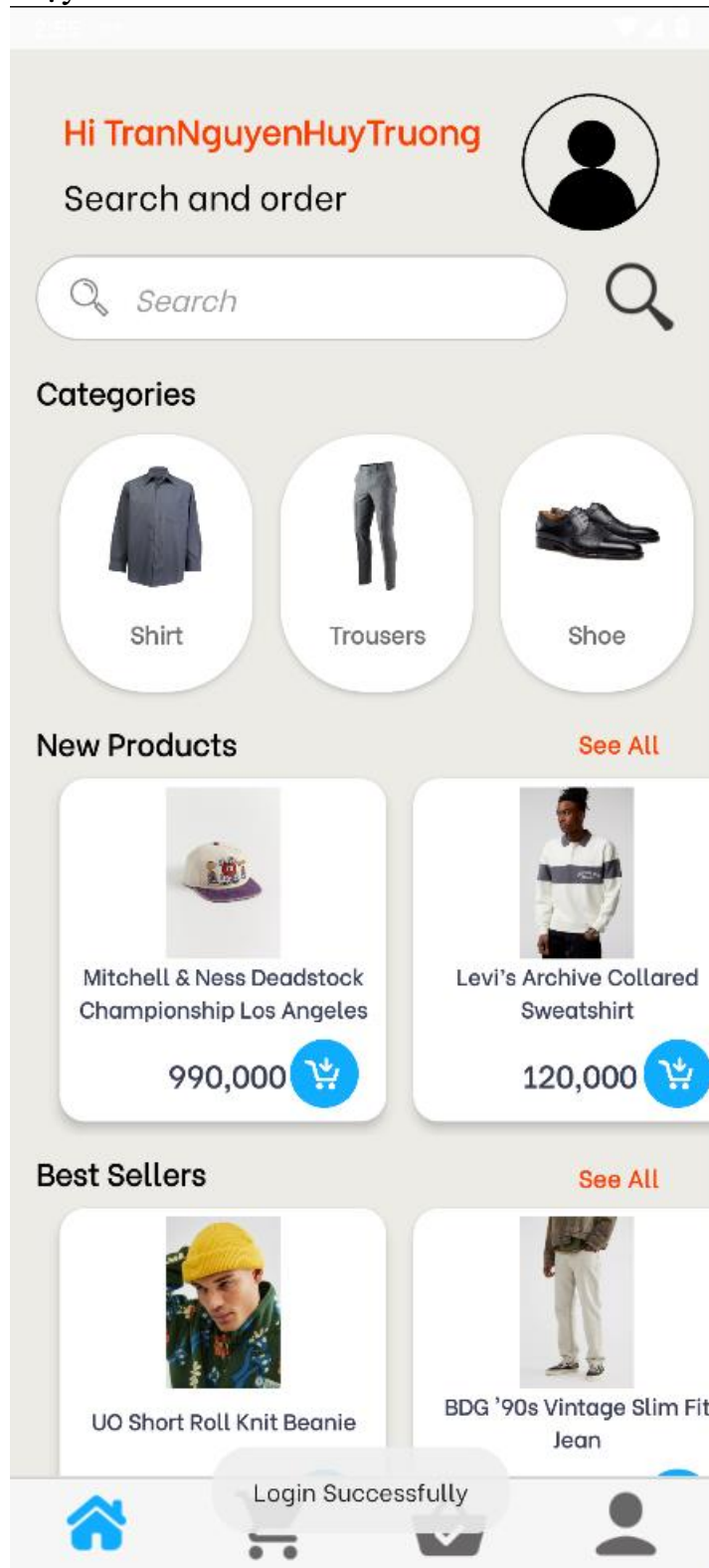
        clForgotPassword.setVisibility(View.VISIBLE);
    });
}
else{
    tvUserNameNotCorrect.setVisibility(View.VISIBLE);
    Toast.makeText(ForgotPasswordActivity.this, "Your
username is not correct!", Toast.LENGTH_SHORT).show();
}
}

@Override
public void onFailure(Call<String> call, Throwable t) {
    Log.e("===++", t.getMessage());
    progressDialog.dismiss();
}
});
}
}

```

### 5.5. Giao diện trang chính

Chức năng: List các danh mục sản phẩm. Các sản phẩm mới và các sản phẩm bán chạy



Phần code chính thực hiện:

```
@GET ("/newproduct")  
Call<List<Product>> getNewProduct () ;
```

```
@GET("bestsellers")
Call<List<Product>> getBestSellers();
```

```
private void LoadBestSellers() {
    LinearLayoutManager layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.HORIZONTAL, false);
    recyclerViewBestSellersList = findViewById(R.id.view3);
    recyclerViewBestSellersList.setLayoutManager(layoutManager);

    //GET API
    ProductAPI.productApi.getBestSellers().enqueue(new
Callback<List<Product>>() {
        @Override
        public void onResponse(Call<List<Product>> call,
Response<List<Product>> response) {
            List<Product> newProductsList = response.body();
            adapter3 = new ProductAdapter(newProductsList,
MainActivity.this);
            recyclerViewBestSellersList.setAdapter(adapter3);
        }

        @Override
        public void onFailure(Call<List<Product>> call, Throwable t) {
            Log.e("====", "Call API Get Best Sellers fail");
        }
    });
}

private void LoadNewProducts() {
    LinearLayoutManager layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.HORIZONTAL, false);
    recyclerViewNewProductList = findViewById(R.id.view2);
    recyclerViewNewProductList.setLayoutManager(layoutManager);

    //GET API
    ProductAPI.productApi.getNewProduct().enqueue(new
Callback<List<Product>>() {
        @Override
        public void onResponse(Call<List<Product>> call,
Response<List<Product>> response) {
            List<Product> newProductsList = response.body();
            adapter2 = new ProductAdapter(newProductsList,
MainActivity.this);
            recyclerViewNewProductList.setAdapter(adapter2);
        }

        @Override
        public void onFailure(Call<List<Product>> call, Throwable t) {
            Log.e("++++", t.getMessage());
            Log.e("====", "Call API Get New Products fail");
        }
    });
}

private void LoadCategories() {
    LinearLayoutManager layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.HORIZONTAL, false);
    recyclerViewCategoryList = findViewById(R.id.view1);
    recyclerViewCategoryList.setLayoutManager(layoutManager);
    //Get API
    CategoryAPI.categoryAPI.GetAllCategories().enqueue(new
Callback<List<Category>>() {
```



```

        @Override
        public void onResponse(Call<List<Category>> call,
Response<List<Category>> response) {
            List<Category> categoriesList = response.body();
            adapter = new CategoryAdapter(categoriesList, MainActivity.this);
            recyclerViewCategoryList.setAdapter(adapter);
        }

        @Override
        public void onFailure(Call<List<Category>> call, Throwable t) {
            Log.e("====", "Call API Get Categories fail");
        }
    });
}

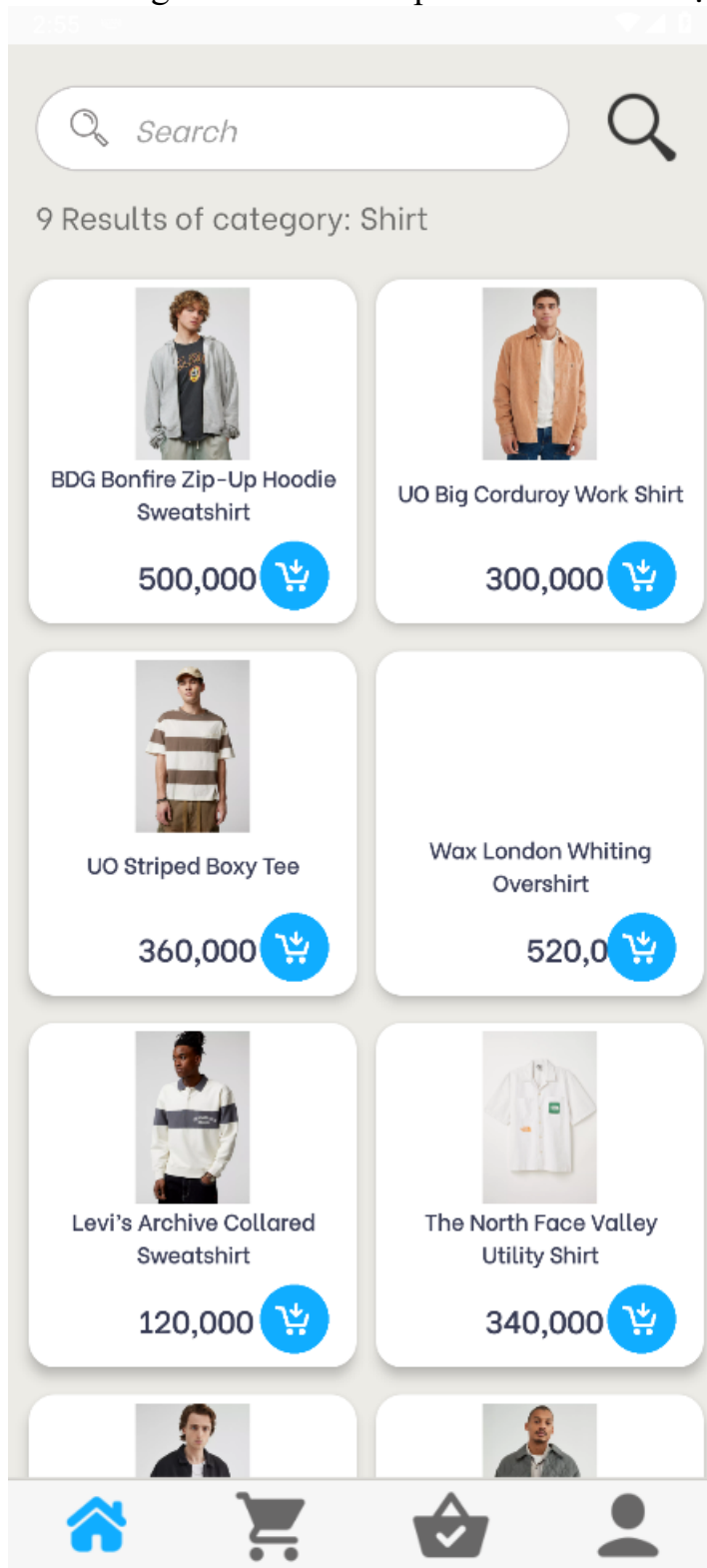
private void LoadUserInfor() {
    user =
ObjectSharedPreferences.getSavedObjectFromPreference(MainActivity.this,
"User", "MODE_PRIVATE", User.class);
    tvHiName.setText("Hi " + user.getUser_Name());

Glide.with(getApplicationContext()).load(user.getAvatar()).into(ivAvatar);
}

```

## 5.6. Giao diện sản phẩm theo danh mục

Chức năng: Danh sách sản phẩm theo danh mục sản phẩm



Các phần code chính:

```
private void loadProductCategory(Category category) {  
    tvResult.setText(category.getProduct().size() + " Results of category: " +  
category.getCategory_Name());  
    productAdapter=new
```

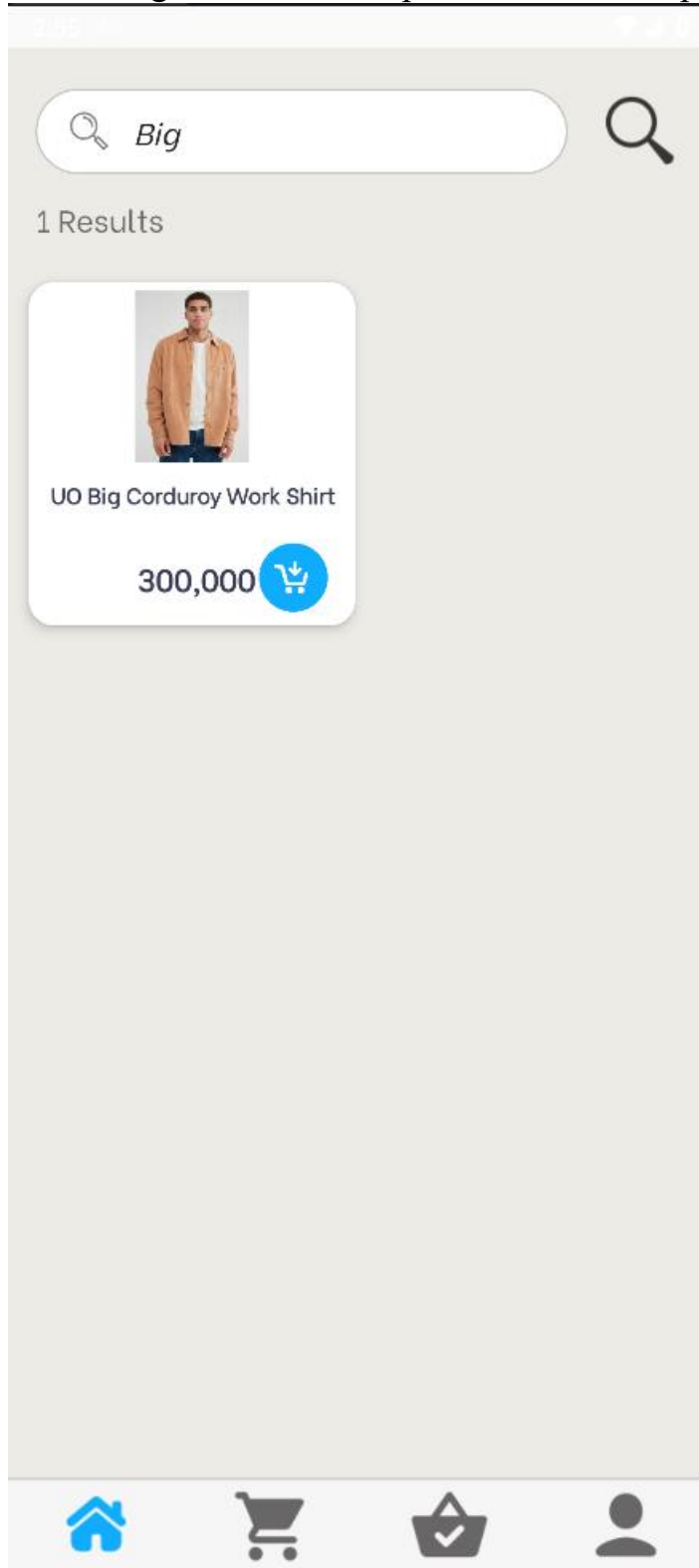
```

ProductAdapter(category.getProduct(),ProductsActivity.this);
    rcProduct.setHasFixedSize(true);
    GridLayoutManager layoutManager=new
GridLayoutManager(getApplicationContext(),2);
    rcProduct.setLayoutManager(layoutManager);
    rcProduct.setAdapter(productAdapter);
}
etSearch.setOnKeyListener(new View.OnKeyListener() {
    public boolean onKey(View v, int keyCode, KeyEvent event) {
        // If the event is a key-down event on the "enter" button
        if ((event.getAction() == KeyEvent.ACTION_DOWN) &&
            (keyCode == KeyEvent.KEYCODE_ENTER)) {
            // Perform action on key press
            Intent intent = new Intent(ProductsActivity.this,
ProductsActivity.class);
            intent.putExtra("searchContent", etSearch.getText().toString());
            intent.putExtra("category_id", "-1");
            startActivity(intent);
            return true;
        }
        return false;
    }
});

```

## 5.7. Giao diện tìm kiếm sản phẩm

Chức năng: Tìm kiếm sản phẩm theo tên sản phẩm



Các phần code chính:

```
@GET("/search")
Call<List<Product>> search(@Query("searchContent") String searchContent);
```

```
private void loadProductSearch() {
    ProductAPI.productApi.search(etSearch.getText().toString()).enqueue(new
```

```

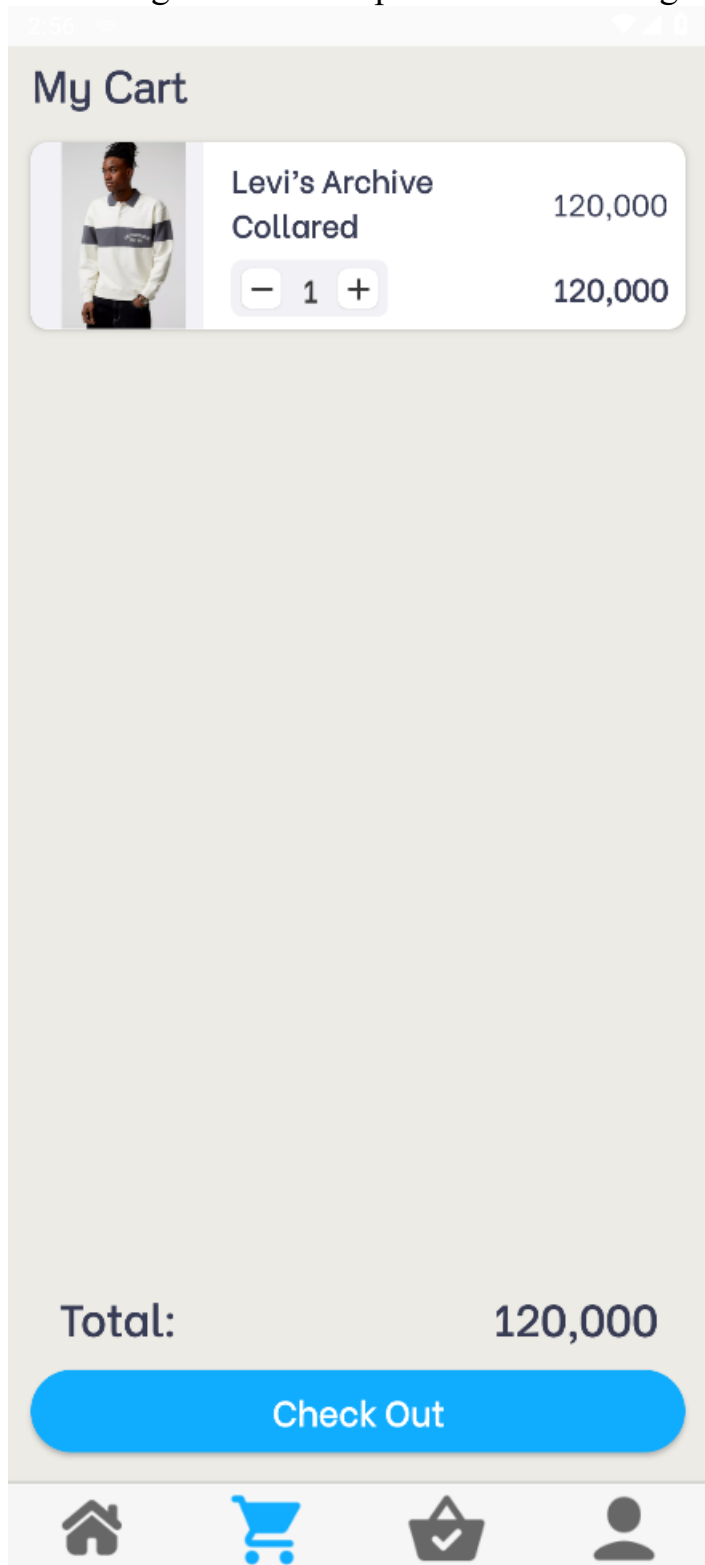
Callback<List<Product>>() {
    @Override
    public void onResponse(Call<List<Product>> call,
Response<List<Product>> response) {
        List<Product> newProductsList = response.body();
        tvResult.setText(newProductsList.size() + " Results");
        productAdapter=new
ProductAdapter(newProductsList,ProductsActivity.this);
        rcProduct.setHasFixedSize(true);
        GridLayoutManager layoutManager=new
GridLayoutManager(getApplicationContext(),2);
        rcProduct.setLayoutManager(layoutManager);
        rcProduct.setAdapter(productAdapter);
    }

    @Override
    public void onFailure(Call<List<Product>> call, Throwable t) {
        Log.e("====", "Call API Search fail");
    }
});
}

```

## 5.8. Giao diện giỏ hàng

Chức năng: List các sản phẩm đã thêm vào giỏ hàng



Các phần code chính:

```
holder.btnAdd.setOnClickListener(v -> {  
    User user = ObjectSharedPreferences.getSavedObjectFromPreference(context,  
    "User", "MODE_PRIVATE", User.class);  
    CartAPI.cartAPI.addToCart(user.getId(), product.getId(), 1).enqueue(new  
    Callback<Cart>() {
```

```

@Override
public void onResponse(Call<Cart> call, Response<Cart> response) {
    Cart cart = response.body();
    if (cart != null) {
        Toast.makeText(context.getApplicationContext(), "Thêm vào giỏ
thành công", Toast.LENGTH_SHORT).show();
    }
    else {
        Toast.makeText(context.getApplicationContext(), "Thêm vào giỏ
thất bại", Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onFailure(Call<Cart> call, Throwable t) {
    Toast.makeText(context.getApplicationContext(), "Call API Add to
cart fail", Toast.LENGTH_SHORT).show();
}
});
});

```

```

private void LoadCart() {
    LinearLayoutManager layoutManager = new
LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false);
    recyclerViewCart = findViewById(R.id.view);
    recyclerViewCart.setLayoutManager(layoutManager);
    User user =
ObjectSharedPreferences.getSavedObjectFromPreference(CartActivity.this,
"User", "MODE_PRIVATE", User.class);
    CartItemInterface cartItemInterface = this;
    //GET API
    CartAPI.cartAPI.cartOfUser(user.getId()).enqueue(new
Callback<List<Cart>>() {
        @Override
        public void onResponse(Call<List<Cart>> call,
Response<List<Cart>> response) {
            listCart = response.body();
            if (listCart.isEmpty()) {
                clCartIsEmpty.setVisibility(View.VISIBLE);
                clCart.setVisibility(View.GONE);
            }
            else {
                Log.e("++==", String.valueOf(listCart));

                cartAdapter = new CartAdapter(cartItemInterface,
listCart, CartActivity.this);
                recyclerViewCart.setAdapter(cartAdapter);

                int Total= 0;
                for (Cart y: listCart) {
                    Total += y.getCount() * y.getProduct().getPrice();
                }
                Locale localeEN = new Locale("en", "EN");
                NumberFormat en = NumberFormat.getInstance(localeEN);
                tvTotalPrice.setText(en.format(Total));
            }
        }
    });
    @Override
    public void onFailure(Call<List<Cart>> call, Throwable t) {

```

```
        Log.e("===", "Call API Cart of user fail");  
    }  
    });  
}
```



## 5.9. Giao diện đặt hàng

Chức năng: List các sản phẩm cần thanh toán. Có các tùy chọn thanh toán khi nhận hàng hoặc thanh toán bằng zalo pay

2:56

< Checkout


Shipping address

TranNguyenHuyTruong (0779847539) [Change](#)

19/25

Việt Nam

Product Item

 Levi's Archive Collared Sweatshirt  
120,000 x 1

Total: 120,000

☐ Pay on Delivery

☐ Pay With ZaloPay

Total Price 120,000

Place Order

Các phần code chính:

```

@FormUrlEncoded
@POST("/addtocart")
Call<Cart> addToCart(@Field("user_id") String user_id, @Field("product_id")
int product_id, @Field("count") int count);

```

```

private void radioButtonClick() {
    tvPayOnDelivery.setOnClickListener(v -> {
        rbPayOnDelivery.setChecked(true);
    });
    tvPayWithZaloPay.setOnClickListener(v -> {
        rbPayWithZaloPay.setChecked(true);
    });
}

private void btnPlaceOrderClick() {
    btnPlaceOrder.setOnClickListener(v -> {
        if (rbPayOnDelivery.isChecked()) {
            newOrder("Pay on Delivery");
        } else if (rbPayWithZaloPay.isChecked()) {
            StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
            // ZaloPay SDK Init
            ZaloPaySDK.init(2553, Environment.SANDBOX);
            CreateOrder orderApi = new CreateOrder();

            try {
                String amount =
tvTotalPrice.getText().toString().replace(",", "");
                JSONObject data = orderApi.createOrder(amount);
                Log.e("Amount", amount);
                String code = data.getString("return_code");

                if (code.equals("1")) {
                    Dialog dialog = new Dialog(CheckOutActivity.this);
                    String token = data.getString("zp_trans_token");

                    ZaloPaySDK.getInstance().payOrder(CheckOutActivity.this, token,
"demozpdk://app", new PayOrderListener() {
                        @Override
                        public void onPaymentSucceeded(String s, String
s1, String s2) {
                            newOrder("Pay with ZaloPay");
                        }

                        @Override
                        public void onPaymentCanceled(String s, String
s1) {
                            dialog.setContentView(R.layout.dialog_pay_with_zalopay);

                            dialog.getWindow().setLayout(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.WRAP_CONTENT);

                            dialog.getWindow().getAttributes().windowAnimations = R.style.animation;
                            dialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
                            dialog.show();

                            Button btnOk =
dialog.findViewById(R.id.btnOk);

```

```

        btnOk.setOnClickListener(v1 -> {
            dialog.dismiss();
        });
    }

    @Override
    public void onPaymentError(ZaloPayError
zaloPayError, String s, String sl) {

dialog.setContentview(R.layout.dialog_pay_with_zalopay);

dialog.getWindow().setLayout(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.WRAP_CONTENT);

dialog.getWindow().getAttributes().windowAnimations = R.style.animation;
dialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
dialog.show();
Button btnOk =
dialog.findViewById(R.id.btnOk);
        btnOk.setOnClickListener(v1 -> {
            dialog.dismiss();
        });
    }
}

    } catch (Exception e) {
        e.printStackTrace();
    }
}
else {
    Toast.makeText(getApplicationContext(), "Please choose a
payment method", Toast.LENGTH_LONG).show();
}
});

}

    private void newOrder(String method) {
        Address address =
ObjectSharedPreferences.getSavedObjectFromPreference(CheckOutActivity.this,
"address", "MODE_PRIVATE", Address.class);
        if (address == null) {
            Toast.makeText(CheckOutActivity.this(getApplicationContext(),
"Please add your address before place order", Toast.LENGTH_SHORT).show();
        } else {
            User user =
ObjectSharedPreferences.getSavedObjectFromPreference(CheckOutActivity.this,
"User", "MODE_PRIVATE", User.class);
            OrderAPI.orderAPI.placeOrder(user.getId(),
tvUserName.getText().toString(),
tvPhoneNumber.getText().toString().replace("(", "").replace(")", ""),
tvAddress.getText().toString(), method).enqueue(new Callback<Order>() {
                @Override
                public void onResponse(Call<Order> call, Response<Order>
response) {
                    Order order = response.body();
                    if (order != null) {
                        placeOrder.setVisibility(View.GONE);
                        placeOrderSuccess.setVisibility(View.VISIBLE);
                        Button btnContinueShopping =

```

```

findViewById(R.id.btnContinueShopping);
        btnContinueShopping.setOnClickListener(v1 -> {
            placeOrder.setVisibility(View.VISIBLE);
            placeOrderSuccess.setVisibility(View.GONE);
            startActivity(new Intent(CheckOutActivity.this,
MainActivity.class));
        });
    }

    @Override
    public void onFailure(Call<Order> call, Throwable t) {
    }
});
}
}

```

## 5.10. Giao diện đổi địa chỉ

Chức năng: Thay đổi địa chỉ đặt hàng

Full Name

TranNguyenHuyTruong

Phone Number

0779847539

Address

19/25

Save Address

Các phần code chính:

```
private void saveAddressClick() {  
    saveAddress.setOnClickListener(v -> {  
        if (TextUtils.isEmpty(etFullName.getText().toString())) {  
            Toast.makeText(AddressActivity.this, "Full Name is required",  
                Toast.LENGTH_SHORT).show();  
            return;  
        }  
        if (TextUtils.isEmpty(etPhoneNumber.getText().toString())) {  
            Toast.makeText(AddressActivity.this, "Phone Number is required",  
                Toast.LENGTH_SHORT).show();  
            return;  
        }  
        if (TextUtils.isEmpty(etAddress.getText().toString())) {  
            Toast.makeText(AddressActivity.this, "Address is required",  
                Toast.LENGTH_SHORT).show();  
            return;  
        }  
        saveAddress();  
    });  
}
```

```

"Please enter your full name", Toast.LENGTH_SHORT).show();
    etFullName.requestFocus();
    return;
}

    if (TextUtils.isEmpty(etPhoneNumber.getText().toString())){
        Toast.makeText(AddressActivity.this.getContext(),
"Please enter your phone number", Toast.LENGTH_SHORT).show();
        etPhoneNumber.requestFocus();
        return;
    }
    if (TextUtils.isEmpty(etAddress.getText().toString())){
        Toast.makeText(AddressActivity.this.getContext(),
"Please enter your address", Toast.LENGTH_SHORT).show();
        etAddress.requestFocus();
        return;
    }
    Address address = new Address(etFullName.getText().toString(),
etPhoneNumber.getText().toString(), etAddress.getText().toString());

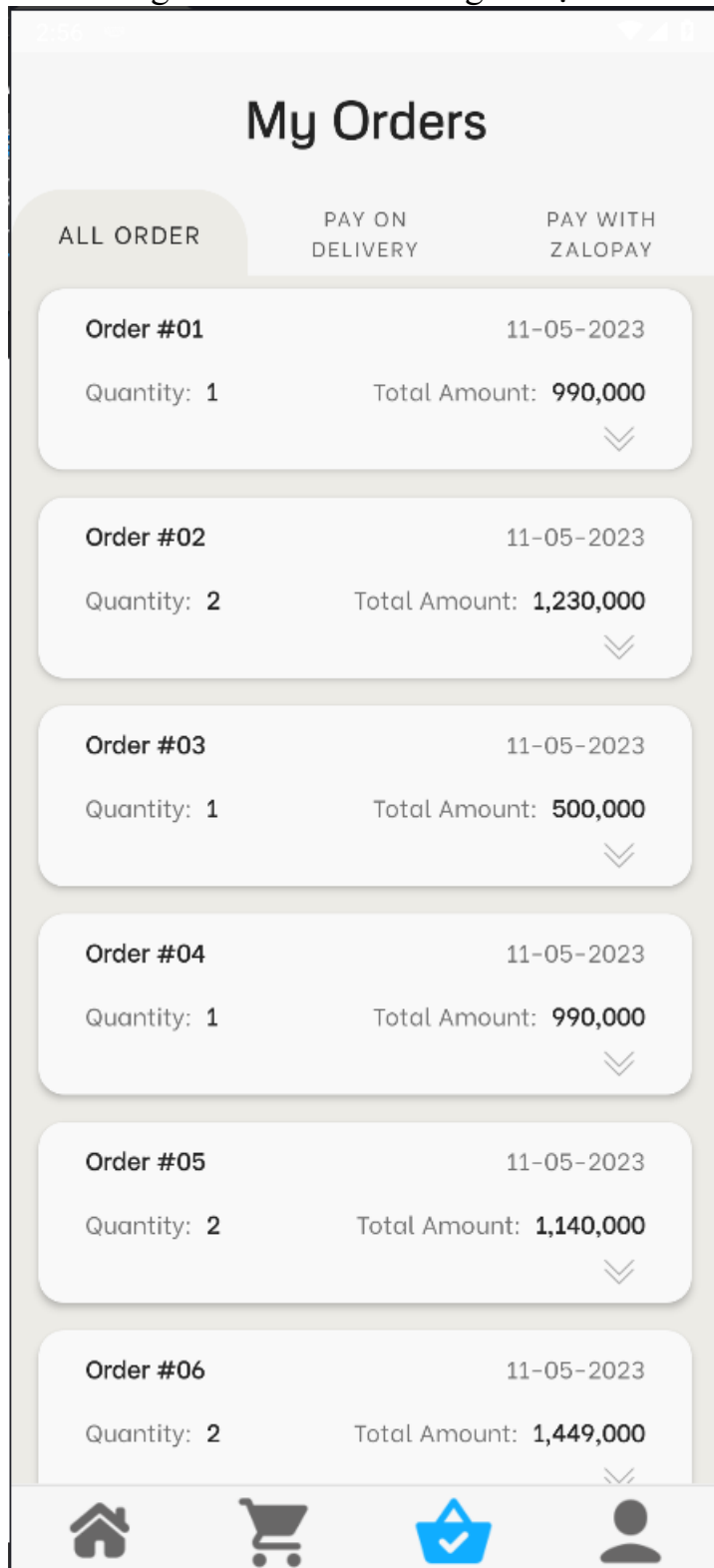
ObjectSharedPreferences.saveObjectToSharedPreference(AddressActivity.this,
"address", "MODE_PRIVATE", address);
    startActivity(new Intent(AddressActivity.this,
CheckoutActivity.class));
    finish();
});
}

private void LoadInformation() {
    Address address =
ObjectSharedPreferences.getSavedObjectFromPreference(AddressActivity.this,
"address", "MODE_PRIVATE", Address.class);
    if(address != null){
        etFullName.setText(address.getFullName());
        etPhoneNumber.setText(address.getPhoneNumber());
        etAddress.setText(address.getAddress());
    }
}
}

```

## 5.11. Giao diện liệt kê đơn hàng

Chức năng: List tất cả đơn hàng đã đặt mua



Các phần code chính:

```
private void LoadData() {  
    LinearLayoutManager layoutManager = new
```

```

LinearLayoutManager(getContext(), LinearLayoutManager.VERTICAL, false);
        rvAllOrder.setLayoutManager(linearLayoutManager);
        User user =
ObjectSharedPreferences.getSavedObjectFromPreference(getContext(), "User",
"MODE_PRIVATE", User.class);
        OrderAPI.orderAPI.getOrderById(user.getId()).enqueue(new
Callback<List<Order>>() {
            @Override
            public void onResponse(Call<List<Order>> call,
Response<List<Order>> response) {
//                Log.e("=====", response.body().toString());
                orderList = response.body();
                orderAdapter = new OrderAdapter(orderList,
getContext().getApplicationContext());
                rvAllOrder.setAdapter(orderAdapter);
            }

            @Override
            public void onFailure(Call<List<Order>> call, Throwable t) {

            }

        });
    }

private void LoadData() {
        LinearLayoutManager linearLayoutManager = new
LinearLayoutManager(getContext(), LinearLayoutManager.VERTICAL, false);
        rvAllOrder.setLayoutManager(linearLayoutManager);
        User user =
ObjectSharedPreferences.getSavedObjectFromPreference(getContext(), "User",
"MODE_PRIVATE", User.class);
        OrderAPI.orderAPI.getOrderByIdAndPaymentMethod(user.getId(), "Pay
with ZaloPay").enqueue(new Callback<List<Order>>() {
            @Override
            public void onResponse(Call<List<Order>> call,
Response<List<Order>> response) {
//                Log.e("=====", response.body().toString());
                orderList = response.body();
                orderAdapter = new OrderAdapter(orderList,
getContext().getApplicationContext());
                rvAllOrder.setAdapter(orderAdapter);
            }

            @Override
            public void onFailure(Call<List<Order>> call, Throwable t) {

            }

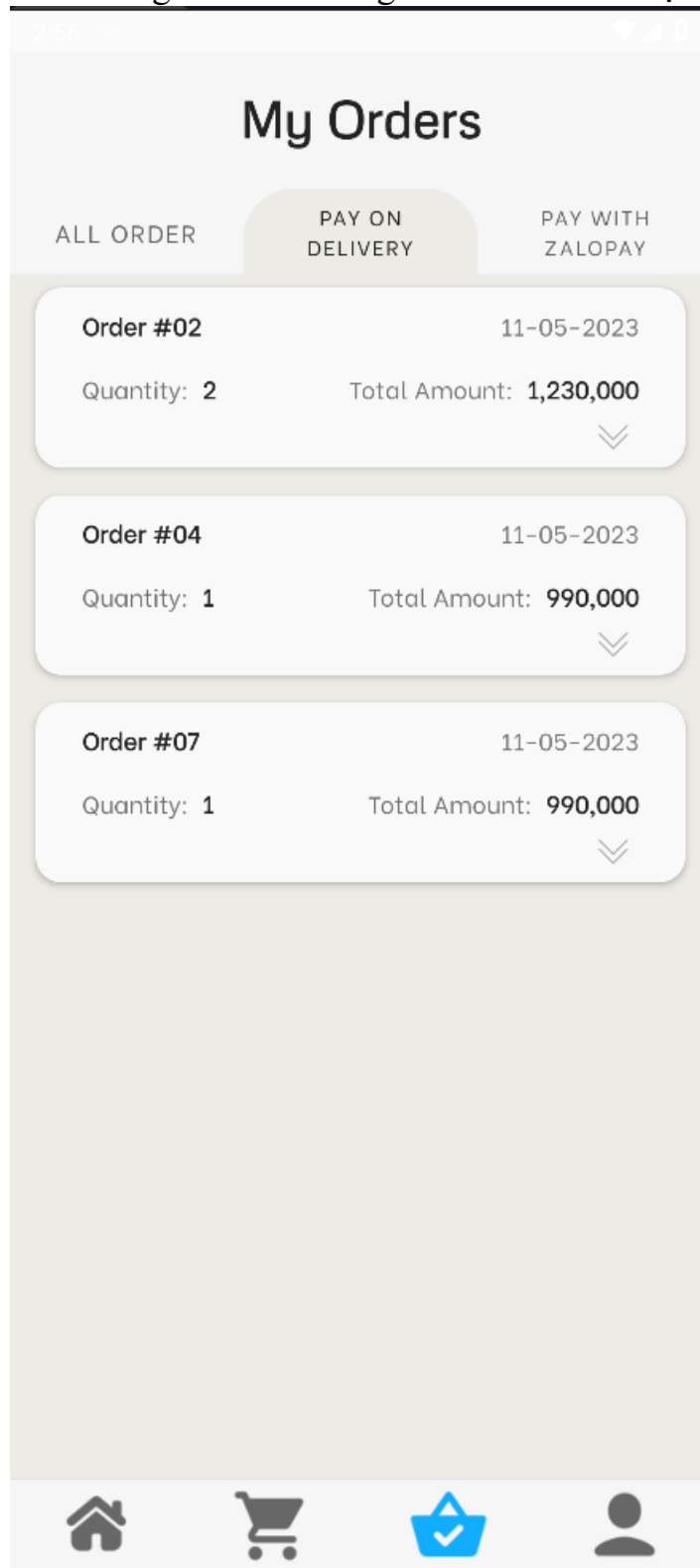
        });
    }
}

```



## 5.12. Giao diện liệt kê đơn hàng thanh toán tiền mặt

Chức năng: List đơn hàng thanh toán tiền mặt



Các phần code chính:

```

private void LoadData() {
    LinearLayoutManager layoutManager = new
LinearLayoutManager(getContext(), LinearLayoutManager.VERTICAL, false);
    rvAllOrder.setLayoutManager(layoutManager);
    User user =
ObjectSharedPreferences.getSavedObjectFromPreference(getContext(), "User",
"MODE_PRIVATE", User.class);
    OrderAPI.orderAPI.getOrderByIdAndPaymentMethod(user.getId(), "Pay
on Delivery").enqueue(new Callback<List<Order>>() {
        @Override
        public void onResponse(Call<List<Order>> call,
Response<List<Order>> response) {
//            Log.e("=====", response.body().toString());
            orderList = response.body();
            orderAdapter = new OrderAdapter(orderList,
getContext().getApplicationContext());
            rvAllOrder.setAdapter(orderAdapter);
        }

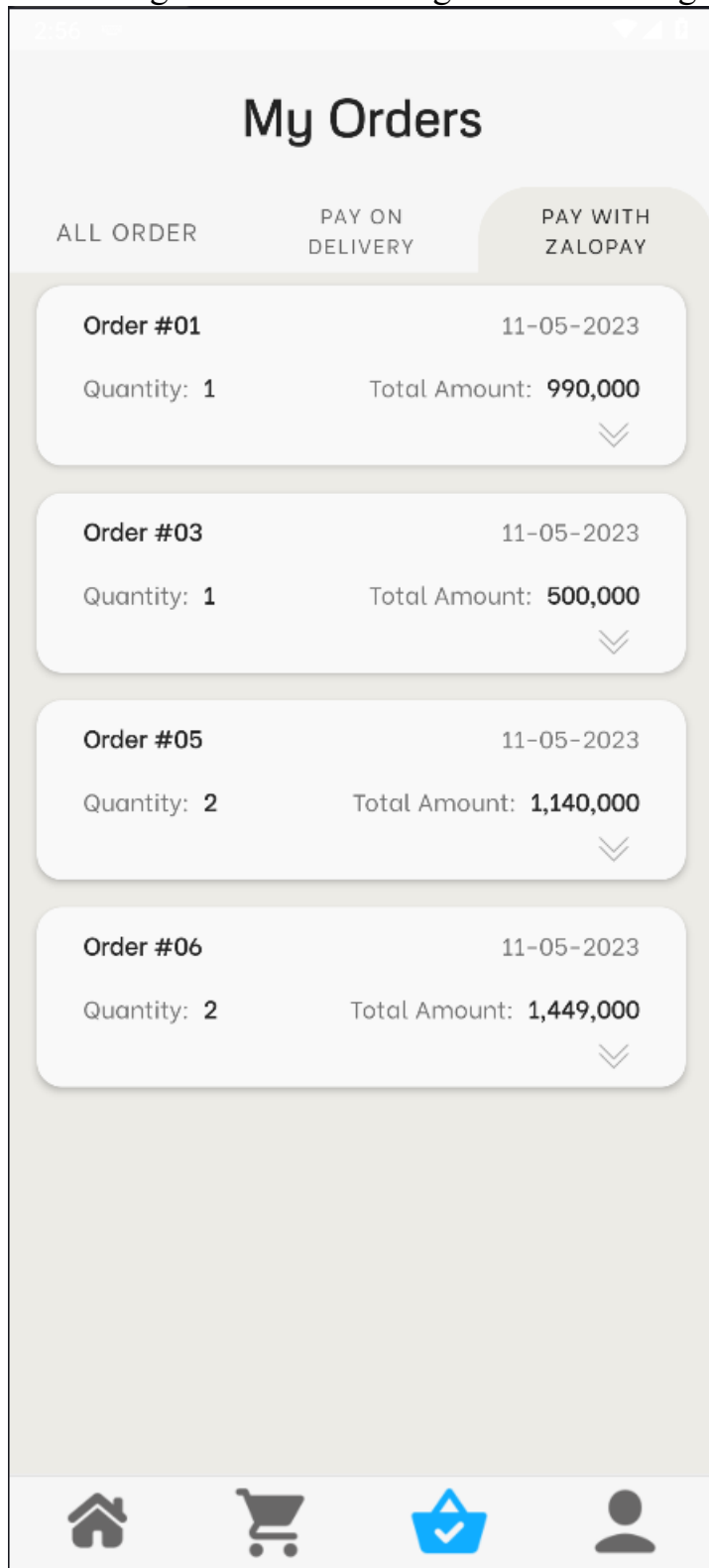
        @Override
        public void onFailure(Call<List<Order>> call, Throwable t) {

        }
    });
}

```

### 5.13. Giao diện liệt kê đơn hàng thanh toán bằng zalo pay

Chức năng: List các đơn hàng thanh toán bằng zalopay



Các phần code chính:

```
private void LoadData() {  
    LinearLayoutManager layoutManager = new
```

```

LinearLayoutManager(getContext(), LinearLayoutManager.VERTICAL, false);
    rvAllOrder.setLayoutManager(linearLayoutManager);
    User user =
ObjectSharedPreferences.getSavedObjectFromPreference(getContext(), "User",
"MODE_PRIVATE", User.class);
    OrderAPI.orderAPI.getOrderByIdAndPaymentMethod(user.getId(), "Pay
with ZaloPay").enqueue(new Callback<List<Order>>() {
        @Override
        public void onResponse(Call<List<Order>> call,
Response<List<Order>> response) {
//            Log.e("====", response.body().toString());
            orderList = response.body();
            orderAdapter = new OrderAdapter(orderList,
getContext().getApplicationContext());
            rvAllOrder.setAdapter(orderAdapter);
        }

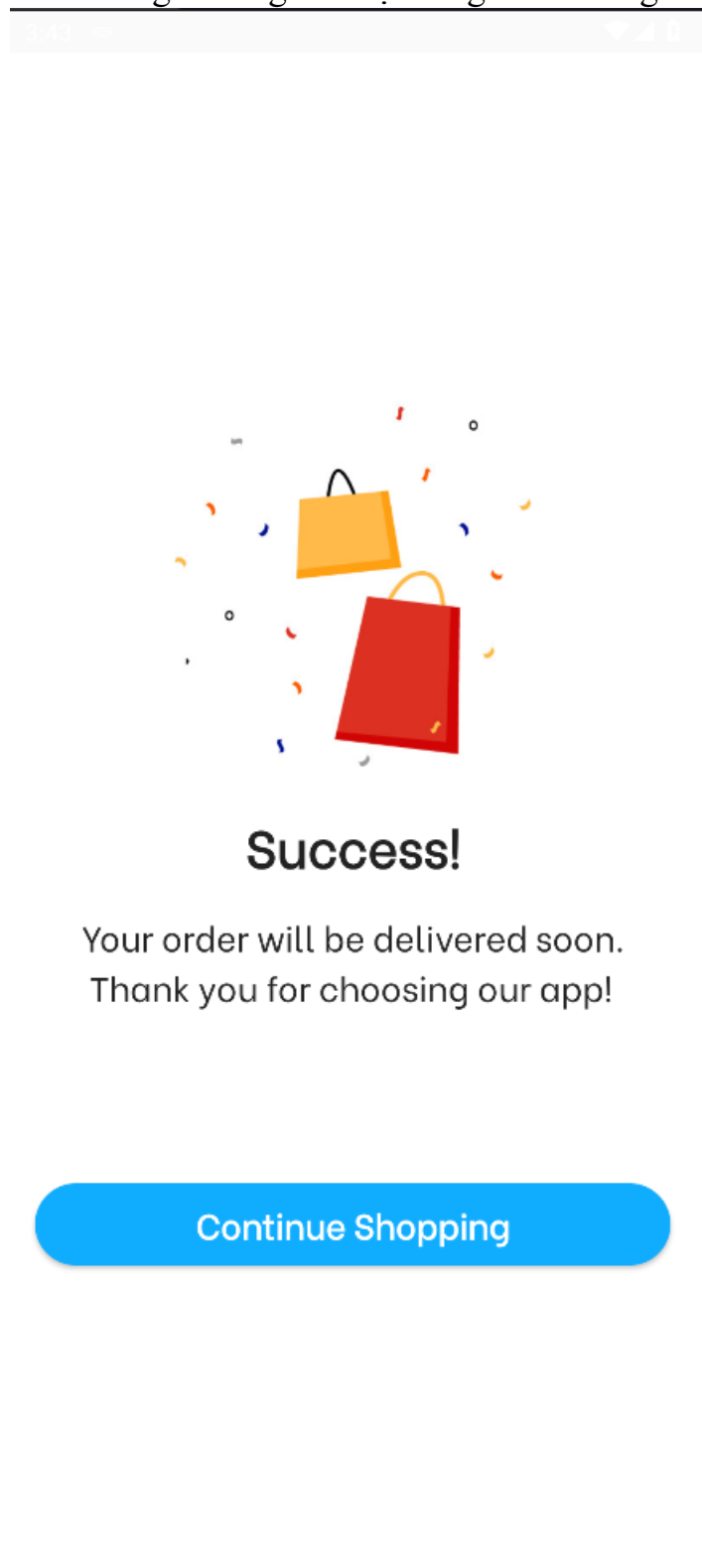
        @Override
        public void onFailure(Call<List<Order>> call, Throwable t) {

        }
    });
}

```

## 5.14. Giao diện đặt hàng thành công

Chức năng: Thông báo đặt hàng thành công



Các phần code chính:

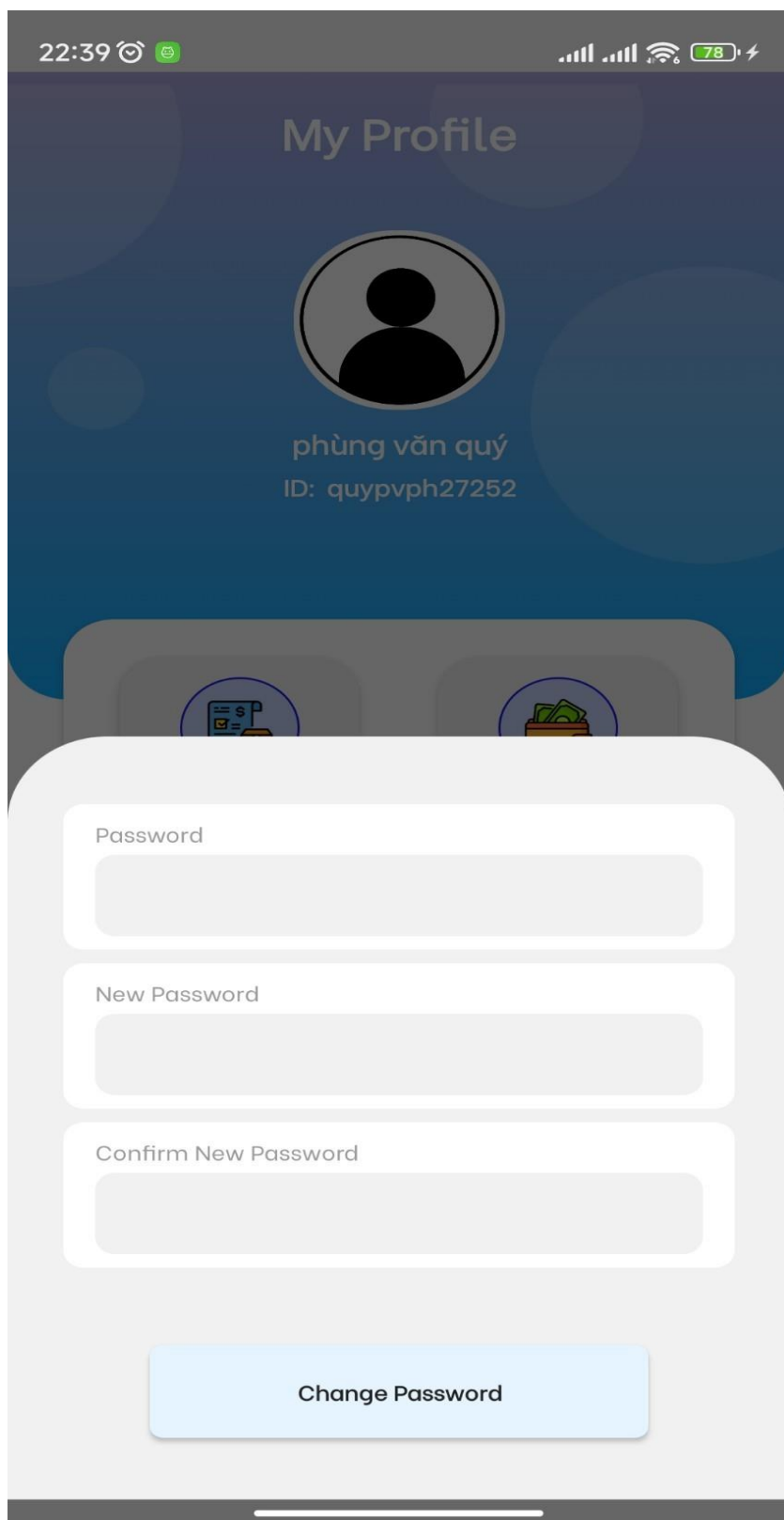
```

@Override
public void onResponse(Call<Order> call, Response<Order> response) {
    Order order = response.body();
    if (order != null) {
        placeOrder.setVisibility(View.GONE);
        placeOrderSuccess.setVisibility(View.VISIBLE);
        Button btnContinueShopping = findViewById(R.id.btnContinueShopping);
        btnContinueShopping.setOnClickListener(v1 -> {
            placeOrder.setVisibility(View.VISIBLE);
            placeOrderSuccess.setVisibility(View.GONE);
            startActivity(new Intent( packageContext: CheckoutActivity.this, MainActivity.class));
        });
    }
}
}

```

### 5.15. Giao diện Profile

Chức năng: Xem thông tin người dùng. Tổng đơn hàng và tổng chi phí đã đặt hàng



The screenshot shows a mobile application interface. At the top, the status bar displays the time 22:39, a location icon, a WhatsApp icon, signal strength bars, Wi-Fi, and a battery level of 78%. The main header is 'My Profile' in a light blue font. Below the header is a circular profile picture placeholder with a black silhouette. Underneath the profile picture, the name 'phùng văn quý' and the ID 'ID: quyvpvh27252' are displayed. Below the profile information, there are two icons: a shopping cart and a wallet. A white modal dialog is open in the foreground, containing three input fields labeled 'Password', 'New Password', and 'Confirm New Password'. At the bottom of the modal is a blue button labeled 'Change Password'.

22:39 📍 📞

My Profile

phùng văn quý  
ID: quyvpvh27252

🛒 🏠

Password

New Password

Confirm New Password

Change Password

## Các phần code chính:

```
private void btnLogoutClick() {
    btnLogout.setOnClickListener(v -> {
        user =
ObjectSharedPreferences.getSavedObjectFromPreference(UserActivity.this,
"User", "MODE_PRIVATE", User.class);
        if (user.getPassword()==null) {
            googleSignInClient.signOut().addOnCompleteListener(new
OnCompleteListener<Void>() {
                @Override
                public void onComplete(Task<Void> task) {
                    finish();

ObjectSharedPreferences.saveObjectToSharedPreference(UserActivity.this,
"User", "MODE_PRIVATE", null);
                    startActivity(new
Intent(UserActivity.this, LoginActivity.class));
                }
            });
        }
        else{

ObjectSharedPreferences.saveObjectToSharedPreference(UserActivity.this,
"User", "MODE_PRIVATE", null);
            startActivity(new Intent(UserActivity.this,
LoginActivity.class));
        }

    });
}

private void LoadData() {
    user =
ObjectSharedPreferences.getSavedObjectFromPreference(UserActivity.this,
"User", "MODE_PRIVATE", User.class);
    UserAPI.userApi.Login(user.getId(), user.getPassword()).enqueue(new
Callback<User>() {
        @Override
        public void onResponse(Call<User> call, Response<User> response) {
            user = response.body();

ObjectSharedPreferences.saveObjectToSharedPreference(UserActivity.this,
"User", "MODE_PRIVATE", user);

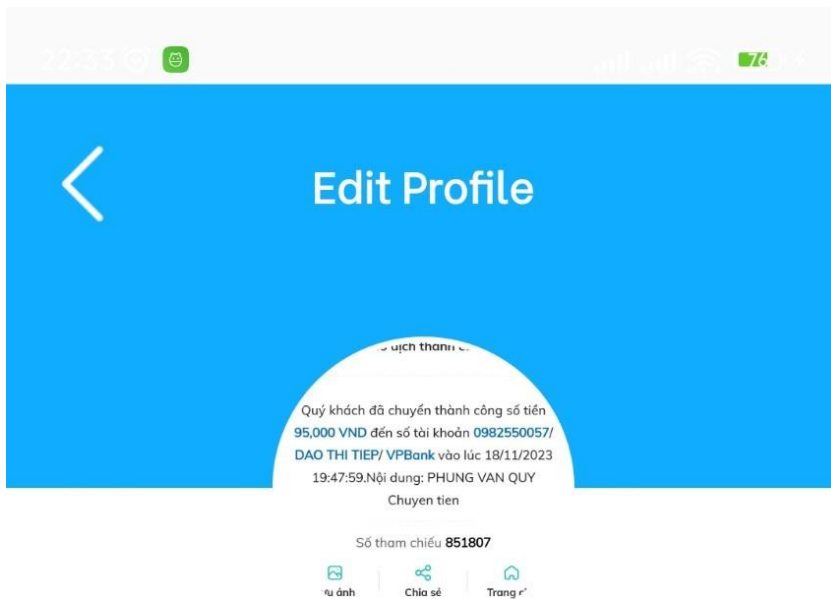
Glide.with(getApplicationContext()).load(user.getAvatar()).into(ivAvatar);
            tvFullName.setText(user.getUser_Name());
            tvId.setText(user.getId());
            tvTotalOrder.setText(user.getOrder().size()+"");
            int totalPrice = 0;
            for (Order o : user.getOrder()){
                totalPrice += o.getTotal();
            }
            Locale localeEN = new Locale("en", "EN");
            NumberFormat en = NumberFormat.getInstance(localeEN);
            tvTotalPrice.setText(en.format(totalPrice));
            tvPhone.setText(user.getPhone_Number());
            tvAddress.setText(user.getAddress());
            tvEmail.setText(user.getEmail());
            if(user.getLogin_Type().equals("google")){
```



```
        tvChangePassword.setVisibility(View.GONE);
    }
    @Override
    public void onFailure(Call<User> call, Throwable t) {
    }
    });
}
```

### **5.16. Giao diện edit profile**

Chức năng: Cập nhật thông tin người dùng



Change Picture

Full Name:

QUY

Email:

preci0us.03.pt@gmail.com

Phone Number:

0986669684

Address:

khu 1 mạn lạn thanh ba phú thọ

Update

### Các phần code chính:

```
private void btnUpdateClick() {
    btnUpdate.setOnClickListener(v -> {

        if(etFullName.getText().toString().isEmpty()){
            tvError.setText("Please enter your name!");
            return;
        }
        if (etEmail.getText().toString().isEmpty()){
            tvError.setText("Please enter your email!");
            return;
        }
        if(etPhoneNumber.getText().toString().isEmpty()){
            tvError.setText("Please enter your phone number!");
            return;
        }
        if (etAddress.getText().toString().isEmpty()){
            tvError.setText("Please enter your address!");
            return;
        }
        user =
ObjectSharedPreferences.getSavedObjectFromPreference(EditProfileActivity.this
, "User", "MODE_PRIVATE", User.class);
        RequestBody userId =
RequestBody.create(user.getId(), MediaType.parse("multipart/form-data"));
        RequestBody fullName =
RequestBody.create(etFullName.getText().toString(),
MediaType.parse("multipart/form-data"));
        RequestBody email = RequestBody.create(etEmail.getText().toString(),
MediaType.parse("multipart/form-data"));
        RequestBody phoneNumber =
RequestBody.create(etPhoneNumber.getText().toString(),
MediaType.parse("multipart/form-data"));
        RequestBody address =
RequestBody.create(etAddress.getText().toString(),
MediaType.parse("multipart/form-data"));
        MultipartBody.Part avatar = null;
        if(mUri!=null){
            String IMAGE_PATH = RealPathUtil.getRealPath(this, mUri);
            Log.e("ffff", IMAGE_PATH);
            File file = new File(IMAGE_PATH);
            RequestBody requestFile = RequestBody.create(file,
MediaType.parse("multipart/form-data"));
            avatar=MultipartBody.Part.createFormData("avatar",
file.getName(), requestFile);
        }

        progressDialog = new ProgressDialog(EditProfileActivity.this);
        progressDialog.setMessage("Loading..."); // Setting Message
        progressDialog.setTitle("Update Profile"); // Setting Title
        progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER); //
Progress Dialog Style Spinner
        progressDialog.show(); // Display Progress Dialog
        progressDialog.setCancelable(false);
        UserAPI.userApi.update(userId,avatar, fullName, email, phoneNumber,
address).enqueue(new Callback<User>() {
```

```

        @Override
        public void onResponse(Call<User> call, Response<User> response)
        {
            User userUpdate = response.body();
            if(userUpdate!=null){
                progressDialog.dismiss();

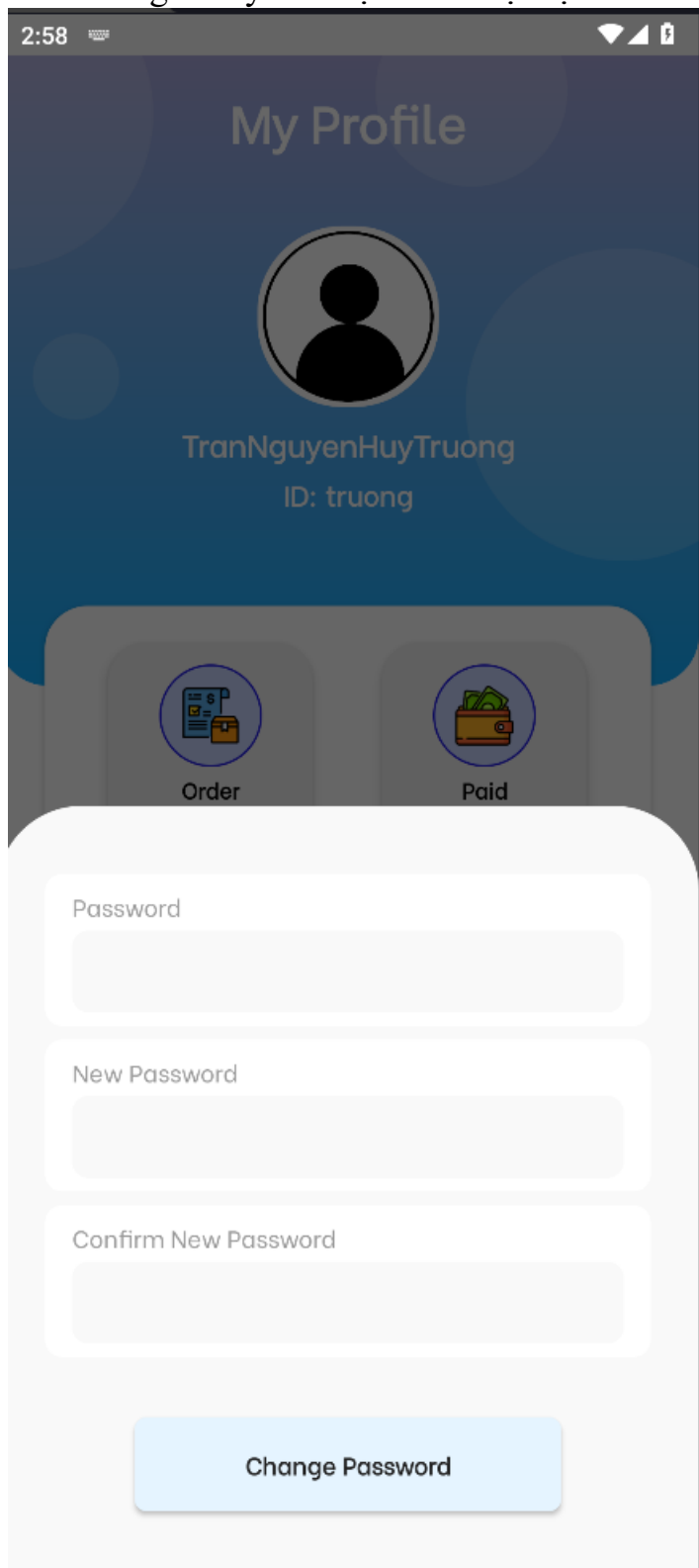
ObjectSharedPreferences.saveObjectToSharedPreference(EditProfileActivity.this
, "User", "MODE_PRIVATE", userUpdate);
                startActivity(new Intent(EditProfileActivity.this,
UserActivity.class));
                finish();
            }
        }

        @Override
        public void onFailure(Call<User> call, Throwable t) {
            progressDialog.dismiss();
            Log.e("===", "call fail + " + t.getMessage());
        }
    });
}

```

### 5.17. Giao diện thay đổi mật khẩu

Chức năng: Thay đổi mật khẩu hiện tại



The screenshot shows a mobile application interface. At the top, the status bar displays the time 2:58 and signal icons. The main screen is titled "My Profile" and features a circular profile picture placeholder. Below the profile picture, the name "TranNguyenHuyTruong" and the ID "ID: trung" are displayed. At the bottom of the main screen, there are two buttons: "Order" and "Paid". A modal dialog is open in the foreground, containing three text input fields labeled "Password", "New Password", and "Confirm New Password". Below these fields is a blue button labeled "Change Password".

Các phần code chính:

```
private void tvChangePasswordClick() {  
    tvChangePassword.setOnClickListener(v -> {
```

```

        Dialog dialog = new Dialog(UserActivity.this);
        dialog.setContentView(R.layout.dialog_change_password);
        dialog.getWindow().setLayout(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
        dialog.getWindow().getAttributes().windowAnimations =
R.style.animation;
        dialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
        dialog.getWindow().setGravity(Gravity.BOTTOM);
        dialog.show();
        //Anh xa dialog
        ConstraintLayout clChangePassword =
dialog.findViewById(R.id.clChangePassword);
        ConstraintLayout clChangePasswordSuccess =
dialog.findViewById(R.id.clChangePasswordSuccess);
        EditText etOldPassword = dialog.findViewById(R.id.etOldPassword);
        EditText etNewPassword = dialog.findViewById(R.id.etNewPassword);
        EditText etReNewPassword = dialog.findViewById(R.id.etReNewPassword);
        TextView tvErrorChangePassword =
dialog.findViewById(R.id.tvErrorChangePassword);
        Button btnChangePassword =
dialog.findViewById(R.id.btnChangePassword);
        Button btnOK = dialog.findViewById(R.id.btnOk);
        //====
        btnChangePassword.setOnClickListener(v1 -> {
            String password = etOldPassword.getText().toString();
            String newPassword = etNewPassword.getText().toString();
            String reNewPassword = etReNewPassword.getText().toString();
            user =
ObjectSharedPreferences.getSavedObjectFromPreference(UserActivity.this,
"User", "MODE_PRIVATE", User.class);
            if(password.equals(user.getPassword())){
                if(newPassword.equals(reNewPassword)){
                    UserAPI.userApi.changePassword(user.getId(),
newPassword).enqueue(new Callback<String>() {
                        @Override
                        public void onResponse(Call<String> call,
Response<String> response) {
                            String pass = response.body();
                            if(pass!= null){
                                clChangePassword.setVisibility(View.GONE);
                                clChangePasswordSuccess.setVisibility(View.VISIBLE);
                                user.setPassword(pass);
                                ObjectSharedPreferences.saveObjectToSharedPreference(UserActivity.this,
"User", "MODE_PRIVATE", user);
                                btnOK.setOnClickListener(v2 -> {
                                    dialog.dismiss();
                                });
                            }
                        }
                        @Override
                        public void onFailure(Call<String> call, Throwable t)
{
                            }
                    });
                }else {
                    tvErrorChangePassword.setText("New password and confirm
password do not match!");

```

```
        }  
    } else {  
        tvErrorChangePassword.setText("Your password is not  
correct");  
    }  
    });  
});  
}
```



## CÁC GIAO DIỆN ADMIN ĐƯỢC LÀM TỪ PAGEVIEW:

```
public class AdminActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_admin);

        // Find the WebView by its unique ID
        WebView webView = findViewById(R.id.webview);

        // loading https://www.geeksforgeeks.org url in the WebView.
        webView.loadUrl("http://172.16.2.26:8080/signin-admin");
        // this will enable the javascript.
        webView.getSettings().setJavaScriptEnabled(true);

        // WebViewClient allows you to handle
        // onPageFinished and override Url loading.
        //
        webView.setWebViewClient(new WebViewClient());
        webView.setWebViewClient(new WebViewClient()
        {
            @Override
            public void onPageStarted(WebView view, String url, Bitmap
favicon) {
                super.onPageStarted(view, url, favicon);
                //Here check whether the url is after logged in
                if(url.equals("http://192.168.1.7:8080/logout-admin")){
                    startActivity(new Intent(AdminActivity.this,
LoginActivity.class));
                }
            }

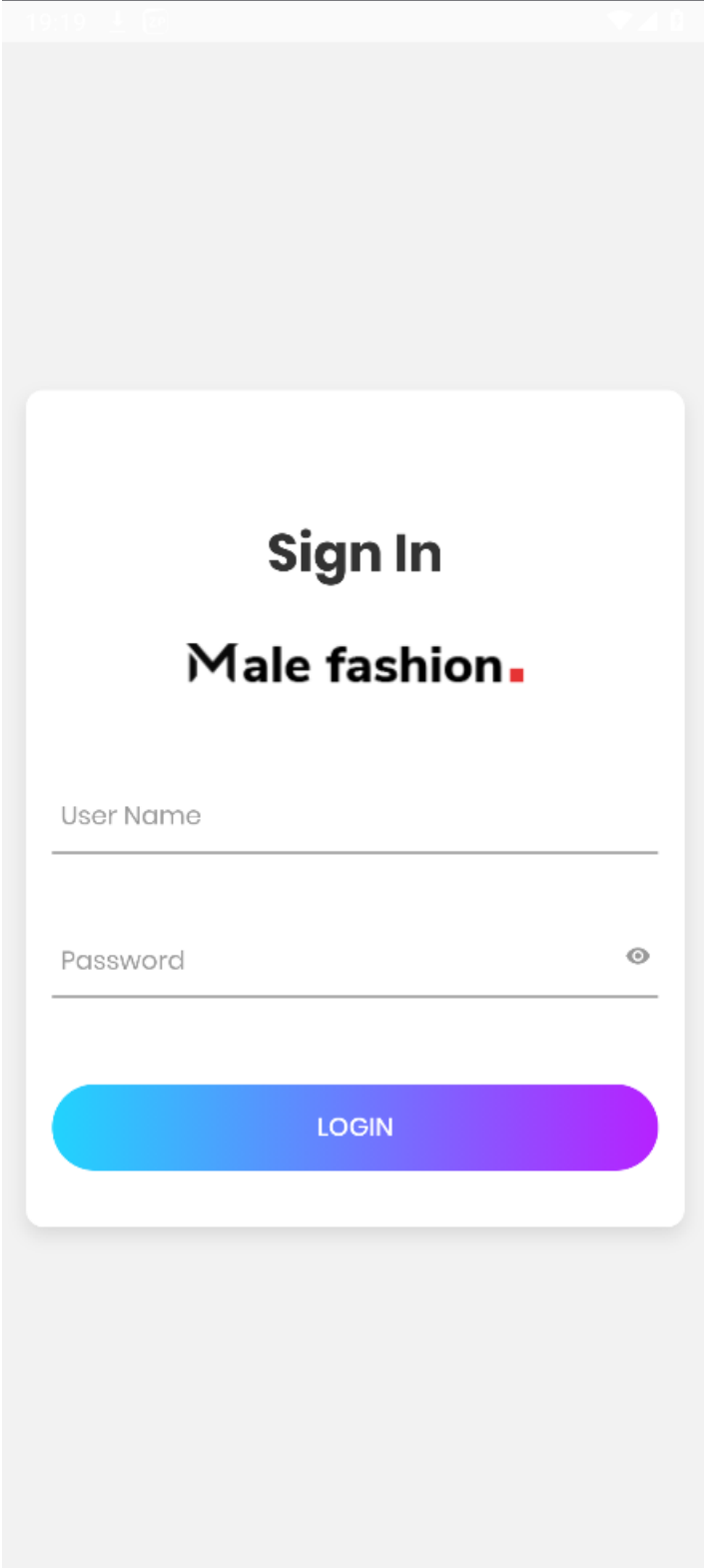
            @Override
            public void onPageFinished(WebView view, String url) {
                Log.d("WebView", "your current url when webpage loading..
finish" + url);
                super.onPageFinished(view, url);
            }

            @Override
            public void onLoadResource(WebView view, String url) {
                // TODO Auto-generated method stub
                super.onLoadResource(view, url);
            }

            @Override
            public boolean shouldOverrideUrlLoading(WebView view, String url)
{
                System.out.println("when you click on any interlink on
webview that time you got url :-" + url);
                if(url.equals("http://172.16.2.26:8080/redirect")){
                    startActivity(new Intent(AdminActivity.this,
LoginActivity.class));
                }
                return super.shouldOverrideUrlLoading(view, url);
            }
        });
    }
}
```

### 5.17. Giao diện đăng nhập tài khoản admin

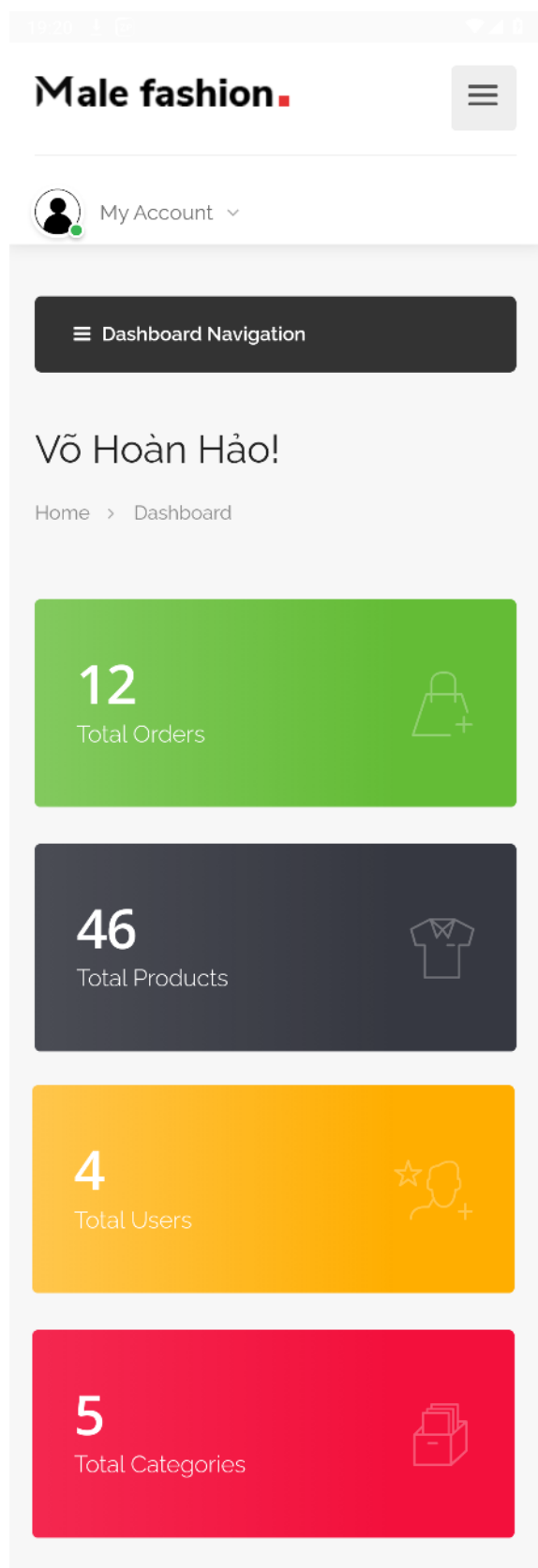
Chức năng: Đăng nhập tài khoản admin bằng username và password



The image shows a mobile application interface for a login screen. At the top, there is a status bar with the time 12:19 and battery level 65%. The main content area has a light gray background. In the center, there is a white rounded rectangle containing the login form. The form has the title "Sign In" in bold black text, followed by "Male fashion" in bold black text with a small red square icon. Below the title, there are two input fields: "User Name" and "Password". The "Password" field has a toggle icon (an eye) to its right. At the bottom of the form, there is a large, rounded button with a blue-to-purple gradient and the text "LOGIN" in white capital letters.

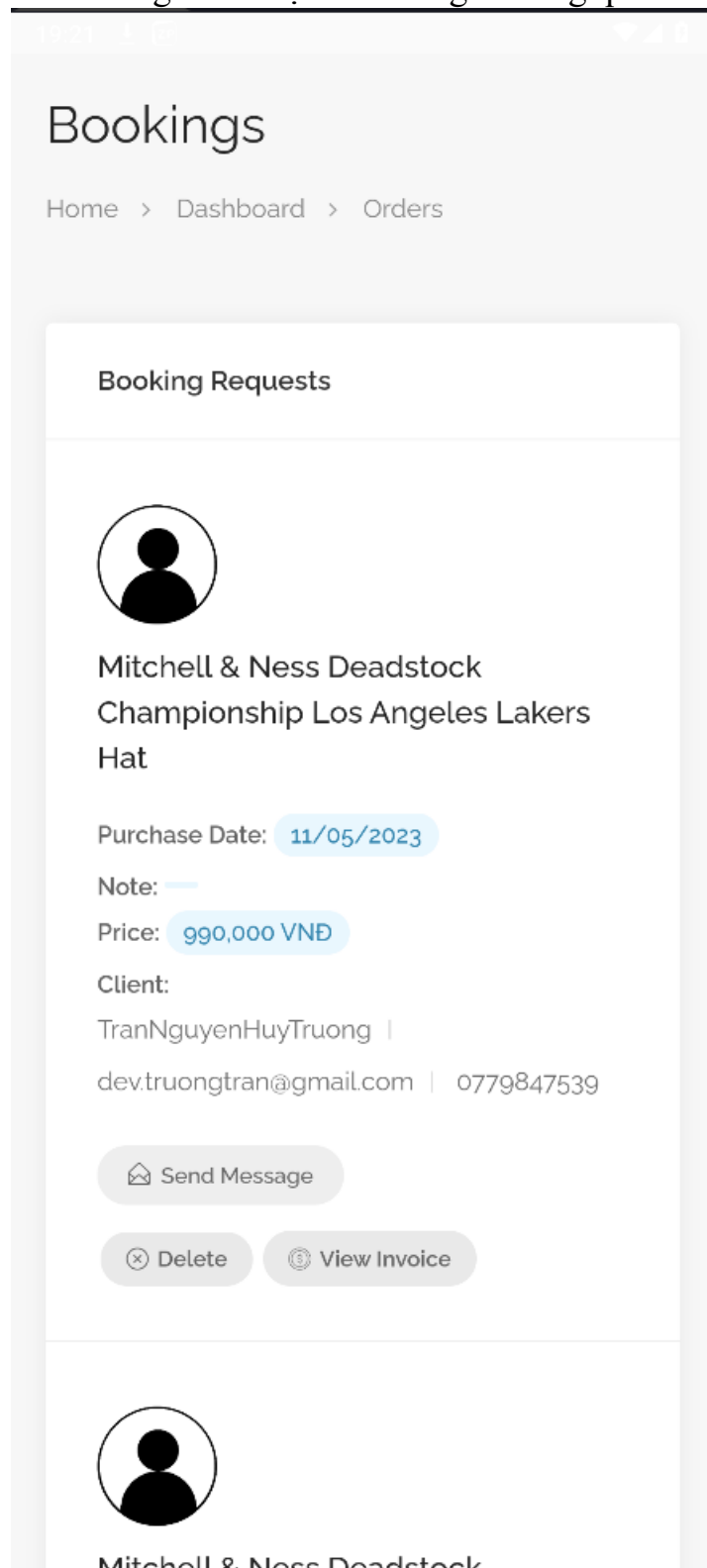
### 5.18. Giao diện chính của admin

Chức năng: Thể hiện tổng số lượng : Orders, Products, Users và Categories



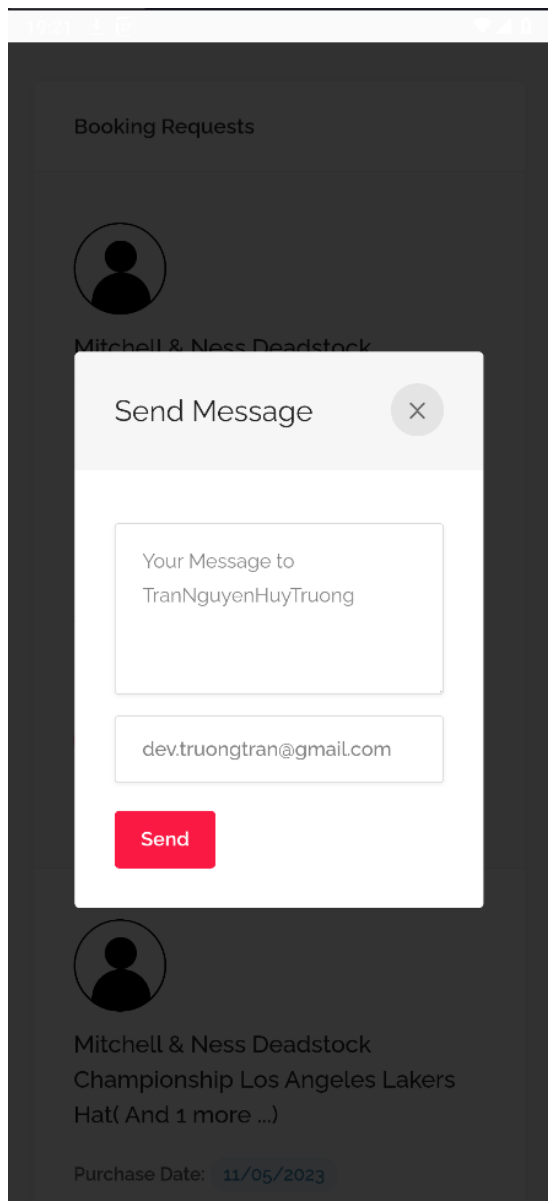
### 5.19. Giao diện danh sách tất cả các đơn hàng đã order

Chức năng: Thể hiện đơn hàng và tổng quát của một đơn hàng



### 5.20. Giao diện gửi mail đến đơn hàng của người dùng

Chức năng: Từ vai trò admin gửi mail đến người dùng của đơn hàng



### 5.21. Giao diện chi tiết hoá đơn của một đơn hàng

Chức năng: Khi bấm vào một đơn hàng sẽ chuyển sang chi tiết hoá đơn của đơn hàng đó

Issued:11-05-2023

Due 7 days from date of issue

# Invoice

## Supplier

Male-Fashion

498 Lê Văn Việt - Tầng Nhơn Phú A

TP. Thủ Đức - TP. Hồ Chí Minh

## Customer

TranNguyenHuyTruong

SĐT: 0779847539

Email: dev.truongtran@gmail.com

Description	Quantity	Total
Mitchell & Ness		
Deadstock	1	990000
Championship Los		VNĐ
Angeles Lakers Hat		

Address: 19/25

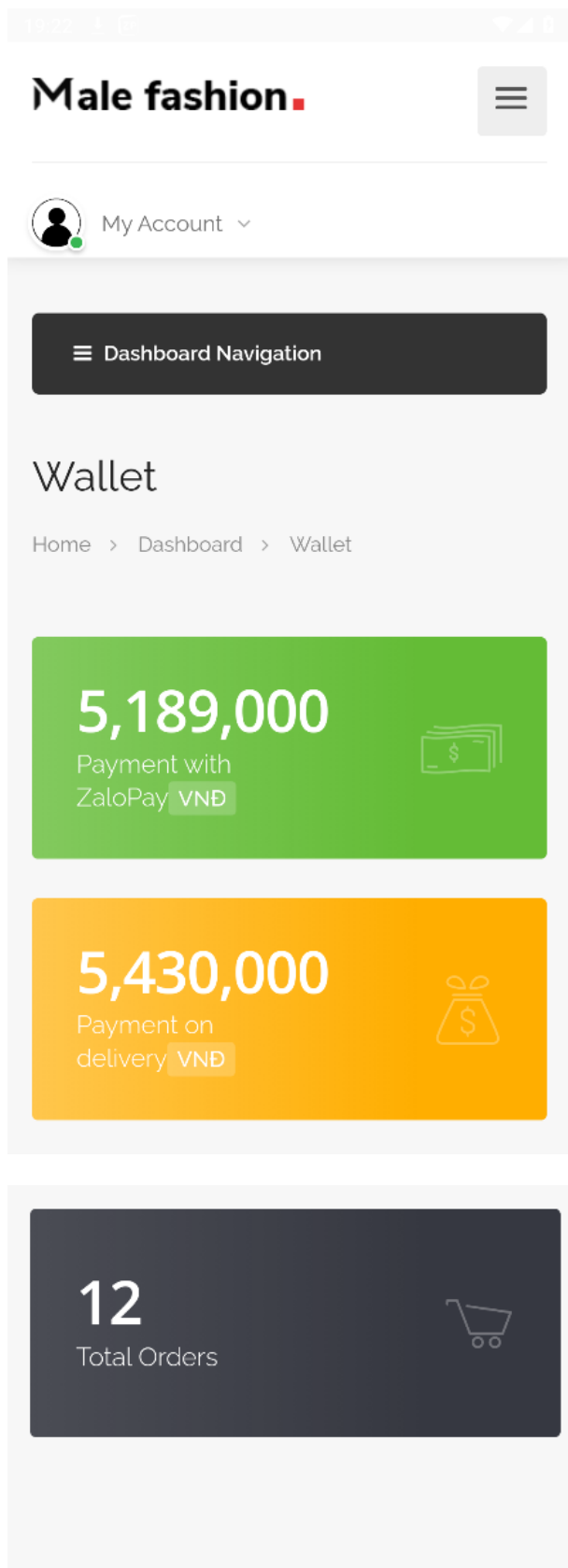
Payment method: Pay with ZaloPay

Total Due 990000 VNĐ

Các phần code chính:

## 5.22. Giao diện Wallet

Chức năng: Tổng số tiền bán được bằng zalopay, tổng số tiền bán được bằng giao hàng trực tiếp, tổng số đơn hàng đã bán



## **PHẦN 6: KẾT LUẬN**

### **5.1. Đánh giá kết quả thu được**

Các chức năng của chương trình hoạt động đúng với kế hoạch ban đầu đã đặt ra. Thể hiện được những nhu cầu cơ bản của bài toán được đặt ra thông qua những kiến thức đã được học trong bộ môn lập trình di động. Các thành viên hoạt động rất năng nổ, quá trình đồng bộ code diễn ra tương đối thuận lợi.

### **5.2. Phương hướng phát triển**

Vấn đề quản lý dữ liệu là một bài toán mở, tiềm năng phát triển là rất lớn trong đó có thể kể đến:

- + Thêm các chức năng: tích hợp công nghệ AI vào việc lọc nhu cầu tìm kiếm người dùng
- + Thêm chức năng thanh toán bằng thẻ ngân hàng
- + Thêm chức năng đăng nhập bằng facebook
- + Phát triển thêm phần đồ họa bắt mắt.
- + Tối ưu hóa thuật toán và luôn chạy chương trình.



## **TÀI LIỆU THAM KHẢO**

- [1] <https://www.javatpoint.com/android-tutorial>
- [2] <https://www.tutorialspoint.com/android/index.htm>
- [3] <https://stackoverflow.com/>
- [4] <https://www.youtube.com/>