



BÀI TẬP LỚN

HỌC PHẦN: LẬP TRÌNH MOBILE

Giảng viên hướng dẫn: Trần Đức Minh

Sinh viên thực hiện:

STT	Mã sv	Họ và tên	Lớp
1	1451020062	Hoàng Trung Đức	CNTT 14-01
2	1451020273	Nguyễn Xuân Viễn	CNTT 14-01
3	1451020145	Phạm Tùng Linh	CNTT 14-01
4	1451020044	Tô Đăng Dũng	CNTT 14-01

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM



BÀI TẬP LỚN
HỌC PHẦN: LẬP TRÌNH MOBILE

STT	Mã SV	Họ và Tên	Ngày sinh	Điểm	
				Bảng số	Bảng chữ
1	1451020062	Hoàng Trung Đức	27/06/2002		
2	1451020273	Nguyễn Xuân Viễn	05/11/2002		
3	1451020145	Phạm Tùng Linh	26/04/2002		
4	1451020044	Tô Đăng Dũng	26/09/2002		

CÁN BỘ CHẤM THI 1

CÁN BỘ CHẤM THI 2

Hà Nội, tháng 12 năm 2023

LỜI NÓI ĐẦU

Trong sự phát triển mạnh mẽ của khoa học công nghệ thế giới, có thể nói rằng ngành công nghệ thông tin phát triển với tốc độ nhanh nhất và ngày càng thể hiện được vai trò to lớn, hết sức quan trọng đối với toàn bộ đời sống con người. Hầu hết các lĩnh vực trong xã hội đều ứng dụng công nghệ thông tin, nhiều phần mềm đã mang lại hiệu quả không thể phủ nhận. Song thực tế luôn đặt ra những yêu cầu mới đòi hỏi ngành công nghệ thông tin không ngừng phát triển để thỏa mãn và đáp ứng những thay đổi của cuộc sống. Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố quyết định trong hoạt động của các chính phủ, tổ chức cũng như các công ty lớn và nhỏ, chúng đóng vai trò hết sức quan trọng, có thể tạo ra những bước phát triển mạnh mẽ.

Việc sử dụng ứng dụng thời tiết có thể cung cấp thông tin thời tiết chi tiết, dự báo thời tiết trong thời gian thực và các tính năng khác như cảnh báo thời tiết xấu, chỉ số chất lượng không khí, và các thông tin liên quan khác. Giao diện của ứng dụng có thể thiết kế thân thiện, dễ sử dụng và hấp dẫn, giúp người dùng dễ dàng truy cập và tìm kiếm thông tin thời tiết theo địa điểm mong muốn đang được người dùng rất quan tâm. Vì vậy, trong đồ án này, em xin đưa ra ý tưởng xây dựng một ứng dụng quản lý thời tiết nhằm giúp cho việc cập nhật thông tin thời tiết dễ dàng hơn và người dùng yên tâm, hài lòng về thông tin thời tiết mà họ sử dụng.

Ngoài ra, ứng dụng cũng có thể tích hợp các tính năng như thông báo thời tiết tự động, cập nhật thông tin thời tiết theo vị trí hiện tại của người dùng, và khả năng tùy chỉnh để người dùng có thể theo dõi thông tin thời tiết của nhiều địa điểm khác nhau.

MỤC LỤC

LỜI NÓI ĐẦU	3
CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI.....	6
1. Tên đề tài	6
2. Mục tiêu của đề tài	6
3. Ý nghĩa của đề tài	6
4. Đối tượng của đề tài.	6
5. Một số trang chính của ứng dụng.	6
CHƯƠNG 2. KỸ THUẬT XÂY DỰNG ỨNG DỤNG THỜI TIẾT	7
2.1. Tìm hiểu về dịch vụ Open Weather Map	7
2.1.1. Khái niệm	7
2.1.2. Các tính năng nổi bật của API thời tiết Open Weather Map.....	7
2.1.3. Đăng ký và kích hoạt sử dụng key API Openweathermap.....	7
2.2. Tìm hiểu về firebase	9
2.2.1. Khái niệm	9
2.2.2. Các dịch vụ chính của Firebase bao gồm	9
2.2.3. Các ưu điểm của Firebase.....	9
2.2.4. Các điểm hạn chế của Firebase	10
2.3. Tìm hiểu về SQLite	10
2.3.1. Khái niệm	10
2.3.2. Tính năng nổi bật của SQLite.....	11
2.3.3. Ưu điểm của SQLite.....	11
2.3.4. Nhược điểm của SQLite	11
CHƯƠNG 3: THIẾT KẾ CHƯƠNG TRÌNH	12
3.1. Thiết kế giao diện	12
3.2. Phần xử lý.....	16

CHƯƠNG 4: CHƯƠNG TRÌNH THỰC NGHIỆM.....	29
4.1. Phát biểu bài toán	29
4.2. Mô hình chức năng	29
4.3. Mô hình hoạt động.....	30
4.4. Giao diện chương trình.....	31
4.4.1. Giao diện màn hình Register và Login.....	31
4.4.2. Giao diện màn hình Main	32
4.4.3. Giao diện màn hình List 5 days next.....	33
4.4.4. Giao diện màn hình Favorite city	34
KẾT LUẬN	35
TÀI LIỆU THAM KHẢO	36

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1. Tên đề tài

Thiết kế ứng dụng dự báo thời tiết

2. Mục tiêu của đề tài

Thiết kế và xây dựng ứng dụng dự báo thời tiết trên android

3. Ý nghĩa của đề tài

Ứng dụng thời tiết có ý nghĩa quan trọng trong cuộc sống hàng ngày của chúng ta vì nó cung cấp thông tin về tình hình thời tiết hiện tại và dự báo trong tương lai. Giúp người dùng lên kế hoạch cho hoạt động trong ngày hoặc trong thời gian sắp tới.

Ứng dụng thời tiết cho phép người dùng tùy chỉnh thông tin theo địa điểm mong muốn. Người dùng có thể xem thông tin thời tiết của nhiều địa điểm khác nhau như thành phố yêu thích... Đặc biệt hữu ích khi đi du lịch hoặc theo dõi thời tiết của gia đình và bạn bè ở xa.

4. Đối tượng của đề tài.

Đối tượng nghiên cứu chủ yếu là các ứng dụng dự báo thời tiết.

5. Một số trang chính của ứng dụng.

- Trang chủ: Đây là trang giới thiệu sơ lược của ứng dụng.
- Trang hiển thị thời tiết: Hiển thị thông tin chi tiết thời tiết
- Trang hiển thị thành phố: Hiển thị thông tin thành phố yêu thích
- Trang đăng ký: Người dùng đăng ký tài khoản để trở thành khách hàng của trang.
- Trang đăng nhập: Người dùng đăng nhập tài khoản để đặt mua sản phẩm của trang.

CHƯƠNG 2. KỸ THUẬT XÂY DỰNG ỨNG DỤNG THỜI TIẾT

2.1. Tìm hiểu về dịch vụ Open Weather Map

2.1.1. Khái niệm

OpenWeatherMap API này cung cấp dịch vụ dữ liệu thời tiết và dự báo miễn phí, thích hợp cho bất kỳ dịch vụ bản đồ như các ứng dụng web và điện thoại thông minh. Ý tưởng được lấy cảm hứng từ OpenStreetMap và Wikipedia nhằm cung cấp thông tin miễn phí và sẵn có cho mọi người.

OpenWeatherMap cung cấp nhiều dữ liệu thời tiết như bản đồ thời tiết hiện tại, dự báo tuần, lượng mưa, gió, mây, dữ liệu từ các trạm thời tiết và nhiều thứ khác. Dữ liệu thời tiết được nhận từ các dịch vụ phát sóng khí tượng toàn cầu và hơn 40.000 trạm khí tượng.

Bạn có thể nhận được bất kỳ dữ liệu thời tiết nào cho ứng dụng của bạn bằng cách sử dụng chuỗi JSON hoặc XML.

2.1.2. Các tính năng nổi bật của API thời tiết Open Weather Map

Dữ liệu thời tiết hiện tại (Current weather data) Truy cập dữ liệu thời tiết hiện tại cho bất kỳ vị trí nào bao gồm hơn 200.000 thành phố. Thời tiết hiện tại thường xuyên được cập nhật dựa trên các mô hình và dữ liệu toàn cầu từ hơn 40.000 trạm thời tiết. Dữ liệu có sẵn ở định dạng JSON, XML hoặc HTML

Dự báo thời tiết 5 ngày tiếp theo cập nhật liên tục sau 3 giờ. Dự báo 5 ngày có sẵn tại bất kỳ địa điểm hoặc thành phố nào. Dự báo 5 ngày bao gồm dữ liệu thời tiết cứ sau 3 giờ. Dự báo có sẵn trong JSON và XML.

Dự báo 16 ngày tiếp theo Dự báo 16 ngày có sẵn tại bất kỳ địa điểm hoặc thành phố nào. Dự báo 16 ngày bao gồm thời tiết hàng ngày. Dự báo có sẵn trong JSON và XML

Dữ liệu lịch sử thời tiết API OpenWeatherMap cung cấp dữ liệu lịch sử thời tiết cho hơn 37.000 thành phố. Dữ liệu lịch sử trong 1 tháng trước đó có sẵn trong tài khoản Starter, cho 1 năm trước trong tài khoản Medium và trong 5 năm trong History Bulk

2.1.3. Đăng ký và kích hoạt sử dụng key API Openweathermap

Đầu tiên ta đăng ký tài khoản Open weather API để lấy API Key Ta truy cập đường link https://home.openweathermap.org/users/sign_up để đăng ký tài khoản miễn phí của OpenWeatherMap và nhận key miễn phí duy nhất cho tài khoản đó.

Hình 1: Trang chủ đăng ký tài khoản OpenWeatherMap

Khi đăng ký và đăng nhập thành công, chúng ta sẽ có API key bên dưới:

Hình 2: Trang chủ OpenWeatherMap

Kiểm tra chuỗi JSON truy vấn dữ liệu từ API URL lấy thông tin thời tiết theo kinh độ vĩ độ <http://api.openweathermap.org/data/2.5/weather?q=HaiPhong&appid=d0e912badechb079b4f4658f6625cd097>



Hình 3: URL thông tin thời tiết

2.2. Tìm hiểu về firebase

2.2.1. Khái niệm

Firestore là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính là giúp người dùng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

Cụ thể là những giao diện lập trình ứng dụng API đơn giản. Mục đích nhằm tăng số lượng người dùng và thu lại nhiều lợi nhuận hơn.

Đặc biệt, còn là dịch vụ đa năng và bảo mật cực tốt. Firestore hỗ trợ cả hai nền tảng Android và IOS

2.2.2. Các dịch vụ chính của Firestore bao gồm

- Firestore Authentication: Dịch vụ xác thực người dùng, cho phép bạn xác minh danh tính người dùng thông qua các phương thức như email/password, xác thực xã hội (Facebook, Google, Twitter), và xác thực qua số điện thoại.
- Cloud Firestore: Cơ sở dữ liệu NoSQL thời gian thực cho phép lưu trữ và đồng bộ dữ liệu giữa các ứng dụng di động và web.
- Firestore Cloud Messaging (FCM): Dịch vụ thông báo đẩy cho phép gửi thông báo từ máy chủ đến các thiết bị di động và web.
- Firestore Storage: Dịch vụ lưu trữ đám mây cho phép tải lên và tải xuống các tệp tin như hình ảnh, video và tài liệu.
- Firestore Hosting: Dịch vụ lưu trữ và phân phối các ứng dụng web tĩnh.
- Firestore Analytics: Dịch vụ phân tích và theo dõi hiệu suất ứng dụng và hành vi người dùng.

2.2.3. Các ưu điểm của Firestore

- Tạo tài khoản và sử dụng dễ dàng
- Tốc độ phát triển nhanh

- Nhiều dịch vụ trong một nền tảng
- Được cung cấp bởi Google
- Tập trung vào phát triển giao diện người dùng
- Firebase không có máy chủ
- Học máy (Machine Learning)
- Tạo lưu lượng truy cập
- Theo dõi lỗi
- Sao lưu

2.2.4. Các điểm hạn chế của Firebase

- Mặc dù đây là một nền tảng tuyệt vời, nhưng Firebase cũng có một số nhược điểm.
- Không phải là mã nguồn mở
- Người dùng không có quyền truy cập mã nguồn
- Firebase không hoạt động ở nhiều quốc gia
- Chỉ hoạt động với Cơ sở dữ liệu NoSQL
- Truy vấn chậm
- Không phải tất cả các dịch vụ Firebase đều miễn phí
- Firebase khá đắt và giá không ổn định
- Chỉ chạy trên Google Cloud
- Thiếu Dedicated Servers và hợp đồng doanh nghiệp
- Không cung cấp các API GraphQL

2.3. Tìm hiểu về SQLite

2.3.1. Khái niệm

SQLite là một hệ quản trị cơ sở dữ liệu hay còn gọi là hệ thống cơ sở dữ liệu quan hệ nhỏ gọn, khác với các hệ quản trị khác như MySQL, SQL Server, Oracle, PostgreSQL... SQLite là một thư viện phần mềm mà triển khai một SQL Database Engine truyền thống, không cần mô hình client-server nên rất nhỏ gọn. SQLite được sử dụng vào rất nhiều chương trình từ desktop đến mobile hay là website.

2.3.2. Tính năng nổi bật của SQLite

SQLite là một cơ sở dữ liệu nhúng, không yêu cầu giấy phép nào sau khi cài đặt và cho phép các nhà phát triển lưu trữ dữ liệu mà không cần máy chủ. Nó không cần máy chủ bởi nó không cần một quy trình hoặc hệ thống máy chủ khác để hoạt động.

SQLite có tính linh hoạt cao bởi, bạn có thể dùng nó để làm việc đồng thời trên nhiều cơ sở dữ liệu trong cùng một phiên. Chạy trên đa nền tảng, mang đến cho doanh nghiệp các giải pháp đáng tin cậy để lưu trữ dữ liệu cục bộ trên các ứng dụng, thiết bị. Cung cấp giao diện đơn giản, dễ sử dụng nhưng vẫn đáp ứng đủ các tiện ích cơ bản có trong các hệ thống cơ sở dữ liệu phức tạp và mở rộng hơn. Ngoài ra, SQLite cũng có bộ tiện ích mở rộng đa dạng với các chứng năng bổ sung hữu ích.

2.3.3. Ưu điểm của SQLite.

- SQLite không cần mô hình client – server để hoạt động.
- SQLite không cần phải cấu hình tức là bạn không cần phải cài đặt.
- SQLite database được lưu trữ trên một tập tin duy nhất.
- SQLite hỗ trợ hầu hết các tính năng của ngôn ngữ truy vấn SQL theo chuẩn SQL92.
- SQLite rất nhỏ gọn bản đầy đủ các tính năng nhỏ hơn 500kb, và có thể nhỏ hơn nếu lược bớt một số tính năng.
- Các thao tác dữ liệu trên SQLite chạy nhanh hơn so với các hệ quản trị cơ sở dữ liệu theo mô hình client – server.
- SQLite rất đơn giản và dễ dàng sử dụng.
- SQLite tuân thủ 4 tính chất ACID (là tính nguyên tử (Atomic), tính nhất quán (Consistent), tính cô lập (Isolated), và tính bền vững (Durable)).

Với đặc tính nhỏ gọn, truy xuất dữ liệu nhanh SQLite thường được sử dụng để nhúng vào các dự án.

2.3.4. Nhược điểm của SQLite

Ngoài những ưu điểm đã kể ra ở trên SQLite cũng có một số mặt hạn chế nếu đem so sánh với các hệ quản trị khác.

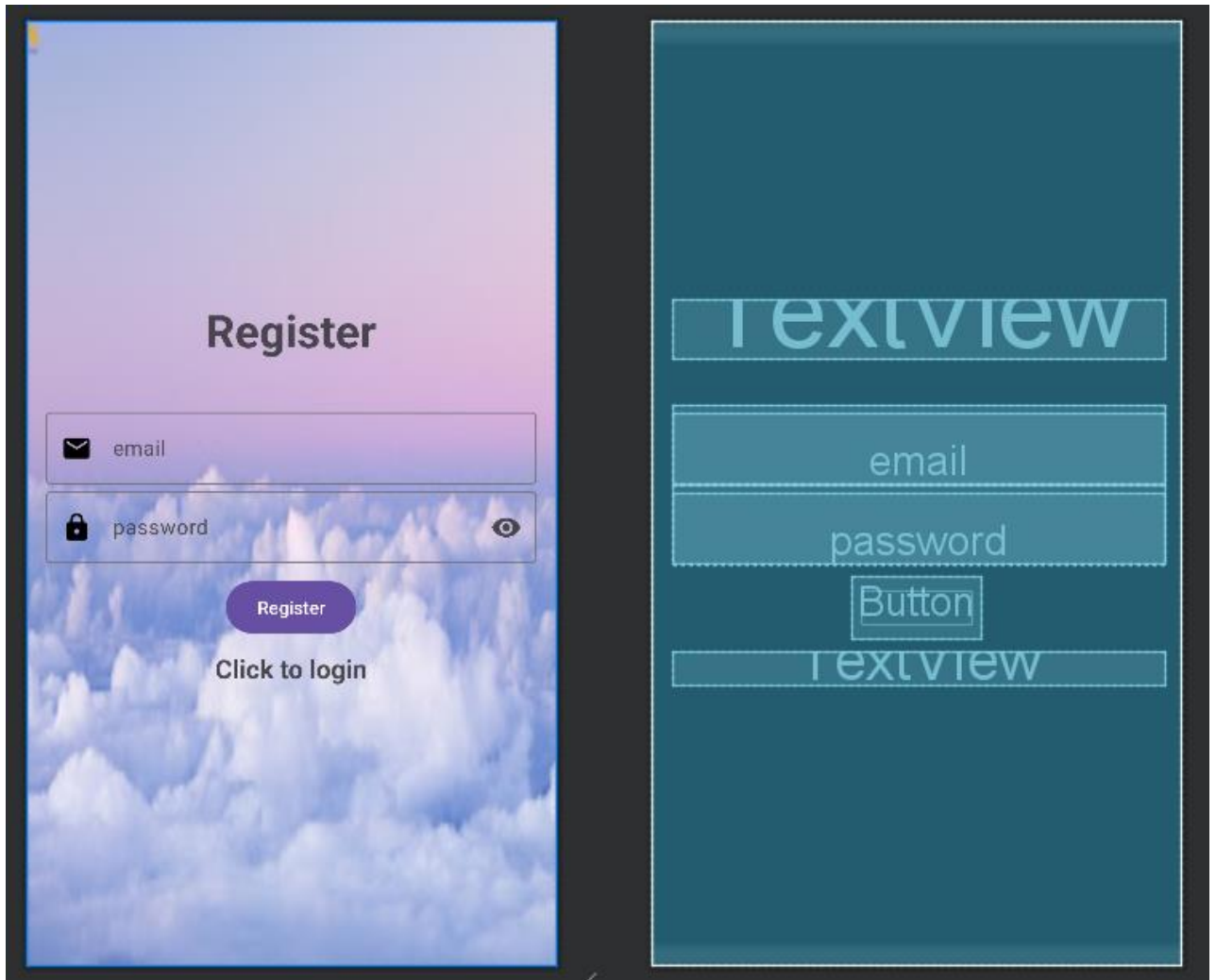
Do sử dụng cơ chế coarse-grained locking nên trong cùng một thời điểm SQLite có thể hỗ trợ nhiều người đọc dữ liệu, nhưng chỉ có 1 người có thể ghi dữ liệu.

SQLite không phải là lựa chọn hoàn hảo để đáp ứng các nhu cầu xử lý trên một khối lượng dữ liệu lớn, phát sinh liên tục.

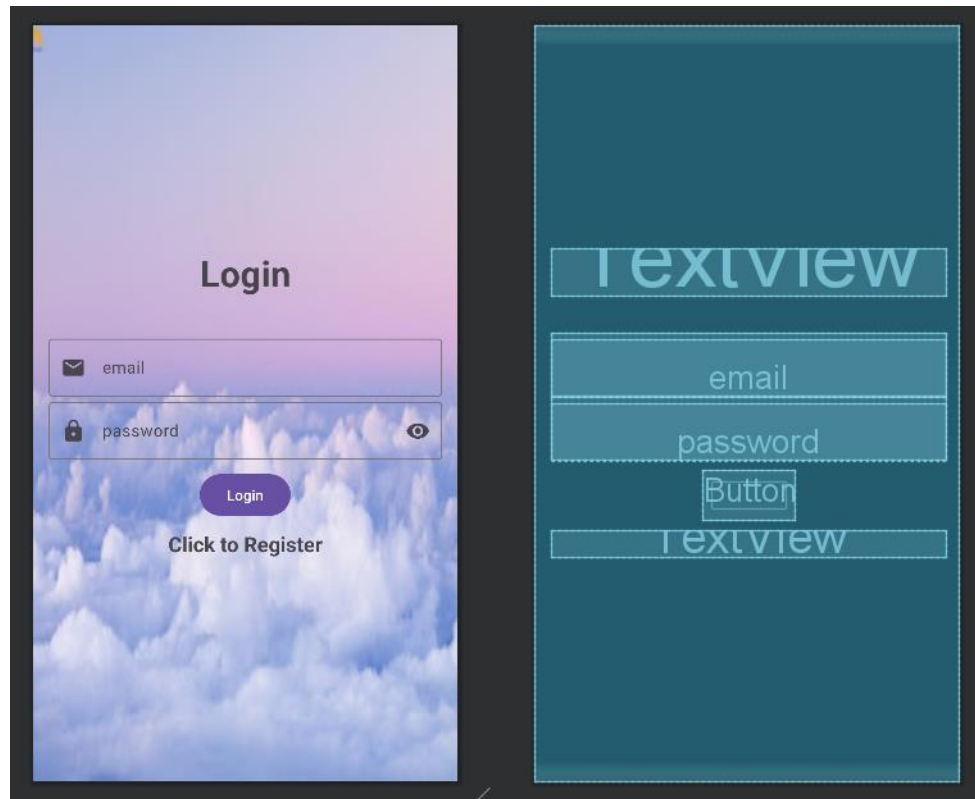
CHƯƠNG 3: THIẾT KẾ CHƯƠNG TRÌNH

3.1. Thiết kế giao diện

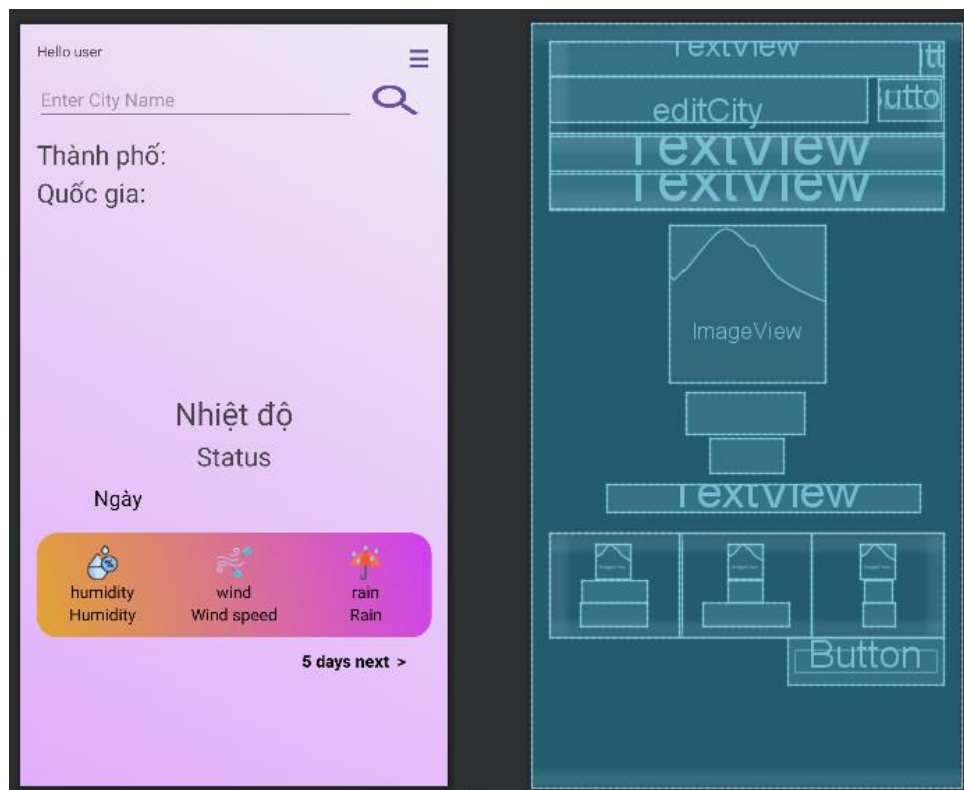
Trang Register



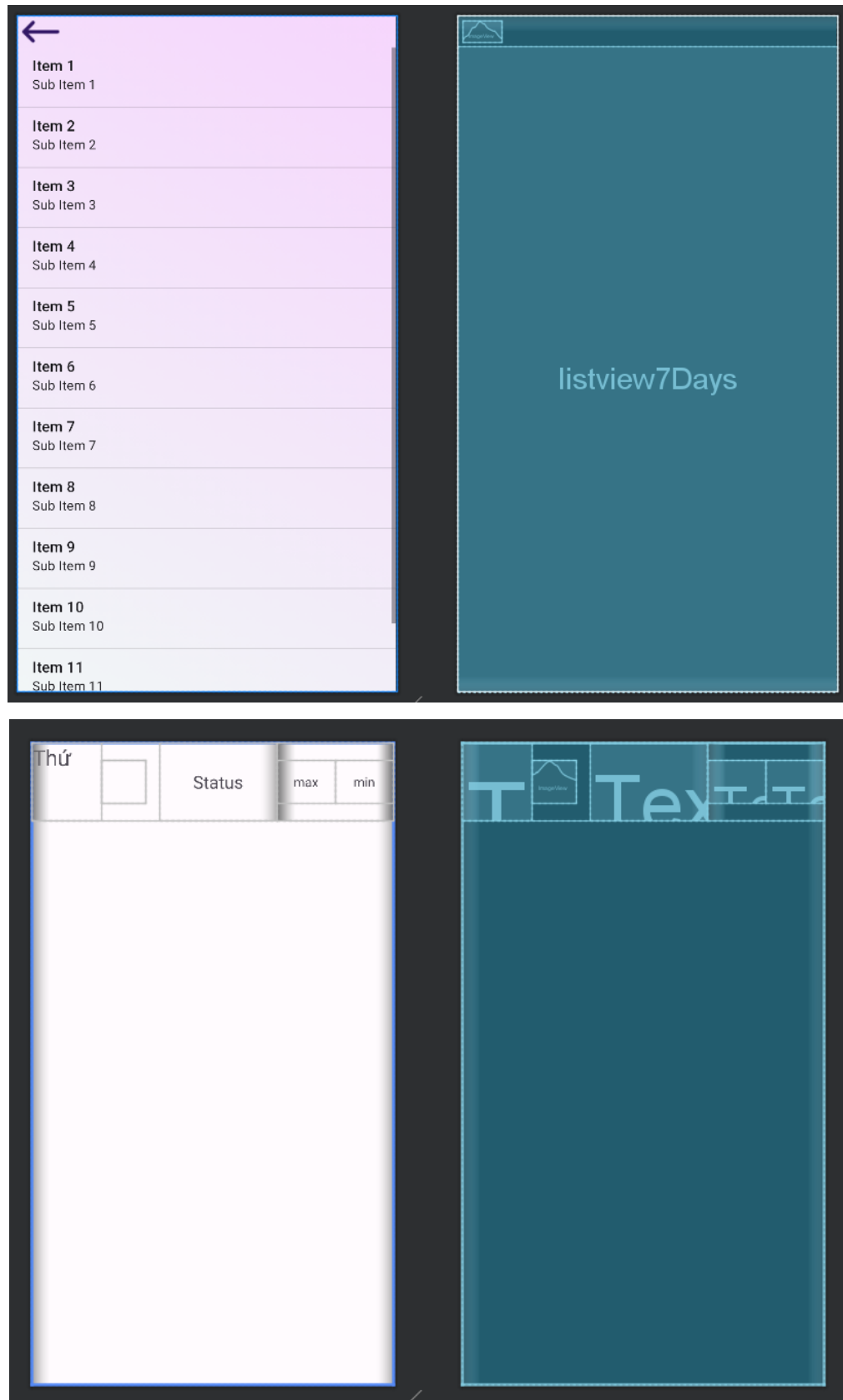
Trang Login

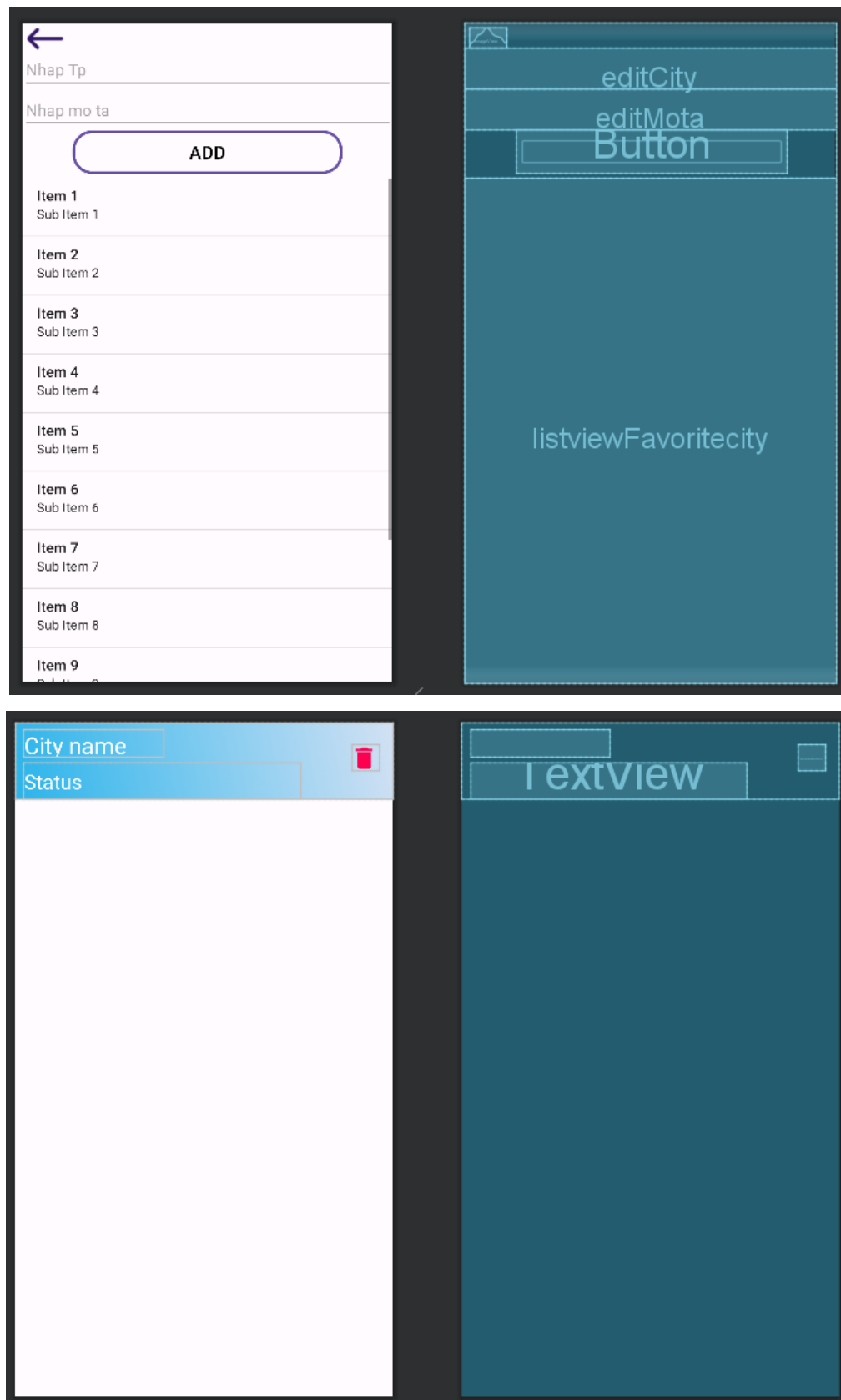


Trang Main



Trang List 5 days next



Trang Favorite city

3.2. Phần xử lý

Trang Register

```

public class Register extends AppCompatActivity {

    TextInputEditText editTextEmail, editTextPassword;
    Button btnRegister;
    FirebaseAuth mAuth;
    ProgressBar progressBar;
    TextView textView;

    @Override
    public void onStart() {
        super.onStart();
        // Check if user is signed in (non-null) and update UI
        accordingly.
        FirebaseUser currentUser = mAuth.getCurrentUser();
        if(currentUser != null){
            Intent intent = new Intent(getApplicationContext(),
Login.class);
            startActivity(intent);
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        mAuth = FirebaseAuth.getInstance();
        editTextEmail = findViewById(R.id.email);
        editTextPassword = findViewById(R.id.password);
        btnRegister = findViewById(R.id.btn_register);
        progressBar = findViewById(R.id.progressBar);
        textView = findViewById(R.id.loginNow);
        textView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(),
Login.class);
                startActivity(intent);
                finish();
            }
        });

        btnRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                progressBar.setVisibility(View.VISIBLE);
                String email, password;

```



```

        email = String.valueOf(editTextEmail.getText());
        password =
String.valueOf(editTextPassword.getText());

        if (TextUtils.isEmpty(email)) {
            Toast.makeText(Register.this, "Nhập Email",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (TextUtils.isEmpty(password)) {
            Toast.makeText(Register.this, "Nhập Password",
Toast.LENGTH_SHORT).show();
            return;
        }
        mAuth.createUserWithEmailAndPassword(email,
password)
            .addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull
Task<AuthResult> task) {
                    progressBar.setVisibility(View.GONE);
                    if (task.isSuccessful()) {
                        Toast.makeText(Register.this,
"Tạo tài khoản thành công",
Toast.LENGTH_SHORT).show();

                        Intent intent = new
Intent(Register.this, Login.class);
                        startActivity(intent);
                    } else {
                        // If sign in fails, display a
message to the user.
                        Toast.makeText(Register.this,
"Tạo tài khoản thất bại",
Toast.LENGTH_SHORT).show();
                    }
                }
            });
    }
}

```

Trang Login

```

public class Login extends AppCompatActivity {
    TextInputEditText editTextEmail, editTextPassword;
    Button btnLogin;
    FirebaseAuth mAuth;
    ProgressBar progressBar;
    TextView textView;

    @Override
    public void onStart() {
        super.onStart();
        // Check if user is signed in (non-null) and update UI
        accordingly.
        FirebaseUser currentUser = mAuth.getCurrentUser();
        if(currentUser != null){
            Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
            startActivity(intent);
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        mAuth = FirebaseAuth.getInstance();
        editTextEmail = findViewById(R.id.email);
        editTextPassword = findViewById(R.id.password);
        btnLogin = findViewById(R.id.btn_login);
        progressBar = findViewById(R.id.progressBar);
        textView = findViewById(R.id.RegisterNow);
        textView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(),
Register.class);
                startActivity(intent);
            }
        });

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                progressBar.setVisibility(View.VISIBLE);
                String email, password;
                email = String.valueOf(editTextEmail.getText());
                password =
String.valueOf(editTextPassword.getText());

```

```

        if (TextUtils.isEmpty(email)) {
            Toast.makeText(Login.this, "Nhập Email",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (TextUtils.isEmpty(password)) {
            Toast.makeText(Login.this, "Nhập Password",
Toast.LENGTH_SHORT).show();
            return;
        }

        mAuth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener( new
OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull
Task<AuthResult> task) {

progressBar.setVisibility(View.GONE);
                    if (task.isSuccessful()) {

Toast.makeText(getApplicationContext(), "Đăng nhập thành công",
Toast.LENGTH_SHORT).show();

                        Intent intent = new
Intent(getApplicationContext(), MainActivity.class);
                        startActivity(intent);
                    } else {
                        Toast.makeText(Login.this,
"Đăng nhập thất bại",
Toast.LENGTH_SHORT).show();
                    }
                }
            });
    }
}
}

```

Trang Main

```

public class MainActivity extends AppCompatActivity {

    FirebaseAuth auth;
    Button btn, btnGet, btnMenu, btnChangeActivity;
    TextView textView, txtviewName, txtcountry, txtTemp, txtStatus,
    txtHumi, txtRain, txtWind, txtviewDay;
    ImageView imgIcon;
    FirebaseUser user;
    EditText editCity;
    String City = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        auth = FirebaseAuth.getInstance();
        btn = findViewById(R.id.Login);
        textView = findViewById(R.id.user_details);
        user = auth.getCurrentUser();
        GetCurrentWeatherData("Hanoi");
        Anhxa();
        //lấy thông tin email người dùng
        if (user == null) {
            Intent intent = new Intent(getApplicationContext(),
Login.class);
            startActivity(intent);
            finish();
        } else {
            textView.setText(user.getEmail());
        }

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FirebaseAuth.getInstance().signOut();
                Intent intent = new Intent(getApplicationContext(),
Login.class);
                startActivity(intent);
                finish();
            }
        });

        btnGet.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String city = editCity.getText().toString();
                if (city.equals("")) {

```

```

        City = "Hanoi";
        GetCurrentWeatherData(City);
    }else{
        City = city;
        GetCurrentWeatherData(City);
    }
    }
});
btnChangeActivity.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String city = editCity.getText().toString();
        Intent intent = new Intent(MainActivity.this,
MainActivity2.class);
        intent.putExtra("name",city);
        startActivity(intent);
    }
});
btnMenu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        PopupMenu popupMenu = new
PopupMenu(MainActivity.this, btnMenu);

        popupMenu.getMenuInflater().inflate(R.menu.menu_popup,
popupMenu.getMenu());

        popupMenu.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem item) {

                int id = item.getItemId();
                if(id== R.id.FC){
                    Toast.makeText(MainActivity.this,"Đã
chuyển ", Toast.LENGTH_SHORT).show();
                    startActivity(new
Intent(getApplicationContext(), FavoriteCity.class));
                }
                else if(id == R.id.Logout) {
                    Toast.makeText(MainActivity.this,"Đã
thoát", Toast.LENGTH_SHORT).show();
                    startActivity(new
Intent(getApplicationContext(), Login.class));
                }
                return false;
            }
        });
});

```

```

        popupMenu.show();

    }

});

}

private void Anhxa() {
    editCity = findViewById(R.id.editCity);
    btnGet = findViewById(R.id.btnGet);
    btnChangeActivity = findViewById(R.id.btnChangeActivity);
    btnMenu = findViewById(R.id.btnMenu);
    txtviewName = findViewById(R.id.txtviewName);
    txtcountry = findViewById(R.id.txtCountry);
    txtTemp = findViewById(R.id.txtTemp);
    txtStatus = findViewById(R.id.txtStatus);
    txtHumi = findViewById(R.id.txtviewHumi);
    txtRain = findViewById(R.id.txtviewRain);
    txtWind = findViewById(R.id.txtviewWind);
    txtviewDay = findViewById(R.id.txtviewDay);
    imgIcon = findViewById(R.id.imgIcon);
}

public void GetCurrentWeatherData(String data){
    //thực thi những yêu cầu mình gửi đi
    //ba6915e1e5de61a0b628a45f4517673d key free
    RequestQueue requestQueue =
Volley.newRequestQueue(MainActivity.this);
    String url =
"https://api.openweathermap.org/data/2.5/weather?q="+data+"&appid=b
a6915e1e5de61a0b628a45f4517673d&units=metric";
    StringRequest stringRequest = new
StringRequest(Request.Method.GET, url,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {

                try {
                    JSONObject jsonObject = new
JSONObject(response);
                    String day =
jsonObject.getString("dt");
                    String name =
jsonObject.getString("name");
                    txtviewName.setText("City: " + name);

                    long l = Long.valueOf(day);
                    Date date = new Date(l * 1000L);
                    SimpleDateFormat simpleDateFormat = new
SimpleDateFormat("HH:mm EEEE dd-MM-yyyy");
                    String Day =

```

```

simpleDateFormat.format(date);
                txtviewDay.setText(Day);

                JSONArray jsonArrayWeather =
jsonObject.getJSONArray("weather");
                JSONObject jsonObjectWeather =
jsonArrayWeather.getJSONObject(0);
                String status =
jsonObjectWeather.getString("main");
                String icon =
jsonObjectWeather.getString("icon");

Picasso.get().load("https://openweathermap.org/img/wn/"+icon+".png"
).into(imgIcon);

                txtStatus.setText(status);

                JSONObject jsonObjectMain =
jsonObject.getJSONObject("main");
                String temp =
jsonObjectMain.getString("temp");
                String humi =
jsonObjectMain.getString("humidity");
                //Đổi nhiệt độ, độ ẩm qua dạng chuỗi
                Double a = Double.valueOf(temp);
                String TEMP =
String.valueOf(a.intValue());
                txtTemp.setText(TEMP+"°C");
                txtHumi.setText(humi+"%");

                JSONObject jsonObjectWind =
jsonObject.getJSONObject("wind");
                String wind =
jsonObjectWind.getString("speed");
                txtWind.setText(wind+"m/s");

                JSONObject jsonObjectRain =
jsonObject.getJSONObject("clouds");
                String rain =
jsonObjectRain.getString("all");
                txtRain.setText(rain+"%");

                JSONObject jsonObjectSys =
jsonObject.getJSONObject("sys");
                String country =
jsonObjectSys.getString("country");
                txtcountry.setText("Country:
"+country);
            } catch (JSONException e) {
                throw new RuntimeException(e);
            }

```

```

        }

        }

    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error)
    {

        }

    });
    requestQueue.add(stringRequest);
}
}

```

Trang List 5 days next

```

public class MainActivity2 extends AppCompatActivity {

    String tentp = "";
    ImageView imgback, imgviewIcon;
    TextView txtviewDate, txtviewDescription , txtviewMaxtemp,
    txtviewMintemp ;
    ListView listview;
    CustomAdapter customAdapter;
    ArrayList<Weather> arrayList;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        Anhxa();
        Intent intent = getIntent();
        String city = intent.getStringExtra("name");

        Log.d("ketqua", "Dữ liệu truyền qua:" + city);
        if (city.equals("")){
            tentp = "Hanoi";
            Get7Daysdata(tentp);
        }else {
            tentp = city;
            Get7Daysdata(tentp);
        }

        imgback.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(),
                MainActivity.class);
                startActivity(intent);
                finish();
            }
        });
    }
}

```



```

new SimpleDateFormat("EEEE");
SimpleDateFormat("HH:mm");
simpleDateFormat.format(date);
timeFormat.format(date);
gio);

SimpleDateFormat simpleDateFormat =
SimpleDateFormat timeFormat = new
String Thu =
String gio =
Log.d("ketqua", "thu: " + Thu +

//lấy mô tả thời tiết
JSONArray jsonArrayweather =
Log.d("ketqua", "thoitiet: "+
JSONObject jsonObjectweather =
Log.d("ketqua", "we: "+
String status7days =
Log.d("ketqua", "dews: "+
String icon7days =
jsonObjectweather.getString("description");
status7days);
jsonObjectweather.getString("icon");
//
Picasso.get().load("https://openweathermap.org/img/wn/"+icon7days+
.png").into(imgviewIcon);
Log.d("ketqua", "icon: "+
icon7days);
//
txtviewDescription.setText(status7days);

//Lấy nhiệt độ
JSONObject jsonObjectMain =
jsonObjectList.getJSONObject("main");
Log.d("ketqua", "main: "+
jsonObjectList );
Double max =
jsonObjectMain.getDouble("temp_max");
Double min =
jsonObjectMain.getDouble("temp_min");
Double a = Double.valueOf(max);
Double b = Double.valueOf(min);
Double MAX =
Double.valueOf(a.intValue());
Double MIN =
Double.valueOf(b.intValue());

```

```

//                                txtviewMaxtemp.setText(MAX+"°C");
//                                txtviewMintemp.setText(MIN+"°C");

                                arrayList.add(new
Weather(Thu.substring(0,3) +" "+ gio, status7days, icon7days, max,
min));
                                }
                                customAdapter.notifyDataSetChanged();
                                } catch (JSONException e) {
                                e.printStackTrace();

                                }
                                }
                                },
                                new Response.ErrorListener() {
                                @Override
                                public void onErrorResponse(VolleyError error)
                                {
                                Toast.makeText(MainActivity2.this, "No data
for this city", Toast.LENGTH_SHORT).show();
                                Log.d("MyError: ",error.toString());
                                }
                                });
                                requestQueue.add(stringRequest);
                                }
                                }

```

Trang Favorite city

```

public class FavoriteCity extends AppCompatActivity {

    EditText editCity, editMota;
    Button btnAdd;
    ImageView imgviewBack;
    ListView listviewCity;
    CityAdapter cityAdapter;
    ArrayList<Model> arrayListModel;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_favorite_city);

        editCity = findViewById(R.id.editCity);
        editMota = findViewById(R.id.editMota);
        btnAdd = findViewById(R.id.btnAdd);
        imgviewBack = findViewById(R.id.imgviewBack);
        listviewCity = findViewById(R.id.listviewFavoritecity);

        btnAdd.setOnClickListener(new View.OnClickListener() {

```

```

        @Override
        public void onClick(View v) {
            //          DbResponse dbResponse = new
            DbResponse(FavoriteCity.this);
            //          Model model = new Model();
            //          Toast.makeText(FavoriteCity.this, "them thanh
            cong", Toast.LENGTH_LONG).show();
            //          String Cityname = editCity.getText().toString();
            //          String Mota = editMota.getText().toString();
            //          if(TextUtils.isEmpty(Cityname)||
            //          TextUtils.isEmpty(Mota)){
            //          Toast.makeText(FavoriteCity.this, "thong tin
            trong", Toast.LENGTH_LONG).show();
            //          }
            //          model.setTenCity(Cityname);
            //          model.setMota(Mota);
            //          dbResponse.Them(model);
            //          cityAdapter = new CityAdapter(FavoriteCity.this,
            R.layout.list_city, arrayListModel);
            //          listViewCity.setAdapter(cityAdapter);
            Toast.makeText(FavoriteCity.this, "oke",
            Toast.LENGTH_LONG).show();
            ThemDL();
            HienThiDL();
        }
    });

}

private void ThemDL() {
    DbResponse dbResponse = new DbResponse(this);
    Model model = new Model();
    model.setMota(editMota.getText().toString());
    model.setTenCity(editCity.getText().toString());
    dbResponse.Them(model);
}

private void HienThiDL() {
    DbResponse dbResponse = new DbResponse(this);
    arrayListModel = dbResponse.LayDL();
    cityAdapter= new
    CityAdapter(this,R.layout.list_city,arrayListModel);
    listViewCity.setAdapter(cityAdapter);
}
}

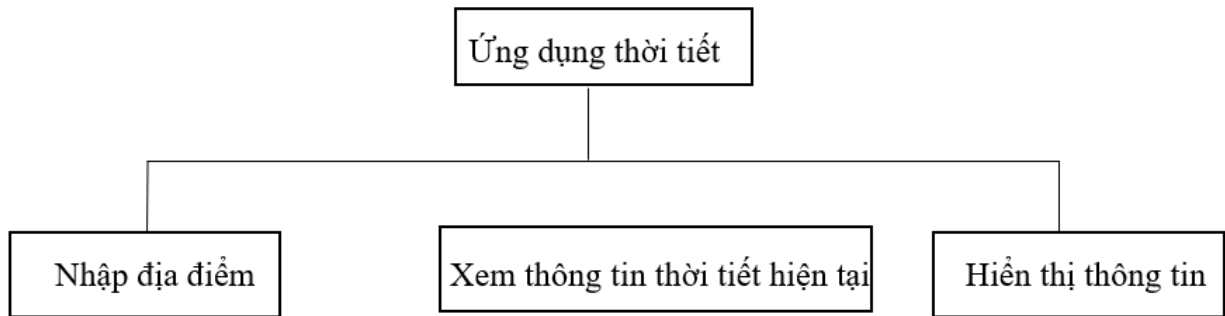
```

CHƯƠNG 4: CHƯƠNG TRÌNH THỰC NGHIỆM

4.1. Phát biểu bài toán

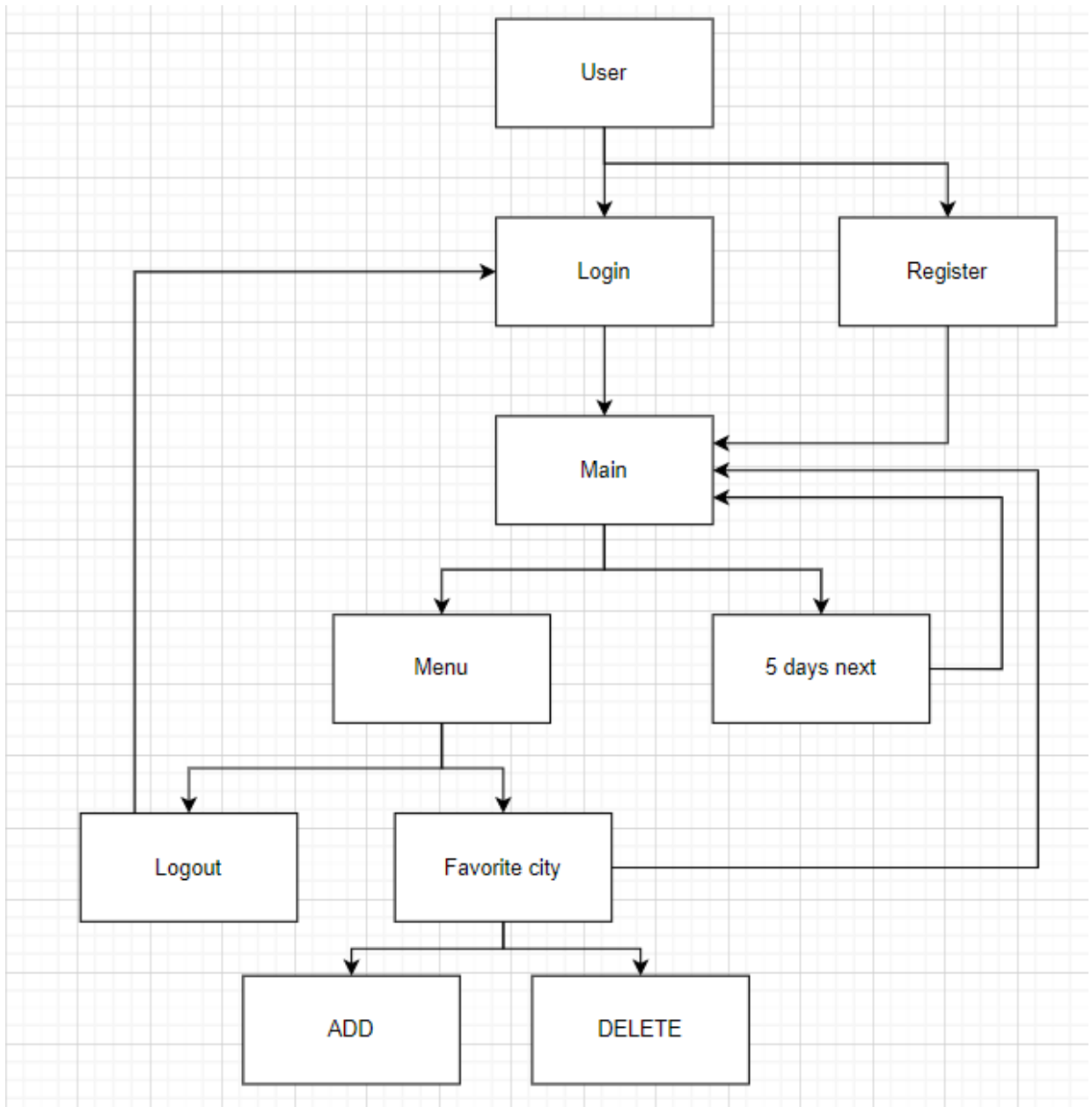
Ngày nay nhu cầu đi lại, tổ chức sự kiện của con người ngày càng tăng cao. Vì vậy, nếu nắm bắt được tình hình thời tiết như lượng mưa, độ ẩm, nhiệt độ thì sẽ dễ dàng sắp xếp cũng như tổ chức sự kiện, cuộc vui chơi một cách tốt nhất. Hiểu được nhu cầu cần thiết đó, em đã quyết định xây dựng ứng dụng “Dự báo thời tiết” để giúp người dùng có thể cập nhật tình hình thời tiết một cách nhanh nhất, mới nhất và chính xác nhất để người dùng có thể dễ dàng có những quyết định tốt nhất cho những dự định sắp đến của mình. Tốc độ phát triển nhanh chóng của các công nghệ di động hiện nay đã tạo ra các cơ hội cho những lập trình viên tự do thể hiện những ý tưởng sáng tạo của mình. Làm cho các ứng dụng ngày càng thân thiện để đáp ứng nhu cầu cuộc sống của mỗi cá nhân.

4.2. Mô hình chức năng



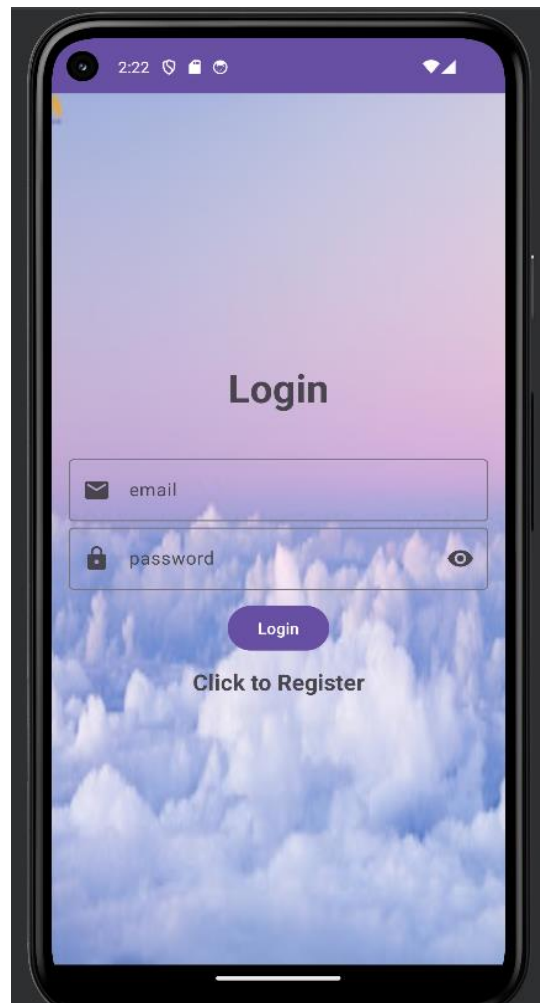
- Nhập địa điểm : Ứng dụng cho phép người dùng nhập địa điểm cần tìm
- Xem thông tin thời tiết hiện tại : Ứng dụng hiển thị thông tin tình hình thời tiết như lượng mưa, độ ẩm, nhiệt độ
- Hiển thị thông tin thời tiết: 7 ngày tiếp theo ứng dụng hiển thị thông tin tình hình nhiệt độ cao nhất và thấp nhất .

4.3. Mô hình hoạt động



4.4. Giao diện chương trình

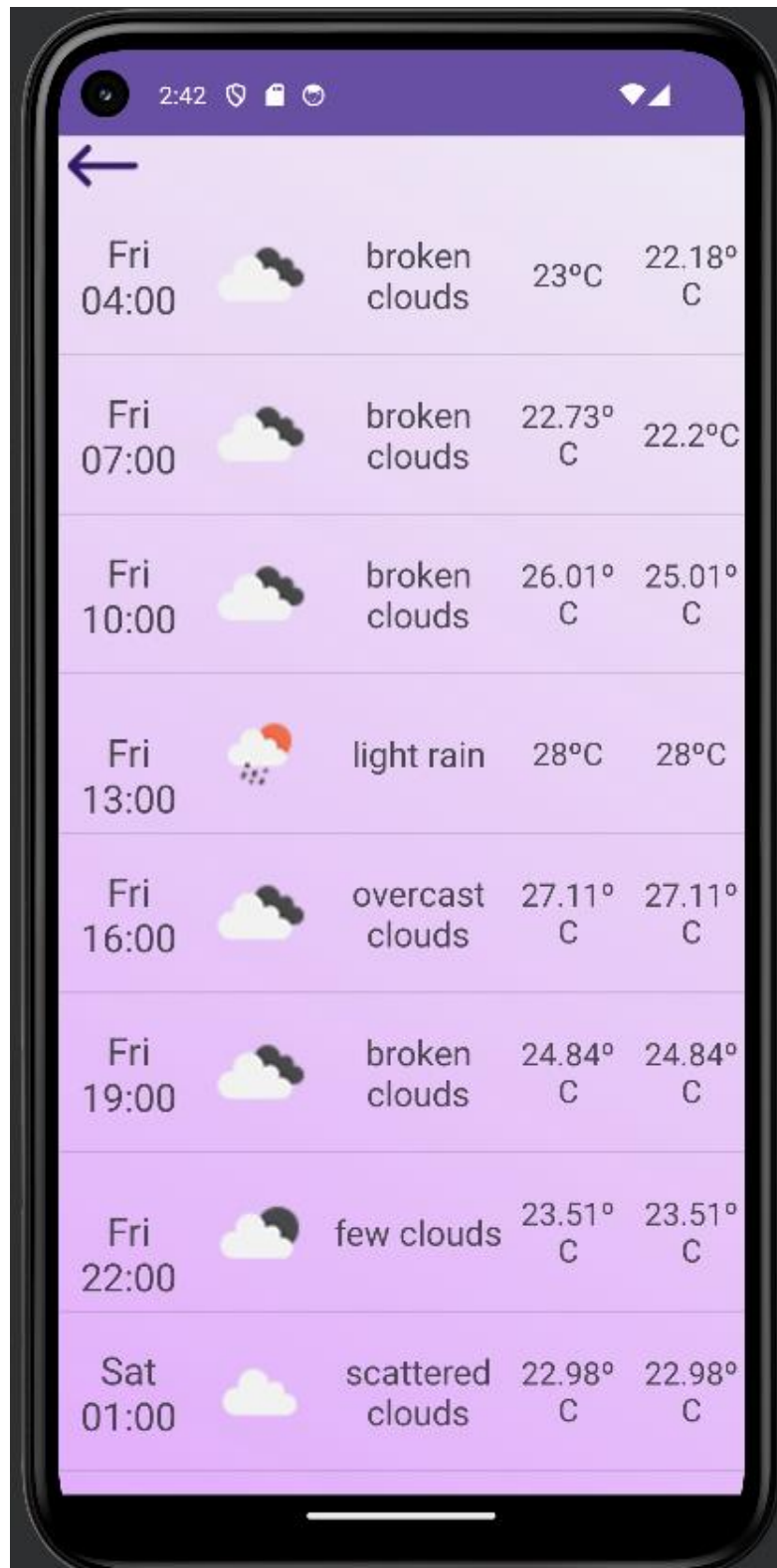
4.4.1. Giao diện màn hình Register và Login



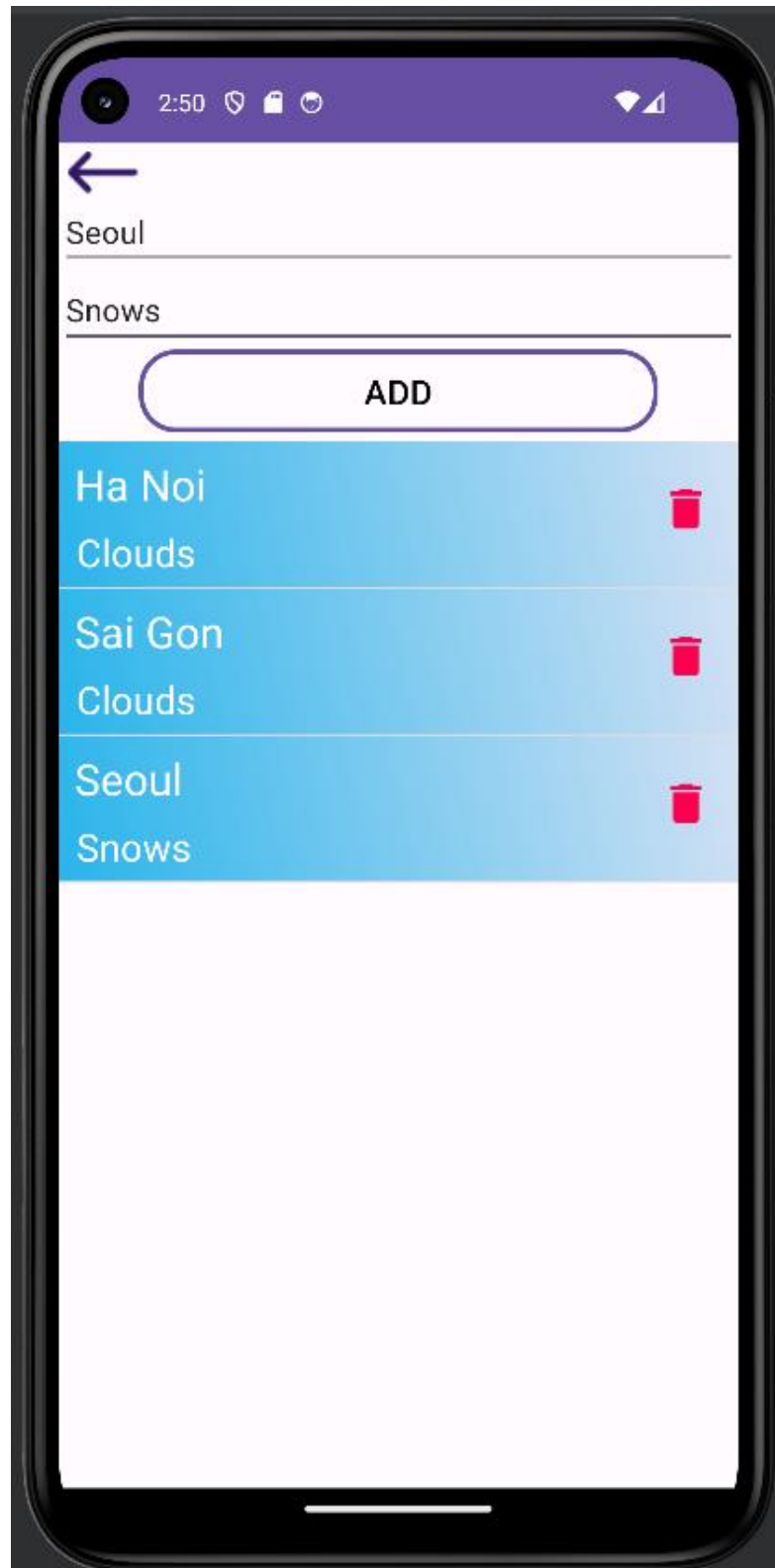
4.4.2. Giao diện màn hình Main



4.4.3. Giao diện màn hình List 5 days next



4.4.4. Giao diện màn hình Favorite city



KẾT LUẬN

Sau một thời gian tìm hiểu đề tài “Xây dựng ứng dụng Android lấy thông tin dự báo thời tiết”, chúng em đã thực hiện được đề tài của mình theo như dự kiến. Với mục đích là tìm hiểu về hệ điều hành Android, hiểu biết về ngôn ngữ lập trình mới cho điện thoại di động và tìm hiểu các ứng dụng của hệ điều hành Android trong thế giới di động ngày nay. Thông qua việc tìm hiểu và nắm bắt lý thuyết về hệ điều hành Android để bước đầu tìm hiểu ứng dụng dự báo thời tiết trên nền hệ điều hành Android cho điện thoại di động. Qua quá trình nghiên cứu và cố gắng thực hiện dưới sự hướng dẫn tận tình của Giảng viên, sự giúp đỡ của bạn bè trong nhóm nghiên cứu thể hiện được những điều sau:

Kết quả đạt được:

- Sau quá trình tìm hiểu và nghiên cứu đề tài, em đã bước đầu hiểu về hệ điều hành Android chạy trên điện thoại. Biết và nắm vững quá trình tạo một file project trong lập trình trên nền Android.
- Hiểu và sử dụng được các lớp đối tượng, phương thức hỗ trợ lập trình trên nền hệ điều hành Android.
- Thiết kế thành công ứng dụng dự báo thời tiết, đáp ứng được yêu cầu của người sử dụng.
- Giao diện website được thiết kế đơn giản, thân thiện và dễ sử dụng.

Về mặt ứng dụng:

- Xem được thời tiết địa điểm hiện tại của thiết bị.
- Xem được thời tiết tại địa điểm nhập vào bất kỳ.
- Xem được thời tiết các ngày kế tiếp.

Tuy nhiên vẫn còn những hạn chế:

- Ứng dụng còn ít dữ liệu, chưa đáp ứng đủ nhu cầu thông tin từ người sử dụng
- Còn ít chức năng, và chưa tối ưu khả năng trải nghiệm của người dùng

Hướng phát triển: Vì thời gian không cho phép và kiến thức về thiết kế web còn hạn chế nên chúng em chưa tìm hiểu rõ và kỹ hơn về đề tài này. Vì thế đề tài cũng còn có nhiều thiếu sót, mong thầy giúp đỡ chúng em nhiều hơn và em sẽ tiếp tục tìm hiểu sâu hơn về công nghệ này, nhằm mục đích thiết kế website: Giao diện đẹp, thân thiện với người dùng, có nhiều chức năng và tương tác tốt hơn với người dùng; thiết kế, bổ sung các tính năng còn thiếu hay chưa thật sự hoàn chỉnh trong quá trình sử dụng.

TÀI LIỆU THAM KHẢO

- [1]. <https://viblo.asia/p/kien-truc-cua-he-dieu-hanh-android-PaLGdYdaelX>
- [2]. https://vi.wikipedia.org/wiki/Android_Studio
- [3]. <https://laptrinhstuduy.wordpress.com/2014/04/27/activity-trong-android/>