



TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO KẾT THÚC MÔN HỌC

Lập trình android nâng cao

Xây dựng ứng dụng “My Weather”

Giảng viên hướng dẫn: Trương Bá Thái

Sinh viên thực hiện: Nhóm 3

1. Võ Hoàng Phú

2. Phạm Khánh Duy

3. Đoàn Xuân Trung

4. Trần Phan Thế Bảo

5. Mai Đức Hưng

Khoa: Công nghệ thông tin

Khóa: 2014

Ngày 27/07/2017

NHẬT KÝ HOẠT ĐỘNG NHÓM 3

STT	HỌ TÊN	CÔNG VIỆC	TỰ ĐÁNH GIÁ	NHÓM ĐÁNH GIÁ	KÝ TÊN
1	Võ Hoàng Phú	Nghiên cứu đề tài Viết SRS Viết SDS Viết báo cáo Viết test case Làm màn hình 1,2 Làm navigation drawer menu Báo cáo cuối kỳ	20%	20%	
2	Phạm Khánh Duy	Nghiên cứu đề tài Viết SRS Viết SDS Làm màn hình 4 Lấy API Đọc và load chuỗi JSON Hỗ trợ code, tổng hợp Thiết kế custom listview Báo cáo cuối kỳ	20%	20%	
3	Trần Phan Thế Bảo	Nghiên cứu đề tài Viết SRS Viết SDS Làm màn hình 6 Thiết kế custom listview Database Báo cáo cuối kỳ	20%	20%	
4	Đoàn Xuân Trung	Nghiên cứu đề tài Viết SRS Viết SDS Auto complete Làm màn hình 5 Báo cáo cuối kỳ	20%	20%	
5	Mai Đức Hưng	Nghiên cứu đề tài Viết SRS Viết SDS Lấy API Đọc và load chuỗi JSON Làm màn hình 3 Báo cáo cuối kỳ	20%	20%	

Mục lục

CHƯƠNG 1. MỞ ĐẦU	4
1.1 Giới thiệu môn học và nhóm thực hiện.....	4
1.2 Mô tả ứng dụng	4
1.3 Ứng dụng trong cuộc sống	4
CHƯƠNG 2 PHÂN TÍCH CẤU TRÚC HỆ THỐNG	5
2.1 Phân tích thiết kế hệ thống.....	5
2.2 Thiết kế hệ thống.....	19
CHƯƠNG 3. CÀI ĐẶT	34
3.1 Màn hình home	34
3.2 Màn hình xem thông tin thời tiết tại TP.HCM	35
3.3 Màn hình xem thời tiết 7 ngày tiếp theo	36
3.4 Màn hình tìm kiếm.....	37
3.5 Màn hình lịch sử.....	38
CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC.....	40
4.1 Kết quả đạt được	40
4.2 Kết luận	40

CHƯƠNG 1. MỞ ĐẦU

1.1 Giới thiệu môn học và nhóm thực hiện

Môn di động 2 là môn nâng cao của di động 1. Môn học này giúp sinh viên phát triển kỹ năng lập trình, tự tìm kiếm, cập nhật kiến thức mới giúp áp dụng cho các dự án thực tế.

Kiến thức mới: API, JSON, splash screen, navigation drawer, tìm hiểu và sử dụng thư viện Volley, thư viện Picasso.

Thành viên thực hiện:

- Phạm Khánh Duy (Leader)
- Võ Hoàng Phú (Tester)
- Mai Đức Hưng
- Đoàn Xuân Trung
- Trần Phan Thế Bảo

1.2 Mô tả ứng dụng

- Tên ứng dụng: myWeather
- Cần kết nối mạng
- Android 4.0+
- Thông tin thời tiết(API) tại trang openweathermap.org
- Xem thời tiết mọi nơi trên thế giới
- Thời tiết trong 7 ngày tiếp theo
- Lưu lịch sử xem

1.3 Ứng dụng trong cuộc sống

Với việc điện thoại thông minh, thiết bị mạng ngày càng phát triển vì vậy các ứng dụng có kết nối mạng được sử dụng liên tục.

Ứng dụng dự báo thời tiết giúp dự báo thời tiết mọi lúc mọi nơi chỉ cần 1 điện thoại android và 1 kết nối mạng.

CHƯƠNG 2 PHÂN TÍCH CẤU TRÚC HỆ THỐNG

2.1 Phân tích thiết hệ thống

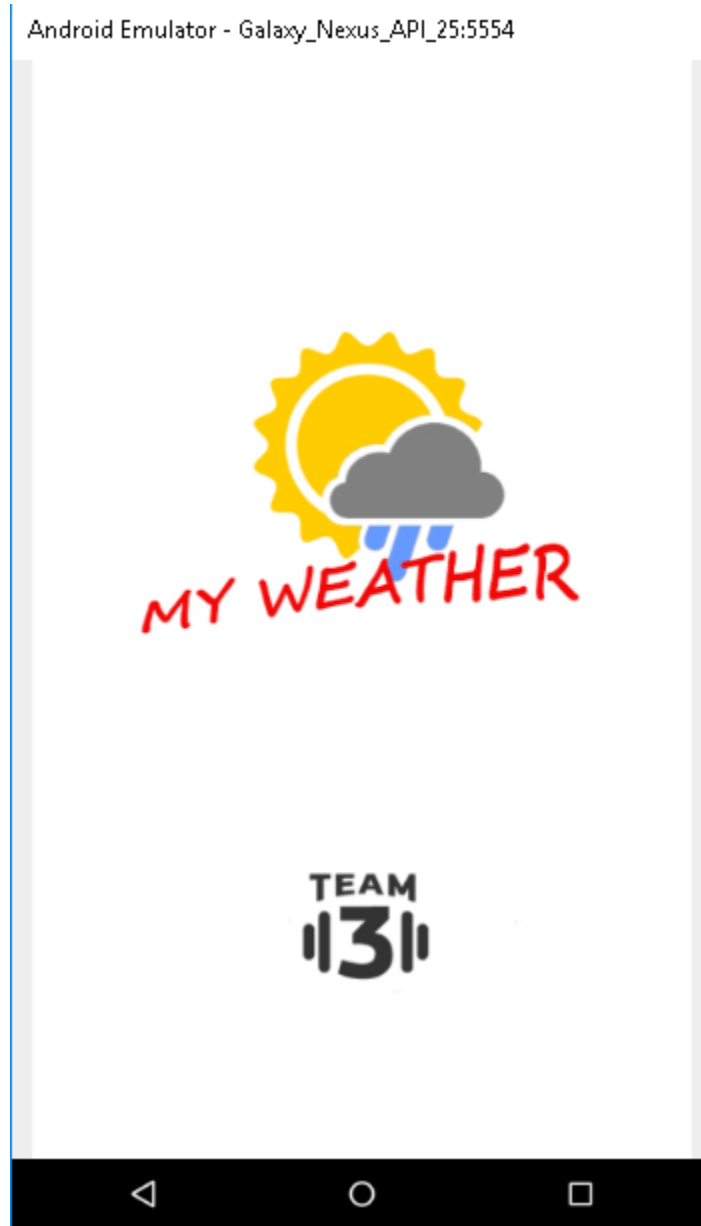
Tổng quan: Đây là ứng dụng dùng để theo dõi tình hình thời tiết, nhiệt độ. Xem thời tiết tại địa điểm hiện hành của thiết bị hoặc địa điểm do người dùng nhập vào, báo trước thời tiết các ngày tiếp theo.

Phạm vi đề tài:

- Theo dõi thời tiết và tìm kiếm
- Lập trình trên android studio v2.3.3
- Hệ điều hành android: 4.0+
- Ứng dụng lấy API từ Website: <http://openweathermap.org>
- Thiết bị thử nghiệm: Galaxy Nesus, galaxy dous
- Độ phân giải màn hình 480x800px 5inch

a. Chức năng #1: Màn hình giao diện 1(Võ Hoàng Phú)

- **Giao diện người dùng**



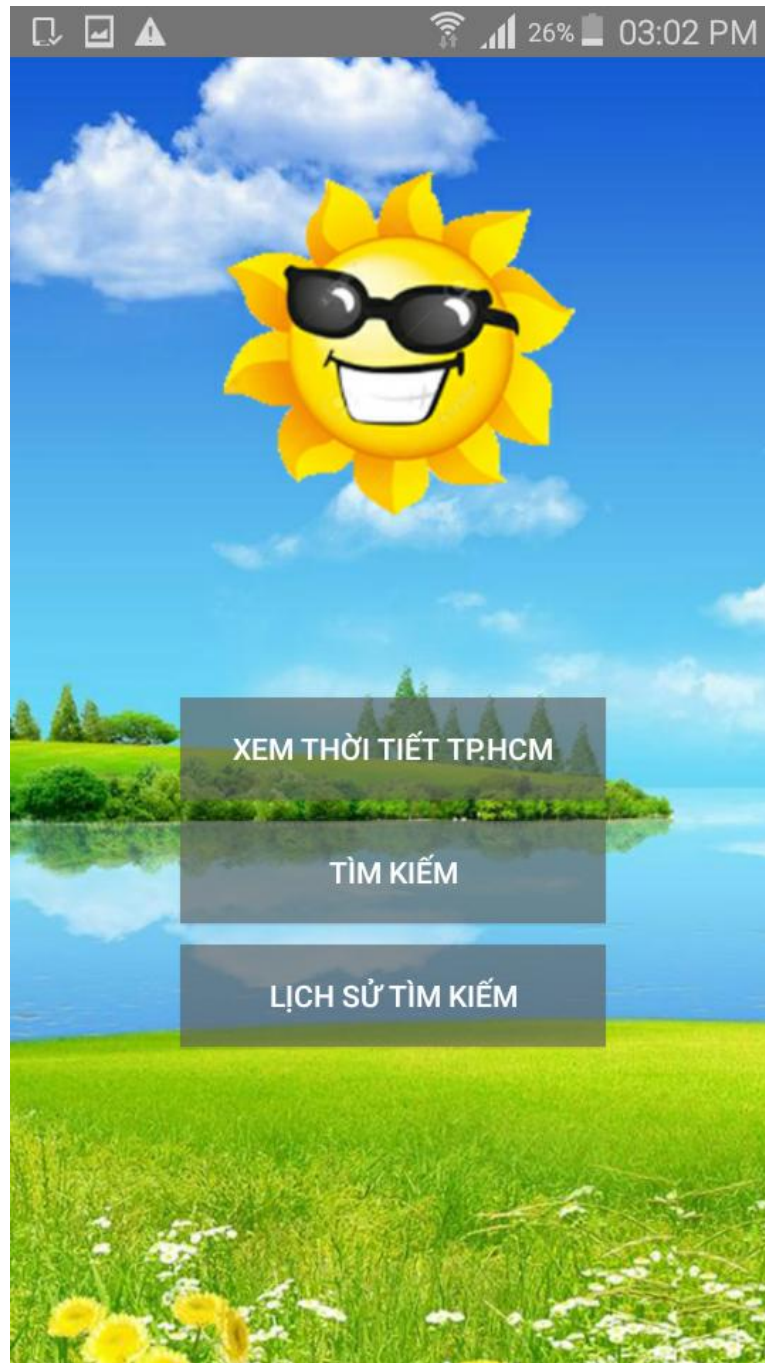
Hình 1 Màn hình splash

○ **Yêu cầu chức năng**

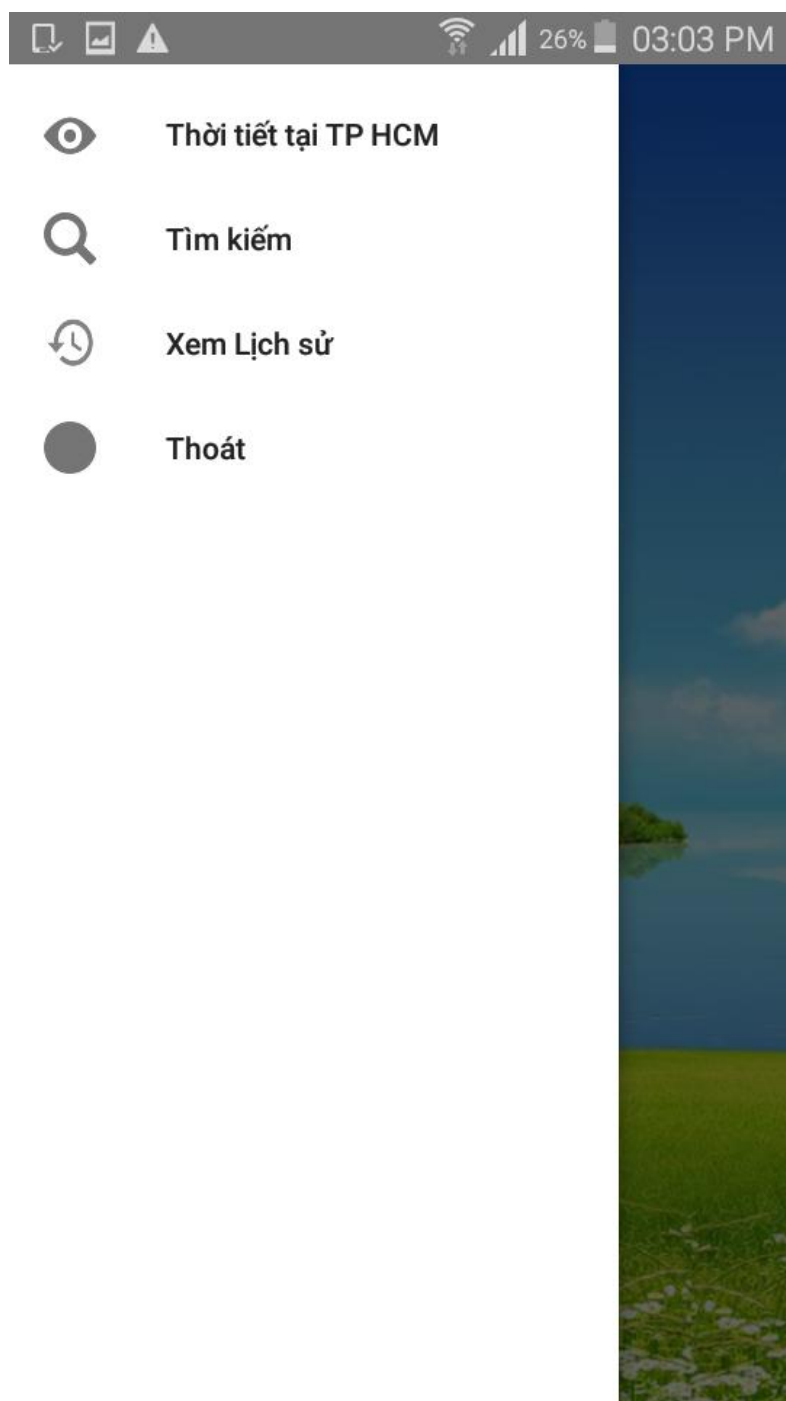
Tiêu đề	Mô tả	Quy trình	Kết quả
Màn hình splash	Hiển thị logo	Sau 3s chuyển qua màn hình chính	Hoàn thành

b. Chức năng #2: Màn hình giao diện 2(Võ Hoàng Phú)

- **Giao diện người dùng**



Hình 1 Màn hình home



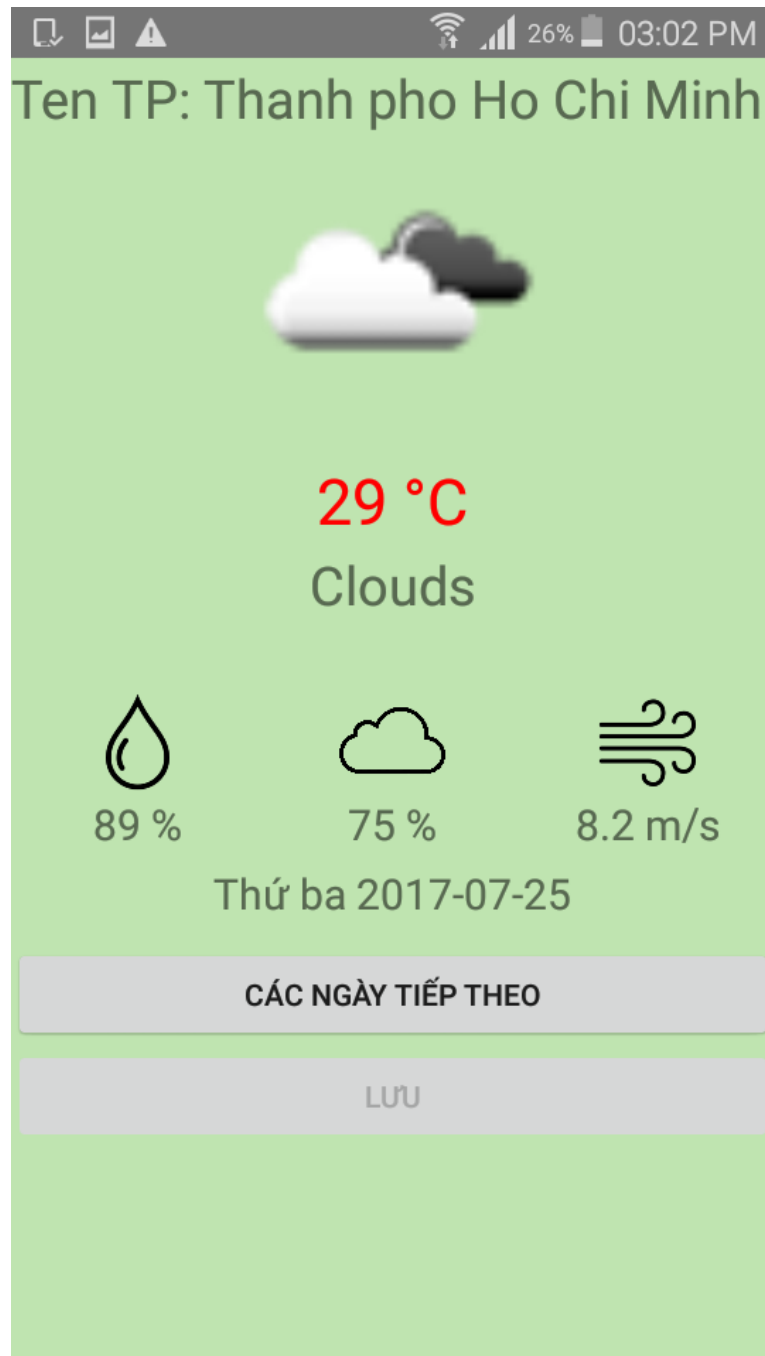
Hình 1 Navigation drawer menu

○ **Yêu cầu chức năng**

Tiêu đề	Mô tả	Quy trình	Kết quả
Màn hình home	Màn hình chính có: - 1 image background - 3 button	Thiết kế	Hoàn thành
Button “Xem thời tiết”	Chuyển sang giao diện “Xem thời tiết”	Click	Chuyển sang giao diện “Xem thời tiết”
Button “Tìm kiếm”	Chuyển sang giao diện “Tìm kiếm”	Click	Chuyển sang giao diện “Tìm kiếm”
Button “Xem lịch sử”	Chuyển sang giao diện “Lịch sử”	Click	Chuyển sang giao diện “Lịch sử”
Navigation drawer menu “Thời tiết tại tp Hồ Chí Minh”	Chuyển sang giao diện “thời tiết tại tp Hồ Chí Minh”	Click	Chuyển sang giao diện “thời tiết tại tp Hồ Chí Minh”
Navigation drawer menu “Tìm kiếm”	Chuyển sang giao diện “Tìm kiếm”	Click	Chuyển sang giao diện “Tìm kiếm”
Navigation drawer menu “Xem lịch sử”	Chuyển sang giao diện “Xem lịch sử”	Click	Chuyển sang giao diện “Xem lịch sử”
Navigation drawer menu “Thoát”	Thoát	Click	Thoát

c. Chức năng #3: Màn hình giao diện 3 (Mai Đức Hưng)

- Màn hình thời tiết



Hình 1 Màn hình xem thời tiết

○ **Yêu cầu chức năng**

Tiêu đề	Mô tả	Qui trình	Kết quả
Màn hình thời tiết hiện tại	Màn hình chính có: 6 textview 4 imageview 2 button	N/A	N/A
Button “Các ngày tiếp theo”	Chuyển qua màn hình các ngày tiếp theo	Click	Chuyển qua màn hình các ngày tiếp theo
Button “Lưu”	Lưu vào lịch sử xem khi được truyền từ màn hình tìm kiếm qua. Chức năng lưu sẽ bị tắt khi xem màn hình thời tiết tại tp.Hồ Chí Minh	Click	Lưu vào lịch sử xem
Image thời tiết, độ ẩm, gió, mây.	Lấy dữ liệu từ API	Click	N/A

d. Chức năng #4: Màn hình giao diện 4 (Phạm Khánh Duy)

- **Màn hình các ngày tiếp theo**



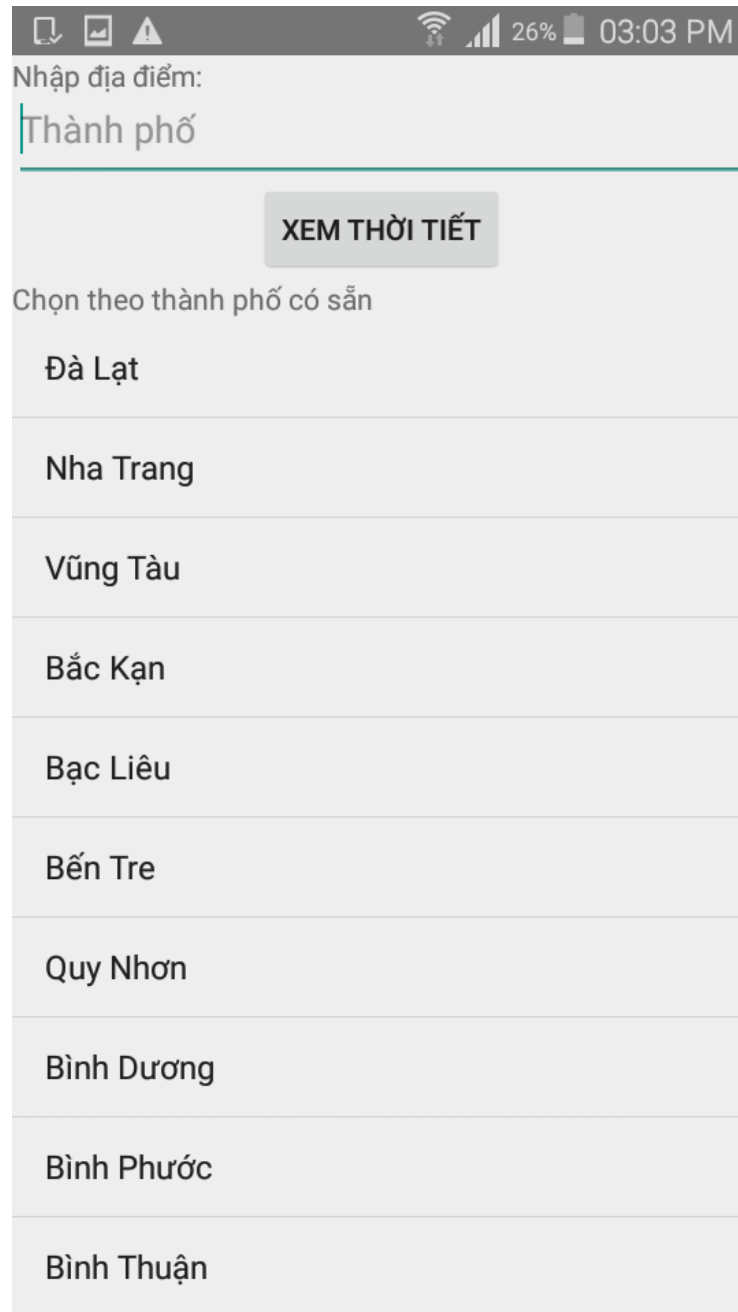
Hình 1 Màn hình Xem thời tiết các ngày tiếp theo

○ **Yêu cầu chức năng**

Tiêu đề	Mô tả	Qui trình	Kết quả
Layout	Màn hình chính có: - 1 custom listview - 1 textview -1 button	N/A	N/A
Custom listview	Listview hiển thị thông tin thời tiết của các ngày trong tuần và các thông tin như nhiệt độ, lượng mưa,....	Load dữ liệu API	Hiển thị danh sách các ngày trong tuần và dự báo
imageview “back”	Quay về màn hình xem thời tiết	Click	quay về màn hình xem thời tiết
Button “Home”	Quay về màn hình Home	Click	Quay về màn hình home

e. Chức năng #5: Màn hình giao diện 5 (Đoàn Xuân Trung)

- **Màn hình Tìm kiếm**



Nhập địa điểm:

Thành phố

XEM THỜI TIẾT

Chọn theo thành phố có sẵn

Đà Lạt
Nha Trang
Vũng Tàu
Bắc Kạn
Bạc Liêu
Bến Tre
Quy Nhơn
Bình Dương
Bình Phước
Bình Thuận

Hình 1 màn hình tìm kiếm

○ **Yêu cầu chức năng**

Tiêu đề	Mô tả	Quy trình	Kết quả
Màn hình tìm kiếm thời tiết	Màn hình chính có: 1 Textbox để nhập tên thành phố 1 button để xem thông tin 1 Listview danh sách các thành phố cho trước	N/A	Hoàn thành
Autocomplete”Nhập thành phố tìm kiếm”	Nhập tên thành phố xem thời tiết	view	Hiện thị gợi ý tên thành phố cần tìm
Button “Xem thời tiết”	Xem thông tin thời tiết địa điểm đã nhập, chuyển sang màn hình xem thời tiết	Click	Chuyển sang màn hình xem thời tiết, hiển thị thông tin thời tiết thành phố đã tìm
Listview “danh sách”	Hiện thị tất cả thành phố cho sẵn	Click	

f. Chức năng #6: Màn hình giao diện 6 (Trần Phan Thế Bảo)

- **Màn hình lịch sử**



Hình 1 Màn hình lịch sử

○ **Yêu cầu chức năng**

Tiêu đề	Mô tả	Qui trình	Kết quả
Màn hình lịch sự	Màn hình chính có: - 1 textview - 1 listview - 2 Button	Thiết kế	Hoàn thành
Button “Home”	Chuyển sang giao diện “Màn hình chính”	Click	Chuyển sang giao diện “Màn hình chính”
Button “Xóa”	Dùng để xóa lịch sử tìm kiếm	Click	Các thông tin lịch sử được xóa
Listview	Hiển thị các thông tin của lịch sử tìm kiếm	View	Hiển thị các thông tin của lịch sử tìm kiếm

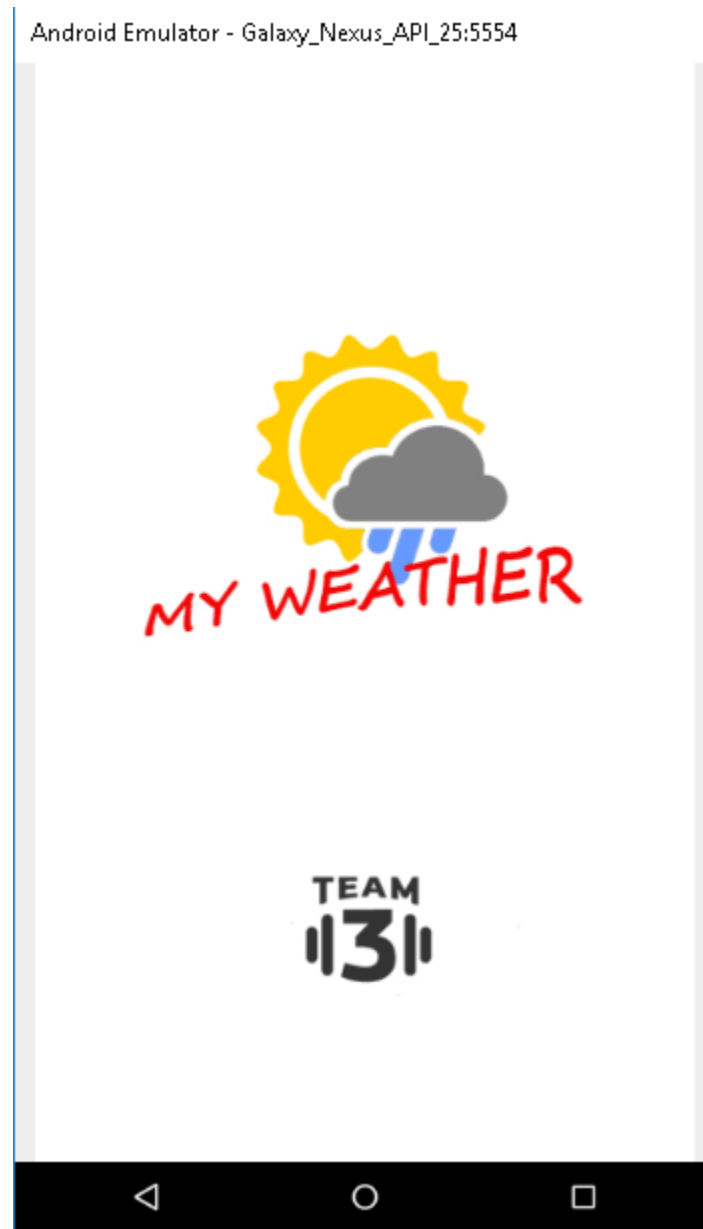
2.2 Thiết kế hệ thống

Tổng quan: Tài liệu dùng để cho nhóm trưởng phân công công việc cho các thành viên giúp phân công công việc 1 cách dễ dàng, bố trí việc làm và phân bổ công việc hợp lý. Mô tả chi tiết công việc cho từng màn hình.

Yêu cầu chức năng:

a. Màn hình splash

- Giao diện người dùng



Hình 1 Splash screen

○ **Yêu cầu chức năng**

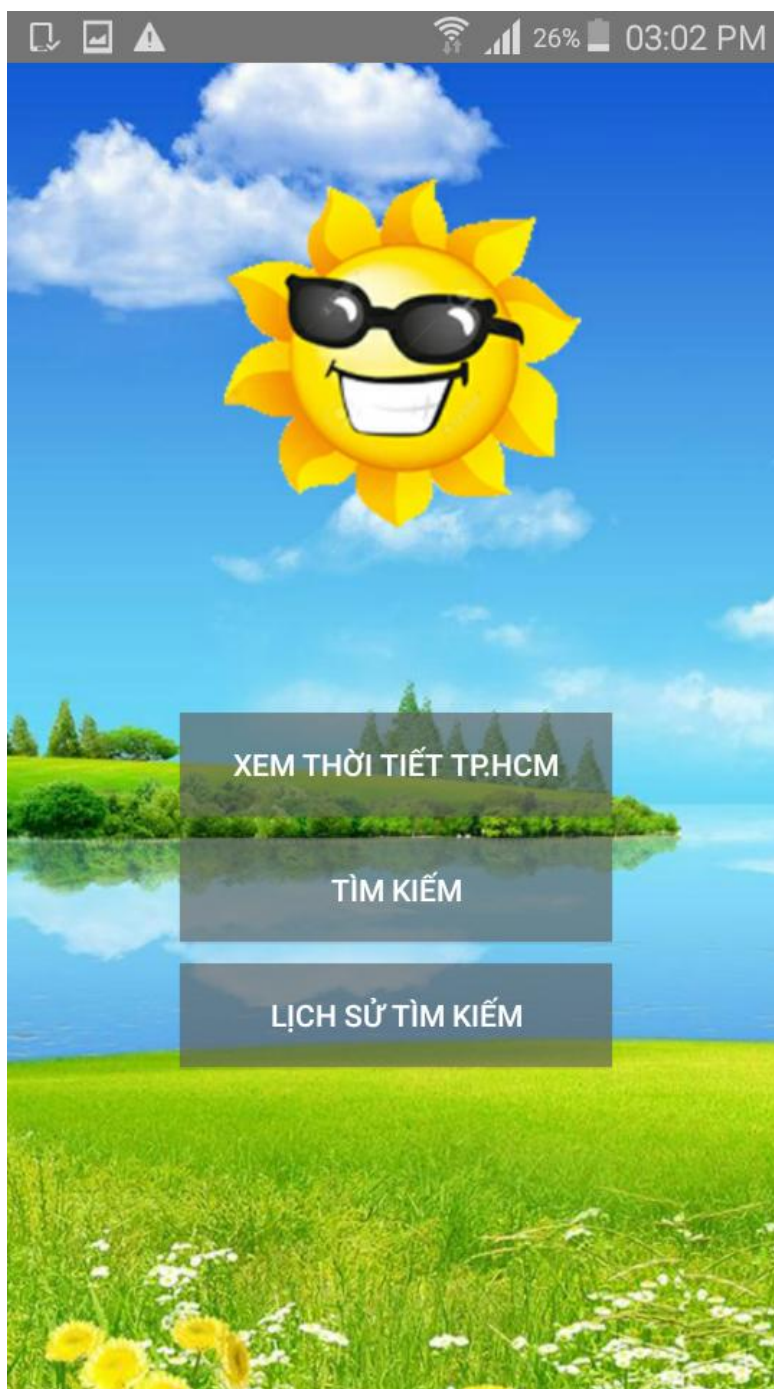
Tiêu đề	Mô tả	Quy trình	Kết quả
Splash screen	Màn hình Splasd	Sau 3s chuyển màn hình	Chuyển qua màn hình chính

Làm thế nào: Splash screen

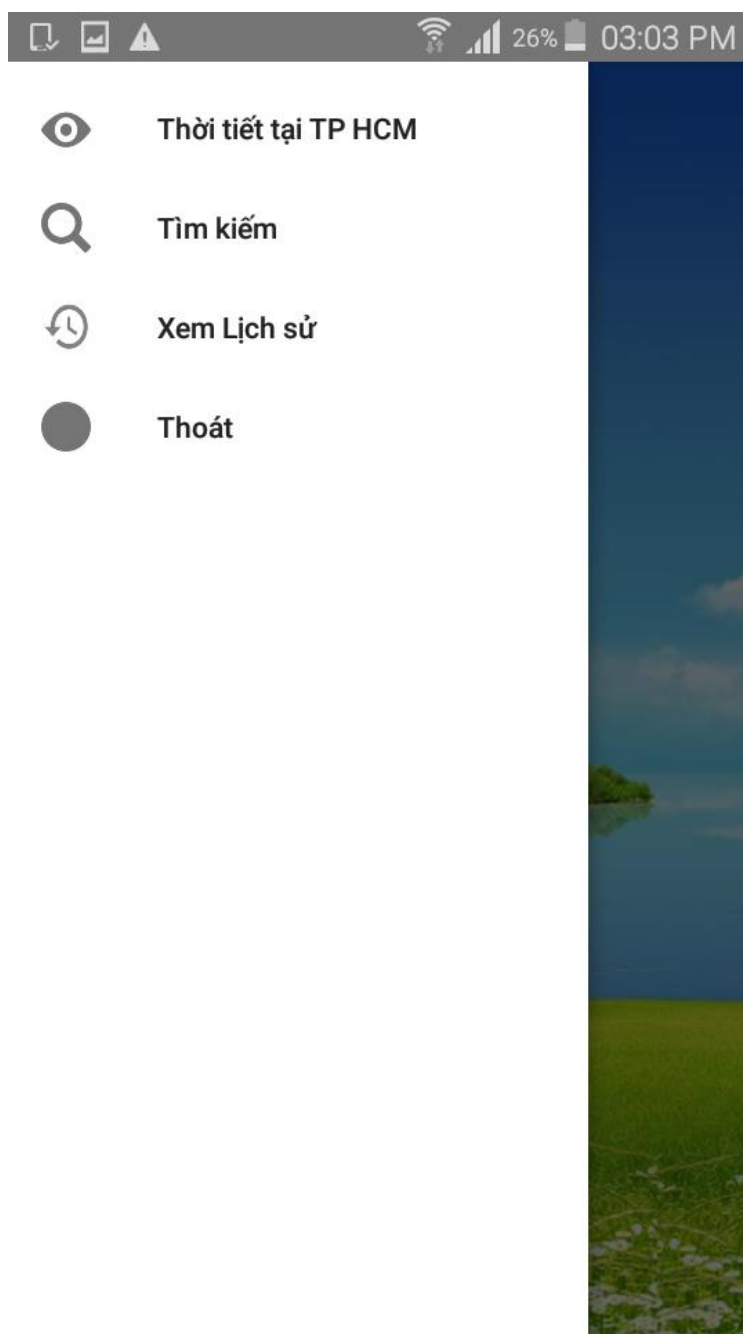
- Chạy ứng dụng
- Chương trình sẽ chạy màn hình Splash
- Sau 3s sẽ tự động chuyển qua màn hình chính

b. Màn hình chính

- **Giao diện người dùng**



Hình 2 Màn hình chính



Hình 1 Navigation drawer menu

○ **Yêu cầu chức năng**

Tiêu đề	Mô tả	Quy trình	Kết quả
Màn hình home	Màn hình chính có: - 1 image background - 3 button	Thiết kế	Hoàn thành
Button “Xem thời tiết”	Chuyển sang giao diện “Xem thời tiết”	Click	Chuyển sang giao diện “Xem thời tiết”
Button “Tìm kiếm”	Chuyển sang giao diện “Tìm kiếm”	Click	Chuyển sang giao diện “Tìm kiếm”
Button “Xem lịch sử”	Chuyển sang giao diện “Lịch sử”	Click	Chuyển sang giao diện “Lịch sử”
Navigation drawer menu “Thời tiết tại tp Hồ Chí Minh”	Chuyển sang giao diện “thời tiết tại tp Hồ Chí Minh”	Click	Chuyển sang giao diện “thời tiết tại tp Hồ Chí Minh”
Navigation drawer menu “Tìm kiếm”	Chuyển sang giao diện “Tìm kiếm”	Click	Chuyển sang giao diện “Tìm kiếm”
Navigation drawer menu “Xem lịch sử”	Chuyển sang giao diện “Xem lịch sử”	Click	Chuyển sang giao diện “Xem lịch sử”
Navigation drawer menu “Thoát”	Thoát	Click	Thoát

Làm thế nào: Màn hình chính

ImageButton “Thời tiết”

- Click button thời tiết
- Ứng dụng chuyển qua màn hình thời tiết

ImageButton “Nhập địa điểm”

- Click button thời tiết
- Ứng dụng chuyển qua màn hình Nhập địa điểm

ImageButton “Lịch sử”

- Click button thời tiết
- Ứng dụng chuyển qua màn hình thời tiết

ImageButton “Exit”

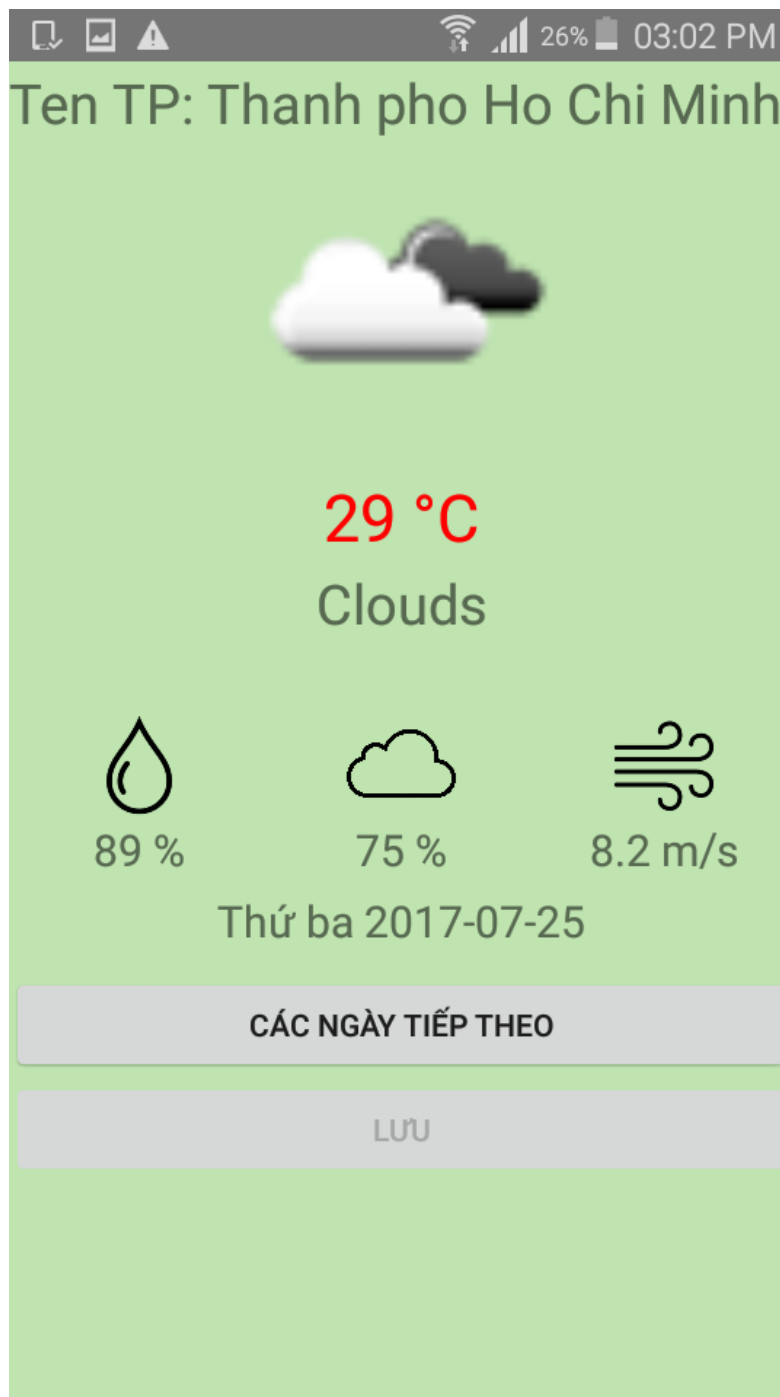
- Click button thời tiết
- Thoát ứng dụng

Navigation drawer menu - CLICK

- Navigation drawer menu “Thời tiết tại tp Hồ Chí Minh”
- Navigation drawer menu “Tìm kiếm”
- Navigation drawer menu “Xem lịch sử”
- Navigation drawer menu “Thoát”

c. Màn hình thời tiết

- **Giao diện người dùng**



Hình 1 Màn hình thời tiết chi tiết

○ **Yêu cầu chức năng**

Tiêu đề	Mô tả	Qui trình	Kết quả
Màn hình thời tiết hiện tại tp.HCM	Màn hình chính có: 6 textview 4 imageview 2 button	N/A	N/A
Button “Các ngày tiếp theo”	Chuyển qua màn hình các ngày tiếp theo	Click	Chuyển qua màn hình các ngày tiếp theo
Button “Lưu”	Màn hình thời tiết tại tp.Hồ Chí Minh chức năng lưu sẽ bị tắt. Lưu lại kết quả tìm kiếm vào lịch sử xem chức năng lưu sẽ hoạt động khi được truyền từ màn hình tìm kiếm qua	Click	Lưu vào lịch sử xem
Image thời tiết, độ ẩm, gió, mây.	Lấy dữ liệu từ api	Click	N/A

Làm thế nào: Màn hình thời tiết tại Hồ Chí Minh

Màn hình chính

- Ứng dụng đưa ra màn hình những thông tin thời tiết:
- Tên thành phố
- Hình ảnh minh họa trạng thái thời tiết
- Nhiệt độ
- Trạng thái thời tiết
- Độ ẩm, % lượng mây, tốc độ gió

Button “Home”

- Click button Home
- Ứng dụng chuyển qua màn hình chính

Button “các ngày tiếp theo”

- Click button các ngày tiếp theo
- Ứng dụng sẽ chuyển sang màn hình giao diện các ngày tiếp theo

d. Màn hình các ngày tiếp theo

- **Giao diện người dùng**



Hình 1 Màn hình thời tiết các ngày tiếp theo

○ **Yêu cầu chức năng**

Tiêu đề	Mô tả	Quy trình	Kết quả
Layout	Màn hình chính có: - 1 custom listview - 1 textview - 1 imageView - 1 Button	N/A	N/A
Custom listview	Listview hiển thị thông tin thời tiết của các ngày trong tuần và các thông tin như nhiệt độ, lượng mưa,....	Load dữ liệu API	Hiển thị danh sách 7 ngày trong tuần và dự báo
ImageView “Back”	Quay về màn hình xem thời tiết	Click	quay về màn hình xem thời tiết
Button “Home”	Quay về màn hình Home	Click	Quay về màn hình Home

Làm thế nào: Màn hình thời tiết các ngày tiếp theo

Lấy API thời tiết, api link sẽ trả về 1 chuỗi JSON, đọc chuỗi JSON đó ra 1 custom listview

Listview hiển thị thông tin thời tiết 7 ngày tiếp theo trong tuần của địa điểm tìm kiếm (mặc định sẽ là TP Hồ Chí Minh nếu không tìm kiếm theo địa điểm)

Mỗi ngày trong list sẽ hiển thị các thông tin:

- Trạng thái thời tiết (chữ cam)
- Hình minh họa trạng thái thời tiết
- Nhiệt độ cao nhất trong ngày (chữ đỏ), nhiệt độ thấp nhất trong ngày (chữ xanh)

1 ImageView quay về màn hình xem thời tiết

1 Button Home quay về màn hình Home

e. Màn hình chọn tìm kiếm

○ **Giao diện người dùng**

Nhập địa điểm:

Thành phố

XEM THỜI TIẾT

Chọn theo thành phố có sẵn

Đà Lạt

Nha Trang

Vũng Tàu

Bắc Kạn

Bạc Liêu

Bến Tre

Quy Nhơn

Bình Dương

Bình Phước

Bình Thuận

Hình 1 màn hình nhập tìm kiếm

○ **Yêu cầu chức năng**

Tiêu đề	Mô tả	Quy trình	Kết quả
Màn hình tìm kiếm thời tiết	Màn hình chính có: 1 Textbox để nhập tên thành phố 1 button để xem thông tin 1 Listview danh sách các thành phố cho trước	N/A	Hoàn thành
Autocomplete”Nhập thành phố tìm kiếm”	Nhập tên thành phố xem thời tiết	view	Hiện thị gợi ý tên thành phố cần tìm
Button “Xem thời tiết”	Xem thông tin thời tiết địa điểm đã nhập, chuyển sang màn hình xem thời tiết	Click	Chuyển sang màn hình xem thời tiết, hiển thị thông tin thời tiết thành phố đã tìm
Listview “danh sách”	Hiện thị tất cả thành phố cho sẵn	Click	Hiện thị tất cả thành phố cho sẵn

Làm thế nào: Màn hình tìm kiếm

Button “Tìm kiếm”

- Click button tìm kiếm
- Ứng dụng chuyển qua màn hình nhập địa điểm tìm kiếm

AutoCompleteTextView “Tìm kiếm”

- Nhập tên thành phố muốn xem thời tiết, hiện gợi ý tìm kiếm

Button “xem ”

- Click button xem
- Ứng dụng chuyển qua màn hình thời tiết

Listview

- Hiện thị tên của một vài tỉnh, thành phố lớn cho người dùng tìm kiếm nhanh

f. Màn hình lịch sử

- **Giao diện người dùng**



Hình 1 Màn hình lịch sử

○ **Yêu cầu chức năng**

Tiêu đề	Mô tả	Quy trình	Kết quả
Màn hình lịch sự	Màn hình chính có: - 1 textview - 1 listview - 2 Button	Thiết kế	Hoàn thành
Button “Home”	Chuyển sang giao diện “Màn hình chính”	Click	Chuyển sang giao diện “Màn hình chính”
Button “Xóa”	Dùng để xóa lịch sử tìm kiếm	Click	Các thông tin lịch sử được xóa
Listview	Hiển thị các thông tin của lịch sử tìm kiếm	View	Hiển thị các thông tin của lịch sử tìm kiếm

Làm thế nào: Màn hình lịch sử

Button “Home”

- Click button Home
- Ứng dụng chuyển qua màn hình chính

Button “Xóa”

- Click button thời tiết
- Ứng dụng xóa các thông tin thời tiết đang hiện trên màn hình

Listview

- Ứng dụng đưa ra màn hình những thông tin thời tiết bao gồm : tên thành phố , nhiệt độ , thông tin thời tiết tại đó , đã từng được tìm kiếm được lưu trong cơ sở dữ liệu

CHƯƠNG 3. CÀI ĐẶT

3.1 Màn hình home

Sử dụng Navigation drawer menu

```

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

    btnXem = (Button) findViewById(R.id.buttonXem);
    btnTim = (Button) findViewById(R.id.buttonTim);
    btnLichsu = (Button) findViewById(R.id.buttonLichsu);

    toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    // Find our drawer view
    mDrawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    // Find our drawer view
    nvDrawer = (NavigationView) findViewById(R.id.nvView);
    // Setup drawer view
    setupDrawerContent(nvDrawer);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // The action bar home/up action should open or close the drawer.
    switch (item.getItemId()) {
        case android.R.id.home:
            mDrawer.openDrawer(GravityCompat.START);
            return true;
    }
    return super.onOptionsItemSelected(item);
}

private void setupDrawerContent(NavigationView navigationView) {
    navigationView.setNavigationItemSelectedListener(
        (menuItem) -> {
            selectDrawerItem(menuItem);
            return true;
        }
    );
}

public void selectDrawerItem(MenuItem menuItem) {
    switch(menuItem.getItemId()) {
        case R.id.nav_first_fragment:
            Intent src = new Intent(Home.this, MainActivity.class);
            startActivity(src);
            break;
        case R.id.nav_second_fragment:
            Intent src2 = new Intent(Home.this, TimKiem.class);
            startActivity(src2);
            break;
        case R.id.nav_third_fragment:

```

3.2 Màn hình xem thông tin thời tiết tại TP.HCM

Hàm GetCurrentWeatherData:

```
//Hàm gửi response và trả về chuỗi JSON (sử dụng thư viện Volley)
public void GetCurrentWeatherData(String data){
    RequestQueue requestQueue = Volley.newRequestQueue(MainActivity.this);

    //Đọc url, biến data gán tên thành phố
    String url = "http://api.openweathermap.org/data/2.5/weather?q="+data+"&units=metric&appid=defcb81b976af17f632cef86d49739e2";
    StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
        (response) -> {

            //Log.d("ketqua", response);
            try {
                JSONObject jsonObject = new JSONObject(response);

                //Gán JSON tên thành phố vào txtTen
                String name = jsonObject.getString("name");
                txtTen.setText("Ten TP: " + name);

                //Gán JSON ngày vào txtDay
                String day = jsonObject.getString("dt");
                long lv = Long.valueOf(day);
                Date date = new Date(lv*1000L);
                SimpleDateFormat simpleDateFormat = new SimpleDateFormat("EEEE yyyy-MM-dd");
                String Day = simpleDateFormat.format(date);
                txtDay.setText(Day);

                //Gán JSON vào txtStatus
                JSONArray jsonArrayWeather = jsonObject.getJSONArray("weather");
                JSONObject jsonObjectWeather = jsonArrayWeather.getJSONObject(0);
                String status = jsonObjectWeather.getString("main");
                txtStatus.setText(status);

                //Gán JSON vào image icon (Sử dụng thư viện Picasso)
                String icon = jsonObjectWeather.getString("icon");
                Picasso.with(MainActivity.this).load("http://openweathermap.org/img/w/"+icon+".png").into(imgIcon);

                //Gán JSON vào txtTemp
                JSONObject jsonObjectMain = jsonObject.getJSONObject("main");
                String nhietdo = jsonObjectMain.getString("temp");
                Double a = Double.valueOf(nhietdo);
                String Nhietdo = String.valueOf(a.intValue());
                txtTemp.setText(Nhietdo + " °C");

                //Gán JSON vào txtHumidity
                String doam = jsonObjectMain.getString("humidity");
                txtHumidity.setText(doam + "%");

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

                //Gán JSON vào txtCloud
                JSONObject jsonObjectCloud = jsonObject.getJSONObject("clouds");
                String may = jsonObjectCloud.getString("all");
                txtCloud.setText(may + "%");
            } catch (JSONException e) {
                e.printStackTrace();
            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
            }
        }
    );
}
```

```

    });
    requestQueue.add(stringRequest);
}

```

3.3 Màn hình xem thời tiết 7 ngày tiếp theo

Hàm Get7DayData

```

private void Get7DayData(String data) {
    String url = "http://api.openweathermap.org/data/2.5/forecast/daily?q="+data+"&units=metric&cnt=7&appid=defcb81b976af17f632cef86d49739e2";
    RequestQueue requestQueue = Volley.newRequestQueue(CacNgayTiepTheo.this);
    StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
        (response) → {
            //Log.d("Ket qua", "Json" + response);
            try {
                JSONObject jsonObject = new JSONObject(response);
                JSONObject jsonObjectCity = jsonObject.getJSONObject("city");
                String name = jsonObjectCity.getString("name");
                txtTenThanhPho.setText(name);

                JSONArray jsonArrayList = jsonObject.getJSONArray("list");
                for (int i = 0; i < jsonArrayList.length(); i++) {
                    JSONObject jsonObjectList = jsonArrayList.getJSONObject(i);

                    String ngay = jsonObjectList.getString("dt");
                    long lv = Long.valueOf(ngay);
                    Date date = new Date(lv*1000L);
                    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("EEEE yyyy-MM-dd");
                    String Day = simpleDateFormat.format(date);

                    JSONObject jsonObjectTemp = jsonObjectList.getJSONObject("temp");
                    String max = jsonObjectTemp.getString("max");
                    String min = jsonObjectTemp.getString("min");

                    Double a = Double.valueOf(max);
                    String NhietdoMax = String.valueOf(a.intValue());
                    Double b = Double.valueOf(min);
                    String NhietdoMin = String.valueOf(b.intValue());

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

                    mangthoitiets.add(new ThoiTiet(Day, status, icon, NhietdoMax, NhietdoMin));
                }
                customAdapter_thoitiets.notifyDataSetChanged();
            } catch (JSONException e) {
                e.printStackTrace();
            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
            }
        });
    requestQueue.add(stringRequest);
}

```

3.4 Màn hình tìm kiếm

Sử dụng AutoCompleteTextView

```
public void addEvent() {

    edtSearch.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after) {
            adapterThanhpho.getFilter().filter(s);
        }

        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {

        }

        @Override
        public void afterTextChanged(Editable s) {

        }
    });
    lv.setOnItemClickListener((parent, view, position, id) → {
        edtSearch.setText(arr[position]);
    });

    btnSearch.setOnClickListener((v) → {
        String city = edtSearch.getText().toString();
        Intent intent = new Intent(TimKiem.this, MainActivity.class);
        intent.putExtra("thanhpho", city);
        startActivity(intent);
    });
}
```

3.5 Màn hình lịch sử

Hàm tạo database và insert dữ liệu vào database (tạo sự kiện cho nút “Lưu” màn hình xem thời tiết)

```
private void OpenDB() {
    // TODO Auto-generated method stub
    database = openOrCreateDatabase("History.db", MODE_PRIVATE, null);
    String sql = "CREATE TABLE IF NOT EXISTS lichsu (thanhpho varchar, nhietdo varchar, status varchar)";
    database.execSQL(sql);
}

//Hàm xử lý nhập
public void xulyNhap() {

    if (insert(edtTen.getText().toString(), txtTemp.getText().toString(), txtStatus.getText().toString())) {
        Toast.makeText(getApplicationContext(), "Đã lưu", Toast.LENGTH_LONG)
            .show();
    } else {
        Toast.makeText(getApplicationContext(), "Không thể lưu", Toast.LENGTH_LONG)
            .show();
    }
}

//Hàm nhập vào database
public boolean insert(String thanhpho, String nhietdo, String status) {
    ContentValues value = new ContentValues();
    value.put("thanhpho", thanhpho);
    value.put("nhietdo", nhietdo);
    value.put("status", status);

    if (database.insert("lichsu", null, value) > 0) {
        return true;
    }
    return false;
}
```

Hàm GetData và hàm xóa (sự kiện cho nút Xóa và add vào Listview màn hình lichsutimkiem)

```
private void getData() {  
    // TODO Auto-generated method stub  
    manglichsu = new ArrayList<LichsuData>();  
    Cursor c = database.query("lichsu", null, null, null, null, null, null);  
    c.moveToFirst();  
    while (!c.isAfterLast()) {  
        manglichsu.add(new LichsuData(c.getString(0), c.getString(1), c.getString(2)));  
        c.moveToNext();  
    }  
}  
  
public void Clear() {  
    String smg = "";  
    int res = database.delete("lichsu", null, null);  
  
    if (res == 0) {  
        smg = "Không thể xóa";  
    } else {  
        smg = "Đã xóa";  
    }  
    Toast.makeText(LichSuList.this, smg, Toast.LENGTH_LONG).show();  
}  
  
public void xulyXoa()  
{  
    for(int i=lv.getChildCount()-1;i>=0;i--)  
    {  
        View v = lv.getChildAt(i);  
        manglichsu.remove(i);  
    }  
    adapter_lichsu.notifyDataSetChanged();  
}
```

CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC

4.1 Kết quả đạt được

- Hệ thống trả về được các thông tin về thời tiết theo các mục đã chọn theo địa điểm
- Hệ thống tìm kiếm thời tiết theo từng khu vực
- Hệ thống các địa điểm được cho trước
- Hệ thống thời tiết của 7 ngày tiếp theo
- Hệ thống lịch sử

4.2 Kết luận

Những việc đã làm được:

- Lấy được thông tin thời tiết
- Lấy thông tin thời tiết 7 ngày tiếp theo
- Lưu lịch sử trong SQL

Những việc chưa làm được:

- Tên địa điểm chưa chính xác 100%(gần giống)
- Chưa xử lý được lưu trùng tên
- Chưa lưu được ngày tháng trong database
- Giao diện chưa tối ưu
- Chưa làm được màn hình load động
- Chưa xử lý khi không có internet