

TRƯỜNG ĐẠI HỌC NÔNG LÂM THÀNH PHỐ HỒ CHÍ MINH

KHOA: CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN HỌC

DATA WAREHOUSE

Giáo viên hướng dẫn: Nguyễn Đức Công Song

Các thành viên trong nhóm: Nguyễn Bảo Tâm – 23130286

Phạm Tấn Đức – 23130068

Lê Tiến Hoàng – 23130116

Trần Nguyễn Thanh Tú – 23130365

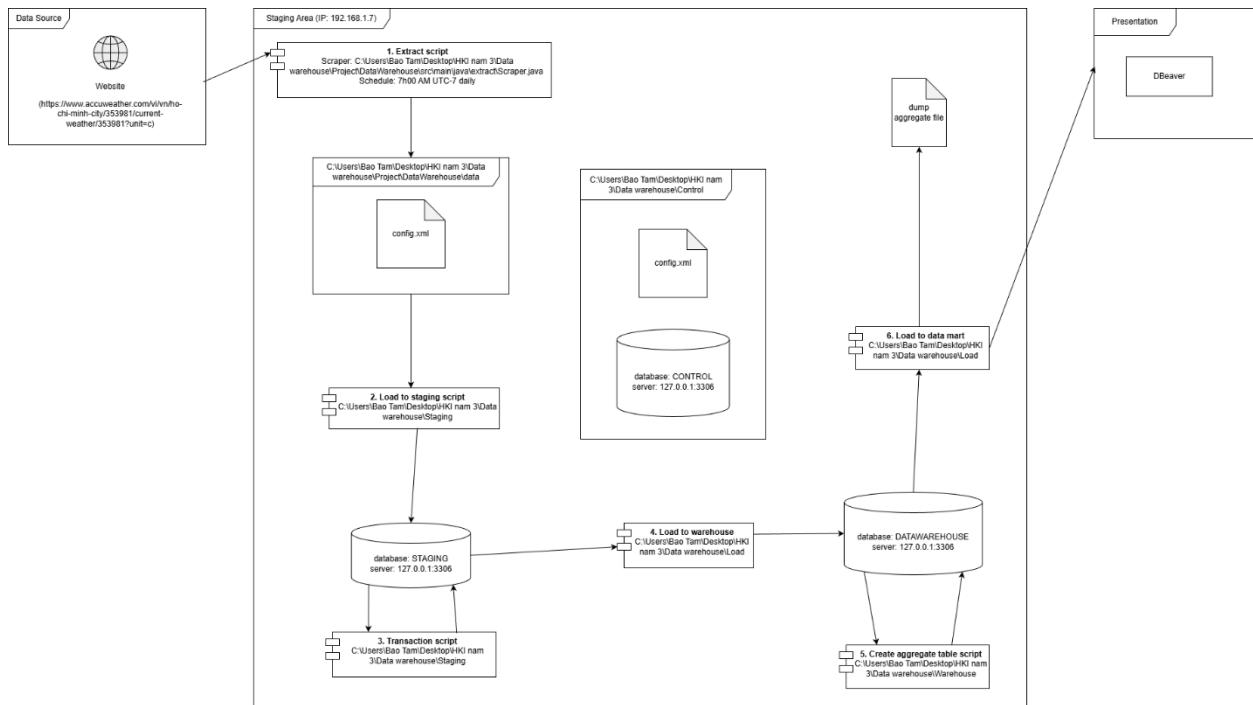
Sinh viên thực hiện: Nguyễn Bảo Tâm – 23130286

MỤC LỤC

1.	KIẾN TRÚC HỆ THỐNG	3
1.1.	Sơ đồ kiến trúc hệ thống.....	3
1.2.	Thông tin các quy trình	3
2.	THÔNG TIN DỮ LIỆU NGUỒN.....	4
2.1.	Nguồn accuweather - https://www.accuweather.com/vi/vn/ho-chi-minh-city/353981/current-weather/353981?unit=c	4
2.1.1.	Thông tin các thuộc tính.....	4
2.1.2.	Thông tin lưu trữ	4
3.	THÔNG TIN HỆ THỐNG	5
3.1.	Hướng dẫn triển khai hệ thống.....	5
3.1.1.	Yêu cầu chung cho các hệ thống.....	5
3.1.2.	Lịch chạy hệ thống	5
3.1.3.	Hướng dẫn chạy thủ công.....	6
3.2.	Ký hiệu các trạng thái	6
3.2.1.	Ký hiệu trạng thái của file	6
3.2.2.	Ký hiệu trạng thái của process	6
3.3.	Quy trình xử lý lỗi	7
4.	CẤU TRÚC CƠ SỞ DỮ LIỆU	7
4.1.	Staging Area	7
4.1.1.	Control database.....	7
4.1.2.	Straging database.....	8
4.1.3.	Warehouse database.....	8
4.2.	Data presentation.....	8
4.2.1.	Mart I	8
5.	QUY TRÌNH CÁ NHÂN ĐẢM NHẬN	9
5.1.	Quy trình <?>.....	9
5.1.1.	Workflow.....	9
5.1.2.	Chi tiết mã trong quy trình	10

1. KIẾN TRÚC HỆ THỐNG

1.1. Sơ đồ kiến trúc hệ thống.



1.2. Thông tin các quy trình

STT	Tên quy trình	Mô tả	Thành viên thực hiện
1	Lấy dữ liệu từ nguồn về file	Lấy dữ liệu thời tiết từ website accuweather về thành các file .csv lưu trữ trên github	Nguyễn Bảo Tâm
2	Load dữ liệu từ file .csv vào db.staging	Load toàn bộ dữ liệu từ file .csv vào bảng tạm db.staging	Trần Nguyễn Thanh Tú
3	Transform dữ liệu trong db.staging	Chuyển đổi dữ liệu, chuẩn hóa trong bảng tạm sang bảng chính trong db.staging	Phạm Tân Đức
4	Load dữ liệu từ db.staging vào db.warehouse	Load dữ liệu từ bảng chính trong db.staging sang db.warehouse	Lê Tiên Hoàng
5	Tạo aggregate table trong db.warehouse	Tạo aggregate table từ các bảng dim trong db.warehouse	Phạm Tân Đức
6	Load dữ liệu từ aggregate table vào data mart	Dump dữ liệu aggregate table thành các file và thực hiện load vào data mart	Phạm Tân Đức

2. THÔNG TIN DỮ LIỆU NGUỒN

2.1. Nguồn accuweather - <https://www.accuweather.com/vi/vn/ho-chi-minh-city/353981/current-weather/353981?unit=c>

2.1.1. Thông tin các thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả	Ví dụ
1	FullDate	DATETIME	Thời điểm hệ thống ghi nhận dữ liệu thời tiết (bao gồm ngày và giờ)	2025-09-26 01:09:50
2	Weekday	VARCHAR(20)	Thứ trong tuần(Từ thứ 2 đến chủ nhật)	Thứ Sáu
3	Day	VARCHAR(20)	Ngày và tháng	2 tháng 9
4	Temperature	DECIMAL(4,1)	Nhiệt độ hiện tại (°C)	26
5	UVValue	DECIMAL(4,2)	Chỉ số tia cực tím (UV value)	0.6
6	UVLevel	NVARCHAR(20)	Mức độ UV đi kèm mô tả (ví dụ: “Thấp”, “Trung bình”, “Cao”)	Thấp
7	WindDirection	NVARCHAR(10)	Hướng gió (ví dụ: “TTN” nghĩa là Tây Tây Nam)	TTN
8	WindSpeed	DECIMAL(5,2)	Tốc độ gió (km/h)	9
9	Humidity	DECIMAL(4,1)	Độ ẩm tương đối của không khí (%)	94
10	DewPoint	DECIMAL(4,1)	Nhiệt độ điểm sương (°C) – chỉ mức độ hơi ẩm trong không khí	25
11	Pressure	DECIMAL(6,2)	Áp suất khí quyển (millibar – mb)	1009
12	Cloud	DECIMAL(5,2)	Tỷ lệ mây che phủ bầu trời (%)	91
13	Visibility	DECIMAL(5,2)	Tầm nhìn xa (km)	16
14	CloudCeiling	INT	Độ cao trần mây (mét) – khoảng cách từ mặt đất đến lớp mây thấp nhất	5500

2.1.2. Thông tin lưu trữ

- Định dạng lưu trữ:** Dữ liệu được thu thập từ AccuWeather và lưu dưới dạng file. csv
- Đường dẫn lưu trữ:** \data
- Ví dụ mẫu dữ liệu:**

Thời gian,Thứ,Ngày,Nhiệt độ,UV,Gió,Độ ẩm,Điểm sương,Khí áp,Mây,Tầm nhìn,Trần mây

2025-09-26 01:09:50,Thứ Sáu, 26 tháng 9,26°C,0.6 (Thấp),TTN 9 km/h,94%,25° C,↑ 1009 mb,91%,16 km,5500 m

- Phương thức thu thập dữ liệu:**
 - Sử dụng script để lấy dữ liệu định kỳ từ website
 - Thời gian chạy: mỗi ngày 1 lần (7h00 AM UTC-7)

3. THÔNG TIN HỆ THỐNG

3.1. Hướng dẫn triển khai hệ thống

3.1.1. Yêu cầu chung cho các hệ thống

- Yêu cầu về phần cứng**

Thành phần	Thông số tối thiểu	Thông số khuyến nghị	Ghi chú
CPU	Intel Core i3 hoặc tương đương	Intel Core i5 / AMD Ryzen 5 trở lên	Đảm bảo hiệu suất khi xử lý ETL và truy vấn
RAM	4 GB	8 GB hoặc cao hơn	Hỗ trợ chạy đồng thời Python và SQL Server
Dung lượng ổ cứng	10 GB trống	≥ 50 GB	Lưu trữ dữ liệu thô, staging và warehouse
Kết nối mạng	Internet ổn định	Tốc độ ≥ 10 Mbps	Phục vụ truy cập và thu thập dữ liệu AccuWeather

- Yêu cầu về phần mềm**

Thành phần	Phiên bản / Công cụ	Mục đích sử dụng
Hệ điều hành	Windows 10 / 11 (64-bit)	Môi trường triển khai hệ thống
Ngôn ngữ lập trình	Java (JDK 17 trở lên)	Crawl dữ liệu và xử lý ETL
Thư viện Java	mysql-connector-j-9.5.0.jar	Hỗ trợ lấy dữ liệu, trích xuất và làm sạch
CSDL tạm (Staging Database)	MySQL 8.0	Lưu dữ liệu trung gian
CSDL kho (Warehouse Database)	MySQL 8.0	Lưu dữ liệu phân tích dài hạn
Công cụ trực quan hóa	Dbeaver	Xây dựng báo cáo, dashboard
Trình quản lý CSDL	MySQL Workbench	Quản trị, truy vấn và kiểm tra dữ liệu
Trình lập lịch (Scheduler)	Windows Task Scheduler	Tự động hóa quá trình ETL định kỳ

3.1.2. Lịch chạy hệ thống

Quy trình	Mô tả	Tần suất	Thời điểm chạy	Ghi chú
Lấy dữ liệu từ nguồn về file	Thu thập dữ liệu thời tiết từ AccuWeather	1 lần/ngày	7h00 AM UTC-7	Dữ liệu được lưu dưới dạng CSV nằm trong \data
Load dữ liệu từ file .csv vào db.staging				

Transform dữ liệu trong db.staging				
Load dữ liệu từ db.staging vào db.warehouse				
Tạo aggregate table trong db.warehouse				
Load dữ liệu từ aggregate table vào data mart				

3.1.3. Hướng dẫn chạy thủ công

3.2. Ký hiệu các trạng thái

3.2.1. Ký hiệu trạng thái của file

KH	Ý nghĩa	Mô tả	Ghi chú
FS	File Success	File được đánh dấu đã xử lý xong	
FF	File Fail	File lỗi	

3.2.2. Ký hiệu trạng thái của process

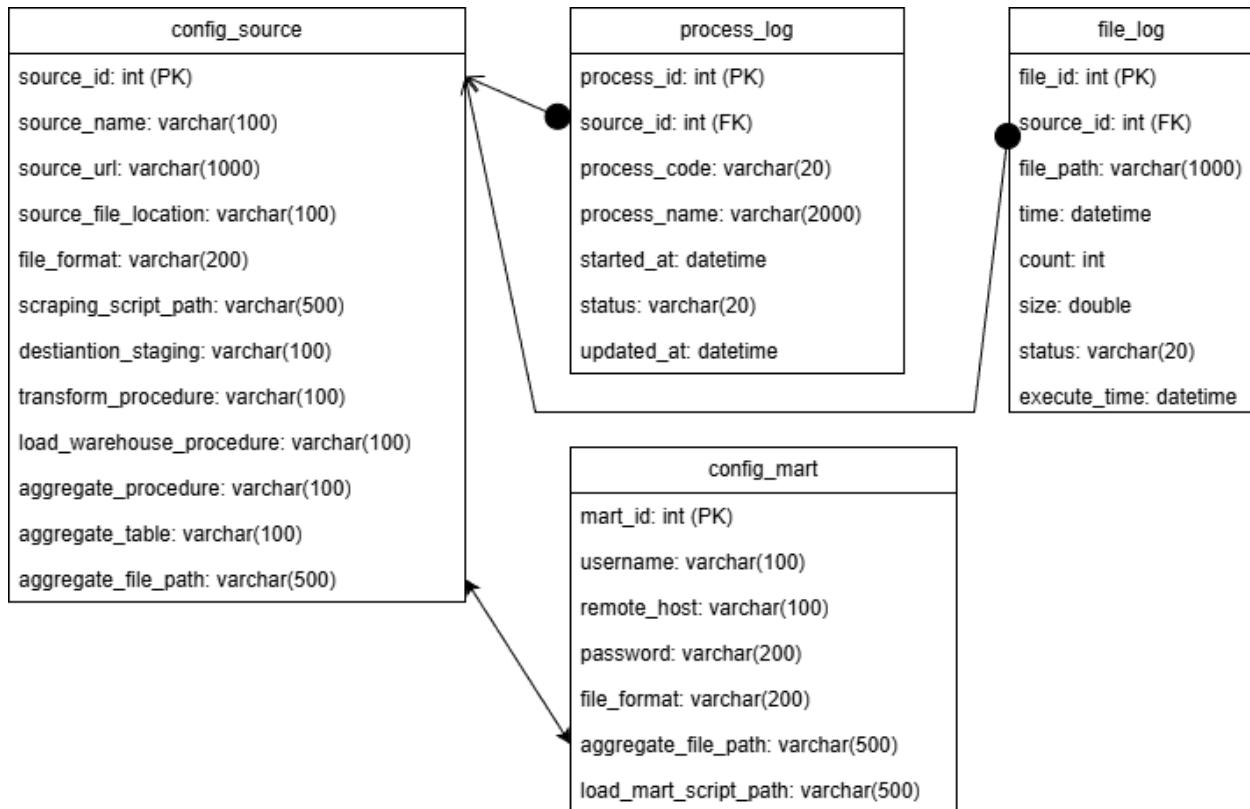
KH	Ý nghĩa	Mô tả	Ghi chú
ER	Extract Ready	Sẵn sàng để extract	
EO	Extract Ongoing	Đang thực hiện extract	
EF	Extract Fail	Extract thất bại	
TR	Transform Ready	Sẵn sàng để transform	
TO	Transform Ongoing	Đang thực hiện transform	
TF	Transform Fail	Transform thất bại	
LR	Load Ready	Sẵn sàng để load vào warehouse	
LO	Load Ongoing	Đang thực hiện load vào warehouse	
LF	Load Fail	Load vào warehouse thất bại	
SC	Success	Hoàn thành ETL trong staging	
F	Fail	Thất bại ETL trong staging	

3.3. Quy trình xử lý lỗi

4. CẤU TRÚC CƠ SỞ DỮ LIỆU

4.1. Staging Area

4.1.1. Control database



- **Giải thích thành phần:**

- 1) Bảng config_source

Tên cột	Kiểu dữ liệu	Mô tả
source_id	INT (PK)	ID của nguồn dữ liệu
source_name	VARCHAR(100)	Tên nguồn dữ liệu
source_url	VARCHAR(1000)	URL của nguồn dữ liệu
source_file_location	VARCHAR(100)	Đường dẫn file nguồn
file_format	VARCHAR(200)	Định dạng file
scraping_script_path	VARCHAR(500)	Đường dẫn đến script thu thập dữ liệu
destination_staging	VARCHAR(100)	Đường dẫn đến thư mục staging
transform_procedure	VARCHAR(100)	Procedure thực hiện biến đổi dữ liệu
load_warehouse_procedure	VARCHAR(100)	Procedure load dữ liệu vào data warehouse
aggregate_procedure	VARCHAR(100)	Procedure thực hiện tổng hợp dữ liệu
aggregate_table	VARCHAR(100)	Bảng aggregate sau khi ETL
aggregate_file_path	VARCHAR(500)	Đường dẫn lưu bảng aggregate

- 2) Bảng config_mart

Tên cột	Kiểu dữ liệu	Mô tả
mart_id	INT (PK)	ID của Data Mart
username	VARCHAR(100)	Tên đăng nhập
remote_host	VARCHAR(100)	Địa chỉ máy chủ hoặc host của Data Mart.
password	VARCHAR(200)	Mật khẩu
file_format	VARCHAR(200)	Định dạng file
aggregate_file_path	VARCHAR(500)	Đường dẫn lưu bảng aggregate
load_mart_script_path	VARCHAR(500)	Đường dẫn script load dữ liệu vào Mart.

3) Bảng file log

Tên cột	Kiểu dữ liệu	Mô tả
file_id	INT (PK)	ID của file
source_id	INT (FK)	ID của nguồn dữ liệu
file_path	VARCHAR(1000)	Đường dẫn file đang được xử lý.
time	DATETIME	Thời gian bắt đầu chạy của file
file_format	VARCHAR(200)	Định dạng file
count	INT	Số bản ghi trong file
size	DOUBLE	Kích thước file
status	VARCHAR(20)	Trạng thái file
execute_time	DATETIME	Thời điểm file được xử lý xong

4) Bảng process log

Tên cột	Kiểu dữ liệu	Mô tả
process_id	INT (PK)	ID của process
source_id	INT (FK)	ID của nguồn dữ liệu
process_code	VARCHAR(20)	Mã tiến trình
process_name	VARCHAR(2000)	Tên tiến trình
started_at	DATETIME	Thời điểm tiến trình bắt đầu.
status	VARCHAR(20)	Trạng thái tiến trình
updated_at	DATETIME	Thời điểm trạng thái tiến trình được cập nhật lần cuối.

4.1.2. Staging database

4.1.3. Warehouse database

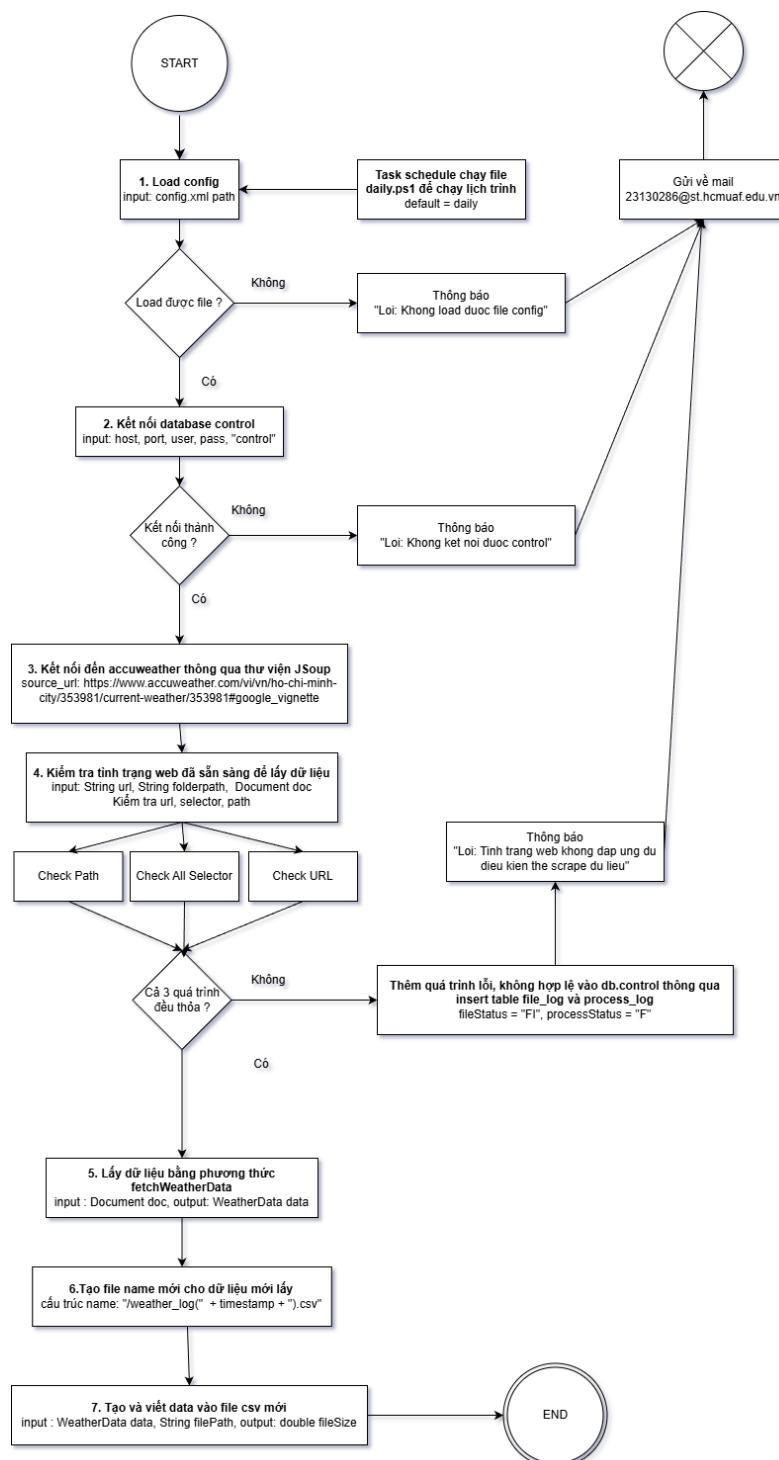
4.2. Data presentation

4.2.1. Mart I

5. QUY TRÌNH CÁ NHÂN ĐẨM NHẬN

5.1. Quy trình < Lấy dữ liệu từ nguồn về file >

5.1.1. Workflow



5.1.2. Chi tiết mã trong quy trình

0. File daily.ps1: Task schedule sẽ đọc file này mỗi ngày và chạy file tes.class

```
# =====
# File: daily.ps1
# Chạy chương trình Java
# =====

# Đường dẫn tới JDK
$JAVA_HOME = "C:\Program Files\Java\jdk-17"
$JAVA = "$JAVA_HOME\bin\java.exe"

# Thư mục project target
$PROJECT = "C:\Users\Bao Tam\Desktop\HKI nam 3\Data warehouse\Project\DataWarehouse\target"

# Classpath: main classes + test classes + tất cả jar trong dependency
$CLASSPATH = "$PROJECT\classes;$PROJECT\test-classes;$PROJECT\dependency\*"

# Class chính (full qualified name)
$MAIN_CLASS = "com.example.DataWarehouse.Test"

# Chạy chương trình
& "$JAVA" -cp "$CLASSPATH" "$MAIN_CLASS"
```

1. Load config: tải lên file config.xml để đọc cấu hình

```
public static Config readConfig() {
    try {
        // Tao một đối tượng XmlMapper để đọc và ánh xạ dữ liệu XML sang đối tượng Java
        XmlMapper xmlMapper = new XmlMapper();

        // Đọc file "config.xml" và chuyển đổi nội dung của nó thành đối tượng Config
        return xmlMapper.readValue(new File("config.xml"), Config.class);
    } catch (Exception e) {
        // Nếu có lỗi xảy ra (ví dụ: file không tồn tại, định dạng XML sai)
        // Gửi email thông báo lỗi với tiêu đề và nội dung chi tiết lỗi
        EmailUtils.send("Lỗi hệ thống: không thể đọc file config.xml", "Chi tiết lỗi: " + e.getMessage());
    }

    // Nếu không thể đọc file hoặc xảy ra lỗi, trả về null
    return null;
}
```

2. Kết nối database control: kết nối tới database control

```
// Phương thức kết nối tới database MySQL
public static Connection connectDB(String host, int port, String user, String pass, String name) {
    try {
        // Tao chuỗi kết nối JDBC với thông tin host, port, tên database và các tham số kết nối
        String url = "jdbc:mysql://" + host + ":" + port + "/" + name +
        "?useSSL=false&serverTimezone=UTC";

        // Tao kết nối tới database bằng DriverManager
        Connection conn = DriverManager.getConnection(url, user, pass);

        // Thông báo kết nối thành công ra console
        System.out.println("Kết nối MySQL thành công!");

        // Trả về đối tượng Connection để thực hiện các thao tác với database
        return conn;
    } catch (SQLException e) {
        // Nếu kết nối thất bại, gửi email thông báo lỗi với chi tiết
        EmailUtils.send("Lỗi hệ thống: không thể kết nối database: " + name, "Chi tiết lỗi: " +
        e.getMessage());

        // Trả về null nếu không thể kết nối
        return null;
    }
}
```

3. Kết nối đến accuweather thông qua thư viện Jsoup: Sử dụng thư viện Jsoup để tạo Document qua url.

```
// Phương thức kết nối tới một website và lấy nội dung HTML
public static Document connectToWebsite(String url) {
    try {
        // Sử dụng Jsoup để kết nối tới URL và lấy toàn bộ nội dung HTML
        return Jsoup.connect(url).get();
    } catch (Exception e) {
        // Nếu kết nối thất bại, gửi email thông báo lỗi với chi tiết
        EmailUtils.send("Lỗi hệ thống: không thể kết nối tới website: " + url,
        "Chi tiết lỗi: " + e.getMessage());
    }
    // Trả về null nếu không thể kết nối
    return null;
}
```

4. Kiểm tra tình trạng web đã sẵn sàng để lấy dữ liệu: Kiểm tra 3 thành phần là checkPath, checkAllSelector và checkURL

```

// Phương thức kiểm tra xem dữ liệu có sẵn sàng để thực hiện bước Extract hay không
public static boolean checkExtract(Config config, Document doc) throws Exception {
    // Kiểm tra các điều kiện cần thiết để extract dữ liệu:
    // 1. URL của nguồn dữ liệu có thể truy cập được
    // 2. Thư mục lưu trữ dữ liệu nguồn tồn tại và hợp lệ
    // 3. Các selector trong tài liệu HTML hợp lệ và tồn tại
    if (Scraper.checkURL(config.source.source_url)
        && Scraper.checkPath(config.source.source_folder_path)
        && Scraper.checkAllSelectors(doc)) {
        return true; // Nếu tất cả điều kiện đều đúng, trả về true
    }

    // Nếu bất kỳ điều kiện nào không thỏa, trả về false
    return false;
}

```

- **checkPath**

```

// Phương thức kiểm tra folder lưu file có tồn tại và có quyền ghi hay không
public static boolean checkPath(String folderPath) throws IOException {
    // Tạo đối tượng File đại diện cho folder
    File folder = new File(folderPath);

    // Nếu folder không tồn tại, thử tạo folder mới
    // Nếu không thể tạo được folder, in lỗi và trả về false
    if (!folder.exists() && !folder.mkdirs()) {
        System.err.println("✖ Không thể tạo folder: " + folderPath);
        return false;
    }

    // Kiểm tra quyền ghi file bằng cách tạo 1 file tạm trong folder
    File testFile = new File(folderPath + "/test_write.tmp");
    try (FileWriter fw = new FileWriter(testFile)) {
        fw.write("test"); // Ghi 1 nội dung nhỏ vào file
    }
    // Xóa file tạm sau khi test
    testFile.delete();

    // Nếu mọi thứ thành công, in thông báo hợp lệ và trả về true
    System.out.println("✓ folderPath hợp lệ.");
    return true;
}

```

- **checkURL**

```

// Phương thức kiểm tra URL có thể truy cập được hay không
public static boolean checkURL(String url) throws IOException {
    // Kết nối tới URL sử dụng Jsoup với timeout 5 giây
    // ignoreHttpErrors(true) để không ném lỗi nếu HTTP code khác 200
    Connection.Response resp = Jsoup.connect(url)
        .timeout(5000)
        .ignoreHttpErrors(true)
        .execute();

    // Nếu HTTP status code khác 200, in lỗi và trả về false
    if (resp.statusCode() != 200) {
        System.err.println("❌ URL không phản hồi hoặc sai: HTTP " + resp.statusCode());
        return false;
    }

    // Nếu URL hợp lệ và phản hồi HTTP 200, in thông báo hợp lệ và trả về true
    System.out.println("✅ url hợp lệ.");
    return true;
}

```

- **checkAllSelectors**

```

// Phương thức kiểm tra tất cả các selector cần thiết có tồn tại trong tài liệu HTML hay không
public static boolean checkAllSelectors(Document doc) {
    // Duyệt qua từng cặp key-value trong REQUIRED_SELECTORS
    // key = tên trường, value = CSS selector
    for (Map.Entry<String, String> entry : REQUIRED_SELECTORS.entrySet()) {
        String field = entry.getKey();      // Tên trường dữ liệu
        String selector = entry.getValue(); // CSS selector tương ứng

        // Nếu selector không tìm thấy phần tử trong Document, in lỗi và trả về false
        if (doc.select(selector).isEmpty()) {
            System.err.println("❌ Missing selector for field: " + field + " | Selector: " + selector);
            return false;
        }
    }

    // Nếu tất cả selector đều tìm thấy, in thông báo hợp lệ và trả về true
    System.out.println("✅ Tất cả selector đều hợp lệ.");
    return true;
}

```

5. Lấy dữ liệu bằng phương thức fetchWeatherData: input: Document doc, output: WeatherData data

```
// Phương thức lấy dữ liệu thời tiết từ Document HTML
public static WeatherData fetchWeatherData(Document doc) throws Exception {

    // Tạo đối tượng WeatherData mới để lưu thông tin thời tiết
    WeatherData data = new WeatherData();

    // Gán thời gian hiện tại vào dữ liệu
    data.time = getCurrentTime();

    // Duyệt qua tất cả các selector cần thiết trong REQUIRED_SELECTORS
    for (Map.Entry<String, String> entry : REQUIRED_SELECTORS.entrySet()) {
        String field = entry.getKey();          // Tên trường dữ liệu
        String selector = entry.getValue();     // CSS selector tương ứng

        // Lấy giá trị text từ selector trong Document
        String value = doc.select(selector).text();

        // Gán giá trị vào đúng trường của đối tượng WeatherData dựa trên tên field
        switch (field) {
            case "dayDate":
                data.dayDate = value;
                break;
            case "temperature":
                data.temperature = value;
                break;
            case "uvIndex":
                data.uvIndex = value;
                break;
            case "wind":
                data.wind = value;
                break;
            case "humidity":
                data.humidity = value;
                break;
            case "dewPoint":
                data.dewPoint = value;
                break;
            case "pressure":
                data.pressure = value;
                break;
            case "cloudCover":
                data.cloudCover = value;
                break;
            case "visibility":
                data.visibility = value;
                break;
            case "ceiling":
                data.ceiling = value;
                break;
        }
    }

    // Trả về đối tượng WeatherData đã được điền đầy đủ dữ liệu
    return data;
}
```

6.Tạo file name mới cho dữ liệu mới lấy: cấu trúc name: "/weather_log(" + timestamp + ").csv"

```
// Phương thức tạo tên file mới cho dữ liệu thời tiết
public static String generateFileName(String folderPath) {
    // Lấy thời gian hiện tại và định dạng theo "dd-MM-yyyy_HH-mm-ss"
    String timestamp = LocalDateTime.now().format(DateTimeFormatter.ofPattern("dd-MM-yyyy_HH-mm-ss"));

    // Tạo đối tượng File đại diện cho folder lưu trữ
    File folder = new File(folderPath);

    // Nếu folder chưa tồn tại, tạo mới
    if (!folder.exists())
        folder.mkdirs();

    // Trả về đường dẫn đầy đủ của file, kèm timestamp để tránh trùng tên
    return folderPath + "/weather_log(" + timestamp + ").csv";
}
```

7. Tạo và viết data vào file csv mới: input : WeatherData data, String filePath, output: double fileSize

```
public static double writeToCSV(WeatherData data, String filePath) {
    double kilobytes = 0;

    // Sử dụng FileWriter và PrintWriter để ghi dữ liệu vào file
    try (FileWriter fw = new FileWriter(filePath); PrintWriter pw = new PrintWriter(fw)) {

        // Ghi header của file CSV
        pw.println("FullDate,WeekDay,Day,Temperature,UVValue,WindDirection,Humidity,DewPoint,Pressure,Cloud,Visibility,CloudCeiling");

        // Ghi dữ liệu thời tiết theo định dạng CSV
        pw.printf("%s,%s,%s,%s,%s,%s,%s,%s,%s,%s%n", data.time, data.dayDate, data.temperature,
data.uvIndex,data.wind, data.humidity, data.dewPoint, data.pressure, data.cloudCover, data.visibility,data.ceiling);

    } catch (Exception e) {
        // Nếu có lỗi, in stack trace ra console
        e.printStackTrace();
    }

    // Tính dung lượng file sau khi ghi
    File file = new File(filePath);
    if (file.exists()) {
        long bytes = file.length(); // Lấy dung lượng file tính bằng byte
        kilobytes = (bytes / 1024.0); // Chuyển đổi sang KB
    }

    // Trả về dung lượng file (KB)
    return kilobytes;
}
```

