

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙠🙟🕮🙝🙢

A logo of hands holding a book

Description automatically generated

**BÁO CÁO ĐỒ ÁN CUỐI KÌ**

**NGÔN NGỮ LẬP TRÌNH PYTHON**

**Tên đề tài:**

|  |  |
| --- | --- |
| **Nhóm sinh viên thực hiện** | |
| **Họ và tên:** Trần Phúc Toàn | **MSSV:** 23110344 |
| **Họ và tên:** Phan Việt Tuấn | **MSSV:** |
| **Họ và tên:** Bùi Phúc Nhân | **MSSV** |

**Giảng viên:** Phan Thị Thể

**Thành phố Hồ Chí Minh, Tháng 7 năm 2024**

**TÓM TẮT ĐỀ TÀI**

**PHẦN 1: MỞ ĐẦU**

1. **Tính cấp thiết của đề tài.**

Trong bối cảnh công nghệ số phát triển nhanh chóng và nhu cầu tiếp cận thông tin ngày càng gia tăng, việc xây dựng một ứng dụng tích hợp tin tức, dự báo thời tiết và quản lý lịch trình trở thành một giải pháp đầy tiềm năng và thực tiễn. Nhóm tác giả chọn đề tài này xuất phát từ những thách thức và nhu cầu thực tế của người dùng trong môi trường công nghệ hiện đại.

Trước hết, cuộc sống hiện đại đang đòi hỏi những giải pháp công nghệ thông minh, tiện lợi và đa chức năng. Người dùng ngày nay không chỉ mong muốn tiếp cận thông tin nhanh chóng mà còn mong đợi sự tích hợp liền mạch giữa các dịch vụ. Một ứng dụng kết hợp tin tức, thời tiết và lịch trình sẽ là công cụ giải quyết triệt để nhu cầu này. Người dùng có thể dễ dàng tra cứu thông tin thời sự, kiểm tra dự báo thời tiết và quản lý các hoạt động cá nhân tại một địa điểm duy nhất, tiết kiệm tối đa thời gian.

Về mặt công nghệ, đề tài này mở ra cơ hội tuyệt vời để nghiên cứu và ứng dụng các công nghệ tiên tiến. Việc tích hợp các API để lấy dữ liệu thời tiết, sử dụng RSS feed để lấy thông tin về tin tức và phát triển hệ thống quản lý lịch trình đòi hỏi các kỹ năng lập trình chuyên sâu. Thông qua đó nhóm tác giả sẽ được trải nghiệm thực tế trong việc xử lý dữ liệu, thiết kế giao diện người dùn, và giải quyết các thách thức kỹ thuật phức tạp như đồng bộ hóa dữ liệu, cập nhật thông tin theo thời gian thực và đảm bảo trải nghiệm người dùng mượt mà.

Tính ứng dụng của đề tài còn được thể hiện ở khả năng giải quyết những nhu cầu thực tế của người dùng. Trong một xã hội năng động, việc có một công cụ giúp dự đoán và chuẩn bị cho các hoạt động là vô cùng quan trọng. Ứng dụng sẽ hỗ trợ người dùng lập kế hoạch chi tiết dựa trên các thông tin thời tiết chính xác, giúp họ chủ động điều chỉnh lịch trình, lựa chọn trang phục phù hợp, hoặc thậm chí là quyết định những hoạt động ngoài trời một cách thông minh.

Với xu hướng số hóa ngày càng mạnh mẽ và nhu cầu trải nghiệm công nghệ tiện ích, sản phẩm này có thể trở thành một giải pháp hữu hiệu cho nhiều đối tượng người dùng, từ sinh viên, nhân viên văn phòng cho đến những người thích khám phá và du lịch.

Với những lý do trên, việc xây dựng ứng dụng tin tức kết hợp dự báo thời tiết và quản lý lịch trình không chỉ là một đề tài học thuật đơn thuần, mà còn là một dự án có ý nghĩa, mang tính ứng dụng cao và đầy triển vọng trong bối cảnh công nghệ hiện nay.

1. **Mục đích nghiên cứu**

Phát triển ứng dụng đa chức năng, tích hợp các API tin tức và thời tiết nhằm cung cấp thông tin tin tức nhanh chóng, hỗ trợ tra cứu dự báo thời tiết và công cụ quản lí lịch trình cá nhân hiệu quả

Mục tiêu dài hạn của dự án là xây dựng nền tảng ứng dụng có tính mở rộng, tạo tiền đề cho các tính năng nâng cao như trợ lý ảo và gợi ý cá nhân hóa. Sản phẩm không chỉ là đề tài nghiên cứu, mà còn là giải pháp công nghệ thực tiễn, đáp ứng nhu cầu người dùng trong kỷ nguyên số.

1. **Đối tượng và phạm vi nghiên cứu**

* Đối tượng nghiên cứu chính của đồ án là việc xây dựng ứng dụng desktop tích hợp tin tức và dự báo thời tiết. Sử dụng Python với CustomTkinter để thiết kế giao diện, tích hợp RSS Feed và API thời tiết để lấy dữ liệu..
* Nghiên cứu các phương pháp thiết kế giao diện người dùng đơn giản, thân thiện, tối ưu hóa trải nghiệm sử dụng trên desktop.
* Khảo sát cách tích hợp các tính năng quản lý lịch trình, nhắc nhở người dùng về sự kiện quan trọng song song với việc hiển thị thông tin thời tiết và tin tức.
* Đối tượng người dùng: Sinh viên, người đi làm cần công cụ cập nhật tin tức, thời tiết và quản lý công việc hàng ngày.
* Giới hạn: Chỉ hỗ trợ trên Windows và giới hạn tin tức từ các nguồn RSS Feed cố định.

1. **Cấu trúc của báo cáo:**

Bài báo cáo gồm 3 phần chính và được triển khai như sau

* *Phần 1: Lời mở đầu:*

Trình bày lý do chọn đề tài, mục đích nghiên cứu, đối tượng và phạm vi nghiên cứu. Đồng thời, phần mở đầu giới thiệu tổng quan về cấu trúc bài tiểu luận.

* *Phần 2: Nội dung* - Đây là phần trọng tâm, gồm hai chương chính:

Chương 1: Cơ sở lý thuyết – Cung cấp các kiến thức nền tảng liên quan đến đề tài, bao gồm công nghệ tích hợp API, RSS Feed, nguyên tắc thiết kế giao diện người dùng, và các lý thuyết liên quan đến quản lý lịch trình.

Chương 2: Thiết kế và triển khai hệ thống – Trình bày chi tiết quy trình phân tích yêu cầu, thiết kế kiến trúc hệ thống, triển khai các chức năng chính như hiển thị tin tức, dự báo thời tiết, và quản lý lịch trình. Phần này cũng đánh giá kết quả thử nghiệm và phân tích ưu, nhược điểm của ứng dụng.

* *Phần 3: Kết luận và khuyến nghị*

Tóm lược những nội dung chính của bài tiểu luận, đánh giá mức độ đạt được của mục tiêu nghiên cứu, chỉ ra những hạn chế và đề xuất hướng phát triển ứng dụng trong tương lai.

**PHẦN 2: NỘI DUNG**

**CHƯƠNG 1: Cơ sở lí thuyết**

* 1. ***Khái quát sơ lược về API và RSS Feed***
     1. *Khái niệm và cơ chế hoạt động của API*

API (Application Programming Interface) là một tập hợp các quy tắc và giao thức cho phép các ứng dụng hoặc dịch vụ giao tiếp với nhau. Nó đóng vai trò trung gian, giúp một phần mềm hoặc ứng dụng truy cập các chức năng, dữ liệu, hoặc dịch vụ của một phần mềm khác mà không cần biết chi tiết bên trong [1].

API chủ yếu bao gồm Web API, sử dụng HTTP hoặc HTTPS để giao tiếp giữa client và server. Các loại API khác có thể kể đến là Library API (do thư viện phần mềm cung cấp) và Operating System API (giúp ứng dụng truy cập tài nguyên hệ điều hành). API hoạt động qua ba bước chính: Đầu tiên, client gửi yêu cầu tới server, bao gồm URL, phương thức HTTP, tham số đầu vào và thông tin xác thực. Sau đó, server xử lý yêu cầu, kiểm tra dữ liệu và thực hiện hành động tương ứng. Cuối cùng, server gửi phản hồi về client dưới dạng dữ liệu (thường là JSON hoặc XML) [2].

| **Bước** | **Mô tả** |
| --- | --- |
| **1.Request (Yêu cầu)** | Ứng dụng client gửi yêu cầu tới server thông qua API. Yêu cầu có thể bao gồm: URL, phương thức HTTP (GET, POST), tham số đầu vào (như tên thành phố), và thông tin xác thực (API key). |
| **2. Server xử lý yêu cầu** | Server nhận yêu cầu, xác minh dữ liệu, thực hiện hành động cần thiết (như truy xuất dữ liệu từ cơ sở dữ liệu hoặc dịch vụ khác) và chuẩn bị phản hồi. |
| **3. Response (Phản hồi)** | Server gửi lại phản hồi dưới dạng dữ liệu (thường là JSON hoặc XML) cho ứng dụng client. Phản hồi này có thể bao gồm thông tin yêu cầu, ví dụ: dữ liệu thời tiết, tin tức, v.v. |

API thời tiết là công cụ mạnh mẽ cho phép các ứng dụng và hệ thống truy xuất thông tin thời tiết từ các dịch vụ trực tuyến như các đài khí tượng, vệ tinh, và các nguồn dữ liệu khác [3]. Khi người dùng hoặc ứng dụng cần thông tin thời tiết, một yêu cầu HTTP được gửi đến server cung cấp API. Yêu cầu này có thể bao gồm thông tin như tên thành phố, tọa độ địa lý (vĩ độ, kinh độ), hoặc mã bưu điện để xác định vị trí. API yêu cầu người dùng cung cấp một API key – một mã khóa duy nhất để xác thực và giám sát người dùng truy cập dịch vụ.

Sau khi nhận yêu cầu, server API truy vấn cơ sở dữ liệu hoặc dịch vụ đối tác để lấy thông tin thời tiết và trả về dữ liệu dưới dạng JSON hoặc XML, bao gồm nhiệt độ, độ ẩm, gió, tình trạng thời tiết và dự báo. OpenWeather API cung cấp các dữ liệu chi tiết như dự báo hiện tại, 5 ngày, theo giờ, và lịch sử thời tiết [4]. Các ứng dụng có thể yêu cầu thông tin nhanh chóng và tự động, nhưng dữ liệu có thể không chính xác hoàn toàn, đặc biệt ở các khu vực thiếu trạm đo đạc hoặc trong thời tiết cực đoan.

* + 1. *Khái niệm và nguyên tắc hoạt động của RSS Feed*

RSS (Really Simple Syndication) là công nghệ cho phép website tự động cung cấp thông tin qua tệp XML, bao gồm bài viết, bản tin, podcast hoặc video từ các nguồn như trang web tin tức, blog, và dịch vụ trực tuyến. RSS Feed giúp người dùng nhận thông tin mới mà không cần truy cập trực tiếp vào website. là một công nghệ cho phép các website [5].

RSS Feed là tệp XML gồm hai phần chính: Channel và Item. Channel chứa thông tin về kênh như tên, mô tả và đường dẫn website, trong khi Item chứa chi tiết bài viết như tiêu đề, mô tả, đường dẫn và ngày xuất bản.*Nguyên tắc hoạt động của RSS Feed:*

RSS Feed hoạt động theo một chu trình đơn giản nhưng hiệu quả. Đầu tiên, các website hoặc dịch vụ trực tuyến tạo ra một file RSS Feed, thông qua đó cung cấp thông tin mới nhất từ website của họ. File RSS Feed này chứa các thông tin như tiêu đề bài viết, mô tả ngắn gọn và đường dẫn đến bài viết đầy đủ.[5].

Khi người dùng đăng ký kênh RSS, trình đọc RSS tự động kiểm tra các Feed để tìm bài viết mới. Khi có bài viết mới, Feed được cập nhật và trình đọc hiển thị thông tin mới, giúp người dùng nhận thông tin mà không cần truy cập từng website.

* 1. ***Tổng quan về các công cụ và thư viện Python hỗ trợ phát triển ứng dụng***

*1.2.1. Tổng quan về Python và khả năng ứng dụng trong phát triển phần mềm*

Python là ngôn ngữ lập trình bậc cao, nổi bật với cú pháp đơn giản, dễ học và ứng dụng rộng rãi. Ra mắt vào năm 1991, Python nhanh chóng trở thành lựa chọn phổ biến nhờ tính dễ tiếp cận và mã nguồn mở, giúp lập trình viên viết mã dễ dàng và giảm thiểu lỗi [6].

Python trở thành công cụ phổ biến nhờ vào hệ sinh thái thư viện phong phú và khả năng tích hợp công nghệ tiên tiến. Các thư viện như Tkinter, CustomTkinter và API như requests giúp xây dựng ứng dụng desktop và web dễ dàng, từ việc truy xuất dữ liệu qua API (như thời tiết) đến việc tạo giao diện người dùng trực quan và phát triển tính năng phức tạp với ít mã lệnh.

Trong phát triển ứng dụng tin tức, dự báo thời tiết và quản lý lịch trình, Python là lựa chọn lý tưởng. Thư viện requests giúp lấy dữ liệu từ các nguồn tin tức (RSS) và dự báo thời tiết qua API. CustomTkinter cho phép xây dựng giao diện đẹp, dễ sử dụng cho ứng dụng desktop. Python cũng hỗ trợ các thư viện như schedule và datetime để quản lý lịch trình và gửi nhắc nhở người dùng về các sự kiện quan trọng. *1.2.2. Thư viện Custom Tkinter*

CustomTkinter là thư viện hỗ trợ xây dựng giao diện người dùng (GUI) cho ứng dụng Python. Với khả năng tạo giao diện đơn giản nhưng hiện đại, CustomTkinter là sự lựa chọn lý tưởng cho việc phát triển ứng dụng tích hợp tin tức, dự báo thời tiết và quản lý lịch trình. Thư viện này giúp tạo các thành phần giao diện như bảng tin tức, dự báo thời tiết, và các chức năng tương tác, đồng thời tối ưu hóa trải nghiệm người dùng.

CustomTkinter hỗ trợ các widget như thanh cuộn (scrollbars), khung chứa nội dung (frame), và nút bấm (button), giúp dễ dàng tạo giao diện mượt mà và dễ sử dụng [7]. Các widget này cho phép người dùng duyệt qua tin tức hoặc lịch trình một cách thuận tiện mà không gặp phải vấn đề về không gian màn hình.CustomTkinter cũng hỗ trợ đồng bộ hóa dữ liệu thời gian thực từ các API thời tiết hoặc RSS feed, nâng cao hiệu quả sử dụng và trải nghiệm người dùng.  
*1.2.3. Các công cụ hỗ trợ kết nối API, RSS Feed và xử lý dữ liệu*

Kết nối API: Trong Python, thư viện Requests là công cụ phổ biến để gửi yêu cầu HTTP đến các API như thời tiết hay tin tức. Điển hình, thư viện Requests còn giúp lấy dữ liệu thời tiết từ API OpenWeatherMap.

Kết nối và xử lý RSS Feed: Để lấy dữ liệu từ RSS Feed, thư viện Feedparser giúp phân tích và trích xuất thông tin từ các RSS Feed một cách dễ dàng. Feedparser có thể dùng để hiển thị tiêu đề và liên kết bài viết từ một nguồn tin RSS.

Xử lý và phân tích dữ liệu: BeautifulSoup là công cụ lý tưởng khi cần trích xuất dữ liệu từ các trang web không có API.

Xử lý dữ liệu JSON: Dữ liệu từ API thường ở định dạng JSON, và Python cung cấp thư viện JSON tích hợp để dễ dàng xử lý. Nó giúp chuyển đổi dữ liệu JSON thành dictionary để dễ thao tác

* 1. ***Nguyên tắc thiết kế giao diện người dùng***

Khi xây dựng ứng dụng, thiết kế giao diện người dùng (UI) đóng vai trò quan trọng trong việc tạo ra trải nghiệm mượt mà và dễ sử dụng. Một giao diện tốt cần phải đảm bảo tính đơn giản và rõ ràng, giúp người dùng dễ dàng nhận biết và sử dụng các chức năng mà không bị rối mắt [8].

Tính nhất quán trong thiết kế giúp người dùng dễ dàng làm quen với ứng dụng, từ việc sử dụng các biểu tượng, màu sắc đến kiểu chữ. Các yếu tố trong giao diện cần phản hồi ngay lập tức với người dùng, thể hiện qua phản hồi nhanh và rõ ràng mỗi khi người dùng thực hiện thao tác.

Một giao diện tương tác dễ dàng và trực quan sẽ giúp người dùng hoàn thành nhiệm vụ nhanh chóng. Thiết kế đáp ứng đảm bảo ứng dụng hiển thị đúng trên nhiều thiết bị khác nhau, từ máy tính đến điện thoại, mà không làm mất đi tính thẩm mỹ.

Ngoài ra, việc sử dụng màu sắc hợp lý giúp làm nổi bật các yếu tố quan trọng và tạo cảm giác dễ chịu. Giao diện cần tối ưu hóa cho người dùng, tức là cần nghiên cứu và hiểu được nhu cầu người dùng để đưa ra thiết kế phù hợp.

Để hiển thị thông tin rõ ràng, các thông báo và dữ liệu quan trọng cần được bố trí hợp lý, dễ đọc. Cuối cùng, khả năng truy cập (accessibility) phải được đảm bảo, giúp ứng dụng dễ dàng sử dụng cho mọi đối tượng người dùng, bao gồm những người có khuyết tật.

**CHƯƠNG 2: Thiết kế - triển khai hệ thống. Đánh giá kết quả đạt được**

***2.1. Phân tích yêu cầu hệ thống và kịch bản hoạt động****2.1.1. Yêu cầu chức năng của ứng dụng*

| **Yêu Cầu Chức Năng** | **Mô Tả** | **Chức Năng Cụ Thể** |
| --- | --- | --- |
| **Kết nối và Hiển thị tin tức** | Ứng dụng cần kết nối với các nguồn tin tức qua RSS Feed hoặc API và hiển thị các bài viết mới nhất. | - Kết nối và lấy dữ liệu từ RSS Feed hoặc API. - Cập nhật tin tức theo thời gian thực. - Hiển thị tiêu đề, mô tả.  -Mở bài viết thông qua trình duyệt mặc định của máy tính |
| **Dự báo Thời tiết** | Cung cấp thông tin thời tiết hiện tại và dự báo cho các ngày tiếp theo. | - Kết nối với API thời tiết. - Hiển thị nhiệt độ, độ ẩm, điều kiện thời tiết. - Cập nhật thông tin thời tiết thời gian thực. |
| **Quản lý Lịch trình Cá nhân** | Quản lý các sự kiện, công việc và cuộc hẹn trong ngày. | - Tạo và chỉnh sửa sự kiện. - Hiển thị lịch trình theo tháng, tuần, ngày. - Nhắc nhở các sự kiện quan trọng. |
| **Giao diện Người Dùng (UI)** | Giao diện cần đơn giản, trực quan, dễ sử dụng. | - Hiển thị tin tức, thời tiết, và lịch trình một cách dễ đọc. - Cung cấp thanh cuộn và các nút bấm điều hướng. |
| **Tính Năng Tương Tác và Cập Nhật Thời Gian Thực** | Cập nhật dữ liệu tin tức, thời tiết, và lịch trình mà không cần tải lại ứng dụng. | - Cập nhật dữ liệu trong thời gian thực. - Sử dụng kỹ thuật bất đồng bộ để tối ưu hóa hiệu suất. |
| **Quản lý Dữ liệu Cá nhân và Cài đặt** | Lưu trữ và quản lý thông tin người dùng và cài đặt ứng dụng. | - Lưu trữ sự kiện và thông tin cá nhân. - Cho phép chỉnh sửa và xóa dữ liệu. - Cung cấp cài đặt tùy chỉnh (giao diện, thông báo). |

*2.1.2. Yêu cầu phi chức năng*

| **Yêu Cầu Phi Chức Năng** | **Mô Tả** | **Chi Tiết** |
| --- | --- | --- |
| **Hiệu suất (Performance)** | Ứng dụng cần hoạt động mượt mà và nhanh chóng, ngay cả khi người dùng yêu cầu nhiều thao tác hoặc tải dữ liệu từ nhiều nguồn. | - Thời gian phản hồi nhanh khi người dùng tương tác với giao diện. - Cập nhật dữ liệu tin tức và thời tiết trong thời gian thực mà không gây gián đoạn. |
| **Tính ổn định (Stability)** | Ứng dụng cần duy trì sự ổn định và hoạt động liên tục, không bị crash hoặc lỗi trong quá trình sử dụng. | - Đảm bảo không có lỗi hay sự cố khi tải dữ liệu từ API. - Quản lý lỗi hiệu quả và thông báo lỗi cho người dùng khi có sự cố xảy ra. |
| **Tính khả dụng (Availability)** | Ứng dụng cần sẵn sàng hoạt động liên tục và có thể truy cập được mọi lúc. | - Dữ liệu từ các API như thời tiết và tin tức cần luôn sẵn sàng và cập nhật thường xuyên. - Ứng dụng cần có tính khả dụng cao, đặc biệt trong các giờ cao điểm. |
| **Khả năng sử dụng (Usability)** | Giao diện người dùng phải trực quan, dễ sử dụng và dễ dàng tiếp cận các tính năng của ứng dụng. | - Giao diện đơn giản, không rối mắt, người dùng dễ dàng điều hướng. - Đảm bảo khả năng sử dụng cho nhiều nhóm người dùng, bao gồm cả những người không chuyên về công nghệ. |

*2.1.3. Đối tượng sử dụng và kịch bản hoạt động*

Kịch bản 1: Xem tin tức và thời tiết

* Người dùng mở ứng dụng và được hiển thị thông tin thời tiết hiện tại cùng với các tin tức nổi bật.
* Người dùng có thể cuộn xuống để xem dự báo thời tiết trong những ngày tới hoặc theo dõi các chuyên mục tin tức khác nhau.
* Họ cũng có thể chọn các nguồn tin khác nhau để xem tin tức từ các website hoặc báo chí yêu thích.

Kịch bản 2: Quản lý lịch trình cá nhân

* Người dùng có thể vào phần lịch trình của ứng dụng và thêm các sự kiện hoặc nhiệm vụ cần thực hiện trong ngày.
* Các thông báo nhắc nhở sẽ được thiết lập để người dùng không bỏ lỡ các sự kiện quan trọng như cuộc họp, lớp học, hoặc các cuộc hẹn cá nhân.
* Thông tin thời tiết sẽ được hiển thị kèm theo mỗi sự kiện để người dùng có thể điều chỉnh kế hoạch tùy thuộc vào điều kiện thời tiết.

***2.2. Thiết kế hệ thống***

*2.2.1. Sơ đồ tổng quan kiến trúc hệ thống*

*MainProgram.py: Quản lý Cấu hình và Giao diện Chính*

* Class ApplicationConfig:
  + Chức năng: Quản lý cấu hình ứng dụng, tải cấu hình từ tệp JSON hoặc tạo mới nếu không tồn tại. Nếu cấu hình bị lỗi, chương trình sẽ sử dụng cấu hình mặc định.
  + Mối quan hệ: Được sử dụng để cấu hình các tham số ban đầu cho ứng dụng như giao diện người dùng, thông tin về các panel, và các thông số hệ thống.
* Class FunctionExecute:
  + Chức năng: Xây dựng giao diện người dùng bằng CustomTkinter, bao gồm khởi tạo cửa sổ chính và các bảng điều khiển (panels) cho các chức năng tin tức, thời tiết, lịch trình và cài đặt. Quản lý điều hướng giữa các bảng điều khiển qua sidebar.
  + Mối quan hệ: Quản lý các tương tác và giao diện người dùng chính của ứng dụng, kết nối với các panel khác như tin tức, thời tiết và lịch trình.

*NewsPanel.py: Quản lý Tin tức và Hình ảnh*

* Class ImageCacheManager:
  + Chức năng: Quản lý tải và lưu trữ hình ảnh từ các nguồn tin tức, bao gồm tải hình ảnh bất đồng bộ, bộ đệm hình ảnh trong bộ nhớ và đĩa, và xóa các hình ảnh ít sử dụng khi bộ đệm đầy.
  + Mối quan hệ: Hoạt động cùng với NewsPanel để tối ưu hóa việc tải hình ảnh cho các bài viết tin tức.
* Class NewsPanel:
  + Chức năng: Hiển thị tin tức từ các nguồn RSS Feed, bao gồm tải bài viết, trích xuất thông tin tiêu đề, mô tả và hình ảnh (nếu có). Hình ảnh được tải và lưu vào bộ nhớ cache bất đồng bộ, giúp giao diện không bị gián đoạn. Các bài viết có thể nhấp vào tiêu đề để mở liên kết bài viết.
  + Mối quan hệ: Hiển thị thông tin tin tức, được kết nối với ImageCacheManager để tải và lưu trữ hình ảnh. Tin tức và hình ảnh sẽ được hiển thị trong panel của giao diện chính.

*SchedulePanel.py: Quản lý Lịch và Sự kiện*

* Class SchedulePanel:
  + Chức năng: Quản lý lịch và sự kiện của người dùng, cho phép tạo, chỉnh sửa, xóa và xem các sự kiện trong giao diện người dùng. Người dùng có thể chọn ngày, thêm sự kiện với thời gian, tiêu đề và nội dung chi tiết. Cung cấp tính năng xem lịch theo tháng và các sự kiện sắp tới trong vòng 7 ngày. Dữ liệu sự kiện được lưu trữ trong tệp JSON.
  + Mối quan hệ: Chịu trách nhiệm hiển thị và quản lý các sự kiện trên giao diện người dùng. Dữ liệu sự kiện sẽ được lưu trữ và truy xuất khi cần thiết.

*WeatherForecastPanel.py: Dự báo Thời tiết*

* Class WeatherPanel:
  + Chức năng: Hiển thị thông tin thời tiết hiện tại và dự báo thời tiết cho các thành phố. Sử dụng API OpenWeatherMap để lấy dữ liệu thời tiết, bao gồm nhiệt độ, độ ẩm, tốc độ gió, và các biểu tượng thời tiết. Người dùng có thể nhập tên thành phố để xem thời tiết và dự báo trong 5 ngày tiếp theo.
  + Mối quan hệ: Liên kết với API OpenWeatherMap để lấy thông tin thời tiết và hiển thị chúng trên giao diện người dùng.

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

*2.2.2. Thiết kế giao diện người dùng (UI) với CustomTkinter*

*2.2.3. Tích hợp dữ liệu tin tức qua RSS Feed*

*2.2.4. Tích hợp API thời tiết thời gian thực*

*2.2.5. Thiết kế chức năng quản lý lịch trình*

***2.3. Quy trình triển khai***

*2.3.1. Xây dựng giao diện chính của ứng dụng*

*2.3.2. Kết nối và xử lý dữ liệu tin tức từ RSS Feed*

*2.3.3. Triển khai chức năng dự báo thời tiết với API*

*2.3.4. Lập trình và tối ưu hệ thống quản lý lịch trình*

***2.4. Thử nghiệm và đánh giá kết quả***

*2.4.1. Kết quả chạy thử nghiệm các chức năng chính*

*2.4.2. Phân tích ưu điểm và hạn chế của ứng dụng*

*2.4.3. Đánh giá mức độ đáp ứng yêu cầu ban đầu*

***2.5. Chạy demo và kết quả kiểm thử ứng dụng***

**KẾT LUẬN VÀ KHUYẾN NGHỊ**

|  |  |  |
| --- | --- | --- |
| **Nội dung thực hiện** | **Sinh viên thực hiện** | **Nhóm tự đánh giá mức độ hoàn thành**  **(Tốt/Khá/Kém)** |
|  |  |  |