**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**TÀI LIỆU THIẾT KẾ**

*Thực hiện: Nhóm 14*

*21120505 – Bùi Thị Thanh Ngân*

*21120531 – Nguyễn Văn Phúc*

*21120540 – Trần Tôn Bửu Quang*

*21120600 – Nguyễn Thị Như Ý*

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Bộ môn Công nghệ phần mềm

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

**MỤC LỤC**

[1 Bảng đánh giá thành viên 2](#_Toc152795323)

[2 Mô hình quan niệm 3](#_Toc152795324)

[3 Thiết kế kiến trúc 4](#_Toc152795325)

[4 Thiết kế dữ liệu 5](#_Toc152795326)

[4.1 Sơ đồ dữ liệu 5](#_Toc152795327)

[4.2 Đặc tả dữ liệu 5](#_Toc152795328)

**TÀI LIỆU THIẾT KẾ**

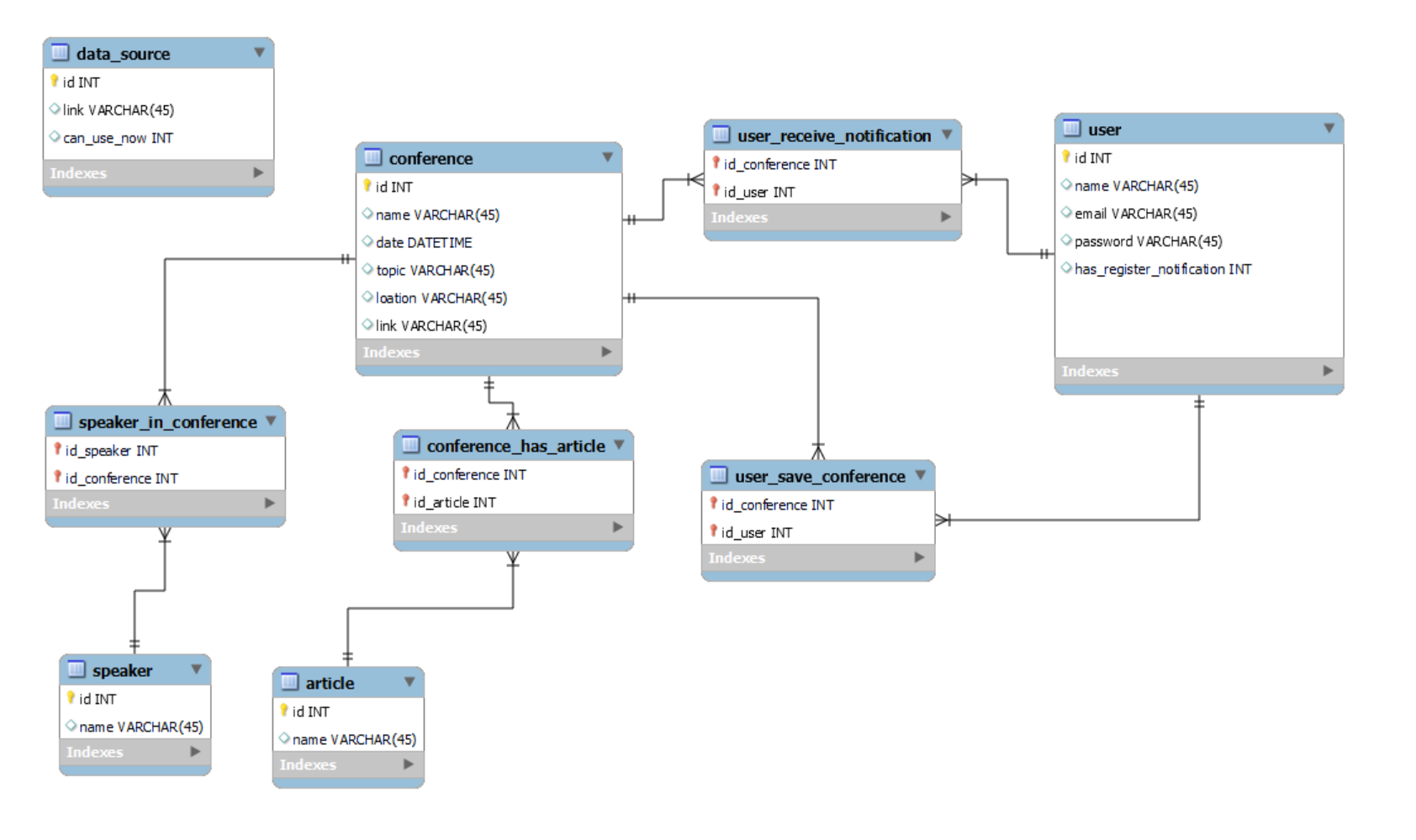
Tài liệu tập trung vào các chủ đề:

* Tạo ra tài liệu thiết kế phần mềm.
* Hoàn chỉnh tài liệu thiết kế phần mềm với các nội dung:
  + Mô hình quan niệm
  + Thiết kế kiến trúc
  + Thiết kế dữ liệu
  + Thiết kế giao diện người dùng
* Đọc hiểu tài liệu thiết kế phần mềm.

# Bảng đánh giá thành viên

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ Tên** | **% đóng góp (tối đa 100%)** | **Chữ ký** |
| 21120505 | Bùi Thị Thanh Ngân | 25% |  |
| 21120531 | Nguyễn Văn Phúc | 25% |  |
| 21120540 | Trần Tôn Bửu Quang | 25% |  |
| 21120600 | Nguyễn Thị Như Ý | 25% |  |

# Mô hình quan niệm

*[Trình bày sơ đồ thể hiện các thực thể ngữ nghĩa trong phần mềm, có thể sử dụng mô hình EER tạo bởi* ***MySQL Workbench*** *hoặc* ***Power Designer*** *để thực hiện phần này]*

# Thiết kế kiến trúc

*[Trình bày hình vẽ cây phân rã hệ thống, cho biết hệ thống có các thành phần như thế nào (phân rã theo chiều dọc)]*

*[Trình bày các điểm đặc biệt trong kiến trúc, ví dụ như áp dụng mẫu thiết kế (Design Pattern), sử dụng kiến trúc Client-Server/Tier/MVC…, hỗ trợ cơ chế plug-in…]*

Cây phân rã hệ thống theo chiều dọc:

Đặc điểm của kiến trúc:

**Kiến trúc client-server:**

* Áp dụng vào việc xây dựng một web client chạy trên máy tính của người dùng, đây sẽ là nơi triển khai giao diện của phần mềm này.
* Người dùng sẽ thực hiện các thao tác, các request tới máy chủ để lấy thông tin mình mong muốn.
* Server là nơi sẽ thu nhận các yêu cầu, thông tin từ người dùng để đưa ra các phản hồi, các kết quả phù hợp. Đồng thời server cũng là nơi sẽ xử lý thông tin để cập nhật các tin tức mới nhất, là nơi sẽ tiến hành phân tích các URL của trang web để đưa đến nguồn tin vắn tắt nhất thể hiện tại màn hình giao diện của client

**Model:**

1. Model Data:

* Mô tả: Chứa dữ liệu của hệ thống, bao gồm thông tin về các hội nghị, diễn giả, bài báo, và thông tin khác liên quan.
* Triển khai: Kết nối với cơ sở dữ liệu để lưu trữ và truy xuất dữ liệu.

1. Business Logic:

* Mô tả: Thực hiện xử lý dữ liệu, lọc thông tin từ crawler, và cung cấp dữ liệu cho Controller. (Một phần của backend, xử lý logic của ứng dụng.)

**View:**

1. User Interface (UI):

* Mô tả:

+ Định dạng và xử lý dữ liệu lấy được trong response trả về từ server

+ Hiển thị dữ liệu đã được định dạng, và tương tác với người dùng.

* **Controller:**

1. Request Handling:

* Mô tả: Nhận và xử lý yêu cầu từ phía người dùng (UI), xác thực và điều hướng họ đến các trang yêu cầu. (Phần của backend, xử lý logic điều khiển.)

1. Data Flow Control:

* Mô tả: Quản lý luồng dữ liệu giữa Model và View, đảm bảo sự tương tác hiệu quả.

**Ưu điểm áp dụng MVC trong phần mềm Crawler:**

1. Tách Biệt Logic:

* Ưu điểm: Frontend (View) và Backend (Model và Controller) được tách biệt, giúp quản lý code dễ dàng và tái sử dụng.

1. Dễ Bảo Trì và Mở Rộng:

* Ưu điểm: Mỗi thành phần có trách nhiệm cụ thể, giúp việc bảo trì và mở rộng hệ thống trở nên dễ dàng.

1. Tính Tương Tác:

* Ưu điểm: Controller quản lý sự tương tác giữa View và Model, giúp duy trì sự đồng bộ và hiệu quả.

1. Thử Nghiệm Dễ Dàng:

* Ưu điểm: Tính tách biệt của Model, View, và Controller làm cho quá trình thử nghiệm trở nên dễ dàng hơn.

1. Phân Phối Công Việc:

* Ưu điểm: Mô hình giúp phân phối công việc giữa các thành viên của nhóm phát triển.

Việc áp dụng mô hình MVC giúp quản lý và phát triển dự án một cách hiệu quả hơn, đồng thời tối ưu hóa sự tương tác giữa các thành phần

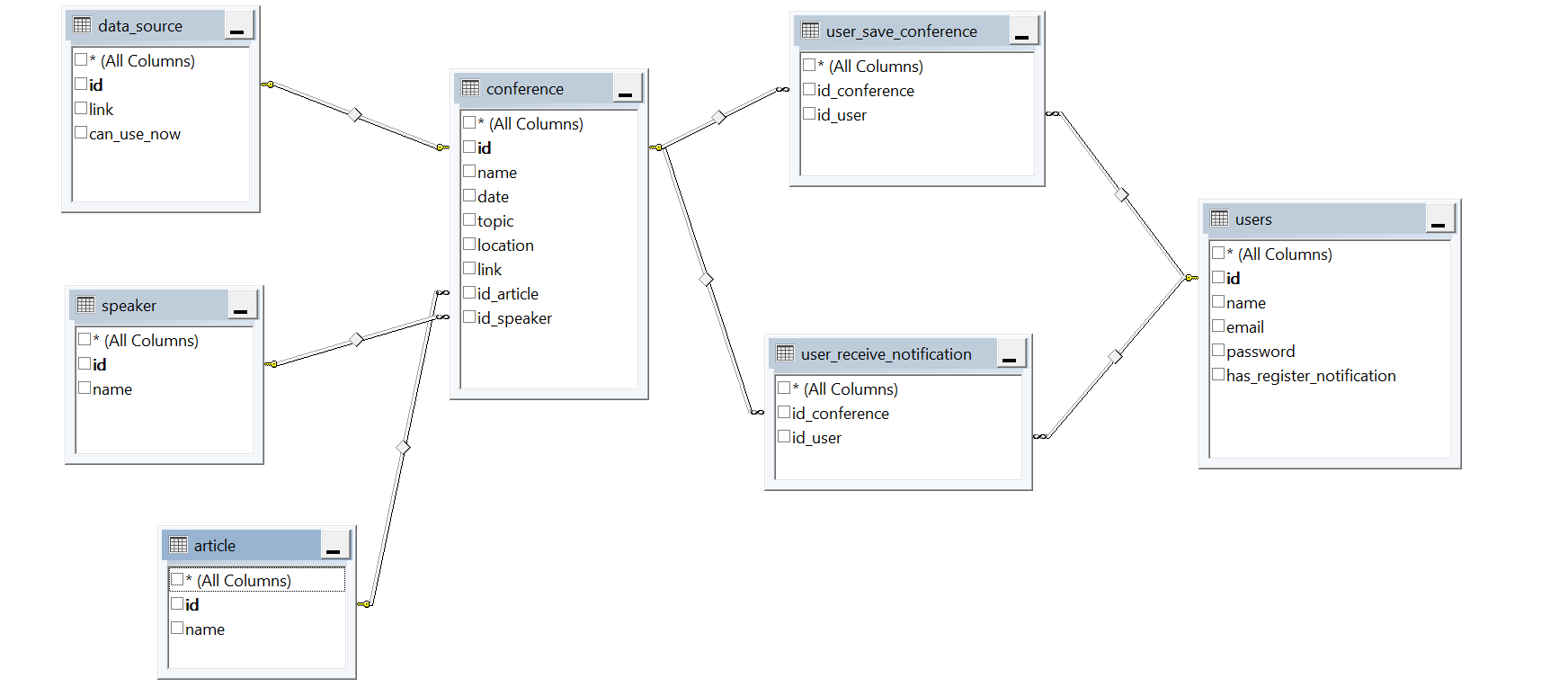
**Áp dụng Strategy Pattern**

* Cách đọc, trích xuất thông tin ở mỗi một trang web là hoàn toàn khác nhau(một số trang web chống crawler, một số trang web bị giới hạn truy cập). Việc yêu cầu sử dụng thuật toán linh hoạt cho nhiều trường hợp đặc biệt như vậy rất thích hợp để sử dụng Strategy. Việc sử dụng Strategy cho phép tách biệt các thuật toán xử lý logic một cách riêng biệt. Nếu gặp các trường hợp cần xử lý bằng thuật toán riêng biệt thì vẫn có thể thêm chiến lược xử lý mới mà không gây ảnh hưởng tới các phần trước đó
* Dễ mở rộng và bảo trì: Strategy cho phép thêm các chiến lược mới mà không gây ảnh hưởng gì tới các thành phần khác. Điều này giúp giảm thiểu tối đa sự phụ thuộc giữa các thành phần và tăng tính mô đun cho hệ thống

# Thiết kế dữ liệu

## Sơ đồ dữ liệu

*[Vẽ sơ đồ dữ liệu của hệ thống, trong đó xác định các thành phần dữ liệu cần lưu trữ, thể hiện mối quan hệ giữa chúng]*



## Đặc tả dữ liệu

*[Nếu dùng CSDL, mô tả các bảng dữ liệu, thông tin của từng cột dữ liệu bao gồm tên thuộc tính, ràng buộc kiểu dữ liệu và giá trị, ràng buộc khóa..., diễn giải thuộc tính]*

*[Nếu dùng XML hoặc file có cấu trúc tự định nghĩa để lưu trữ dữ liệu, cần mô tả cụ thể cấu trúc của file, các thông tin thuộc tính, ràng buộc kiểu dữ liệu và giá trị. Nên kèm theo ví dụ về nội dung cho tập tin lưu trữ thông tin]*

1. Bảng conference

* Mô tả: Lưu thông tin về các hội nghị.

|  |  |  |
| --- | --- | --- |
| **Field** | **Kiểu dữ liệu** | **Mô tả** |
| id | INT | Primary key, auto generate |
| name | Nvarchar(100) | Tên hội nghị (not null) |
| date\_start | Date | Ngày bắt đầu diễn ra hội nghị |
| date\_end | Date | Ngày kết thúc hội nghị |
| topic | Nvarchar(100) | Chủ đề chính của hội nghị |
| location | Nvarchar(100) | Địa điểm diễn ra hội nghị |
| link | Varchar(45) | Đường dẫn đến bài viết chứa thông tin chi tiết của hội nghị |
| id\_article | INT | Khóa ngoại đến các bài báo được thông qua trong hội nghị |
| id\_speaker | INT | Khóa ngoại đến các diễn giả tham gia trong hội nghị |

1. Bảng user

* Mô tả: Lưu thông tin đăng ký và theo dõi của người dùng.

|  |  |  |
| --- | --- | --- |
| **Field** | **Kiểu dữ liệu** | **Mô tả** |
| id | INT | Primary key, auto generate |
| name | varchar(100) | Tên người dùng |
| mail | varchar(100) | Địa chỉ mail của người dùng  (not null, unique) |
| password | varchar(255) | Mật khẩu đã hash do người dùng đăng ký |
| has\_register\_notification | INT | Giá trị cho biết người dùng có đăng ký nhận thông báo qua mail không (0 | 1) |

1. Bảng article

* Mô tả: lưu thông tin về các bài báo được thông qua trong hội nghị

|  |  |  |
| --- | --- | --- |
| **Field** | **Kiểu dữ liệu** | **Mô tả** |
| id | INT | Khóa chính, auto generate |
| name | Nvarchar(45) | Tên bài báo |
| link | Varchar(255) | Link dẫn đến bài báo |

1. Bảng speaker

* Mô tả: Lưu thông tin về các diễn giả.

|  |  |  |
| --- | --- | --- |
| **Field** | **Kiểu dữ liệu** | **Mô tả** |
| id | INT | Primary key, auto generate |
| name | Nvarchar(100) | Tên diễn giả tham gia hội nghị |
| bio | TEXT | Thông tin về diễn giả (công việc, thành tựu..) |

1. Bảng data\_source

* Mô tả: Lưu thông tin về các urls để crawler

|  |  |  |
| --- | --- | --- |
| **Field** | **Kiểu dữ liệu** | **Mô tả** |
| id | INT | Primary key, auto generate |
| link | varchar(100) | Đường link của trang web đang chờ xử lý |
| status | INT | Tình trạng của đường link(lỗi, đang được cập nhật, sẵn sàng...) |

1. Bảng user\_save\_conference

* Mô tả: Lưu thông tin về các bài hội nghị mà người dùng đã đăng ký theo dõi

|  |  |  |
| --- | --- | --- |
| **Field** | **Kiểu dữ liệu** | **Mô tả** |
| Id\_conference | INT | Khóa chính, khóa ngoại tới hội nghị mà người dùng muốn theo dõi, nhận thông báo |
| id\_user | INT | Khóa chính, khóa ngoại tới user đăng ký theo dõi thông tin về hội nghị |