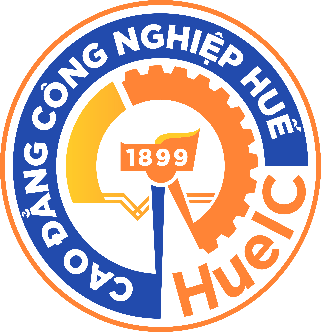
BỘ CÔNG THƯƠNG

TRƯỜNG CĐ CÔNG NGHIỆP HUẾ

**KHOA CÔNG NGHỆ THÔNG TIN – TRUYỀN THÔNG**

**🙞 🕮 🙜**



BÁO CÁO ĐỒ ÁN

**Đề tài : MẠNG VIỆC LÀM**

Sinh viên thực hiện: **Nguyễn Như Phú**

Lớp : **22CDTH41**

Khóa: **2022-22CD**

Giáo viên hướng dẫn: **Nguyễn Hoàng Tùng**

**Huế , tháng 6 năm 2023**

CHƯƠNG 1: TỔNG QUAN VỀ INTERNET VÀ THỰC TRẠNG VẤN ĐỀ

* 1. **Giới thiệu tổng quan và lý do chọn đề tài :**

Internet là một mạng máy tính nối hàng triệu máy tính với nhau trên phạm vi toàn thế giới. Internet có lịch sử rất ngắn, nó có nguồn gốc từ một dự án của Bộ Quốc Phòng Mỹ có tên là ARPANET vào năm 1969, dự án nhằm thực hiện xây dựng một mạng nối các trung tâm nghiên cứu khoa học và quân sự với nhau. Đến năm 1970 đã có thêm hai mạng: Store-and-forwarrd và ALOHAnet, đến năm 1972 hai mạng này đã được kết nối với ARPANET. Cũng trong năm 1972 Ray Tomlinson phát minh ra chương trình thư tín điện tử E-mail. Chương trình này đã nhanh chóng được ứng dụng rộng rãi để gửi các thông điệp trên mạng phân tán.

* 1. **Mục tiêu của đề tài**
* Về tuyển dụng trực tuyến là sự tích hợp của công nghệ Internet nhằm nâng cao hiệu suất và hiệu quả của quy trình tuyển dụng, hầu hết các công ty đã bắt đầu nhận ra được lợi ích của loại hình dịch vụ này và đang áp dụng một cách rộng rãi.
* Nền tảng tập trung: thu thập dữ liệu ứng viên theo một định dạng chuẩn và tổng hợp các dữ liệu từ nhiều nguồn khác nhau.
* Giảm thiểu công việc hành chính, quản lý, theo dõi và lưu trữ ứng viên theo những tiêu chí phù hợp với yêu cầu tuyển dụng.
* Tiết kiệm chi phí và thời gian, nâng cao hiệu suất và kết quả làm việc.
  1. **Mô tả bài toán**

Với sự phát triển mạng như hiện nay thì có rất nhiều giao thức chuẩn ra đời nhằm đáp ứng nhu cầu phát triển. Các chuẩn giao thức được sử dụng rộng rãi nhất hiện nay như giao thức TCP/IP, giao thức SNA của IBM, OSIISDN, X.25 hoặc giao thức LAN to LAN net BIOS. Giao thức được sử dụng rộng rãi nhất hiện nay trên mạng là TCP/IP. Giao thức này cho phép dữ liệu được gửi dưới dạng các “gói” (packet) thông tin nhỏ. Nó chứa hai thành phần: Internet Protocol (IP) và Transmission Control Protocol (TCP).

* 1. **Yêu cầu về chức năng**
* Thư điện tử (E-mail): dịch vụ E-mail có thể dùng để trao đổi thông tin giữa các cá nhân với nhau, các cá nhân với tổ chức và giữa các tổ chức với nhau.
* WWW (world wide web): đây là khái niệm mà người dùng Internet quan tâm nhiều nhất hiện nay.
* Dịch vụ truyền file (FTP – File Transfer Protocol): là dịch vụ dùng để trao đổi các tệp tin từ máy chủ xuống các máy cá nhân và ngược lại.
* Gropher: dịch vụ này hoạt động như menu đủ loại. Thông tin hệ thống menu phân cấp giúp người sử dụng từng bước xác định được những thông tin cần thiết để đi tới vị trí cần đến.
* Telnet: dịch vụ này cho phép truy cập tới Server được xác định rõ như một TelnetSite tìm kiếm Server.
  1. **Yêu cầu về phi chức năng**

Về phi chức ăng nó sẽ khác hơn với chức năng ở tính chất và và ràng buộc của phần mềm hay hệ thống

Vì vậy , trong thực tế yêu cầu về chức năng sễ được đánh giá là có phần quan trọng hơn .Nếu không thỏa mãn được các yêu cầu này thì phần mềm hoặc hệ thống sẽ không thể đưa vào sử dụng .

Chương 2: Phân tích hệ thống

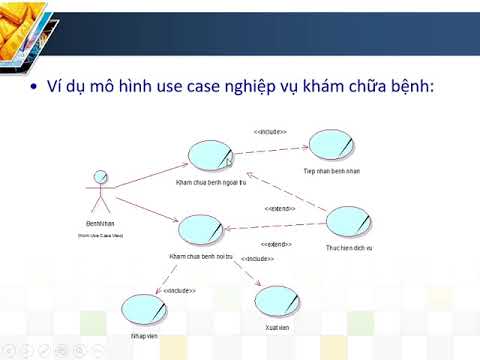
* 1. **Mô hình nghiệp vụ**

Mô tả các chức năng nghiệp vụ của một tổ chức (hay một miền đựoc nghiên cứu của tổ chức) và những mối quan hệ bên trong gi ữa các chức năng đó cũng như các mối quan hệ của chúng với môi trường bên ngoài. Mô hình nghiệp vụ được thể hiện bằng một số dạng khác nhau. Mỗi một dạng mô tả một khía cạnh của hoạt động nghiệp vụ. Tất cả các dạng đó cho ta m ột bức tranh toàn cảnh về hoạt động nghiệp vụ.

* 1. **Mô Hình Use Case**
* Bước 1: Trích phần Use Case của chức năng tương ứng từ biểu đồ Use Case tổng quan.
* Bước 2: Phân rã Use Case chính thành các Use Case con: mỗi giao diện (hoặc một số giao diện) tương tác với người dùng có thể đề xuất thành một Use Case con.
* Bước 3: Xác định quan hệ của Use Case con với Use Case chính: generalization, include, hay extend.
* Bước 4: Gộp các Use Case con tương tự nhau bằng cách dùng các Use Case trừu tượng tổng quát hơn.

Lưu ý tránh sai sót trong biểu đồ Use Case:

* Tên use case phải là động từ chỉ hành động của actor. Không nên là động từ chỉ hành động của hệ thống. Cũng không nên là danh từ, tính từ…
* Mỗi use case phải có tương tác với ít nhất một actor, có thể là trực tiếp hoặc gián tiếp: phải tồn tại ít nhất một đường đi từ một actor nào đó đến use case theo hướng: quan hệ include thì theo chiều mũi tên, quan hệ extend thì ngược chiều mũi tên, quan hệ kế thừa thì gộp lại.



**2.3 Các tác nhân**

Là một đối tượng bên ngoài hệ thống giao tiếp hệ thống, thông qua :

Tương tác ,trao đổi thông tin với hệ thống hoặc sử dụng các phương thức của hệ thống .

Cung cấp đầu vào hoặc nhận đầu ra từ hệ thống

Không điều khiển hoạt động của hệ thống .

Tên tác nhân là theo danh từ ,vai trò của tác nhân

Phân loại các tác nhân :

Tác nhân chính là những người suer dụng những chức năng chính của hệ thống.

Tác nhân phụ là những người làm những công việc quản lý ,bảo dưỡng hệ thống

Các thiết bị ngoài :Là thiết bị được hệ thống điều khiển .

Các hệ thống khác :là các hệ thống không thuộc hệ thống đang xét nhưng tương tác với nó

Tác nhân có thể coi là vai trò tham gia vào hệ thống.

Một đối tượng có t hể tham gia vào một hoặc nhiều vai trò

Để tìm các tác nhân có thể trả lời câu hỏi sau:

Ai hay hệ thống nào sẽ là người dùng những chức năng chính của hệ thống ?

Ai cần sự hỗ trợ của hệ thống để thực hiện những công việc hàng ngày của họ?

Ai sẽ cần bảo trì ,quản lí và đảm bảo cho hệ thống hoạt động ?

Hệ thống sẽ phải xử lý và làm việc với những trang thiết bị phần cứng nào ?

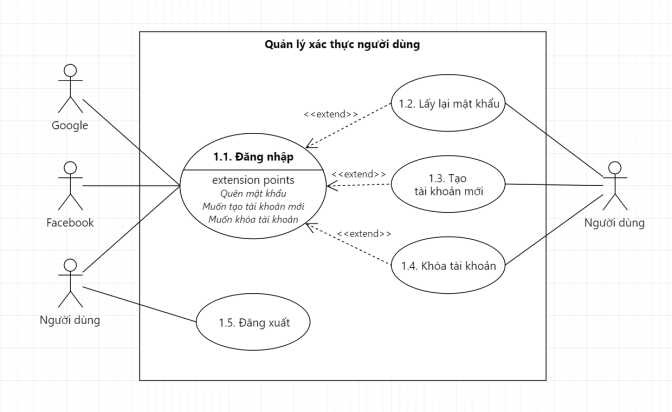
**2.2.3 Mô tả Use Case**

Giả dụ trường hợp ở đây: Anh em đã có Use Case Diagram, đã capture được tổng quan các requirement **theo góc nhìn của người dùng**. Đó là thứ để chúng ta bỏ vào các document như FRD hoặc SRS.

Tuy nhiên, Use Case Diagram khá là chung chung để các stakeholders có cái nhìn trực quan về những requirements được mô tả. Do đó, anh em cần phải **diễn đạt nó một cách chi tiết hơn** nữa.

Anh em hãy cùng xét tới một Use Case đơn giản nhất, đó là Đăng nhập.

Ví dụ đối với diễn đàn Medium đi chẳng hạn. Chúng ta sẽ vẽ Use Case Diagram và viết đặc tả Use Case cho phân hệ quản lý xác thực người dùng như sau.



USE CASE SPECIFICATION

|  |  |
| --- | --- |
| **Use Case ID** | UC-1.1 |
| **Use Case Name** | Đăng nhập |
| **Description** | Là người dùng, tôi muốn đăng nhập vào ứng dụng để sử dụng dịch vụ  từ ứng dụng. |
| **Actor(s)** | Khách hàng, Google, Facebook |
| **Priority** | Must Have |
| **Trigger** | Người dùng muốn đăng nhập vào ứng dụng Medium |
| **Pre-Condition(s):** | * Tài khoản người dùng đã được tạo sẵn * Tài khoản người dùng đã được phân quyền * Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập |
| **Post-Condition(s):** | * Người dùng đăng nhập ứng dụng thành công * Hệ thống ghi nhận hoạt động đăng nhập thành công vào Activity Log. |
| **Basic Flow** | 1. Người dùng truy cập ứng dụng Medium.  2. Người dùng chọn phương thức đăng nhập bằng tài khoản Medium  3. Người dùng nhập tài khoản Medium và chọn lệnh đăng nhập  4. Hệ thống xác thực thông tin đăng nhập thành công và cho phép người  dùng truy cập ứng dụng  5. Hệ thống ghi nhận hoạt động đăng nhập thành công vào Activity Log. |
| **Alternative Flow** | 2a. Người dùng chọn phương thức đăng nhập bằng tài khoản Gmail  2a1. Hệ thống chuyển sang màn hình đăng nhập của Google  3a. Người dùng nhập tài khoản Google và chọn lệnh đăng nhập  4a. Google xác thực thông tin đăng nhập thành công và cho phép người  dùng truy cập ứng dụng  *Use Case tiếp tục bước 5.*    2b. Người dùng chọn phương thức đăng nhập bằng tài khoản Facebook  2b1. Hệ thống chuyển sang màn hình đăng nhập của Facebook  3b. Người dùng nhập tài khoản Facebook và chọn lệnh đăng nhập  4b. Facebook xác thực thông tin đăng nhập thành công và cho phép  người dùng truy cập ứng dụng  *Use Case tiếp tục bước 5.* |
| **Exception Flow** | 4c. Hệ thống xác thực thông tin đăng nhập không thành công và hiển  thị thông báo.  4c1.Người dùng chọn lệnh hủy đăng nhập. *Use Case dừng lại.*  4c2. Người dùng chọn lệnh lấy lại mật khẩu *Use Case tiếp tục Use Case UC1-3*  4c3. Người dùng chọn lệnh khóa tài khoản *Use Case tiếp tục Use Case UC1-4* |
| **Business Rules** | BR1.1-1: Người dùng nhập sai thông tin đăng nhập ở lần thứ 6 liên tiếp sẽ bị  khóa tài khoản 30 phút. |
| **Non-Functional Requirement** | NFR1.1-1: Time out cho màn hình đăng nhập dưới 60 giây.  NFR1.1-2: Mật khẩu của người dùng phải được hash bằng MD5. |

* 1. **Mô hình trạng thái**

Một trong năm biểu đồ UML được sử dụng để mô hình hóa bản chất động của hệ thống. Chúng xác định các trạng thái khác nhau của một đối tượng trong suốt thời gian tồn tại của nó và các trạng thái này được thay đổi bởi các sự kiện.

Chương 3: Thiết kế hệ thống

**3.1 Biểu đồ tuần tự**

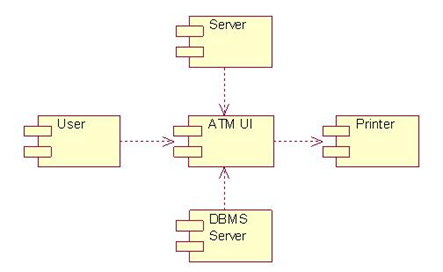
Biểu đồ tuần tự là một loại sơ đồ UML đại diện cho cách thức và thứ tự các đối tượng trong một hệ thống tương tác với nhau.

Biểu đồ tuần tự được sử dụng để thiết kế, ghi lại và xác nhận kiến trúc, giao diện và logic của các hệ thống bằng cách mô tả chuỗi các hành động cần được thực hiện để hoàn thành một nhiệm vụ.

**3.2** **Mô hình lớp chi tiết**

**3.2 Biểu đồ thành phần**

Khi thiết kế các hệ thống phức tạp chúng ta nên chia chúng ra thành nhiều hệ thống con (subsystem) để dễ thiết kế. Mỗi hệ thống con sau khi xây dựng có thể đóng gói thành một thành phần phần mềm được triển khai độc lập. Bản vẽ Component Diagram sẽ giúp chúng ta thể hiện cách chia hệ thống ra nhiều thành phần và quan hệ của chúng. Component Diagram là bản vẽ cho biết cấu trúc của hệ thống theo thành phần phần mềm. Chúng ta xem một ví dụ về Component Diagram như sau:



Ví dụ trên cho thấy hệ thống phần mềm ATM chia ra thành 5 thành phần là ATM UI, Server, User, DBMS Server và Printer. Trong đó ATM UI sử dụng chức năng của các thành phần còn lại để vận hành hệ thống.

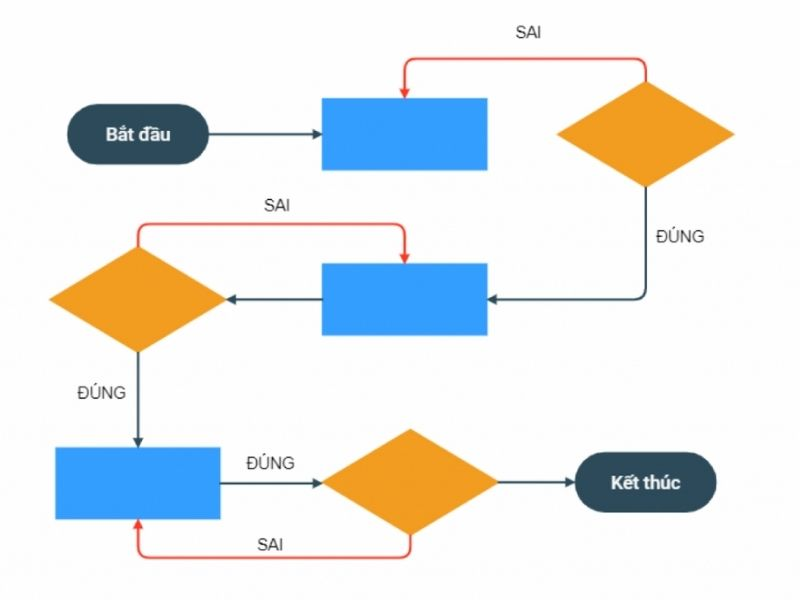
**3.4 Biểu đồ triển khai:**

Các bước để tạo Flowchart rất đơn giản nếu như bạn thực hiện đủ và chính xác 3 bước mà Tanca tổng hợp giúp bạn trong bài viết dưới đây. Tuy nhiên, trước đó Tanca sẽ đưa ra cho bạn một vài **quy tắc vẽ Flowchart**quan trọng nhất  nhé!

**Quy tắc 1: Sử dụng nhất quán các thành phần**

Yếu tố đầu tiên chính là tính nhất quán từ hình khối, đường dẫn, văn bản. Trong đó:

* Hình elip: Thể hiện điểm bắt đầu và kết thúc của một quy trình.
* Hình chữ nhật: Các bước hay hành động triển khai do cá nhân đảm nhiệm.
* Hình thoi: Dùng khi có quyết định hay phê chuẩn cần lựa chọn.
* Đường dẫn mũi tên chỉ hướng của dòng chảy các bước.



*Flowchart với màu sắc bắt mắt, tiện theo dõi*

#### **Quy tắc 2: Sắp xếp luồng dữ liệu khoa học trên cùng một trang**

Khi bạn muốn tạo ra một biểu đồ hoàn chỉnh, hiệu quả và tốt nhất thì nên tối ưu trên cùng một trang. Đây có thể nói là một quy tắc vô cùng quan trọng và then chốt trong quá trình vẽ biểu đồ Flowchart.

Trong trường hợp, biểu đồ có kích thước lớn, bạn có thể sử dụng các mẹo như sau:

* Tối ưu câu từ, ý nghĩa một cách ngắn gọn và xúc tích nhất, ;làm cho lưu đồ nhỏ hơn. Phông chữ đi kèm lúc này cần hiệu chỉnh to lên để bù đắp việc giảm tỷ lệ trên. Tổng quan về phần nhìn bạn dễ nhận thấy giao diện không bị quá bé.
* Tùy theo số lượng bước, người dùng có thể sắp xếp đường dẫn từ trái sang phải. Sau đó, những trình tự còn lại sẽ hạ xuống dòng dưới thực hiện tiếp.
* Ngoài ra, một cách xử lý khác là chia Flowchart lớn ra thành các tập hợp nhỏ. Đầu tiên, bạn hãy vẽ lưu đồ miêu tả tổng quan các bước hoàn chỉnh của quy trình. Trong mỗi bản chính này sẽ chứa một siêu liên kết đến các sơ đồ riêng biệt hiển thị chi tiết bước đó.

#### **Quy tắc 3: Đặt dòng trả về phía dưới biểu đồ**

Quy tắc còn lại trong vẽ Flowchart đó chính là đặt dòng trả về phía dưới biểu đồ luồng. Thực tế, chúng ta đọc văn bản từ đầu trang xuống một cách tự nhiên theo tuần tự. Do đó, mọi dòng trả ngược về trước cần đặt bên dưới.

**3.5 Thiết kế cơ sở dữ liệu**

Trong trường hợp xuất hiện 2 dòng cần trả về tuyệt đối không được trùng nhau.

1. Bước đầu tiên để thiết kế cơ sở dữ liệu là xác định mục đích của thiết kế.
2. Bước tiếp theo là thu thập tất cả các loại thông tin mà bạn muốn ghi lại trong cơ sở dữ liệu.
3. Hãy chia thông tin của bạn thành các đối tượng hoặc chủ đề chính. Mỗi chủ đề sẽ là một bảng riêng
4. Hãy quyết định thông tin bạn muốn lưu trữ trong bảng. Mỗi mục sẽ trở thành một trường và được hiển thị dưới dạng cột trong bảng.
5. Chọn khóa chính của mỗi bảng. Khóa chính là một cột được sử dụng để xác định mỗi hàng.
6. Thiết lập mối quan hệ trong bảng và quyết định cách dữ liệu trong bảng có liên quan đến dữ liệu trong bảng khác.

**3.6 Thiết kế giao diện**

Chương 4: Cài đặt

Để thiết kế trang web ta sử dụng công cụ máy tính và những kĩ thuật chúng ta đã học được công nghệ sử dụng là apps **visual studio code**

Các chức năng đã cài được là :Atom one dark Theme,Live sever,Color Picker,Auto Close Tag,....

Kết luận : Làm việc với Visual studio code tôi rất hài lòng khi dùng được những tính năng hữu ích , và khi tải xuống nó cũng rất mau , sau khi làm việc với visua studio code tôi cảm thấy rất hào hứng và muốn ngồi với nó cả ngày .

Đề tài của tôi là :**Mạng việc làm**

**Cảm ơn các bạn đã theo dõi trình bày đồ án của mình!**