**Báo cáo tiến độ lần 1**

**Chủ đề: Quản lý đề cương chi tiết Khoa Kỹ Thuật và Công Nghệ trường Đại học Trà Vinh**

**TỔNG QUAN**

1. **Lý do chọn đề tài:**

Hiện nay giảng viên của Khoa KTCN vẫn đang sử dụng, quản lý đề cương chi tiết theo cách truyền thống, sẽ có những hạn chế như: việc kiểm tra giáo viên có nộp đầy đủ đề cương chi tiết hay không, tốn chi phí in ấn, giảng viên có thể để quên, thất lạc, bảo quản không được lâu. Thông qua việc đó, chúng tôi muốn làm một trang web “*quản lý đề cương chi tiết*” để việc quản lý của các giáo viên được dễ dàng hơn.

1. **Mô tả hệ thống**

Mỗi giáo viên cần quản lý đề cương chi tiết của sinh viên theo từng học kỳ, mỗi giáo viên sẽ được cấp một tài khoản đăng nhập vào “*quản lý đề cương chi tiết online*”.

Mỗi khi học kì mới bắt đầu, giáo viên sẽ tải đề cương chi tiết của học sinh mà mình hướng dẫn lên website để quản lý. Thanh tra kiểm tra sẽ được cấp tài khoản có quyền admin, có quyền vào xem và kiểm tra các đề cương chi tiết mà giáo viên khác tải lên.

Tài khoản admin của thanh tra có chức năng phân quyền bất kì tài khoản giáo viên nào mình muốn, giáo viên được phân quyền sẽ có các quyền và chức năng như tài khoản admin.

**CHƯƠNG 1: XÁC ĐỊNH YÊU CẦU**

1. **. Yêu cầu chức năng nghiệp vụ**

Bộ phận người dùng: Giảng viên khoa KTCN trường Đại học Trà Vinh và thanh tra kiểm tra.

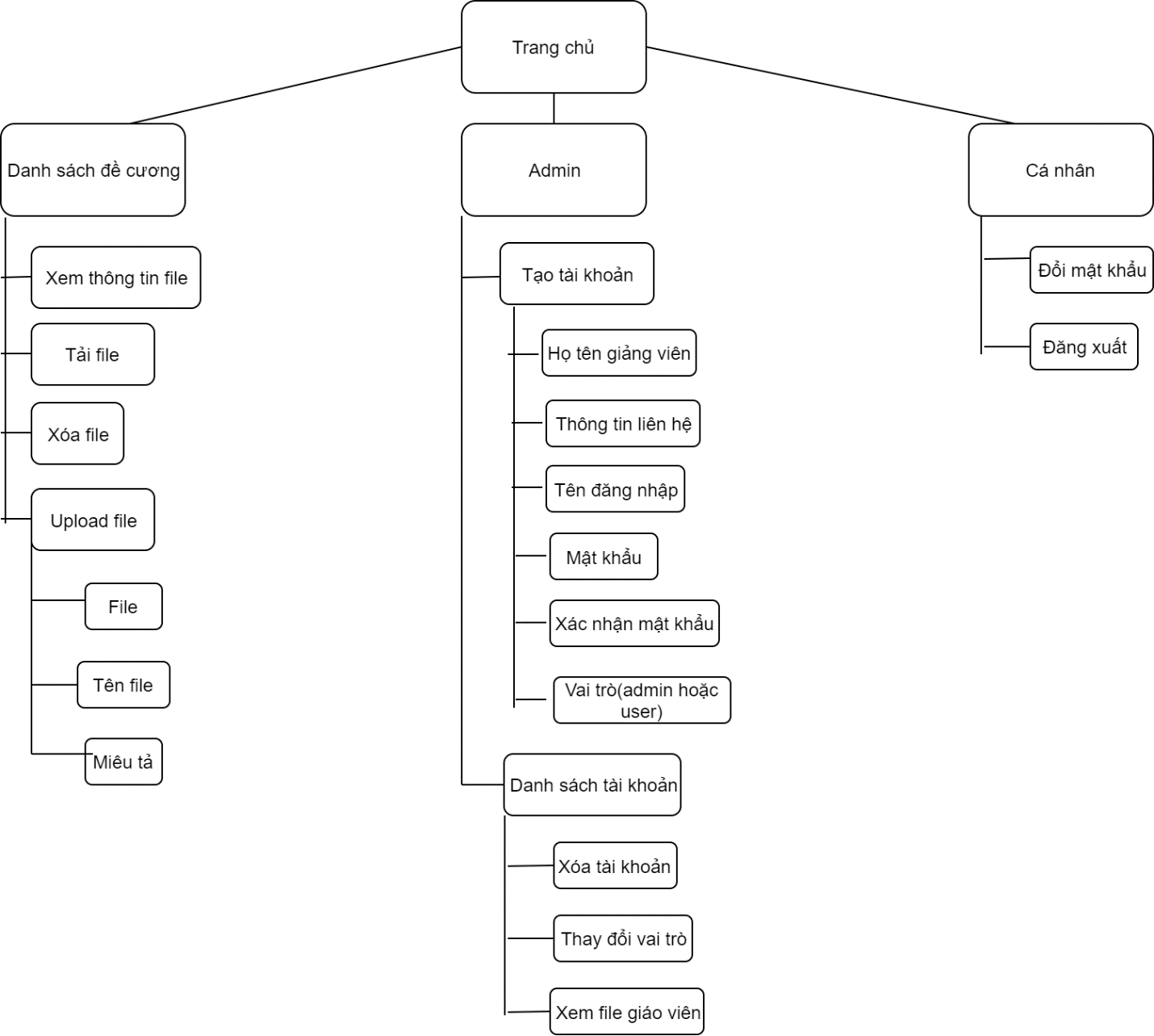
Chức năng:

* Nhập liệu:
  + Nhập thông tin giảng viên: họ tên giảng viên, mã giảng viên, bộ môn, năm học.
  + Cập nhật: tạo mới, xóa , sửa.
* Lưu trữ:
  + Lưu thông tin giảng viên, file đề cương chi tiết.

1. **. Yêu cầu chất lượng**

* Cập nhật
* Tiện dụng
* Hiệu quả

1. **. Cây giao diện**



**Ngôn ngữ và cơ sở dữ liệu sử dụng**

1. **JavaScript và NodeJS**
   1. **Tổng quan:**

JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ (small and lightweight). Khi nằm bên trong một môi trường (host environment), JavaScript có thể kết nối tới các object của môi trường đó và cung cấp các cách quản lý chúng (object).

Javascript ngày càng trở nên phổ biến hơn với nhiều tính năng và các thư viện được hỗ trỡ cho developer, điều đó khiến cho các giao diện web càng trở nên sinh động hơn. Mọi thứ mà chúng ta có thể làm được trên web ngày nay là Javascript có thể chạy được trên server, cũng như chạy được trên browser, điều này là khó tưởng tượng trong những năm trở lại đây, hoặc nó chỉ đóng gói trong môi trường sandboxed như Flash hoặc JavaApplets. Thật vậy, rõ ràng rằng từ xưa đến nay chúng ta vẫn quan niệm rằng lập trình bên phía server chỉ dùng được những ngôn ngữ như php, ruby, ... nhưng từ khi node js ra đời nó mang đến tư tưởng mới cho việc lập trình cả bên phía server cũng như bên phía client. Và đối với node js thì chúng ta biến những điều mà trước kia chỉ có thể thao tác được bên phía client thì nay cũng thao tác và xử lý được trên server, và ngược lại.

NodeJS được viết bằng ngôn ngữ javascript, nó là một trình biên đóng gói của Google’s V8 JavaScript engine, libuv platform abstraction layer, và một thư viện lõi được viết bằng Javascript. Mục tiêu của NodeJS là làm cho web có khả năng push như trong một số ứng dụng gmail. NodeJS cung cấp công cụ giúp lập trình viên có thể làm việc trong non-blocking, mô hình I/O . Sau hơn 20 năm nghiên cứu, xây dựng và phát triển, nhóm kĩ sư đã cho ra đời sản phẩm ứng dụng web node js chạy thời gian thực và kết nối 2 chiều client và server, cho phép trao đổi dữ liệu một cách tự do.

* 1. **Ưu điểm NodeJS:**
* Đặc điểm nổi bật của NodeJS là nó nhận và xử lý nhiều kết nối chỉ với một single-thread. Điều này giúp hệ thống tốn ít RAM nhất và chạy nhanh nhất khi không phải tạo thread mới cho mỗi truy vấn giống PHP. Ngoài ra, tận dụng ưu điểm non-blocking I/O của Javascript mà NodeJS tận dụng tối đa tài nguyên của server mà không tạo ra độ trễ như PHP.
* Ứng dụng trên 1 trang( Single page Application): Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh.
* Shelling tools unix: NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu xuất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.
* Streamming Data (Luồng dữ liệu): NodeJS xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.
* Ứng dụng Web thời gian thực: Với sự ra đời của các ứng dụng di động & HTML 5 nên NodeJS rất hiệu quả khi xây dựng những ứng dụng thời gian thực (real-time applications) như ứng dụng chat, các dịch vụ mạng xã hội như Facebook, Twitter,…
  1. **Nhược điểm của NodeJS:**
* Ứng dụng tốn tài nguyên: NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm 1 trình biên dịch của NodeJS sẽ lâu hơn 1 chút nếu xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding, encryption... hoặc các ứng dụng tương tự như vậy thì không nên dùng Node.js.
* NodeJS được giới thiệu khá muộn so với các ngôn ngữ khác cho xử lý sever appweb nên cộng đồng hướng dẫn tại Việt Nam còn ít.

1. **. Mongodb, NoSql là gì?**

MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++. Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng.

NoSQL là 1 dạng CSDL mã nguồn mở không sử dụng Transact-SQL để truy vấn thông tin. NoSQL viết tắt bởi: None-Relational SQL, hay có nơi thường gọi là Not-Only SQL. CSDL này được phát triển trên Javascript Framework với kiểu dữ liệu JSON. (Cú pháp của JSON là “key:value”) NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu xót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS về tốc độ, tính năng, khả năng mở rộng, memory cache,...

1. **. Ưu, nhược điểm mongodb**
   1. **Ưu điểm**

* Ít schema hơn: Vì schema được sinh ra là để nhóm các đối tượng vào 1 cụm, dễ quản lý. Ví dụ như tạo 1 schema tên là Students chẳng hạn thì chỉ có những gì liên quan đến student thì mới được cho vào schema này. Trong khi đó trong mongodb thì chỉ 1 collection ta có thể chứa nhiều document khác nhau . Với mỗi document thì số trường, nội dung, kích thước lại có thể khác nhau.
* Cấu trúc của một đối tượng rõ ràng.
* Không có các Join phức tạp.
* Khả năng mở rộng cực lớn: việc mở rộng dữ liệu mà không phải lo đến các vấn đề như khóa ngoại, khóa chính, kiểm tra ràng buộc, ... MongoDB cho phép thực hiện replication và sharding nên việc mở rộng cũng thuận lợi hơn.
* Sử dụng bộ nhớ trong để lưu giữ cửa sổ làm việc cho phép truy cập dữ liệu nhanh hơn. Việc cập nhật được thực hiện nhanh gọn nhờ update tại chỗ (in-place).
  1. **Nhược điểm**
* Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn
* Như đã giới thiệu ở trên, mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

1. **. Lý do sử dụng mongodb cho đề tài**

* Website có tính chất INSERT cao Bởi vì mặc định MongoDB có sẵn cơ chế ghi với tốc độ cao và an toàn.
* Website ở dạng thời gian thực nhiều, nghĩa là nhiều người thao tác với ứng dụng. Nếu trong quá trình load bị lỗi tại một điểm nào đó thì nó sẽ bỏ qua phần đó nên sẽ an toàn.