# PHẦN GIỚI THIỆU

## Đặt vấn đề

Ngày nay nhu cầu học thêm của học sinh, sinh viên ngày càng tăng cao. Nhưng đa số học sinh tìm kiếm gia sư dựa trên lời giới thiệu của bạn bè hay nhờ các giáo viên trong trường nên không có nhiều lựa chọn, khó tìm được gia sư phù hợp. Bên cạch đó, các trung tâm gia sư cũng xuất hiện rất nhiều nhưng lại rất khó để biết được chất lượng của trung tâm cũng như của giáo viên. Học sinh rất cần một kênh để tìm kiếm gia sư mà ở đó học sinh có thể tìm những gia sư phù hợp về vị trí địa lý, môn học hay kinh nghiệm...

Những gia sư có khả năng và trình độ cần một kênh để quảng bá hình ảnh bản thân. Đồng thời, họ cũng khó khăn trong việc liên lạc với những học sinh đang cần sự giúp đỡ của mình.

Do đó, chúng tôi đề xuất “Xây dựng hệ thống mạng gia sư”.

## Mục tiêu của đề tài

* Xây dựng một mạng xã hội dành riêng cho gia sư và phụ huynh học sinh với đầy đủ tính năng cần thiết.
* Tích hợp Google Map vào việc hỗ trợ tìm kiếm gia sư và học sinh.

## 3. Đối tượng nghiên cứu và phạm vi nghiên cứu

### 3.1 Đối tượng nghiên cứu:

* Nhu cầu học thêm và dạy thêm của học sinh, sinh viên, giáo viên.
* Mạng xã hội có và không có hoạt động trong lĩnh vực giáo dục.

### 3.2 Phạm vi nghiên cứu:

* Về lý thuyết:
* Nghiên cứu cách phân tích, thiết kế hệ thống và thiết kế các mô hình: Sơ đồ Use Case, mô hình dữ liệu mức quan niệm (CDM), thiết kế cở sở dữ liệu.
* Về chức năng:

### Nội dung nghiên cứu

Nội dung nghiên cứu của đề tài chủ yếu các công nghệ: NodeJS, ExpressJS, AngularJS và MongoDB. Với nội dung này, nhóm có 2 thành viên tham gia thực hiện và nghiên cứu.

Huỳnh Nguyễn Minh Nguyệt: phát triển phân hệ gia sư

Tuấn: phát triển phân hệ PHHS

## 4.1 Về lý thuyết

* Cả hai cùng nghiên cứu công nghệ NodeJS, ExpressJS, AngularJS và MongoDB

## 4.2 Về chức năng

Nguyệt xây dựng các chức năng như: tìm gia sư theo môn học; liên lạc với gia sư; mở lớp; đăng ký lớp; tạo, cập nhật thông báo của gia sư, cập nhật thông tin gia sư

### Bố cục của quyển luận văn

Nội dung của quyển luận văn bao gồm 3 phần:

**Phần giới thiệu**: đưa ra lý do thực hiện đề tài này cùng với mục tiêu, đối tượng và nội dung nghiên cứu của đề tài. Nội dung chính của phần này gồm: đặt vấn đề, mục tiêu của đề tài, đối tượng nghiên cứu, nội dung nghiên cứu, bố cục của quyển luận văn

**Phần nội dung**: phần này gồm có 4 chương

**Chương 1:** đặt tả yêu cầu. Chương này giới thiệu tổng quan về mạng gia sư cũng như quy trình tìm kiếm gia sư và học sinh.

**Chương 2:** cơ sở lý thuyết. Giới thiệu sơ lược về kiến trúc chương trình với công nghệ MongoDB, ExpressJS, NodeJS, AngularJS và 4 công nghệ này.

**Chương 3:** thiết kế giải pháp. Chương này sẽ đưa ra thiết kế kiến trúc, thiết kế dữ liệu, thiết kế giải thuật, thiết kế giao diện cho chức năng trong hệ thống.

**Chương 4:** đánh giá kiểm thử. Đưa ra mục tiêu kế hoạch, các trường hợp kiểm thử đối với chức năng của hệ thống.

**Phần kết luận**: nêu lên những kết quả đạt được và hướng phát triển cho đề tài.

# PHẦN NỘI DUNG

# CHƯƠNG 1. ĐẶT TẢ YÊU CẦU

## 1.1 Tổng quan về mạng gia sư

Mạng gia sư hổ trợ cho phụ huynh học sinh tìm gia sư và gia sư tìm học sinh theo môn hoặc theo vị trí địa lý. Hệ thống cung cấp chức năng cho 3 loại người dùng: người quản trị, phụ huynh học sinh và gia sư.

Hình 1.1 Hệ thống mạng gia sư

## 1.2 Dành cho phụ huynh học sinh

Phụ huynh học sinh có thể tìm gia sư theo một trong hai cách: tìm theo môn học và tìm theo vị trí địa lý. Bên cạnh đó, PHHS còn có thể đăng ký vào các lớp học do gia sư mở hoặc yêu cầu mở lớp học mới.

Hình 1.2 Sơ đồ use case của phụ huynh học sinh

### 1.2.1. Tìm gia sư theo môn học

Khi PHHS truy cập vào hệ thống, PHHS có thể sử dụng chức năng tìm kiếm gia sư bằng môn học để tìm gia sư theo nhu cầu. Kết quả tìm kiếm là danh sách các gia sư phù hợp với yêu cầu cùng với thông tin cơ bản của họ như: avatar, giới tính, họ tên, nơi công tác, số năm kinh nghiệm,.. Để liên lạc với gia sư, PHHS cần phải đăng nhập vào hệ thống và chọn chức năng liên lạc. Sau đó, hệ thống sẽ hiển thị form liên lạc với nội dung: môn muốn học, khả năng của học sinh, hạn chế (khuyết điểm) của học sinh, nguyện vọng của học sinh, thông tin khác. Sau khi gia sư hồi âm hệ thống sẽ gửi thông báo đến tài khoản của PHHS ở Tutor.ct, đồng thời gửi email cho PHHS.

### 1.2.2 Tìm gia sư theo vị trí địa lý

## 1.3 Dành cho gia sư

Gia sư có thể tìm học sinh theo một trong hai cách: tìm theo môn học và tìm theo vị trí địa lý. Ngoài ra, gia sư có thể mở lớp theo môn học để PHHS có thể đăng ký học.

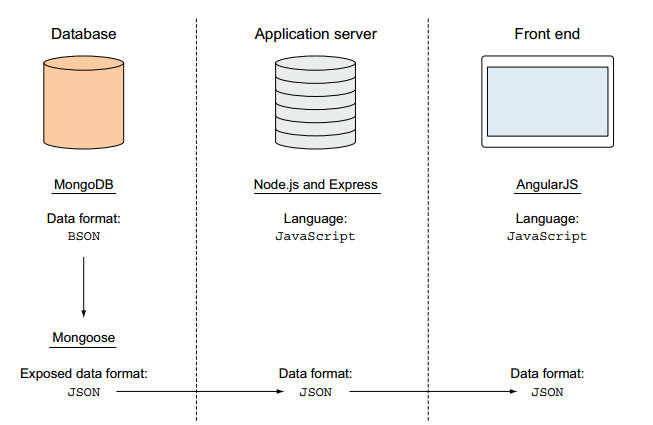
Hình 1.3 Sơ đồ use case của phụ huynh học sinh

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## Tổng quan về kiến trúc chương trình

Ứng dụng đươc web sử dụng 4 công nghệ MongoDB, Express, Node.js và AngularJS được gọi tắt là MEAN. Trong đó, mỗi công nghệ có vai trò:

* Node.js là môi trường thực thi Javascript phía server. Đây là nền tảng được xây dựng trên Google Chrome’s V8 JavaScript runtime.
* Express là framework nhẹ dùng để xây dựng ứng dụng web trên nền Node.
* MongoDB là hệ thống cơ sở dữ liệu NoSQL. MongDB lưu dữ liệu ở dạng JSON.
* AngulaJS là JavaScript framework. Nó cung cấp mộ số tính năng tuyệt vời như ràng buộc dữ liệu (data-binding) hai chiều. Đây là giải pháp hoàn chỉnh để phát triển giao diện.



Hình 2.1

## MongoDB

2.2.1 Giới thiệu

* MongoDB là cơ sở dữ liệu tài liệu (document database) mã nguồn mở cung cấp hiệu năng cao, khả năng sẵn sàng cao và có thể tự động mở rộng.
* Một bản ghi trong MongDB là một tài liệu, một cấu trúc dữ liệu bao gồm các cặp trường (field) và giá trị (value). Các tài liệu trong MongoDB tương tự như các đối tượng JSON. Các giá trị của trường có thể bao gồm các tài liệu, mảng và mảng các tài liệu.

{

ten: sue, 🡨 trường: giá trị

tuoi: 26, 🡨 trường: giá trị

so\_thich: [“ xem phim”, “doc sach”] 🡨 trường: giá trị

}

Hình 2.2 Cấu trúc document

2.2.2 Khái niệm

**Cơ sở dữ liệu**

Trong MongoDB, CSDL là tập hợp các collections.

**Collection**

Collection là một nhóm các document. Nó tương đương với các bảng trong CSDL quan hệ

**Document**

Một document trong MongoDB, có cấu trúc tương tự kiểu dữ liệu JSON, là một tập hợp các cặp trường – giá trị. Các document có schema động, nghĩa là document trong cùng một Collection không cần thiết phải có cùng một tập hợp các trường hoặc cấu trúc giống nhau, và các trường chung trong document của một collection có thể giữ các kiểu dữ liệu khác nhau. Document có thể xem là tương đương với các hàng trong bảng trong CSDL quan hệ.

Cấu trúc của một document:

{

<trường>: <giá trị>,

<trường>: <giá trị>,

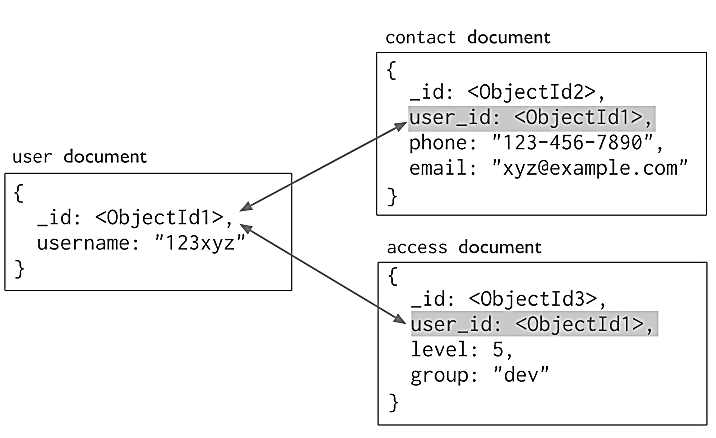
…

}

2.2.3 Quan hệ (relationship)

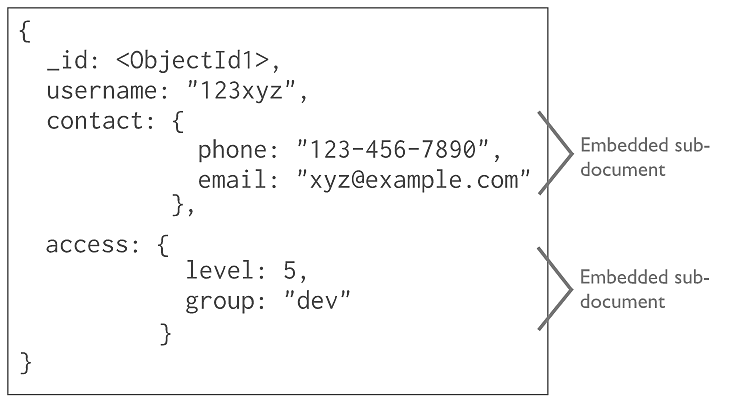
MongoDB cung cấp 2 cách để tạo quan hệ là references và embedded

* **References**: References lưu trữ các mối quan hệ giữa dữ liệu bằng cách chứa liên kết hoặc tham chiếu từ tài liệu này sang tài liệu khác. Các ứng dụng có thể dùng những references này để truy cập dữ liệu có liên quan.



Hình 2. Mô tả quan hệ references

* **Embedded**: Embedded lưu trữ mối quan hệ giữa dữ liệu bằng cách lưu trữ dữ lilệu liên quan trong một cấu trúc tài liệu đơn. Trong MongDB, document có thể nhúng (embed) vào như một trường (field) hay một mảng.



Hình 2. Mô tả quan hệ embeded

2.2.4 Các thao tác và lệnh cơ bản

**Tạo CSDL**

Để tạo CSDL dùng lệnh **use <tên CSDL>.** Nếu CSDL chưa tồn tại thì MongoDB sẽ tạo ra CSDL mới. Nếu CSDL đã tồn tại thì sẽ trả về CSDL đang tồn tại. Tuy nhiên, CSDL chỉ thực sự được tạo ra khi người dùng insert giá trị đầu tiên vào CSDL này. Lệnh này cũng dùng để chọn CSDL khi muốn thực hiện các thao tác lên CSDL.

Sử dụng lệnh **db** để xe danh sách các CSDL có trong MongDB.

**Thêm document**

Thêm 1 document

**db.collection.insertOne(<document>)** dùng để thêm 1 document vào collection.

Ví dụ:

**db.inventory.insertOne(**

{ item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" }

}

)

Thêm nhiều document

**db.collection.insertMany([<document>,<document>,…])** dùng để thêm nhiều document vào collection. Tham số của phương thức này là mảng các documents.

Ví dụ:

**db.inventory.insertMany**([

{ item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } },

{ item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom: "cm" } },

{ item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" } }

])

**Truy vấn**

Để chọn tất cả document trong collection, truyền document rỗng vào phương thức find **db.<tên collection>.find({})**

Ví dụ: **db.inventory.find( {} )**

Các thao tác này tương ứng với câu lệnh SQL: SELECT \* FROM inventory

Để tìm theo giá trị, sử dụng <trường>: <giá trị> để truyền vào phương thức find.

Ví dụ: **db.inventory.find( { status: "D" } )**

Các thao tác này tương ứng với câu lệnh SQL: SELECT \* FROM inventory WHERE status = “D”.

**Cập nhật document**

Cập nhật 1 document

**db.collection.updateOne(<filter>, <update>, <options>)** dùng để cập nhật giá trị của 1 document. Dùng $set để đưa giá trị cập nhật vào.

Ví dụ:

db.inventory.updateOne(

{ item: "paper" },

{

$set: { "size.uom": "cm", status: "P" }

}

)

Cập nhật nhiều document

**db.collection.updateMany(<filter>, <update>, <options>)** dùng để cập nhật giá trị của nhiều document. Dùng $set để đưa giá trị cập nhật vào.

Ví dụ:

db.inventory.updateMany(

{ "qty": { $lt: 50 } }, {

$set: { "size.uom": "cm", status: "P" }

}

)

Cập nhật tất cả document có qty < 50.

**Xóa document**

Xóa 1 document

Để xóa 1 document phù hợp với điều kiện lọc sử dụng phương thức **db.inventory.deleteOne(<filter>)**

Ví dụ: **db.inventory.deleteOne( { status: "D" } )**

Xóa nhiều document

Để xóa nhiều document phù hợp với điều kiện lọc sử dụng phương thức **db.inventory.deleteMany().**

Ví dụ: db.inventory.deleteMany({ status : "A" })

## NodeJS

## 2.4 ExpressJS

## 2.5 AngularJS

CHƯỞNG 3. THIẾT KẾ GIẢI PHÁP

CHƯƠNG 4. ĐÁNH GIÁ KIỂM THỬ

PHẦN KẾT LUẬN

# TÀI LIỆU THAM KHẢO

Simon Holmes, 2016. Getting MEAN with Mong, Express, Angular, and Node. Manning Publications

Jay Rai, 2014. An Introduction to the MEAN Stack, <https://www.sitepoint.com/introduction-mean-stack/>

<https://docs.mongodb.com/manual/>

PHỤ LỤC