

**Báo Cáo Khóa Luận**

**XÂY DỰNG ỨNG DỤNG**

**TẠO THỜI KHÓA BIỂU TỰ ĐỘNG CHO KHOA TOÁN – TIN HỌC**

Sinh viên thực hiện:

Võ Trần Gia Hào

Lương Võ Nhật Nhi

Giảng viên hướng dẫn:

TS. Trần Anh Tuấn (A)

**LỜI CẢM ƠN**

Chúng em xin dành lời cảm ơn chân thành đến Thầy Trần Anh Tuấn - người đã hướng dẫn chúng em trong quá trình thực hiện báo cáo khóa luận này. Sự hỗ trợ của Thầy đã giúp chúng em hoàn thành tốt báo cáo này, chúng em rất biết ơn và trân trọng công lao của Thầy. Cũng xin gửi lời cảm ơn đến gia đình, bạn bè đã ở bên giúp đỡ, động viên con/tôi trong suốt khoảng thời gian vừa qua.

**MỤC LỤC**

[Phụ lục hình ảnh 4](#_heading=h.gjdgxs)

[Phụ lục bảng 5](#_heading=h.30j0zll)

[Tóm tắt 6](#_heading=h.1fob9te)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 7](#_heading=h.2et92p0)

[1.1 Giới thiệu bài toán 7](#_heading=h.tyjcwt)

[1.2 Mục tiêu và nhiệm vụ 7](#_heading=h.wd3c8mj5aqyz)

[1.3 Tính thực tiễn của đề tài 8](#_heading=h.7wi4xucxhuol)

[1.4 Hướng giải quyết 9](#_heading=h.a0498zc7eek3)

[1.5 Bố cục luận văn 9](#_heading=h.uo99idz7d5dn)

[CHƯƠNG 2 - CƠ SỞ LÝ THUYẾT VÀ KỸ THUẬT LIÊN QUAN 10](#_heading=)

[2.1 Bài toán sắp xếp thời khóa biểu 10](#_heading=)

[2.1.1 Các tham số đầu vào của bài toán 10](#_heading=)

[2.1.2 Nhắc lại các ràng buộc của bài toán 10](#_heading=)

[2.1.3 Mục tiêu của bài toán 12](#_heading=)

[2.2 Các kỹ thuật/thuật toán liên quan 12](#_heading=h.euugrj301bs4)

[2.2.1 Giới thiệu thuật toán di truyền Genetic Algorithm 12](#_heading=h.gj12nfth2i0q)

[2.2.1.1 Khái niệm Di truyền và Tiến hóa 12](#_heading=h.drl8ui3m0aft)

[2.2.1.2 Thuật toán di truyền Genetic Algorithm 12](#_heading=h.6ksyv57diu21)

[2.2.1 Mô hình hóa bài toán (In progress) 14](#_heading=h.z0bdxkfr6a5s)

[2.3 Kiến trúc hệ thống 14](#_heading=)

[2.3.2 Kiến trúc client-server 15](#_heading=h.clz7i44i4l26)

[2.4 Các công nghệ sử dụng 17](#_heading=)

[2.4.1 Angular framework 17](#_heading=h.aomtwvsmd73s)

[2.4.1.1 Sơ lược về front-end framework 17](#_heading=h.lbhxlzkdk5o8)

[2.4.1.2 Angular framework 17](#_heading=h.3q9xs2j9ysrk)

[2.4.1.3 Sử dụng Angular framework trong phần mềm sắp xếp thời khóa biểu tự động 17](#_heading=h.wa9mu4pxhffl)

[2.4.2 Express js 18](#_heading=h.gg4ob0si196e)

[2.4.2.1 Sơ lược về Express js 18](#_heading=h.dlgpdiu5x962)

[2.4.2.2 Express js trong phần mềm tạo thời khóa biểu tự động 19](#_heading=h.kpqgnmrstwd4)

[2.4.3 Hệ quản trị cơ sở dữ liệu MySQL 19](#_heading=h.ikucavobj4tg)

[2.4.3.1 Sơ lược về hệ quản trị cơ sở dữ liệu MySQL 19](#_heading=h.y0ni0hdfwqew)

[2.4.3.2 Sử dụng MySQL trong phần mềm sắp xếp thời khóa biểu tự động 20](#_heading=h.30fgqz5oc3g1)

[2.4.4 Socket IO 21](#_heading=h.70ati9ojg9v2)

[2.4.4.1 Sơ lược về Socket IO 21](#_heading=h.adte80ezldmt)

[2.4.4.2 Sử dụng Socket IO trong phần mềm sắp xếp thời khóa biểu tự động 21](#_heading=h.geokpftibagd)

[2.4.5 JWT (JSON Web Token) 21](#_heading=h.ulsrs7ktk4se)

[2.4.5.1 Sơ lược về JWT (JSON Web Token) 21](#_heading=h.kyzb3n2ox3i)

[2.4.5.2 Sử dụng JWT trong phần mềm xây dựng thời khóa biểu tự động 22](#_heading=h.3nxlymokz0n4)

[CHƯƠNG 3 - PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 23](#_heading=h.mpb1jm69lq4)

[3.1 Đặc tả hệ thống 23](#_heading=h.tfheho791c8g)

[3.1.1 Nhiệm vụ 23](#_heading=h.38gdbgr1xk72)

[3.1.2 Quy trình nghiệp vụ 23](#_heading=h.bbmmosoqfcwo)

[3.2 Phân tích hệ thống 23](#_heading=h.111o5pdbh358)

[3.2.1 Yêu cầu ứng dụng 23](#_heading=h.bayl2tpfjfty)

[3.2.2 Yêu cầu người dùng 23](#_heading=h.f5zhwzeom9zm)

[3.3 Thiết kế hệ thống 23](#_heading=h.8d1pa0y5ax30)

[3.3.1 Mô hình cơ sở dữ liệu 23](#_heading=h.zfry9hgt4j81)

[3.3.2 Sơ đồ Use case (Use case diagram) 23](#_heading=h.f5r3o7yqetlo)

[3.3.3 Sơ đồ tuần tự (Sequence diagram) 23](#_heading=h.l2jhlg60p14k)

[3.3.4 Mô hình phân rã chức năng (BFD) 23](#_heading=h.xcojo890q240)

[3.3.5 Mô hình dòng dữ liệu (DFD) 23](#_heading=h.j394oc97dgsg)

[CHƯƠNG 4 - HIỆN THỰC CHƯƠNG TRÌNH 23](#_heading=h.mc9xrhyrl93r)

[CHƯƠNG 5 - TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI 23](#_heading=h.f9ghqtxdvoh6)

[TÀI LIỆU THAM KHẢO 23](#_heading=h.kfwz8js0dvna)

[KẾT LUẬN 24](#_heading=h.3dy6vkm)

# Phụ lục hình ảnh

# Phụ lục bảng

# Tóm tắt

# **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

## Giới thiệu bài toán

Trong thời đại công nghệ 4.0 như hiện nay, mọi ngành nghề, mọi quy trình đang dần áp dụng các công nghệ tự động hóa bằng cách tạo ra các hệ thống, phần mềm chuyên dụng, nhằm đẩy nhanh tiến độ, mang lại hiệu quả trong công việc và nâng cao chất lượng của kết quả hoặc sản phẩm đầu ra. Chẳng hạn, trong lĩnh vực sản xuất, các nhà máy đã áp dụng robot hóa để tự động hóa các quy trình sản xuất như lắp ráp, kiểm tra chất lượng sản phẩm.

Mặc dù vậy, vẫn còn rất nhiều nhu cầu chưa được đáp ứng hoặc không thể đáp ứng do quá trình xây dựng hệ thống, phần mềm gặp nhiều khó khăn, hoặc đã phát triển được phần mềm nhưng không giải quyết được vấn đề hiện có vì nhiều lý do. Một trong những lý do đó là tính cụ thể, riêng biệt của từng vấn đề. Quay trở lại bài toán mà khóa luận này đề ra, trong việc quản lý thời khóa biểu của một trường đại học, cao đẳng, mỗi lớp học, mỗi giáo viên có thể có những yêu cầu khác nhau về thời gian học, sự sắp xếp phòng học hoặc thời lượng của mỗi buổi học, vv. Thậm chí, mỗi khoa trong trường cũng có những hiện trạng và nhu cầu khác nhau trong việc sắp xếp thời khóa biểu. Những yếu tố này tạo ra một độ phức tạp đáng kể khi thiết kế thời khóa biểu chung cho cả trường.

Chính vì lý do này, các trường đại học, cao đẳng, gặp nhiều khó khăn trong việc sắp xếp thời khóa biểu mặc dù đã có rất nhiều các phần mềm sắp xếp thời khóa biểu tự động đã được phát hành. Vì vậy, việc tạo ra một phần mềm sắp xếp thời khóa biểu tự động dành riêng cho từng khoa hoặc từng trường đại học sẽ giúp giải quyết những nhu cầu và thách thức cụ thể mà từng đơn vị đang phải đối mặt. Đồng thời, điều này cũng sẽ là nền tảng cho việc phát triển các giải pháp tự động hóa thời khóa biểu cho các trường đại học và cao đẳng khác trên toàn quốc. Và đó cũng là lý do nhóm chúng em quyết định chọn đề tài “Xây dựng phần mềm sắp xếp thời khóa biểu tự động dành cho Khoa Toán – Tin học”.

## Mục tiêu và nhiệm vụ

- Tìm hiểu về các yêu cầu, nhu cầu và hiện trạng của khoa liên quan đến việc tạo thời khóa biểu

- Xây dựng phần mềm sắp xếp thời khóa biểu tự động

- Nghiên cứu thuật toán tối ưu để sắp xếp thời khóa biểu tự động

- Giải quyết một cách tối ưu các vấn đề đặt ra trong việc sắp xếp thời khóa biểu

## Tính thực tiễn của đề tài

Qua quá trình khảo sát về việc sắp xếp thời khóa biểu của khoa, nhóm em đã có nhìn nhận tổng quan về việc sắp xếp thời khóa biểu của khoa như sau: mỗi học kỳ khoa sẽ mở nhiều lớp học tương ứng với mỗi môn học, mỗi lớp học sẽ do một hoặc nhiều giảng viên phụ trách và tùy theo nhu cầu thì số lượng sinh viên cũng sẽ khác nhau. Mặt khác, mỗi giảng viên cũng phụ trách một hoặc nhiều lớp học cũng như môn học. Vì vậy, việc tính toán và sắp xếp giờ học cũng như phòng học sao cho phù hợp là việc tốn thời gian và công sức, đặc biệt là khi khoa không có phần mềm chuyên dụng để hỗ trợ. Chưa kể đến việc đáp ứng các yêu cầu cao hơn của một thời khóa biểu như: trong cùng một ngày, giảng viên không được có tiết ở hai cơ sở hoặc trong cùng một ngày giảng viên không được có hai tiết cách quá xa nhau,...

Hiện nay, khoa Toán – Tin học vẫn sử dụng phương pháp truyền thống – tạo thời khóa biểu thủ công dựa trên thời khóa biểu của học kỳ trước. Mặc dù phương pháp này vẫn mang lại kết quả như mong muốn, tuy nhiên không thực sự hiệu quả do có nhiều ràng buộc khi tạo thời khóa biểu.

Một số ràng buộc chính như sau:

*Ràng buộc về thời gian*

Đảm bảo thời lượng mỗi lớp học nằm trong một ca học

Đảm bảo sắp xếp các tiết học liên tục cho một môn học để tránh bị lủng tiết giữa các tiết học

*Ràng buộc về phòng học*

Đảm bảo sử dụng phòng học một cách tối ưu giữa các lớp với nhau

Đảm bảo mỗi phòng học trong một khung giờ chỉ được sử dụng bởi một lớp học

Đảm bảo sắp xếp phòng học phù hợp dành cho lớp lý thuyết và thực hành

Đảm bảo xác định phòng học phù hợp với số lượng sinh viên đăng ký môn học

*Ràng buộc về giảng viên*

Đảm bảo sắp xếp thời khóa biểu phù hợp với giờ dạy mà giảng viên cung cấp

Đảm bảo phù hợp với giảng viên có lịch nghỉ, lịch bận, xa nhà

Đảm bảo các lớp mà giảng viên đảm nhiệm không bị trùng lịch với nhau

Đảm bảo các lớp học trong cùng một ngày của giảng viên không cách quá xa nhau

Có thể xử lý được các yêu cầu đặc biệt từ giảng viên về việc sắp xếp thời khóa biểu

*Ràng buộc về địa điểm*

Đảm bảo sắp xếp thời gian dạy hai lớp khác cơ sở hợp lý sao cho giảng viên có đủ thời gian để di chuyển

Đảm bảo sắp xếp địa điểm các lớp dành cho sinh viên năm 1 và năm 2 ở cơ sở 2 (Linh Trung)

Các ràng buộc này làm cho việc sắp xếp thời khóa biểu thủ công không hiệu quả và kết quả đầu ra đôi khi vẫn không thể đáp ứng được một số các ràng buộc trên gây ra những khó khăn cho giảng viên cũng như sinh viên trong quá trình dạy và học.

## Hướng giải quyết

Qua quá trình khảo sát về quy trình sắp xếp thời khóa biểu của khoa và qua việc nghiên cứu, thảo luận, nhóm chúng em quyết định xây dựng một phần mềm sắp xếp thời khóa biểu tự động dựa trên những nhu cầu, vấn đề trong việc lập lịch cũng như quy trình lập lịch của khoa đang sử dụng, cụ thể như sau.

Nhóm em vẫn giữ nguyên quy trình sắp xếp thời khóa biểu của khoa: mỗi kỳ, khoa sử dụng lại thời khóa biểu của học kỳ trước tương ứng. Trong trường hợp giảng viên bị trùng giờ, giảng viên sẽ gửi yêu cầu đến khoa, yêu cầu đổi giờ dạy. Khoa sẽ tiếp nhận tất cả các yêu cầu của giảng viên liên quan đến việc đổi giờ dạy và tiến hành xử lý yêu cầu - chỉnh sửa lại thời khóa biểu. Quá trình này lặp đi lặp lại cho đến khi mọi giảng viên đã thống nhất được giờ dạy.

Dựa trên quy trình đó, chúng em sẽ thiết kế hệ thống có tính năng ghi nhận yêu cầu thay đổi giờ dạy của giảng viên. Trong đó, giảng viên sẽ cung cấp hai sự lựa chọn cho mỗi lớp học mà bản thân yêu cầu thay đổi. Hệ thống sẽ dựa trên dữ liệu lịch trống mà giảng viên cung cấp, cùng với các dữ liệu và ràng buộc khác để tính toán tự động và cho ra một thời khóa biểu đáp ứng được các yêu cầu của giảng viên và thỏa mãn các ràng buộc đã trình bày ở mục 1.3.

## Bố cục luận văn

Bố cục của bài luận văn được chia theo các chương chính như sau:

CHƯƠNG 1 - TỔNG QUAN VỀ ĐỀ TÀI

CHƯƠNG 2 - CƠ SỞ LÝ THUYẾT VÀ KỸ THUẬT LIÊN QUAN

CHƯƠNG 3 - PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

CHƯƠNG 4 - HIỆN THỰC CHƯƠNG TRÌNH

CHƯƠNG 5 - TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

TÀI LIỆU THAM KHẢO

# **CHƯƠNG 2 - CƠ SỞ LÝ THUYẾT VÀ KỸ THUẬT LIÊN QUAN**

## 2.1 Bài toán sắp xếp thời khóa biểu

### *2.1.1 Các tham số đầu vào của bài toán*

Các tham số đầu vào của bài toán bao gồm: danh sách các môn học được mở trong học kỳ, danh sách giảng viên, danh sách các khóa, danh sách lớp học, danh sách lớp - môn học, danh sách cơ sở, danh sách các phòng, phân loại theo phòng học dành cho lớp - môn học lý thuyết, lớp - môn học thực hành; bảng phân công giảng viên giảng dạy cho lớp - môn học.

* Danh sách các môn học được mở trong học kỳ: mỗi môn học có thông tin như tên môn, thời lượng, loại học phần (lý thuyết, thực hành).
* Danh sách giảng viên: mỗi giảng viên có thông tin về tên, khả năng giảng dạy các môn học, lịch trống (khi họ không thể dạy).
* Danh sách các khóa: các khóa sinh viên như: K2020, K2021, K2022,...
* Danh sách lớp học: danh sách các lớp học với các thông tin liên quan như tên lớp, khóa.
* Danh sách lớp - môn học: Danh sách các lớp - môn học, thông tin về số lượng sinh viên, loại lớp học (lý thuyết hoặc thực hành), thông tin lớp học liên quan.
* Danh sách cơ sở: Thông tin về các cơ sở của trường, bao gồm địa chỉ và thông tin liên quan.
* Danh sách các phòng: Danh sách các phòng học có sẵn, thông tin về sức chứa, loại phòng (lý thuyết, thực hành).
* Phân loại theo phòng học dành cho lớp - môn học: Xác định phòng học phù hợp cho mỗi lớp - môn học, dựa trên số lượng sinh viên và loại học phần.
* Bảng phân công giảng viên giảng dạy cho lớp - môn học: Xác định giảng viên nào sẽ dạy cho mỗi lớp - môn học.

### *2.1.2 Nhắc lại các ràng buộc của bài toán*

Các ràng buộc của bài toán được chia thành hai loại: ràng buộc cứng và ràng buộc mềm, cụ thể như sau:

*2.1.2.1 Ràng buộc cứng*

* Ràng buộc cứng về thời gian

Đảm bảo thời lượng mỗi lớp học nằm trong một ca học

Đảm bảo sắp xếp các tiết học liên tục cho một môn học để tránh bị lủng tiết giữa các tiết học

* Ràng buộc cứng về phòng học

Đảm bảo mỗi phòng học trong một khung giờ chỉ được sử dụng bởi một lớp học

Đảm bảo sắp xếp phòng học phù hợp dành cho lớp lý thuyết và thực hành

* Ràng buộc cứng về giảng viên

Đảm bảo sắp xếp thời khóa biểu phù hợp với giờ dạy mà giảng viên cung cấp

Đảm bảo các lớp mà giảng viên đảm nhiệm không bị trùng lịch với nhau

* Ràng buộc cứng về địa điểm

Đảm bảo sắp xếp thời gian dạy hai lớp khác cơ sở hợp lý sao cho giảng viên có đủ thời gian để di chuyển

*2.1.2.2 Ràng buộc mềm*

* Ràng buộc mềm về phòng học

Đảm bảo sử dụng phòng học một cách tối ưu giữa các lớp với nhau

Đảm bảo xác định phòng học phù hợp với số lượng sinh viên đăng ký môn học

* Ràng buộc mềm về giảng viên

Đảm bảo phù hợp với giảng viên có lịch nghỉ, lịch bận, xa nhà

Đảm bảo các lớp học trong cùng một ngày của giảng viên không cách quá xa nhau

Có thể xử lý được các yêu cầu đặc biệt từ giảng viên về việc sắp xếp thời khóa biểu

* Ràng buộc mềm về địa điểm

Đảm bảo sắp xếp địa điểm các lớp dành cho sinh viên năm 1 và năm 2 ở cơ sở 2 (Linh Trung)

### *2.1.3 Mục tiêu của bài toán*

Mục tiêu của bài toán là đáp ứng được tất cả các ràng buộc chính và cố gắng đáp ứng tối đa các ràng buộc mềm.

## 2.2 Các kỹ thuật/thuật toán liên quan

### 2.2.1 Giới thiệu thuật toán di truyền Genetic Algorithm

#### 2.2.1.1 Khái niệm Di truyền và Tiến hóa

Di truyền là quá trình truyền các đặc điểm gen từ cha mẹ sang con cái thông qua việc chuyển giao các đoạn gen trong quá trình sinh sản.

Tiến hóa là quá trình thay đổi các đặc điểm di truyền của các quần thể sinh vật qua các thế hệ. Sự thay đổi này có thể dẫn đến sự xuất hiện của các loài mới và sự biến đổi trong đa dạng sinh học trên Trái Đất. Tiến hóa được điều khiển bởi các cơ chế chính sau:

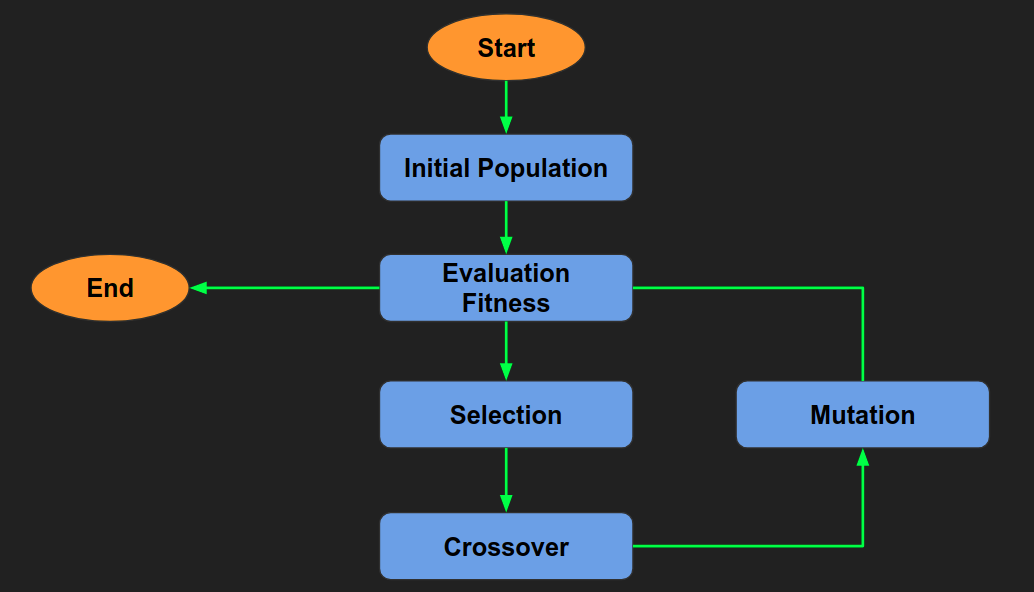
* Chọn lọc tự nhiên: được nghiên cứu bởi Charles Darwin, chọn lọc tự nhiên là quá trình trong đó các cá thể có đặc điểm phù hợp với môi trường sống có nhiều khả năng sống sót và sinh sản hơn, sau đó truyền lại các đặc điểm này cho thế hệ sau.
* Đột biến: đột biến là những thay đổi ngẫu nhiên trong vật liệu di truyền (DNA). Chúng có thể tạo ra các alen mới, cung cấp nguyên liệu thô cho tiến hóa.
* Di cư: di cư là sự trao đổi gen giữa các quần thể thông qua di chuyển của các cá thể hoặc giao tử. Điều này có thể làm thay đổi tần số alen trong quần thể.
* Giao phối chọn lọc: giao phối không ngẫu nhiên xảy ra khi cá thể chọn bạn tình dựa trên các đặc điểm nhất định, dẫn đến sự thay đổi tần số alen.
* Trôi dạt di truyền: trôi dạt di truyền là sự thay đổi ngẫu nhiên trong tần số alen của một quần thể nhỏ do sự ngẫu nhiên của các sự kiện sinh sản.

#### 2.2.1.2 Thuật toán di truyền Genetic Algorithm

Thuật toán di truyền (Genetic Algorithm) là phương pháp tìm kiếm tối ưu ngẫu nhiên bằng cách mô phỏng quá trình tiến hóa của sinh vật trong tự nhiên. Thuật toán này được sử dụng để giải quyết các bài toán tối ưu hóa phức tạp, đặc biệt là những bài toán có không gian tìm kiếm lớn và không thể sử dụng những thuật toán thông thường.

Các bước cơ bản của một thuật toán di truyền như sau:

* Khởi tạo quần thể (Initial population): tạo một quần thể ban đầu ngẫu nhiên, mỗi cá thể đại diện cho một giải pháp của bài toán. Số lượng cá thể của quần thể ban đầu phụ thuộc vào độ lớn của bài toán và nguồn lực tính toán
* Đánh giá (Evaluation fitness): mỗi cá thể được đánh giá bằng một hàm thích nghi (fitness function) để xác định độ tốt của nó trong việc giải quyết vấn đề
* Chọn lọc (Selection): chọn các cá thể từ quần thể hiện tại để sinh sản thế hệ tiếp theo. Có thể chọn theo một trong những phương pháp sao:
  + Chọn lọc roulette: cá thể có giá trị thích hợp cao hơn có nhiều khả năng được chọn hơn.
  + Chọn lọc giải đấu: chia nhỏ quần thể thành các nhóm nhỏ và chọn cá thể có giá trị thích hợp cao nhất trong mỗi nhóm.
  + Chọn lọc ngẫu nhiên: chọn cá thể ngẫu nhiên từ quần thể.
* Lai ghép (Crossover): ghép nối các cặp cá thể được chọn ngẫu nhiên từ tập hợp con để tạo ra con lai.
* Đột biến (Mutation): thay đổi ngẫu nhiên một phần nhỏ của cá thể con để tạo ra sự đa dạng và tránh bị kẹt trong các giải pháp không tối ưu.
* Lặp lại (Loop): Lặp lại các bước 2 đến 5 cho đến khi đạt được một trong các điều kiện sau:
  + Tìm thấy giải pháp tối ưu hoặc gần tối ưu.
  + Đạt đến số lượng thế hệ tối đa.
  + Tiêu chuẩn dừng khác được đáp ứng.



### 2.2.1 Mô hình hóa bài toán (In progress)

#### 2.2.1.1 Biểu diễn cá thể

#### 2.2.1.2 Khởi tạo quần thể

#### 2.2.1.3 Đánh giá

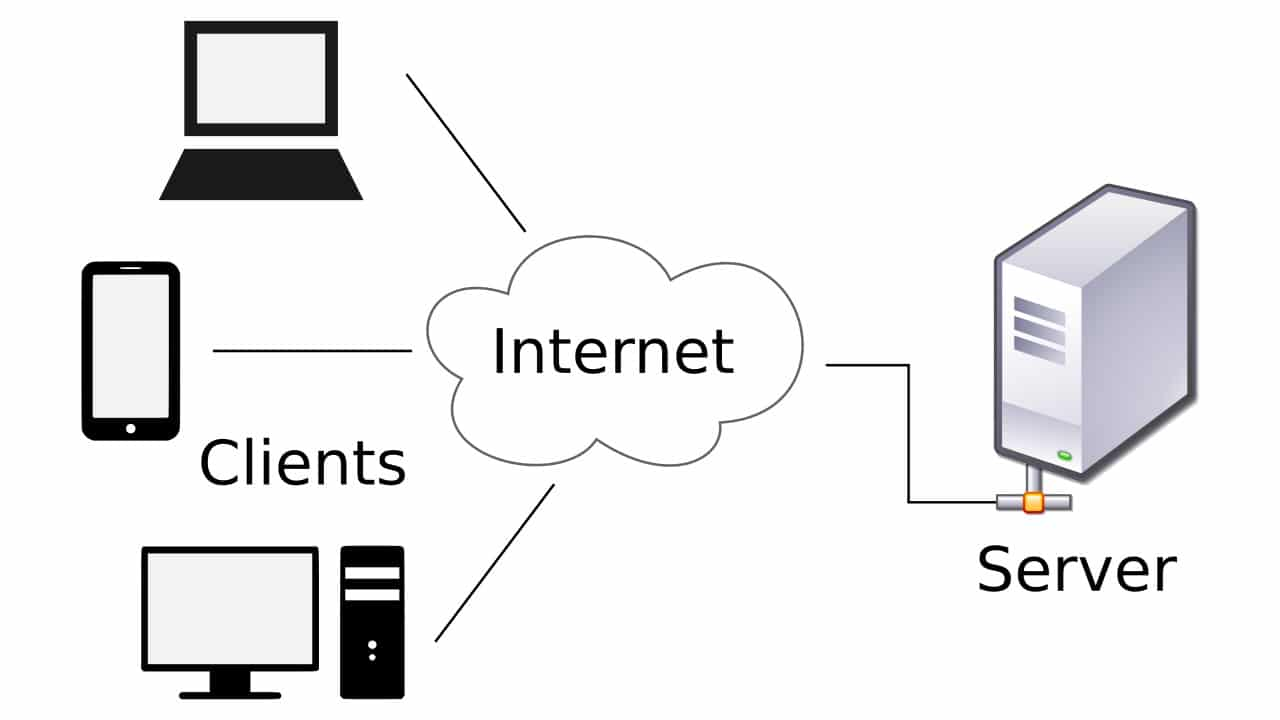
## 2.3 Kiến trúc hệ thống

### 2.3.1 Kiến trúc client-server

Client-server là một kiến trúc phổ biến trong việc phát triển phần mềm. Với kiến trúc này, phần mềm được chia thành hai phần: client và server. Mỗi thành phần đều có những đặc điểm và vai trò riêng, góp phần tạo nên ứng dụng hoàn chỉnh.

Client có thể hiểu là phần mềm chạy trên máy tính hoặc các thiết bị của người dùng cuối, nó có thể là một ứng dụng trên máy tính, trình duyệt web, ứng dụng di động hoặc các thiết bị có thể kết nối mạng. Client được sử dụng để tương tác với người dùng và gửi yêu cầu của người dùng đến máy chủ (server).

Server là phần mềm chạy trên máy chủ của bên phát triển ứng dụng. Server có nhiệm vụ tiếp nhận yêu cầu, thông tin từ client và xử lý chúng, sau đó trả về kết quả thích hợp đối với từng yêu cầu của client. Server có thể là máy chủ ứng dụng, máy chủ cơ sở dữ liệu, máy chủ web, hoặc các loại máy chủ khác.



Do cấu trúc, client và server chạy trên những thiết bị và máy chủ khác nhau, kết nối thông qua mạng, làm cho kiến trúc này có tính phân tán về mặt vật lý và địa lý. Điều này lại là một ưu điểm của mô hình client-server, việc cho phép phân tán tài nguyên giúp giảm áp lực cho các thành phần trong hệ thống và tăng khả năng mở rộng, phát triển hệ thống.

Đa số logic xử lý được chia ra giữa client và server. Client sẽ chịu trách nhiệm xử lý giao diện người dùng và xử lý sự kiện người dùng, server sẽ xử lý các dữ liệu, logic kinh doanh (logic business) và bảo mật. Nhờ sự phân chia rõ ràng như vậy, việc quản lý, bảo trì và nâng cấp hệ thống trở nên dễ dàng hơn so với các tính chất phức tạp.

Bên cạnh những lợi ích mang lại, mô hình client-server cũng còn những điểm bất cập. Sự phụ thuộc vào mạng trong kiến trúc client-server có thể gây ra nhiều vấn đề như giảm hiệu suất ứng dụng nếu như hệ thống mạng hoạt động không ổn định. Ngoài ra, đối với các ứng dụng với một máy chủ duy nhất, nếu như máy chủ gặp sự cố, nó sẽ gây ra sự cố cho toàn bộ hệ thống.

## 2.4 Các công nghệ sử dụng

### 2.4.1 Angular framework

#### 2.4.1.1 Sơ lược về front-end framework

Front-end framework là một bộ công cụ và thư viện được sử dụng để xây dựng và duy trì giao diện người dùng một cách nhanh chóng và dễ dàng thông qua cách cung cấp các thành phần và tính năng chuẩn để phát triển ứng dụng web. Bên cạnh đó, các framework front-end cũng cung cấp các thành phần và mã JavaScript có thể tái sử dụng mà các nhà phát triển có thể tích hợp vào ứng dụng của mình, giúp tăng hiệu suất và tiết kiệm thời gian phát triển.

#### 2.4.1.2 Angular framework

Angular là một front-end framework mã nguồn mở, được phát triển bởi Google và được sử dụng để xây dựng một cách linh hoạt và hiệu quả các ứng dụng web SPA (Single-page applications) - ứng dụng web động mà tất cả nội dung và tương tác người dùng được hiển thị trên một trang web duy nhất.

Angular được biết đến rộng rãi thông qua một số đặc điểm nổi bật như

* Tuân thủ mô hình MVVM (Model-View-Viewmodel) giúp tách biệt logic dữ liệu (Model), hiển thị (View) và logic điều khiển (ViewModel), làm cho việc quản lý và bảo trì mã nguồn trở nên dễ dàng.
* Cung cấp tính năng two-way data binding, cho phép tự đồng bộ hóa dữ liệu giữa model và view mà không cần phải viết code tùy chỉnh, nhờ đó làm giảm khối lượng mã nguồn và tăng khả năng, năng suất trong việc phát triển ứng dụng.
* Angular sử dụng dependency injection để quản lý các phụ thuộc và cung cấp các thành phần dễ kiểm soát, tái sử dụng và kiểm thử.
* Cung cấp bộ router thích hợp để quản lý và điều hướng trang, tạo ra các ứng dụng đa trang (SPA) với URL thân thiện với người dùng
* Các công cụ đi kèm của Angular như Angular CLI hoạt động mạnh mẽ, giúp tạo, triển khai và kiểm thử ứng dụng dễ dàng hơn.

#### 2.4.1.3 Sử dụng Angular framework trong phần mềm sắp xếp thời khóa biểu tự động

Đối với phần mềm sắp xếp thời khóa biểu tự động, Angular được sử dụng cho phần front end để xây dựng giao diện người dùng một cách hiệu quả. Cụ thể như sau:

* Giao diện người dùng động: Angular cho phép tạo ra giao diện người dùng động và phản ứng, giúp người dùng tương tác với ứng dụng một cách tự nhiên và dễ dàng hơn.
* Quản lý trạng thái ứng dụng: các công cụ mạnh mẽ của Angular trong việc trạng thái quản lý ứng dụng như Observables và RxJS góp phần vào quá trình xử lý dữ liệu và sự kiện theo thời gian thực.
* Routing: hệ thống routing mạnh mẽ cho phép người dùng quản lý và điều hướng qua lại các trang trong ứng dụng một cách dễ dàng. Điều này sẽ hữu ích khi người dùng muốn xem thời khóa biểu của các kỳ và các năm học khác nhau hoặc…
* Tương tác với backend: Angular có thể gửi các yêu cầu HTTP đến backend để nhận dữ liệu mới, thêm hoặc sửa đổi thông tin thời khóa biểu dựa trên hành động của người dùng
* Thiết kế đẹp và linh hoạt: nhờ sự kết hợp với các thư viện của CSS như Bootstrap hoặc Angular Material, tạo ra giao diện người dùng đẹp mắt và linh hoạt

### 2.4.2 Express js

#### 2.4.2.1 Sơ lược về Express js

Express.js là một framework web tiện lợi, linh hoạt và tối giản cho Node.js, giúp xây dựng ứng dụng web và API một cách nhanh chóng và dễ dàng. Bên cạnh đó, Express.js cung cấp các phương thức và công cụ hữu ích cho việc xử lý yêu cầu HTTP và tạo các endpoint API.

Express.js có một số đặc điểm chính như sau:

* Middleware và router: bằng cách sử dụng middleware và router, Express.js giúp cho việc tổ chức chức năng của ứng dụng dễ dàng hơn. Các hàm middleware cho phép người dùng xử lý các nhiệm vụ như xác thực, ghi lại log và xử lý lỗi. Router đảm bảo rằng các yêu cầu gửi đến được chuyển hướng đến các bộ xử lý thích hợp.
* Thiết kế tối giản: Express.js tuân theo triết lý thiết kế đơn giản. Sự đơn giản này cho phép người dùng thiết lập máy chủ một cách nhanh chóng, xác định các route và xử lý yêu cầu HTTP một cách hiệu quả.
* Tính linh hoạt: Express.js không áp đặt một kiến trúc ứng dụng nhất định. Người dùng có thể tự do trong cách tổ chức mã của mình. Dù là xây dựng một RESTful API hay một ứng dụng web với đầy đủ chức năng, Express.js vẫn có thể đáp ứng được.
* Khả năng mở rộng: Express.js được thiết kế để xử lý một lượng lớn yêu cầu cùng một lúc mà không làm giảm hiệu suất của ứng dụng. Đây là một phần quan trọng giúp Express.js trở nên linh hoạt và mạnh mẽ trong việc xây dựng các ứng dụng web quy mô lớn.

#### 2.4.2.2 Express js trong phần mềm tạo thời khóa biểu tự động

Trong phần mềm xây dựng thời khóa biểu tự động này, Express.js được sử dụng để phát triển phần backend của phần mềm, quản lý việc xử lý yêu cầu từ phía client và cung cấp dữ liệu cần thiết cho phần front end. Cụ thể như sau:

* Routing: Express.js cho phép định tuyến các yêu cầu HTTP đến các hàm xử lý tương ứng, hỗ trợ việc xử lý yêu cầu từ phía client như lấy danh sách giảng viên, thời khóa biểu của một kỳ hoặc thêm/sửa/xóa các thông tin liên quan đến lớp học.
* Xử lý dữ liệu đầu vào: khi có yêu cầu gửi đến backend, Express.js có thể xử lý dữ liệu đầu vào từ các yêu cầu này, kiểm tra tính hợp lệ và xử lý sao cho phù hợp với logic của ứng dụng.
* Kết nối cơ sở dữ liệu: Express.js có thể dùng để thiết lập kết nối với cơ sở dữ liệu (ở đây sử dụng MySQL) để lấy các dữ liệu như thông tin giảng viên, môn học, phòng học và một số dữ liệu liên quan khác.
* Xử lý tác vụ nền: Express.js sử dụng các thư viện như Agenda hoặc Bull để xác định và xử lý các tác vụ nền, những tác vụ này có thể được lên lịch hoặc kích hoạt dựa trên các sự kiện cụ thể. Trong phần mềm tạo thời khóa biểu tự động, có thể xem xét tạo ra chức năng gửi thông báo cho giảng viên khi thời khóa biểu có thay đổi.
* Bảo mật: Express.js cung cấp các công cụ và middleware để xử lý bảo mật, bao gồm xác thực người dùng, quản lý phiên và xác thực JWT (JSON Web Tokens) để bảo vệ các API và dữ liệu quan trọng.

Bên cạnh đó, việc kết hợp với Angular cho phần front-end giúp Express.js trở thành một công cụ mạnh mẽ giúp xây dựng phần mềm tạo thời khóa biểu tự động linh hoạt và hiệu quả.

### 2.4.3 Hệ quản trị cơ sở dữ liệu MySQL

#### 2.4.3.1 Sơ lược về hệ quản trị cơ sở dữ liệu MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở được phát triển bởi MySQL AB, hiện thuộc sở hữu của Oracle Corporation. MySQL cho phép lưu trữ dữ liệu trong một định dạng có cấu trúc sử dụng các hàng và cột, hỗ trợ người dùng tạo, quản lý và thao tác với các cơ sở dữ liệu. MySQL nổi tiếng với tính đáng tin cậy, khả năng mở rộng và dễ sử dụng.

MySQL là lựa chọn phổ biến để quản lý cơ sở dữ liệu vì một số đặc điểm nổi bật dưới đây:

Mã nguồn mở: MySQL là phần mềm mã nguồn mở, tức là nó cho phép người dùng tự do tải về, sử dụng và sửa đổi mã nguồn. Ngoài ra, phần mềm này cũng có một cộng đồng lớn các nhà phát triển đóng góp cải tiến.

Quan hệ: MySQL tuân theo mô hình cơ sở dữ liệu quan hệ, nghĩa là nó cho phép người dùng tổ chức dữ liệu dưới dạng bảng, liên quan đến nhau thông qua các khóa, giúp lưu trữ và xuất dữ liệu hiệu quả.

Hiệu suất và khả năng mở rộng: MySQL được thiết kế để xử lý khối lượng lớn dữ liệu và nhiều người dùng đồng thời một các hiệu quả. Ngoài ra, MySQL có thể mở rộng theo cả chiều dọc và chiều ngang để đáp ứng được lượng dữ liệu và người dùng ngày càng tăng.

Tính tương thích: MySQL được hỗ trợ rộng rãi bởi nhiều ngôn ngữ lập trình. Nó cung cấp các kết nối và API cho các ngôn ngữ phổ biến như PHP, Python, Java,...giúp dễ dàng tích hợp với hệ thống phần mềm hiện có.

Bảo mật: MySQL cung cấp các tính năng bảo mật mạnh mẽ để bảo vệ dữ liệu của người dùng bao gồm kiểm soát truy cập và mã hóa dữ liệu để bảo vệ thông tin. Với cấu hình thích hợp, người dùng có thể yên tâm rằng chỉ những người dùng được ủy quyền mới có quyền truy cập vào thông tin nhạy cảm.

Tính ứng dụng: MySQL thường được sử dụng làm cơ sở dữ liệu cho các ứng dụng web, trong các nền tảng thương mại điện tử và các ứng dụng doanh nghiệp.

#### 2.4.3.2 Sử dụng MySQL trong phần mềm sắp xếp thời khóa biểu tự động

Trong ứng dụng tạo thời khóa biểu tự động này, MySQL có thể được sử dụng để:

* Lưu trữ dữ liệu: lưu trữ một số thông tin như sinh viên, giảng viên, môn học, phòng học, và lịch trình lớp học.
* Truy vấn dữ liệu: thực hiện các truy vấn để lấy dữ liệu cần thiết cho việc tạo và hiển thị thời khóa biểu.
* Quản lý người dùng: quản lý thông tin người dùng và quyền truy cập của họ vào hệ thống.

### 2.4.4 Socket IO

#### 2.4.4.1 Sơ lược về Socket IO

Socket.IO là một thư viện JavaScript phổ biến cho phép truyền thông hai chiều, thời gian thực giữa máy chủ và các client (người dùng). Nó hỗ trợ cả phía client (trình duyệt) và phía server (Node.js). Socket.IO thường được sử dụng để phát triển các ứng dụng yêu cầu cập nhật dữ liệu theo thời gian thực, như chat trực tuyến, thông báo, trò chơi trực tuyến, và các ứng dụng tương tác cao khác.

Socket.IO có một số đặc điểm chính như sau:

Độ tin cậy: Socket.IO có thể thiết lập kết nối ngay cả khi có sự hiện diện của các proxy, bộ cân bằng tải, tường lửa và phần mềm chống virus.

Hỗ trợ tự động kết nối lại: trừ khi có được đề cập rõ ràng trong mã nguồn, nếu không, Socket.IO sẽ tự động cố gắng kết nối lại cho đến khi server hoạt động trở lại.

Phát hiện ngắt kết nối: Socket.IO có khả năng phát hiện khi nào client hoặc server bị ngắt kết nối và thực hiện các hành động cần thiết.

Hỗ trợ đa kênh: Socket.IO cho phép có nhiều kênh giao tiếp trên cùng một kết nối nền.

Hỗ trợ truyền tải nhị phân: Socket.IO cho phép truyền tải dữ liệu nhị phân bằng cách sử dụng các phương thức như send() hoặc emit() để gửi dữ liệu. Điều này cho phép người dùng truyền tải các đối tượng nhị phân như ArrayBuffer hoặc Blobs giữa client và server một cách hiệu quả.

#### 2.4.4.2 Sử dụng Socket IO trong phần mềm sắp xếp thời khóa biểu tự động

Trong ứng dụng tạo thời khóa biểu tự động, Socket.IO có thể được sử dụng để:

* Cập nhật thời khóa biểu thời gian thực: khi có thay đổi trong thời khóa biểu (ví dụ: thêm/sửa/xóa lớp học), các client đang kết nối sẽ nhận được cập nhật ngay lập tức.
* Thông báo thời gian thực: gửi thông báo đến giảng viên khi có sự kiện quan trọng như thay đổi phòng học, thay đổi thời gian lớp học.
* Tương tác người dùng: Cho phép nhiều người dùng tương tác với hệ thống cùng một lúc.

### 2.4.5 JWT (JSON Web Token)

#### 2.4.5.1 Sơ lược về JWT (JSON Web Token)

JWT (JSON Web Token) là một tiêu chuẩn mở (RFC 7519) dùng để chia sẻ thông tin an toàn giữa hai bên (thường là giữa máy chủ và trình duyệt) dưới dạng một đối tượng JSON. Nó được sử dụng rộng rãi trong các hệ thống xác thực để đảm bảo rằng người dùng đã được xác minh và có quyền truy cập vào các tài nguyên cụ thể.

Cấu trúc của JWT gồm ba phần chính, được phân tách bằng dấu chấm (.):

* Header: Phần này chứa thông tin về loại token và thuật toán mã hóa được sử dụng.
* Payload: Phần này chứa dữ liệu (claims), tức là các thông tin về người dùng hoặc dữ liệu cần truyền. Dữ liệu này có thể bao gồm thông tin như ID người dùng, tên, quyền hạn, và thời gian hết hạn của token.
* Signature: Phần này dùng để xác thực tính toàn vẹn của token. Nó được tạo ra bằng cách mã hóa header và payload với một khóa bí mật hoặc một cặp khóa công khai/riêng tư.

Cách JWT Hoạt Động:

* Tạo Token: khi người dùng đăng nhập vào hệ thống, server sẽ tạo một JWT chứa thông tin người dùng và gửi lại cho client.
* Lưu Trữ Token: client lưu trữ token này (thường trong localStorage hoặc sessionStorage).
* Gửi Token: với mỗi yêu cầu từ client đến server, client gửi JWT trong header của yêu cầu HTTP (thường là header Authorization: Bearer <token>).
* Xác Thực Token: server nhận token, kiểm tra chữ ký và giải mã token để xác thực người dùng. Nếu token hợp lệ, server sẽ xử lý yêu cầu.

Ngoài ra, JWT được biết đến rộng rãi với những ưu điểm sau

* Bảo mật: JWT sử dụng chữ ký để đảm bảo tính toàn vẹn và tính xác thực của token. Nếu bất kỳ phần nào của token bị thay đổi, chữ ký sẽ không hợp lệ.
* Tự chứa: JWT chứa tất cả thông tin cần thiết để xác thực người dùng, giúp server không cần duy trì trạng thái phiên người dùng.
* Nhẹ: JWT là các chuỗi văn bản nhỏ gọn và có thể dễ dàng gửi qua HTTP headers.
* Khả năng mở rộng: Với JWT, bạn có thể dễ dàng mở rộng ứng dụng bởi vì server không cần lưu trữ thông tin phiên người dùng, tất cả được mã hóa trong token.

#### 2.4.5.2 Sử dụng JWT trong phần mềm xây dựng thời khóa biểu tự động

Trong ứng dụng tạo thời khóa biểu tự động, JWT có thể được sử dụng để:

* Xác Thực Người Dùng: khi người dùng đăng nhập, server sẽ tạo ra một JWT và gửi lại cho client. Mỗi yêu cầu tiếp theo từ client sẽ đính kèm JWT để xác thực.
* Ủy Quyền: dựa trên thông tin trong JWT, server có thể kiểm tra quyền hạn của người dùng (ví dụ: giảng viên, quản trị viên) và cung cấp quyền truy cập tương ứng.
* Bảo Mật API: Các API yêu cầu xác thực sẽ kiểm tra JWT để đảm bảo chỉ những yêu cầu hợp lệ mới được xử lý.

# **CHƯƠNG 3 - PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## 3.1 Đặc tả hệ thống

### 3.1.1 Nhiệm vụ

### 3.1.2 Quy trình nghiệp vụ

## 3.2 Phân tích hệ thống

### 3.2.1 Yêu cầu ứng dụng

### 3.2.2 Yêu cầu người dùng

## 3.3 Thiết kế hệ thống

### 3.3.1 Mô hình cơ sở dữ liệu

### 3.3.2 Sơ đồ Use case (Use case diagram)

### 3.3.3 Sơ đồ tuần tự (Sequence diagram)

### 3.3.4 Mô hình phân rã chức năng (BFD)

### 3.3.5 Mô hình dòng dữ liệu (DFD)

# **CHƯƠNG 4 - HIỆN THỰC CHƯƠNG TRÌNH**

# CHƯƠNG 5 - TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

# TÀI LIỆU THAM KHẢO

# KẾT LUẬN

# 