

DANH SÁCH ĐỀ TÀI LẬP TRÌNH MẠNG

ĐỀ TÀI 1: XÂY DỰNG CHƯƠNG TRÌNH CHAT

Yêu cầu về chức năng phía client (phải có GUI):

- Đăng ký tài khoản mới, đăng nhập, cập nhật thông tin tài khoản. Thông tin tối thiểu cho mỗi tài khoản gồm có các trường: username (chính là địa chỉ email), password (hash), họ tên, giới tính, ngày sinh. Ngoài ra, địa chỉ email lúc đăng ký phải được xác thực bằng OTP (timeout 10 phút) trước khi tài khoản đó được kích hoạt.
- Tạo nhóm chat mới hoặc gia nhập/rời khỏi các nhóm chat có sẵn.
- Gửi tin nhắn 1-1 hoặc 1-nhóm.
- Gửi file đính kèm (dung lượng dưới 1MB), gửi sticker.
- Chặn nhận tin nhắn từ 1 người dùng hoặc 1 nhóm.
- Xác nhận tin nhắn đã gửi/nhận, đã xem.
- Friend list: lưu giữ danh sách và cho biết trạng thái (online hoặc offline) của những người mà user đã từng chat, có thông báo khi bất cứ user nào trong friend list thay đổi trạng thái.
- Chức năng quản lý và lưu giữ tin nhắn: toàn bộ tin nhắn của người dùng được lưu trữ trên server, khi client đăng nhập, những tin nhắn gần đây nhất sẽ được đẩy về giao diện phía client.

Yêu cầu về chức năng phía server (không cần GUI):

- Thống kê được tổng số user, các user đang online.
- Ghi log khi user login/logout hoặc thực hiện các chức năng: đăng ký tài khoản; tạo nhóm.
- Block user bằng lệnh (hoặc giao diện đồ họa nếu có).
- Broadcast tin nhắn từ hệ thống đến toàn bộ user.

Yêu cầu chung:

- Mã hóa nội dung tin nhắn giữa client – server. Phải sử dụng key khác nhau cho các client.
- Các client khi chấm bài phải chạy trên các máy tính khác nhau.

ĐỀ TÀI 2: XÂY DỰNG CHƯƠNG TRÌNH CHƠI CỜ CARO

Yêu cầu về chức năng phía client (phải có GUI):

- Đăng ký tài khoản mới, đăng nhập, cập nhật thông tin tài khoản tương tự đề tài 1 (loại trừ phần xác thực email).
- Chức năng chơi game:
 - ✓ Cứ một cặp user đăng nhập sẽ được ghép đôi thành 1 phiên chơi sau khi có xác nhận đồng ý từ cả hai user.
 - ✓ Nếu 1 user đăng nhập vào trường hợp lẻ, user sẽ được đưa vào hàng đợi và có thể chọn xem 1 phiên chơi của một cặp bất kỳ. User này sẽ tự động được ghép đôi tạo thành 1 phiên chơi mới khi có một user nữa đăng nhập (cần có xác nhận đồng ý từ cả hai user). Trong trường hợp một trong hai user không đồng ý ghép đôi, cả hai sẽ được đưa lại vào hàng đợi và user từ chối ghép đôi sẽ bị trừ 1 điểm thành tích.
 - ✓ Mỗi phiên chơi giới hạn trong 10 phút, nếu hết thời gian thì xem như 2 user hòa. Thời gian giới hạn cho mỗi nước đi là 30 giây, user nào không đi nước mới trong thời gian giới hạn sẽ bị xử thua. Điểm phiên chơi được tính theo tỉ lệ:
 - Thắng: cộng 3 điểm.
 - Hòa: cộng 1 điểm.
 - Thua: trừ 1 điểm.
- Chat nhóm trong khi chơi: giữa người chơi 1 – người chơi 2 – người xem.
- Xem thành tích cá nhân: số trận thắng/thua, tỉ lệ thắng, chuỗi trận thắng/thua dài nhất, điểm thành tích (nhóm tự xác định công thức tính điểm thành tích sao cho hợp lý).
- Xếp hạng: liệt kê thứ hạng tất cả các user đã tham gia dựa trên điểm thành tích; chuỗi trận thắng; tỉ lệ thắng (cần quan tâm đến trường hợp user mới tham gia vài trận).

Yêu cầu về chức năng phía server (không cần GUI):

- Thống kê được tổng số user, số user đang online.
- Block user khỏi hệ thống.
- Ghi log các trận đấu (các user tham gia, thời gian chơi, thời gian bắt đầu,) và query được các thông tin cơ bản (user thắng nhiều nhất, trận đấu ngắn nhất,...).

Yêu cầu chung:

- Mã hóa nội dung trao đổi giữa client – server. Phải sử dụng key khác nhau cho các client.
- Các client phải chạy trên các máy tính khác nhau.

ĐỀ TÀI 3: XÂY DỰNG CHƯƠNG TRÌNH NHẬN DIỆN KHUÔN MẶT/ĐỐI TƯỢNG

Yêu cầu về chức năng phía client (phải có GUI):

- Đối với chức năng nhận diện khuôn mặt:
 - ✓ Gửi dữ liệu ảnh về server: client có thể gửi dữ liệu khuôn mặt của 1 đối tượng về server bằng 1 trong 2 phương pháp dưới đây (nhóm phải cài đặt cả 2 phương pháp):
 - Sử dụng webcam của client
 - Sử dụng ảnh tĩnh có sẵn trên thiết bị.
 - ✓ Nhận kết quả phản hồi từ server: in ra đối tượng ứng với khuôn mặt nhận diện được. Trường hợp nhận diện được/không nhận diện được, server cần trả về kết quả tỉ lệ % hoặc thông số cho biết mức độ khớp giữa ảnh đầu vào và các đối tượng trên CSDL. CSDL trên server phải có tối thiểu 5 đối tượng.
 - ✓ Bổ sung khuôn mặt: vẫn sử dụng hai phương pháp truyền bằng hình ảnh tĩnh hoặc lấy ảnh trực tiếp từ webcam.
- Đối với chức năng nhận diện đối tượng (object):
 - ✓ Client chỉ gửi dữ liệu ảnh tĩnh về server, server phân tích ảnh và trả kết quả các đối tượng xuất hiện trong ảnh cho client.

Yêu cầu về chức năng phía server (GUI do SV tự quyết định):

- Server lưu dữ liệu mẫu của một số người dùng (tối thiểu là 5 người khác nhau, chưa kể các người dùng mới sẽ được thêm và test trong quá trình chấm bài), sử dụng cho việc nhận diện.
- Nhóm SV có thể sử dụng các thư viện/API/ngôn ngữ để giải quyết vấn đề phía server. Tuy nhiên, giao tiếp giữa client và server bắt buộc phải thông qua Java socket.

Yêu cầu chung:

- Mã hóa nội dung trao đổi giữa client – server. Phải sử dụng key khác nhau cho các client.
- Các client phải chạy trên các máy tính khác nhau.

Một số từ khóa tham khảo:

- OpenCV, Tensorflow
- <https://rapidapi.com/blog/top-facial-recognition-apis/>

ĐỀ TÀI 4: XÂY DỰNG CHƯƠNG TRÌNH CHIA SẺ ĐỀ VÀ THI TRẮC NGHIỆM

Yêu cầu về chức năng phía client (phải có GUI):

- Đăng ký tài khoản mới, đăng nhập, cập nhật thông tin tài khoản: user có thể đăng ký tài khoản mới và đăng nhập từ phía client. Thông tin tối thiểu cho mỗi tài khoản gồm có: username (chính là địa chỉ email), password (hash), họ tên, giới tính, ngày sinh. Ngoài ra, địa chỉ email lúc đăng ký phải được xác thực bằng OTP (timeout 10 phút) trước khi tài khoản đó được kích hoạt.
- Tạo đề thi mới: client sau khi đăng nhập có thể bổ sung bộ đề mới vào ngân hàng đề để các người dùng khác thi thử. Mỗi bộ đề cần có các thông tin tối thiểu:
 - ✓ Số lượng câu hỏi trong bộ đề do người tạo đề quyết định (nên gợi ý sẵn số lượng câu)
 - ✓ Mỗi câu hỏi luôn có 4 đáp án và có duy nhất 1 đáp án đúng.
 - ✓ Điểm mỗi bộ đề là 10 và sẽ chia đều cho số câu hỏi.
 - ✓ Thời gian thi do người tạo đề quyết định (nên gợi ý sẵn các mức thời gian)
 - ✓ Người tạo đề được phép chỉnh sửa bộ đề thi chưa có ai thi.
- Chức năng thi: client sau khi đăng nhập sẽ thấy tất cả các bộ đề đã được những người dùng khác (kể cả mình) tạo. Khi chọn 1 bộ đề để thi, câu hỏi được phân phối ngẫu nhiên thứ tự câu và cả đáp án. Sau khi hoàn tất mỗi câu, kết quả đúng/sai sẽ được báo về client. Khi hoàn tất toàn bài thi hoặc hết thời gian làm bài, kết quả tổng hợp kèm xếp hạng sẽ được gửi về client.
- Chức năng thống kê: client xem được thống kê sơ bộ về các đề thi đang có trên hệ thống (mỗi đề có bao nhiêu người đã làm, điểm số trung bình, điểm cao nhất, ...)

Yêu cầu về chức năng phía server (không cần GUI):

- Thống kê được số lượng người dùng, người dùng đang online.
- Block người dùng không cho đăng nhập; không cho tạo đề thi; không cho thi;
- Thực hiện các query cơ bản: tổng số đề thi, điểm thi cao nhất/thấp nhất,... (nhóm tự đề xuất thêm)

Yêu cầu chung:

- Mã hóa nội dung tin nhắn giữa client – server. Phải sử dụng key khác nhau cho các client.
- Các client phải chạy trên các máy tính khác nhau.

ĐỀ TÀI 5: XÂY DỰNG CHƯƠNG TRÌNH TRA CỨU KẾT QUẢ HỌC TẬP SINH VIÊN

Yêu cầu về chức năng phía client (phải có GUI):

- Tra cứu điểm theo MSSV:
 - ✓ Client gửi một chuỗi dữ liệu là MSSV đến server. Server phản hồi về:
 - Các thông tin cá nhân tương ứng MSSV: họ tên; phái; ngày sinh; nơi sinh; lớp; ngành
 - Điểm tất cả các học phần tương ứng với MSSV nhận được hoặc gửi thông báo lỗi nếu MSSV đó không tồn tại.
 - ✓ Thống kê sơ lược về quá trình học đến thời điểm hiện tại: điểm trung bình tích lũy (hệ 4 và hệ 10), số tín chỉ tích lũy, tổng số môn đã học (các lần học cải thiện cũng chỉ tính là 1 môn).
- Xếp hạng sinh viên theo điểm trung bình chung:
 - ✓ Client gửi chuỗi dữ liệu MSSV đến server kèm theo tùy chọn muốn biết xếp hạng theo tiêu chí nào: Khóa, Ngành, Khoa hoặc kết hợp các tiêu chí trên.
 - ✓ Server trả về thứ hạng của client kèm danh sách 10 người có thứ hạng cao hơn và thấp hơn client. Đồng thời cho phép client xem điểm chi tiết bất cứ người nào trong danh sách trên.

Yêu cầu về chức năng phía server (không cần GUI):

- Nhận MSSV từ client, kết nối đến thongtindaotao.sgu.edu.vn, gửi và phân tích dữ liệu nhận được để cho ra kết quả như yêu cầu.
- Server ngoài sử dụng ngôn ngữ Java được phép sử dụng thêm các ngôn ngữ/công cụ khác (nếu cần) để trích xuất dữ liệu. Gợi ý: sử dụng API hoặc Jsoup, Okhttp

Yêu cầu chung:

- Mã hóa nội dung tin nhắn giữa client – server. Phải sử dụng key khác nhau cho các client.
- Các client phải chạy trên các máy tính khác nhau.

ĐỀ TÀI 6: ỨNG DỤNG THEO DÕI HỆ THỐNG BẰNG JAVA

Yêu cầu về chức năng phía client giám sát (GUI) –phiên bản admin:

- Xem thông tin hệ thống (CPU, RAM, disk,...) trên tất cả user.
- Theo dõi thông tin hệ thống (mức độ sử dụng CPU, RAM, disk, process,) trên tất cả user.
- Quản lý máy user: có thể kết thúc process trên user, lấy ảnh màn hình (hoặc video theo thời gian thực – chức năng nâng cao), ghi phím + clipboard, logout hoặc shutdown máy user.
- Có thể quản lý/theo dõi một lúc nhiều user (nhiều máy tính), tự động lấy quyền điều khiển khi user bật ứng dụng client.

Yêu cầu về chức năng phía client bị giám sát (GUI do SV quyết định) – phiên bản user:

- Chạy trên máy user nhằm cho phép admin kết nối và điều khiển.

Yêu cầu về chức năng phía server (không cần GUI):

- Đóng vai trò trung gian, phục vụ liên lạc giữa phiên bản client admin và phiên bản client user.

Yêu cầu chung:

- Các client phải chạy trên các máy tính khác nhau.

ĐỀ TÀI 7: XÂY DỰNG AI CHATBOT

Yêu cầu về chức năng phía client (phải có GUI):

- Sử dụng API của Simsimi (hoặc tự chọn lựa một công cụ khác) để xây dựng AI chatbot. Đồng thời với chat tự động, chatbot sẽ trả lời thông tin thời tiết, xác định vị trí IP, quét port khi người dùng đặt câu hỏi, chi tiết như sau:
 - ✓ Tra cứu thời tiết: gửi yêu cầu là tên một tỉnh/thành phố hoặc địa danh (bằng tiếng Việt). Kết quả phản hồi là thời tiết ngày hiện tại và một số ngày/tuần (do SV quyết định) kế tiếp của địa điểm đó.
 - ✓ Xác định vị trí IP: gửi yêu cầu là 1 địa chỉ IP bất kỳ. Kết quả phản hồi là tọa độ/địa điểm tương ứng với địa chỉ IP đó. SV tham khảo kết quả từ <https://www.iplocation.net/> để biết các thông tin cần trả về. Nhóm SV có thể sử dụng API có sẵn hoặc trích xuất kết quả từ một website có công cụ này.
 - ✓ Quét port: gửi yêu cầu là một địa chỉ IP. Kết quả phản hồi là các port đang mở trong giới hạn từ port x đến port y (với x, y là dữ liệu người dùng nhập).
 - ✓ Các chức năng khác nếu có cài đặt sẽ có điểm cộng.

Yêu cầu về chức năng phía server (không cần GUI):

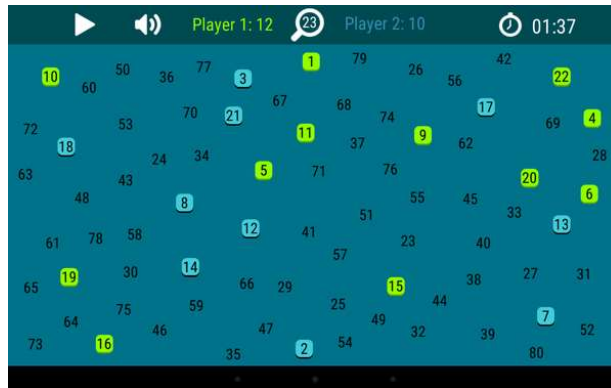
- Nhận các yêu cầu từ client, sử dụng Java socket kết hợp API hoặc (các tools/ngôn ngữ khác) để lấy kết quả trả về cho client. Tất cả xử lý phải nằm ở phía server.

Yêu cầu chung:

- Mã hóa nội dung tin nhắn giữa client – server. Phải sử dụng key khác nhau cho các client.
- Các client phải chạy trên các máy tính khác nhau.

ĐỀ TÀI 8: TRÒ CHƠI TÌM SỐ

Yêu cầu về chức năng phía client (phải có GUI):



Nguồn: Tinhhte.vn

- Cách chơi: SV xem hình minh họa ở trên, game mô phỏng lại trò chơi tìm số trên giấy, các số cần tìm không cần theo thứ tự tăng dần mà theo thứ tự chương trình đưa ra.
- Chức năng đăng ký tài khoản mới, đăng nhập, cập nhật thông tin tài khoản tương tự đề tài 1 (loại trừ phần xác thực email).
- Chức năng chơi game:
 - ✓ Cho phép 2-3 người chơi đồng thời.
 - ✓ Hiển thị 100 số từ 1->100 (có thể cấu hình lại từ phía server).
 - ✓ Thời gian chơi 1 ván là 3 phút (có thể cấu hình lại từ phía server). Nếu hết thời gian chơi mà chưa tìm hết số thì kết quả thắng/thua dựa trên số lượng số mỗi người tìm được.
 - ✓ Số nào đã tìm được thì đánh dấu bằng màu dựa trên người chơi.
 - ✓ Thông báo số cần tìm kế tiếp khi có người đã tìm ra số hiện tại. Các số cần tìm hiển thị ngẫu nhiên, không cần theo thứ tự 1->100.
 - ✓ Các chức năng tăng cường cho người chơi:
 - Tăng điểm: cộng điểm nhiều hơn khi tìm được số may mắn.
 - Ưu tiên: che hết số của những người chơi khác trong vòng 3 giây.
- Chức năng xếp hạng: liệt kê thứ hạng dựa trên thành tích của người dùng. Tra cứu được thứ hạng, tỉ lệ thắng/thua của từng người chơi.

Yêu cầu về chức năng phía server (không cần GUI):

- Xử lý mọi dữ liệu từ client và gửi ngược lại cho client cập nhật.
- Thống kê được số lượng người dùng, người dùng đang online.
- Cấu hình được số lượng số / ván chơi, thời gian / ván chơi.

Yêu cầu chung:

- Các client phải chạy trên các máy tính khác nhau.

ĐỀ TÀI 9: ỨNG DỤNG MINH HỌA HOẠT ĐỘNG CỦA FTP

Yêu cầu về chức năng phía client (phải có GUI):

- Tạo tài khoản mới (yêu cầu về thông tin tài khoản, OTP tương tự đề tài 1).
- Mỗi người dùng đăng nhập được tự động đẩy đến thư mục chứa data của riêng người dùng đó.
- Chức năng upload/download: cho phép upload/download dữ liệu lên/xuống **thư mục có quyền**. Có thể pause/resume quá trình upload/download.
- Có thể đăng nhập tài khoản anonymous để upload/download vào thư mục chung.
- Chia sẻ thư mục/file bất kỳ với quyền chỉ đọc hoặc full quyền cho một người dùng khác. Người dùng đó sẽ nhận được thông báo khi có người chia sẻ file/thư mục cho mình.

Yêu cầu về chức năng phía server (phải có GUI):

- Cho phép/block người dùng download/upload.
- Gán quyền user đối với thư mục. User chỉ được upload/download với các thư mục có quyền.
- Cấu hình dung lượng lưu trữ tối đa của từng user. Kích thước file tối đa upload và download.
- Cho phép/block chức năng anonymous với user bất kỳ.

Yêu cầu chung:

- Các client phải chạy trên các máy tính khác nhau.

ĐỀ TÀI 10: ỨNG DỤNG KIỂM TRA VÀ THỰC THI CODE

Yêu cầu về chức năng phía client (phải có GUI):

- Client gửi tới server code của một trong 4 ngôn ngữ lập trình bất kỳ (ngôn ngữ lập trình do nhóm SV chọn, tối thiểu 4 ngôn ngữ), dữ liệu được gửi qua textbox hoặc dưới dạng upload file code (phải cài đặt cả hai chức năng). Code có thể đúng hoặc sai syntax.
- Client nhận kết quả phản hồi từ server và xuất ra màn hình:
 - ✓ Kết quả thực thi code + code được format đúng chuẩn nếu code đúng syntax.
 - ✓ Lỗi và hướng dẫn sửa lỗi nếu code sai syntax.

VD input:

```
if ('this_is'==/an_example/){of_beautifier();}else{var a=b?(c%d):e[f];}
```

VD output:

```
if ('this_is' == /an_example/) {  
    of_beautifier();  
} else {  
    var a = b ? (c % d) : e[f];  
}
```

Yêu cầu về chức năng phía server (không cần GUI):

- Nhận các yêu cầu từ client, sử dụng Java socket kết hợp API (hoặc các tools/ngôn ngữ khác) để lấy kết quả trả về cho client. Tất cả xử lý phải nằm ở phía server, client chỉ hiển thị kết quả, không kết nối đến các dịch vụ khác cũng như không sử dụng thêm công cụ/ngôn ngữ khác. Hai chức năng chính của server:
 - ✓ Code Checker: kiểm tra lỗi cú pháp, lỗi lập trình của đoạn code được upload lên server và đưa ra phản hồi chi tiết về client: danh sách các lỗi có trong code, vị trí dòng code có lỗi, gợi ý sửa.
 - ✓ Code Formatter: định dạng code đang trình bày lộn xộn thành dạng đúng chuẩn

Yêu cầu chung:

- Mã hóa nội dung tin nhắn giữa client – server. Phải sử dụng key khác nhau cho các client.
- Các client phải chạy trên các máy tính khác nhau.

ĐỀ TÀI 11: ỨNG DỤNG TÌM THÔNG TIN BÀI HÁT

Mô tả chức năng ứng dụng:

- Chức năng 1: ứng dụng cho phép người dùng tra cứu thông tin bài hát dựa vào tên bài hát, tên bài hát có thể là tiếng Việt (có dấu/không dấu) hoặc tiếng Anh.
 - ✓ Trường hợp tên bài hát chính xác, client hiển thị các thông tin:
 - Lời bài hát
 - Tên nhạc sỹ.
 - Một số audio/video (trên Youtube hoặc các trang nghe nhạc) để người dùng có thể nghe trực tiếp trên client.
 - ✓ Trường hợp tên bài hát không chính xác, có lỗi đánh máy: gợi ý hoặc trả về bài hát gần đúng nhất.
- Chức năng 2: tra cứu thông tin ca sĩ Việt Nam dựa vào tên
 - ✓ Trường hợp tên ca sĩ chính xác, client hiển thị các thông tin (có thể lấy từ Wikipedia):
 - Họ tên, ngày sinh, quê quán
 - Thông tin về sự nghiệp
 - Các bài hát, album nổi tiếng
 - Một số audio/video (trên Youtube hoặc các trang nghe nhạc để người dùng có thể nghe trực tiếp trên client)
 - ✓ Trường hợp tên ca sĩ không chính xác, có lỗi đánh máy hoặc không tìm thấy thông tin: báo lỗi

Yêu cầu về chức năng phía server (không cần GUI):

- Nhận dữ liệu từ client, tìm kiếm và xử lý yêu cầu, trả kết quả lại cho client. Toàn bộ các thao tác xử lý nằm ở server, client chỉ hiển thị kết quả do server trả về và chạy audio/video do server gửi link.

Yêu cầu chung:

- Các client phải chạy trên các máy tính khác nhau.
- Mã hóa dữ liệu giữa client – server.

ĐỀ TÀI 12: ỨNG DỤNG TRANH TÀI KIẾN THỨC

Mô tả chức năng ứng dụng:

- Ứng dụng mô phỏng cuộc thi “Học viện Momo”. Mỗi người chơi sau khi đăng ký thành viên (thực hiện các yêu cầu về đăng ký, quản lý tài khoản tương tự đề tài 1), sẽ tham gia các trận đấu trắc nghiệm kiến thức để tích lũy điểm.
- Chức năng thi trắc nghiệm:
 - ✓ Người chơi tự động được ghép cặp với nhau trong 60s (hết thời gian mà chưa tìm được người ghép cặp thì hệ thống đẩy ra khỏi chức năng thi trắc nghiệm, vào lại phòng chờ).
 - ✓ Mỗi cặp chơi phải cùng trả lời 5 câu hỏi, mỗi câu hỏi có đúng 4 đáp án. Thời gian trả lời mỗi câu hỏi tối đa 10s, điểm được tính dựa trên thời gian còn lại. Người chơi không trả lời hoặc trả lời sai thì điểm câu đó = 0.
 - ✓ Khi cả hai người chơi cùng trả lời xong 1 câu, mỗi người chơi sẽ nhìn thấy lựa chọn đáp án của đối thủ và đáp án đúng của câu hỏi.
 - ✓ Các câu hỏi được server đẩy về client đảm bảo sao cho câu hỏi không trùng nhau, cả thứ tự câu hỏi và thứ tự đáp án phải được xáo trộn.
 - ✓ Sau khi kết thúc 5 câu hỏi, người chơi có điểm cao hơn sẽ là người thắng trận.
- Chức năng xếp hạng:
 - ✓ Tra cứu thứ hạng người chơi (nhóm tự xác định công thức)
 - ✓ Chuỗi bàn thắng/thua dài nhất. Tỷ lệ thắng/thua
- Chức năng test IQ bằng hình ảnh:
 - ✓ Chức năng này cho phép người chơi tự xác định IQ dựa trên bộ câu hỏi test bằng hình ảnh. Chức năng này được dùng khi người chơi chờ đợi thêm người dùng vào hệ thống để ghép cặp do đó không được dùng để xếp hạng. Tuy nhiên, điểm IQ sẽ được lưu vào profile của người dùng đó.

Yêu cầu về chức năng phía server (phải có GUI):

- Thêm câu hỏi vào CSDL
- Cấu hình số lượng câu hỏi, thời gian trả lời mỗi câu, điểm trận đấu.
- Thống kê tổng người chơi, người chơi đang online.
- Các query cơ bản: người chơi tham gia nhiều trận, thắng nhiều trận nhất, chuỗi thắng dài nhất.
- Block người dùng khỏi hệ thống.

Yêu cầu chung:

- Các client phải chạy trên các máy tính khác nhau.
- Mã hóa dữ liệu giữa client – server.

ĐỀ TÀI 13: ỨNG DỤNG XẾP THỜI KHÓA BIỂU ĐĂNG KÝ MÔN HỌC

Mô tả chức năng ứng dụng:

- Ứng dụng giải quyết bài toán chọn nhóm lớp các học phần để đăng ký môn học vào đầu học kỳ sao cho học được tất cả các học phần sinh viên mong muốn và không bị trùng thời khóa biểu. Thậm chí tìm thời khóa biểu tối ưu theo tiêu chí (ví dụ đi ít buổi nhất trong tuần).
- Dữ liệu người dùng nhập vào client sẽ là danh sách các học phần muốn đăng ký học trong học kỳ, các học phần được sắp theo thứ tự độ quan trọng giảm dần.
- Server thực hiện các thao tác xử lý:
 - ✓ Kiểm tra các học phần đó có mở ở học kỳ và báo lỗi (nếu có)
 - ✓ Tìm kiếm thời khóa biểu các nhóm của từng học phần (kể cả lý thuyết và thực hành)
 - ✓ Phân tích và xác định một hoặc nhiều phương án đăng ký môn học sao cho học được tất cả các học phần client đã gửi. Trong trường hợp không tìm được bất kỳ phương án nào, server cần loại bỏ bớt 1 học phần ít quan trọng nhất và lặp lại quá trình xử lý cho tới khi tìm được phương án đăng ký học phần.
 - ✓ Gửi trả kết quả về client.
- Client xuất phương án đăng ký ra màn hình dưới dạng thời khóa biểu tuần, thể hiện được lịch học cùng với các thông tin về số tiết, phòng học.

Yêu cầu về chức năng phía server (không cần GUI):

- Nhận yêu cầu từ client, kết nối đến thongtindaotao.sgu.edu.vn để lấy dữ liệu và xử lý.
- Trả kết quả xử lý về client

Yêu cầu chung:

- Các client phải chạy trên các máy tính khác nhau.
- Mã hóa dữ liệu giữa client – server.

ĐỀ TÀI 14: ỨNG DỤNG TÌM ĐƯỜNG ĐI NGẮN NHẤT VÀ LẬP LỊCH CPU

Yêu cầu về chức năng phía client (phải có GUI):

- Chức năng tìm đường đi:
 - ✓ Nhận dữ liệu đầu vào là 1 file text chứa thông tin các đỉnh đồ thị, các cạnh nối giữa các đỉnh và trọng số + thông tin đỉnh source - destination. File input cần có tối thiểu 10 đỉnh và không có đỉnh nào tách rời khỏi đồ thị. Cấu trúc file và định dạng file (sử dụng plaintext, XML hoặc định dạng khác do SV tự quyết định).
 - ✓ Nếu sử dụng file XML hoặc một dạng file có cấu trúc khác: SV cần phải viết thêm chức năng convert từ file plaintext sang file có cấu trúc đó để tạo thuận lợi cho người dùng lúc nhập dữ liệu. Nếu sử dụng file text đơn thuần thì không cần làm chức năng này. Chức năng này có thể thực hiện ngay phía client, không cần gửi dữ liệu lên server.
 - ✓ Nhận output từ server là đường đi ngắn nhất từ source -> destination và vẽ cả đồ thị ra màn hình, riêng đường đi từ source -> destination sẽ được tô đậm hoặc làm nổi bật, in ra chi phí đường đi đó.
 - ✓ Chức năng export đồ thị đã được vẽ thành file ảnh.
- Chức năng giải bài toán lập lịch CPU (cài đặt tối thiểu 4 loại: FCFS, SJF, Priority, RR):
 - ✓ Dữ liệu đầu vào được lấy từ form hoặc text file như chức năng tìm đường đi, bao gồm các thông tin: tên tiến trình, thời gian dùng CPU, thời gian đến, độ ưu tiên.... Các thông tin này có thể khác nhau tùy theo thuật toán lập lịch được chọn.
 - ✓ Nhận output từ server và xuất kết quả ra màn hình, đồng thời vẽ giản đồ Gantt minh họa.

Yêu cầu về chức năng phía server (phải có GUI):

- Kiểm tra định dạng/cấu trúc và dữ liệu đầu vào của file input có hợp lệ.
- Tính toán cách vẽ đồ thị, tính toán đường đi từ source -> destination và gửi về client. Client chỉ vẽ theo kết quả từ server trả về, không thực hiện tính toán gì thêm.
- Thuật toán tìm đường do nhóm SV tự chọn.
- Giải bài toán lập lịch theo thuật toán mà client gửi -> gửi trả kết quả lại client.

Yêu cầu chung:

- Các client phải chạy trên các máy tính khác nhau.
- Mã hóa dữ liệu giữa client – server.

ĐỀ TÀI 15: MÔ PHỎNG HỆ THỐNG EMAIL BẰNG JAVA SOCKET

Yêu cầu về chức năng phía client (phải có GUI):

- Chức năng đăng ký/đăng nhập. Thông tin tối thiểu cần cung cấp: username (sẽ chính là địa chỉ email được cấp); họ tên; password.
- Xây dựng các chức năng gửi nhận thư tương tự một mail client đơn giản:
 - ✓ Send (bao gồm cả CC và BCC) tới 1 hoặc nhiều địa chỉ email: nội dung email có thể format đơn giản (cỡ chữ, màu chữ, in đậm/nghiêng,...) và có thể nhúng hình ảnh, file đính kèm.
 - ✓ Receive
 - ✓ Reply; reply all
 - ✓ Mark spam email
 - ✓ Schedule send
 - ✓ Delete
- Mô phỏng cách phân loại email vào các nhóm: Inbox (dành cho email mới đến và chưa được đọc); Read (email đã được đọc); Sent (dành cho email đã gửi đi); Spam (dành cho email bị đánh dấu spam); Bin (dành cho email bị xóa).

Yêu cầu về chức năng phía server (không cần GUI):

- Sử dụng một tên miền giả (do SV) để làm tên miền email.
- Server ghi log về quá trình login/logout của client.
- Server có thể đặt và thay đổi dung lượng hộp thư của user. Mặc định là 100 MB / user.
- Server có thể khóa hoặc gửi email thông báo tới tất cả user.

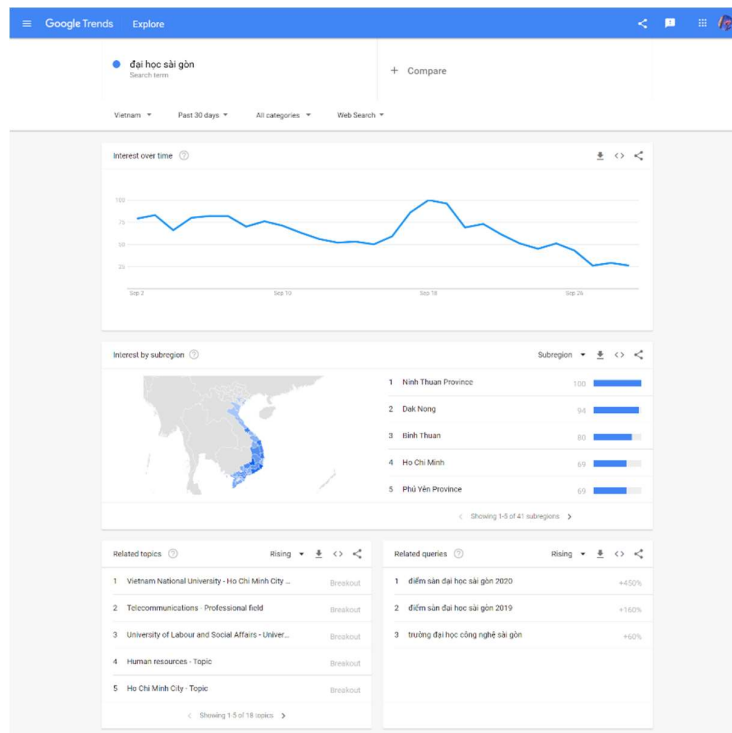
Yêu cầu chung:

- Mã hóa nội dung tin nhắn giữa client – server. Phải sử dụng key khác nhau cho các client.
- Các client phải chạy trên các máy tính khác nhau.

ĐỀ TÀI 16: ỨNG DỤNG THỐNG KÊ, SO SÁNH XU HƯỚNG TỪ KHÓA THEO THỜI GIAN

Mô tả chức năng ứng dụng:

- Ứng dụng nhằm cho biết mức độ quan tâm của người dùng ở một khu vực đối với một từ khóa nào đó và cho biết các số liệu thống kê chi tiết về từ khóa cũng như các từ khóa khác có liên quan với từ khóa đó. Khi người dùng chọn xem chi tiết 1 từ khóa, ứng dụng liệt kê các bài báo/blog có nội dung liên quan đến từ khóa để người dùng có thể truy cập.
- Tại client, người dùng nhập/chọn một số thông tin như: từ khóa muốn tra cứu; Phạm vi tra cứu: quốc gia cụ thể (như Việt Nam) hoặc toàn thế giới; Khoảng thời gian tra cứu: giờ, ngày, tháng, năm hoặc khoảng thời gian xác định. Client đẩy các thông tin trên đến server, server sử dụng kết quả từ Google Trend để trả về các thông tin cần thiết cho client. Xem hình minh họa:



- Ngoài ra, ứng dụng cho phép so sánh từ khóa người dùng đã nhập với 1 từ khóa bất kỳ (về số lượt tìm kiếm, khu vực tìm kiếm, ...) tham khảo link: <https://trends.google.com/trends/explore?date=today%201m&geo=VN&q=%C4%91%E1%BA%A1i%20h%E1%BB%8Dc%20s%C3%A0i%20g%C3%B2n,%C4%91%E1%BA%A1i%20h%E1%BB%8Dc%20s%C6%B0%20ph%E1%BA%A1m>
- Đây là đề tài mở, nhóm SV dựa trên dữ liệu từ Google Trends và có thể trao đổi thêm với GV về các chức năng mở rộng.

Yêu cầu về chức năng phía server (không cần GUI):

- Nhận request từ client -> Google Trend và xử lý kết quả trả về từ Google Trend.
- Mã hóa nội dung tin nhắn giữa client – server. Phải sử dụng key khác nhau cho các client
- Các client hoạt động trên các máy tính khác nhau.

ĐỀ TÀI 17: ỨNG DỤNG TRA CỨU LỊCH SỬ GIÁ SẢN PHẨM

Mô tả chức năng ứng dụng:

- Ứng dụng giúp người dùng theo dõi sự biến động giá của 1 sản phẩm trên 1 trang thương mại điện tử bất kỳ (do SV chọn) trước khi quyết định đặt mua. Client gửi tới server tên 1 sản phẩm và server sẽ phản hồi về các mức giá theo nhiều mốc thời điểm từ quá khứ đến hiện tại kèm theo đồ thị thể hiện thay đổi giá.
- Nhằm giúp việc tìm kiếm đạt kết quả tốt nhất, chương trình cần gợi ý hoặc có giải pháp để giúp người dùng nhập hoặc chọn lựa chính xác tên sản phẩm.
- Nhóm SV cần chuẩn bị sẵn CSDL sản phẩm cần theo dõi thuộc tối thiểu 8 nhóm hàng hóa với tổng số sản phẩm tối thiểu là 4000.
- Ngoài chức năng theo dõi biến động giá, ứng dụng có thể cung cấp review của người dùng (dựa trên kết quả của trang thương mại điện tử nhóm chọn/các trang thương mại điện tử khác/Google – tùy nhóm quyết định) đối với sản phẩm đó.

Yêu cầu về chức năng phía server (không cần GUI):

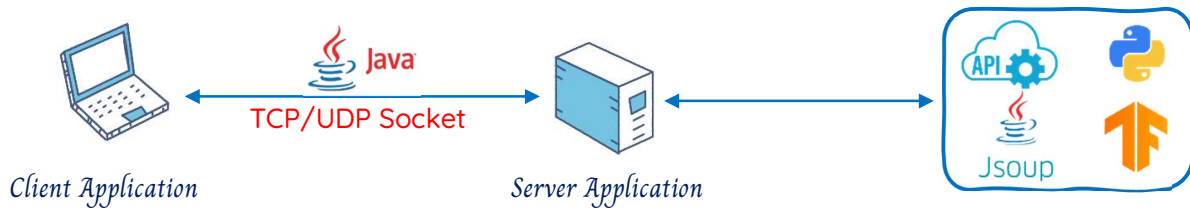
- Server nhận dữ liệu từ client và gửi lên các API hoặc tự xử lý.

Yêu cầu chung:

- Mã hóa nội dung tin nhắn giữa client – server. Phải sử dụng key khác nhau cho các client.
- Các client phải chạy trên các máy tính khác nhau.

YÊU CẦU CHUNG ĐỐI VỚI TẤT CẢ ĐỀ TÀI

- Ngôn ngữ chính được sử dụng là Java, sử dụng mô hình client-server, giao tiếp bằng TCPSocket hoặc UDPSocket. Để giải quyết các vấn đề xử lý tại server, nhóm sinh viên có thể sử dụng các thư viện, gọi API hoặc kết hợp các ngôn ngữ khác để trả kết quả xử lý về server.



- Yêu cầu chung đối với tất cả đề tài:
 - ✓ Mã hóa dữ liệu trao đổi giữa Client-Server, phương pháp mã hóa do nhóm tùy chọn.
 - ✓ Các client/server phải chạy trên các máy khác nhau (qua LAN hoặc Internet)
 - ✓ GUI bắt buộc đối với client, phiên bản server tùy đề tài có thể xây dựng GUI.
- Những nội dung bị trừ điểm đề tài:
 - ✓ Không thực hiện hoặc thực hiện sai các yêu cầu đã nêu trong mô tả.
 - ✓ Thành viên báo cáo không giải thích được chức năng hoặc không sửa được các lỗi phát sinh lúc báo cáo.
 - ✓ Không kiểm tra hết lỗi dữ liệu đầu vào, để phát sinh bất kỳ lỗi nằm ngoài dự kiến.
 - ✓ Sử dụng control không đúng chức năng hoặc thiết kế giao diện không trực quan
 - ✓ Hẹn lại lịch báo cáo.

Lưu ý: nội dung trong file mô tả và yêu cầu này chưa phải là bản cuối cùng. Nếu có sự thay đổi, GV sẽ báo trực tiếp cho SV trong giờ lên lớp.