



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÀI TẬP TUẦN 1

Kỳ 20231

Mục lục

1. Yêu cầu tổng quan của đề tài
2. Quy trình bảo vệ
3. Yêu cầu chức năng và phi chức năng
4. Cơ sở dữ liệu
5. Danh sách API
6. Đăng ký đăng nhập

Mục lục

1. **Yêu cầu tổng quan của đề tài**
2. Quy trình bảo vệ
3. Yêu cầu chức năng và phi chức năng
4. Cơ sở dữ liệu
5. Danh sách API
6. Đăng ký đăng nhập

1. Yêu cầu tổng quan của đề tài

- Phát triển một hệ thống có client (ứng dụng đa nền tảng), hệ thống database server và service server (tùy chọn công nghệ) để xây dựng lại (bắt chước một phần giao diện và một phần chức năng) một ứng dụng phổ biến với người dùng Việt Nam.
- Trong học kỳ này, ứng dụng được lựa chọn là Facebook. SV cần hoàn thành không những một ứng dụng chạy trên di động mà còn xây dựng (các) server hỗ trợ
 - Database lưu trữ các bài viết, thông tin của người dùng
 - Service server cung cấp các api để ứng dụng di động truy cập
 - Các push notification được thực hiện bằng Firebase (hoặc SV tự phát triển)

1. Yêu cầu tổng quan của đề tài

Tuy nhiên mạng xã hội này có sự khác biệt so với Facebook:

- + Phải nạp coins mới được tiếp tục đăng bài viết
- + Phải trừ coins nếu tin bị nhiều người nhận xét là tin giả
- + Phải trừ coins nếu nhận xét sai về tin do người khác đăng
- + Sẽ được thưởng coins nếu đăng tin đúng hoặc nhận xét đúng về tin
- + Sẽ được phép chuyển coins cho tài khoản khác.

Mạng xã hội này được gọi là Anti-Fakebook (AFB)

1. Yêu cầu tổng quan của đề tài

Lịch trình giới thiệu API và giao diện của bài tập tuần:

Tuần 1: API Đăng ký/Đăng nhập

Tuần 2:

- API Đăng ký/Đăng nhập (tiếp theo)
- Giao diện đăng ký/đăng nhập
- API đăng bài

Tuần 3:

- Giao diện đăng bài/hiển thị bài
- API truy vấn bài/sửa bài

Tuần 4:

- Giao diện sửa/xóa/báo cáo/xem bình luận bài
- API xóa/báo cáo/thích/xem BL/đăng BL của bài

Tuần 5:

- API danh sách bài/kiểm tra phần tử mới
- Giao diện trang chủ

1. Yêu cầu tổng quan của đề tài

Lịch trình giới thiệu API và giao diện của bài tập tuần (tiếp):

Tuần 6:

- API tìm kiếm/xem tìm kiếm/xóa tìm kiếm
- Giao diện tìm kiếm

Tuần 7:

- API yêu cầu kết bạn/lấy DS video và bạn bè
- Giao diện tab bạn bè/video

Tuần 8:

- API xử lý yêu cầu/xem danh sách gợi ý/yêu cầu kết bạn và xem danh sách chặn
- Giao diện tab thông báo/bạn bè

Tuần 9:

- API đổi mật khẩu và đọc/sửa cài đặt thông báo
- API chặn người dùng, kiểm tra phiên bản

1. Yêu cầu tổng quan của đề tài

Lịch trình giới thiệu API và giao diện của bài tập tuần (tiếp):

Tuần 10:

- API lấy/đọc thông báo
- API gán mã thiết bị
- API lấy/sửa thông tin cá nhân
- Giao diện trang người dùng

Tuần 11:

- Các API và giao diện cuối cùng phục vụ việc quản lý phía admin

Mục lục

1. Yêu cầu tổng quan của đề tài
2. **Quy trình bảo vệ**
3. Yêu cầu chức năng và phi chức năng
4. Cơ sở dữ liệu
5. Danh sách API
6. Đăng ký đăng nhập

2. Quy trình bảo vệ

2.1. Thành viên

2.2. Thời gian bảo vệ

2.3. Hình thức bảo vệ

2. Quy trình bảo vệ

2.1. Thành viên

2.2. Thời gian bảo vệ

2.3. Hình thức bảo vệ

2.1. Thành viên

Mỗi nhóm cần tối đa năm sinh viên để thực hiện các công việc sau (không bắt buộc):

- + Một bạn phát triển Database server, service server để có các dữ liệu bài viết, thông tin cá nhân, lịch sử hoạt động của người dùng
- + Một bạn phát triển thư viện xử lý JSON trả về từ server, phát triển chức năng quản trị phía admin
- + Một bạn phát triển giao diện và xử lý sự kiện cho ứng dụng di động
- + Một bạn xử lý push notification phía clients
- + Một bạn đảm nhiệm kiểm thử tổng hợp. Các thành viên có thể thực hiện kiểm thử chéo.

2. Quy trình bảo vệ

2.1. Thành viên

2.2. Thời gian bảo vệ

2.3. Hình thức bảo vệ

2.2. Thời gian bảo vệ

Chậm nhất là trước khi vào học **tuần thứ 3**, các sinh viên gửi thông tin về nhóm của mình cùng thông tin liên lạc của nhóm trưởng

Thứ tự bảo vệ sẽ được gửi cho lớp trước khi bảo vệ

Các nhóm đến bảo vệ theo đúng lịch của mình, với **đầy đủ** các thành viên.

Bảo vệ vào các buổi học **thứ 14, 15, 16, 17** của môn học

Điểm bài tập lớn sẽ được tính vào **điểm quá trình** của môn học

2.2. Thời gian bảo vệ (2)

Các nhóm sẽ được bảo vệ **thử** vào buổi học **thứ 13**

Một số nhóm sẽ được chọn lựa để đi bảo vệ thử vào buổi thứ 13.

Các nhóm chỉ lên trình bày công việc mình làm được, nếu chưa làm được gì hoặc chưa hoàn thành xong hết hoặc không đến thì vẫn được chấp nhận.

Các nhóm trưởng khác **nên** có mặt để tham khảo

Các nhóm nên chuẩn bị như thể đây là một buổi bảo vệ chính thức, để có thể có kết quả tốt trong buổi tiếp theo

2. Quy trình bảo vệ

2.1. Thành viên

2.2. Thời gian bảo vệ

2.3. Hình thức bảo vệ

2.3. Hình thức bảo vệ

Sinh viên nộp báo cáo bản cứng vào hôm bảo vệ và bản mềm ba ngày trước khi nhóm bảo vệ. Trong báo cáo ghi rõ phân công công việc, link github của mã nguồn hai phía.

Nhóm sinh viên chuẩn bị đầy đủ ít nhất là 2 điện thoại **THẬT** (kết nối mạng 3G/4G), server để demo. Không chấp nhận chỉ sử dụng bộ giả lập.

Khi một nhóm bảo vệ (các thành viên ngồi ở hai bàn đầu, dãy giữa), các nhóm khác ngồi cách xa **ít nhất hai bàn**

Bảo vệ xong, nhóm cần ở lại đến phút cuối của buổi học để có thể trả lời các câu hỏi khác (nếu có)

2.3. Hình thức bảo vệ (2)

Nhóm sinh viên không nhất thiết phải hoàn thành tất cả các chức năng trong danh sách

- Nhóm được cao điểm nhất là nhóm hoàn thành được nhiều chức năng nhất.

Để được điểm tối đa, nhóm cần phải hoàn thành tối thiểu được 80% các chức năng được yêu cầu.

Nhóm đó phải hoàn thiện chức năng push notification

Mục lục

1. Yêu cầu tổng quan của đề tài
2. Quy trình bảo vệ
3. **Yêu cầu chức năng và phi chức năng**
4. Cơ sở dữ liệu
5. Danh sách API
6. Đăng ký đăng nhập

3. Yêu cầu chức năng và phi chức năng

3.1. Yêu cầu chức năng

3.2. Yêu cầu phi chức năng

3. Yêu cầu chức năng và phi chức năng

3.1. Yêu cầu chức năng

3.2. Yêu cầu phi chức năng

Thuật ngữ

Trong mô tả dưới đây có khái niệm mark cho bài viết, nó hoàn toàn tương đương với bình luận cho bài viết

Một mark cũng sẽ có bình luận của nó, nhưng bình luận này được coi là bình luận của mark, không phải là bình luận của bài viết

Bình luận của mark cũng có thể có bình luận của nó.

Vậy từ nay “bình luận của bài viết” còn được gọi là mark của bài viết; “bình luận của mark” và “bình luận của bình luận” còn được gọi chung là bình luận

3.1. Yêu cầu chức năng (1)

Hệ thống cần phát triển có các chức năng sau:

- + Cho phép người dùng đăng ký (làm guest) và đăng nhập vào ứng dụng. Tên đăng nhập là email hợp lệ, mật khẩu không phải là tên email, không chứa các ký tự đặc biệt
- + Mọi guest đăng nhập lần đầu sẽ được cấp 10 coins, guest phải nạp tiền thì mới thành user (~~10.000VND~~ → 1 coins)
- + Cho phép user (guest đặc biệt) đăng bài mới với một chế độ duy nhất là public, số lượng bài bị giới hạn trong số coins.
- + Bài viết của người dùng giới hạn tối đa **bốn** hình ảnh HOẶC **một** video.
- + Bất kỳ guest nào cũng viết được mark (trust/fake) cho bài viết, khi đó họ phải cho cảm xúc: kudos/disappointed; và phải bị trừ coins.
- + Khi bài được đăng, bất kỳ ai (trừ khi bị chặn) cũng có thể xem. Số lần bày tỏ cảm xúc và số lần viết mark không vượt quá số coins.
- + Một người chỉ được viết tối đa 1 mark cho một bài viết.
- + Một tài khoản có thể kết bạn được với tối đa 5000 bạn

3.1. Yêu cầu chức năng (2)

- + Sau khi bị trừ tiền, hết 30 ngày tiếp theo, bài viết phải được admin xác nhận tin là thật (hoặc mark viết đúng) thì người đăng bài (hoặc mark đúng) được thưởng lại một số coins.
- + Một bài viết bất kỳ đều có chức năng cho phép người khác báo cáo
- + Một người có thể chặn (**block**) và bỏ chặn (**unblock**) bất kỳ ai. Việc chặn sau khi bỏ chặn không bị giới hạn thời gian như trên facebook
- + Việc chặn một người sẽ khiến người đó không còn là bạn bè của người thực hiện thao tác chặn
- + Việc người A chặn một người không khiến mark, **bình luận, cảm xúc** của người đó bị xóa khỏi trang cá nhân của A (nhưng A sẽ không thấy các mark/bình luận/cảm xúc này)
- + Việc chặn một người sẽ khiến hai người không xem được trang cá nhân của nhau.

3.1. Yêu cầu chức năng (3)

- + Một bài viết sau khi đăng thì có 1 ngày đầu để sửa, hết thời hạn thì tác giả không được sửa nữa. Sửa sẽ bị trừ coins như bài viết mới.
- + Nếu sau khi mark/bày tỏ cảm xúc mà tác giả bài viết sửa thì mark/cảm xúc sẽ chỉ hiện với người chủ mark/cảm xúc. Coins sẽ quay lại với những người đó. Bình luận cho mark sẽ biến mất cho đến khi mark được sửa lại (hoặc chủ mark xác nhận không sửa)
- + Chủ mark toàn quyền sửa mark của mình, nếu chỉ sửa nội dung sẽ không bị trừ coins. Đổi mark từ fake sang trust (hoặc ngược lại) sẽ bị trừ coin như tạo mark mới
- + Hiện tại chưa cho phép tiến hành lập nhóm nhiều người.
- + Khi một người đăng nhập thành công vào ứng dụng hệ thống sẽ hiển thị các bài viết do hệ thống gợi ý hoặc các bài viết của bạn bè.
- + Phải ưu tiên hiển thị những tin quan trọng (do hệ thống gợi ý) - dù tác giả không phải là bạn bè.
- + Bình luận của mark thì không bị trừ tiền.
- + Không bày tỏ cảm xúc cho mark và bình luận được.
- + Chủ cảm xúc/bình luận sửa được cảm xúc/bình luận của họ. Nhưng khi sửa cảm xúc (từ kudos sang disappointed hoặc ngược lại) sẽ bị trừ coins

3.1. Yêu cầu chức năng (4)

- + Không xem được danh sách những người bày tỏ cảm xúc một bài viết (nhưng biết được số lượng bày tỏ cảm xúc của bài viết).
- + Mark bắt buộc chứa văn bản, có thể chứa một trong số HTTP link, image, video; nhưng không thể chỉ có mỗi văn bản
- + Bình luận có thể chứa một trong số văn bản, HTTP link, image, video
- + Phía app mobile, được phép tùy chọn âm thanh push notification sao cho thể hiện được “trải nghiệm” tốt nhất
- + Các thông tin được push: có người bày tỏ cảm xúc bài viết của bạn, có người gắn mark cho bài viết của bạn, có người bình luận mark/bình luận của bạn; bạn bè đăng bài viết mới, có người muốn kết bạn với bạn, tài khoản đăng nhập trên máy khác... (xem chi tiết bài tập tuần 9)

3.1. Yêu cầu chức năng (5)

- + Tải ảnh có kích thước tối đa 4MB, không làm việc với ảnh động
- + Tải video có kích thước tối đa 10MB, thời lượng không quá 10s và phải 1s trở lên.
- + Hoàn toàn có thể sửa, xóa bài viết. Nhưng một bài viết không chứa lẫn lộn cả ảnh lẫn video
- + Việc xem video và xem ảnh cá nhân, ảnh bài viết đều có trải nghiệm giống như ứng dụng Facebook
- + Ở phía app, một khi bài viết được đăng xong hoặc sửa xong hoặc xóa xong sẽ chuyển về một trang nào đó (theo đúng kịch bản của facebook)
- + Không có việc tham gia và tạo lập nhóm

3.1. Yêu cầu chức năng (6)

- + Không cho phép tag thành viên khác vào bài viết hoặc đoạn chat hoặc bình luận
- + Ứng dụng cho phép việc chụp ảnh và quay video trong app. Ngoài ra vẫn có thể chọn các file ảnh và video có sẵn ở trong máy
- + Ứng dụng cho phép chèn các emoticon trong bài viết hoặc bình luận bằng cách nhập các ký tự đặc biệt
- + Không cho phép màn hình đăng nhập được chứa thêm nhiều tài khoản
- + Bài viết dài sẽ có nút xem thêm để thu gọn lại. Nhấn vào “Xem thêm” sẽ mở rộng nhưng sau đó không thu gọn được
- + Tất cả các ảnh upload lên sẽ được hiển thị ở phần trung tâm của ảnh

3.1. Yêu cầu chức năng (7)

- + Cho phép thiết lập tắt push gửi về hoặc thiết lập có push gửi về nhưng dùng âm thanh mặc định (âm kêu vang lên) hoặc âm có tiếng nói (yêu cầu SV thu âm giọng nói để làm file phát ra)
- + Cho phép tìm kiếm trong màn hình trang chủ của ứng dụng. Chẳng hạn tìm kiếm với từ khóa “bóng đội Nam Việt đá” thì vẫn ra được các kết quả về các bài viết có chứa các từ khóa “đội bóng Việt Nam đá”.
- + KHÔNG cho phép người dùng tự xóa tài khoản của mình
- + Xây dựng trang quản trị để người quản trị có thể dễ dàng thêm/sửa/xóa trực tiếp các dữ liệu cần thiết (về API Web thì các nhóm được quyền share cho nhau)

3. Yêu cầu chức năng và phi chức năng

3.1. Yêu cầu chức năng

3.2. Yêu cầu phi chức năng

3.2. Yêu cầu phi chức năng (1)

- + Tất cả dữ liệu trước khi gửi lên nên được kiểm tra ở phía **client** trước
- + Trừ lần đăng nhập đầu tiên, hệ thống cần có chức năng **cache** dữ liệu để tạo cảm giác **load** nội dung cực kỳ mau chóng cho người dùng. Việc **cache** chỉ áp dụng cho màn trang chủ.
- + Các **animation** cần phải bắt chước theo **Facebook**
- + Tính năng **pull down to refresh** và **pull up to load more** cần được cài đặt
- + Ứng dụng cần hoạt động thật trơn tru, tránh việc bị **crash** đột ngột
- + Một khi ứng dụng gửi yêu cầu lên **server**, cần có màn hình **loading**, sau đó một khi không có dữ liệu trả về hoặc dữ liệu trả về không đúng định dạng thì hiện thông báo về lỗi kết nối.
- + Các dữ liệu liên quan đến **privacy** của người dùng cần được mã hóa

3.2. Yêu cầu phi chức năng (2)

- + Khi tắt mạng của điện thoại, nhập dữ liệu và gửi yêu cầu thì ứng dụng vẫn tự xác thực input, nếu input đúng đắn mới kiểm tra trạng thái mạng.
- + Các thông báo cần có sự thống nhất với nhau, chẳng hạn ở màn hình **Login** và màn hình **Signup**, nếu người dùng nhập sai email (thừa số hoặc thiếu số) thì đều có chung một thông báo “email không đúng định dạng”
- + Việc mất kết nối **Internet** phải thông báo càng sớm càng tốt để hệ thống tránh các thao tác làm tổn tài nguyên cũng như mất thời gian của người dùng.
- + Các thông báo về việc người khác đã đăng nhập vào tài khoản trên máy khác cần được báo càng sớm càng tốt

3.2. Yêu cầu phi chức năng (2)

- + Khi tắt mạng của điện thoại, nhập dữ liệu và gửi yêu cầu thì ứng dụng vẫn tự xác thực input, nếu input đúng đắn mới kiểm tra trạng thái mạng.
- + Các thông báo cần có sự thống nhất với nhau, chẳng hạn ở màn hình **Login** và màn hình **Signup**, nếu người dùng nhập sai email (thừa số hoặc thiếu số) thì đều có chung một thông báo “email không đúng định dạng”
- + Việc mất kết nối **Internet** phải thông báo càng sớm càng tốt để hệ thống có thể xử lý kịp thời
- + Các thông báo lỗi phải hiển thị rõ ràng, dễ hiểu, tránh gây khó khăn cho người dùng

CÂU HỎI 1: LÀM THẾ NÀO ĐỂ THỰC HIỆN ĐƯỢC VIỆC CÓ CHUNG MỘT THÔNG BÁO NHƯ VẬY?

3.2. Yêu cầu phi chức năng (3)

- + Nhóm làm app mobile sẽ không gán các màu trong ứng dụng bằng các tên mà phải gán bằng các mã màu cụ thể, có lưu trữ danh sách các mã màu vào một file duy nhất để tiện quản lý.

=> TẠI SAO?

=> LÀM THẾ NÀO ĐỂ BIẾT MÃ MÀU CỦA TỪNG MÀU MỘT TRONG ỨNG DỤNG FACEBOOK?

3.2. Yêu cầu phi chức năng (4)

+ Nhóm cần làm cho các file cài không quá lớn.

=> **TẠI SAO**

=> **BẰNG CÁCH NÀO?**

Mục lục

1. Yêu cầu tổng quan của đề tài
2. Quy trình bảo vệ
3. Yêu cầu chức năng và phi chức năng
4. **Cơ sở dữ liệu**
5. Danh sách API
6. Đăng ký đăng nhập

4. Cơ sở dữ liệu

Cần thiết kế ra được cơ sở dữ liệu phục vụ các dữ liệu liên quan đến (dự kiến):

- + Tài khoản cá nhân: id, tên, mật khẩu, email, link avatar, token, session, danh sách id bị chặn bởi cá nhân
- + Bài viết cá nhân: id, id của chủ bài viết, nội dung bài viết, media, danh sách các id của bình luận, danh sách các id của người thích, thời điểm tạo
- + Bình luận: id, id của người viết bình luận, nội dung bình luận, thời điểm tạo
- + Một số bảng cơ sở dữ liệu khác (phục vụ cho push)
- + Đội phát triển được toàn quyền sử dụng hệ quản trị CSDL nào hoặc tự đề xuất cấu trúc của bảng
- + Bắt buộc phải sử dụng CSDL quan hệ

4. Cơ sở dữ liệu

Cần thiết kế ra được cơ sở dữ liệu phục vụ các dữ liệu liên quan đến (dự kiến):

- + Tài khoản cá nhân: id, tên, mật khẩu, email, link avatar, token, session, danh sách id bị chặn bởi cá nhân
- + Bài viết cá nhân: id, id của chủ bài viết, nội dung bài viết, media
- + Bình luận cá nhân: id, id của chủ bài viết, nội dung bình luận, người bình luận, thời điểm
- + Một số tính năng khác
- + Đội phát triển được toàn quyền sử dụng hệ quản trị CSDL nào hoặc tự đề xuất cấu trúc của bảng
- + Bắt buộc phải sử dụng CSDL quan hệ

CÂU HỎI 2: ĐỘI PHÁT TRIỂN CÓ ĐỀ XUẤT THIẾT KẾ CƠ SỞ DỮ LIỆU NÀO KHÁC KHÔNG?

Mục lục

1. Yêu cầu tổng quan của đề tài
2. Quy trình bảo vệ
3. Yêu cầu chức năng và phi chức năng
4. Cơ sở dữ liệu
5. **Danh sách API**
6. Đăng ký đăng nhập

5. Danh sách API

Danh sách 41 API cần được phát triển:

Login, logout, register, get_verify_code, verify_code, get_list_posts, get_post, post (post), post (patch), post (delete), comment/:postId (get), comment/:postId (post), comment/:postId (delete), report_post, reactions/:postId, search, get_saved_search, del_saved_search, friends (get), get_user_info, user (patch), get_list_videos, user/block (get), user/:userId/block (post), friends/:userId/accept (patch), friends/request (get), friends/:userId/request (post), get_push_settings, set_push_settings, change_password, check_new_version, set_devtoken, ~~get_conversation, delete_message, get_list_conversation, delete_conversation,~~ get_list_suggested_friends, check_new_item, get_notification, ~~set_read_message,~~ set_read_notification, buy_token, transfer_token, deactivate_user

5. Danh sách API

CÂU HỎI 3: NẾU ỨNG DỤNG CÓ TRUY VẤN DỮ LIỆU TỪ SERVER, THÌ NÓ CÓ ƯU ĐIỂM GÌ?

Danh sách 41 API

Login, logout, register, get_verify_code, verify_code, get_list_posts, get_post, post (post), post (patch), post (delete), comment/:postId (get), comment/:postId (post), comment/:postId (delete), report_post, reactions/:postId, search, get_saved_search, del_saved_search, friends (get), get_user_info, user (patch), get_list_videos, user/block (get), user/:userId/block (post), friends/:userId/accept (patch), friends/request (get), friends/:userId/request (post), get_push_settings, set_push_settings, change_password, check_new_version, set_devtoken, ~~get_conversation, delete_message, get_list_conversation, delete_conversation,~~ get_list_suggested_friends, check_new_item, get_notification, ~~set_read_message,~~ set_read_notification, buy_token, transfer_token, deactivate_user

RESTful API

Đường dẫn cơ bản:

<https://ABC.def/it4788/>

Tên miền ABC.def sinh viên tự thiết lập, cho phép local

Với việc sử dụng API Login thì truy cập

<https://ABC.def/it4788/login?>... với các tham số đi kèm

DANH SÁCH MÃ RESPONSE

Code	Message	Note
1000	OK	
9992	Post is not existed	Bài viết không tồn tại
9993	Code verify is incorrect	Mã xác thực không đúng
9994	No Data or end of list data	Không có dữ liệu hoặc không còn dữ liệu
9995	User is not validated	Không có người dùng này
9996	User existed.	Người dùng đã tồn tại
9997	Method is invalid	Phương thức không đúng
9998	Token is invalid.	Sai token
9999	Exception error.	Lỗi exception
1001	Can not connect to DB.	Lỗi mất kết nối DB/hoặc thực thi câu SQL
1002	Parameter is not enough.	Số lượng Paramater không đầy đủ
1003	Parameter type is invalid.	Kiểu tham số không đúng đắn.
1004	Parameter value is invalid.	Giá trị của tham số không hợp lệ
1005	Unknown error.	Unknown error
1006	File size is too big.	cỡ file vượt mức cho phép
1007	Upload File Failed!.	upload thất bại
1008	Maximum number of images.	số lượng images vượt quá quy định
1009	Not access.	không có quyền truy cập tài nguyên
1010	action has been done previously by this user.	hành động đã được người dùng thực hiện trước đây

DANH SÁCH MÃ RESPONSE (2)

1011 | Could not publish this post (Bài đăng vi phạm tiêu chuẩn cộng đồng)

1012 | Limited access (Bài đăng bị giới hạn ở một số quốc gia)

9991| Not enough coins (Không đủ coins để thực hiện hành động này)

Mục lục

1. Yêu cầu tổng quan của đề tài
2. Quy trình bảo vệ
3. Yêu cầu chức năng và phi chức năng
4. Cơ sở dữ liệu
5. Danh sách API
6. **Đăng ký đăng nhập**

6. Đăng ký đăng nhập

6.1. Signup

6.2. Login

6.3. Logout

6.4. Get_verify_code

Signup

O: bắt buộc, X: không bắt buộc

Input

NO	Tên parameter	Type	NN	Mô tả
1	email	string	0	
2	password	string	0	password
3	uuid	string	0	UUID thiết bị

Loại dữ liệu trả về là
JSON

Output

NO	Tên parameter	Type	Mô tả
1	code	string	(tham khảo phân response common)
2	message	string	
3	data		

Mô tả signup

API thực hiện việc cho phép đăng ký một tài khoản mới của người dùng

Request dạng POST

Tham số: **email** (địa chỉ email của người dùng), **password** (mật khẩu của người dùng), **uuid** (mã số định danh của thiết bị)

Kết quả đầu ra: Nếu đăng ký thành công thì trả về thông báo phù hợp đi kèm với việc sinh ra mã xác thực lưu ở server. Sẽ báo lỗi nếu có các trường hợp bất thường xảy ra.

Các test case cho Signup

1. Người dùng nhập email vốn là địa chỉ chưa được đăng ký trên hệ thống. Nhập đúng mật khẩu là xâu không trùng với email, không chứa ký tự đặc biệt.

Kết quả mong đợi: 1000 | OK (Thông báo thành công) kèm với mã xác thực được lưu ở phía server

Mã này đảm bảo không trùng nhau với hai số khác nhau, gồm cả số kèm chữ, có 6 ký tự

Chú ý: email phải đúng định dạng (có @, có tên miền...)

Mật khẩu phải có từ 6 đến 10 ký tự.

Các test case cho Signup (2)

2. Người dùng nhập email vốn đã được đăng ký thành công từ trước. Người dùng cũng nhập mật khẩu đúng quy định

Kết quả mong đợi: 9996 | User existed

3. Người dùng nhập email không đúng định dạng (không đủ ký tự hoặc chứa ký tự đặc biệt hoặc không có @) và nhập mật khẩu đúng quy định

Kết quả mong đợi: ứng dụng kiểm tra ngay và đưa ra thông báo sai định dạng của email. Kể cả khi không có mạng Internet thì ứng dụng vẫn kiểm tra và đưa thông báo.

Các test case cho Signup (3)

4. Người dùng nhập đúng định dạng của email nhưng mật khẩu không đúng định dạng (quá ngắn hoặc quá dài hoặc chứa ký tự đặc biệt hoặc trùng với email). Ứng dụng cần phải kiểm tra ngay trước khi gửi dữ liệu lên server

Kết quả mong đợi: ứng dụng kiểm tra ngay và đưa ra thông báo sai định dạng của mật khẩu. Kể cả khi không có mạng Internet thì ứng dụng vẫn kiểm tra và đưa thông báo.

5. Người dùng bỏ qua không nhập cả email và mật khẩu nhưng nhấn vào nút “Đăng ký” (hoặc “Sign up” tùy vào giao diện ứng dụng).

Kết quả mong đợi: ứng dụng kiểm tra ngay và đưa ra thông báo sai định dạng của email. Kể cả khi không có mạng Internet thì ứng dụng vẫn kiểm tra và đưa thông báo.

Login

O: bắt buộc, X: không bắt buộc

Input

NO	Tên parameter	Type	NN	Mô tả
1	email	string	0	
2	password	string	0	

Output

Loại dữ liệu trả về là
JSON

NO	Tên parameter	Type	NN	Mô tả
1	code	string	0	(tham khảo phân response common)
2	message	string	0	
3	data	string	0	Thông tin user
	id	string	0	
	username	string	0	
	token	string	0	
	avatar	string	0	-1: đợi đổi tên và ava; 1: actived
	active	string	0	0: non-actived
	coins	string	0	

Mô tả Login

API thực hiện việc cho phép đăng nhập vào một tài khoản của người dùng

Request dạng POST

Tham số: **email** (email của người dùng), **password** (mật khẩu của người dùng). **THÊM TRƯỜNG DEVICE ID** truyền lên nữa

Kết quả đầu ra: Nếu đăng nhập thành công thì (i) nhận thông báo thành công và (ii) chuyển ngay sang màn hình trang chủ. Sẽ báo lỗi nếu có các trường hợp bất thường xảy ra.

Với id (định danh người dùng), username (tên người), token (mã phiên đăng nhập), avatar (link đến avatar), active (1: được phép sử dụng tài khoản), coins (số coins có trong tài khoản)

Mô tả Login

API thực hiện việc cho phép đăng nhập vào một tài khoản của người dùng

Request dạng POST

Tham số: **email** (email của người dùng), **password** (mật khẩu của người dùng). **THÊM TRƯỜNG DEVICE ID** truyền lên nữa

Kết **CÂU HỎI 4: Mã TOKEN sẽ khác nhau nếu một người đăng nhập**
thà **cùng một tài khoản trên các máy khác nhau. Theo em tại sao lại**
nếu **cần mã token này?**

Với
phiên đăng nhập), avatar (link đến avatar), active (1: được phép sử dụng tài khoản), coins (số coins có trong tài khoản)

Mô tả Login

API thực hiện việc cho phép đăng nhập vào một tài khoản của người dùng

Request dạng POST

Tham số: **email** (email của người dùng), **password** (mật khẩu của người dùng). **THÊM TRƯỜNG DEVICE ID** truyền lên nữa

Kết **CÂU HỎI 5: Sẽ gửi về kết quả gì cho ứng dụng nếu như người**
thà **dùng vẫn sử dụng avatar mặc định?**
nếu

Với id (định danh người dùng), username (tên người), token (mã phiên đăng nhập), avatar (link đến avatar), active (1: được phép sử dụng tài khoản), coins (số coins có trong tài khoản)

Các test case cho Login

1, Người dùng nhập email vốn đã được đăng ký trên hệ thống. Nhập đúng mật khẩu và xâu không trùng với email, không chứa ký tự đặc biệt, thiết bị cũng gửi đúng mã devtoken (device ID).

Kết quả mong đợi: 1000 | OK (Thông báo thành công) và ứng dụng sẽ chuyển sang màn hình trang chủ. Chú ý, nếu ĐÃ đăng nhập trên máy khác thì (nếu chức năng push hoàn thành và máy khác có mạng) máy khác sẽ bị đăng xuất.

Mã token được lưu trữ ở cả client và server

2. Người dùng nhập email vốn CHƯA được đăng ký trên hệ thống. Nhập mật khẩu đúng định dạng và có mã devtoken hợp lệ.

Các test case cho Login (2)

Kết quả mong đợi: **CÂU HỎI 6: THEO EM KẾT QUẢ MONG ĐỢI LÀ GÌ?**

3. Người dùng nhập email không đúng định dạng và nhập mật khẩu đúng quy định, devtoken hợp lệ.

Kết quả mong đợi: ứng dụng kiểm tra ngay và đưa ra thông báo sai định dạng của email. Kể cả khi không có mạng Internet thì ứng dụng vẫn kiểm tra và đưa thông báo.

Các test case cho Login (3)

4. Người dùng nhập đúng định dạng của email nhưng mật khẩu không đúng định dạng (quá ngắn hoặc quá dài hoặc chứa ký tự đặc biệt hoặc trùng với email), devtoken hợp lệ. Ứng dụng cần phải kiểm tra ngay trước khi gửi dữ liệu lên server

Kết quả mong đợi: ứng dụng kiểm tra ngay và đưa ra thông báo sai định dạng của mật khẩu. Kể cả khi không có mạng Internet thì ứng dụng vẫn kiểm tra và đưa thông báo.

5. Người dùng bỏ qua không nhập cả email và mật khẩu nhưng nhấn vào nút “Đăng nhập” (hoặc “Log in” tùy vào giao diện ứng dụng).

Kết quả mong đợi: ứng dụng kiểm tra ngay và đưa ra thông báo sai định dạng của email. Kể cả khi không có mạng Internet thì ứng dụng vẫn kiểm tra và đưa thông báo.

Các test case cho Login (4)

6. Người dùng nhập đúng định dạng của email và mật khẩu, devtoken hợp lệ nhưng không có kết nối mạng.

Kết quả mong đợi: Ứng dụng cần phải đưa ra thông báo càng sớm càng tốt về việc “Không thể kết nối Internet”. Mẹo: Kể cả khi có kết nối mạng Internet nhưng server không trả về kết quả hoặc kết quả không đúng định dạng JSON thì vẫn thông báo là “Không thể kết nối Internet”.

7. Người dùng nhập email và mật khẩu có nội dung giống email rồi nhấn vào nút “Đăng nhập” (hoặc “Log in” tùy vào giao diện ứng dụng), devtoken hợp lệ.

Kết quả mong đợi: ứng dụng kiểm tra ngay và đưa ra thông báo “Mật khẩu sai định dạng”. Kể cả khi không có mạng Internet thì ứng dụng vẫn kiểm tra và đưa thông báo.

Các test case cho Login (4)

6. Người dùng nhập đúng định dạng của email và mật khẩu, devtoken hợp lệ nhưng không có kết nối mạng.

Kết quả mong đợi: Ứng dụng cần phải đưa ra thông báo càng sớm càng tốt về việc “Không thể kết nối Internet”. Mẹo: Kể cả khi có kết nối mạng Internet nhưng server không trả về kết quả hoặc kết quả không thể kết nối.

CÂU HỎI 7: THEO EM VIỆC LIÊN TỤC KIỂM TRA DỮ LIỆU TRƯỚC Ở PHÍA ỨNG DỤNG SẼ CÓ ƯU ĐIỂM GÌ?

7. Người dùng nhập sai định dạng của email hoặc mật khẩu (hoặc devtoken không hợp lệ), devtoken hợp lệ.

Kết quả mong đợi: ứng dụng kiểm tra ngay và đưa ra thông báo “Mật khẩu sai định dạng”. Kể cả khi không có mạng Internet thì ứng dụng vẫn kiểm tra và đưa thông báo.

Các test case cho Login (5)

8. Người dùng đăng nhập thành công trên máy A. Rồi đăng nhập thành công trên máy B.

Kết quả mong đợi: 1000 | OK (Thông báo thành công) và token cũ (của lần đăng nhập trên máy A) sẽ bị xóa đi bên phía server.

9. Ứng dụng gửi thông tin đăng nhập nhưng không gửi devtoken.

Kết quả mong đợi: Ứng dụng phải tự kiểm tra nhưng nếu lỡ gửi lên thì server phải về thông báo lỗi thiếu tham số.

Logout

Input

NO	Tên parameter	Type	NN	Mô tả
1	token	string	0	token từ sever

Output

NO	Tên parameter	Type	NN	Mô tả
1	code	string	0	(tham khóa phần response common)
2	message	string	0	

Mô tả Logout

API thực hiện việc cho phép đăng xuất khỏi một tài khoản của người dùng

Request dạng POST

Tham số: **token** (mã phiên đăng nhập của người dùng)

Kết quả đầu ra: Nếu đăng xuất thành công thì (i) nhận thông báo thành công và (ii) chuyển ngay sang màn hình đăng nhập. Sẽ báo lỗi nếu có các trường hợp bất thường xảy ra.

Các test case cho Logout

1. Người dùng chỉ đăng nhập trên một thiết bị và sau đó đăng xuất.

Kết quả mong đợi: 1000 | OK (Thông báo thành công) và ứng dụng sẽ chuyển sang màn hình đăng nhập. Token cũ sẽ bị xóa đi

2. Người dùng đăng nhập trên máy A, sau đó tắt mạng. Rồi đăng nhập trên máy B. Tiếp theo mở mạng của máy A ra và (push chưa kịp chuyển về máy A hoặc nhóm không làm push) đăng xuất.

Kết quả mong đợi: 1000 | OK (Thông báo thành công) và ứng dụng sẽ chuyển sang màn hình đăng nhập. Tất cả token sẽ bị xóa hết

Các test case cho Logout (2)

3. Người dùng đăng nhập trên máy A, sau đó tắt mạng. Rồi đăng nhập trên máy B, sau đó đăng xuất trên máy B. Tiếp theo mở mạng của máy A ra và (push chưa kịp chuyển về máy A hoặc nhóm không làm push) đăng xuất.

Kết quả mong đợi: 1000 | OK (Thông báo thành công) và ứng dụng ở máy A sẽ chuyển sang màn hình đăng nhập. Tất cả token sẽ bị xóa hết ở cả 2 phía

4. Người dùng đăng nhập trên máy A và ngắt kết nối mạng. Rồi đăng nhập trên máy B, rồi đăng xuất trên máy B.

Kết quả mong đợi: 1000 | OK (Thông báo thành công) và không còn bất kỳ trường token nào dành cho người dùng này ở cả hai phía

get_verify_code

Input

NO	Tên parameter	Type	NN	Mô tả
1	email	string	0	

Output

NO	Tên parameter	Type	NN	Mô tả
1	code	string	0	(tham khóa phần response common)
2	message	string	0	

Mô tả get_verify_code

API thực hiện việc nhận mã xác thực cho một tài khoản của người dùng

Request dạng POST

Tham số: **email**

Kết quả đầu ra: nếu nhận thành công thì sẽ có mã xác thực được tạo ra phía server. Server sẽ lưu trữ mã này. Nếu không thành công thì sẽ có các thông báo lỗi tương ứng

Các test case cho get_verify_code

1. Người dùng gọi request này lại khi mà chưa nhận được hoặc làm mất mã xác thực.

Kết quả mong đợi: 1000 | OK (Thông báo thành công) và mã xác thực cũ sẽ vẫn được tồn tại

1. Người dùng liên tục gửi request này lại trong thời gian quá ngắn.

Kết quả mong đợi: Nếu dưới 120 giây sau khi đã gửi request này thì ứng dụng sẽ làm mờ nút gửi mã xác thực. Nếu một lý do nào đó request này đến được server thì sẽ có thông báo trả về mã 1010 hoặc 1009

Các test case cho get_verify_code (2)

3. Người dùng truyền tham số với email đã được hoàn tất quy trình đăng ký (tức việc xác thực cũng đã xong).

Kết quả mong đợi: mã lỗi 1010

4. Người dùng truyền tham số với email chưa được đăng ký.

Kết quả mong đợi: trả về mã lỗi 1004 hoặc 9995

5. Người dùng truyền tham số với email sai định dạng.

Kết quả mong đợi: Ứng dụng sẽ phải kiểm tra từ trước để thông báo lỗi không đúng định dạng. Nếu như được gửi lên server sẽ được nhận về mã 1004

HẾT TUẦN 1



25 YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you
for your
attentions!**



soict.hust.edu.vn/



fb.com/groups/soict

