BÀI TẬP TUẦN 10

Nhắc lại

Đã nắm được thêm một số API và giao diện của màn trang chủ

Tiếp tục với những API của màn trang chủ và giao diện trang cá nhân

Danh sách API

Danh sách 41 API cần được phát triển:

Login, logout, signup, get_verify_code, *check_verify_code*, *change_info_after_signup*, *get_list_posts*, *get_post*, *add_post*, *edit_post*, *delete_post*, *get_comment*, *set_comment*, *report_post*, *like*, *search*, *get_saved_search*, *del_saved_search*, *get_user_friends*, get_user_info, set_user_info, *get_list_videos*, *get_list_blocks*, *set_block*, *set_accept_friend*, *get_requested_friends*, *set_request_friend*, *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,

Mục lục

Mục lục

get_notification

Input

NO	Tên parameter	Type	NN	Mô tả
1	token	string	O	
2	index	integer	O	
3	count	integer	O	

Output

NO	Tên parameter	Type	NN	Mô tả
1	code	string	O	(tham khảo phần response common)
2	message	string	O	
3	data	string	O	
	type	string	O	
	object_id	string	O	id
	title	string	O	
	notification_id	string	O	id của notification
	created	string	O	
	avatar	string	O	
	group	string	O	0: notification; 1: action
	read	string	O	0: chua doc; 1: da doc
	last_update	string	O	thời gian server
	badge	string	O	

Mô tả get_notification

API thực hiện việc lấy danh sách các thông báo đẩy

Request dạng POST

Tham số: **token,** index và count

Kết quả đầu ra: Nếu thành công thì mã thông báo thành công và các dữ liệu khác được trả về. Nếu không thành công thì sẽ có các thông báo lỗi tương ứng

Mô tả get_notification (2)

Giải thích ý nghĩa của từng trường output trả về:

- type: đặc tả tên giao diện cần được chuyển đến khi người dùng nhấn vào tin thông báo.
- object_id: id để truyền tham số cho việc chuyển màn hình.
- title: tiêu đề của thông báo
- notification_id: id của tin thông báo
- created: thời gian tạo ra thông báo
- avatar: đường dẫn đến ảnh avatar đại diện cho thông báo push. Đó có thể là ảnh của người dùng hoặc ảnh local một số biểu tượng nào đó (video, báo cáo)
- group: 0 nghĩa là nhấn vào sẽ không chuyển sang trang nào cả (nếu trong app) hoặc mở ra trang chủ (nếu không trong app).

Mô tả get_notification (3)

- Khi group = 1 nghĩa là nhấn vào thì sẽ chuyển sang một giao diện nào đó.
- read: 0 nghĩa là chưa được người dùng nhấn vào, 1 nghĩa là người dùng đã nhấn rồi
- badge: giá trị số đếm biểu thị có bao nhiêu notification mới xuất hiện
- last_update: thời gian server tính toán ra giá trị badge ở trên. Chỉ sử dụng giá trị badge nào có thời gian gần với hiện tại nhất.
- Các dữ liệu của API get_notification là được lấy từ HTTP request có thể là một mảng nhiều tin thông báo khác nhau. Như vậy, sẽ có khác biệt với dữ liệu từ push notification trả về (qua Firebase chẳng hạn)

Mô tả get_notification (4)

- Dữ liệu push được trả về sẽ có cấu trúc tương tự như output của API này, tuy vậy một push chỉ đại diện cho một thông báo duy nhất.
- Cần hiểu rằng push notification trả về sẽ có độ trễ nhất định, không ai nắm được giá trị cụ thể đó.
- Vì thế nếu badge thu được từ push và từ API HTTP request khác nhau thì ta sử dụng giá trị nào có last_update gần thời điểm hiện tại nhất.

Các test case cho get_notification

1. Người dùng truyền đúng mã phiên đăng nhập và các tham số khác

   Kết quả mong đợi: 1000 | OK (Thông báo thành công), gửi cho ứng dụng các thông tin cần thiết.

2. Người dùng gửi sai mã phiên đăng nhập (mã bị trống hoặc quá ngắn hoặc mã phiên đăng nhập cũ).

   Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập. Xem lại test case 3 của change_info_after_signup

Các test case cho get_notification (2)

3. Người dùng truyền đúng mã phiên đăng nhập nhưng hệ thống không thể thiết lập việc xử lý yêu cầu (do lỗi truy cập CSDL chẳng hạn)

    Kết quả mong đợi: thông báo cho người dùng, chẳng hạn như “Không thể kết nối Internet”

4. Người dùng truyền đúng mã phiên đăng nhập. Nhưng người dùng đã bị khóa tài khoản (do hệ thống khóa đi).

    Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập. Xem lại test case 3 của change_info_after_signup

Các test case cho get_notification (3)

5. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác nhưng hệ thống không trả về last_update.

    Kết quả mong đợi: Ứng dụng bỏ qua các giá trị badge nào mà không có trường last_update đi kèm.

6. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống trả về có tin thông báo không có title.

    Kết quả mong đợi: những tin nào không có title thì mặc định ứng dụng bỏ qua.

Các test case cho get_notification (4)

7. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống trả về có tin thông báo bị lỗi ở các trường sau: notification_id, created, object_id (nếu cần phải chuyển giao diện).

    Kết quả mong đợi: ứng dụng tự động loại bỏ những tin thông báo bị lỗi ở các trường đó.

8. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống trả về có tin thông báo bị lỗi ở các trường sau: type, avatar, group, read, object_id (nếu không cần chuyển giao diện).

Các test case cho get_notification (5)

Kết quả mong đợi: Ứng dụng tự gán các giá trị mặc định type là màn hình trang chủ, avatar là icon của ứng dụng, group là 0, read là 0 (chưa đọc), object_id bằng 0

9. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống báo sai giá trị badge (âm hoặc sai định dạng).

Kết quả mong đợi: Ứng dụng coi như giá trị đó bằng 0.

10. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ và hệ thống báo giá trị badge hợp lệ.

Các test case cho get_notification (6)

Kết quả mong đợi: Ứng dụng hiển thị một hình tròn đỏ ở trên hình cái chuông, trong đó có hiển thị số badge. Nếu số đó lớn hơn 100 thì ghi 99+. Chú ý vòng tròn đỏ phải to dần nếu cần hiển thị 1, 2, hoặc 3 ký tự.

11. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ và hệ thống trả về các trường thông báo trong đó có thông báo bị lỗi trường type.

Kết quả mong đợi: type có giá trị là màn trang chủ. Tuy vậy nếu người dùng đang ở màn trang chủ mà nhấn vào đó sẽ vẫn ở màn trang chủ.

12. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác. Hệ thống gửi về có tất cả một JSON hợp lệ (chuyển đổi thành đối tượng được)

Các test case cho get_notification (7)

Kết quả mong đợi: ứng dụng cần lưu lại cache, cần đảm bảo các hiệu ứng pull down và pull up như bình thường.

API get_notification này được gọi khi nào?

Khi vào màn hình trang chủ, nếu chưa có cache lưu trữ các tin thông báo đẩy thì sẽ gọi API này. Hoặc khi người dùng tạo hiệu ứng pull down ở trong tab thông báo (hoặc pull up khi đã hiển thị được hết dữ liệu trong cache)

Các thông báo do push chuyển về máy có cần phải lưu vào cache không? Để đơn giản thì không.

CÂU HỎI: THEO EM CÒN CÓ CÁC TEST CASE NÀO KHÁC NỮA?

Mục lục

set_read_notification

Input

NO	Tên parameter	Type	NN	Mô tả
	token	string	O	
	notification_id	string	O	

Output

NO	Tên parameter	Type	NN	Mô tả
1	code	string	O	(tham khảo phần response common)
2	message	string	O	
3	data	string	O	
	badge	string	O	số noti chưa đọc
	last_update	string	O	

Mô tả set_read_notification

API thực hiện việc ghi nhận người dùng đã đọc tin thông báo đẩy

Request dạng POST

Tham số: **token,** notification_id

Kết quả đầu ra: Nếu thành công thì mã thông báo thành công và các dữ liệu khác được trả về. Nếu không thành công thì sẽ có các thông báo lỗi tương ứng. Dẫu thành công hay không, ở client vẫn luôn chấp nhận rằng người dùng đã đọc tin thông báo và đổi màu của tin.

Các test case cho set_read_notification

1. Người dùng truyền đúng mã phiên đăng nhập và các tham số khác

   Kết quả mong đợi: 1000 | OK (Thông báo thành công), gửi cho ứng dụng các thông tin cần thiết.

2. Người dùng gửi sai mã phiên đăng nhập (mã bị trống hoặc quá ngắn hoặc mã phiên đăng nhập cũ).

   Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập. Xem lại test case 3 của change_info_after_signup

Các test case cho set_read_notification (2)

3. Người dùng truyền đúng mã phiên đăng nhập nhưng hệ thống không thể thiết lập việc xử lý yêu cầu (do lỗi truy cập CSDL chẳng hạn)

    Kết quả mong đợi: thông báo cho người dùng, chẳng hạn như “Không thể kết nối Internet”

4. Người dùng truyền đúng mã phiên đăng nhập. Nhưng người dùng đã bị khóa tài khoản (do hệ thống khóa đi).

    Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập. Xem lại test case 3 của change_info_after_signup

Các test case cho set_read_notification (3)

5. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác nhưng hệ thống không trả về last_update.

    Kết quả mong đợi: Ứng dụng bỏ qua các giá trị badge nào mà không có trường last_update đi kèm.

6. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống trả về có tin thông báo không có badge.

    Kết quả mong đợi: những tin nào không có badge thì mặc định ứng dụng bỏ qua.

Các test case cho set_read_notification (4)

7. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống phát hiện rằng tin này đã được báo “đọc” từ trước đó.

Kết quả mong đợi: dẫu hệ thống trả về nội dung gì, ở client vẫn luôn chấp nhận rằng người dùng đã đọc tin thông báo và đổi màu của tin.

Mục lục

1. API lấy các thông báo
2. API đã đọc thông báo
3. **API gán mã thiết bị**
4. API lấy thông tin người dùng
5. API cập nhật thông tin người dùng
6. Giao diện trang người dùng

set_devtoken

Input

NO	Tên parameter	Type	NN	Mô tả
1	token	string	O	
2	devtype	string	O	0: ios, 1:android
2	devtoken	string	O	

Output

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

Mô tả set_devtoken

API thực hiện việc ghi nhận device cho token của người dùng

Request dạng POST

Tham số: **token,** devtype (kiểu thiết bị), devtoken (mã thiết bị UID)

Kết quả đầu ra: Nếu thành công thì mã thông báo thành công và các dữ liệu khác được trả về. Nếu không thành công thì sẽ có các thông báo lỗi tương ứng.

Mô tả set_devtoken (2)

Vì sao phải có API này?

API này giúp hệ thống thực hiện được các thống kê về thói quen sử dụng của người dùng Android và iOS, từ đó có những chiến lược điều chỉnh đúng đắn để phục vụ khách hàng tốt hơn

Mỗi khi bật ứng dụng lên, ứng dụng sẽ kiểm tra mình đã gửi chưa và gửi đi nếu chưa gửi API này bao giờ.

Mô tả set_devtoken (3)

Kẻ giả mạo có thể lấy cắp được token của người dùng rồi đăng nhập ở thiết bị khác.

CÂU HỎI: API NÀY CÓ GIÚP GÌ CHO VIỆC XÁC THỰC NGƯỜI DÙNG HAY KHÔNG?

Các test case cho set_devtoken

1. Người dùng truyền đúng mã phiên đăng nhập và các tham số khác

   Kết quả mong đợi: 1000 | OK (Thông báo thành công), gửi cho ứng dụng các thông tin cần thiết. Ghi nhớ đã gửi để lần sau không gửi tiếp nữa. Chú ý, khi đăng xuất thì sự ghi nhớ sẽ biến mất.

2. Người dùng gửi sai mã phiên đăng nhập (mã bị trống hoặc quá ngắn hoặc mã phiên đăng nhập cũ).

   Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập. Xem lại test case 3 của change_info_after_signup

Các test case cho set_devtoken (2)

3. Người dùng truyền đúng mã phiên đăng nhập nhưng hệ thống không thể thiết lập việc xử lý yêu cầu (do lỗi truy cập CSDL chẳng hạn)

    Kết quả mong đợi: thông báo cho người dùng, chẳng hạn như “Không thể kết nối Internet”

4. Người dùng truyền đúng mã phiên đăng nhập. Nhưng người dùng đã bị khóa tài khoản (do hệ thống khóa đi).

    Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập. Xem lại test case 3 của change_info_after_signup

Các test case cho set_devtoken (3)

5. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác nhưng hệ thống phát hiện devtype không có giá trị hợp lệ.

    Kết quả mong đợi: Ứng dụng phải cố gắng gửi giá trị đúng đắn. Nếu lỡ gửi và bị lỗi trả về, ứng dụng không hiển thị thông báo gì cho người dùng.

6. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống phát hiện devtoken không có hoặc không hợp lệ.

    Kết quả mong đợi: Ứng dụng phải cố gắng gửi giá trị đúng đắn. Nếu lỡ gửi thì hệ thống cố kiểm tra devtoken nào là giả và gửi lỗi trả về. Khi ấy, ứng dụng không hiển thị thông báo gì cho người dùng.

Mục lục

get_user_info

Input

NO	Tên parameter	Type	NN	Mô tả
1	token	string	X	
2	user_id	integer	X	nếu get info của chính User đang login thì không cần user_id

Output

NO	Tên parameter	Type	NN	Mô tả
1	code	string	O	(tham khảo phần response common)
2	message	string	O	
3	data	array	O	
	id	string	X	
	username	string	X	
	created	string	O	
	description	string	O	
	avatar	string	O	
	cover_image	string	O	
	link	string	O	liên kết
	address	string	O	
	city	string	O	
	country	string	O	
	listing	string	O	số lượng bạn bè của user
	is_friend	string	O	người dùng hiện tại có phải là bạn của người này không?
	online	string	O	1: onine; 0: offline

Mô tả get_user_info

API thực hiện việc ghi nhận lấy thông tin cá nhân của một người dùng

Request dạng POST

Tham số: **token,** user_id (nếu lấy thông tin cá nhân của chính người chủ tài khoản thì trường này có thể bỏ qua)

Kết quả đầu ra: Nếu thành công thì mã thông báo thành công và các dữ liệu khác được trả về. Nếu không thành công thì sẽ có các thông báo lỗi tương ứng.

Các test case cho get_user_info

1. Người dùng truyền đúng mã phiên đăng nhập và các tham số khác

   Kết quả mong đợi: 1000 | OK (Thông báo thành công), gửi cho ứng dụng các thông tin cần thiết.

2. Người dùng gửi sai mã phiên đăng nhập (mã bị trống hoặc quá ngắn hoặc mã phiên đăng nhập cũ).

   Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập. Xem lại test case 3 của change_info_after_signup

Các test case cho get_user_info (2)

3. Người dùng truyền đúng mã phiên đăng nhập nhưng hệ thống không thể thiết lập việc xử lý yêu cầu (do lỗi truy cập CSDL chẳng hạn)

    Kết quả mong đợi: thông báo cho người dùng, chẳng hạn như “Không thể kết nối Internet”

4. Người dùng truyền đúng mã phiên đăng nhập. Nhưng người dùng đã bị khóa tài khoản (do hệ thống khóa đi).

    Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập. Xem lại test case 3 của change_info_after_signup

Các test case cho get_user_info (3)

5. Người dùng truyền đúng mã phiên đăng nhập, nhưng tham số user_id không tồn tại (hoặc đã bị khóa) - không phải là id của người chủ tài khoản.

    Kết quả mong đợi: Ứng dụng nhận được thông báo về tài khoản không tồn tại.

6. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống trả về có không có username hoặc id.

    Kết quả mong đợi: ứng dụng coi như đó là lỗi và thông báo về tài khoản không tồn tại.

Các test case cho get_user_info (4)

7. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống phát hiện rằng người dùng này đã chặn người chủ tài khoản.

    Kết quả mong đợi: Ứng dụng nhận được thông báo về tài khoản không tồn tại.

8. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống trả về có không có mô tả.

    Kết quả mong đợi: ứng dụng coi như người dùng đó không có thông tin mô tả.

Các test case cho get_user_info (5)

9. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống trả về không có (hoặc sai định dạng) **created, avatar, cover_image, link, address, country, listing, is_friend, online.**

    Kết quả mong đợi: Ứng dụng coi như người dùng không có các thông tin kia và tự gán giá trị mặc định.

10. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống trả về đa số đúng định dạng trừ việc không có (hoặc sai định dạng) một trong ba trường **country/address/city**.

    Kết quả mong đợi: Ứng dụng coi như người dùng không có các thông tin kia và tự gán giá trị mặc định.

Các test case cho get_user_info (6)

Kết quả mong đợi: Ứng dụng coi như việc thiếu (hoặc sai định dạng) một trường cấp cao sẽ khiến các trường cấp thấp bị mất ý nghĩa. Trường cao nhất là country, kế đến city và cuối cùng là address.

Hệ thống không bắt buộc phải lấy danh sách các quốc gia/tỉnh/thành theo đúng danh sách trên thực tế (nhóm SV làm thì càng tốt).

CÂU HỎI: THEO EM CÒN CÓ CÁC TEST CASE NÀO KHÁC

Mục lục

1. API lấy các thông báo
2. API đã đọc thông báo
3. API gán mã thiết bị
4. API lấy thông tin người dùng
5. **API cập nhật thông tin người dùng**
6. Giao diện trang người dùng

set_user_info

Input

NO	Tên parameter	Type	NN	Mô tả
1	token	string	O	
3	username	string	X	
4	description	string	X	
5	avatar	file	X	
8	address	string	X	
9	city	string	X	
10	country	string	X	
11	cover_image	file	X	
12	link	string	X	

Output

NO	Tên parameter	Type	NN	Mô tả
1	code	string	O	(tham khảo phần response common)
2	message	string	O	
3	data	string	O	
	avatar	string	X	đường dẫn đến liên kết
	cover_image	string	X	
	link	string	X	
	city	string	X	
	country	string	X	

Mô tả set_user_info

API thực hiện việc cập nhật thông tin cá nhân của một người dùng

Request dạng POST

Tham số: **token,** user_name, mô tả người dùng, avatar, địa chỉ, city, country, ảnh cover, liên kết (trừ token, các tham số khác tùy chọn)

Kết quả đầu ra: Nếu thành công thì mã thông báo thành công và các dữ liệu khác được trả về. Nếu không thành công thì sẽ có các thông báo lỗi tương ứng.

Mô tả set_user_info (2)

Trường hợp người dùng cập nhật ảnh avatar và cover_image thì cần chú ý:

- Do avatar là ảnh cá nhân cần hiển thị ở nhiều nơi nên nhìn chung ứng dụng sẽ tự biết lưu cache (người dùng không xóa được cache này) để tiết kiệm băng thông.
- Khi người dùng cập nhật avatar, server có hai tùy chọn: (i) chỉ đổi file, đường dẫn vẫn giữ nguyên. (ii) đổi cả file và đổi cả đường dẫn.
- Nên chọn cách thứ hai.

Các test case cho set_user_info

1. Người dùng truyền đúng mã phiên đăng nhập và các tham số khác

   Kết quả mong đợi: 1000 | OK (Thông báo thành công), gửi cho ứng dụng các thông tin cần thiết. Nếu có cập nhật avatar thì sẽ lưu lại đường dẫn.

2. Người dùng gửi sai mã phiên đăng nhập (mã bị trống hoặc quá ngắn hoặc mã phiên đăng nhập cũ).

   Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập. Xem lại test case 3 của change_info_after_signup

Các test case cho set_user_info (2)

3. Người dùng truyền đúng mã phiên đăng nhập nhưng hệ thống không thể thiết lập việc xử lý yêu cầu (do lỗi truy cập CSDL chẳng hạn)

    Kết quả mong đợi: thông báo cho người dùng, chẳng hạn như “Không thể kết nối Internet”

4. Người dùng truyền đúng mã phiên đăng nhập. Nhưng người dùng đã bị khóa tài khoản (do hệ thống khóa đi).

    Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập. Xem lại test case 3 của change_info_after_signup

Các test case cho set_user_info (3)

5. Người dùng truyền đúng mã phiên đăng nhập, nhưng tham số user_name bị sai định dạng (bị trống hoặc có chứa con số hoặc ký tự đặc biệt khác underscore, hoặc có ký tự đặc biệt ở đầu tiên hoặc quá dài).

    Kết quả mong đợi: Ứng dụng cố gắng tự bắt lỗi này trước khi gửi. Nếu lỡ gửi thì ứng dụng hiển thị thông báo phù hợp về lỗi này cho người dùng

6. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ user_name không phải xâu chuẩn.

    Kết quả mong đợi: ứng dụng tự chuẩn hóa xâu trước khi gửi lên. Khi cập nhật thành công (phía server) thì ứng dụng cũng chỉ hiện xâu chuẩn.

Các test case cho set_user_info (4)

7. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống phát hiện rằng đất nước của người dùng không được hỗ trợ bởi hệ thống (chẳng hạn Bắc Triều Tiên - North Korea/NorthKorea).

    Kết quả mong đợi: Ứng dụng nhận được thông báo sao cho đăng xuất.

8. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng trường description quá dài (hơn 150 ký tự).

    Kết quả mong đợi: Ứng dụng cố gắng tự bắt lỗi này trước khi gửi. Nếu lỡ gửi thì ứng dụng hiển thị thông báo phù hợp về lỗi này cho người dùng

Các test case cho get_user_info (5)

9. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ (có cả avatar và/hoặc cover_image và/hoặc link) nhưng hệ thống trả về không có đường dẫn của các trường đó.

    Kết quả mong đợi: Ứng dụng coi như người dùng không chỉnh sửa các thông tin kia và tự gán giá trị mặc định.

10. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ nhưng hệ thống phát hiện đường dẫn liên kết là đường dẫn bị cấm (để ví dụ, ta chọn trang vnhackers.com).

    Kết quả mong đợi: hệ thống phải chặn các trang bị cấm, báo cho ứng dụng và ứng dụng có cách hiển thị phù hợp.

Các test case cho get_user_info (6)

11. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác đầy đủ (có cả city và/hoặc country) nhưng hệ thống trả về không có các trường đó.

Kết quả mong đợi: Ứng dụng coi như người dùng không chỉnh sửa các thông tin kia và tự gán giá trị mặc định.

CÂU HỎI: THEO EM CÒN CÓ CÁC TEST CASE NÀO KHÁC

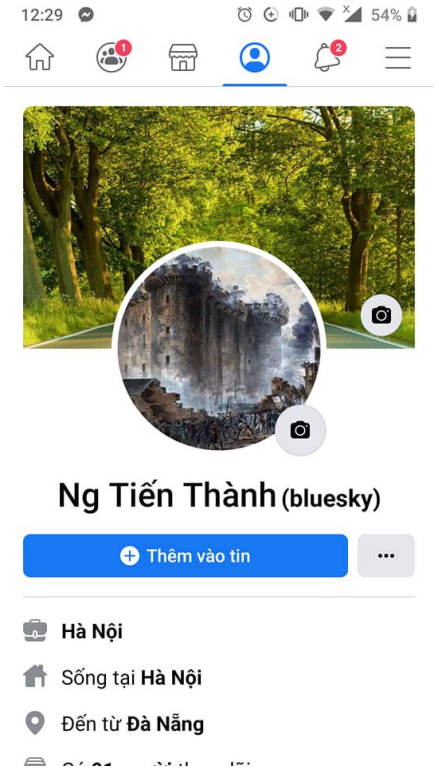
Mục lục

6. Giao diện trang người dùng

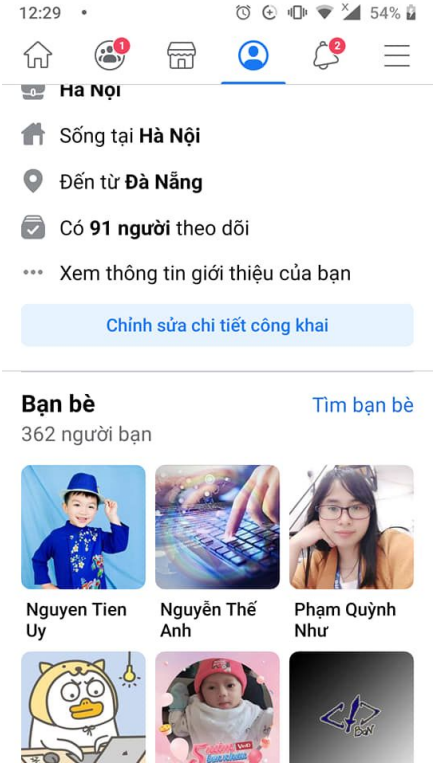
Đây là trang dùng để hiển thị các thông tin cá nhân của người dùng:

- Tên người dùng
- Ảnh đại diện
- Ảnh nền trang cá nhân
- Mô tả bản thân (150 ký tự)
- Địa chỉ nhà/tỉnh/thành/đất nước
- Trang liên kết
- Số lượng và danh sách bạn bè
- Danh sách các bài viết và video của cá nhân
- Số lượng like/bình luận của bài viết cá nhân
- Nếu là trang của chủ tài khoản sẽ hiển thị cả các bài bị khóa.

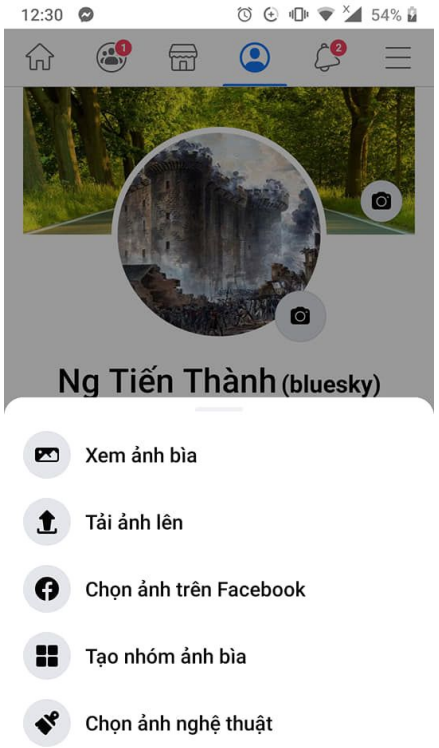
6. Giao diện trang người dùng (2)



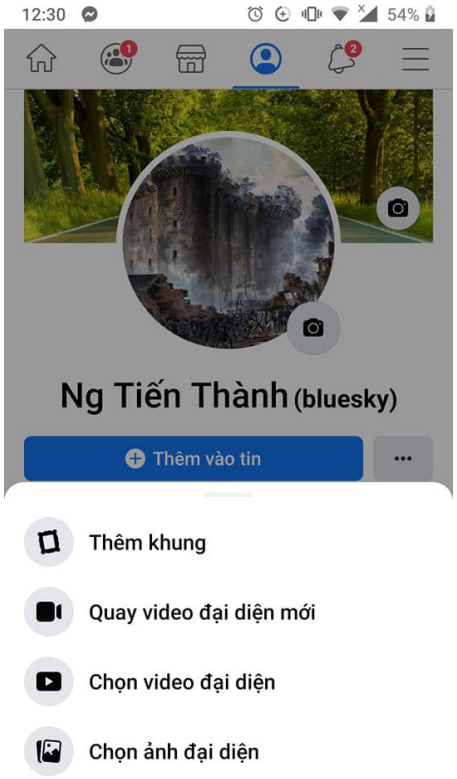
- Ứng dụng hiện chưa hỗ trợ tên biệt danh
- Không có chức năng "Thêm vào tin"
- Không hiển thị nơi học cũ
- Chỉ có một thành phố
- Không hiển thị số người theo dõi
- Không hỗ trợ thông tin giới thiệu



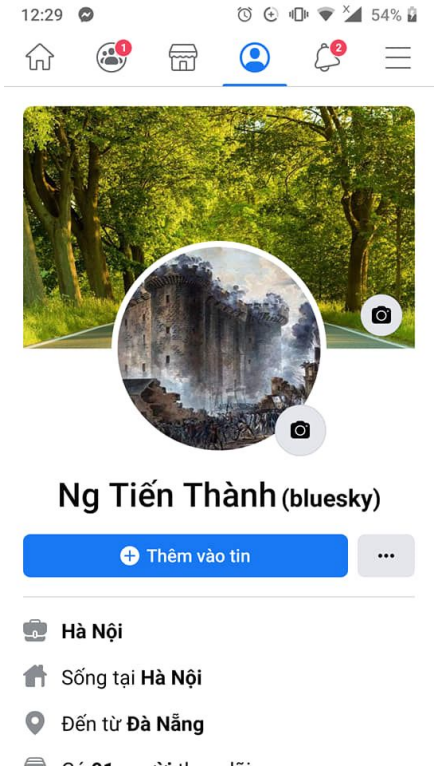
6. Giao diện trang người dùng (3)



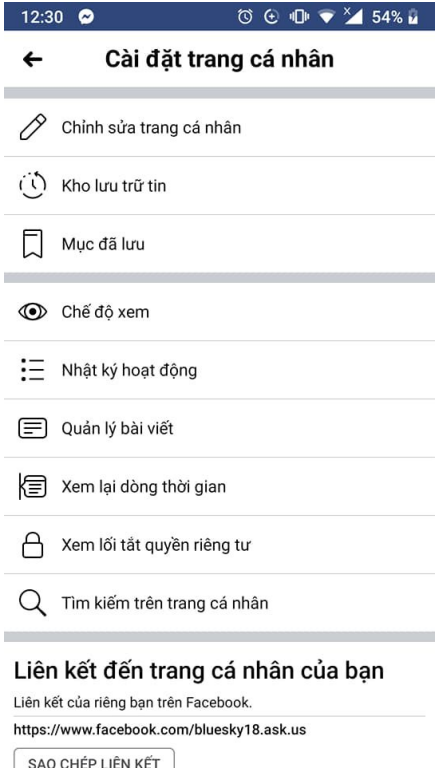
- Có hỗ trợ Xem ảnh bìa và Tải ảnh lên
- Không hỗ trợ Chọn ảnh trên Facebook, Tạo nhóm ảnh bìa hoặc Chọn ảnh nghệ thuật
- Không hỗ trợ Thêm khung, Quay video đại diện mới hoặc Chọn video đại diện
- Có hỗ trợ chọn ảnh đại diện



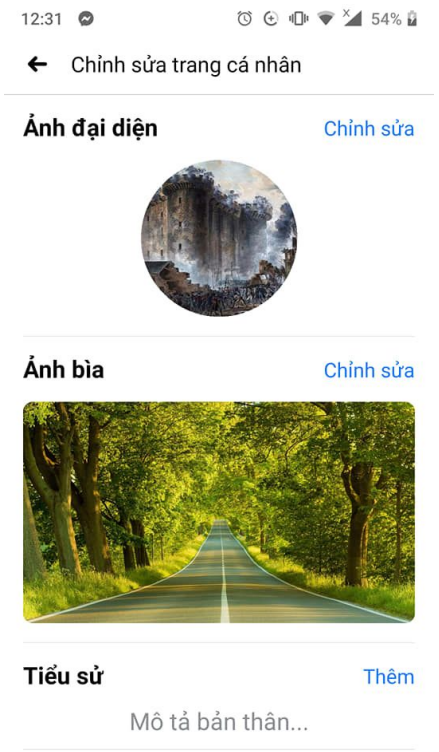
6. Giao diện trang người dùng (4)



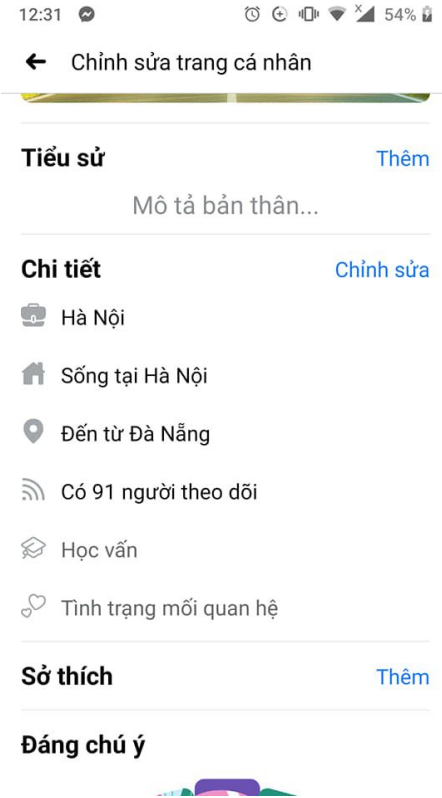
- Có hỗ trợ "Cài đặt trang cá nhân" - là nút có dấu ba chấm
- Trong Cài đặt trang cá nhân có cho phép "Chỉnh sửa trang cá nhân"
- Cho phép "Tìm kiếm trên trang cá nhân"
- Mục "Liên kết đến trang cá nhân của bạn" chính là hiển thị đường dẫn trong trường link (nếu có).
    - Cho phép sao chép liên kết



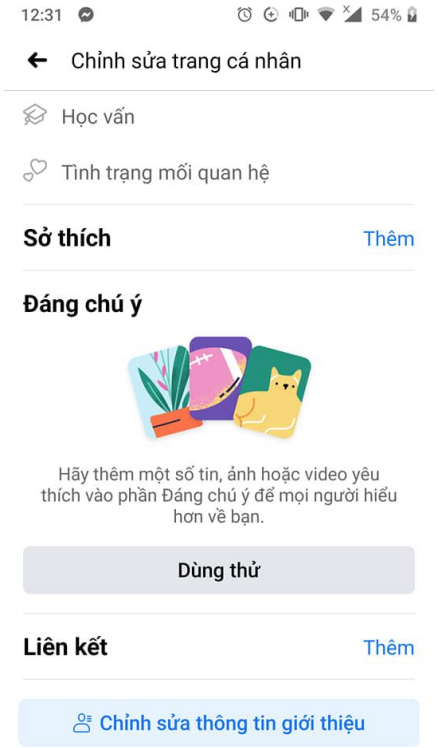
6. Giao diện trang người dùng (5)



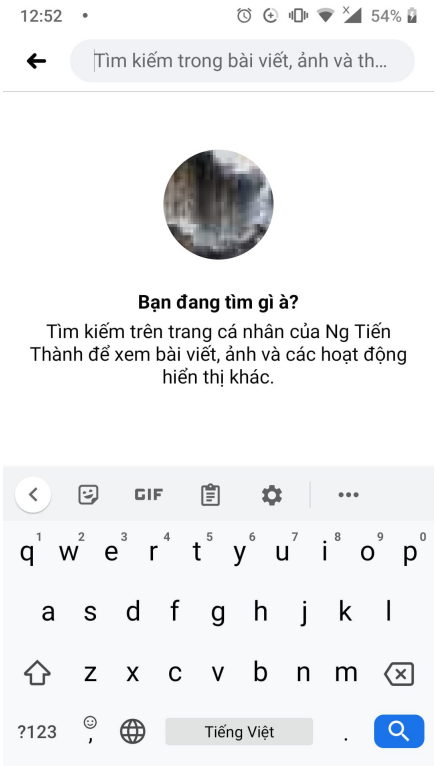
- Trong Cài đặt trang cá nhân có cho phép "Chỉnh sửa trang cá nhân" và "Tìm kiếm trên trang cá nhân"
- Tìm kiếm trên trang cá nhân sử dụng API search với tham số user_id nữa
- Mục "Liên kết đến trang cá nhân của bạn" chính là hiển thị đường dẫn trong trường link (nếu có).
    - Cho phép sao chép liên kết



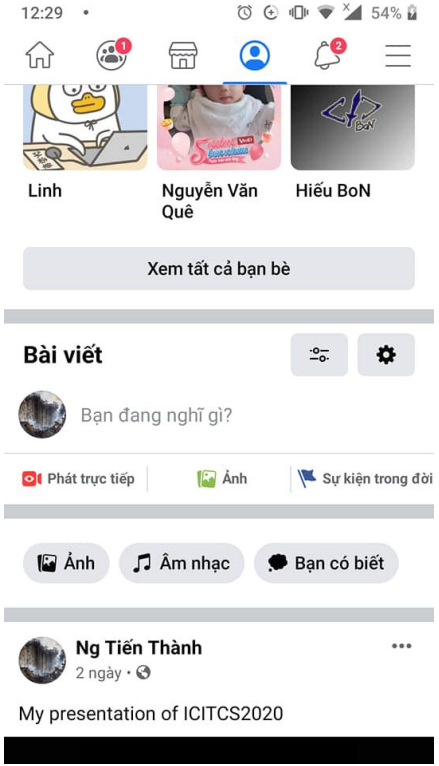
6. Giao diện trang người dùng (6)



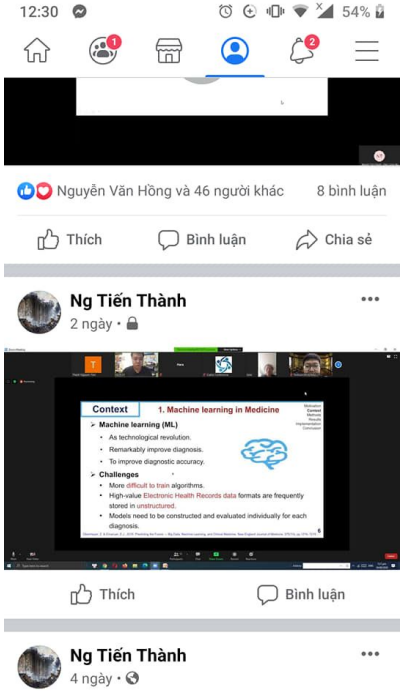
- Trong Cài đặt trang cá nhân có cho phép "Chỉnh sửa trang cá nhân" và "Tìm kiếm trên trang cá nhân"
- Tìm kiếm trên trang cá nhân sử dụng API search với tham số user_id nữa
- Mục "Liên kết đến trang cá nhân của bạn" chính là hiển thị đường dẫn trong trường link (nếu có).
  - Cho phép sao chép liên kết



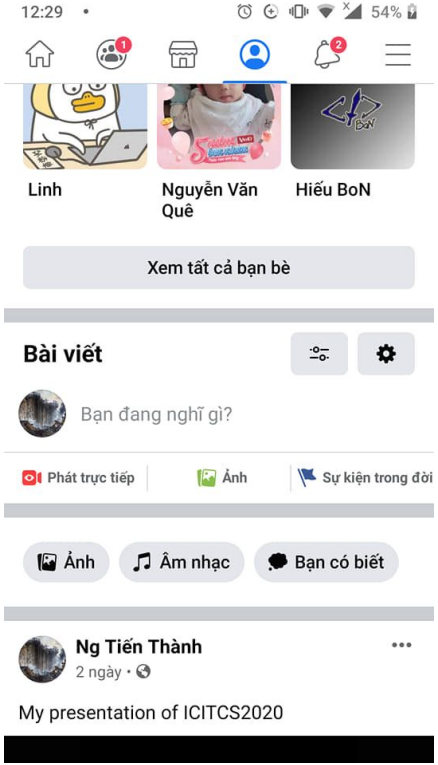
6. Giao diện trang người dùng (7)



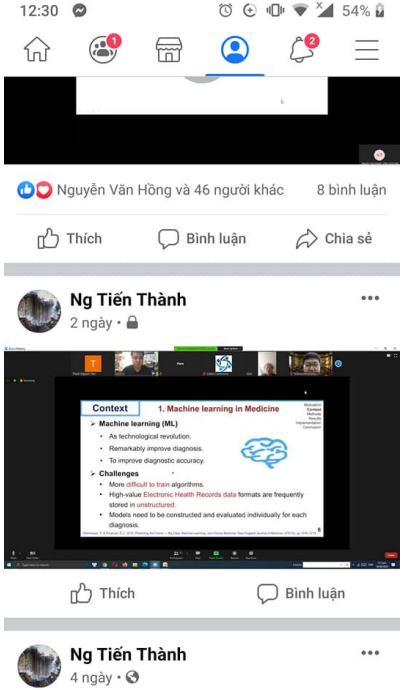
- Trong trang cá nhân, ta xem được danh sách các người dùng bằng cách gọi thêm API mang tên **get_user_friends**
- Để xem danh sách bài viết của một người ở trang cá nhân, ta gọi API **get_list_posts** với tham số **user_id**, các trường **in_campaign**, **campaign_id**, **last_id** không có mặt.
- Nếu trong trang này có push và push đó có cập nhật thông tin like/bình luận của bài viết trong trang này thì ta có cập nhật.



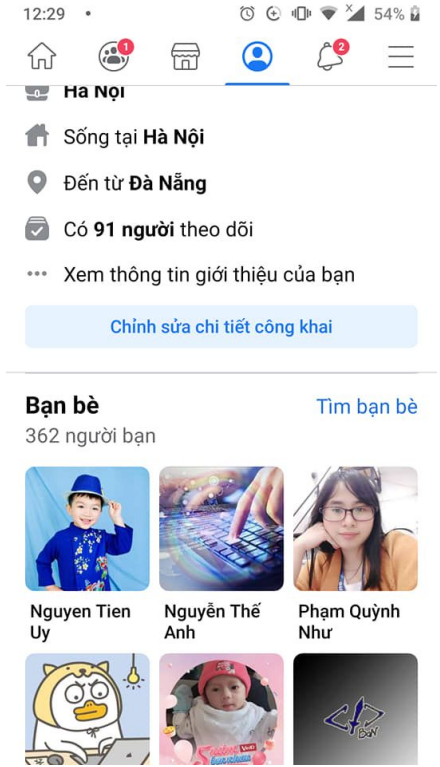
6. Giao diện trang người dùng (8)



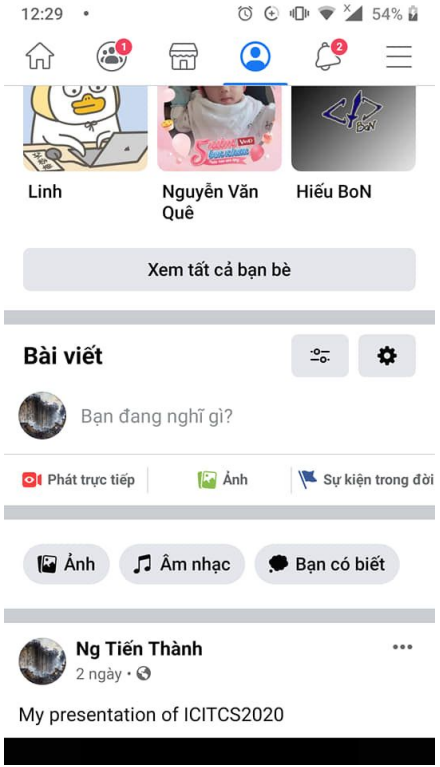
- Nếu trong trang này có push và push đó có cập nhật thông tin like/bình luận của bài viết trong trang này thì ta có cập nhật.



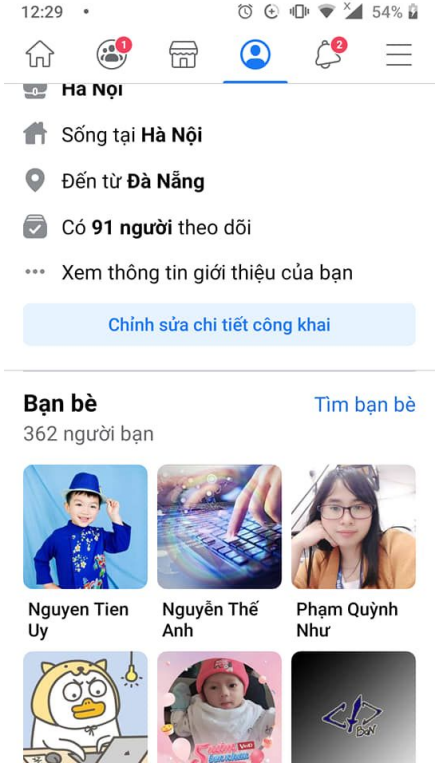
6. Giao diện trang người dùng (9)



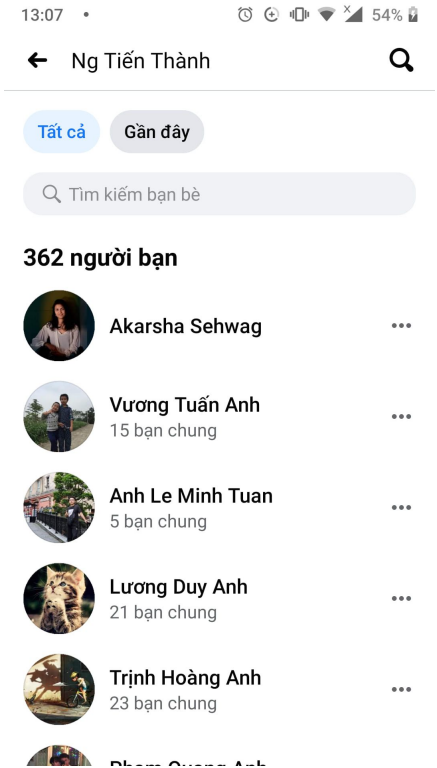
- Các bạn bè được sắp xếp theo trình tự thời gian kết bạn (mới nhất cho lên trước)
- "Tìm bạn bè" sẽ chuyển sang trang hiển thị các lời yêu cầu kết bạn từ người khác
- Bên cạnh **Bài viết** sẽ không có các nút cài đặt hoặc bộ lọc (bên phải), không có "Phát trực tiếp", "Ảnh", "Sự kiện trong đời", "Âm nhạc", "Bạn có biết"



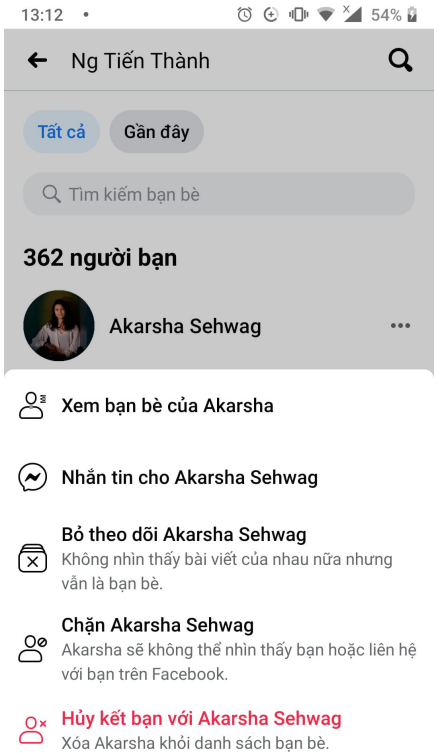
6. Giao diện trang người dùng (10)



- Nếu số bạn vượt 6 thì hiển thị 6 người mới nhất theo trình tự thời gian
- Khi xem tất cả bạn bè thì hiển thị theo thứ tự ABC.
- Khi nhấn vào nút ba chấm ở bên phải tên từng người bạn, sẽ hiển thị popup tùy chọn

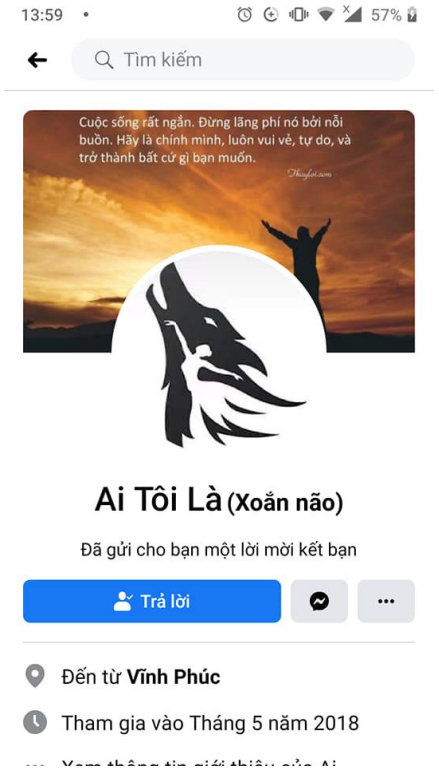


6. Giao diện trang người dùng (11)



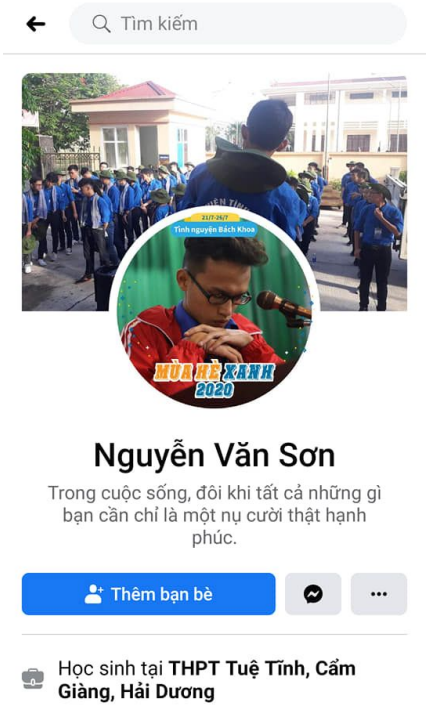
- Khi nhấn vào nút ba chấm ở bên phải tên từng người bạn, sẽ hiển thị popup tùy chọn
- Hỗ trợ "Xem bạn bè của X" sẽ chuyển sang trang hiển thị danh sách bạn bè của X
- Không hỗ trợ "Bỏ theo dõi X" mà thay vào đó là "Xem trang cá nhân của X"
- Các tùy chọn "Chặn…" và "Hủy…" vẫn được xây dựng.

6. Giao diện trang người dùng (12)



- Khi chuyển vào trang một người dùng (đã gửi lời mời kết bạn) thì sẽ có giao diện như bên trái:
  - Vẫn giữ nút "Trả lời" và nút cho phép nhắn tin
- Nhấn vào nút ba chấm sẽ chuyển sang các tùy chọn với trang cá nhân này
  - Bỏ qua thao tác "Tìm hỗ trợ…"
  - Có hỗ trợ Chặn
  - Các thao tác còn lại vẫn giữ như trang cá nhân của người bình thường

6. Giao diện trang người dùng (13)



Với một người dùng bình thường (không phải bạn bè và chưa có lời gửi kết bạn từ hai phía) sẽ hiển thị như hình bên trái

Còn một khi đã gửi yêu cầu thì sẽ có giao diện như hình bên phải.

HẾT TUẦN 10