

Document for notification service

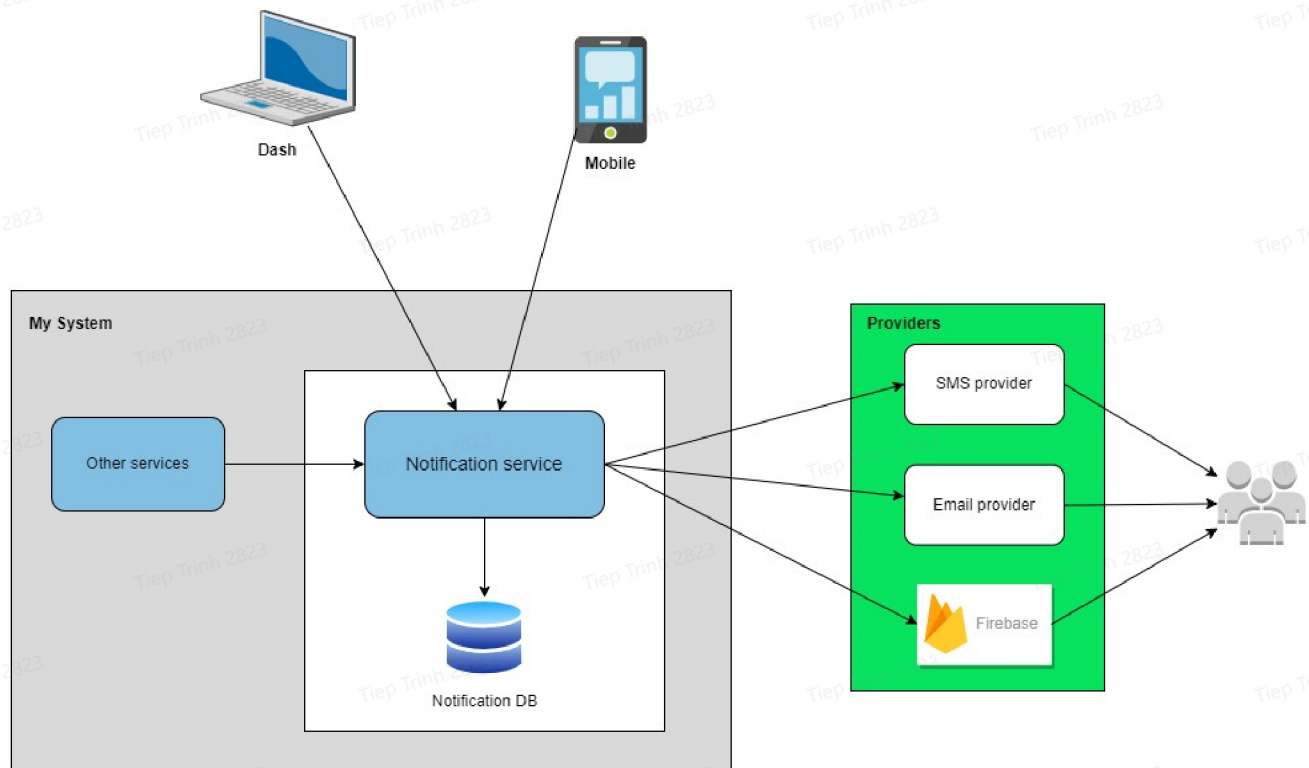
1. Functional requirement:

Hệ thống gửi thông báo đến end users

- Hệ thống có thể hỗ trợ gửi thông báo qua một số channels khác nhau như: Sms, email, ... Mỗi channel có thể có 1 số nhà cung cấp dịch vụ khác nhau
- Hệ thống hỗ trợ 3 API cơ bản:
 - send: 1 service hay bên nào đó sẽ gửi yêu cầu thông báo đến hệ thống, hệ thống sẽ triển khai và gửi đến end users
 - getList: ở phía client web app (dash) thì có thể gọi api này để lấy ra danh sách thông báo chứa các thông tin liên quan
 - getListForCustomer: ở phía client mobile app có thể gọi api này để lấy ra danh sách thông báo của end user.
- Ngoài ra, hệ thống cần phải có logging với đầy đủ thông tin: incoming traffic, outgoing traffic, important logic, traceID (chưa cần monitoring, chỉ cần logging cơ bản)

2. High level design:

Dưới đây là bản phác thảo về kiến trúc hệ thống và các luồng xử lý liên quan

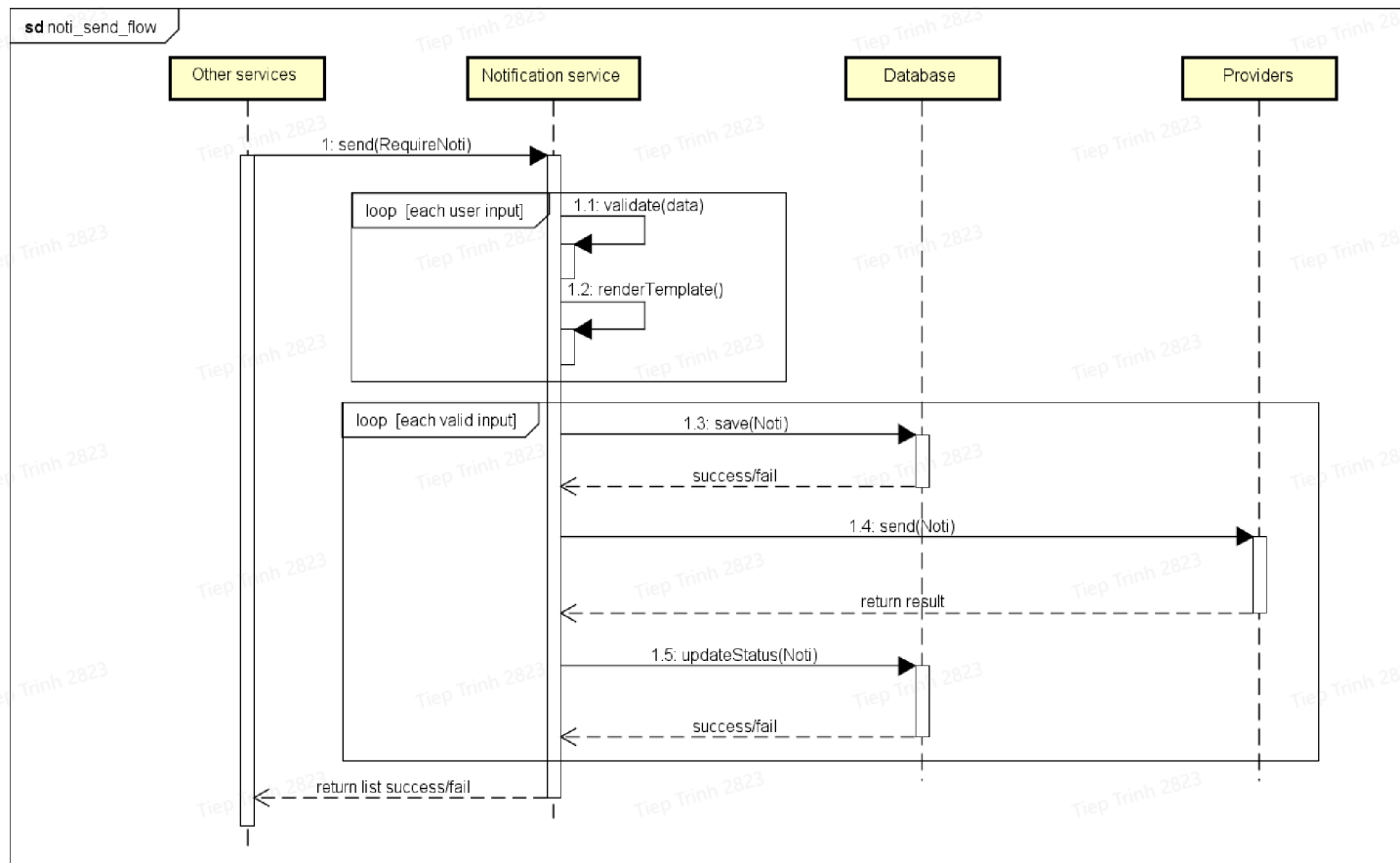


• Notification service:

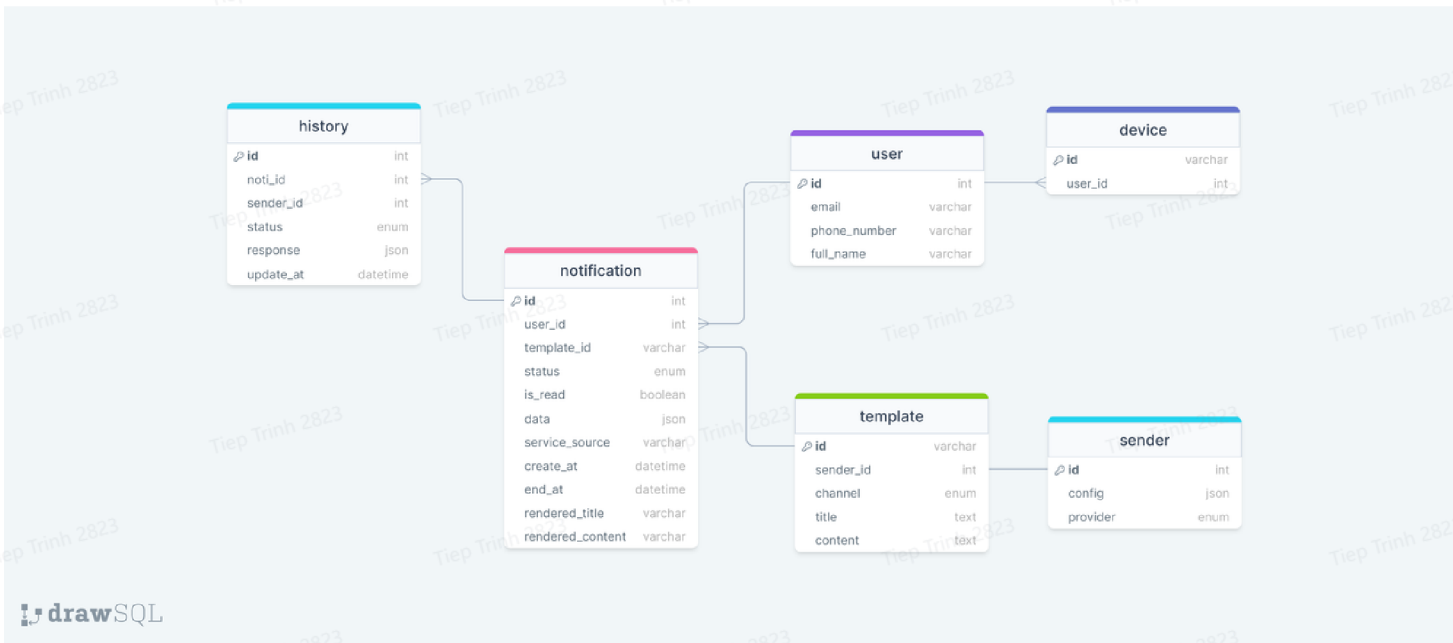
- Yêu cầu chức năng: quản lý thông báo, giúp các services khác trong hệ thống có thể gửi thông báo, lưu trữ lại thông báo giúp client (dash, mobile) có thể call API nhận về danh sách thông báo
- Yêu cầu phi chức năng: hệ thống đơn giản, giao tiếp đồng bộ, nhanh chóng giữa các services và bên providers
- Chú ý: hệ thống không được response các thông tin không phù hợp qua việc call API từ các đối tượng client khác nhau

- **Notification DB:** là nơi lưu trữ database liên quan đến thông báo. Ngoài ra còn có thêm một table cho thông tin để tương tác với user và các channels mà user đăng ký
- **Other services:** là một vài service nào đó trong hệ thống gửi yêu cầu send notification đến.

3. Data flow design



4. Database design



Thiết kế database, giải quyết các nhập nhằng, ràng buộc:

Một thông báo => gửi tới nhiều user => mỗi user có nhiều device

Một channel => có thể có nhiều providers => mỗi provider có thể hỗ trợ gửi các template khác nhau

Mô tả các bảng:

- Bảng **notification**: lưu trữ thông báo theo user

STT	Tên	Kiểu dữ liệu	Mô tả	Ghi chú
1	id	int	Primary key của bảng	
2	user_id	int	Khoá ngoại, xác định notification được gửi cho user nào	
3	template_id	varchar	Khoá ngoại, xác định template nào được áp dụng cho thông báo	
4	status	enum	trạng thái của thông báo	
5	isRead	boolean	trạng thái người dùng đã đọc chưa	
6	data	json	dữ liệu tương ứng với các biến trong template của thông báo	
7	service_source	varchar	tên service yêu cầu gửi đi	
8	create_at	datetime	thời gian thông báo được tạo	
9	end_at	datetime	thời gian việc gửi thông báo kết thúc (có thể kết thúc thành công/thất bại)	
10	rendered_title	text	Title của notification sau khi đã render, cái mà người dùng thực sự nhận được	
11	rendered_content	text	Content của notification sau khi đã render, cái mà người dùng thực sự nhận được	

- Bảng **sender**: lưu trữ thông tin về bên yêu cầu gửi thông báo

STT	Tên	Kiểu dữ liệu	Mô tả	Ghi chú
1	id	varchar	Primary key, lưu trữ id của sender	
2	config	json	thông tin cấu hình của sender	
3	provider	enum	xác định provider nào sẽ gửi đi	

- Bảng **user**: lưu trữ thông tin người dùng mà thông báo gửi đến

STT	Tên	Kiểu dữ liệu	Mô tả	Ghi chú
1	id	int	Primary key của bảng	
2	email	varchar	Thông tin về email của người dùng	
3	phone_number	varchar(11)	Số của người dùng	
4	full_name	varchar	Họ tên của người dùng	

- Bảng **device**: lưu trữ các device_id khác nhau của user

STT	Tên	Kiểu dữ liệu	Mô tả	Ghi chú
-----	-----	--------------	-------	---------

1	id	varchar	Id (token) của thiết bị	Do providers cung cấp
2	user_id	int	Khoá ngoại, Id của người dùng	

- Bảng **template**: lưu trữ các template cho thông báo

STT	Tên	Kiểu dữ liệu	Mô tả	Ghi chú
1	id	varchar	Primary key, id của thiết bị	
2	channel	enum	Thông tin channel của template	
3	form	text	định dạng form của thông báo	
4	sender_id	varchar	Khoá ngoại, xác định sender	

- Bảng **history**:

STT	Tên	Kiểu dữ liệu	Mô tả	Ghi chú
1	id	int	Primary key	
2	noti_id	int	Kháo ngoại, xác định đây là lịch sử của notification nào	
3	sender_id	varchar	xác định sender nào thực hiện bản thông báo này	
4	status	enum	trạng thái của thông báo	
5	response	json	kết quả trả về sau khi gửi	
6	update_at	datetime	thời gian thực hiện gửi thông báo	

5. API design

No	Description	Path	Method	Request	Response
1	Gửi thông báo	/api/v1/internal/notification/send	POST	Body: <pre>{ "to": [{ "user_id": 11, "data": [{ "key": "name", "value": "TiepTD" }] }], "template": "code_tp",</pre>	<pre>{ "code": "code", "message": "message", "result": [{ "user_id": "user_id", "status": "success/fail", "response": "response" }] }</pre>

				<pre> "service_source": "Payment Service" } </pre>	
2	Lấy danh sách thông báo từ trang dash	/api/v1/admin/notifica tion	GET	<pre> Params: page* size* user_id* channel* email* phone_number* notification_id* provider* sender_id* status* create_at* end_at* </pre>	<pre> { "code": "code", "message": "message", "result": [{ "content": { "id": 11, "user_id": "user_id", "template_id": "template_id", "sender_id": "sender_id", "status": "status", "channel": "channel", "provider": "provider", "is_read": false, "title": "title", "content": "content" "service_source": "Payment service", "create_at": time, "end_at": time }, "totalPages": 0, "totalElements": 0, }] } </pre>
3	Lấy danh sách thông báo cho một người dùng từ mobile app	/api/v1/client/notificati on	GET	<pre> Params: page* size* is_read* end_at* </pre>	<pre> { "code": "code", "message": "message", "result": [{ "content": { "title": "title", "content": "content", "is_read": true, "end_at": "date_time", "metadata": "metadata" }, "totalPages": 0, "totalElements": 0, }] } </pre>

