**MỤC LỤC**

[PHẦN 1: GIỚI THIỆU 3](#_Toc443578492)

[PHẦN 2: MÔ HÌNH KIẾN TRÚC TỔNG THỂ 4](#_Toc443578493)

[I. Mô hình kiến trúc hệ thống SSO 4](#_Toc443578494)

[1. Mô hình triển khai hệ thống 4](#_Toc443578495)

[2. Mô hình kiến trúc logic 5](#_Toc443578496)

[II. Lựa chọn công nghệ 6](#_Toc443578497)

[1. Mã nguồn mở Jasig Cas 6](#_Toc443578498)

[2. Memcache 8](#_Toc443578499)

[PHẦN 3: THIẾT KẾ NGHIỆP VỤ HỆ THỐNG SSO 11](#_Toc443578500)

[I. Xác thực người dùng 11](#_Toc443578501)

[1. Dịch vụ xác thực trên giao diện web 11](#_Toc443578502)

[2. Xác thực người dùng thông qua webservice 12](#_Toc443578503)

[3. Tích hợp ứng dụng với SSO 12](#_Toc443578504)

[4. Chức năng đổi mật khẩu khi quên mật khẩu 13](#_Toc443578505)

[5. Chức năng đổi mật khẩu 15](#_Toc443578506)

[6. Chính sách mật khẩu mạnh 16](#_Toc443578507)

[II. Cấp phát tài khoản người dùng 17](#_Toc443578508)

[III. Quản lý hệ thống 17](#_Toc443578509)

[1. Quản lý log 17](#_Toc443578510)

[2. Quản lý người dùng 18](#_Toc443578511)

[3. Quản lý Ứng dụng 19](#_Toc443578512)

[PHẦN 4: TRIỂN KHAI THỰC TẾ 21](#_Toc443578513)

[Danh sách ứng dụng CNTT đã tích hợp với hệ thống SSO 21](#_Toc443578514)

[Tăng trưởng người dùng của hệ thống SSO 23](#_Toc443578515)

# PHẦN 1: GIỚI THIỆU

Hiện nay Viettel đã và đang triển khai nhiều hệ thống Công nghệ thông tin phục vụ điều hành sản xuất kinh doanh, tuy nhiên các hệ thống chưa liên thông tích hợp cơ sở dữ liệu người dùng (User Database). Mỗi hệ thống CNTT có cơ sở dữ liệu người dùng riêng, cấu trúc tài khoản đăng nhập và chính sách mật khẩu không đồng nhất nên tồn tại một số bất cập như:

* + Cán bộ công nhân viên (CBCNV) phải nhớ rất nhiều tên tài khoản/mật khẩu, địa chỉ ứng dụng (url) khác nhau cho mỗi hệ thống CNTT
  + Hàng ngày các CBCNV phải thao tác đăng nhập lặp đi lặp lại nhiều lần khi sử dụng các ứng dụng khác nhau nên bất tiện, tốn thời gian.
  + Công tác quản trị người dùng, triển khai chính sách an toàn thông tin không đồng bộ do tồn tại nhiều hệ thống xác thực rải rác ở các đơn vị trong Tập đoàn.
  + Tốn nguồn lực phát triển chức năng xác thực người dùng cho mỗi ứng dụng.

Để khắc phục bất cập, Phòng Công nghệ thông tin phối hợp Trung tâm phần mềm 02 Viettel nghiên cứu, xây dựng hệ thống xác thực tập trung Single Sign On Tập đoàn, cung cấp giao diện xác thực người dùng duy nhất, đồng bộ cho tất cả hệ thống Công nghệ thông tin (CNTT) tại Tập đoàn. Hệ thống cho phép mỗi CBCNV chỉ phải nhớ và dùng một tài khoản duy nhất để đăng nhập sử dụng các ứng dụng CNTT khác nhau, đồng thời là công cụ quản lý, cấp phát người dùng tự động cho toàn bộ hệ thống CNTT tại Tập đoàn.

Hệ thống xác thực Tập trung Single Sign On Tập đoàn do đội ngũ kỹ sư Viettel làm chủ công nghệ hoàn toàn, sử dụng phần mềm nguồn mở Jasig CAS kết hợp với công nghệ xử lý dữ liệu trong bộ nhớ trong (Data-in-memory) đảm bảo hiệu năng phục vụ hoạt động thường nhật của toàn bộ CBCNV Tập đoàn.

# PHẦN 2: MÔ HÌNH KIẾN TRÚC TỔNG THỂ

## I. Mô hình kiến trúc hệ thống SSO

### Mô hình triển khai hệ thống



*Hình 1: kiến trúc mô hình vật lý hệ thống SSO*

Hệ thống SSO bao gồm các module

* Load balancer: Cân bằng tải hệ thống
* SSO App: Bao gồm 4 server vật lý, chứa các webserver app chịu trách nhiệm xử lý toàn bộ các request từ phía người dùng.
* Memcached: Bao gồm 2 server memcached, lưu trữ bản tin ticket.
* Database: Lưu trữ dữ liệu xuống Storage, sử dụng công nghệ Oracle Real Application Clusters (RAC)
* Storage: Lưu trữ dữ liệu, sao lưu định kỳ.

### Mô hình kiến trúc logic



*Hình 2: kiến trúc mô hình logic của hệ thống SSO*

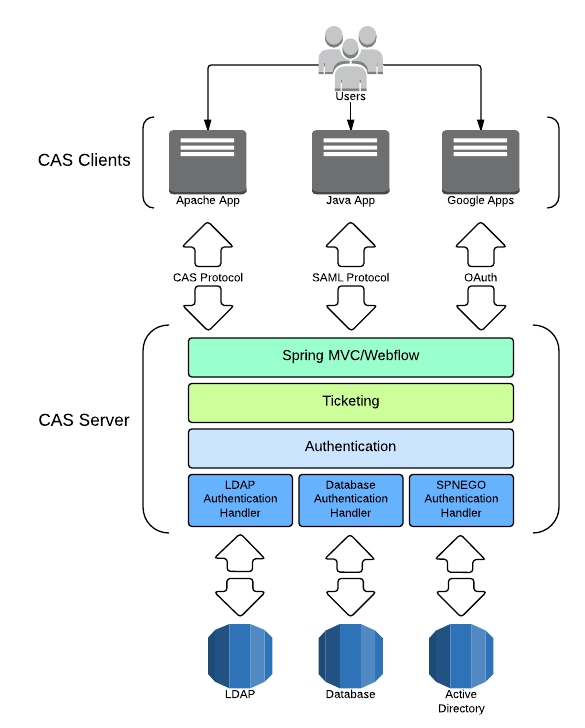
* Bước 1: Browser gửi request yêu cầu truy cập vào hệ thống ứng dụng (applcation server)
* Bước 2:
  + - * 2.1: Ứng dụng Redirect sang hệ thống SSO sử dụng Jasig-Cas, hiển thị cho người dùng form đăng nhập
      * 2.2: Người dùng nhập thông tin user name và mật khẩu
* Bước 3: Hệ thống SSO xác thực người dùng thành công, sinh ra GTC và ST.
* Bước 4: Hệ thống SSO lưu lại GTC và ST vào server memcached
* Bước 5: Hệ thống SSO trả lại GTC và ST cho client.
* Bước 6: Client lưu lại GTC tại local
* Bước 7: Gửi ST cho server ứng dụng
* Bước 8: Gửi ST cho SSO để SSO xác thực ST client gửi lên với ST lưu trong memcached
* Bước 9: SSO xác thực chính xác ST thì hủy ST trong memcached, gửi trả lại thông tin người dùng cho ứng dụng
* Bước 10: Ứng dụng gửi thông tin người dùng đã xác thực lên VSA(hoặc 1 hệ thống phân quyền khác) để lấy thông tin về role, phòng ban, list object menu…
* Bước 11:VSA (hoặc 1 hệ thống phân quyền khác) trả lại thông tin về role, phòng ban, list object menu… cho ứng dụng
* Bước 12: Ứng dụng trả lại thông tin tài nguyên cho người dùng

## II. Lựa chọn công nghệ

### Mã nguồn mở Jasig Cas

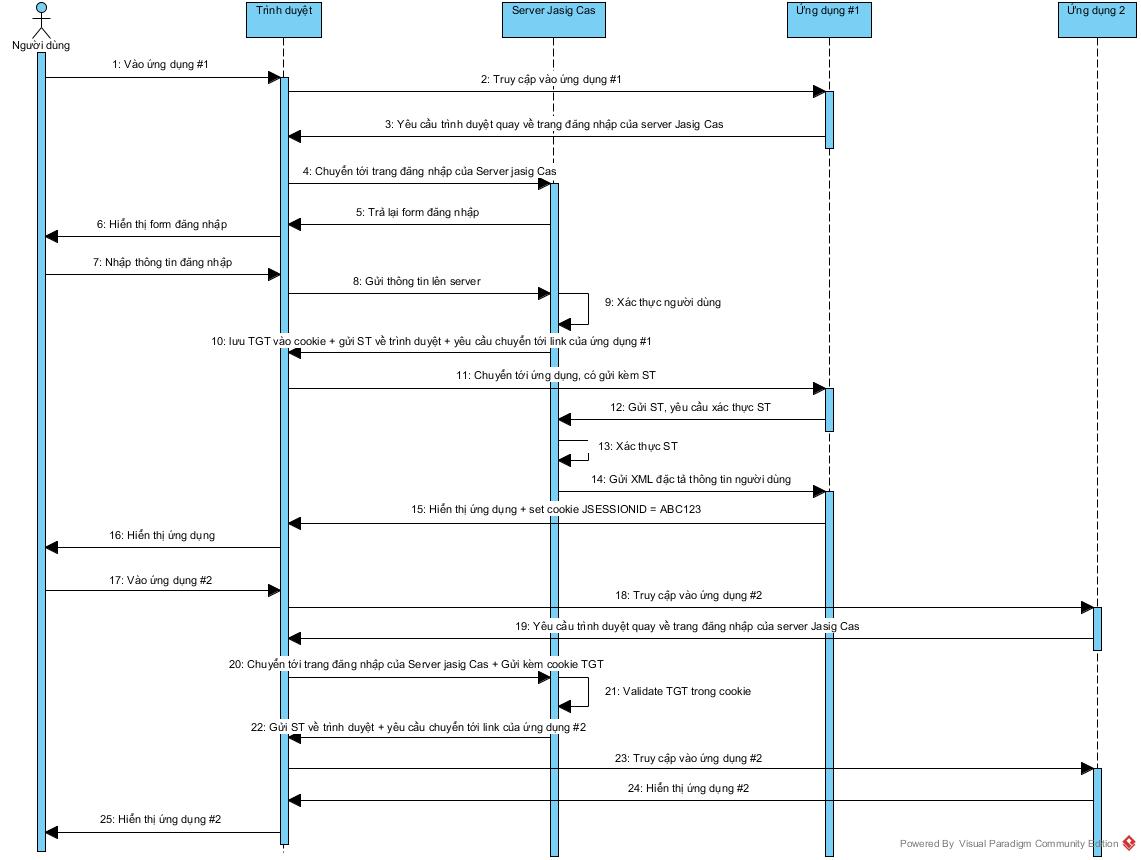
1. ***Tổng quan về Jasig Cas***

* JA-SIG Central Authentication Service là một mã nguồn mở, cung cấp giải pháp đăng nhập một lần (Single Sign On), hỗ trợ nhiều ngôn ngữ thông dụng như PHP, Java, C#...
* Dự án ban đầu có tên là Central Authentication Service (CAS), bắt đầu phát triển từ năm 2001 tại đại học Yale, người đầu tiên phát triển dự án này là Shawn Bayern.
* Năm 2002, đại học Yale công bố phiên bản CAS 2.0
* Vào tháng 9/2004, đại học Yale và Rutgers, Đại học bang New Jersey đã tổ chức các cuộc thảo luận về một phiên bản mới của CAS. Trong tháng mười hai năm 2005, CAS đã chính thức có tên gọi mới, trở thành dự án Jasig CAS. Mặc dù không thay đổi các giao thức, nhưng CAS 3.0 đã giới thiệu một kiến trúc hoàn toàn mới, cho phép đơn giản hóa việc triển khai và tích hợp vào các máy chủ CAS mà không cần phải thay đổi mã nguồn. Chính khả năng mới này đã giúp CAS được biết đến nhiều hơn trong giới lập trình. Tới khi phát hành chính thức vào tháng 6/2005, kiến trúc của CAS không thay đổi nhiều đã cho phép gia tăng nhanh chóng các tiện ích và tính năng mới.
* Kiến trúc hệ thống Jagis CAS



1. ***Giải pháp của mã nguồn mở Jasig Cas***

* Jasig Cas sẽ tạo ra trong mỗi lần đăng nhập thành công của người dùng 2 loại ticket: Service ticket (ST) và Ticket Grand Ticket (TGT)
* Service ticket sẽ lưu trữ thông tin đăng nhập của người dùng vào một ứng dụng cụ thể. Trong khi đó Ticket Grand Ticket sẽ được lưu vào cookie của trình duyệt để lưu lại thông tin của cả phiên đăng nhập.
* Khi người dùng đăng nhập vào một ứng dụng, Jasig Cas sẽ tạo ra ST tương ứng với ứng dụng đó và một TGT tương ứng với cả phiên đăng nhập.
* Khi người dùng chuyển sang sử dụng một ứng dụng khác, ứng dụng sẽ redirect tới server Jasig Cas đồng thời gửi kèm TGT đã sinh ở lần đăng nhập ứng dụng trước. Khi nhận thấy có TGT trong request gửi tới, server Jasig Cas sẽ bỏ qua xử lý đăng nhập vào tạo mới một ST tương ứng với phiên đăng nhập của ứng dụng mới.



*Hình 3: sơ đồ tuần tự luồng xử lý của Jasig Cas*

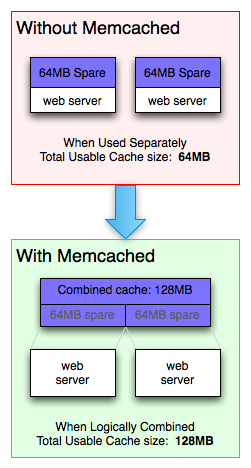
### Memcache

1. ***Tổng quan về memcached***

* Được phát triển bởi công ty Danga Interactive. Lúc đầu được sử dụng trong các server chạy bằng PHP, nhưng ngày nay memcached đã hiện hữu và được sử dụng trong rất nhiều các loại máy chủ khác.
* Memcached được viết bằng ngôn ngữ C++
* Memcached là một công nghệ cho phép tăng cao khả năng lưu trữ và truy xuất dữ liệu.
* Memcached sử dụng công nghệ data in memory, sử dụng chính bộ nhớ đệm của hệ thống để tiến hành lưu trữ dữ liệu

1. ***Hiệu quả khi sử dụng memcached***

* Memcached đạt hiệu suất rất cao trong truy xuất và lưu trữ dữ liệu có kích thước nhỏ nhưng đòi hỏi việc truy xuất diễn ra liên tục với tần suất lớn.
* Thay vì truy xuất và lưu trữ dữ liệu trên ổ đĩa cứng, memcached sử dụng chính bộ nhớ đệm của hệ thống (RAM) để lưu trữ dữ liệu. Do tốc độ đọc ghi của RAM lớn hơn rất nhiều so với trên ổ cứng nên memcached có tốc độ đọc ghi rất cao
* Cho phép nhiều ứng dụng đồng thời truy xuất. Tất cả các ứng dụng có thể chia sẻ dữ liệu với nhau thông qua memcached thay vì chỉ có thể sử dụng bộ nhớ của riêng mình.



*Hình 4: memcached cho phép ứng dụng chia sẻ dữ liệu với nhau*

* Dễ triển khai, tiết kiệm chi phí, nâng cao tính đồng bộ dữ liệu giữa các server ứng dụng. Bảng chi tiết đánh giá tính hiệu quả của memcached so với các phương pháp chia sẽ dữ liệu khác.

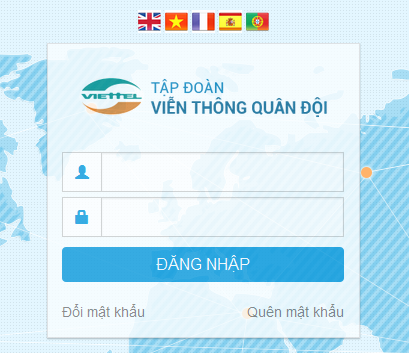
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Lưu trữ trên server CAS** | **Lưu trữ trên database** | **Lưu trữ trên memcached** |
| *Giải pháp* | Khi lưu ticket trên 1 server, xây dựng tiến trình đổng bộ sang các server khác | Sử dụng giải pháp HA của hãng (RAC với Oracle, Cluster với MySQL …v.v) | Khi lưu trữ ticket, CAS lưu lên tất cả các server memcached. Khi đọc ticket đọc từng server cho đến khi lấy thành công ticket |
| *Ưu điểm* | Không phải triển khai thêm server lưu trữ |  | - Triển khai đơn giản  - Không bị mất đồng bộ |
| *Nhược điểm* | - Chi phí cho việc đồng bộ ticket lớn - Xảy ra tình trạng mất đồng bộ | - Triển khai phức tạp  - Không đúng bản chất lưu ticket, không tự hủy ticket khi expired 🡪 Phải xử lý xóa ticket thủ công |  |

# PHẦN 3: THIẾT KẾ NGHIỆP VỤ HỆ THỐNG SSO

## I. Xác thực người dùng

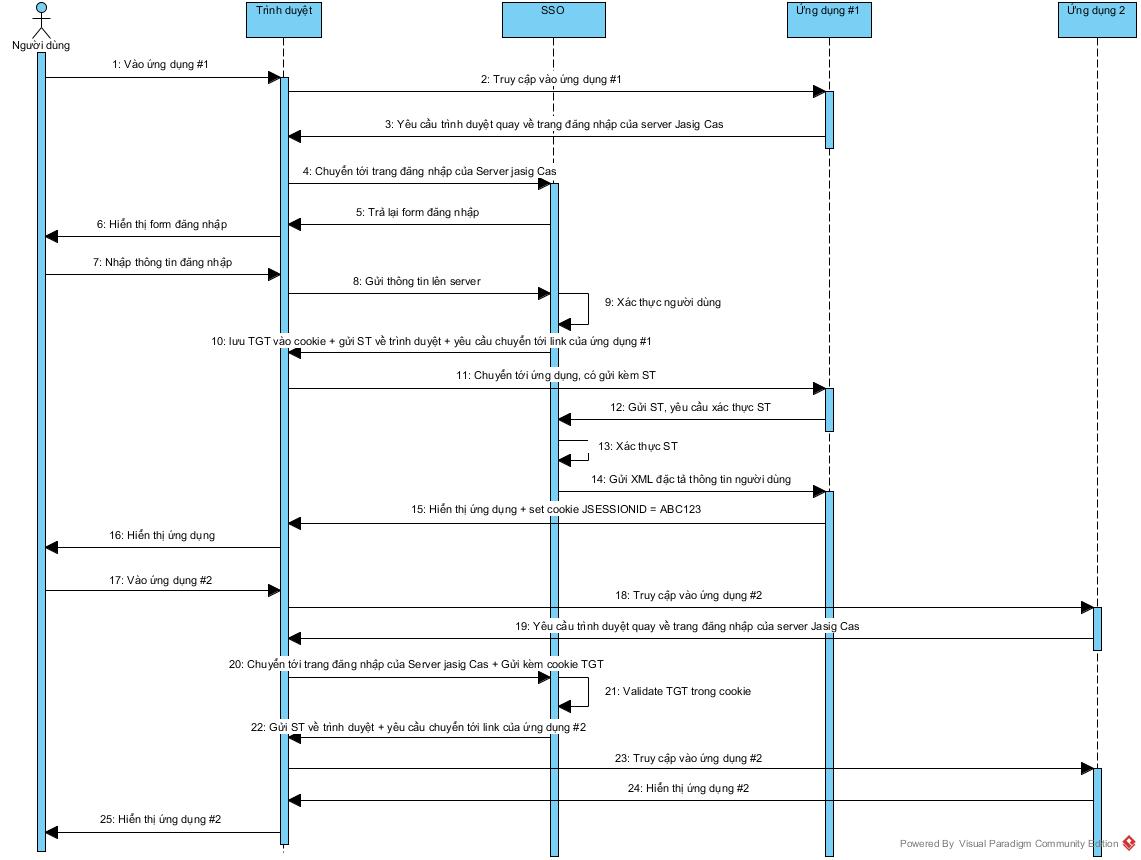
### Dịch vụ xác thực trên giao diện web

* Giao diện web xác thực của hệ thống SSO



*Hình 5: Giao diện đăng nhập của hệ thống SSO*

* Sơ đồ đặc tả luồng xử lý xác thực trên giao diện web



* Các bước thực hiện như sau
  + Người dùng nhập link ứng dụng muốn truy cập
  + Trình duyệt truy cập vào ứng dụng
  + Ứng dụng sẽ check người dùng đã đăng nhập chưa, nếu chưa sẽ yêu cầu trình duyệt chuyển sang trang đăng nhập của SSO
  + SSO sẽ hiển thị màn hình đăng nhập, người dùng nhập thông tin tài khoản và mật khẩu
  + SSO sẽ xác thực người dùng, sinh ra service ticket (ST) và ticket grand ticket (TGT). TGT sẽ lưu vào trong cookie của trình duyệt
  + Trình duyệt sẽ quay trở lại ứng dụng, gửi kèm ST lên cho ứng dụng
  + Ứng dụng sẽ gửi ST sang SSO, yêu cầu xác thực ST trình duyệt gửi lên
  + SSO tiến hành xác thực ST, nếu chính xác, sẽ gửi lại ứng dụng XML đặc tả thông tin người dùng
  + Ứng dụng nhận được thông tin XML gửi về sẽ cho hiển thị ứng dụng trên trình duyệt

### Xác thực người dùng thông qua webservice

SSO cung cấp cho các ứng dụng desktop khả năng xác thực thông qua webservcie.

* Các ứng dụng có thể kết nối vào 1 trong 3 link webservice bất kỳ của SSO bao gồm

http://10.60.7.126:8660/passportv3/passportWS?wsdl

http://10.60.7.126:8661/passportv3/passportWS?wsdl

http://10.60.7.126:8662/passportv3/passportWS?wsdl

* Đặc tả webservice của hệ thống SSO

|  |  |  |
| --- | --- | --- |
| Method | Param input | Response output |
| Authen | - String userName: tên đăng nhập - String password: mật khẩu - String domainCode: mã ứng dụng muốn truy cập | Response response: + ErrorCode: mã lỗi (1 là có lỗi đi kèm mô tả, 0 là không có lỗi) + Desciption: mô tả lỗi |

### Tích hợp ứng dụng với SSO

1. ***Các ứng dụng web đang sử dụng VSA của VTT và VTNET***

* Thay đổi link Url passport trong file ../scr/java/cas.properties hoặc ../scr/java/cas\_en\_US.properties

loginUrl=http://10.60.7.126:8660/passportv3/login

logoutUrl=http://10.60.7.126:8660/passportv3/logout

validateUrl=http://10.60.7.126:8660/passportv3/validate

passportServiceUrl=http://10.60.7.126:8660/passportv3/passportWS?wsdl

1. ***Các ứng dụng sử dụng Java, đã tích hợp hệ thống VSA Passport nhưng không thuộc VTT và VTNet***

* Bước 1: Chỉnh sửa file ../scr/java/cas\_en\_US.properties.

Thay đổi link passport-SSO Url về địa chỉ của server SSO

loginUrl=http://10.60.7.126:8660/passportv3/login

logoutUrl=http://10.60.7.126:8660/passportv3/logout

validateUrl=http://10.60.7.126:8660/passportv3/validate

passportServiceUrl=http://10.60.7.126:8660/passportv3/passportWS?wsdl

* Bước 2: Thêm mới filter, bổ sung sử lý lấy thông tin quyền của tài khoản vào giá trị UserToken mà SSO trả về

1. ***Các ứng dụng desktop***

* Sử dụng xác thực qua webservice

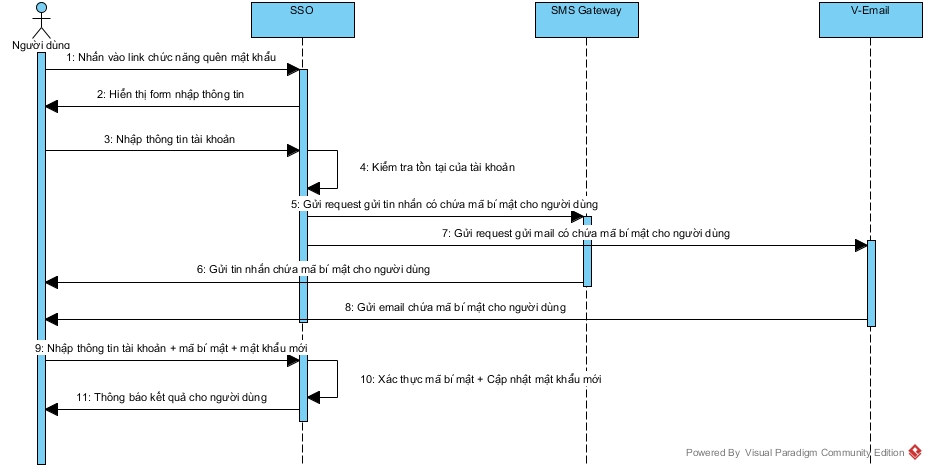
### Chức năng đổi mật khẩu khi quên mật khẩu

* Giao diện web chức năng quên mật khẩu



*Hình 6: Giao diện chức năng quên mật khẩu của hệ thống SSO*

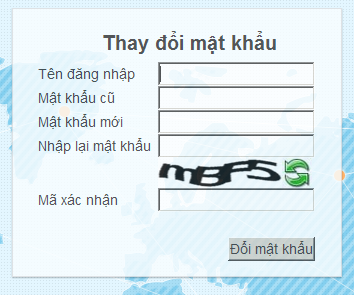
* Sơ đồ đặc tả luồng xử lý chức năng quên mật khẩu



* Các bước thực hiện như sau:
  + Người dùng nhấn vào link chức năng quên mật khẩu
  + SSO sẽ trả lại form nhập thông tin cho người dùng
  + Người dùng nhập thông tin tên tài khoản
  + SSO sẽ tiến hành kiểm tra trên hệ thống có tồn tại tên tài khoản tương ứng không
  + Nếu chính xác, SSO sẽ gửi request tới SMSGateway và V-email, để gửi mã bí mật đổi mật khẩu tới người dùng thông qua SMS và email
  + Người dùng nhận được email, sẽ tiến hành nhập mã bí mật vào form thông tin
  + SSO xác thực lại mã bí mật, nếu chính xác sẽ tiến hành đổi mật khẩu
  + SSO thông báo lại kết quả cho người dùng đã cập nhật thành công

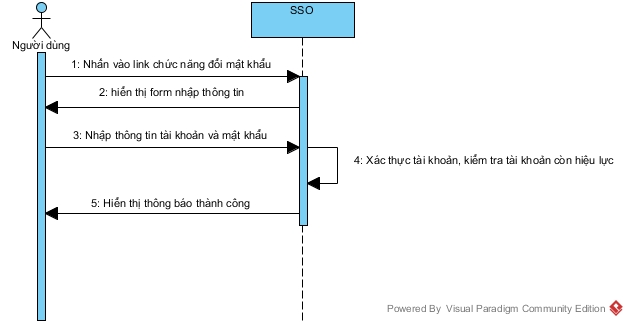
### Chức năng đổi mật khẩu

* Giao diện web chức năng đổi mật khẩu



*Hình 6: Giao diện chức năng quên mật khẩu của hệ thống SSO*

* Sơ đồ đặc tả luồng xử lý chức năng đổi mật khẩu trên SSO



* Các bước thực hiện như sau
  + Người dùng nhấn vào link chức năng đổi mật khẩu
  + SSO sẽ hiển thị form nhập thông tin cho người dùng
  + Người dùng nhập thông tin tài khoản, mật khẩu cũ và mật khẩu mới
  + SSO xác thực tài khoản, kiểm tra dữ liệu trong form người dùng nhập
  + SSO hiển thị thông báo đổi mật khẩu thành công

### Chính sách mật khẩu mạnh

Người dùng sử dụng hệ thống SSO phải tuân thủ nguyên tắc mật khẩu mạnh của tập đoàn. Bao gồm các nguyên tắc sau

* Phải chứa ký tự, chữ số, chữ in hoa, ký tự đặc biệt: ~ ! @ # $ % ^ & \* ( ) \_ + | ` - = { } [ ] : " ; '<>?,./
* Độ dài tối thiểu là 8
* Không chứa tên tài khoản và bắt đầu bằng 123, 456, 789, 147, 258, 852, 741, abc, 369
* Không chứa các cụm từ dễ đoán: qwerty, password, passw0rd, password@123, abc123, iloveyou, viettel@123, admin@123, 123qwea@, abc@123, qwerty@123, vtt@2014, 123@123, 123123, 696969
* Không chứa thông tin cá nhân: họ và tên, ngày tháng năm sinh, số điện thoại, CMT, mã nhân viên
* Không chứa các dãy 4 ký tự tăng dần hoặc giảm dần: 1234, 4321, 6789, abcd...

## II. Cấp phát tài khoản người dùng

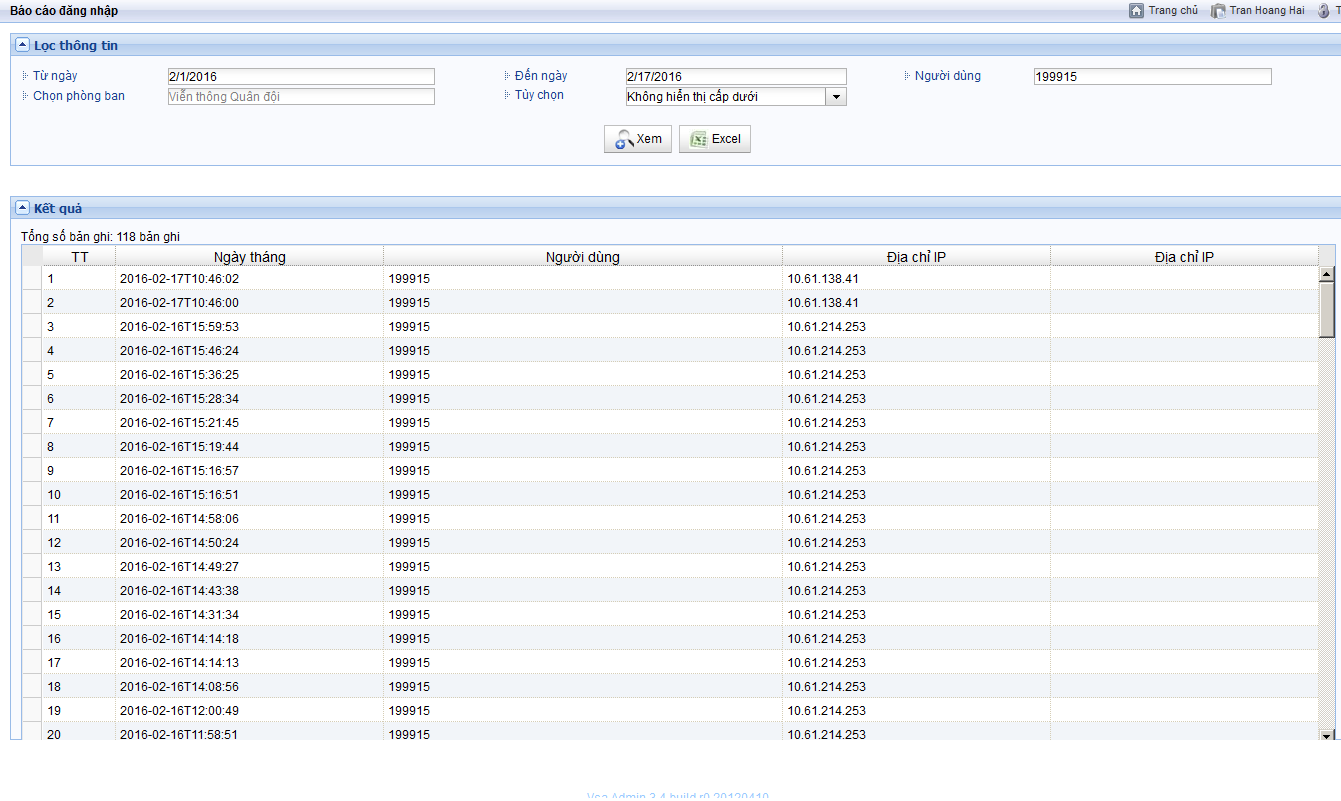
* Mỗi người dùng Viettel sẽ chỉ có một tài khoản duy nhất để truy cập toàn bộ tất cả các hệ thống công nghệ thông tin.
* Việc cấp phát tài khoản cho người hoàn toàn tự động và được đồng bộ định ký với hệ thống quản lý nhân sự VHR
* Định kỳ vào 2h sáng mỗi ngày, một tiến trình sẽ cập nhật lại thông tin người dùng
* Với những người dùng mới, tiến trình sẽ tạo mới tài khoản người dùng, gửi tin nhắn thông báo mật khẩu
* Với người dùng nghỉ việc, tiến trình sẽ cập nhật khóa người dùng
* Với người dùng có sự thay đổi thông tin về chức danh, phòng ban, thông tin cá nhân, tiến trình cũng tự động cập nhật lại để trùng khớp thông tin với hệ thống quản lý nhân sự VHR



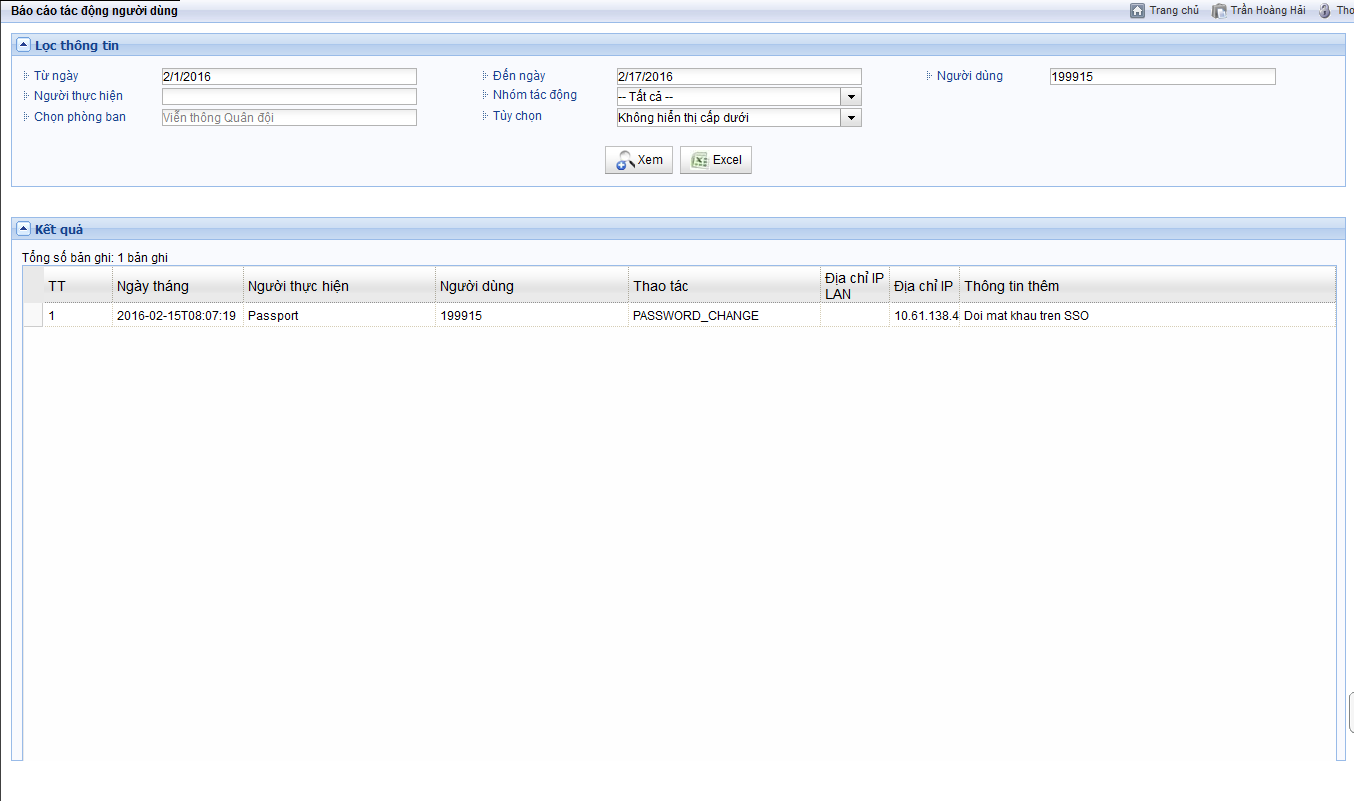
## III. Quản lý hệ thống

### Quản lý log

* Hệ thống SSO quản lý chặt chẽ và lưu lại log đăng nhập cũng như tất cả log tất cả tác động đối với tài khoản người dùng
* Cho phép hiển thị, tìm kiếm xuất dữ liệu trên ứng dụng quản lý
* Giao diện tìm kiếm tra cứu log đăng nhập của người dùng



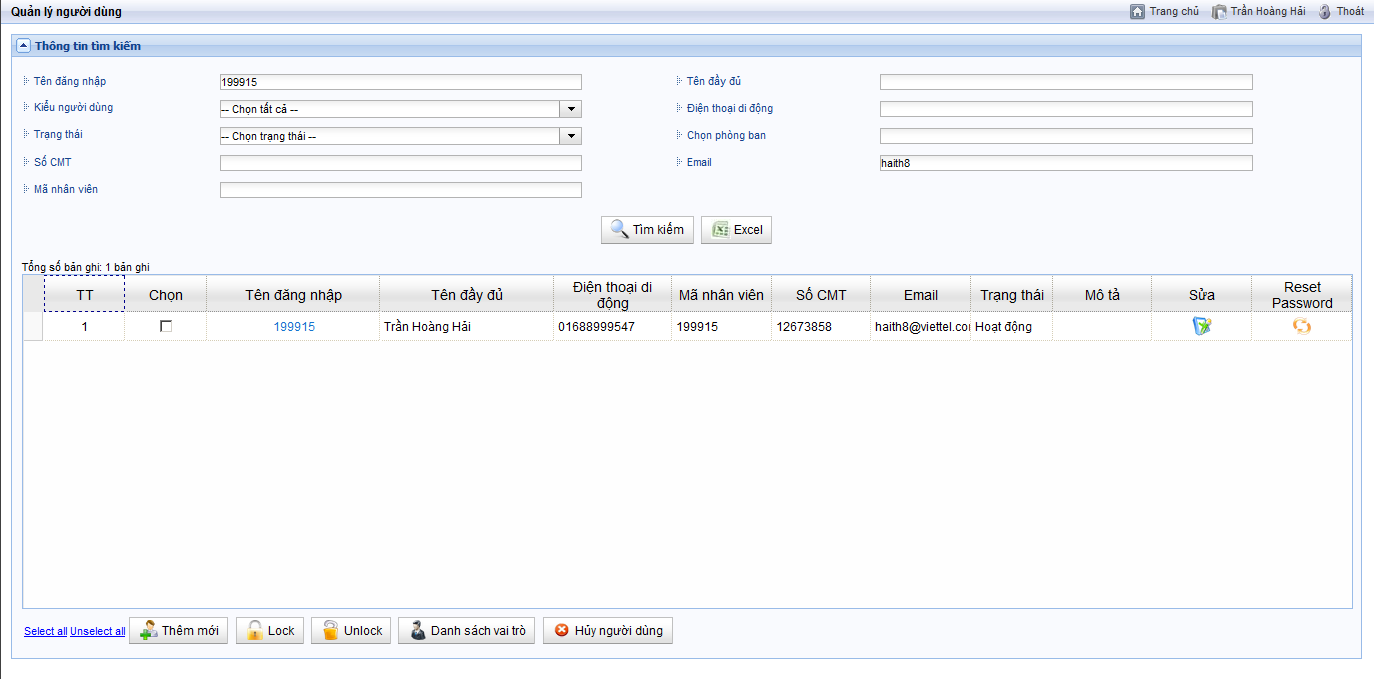
* Giao diện tìm kiếm tra cứu log tác động của ngươi dùng



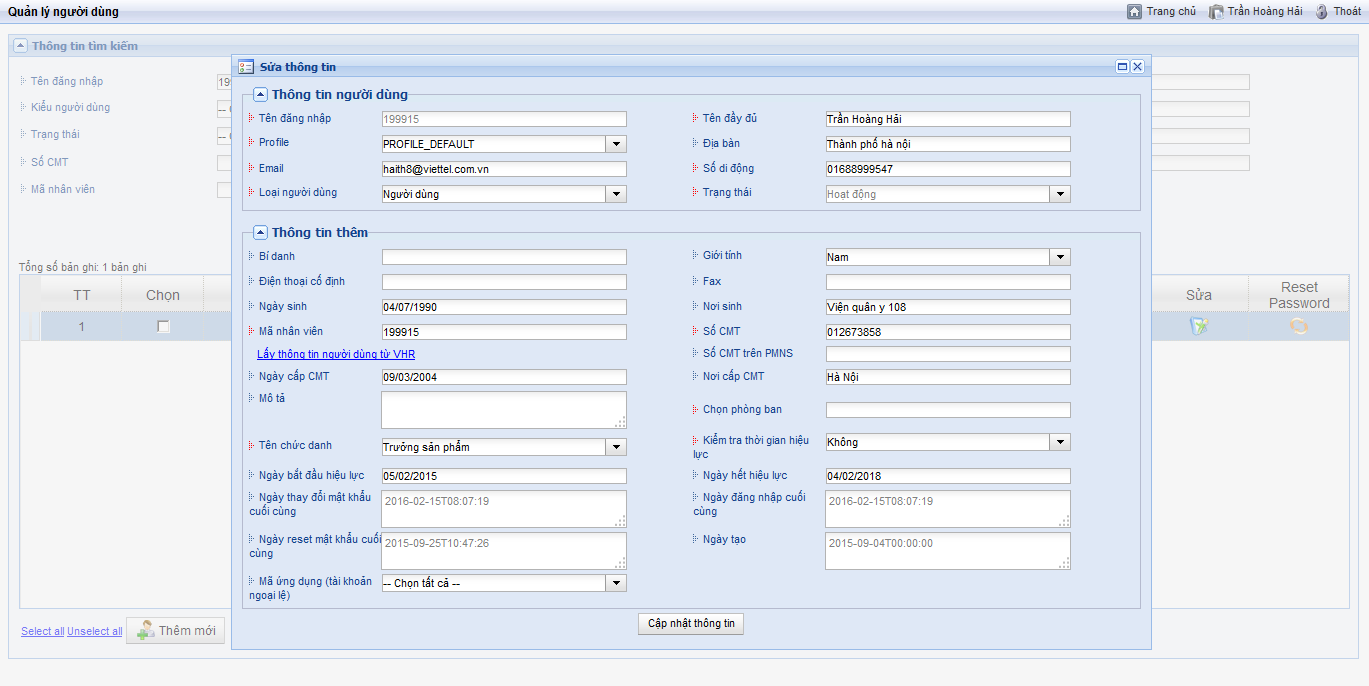
### Quản lý người dùng

Hệ thống SSO cho phép quản trị viên có thể tìm kiếm, cập nhật, khóa, mở khóa người dùng trên giao diện web

* Giao diện tìm kiếm tra cứu thông tin người dùng.

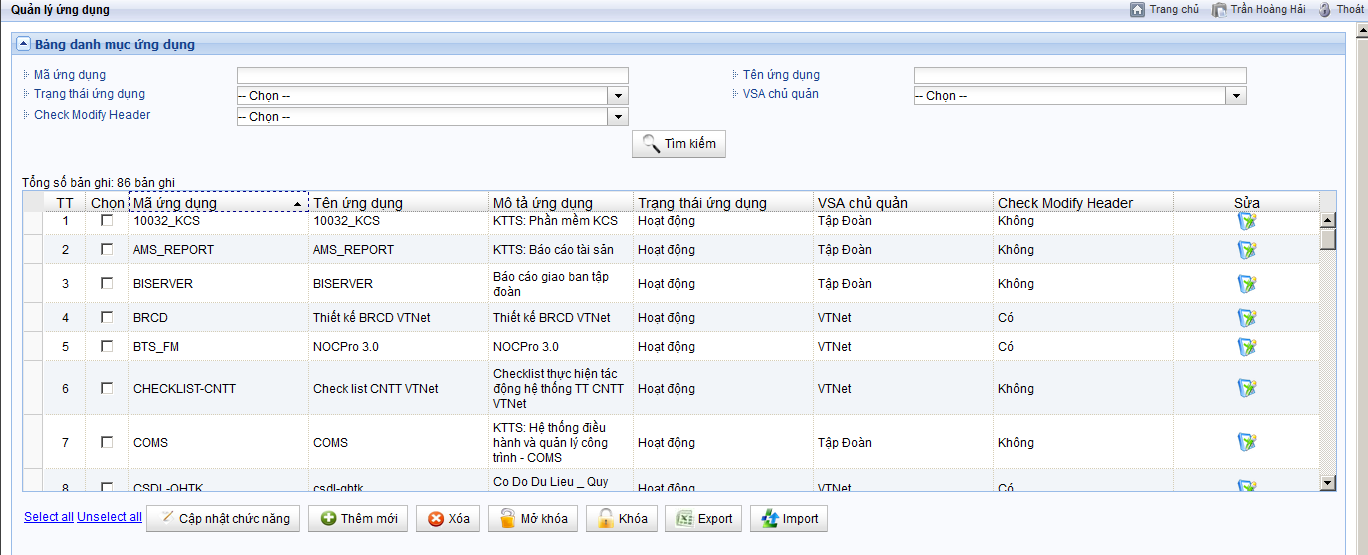


* Giao diện thay đổi thông tin người dùng

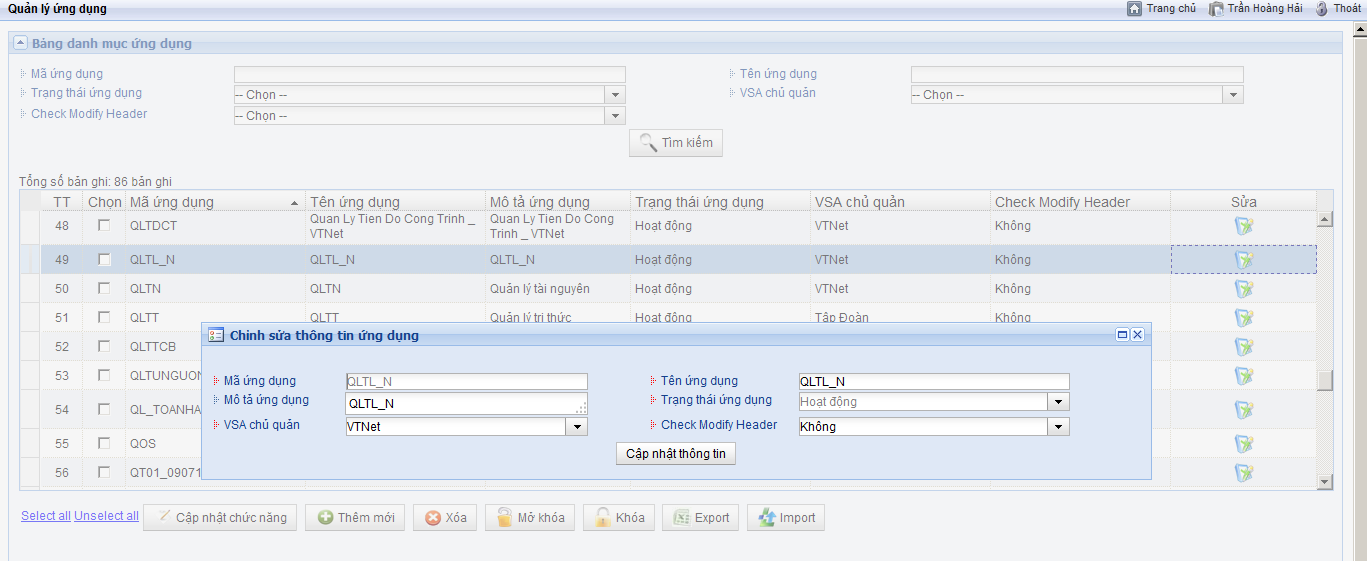


### Quản lý Ứng dụng

* Tất cả các ứng dụng muốn tích hợp với SSO đều phải khai báo trên hệ thống SSO.
* Quản trị viên hệ thống hoàn toàn có thể kiểm soát việc cho phép hoặc không cho phép ứng dụng được tích hợp với hệ thống SSO
* Giao diện tìm kiếm ứng dụng đang tích hợp trên SSO



* Giao diện cập nhật thông tin, thêm mới ứng dụng trên SSO



* Thông qua giao diện quản lý ứng dụng, quản trị viên cũng có thể biết được ứng dụng thuộc về đơn vị VTT hay VTNet.

# PHẦN 4: TRIỂN KHAI THỰC TẾ

Hệ thống SSO chính thức triển khai từ tháng 09/2015. Sau khi triển khai đã thay đổi được thói quen người dùng, tăng tính tiện lợi, đơn giản và tiết kiệm thời gian của người dùng cuối khi thực hiện thao tác trên hệ thống CNTT của tập đoàn.

Đơn giản hóa được việc quản lý người dùng trên toàn bộ hệ thống CNTT của tập đoàn. Mọi tác vụ cấp mới, cập nhật thông tin hoàn toàn tự động và luôn đồng bộ với hệ thống quản lý nhân sự.

Tính tới thời điểm tháng 01/2016 hệ thống SSO đã tích hợp 55 hệ thống CNTT, lượng người dùng sử dụng luôn đạt trên 50.000 người.

## Danh sách ứng dụng CNTT đã tích hợp với hệ thống SSO

|  |  |
| --- | --- |
| STT | Tên hệ thống |
| 1 | QLCR Mobile |
| 2 | Nocpro 4.0 |
| 3 | Nocpro 3 |
| 4 | NQT (Test End2End) |
| 5 | Gatepro 3.0 |
| 6 | Quản lý tài nguyên |
| 7 | Điều hành Kỹ thuật/SXKD |
| 8 | VSA/Passport VTNet |
| 9 | QLCR web |
| 10 | Mediation Noc 4.0 |
| 11 | Cảnh báo sớm |
| 12 | CĐ thị trường |
| 13 | Mediation CBS |
| 14 | Nhà trạm thông minh |
| 15 | Security |
| 16 | Quản lý tủ nguồn cơ điện |
| 17 | Quản lý tiến độ công trình (QLTDCT) |
| 18 | Quản lý Công tác Kỹ thuật |
| 19 | Hệ thống Profile node mạng |
| 20 | Hệ thống view thông số NodeB, RNC |
| 21 | TKTƯ |
| 22 | OCS tool |
| 23 | Send SMS (Thay thế bulk sms) |
| 24 | Tool HLR |
| 25 | Trace thuê bao |
| 26 | NPMS |
| 27 | Cơ Điện |
| 28 | NCMS- Quản lý cấu hình |
| 29 | Quản lý tài liệu |
| 30 | Hệ thống phòng chống thiên tai |
| 31 | Tác động hệ thống viễn thông |
| 32 | Tool khai báo accesslist tự động |
| 33 | NIMS |
| 34 | Quản lý sự cố (NTMS) |
| 35 | V-Book |
| 36 | Vsmart (QLCTKT Tinh version Mobile) |
| 37 | GNOC |
| 38 | OPenVPN |
| 39 | Quản lý cán bộ cao cấp |
| 40 | Quản lý chi phí hành chính |
| 41 | Quản lý đoàn thanh niên |
| 42 | Website Nhịp sống |
| 43 | Quản lý ô tô |
| 44 | Báo cáo giao ban tập đoàn |
| 45 | Sáng kiến ý tưởng |
| 46 | Kho tàng tài sản |
| 47 | Wiki Viettel |
| 48 | Thông tin nhân sự |
| 49 | Quản lý tài chính kế toán |
| 50 | Phần mềm tài chính kế toán |
| 51 | V-Office |
| 52 | V-Email |
| 53 | ViettelFamily |
| 54 | Hệ thống đào tạo Elearning |
| 55 | Quản lý tri thức |

## Tăng trưởng người dùng của hệ thống SSO