**TRƯỜNG ĐẠI HỌC QUỐC TẾ MIỀN ĐÔNG**

**KHOA KỸ THUẬT**

**NGÀNH KỸ THUẬT PHẦN MỀM**

**🙢 🕮 🙠**

**Báo cáo thực tập**

**Xây dựng Web Portal cho dịch vụ CloudBackup**

**Giảng viên hướng dẫn:**

ThS.Hà Minh Ngọc

**Sinh viên thực hiện:**

Nguyễn Mạnh Phúc

Lưu Nhật Phi

Trần Công Duy

**MỤC LỤC**

[Mục lục hình 3](#_Toc397536959)

[1 Giới thiệu 4](#_Toc397536960)

[1.1 Công ty cổ phần công nghệ truyền thông Việt Nam 4](#_Toc397536961)

[1.2 Dịch vụ CloudBackup 4](#_Toc397536962)

[1.3 Hiện trạng ứng dụng Portal 5](#_Toc397536963)

[2 Dashboard 6](#_Toc397536964)

[2.1 Định nghĩa 6](#_Toc397536965)

[2.2 Chức năng 7](#_Toc397536966)

[2.3 Các vấn đề 10](#_Toc397536967)

[2.4 Giải pháp 12](#_Toc397536968)

[2.5 Trải nghiệm người dùng 12](#_Toc397536969)

[2.5.1 Build Tree 12](#_Toc397536970)

[2.5.2 Build popup 13](#_Toc397536971)

[2.5.3 Phát hiện các thao tác không chủ đích 13](#_Toc397536972)

[2.5.4 Tối ưu vị trí hiện thị popup 14](#_Toc397536973)

[3 Tổng quát hóa phân tích dữ liệu 17](#_Toc397536974)

[3.1 Định nghĩa 17](#_Toc397536975)

[3.2 Vấn đề 17](#_Toc397536976)

[3.3 Phân tích 18](#_Toc397536977)

[3.4 Giải pháp 18](#_Toc397536978)

[4 Unit Test 21](#_Toc397536979)

[4.1 Định nghĩa 21](#_Toc397536980)

[4.2 Lợi ích của Unit Test 21](#_Toc397536981)

[4.3 Một số Framework Unit tests trên các ngôn ngữ khác nhau 21](#_Toc397536982)

[4.4 Unit test trong Portal 22](#_Toc397536983)

[4.4.1 Một số kỹ thuật được sử dụng trong Unit Test 23](#_Toc397536984)

[4.5 Công việc thực hiện 24](#_Toc397536985)

[5 Kết quả 25](#_Toc397536986)

[5.1 Đạt được 25](#_Toc397536987)

[5.2 Đề xuất cải tiến 25](#_Toc397536988)

Mục lục hình

[Hình 1 Dashboard dạng ngày 6](file:///C:\Users\HOME\Desktop\Report.docx#_Toc397536645)

[Hình 2 Dashboard dạng giờ 7](#_Toc397536646)

[Hình 3. Popup thông tin tổng hợp. 8](file:///C:\Users\HOME\Desktop\Report.docx#_Toc397536647)

[Hình 4. Thông tin tổng hợp tại header 8](#_Toc397536648)

[Hình 5. Thông tin tài khoản admin 9](file:///C:\Users\HOME\Desktop\Report.docx#_Toc397536649)

[Hình 6. Thông tin tài khoản manager 9](#_Toc397536650)

[Hình 7. Thông tin BackupSet 10](#_Toc397536651)

[Hình 8. Thay đổi vị trí manager 10](#_Toc397536652)

[Hình 9. Popup hiển thị thông tin backup job 13](file:///C:\Users\HOME\Desktop\Report.docx#_Toc397536653)

[Hình 11. Popup với vị trí hiển thị dưới 15](file:///C:\Users\HOME\Desktop\Report.docx#_Toc397536654)

[Hình 10. Popup với vị trí hiển thị trái 15](file:///C:\Users\HOME\Desktop\Report.docx#_Toc397536655)

[Hình 12. Popup với vị trí hiển thị trên 16](file:///C:\Users\HOME\Desktop\Report.docx#_Toc397536656)

[Hình 13. Dữ liệu OBS trả về 18](#_Toc397536657)

[Hình 14. Class diagram 19](file:///C:\Users\HOME\Desktop\Report.docx#_Toc397536658)

[Hình 15. Unit test 22](#_Toc397536659)

[Hình 16. Multi test 24](#_Toc397536660)

# Giới thiệu

## Công ty cổ phần công nghệ truyền thông Việt Nam

Công ty cổ phần công nghệ truyền thông Việt Nam (VNTT) được thành lập vào tháng 01 năm 2008 trên cơ sở góp vốn của 03 cổ đông sáng lập là Tập đoàn bưu chính viễn thông Việt Nam (VNTT), Công ty Đầu tư và Phát triển Công nghiệp (Becamex IDC) và Ngân hàng đầu tư và phát triển (BIDV) Trụ sở chính đặt tại Đường số 1 - khu công nghiệp Việt Nam - Singapore II với vốn điều lệ đăng ký là 500 tỷ đồng.

Các lĩnh vực kinh doanh tập trung: Dịch vụ truyền thông, khảo sát, tư vấn, lắp đặt, bảo dưỡng các công trình viễn thông, truyền thông, công nghệ thông tin và các công trình liên quan đến quản lý tự động, sản xuất, gia công, cung ứng, xuất khẩu thiết bị điện tử, cơ khí chính xác, tự động, chuyển giao công nghệ, gia công xuất khẩu phần mềm, đại lý các dịch vụ điện thoại, ADSL,… đặc biệt các dịch vụ cung cấp trong lĩnh vực viễn thông, công nghệ thông tin cho các nhà đầu tư trong các khu công nghiệp và đô thị do Becamex IDC đầu tư và quản lý.

## Dịch vụ CloudBackup

vGuard Cloud Backup (Online Backup) là dịch vụ sao lưu dữ liệu trong máy tính KH (server, desktop, laptop) về lưu trữ tại Data Center của VNTT thông qua kết nối Internet. Khi gặp sự cố mất mát/hư hỏng dữ liệu, KH có thể phục hồi lại từ các bản sao đang được lưu trữ tại Data Center của VNTT. Dịch vụ hỗ trợ backup nhiều loại dữ liệu khác nhau như: e-mail, tập tin, cơ sở dữ liệu….

Khách hàng sử dụng dịch vụ vGuard Cloud Backup theo cách như sau:

* Mỗi khách hàng được cung cấp một (hoặc nhiều) tài khoản dịch vụ. Mỗi tài khoản có giới hạn dung lượng xác định (50GB, 100GB, 500GB,…)
* Khách hàng cài đặt và cấu hình phần trên máy tính cần sao lưu (vGuard Lite cho laptop/desktop, vGuard Pro cho server).
* Đến thời điểm trong lịch sao lưu, phần mềm tự động nén, mã hóa và gửi dữ liệu về Data Center của VNTT qua Internet. Bên cạnh sao lưu tự động, phần mềm cũng cho phép khách hàng sao lưu thủ công. Để quản lý tài khoản và quá trình sao lưu, KH truy cập Web-console tại địa chỉ <https://sv.vguard.vn>.
* Để phục hồi dữ liệu, khách hàng sử dụng phần mềm để lấy lại, hoặc phục hồi qua web tại Web-console

## Hiện trạng ứng dụng Portal

Các loại tài khoản trong Portal gồm:

* Admin: là tài khoản được sử dụng trên hệ thống CloudBackup cũ, được dùng để quản lý trực tiếp các BackupSet.
* Manager: quản lý tất cả tài khoản admin của một khách hàng (một khách hàng có thể có nhiều tài khoản admin), được quyền cấp phát dung lượng cho các tài khoản admin (dựa trên dung lượng tổng của tài khoản), tạo mới tài khoản admin, xóa tài khoản admin… Loại tài khoản này được tạo bởi tài khoản vnttadmin hoặc vnttdirector.
* VNTTAdmin là tài khoản được tạo sẵn để VNTT có thể tạo mới và quản lý manager cho khách hàng.
* VNTTDirector tương tự tài khoản VNTTAdmin nhưng có quyền tạo ra các tài khoản VIP.
* VNTTAgent là tài khoản dùng cho nhân viên VNTT có thể quan sát hoạt động của hệ thống CloudBackup qua dashboard dạng giờ.
* Trước khi bắt đầu xây dụng dashboard thì các tính năng về tạo tài khoản (manager, admin), các tính năng về BackupSet, các nội dung và report đã hoàn thành.
* Dashboard được xây dụng như trang chủ của tất cả các loại tài khoản, đồng thời đóng vai trò như một bảng tổng hợp để người dùng kiểm tra sự hoạt động của hệ thống CloudBackup một cách tiện lợi, nhanh chóng.

# **Dashboard**

## **Định nghĩa**

Dashboard là bảng gồm các hàng và các cột.

* Mỗi ô trong bảng là tổng hợp trạng thái của một tài khoản manager (admin hoặc BackupSet) trong một thời gian cố định (một ngày hoặc một giờ).
* Các ô được hiển thị với màu sắc khác nhau tương ứng với trạng thái của tài khoản (BackupSet) đó trong đơn vị thời gian. Những trạng thái này được xét theo thứ tự ưu tiên nếu có nhiều BackupJob trong cùng đơn vị thời gian.
* Mỗi dòng trong dashboard tương ứng với một tài khoản (manager, admin) hoặc BackupSet.

Dashboard được thể hiện ở 2 dạng:

* Dạng theo ngày: tổng hợp dữ liệu theo ngày, được hiển thị thành một tháng được chọn (manager, admin, vnttadmin, vnttdirector)
* Dạng theo giờ: tổng hợp dữ liệu theo giờ, được hiển thị thành một ngày tính từ thời điểm hiện hành, dạng này được sử dụng bởi tài khoản vnttagent.

Hình 1 Dashboard dạng ngày



Hình 2 Dashboard dạng giờ

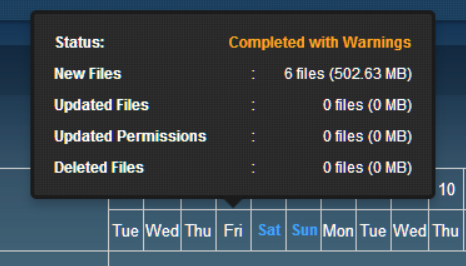
* Dashboard là một tính năng mới, chưa xuất hiện trong console. Dashboard cho phép user có cái nhìn tổng quát về trạng thái của tài khoản (manager, admin) hoặc theo dõi trạng thái của khách hàng (vnttadmin, vnttagent, vnttdirector). Đây là điều rất quan trọng đặc biệt là đối với user mới vì có thể theo dõi được hoạt động của tài khoản. Đồng thời, công ty VNTT cũng có thể theo dõi tình trạng sử dụng của khách hàng một cách dễ hàng để có thể hỗ trợ nhanh chóng, nâng cao chất lượng dịch vụ.

## **Chức năng**

* Hiển thị trạng thái tổng hợp theo ngày/giờ của user hiện hành (áp dụng đối với manager và admin).
* Hiển thị trạng thái tổng hợp theo ngày/giờ của toàn bộ tài khoản khách hàng (áp dụng đối với VNTTAdmin, VNTTAgentvà VNTTDirector).
* Hiển thị popup tổng hợp thông tin khi rê chuột vào các ô trên dashboard.

Hình 3. Popup thông tin tổng hợp.

* Hiển thị popup tổng hợp thông tin khi rê chuột vào các ô header trên dashboard (dữ liệu theo ngày hoặc giờ).



Hình 4. Thông tin tổng hợp tại header

* Hiển thị popup thông tin tài khoản/BackupSet khi rê chuột vào ô tương ứng. 

Hình 5. Thông tin tài khoản admin

Hình 6. Thông tin tài khoản manager



Hình 7. Thông tin BackupSet

* Tạo liên kết đến các trang setting, BackupSet, report…
* Cho phép thu gọn (collapse) và mở rộng (expand) để xem chi tiết.
* Hỗ trợ thay đổi thứ tự manager đối với dashboard dạng giờ.



Hình 8. Thay đổi vị trí manager

* Cho phép chọn tháng hiển thị (dạng ngày) hoặc thay đổi ngày dạng giờ).

## Các vấn đề

* Lượng dữ liệu lớn: Số BackupSet trong một tài khoản manager là

Trong đó:

n là số lượng account vguard pro

m là số lượng account vguard lite (mỗi tài khoản chỉ có tối đa một BackupSet).

ai là số BackupSet của tài khoản vguard pro thứ i

Với n = 5, ai = 5 ∀i, m = 5. Ta có S = 30.

Với 200 manager thì có khoảng 30 x 200 BackupSet = 6000 BackupSet.

Nếu mỗi BackupSet thực hiện backup mỗi 2h thì số BackupJob là 6000 x 12 = 72000 BạckupJob.

* Tốc độ OBS: Thời gian trả lời của OBS khi truy xuất các record là khá lâu và API của OBS chỉ hỗ trợ truy xuất BackupJob trên từng tài khoản admin vào một ngày xác định.

Ngoài ra, thông tin cung cấp bởi chức năng truy xuất trên không thể cung cấp đầy đủ thông tin cho các backup job mà đòi hỏi với mỗi BackupJob cần một câu truy vấn OBS khác để lấy thông tin. Tất cả dẫn đến thời gian truy vấn trực tiếp trên OBS để lấy dữ liệu về BackupJob là rất dài.

* Tần suất xem: Các thông tin về tài khoản (manager, admin), BackupSet được xem với tần số thấp và có thể được thay đổi trong một phiên nên không được lưu trữ trên web mà được truy vấn với mỗi khi có yêu cầu xem từ phía người dùng. Tuy nhiên, các dữ liệu tổng hợp từng ô yêu cầu xử lý lớn nên cần được lưu trữ để có thể sử dụng dễ dàng.
* Sự thay đổi: API của CloudBackup cung cấp chức năng rất hạn chế để có thể cập nhật dữ liệu BạckupJob. Việc cập nhật dữ liệu chỉ có thể thực hiện thông qua nhiều truy vấn nên thời gian rất chậm. Đồng thời, các BackupJob có trạng thái là chưa hoàn thành phải được cập nhật lại. Có thể những BackupJob kéo dài sẽ dẫn đến việc phải cập nhật dữ liệu lại nhiều ngày. Do đó, cần phải xác định một giới hạn thời gian để có thời tránh trường hợp BackupJob không kết thúc hoặc kéo dài quá lâu.
* Thay đổi thứ tự manager: thứ tự tài khoản manager phải cố định đến khi được yêu cầu thay đổi. Do đó, thứ tự cần được lưu trữ để sử dụng mỗi lần được truy vấn. Ngoài ra, cần có cơ chế để xử lý trường hợp khách hàng nhập sai giá trị thứ tự mới.

## Giải pháp

* Cache (Database, Web):
* Database: Lưu trữ toàn bộ dữ các record BackupJob trên database, do đó giảm số lượng truy vấn trên OBS. Đây là điểm then chốt giải quyết vấn đề về tốc độ OBS và số lượng dữ liệu lớn được truy vấn.
* Web: Dữ liệu tổng hợp của các ô được cache lại trên web để giảm thời gian truy vấn, tổng hợp theo ngày (giờ) của toàn bộ các admin khi được yêu cầu.
* Kiểm soát sự thay đổi:
* Bắt đầu cập nhật từ ngày đầu tiên có BackupJob chưa hoàn thành.
* Nếu ngày này trước thời gian hiện tại quá 7 ngày thì sẽ bắt đầu lúc 7 ngày trước ngày hiện tại.
* Nếu không có BackupJob nào chưa hoàn thành thì sẽ cập nhật từ ngày hiện tại.

Thay đổi thứ tự manager: Dùng lệnh update trong SQL để cập nhật dữ liệu. Dựa trên các giá trị vị trí cũ và mới để cập nhật vị trí mới. Đồng thời, dựa trên số lượng tài khoản manager để có thể sửa lỗi các trường hợp nhập sai của người dùng (nhập số lớn hơn số tài khoản manager hoặc nhâp số bé hơn 1).

## Trải nghiệm người dùng

### Build Tree

**Mục đích:** cho phép người dùng đóng/mở các node để có thể xem chi tiết/ tổng quát hoạt các hoạt động trên CloudBackup. Ngoài ra, việc phân cấp thành tài khoản manager, admin và backupset giúp cho việc quan sát, kiểm soát dễ dàng hơn.

Dựa trên cấu trúc phân cấp từ cao đến thấp lần lượt là manager, admin, BackupSet. Khi đó sử dụng javascript và jquery để xử lý việc đóng mở các node trên cây. Với những nút đang mở thì ẩn tất cả các con (level thấp hơn) của nó. Đối với những nút đang đóng thì hiện tất cả những con của nó.

### Build popup

**Mục đích:** hiển thị thông tin về backup job.

**Vấn đề:** cần xây dựng một popup hiển thị đầy đủ thông tin cần thiết. Tuy nhiên cũng cần phải tối ưu về mặt truy vấn dữ liệu để cải thiện tốc độ đáp ứng người dùng từ đó cần phải giải quyết một số vấn đề.

Hình 9. Popup hiển thị thông tin backup job

### Phát hiện các thao tác không chủ đích

**Vấn đề:** khi người dùng tương tác với trang web, ngoài các thao tác bình thường còn có các thao tác ngoài ý không chủ đích. Ví dụ như khi người dùng vô tình rê chuột qua một ô. Rõ ràng người dùng không muốn hiển thị thông tin popup về BạckupJob. Nhưng sự kiện mouse over đã phát sinh sẽ hiển thị popup. Và khi chuột rê ra ngoài ô sẽ phát sinh sự kiện mouse leave tắt hiển thị popup. Nếu người dùng rê chuột liên tục qua các ô sẽ gây hiện tượng giật rất khó chịu cho người dùng.

**Giải pháp**: khi phát sinh sự kiện mouse over, cần phải delay thời gian hiển thị popup. Nếu sau thời gian delay, con trỏ chuột vẫn nằm trong ô thì hiển thị popup. Ngược lại thì không hiển thị.

Sau khi hiển thị, delay một khoảng thời gian. Trong khoảng thời gian này vẫn hiện thị popup. Sau khi kết thúc thời gian delay, nếu con trỏ chuột rời khỏi ô thì tắt hiển thị popup. Việc này nhằm tránh trường hợp popup hiển thị và tắt quá nhanh, gây khó chịu cho người dùng.

### Tối ưu vị trí hiện thị popup

**Vấn đề:** khi hiển thị popup cho người dùng, ta cần tính toán vị trí hiển thị sao cho hợp lý. Ví dụ khi vùng cần hiển thị gần biên trái trình duyệt thì cần hiện thị popup bên trái và hiển thị popup bên phải khi vùng hiển thị gần biên trái.

**Giải pháp:** Để hiển thị hợp lý cần tính được các yếu tố sau:

* Vị trí cần hiển thị: thông qua phương thức offset của jQuery. Phương thức offset sẽ trả về vị trí của thành phần được chọn tính theo vị trí trong thành phần document.
* Vị trí của cửa sổ hiện tại: dựa trên 4 phương thức của jQuery là:
  + scrollTop(): trả về kết và là vị trí của thanh cuộn dọc cho thành phần được chọn.
  + scrollLeft(): trả về kết quả là vị trí của thanh cuộn ngang cho thành phần được chọn.
  + $(window).width(): trả về chiều rộng của cửa sổ web hiện tại.
  + $(window).height(): trả về chiều cao của cửa sổ web hiện tại.
* Kích thước của popup cần hiển thị.

Sau khi có đủ các thông số cần thiết. Xét các điều kiện sau để hiển thị:

* Nếu vị trí hiện tại trừ chiều cao của popup không vượt quá biên trên của trình duyệt và vị trí hiện tại ± một nữa chiều rộng của popup không vượt quá biên trái và biên phải thì giá trị vị trí hiển thị là trên.
* Nếu vị trí hiện tại cộng chiều cao của popup không vượt quá biên dưới của trình duyệt và vị trí hiện tại ± một nữa chiều rộng của popup không vượt quá biên trái và biên phải thì giá trị vị trí hiển thị là dưới.
* Nếu cả 2 trường hợp trên đều không thỏa, xét nếu vị trí hiện tại cộng chiều cao popup và vị trí hiện tại trừ chiều rộng của popup không vượt quá biên trên và biên trái của trình duyệt thì giá trị vị trí hiển thị là trái.
* Ngược lại giá trị vị trí hiển thị là phải.



****

Hình 11. Popup với vị trí hiển thị dưới

Hình 10. Popup với vị trí hiển thị trái



Hình 12. Popup với vị trí hiển thị trên

# Tổng quát hóa phân tích dữ liệu

## Định nghĩa

Các dữ liệu trả về ban đầu là khối lượng lớn dữ liêu thô, dưới định dạng XML và không có đặc tả cho dữ liệu rõ ràng, do đó cần một quá trình phân tích, biến đổi tổng quát các dữ liệu này thành các đối tượng để sử dụng một cách thuận tiện nhất trong các tác vụ khác liên quan đến dữ liệu.

## Vấn đề

* Toàn bộ dữ liệu của chương trình được truy xuất thông qua các OBS server, dữ liệu gồm thông tin về users, backup set, backup job và các thông tin về hệ thống. Tuy nhiên các thông tin này được chia làm các API khác nhau và các services được viết trên các API tạo nên 32 services khác nhau.
* Thông số đầu vào (paramaters): mỗi service được định nghĩa với thông số đầu vào khác nhau. Do đó, số lượng dữ liệu đầu vào lớn dẫn đến gây khó ngăn trong quá trình kiểm soát và sai sót trong quá trình thao tác với các services.
* Dữ liệu trả về (Service Response): tương tự thông số đầu vào số lượng dữ liệu lớn ngoài ra dữ liệu được trả về với nhiều cấu trúc dữ liệu khác nhau. Do đó, nếu dữ liệu không được xử lý sẽ tốn chi phi và thời gian cho quá trình kiểm soát và tính chính xác của dữ liệu không được đảm bảo.
* Ngoài ra, quá trình test dữ liệu trong các tác vụ xử lý khác sẽ gặp khó khăn dẫn đến rủi ro cao.
* Dữ liệu các servers OBS cung cấp được định dạng dữ liệu kiểu XML với số lượng thuộc tính lớn, do đó việc này sẽ gây khó khăn trong quá trình kiểm tra lại dữ liệu khi truy xuất. Ngoài ra tuy dữ liệu XML là dạng dữ liệu có cấu trúc nhưng do không được cung cấp đặc tả cấu trúc dữ liệu rõ ràng, mỗi API lại trả cấu trúc khác và nhu cầu lưu trữ dự liệu khác nhau, do đó không thể áp dụng được các công nghệ serialization và deserialization XML của C#.



Hình 13. Dữ liệu OBS trả về

* Ngoài những khó khăn trên, trong dữ liệu trên OBS có một số dữ liệu cần được truy xuất thường xuyên, nếu không được xứ lý thích hợp chi phí và thời gian bỏ trong quá trình truy xuất lên OBS sẽ gây ra ảnh hưởng lớn tốc độ của chương trình.

## Phân tích

* Số lượng các thông số đầu vào và dữ liệu trả về của service rất lớn nhưng chúng có các thuộc tính chung .
* Dữ liệu XML không được cung cấp đặc tả cụ thể nhưng ta có thể thấy cấu trúc lưu trữ dữ liệu được lặp lại một số dạng nhất định.
* Các dữ liệu thường xuyên được sử dụng nhưng không phải lúc nào thông tin đó cũng thay đổi nên ta có thể sử dụng một số kĩ thuật để kiểm tra và truy xuất khi cần thiết.

## Giải pháp

* Chúng ta sẽ viết các phương thức để chuyển đổi dữ liệu dưới dạng XML thành các đối tượng trong C#, việc này sẽ làm cho các tác vụ liên quan đến dữ liệu được xử lý dễ dàng và độ chính xác được đảm bảo. Tuy nhiên khi tạo các lớp này số lượng các thuộc tính vẫn còn lớn để đảm bảo chất lượng không sai xót khi tạo các lớp này, có một giải pháp khá hữu ích là viết một chương trình đơn giản để tự động lọc ra các tên thuộc tính của các API do OBS cung cấp và chuyển thành định dạng các thuộc tính trong C#. Việc này giảm thời gian đáng kể và đảm bảo tên thuộc tính chính xác.

Hình 14. Class diagram



* Cấu trúc dữ liệu XML được lưu dưới dạng các node bao gồm các thuộc tính và giá trị, sau khi được phân tích ta có thể thấy dữ liệu có một số cấu trúc chính như Object, Collection, Dictionary và Primitive. Sau đó, viết phương thức duyệt lần lượt các node phía ngoài sau đó lần lượt duyệt vào trong để lấy dữ liệu về các thuộc tính và giá trị của node. Để viết phương thức này ta áp dụng một số kỹ thuật của C# :
  + Generic method: để cho phương thức linh hoạt hơn, tránh lập lại code => code được trình bày dễ hiểu, ngắn gọn
  + Activator Class: bao gồm phương tạo ra các lớp đối tượng
  + Unit Test: Đảm bảo tính đúng đắn của phương thức.
* Các dữ liệu cần thiết ví dụ như thông tin Users cần được cập nhật liên tục, do đó,việc tạo ra cache để lưu lại danh sách users và phân tích tính toán thời gian hợp lý để cập nhật lại danh sách. Do đó danh sách này chỉ được cập nhật khi User được kiểm tra chưa nằm trong danh sách hoặc sau khoảng thời gian nhất định. Ngoài ra, chúng ta có một phương thức cập nhật tức thời để dành cho một số trường hợp như tạo tài khoản.

# Unit Test

## Định nghĩa

* + Unit Test là quá trình kiểm thử tự động các thành phần của chương trình như các phương thức, các lớp đối tượng hoặc các đơn vị khác trong code. Unit Test sẽ giúp developers có khả năng để xác định chức năng của họ hoạt động đúng như mong đợi.

## Lợi ích của Unit Test

* Unit Test giúp xác định giải thuật, logic không đúng hoặc các lỗi trong quá trình lập trình từ sớm để sửa chữa và nâng cao chất lượng code.
* Unit Test giúp có thể kiểm tra lại code một cách nhanh chóng và chính xác khi thay đổi một phần nhỏ trong chương trình.
* Unit Test chứng minh rằng code của bạn đang hoạt động.
* Test-first buộc developer phải có kế hoạch trước khi lập trình.
* Chi phí và thời gian hiện thực phần mềm giảm, bên cạnh đó chất lượng code được nâng cao và giảm rủi ro của từng chức năng một trong phần mềm.

## Một số Framework Unit tests trên các ngôn ngữ khác nhau

* C#:

*Visual studio*: Visual Studio Unit Testing Framework lần đầu tiên được bao gồm trong Visual Studio Team System 2005, nơi nó được tích hợp với IDE, nhưng không có sẵn trong hầu hết được sử dụng Standard Edition. Từ Visual Studio 2008 nó có sẵn cũng Professional Edition. Bắt đầu với Visual Studio Express 2013, nó được bao gồm các phiên bản với Visual Studio Express.

* Java:

*JUnit* là một framework unit test cho các ngôn ngữ lập trình Java. JUnit là một phần quan trọng trong sự phát triển của test-driven development, dựa trên kiến ​​trúc xUnit.

* C++:

Google C++ Testing Framework: một thư viện Unit Test cho các ngôn ngữ lập trình C ++, dựa trên kiến ​​trúc xUnit

## Unit test trong Portal

* Các phương thức được viết đều được thiết kế và hiện thực các test cases tự động trước. Các dữ liệu trong database được sử dụng kĩ thuật mock để giả lập cho các unit tests để đảm bảo hạn chế lỗi tối đa khi chạy trong môi trường thực tế. Bên cạnh đó, số lượng test cases nhiều đảm bảo các chức năng chạy đúng như mong muốn do đó số lượng bug được tìm ra và sửa chữa sớm. Trong quá trình sửa chữa, các test cases luôn được chạy lại để đảm bảo tính đúng đắn của chương trình.



Hình 15. Unit test

### Một số kỹ thuật được sử dụng trong Unit Test

#### Kỹ thuật Mock

* Mock là một phương thức/đối tượng mô phỏng hành vi của các phương thức/đối tượng thực tế.
* Mocking chủ yếu được sử dụng trong Unit Test. Một đối tượng được kiểm tra có thể phải phụ thuộc (dependencies objects or methods) vào các đối tượng phức tạp hoặc đối tượng bên ngoài khác. Để tách biệt các hành vi của đối tượng muốn kiểm tra ta đối tượng bằng các đối tượng Mocks, điều này rất hữu ích nếu các đối tượng thực tế không thể tích hợp vào Unit Test (cơ sở dữ liệu, một số trạng thái lỗi,…).

#### Kỹ thuật multi test

* Trong thực tế số lượng test cases rất lớn nếu mỗi test cases là một Unit Test đơn sẽ gây nhiều khó khăn trong quá trình hiện thực Unit Test như khó kiểm soát các test cases, đặt tên, số lượng code lớn,…
* Các test cases được phân tích thành một tập hợp cùng với kì vọng kết quả của mỗi phần tử trong tập hợp đó. Các test này sẽ được tự động thực hiện lần lượt chỉ với một hàm test.



Hình 16. Multi test

## Công việc thực hiện

* Tạo một kế hoạch thử nghiệm
* Tạo các test cases và test data
* Nếu có thể tạo ra các scripts để chạy các test cases.
* Chạy test cases cho mỗi khi hoàn thành code.
* Sửa chữa các lỗi nếu có và chạy lại test cases.
* Lặp lại chu kỳ test cho đến khi "đơn vị" không còn lỗi.

# Kết quả

## Đạt được

Trong quá trình thực hiện, đã đạt được các kết quả sau:

* Xây dựng trang dashboard đáp ứng các yêu cầu của requirements.
* Xử lý tối ưu được các yêu cầu của người dùng.
* Nâng cao trải nghiệm người dùng.
* Thành công trong việc phân tích và xử lý dữ liệu trả về từ OBS.
* Áp dụng các chuẩn Unit test chuyên nghiệp vào dự án.
* Được tham gia vào dự án thực tế ở doanh nghiệp.
* Học hỏi thêm các kĩ năng và công nghệ mới.

## Đề xuất cải tiến

Trong quá trình hiện thực, nhóm đề xuất một số ý tưởng cải tiến như sau:

* Trang web chưa hỗ trợ nhiều thiết bị khác nhau. Đặc biệt là màn hình trên các loại điện thoại di động.
* Chưa hiện thực phân trang cho cột manager, admin, BackupSet.

|  |  |
| --- | --- |
| **Công ty VNTT** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  *Độc lập - Tự do - Hạnh Phúc* |

**PHIẾU NHẬN XÉT, ĐÁNH GIÁ THỰC TẬP TỐT NGHIỆP**

Ngành: Niên khóa:

Họ và Tên SV thực tập 1: Mã số

Họ và Tên SV thực tập 2: Mã số

Họ và Tên SV thực tập 3: Mã số

Đơn vị thực tập:

Thời gian thực tập: Từ đến

Đề tài thực tập tốt nghiệp:

1. **Chấp hành nội quy, quy định của cơ quan:**

1. **Tinh thần trách nhiệm, năng lực thực sự:**

1. **Nội dung thực tập (Nội dung khoa học của đề tài, Tính sáng tạo):**

1. **Kết quả đánh giá:**

|  |  |
| --- | --- |
| **Xác nhận của cơ quan, đơn vị thực tập**  *(Ký và ghi rõ họ tên và đóng dấu)* | **…………….,** *ngày tháng năm 2014*  **Người đánh giá**  *(Ký và ghi rõ họ tên)* |