# Giới thiệu

* VNTT
* Dịch vụ Cloudbackup:

# Nhu cầu

## Hiện trạng ứng dụng portal

## Các chức năng

Nội dung công việc

## Các vấn đề và giải pháp

* Tốc độ obs: Với mỗi tài khoản manager sẽ có nhiều tài khoản admin, mỗi tài khoản admin lài có nhiều backupset. Do đó số lượng backupset trong mỗi tài khoản admin là khá lớn. Trong một tháng( hoặc một ngày) được hiển thị trên dashboard, số lượng các dòng(record) lưu thông tin của backup job là rất lớn (mỗi backupset có thể thực hiện backupjob rất nhiều lần mỗi ngày).

Thời gian trả lời của obs khi truy xuất các record là khá lâu và API của Obs chỉ hỗ trợ truy xuất backup Job trên từng tài khoản admin vào một ngày xác định. Ngoài ra các 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 có 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

* Số lượng dữ liệu cho dashboard: như nói ở trên, số lượng backupJob cần hiển thị trên dashboard là rất lớn. Việc truy xuất tất cả các backupjob ra portal để xử lý sẽ tốn nhiều thời gian
* 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 trư trên web mà được truy vấn với mỗi khi có yêu cầu xem từ phí người dung.
* Vấn đề bất đồng bộ
* Kiểm tra chất lượng các hàm truy vấn Obs:

## Công nghệ:

* Unit Test
* Bất đồng bộ trên giao diện
* Lưu trữ: Để giải quyết vấn đề truy xuất dữ liệu trên obs tốn nhiều thời gian nên để tối ưu thời gian xử lý dashboard nên nhưng thông tin về backupjob được lưu trữ trên portal để có thể xử lý dễ dàng hơn. Đồng thời những thông tin về status của backupjob và report được gộp chung lại để lưu trữ thống nhất (nhũng thông tin này được truy xuất rời rạc bằng 2 lần truy vấn trên obs).
* SQL: Như đã đề cập, dữ liệu trên dashboard được tổng hợp từ các record backupjob nhưng gặp vấn đề về tốc độ do phải truyền tải dữ liệu lớn từ database sang portal để xử lý. Do đó, việc tổng hợp dữ liệu từ database rồi mới xử lý tiếp theo ở portal sẽ giúp tối ưu thời gian đồng thời giảm được lượng dữ liệu truyền tải giữa database và portal.

## Làm việc nhóm

* Team Foudation

# Kết quả

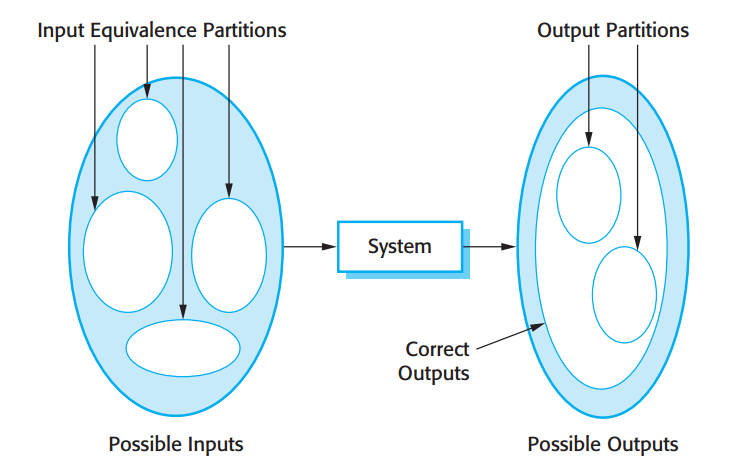
## Đạt được

## Hạn chế

Unit Test.

1. Định nghĩa.

Unit Test là quá trình kiểm thử 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.



1. 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.
* 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.

1. Unit Test trong Dashboard.

Dữ liệu của dashboard được tổng hợp từ nhiều dữ liệu khác nhau trên server trả về, bên cạnh đó chức năng chính của dashboard là hiển thị chi tiết các dữ liệu tổng hợp cho khách hàng cũng như admin. Do đó, các thông tin này cần được đảm bảo chính xác khi lấy từ server và trong các phương thức tổng hợp.

* Tất cả các đối tượng lưu trữ thông lấy từ server và các phương thức xử lý dữ liệu đều phải thực hiện Unit Test.

1. Công việc thực hiện.

* Xác định và thiết kế các trường hợp input testcases có thể xảy ra và kết quả mong đợi của các testcase đó trước khi lập trình chi tiết các chức năng chính.
* Viết tất cả các unit test theo các thiêt kế trên, có thể bổ sung trong quá trình hiện thực các chức năng chính.
* Chạy test cases thường xuyên trong quá trình hiện thực và trong quá trình thay đổi và chỉnh sửa các chi tiết trong chương trình.