Contents

[1. Tìm hiểu về Subversion 2](#_Toc505373315)

[a. Download và cài đặt SVN Server và SVN Client 2](#_Toc505373316)

[b. Các khái niệm 1](#_Toc505373317)

[c. Tạo tài khoản và phân quyền 1](#_Toc505373318)

[d. Lưu trữ và lấy tài liệu 1](#_Toc505373319)

[e. Upload files sau khi thay đổi 1](#_Toc505373320)

[2. Triển khai Phần mềm quản lý phiên bản trên mạng máy tính (GitHub) 1](#_Toc505373321)

[a. Mô tả ngắn gọn 1](#_Toc505373322)

[b. Cách cài đặt 1](#_Toc505373323)

[3. Tạo nơi chứa một dự án 1](#_Toc505373324)

[a. Mô tả 1](#_Toc505373325)

[b. Cách lấy tài liệu 1](#_Toc505373326)

[c. Phần mềm hỗ trợ 1](#_Toc505373327)

[d. Thử nghiệm 1](#_Toc505373328)

[4. Thống kê lượt truy cập 1](#_Toc505373329)

# Tìm hiểu về Subversion.

## Download và cài đặt SVN Server và SVN Client.

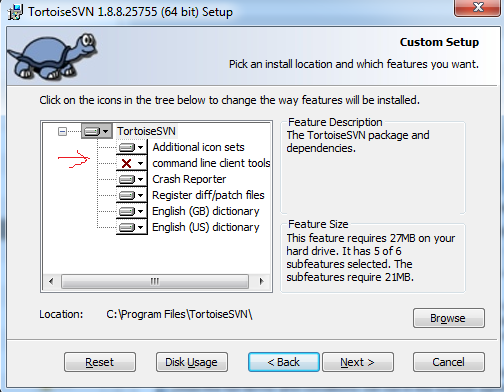
SVN Server:

Link download: <https://www.visualsvn.com/server/download/>

SVN Client:

Link download: <https://tortoisesvn.net/downloads.html>

Chú ý đến bước này cần thay đổi, phải click vào biểu tượng ở dòng command line…thay đổi cho giống với những cái còn lại.



## Các khái niệm.

* SVN server (Visual SVN):

Visual SVN là một sản phẩm của Microsoft, tuy nhiên nó cũng có nhiều phiêu bản chuẩn miễn phí.

Visual SVN sử dụng như là một Repository Server (Kho chứa). Dùng để lưu trữ các file dư liệu chia sẻ giữa các thành viên trong nhóm. Visual SVN được cài trên 1 máy chủ, các thành viên trong nhóm sẽ cài đặt chương trình Client (Subvertion Client).

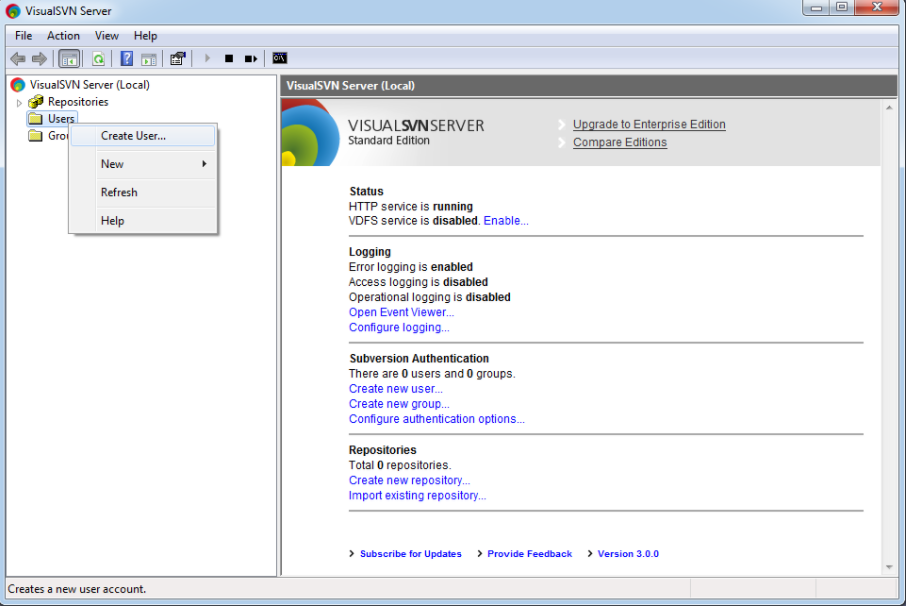
* SVN client (Tortoise SVN):

Tortoise SVN là một chương trình SVN Client nó được cài đặt tại các mày tính của các thành viên trong nhóm, giúp dễ dàng chia sẻ dữ liệu lên SVN Server và cũng như lấy dữ liệu về từ SVN Server.

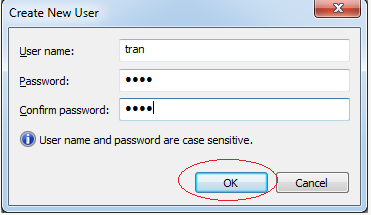
## Tạo tài khoản và phân quyền.

* Tạo tài khoản:

Chúng ta sẽ tạo mới một user.

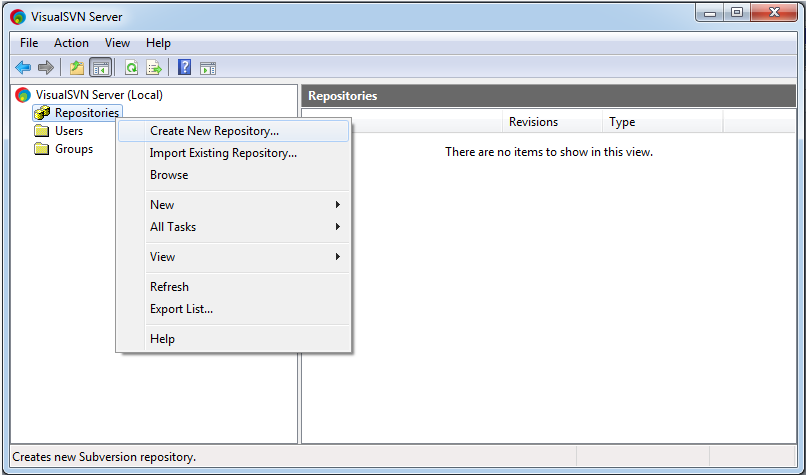


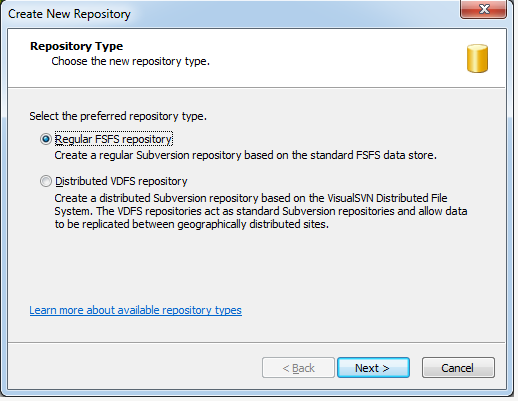
Nhập thông tin username và password.

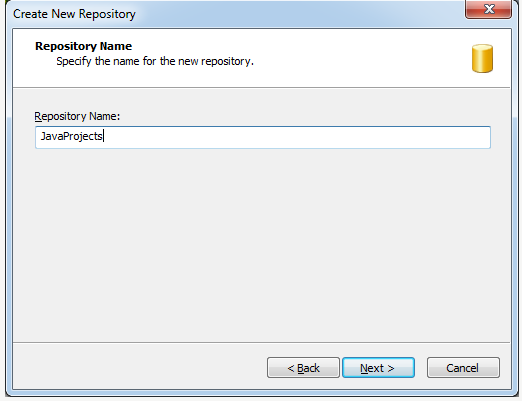


* Phân quyền:

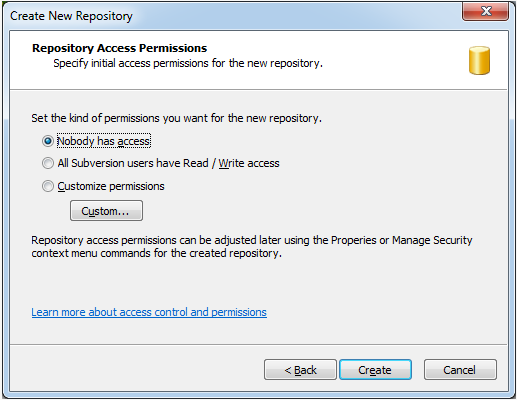
Chúng ta tạo mới một **Repository**với tên **JavaProjects**. Đây là một **Repository**rỗng chưa có dữ liệu, dữ liệu sẽ được đẩy lên **Server**từ một **Subversion Client** (bởi một thành viên trong nhóm lập trình). Và được các thành viên khác lấy về. Các thành viên có thể sửa đổi dữ liệu và đẩy lên **Repository**.



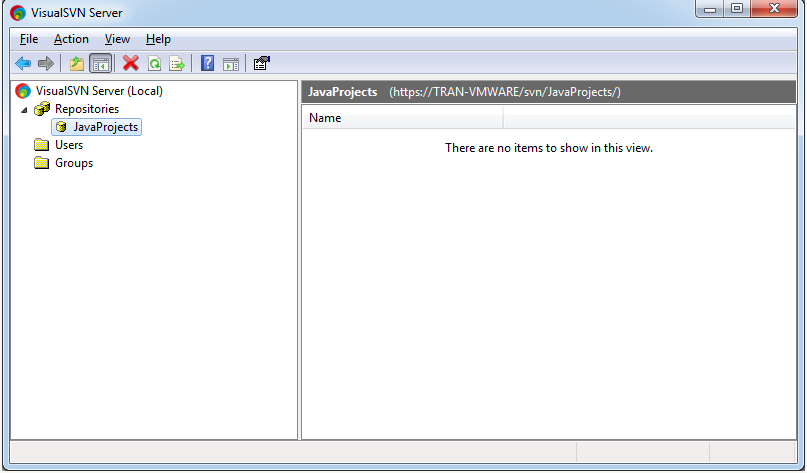




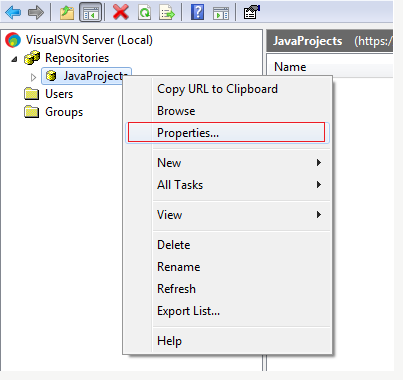
Mặc định là không ai được phép truy cập vào Repository này, chúng ta sẽ phân quyền sau.



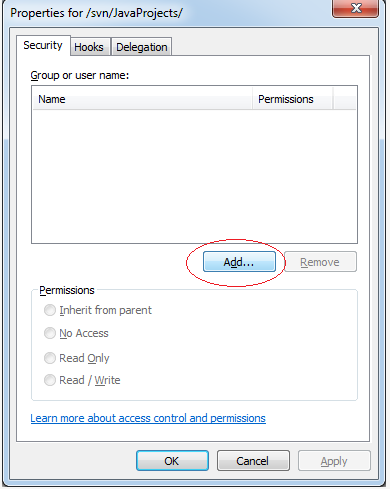
Kết quả:

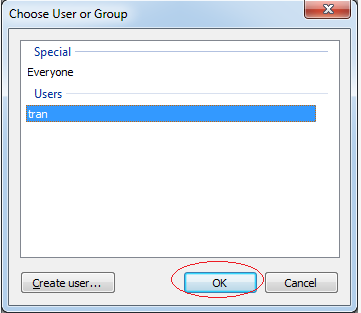


Nhấn phải chuột vào **Repository "JavaProjects"** và chọn **Properties:**

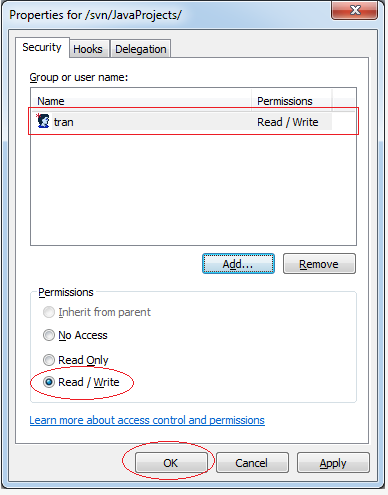


Nhấn **Add** để thêm **user** có quyền truy cập vào **Repository** này.



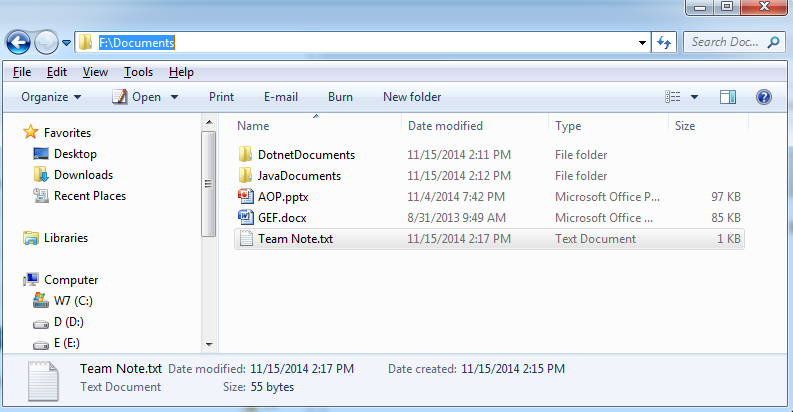


Lựa chọn quyền hạn:

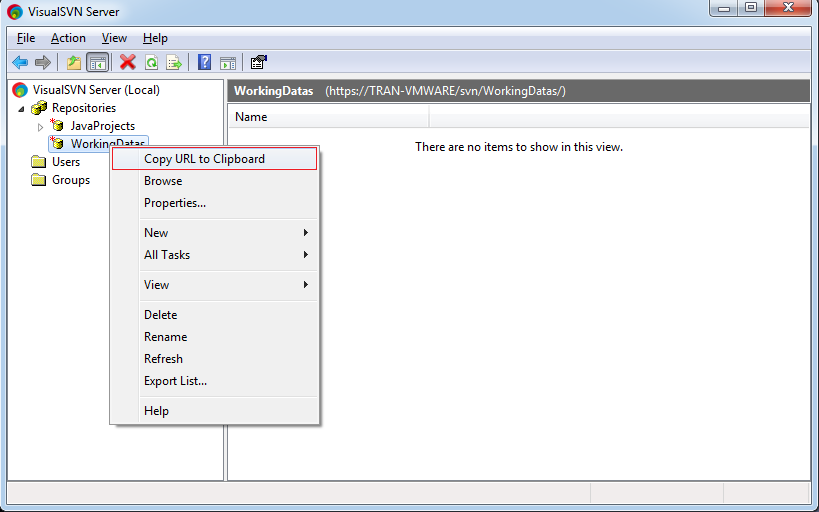


## Lưu trữ và lấy tài liệu.

* Lưu trữ: Giả sử ta muốn đưa file Documents lên SVN Server. Và file Documents có những mục sau:



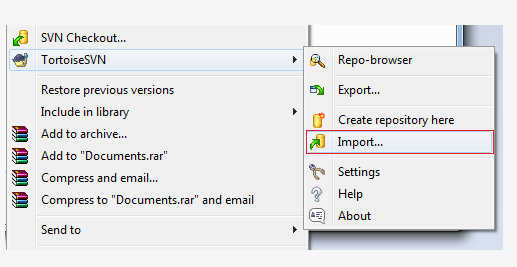
Trên SVN Server đã có sẵn Repository tên WorkingData, ta click chuột phải vào đó sẽ được như sau:



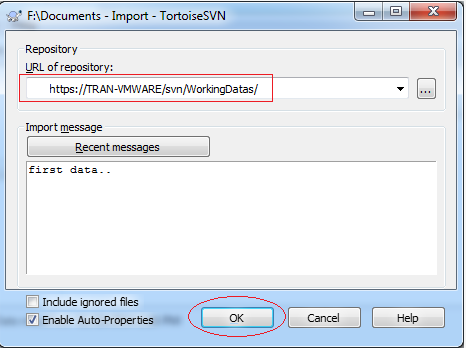
URL mà ta copy được có thể chứa tên của máy chủ SVN, nó có dạng như sau:

<https://TRAN-VMWARE/svn/WorkingDatas/>

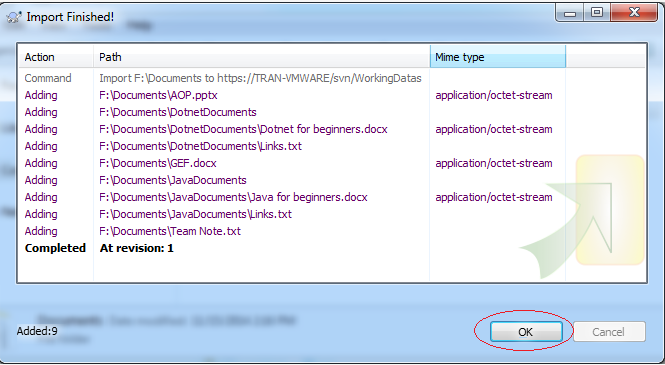
Quay lại với file Document, ta click chuột phải vào nó và chọn như ảnh ở dưới:



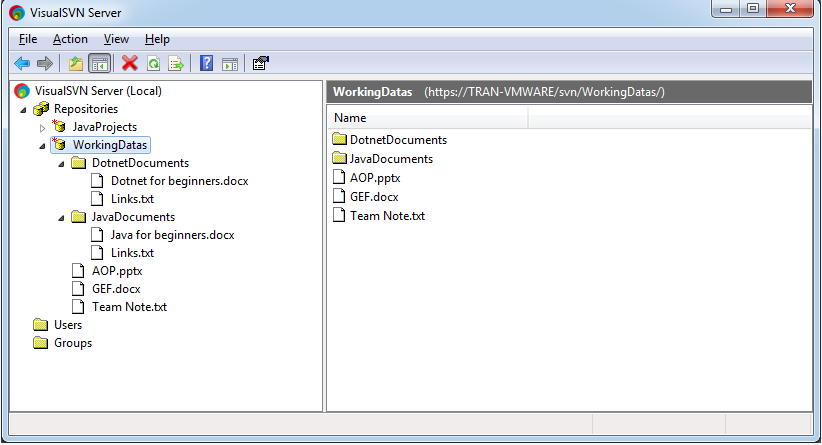
Nhập URL của SVN Repository:



Dữ liệu đã được lưu trữ trên SVN Server:

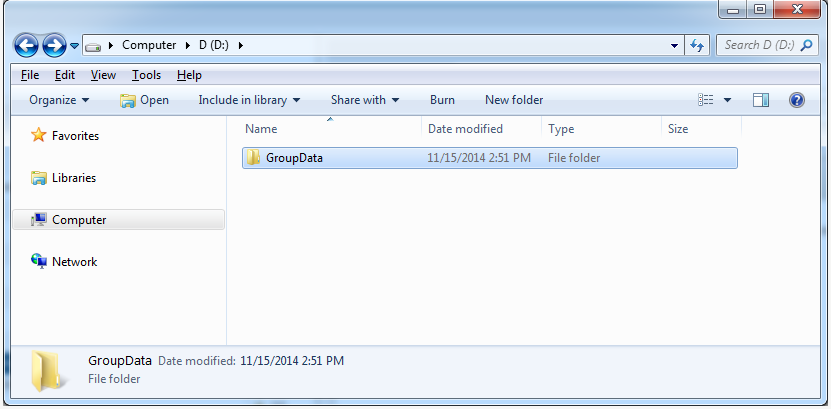


Kết quả:

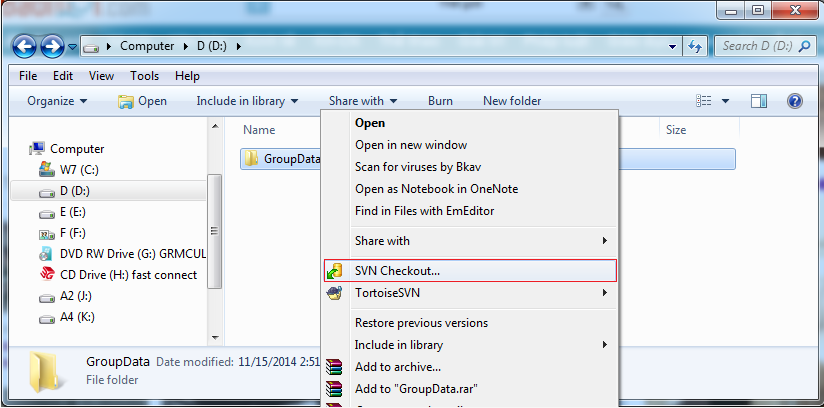


* Cách lầy tài liệu từ SVN Server về.

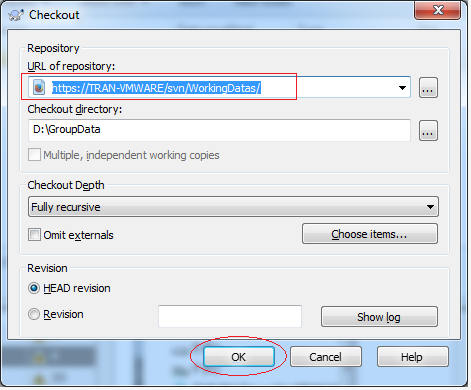
Tạo một thư mục để lựu trữ dữ liệu:



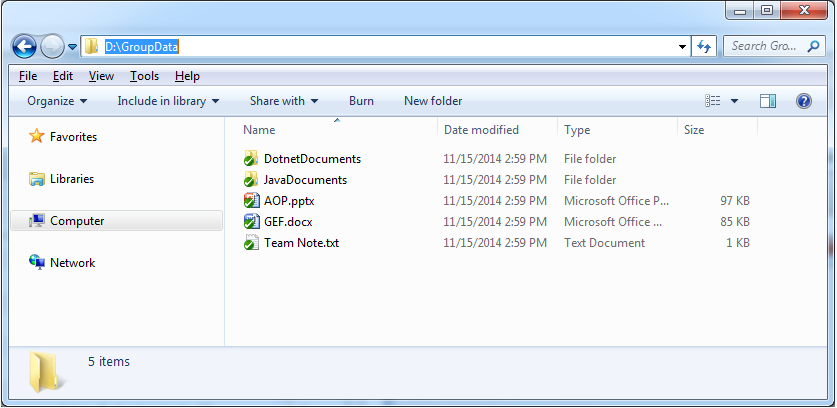
Click chuột phải vào file vừa tạo và chọn như sau:



Nhập URL của SVN Repository:



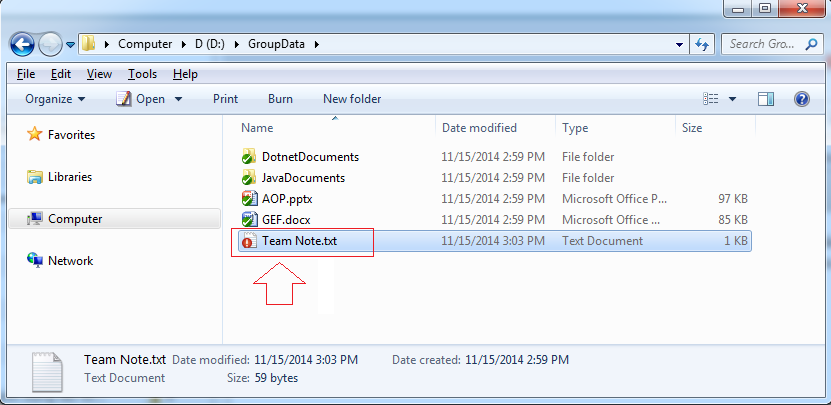
Dữ liệu được lấy về:



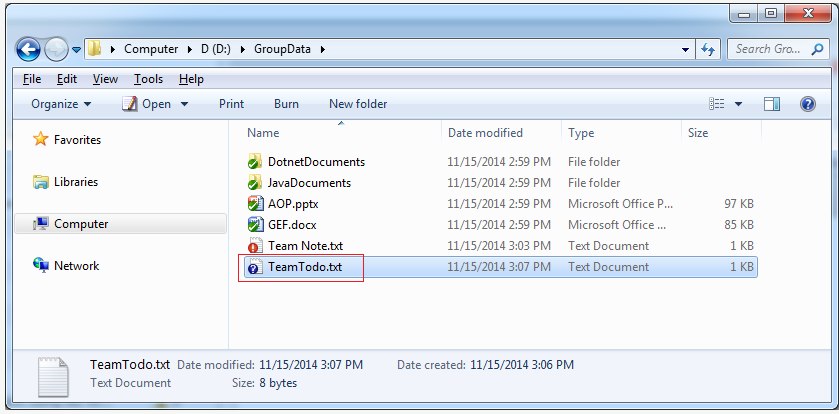
## Upload files sau khi thay đổi.

Việc đưa dữ liệu đã thay đổi lên Repository được gọi là Commit. Và việc cập nhập các dữ liệu đã thay đổi về là hành động Update. Hai hành động này được gọi là Upload.

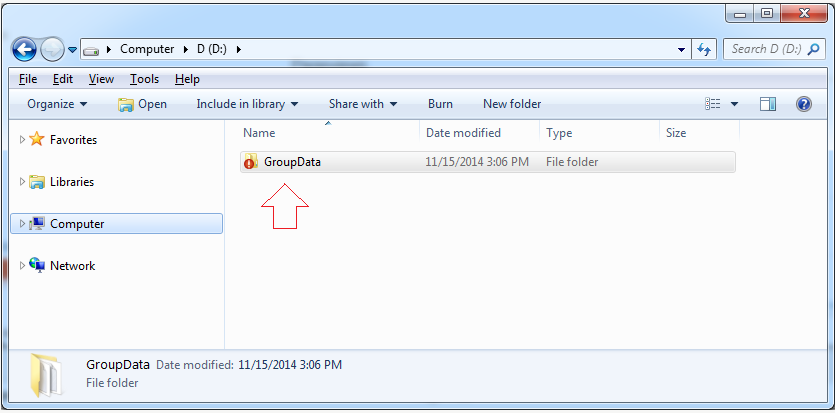
Ví dụ ta thay đổi file TeamNote.txt và lưu lại, khi đó biểu tượng SVN của file này thay đổi, thông báo rằng file đã bị sửa đổi.



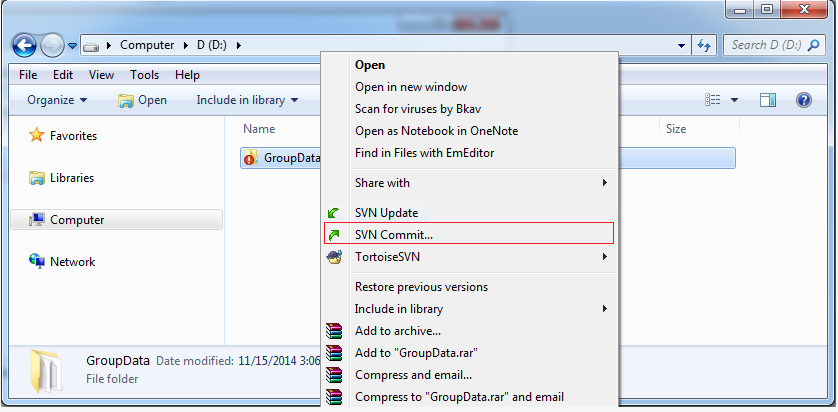
Tiếp theo ta tạo mới một file TeamTodo.txt. Biểu tượng của file mới được tạo ra có dấu ‘?’, thông báo rằng file này mới, chưa được quản lý bởi SVN.



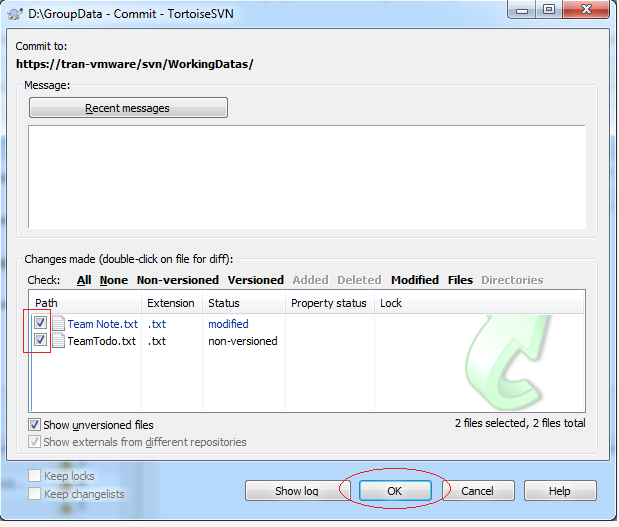
Thư mục SVN chứa file có dữ liệu bị sửa đổi cũng thay đổi biểu tượng.

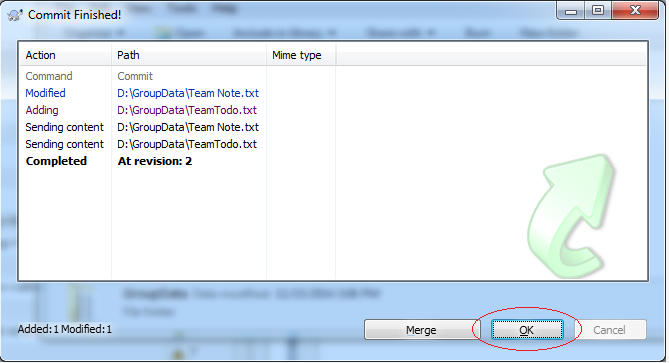


Để **Commit**dữ liệu lên **SVN Repository**, nhấn phải chuột vào file hoặc thư mục cần **Commit**, chọn **SVN Commit...**

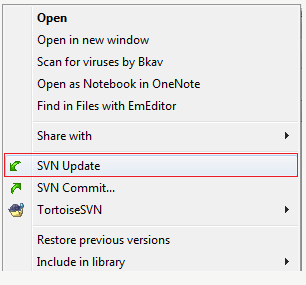


Check chọn các file cần **Commit**trên danh sách. Thường thì Tortoise SVN tự động check các file có sửa đổi.





Các thành viên khác trong nhóm có thể Update các dữ liệu mới nhất từ SVN Repository:



# Triển khai Phần mềm quản lý phiên bản trên mạng máy tính (GitHub).

## Mô tả ngắn gọn.

**Git** - Trình quản lý phiên bản sử dụng rất rộng rãi trên thế giới, **Github** là một mạng xã hội cho lập trình viên dựa trên Git. Git giúp bạn quản lý được mã nguồn, làm việc nhóm, xử lý các thao tác hợp nhất, lịch sử mã nguồn …​ Có thể trong quá trình làm việc với quyển sách này, bạn sẽ cần sử dụng các thư viện mã nguồn mở cho Arduino từ Github, nên việc cài đặt và sử dụng công cụ khá cần thiết cho việc đó. Chưa kể, nó sẽ giúp bạn quản lý mã nguồn và dự án ngày càng chuyên nghiệp hơn.

Làm việc theo nhóm là một phương pháp tính cực nhưng khi làm việc nhóm có một số bất cập như :

* Việc tự code riêng và gửi cho nhau qua email rất mất thời gian của nhau. Giá như anh A có thể chủ động xem những thay đổi của anh B từ xa và tiến hành gộp trực tiếp những thay đổi của anh B vào sản phẩm.
* Việc sửa code mà không sao lưu khiến cho họ phải viết lại code từ đầu khi phần mềm bị lỗi.

Và hai vấn đề trên Git có thể sẽ giúp giải quyết nhanh gọn lẹ và đơn giản hơn rất nhiều. Ngoài ra Git còn làm nhiều việc hơn nữa.

**Git** là tên gọi của một **Hệ thống quản lý phiên bản phân tán** (Distributed Version Control System – ***DVCS***) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. DVCS nghĩa là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (**clone**) từ một kho chứa mã nguồn (**repository**), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (**commit**) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia. Trong Git, thư mục làm việc trên máy tính gọi là **Working Tree**.

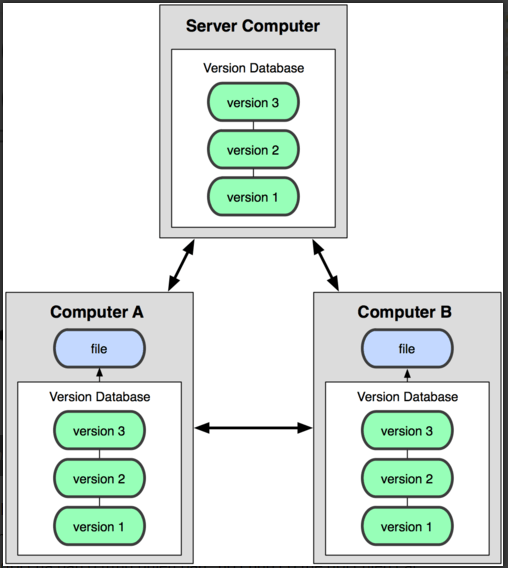
Ngoài ra, có một cách hiểu khác về Git đơn giản hơn đó là nó sẽ giúp bạn lưu lại các phiên bản của những lần thay đổi vào mã nguồn và có thể dễ dàng khôi phục lại dễ dàng mà không cần copy lại mã nguồn rồi cất vào đâu đó. Và một người khác có thể xem các thay đổi của bạn ở từng phiên bản,  họ cũng có thể đối chiếu các thay đổi của bạn rồi gộp phiên bản của bạn vào phiên bản của họ. Cuối cùng là tất cả có thể đưa các thay đổi vào mã nguồn của mình lên một kho chứa mã nguồn.

Cơ chế lưu trữ phiên bản của Git là nó sẽ tạo ra một “*ảnh chụp*” (*snapshot*) trên mỗi tập tin và thư mục sau khi commit, từ đó nó có thể cho phép bạn tái sử dụng lại một ảnh chụp nào đó mà bạn có thể hiểu đó là một phiên bản. Đây cũng chính là lợi thế của Git so với các DVCS khác khi nó không “lưu cứng” dữ liệu mà sẽ lưu với dạng snapshot.

Có rất nhiều lợi thế để sử dụng Git trong việc lập trình:

* Git dễ sử dụng, an toàn và nhanh chóng.
* Có thể giúp quy trình làm việc code theo nhóm đơn giản hơn rất nhiều bằng việc kết hợp các phân nhánh (branch).
* Bạn có thể làm việc ở bất cứ đâu vì chỉ cần clone mã nguồn từ kho chứa hoặc clone một phiên bản thay đổi nào đó từ kho chứa, hoặc một nhánh nào đó từ kho chứa.
* Dễ dàng trong việc deployment sản phẩm.
* Và nhiều hơn thế nữa.

Sự khác nhau giữa Git và GitHub: Git là **các máy tính có thể clone lại mã nguồn từ một repository** và **GitHub chính là một dịch vụ máy chủ repository** công cộng, mỗi người có thể tạo tài khoản trên đó để tạo ra các kho chứa của riêng mình để có thể làm việc. Nói tóm lại Github thực chất là một cái giao diện dùng để thao tác GIT và chứa code trên đó

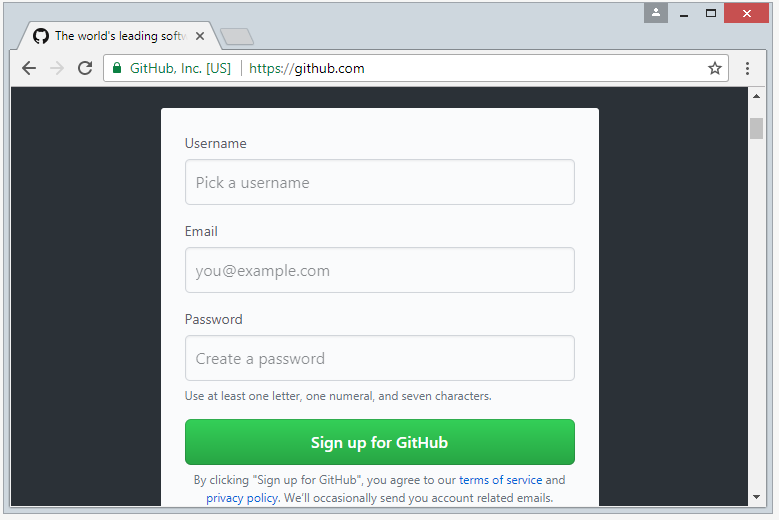


## Cách cài đặt.

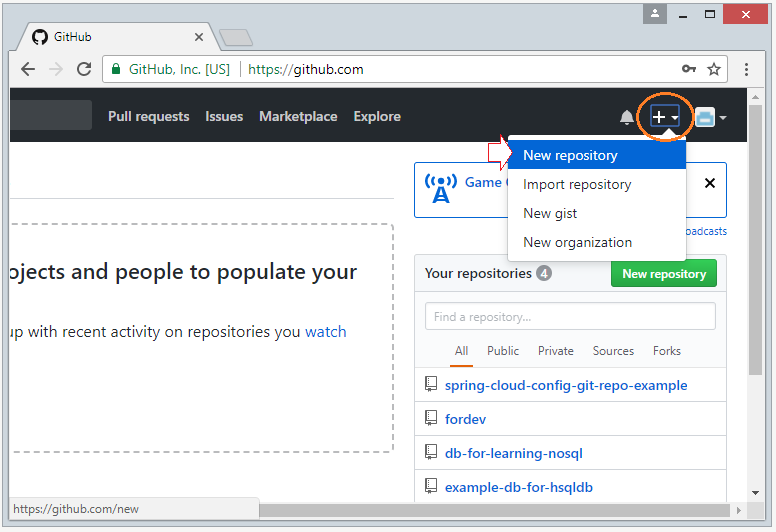
Để sử dụng trước hết cần đăng kí miễn phí một tài khoản GitHub bằng cách vào trang chủ của GitHub:

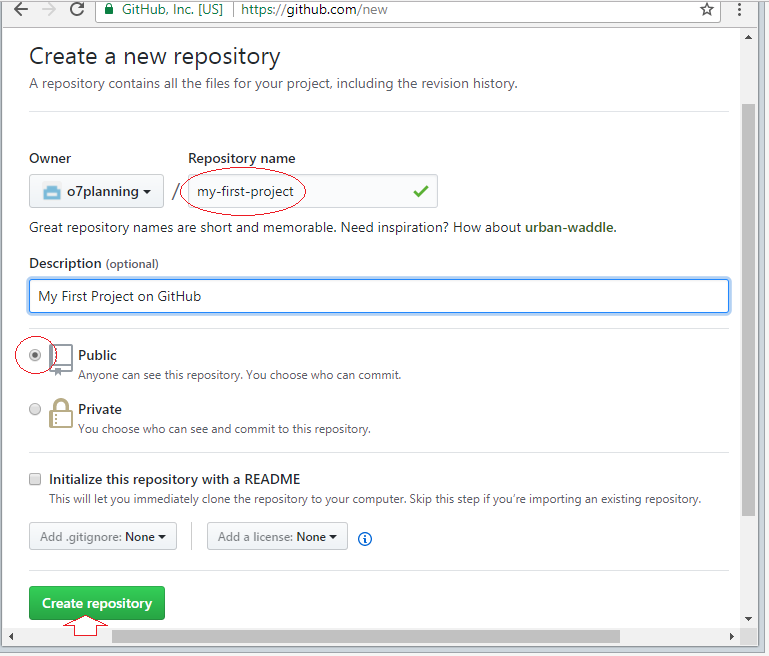
Link: [https://github.com](https://github.com/)

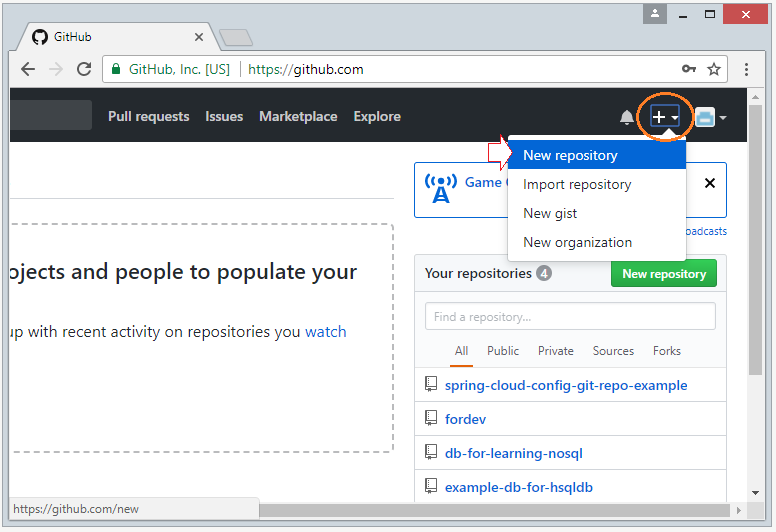
Sau khi click vào link sẽ hiện ra giao diện màn hình:

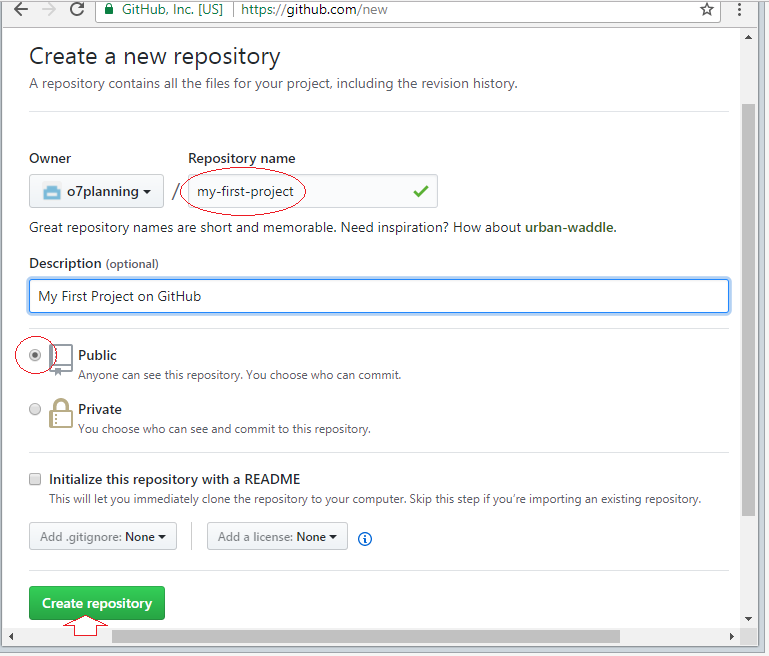


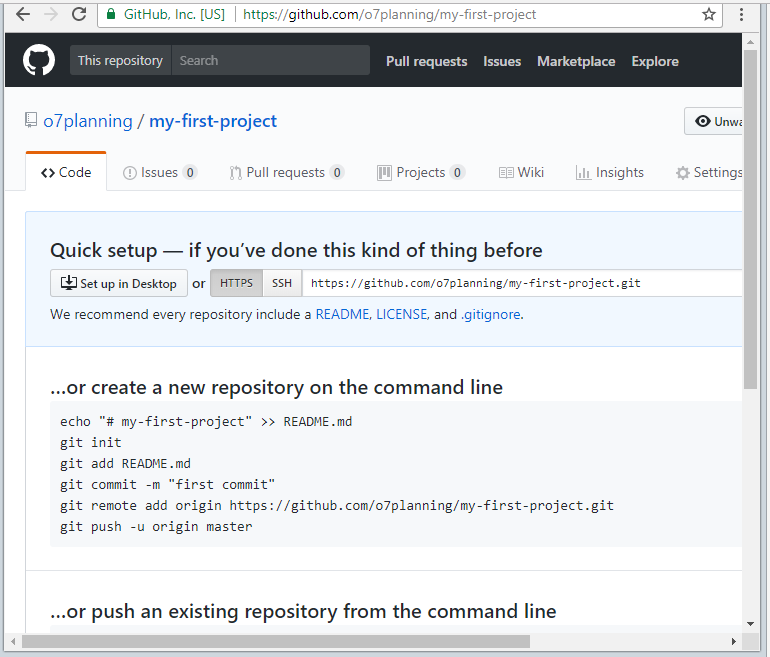
Tiếp tục là cần điền thông tin như username, email và password và vào email để xác nhận kích hoạt tài khoản. Sau khi đăng ký xong tài khoản **GitHub**, và đăng nhập vào. Bạn có thể tạo một **GitHub Repository**.





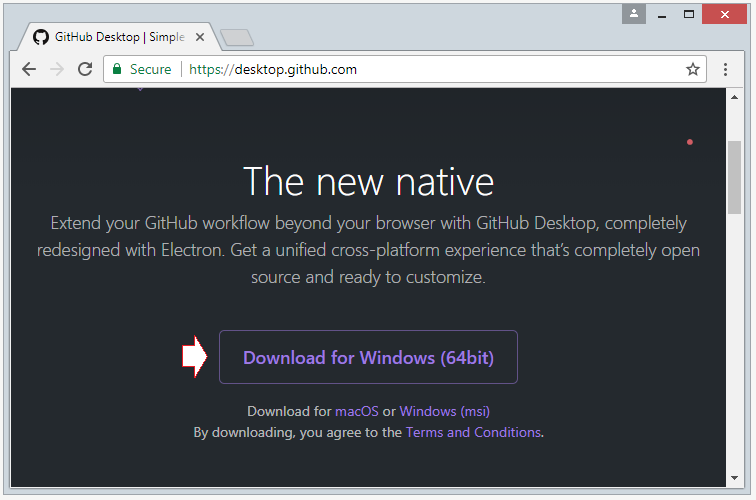


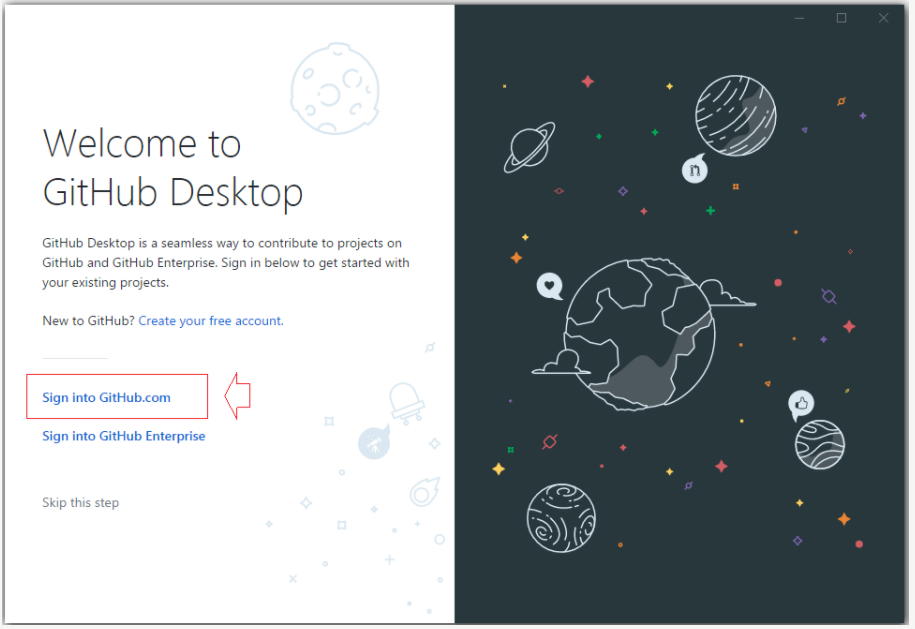




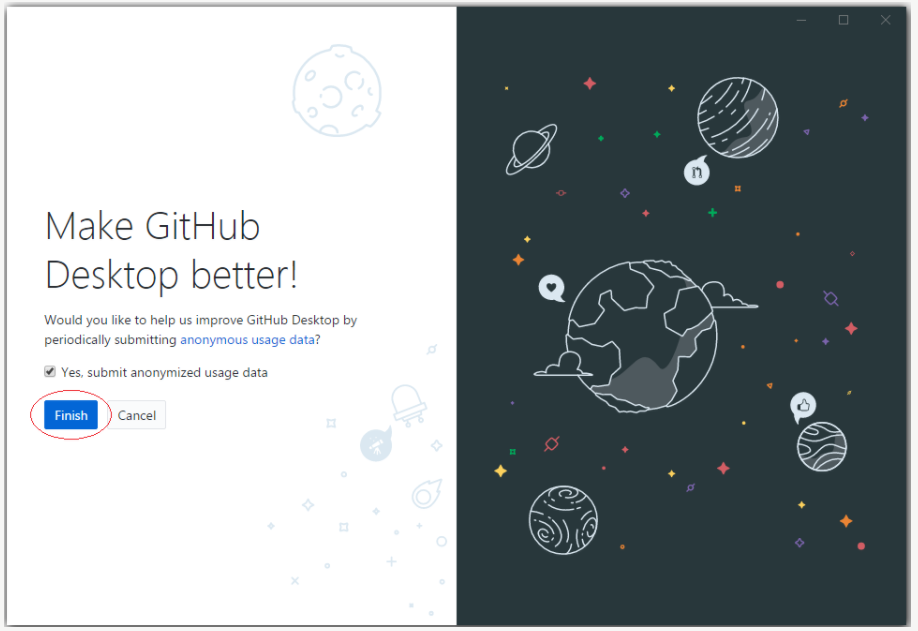
Như vậy, respository đã được tạo.

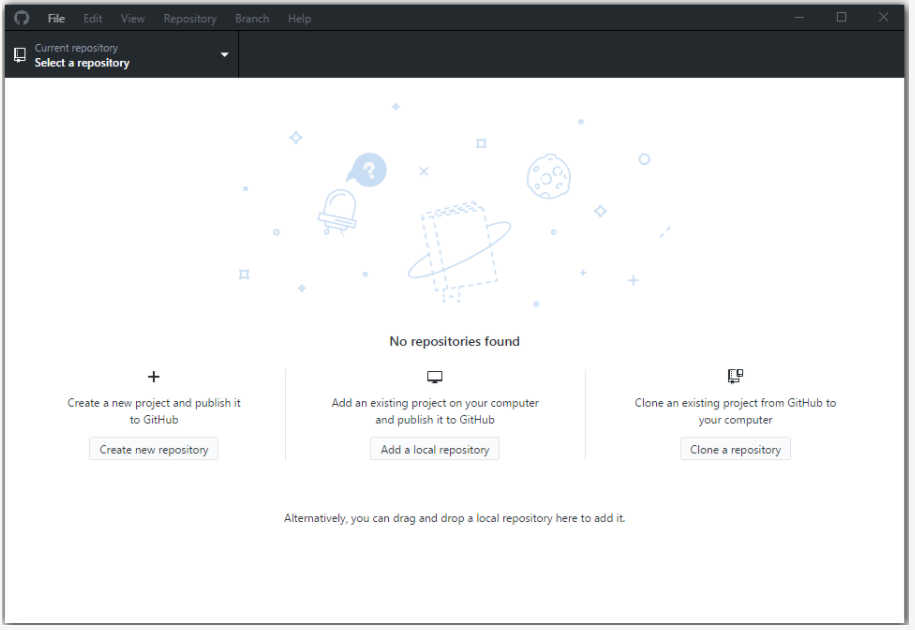
Download và cài đặt GitHup Destop tại link: <https://desktop.github.com/>



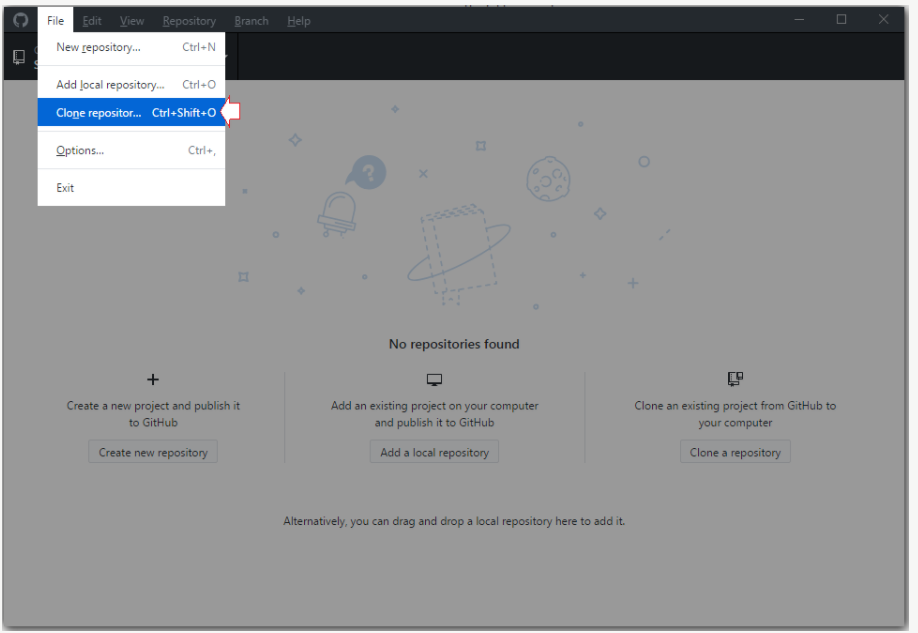


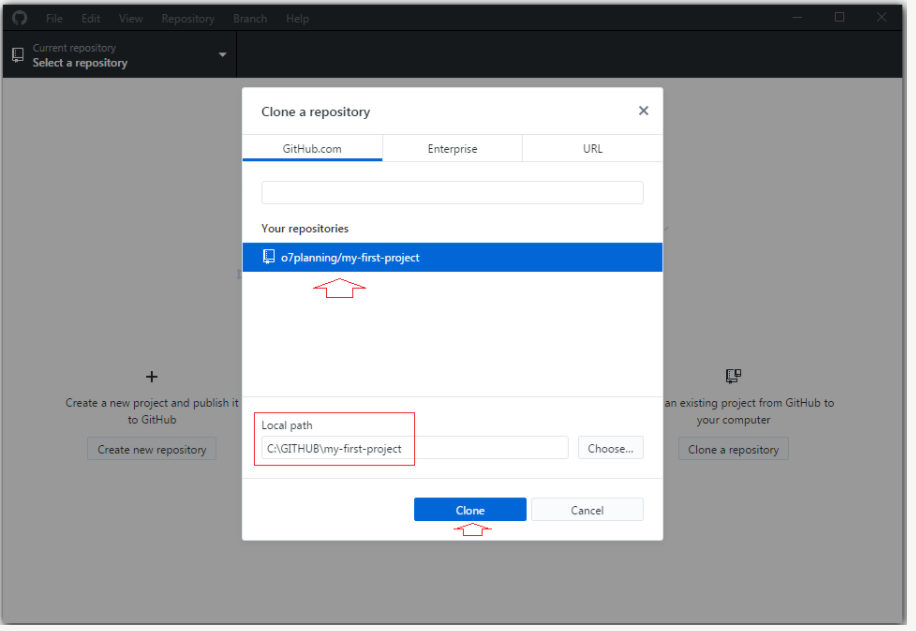
Sau đó đăng nhập tài khoản GitHub vào sẽ hiển thị giao diện màn hình:





Kết nối GitHub với GitHub Destop

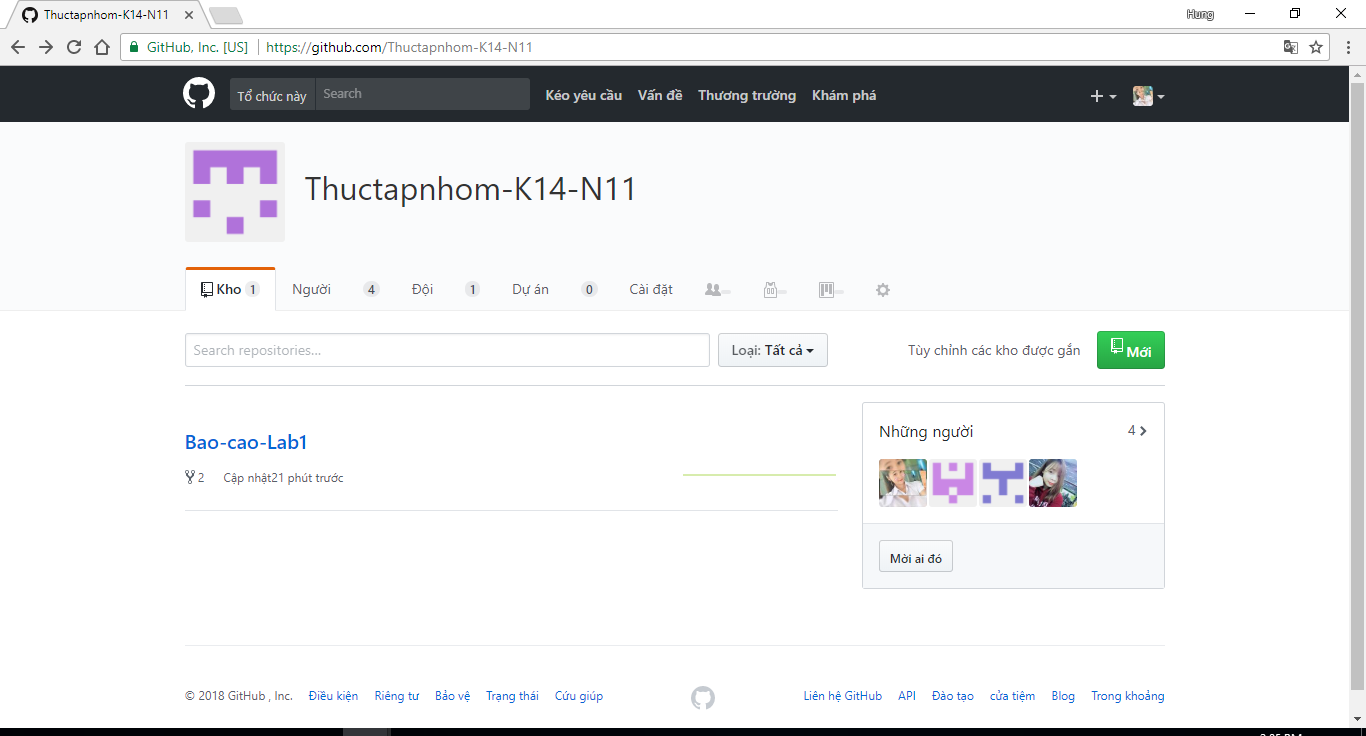




Lúc này trên **GitHub Desktop** bạn sẽ thấy một **Local Repository** đã được tạo ra.

# Tạo nơi chứa một dự án.

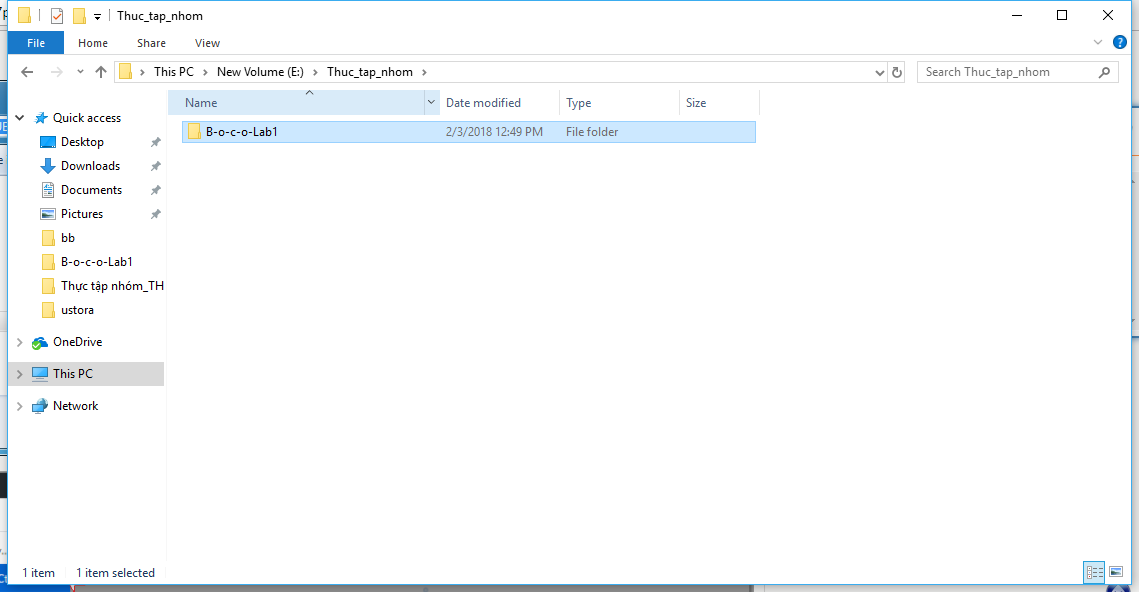
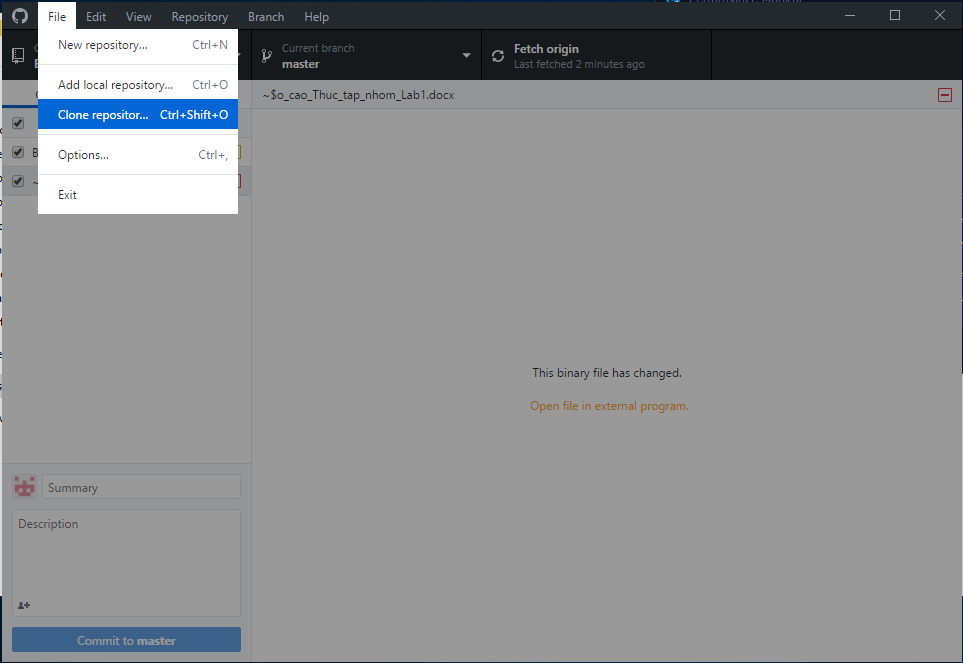
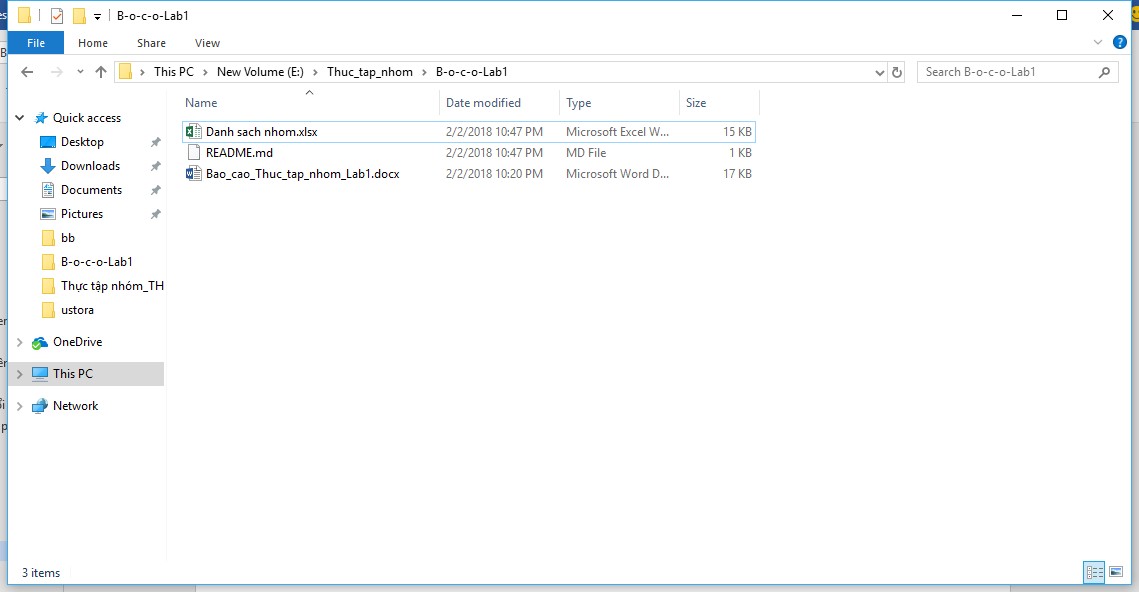
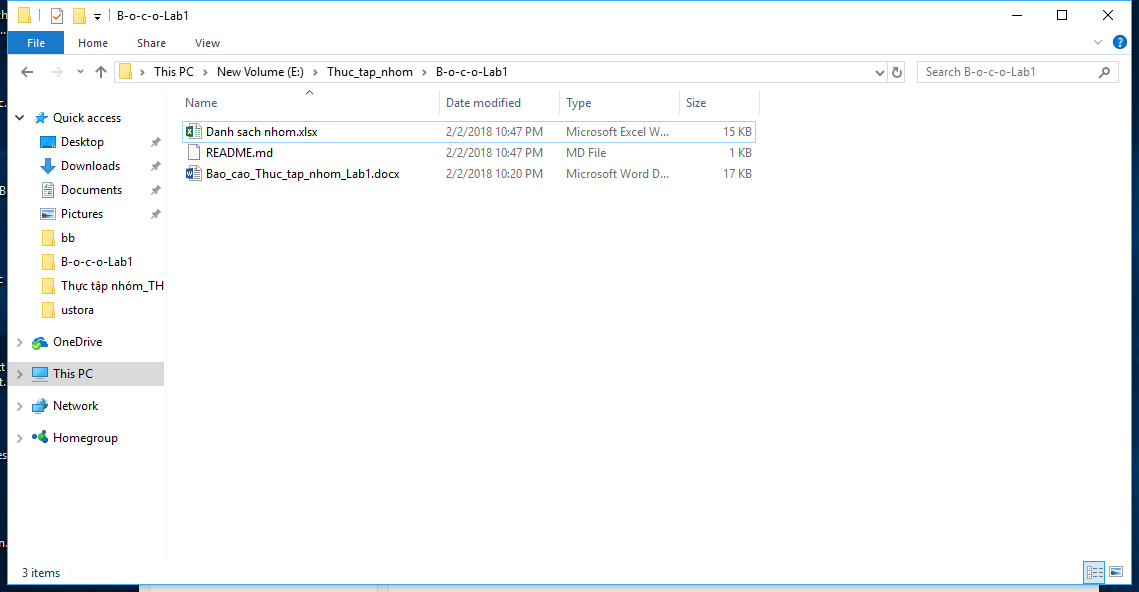
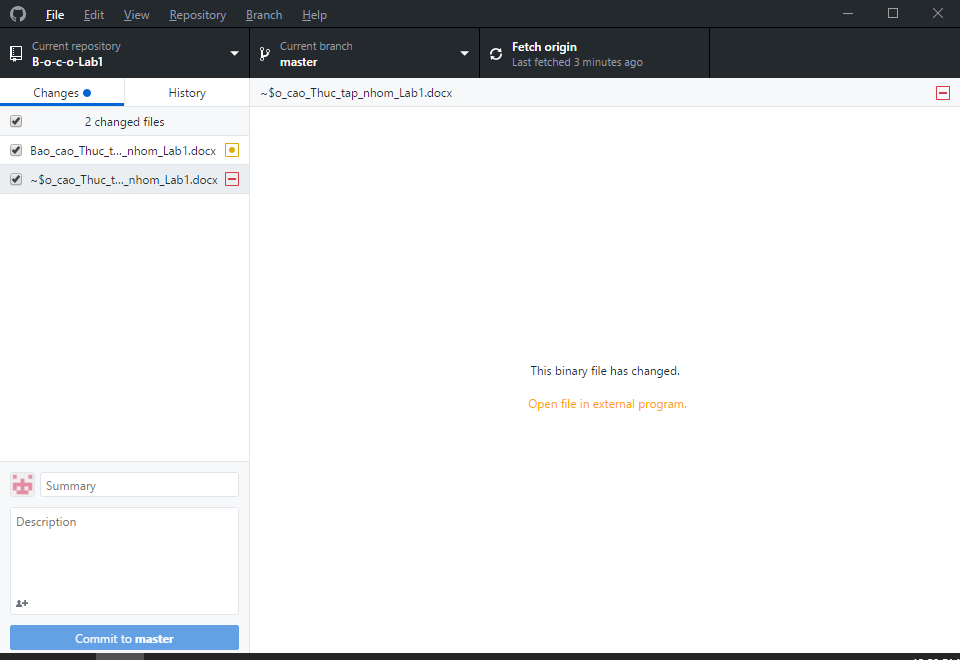
## Mô tả.

* Với tài khoản đã lập, tạo 1 tổ chức gồm 4 thành viên để cùng thực hiện công việc
* Tên là Thuctapnhom-K14-N11
* Tổ chức chung có đường link là <https://github.com/Thuctapnhom-K14-N11>. Tại đây lưu toàn bộ những project của nhóm thực hiện  
  
* Với tổ chức này t phân quyền cho các thành viên có thể xem, sửa và sao lưu
* Trong đây ta tạo từng Reponsitory ứng với mỗi project khác nhau
* Bên dưới là list những tài liệu mà thành viên đã đưa lên trong quá trình làm việc

## Cách lấy dữ liệu.

* Có 2 các lấy dữ liệu là lấy trực tiếp trên web hoặc thực hiện qua phần mềm hỗ trợ là GitHub desktop đc trình bày bên dưới

## Phần mềm hỗ trợ.

* + Sử dụng GitHub desktop để quản lý Local Repository trên máy tính cá nhân
  + Link download: <https://desktop.github.com/>
  + Sau khi download tiến hành cái đặt vào máy
  + Tiến hành mở GitHub desktop và đăng nhập để kết nối với GitHub trên web
  + Cách thức kết nối
    - Tạo 1 thư mục rỗng làm vị trí lưu dữ liệu trên máy tính cá nhân  
      
    - Trên GitHub desktop lựa chọn 1 Repository đã tạo trên GitHub để tạo 1 bản sao chép trên máy tính cá nhân  
      
    - Lúc này các dữ liệu ở Repository bạn đã kết nốt sẽ xuất hiện trong thư mục lưu trữ  
      
  + Cách lấy dữ liệu và upload lên GitHub
    - Ta có thể thao tác trên các file trong thư mục lưu trữ ở trên
    - Khi thay đổi dữ liệu ở các file thì trong GitHub desktop sẽ hiển thị các files đước thay đổi và các files thêm mới  
        
      
    - Tiến hành điền mô tả và click Commit to master để upload files lên GitHub
    - Sau khi commit ấn push to origin để upload

## Thử nghiệm.

# Thống kê lượt truy cập.