**Báo cáo SVN kết hợp redmine**

1. **SVN**
2. Định nghĩa

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

1. Tại sao nên dùng svn.

Như đã đề cập ở trên, SVN hỗ trợ làm việc dự án cho nhóm. Vậy nếu có dự án một người ta sử dụng được không? Câu trả lời là được, nhưng một người không thể tối ưu hóa tất cả các tính năng của SVN được, chỉ có thể sử dụng để lưu source code lại mà thôi. Vì vậy, làm việc nhóm là yêu cầu cơ bản để chúng ta nên sử dụng SVN. Thông thường làm việc nhóm thường gây ra nhiều vấn đề:

* Khi làm dự án, nếu khi mỗi người trong nhóm làm một phần, khi gộp lại sẽ rất khó khăn, có thể thiếu thư viện, đặt trùng tên, hoặc code không tương thích, …
* Việc chỉnh sửa hoặc cập nhật công việc phải gộp lại từ đầu rất khó khăn.
* Nếu xảy ra lỗi khó có thể sửa được, thì khó có thể backup về trạng thái ban đầu.
* Dễ xảy ra mất project nếu không lưu trữ cẩn thận.

1. Hệ thống quản lý phiên bản trong SVN (Version Control System – VCS) là gì?

VCS là một phần mềm giúp nhà phát triển phần mềm làm việc cùng nhau và duy trì đầy đủ tiến độ công việc (đang làm hoặc đã hoàn thành) của họ. Nghĩa là VCS giúp người dùng lưu trữ toàn bộ dự án công việc của họ.

Tại sao lại cần quan tâm đến phiên bản của SVN:

* Cho phép những nhà phát triển làm việc cùng một lúc. Vì vậy, nhiều người có thể cùng làm việc chung một source code dự án.
* Không ghi đè lên những thay đổi của nhau.
* Mỗi khi thay đổi version của dự án, hệ thống sẽ lưu lại toàn bộ version cũ.

1. Vòng đời của SVN:
   1. **Create reposity:**Tạo một reposity mới, mỗi dự án chỉ cần tạo reposity một lần. Và tất cả thành viên team sẽ làm việc trên chung một reposity đó.
   2. **Checkout:** Sử dụng để tạo một working place từ reposity. Working copy là một thư mục làm việc cục bộ (lưu trên máy) và những nhà phát triển thực hiện các thay đổi của họ, và sau đó submit các thay đổi này lên reposity.
   3. **Update:** Dùng để cập nhật working copy. Vì khi người dùng check out phiển bản mới nhất từ reposity, nếu người commit thay đổi lên reposity, thì phiển bản hiện tại sẽ bị cũ và ta cần update để lấy những thay đổi từ những thành viên khác về.

**d. Perform changes:**

d.1 Sau khi checkout, một thành viên có thể làm nhiều thay đổi khác đối với dự án. Chỉnh sửa là phương thức phổ biến nhất, có thể sửa nội dung file hoặc xóa nội dung trong file đó.

d.2 Người khác có thể thêm file hoặc thư mục. Nhưng khi thay đổi thì sẽ không thay đổi lập tức trên reposity mà được xếp vào hàng đợi, và nếu việc commit được chấp thuận thì đó sẽ được coi như một phần của reposity. Việc delete cũng xảy ra tương tự.

1. **Review changes:** Svn cung cấp cho người dùng phương thức status để xem thay đổi của working copy so với reposity, bạn có thể xem file nào đã thay đổi để dễ theo dõi nếu cần commit công việc. Để xem chi tiết sự thay đổi cần sử dụng phương thức *diff.*
2. **Fix mistake:** Trường hợp bạn đang làm dự án trên máy đã có nhiều sự thay đổi, nhưng bạn cảm thấy không cần thiết hoặc dữ liệu bị lỗi và muốn lấy lại dữ liệu ban đầu. SVN cung cấp phương thức reverts giúp working copy của bạn trở về phiên bản mà bạn mong muốn trên reposity. Có thể reverts một file/thư mục hoặc toàn bộ dự án.
3. **Resolve conflicts:**

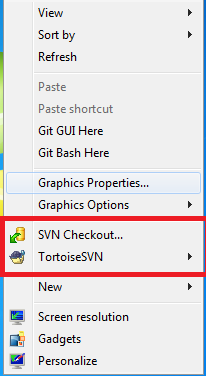
g.1 Xung đột (Conflict) xảy ra khi có nhiều developer cùng làm việc với một working copy và cùng tiến hành cập nhật trên một tập tin nào đó (Merge). Ví dụ A và B cùng checkout revision 10 của file index.js. Sau đó A sửa một hàm trong file đó sau đó commit lên revision 11, lúc này B cũng sửa và commit lên, nhưng server sẽ báo có phiên bản mới và B cần checkout, lúc này sẽ xảy ra conflict do SVN không biết phiên bản của A hay B là mới nhất.

g.2 Giải quyết: Lúc này, B sẽ xem lại code trong đoạn conflict đó, có thể giữ lại hoặc sửa, sau đó đánh dấu conflict đã được resolve.

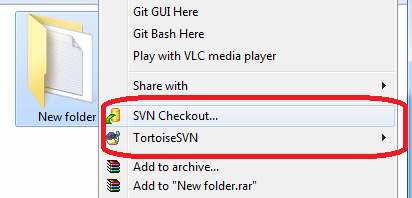
g.3 Một conflict khác xảy ra khi người dùng thay đổi working copy tới một url khác hoặc đổi tên, xóa file hoặc thư mục đó. Ở ví dụ trên là xung đột file (file conflicts). Một loại thứ 2 mới đề cập là xung đột cây (tree conflicts)

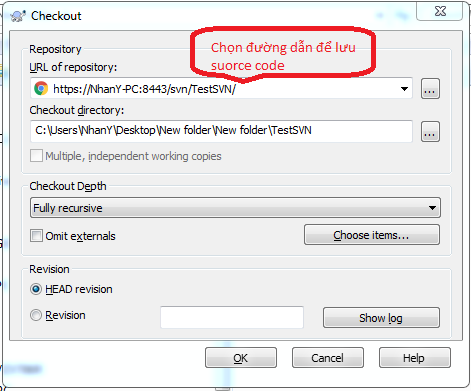
1. **Commit:** Thành viên team đưa các thay đổi ở working copy của mình lên reposity.
2. Sữ dụng SVN.

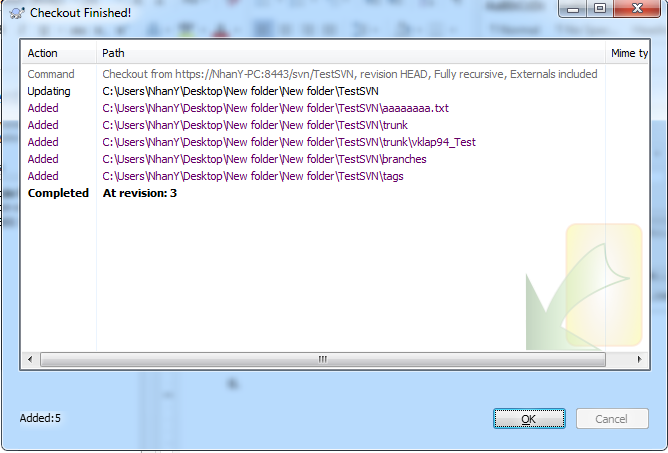
Sau khi cài đặt xong, nhấn chuột phải ta sẽ thấy như hình.



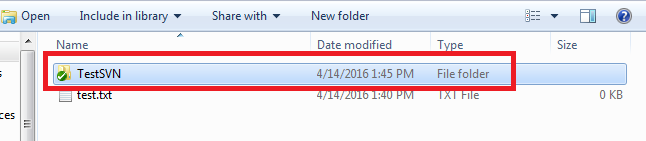
Tạo một thư mục để test như hình

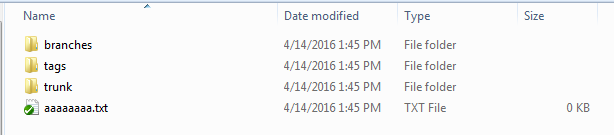






Sau khi nhấn OK source sẽ được lưu lại thư mục sẽ như hình sau





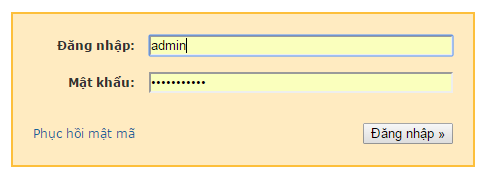
1. **Redmine**
2. Redmine là gì?

[Redmine](http://www.redmine.org/) là một công cụ quản lý dự án dựa vào web mà tôi sử dụng tích cực. Nó mạnh, chạy trên Ruby và Rails, và được cấp phép theo GPLv2.

1. Sữ dụng Redmine

Sau khi cài đặt Redmine để truy cập trên localhost ta vào: <http://127.0.0.1/redmine>

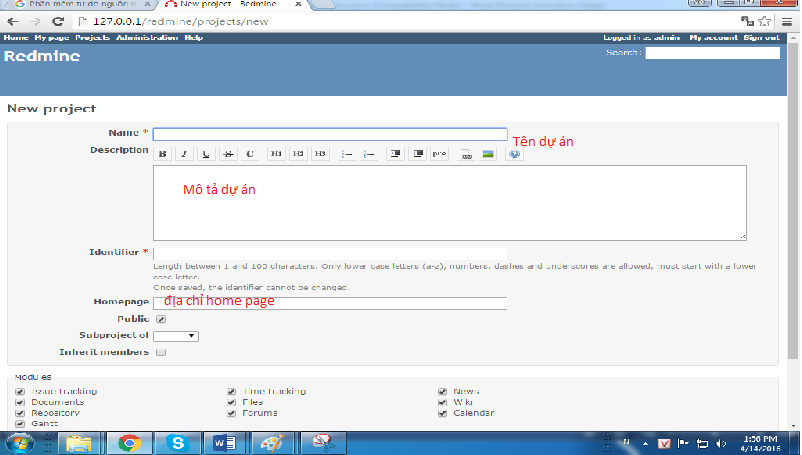
Có màn hình giao diện đăng nhập và vào trang home.



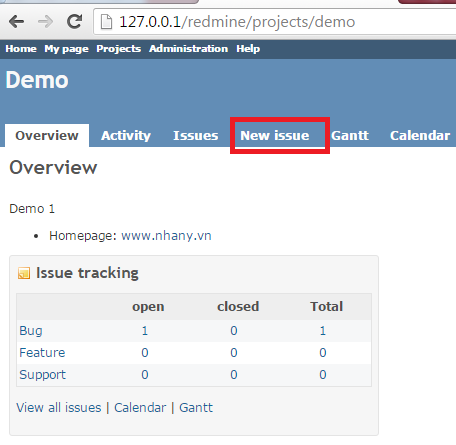
Màn hình giao diện chính



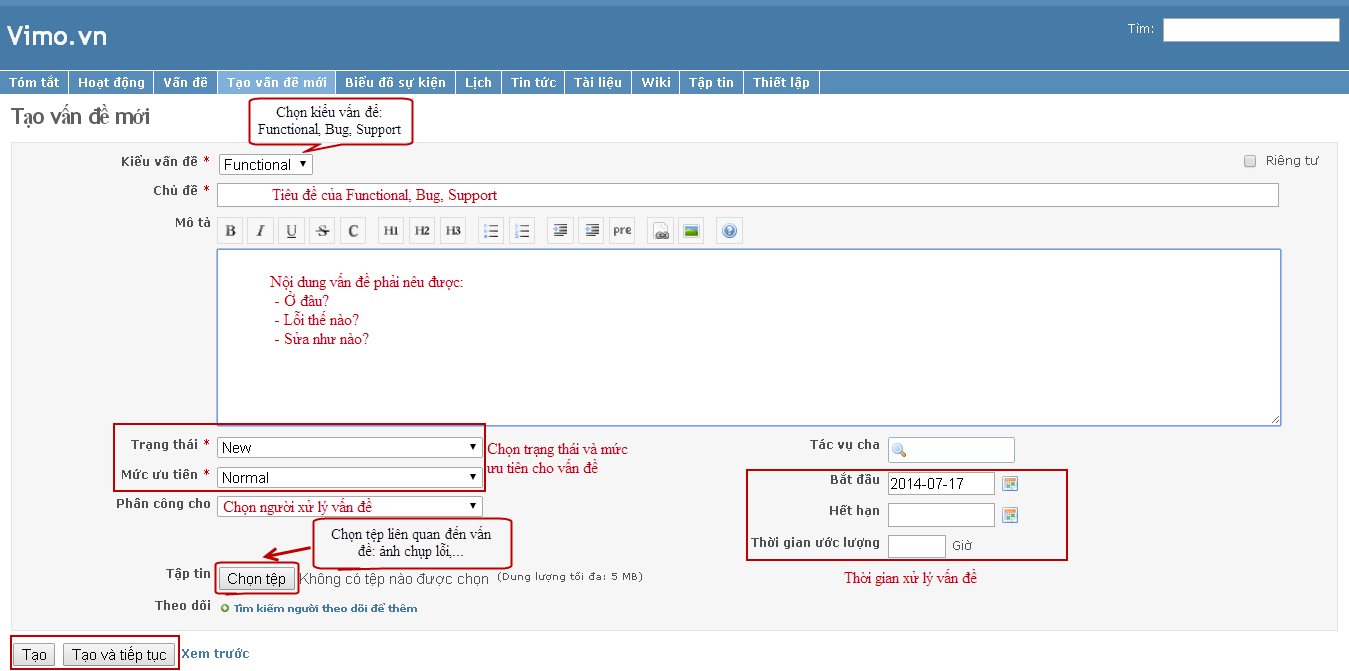
Ta chọn mục Projet được bôi đỏ như hình trên để xem các dự án nếu chưa ta tạo một dự án mới như sau



Xong ta tạo các vấn đề cho project bằng cách nhấn chỗ bôi đỏ trên hình

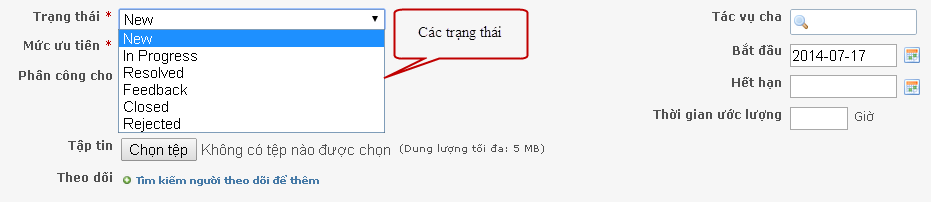


Ta tạo các vấn đề như hình sau



Hoàn thành tạo một vấn đề

1. Các trạng thái của vấn đề vừa mới tạo

  
3.1 Ý nghĩa các trạng thái

* **New:** Lỗi khi được tạo mới và log lên hệ thống
* **In Progress:** Lỗi đang trong tiến trình xứ lý.
* **Reject:** Lỗi tạo ra không phải là lỗi hoặc không phù hợp với thiết kế, nằm ngoài thiết kế
* **Resolved:** Lỗi sau khi được Deverloper sửa và đang được kiểm tra lại
* **Feedback:** Sau khi kiểm tra lại lỗi mà Deverloper đã sửa nếu kiểm tra lại không đạt yêu cầu có thể Feedback
* **Close:** Lỗi đã được sửa và test lại thành công

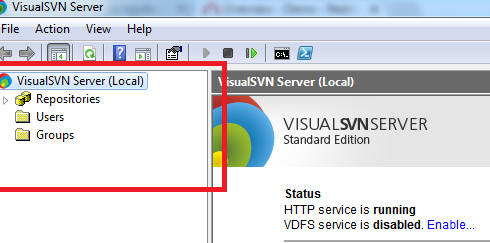
3.2 Vòng đời của một issue

## 

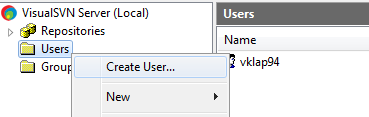
1. **Kết hợp giữa SVN và Redmine**
2. Chuẩn bị

Sau khi cài đặt SVN ta cần cài thêm visual SVN

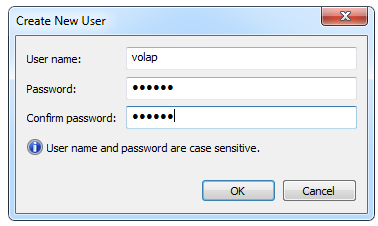
Hoàn tất việc cài đặt visual SVN ta có giao diện sau đây



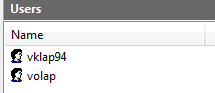
Tạo một user mới cho SVN nhấn vào User như hình sau



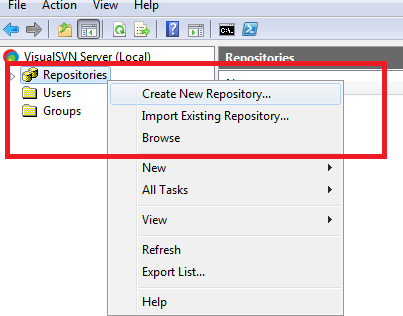
Tạo user và password cho user

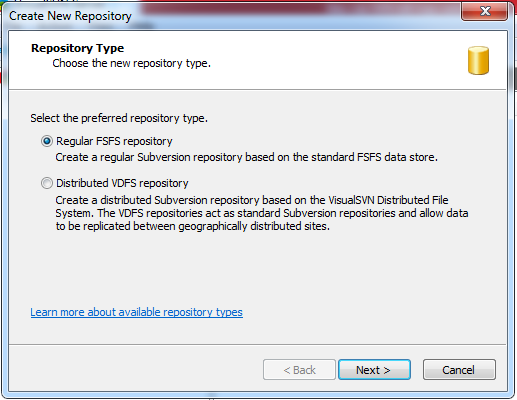


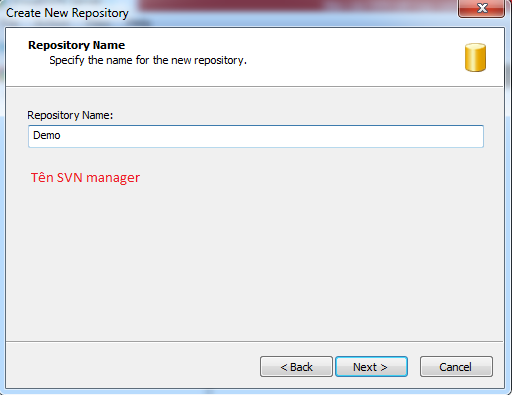
Nhấn OK sẽ hoàn thành việt tạo User



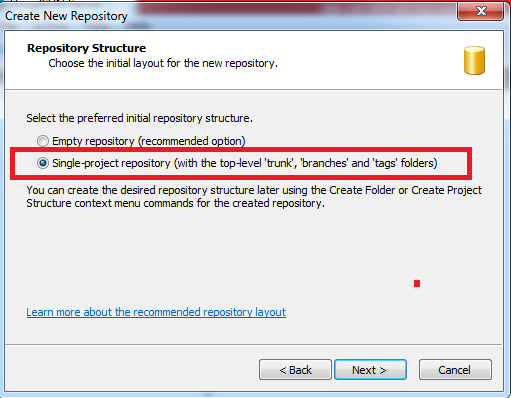
Tạo một SVN manager



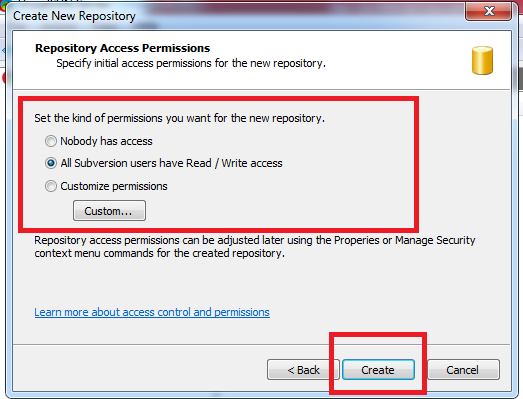




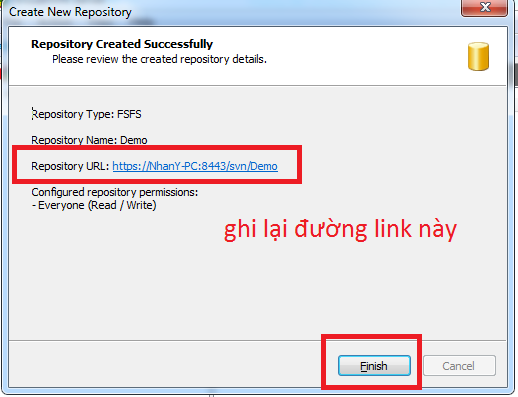
Chọn như hình dưới để có một thư mục SVN đầy đủ



Tác vụ phân quyền như hình dưới



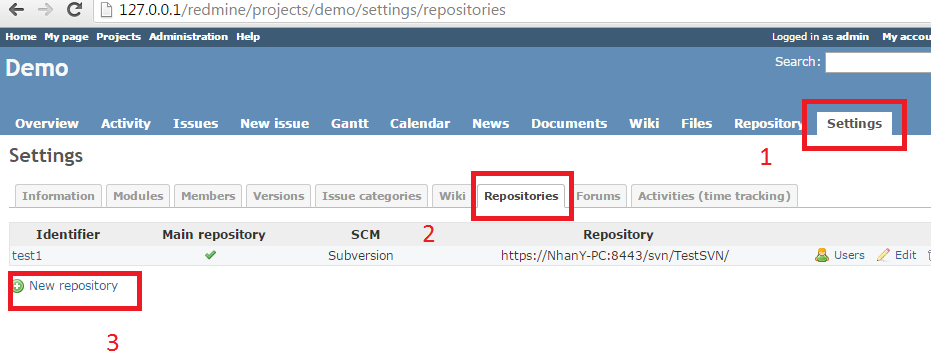
Nhấn create



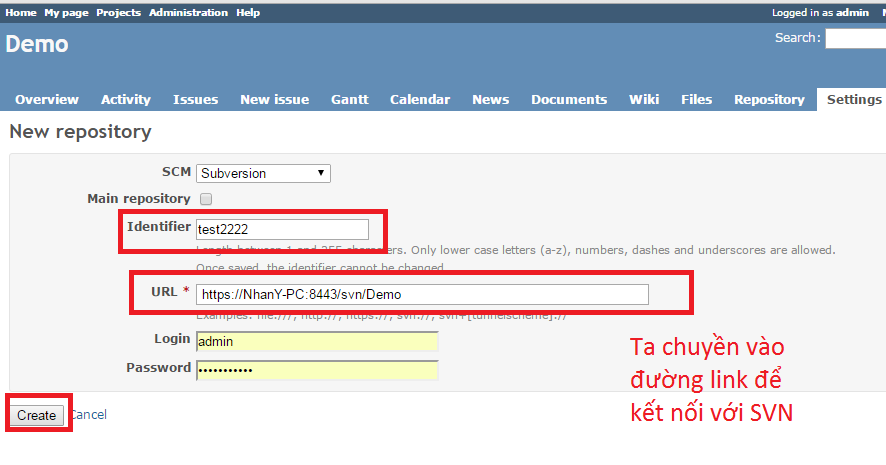
Nhấn finish.

1. Kết nối với Redmine

Sau khi tạo xong dự án và vấn đề dự án làm như hình sau



Ta điền các thông tin như hình dưới và nhấn create



Nhấn vào dấu bôi đỏ trên hình để thấy được các dữ liệu trong svn đã được load lên trên redmine

