注：红色表示存在问题，绿色表示已经clear

需要实现的功能：

1 能上传文件(该功能暂时废除)

1.1 实现工具：前端:jquery上传组件，后端：svnkit

1.2实现步骤: 1 检查用户是否具有本目录下的权限

2 判断是否有重名文件

3 无重名文件便上传，有则在其后加序号上传。

2能查看有权限的目录

2.1 指远程查看svn服务器上有权限的目录。（仅查看服务器端的目录，格式可参考github）

2.2 实现工具：svnkit

2.2实现步骤：1 直接调用svnkit显示目录内容.

2 如果显示的目录数量>10，则只显示10个，其他的目录暂时存在collection中。

3 对于属性为文件夹的，提供链接。属性为文件的，则不提供链接。

4 当点击链接时，则进入文件夹，显示该文件夹的内容，后转至第二步。

5 如果文件夹中无文件，则终止。

1 在文件列表上方显示文件夹的地址如(Master/项目名/文件夹/)

2 当点击Master或项目名或文件夹可返回所在层次。

3能下载有权限的目录(这个可能不必要，因为tortoiseSVN工具可能下载更为方便)

可以考虑只实现将项目文件夹所在的http地址复制到clipboard，然后在tortoiseSVN中填入http地址，然后下载，可能比下载一个zip文件更为方便。

3.1 实现工具：svnkit，j2ee

3.1 下载过程：1点击想下载的项目目录整个打包称zip压缩文件下载。

4查找文件功能

4.1实现功能：可通过一个界面在不同的项目内查找文件名部分或全部匹配的文件或文件夹

4.2 可能的实现工具:apache lucene3(?), svnkit

4.3 实现步骤：1 当输入文字时，搜索该项目下的文件名模糊匹配的文件或文件夹。

(搜索过程的实现可能比较复杂)http://svn-search.sourceforge.net/



5 开启项目顶层目录并赋予相关人员权限

5.1 当项目合同签订后，建立顶层目录，并设置相关开发、测试、主管、项目经理的权限。

5.2 实现工具:svnkit（建立文件夹），python(设置权限?) <http://blog.chinaunix.net/uid-25525723-id-363023.html>

5.3 实现步骤：1点击网页上的建立目录按钮

2 调用svnkit中的editor(?)建立项目文件夹及其下的(仅拥有操作repository级别的角色可建立，其他人无权建立)。

3 建立11个文件夹（项目计划，用户需求等）

4 设置所有文件夹的权限（python? Hook? Java?）

6可能的实现工具

6.1 前台:jsp,js,jquery,html5,css3

6.2 后台:spring,spring on boot, spring mvc, apache lucene, svnkit,python（可能？）

6.3 查看有权限的目录可能的实现过程：1 后台可实验利用svnkit搜索同层的文件及文件夹。

在家中笔记本上安装maven,svn,idea,jdk

2 选择其中其中一个点击进入下层文件夹

3 试验spring,spring on boot,spring mvc

4 试验前台，分页？

5 先学习html5,css3,javascript,jquery,bootstrap

6 先利用html,css制作一个静态的网页

7 在利用javascript,jquery,bootstrap制作动态网页

8 利用spring on boot,spring mvc开发一个示例。

6.4 能下载有权限的目录可能的实现过程:1 在web端点击copy to the clipboard按钮。

2 研究如何将所选内容复制到clipboard上。

6.5 查找文件功能可能的实现过程:1暂时不确定如何实现。可能有两种实现办法，第一种是利用数据库，将所有的文件名存储在数据库中方便查找。第二种是index所有的repository,然后利用，然后用apache lucene检索所有用户具备权限的index。对于第一种方式，可以每新建一个项目目录新建一个表，表中记录文件名，作者，时间，地址等信息。利用sql语句快速搜索。第二种方法怎么做尚不清楚。

6.6 开启项目顶层目录并赋予相关人员权限可能的实现过程:1利用svnkit新建文件夹

2 利用svn hook捕捉事件，在建立文件夹后设置权限(可行性有待验证)。

20：20分

讨论后发现实现两棵树暂时找不到办法。6.6暂时是需要实现的功能。

1月26日

1 志华的问题：在华安时候代码管理怎么搞，是需求对应代码还是一个版本对应代码？

公司总体项目数量(维护的，在开发的等)为706个。

2 6.6的功能需要实现，可以通过svn脚本实现，具体实现方法需要研究。

第二个需要实现的功能是在项目开发到某个时间节点的时刻（可能与任务系统的流程引擎有关）打branch,比如alpha,beta,cr,stable时刻。

1月28日

本星期的主要工作是上传某个文档文件(例如xxx.doc)至svn服务器，如果之前目录没有该文件，则直接上传。如果有该文件，则会返回SVNException，之后重新commit上去。

使用string中的contains方法可用于部分匹配SVNException.getMessage()。

2月1日

1 在项目的伊始阶段，在某个界面中填入某个repository的svn地址和svn的用户名和密码！！！

2 在任务系统上传文档文件流程：

1从浏览器端获取所需要上传的文件的名字。

~~2 依据上传界面判断上传文件的所在目录，在目录下搜索同一文件是否存在，如果存在则判断该文件的用户(getAuthor)是否为同一人，如果是，则覆盖该文件，如果不是，则提醒该用户改名字(???)~~

2 判断上传文件所在目录，在目录下搜索同一文件是否存在，如果存在，则提示用户选择覆盖（已经完成）或是新增。新增的话则重新命名（重命名方式：文件名+服务器端日期，精确到秒）。

~~文件名冲突的时候：1 提示用户选择 覆盖 还是 新增（重命名方式：文件名+日期）~~

~~传输多个文件的时候？~~

3 权限分配问题

1 根据志华给出的菜单设计和目录规划，建立起一张表格每个repository下文件夹的权限分配表。

2 可能的实现过程：1 部门助理新建一个版本库（版本库其实可以靠svnkit建立本地的版本库，但是ubersvn估计是不认的。。。。）。

2 将版本库的地址填入任务系统后，系统自动建立trunk,branches,tag(ubersvn可自动建立),之后再在trunk目录下自动建立起11个文件夹（包括00.项目计划----11.规范）（2月3日完成），并为这11个文件夹自动分配权限(可能需要写hook script)。

~~分配权限的方法：为repository选择参与该项目的所有人员。之后设置11个文件夹的角色权限。~~

2月3日

1 ~~实验是否可以通过svnkit库中的函数创建一个版本库，是否能被ubersvn识别。（已经成功，但是暂时不需要）~~

2 ~~正在思考是否需要做一个web端的svn浏览界面，可参考的原型包括visual svn server的浏览界面，sventon(github地址：)~~

3 自动建立repository（visual svn server上已经建立好了），剩下在uberSvn上的建立还没尝试（暂时不需要）

4 回家看一下JetBrains的TeamCity是什么？该怎么用？TeamCity是用于持续集成的工具

5 根据刚才的试验，似乎在更改了设置（比如某个repo的权限）之后需要重新启动apache的service ，该步骤是否可以自动实现有待验证。

2月4日

1 经过测试后发现设置alternative\_svn.authz文件，可定义角色，角色中的成员（必须在uberSvn中存在），实现顶层版本库的权限管理。文件夹层级的管理已经实现 ~~（文件夹层级的管理尚未实现）~~

2 经过实际分析与试验，alternative\_svn.passwd文件是uberSVN管理用户，密码的文件。Apache httpd是一个apache的密码管理程序。

3 志华的要求：新建项目，授权，人员离职，新来员工加人。

2月5日

1 现在公司中项目数量总记706个，如果每个项目（独立的版本库）都有一个权限控制文件，那么就会有706个权限控制文件。

2 经过测试，uberSVN支持每个版本库使用独立的权限控制文件，只需要在C:\Program Files (x86)\WANdisco\uberSVN\conf\conf.d\50-repositories文件中找到对应的版本库的location，然后修改权限控制文件位置。之后将生成好的权限控制文件放入指定的文件夹中。

可能的实现技术：

Java, Apache Lucene(全文检索)

注：公司员工的角色定义是固定的，不得更改。

授权过程：1 在某个界面中输入属于该项目的成员（多个），依照该成员所属的用户组依次写入权限控制文件中对应的组，然后对于每个文件夹建立用户组的权限。

员工离职流程：1 在用户管理界面中选中用户，然后将其状态改为离职。

2 利用apache lucene打开C:\Program Files (x86)\WANdisco\uberSVN\conf\svn.passwd文件，将其用户名所在的行删除。

3 利用apache lucene搜索所有版本库中的权限控制文件，将存在该用户名字的删除。

4 完成。

5 新员工加入流程：1 设置用户的用户名，密码。

2 设置用户所属的角色。

3 完成设置。

6 新建项目流程： 1 项目助理登录验证是否具有新建repository的权限。

2 填写repository的名称，然后系统检查是否重名。

3 没有重名则创建新版本库，如果有重名，则重新填写版本库名称。

4 创建版本库。

2月14日

试验项目：1 ~~利用svn某个用户的名字建立索引。~~

2 尝试编写ini文件解析器，用以解析权限文件和全局用户文件(可能)，可能之后可以结合apache lucene一同使用。

3 使ini文件解析器具备读写功能。利用状态机分析ini文件。

4 100个authz权限文件的建立索引完成用时23秒，搜索只需0.1秒，成功。

5 删除512个权限文件内的指定用户成功。

6 不使用lucene，一个一个搜索700个文件时间几乎一致，在1秒内。

2月15日

试验项目：1 ~~试验apache tika解析权限文件。~~(tika主要用户文件内容的抽取，比如网站某个html页面的内容抽取)，解析权限文件意义不大。Nutch是一个开源的web搜索引擎。试验后发现，解析权限文件不行。可抓取网络上的内容(爬虫)。（某些网站不允许网络爬虫）

2 试验自写解析器，解析ini权限文件，使其能添加组用户，定义组用户的读写权限。

3 试验自写50-repositories.conf文件解析器，使其能自定义地添加或删除文件中内容。

2月16日

1 根据搜索发现，解析器可以参考编译原理中的词法分析（可能还有语法分析）。还可以参考conf-c或者是conf配置文件解析.c，完成java版本的ini文件和conf文件的解析。

~~2 根据配置文件写新的~~

2月18日

1 uberSVN的密码管理，配置文件利用apache http server进行管理。分析uberSVN的目录结构的分析后如下：

1.1 uberSVN的web管理页面存放位置:C:\Program Files (x86)\WANdisco\uberSVN\tomcat\webapps\ubersvn

可以看出已经编译好，放在tomcat的容器中。

1.2 uberSVN的C:\Program Files (x86)\WANdisco\uberSVN\conf\httd.conf用于配置apache http server， 其中Include conf/conf.d/\*.conf表示conf.d目录下的文件加为头文件。

1.3 svn.passwd文件中存储svn用户与用户名,MD5加密。操作该文件的程序在C:\Program Files (x86)\WANdisco\uberSVN\bin中httpd（从apache http server中拷贝而来）。httpd命令的使用方法：<http://www.linuxidc.com/Linux/2012-11/73915.htm>

2月19日

1 设计助理操作SVN的命令。志华的要求：新建项目，授权，人员离职，新来员工加入。

1.1 版本库管理：

SvnRepo -[c|cu|d|r|q] 项目名 (项目名)

-c 表示create，创建新版本库，目录结构按照trunk,tags,branches,trunk目录下doc文件夹建立11个文件夹。

-cu 表示custom，定制，选择该选项，不会创建标准的目录结构，仅会创建braches,tags,trunk

-d表示delete，删除版本库

-r表示revise, 修改版本库名字

-q表示query，查看版本库，如果填写版本库名字，则能查看版本库是否存在，如果不填写，则默认显示所有版本库。

例1：SvnRepo -c 项目管理系统

创建项目管理系统版本库

例2：SvnRepo -cu 项目管理系统

创建自定义的项目管理系统版本库(至包含trunk,tags,branches)

例3：SvnRepo -d 项目管理系统

删除项目管理系统版本库(慎用！！！)

例4：SvnRepo -r 项目管理系统 代码管理系统

将名字为项目管理系统的版本库名字改为代码管理系统。

例5：SvnRepo -q 项目名

查询某项目名是否存在

~~1.2 添加项目组成员：~~

~~Svntmem (组员名字)\*~~

~~最多可一次添加10个用户。~~

1.3 员工权限管理：

SvnAuthz -[a|d|r|q] (员工名字)\* [-r] (角色)\* [-p] (项目名)\*

-a表示add，表示添加用户，分配角色。

-d表示delete，表示删除用户角色。

-r表示revise，表示修改用户角色。

-q表示query，表示查找某位员工，或某个全局角色，或某个项目中某个人的角色或者某个角色有多少人。

-r表示role，表示操作所涉及到的角色。

-p表示project，表示操作所涉及到的项目。

具体用法:

例1：SvnAuthz -a (员工名字)\* -r (角色)\*

增加一位或多位员工一个或多个角色。

例2：SvnAuthz -a (员工名字)\* -r (角色)\* -p (项目名)\*

增加一位或多位员工的在一个或多个项目中的角色。

例3：SvnAuthz -d (员工名字)\*

删除一位或多位员工的在所有项目中的角色和全局角色。

例4：SvnAuthz -d (员工名字)\* -r (角色)

删除一位或多位员工的全局角色

例5：SvnAuthz -d (员工名字)\* -p (项目名)\*

删除一位或多位员工子在一个或多个项目中的所有角色。

例6：SvnAuthz -d (员工名字)\* -r (角色) -p (项目名)\*

删除一位或多位员工在一个或多个项目中的角色。

例7：SvnAuthz -q (员工名字)

查询某位员工的所有项目的角色和全局角色。

例8：SvnAuthz -q (员工名字) -r

查询某位员工的全局角色。

例9：SvnAuthz -q (员工名字) -p (项目名字)\*

查询某位员工在一个或多个项目中的角色。

例10：SvnAuthz -q -r (角色)

查询某个全局角色的所有员工。

例11：SvnAuthz -q -r (角色) -p (项目名字)

查询某个项目的某个角色的所有成员。

例12：SvnAuthz -q -p (项目名字)

查询某个项目中所有成员的角色。

例13：SvnAuthz -r (员工名字)\* -r (角色)

修改一位或多位员工的全局角色。

例14：SvnAuthz -r (员工名字)\* -r (角色) -p (项目名字)

修改某个项目中一位或多位员工的角色。

1.4 用户管理

Svnuser -[a|d|r|q] (员工名字)\* (用户密码)

-a表示add，添加员工与密码。

-d表示delete，删除员工名字，最多一次可删除10个。

-ru表示revise，修改用户的名字。

-rp表示revise，修改用户密码。

-q表示query，查询某位用户是否存在，或者遍历所有用户。

具体用法：

例1：Svnuser -a kr.huang 123

表示新增员工kr.huang，密码：123

例2：Svnuser -d kr.huang kk.wang

删除kr.huang，kk.wang员工

例3：Svnuser -ru kr.huang kk.huang

将员工名字从kr.huang修改为kk.huang(注意顺序！！)

例4：Svnuser -rp kr.huang 123

将员工的密码修改为123

例5：Svnuser -q

查询所有用户。

例6：Svnuser -q kr.huang

查询kr.huang是否为员工

~~Svnuser (组员名字)\* 项目名字~~

~~-a表示add，在项目中添加组员名字，最多一次可添加10个~~

~~-d表示delete，在项目中删除组员名字，最多一次可删除10个~~