# DocSys文件管理系统设计

## 1、系统简介

DocSys是一个网页版本的文件管理系统，方便用户进行文件存储和管理，用户操作需要尽可能的接近本地目录的操作习惯（例如，拖放、右键新建、删除、修改等）。

为了方便用户对文件进行分类管理和不同的存储系统，系统支持多仓库，不同仓库有自己独立的目录结构和权限管理，不同仓库可以采用不同的存储方式。

## 2、概要设计

系统参考了Wiki、自由团队、自由团队管理后台三个系统的界面设计，为了保证界面的风格统一但又相对独立，界面部分代码基本还是分开目录存放，但会使用相同的css和js代码。

### 传统文件系统实现

传统文件系统设计需要以下几个关键信息：

1. 块位图信息：用于表示数据块的使用情况
2. 文件节点位图信息：用于表示文件节点的使用情况
3. 文件节点信息表：与文件节点一一对应，用于描述文件/目录的信息
4. 数据块：数据存储区

传统文件系统实现有两个难点：

1. 在于文件操作（增、删、改）过程中如何维护块位图信息和文件节点信息的正确性，这个是保证文件系统稳定性的关键，因此保证文件节点的写操作是原子操作是核心
2. 文件的查找速度，文件系统总是通过文件名找到文件节点索引，才能实现对文件内容的访问，因此一个快速高效的搜索算法是非常必要。

### 网页版文件系统实现

网页版文件系统的文件存储部分是建立在操作系统的文件系统之上的，因此不需要考虑维护块数据位图的维护，可以假设每个文件节点只使用一个数据块，而且这个数据块可以自由伸缩（因为这个数据块实际就是一个真实的文件），因此我们设计需要考虑的就是如何维护文件节点位图和文件节点信息表。

DocSys采用数据库来保存文件节点信息，相当于将节点为图和节点信息表合在了一起，每条存在一条记录就表示一个文件节点，不需要考虑如何去维护文件节点位图，但需要保证文件节点的写入操作是原子操作（即同一时间只能有一个用户在进行写入操作）。

以上提到的原子操作，只是为了保证文件系统底层数据的正常，但对于网页版文件系统而言，做到这一点还是不够的，比如用户在编辑文件的过程中（没对文件进行写操作），发现文件内容突然被其他给修改了，因此作为网页版文件系统，不仅要保证底层文件的原子操作，还需要保证在用户层的原子操作（即用户一旦启动了编辑模式，该文件的写入操作就杯占用，直到用户主动释放或者系统超时自动释放）。

### 文件下载

支持多文件下载和单目录下载；文件下载使用数据流方式下载，避免用户直接获取到下载链接，方便进行权限管理。

#### 2.3.1 普通文件下载

普通文件下载实现分目录和文件下载：

1. 文件下载

文件下载比较简单，实际上就是将文件以数据流的方式传回前台。

1. 目录下载

目录下载需要先将目录打包成zip文件放到/tmp/UserID/目录下，再以数据流的方式传回前台。

后期也可以考虑以下载文件历史版本的方式实现，也就是说下载文件的最新版本即可。

#### 2.3.2 虚拟文件下载

单个虚拟文件下载，跟普通文件系统的目录下载功能相同。但如果需要考虑以目录关系下载的方式的话，则需要将多个虚拟文件目录打包后下载。

另外考虑到以目录结构的方式展示给用户的话，还需要将md文件的内容以html文件的方式提供给用户，这个功能对于利用虚拟文件系统来记录一些接口定义是会非常方便的。

### 文件编辑

文件系统包括虚拟文件和普通文件，虚拟文件是基于Markdown文件实现的，需要保证文件内容指定的外联资源的正常显示，因此实际上并不是一个单一的文件，它在后台存储上是一个以DocID标示的目录，其中包括markdown文件以及关联的资源文件/DocID/DocName.md /DocID/res/xxx.png…，其中md文件内容还有一份是放在DataBase中的。

普通文件就是一个单一的文件：例如txt、word、markdown等文件，普通文件是不保证像markdown文件这种需要链接外部资源的正常显示的。

#### 2.4.1 虚拟文件编辑

当用户编辑虚文件时，需要将虚拟文件目录复制到tmp/UserID目录下，并将markdown文件内容回传给前台，前台根据当前是编辑状态和用户ID，来获取后台相应的图片文件用于展示。

编辑过程中，如果加入了图片，则图片会被上传到tmp/UserID/DocID/res目录，用于编辑过程中的预览。

编辑过程中，前台会定时将修改内容保存到md文件。

用户点击保存后，后台需要将tmp/UserID/DocID目录替换/vdata/DocID目录。

用户点击退出编辑时，如果文件有修改，则提示用户是否先保存修改，如果否的话，则直接进入预览状态，否则后台保存后，再进入预览状态。

预览实际上就是展示/vdata/DocID的md文件。

#### 2.4.2 普通文件编辑

当用户编辑普通文件时，需要将文件复制到/tmp/UserID目录，并根据不同的文件类型将文件内容解析成前台可以展示的数据，前台需要根据不同文件类型选择不同的编辑容器。

用户插入的图片会被保存在/tmp/UserID/res目录下，如果Word之类的文档，则保存时图片会以数据的格式进入文件，但如果类似与md文件本身不保存res文件的话，那么这些图片只是用于临时的展示用途。

编辑过程中，前台会定时将修改内容保存到文件。

用户点击保存后，后台需要将tmp/UserID/目录下的文件替换/rdata/目录下的文件。同时将/tmp/UserID/res目录下的文件复制到/web/images/fordoc

用户点击退出编辑时，如果文件有修改，则提示用户是否先保存修改，如果否的话，则直接进入预览状态，否则后台保存后，再进入预览状态。

预览实际上就是展示/rdata/文件，如果文件使用了图片，则尝试在/web/images/fordoc目录下去找。

#### 1、文件撤销功能

对于本地文件系统而言，由于操作系统和编辑软件的支持，通常不会发生编辑过程中，文件内容彻底丢失的问题，但对于网页来说，彻底丢失是很正常的事情，因此如何保证在网页编辑过程可以跟在本地操作一样安全是一个必须考虑的设计。

用户启动编辑时，后台将原始文件复制到临时目录下，用户点击保存时才会用临时文件替换原始文件，用户可以执行撤销（CTRL + Z）时，系统会将指令传递给后台，由后台来对临时文件的内容编辑进行撤销操作（这个能否实现未验证）；另外一个方案就是保留最近50次操作的文件备份，但这样对于后台的存储文件有着非常高的要求。

仓库目录下共三个目录: data、vdata和tmp目录，其中data用于存放实际的文件，vdata用于存放虚文件（全部是markdown文件： ID+文件名），tmp目录是用于实行文件编辑过程中的撤销功能存在，用于保存用户编辑过程中的内容（用户编辑过程中没隔一段时间保存一个备份，从而实现在线编辑的撤销功能，撤销功能只是一个提高用户体验的一个东西，系统不能保证撤销总是会成功）。

#### 2、文件并发访问

编辑状态的文件及其父节点都不能被删除，这个状态是否记录在数据库中呢？？记录在数据库中的方法是最简单的。设计时需要考虑以下几点：

1. 文件处于编辑状态下，其他人不能编辑；
2. 编辑的用户下线后，如果忘记退出编辑，48小时候自动释放；VSS配置库会长期占用，这是不合理的。
3. 处于编辑状态中的文件，用户重新登陆可以看到该文件仍然处于编辑状态，点击该文件将直接进入编辑状态，且文件内容是上次编辑是的内容。

参考SVN和GIT的实现，将并非访问的冲突处理放在了底层，即编辑过程独立，每个用户都有自己的编辑空间，只有在替换原始文件以及在提交版本的时候进行冲突检测，这样即使发生了冲突每个用户编辑后的数据还是存在，可以再次尝试提交；而且利用了文件系统和SVN以及GIT本身的冲突处理功能，避免了在上层进行复杂的冲突检测处理，而且用户也会有更好的体验。

### 文件版本管理

文件版本管理是一个非常重要的功能，用户主动提交（或者叫保存，可以参考VSS）才会产生历史版本，历史版本的保存使用SVN仓库或GIT仓库进行管理。

文件版本管理相关有以下几个操作节点：

#### 2.5.1新建仓库

仓库存储目录创建成功时，如果该仓库指定了版本控制，则根据设置的SVN或GIT路径进行版本仓库初始化（SVN或GIT路径未设置，需要先创建一个本地的SVN仓库或GIT 仓库，作为版本仓库使用），版本仓库初始化具体操作如下：

（1）远程目录结构修改

如果版本仓库目录结构与本地目录结构不一致的时候，需要先删除版本仓库上存在而本地仓库不存在的文件或目录。

（2）CheckOut

将版本仓库Checkout到本地目录，形成working copy，注意CheckOut时不得对本地文件进行修改。

（3）提交Working Copy的Changes

如果本地目录的文件与版本仓库不一致，则需要将这些Changes全部提交到版本仓库中去，完成同步。

过程中发生错误，新建仓库都会失败，避免出现初始状态不正确的仓库。

#### 2.5.2仓库参数修改

仓库参数如果修改了仓库存储地址和版本管理仓库地址，需要进行处理。

1. 仓库存储地址改变时，需要将当前仓库目录整个移动到新的位置（包括了仓库信息），如果版本管理仓库地址未修改则可以考虑不重新进行版本仓库初始化，（理论上是可以的，当前安全起见还是执行一次）。

如果发生错误，需要删除刚复制过来的目录，以还原现场。

1. 版本管理仓库地址改变时，需要重新进行仓库初始化。

如果发生错误的话，有可能出现不可逆的问题，安全起见就是将WorkingCopy先保护起来，因此在真正完成仓库初始化之前，需要备份本地仓库目录，如果失败的话需要还原回来（我们需要假设总是可以还原的，否则这将是一个永远无法结束的循环）。

#### 2.5.3 新增文件

本地文件已存在 🡪 失败

版本仓库文件已存在🡪失败

提交版本成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次失败🡪失败

失败处理：还原现场（删除新增的本地文件）

#### 2.5.4 删除文件

本地文件不存在 🡪 失败

版本仓库文件不存在🡪失败

提交版本成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次失败🡪失败

失败处理：还原现场（恢复被删除的本地文件）

#### 2.5.5 编辑文件

本地文件不存在 🡪 失败

本地文件替换失败 🡪 失败

版本仓库文件不存在🡪失败

提交版本成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次失败🡪失败

失败处理：还原现场（恢复被修改的本地文件）

#### 2.5.6 上传文件

上传文件如果是新增操作，流程同新增文件。

上传文件如果是覆盖操作，流程同修改文件。

#### 2.5.7 重命名文件

源本地文件不存在 🡪 失败

目标本地文件已存在🡪 失败

版本仓库源文件不存在🡪失败

版本仓库目标文件已存在🡪失败

提交版本成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次失败🡪失败

失败处理：还原现场（将目标本地文件重命名回源本地文件）

#### 2.5.8 移动文件

源本地文件不存在 🡪 失败

目标本地文件已存在🡪 失败

版本仓库源文件不存在🡪失败

版本仓库目标文件已存在🡪失败

提交版本成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次失败🡪失败

失败处理：还原现场（将目标本地文件复制回源本地文件）

#### 2.5.9 复制文件

源本地文件不存在 🡪 失败

目标本地文件已存在🡪 失败

版本仓库源文件不存在🡪失败

版本仓库目标文件已存在🡪失败

提交版本成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次成功🡪成功

提交版本失败🡪更新WorkingCopy后再提交一次失败🡪失败

失败处理：还原现场（将目标本地文件删除）

### 权限控制

系统需要对用户仓库和仓库目录的访问进行权限控制：

1. 仓库
   1. 访问权限包括： 管理权限、读权限、写权限、增加权限、删除权限
   2. 仓库拥有人

拥有仓库的最大权限，包括所有访问权限，不受权限表控制，可以删除仓库本身，可以设置仓库密码

* 1. 仓库管理员

仓库管理员不能设置仓库拥有人、超级管理员、其他管理员的权限外，可以设置所有人员的权限，但可以查看除超级管理员之外的用户权限。

1. 仓库文件节点访问
   1. 文件节点访问权限包括：管理权限、读权限、写权限、增加权限、删除权限
   2. 仓库拥有人

拥有仓库中所有文件的控制权限和访问权限；

* 1. 文件管理员

文件管理员可以管理除仓库拥有人、超级管理员和管理员之外的所有其他该文件下的所有人的权限，但可以查看除超级管理员之外的用户权限。

1. 权限申请
   1. 用户可以申请自己看得到得仓库的访问权限，仓库的管理员会受到申请通知。

### 管理后台

1. 系统管理（超级管理员）

用户自主注册功能关闭与打开

用户自主新建仓库功能的打开与关闭

1. 角色管理（超级管理员）

角色增加、删除、修改

1. 用户管理（超级管理员）

用户增加、删除、修改

（3）仓库管理（管理员、超级管理员）

仓库增加、删除、修改权限管理

仓库目录权限管理

### 用户

系统的用户类型有三种0、1、2，分别是普通用户、管理员、超级管理员：

1. 超级管理员

拥有最大权限，系统的所有功能访问都不受限制；

为了保证系统的安全，超级管理员目前只能设置一个，用户管理里不能添加超级管理员。

1. 管理员

管理员可以访问管理后台，但可访问的管理后台的模块受角色的权限表控制，因此避免管理员通过修改角色定义和用户角色来修改自己的权限，角色管理和用户管理只有超级管理员才拥有权限。

1. 普通用户

普通用户不能访问管理后台，其他功能跟管理员相同。

#### 2.8.1 用户登录

用户登录包括系统登录和管理后台登录，后台需要使用两个不同接口，但用户基本信息检查使用相同的接口，管理后台登录除了检查基本信息外，还需要检查用户角色是否是管理员，只有是管理员才能登录管理后台。

登录成功后，用户信息需要放入session中，用于后续用户信息展示和系统访问控制。

用户登录时，勾选了记住用户，则登录信息会保存进cookie里，方便用户下次自动登录，有效期为1周。

#### 2.8.2 用户注册

用户可以通过邮箱和手机进行自主注册，完成注册需要输入验证码。

### 关键技术

1. zTree控件：

目录结构展示和文件操作

1. MD文件编辑控件

虚文件编辑功能

1. 右键菜单控件

文件右键操作

1. 文件流上传

文件大小可以脱离Tomcat的限制

1. 文件流下载

实现文件下载的权限控制

1. 文件拖拽控件

实现文件拖拽上传

1. 页面Div区域大小拖放控制

目录展示区、当前目录展示区、虚文件预览区、实文件预览区、在线编辑区、在线编辑预览区

1. SVN版本管理
2. GIT版本管理
3. Word在线预览与编辑
4. Pdf在线预览
5. Txt文件在线预览

## 详细设计

### 3.1 数据库设计

为了数据库能够兼容Oracle数据库，因此所有字段全部设计成大写字母。

#### 3.1.1仓库表

|  |  |  |
| --- | --- | --- |
| 字段 | 描述 | 备注 |
| ID | 仓库ID，每个仓库的唯一标示，不可重复 |  |
| NAME | 仓库名字 |  |
| TYPE | 仓库存储类型：  0: 虚拟文件系统  1: 实文件系统 |  |
| PATH | 远程仓库根目录 |  |
| VER\_CTRL | 版本控制类型  0：无  1：SVN  2：GIT |  |
| SVN\_PATH | SVN/GIT路径 |  |
| SVN\_USER | SVN/GIT访问帐户 |  |
| SVN\_PWD | SVN/GIT访问账户密码 |  |
| INFO | 仓库简介 |  |
| MENU | 仓库目录结构 |  |
| PWD | 仓库访问密码 |  |
| OWNER | 仓库所有人（默认是创建人） |  |
| CREATE\_TIME | 仓库创建时间 |  |

#### 3.1.2 文件节点表

这里的文件包括文件和目录

|  |  |  |
| --- | --- | --- |
| 字段 | 描述 | 备注 |
| ID | 文件ID,每个文件的唯一ID |  |
| NAME | 文件名称 |  |
| TYPE | 文件类型   1. 目录 2. 文件 |  |
| CONTENT | 虚文件内容 |  |
| PATH | 目录路径：相对于仓库根目录的路径 |  |
| PID | 父目录ID |  |
| VID | 所属仓库ID |  |
| PWD | 文件节点密码 |  |

常见错误:

1、数据库新增了三个字段，导致addDoc失败，提示没有Default值，将空打勾就可以了

#### 3.1.3用户表

|  |  |  |
| --- | --- | --- |
| 字段 | 描述 | 备注 |
| ID | 用户ID |  |
| NAME | 用户名（字母+数字） |  |
| PWD | 用户密码 |  |
| TYPE | 用户类型：0:普通用户; 1:管理员；2超级管理员 |  |
| ROLE | 角色ID：不同角色可访问的管理后台的模块不同，可自定义 |  |
| REAL\_NAME | 真实姓名 |  |
| NICK\_NAME | 昵称 |  |
| INTRO | 用户简介 |  |
| IMG | 用户头像信息 |  |
| TEL | 手机 |  |
| TEL\_VALID | 手机是否已验证 |  |
| EMAIL | 邮箱 |  |
| EMAIL\_VALID | 邮箱是否已验证 |  |
| LAST\_LOGIN\_TIME | 最近登陆时间 |  |
| LAST\_LGOIN\_IP | 最近登陆IP |  |
| LAST\_LOGIN\_CITY | 最近登陆城市 |  |
| CREATE\_TYPE | 用户创建类型：  0：管理员创建  1：用户注册 |  |
| CREATE\_TIME | 用户创建时间 |  |

#### 3.1.4角色表

|  |  |  |
| --- | --- | --- |
| 字段 | 描述 | 备注 |
| ID | ID |  |
| ROLE\_ID | 角色ID |  |
| NAME | 角色名 |  |
| AUTH | 角色的访问权限，用json格式数据来保存，方便扩展 |  |

#### 3.1.5 仓库权限表

|  |  |  |
| --- | --- | --- |
| 字段 | 描述 | 备注 |
| ID | 每个权限的唯一ID |  |
| USER\_ID | 用户ID |  |
| REPOS\_ID | 仓库ID |  |
| IS\_ADMIN | 是否是仓库管理员：仓库管理员可以管理其他人的权限 |  |
| ACCESS | 访问权限：  0：不可访问  1：可访问 |  |
| EDIT\_EN | 编辑权限：  0：不可编辑  1：可编辑 |  |
| ADD\_EN | 文件新增权限：  0：无  1：有 |  |
| DELETE\_EN | 文件删除权限  0：无  1：有 |  |

#### 3.1.6 文件节点权限表

|  |  |  |
| --- | --- | --- |
| 字段 | 描述 | 备注 |
| ID | 每个权限的唯一ID |  |
| USER\_ID | 用户ID |  |
| DOC\_ID | 文件节点ID |  |
| REPOS\_ID | 仓库ID |  |
| IS\_ADMIN | 是否是文件节点管理员，该属性继承父节点的属性，如果父节点在根目录下，则继承仓库的属性 |  |
| ACCESS | 访问权限：  0：不可见  1：可见 |  |
| EDIT\_EN | 编辑权限：  0：不可编辑  1：可编辑 |  |
| ADD\_EN | 文件新增权限：  0：无  1：有 |  |
| DELETE\_EN | 文件删除权限  0：无  1：有 |  |

#### 3.1.7 系统配置表

|  |  |  |
| --- | --- | --- |
| 字段 | 描述 | 备注 |
| ID | 配置表ID |  |
| REG\_ENABLE | 用户自主注册功能  0：关闭  1：打开 |  |
| PRIVATE\_REPOS\_ENABLE | 普通用户私有仓库创建功能  0：关闭  1：打开 |  |

### 3.2 系统详细设计

#### 3.2.1 用户注册

（1）用户在主页点击注册，弹出注册窗口

（2）用户填写邮箱或手机，点击获取验证码

（3）后台检查邮箱或手机是否已注册，发送验证码到邮箱或手机

（5）验证码输入后，点击注册，后台新增一个新用户

（6）系统自动弹出登陆页面，登陆用户为新注册的用户

#### 3.2.2 用户登录

##### 3.2.2.1 系统登录

1、前台功能

（1）系统主页用户点击登陆弹出登陆窗口

（2）有登陆历史则使用上次的登陆用户，否则用户填写邮箱或手机

（3）输入密码

（4）点击登陆，后台验证成功后，进入系统仓库列表页

2、后台接口

接口：login.do

参数：userName 登录用户名，邮箱或手机

pwd 登录密码

rememberMe 是否记住登录信息

接口功能：

根据用户名和密码信息，检查该用户是否存在，如果信息符合则返回登录成功信息（当前登录用户信息），当前登录用户信息保存在session中用于权限控制。

##### 3.2.2.1 管理后台登录

#### 3.2.2 忘记密码

（1）用户在主页点击忘记密码，弹出找回密码窗口

（2）用户填写邮箱或手机，点击获取验证码

（3）后台检查邮箱或手机是否已注册，发送验证码到邮箱或手机

（5）验证码输入后，点击继续，则弹出密码修改页面

（6）密码修改完成后，自动进入用户登陆页面

#### 3.2.3 用户主页

1、用户登陆成功后，系统右侧显示用户头像

2、点击后有下拉菜单，点击可以进入用户主页的对应页面

3、用户主页包括：用户信息、我的好友、权限管理、密码修改、意见反馈

（1）用户信息：修改用户信息

（2）我的好友：管理好友，添加或删除好友？？

（3）权限管理：点击进入仓库列表页面，可以删除或增加仓库，可以设置仓库的访问权限，可点击管理仓库目录权限按键（拥有该仓库管理权限的人员才能操作，后台控制），进入仓库目录权限管理页面（该页面列出整个仓库的目录结构，可以对每个目录和文件进行权限设置）

#### 3.2.4 仓库权限管理

1、新建仓库时

（1）将创建人加入到仓库权限列表中，权限2(该仓库的管理员)，可以删除仓库和更改仓库访问权限和仓库目录的访问权限；

（2）将创建人加入到仓库根目录的权限列表中，拥有所有权限；

2、可访问仓库列表，通过合并查询仓库列表和仓库权限列表，进行控制。

#### 3.2.5 仓库目录权限管理

1、可访问仓库目录列表，通过合并查询仓库目录列表和仓库目录权限列表进行控制，

2、仓库的访问权限在后台进行控制，前台只做展示和触发后台文件操作。

3、如果拥有文件密码设置权限的人，能够设置文件的访问密码

#### 3.2.4 管理后台

管理后台部分的代码的网址配置在config.js中设置，需要修改的话请在config.js中进行设置。

3.2.4.1 管理后台登陆页面

|  |
| --- |
| [Logo] DocSys |
| 用户名 [ ]  密 码 [ ]  [ ]记住我 忘记密码  [ 登陆按键] |

3.2.4.2 管理后台主页面

|  |  |
| --- | --- |
| [Logo] DocSys [头像] 名字[角色] | |
| 系统管理  角色管理  系统参数  用户管理  仓库管理 | 数据列表区 [增加]  数据1 [删除] [修改] [设置] |

1、点击左侧的功能按键，右侧显示不同的管理页面；

2、右侧管理页面均以列表方式展示需要管理的内容

3、角色管理

角色管理包括角色的增加、删除、修改

角色信息包括角色ID（由系统自动生成无法修改）、角色名字、角色权限

角色权限用于设置角色能够进入的管理后台的哪些模块

1. 系统参数

系统参数管理包括参数的增加、删除、修改、启用

系统参数包括：是否开始开启自主注册、是否开启私人仓库、是否开启私人仓库的删除功能、私人仓库的个数等参数

可以增加多组系统参数，但只能使用其中一组，默认为ID为1的系统参数

1. 用户管理

用户管理包括用户的增加、删除、修改

用户信息包括用户名、邮箱、手机等信息

1. 仓库管理

仓库管理包括仓库的增加、删除、修改、权限管理

仓库信息包括：仓库名、仓库路径、仓库描述、仓库所有人、仓库密码

权限管理：打开新的页面，展示出仓库的数状结构（需要包括仓库本身，仓库本身作为id为0的节点存在），这个页面对于普通用户也可以使用

（1）角色管理

（2）用户管理

（3）权限管理

仓库权限管理

仓库1 [ ] 可见 [ ]不可见

目录权限管理

目录树 [ ] 不可见 [ ]可见 [ ] 修改 [ ]增加 [ ]删除