# 系统概述

本次设计为简版的cms

Admin用户：系统管理员；

支持各类角色维护各类信息；

前台：首页，列表页，详细页；

后台：admin，各个角色；

角色的差别在于两点：

1. 维护信息的类型的不同
2. 功能模块的不同；

# 控制器设计:登录管理

app/controllers/users目录下，这是继承自devise类的代码；

登录 sessions

修改密码 password

注册 registration

解锁 unlocks

确认 confirmation

# 控制器设计:Admin后台

admin的代码放在 app/controllers/admin namespace中。

## 用户管理

app/controllers/admin/users\_controller.rb

列表

增加

删除

修改

## 角色管理

app/controllers/admin/role\_controller.rb

列表

增加

删除

修改

## 信息类型管理

app/controllers/admin/info\_types\_controller.rb

列表

增加

删除

修改

# 控制器设计之角色后台

各个角色功能所在目录

app/controllers/user

info\_controller.rb 完成信息管理方面的功能；

## 信息管理

app/controllers/user/info\_controller.rb 完成信息管理方面的功能；

列表

增加

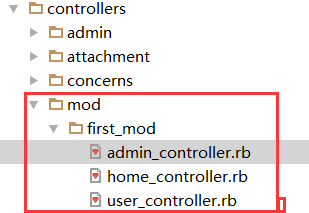
删除

修改

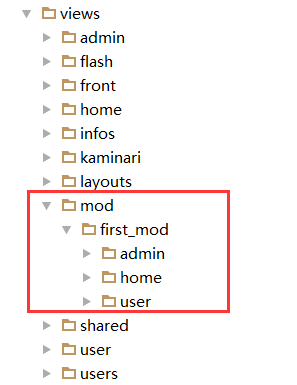
## 功能模块

功能模块均按照插件的方式设计

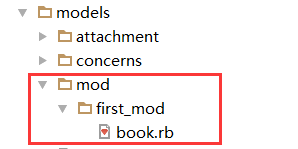
控制器controller下建立mod namespace，表明是mod(module), 然后再建立一个实际的mod，这里是FirstMod,里面建立三个控制器admin：是系统后台需要使用的；user，是用户后台，home则是未授权的功能。



==========================================



模型：建立Mod namespace，其下建立与功能插件相关的模型。



# 控制器设计之Guest页面或首页

app/controllers/home\_controller.rb

首页

列表页

详细页

# 模型设计

# 服务设计

# 功能设计

app/controllers/function目录下

如调研等功能。