# 网站设计运行说明书

设计理念

本网站是用于展示各大影院所拥有的影片的信息检索系统，页面主要以灰色调风格为主，站点分有登录页、首页展示、影院信息、高级搜索查询页等各个方面子页面。

本站主要采用的技术是MongoDB+AngularJS+Nodejs+Bootstrap。

设计流程

首先搭建后端功能，通过使用Resteasy网站完成了post、delete等http请求的测试，使代码具备了查询电影的初步功能。之后通过一系列编译、测试，陆续完成了用户登录，高级搜索查询、获取更多信息等后端功能。

其次构思前端展示页面，在查找资料过程中了解到Bootstrap框架，经过学习，本人下载并使用了其提供的部分组件于前端用户交互页面。

在使用Bootstrap完成了前端展示页面的制作后，本人使用AngularJS完成了前后端交互工作。

使用的工具与技术

网页制作工具：VS CODE

应用到的设计技术：MongoDB+AngularJS+Nodejs+Bootstrap

网站设计说明

网站名称：基于MongoDB的电影信息检索系统

在一切开始之前请先查看路由分配

## 登录功能

登录是进入展示页前所必要进行的操作，系统通过区分登录用户是否为管理员来展示不同的界面

* 若是普通用户，则展示所有电影信息，提供高级搜索查询与影院信息查看。
* 若是管理用户，则展示其旗下所拥有的院线电影，并提供添加、修改、删除电影等需具有管理权限的操作。严格进行了管理权限的区分。

打开根目录下的app.js文件以及models文件夹，并打开models文件夹下的User.js

后端采用Mongoose链接Mongodb数据库，首先使用Mongoose.Schema创建Model，Model代表的是数据库中的集合，通过Model才能对数据库进行操作，下图展示用户的MongooseSchma创建，分有三个属性：name是用户名，password是登录密码，admin是管理员属性。

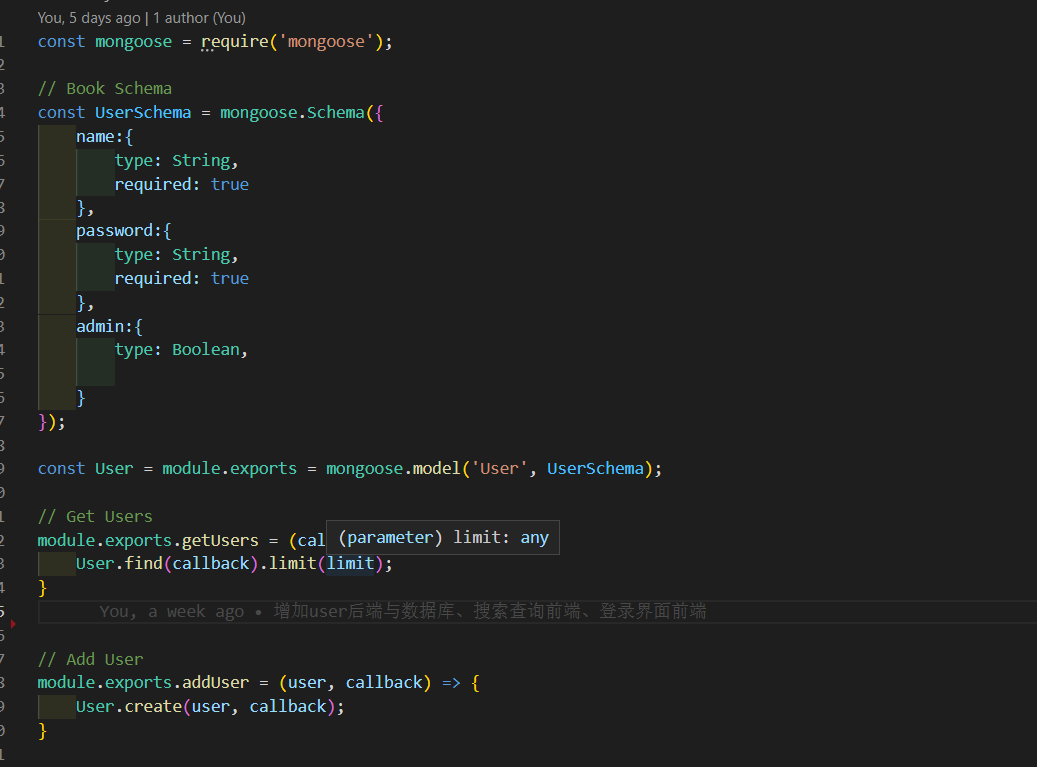


图 1 ./models/User.js文件截图

由于登录与注册行为涉及到用户隐私（密码），故采用post的http请求方法来实现表单的传参。下图展示的是app回应注册请求以及登录请求，值得注意的是：model.find方法返回的是一个数组，不能只是简单通过if(user)判断是否找到用户，否则会一直返回真值，不能判断是否查询到指定用户。



图 2 ./app.js文件关于检测用户登录的截图

下图展示登录界面以及注册界面的前端



图 3登录界面展示



图 4登录界面前端实现



图 5 注册前端展示



图 6 注册界面前端实现

点击高级搜索，可按用户给出的条件进行复杂条件搜索。若无输入条件，则默认输出所有电影。系统也将自动区分某一条件是否有输入。若无输入，则不将该条件加入搜索语句中。