**电 子 科 技 大 学**

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

**学 士 学 位 论 文**

BACHELOR DISSERTATION

**论文题目 基于Web技术的电影交流分享平台的设计与实现**

**学生姓名 李卓**

**学　　号 2012222010007**

**学　　院 信息与软件工程学院**

**专　　业 软件技术**

**指导教师 李美蓉**

**指导单位 成都了了科技有限公司**

**年 　　 月 　　 日**

# 基于Web技术的电影交流分享平台的设计

# 摘要

2015年，中国电影发展取得突破性成果，全国电影总票房达440.69亿元，同比增长48.7%，其中，国产电影票房达271.36亿元，占票房总额的61.58%。看电影成为一种生活方式，社交活动和沟通方式。

本程序设计并实现了一个电影分享交流平台，尽可能的将电影宣传信息第一时间发布，让浏览者更加清晰和详细的了解电影详情，也尽可能详细的展示、介绍不同类型的电影，同时为爱电影的人提供一个交流的平台。在“IMovie”的设计过程中使用了HTML、CSS、JavaScript等开发工具。通过对“IMovie”桌面应用的需求分析、总体设计、详细设计和测试，最终完成了首屏、各分类页、详情展示页、注册页、登录页等，完成了“IMovie”的课题设计与实现。

关键词：电影，桌面应用，Javacript, Web技术

# The Design and Implementation of Film Sharing Platform Based on Web Technology

# Abstract

In 2015, the Chinese Films have a breakthrough achievements. The national film box office grossed up to 44.069 billion yuan, an increase of 48.7%. Among those, the domestic box office reached 27 billion 136 million yuan, accounting for 61.58% of total box office. Watching movies become a way of life, social activities and communication.

This program has designed and implemented a movie sharing platform, release the film publicity information for the first time as far as possible, let visitors more clear and detailed understanding of the film details. As much as possible detailed presentation of different types of movies, at the same time to provide a communication platform for the people who love movies. In the design of "IMovie" is used in the HTML, CSS, JavaScript and other development tools. Design and implementation of topic "IMovie". through the "lMovie" application needs analysis, overall design, detailed design and testing, the final completion of the first screen, the classification page, details page, display the registration page, login page, complete the design and implementation of IMovie project ".

Keywords：Movies,  [Desktop Applications](http://www.baidu.com/link?url=UXKzhMGhFUCO-yLp9EzJPkfyf9OEg2VYENFDQp1aqNRtvbwAIiyglWnh7tkbUE_g5DDwilosO407HJN0cFk_mGEL9eTY36XH8QSNM3uwi1jIjvT9c3V1_JFoh9BeS0Up" \t "https://www.baidu.com/_blank), Javascript, Web Technology

目 录

[第一章 毕业设计（顶岗实习）概况 1](#_Toc451259446)

[1.1 实习单位与实习岗位概况，对企业所属行业的认识 1](#_Toc451259447)

[1.2 实习项目课题背景、价值、意义以及国内外研究现状 1](#_Toc451259448)

[1.3 实习项目整体执行完成情况概述 1](#_Toc451259449)

[第二章 需求分析 2](#_Toc451259450)

[2.1 功能需求 2](#_Toc451259451)

[2.1.1 ×××××× 2](#_Toc451259452)

[2.1.2 ×××××× 2](#_Toc451259453)

[2.2 ×××××× 3](#_Toc451259454)

[2.2.1 ×××××× 3](#_Toc451259455)

[2.2.2 ×××××× 3](#_Toc451259456)

[2.2.3 ×××××× 4](#_Toc451259457)

[2.2.4 ×××××× 4](#_Toc451259458)

[2.3 ×××××× 4](#_Toc451259459)

[2.3.1 ×××××× 4](#_Toc451259460)

[2.3.2 ×××××× 4](#_Toc451259461)

[2.3.3 ×××××× 4](#_Toc451259462)

[2.4 本章小结 4](#_Toc451259463)

[第三章 ×××××× 5](#_Toc451259464)

[3.1 ×××××× 5](#_Toc451259465)

[3.1.1 ×××××× 5](#_Toc451259466)

[3.1.2 ×××××× 5](#_Toc451259467)

[3.1.3 ×××××× 5](#_Toc451259468)

[3.1.4 ×××××× 6](#_Toc451259469)

[3.2 ×××××× 6](#_Toc451259470)

[3.2.1 ×××××× 6](#_Toc451259471)

[3.2.2 ×××××× 7](#_Toc451259472)

[3.2.3 ×××××× 7](#_Toc451259473)

[3.2.4 ×××××× 8](#_Toc451259474)

[3.3 本章小结 8](#_Toc451259475)

[第四章 ×××××× 9](#_Toc451259476)

[第五章 结束语 10](#_Toc451259477)

[5.1 本文内容 10](#_Toc451259478)

[5.2 顶岗实习项目课题有待进一步解决的问题及方向 10](#_Toc451259479)

[5.3 对软件工程实践以及软件工程领域发展的认识 10](#_Toc451259480)

[5.4 本人毕业设计（顶岗实习）收获及体会 10](#_Toc451259481)

[参考文献 11](#_Toc451259482)

[致谢 13](#_Toc451259483)

[外文资料原文 14](#_Toc451259484)

[外文资料译文 15](#_Toc451259485)

第一章 毕业设计（顶岗实习）概况

## 1.1 实习单位与实习岗位概况，对企业所属行业的认识

成都了了科技有限公司是一家小型的创业公司，主要产品是Skylark，Skylark致力于帮助组织优雅地管理组织消息。Skylark可以便捷组织架构管理、灵活消息和表单管理、强大的三级推送、让消息零遗漏，提高工作效率。

我作为一名前端开发工程师实习生，主要的岗位职责：

1. 负责HTML/JavaScript/CSS代码的编写，掌握HTML5特性,实现web端的界面效果、交互和功能；

2. 配合产品完成Web项目迭代，不断优化和提升用户体验；

3. 深入分析和解决前端遇到的各种技术、性能、跨终端兼容问题；

4. 完成前端模块业务代码优化及JS组件的封装。

1.2 实习项目课题背景、价值、意义以及国内外研究现状

1.2.1 课题背景

从1945年美国电子数字积分计算机ENIAC诞生以来，短短四、五十年，计算机技术发展异常迅速，计算机尤其是微型计算机逐渐普及，计算机的应用也由原先单纯的数值发展，到数据处理如办公自动化，辅助教学，工业自动控制，人工智能，可以说渗透到生产生活的各个领域，计算机的应用提高了效率，降低了成本，节省了人力物力。但是要发挥其巨大作用，还需提高人们的应用水平。

一百年前，电影在西方被发明，电影是一种[表演艺术](http://zh.wikipedia.org/wiki/%E8%A1%A8%E6%BC%94%E8%97%9D%E8%A1%93" \o "表演艺术)、[视觉艺术](http://zh.wikipedia.org/wiki/%E8%A6%96%E8%A6%BA%E8%97%9D%E8%A1%93" \o "视觉艺术)及[听觉艺术](http://zh.wikipedia.org/wiki/%E5%90%AC%E8%A7%89%E8%89%BA%E6%9C%AF" \o "听觉艺术)，利用[胶卷](http://zh.wikipedia.org/wiki/%E8%86%A0%E5%8D%B7" \o "胶卷)、[录像带](http://zh.wikipedia.org/wiki/%E5%BD%95%E5%83%8F%E5%B8%A6" \o "录像带)或[数字媒体](http://zh.wikipedia.org/wiki/%E6%95%B8%E4%BD%8D%E5%AA%92%E9%AB%94" \o "数字媒体)将影像和声音捕捉起来，再加上后期的编辑工作而成。电影的奥秘它能够借由其影像表现出现实世界之丰富，而其本身任然停留在影像。电影是现在人们休闲的重要方式之一。电影里面往往都是：爱情、友情、生活、生命、现在、未来，不管是真实的、还是虚幻的，几乎每一部电影都能把一些种子植入你的大脑中，指不定哪天就发芽了。真的把一部部电影看进去了，它解决的绝对不是你的孤独、无聊。它会让你对生活有了新的看法，甚至会改变你人生的轨迹。

刚刚过去的2015年，中国电影票房规模达到了440亿美元，面对竞争压力增大和全球电影市场疲软的局面，中国电影却逆势而行。改革开放30多年，中国电影市场从无到有，人们的文化，娱乐与精神需求不断提高，看电影成为一种生活方式，社交活动和沟通方式。人们对观看电影逐渐有着更高的需求。

如今的网络上，有着大量的电影网站，其中每个电影网站的侧重点各有不同，有的侧重热门电影，有的侧重冷门电影，有的侧重社交，有的侧重电影评价，在这种情况下，各类型的电影都被广泛的宣传开来。由此可见，电影平台是电影的一种重要宣传手段，而且电影信息可以实时更新，人们对电影评论也可以更广泛的传播开来，人们对电影的评论可以让其他人更好的了解该电影而产生去看电影的兴趣，所以电影交流平台对推动电影的发展，提高人们生活娱乐水平有着极其重要的意义。

在这种电影业蓬勃发展的趋势下，在电影的宣传和评论都逐渐的由实体宣传和评论转到了网络宣传的趋势下，在人们了解电影信息也开始逐渐的由实体媒体了解转向到了网络上了解最新的电影信息的趋势下，综合以上所考虑的全部研究背景，特此研究出本中小型电影交流分享应用。

1.2.2 国内外研究现状

在网络刚出现的时候，国外就涌现了许多电影有关的网站，有电影公司的，也有电影院进行宣传的。而在网络发展几年之后，又出现了许多电影交流平台，例如著名的IMDB。

这种风潮很快进入中国，并发展开来。从2001年中国电影“院线化”改革开始，中国电影业发展已过十余年，如今互联网对电影行业产生颠覆性改变，国内出现了大量的电影网站平台，如豆瓣网，时光网等。到了现在，电影交流平台已经成为各个电影公司宣传的不二选择。

随着经济的发展，中小型电影交流分享平台是计算机技术与电影宣传相结合的产物，通过电影交流分享平台实现对在线电影宣传的高效管理，随着计算机技术的不断提高，计算机网络已经深入到社会生活的各个角落。电影交流分享平台可以拉近相同兴趣人之间的距离，让用户更直接的了解电影，对自己看过的电影进行评价打分以及收藏。

1.2.3 研究目的和意义

根据研究背景和研究意义所阐述，本课题最终研究的主要内容是开发一款适合大众的中小型电影交流分享平台，能快速的进行对电影信息的发表，好让用户能够在第一时间内了解到最新的电影资讯，对电影进行评分，发表对电影的评论，在网上的电影评论网站逐渐多起来的时代，怎么快速有效并且高时效性的让用户能够评价电影是网上电影评论网站的开发要点。

即使在移动端和云端大行其道而，桌面端日渐落末的现在，我仍觉得桌面应用还是自己的优点。和 Web 应用比起来桌面应用的优点还是很多的：只要把它们放在开始菜单栏或者 dock 上，你就能随时打开它们；还可以通过 alt-tab 或者 cmd-tab 切换应用；和操作系统之间的交互更良好（快捷键，通知栏等）。

本电影交流分享平台利用关联数据库和互联网交互信息功能，不仅仅单方面提供电影信息资讯，而且发布全面的电影，演职人员，上映电影等信息，浏览者可以更全面的浏览电影及相关资讯，还可以对电影进行评论，给电影爱好者们搭建一个全面浏览电影信息的平台。

管理员通过后台系统与数据库交互，可以方便的管理网站和用户。

1.3 实习项目整体执行完成情况概述

在实习期间，在企业导师的指导下，跟着自己所在的项目小组，完成公司派予的任务。同时，我还学习巩固了与毕业设计有关的各项知识，如javascript，mysql数据库，软件工程、软件项目管理的相关知识等。

实习项目整体有以下几个内容：需求分析，系统分析，可行性分析，系统整体设计，代码编写，软件测试。

需求分析主要是分析用户需求，业务需求，功能需求。需求分析阶段十分重要，就因为他具有决策性、方向性、策略性的作用，他在软件开发的过程中具有举足轻重的地位目前此阶段已完成。

系统分析主要是分析系统功能，如用户功能分析，包括用户的登陆，注册，查询，评论等。系统功能，电影的展示，修改，增加等。目前此阶段已完成。

可行性分析是系统分析阶段的重要活动，是对系统进行全面、概要的分析。它的任务是确定项目开发是否必要和可行。目前此阶段已经完成。

系统整体设计主要是系统功能设计，详细设计，系统界面设计等。目前此阶段已经完成。

代码编写则主要书写应用各个模块的逻辑代码。目前此阶段已经完成。

软件测试的目的是为了发现程序中的错误而执行程序的过程。好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案。成功的测试是发现了到今为止尚未发现的错误的测试。目前此阶段已经完成。

1. 复杂工程问题归纳与实施方案可行性研究

2.1 课题目标与任务

本课题是基于Web技术开发出媲美Windows桌面原生程序的应用，用于让电影爱好者发现爱好相同的人，分享交流电影。

程序主要包含以下模块: 用户个人中心，包括个人信息，收藏电影，评价历史。热映电影以及相关影院的信息。单个电影相关页面，包括电影详情，所有用户的评价与打分。排行榜，电影趋势数据图形化展示。

在研发课题程序时，主要任务包含：需求分析，可行性分析，系统设计，系统详细设计与实现，代码的编写，软件测试。

需求分析的任务是通过详细调查现实世界要处理的对象，充分了解原系统工作概况，明确用户的各种需求然后在此基础上确定新系统的功能。确定对系统的综合要求，虽然功能需求是对软件系统的一项基本需求，但却并不是唯一的需求，通常对软件系统有下述几方面的综合要求。主要包括功能需求分析，性能需求分析，可靠性和可用性分析。

可行性分析的任务是通过对项目的主要内容和配套条件，从技术、经济、工程等方面进行调查研究和分析比较，并对项目建成以后可能取得的财务、经济效益及社会环境影响进行预测，从而提出该项目是否值得投资和如何进行建设的咨询意见，为[项目决策](http://baike.baidu.com/view/1334011.htm" \t "http://baike.baidu.com/view/_blank)提供依据的一种综合性的[系统分析方法](http://baike.baidu.com/view/532763.htm" \t "http://baike.baidu.com/view/_blank)。可行性分析应具有预见性、公正性、可靠性、科学性的特点。

系统设计是本程序的物理设计阶段。根据[系统分析](http://baike.baidu.com/view/170100.htm" \t "http://baike.baidu.com/view/_blank)阶段所确定的程序的[逻辑模型](http://baike.baidu.com/view/1650561.htm" \t "http://baike.baidu.com/view/_blank)、功能要求，在用户提供的环境条件下，设计出一个能在计算机网络环境上实施的方案，建立程序的物理模型。

系统详细设计是[软件工程](http://baike.baidu.com/view/1659.htm" \t "http://baike.baidu.com/view/_blank)中软件开发的一个步骤，就是对[概要设计](http://baike.baidu.com/view/551728.htm" \t "http://baike.baidu.com/view/_blank)的一个细化，就是详细设计每个模块实现算法，所需的局部结构。在[详细设计](http://baike.baidu.com/view/228950.htm" \t "http://baike.baidu.com/view/_blank)阶段，主要是通过需求分析的结果，设计出满足用户需求的产品。

代码编写的任务是在系统分析之后，通过代码的编写对系统进行实现，最后编译出整个产品。

软件测试的任务是通过编写软件测试代码，检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别，是为了发现程序中的错误而执行程序的过程。

2.2 相关复杂工程问题

在设计本程序时，主要有以下几个问题需要解决：

1. 如何通过当下Web技术开发出媲美Windows桌面原生程序的应用，并具有较高的观赏性和易操作性。Web所具有的开发简单、访问容易等特点，也得到web服务技术不断的发展和应用。借助Web技术可以在较短的时间内发展应用程序，保证了系统的高质量和高可行性。
2. 程序的界面该如何设计。网站的结构布局与视觉感受对一个网站是否被关注起着至关重要的作用。例如一个网站的首页如果结构布局凌乱不堪，视觉感官令人厌烦，那么将会造成浏览者对其关注的吸引力下降。因此需要设计一个符合用户习惯，能够吸引用户眼球的界面。
3. 程序如何才能提供丰富的电影条目数据。电影的信息需要一定的来源，否则无法呈现给用户电影信息。
4. 管理员和普通用户登录因权限不同而路由不同的问题，需要给管理员和普通用户分配不同的权限，并且在不同的路由里设置权限，使得普通用户无法进行管理员才拥有的操作。
5. 电影评分的计算以及通过评分排序来制作排行榜。电影评分的计算不能简单的用总分除以人数来算，如果一个电影A的评分有1000人得到平均10分，另一个电影B评分有10000人达到平均分9.8分，则A与B哪部电影更优秀则需要另说。
6. 对于电影趋势数据的图形化展示。将电影的评分，最近几周票房数据需要通过图表更直观的展示给用户。

2.3 实施方案与可行性研究

在通过对相关资料的搜集和研究，针对相关复杂工程问题设计解决方案，并在此基础上进行可行性分析。最后采用以下解决方案：

1. 目前，使用Web技术开发桌面应用，与传统的Java和.Net应用程序一样，它不被系统原生所支持，运行环境需要相应的技术和环境支撑。而Github项目组所开发出的Electron使得开发者可以通过使用 JavaScript 构建桌面应用。基于 Web技术开发的桌面应用,结合了Web开发模式的简易高效性和桌面模式的功能 丰富性,这种应用的核心技术是 HTML 和 Javascript,其运行期环 境需要具有呈现 Web 页面和执行 Javascript 脚本的能力, 即浏览器渲染引擎的功能。
2. 在当今UI设计世界里，从之前的拟物化，到现在流行的扁平化以及谷歌所提倡的Material Design设计风格大为流行。本应用采用Material Design设计风格，因为它对用户更加友好，它的设计具有普遍适应性。
3. 电影的信息一定要有稳定的来源，所以本程序采用爬虫来进行爬取豆瓣上电影信息，并将其导入数据库，通过使用一台服务器进行定时爬取任务进行爬虫工作。
4. 通过在代码中编写Permission规则，并写成Service，在用户登陆时进行权限的判断，来给与不同用户不同路由权限。
5. 电影评分排名采用贝叶斯统计算法，假定每个用户的评分都是独立事件，每次评分只有n个选项可以选择，那么这就服从["多项分布"](http://en.wikipedia.org/wiki/Multinomial_distribution" \t "http://www.ruanyifeng.com/blog/2012/03/_blank)，就可以结合贝叶斯定理，估计该分布的期望值。
6. 过对数据库中数据的采集与分析，借助百度前端的数据可视化框架echarts将数据统计并展示出来，清晰有效地传达与沟通信息。

2.4 可行性分析

1. 技术可行性：技术上的可行性分析只要是分析技术条件是否能顺利完成开发工作，硬、软件是否满足开发者的需求等，能否在规定时间完成整个项目。

1、细化应用目标。分析具体目标的技术可行性。通过相关电影信息总结，了解本应用的设计目标和具体要求，实现每一个具体目标和功能在技术上的功能，同时还要考虑需要什么条件实现，列出每个具体目标的内容、任务的清单。

2、分析应用的可用性。应用的设计要求使浏览者易于使用，而不止是简单的信息堆砌与排列。这个要求与本应用的版面布局和服务器的功能有关。

3、分析应用的交互性。交互性是现代应用发展的趋势，应用的交互性能可以增加对应用的处理功能和存储容量的需求，内部结构设计也要根据交互性能做相应地调整。

4、分析应用的性能。应用的用户数量随着应用的性能和功能不断变化，在保证本应用性能的前提下，不断满足更多用户提出的需求，并不断规划、设计和系统维护。

应用的综合性能一般通过应用的响应时间和处理时间以及用户平均等待时间和系统输出量来衡量。所以提高应用性能，可以通过调查用户数量和信息处理量确定应用服务器的功能。

（2）经济可行性：软件的经济可行性是指软件所能带来的经济效益与开发设计所需要的投资相比，是否相适宜，同时还要看此软件是否能真正给用户带来足够的经济效益。本电影交流分享平台正是考虑系统开发的成本而设计。本系统的开发在经济上是完全可行的。开发此软件是不需要大量的经费，而且是个人独立设计，可以节省很多费用，同时也可提高个人的实际动手能力。经过详细周密的调研，本系统采用功能强大并且使用民用开发环境，借鉴成功的实例，及科学的开发方法，可以大大的降低本系统的开发成本，达到经济高效的目的。

（3）操作可行性，该应用是基于WEB技术的桌面程序，不需要什么技术就可直接在软件上操作。本应用具备友好的用户界面，方便用户使用。用户只需熟练操作计算机，并对此应用软件的使用做简单的了解，因此从使用操作方面来看，电影桌面应用的开发是可行的。

2.5 功能需求

经过分析，认为本电影交流分享平台主要包含2个角色，用户以及管理员。用户可以在系统中进行账号注册，用户可以对电影进行简短评论、打分。管理员上传电影的画报、各类简介信息。系统根据某种规则自动统计、计算每部电影的分数情况。系统自动生成电影的排行榜。系统可以根据电影的年份或分数情况对电影进行自动排序。

2.5.1 管理员功能需求分析

管理员管理平台功能需求分析：

1. 添加电影信息的功能

管理员登陆后可以对电影交流分享平台的电影信息进行添加操作，从而让用户登陆程序的时候就了解到平台的电影更新信息。

1. 修改电影信息的功能

管理员可以用对电影信息进行修改，包括电影的名称，图片，演职人员等等，以便对过去发布电影信息错误的一些修正。

1. 删除电影信息的功能

管理员可以对电影条目进行删除，以便对一些错误录入的电影信息进行删除。

1. 管理员删除电影评论信息

管理员在管理影评的时候对一些不正当的影评信息进行删除操作。

1. 管理员删除用户信息的功能

管理员在查看已注册用户信息的时候因为某些原因需要删除某些用户的时候，可以对想要删除的用户进行删除。

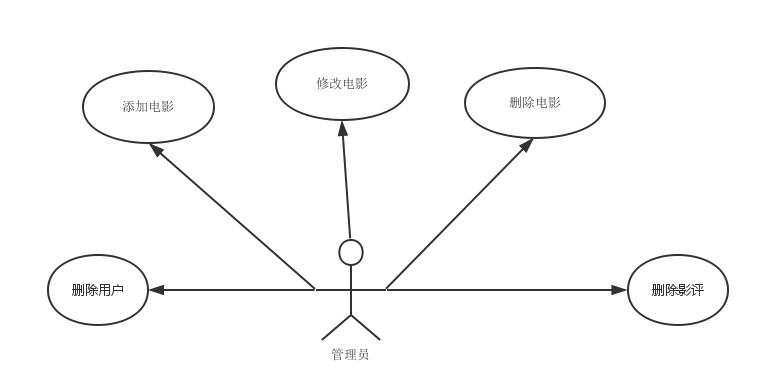


图2-5-1 管理员用例图

2.5.2 用户功能需求分析

用户对电影平台功能需求分析：

1. 用户查看电影信息的功能

用户登入程序之后可以按照需求来查看网站发布的电影信息，从而了解最新的电影信息。

1. 用户搜索电影信息的功能

用户可以对自己需要的电影信息进行查询平台所发布的电影信息。

1. 用户对电影进行评分的

用户对自己已经看过的电影进行评分（满分10分）操作。

1. 用户发布电影评论信息

用户对感兴趣的电影或者已经看过电影需要发布影评信息的时候，可以对想发表的电影进行发布影评的操作。

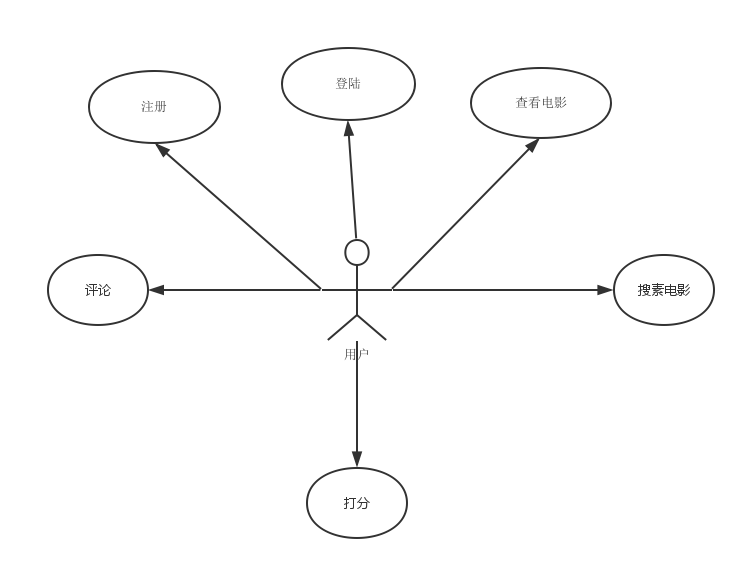


图2-5-2 用户用例图

2.6 界面需求

一个应用的前台界面十分主要，这关乎与用户对程序的第一感受，在很大程度上影响着用户留存率。如果界面不够漂亮，就算功能再全，也很难让用户东西。经过慎重的考虑，本应用使用Google所推出的Material Design设计风格，使得界面美观大方，更能吸引用户，而且各个界面的风格统一，给人一气呵成的感觉。

2.7 性能需求

本平台再满足用户可以对电影进行简短评论、打分。管理员上传电影的画报、各类简介信息等需求的同时，性能上也需要满足：

1. 任何软件都不能排除恶意破坏，盗窃密码等行为，为了防范这种情况，系统需采取措施来保护不法分子的恶意行为。
2. 平台权限管理，需要设置2中权限，即管理员和普通用户，管理员在登陆之后可以对电影信息进行管理，而用户在登陆之后，则不能对电影信息进行增删改等操作，需要对不同的账号设置不同权限。
3. 平台需要提供搜索功能，以便用户能方便快速的查找自己想要查找的电影信息，提高效率。

2.8 系统主要流程图

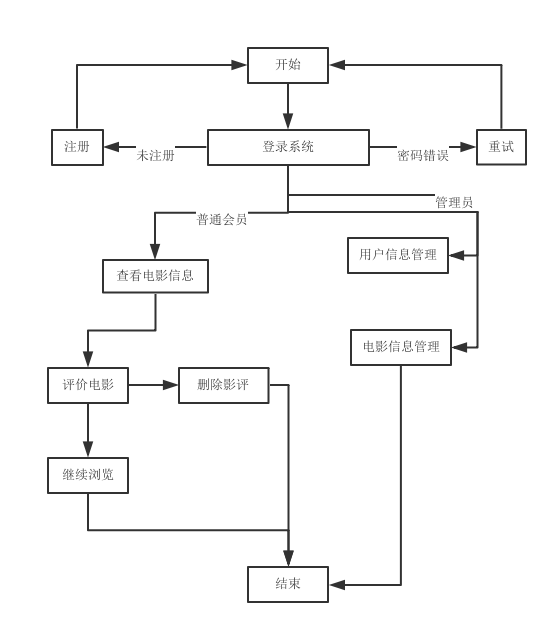


图2-8 系统流程图

2.4 本章小结

本章主要对用户需求的初步分析，说明实习课题目标与任务，并在此基础上归纳说明待解决的相关复杂工程问题。阐述了针对待解决的复杂工程问题所提出的解决方案，并在安全、环境、法律等约束条件下，通过技术经济评价对方案的可行性进行研究。在确认实施方案可行性的基础上，针对用户需求，采用软件工程方法，开展详细需求分析，分别进行了系统功能需求分析，界面需求分析，性能需求分析等。

第三章 针对复杂工程问题的方案设计与实现

3.1 概要设计

3.1.1 系统功能设计

在系统设计的过程中，根据所需要实现的各个功能，将整个程序分为了前台操作的一般用户和后台操作的管理员两个用户。每个用户下分别实现了各个管理模块，管理员用户实现的是电影信息管理模块，影评信息管理模块和用户信息管理模块；一般用户实现的是自身信息操作模块，电影信息操作模块，影评信息操作模块。各个模块设计起来具有相对独立性。可以分别进行构造和实现。为了提高每个模块的高效化，又把每个模块划分为多个字段，这样实现起来更容易而且方便系统的维护，系统的功能设计如下图3-1-1所示。

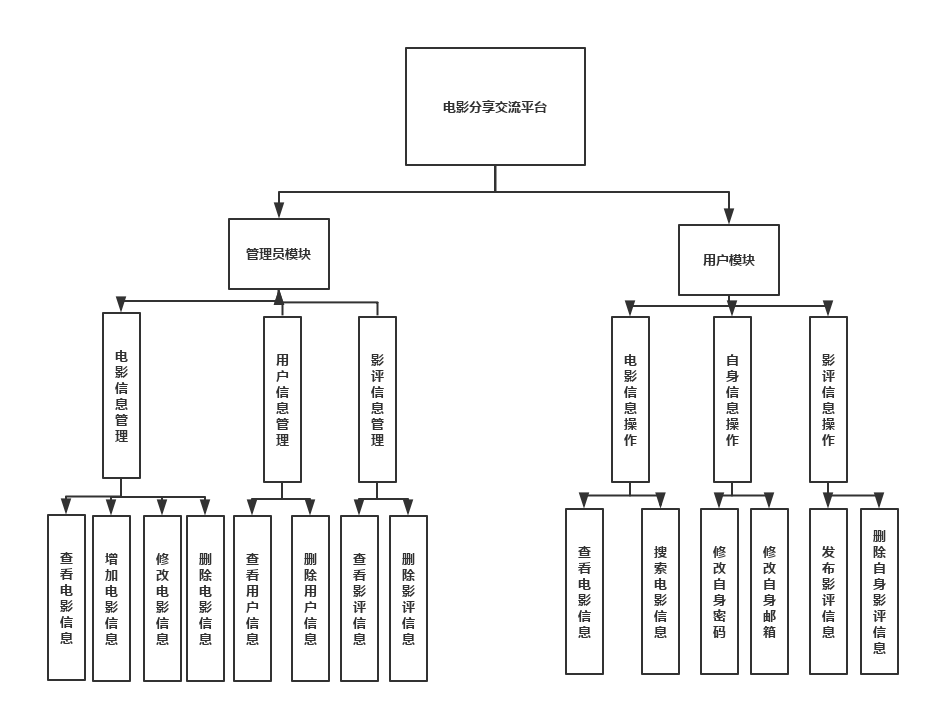


图3-1-1 系统功能模块图

如上图所示，各个模块的细致功能如下：

管理员用户：

1. 电影信息管理模块，主要包含查看电影信息模块，增加电影信息，修改电影信息，和删除单元信息4个子模块。
2. 影评信息管理模块：主要包含查看影评信息和删除影评信息2个子模块。
3. 用户信息管理模块，主要包含查看用户信息和删除用户信息2个子模块。

一般用户：

1. 电影信息操作模块，主要包含查看电影信息和搜索电影信息2个子模块。
2. 自身信息操作模块，主要包含修改密码信息和修改自身邮箱2个子模块。
3. 影评信息操作模块，主要包含发布影评信息和删除自身影评2个子模块。

3.1.2 系统用户用例设计

针本程序的制作目的是为了方便用户想知道最新的电影资讯以及想要查看到别人对电影发布的影评。并且可以实现整个管理员对整体网站的管理。网站用户分为管理员用户和一般用户两个用户，系统用例图如下图3-1-2所示。

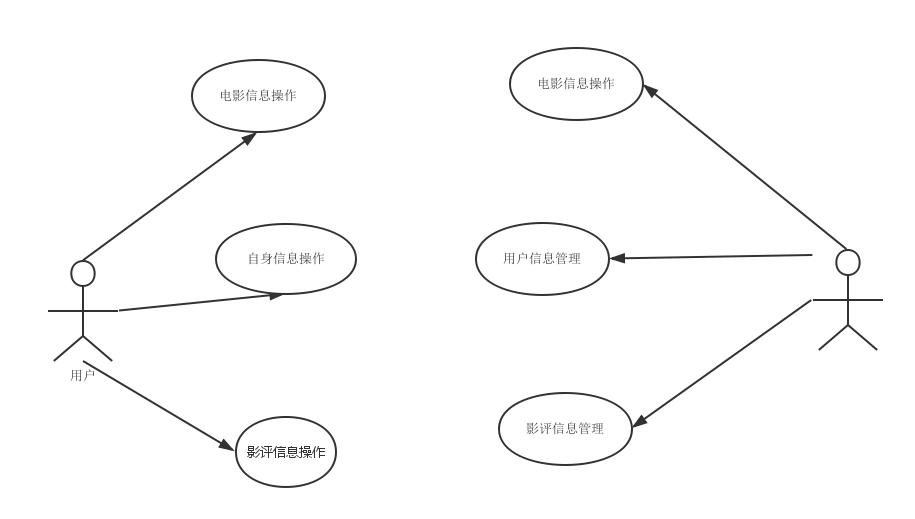


图3-1-2 系统用例图

3.1.3 系统结构设计

针对上面的需求分析，我们把整个系统分成三个模块：1、Model：实现数据抓取，数据处理等功能。2、View：实现网站的界面布局和接收用户响应。3、Controller用来接收view层的响应，转发给model层。

以下就是系统的结构图如图3-1-1所示

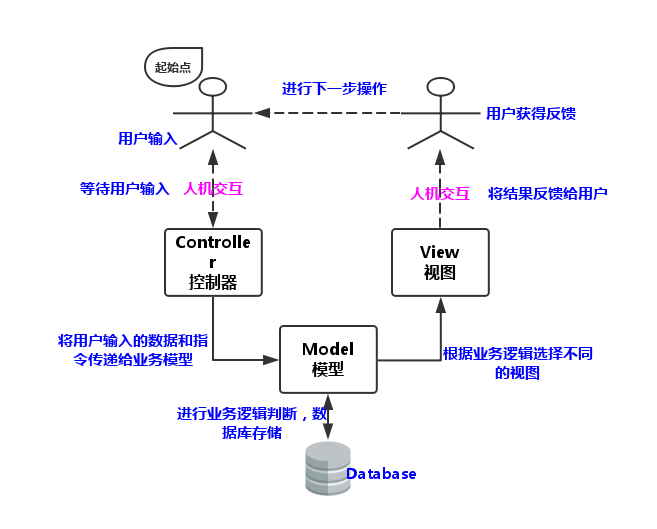


图3-1-3 系统结构图

3.1.4 数据库设计

在设计数据库的时候，应考虑以下事项：

1. 该系统的主要用途，在其应用时要涉及的功能。
2. 数据库规范化规则，防止数据库设计中出现错误。
3. 对数据完整性的包护。
4. 调查用户的实际需要，完整数据库。
5. 数据库和用户权限的安全要求。
6. 选择数据库的类型，在本系统中使用MySql作为该系统的后台数据库。

3.1.4 数据库概述

我们数据库技术是数据管理的最新技术，是计算机科学的一个重要分支，它能指导我们正确地设计数据库系统，它的出现极大地促进了计算机应用的发展。采用数据库技术的原理和方法可以有效地设计实用的数据库系统。一个完整的数据库系统包括数据库管理系统（DBMS），数据库管理员（DBA）、数据库（DB）、应用程序和相应的硬件设施。

目前许多数据库管理系统都基于关系模型，关系模型的主要特点是用表格结构表达实体，用键表示实体与实体之间的联系。与层次模型和网状模型相比，关系模型比较简单，容易为初学者接受。关系模型是由若干个关系模式组成的集合，关系模式相当于记录类型，它的实例称为关系。每个关系是一张表格。表格简单，用户易懂，用户只需用简单的查询语句就可以对数据库进行数据操作，并不涉及到存储结构，访问技术等细节。SQL语言是关系数据库的代表性语言，已经得到广泛应用。

3.1.5 系统数据库E-R图

通过系统的需求分析以及对系统所需要的数据库进行设计，设计出满足系统的数据实体以及相互关系。根据上面的分析，得到的实体有管理员，电影信息，影评信息，用户信息。具体关系E-R图如图3-1-5所示：

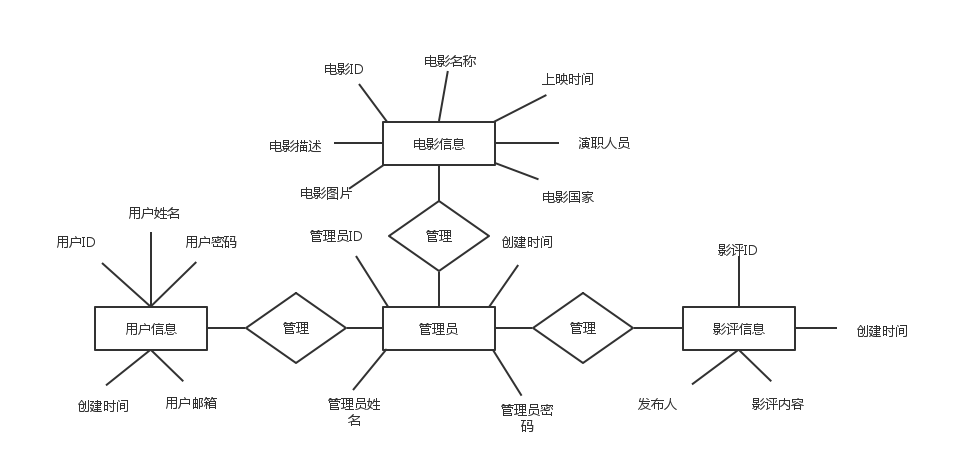


图3-1-5 数据库E-R图

3.1.6 系统数据库表设计

在经过经过数据库的分析以及E-R图的设计，最终设计出本系统用到的数据库表，本系统所用数据库表为：管理员数据表、电影信息数据表、影评信息数据表、用户信息数据表和新闻信息数据表。以上数据表的具体字段信息设置如下:

管理员数据表： 管理员编号，管理员姓名，管理员密码，创建时间。

电影信息数据表： 电影编号，电影名称，电影类型，电影语言，电影国家。

电影描述，电影图片，创建时间。

影评信息数据表： 影评编号，影评内容，用户编号，电影编号，创建时间，

新闻信息数据表： 新闻编号，新闻名称，创建时间，管理员编号，创建时间。

用户信息数据表： 用户编号，用户姓名，用户密码，用户E-mail，创建时间。

以下列出了本系统在数据库内的详细数据库表设计。

### 3.1.6.1 管理员数据表

管理员数据表在数据库内的表名为admin，其中主键为adminid（管理员编号）。

其详细的字段设计如表3-1-6-1所示。

表3-1-6-1 admin（管理员数据表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 为空否 | 字段含义 | 主键否 |
| adminid | Int | 否 | 管理员ID | 是 |
| adminname | varchar（255） | 是 | 管理员姓名 | 否 |
| adminpassword | varchar（255） | 是 | 管理员密码 | 否 |
| createtime | datetime | 是 | 创建时间 | 否 |

3.1.6.2 电影数据表

电影数据表在数据库中的表名为movies, 其中主键为movielid（电影编号）。

其详细的字段设计如表3-1-7-2所示。

表3-1-7-2 movies（电影信息数据表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 为空否 | 字段含义 | 主键否 |
| Movieid | Int | 否 | 电影ID | 是 |
| Name | Varchar(255) | 是 | 电影名称 | 否 |
| Sort | Varchar(255) | 是 | 电影类型 | 否 |
| Lang | Varchar(255) | 是 | 电影语言 | 否 |
| Addr | Varchar(255) | 是 | 电影国家 | 否 |
| Intro | Varchar(255) | 是 | 电影描述 | 否 |
| Images | Varchar(255) | 是 | 电影图片 | 否 |
| Createtime | Datetime | 是 | 创建时间 | 否 |

### 3.1.6.3 用户信息数据表

数据表在数据库中的表名为users，其中主键为userid（用户编号）。

其详细的字段设计如表3-1-6-3所示。

表3-1-6-3 users（用户信息数据表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 为空否 | 字段含义 | 主键否 |
| Userid | Int | 否 | 用户ID | 是 |
| Username | Varchar(255) | 是 | 用户姓名 | 否 |
| Userpassword | Varchar(255) | 是 | 用户密码 | 否 |
| Email | Varchar(255) | 是 | 用户E-mail | 否 |
| Createtime | Datetime | 是 | 创建时间 | 否 |

### 3.1.6.4 影评信息数据表

评论记录数据表在数据库中的表名为comment，其中主键为commentId（影评编号）。

其详细的字段设计如表3-1-6-4所示。

表3-1-6-4 comment（影评信息数据表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 为空否 | 字段含义 | 主键否 |
| CommentId | Int | 否 | 影评Id | 是 |
| Content | Varchar(255) | 是 | 影评内容 | 否 |
| Userid | Int | 是 | 发布人编号 | 否 |
| Movieid | Int | 是 | 电影编号 | 否 |
| Createtime | Datetime | 是 | 创建时间 | 否 |

# 3.2 详细设计

3.2.1 系统主要模块设计

3.2.1.2 管理员管理信息模块设计

管理员管理电影信息模块

电影信息管理模块可以实现管理员添加电影信息，修改电影信息，查看电影信息和删除电影信息。在添加电影信息的时候可以添加最新的电影信息，好让用户了解到最新的电影信息。同时可以删除不需要的电影信息来保证电影信息都保持最新的状态。

电影信息管理模块的流程是：管理员首先可以进行的操作是登录，如果操作成功则进入后台管理的界面在后台管理时选择电影信息管理的功能，可以实现对电影信息的添加，删除，修改和查看等功能。电影信息管理的流程图如图3-2-1-2-1所示。

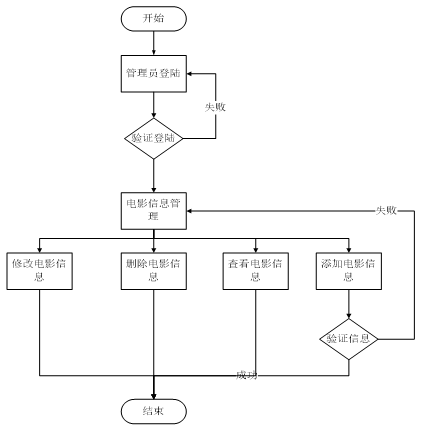


图3-2-1-2-1 管理员管理电影信息模块

管理员管理影评信息模块

影评信息管理模块可以实现管理员对用户所发表的对电影的影评进行查看和删除的操作，有些用户对电影发表影评信息的时候会输入一些误输入的信息，这样管理员就可以删除一些用户发表的影评信息，已达到更高效的管理程序的目的。

电影信息管理模块的流程是：管理员首先可以进行的操作是登录，如果操作成功则进入后台管理的界面在后台管理时选择电影信息管理的功能，可以实现对电影信息的添加，删除，修改和查看等功能。影评信息管理的流程图如图3-2-1-2-1所示。

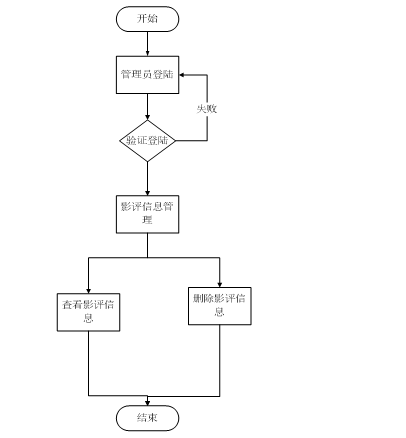


图3-2-1-2-2 管理员管理影评信息模块

用户信息操作模块设计

用户对电影信息操作模块可以实现用户对电影信息的查看和搜索的功能。在用户需要了解最新的电影信息的时候可以查看电影信息，也可以按照自己的需求对电影信息进行搜索。来满足用户对电影信息的需求。

用户电影信息操作的流程是：用户首先进行登录操作，在登陆之后可以点击查看电影信息，也可以按照分类进行电影信息的查看，同时可以按照需求对电影信息进行搜索操作。用户电影信息操作流程图如图3-2-1-2-3所示。

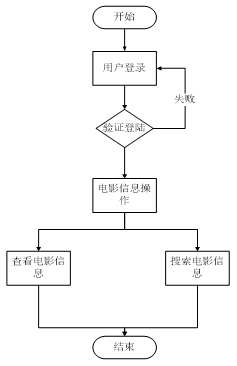


图3-2-1-2-3 用户操作信息模块

3.2.2 系统实现

3.2.2.1 程序主框架实现

通过Web技术设计并实现桌面应用程序，使用了Electron技术框架。Electron为用纯JavaScript创建桌面应用提供了运行时。原理是，Electron调用你在package.[json](http://www.liuhaihua.cn/archives/tag/json" \o "View all posts in json" \t "http://www.liuhaihua.cn/archives/_blank)中定义的main文件并执行它。main文件（通常被命名为main.js）会创建一个内含渲染完的web页面的应用窗口，并添加与你操作系统的原生G[UI](http://www.liuhaihua.cn/archives/tag/ui" \o "View all posts in UI" \t "http://www.liuhaihua.cn/archives/_blank)交互的功能。当用Electron启动一个应用，会创建一个主[进程](http://www.liuhaihua.cn/archives/tag/%e8%bf%9b%e7%a8%8b" \o "View all posts in 进程" \t "http://www.liuhaihua.cn/archives/_blank)。这个主进程负责与你系统原生的GUI进行交互并为你的应用创建GUI。如下图3-2-2-1所示。

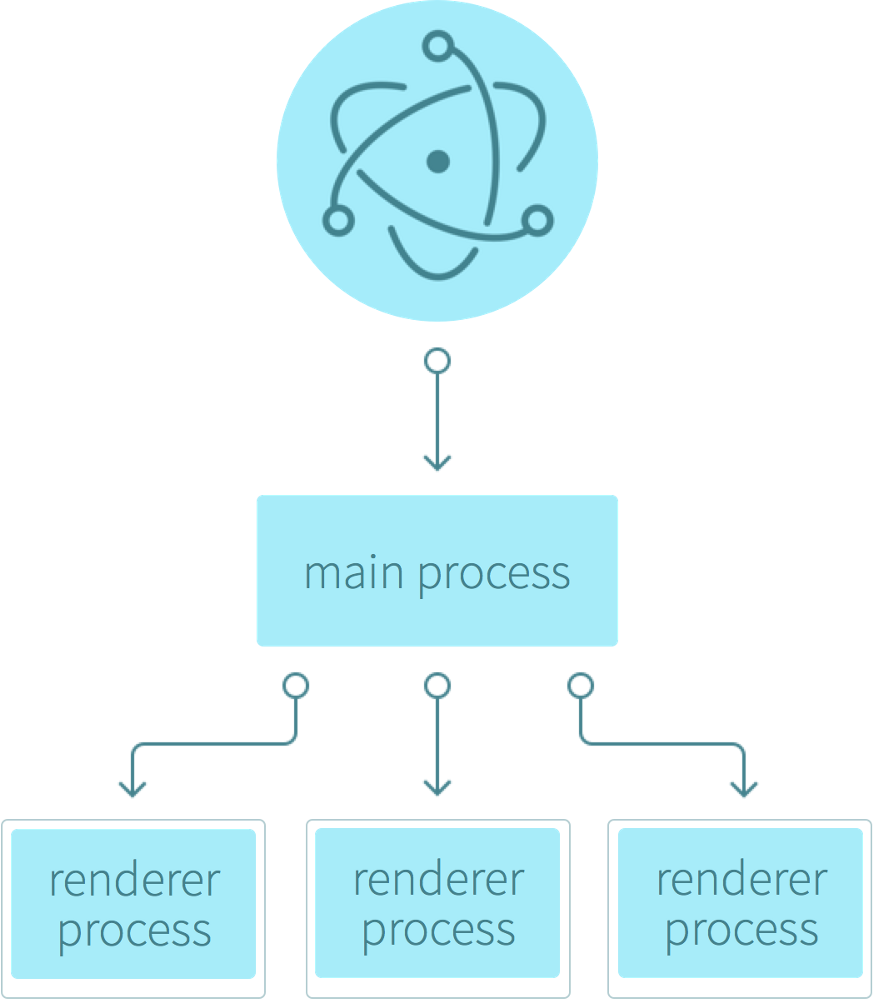


图3-2-2-1 Electron创建流程

其中Main.js的核心代码如下：

**var** app **=** require**(**'app'**);**

**var** BrowserWindow **=** require**(**'browser-window'**);**

// 当窗口冲突时警告.

require**(**'crash-reporter'**).**start**();**

**var** mainWindow **=** **null;**

// 当窗口关闭时退出.

app**.**on**(**'window-all-closed'**,** **function()** **{**

// On OS X it is common for applications and their menu bar

// to stay active until the user quits explicitly with Cmd + Q

**if** **(**process**.**platform **!=** 'darwin'**)** **{**

app**.**quit**();**

**}**

**});**

app**.**on**(**'ready'**,** **function()** **{**

// 创建窗口.设置配置项

mainWindow **=** **new** BrowserWindow**({**

width**:** 1200**,**

height**:** 700**,**

fullscreen**:** **false,**

autoHideMenuBar**:** **true,**

**frame:** **false,**

movable**:** **true,**

webPreferences**:** **{**

plugins**:** **true**

**}**

**});**

// 加载App的index.html.

mainWindow**.**loadUrl**(**'file://' **+** \_\_dirname **+** '/index.html'**);**

// 打开调试工具栏.

mainWindow**.**openDevTools**();**

// 当窗口关闭时进行事件响应

mainWindow**.**on**(**'closed'**,** **function()** **{**

// Dereference the window object, usually you would store windows

// in an array if your app supports multi windows, this is the time

// when you should delete the corresponding element.

mainWindow **=** **null;**

**});**

**});**

3.2.2.2 添加自定义程序按钮

因为程序的最小化，最大化，关闭等按钮是采用操作系统默认按钮，因此如果想要改变为自定义按钮，则需要先把程序默认的外边框frame给隐藏，在上面的main.js已经设置，然后添加自定义按钮，并调用方法来给它们添加各自的功能，打到如图3-2-2-2的效果。

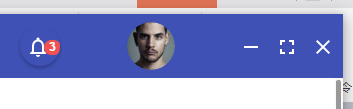


图3-2-2-2 程序按钮

核心代码如下：

angular**.**module**(**'app'**)**

**.**controller**(**'AppCtrl'**,** **[**'$rootScope'**,** '$scope'**,** '$interval'**,** '$timeout'**,** '$mdSidenav'**,** '$log'**,** **function(**$rootScope**,** $scope**,** $interval**,** $timeout**,** $mdSidenav**,** $log**)** **{**

**var** **self** **=** **this;**

**self.**notify **=** **false;**

$scope**.**remote **=** require**(**'remote'**);**

// 引入App模块

**var** App **=** $scope**.**remote**.**require**(**'app'**);**

**var** Win **=** $scope**.**remote**.**require**(**'browser-window'**);**

// 获取当前焦点窗体

**var** broWin **=** Win**.**getFocusedWindow**();**

**this.**searchValue **=** ''**;**

$scope**.**toggleRight **=** buildToggler**(**'left'**);**

$scope**.**isOpenRight **=** **function()** **{**

**return** $mdSidenav**(**'left'**).**isOpen**();**

**};**

**function** buildToggler**(**navID**)** **{**

**return** **function()** **{**

$mdSidenav**(**navID**)**

**.**toggle**()**

**.**then**(function()** **{**

$log**.**debug**(**'toggle ' **+** navID **+** ' is done'**);**

$log**.**debug**(**$mdSidenav**(**'left'**).**isOpen**());**

**});**

**};**

**}**

/\*\*

\* 最小化按钮响应方法

\* @return null

\*/

$scope**.**smallWin **=** **function()** **{**

broWin**.**minimize**();**

**};**

/\*\*

\* 最大化按钮响应方法

\* @return null

\*/

$scope**.**fullWin **=** **function()** **{**

// 如果程序没有最大化，则按钮生效，否则保持原状态

**if** **(!**broWin**.**isMaximized**())** **{**

broWin**.**maximize**();**

**}** **else** **{**

broWin**.**restore**();**

**}**

**};**

/\*\*

\* 关闭按钮响应方法

\* @return null

\*/

$scope**.**closeWin **=** **function()** **{**

App**.**quit**();**

**};**

$scope**.**toggleIt **=** **function()** **{**

**self.**notify **=** **!self.**notify**;**

**}**

$scope**.**logout **=** **function()** **{**

$mdSidenav**(**'left'**).**toggle**();**

**}**

**}]);**

3.2.2.3 系统界面设计

系统界面主要使用material design风格，本着简洁，统一的原则，力求争取最好的用户体验。程序首屏界面如图3-2-2-3-1所示

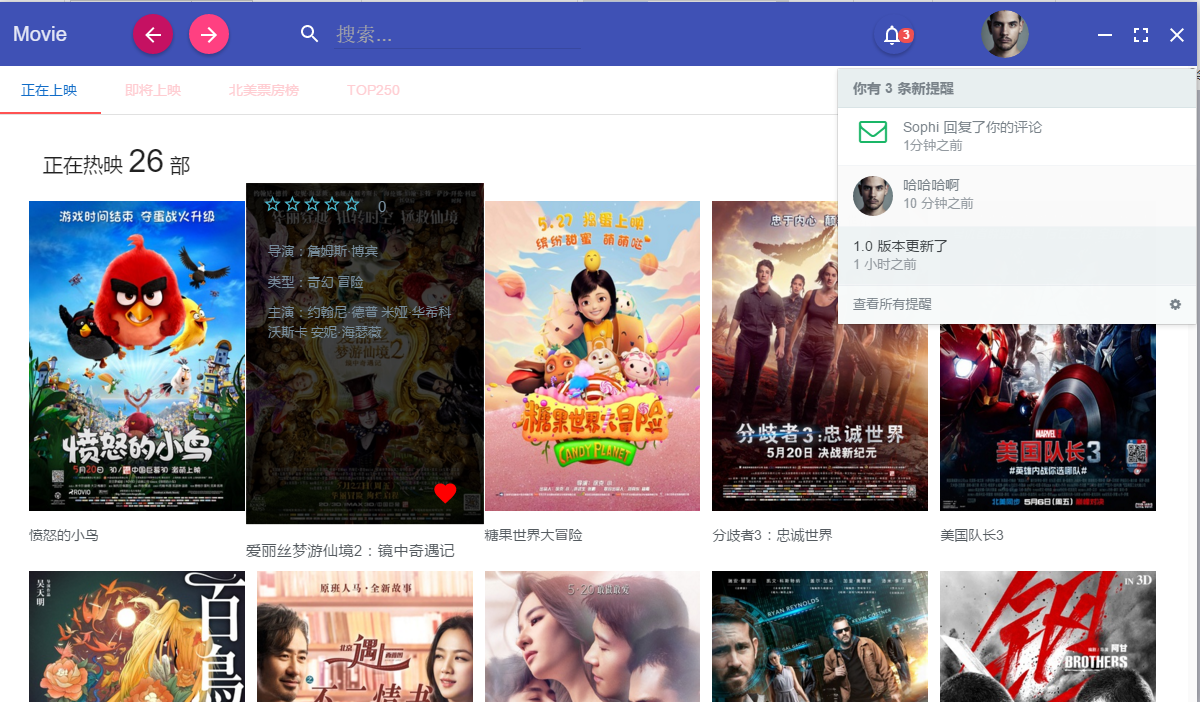


图3-2-2-3-1系统首屏

电影详情界面如图3-2-2-3-2所示



图3-2-2-3-3 电影详情

电影票房统计如图3-2-2-3-4所示

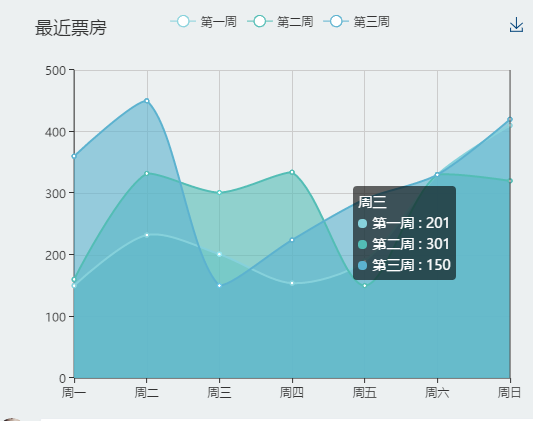


图3-2-2-3-4 电影票房

电影评论统计如图3-2-2-3-5所示



图3-2-2-3-5 电影评论

用户个人中心如图3-2-2-3-6所示



图3-2-2-3-6 用户个人中心侧边栏

3.2.2.3 系统电影数据来源

因为系统的电影数据需要一个稳定的来源，而我目前尚不能提供，所以从用爬虫策略，通过定时爬取豆瓣电影的数据并将其信息导入到数据库中，来填充系统的电影数据。

豆瓣电影目前提供少部分电影的API，所以本系统的策略是通过爬虫抓取豆瓣电影网页中的电影id和title，将其导入到数据库中，电影详情则通过豆瓣所提供的api进行查询，然后动态渲染到页面中。

通过对豆瓣电影前端页面分析，发现豆瓣电影的上映电影条目为一个id为now-playing的ul标签中，id为电影id，data-title为电影名称。

然后写好代码，然后跑在阿里云服务器定时上抓取豆瓣页面，因为豆瓣具有一定的反爬虫策略1分钟限制次数，所以需要将爬虫的频率降低为1s一次请求。

如果爬虫爬取到的电影在数据库中已经存在，则跳过插入数据库操作。

部分代码如下：

**var** mysql **=** require**(**'mysql'**);**

**var** request **=** require**(**'request'**);**

**var** cheerio **=** require**(**'cheerio'**);**

**var** path **=** require**(**'path'**);**

// 爬虫请求配置

**var** options **=** **{**

url**:** 'https://movie.douban.com/nowplaying/chengdu/'**,**

headers**:** **{**

'User-Agent'**:** 'request'

**}**

**};**

// 数据库配置

**var** connection **=** mysql**.**createPool**({**

host**:** 'localhost'**,**

user**:** 'root'**,**

**password:** '\*\*\*\*\*\*\*'**,**

database**:** 'movieTest'

**});**

/\*\*

\* 连接数据库

\* @return {[type]} [description]

\*/

**function** connectSQL**()** **{**

connection**.**connect**(function(**err**)** **{**

**if** **(**err**)** **{**

console**.**error**(**'error connecting: ' **+** err**.**stack**);**

**return;**

**}**

console**.**log**(**'connected as id ' **+** connection**.**threadId**);**

**});**

**}**

/\*\*

\* 插入数据实体

\* @param {object} item 数据实体

\* @return {[type]} null

\*/

**function** insertItem**(**item**)** **{**

connection**.**query**(**'INSERT INTO films SET ?'**,** item**,** **function(**err**,** result**)** **{**

**if** **(**err**)** **throw** err**;**

console**.**log**(**'正在向数据库中插入数据' **+** item**.**id **+** '\t' **+** item**.**title**);**

**});**

**}**

/\*\*

\* 关闭数据库链接

\*

\*/

**function** endConn**()** **{**

connection**.**end**(function(**err**)** **{**

**if** **(**err**)** **{**

console**.**error**(**'error connecting: ' **+** err**.**stack**);**

**return;**

**}**

console**.**log**(**'done!'**);**

**});**

**}**

/\*\*

\* 插入数据判断

\* @param {[number]} id 电影信息id

\* @param {[string]} title 电影名称

\* @return null

\*/

**function** insertData**(**id**,** title**)** **{**

**var** item **=** **[];**

item**.**push**(**id**);**

connection**.**query**(**'SELECT \* FROM `films` WHERE `id` = ?'**,** item**,** **function(**err**,** results**)** **{**

**if** **(**err**)** **throw** err**;**

**if** **(**results**.**length **!==** 0**)** **{**

console**.**log**(**id **+** '此数据已经存在，跳过插入' **+** title**);**

**}** **else** **{**

insertItem**({**

id**:** id**,**

title**:** title

**});**

**}**

**});**

**}**

/\*\*

\* 发起请求

\* @param {[object]} options 请求配置

\* @param {[function]} 回调函数

\* @return null

\*/

request**(**options**,** **function(**error**,** response**,** body**)** **{**

**if** **(!**error **&&** response**.**statusCode **==** 200**)** **{**

$ **=** cheerio**.**load**(**body**);**

//抓取豆瓣网页中电影信息

**var** lists **=** $**(**'#nowplaying .list-item'**);**

**for** **(var** i **=** 0**;** i **<** lists**.**length**;** i**++)** **{**

**var** id **=** lists**[**i**].**attribs**.**id**;**

**var** title **=** lists**[**i**].**attribs**[**'data-title'**];**

insertData**(**id**,** title**);**

**}**

**}**

**});**

设置定时任务的部分代码如下

/\*\*

\* 定时爬取豆瓣电影任务

\* @param {[number]} hour 小时

\* @param {[number]} minute 分钟

\* @param {[number]} hour 任务名

\* @param {[function]} fn 回调函数

\*/

**var** registTask**=function(**hour**,**minute**,**taskname**,**fn**){**

**var** rule **=** **new** schedule**.**RecurrenceRule**();**

//每天这个时刻定时执行任务

rule**.**dayOfWeek **=** **[**0**,** **new** schedule**.**Range**(**1**,** 6**)];**

rule**.**hour **=** hour**;**

rule**.**minute **=** minute**;**

**var** j **=** schedule**.**scheduleJob**(**rule**,**

**function(){**

console**.**log**(**"开始爬取："**+**taskname**);**

fn**&&**fn**();**

**});**

**}**

registTask**(**23**,**30**,**'movies'**,function(){**

getTaoBaoFEDBokes**(**1**);**

//每天23:30开始爬

**});**

爬虫抓取到数据后，前端动态渲染数据，这样用户就可以看到最新电影信息了。

3.2.2.4 路由控制

网站分为管理员和前台用户，必须得根据权限来展示不同的界面和功能。可以利用angular的service来实现。Service可以看成是一个服务，服务实例就是一个实现了该接口的动态代理类。因为service是单例的，可以很方便的在所有view，controller，directives，filters和其他service中共享数据，而不用采取暴露全局变量的方式，封装性也有一定的保障。

在取得当前用户的权限集合后,将这个集合存档到对应的一个service中,然后:

(1) 将permissions存放到factory变量中,使之一直处于内存中,实现全局变量的作用,但却没有污染命名空间.

(2) 通过$broadcast广播事件,当权限发生变更的时候.

主要代码如下：

angular**.element(document).**ready**(function()** **{**

$**.**get**(**'/userPermission'**,** **function(**data**)** **{**

permissionList **=** data**;**

angular**.**bootstrap**(document,** **[**'app'**]);**

**});**

**});**

angular**.**module**(**'app'**)**

**.**service**(**'permissions'**,** **function** **(**$rootScope**)** **{**

**var** permissionList**;**

**return** **{**

setPermissions**:** **function(**permissions**)** **{**

permissionList **=** permissions**;**

$rootScope**.**$broadcast**(**'permissionsChanged'**)**

**}**

**};**

**});**

angular**.**module**(**'app'**)**

**.**factory**(**'permissions'**,** **function** **(**$rootScope**)** **{**

**var** permissionList**;**

**return** **{**

setPermissions**:** **function(**permissions**)** **{**

permissionList **=** permissions**;**

$rootScope**.**$broadcast**(**'permissionsChanged'**)**

**}**

**};**

**});**

确定UI组件的依据权限进行显隐，需要编写一个名为has-permission的directive,它会依据权限关系来进行显示或者隐藏元素.

<!-- 如果用户拥有权限则显示连接 -->

<div has-permission=**'Edit'**>

<a href=**"/#/courses/{{ id }}/edit"**> **{{ name }}**</a>

</div>

<!-- 如果用户没有Edit权限则能显示文本 -->

<div has-permission=**'!Edit'**>

**{{ name }}**

</div>

通过一个has-permission属性校验permission的name,如果当前用户有则显示,没有则隐藏.

angular**.**module**(**'app'**).**directive**(**'hasPermission'**,** **function(**permissions**)** **{**

**return** **{**

**link:** **function(**scope**,** **element,** attrs**)** **{**

**if(!**\_**.**isString**(**attrs**.**hasPermission**))**

**throw** "hasPermission value must be a string"**;**

// 处理下字符串

**var** value **=** attrs**.**hasPermission**.**trim**();**

**var** notPermissionFlag **=** value**[**0**]** **===** '!'**;**

// 判断

**if(**notPermissionFlag**)** **{**

value **=** value**.**slice**(**1**).**trim**();**

**}**

**function** toggleVisibilityBasedOnPermission**()** **{**

**var** hasPermission **=** permissions**.**hasPermission**(**value**);**

**if(**hasPermission **&&** **!**notPermissionFlag **||** **!**hasPermission **&&** notPermissionFlag**)**

**element.**show**();**

**else**

**element.**hide**();**

**}**

toggleVisibilityBasedOnPermission**();**

scope**.**$on**(**'permissionsChanged'**,** toggleVisibilityBasedOnPermission**);**

**}**

**};**

**});**

整个程序的路由主要如下代码所示

//设置App的依赖和路由

**var** app **=** angular**.**module**(**

'app'**,** **[**

'ngRoute'**,**

'ngMaterial'**,**

'ngAnimate'**,**

'ngRateIt'

**]**

**).**config**(**

**[**

'$routeProvider'**,**

**function(**$routeProvider**)** **{**

$routeProvider**.**when**(**

'/'**,** **{**

templateUrl**:** './view/home/content.html'

**}**

**);**

$routeProvider**.**when**(**

'/login'**,** **{**

templateUrl**:** './view/home/login.html'**,**

controller**:** 'LoginCtrl'**,**

permission**:** ''

**}**

**);**

$routeProvider**.**when**(**

'/logout'**,** **{**

templateUrl**:** './view/home/logout.html'**,**

controller**:** 'LogoutCtrl'**,**

permission**:** ''

**}**

**);**

$routeProvider**.**when**(**

'/admin'**,** **{**

templateUrl**:** './view/home/admin.html'**,**

controller**:** 'AdminCtrl'，

permission**:** 'admin'

**}**

**);**

$routeProvider**.**when**(**

'/allItems'**,** **{**

templateUrl**:** './view/home/allItems.html'**,**

controller**:** 'AllItemsCtrl'**,**

permission**:** 'admin'

**}**

**);**

$routeProvider**.**when**(**

'/items/:itemId'**,** **{**

templateUrl**:** './view/home/detail.html'**,**

controller**:** 'DetailCtrl'**,**

permission**:** ''

**}**

**);**

$routeProvider**.**otherwise**({**

redirectTo**:** '/'

**});**

**}**

**]**

**);**

当我们定义一个路由的时候增加一个permission的属性,属性的值就是有哪些权限才能访问当前url.然后通过routeChangeStart事件一直监听url变化.每次变化url的时候,去校验当前要跳转的url是否符合条件,然后决定是跳转成功还是跳转到错误的提示页面.

app**.**controller**(**'AppCtrl'**,** **function(**$scope**,** $location**,** permissions**)** **{**

$scope**.**$on**(**'$routeChangeStart'**,** **function(**scope**,** next**,** current**)** **{**

**var** permission **=** next**.** $$ route**.**permission**;**

**if(**\_**.**isString**(**permission**)** **&&** **!**permissions**.**hasPermission**(**permission**))**

$location**.**path**(**'/unauthorized'**);**

**});**

**});**

在每次view的route跳转前,在父容器的Controller中判断一些它到底有没有跳转的权限即可.

3.2.2.4 电影评分算法

一部电影的评分如果仅仅是用评分总分除以评分人数得到的平均分，这样是没有说服力的。因为如果1部电影的只有10个人看过，10个人评10分，则最终分数为10分，而另一部有10000人看过，但平均分为9.5分,那么这两部电影排名哪个更靠前是无法判定的。

本程序的评分规则采用“贝斯叶”算法，这种算法是如今比较流行的一种投票算法。如公式3-2-2-4所示



公式（3-2-2-4）电影评分算法

R = average for the movie (mean) = (Rating) （是用普通的方法计算出的平均分）  
 v = number of votes for the movie = (votes) （评分人数）  
 m = minimum （最少评分人数）  
 C = the mean vote across the whole report (currently 6.9) （目前所有电影的平均得分）

某种程度上，它借鉴了["贝叶斯推断"](http://www.ruanyifeng.com/blog/2011/08/bayesian_inference_part_one.html" \t "http://www.ruanyifeng.com/blog/2012/03/_blank)（Bayesian inference）的思想：既然不知道评分结果，那就先估计一个值，然后不断用新的信息修正，使得它越来越接近正确的值。

在这个公式中，m（总体平均分）是"先验概率"，每一次新的评分都是一个调整因子，使总体平均分不断向该项目的真实评分结果靠近。评分人数越多，该项目的"贝叶斯平均"就越接近算术平均，对排名的影响就越小。

因此，这种方法可以给一些评分人数较少的项目，以相对公平的排名。

3.2.2.4 数据可视化

[数据可视化](http://baike.baidu.com/view/69231.htm" \t "http://baike.baidu.com/_blank)，是关于数据视觉表现形式的科学技术研究。其中，这种数据的视觉表现形式被定义为，一种以某种概要形式抽提出来的信息，包括相应信息单位的各种属性和变量。

它是一个处于不断演变之中的概念，其边界在不断地扩大。主要指的是技术上较为高级的技术方法，而这些技术方法允许利用图形、[图像处理](http://baike.baidu.com/view/14662.htm" \t "http://baike.baidu.com/_blank)、[计算机视觉](http://baike.baidu.com/view/155265.htm" \t "http://baike.baidu.com/_blank)以及用户界面，通过表达、建模以及对立体、表面、属性以及动画的显示，对数据加以可视化解释。

本程序对数据库中数据的采集与分析，借助百度前端的数据可视化框架echarts将数据统计并展示出来，清晰有效地传达与沟通信息。

前端通过访问服务器，获取最近三周的票房数据，然后渲染图标，核心代码如下。

angular**.**module**(**'app'**)**

**.**controller**(**'DetailCtrl'**,** **[**'DataService'**,** 'VideoService'**,** '$scope'**,** '$q'**,** '$interval'**,** '$mdIcon'**,** '$routeParams'**,** **function(**DataService**,** VideoService**,** $scope**,** $q**,** $interval**,** $mdIcon**,** $routeParams**)** **{**

**var** **self** **=** **this;**

**var** dataBox **=** **{};**

**self.**activated **=** **true;**

**self.**determinateValue **=** 30**;**

// Iterate every 100ms, non-stop and increment

// the Determinate loader.

$interval**(function()** **{**

**self.**determinateValue **+=** 1**;**

**if** **(self.**determinateValue **>** 100**)** **{**

**self.**determinateValue **=** 30**;**

**}**

**},** 100**);**

**var** id **=** $routeParams**.**itemId**;**

// 电影详情配置

**var** options **=** **{**

url**:** 'https://api.douban.com/v2/movie/subject/' **+** id**,**

headers**:** **{**

'User-agent'**:** 'request'

**}**

**};**

// 电影视频预览信息配置

**var** videoOptions **=** **{**

url**:** 'https://movie.douban.com/subject/' **+** id**,**

headers**:** **{**

'User-agent'**:** 'request'

**}**

**};**

// 获取电影信息

DataService**.**getHotMovies**(**options**,** **true)**

**.**then**(function(**data**)** **{**

**self.**activated **=** **false;**

$scope**.**detail **=** data**;**

renderChart**();**

console**.**log**(**$scope**.**detail**);**

**});**

VideoService**.**getVideo**(**videoOptions**)**

**.**then**(function(**data**)** **{**

$scope**.**videoUrl **=** data**;**

**});**

**var** dataUrl **=** 'box/' **+** 'id'**;**

//与服务器信获取票房数据

$http**.**get**(**dataUrl**)**

**.**success**(function(**response**)** **{**

dataBox **=** response**.**records**;**

**});**

// 渲染图标

**function** renderChart**()** **{**

**var** myChart **=** echarts**.**init**(document.**getElementById**(**'main'**),** 'shine'**);**

// 指定图表的配置项和数据

**var** **option** **=** **{**

title**:** **{**

**text:** "最近票房"

**},**

tooltip**:** **{**

trigger**:** 'axis'

**},**

legend**:** **{**

data**:** dataBox**.**legend

**},**

toolbox**:** **{**

feature**:** **{**

saveAsImage**:** **{}**

**}**

**},**

grid**:** **{**

left**:** '3%'**,**

right**:** '4%'**,**

bottom**:** '3%'**,**

containLabel**:** **true**

**},**

xAxis**:** **[{**

type**:** 'category'**,**

boundaryGap**:** **false,**

data**:** dataBox**.**category

**}],**

yAxis**:** **[{**

type**:** 'value'

**}],**

color**:** **[**

'#86D1DB'**,**

'#53BDB5'**,**

'#5CB2CF'

**],**

series**:** **[{**

name**:** '第一周'**,**

type**:** 'line'**,**

smooth**:** **true,**

areaStyle**:** **{**

normal**:** **{**

opacity**:** 0.6

**}**

**},**

data**:** dataBox**.**data**[**0**]**

**},** **{**

name**:** '第二周'**,**

type**:** 'line'**,**

smooth**:** **true,**

areaStyle**:** **{**

normal**:** **{**

opacity**:** 0.6

**}**

**},**

data**:** dataBox**.**data**[**1**]**

**},** **{**

name**:** '第三周'**,**

type**:** 'line'**,**

smooth**:** **true,**

label**:** **{**

normal**:** **{}**

**},**

areaStyle**:** **{**

normal**:** **{**

opacity**:** 0.6

**}**

**},**

data**:** dataBox**.**data**[**2**]**

**}]**

**};**

// 使用刚指定的配置项和数据显示图表。

myChart**.**setOption**(option);**

**}**

**}]);**

渲染后如图3-2-2-4-1所示

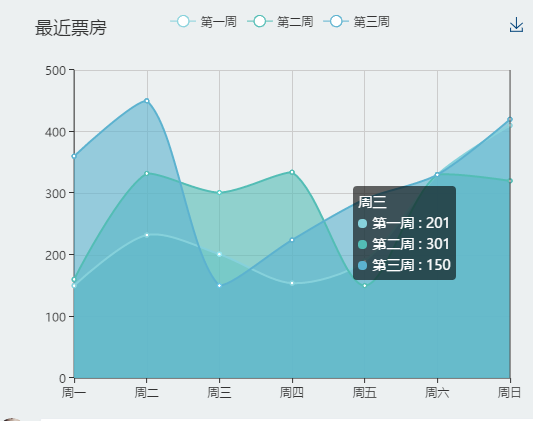


图3-2-2-4-1 最近票房数据图

观影指数的配置如下

**option** **=** **{**

backgroundColor**:** '#404a59'**,**

title**:** **{**

**text:** '美国队长3全国观影指数'**,**

subtext**:** ''**,**

sublink**:** ''**,**

left**:** 'center'**,**

textStyle**:** **{**

color**:** '#fff'

**}**

**},**

tooltip **:** **{**

trigger**:** 'item'

**},**

legend**:** **{**

orient**:** 'vertical'**,**

y**:** 'bottom'**,**

x**:**'right'**,**

data**:[**'观影指数'**],**

textStyle**:** **{**

color**:** '#fff'

**}**

**},**

geo**:** **{**

map**:** 'china'**,**

label**:** **{**

emphasis**:** **{**

show**:** **false**

**}**

**},**

roam**:** **true,**

itemStyle**:** **{**

normal**:** **{**

areaColor**:** '#323c48'**,**

borderColor**:** '#111'

**},**

emphasis**:** **{**

areaColor**:** '#2a333d'

**}**

**}**

**},**

series **:** **[**

**{**

name**:** '观影指数'**,**

type**:** 'scatter'**,**

coordinateSystem**:** 'geo'**,**

data**:** convertData**(**data**),**

symbolSize**:** **function** **(**val**)** **{**

**return** val**[**2**]** **/** 10**;**

**},**

label**:** **{**

normal**:** **{**

formatter**:** '{b}'**,**

position**:** 'right'**,**

show**:** **false**

**},**

emphasis**:** **{**

show**:** **true**

**}**

**},**

itemStyle**:** **{**

normal**:** **{**

color**:** '#ddb926'

**}**

**}**

**},**

**{**

name**:** 'Top 5'**,**

type**:** 'effectScatter'**,**

coordinateSystem**:** 'geo'**,**

data**:** convertData**(**data**.**sort**(function** **(**a**,** b**)** **{**

**return** b**.**value **-** a**.**value**;**

**}).**slice**(**0**,** 6**)),**

symbolSize**:** **function** **(**val**)** **{**

**return** val**[**2**]** **/** 10**;**

**},**

showEffectOn**:** 'render'**,**

rippleEffect**:** **{**

brushType**:** 'stroke'

**},**

hoverAnimation**:** **true,**

label**:** **{**

normal**:** **{**

formatter**:** '{b}'**,**

position**:** 'right'**,**

show**:** **true**

**}**

**},**

itemStyle**:** **{**

normal**:** **{**

color**:** '#f4e925'**,**

shadowBlur**:** 10**,**

shadowColor**:** '#333'

**}**

**},**

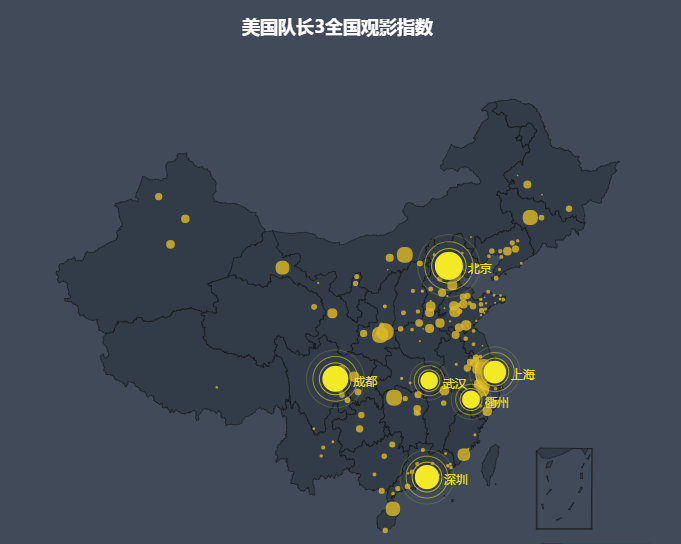
zlevel**:** 1

**}**

**]**

**};**

渲染出如下图3-2-2-4-2的效果



3.3 本章小结

本章在详细需求分析的基础上，对复杂软件工程问题进行推理、分析，阐述了针对复杂软件工程问题而开展的总体设计和详细设计。分别进行了系统总体设计，系统功能设计，系统结构设计，数据库设计，以及数据库表设计。

在详细设计的基础上，进一步阐述针对系统主体框架设计，系统数据来源，如何界面设计，路由控制，评分算法以及数据可视化等复杂工程问题的具体实现以及结果分析。

第四章 ××××××

……

第五章 结束语

5.1 本文内容

本文主要讨论了OFDM系统的一种改进形式SIM-OFDM系统。我们先介绍了OFDM系统，并在此基础上引出了SIM-OFDM系统，介绍了SIM-OFDM系统的基本原理和模型。……

5.2 顶岗实习项目课题有待进一步解决的问题及方向

……

5.3 对软件工程实践以及软件工程领域发展的认识

……

5.4 本人毕业设计（顶岗实习）收获及体会

……

参考文献

[1] 啜钢，王文博，常永宇等.移动通信原理与应用[M].北京：北京邮电大学出版社，2002.1-13

[2] 佟学俭，罗涛等.OFDM移动通信技术原理与应用[M].北京：人民邮电出版社,2003年6月.1-21

[3] B. Stantchev, G. Fettweis. Time-variant distortions in OFDM[J]. *IEEE Communications Letters*, Vol. 4, No. 10, October 2000, pp. 312-314

[4] P.H. Moose. A technique for orthogonal frequency division multiplexing frequency offset correction[J].*IEEE Trans.Communication*,Vol.42,No.10,October 1994, pp. 2908-2914

[5] J.-J. Van de Beek, M. Sandell, P.O. Brjesson. ML Estimation of Timing and Frequency Offset in OFDM Systems[J]. *IEEE Trans. Signal Processing*, Vol. 45,No.7, July 1997, pp. 1800-1805

[6] R. Abualhiga and H. Haas.Subcarrier-index modulation OFDM[C]. in *Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun*, Tokyo, Japan, 2009, pp. 13-16.

[7] Mesleh R Y, Haas H, Sinanovic S. Spatial modulation[J]. *IEEE Transactions on Vehicular Technology*, 2008,Vol.57,No.4, pp. 2228-2241.

[8] D. Tsonev, S. Sinanovic, and H. Haas.Enhanced subcarrier index modulation (SIM) OFDM[C].in *Proc. IEEE GLOBECOM Workshops*, Houston,Tx2011, pp. 728-732.

[9] 李晓峰，周宁.周亮.通信原理[M]。北京：清华大学出版社，1973.22-25

[10] E. Basar, U. Aygolu, E. Panayirci, and H. V. Poor. Orthogonal frequency division multiplexing with index modulation[C]. in *Proc. IEEE GLOBECOM*, 2012,pp. 4741-4746.

[11] E. Basar, U. Aygolu, E. Panayirci, and H. V. Poor. Orthogonal frequency division multiplexing with index modulation[J].*IEEE Trans. Signal Process*., Vol. 61, No. 22, Nov. 2012,pp. 5536–5549.

[12] Youngwook Ko.A Tight Upper Bound on Bit Error Rate of Joint OFDM and Multi-Carrier Index Keying[J]. *IEEE Communications Letters,* Vol. 18, N0.10,Oct. 2014,pp. 1763 - 1766 .

[13] A.Goldsmith.无线通信[M]（杨鸿文，李卫东，郭文彬等译）.北京：人民邮电出版社，2006.143~161

[14] Y.Zhao and S.G.Haggman.BER analysis of OFDM communication systems with intercarrier interference [C].in *IEEE Int. Conf. Communication Technology*,Beijing,China,Oct.1998,pp.1-5.

[15] L.Rugini and P.Banelli.BER of OFDM Systems Impaired by Carrier Frequency Offset in Multipath Fading Channels[J].*IEEE Transactions on Wireless Communications,*Vol.4,No.5,Sep. 2005 ,pp. 2279-2288.

[16] X.Ma,H.Kobayashi and S.C.Schwartz.EFFECT OF FREQUENCY OFFSET ON BER OF OFDM AND SINGLE CARRIER SYSTEMSR[C].*IEEE International Symposium on Personal,Indoor and Mobile Radio Communication Proceedings*,Sep.2003,pp.2239-2243.

致谢

本论文的工作是在我的企业指导导师XX老师与院内代管教师XX老师的悉心指导下完成的，……

外文资料原文

A Tight Upper Bound on Bit Error Rate of Joint OFDM and Multi-Carrier Index Keying

II.Joint MCIK-OFDM System Model

We consider a peer-to-peer M-QAM OFDM transmission with Nc sub-carriers

that consists of n clusters of N sub—carriers(i.eNc=nN).A stream of M-QAM

symbols is first serial-to-parallel converted,where every n()symbols are grouped into a vector  and  are used to modulate sub-carriers,as in the

classical OFDM,but it differs from that the modulated sub-carriers are only those of n activated indices,similar to [4],[5].

……

外文资料译文

基于多载波索引键控的正交多路复用系统的误码率上界

二．基于多载波索引键控的正交频分多路复用系统模型

我们考虑一个端到端的M-QAM，Nc子载波的基于多载波索引键控的正交频分多路复用系统有n个簇，每个簇有N个子载波（Nc=nN）。M-QAM的符号流经过串并转换之后每n个符号组成一个相量，是和传统正交频分多路复用一样是用来调制子载波的，但是不同的是只有这n个活跃子载波进行了调制。……

……