**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](#_Toc4223)

[1.1 实习单位与实习岗位概况，对企业所属行业的认识 1](#_Toc8205)

[1.2 实习项目课题背景、价值、意义以及国内外研究现状 1](#_Toc29825)

[1.2.1 课题背景 1](#_Toc12026)

[1.3 实习项目整体执行完成情况概述 3](#_Toc21634)

[第二章 复杂工程问题归纳与实施方案可行性研究 5](#_Toc12225)

[2.1 课题目标与任务 5](#_Toc6767)

[2.2 相关复杂工程问题 6](#_Toc1365)

[2.3 实施方案与可行性研究 6](#_Toc17934)

[2.4 可行性分析 7](#_Toc14692)

[2.5 功能需求 8](#_Toc22860)

[2.5.1 管理员功能需求分析 8](#_Toc9105)

[2.5.2 用户功能需求分析 9](#_Toc28938)

[2.6 界面需求 10](#_Toc32441)

[2.7 性能需求 10](#_Toc2702)

[2.8 系统主要流程图 11](#_Toc29888)

[2.9 本章小结 12](#_Toc11356)

[第三章 针对复杂工程问题的方案设计与实现 13](#_Toc29187)

[3.1 概要设计 13](#_Toc23925)

[3.1.1 系统功能设计 13](#_Toc26906)

[3.1.2 系统用户用例设计 14](#_Toc7140)

[3.1.3 系统结构设计 14](#_Toc14121)

[3.1.4 数据库设计 15](#_Toc7063)

[3.1.5 数据库概述 16](#_Toc26899)

[3.2 详细设计 19](#_Toc5407)

[3.3 本章小结 45](#_Toc11485)

[第四章 软件测试 46](#_Toc6116)

[4.1 系统测试概论 46](#_Toc27481)

[4.2 本系统采用的测试方法 46](#_Toc8688)

[4.3.2 信息修改测试用力 47](#_Toc23927)

[第五章 知识技能学习情况 49](#_Toc30318)

[5.1 开发环境 49](#_Toc19601)

[5.2 技术学习 49](#_Toc7168)

[第六章 工程计划管控和执行情况 53](#_Toc22880)

[6.1 实施计划 53](#_Toc31403)

[6.2 项目风险](#_Toc13623) **[错误！未定义书签。](#_Toc13623)**

[6.3 本章小结 5](#_Toc27191)4

[第七章 职业素养与工程伦理的学习与培养 55](#_Toc19516)

[第八章 对软件工程实践以及软件工程领域发展的认识 57](#_Toc26615)

[第九章 结束语 59](#_Toc22496)

[9.1 本文内容 59](#_Toc19975)

[9.2 顶岗实习项目课题有待进一步解决的问题及方向 59](#_Toc11357)

[9.3 对软件工程实践以及软件工程领域发展的认识 59](#_Toc22445)

[9.4 本人毕业设计（顶岗实习）收获及体会 60](#_Toc10426)

[参考文献 61](#_Toc2761)

[致谢 62](#_Toc14953)

[外文资料原文 63](#_Toc27093)

[外文资料译文 67](#_Toc8606)

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。

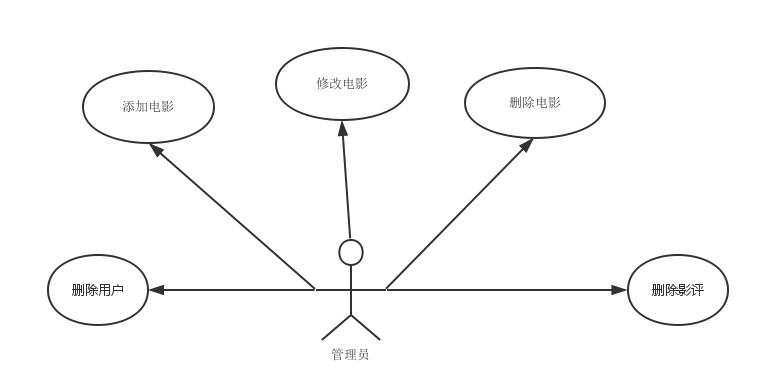


图2-5-1 管理员用例图

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。

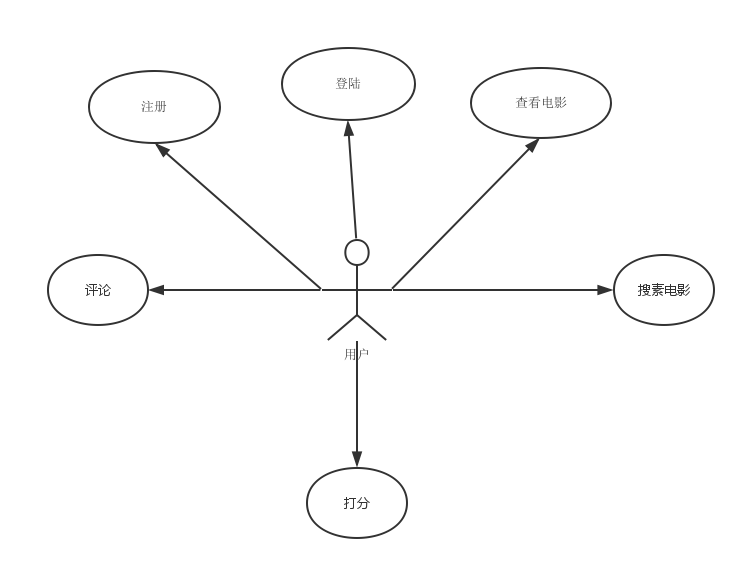


图2-5-2 用户用例图

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

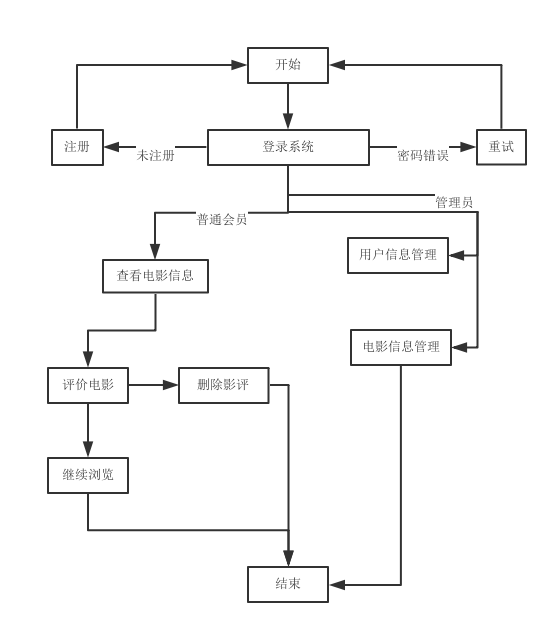


图2-8 系统流程图

2.9 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。

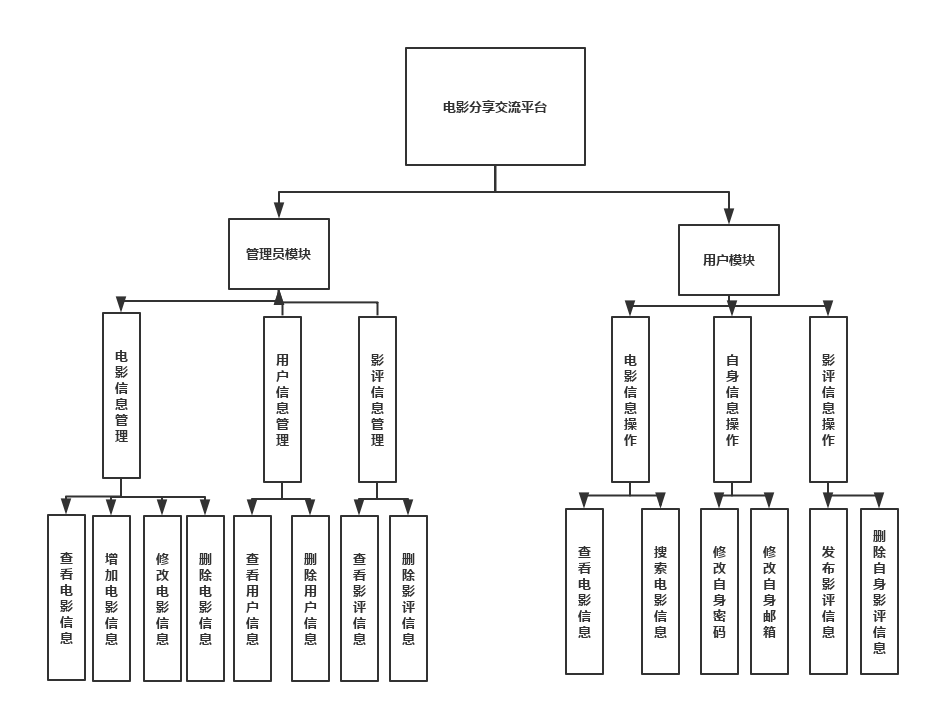


图3-1-1 系统功能模块图

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。

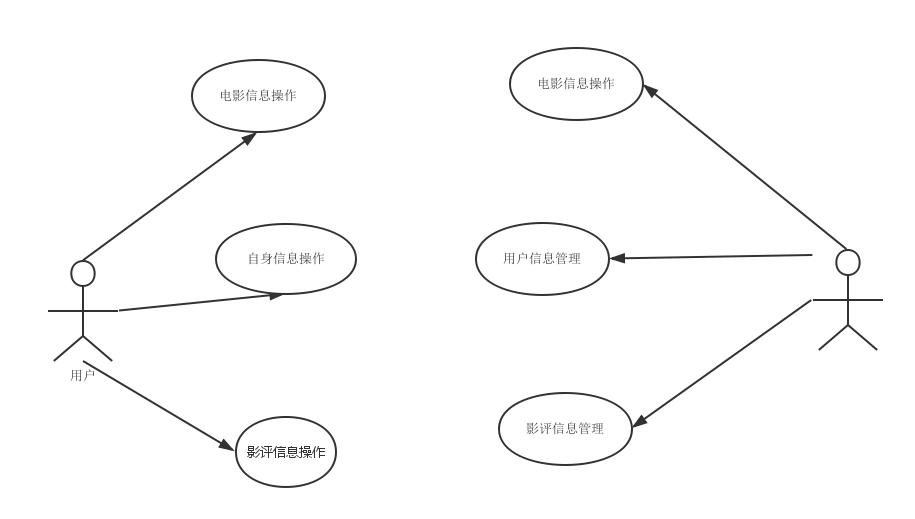


图3-1-2 系统用例图

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

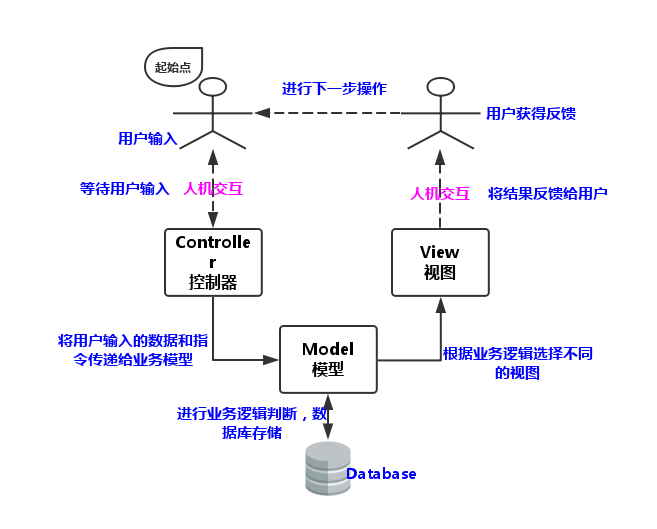


图3-1-3 系统结构图

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：

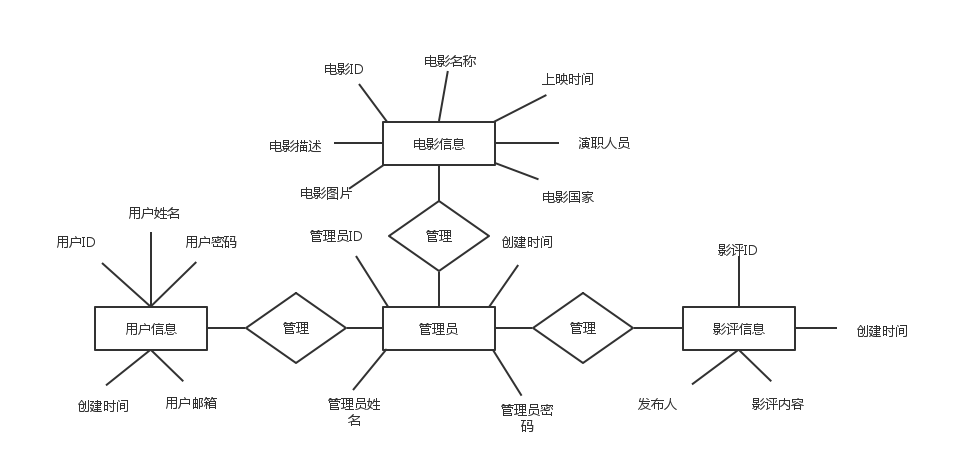


图3-1-5 数据库E-R图

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。

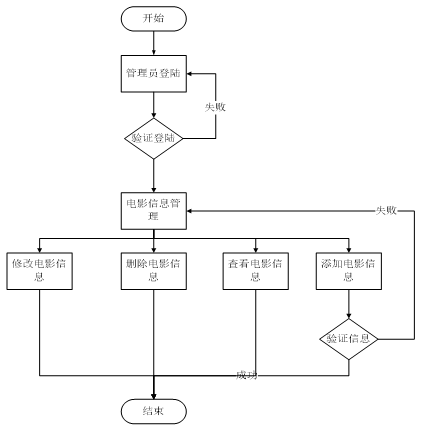


图3-2-1-2-1 管理员管理电影信息模块

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。

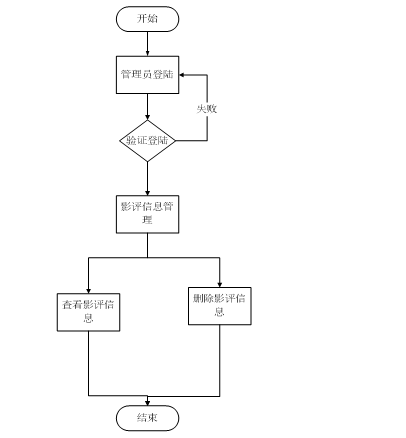


图3-2-1-2-2 管理员管理影评信息模块

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。

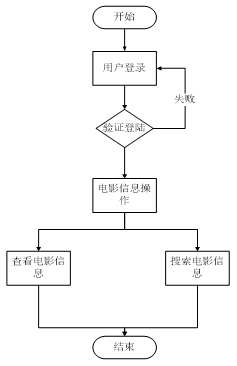


图3-2-1-2-3 用户操作信息模块

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。

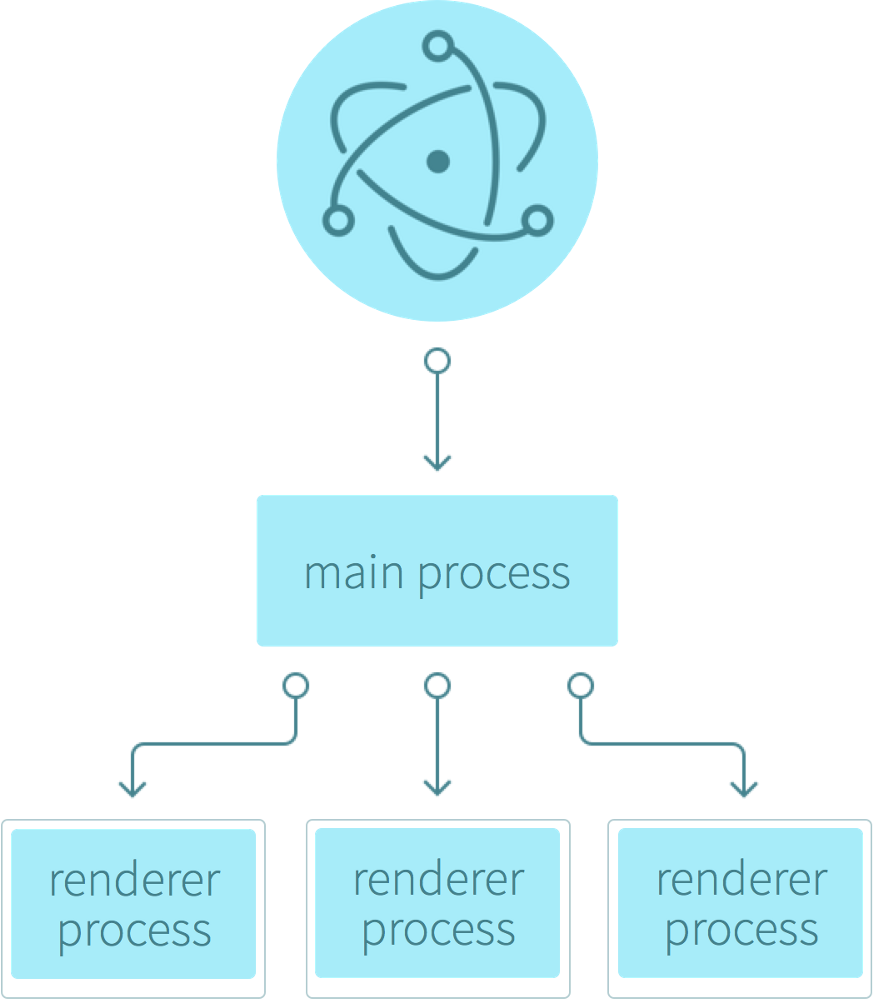


图3-2-2-1 Electron创建流程

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。

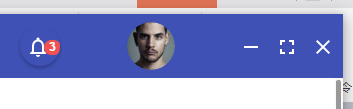


图3-2-2-2 程序按钮

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

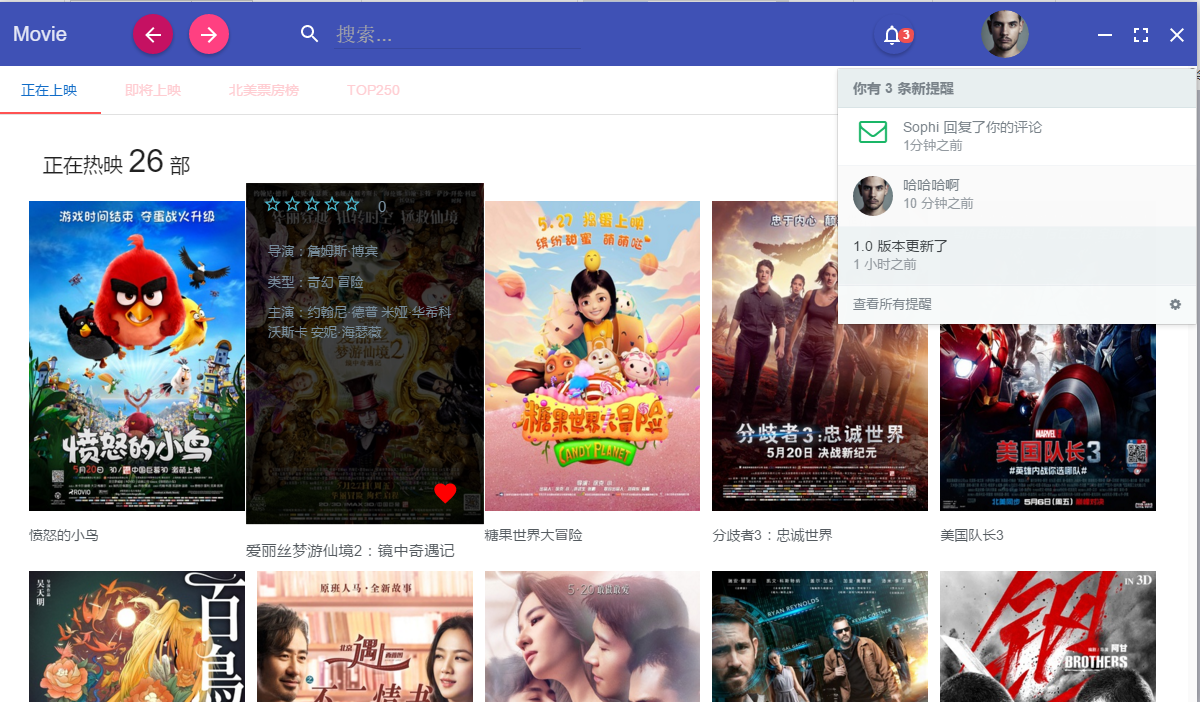


图3-2-2-3-1系统首屏

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

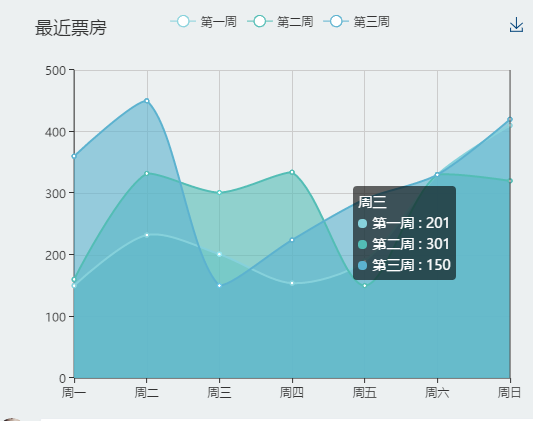


图3-2-2-3-4 电影票房

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

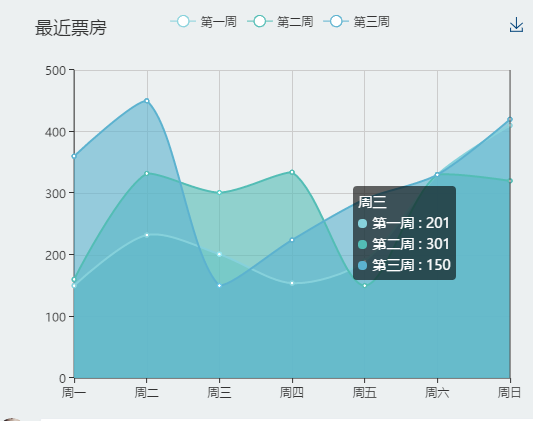


图3-2-2-4-1 最近票房数据图

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'

**}**

**},**

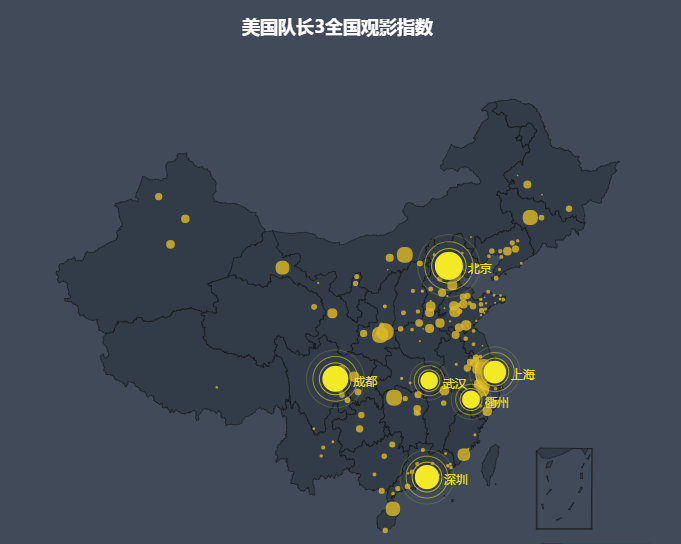
zlevel**:** 1

**}**

**]**

**};**

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 软件测试

## 4.1 系统测试概论

系统测试目的是通过与系统的需求相比较，发现所开发的系统与用户需求不符或矛盾的地方，从而提出更加完善的方案。它的任务是尽可能彻底的检查出程序中的错误，提高软件系统的可靠性，这阶段又可分为：模块测试，测试每个模块的程序是否有错误；确认测试，测试整个软件系统是否满足用户功能和性能的要求。该阶段结束应交付测试报告，说明测试数据的选择，测试用例以及测试结果是否符合预期结果。

在实际开发系统的各个阶段都有可能产生错误，为了使系统最终能够正常的运行，系统开发人员在系统开发的整个过程中需要反复进行系统测试。

## 4.2 本系统采用的测试方法

系统测试分为两种测试方法：白盒测试和黑盒测试。

白盒测试也称结构测试或逻辑驱动测试，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行。按照程序内部的结构测试程序，检验程序中的每条通路是否都有能按预定要求正确工作，白盒测试的主要方法有逻辑驱动、基路测试等，主要用于软件验证。

本系统为黑盒测试，黑盒测试也称功能测试，通过测试来检测每个功能是否都能正常使用。在测试时，在完全不考虑程序内部结构和内部特性的情况下，检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息，并且保持外部信息（如数据库或文件）的完整性。  
 黑盒测试是指不基于内部设计和代码的任何知识，而基于需求和功能性的测试，黑盒测试也称功能测试或数据驱动测试，它是在已知产品所应具有的功能， 通过测试来检测每个功能是否都能正常使用，在测试时，把程序看作一个不能打开的黑盆子，在完全不考虑程序内部结构和内部特性的情况下，测试者在程序接口进 行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数锯而产生正确的输出信息，并且保持外部信息（如数据库或文 件）的完整性。黑盒测试方法主要有等价类划分、边值分析、因—果图、错误推测等，主要用于软件确认测试。  
 “黑盒”法着眼于程序外部结构、不考虑内 部逻辑结构、针对软件界面和软件功能进行测试。“黑盒”法是穷举输入测试，只有把所有可能的输入都作为测试情况使用，才能以这种方法查出程序中所有的错 误。实际上测试情况有无穷多个，人们不仅要测试所有合法的输入，而且还要对那些不合法但是可能的输入进行测试。

4.3 本系统的功能及错误测试

接下来对本系统做以详尽的测试。

4.3.1 系统登录测试用例

进入管理员登录页面时，为了测试管理员的登录功能。在登陆时如果不输入管理员名，单击“登录”按钮后，系统将转到如图4-1所示的用户名或密码有误的页面。

然后测试管理员密码，在管理员的用户名或者密码错误的时候，也将要跳转到管理员名或者密码有误页面。

表4-1 系统登陆测试用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 测试项 | 操作步骤 | 预期结果 | 输入数据 | 实际结果 | 结果比较说明 |
| 001 | 系统登录 | 系统首页不输入用户名密码 | 用户名，密码为真进入系统，否则，提示出错信息。 | null  null | 不符合，请输入登陆的号码 | 符合 |
| 002 | 系统登录 | 在系统首页输入用户名密码 | 用户名，密码为真进入系统，否则，提示出错信息。 | 10000  0000 | 符合，登录系统 | 符合 |

### 4.3.2 信息修改测试用力

在管理员顺利登陆之后，可以选择电影信息管理来进行对电影信息的管理信息，这里我们点击任何一条电影信息的编辑按钮，进入编辑电影信息的页面。在这里我们输入超过数据库定义的长度的电影信息，则会跳转到修改电影信息失败的页面。

表4-3-2 信息修改测试用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 测试项 | 操作步骤 | 预期结果 | 输入数据 | 实际结果 | 结果比较说明 |
| 001 | 电影信息管理 | 在编辑电影信息页面输入新闻信息 | 输入信息超过数据库定义长度，提示出错信息。 | null  null | 不符合，请输入登陆的号码 | 符合 |
| 002 | 电影信息管理 | 在编辑电影信息页面输入新闻信息 | 输入信息超过数据库定义长度，提示出错信息。 | 10000  0000 | 符合，登录系统 | 符合 |

到此，对于系统内的信息输入进行了详细的验证，同时也对系统内的功能进行了完整的测试。本系统的功能测试基本完成。

本系统有待日后继续完善，希望有一天可以达到满足各个层次的用户的使用需求。

第五章 知识技能学习情况

## 5.1 开发环境

1. 硬件：Windows10 PC一台
2. 服务器：阿里云服务器  Ubuntu 14.04 64位
3. IDE：ATOM编辑器
4. 技术标准：Electron NodeJs AngularJs Javacript Html Css

## 5.2 技术学习

5.2.1 Electron

Electron 框架的前身是 Atom Shell，可以使用 [JavaScript](http://www.yyyweb.com/front" \t "http://www.cnblogs.com/lhb25/p/_blank)，[HTML](http://www.yyyweb.com/tag/html5" \t "http://www.cnblogs.com/lhb25/p/_blank)和[CSS](http://www.yyyweb.com/tag/css3" \t "http://www.cnblogs.com/lhb25/p/_blank) 构建跨平台的桌面应用程序。它是基于Node.js 和 Chromium 开源项目。最重要的是，Electron 应用服务构建和运行在 Mac，Windows 和 Linux。

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)（图形界面）交互的功能。详细地说，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。

5.2.2 AngularJs

选择AngularJs的原因：这是一个完全使用JavaScript的客户端技术，配合HTML, CSS, JavaScript使用，开发Web应用更简单，更快捷。AngularJs主要用于构建单页面Web应用，降低了构建复杂应用的难度，它提供了开发者在现代Web应用中经常要用到的一系列高级功能，例如：

解耦应用逻辑，数据模型和视图；

Ajax服务；

依赖注入；

浏览历史（使书签和前进，后退按钮能够像在普通Web应用中一样工作）；

测试等

Angular应用程序由三个部分组成：

模板（Template）

模板是用HTML和CSS编写的文件，展现应用的视图。可以给HTML添加新的元素、属性标记，作为AngularJS编译器的指令。 AngularJS编译器是完全可扩展的，这意味着通过AngularJS可以在HTML中构建自己的HTML标记。

应用程序逻辑（Logic）和行为（Behavior）

应用程序逻辑和行为是用JavaScript定义的控制器。AngularJS与标准AJAX应用程序不同，不需要另外编写侦听器或DOM控制器，因为它们已经内置到AngularJS中了。这些功能使的应用程序逻辑很容易编写、测试、维护和理解。

模型数据（Data）

模型是从AngularJS作用域对象的属性引申的。模型中的数据可能是Javascript对象、数组或基本类型，这都不重要，重要的是，他们都属于AngularJS作用域对象。

AngularJS通过作用域来保持数据模型与视图界面UI的双向同步。一旦模型状态发生改变，AngularJS会立即刷新反映在视图界面中，反之亦然。

在学习AngularJs的时候，发现Angular的数据绑定真的是很酷。它将节省你大量的样板代码编写。 一个典型的Web应用程序可以包含多达80％的代码基础，如遍历，操作，并听取了监听DOM。 数据绑定使得不用编写这些代码，这样你就可以专注于你的应用程序。AngularJs的数据双向绑定如图4-2-2所示

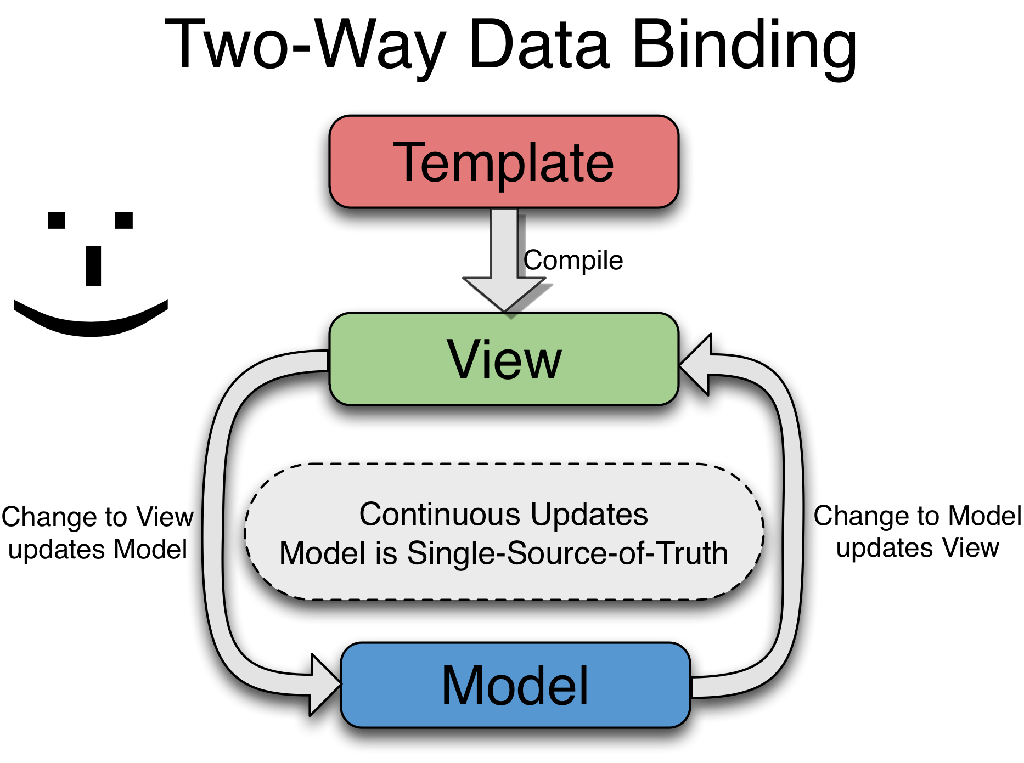


图4-2-2 Angular数据双向绑定

5.2.3 NodeJs

使用Node的原因：如果使用Node.Js，这样前端和后端就可以使用同样一种语言，一个Web开发者学会Node.Js是简单的，学习成本低。

Node.js使用的是事件驱动架构，这就是说在它上面做的每一件事，每一个独立的调用和操作，都是一系列的异步回调。这使得Node.js运行在一个线程上，这一点和其它的Web技术是不同的，它们每个客户端的请求都会生成一个新的线程来处理。这也正是非阻塞I/O的本质所在。

使用Node的原因：如果使用Node.Js，这样前端和后端就可以使用同样一种语言，一个Web开发者学会Node.Js是简单的，学习成本低。

Node.js使用的是事件驱动架构，这就是说在它上面做的每一件事，每一个独立的调用和操作，都是一系列的异步回调。这使得Node.js运行在一个线程上，这一点和其它的Web技术是不同的，它们每个客户端的请求都会生成一个新的线程来处理。这也正是非阻塞I/O的本质所在。

5.2.4 Gulp

Gulp.js 是一个自动化构建工具，你可以用它在项目中自动执行常见任务，

如：压缩优化JS、压缩CSS、压缩图片等。与Grunt相比，Gulp编写任务tasks更加简单，也更易于阅读和维护，插件高质而简洁。Gulp.js基于 Node.js 构建，利用 Node.js的Stream流，使项目可以快速构建并减少 IO 操作。

它拥有以下特性：

1. 易于使用

通过代码优于配置的策略，让简单的任务简单，复杂的任务可管理。

1. 构建快速

利用 Node.js 流的威力，你可以快速构建项目并减少频繁的 IO 操作。

1. 插件高质

Gulp 严格的插件指南确保插件如你期望的那样简洁高质得工作。

1. 易于学习

通过最少的 API，掌握 Gulp 毫不费力，构建工作尽在掌握：如同一系列流管道。

5.2.5 MySQL

MySQL是一个[关系型数据库管理系统](http://baike.baidu.com/view/1450387.htm" \t "http://baike.baidu.com/_blank)，由瑞典MySQL AB 公司开发，目前属于 [Oracle](http://baike.baidu.com/view/15020.htm" \t "http://baike.baidu.com/_blank) 旗下公司。MySQL 最流行的[关系型数据库管理系统](http://baike.baidu.com/view/1450387.htm" \t "http://baike.baidu.com/_blank)，在 WEB 应用方面MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。MySQL是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

本项目是基于Web技术的，采用MySQL，易于管理。

5.2.6 设计风格Material Design

Material design的核心思想，就是把物理世界的体验带进屏幕。去掉现实中的杂质和随机性，保留其最原始纯净的形态、空间关系、变化与过渡，配合虚拟世界的灵活特性，还原最贴近真实的体验，达到简洁与直观的效果。

通过对上述技术的学习，目前已经搭建了基本的开发环境，开发出程序的部分页面和后台逻辑。通过对项目的开发，兼顾前后端，培养了自己的编程与思考能力。运用软件工程的基本原理去分析和设计实现毕业设计的过程。并对系统的开发原理、功能设计和模块的设计做出系统的分析。将大学四年学到的理论知识综合起来，使掌握的知识更加牢固，更加具有趣味性。

1. 工程计划管控和执行情况

软件项目管理是一种科学的管理手段，它是为了使软件项目能够按照预定的成本、进度、质量顺利完成，而对成本、人员、进度、质量、风险等进行分析和管理的活动。

软件项目管理的范围不仅包括传统的软件开发过程，还应该包括开发前的准备工作以及运行中的维护工作和对项目的总结工作，因此具有极其重要的作用。所以，在课题开发的过程中，针对复杂工程问题制定安排计划，让项目计划按期执行。

6.1 实施计划

6.1.1 项目选用的生命周期

本项目采用的是瀑布式模型。此模型的本质是每个阶段的活动只做一次。从上一阶段向下一阶段逐级过度，最终得到所要开发的产品。

6.1.2 进度控制计划

采用在项目生命周期里程碑处对项目的规模、进度、质量、风险进行跟踪； 跟踪机制：

每阶段制定阶段详细计划；

每天记录项目工作日志；

每周汇总项目总工作量、评估项目实际进展以及关键路径、评估项目风险、关键依赖关系。

6.2 项目风险

在本项目中，存在以下的项目风险

|  |  |  |  |
| --- | --- | --- | --- |
| 风险编号 | 风险名称 | 风险等级 | 解决方案 |
| 1 | 课题项目规模太大，无法按期完成 | 中 | 除去无用功能的研发，缩减规模 |
| 2 | 资源不到位，导致无法按期开始研发 | 低 |  |
| 3 | 时间安排不合理 | 中 | 在课题开始时安排项目计划 |
| 4 | 为赶项目进度而导致质量问题 | 中 | 进行软件测试 |
| 5 | 因遇到难以解决的问题而中断开发 | 高 | 多问多查，解决问题 |

6.3 本章小结

管理在软件工程项目中的地位和作用与其它工程项目一样，是十分重要的。软件项目管理始终贯穿于软件生命周期的全过程。在软件开发过程中，我们只有理论联系实践，积极从实践中积累经验，注意项目管理中的关键因素，实施有效的管理，才能使软件项目获得成功。

本课题研发中，对项目的整个生命周期中实行管理计划，做好项目计划管理控制，已让项目能够如期研发完成。

第七章 职业素养与工程伦理的学习与培养

一个软件工程人才培养要注重“基础知识”、“技术”和“工程”三个方面．因此软件工程专业的大学生职业素养包括以下几个方面：

1. 软件基础知识结构能力．软件基础知识结构能力是软件工程专业的大学生对软件基础知识框架的掌握程度及其融会贯通的能力．这种能力是一种软能力，也是该专业最基础的能力，还是培养其他能力的土壤．软件工程专业的大学生都应该注重这种基础性能力的培养。大学生在学习过程要多思考，注意总结，把软件知识学得有条理、有脉络、有框架．这种能力的培养也为我们以后学习新知识打下坚实的基础．

2) 编程能力．编程能力是在一定的平台上利用编程语言编写出计算机能够运行的程序而使某一问题得到解决的能力．这种能力是软件专业大学生所必须具备的能力，因为它是该专业大学生素质的核心体现，是软件工程专业的重要培养目标之一，也是本专业大学生在就业市场上找到自己理想工作的保障．编程能力的培养要有一个过程，对于刚刚进入大学的大学生来说，学习编程语言时要多写程序、多调试程序，不断积累编程经验．编程既锻炼了自己操作计算机的能力，更重要的是锻炼了自己的思维能力和利用计算机解决问题的能力．这种能力是应用能力的核心体现，是软件工程专业大学生的本色，在自己就业时具有至关重要的作用，也关系到自己今后的发展。

3) 搜索能力。大学生是较高水平的知识分子，应该具有较强的知识搜索或查找能力。对于软件工程专业的大学生来说，不仅仅是指一般性知识的搜索和查阅，更多的是指与本专业相关的一些编程工具软件、插件、软件学习网站等的搜索与下载。当代社会是信息爆炸的社会，软件的种类繁多，软件技术的变化El新月异，作为软件工程专业的大学生应该掌握一定的检索技术，以满足自己学习和今后工作的需要。这种搜索能力是自己的充电器，会使自己终生受益。

4) 软件工程实践能力。软件工程实践能力是软件工程专业大学生所必须具备的特殊能力，这是由软件工程专业的特殊性决定的。软件工程专业的大学生是工程性人才，要具有工程性思维和运用工程方法去解决软件项目问题的能力。工程方法是一种比较科学规范、可减少错误、有步骤地解决问题的方法，把这种方法用到软件项目上可以借鉴以往的工程方法和工程经验，从而降低软件开发过程出错可能性、降低软件开发成本、提高软件开发效率。在IT企业中的开发团队也常常用这种方法来开发软件。因此，作为工程性人才也要拥有这种能力，为自己以后的工作奠定基础。

5) 团队协作与沟通能力。在企业中，软件开发人员往往是以团队的形式承接和开发某个工程项目的，团队是这类企业工作的基本单位。软件工程专业的大学生，毕业后往往在IT企业中工作，所以软件工程专业的大学生要具备这种若干个人在一起既分工又合作地完成一个相对庞大而又复杂工作过程的团队协作能力，才能适应工作的需要。团队协作能力较高的人才可以快速地融入团队，并和团队中其他成员一起解决问题。一个团队在开发项目的过程中，团队中的成员不可避免地会因为项目中的一些问题而进行彼此沟通，进而统一认识并找到解决问题的方法。

因此，一个合格的软件开发型人才，必须具有较高的团队协作与沟通能力。

在本课题的设计开发过程中，充分培养了我作为一名软件工程专业的职业素养，在遇到问题时，利用互联网搜索和查阅书籍，培养了我的搜索能力。在整个项目的设计与实现过程中，培养了我的软件基础知识结构能力。而在整个项目的编码过程中，我的编程能力得到了提升，大大加强了实践能力。

第八章 对软件工程实践以及软件工程领域发展的认识

软件工程是应用计算机科学理论和技术以及工程管理原则和方法,按照预算和进度, 实现满足用户要求的软件产品的定义、开发、发布和维护的工程或以之为研究对象的学科。 软件工程与其他工程一样要有自己的目标、活动和原则。

软件工程的基本目标是生产具有正确性、可用性及开销合宜(合算性)的产品.正确性意指软件产品达到预 期功能的程度;可用性意指软件基本结构、实现及文档达到用户可用的程度;开销合宜意指软件开发、运行的 整个开销满足用户的需求.以上目标的实现不论在理论上还是在实践中均存在很多问题有待解决,制约了对过 程、过程模型及工程方法的选取。

软件工程活动是“生产一个最终满足用户需求且达到工程目标的软件产品所需要的步骤”,主要包括需求、 设计、实现、确认以及支持等活动.需求活动是在一个抽象层上建立系统模型的活动,该活动的主要产品是需 求规约,是软件开发人员和客户之间契约的基础,是设计的基本输入.设计活动定义实现需求规约所需的结构, 该活动的主要产品包括软件体系结构、详细的处理算法等.实现活动是设计规约到代码转换的活动.验证/确认 是一项评估活动,贯穿于整个开发过程,包括动态分析和静态分析.主要技术有模型评审、代码“走查”以及程序 测试等.维护活动是软件发布之后所进行的修改,包括对发现错误的修正、对环境变化所进行的必要调整等. 围绕工程设计、工程支持以及工程管理,提出以下软件工程基本原则:

1. 原则是选取适宜的开发风范.以保证软件开发的可持续性,并使最终的软件产品满足客户的要求。
2. 原则是采用合适的设计方法.支持模块化、信息隐蔽、局部化、一致性、适应性、构造性、集成组 装性等问题的解决和实现,以达到软件工程的目标。
3. 原则是提供高质量的工程支持.提供必要的工程支持,例如配置管理、质量保证等工具和环境,以保 证按期交付高质量的软件产品。
4. 原则是有效的软件工程管理.仅当对软件过程实施有效管理时,才能实现有效的软件工程。

20世纪90年代以来，世界软件产业获得了飞速发展。据IDC统计，全球软件业的年均增长率一直保持在15%～20%之间。目前，全球软件业已经开始进入成熟期。产业分工较为明确，产业成熟度较高，成本已成为企业竞争的首要因素。发达国家的软件企业从降低成本考虑，逐步集中力量发展核心业务；利用全球的人力资源，将大量非核心业务向发展中国家转移。

近期国内信息化投资呈现加速趋势，软件产业整体发展较为平稳，得益于持续的政策支持和资金投入，软件服务化进程将不断加快，推动了软件产业结构继续调整优化。工业和信息化部软件服务司司长陈伟表示，目前我国软件产业创新能力逐步增强，大企业日益成为产业创新和规模发展的主导力量，依托自主核心产品和技术提供服务的软件和信息服务企业不断涌现

我个人的理解，软件工程就是按照工程学的管理方式，有组织、有计划的在成本限额以内按时完成开发且能在实际机器上可靠有效地运行的软件，并不断对软件进行优化、维护及管理的工作过程。

软件业作为一个服务业，要想得到发展，首先必须形成一个对软件服务有迫切需要的市场。其次，这个市场中的消费者必须具备足够的购买力。软件的消费群体可以分为个体消费和企业消费。

软件工程的技术和方法贯穿着系统科学的基本思想。软件工程中的一些重要方法，如结构化方法、面向对象方法等都沿用了系统科学的思想方法。软件工程的发展趋势要求在集成和发展现有理论和方法的基础上继续深入研究其支撑理论体系，并建立新的理论基础。从系统论和工程系统论的角度进行研究有助于软件工程重大理论和实践问题的突破。

现在已经是信息化时代，信息化潮流不断涌现，想要掌握主动权就是掌握信息化的发展方向，这就需要我们不断学习，时间，研究，学习国外的先进技术，转变自己的技术，然后融合，创新。

随着社会需要的增长，软件开发模式也遇到诸多困难，原先那些研发模式已经不在满足人们的需求，这叫我们研究开发新模式，现在的软件工程项目不再是一个人手工设计，而是需要一个研发小组，分工合作，研发出令人满意的软件！

大量事实表明，为了解决软件危机，既要有技术措施，又要有必要的组织管理措施。软件工程需要从管理和技术两方面研究如何科学、正确、合理地开发和维护软件，软件开发必须遵循软件工程的思想不断精细化。

第九章 结束语

9.1 本文内容

本文主要讨论了如何从软件工程的角度设计和实现一个基于Web技术的电影交流分享桌面应用，从系统的用户需求分析，系统性能需求分析到系统概要设计，详细设计。如何解决在开发中遇到的复杂工程问题。并进行最后的软件测试，确保软件不会出现大的错误。

9.2 顶岗实习项目课题有待进一步解决的问题及方向

现在网站资源还很少，大部分电影都没有，很多资源都要实时从豆瓣获取，这无疑会拖慢网站的速度，而豆瓣电影却由很强的反爬虫策略。我所使用的设置间隔时间爬虫效率不够高效。后期可以使用代理来抓，通过爬取网上的免费代理，然后随机使用代理群组中的代理进行并发爬取，并改变请求头，以防止被豆瓣电影拒绝访问。

9.3 对软件工程实践以及软件工程领域发展的认识

软件在当今的信息社会中占有重要的地位，软件产业是信息社会的支柱产业之一。随着软件应用日益广泛、软件规模日益扩大，人们开发、使用、维护软件不得不采用工程的方法，以求经济有效地解决软件问题。今天的软件工程己由最初的一个学科方向发展成为以计算机科学技术为基础的一个新兴交叉学科。

软件工程随着信息化时代的到来其地位越来越受到人们的重视，软件工程从一个学科，或是某一个研究方向来说，人员仅仅是过程，方法的执行者，所以人员素质往往被忽略，软件工程是一门实践性很强的学科，所以在实际的软件研究过程中，人员的素质占有很重要的地位。要有出色的软件问世，研发人员的素质至关重要！

作为软件工程的学习者应该不断创新，不断去尝试，不断去实践，不断的研究和学习，中国的软件工程技术依旧滞后于国外一些软件工程技术，作为新一代的学习者应该担当起振兴起中国软件事业，使中国科技得到高速发展！

9.4 本人毕业设计（顶岗实习）收获及体会

不知不觉，实习就结束了。在这半年的时间里，我真的成长了很多，学到了很多知识，认识了很多新的小伙伴，我很感谢大家对我的照顾。

虽然说实习上班的生活基本上是每天两点一线，从住的地方到公司，下班后再回来。刚开始的时候我对实习生活是充满期待的，对所有东西对充满了干劲。学习过程中发现自己真的差的远着呢，很多东西都不知道，越学越感觉自己的无知，真的，现在的web前端开发技术真的更新的很快，成长的很快，过去的JS只能写浏览器脚本，实现一些网页特效，现在呢，世界上的大牛程序员将JS开发的能写前台，还能写后端（node)，还能写安卓APP和IOS的APP以及Windows桌面程序，真的期待JS将来还能干什么。对我们学计算机，写代码的人来说，技术就是自己的饭碗，实践非常重要，而且要跟上时代脚步，不要止步不前，新的技术如果不学习，会被时代淘汰。要善于吸取当下流行技术的精华，累积经验，这样以后才能做到架构师等高级别的工程师。

沟通很重要，因为你要和产品经理，后端开发人员，项目负责人进行沟通，提出疑问，解决问题。沟通不好很容易出问题，要善于表达自己的意思。

在实习的日子里，我真的体验到了什么叫年轻人的激情，在赶项目的时候，大家都充满干劲，只为让用户用到最新的东西，自觉的留下来加班。

实习的小半年日子里，让我得到锻炼，得到提升，充分的将所学进行实践。

参考文献

[1] 陈志泊.数据库原理及应用教程（第二版）[M].北京：人民邮电出版社.2008

[2] 刘晓强.信息系统与数据库技术[M].北京：机械工业出版社.2008

[3] 陆昌辉.吴晓华.MySql核心技术揭秘[M].北京：北京希望电子出版社.2002

[4] 黄伟.基于MVC架构的Web应用系统设计[J].微型机与应用.2004.11.13-15

[5] 陈瑶. 基于Node.js高并发web系统的研究与应用[D].电子科技大学,2014.

[6] [Mike Cantelon](https://book.douban.com/search/Mike Cantelon) ，[TJ Holowaychuk](https://book.douban.com/search/TJ Holowaychuk) ， [Nathan Rajlich](https://book.douban.com/search/Nathan Rajlich)  Node.js In Action [M] 人民邮电出版社 2015-9-1

[7] [Pawel Kozlowski](https://book.douban.com/search/Pawel Kozlowski) ，[Peter Bacon Darwin](https://book.douban.com/search/Peter Bacon Darwin) [李路](https://book.douban.com/search/%E6%9D%8E%E8%B7%AF)，[王永强](https://book.douban.com/search/%E7%8E%8B%E6%B0%B8%E5%BC%BA) ， [马海波](https://book.douban.com/search/%E9%A9%AC%E6%B5%B7%E6%B3%A2) 精通AngularJs[M]. 华中科技大学出版社 北京,2014,11-1

[8] [美]Jon Bentley编著，《编程珠玑》[M]， 人民邮电出版社，2008

[9] 《编程之美》小组编著，《编程之美》[M]， 电子工业出版社，2008

[10] 子柳编著，《淘宝技术这十年》[M]， 电子工业出版社，2013

[11] 郭欣编著，《构建高性能Web站点》[M]， 电子工业出版社，2009

[12] [美][Cal Henderson](http://book.douban.com/search/Cal Henderson) 编著，《构建可扩展的Web站点》[M]， 电子工业出版社，2008

[13] [美][Eric S. Raymond](http://book.douban.com/search/Eric S. Raymond) 编著，《UNIX编程艺术》[M]， 电子工业出版社，2006

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

[15] Damon Armstrong:Pro ASP.NET 2.0 Website Programming,POSTS&TELECOM PRESS,200.2

[7] John Duckett: Beginninng web programming with HTML, XHTML and CSS,2010.1

[8] Thomas Erl :Web Service contract design and versioning for SOA, POSTS&TELECOM PRESS, 2010.01

致谢

四年的大学生活就快走入尾声，伴随着这篇毕业论文的落笔，我四年的本科大学生活也即将画上了句号。回忆这大学四年生活的滴滴点点，美好的大学生活的过往，一切都是历历在目，让人非常留恋，思绪万千。

非常感谢我的同学们。当我在毕业设计过程中遇到困难和问题时，他们给我提出非常多的建议和意见，使我对整个系统有了总体的思路，并耐心的帮我解决了很多技术上的难题，让我毕设的过程轻松了不少。

最后，我要特别感谢我的企业导师徐鸿林和代管老师李美蓉。他们在我们毕业的最后关头给了我重要的鼓励和帮助，从课题的选定到开题报告，任务书，整个系统的完成，论文的编写，格式以及内容的检测，老师全程参与进来，并提出了重要的意见和建议，李美蓉老师严谨的科研精神和认真负责的工作态度都给我树立了正确的榜样，在此表示衷心的感谢。

这是第一次写真正意义上的论文，马上要踏入企业工作了，希望这篇论文不会是学术思考的终点。

外文资料原文

A Web API Design Methodology

Designing, implementing, and maintaining APIs for the Web is more than a challenge; for many companies, it is an imperative. [This series](http://www.infoq.com/articles/Web-APIs-From-Start-to-Finish) takes the reader on a journey from determining the business case for APIs to a design methodology, meeting implementation challenges, and taking the long view on maintaining public APIs on the Web over time. Along the way there are interviews with influential individuals and even a suggested reading list on APIs and related topics.

Designing Web APIs is more than just URLs, HTTP status codes, headers, and payloads. The process of design -- what is essentially a "look and feel" for your API -- is very important and is well-worth the effort. This article briefly outlines a methodology that results in an API design that takes advantage of both HTTP and the Web. And it can work for more than just HTTP. If, at some point you need to implement the same service over WebSockets, XMPP, MQTT, etc. most of the features of the resulting API design will work the same. That can make supporting multiple protocols in the future easier to implement and maintain.

Good Design Goes Beyond URLs, Status Codes, Headers, and Payloads

Typically, Web API design guidance focuses on the the common features such as URL design, proper use of HTTP features such as status codes, methods, headers, and the design of payloads that hold serialized objects or object graphs. These are valuable implementation details, but not much in the way of API design. And it is the design of the API -- the way the essential features of the service are expressed and described -- that can make an important contribution to the success and usability of your Web API.

A good design process or methodology defines a consistent, repeatable set of steps to employ when working to expose a server-side service component as an accessible, usable Web API. That means that a clear methodology can be shared with developers, designers, and software architects in order to help coordinate activities throughout the implementation cycle. An established methodology can also be refined over time as each team discovers ways to improve and streamline the process without adversely affecting implementation details. In fact,implementation details can change (e.g. which platform, OS, frameworks, and UI style to employ) independently of the design process when these two are cleanly separated and defined.

A Seven Step API Design Methodology

What follows is a brief overview of the design methodology covered in the book "[RESTful Web APIs](http://restfulwebapis.com/)" by Richardson and Amundsen. There is not enough room here to go into depth for each step in the process but this article can give the big picture. Also, the reader can use this overview as a guide for developing a unique Web API design process that fits the local skills and goals of your group.

NOTE: Yes, seven seems like quite a few. In reality there are only five design steps and two additional items in the list including implementing and publishing. These last two round out the process to provide and end-to-end experience.

You should plan to reiterate through these steps as needed. You may get through Step 2 (Draw State Diagrams) and realize there is more work to be done in Step 1 (List All the Parts). When you get around to writing the code (Step 6), you may discover a number of things missed in Step 5 (Create a Semantic Profile), etc. The key is to use the process to expose as many details as possible and be willing to go back a step or two in order to capture the items you missed along the way. Iteration is the key to building a more complete picture of your service and clarifying how it can be exposed to client applications.

Step 1 : List All the Parts

The first step is to list all the pieces of data a client application might want to get out of the service or put into it. We'll call these the semantic descriptors. Semantic because they deal with the meaning of data in the application and descriptors since they describe what is happening in the application itself. Note that the point of view here is that of the client, not the service. It's important to design the API as something the client will be using.

Step 2 : Draw State Diagrams

The next step is to draw out state diagrams for the proposed API. Each box in the diagram represents a possible representation -- a document that includes one or more of the semantic descriptors identified in Step 1. You can use arrows to indicate transitions from one box to the next -- from one state to the next. These transitions are initiated by protocol requests.

Step 3 : Reconcile Magic Strings

The next step is to reconcile all the "magic strings" in your service interface. The "magic strings" are all the descriptor names -- they have no intrinsic meaning, they just represent actions or data elements clients will access when communicating with your service. Reconciling these descriptor names means adopting well-known public names from sources like:

* [Schema.org](http://schema.org/)
* [microformats.org](http://microformats.org/)
* [Dublin Core](http://dublincore.org/documents/dces/)
* [IANA Link Relation Values](http://www.iana.org/assignments/link-relations/link-relations.xhtml)

Step 4 : Select a Media Type

The next step in the design process for your API is to select a media type to use when passing messages back and forth between client and server. One of the hallmarks of the Web is that data is passed as standardized documents over a uniform interface. It is important to select a media type that supports both the data descriptors (e.g. "identifier", "status", etc.) as well as the action descriptors (e.g. "search", "edit", etc.).

Step 5 : Create a Semantic Profile

A Semantic Profile is a document that lists all the descriptors in your design and includes details about each one that will help developers when building both client and server implementations. The profile is an implementation guide, not an implementation description. This is an important distinction.

Step 6 : Write Some Code

At this point, you should be able to turn over your design documents (state charts and semantic profile) to developers of both the server and client apps in order to start a specific implementation.

The HTTP server should implement the state diagram created in Step 2 and requests from the client should trigger the proper state transitions in the service. Each representation sent from the service should be in the format selected in Step 3 and should include a link to the profile created in Step 4. Responses should include the appropriate hypermedia controls that implement the actions shown in the state chart and described in the profile document. Client and server developers can build their implementations relatively independently at this point and use test runs to validate compliance with the state diagram and profile.

Step 7 : Publish Your API

Web APIs should publish at least one URL that is promised to always respond to clients -- even in the far future. I call this the "billboard URL" -- the one everyone knows. It is also a good idea to publish the profile document so that new implementations of the service can link to it in responses. You can also publish human-readable documentation, tutorials, etc. to help developers understand and use your service.

Once that is done, you should have a well-designed, stable, accessible service up and running, ready to use.

Hopefully this gives you some ideas on how you can create an optimal API design methodology for your organization and/or team.

外文资料译文

Web API设计方法论

为Web设计、实现和维护API不仅仅是一项挑战；对很多公司来说，这是一项势在必行的任务。[本系列](http://www.infoq.com/articles/web-apis-from-start-to-finish) 将带领读者走过一段旅程，从为API确定业务用例到设计方法论，解决实现难题，并从长远的角度看待在Web上维护公共API。沿途将会有对有影响力的人物的访谈，甚至还有API及相关主题的推荐阅读清单。

设计Web API不止是URL、HTTP状态码、头信息和有效负载。设计的过程--基本上是为了你的API“观察和感受” -- 这非常重要，并且值得你付出努力。本文简要概括了一种同时发挥HTTP和Web两者优势的API设计方法论。并且它不仅对HTTP有效。如果有时你还需要通过WebSockets、XMPP、MQTT等实现同样的服务，大部分API设计的结果同样可用。可以让未来支持多种协议更容易实现和维护。

优秀的设计超越了URL、状态码、头信息和有效负载

一般来说， Web API设计指南的重点是通用的功能特性，比如URL设计，正确使用状态码、方法、头信息之类的HTTP功能特性，以及持有序列化的对象或对象图的有效负载设计。这些都是重要的实现细节，但不太算得上API设计。并且正是API的设计--服务的基本功能特性的表达和描述方式--为Web API的成功和可用性做出了重要贡献。

一个优秀的设计过程或方法论定义了一组一致的、可重复的步骤集，可以在将一个服务器端服务组件输出为一个可访问的、有用的Web API时使用。那就是说，一个清晰的方法论可以由开发人员、设计师和软件架构师共享，以便在整个实现周期内帮助大家协同活动。一个成熟的方法论还可以随着时间的发展，随着每个团队不断发现改善和精简过程的方式而得到精炼，却不会对实现细节产生不利的影响。实际上，当实现细节和设计过程两者都有清晰的定义并相互分离时，实现细节的改变（比如采用哪个平台、OS、框架和UI样式）可以独立于设计过程。

API设计七步法

接下来我们要对Richardson和Amundsen合著的《[REST风格的Web API](http://restfulwebapis.com/)》一书中所介绍的设计方法论做简要地概述。因为篇幅所限，我们不能深入探讨这一过程中的每一步骤，但这篇文章可以让你有个大概的认识。另外，读者可以用这篇概述作为指南，根据自己组织的技能和目标开发一个独有的Web API设计过程。

说明：是的，7步看起来有点儿多。实际上清单中有5个步骤属于设计，额外还有两个条目是实现和发布。最后这两个设计过程之外的步骤是为了提供一个从头到尾的体验。

你应该计划好根据需要重新迭代这些步骤。通过步骤2（绘制状态图）意识到在步骤1（列出所有组成部分）有更多工作要做。当你接近于写代码（步骤6）时，可能会发现第5步（创建语义档案）中漏了一些东西。关键是用这个过程暴露尽可能多的细节，并愿意回退一步或者两步，把前面漏掉的补上。迭代是构建更加完整的服务画面以及澄清如何将它暴露给客户端程序的关键。

步骤1 : 列出所有组成部分

第一步是列出客户端程序可能要从我们的服务中获取的，或要放到我们的服务中的所有数据片段。我们将这些称为语义描述符。语义是指它们处理数据在应用程序中的含义，描述符是指它们描述了在应用程序自身中发生了什么。注意，这里的视点是客户端，不是服务器端。将API设计成客户端使用的东西很重要。

步骤2 : 绘制状态图

下一步是根据建议的API绘制出状态图。图中的每个框都表示一种可能的表示--一个包含在步骤1中确定的一或多个语义描述符的文档。你可以用箭头表示从一个框到下一个的转变--从一个状态到下一个状态。这些转变是由协议请求触发的。

步骤 3 : 调和魔法字符串

下一步是调和服务接口中的所有“魔法字符串”。“魔法字符串” 全是描述符的名称--它们没有内在的含义，只是表示客户端跟你的服务通讯时将要访问的动作或数据元素。调和这些描述符名称的意思是指采用源自下面这些地方的，知名度更高的公共名称：

* [Schema.org](http://schema.org/)
* [microformats.org](http://microformats.org/)
* [Dublin Core](http://dublincore.org/documents/dces/)
* [IANA Link Relation Values](http://www.iana.org/assignments/link-relations/link-relations.xhtml)

步骤 4 : 选一个媒体类型

API设计过程的下一步是选一个媒体类型，用来在客户端和服务器端之间传递消息。Web的特点之一是数据是通过统一的接口作为标准化文档传输的。选择同时支持数据描述符（比如"identifier"、"status"等）和动作描述符（比如"search"、"edit"等）的媒体类型很重要。

步骤 5 : 创建语义档案

语义档案是一个文档，其中列出了设计中的所有描述符，包括对开发人员构建客户端和服务器端实现有帮助的所有细节。这个档案是一个实现指南，不是实现描述。这个差别很重要。

步骤 6 : 写代码

到了这一步，你应该可以将设计文档（状态图和语义档案）交给服务器和客户端程序的开发人员了，让他们开始做具体的实现。

HTTP服务器应该实现在第2步中创建的状态图，并且来自客户端的请求应该触发正确的状态转变。服务发送的每个表示都应该用第3步中选好的格式，并且应该包含一个第4步中创建的指向一个档案的链接。响应中应该包含相应的超媒体控件，实现了在状态图中显示、并在档案文档中描述的动作。客户端和服务器端开发人员在这时可以创建相对独立的实现，并用测试验证其是否遵守了状态图和档案。

有了稳定的可运行代码，还有一步要做：发布。

步骤 7 : 发布你的API

Web API应该至少发布一个总能给客户端响应的URL -- 即便是在遥远的将来。我将其称为“看板URL” --每个人都知道的。发布档案文档也是个好主意，服务的新实现可以在响应中链接它。你还可以发布人类可读的文档、教程等，以帮助开发人员理解和使用你的服务。

做好这个之后，你应该有了一个设计良好的、稳定的、可访问的服务运行起来了，随时可以用。

在你的设计过程中，你可能会发现有遗漏的元素，需要重做某些步骤，以及要做一些折中的决定。这在设计过程中出现得越早越好。将来开发人员要求用新的格式和协议实现时，你还有可能用这个API设计。

希望这篇文章能给你一些启发，让你知道如何给自己的组织以及/或者团队创建一个最佳的API设计方法论。