**学生承诺书**  
 本人郑重声明：所呈交的毕业论文《在线生活百科系统 的设计和开发》，是本人在指导老师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果

由本人承担。

**签名：\_\_\_\_\_\_\_\_\_\_\_\_\_**

摘 要

本文以生活中需要的常用信息（图书，音乐，电影）为研究对象，设计开发了在线生活百科系统，有效的实现了信息的查询。本设计还可以针对登录后的用户可以使用收藏功能。

本设计主要使用React+Redux+Webpack+Babel+Bootrap技术实现前端页面与逻辑的构建。该系统核心分为两部分内容：一，三大模块（图书，音乐，电影）的信息显示，具体条目的细节显示。二，查询模块，根据关键字查询需要的内容。

关键词：生活百科 前端 React Redux Webpack

目 录

[1 前言 1](#_Toc482808694)

[1.1 课题设计的背景、意义及目的 1](#_Toc482808695)

[1.1.1 课题设计的背景 1](#_Toc482808696)

[1.1.2 课题设计的意义及目的 1](#_Toc482808697)

[1.2 国内研究概况 1](#_Toc482808698)

[2 可行性分析 2](#_Toc482808699)

[2.1 社会可行性 2](#_Toc482808700)

[2.2 技术可行性 2](#_Toc482808701)

[2.3 操作可行性 2](#_Toc482808702)

[3 系统技术介绍及环境搭建 3](#_Toc482808703)

[3.1　系统技术介绍 3](#_Toc482808704)

[3.1.1 JavaScript 3](#_Toc482808705)

[3.1.2 JQuery 3](#_Toc482808706)

[3.1.3 Bootstrap 3](#_Toc482808707)

[3.1.4 NodeJS 3](#_Toc482808708)

[3.1.5 Webpack 4](#_Toc482808709)

[3.1.6 Babel 4](#_Toc482808710)

[3.1.7 React 4](#_Toc482808711)

[3.1.8 React-Router 4](#_Toc482808712)

[3.1.9 Redux 4](#_Toc482808713)

[3.2 系统开发平台及运行环境 5](#_Toc482808714)

[3.2.1 系统开发平台 5](#_Toc482808715)

[3.2.2 系统运行环境 5](#_Toc482808716)

[4 需求分析 5](#_Toc482808717)

[4.1 系统功能模块概述 5](#_Toc482808718)

[4.2 系统功能模块设计 5](#_Toc482808719)

[4.3 数据库结构展示与分析 6](#_Toc482808720)

[4.3.1 数据库结构展示 6](#_Toc482808721)

[4.3.2 数据库分析 8](#_Toc482808722)

[5 普通部分功能设计与实现 9](#_Toc482808723)

[5.1 图书，音乐，电影模块设计 9](#_Toc482808724)

[5.2 搜索模块 10](#_Toc482808725)

[5.2.1 搜索信息展示 10](#_Toc482808726)

[5.2.2 条目详细信息 11](#_Toc482808727)

[5.3 主页与脚注部分 11](#_Toc482808728)

[5.3.1 主页 11](#_Toc482808729)

[5.3.3 主页天气功能 12](#_Toc482808730)

[5.3.4 脚注 13](#_Toc482808731)

[5.4 下拉加载更多部分 14](#_Toc482808732)

[6 登录模块设计与实现 16](#_Toc482808733)

[6.1 登录功能展示 16](#_Toc482808734)

[6.2 注册功能展示 17](#_Toc482808735)

[6.3 收藏模块功能展示 18](#_Toc482808736)

[7 Windows系统上环境的搭建 19](#_Toc482808737)

[7.1 环境搭建的相关条件 19](#_Toc482808738)

[7.2 环境搭建的相关步骤 19](#_Toc482808739)

[8 系统测试 20](#_Toc482808740)

[8.1 测试目的 20](#_Toc482808741)

[8.2 测试方法 20](#_Toc482808742)

[8.3 测试用例 21](#_Toc482808743)

[8.4 测试结论 25](#_Toc482808744)

[9 项目管理 26](#_Toc482808745)

[9.1 项目管理重要性 26](#_Toc482808746)

[9.2 Git 和 GitHub 介绍 26](#_Toc482808747)

[9.2.1 Git 26](#_Toc482808748)

[9.2.2 GitHub 26](#_Toc482808749)

[9.3 代码仓库 27](#_Toc482808750)

[9.3.1 项目在GitHub上面托管的地址 27](#_Toc482808751)

[9.3.2 项目在GitHub上面代理的地址 27](#_Toc482808752)

[9.3 项目开发版本 27](#_Toc482808753)

[10 总结 29](#_Toc482808754)

[参 考 文 献 30](#_Toc482808755)

[致 谢 32](#_Toc482808756)

[仲恺农业工程学院毕业设计成绩评定表 33](#_Toc482808757)

1 前言

1.1 课题设计的背景、意义及目的

1.1.1 课题设计的背景

90年代的音乐黄金时代过去后，华语乐坛就一蹶不振，究其原因并不都是因为互联网时代的到来，而是由于中国大陆对于版权的保护制度还有实施的措施不到位，完全更不上技术的发展而导致的。而且，由于历史原因，国人一直都有“免费听歌”的思想惯性。不过，2015年以后，国内的音乐产业迎来了生机，各大音乐平台进行的“版权之战”，加上不断完善的版权制度，在互联网的大背景之下，音乐产业进行着革新，例如：付费专辑，现场的演出直播，网络平台直播等。都大大提高了音乐人的生存环境。

近几年国内发展迅速，电影市场在未来也将保持快速发展。单单2015年期间国内的票房收入就总共达到440多亿元，共有12多亿次观影记录。不过，中国电影票房规模总量虽大，但人均观影次数还是达不到一次，所以国内电影市场发展潜力还是非常的大。

1.1.2 课题设计的意义及目的

随着人们生活质量水平提高，娱乐产业也越来越得到重视。例如：音乐，电影等越来越流行。人们不再只是单纯的为了听而听，看而看。而是挑着看，挑有质量的听。人们对于产品的质量要求越来越高，但同时找不到一个很好了解这些东西的渠道。如果有一个专门提供一些生活中经常需要的东西（图书，音乐和电影等信息）的网站，人们就可以很方便的得到这方面的信息。

1.2 国内研究概况

近年来，随着互联网的快速发展，微博、微信等开始成为图书，音乐与电影营销的新渠道，传统的营销模式并不能满足我们，以互联网为核心的新时代媒体营销模式兴起，给正在迅速发展的中国市场注入了新的血液。作为媒体的一股清泉，新媒体使音乐和电影营销的手段更加多变，将会完全改变既定的营销方式，呈现出新技术下的新气象。一批以点石娱乐为代表的新媒体营销队伍，将帮助中国向着市场化的进程迈进。

2 可行性分析

2.1 社会可行性

生活百科系统设计是以实时便捷的在线查询为设计思路。在日常生中，人们随时都可能需要查询相关的信息。无论是在家里，路上还是其他地方。只要想查询，就很容易的通过本网站查询到。考虑到移动端的庞大流量，本网站还采取了响应式的开发，兼容不同屏幕的显示问题。完美解决移动端的小屏幕显示问题。

2.2 技术可行性

生活百科系统设计的大部分是前端的技术。采用SPA（单页面应用程序）大大提到了用户体验。数据都是采用JSON进行传输。系统的主要由：React+Redux+Webpack+Babel+Bootstrap 构建的。React完美释放DOM操作，所以本网站只涉及很少的直接DOM操作（DOM操作交给Jquery）。而且使用Redux全局统一管理数据（例如当前登录用户）。使用Webpack一体化的构建项目，解决文件的依赖，合并，代码实时更新等大多数问题。还有结合Babel翻译ECMA2015语法规范，使开发更加便捷。同时使用Bootstrap作为前端页面UI框架。很大程度减少了重复造轮子的问题。

网站结合MVC思想，React负责渲染View层，Redux（controller）负责管理module。

2.3 操作可行性

生活百科系统是基于Bomb第三方作为后台服务器的。所以不用担心服务器搭建问题，只要有网络就可以使用。而且由于采用的是浏览器/服务器结构开发。只要终端上有浏览器应用，就可以随时随地使用。而且大部分功能不需要登录即可使用。有效提高了用户的体验。

3 系统技术介绍及环境搭建

3.1　系统技术介绍

3.1.1 JavaScript

JavaScript，一个专门用于浏览器的动态脚本语言。它不需要严格的数据结构定义，支持面向对象（函数）和函数式编程。近几年由于HTML5的出现，前端无疑是最火的，作为前端核心的JavaScript，也出现了非常多有用的框架，诸如：Angular，Vue，React。不过自从NodeJS出现以后，使得JavaScript不再仅限于操作浏览器，现在也能当做一门后台语言作为构建服务器的语言了。

3.1.2 JQuery

Jquery是前端发展前期最著名的类库。它能够兼容不同的浏览器，便捷的选择器方法，简单的操作DOM节点，绑定事件等特色功能。它还包装了Ajax请求方法，不用我们自己封装。不过随着前端的快速发展，在JS代码中操作DOM已经不提倡，所以渐渐的现在Jquery的使用数已大大降低，本系统之所以还使用Jquery是因为Bootstrap的JS文件是基于Jquery的，还有少部分不可避免的DOM操作

3.1.3 Bootstrap

之所以采用Bootstrap，主要是因为Bootstrap提供了一整套UI样式还有自定义组件。免去了自己写HTML结构还有CSS样式。最重要的是Bootstrap是响应式开发的。可以完美解决各种终端不同大小的屏幕显示问题

3.1.4 NodeJS

大前端在近几年的快速发展，离不开NodeJS。NodeJS是一个JavaScript的运行环境。自从NodeJS出现后，前端项目构建就离不开步它。诸如：npm，bower，gulp，webpack，grunt等工具都是基于NodeJS的

3.1.5 Webpack

Webpack是当前最流行的前端构建工具了。它具有gulp的大部分功能，诸如代码压缩，自动更新，解析less/sass文件等功能，也同时拥有自己强大的功能：合并文件，打包第三方库等特色功能

3.1.6 Babel

老版本的JavaScript虽然说也是支持面向对象的，但底层实质还是面向函数编程。但是在2015年发布了ECMA2015（es6，JavaScript标准）以后，JavaScript拥有了很多新功能，诸如：1.同步方式解决异步API调用。2.新增定义变量关键字let，const。3.数据结构的解构等等。大大提高了JavaScript开发效率。不过话说如此，新的东西还是没那么快普及，大部分浏览器都只能部分支持es6新特性，所以需要一个翻译器，这就是Babel。Babel不仅可以实时翻译新语法，还支持先翻译在运行。

3.1.7 React

本系统的核心技术，React是一个声明式的前端框架，来源于Facebook的内部项目。React的思想是网站是一个大组件，由一个个小的组件组合起来的。所以React中最小单元就是组件，通过单向数据绑定，可以实时更新数据和视图。由于React采用虚拟的DOM，只有在数据发生变化时，React只会渲染发生变化的部分。所以渲染效率极高

3.1.8 React-Router

本系统是一个SPA（单页面应用程序），关键点无异于路由部分。本系统使用React-Router解决路由管理问题。整个应用都处于不更新状态，只通过路由的跳转选择渲染不同的组件。提高用户体验。

3.1.9 Redux

React与其他两大框架Angular，Vue对比来说，React只用于页面的渲染，就是View层。Module层是做不了的。所以需要结合Redux，Redux是一个应用数据流框架，最主要作用是应用的状态管理。本系统采用了与React结合更好的React-Redux

3.2 系统开发平台及运行环境

3.2.1 系统开发平台

本系统后台基于Bmob这个第三方网站解决。能使我更加专注于前端的开发。还有部分API来源于豆瓣的API公共接口。

3.2.2 系统运行环境

操作系统Windows，大部分主流手机系统。

浏览器版本要求：IE 9.0+，Firefox 3.5+，Opera 10.5+，Chrome 5.0+，Safari 3.0+

4 需求分析

4.1 系统功能模块概述

生活百科系统主要分为两大部分：非登录部分和登录部分，非登录部分主要有图书，音乐，电影和搜索模块，登录部分主要有注册/登录模块，收藏模块。

4.2 系统功能模块设计

系统两大功能部分：普通部分和登录部分

普通部分具体结构分析：不登录就可以使用的模块

普通部分

天气模块

搜索模块

电影模块

音乐模块

图书模块

搜索天气

搜索列表模块

图书分类信息

详细信息

电影分类信息

音乐分类信息息

图4-1 普通部分具体结构分析

登录部分具体结构分析：登录模块，注册模块，添加收藏模块

登录部分

收藏模块

注册模块

登录模块

注册新用户

用户收藏

判断当前用户

进行用户登录

图4-2 登录部分具体结构分析

4.3 数据库结构展示与分析

4.3.1 数据库结构展示

一个应用程序或者一个网站系统，离不开大量的数据交互，这时就会涉及到数据储存问题，储存数据的方式无异于几种：1，保存在内存中，优点是不占用硬盘空间，缺点是一旦程序关闭数据就会被销毁。2，保存在文件中，优点是能将数据持久化，缺点是当数据一旦复杂，可维护性将大大降低。3，数据库，数据库是目前来说最流行的存储数据办法。它具有读写快，能讲数据持久化，数据关系话等有点。本系统的第三方后台Bmob也是采用数据库的形式保存数据。

生活百科系统设计与后台做交互的数据不多，所以用到的表也不多。分别是：用户表，用户图书收藏表，用户音乐收藏表，用户电影收藏表。

表4-1 用户表(User)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Extra | Comment |
| objectId | varchar(50) | NO | PRI |  | Bmob自动生成的ID |
| \_username | varchar(50) | NO |  |  | \_用户名 |
| \_password | varchar(50) | NO |  |  | \_密码 |
| \_email | varchar(50) | NO |  |  | \_邮箱 |
| createAt | Date | NO |  |  | 创建时间 |
| updateAt | Date | NO |  |  | 修改时间 |

表4-2 用户图书收藏表(BookCollection)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Extra | Comment |
| objectId | varchar(50) | NO | PRI |  | Bmob自动生成的ID |
| bookId | int | NO | 外键 |  | 图书id |
| \_userId | varchar(50) | NO | 外键 |  | \_用户id |
| createAt | Date | NO |  |  | 创建时间 |
| updateAt | Date | NO |  |  | 修改时间 |

表4-3 用户音乐收藏表(MusicCollection)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Extra | Comment |
| objectId | varchar(50) | NO | PRI |  | Bmob自动生成的ID |
| musicId | int | NO | 外键 |  | 图书id |
| \_userId | varchar(50) | NO | 外键 |  | \_用户id |
| createAt | Date | NO |  |  | 创建时间 |
| updateAt | Date | NO |  |  | 修改时间 |

表4-4 用户电影收藏表(FilmCollection)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Extra | Comment |
| objectId | varchar(50) | NO | PRI |  | Bmob自动生成的ID |
| filmId | int | NO | 外键 |  | 图书id |
| \_userId | varchar(50) | NO | 外键 |  | \_用户id |
| createAt | Date | NO |  |  | 创建时间 |
| updateAt | Date | NO |  |  | 修改时间 |

表4-5 轮播图表(FilmCollection)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Extra | Comment |
| objectId | varchar(50) | NO | PRI |  | Bmob自动生成的ID |
| url | varchar(50) | NO |  |  | 图片地址 |
| link | varchar(50) | NO |  |  | 链接地址 |

4.3.2 数据库分析

生活百科系统通过后台数据库查找到相应的数据。对于数据库中数据的查询，增加，删除，修改都需要操作对应的某个具体表。当需要查询的数据不存在或者表中没有对应字段时，都将得不到数据。

每一个系统中，关系型数据库的表间关系都非常重要。它表示了表与表之间的关联关系。通过外键就可以把两个表链接起来。

首先，用户的收藏从用户登录🡪搜索信息🡪收藏信息以及环节中出现的问题如图4-1

用户登录

搜索信息

收藏响应数据

问题汇总

图4-1 用户收藏图

整个收藏的环节中，用户需要先登录成功后，然后搜索想要的信息，在收藏数据时，会先判断当前是否有用户已成功登录。如果有当前用户则允许当前用户收藏该信息。不然就会提示请登录。添加后的想要收藏数据拥有一个信息的标识（条目id），还有一个用户的id，两个字段都是外键，从而可以得到那个用户一共收藏了那些条目信息。

5 普通部分功能设计与实现

系统只有一个index主页面，其中只有一个用于渲染组件的根节点。还有引入样式文件和脚本文件。通过脚本文件，可以快速的将需要呈现的HTML结构渲染到页面上。

5.1 图书，音乐，电影模块设计

三个模块的组件组成十分相像，模块都具有一个子导航，可以导航到不同的分类页面。然后有一个内容区域，内容区域用于渲染条目列表的信息。都拥有一个上拉加载更多的组件，当滚动到页面底部时将会自动加载更多信息。然后还有一个脚注和回到顶部组件。当点击某一个条目时，将会导航到详细页面，详细页面能够更好的展示该条目的具体信息。还可以点击收藏按钮收藏条目。电影模块页面如图5-1



图5-1 电影模块显示

头部有一个不会重复渲染的主导航栏，可以导航到每个子模块。还有一个搜索框，可以进行搜索。回车进入搜索模块。

当进入子模块后，就会如图5-1一样，每个子模块都有自己的子导航。可以去到不同的分类。中间的就是内容展示区域。该区域会根据路由的改变决定渲染的数据。但HTML结构都是类似的。

右边就有一个脚注区域和回到顶部的组件。

5.2 搜索模块

搜索模块是系统的核心部分。该模块允许用户搜索想要的信息。当按下回车后，就会进入搜索模块，如图5-2。页面将有一个大的选项卡组件，分别存放图书，音乐和电影信息。

5.2.1 搜索信息展示

如图5-2，我们搜索的关键子“刘德华”，这是时会请求数据，在获取数据成功后，将会进行页面的渲染。默认我们找到了图书，音乐，电影各10各条目信息。他们以瀑布流的形式展示在页面上。每个条目将会展示一部分信息，如果需要查看更多信息则需要点击条目进入详细页面。

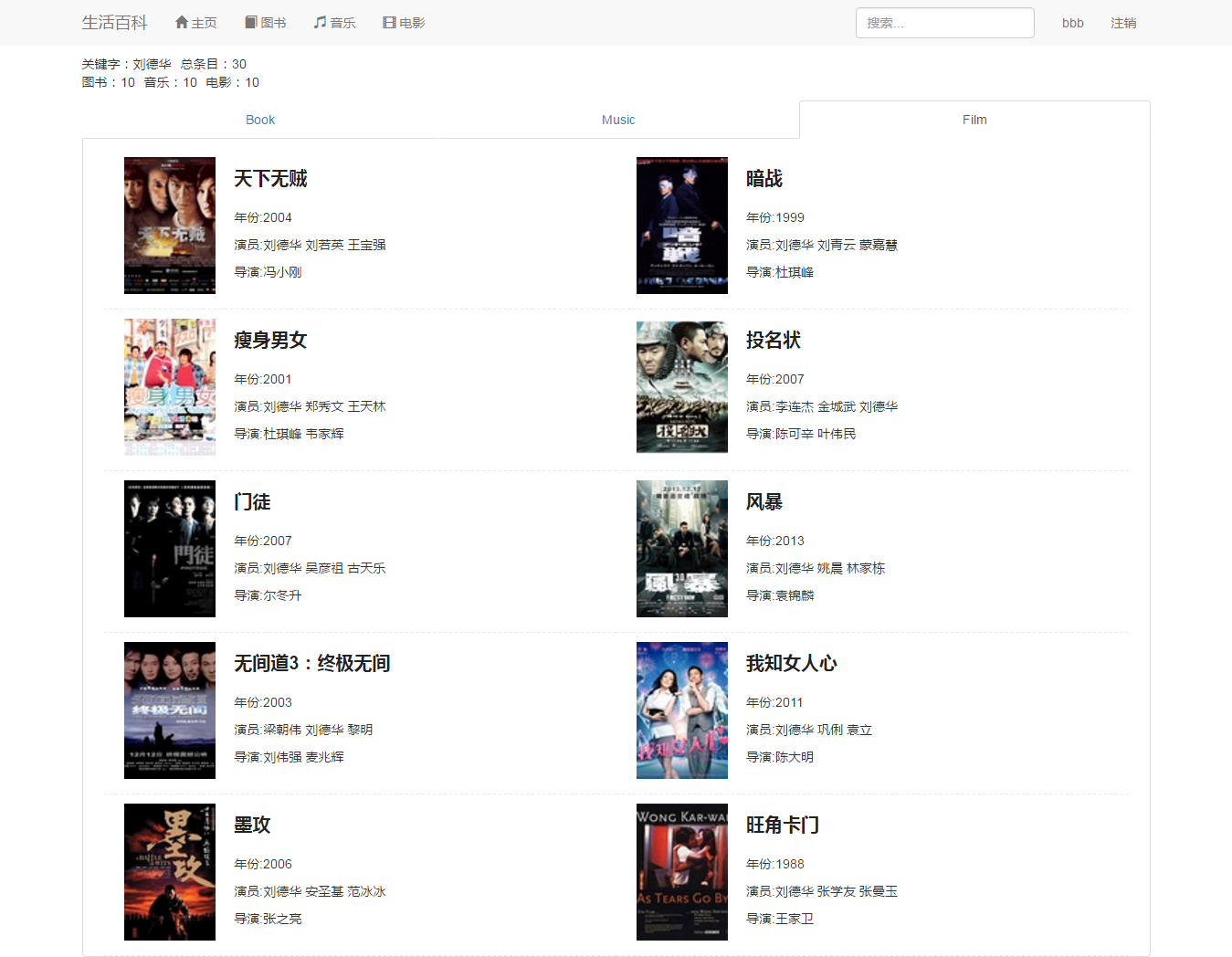


图5-2 搜索信息展示页面

5.2.2 条目详细信息



图5-3 查询页面图

如图5-3，就是点击了“投名状”条目之后进入的详细页面，当路由进入该模块时，模块会先请求“投名状”的信息，当请求成功返回JSON数据后，再取出数据中有用的部分进行页面渲染。该页面展示了“投名状”这部电影的而详细信息，而且还允许用户收藏

5.3 主页与脚注部分

5.3.1 主页

如图5-4，主页比较简单，只有一个Bootstrap的轮播图插件。轮播图的数据时动态的，每次打开系统都会先从后台请求数据，在得到数据后再进行页面的渲染。每隔一段时间就会切换一次图片。点击图片就会跳转到对应的条目页面。跳转依据的是从后台获取到的数据的link字段决定的。



图5-4 主页页面

5.3.3 主页天气功能

主页拥有天气功能，天气也是日常生活经常需要查询的信息了。在网页加载完后，会先请求得到当前位置信息，然后再根据地理位置请求响应的天气信息。得到今天以后4天的天气情况。天气还支持搜索功能，当前仅支持国内城市。当输入正确的城市名称后，就会通过后台向百度的天气API发出请求（前提需要在百度开发者中心注册并创建应用，得到AK码），成功请求后就能得到搜索城市当前的天气信息了。搜索框旁边还有一个切换详细信息的按钮。点击后就能看到各种指数信息。如图5-5，5-6，5-7



图5-5 主页天气部分

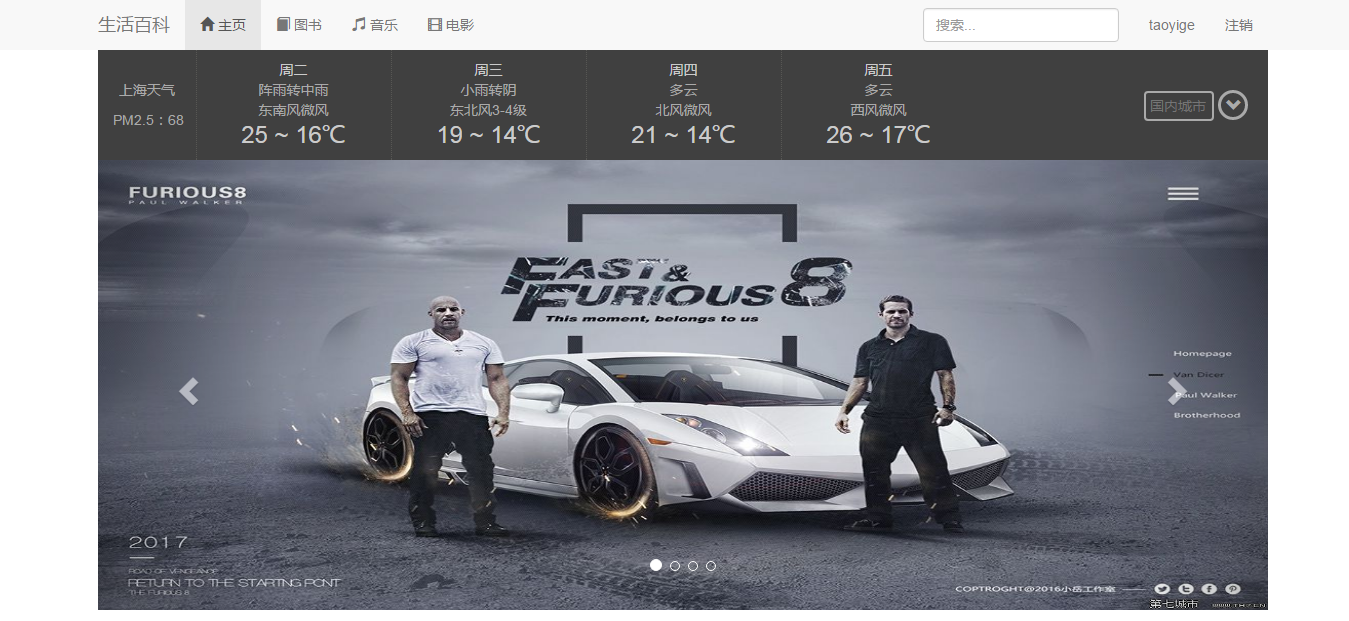


图5-6 搜索上海天气

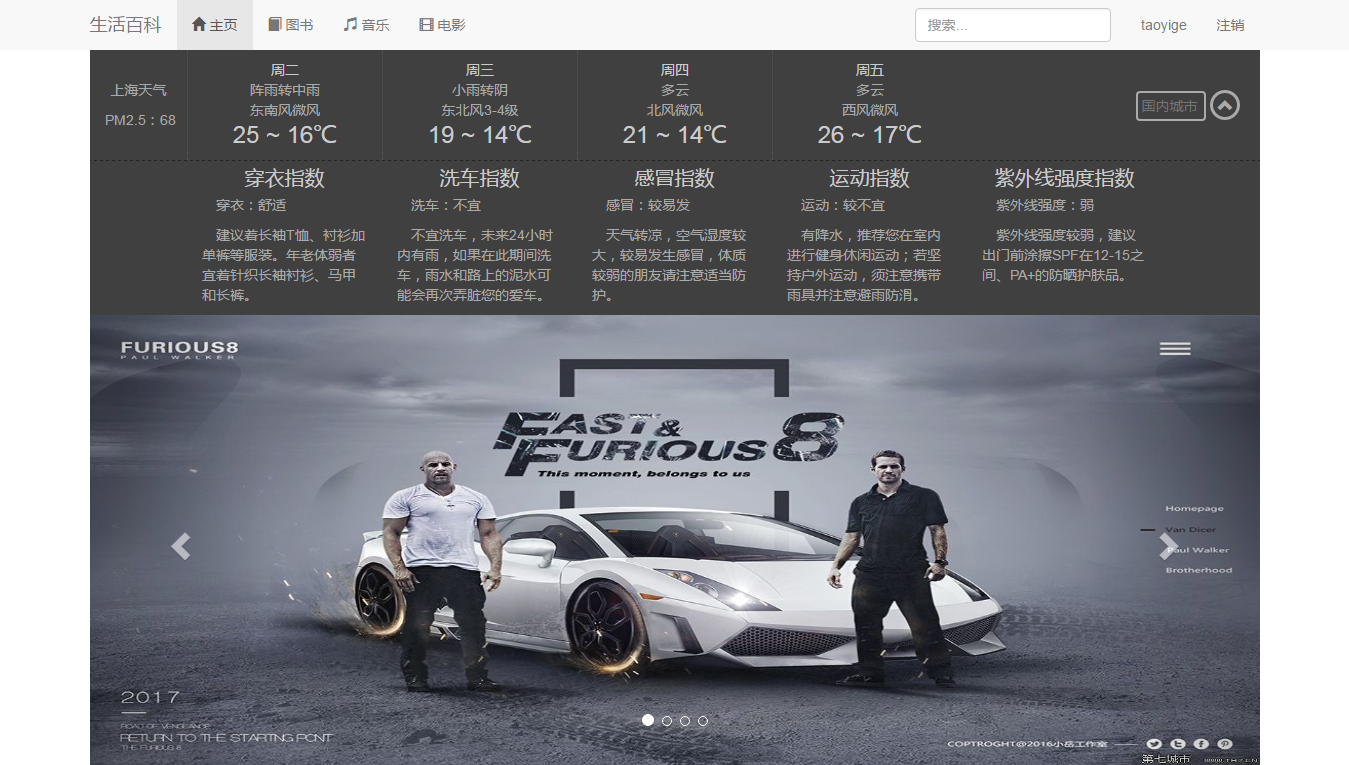


图5-7 海天气详细信息

5.3.4 脚注

如图5-8脚注部分也比较的简单。由两个部分组成，（1）友情链接；（2）GitHub。友情链接列出了相关的链接地址。GitHub部分则列出了GitHub的项目地址还有项目的代理页面。



图5-8 脚注部分

5.4 下拉加载更多部分

下拉加载更多功能，当页面下拉到底部时，会出现一个loading动画，动画是使用CSS3制作的。这时就会去请求数据，请求完成后就会把数据添加到源数据的末尾，再进行数据的渲染。其实loading动画是一直存在于最底部的，每当我们拉到最低端时就能看到，不过由于到达底部会出发加载更多，请求完数据以后就会进行渲染，这是就会把数据插入到loading组件的前面，从而使loading动画的距离顶部的距离加大，感觉是又隐藏了一起。这样就能实现下拉加载更过功能了。如图5-6，数据正在加载中。如图5-7，当数据加载完成后，整个页面高度明显增加了。



图5-9 下拉加载中



图5-10 加载完成

6 登录模块设计与实现

6.1 登录功能展示

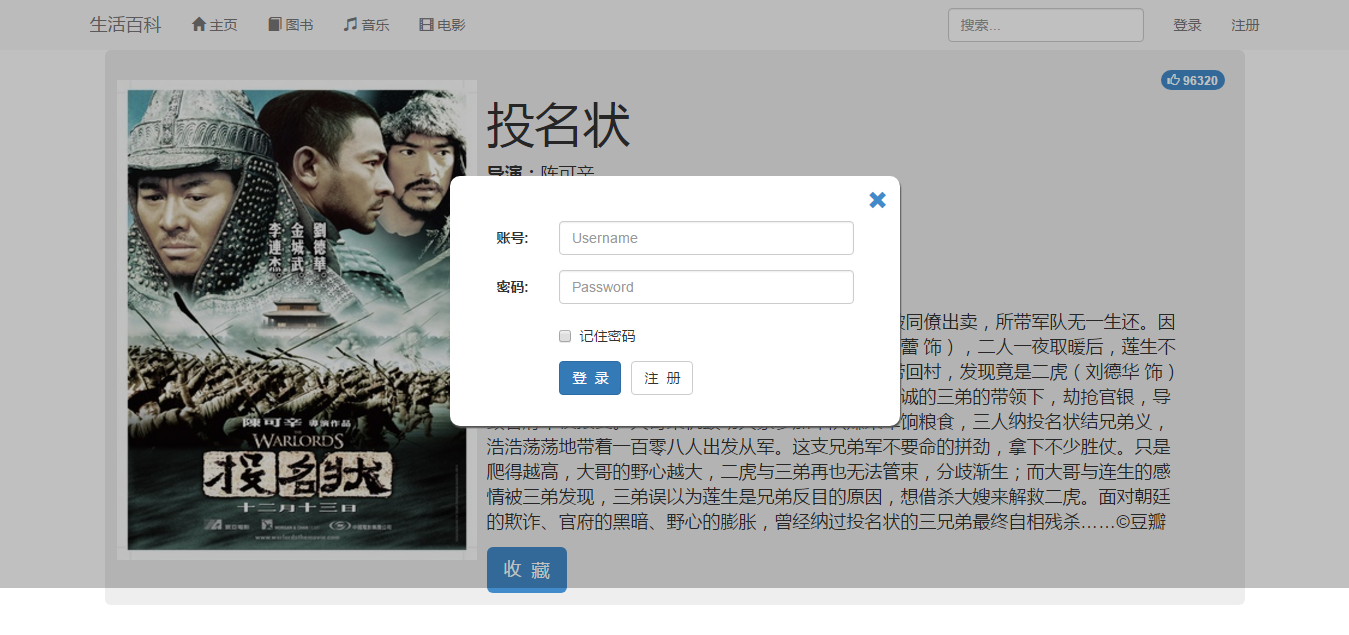


图6-1 管理员登录页面

如图6-1，当点击了登录按钮后，登录组件就会显示出来，底层实现原理其实就是：登录组件一早就已经加载了，只是把它隐藏起来，当点击了登录按钮后，把相应的状态值改为true，这是登录组件就会显示出来。当我们填上用户名和密码后，就可以进行登录了。

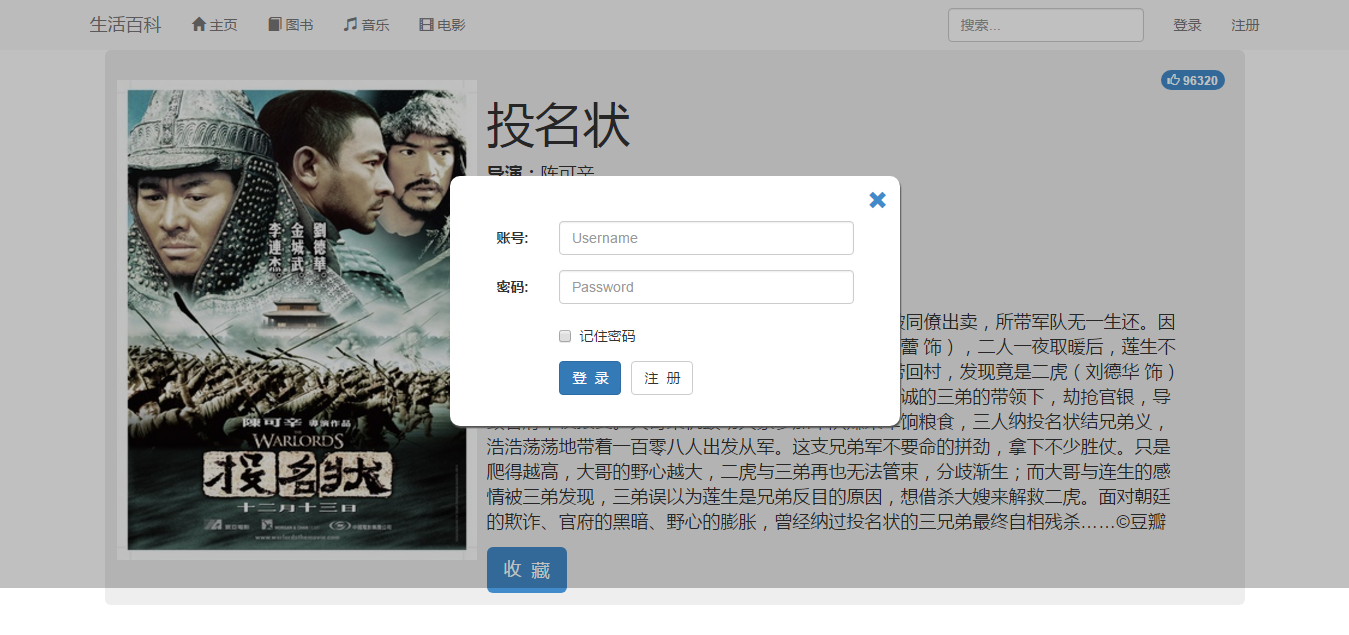


图6-2　登录失败提示信息

当登录失败时，还是友好的提示错误信息。提高用户体验。

6.2 注册功能展示



图6-3　注册功能显示

当点击了注册按钮，就会显示注册组件，其底层实现原理跟登录组件一样，都是先隐藏，改变状态后再显示。注册所需要输入的信息不多，只需要用户名，密码，确认密码还有邮箱。

如图6-3，这是密码不合法的提示信息。本系统注册需要的用户名和密码都是字母开头的3-6位。

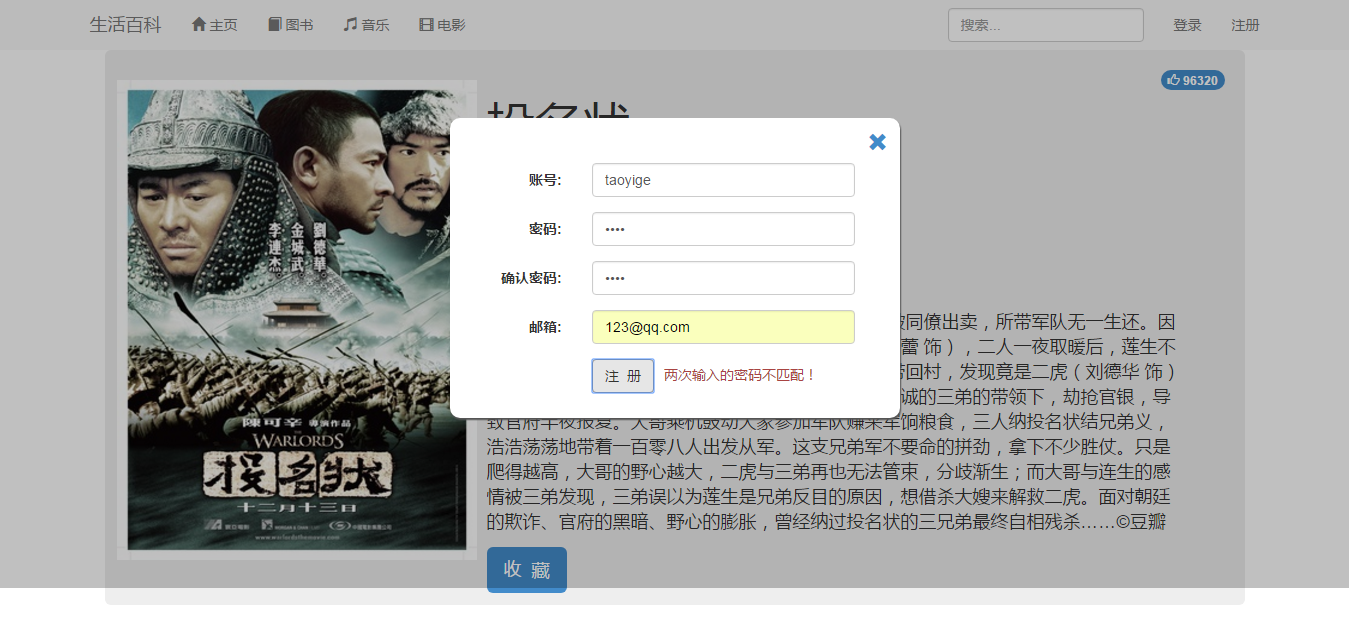


图6-4 注册错误显示

如图6-4，这是两次输入的密码不匹配的提示信息。

6.3 收藏模块功能展示

收藏模块的实现基于用户的成功登录。用户成功登录后，将在全局存在一个当前用户的变量。在进入收藏模块时，会先判断当前是否有成功登录的用户。没有则不进行展示。如果用户成功登录后，点击导航栏上的用户链接就能进入该用户的收藏页面。当然，一开始里面什么数据都没有，用户需要先去收藏一些条目。当收藏了条目后，我们再进入收藏页面，就能找到刚刚我们收藏的条目了。收藏页面也拥有类似搜索模块一样的选项卡，有图书，音乐和电影三个板块，我们可以进行切换来显示不同的板块。每个板块的数据都是在进入收藏模块前，获取当前用户后再向后台请求过来的数据。如果想取消收藏，则需要点击具体的条目，进入条目的详细页面点击取消收藏按钮。



图6-5 图书收藏板块



图6-6 音乐收藏板块

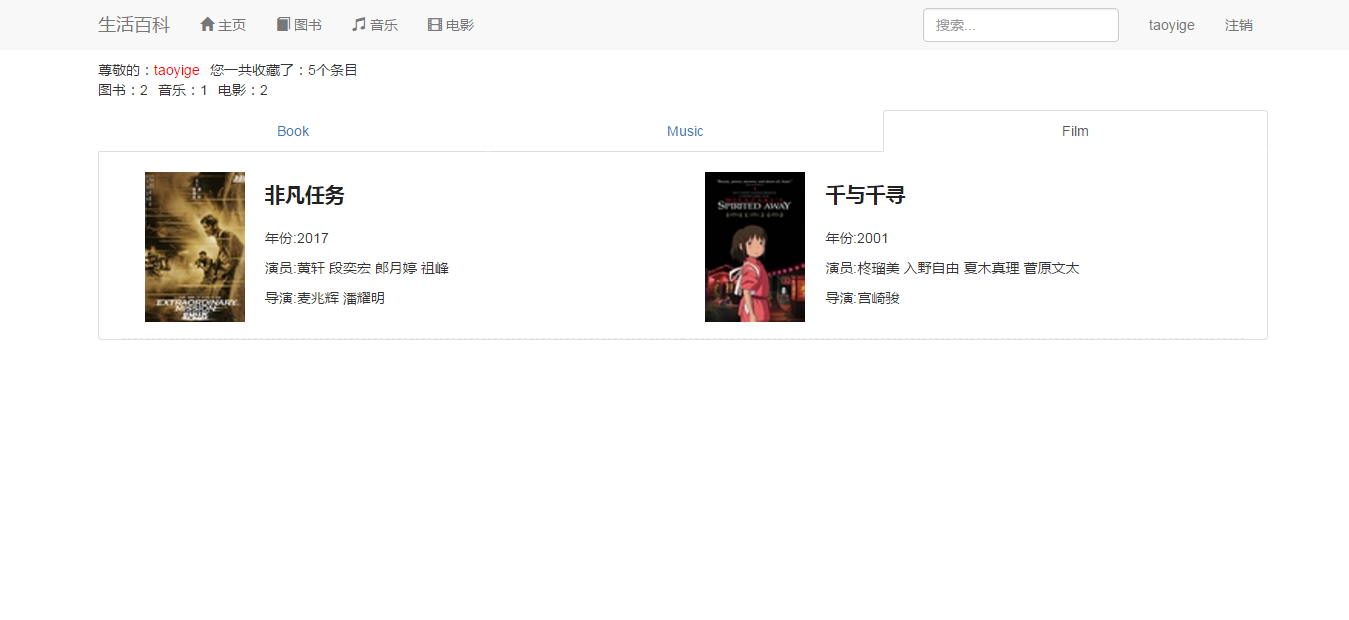


图6-7 电影收藏板块

7 Windows系统上环境的搭建

7.1 环境搭建的相关条件

操作系统：Windows操作系统，浏览器版本要求：IE9.0+，Firefox3.5+，Opera10.5+，Chrome5.0+，Safari3.0+，必须在联网情况下。Node版本：5.5.0

7.2 环境搭建的相关步骤

（1）首先需要安装NodeJS，这里建议安装NVMW（Node Version Manager for Window）Node的版本管理器，因为Node版本更新得很快，所以经常需要切换版本开发。本系统开发是就是基于Node 5.5.0的，现在的Node都已经出现7.0+了。安装NVMW后，配置环境变量。

（2）配置环境变量后，打开终端，使用nvmw install 5.5.0 命令就可以安装了Node了，安装完成就会默认使用该版本。下载完后，在NVMW根目录下就会多了一个5.5.0的文件夹，配置该文件夹路径到环境变量，完成后在终端输入node –version ，如果输出node版本号就表示成功安装Node。

（3）Node安装完后，在项目根目录打开终端，使用npm初始化需要的模块 npm install 这时npm就会根据项目根目录的配置文件package.json中的项目依赖依次下载需要的模块，注意由于本系统依赖模块比较多，所以初始化安装会比较慢。

（4）顺利安装完毕后，就可以输入 webpack-dev-server 命令进行项目调试了。当终端输入类似 webpack：Compiled successfully. 字段就表示编译成功，我们就可以打开浏览器输入 localhost:8080 ，此时就可以打开网站了。

（5）如果需要修改文件要在src文件夹下面修改，修改完成后，使用 webpack

命令就可以打包项目了，此时项目会构建，根目录就会多了一个build文件夹，里面的文件就是打包好项目了。

8 系统测试

8.1 测试目的

每个系统设计完成之后都要经过测试，来检验系统的健壮性，修复bug，提高性能。一个项目，可能前期准备占了30%，编码阶段占了20%，测试占到40%，剩下的可能就是发布和维护了。可以看出，测试占的比重相当大，如果说编码是创造出一个人来，但是这个人只是婴儿，什么都不懂。那么测试好比是这个人成长的经历了。测试会不断完善这个人，让这个人更加的健全。同样的，一个系统同样需要健全性。所以说绝对缺不了测试。

8.2 测试方法

测试方法从不同的角度分为三类。

|  |  |
| --- | --- |
| 测试方法 | 测试角度 |
| 黑盒测试和白盒测试 | 系统内部还是外部 |
| 动态测试和静态测试 | 是否执行程序 |
| \_单元测试、\_集成测试、\_确认测试、系统测试、验收测试、回归测试、 | 开发过程 |

表8-1 测试方法

黑盒测试：黑色的盒子，看不到里面的结构。主要对外部进行一些功能的测试。

白盒测试：白色的盒子，我们可以看到内部的结构了。主要着重于程序内部的测试。

动态测试：动态测试需要把测试目标运行后，把得到的结果与预期结果进行对比，根据运行结果与预期结果的异同，作出我们的测试分析。

静态测试：与动态测试相反的额，静态测试在不运行测试目标的前提下，对测试目标的总体结构进行分析。

单元测试：单元也就是模块测试，主要针对系统每一个模块进行正确性检验的测试，具体体现在定义的每一个组件上，以React定义的每一个组件作为最小检测单位，模块之间都是相互依赖的，所以单元测试可以发现各模块内部出现的问题。

集成测试：单元测试后紧接着就需要集成测试了。也就是组合测试的模式。组合过程中也经常会出现的问题，比如数据丢失、模块之间会不会产生相互的不利影响、是否能够达到预期想要的结果、全局数据是否有问题、每个模块之间的问题会不会被放大。

确认测试：测试目标功能是否达到了一开始设计系统是的需求要求。

系统测试：在特定的系统元素(计算机硬件、浏览器、外设、相关软件、数据、网络差异等)下，进行一系列的集成测试和确认测试。

验收测试：为了检测本系统的性能，兼容性和可扩展的程度。

回归测试：指测试完后由于需求的变更需要加入新的模块，所以要重新进行测试，以确认新模块没对测试目标的功能和性能产生影响。

8.3 测试用例

生活百科系统核心测试主要是对登录，注册地方添加验证。本次测试以验证成功登录和注册为例。

表8-2 注册用户测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能特性 | 注册用户验证 | | | | |
| 测试目的 | 验证输入信息是否合法 | | | | |
| \_编号 | \_操作 | \_数据 | \_预计结果 | \_真实结果 | \_测试状态 |
| 1 | 什么都没输入点击添加 | 都为空 | 无法提交 | 无法提交 | 与预计结果一样 |
| 2 | 用户名输入小于3个字符 | 用户名为”aa” | 用户名不合法！ | 用户名不合法！ | 与预计结果一样 |
| 3 | 不输入密码 | 密码为空 | 密码不合法 | 密码不合法 | 与预期一致 |
| 4 | 邮箱输入一串数字 | 邮箱“1231232” | 警告信息“必须为电子邮件地址” | 显示警告“必须为电子邮件地址” | 与预计结果一样 |
| 5 | 两次输入密码不一致 | 一：aaa  二：bbb | 密码不合法 | 密码不合法 | 与预计结果一样 |
| 6 | 输入正确的用户名，密码和邮箱 | 用户名：“taoyige”  密码：“taoyige”  邮箱：“taoyige@qq.com” | 注册成功 | 用户表多了一条数据 | 与预计结果一样 |

表8-2 添加企业信息测试

测试用例图：



图8-1 输入为空测试图



图8-2 邮件格式错误测试图



图8-3 注册成功测试图

表8-3　登录用户测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能特性 | 登录用户验证 | | | | |
| 测试目的 | 验证输入信息是否合法 | | | | |
| 编号 | 操作 | 数据 | 预计结果 | 真实结果 | 测试状态 |
| 1 | 什么都没输入点击添加 | 都为空 | 无法提交 | 无法提交 | 与预计结果一样 |
| 2 | 用户名输入小于3个字符 | 用户名为”aa” | 用户名不合法！ | 用户名不合法！ | 与预计结果一样 |
| 3 | 不输入密码 | 密码为空 | 密码不合法 | 密码不合法 | 与预期一致 |
| 4 | 输入正确的用户名，密码 | 用户名：“taoyige”  密码：“taoyige” | 登录成功 | 登录成功 | 与预计结果一样 |

表8-2 添加企业信息测试

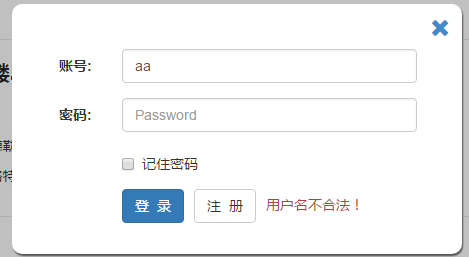


图8-4 用户名不合法测试图



图8-5 密码不合法测试图



图8-6 登录成功测试图

8.4 测试结论

经过本次测试，可以找到系统中很多不得不够好的地方，需要加以修改，直到功能达到设计需求和性能更加完善。生活百科系统中的主模块依赖每一个子模块，如果代码出现问题，那么是不能构建项目的。只有通过测试，找出BUG，然后不断的完善系统，使系统不断的健全和强大。BUG是肯定存在的，我们不可能写出一个完全没有BUG的系统，所以，纠错能力很重要。当然，测试和纠错完全不是一回事。测试只是找出问题所在，然而纠错才是真正的把问题解决掉。测试的前提是你默认系统使可运行的，但是纠错则的前提则是已出现潜在问题需要解决情况下。测试能发现系统的BUG，纠错可以把BUG减少。

9 项目管理

9.1 项目管理重要性

一个完整的项目，从前期的需求分析，功能设计。到中期的代码实现以及后期的测试，修复到最后的上线维护。整个项目的代码会发生巨大的变化。特别是在项目上线运营以后，后期的维护占了很大比重。项目的结构和代码也会发生巨大的变化。这时候如果不能系统的，有效的管理我们项目的结构和代码。项目的迭代就会超级复杂，困难和麻烦。严重时会导致项目瘫痪。这是我们不希望看到的，我们希望的是项目能够健康的有序的进行版本功能的迭代。这是如果我们手动管理的话，就会很麻烦，所以我用到了Git分布式版本控制系统工具帮我管理项目。

9.2 Git 和 GitHub 介绍

9.2.1 Git

Git是一个分布式的版本控制系统。它具有免费，开源，敏捷高效等特点。方便我们管理任何大小的项目。它的作者是Linux系统的开发者。当初只是想单纯的用来管理Liunx的代码而已。后来一把它开源了以后，就越来越流行了，成为现在主流的管理代码仓库。它支持版本回退，提交代码，查询日志以及分支等特点。还支持分布式管理。代码不用集中到同一个地方就可以管理。

9.2.2 GitHub

GitHub则是一个基于Git工具的的开源或私有项目的托管平台。它是其中一个使用Git作为代码管理工具的托管平台，也是最出名的一个。目前有很多知名的开源项目都在上面进行了托管，方便了我们开发者学习和使用。

9.3 代码仓库

9.3.1 项目在GitHub上面托管的地址

https://github.com/taoyige/HouseholdEncyclopedia

9.3.2 项目在GitHub上面代理的地址

https://taoyige.github.io/HouseholdEncyclopedia/build

9.3 项目开发版本

如图9-1，9-2，9-3。整个项目都使用了Git管理。在项目生命周期内，每次一个功能的实现，都会提交一个。

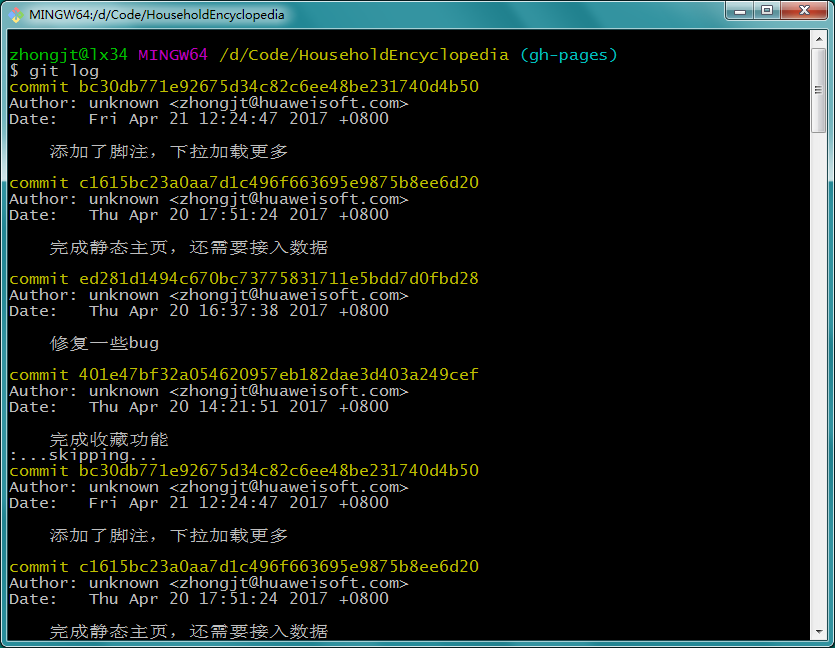


图9-1 Git版本图1

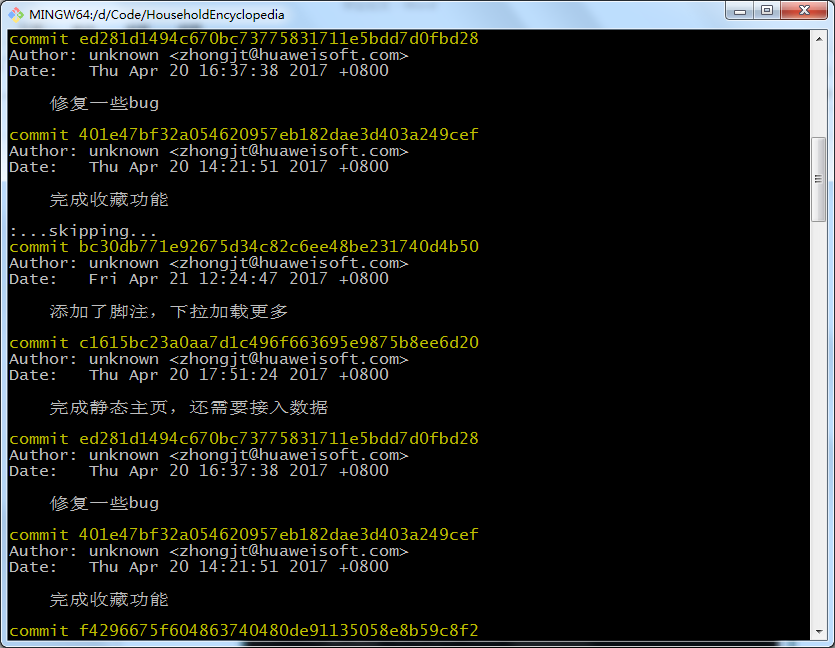


图9-2 Git版本图2



图9-3 Git版本图3

10 总结

生活百科系统的设计与开发，从一开始的需求分析，到功能设计，再到编写代码，后期的测试加上不断的修复BUG。我都学到了很多。

前端近两年发展很快，很多东西无论是产品本身还是思想上面，都是很创新的，例如React就提出了声明式组件编程。这是近段时间才出现的思想，但是却能马上得到回响。学习一样新东西都是需要时间和精力的花费的，我也是在摸着石头过河。一遍写系统一遍学习新东西。就好像刚使用React-Router时，还要边看例子边学着实现。

幸运的是，系统总体能按期按需完成。系统的模块分明。放在component文件夹下的都是定义出来的组件。util文件夹下的都是一些工具类。全局用到的变量都定义在了Global.jsx文件中。各个模块不再耦合，使用es6的新语法规则把需要用到的模块通过导入的方式导入进来使用。同时通过使用React，我也进一步加强了MVC的认识。视图-控制器-模块的分离，使得代码可控程度大大提高。同时也减轻了后期代码维护的难度。

由于我学的是前端，所以对后台认识不多，在技术选型时后台选择了交给第三方的后端云Bmob解决。它也能很好的与我的系统结合。数据交互也很方便。同时大大减轻了我后台的开发，使我的开发更加专注于前端。

可能与老师期望的要求不一致（需要做后台）。但是现在的前端已经不再是几年前的那个简单的页面结构，渲染数据了。下载的前端更加注重于后台的分离。与后台的交互只是单纯的数据请求而已。在浏览器上就完全可以完成以前的后台渲染动态页面的功能。而且加上H5的出现，页面变得丰富起来。很多功能都能在浏览器端实现。特别是三大框架Angular，Vue，React的出现，简直完全颠覆了前端的开发模式。

最后，我需要感谢我的指导老师，他帮助我解决了很多问题。在毕业设计中，我深刻的体会到要完成一个系统，单单编码是完全不行的，前期的设计，后期的测试，维护都非常的重要。无论做什么事都一定要做好准备功夫。不能走一步想一步。例如在项目开展的前期就需要做好需求的分析，功能的设计，技术的选型，所选技术的优缺点和是否成熟，技术能否实现需求，还有写好任务计划书。有一个正确的时间安排对项目的开展十分有效。只有做到有章可循，我们才会立于不败之地，当出现问题了，我们也能有心理准备，不至于什么都不会做。傻傻坐着像个傻瓜。

# 参 考 文 献

[1]新音乐产业观察. 从四个方面聊聊2016年中国音乐行业发展趋势[J].深度报道,2016-02-25.

[2] CY216. 电影产业发展趋势分析[J].中国产业信息，2017-01-18.

[3]阮一峰. ECMAScript 6 入门[M]. 电子工业出版社，2015-12.

[4]阮一峰. Babel 入门教程[J]. 阮一峰的网络日志,2016-01-25.

[5]阮一峰. React 入门实例教程[J]. 阮一峰的网络日志，2015-03-31.

[6]阮一峰. Redux 入门教程. 阮一峰的网络日志[J]，2016-09-18.

[7]泽卡斯 (Zakas. Nicholas C.)、 李松峰. JavaScript高级程序设计(第3版)[M].人民邮电出版社，2012-03.

[8] 张轩、 杨寒星. React全栈:Redux+Flux+webpack+Babel整合开发[M].电子工业出版社，2016-10.

[9]陈屹.深入React技术栈[M].人民邮电出版社,2016-11.

[10]弗里曼 (Adam Freeman)、 谢廷晟. 图灵程序设计丛书:HTML5权威指南[M]. 人民邮电出版社，2014-01.

[11] 陆凌牛. HTML5与CSS3权威指南(上册)(第3版)[M].机械工业出版社,2015-09.

[12] 朴灵. 深入浅出Node.js[M].人民邮电出版社,2013-12.

[13] [美] 科克伦，[美] 惠特利、 李松峰. Bootstrap实战[M]. 人民邮电出版社,2015-05.

[14]阮一峰. ES6标准入门(第2版)[M].电子工业出版社，2015-12.

**The Design and Development of the Online System for Household Encyclopedia**

Zhong Jingtao

(Agronomy Department Zhongkai University of Agriculture and Technology，Guangzhou 510225,China)

**Abstract：**In this paper, the life needs of the common information (books, music, movies) as the research object, the design and development of the online encyclopedia of life system, the effective realization of the information query. This design can also be used for the login user can use the collection function.

This design mainly uses the React+Redux+Webpack+Babel+Bootrap technology to realize the front page and the logic construction. The core of the system is divided into two parts: first, the three modules (books, music, movies) information display, the details of the specific items displayed. Two, the query module, according to the keyword query needs.

**Keywords：** Life Encyclopedia front; React; Redux; Webpack;

致 谢

不知不觉，俨然已经到了毕业季，这应该是我人生中在校园的最后一段日子了。回想四年前那个懵懂的少年走进了仲恺校园。怀抱着远大理想，到现在还不知道算不算已走出了第一步，慢慢来似乎时间也不会等你。加油吧，后面的路还长着。

在这毕业的季节里，毕业设计的完成也预示着大四的落幕，即将离开学校步入社会。

在这里，我想对那些帮助我完成毕业设计的人说一声谢谢。

刘佳老师，我的指导老师，感谢他，没有他我的毕业设计应该不能按期完成吧！毕业设计的顺的完成，真的多亏了刘佳老师的多次指导和修正，从开题报告一直到论文的完成，刘佳老师都有一直发邮件悉心为我解答我遇到的问题，时不时也会询问毕业设计的完成情况。刘佳老师是一个认认真真、尽心尽责的好老师，多亏了刘佳老师认真负责的工作态度和严谨的治学风格，我的毕业论文按期按需地顺利完成了，老师教了我太多太多东西了，例如编程思想，算法实现还有很多东西。感谢老师教授我的一切，感谢老师的悉心教导，感谢老师对我毕业设计上的支持和关怀。

当然了，毕业设计总少不得网上教材，这里我需要特别感谢阮一峰老师，他在网上写了很多很好的文章，帮助新手快速的上手一个新框架。其中的React系列教程让我感受良多。

论文的顺利完成也少不了我同学的功劳，感谢同学不嫌麻烦的和我探讨问题，指正我的错误，感谢你们给我的大学生活增添了新的风景，感谢母校的培养。

最后，感谢百忙之中抽空审阅本论文的良师益友。

# 仲恺农业工程学院毕业设计成绩评定表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓 名 | 林春鹏 | | | | | | 学 号 | | 201210314129 | | | | 2012届 | |
| 院(系)别 | 信息科学与技术学院 | | | | | | | | 专业、班级 | | 电子信息工程121班 | | | |
| 毕业设计题目 | | | 在线食品安全系统的设计和开发 | | | | | | | | | | | |
| 指导教师姓名、职称 | | | 刘佳 副教授 | | | | | | | | | | | |
| 指导教师评语及评分：  签名  年 月 日 | | | | | | | | | | | | | | |
| 评阅老师评语及评分：  签名  年 月 日 | | | | | | | | | | | | | | |
| 答辩记录：  答辩秘书： 年 月 日 | | | | | | | | | | | | | | |
| 答辩小组评价意见及评分：  答辩组长：  年 月 日 | | | | | | | | | | | | | | |
| 设计成绩 | | 指导老师（40%） | | | 评阅老师（20%）  折算分 | | | 答辩小组（40%）  折算分 | | | | 总评分 | | 等级 |
| 原始分 | | 折算分 | 原始分 | 折算分 | | 原始分 | | 折算分 | |
| 96 | | 38.4 | 94 | 18.8 | | 90 | | 36 | | 93.2 | | 优秀 |
| 院系答辩领导小组  (委员会)审核意见 | | 签章  年 月 日 | | | | | | | | | | | | |

注：1、论文(设计)成绩一栏中，折算分分别由指导老师（40%）、评阅老师（20%）、答辩小组（40%）给出的原始分乘以各自百分比例所得。总评分由折算分相加所得。

2、“等级”：90分以上为“优秀”、80―89分为“良好”、70―79分为“中等”、60―69分为“及格”、59分以下为“不及格”。