

浙江大学

项目实训之垂直搜索 “梦奇电影旅行”系统设计报告



学生姓名：	<u>胡洋凡</u>	学号：	<u>3180103167</u>
学生姓名：	<u>蔡灿宇</u>	学号：	<u>3180101972</u>
学生姓名：	<u>刘轩铭</u>	学号：	<u>3180106071</u>
学生姓名：	<u>杨凌霄</u>	学号：	<u>3180103608</u>
学生姓名：	<u>王子腾</u>	学号：	<u>3180102173</u>
学生姓名：	<u>王绍兴</u>	学号：	<u>3180106074</u>

2021 年 07 月 12 日

Verse:1.1

版本变更历史

版本号	作者	修订日期	审核者	审核日期	修订原因
1.0	胡洋凡、蔡灿宇、 王子腾、王绍兴、 杨凌霄、刘轩铭	2021.07.11	蔡灿宇	2021.07.11	原始文档
2.0	胡洋凡、蔡灿宇、 王子腾、王绍兴、 杨凌霄、刘轩铭	2021.07.12	蔡灿宇	2021.07.12	样式的统一和部分 修改

目录

1. 引言	5
1.1 编写目的	5
1.2 背景	5
1.3 参考文件	6
1.4 定义	6
2. 系统设计	7
2.1 用户需求规定	7
2.2 其他需求规定	7
2.3 运行环境	8
2.4 基本设计概念和处理流程	9
2.5 结构	9
2.6 用例设计	11
2.6.1 用户登录:	11
2.6.2 用户搜索:	12
2.6.3 用户查看词云热搜:	14
2.6.4 用户查看推送:	15
2.6.5 用户查看电影详情	16
2.6.6 用户查看下一篇:	18
2.7 数据字典	19
2.8 人工处理过程	20
2.9 尚未解决的问题	20

3. 接口设计	20
3.1 用户接口	20
3.2 外部接口	21
3.3 内部接口	21
3.3.1 前后端合作部分	21
4. 运行设计	26
4.1 运行模块的组合	26
4.2 运行控制	26
4.3 运行时间	27
5. 系统数据结构设计	27
5.1 逻辑结构设计要点	27
5.1.1 MySQL 数据库设计:	27
5.1.2 MongoDB 数据库设计:	29
5.2 物理结构设计要点	30
6. 系统出错设计	31
6.1 出错信息	31
6.2 补救措施	31
7. 系统维护设计	32
7.1 概述	32
7.2 检测点设计	33
7.3 系统维护设计	34

1. 引言

1.1 编写目的

本《系统设计报告》基于前面与用户进行的需求访谈及相关技术人员商讨后，规定下来的系统设计结构及相关说明。

本《系统设计报告》通过引言、运行环境、测试环境、系统结构设计、数据处理及相关对应的系统实现对读者阐述本系统规定的设计计划，明确了本系统的系统概貌、体系设计等，并附带目前版本系统的用户界面截图，建立软件产品和软件项目的总计结构、结构设计、系统出错处理设计等，更好地让开发和维护人员了解本项目的具体架构。

1.2 背景

该项目开发的软件为一个可以为电影和旅行爱好者使用的电影相关信息搜索引擎。梦奇电影旅行是一个影视旅行主题应用，通过它可以沿着某部影视剧或电影了解到其取景地点或者配乐的情况，也能通过选择地点和音乐，了解到曾在此处取景的影视剧的取景片段及剧情发生的时间点或者以该音乐为配乐的众多音乐。

喜欢电影的人应该会有相同的感觉：遇到某部特别喜欢的电影，就很想去了解这部电影的取景地，或者这部电影的配乐信息，甚至想要跟随电影的主人公，走过他走过的路，身临其境，听着熟悉的音乐，感受电影和现实交织在一起的美妙，然后再重温一遍电影，会觉得是一场非常奇妙的体验。

梦奇电影旅行就是一款献给热爱电影和旅行的朋友的一款指南应用，用户可以在应用中搜索熟悉的地点和音乐，或者是电影本身，了解这些事物背后的电影故事。

1.3 参考文件

- [1] 《软件工程——实践者的研究方法》，Roger S. Pressman，机械工业出版社
- [2] 《软件需求（第三版）》，Karl Wieggers, Joy Beatty，清华大学出版社
- [3] 《软件工程开发国家标准》
- [4] 《G14_项目计划报告》
- [5] 《G14_需求规格说明书》

1.4 定义

1. 认证：防范信息的损坏和泄露。
2. 权限：指用户职能的范围，即各种用户所登录界面、所接触数据、所进行操作等的范围。
3. 用例：用户场景通常称为用例，它提供了系统将如何被使用的描述。
4. IPO 图：是输入/处理/输出图的简称，描述输入数据、对数据的处理和输出数据之间的关系。
5. 数据流图：是 SA 方法中用于表示系统逻辑模型的一种工具，它以图形的方式描述数据在系统中流动和处理的过程，它是一种功能模型。
6. 数据词典：是对所有与系统相关的数据元素的一个列表、以及精确严格的定义，使得用户和系统分析员对于输入输出、存储成分和中间计算有共同理解。
7. 系统集成：将不同的系统，根据应用需要，有机地组合成一个一体化的、功能更加强大的新型系统的过程和方法。

2. 系统设计

2.1 用户需求规定

1. 根据关键词搜索影片相关音乐、片场等信息
2. 根据搜索结果查看信息详情
3. 能进行登录以反馈行为信息
4. 查看热搜电影
5. 智能推荐
6. 能通过点击下一篇达到不间断浏览的目的

2.2 其他需求规定

性能需求

服务器性能要求能满足网友日常的访问需求，保持并发承受能力，在界面简洁美观的基础上，使网络延时短

输入输出需求

客户端通过网页形式向用户展现一个简洁友好的操作界面，用户可以点击超链接跳转界面，点击按钮提交表单。服务器后台处理数据后显示在网页界面上。

数据管理需求

安全：数据库对参与数据库操作的数据都会预处理后再加入查询语句，避免通过网页 SQL 注入的方式获取数据库信息或破坏数据库。数据库中密码通过不可逆加密加密存储，限制过短的密码，减低密泄露的风险。

性能：对于需要频繁访问数据库的操作，需要建立持久的数据库连接。

故障处理需求

1. 软件开发团队提供系统维护支持，有完备的内部文档和外部文档。
2. 系统报错信息可读性高

2.3 运行环境

本网站主要服务于众多网友，需要保证至少 200 名网友同时取得服务的需求，包括数据存储能力和网络吞吐能力，保证账户一定的安全性

软件运行环境：

项目	信息
操作系统	Windows 10
服务器软件	Apache5.5
数据库软件	MySQL5.6, MongoDB

硬件运行环境：

项目	名称
操作系统	CPU: Intel core i5 及以上
	内存: 2G 及以上
	硬盘: 200G 及以上
应用服务器	内存: 512M 及以上
	硬盘: 50G 及以上
通讯设备	具有良好的数据传输能力

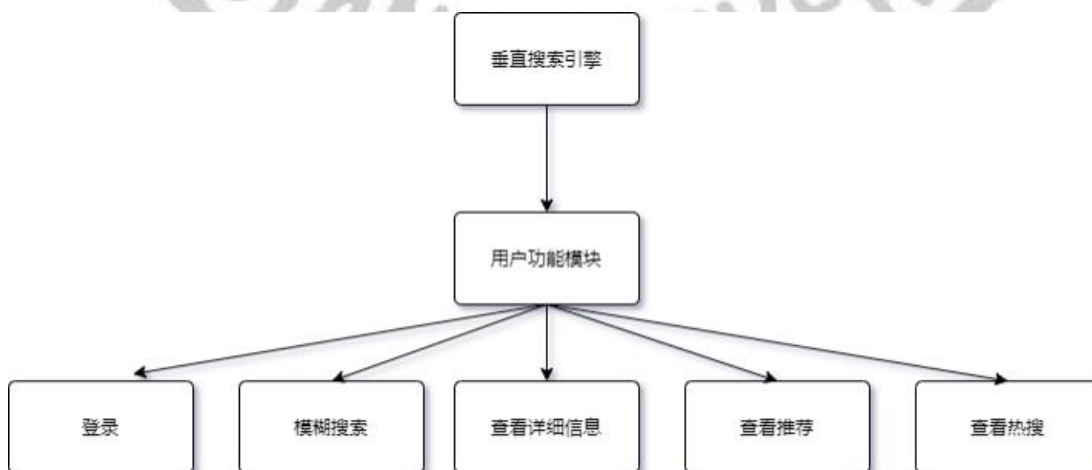
2.4 基本设计概念和处理流程

服务器以 Apache 为服务器，Java 语言编写后端代码，数据库采用 Mysql 和 MongoDB，搜索引擎使用 Elasticsearch。当用户通过浏览器使用网站系统时，浏览器接收用户的请求，并传送到服务器，执行相应的程序，通过搜索引擎 ES 查询相关内容，并且从数据库接口函数向数据库发送 SQL 查询语句，数据库接收 SQL 查询语句后执行，返回查询结果，处理查询结果后返回给前端，并显示在网站页面上。客户端浏览器采用常用的 IE, Chrome, FireFox 等。客户端在不频繁的操作页面时完成操作后断开与数据库的连接以减轻服务器负荷；在操作频繁时保持连接以增加访问度。

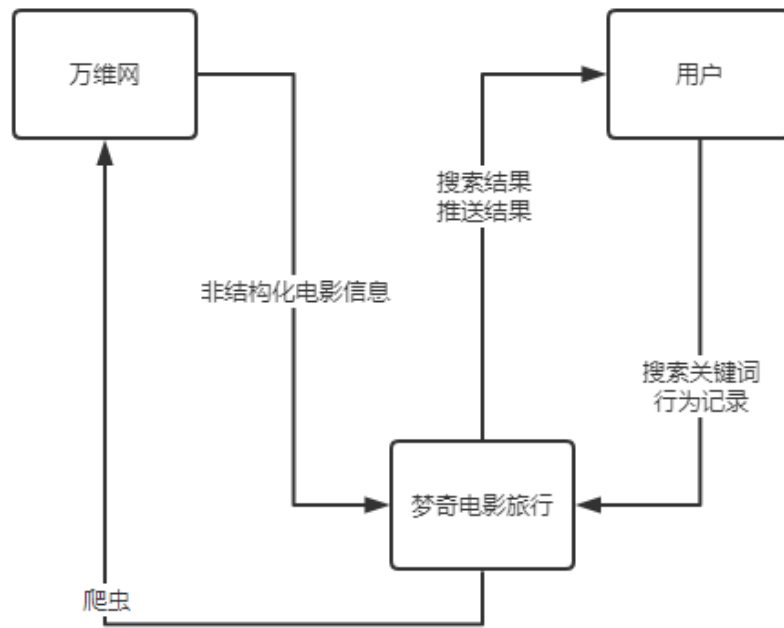
客户端动态页面：嵌入 Vue，动态网页以数据库技术为基础，能降低网站维护的工作量。Vue：页面的各种文本框与按键操作能够完成，同时能实现无刷新页面的一些动画效果，包括下拉菜单等

2.5 结构

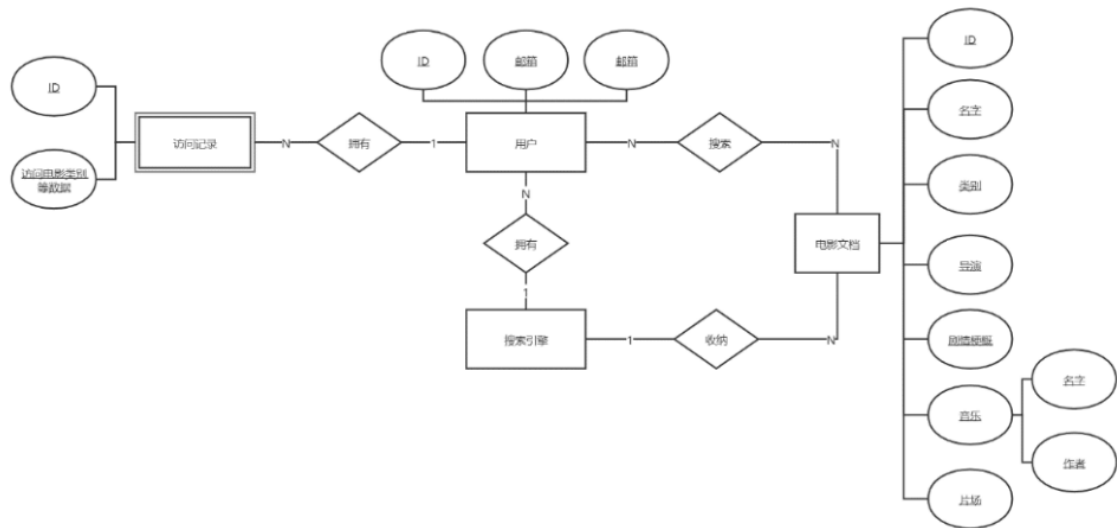
系统模块架构图：



数据流图：



ER 图：



2.6 用例设计

2.6.1 用户登录:

用例编号	UC-LC-01	用例名称	用户登录
创建人	杨凌霄	最后修改人	杨凌霄
创建日期	2021/7/10	最后修改日期	2020/7/10
角色	用户	需求来源	用户代表
主要参与者	任何使用本系统的用户		
描述	用户使用邮箱验证的方式登录系统，输入验证码。验证成功后可以登录，登录成功进入搜索引擎页面。		
前置条件	系统可正常运行 用户已连接网络		
触发器	用户需要进入登陆页面，并输入自己的邮箱		
后置条件	用户进入搜索引擎页面，登录状态切换为已登录		
输入信息	用户的邮箱和验证码		
主干过程	用户通过浏览器进入网站首页 用户输入邮箱，点击验证按钮 系统发送含有验证码的邮件到用户邮箱 用户输入验证码，点击登录按钮 验证码匹配即显示成功登录，用户进入搜索页面		
分支过程	无		
异常	邮箱不存在或者邮箱格式错误(发生在主干进程的第3步)。		

	<p>处理：系统提示“请检查输入的邮箱地址是否正确”，返回步骤 3。</p> <p>验证码不匹配（发生于主干过程步骤 4）。</p> <p>处理：系统提示“验证码错误！”，返回步骤 3。</p> <p>验证码为空（发生于主干过程步骤 4）。</p> <p>处理：提示“验证码不能为空！”，返回步骤 3。</p> <p>系统结束用例。</p>
输出	<p>登陆成功提示。</p> <p>显示系统搜索页面。</p>
包括用例	无
优先级	5
使用频率	高
假设	无
备注与问题	无

2.6.2 用户搜索：

用例编号	UC-LC-02	用例名称	用户搜索
创建人	杨凌霄	最后修改人	杨凌霄
创建日期	2021/7/10	最后修改日期	2020/7/10
角色	用户	需求来源	用户代表
主要参与者	任何使用本系统的用户		
描述	用户使用输入文字的方式进行搜索，也可以选择切换优先搜		

	索方式。切换后会影响搜索结果排序，搜索成功后显示结果。
前置条件	<ol style="list-style-type: none"> 1. 系统可正常运行 2. 用户已连接网络
触发器	用户需要进入搜索页面，已经登录
后置条件	搜索页面呈现搜索结果并分页
输入信息	用户查询的文字信息
主干过程	<p>用户登录后进入搜索页面</p> <p>用户输入相关文字信息</p> <p>用户选择电影/地点/音乐优先搜索或默认不选</p> <p>系统根据用户输入进行搜索</p> <p>系统返回搜索结果并分页</p>
分支过程	无
异常	<ol style="list-style-type: none"> 1. 搜索内容不存在(发生在主干进程的第4步)。 处理：系统提示“搜索无结果”，返回步骤2。 2. 搜索内容为空（发生于主干过程步骤4）。 处理：系统无法进行搜索，返回步骤2。 <p>系统结束用例。</p>
输出	<ol style="list-style-type: none"> 1. 搜索成功。 2. 页面显示搜索结果并分页。
包括用例	无
优先级	7
使用频率	高

假设	无
备注与问题	无

2.6.3 用户查看词云热搜：

用例编号	UC-LC-03	用例名称	用户查看词云
创建人	杨凌霄	最后修改人	杨凌霄
创建日期	2021/7/10	最后修改日期	2020/7/10
角色	用户	需求来源	用户代表
主要参与者	任何使用本系统的用户		
描述	用户进入搜索页面，搜索页面自动呈现词云，词云是根据用户的热搜词语形成的。		
前置条件	1. 系统可正常运行 2. 用户已连接网络		
触发器	用户需要进入搜索页面，已经登录		
后置条件	搜索页面呈现词云		
输入信息	无		
主干过程	用户登录后进入搜索页面 系统自动呈现词云热搜榜		
分支过程	无		
异常	1. 词云内容不存在(发生在主干进程的第2步)。 处理：系统提示“暂无形成词云”，返回步骤1。 系统结束用例。		

输出	1. 页面成功显示词云。
包括用例	无
优先级	4
使用频率	高
假设	无
备注与问题	无

2.6.4 用户查看推送:

用例编号	UC-LC-04	用例名称	用户查看推送
创建人	王子腾	最后修改人	王子腾
创建日期	2021/7/10	最后修改日期	2020/7/10
角色	用户	需求来源	用户代表
主要参与者	任何使用本系统的用户		
描述	在首页，用户可以查看系统给出的推送，推送以小窗的形式展示，其内容根据用户的历史搜索和查看记录智能生成，包含机器学习模块。		
前置条件	系统可正常运行 用户已连接网络 用户已登录		
触发器	用户通过登录后，跳转进入主页		
后置条件	系统给出推送电影，用户可查看或点击进入		
输入信息	用户 ID（登录后保存在 session 中）		

主干过程	用户通过登录部分 用户所在位置为主页 用户查看推送元素 用户点击推送元素可进入详情界面
分支过程	无
异常	用户未登录 处理：系统提示未登录，并跳转至登陆页面 用户未进行过搜索或浏览历史 处理：系统随机选择推送内容
输出	推送内容元素显示
包括用例	无
优先级	7
使用频率	高
假设	无
备注与问题	无

2.6.5 用户查看电影详情

用例编号	UC-LC-05	用例名称	用户查看电影详情
创建人	王子腾	最后修改人	王子腾
创建日期	2021/7/10	最后修改日期	2020/7/10
角色	用户	需求来源	用户代表

主要参与者	任何使用本系统的用户
描述	用户点击首页推送或搜索结果后，系统会为用户跳转至电影详情界面，在详情界面中，用户可查看电影基本信息、相关取景地和配乐信息等，为信息展示页面
前置条件	系统可正常运行 用户已连接网络 用户已进入系统
触发器	用户点击电影链接进入详情
后置条件	系统跳转进入用户所点击电影的详情页面
输入信息	鼠标选中电影点击触发
主干过程	用户点击电影链接 用户进入电影详情界面 用户查看详情界面
分支过程	无
异常	用户未登录 处理：系统提示未登录，并跳转至登陆页面 电影取景地照片未显示 处理：系统重新加载当前页面 电影配乐未加载 处理：系统自动加载或用户手动刷新
输出	电影详细信息显示
包括用例	无

优先级	7
使用频率	高
假设	无
备注与问题	无

2.6.6 用户查看下一篇：

用例编号	UC-LC-06	用例名称	用户查看下一篇
创建人	王子腾	最后修改人	王子腾
创建日期	2021/7/10	最后修改日期	2020/7/10
角色	用户	需求来源	用户代表
主要参与者	任何使用本系统的用户		
描述	在电影详情界面，用户可以通过点击左右翻篇按钮，在系统给出的前后相关电影中进行选择，点击后将跳转至相应的电影详情页面。		
前置条件	系统可正常运行 用户已连接网络 用户已进入详情界面		
触发器	用户点击前/后翻篇按钮		
后置条件	系统给出前/后顺位电影的详情界面		
输入信息	鼠标点击		
主干过程	用户位于详情界面 用户点击翻篇按钮		

	系统跳转至相邻电影详情界面
分支过程	无
异常	<p>用户未登录</p> <p>处理：系统提示未登录，并跳转至登陆页面</p> <p>用户双击翻篇按钮</p> <p>处理：系统只进行一次跳转</p>
输出	电影详细信息显示
包括用例	无
优先级	7
使用频率	高
假设	无
备注与问题	无

2.7 数据字典

用户搜索词条

数据元素	描述	数据类型	数据长度	取值
搜索内容	搜索到的内容	字符串	0-200 位	0-9, a-z, A-Z
词条内容	返回的内容	Moivie 对象	1	Json 数据

词条详细信息

数据元素	描述	数据类型	数据长度	取值
名称	词条的名称	字符串	0-200 位	0-9, a-z, A-Z
Id	词条的 Id	字符串	0-200 位	0-9, a-z, A-Z
地点	词条相关地点	字符串	0-200 位	0-9, a-z, A-Z
音乐	词条相关音乐	字符串	0-200 位	0-9, a-z, A-Z

2.8 人工处理过程

在本系统的运行过程中,可能会出现一些系统无法自动解决的问题,需要人工处理介入来解决,包括词条的删改、词条相关图片视频的爬取等

2.9 尚未解决的问题

暂无

3. 接口设计

3.1 用户接口

本系统为搜索引擎系统,网友所有行为均在网页上实现。

用户通过鼠标点击选择输入优先级,通过键盘输入搜索关键词。除去搜索外,用户通过弹窗登陆。

主要接口分布如下：

一级链接	二级链接	三级链接	四级链接
搜索引擎搜索	搜索结果页面	电影详情页面	音乐搜索页面
			电影详情页面
搜索引擎推送	电影详情页面	电影详情页面	

3.2 外部接口

在应用上线之前，开发者应该提前从数据源爬取数据，将内容导入到数据库中，以达到初始化数据库的目的。

项目上线之后，外部接口仅有一个。当用户在电影详情页面看到了音乐名称后，点击音乐名可以直接跳转到网易云的歌曲搜索页面。

3.3 内部接口

内部接口为前后端合作需要用到的接口，以及后端不同模块之间的合作需要用到的类型定义。

3.3.1 前后端合作部分

枚举类定义：

```
enum SearchType {  
    DEFAULT(0), MOVIE(1), MUSIC(2), LOCATION(3), ERROR(-1);  
    private int code;  
    private SearchType(int code) {  
        this.code=code;  
    }  
}
```

```

    }

    public int getCode() {
        return this.code;
    }

    public static SearchType getObject(int code) {
        if(code == 0)
            return SearchType.DEFAULT;
        else if(code == 1)
            return SearchType.MOVIE;
        else if(code == 2)
            return SearchType.MUSIC;
        else if(code == 3)
            return SearchType.LOCATION;

        //no match, return error
        return SearchType.ERROR;
    }
}

```

合作类型定义:

```

class Music:
    name=string
    composer=[string, string,..., string]
    #网易云链接前端弄

class Location:
    images=[string, string,..., string]#url
    introduction=string
    visit=[string, string,..., string]#所有页面右侧的黑框的所有 visit 变成一个数组

class Movie:
    _id=int
    name=string(同名则在后面括号加年份)
    post=string(封面的 url)
    music=[Music, Music,..., Music]
    location=[Location, Location,..., Location]

    #以下为不关键信息
    year=string(上映年份)

```

```
country=string(出版国家)
director=[string, string, ..., string](导演列表)
genre=[string, string, ...](电影类别, 如动作片)
rating=float(0-10.0)(打分)
popularity=int(评论人数)
description=string(简介)
updateTime=Timestamp(此文档的最后修改时间)

class MovieBrief:
    #返回的电影概要, 出现在搜索结果中
    #一个对象对应一个搜索结果
    _id=int
    post=string(封面的 url)
    name=string(同名则在后面括号加年份)
    year=string(上映年份)
    genre=[string, string, ...](电影类别, 如动作片)
    location=[Location, Location, ..., Location]
    director=[string, string, ..., string](导演列表)
    rating=float(0-10.0)(打分)
    music=[Music, Music, ..., Music]
```

HTTP 响应定义:

```
{
  success:true/false (bool),
  code:200 (int),
  message:'123' (string),
  data:Object
}
```

API:

1. 获取登陆验证码

```
Send:
method: post
url:/app/getVerificationCode
body:
{
  email:'123@qq.com'
}
```

```
Return:
{
  success: ...,
  code: ...,
  message: ...,
  data:null
}
```

2. 验证

```
Send:
method: post
url:/app/verify
body:
{
  email:'123@qq.com',
  verrify:'123456'
}
```

```
Return:
{
  success: ...,
  code: ...,
  message: ...,
  data:null
}
```

3. 搜索

```
Send:
method: get
url:/app/search
params:
{
  keyword:'the avengers',
  priority:0/1/2/3,
  #0 混合
  #1 电影
  #2 音乐
  #3 取景地
  page:1,2... (int)#从 1 开始编号
  pageSiz:10(int)
}
```



```
Return:
{
  success: ...,
  code: ...,
  message: ...,
  data:
  [
    MovieBrief1, (电影简略信息)
    MovieBrief2,
    ...,
    MovieBriefn
  ]
}
```

4. 获取推送

```
Send:
method: get
url:/app/getRecommendation
params:
{
  page:1,2... (int)#从 1 开始编号
  pageSize:10(int)
  #0 混合
  #1 电影
  #2 音乐
  #3 取景地
}

Return:
{
  success: ...,
  code: ...,
  message: ...,
  data:
  [
    MovieBrief1, (电影简略信息)
    MovieBrief2,
    ...,
    MovieBriefn
  ]
}
```

5. 获取电影详情

```
Send:
method: get
url:/app/getMovieDetail
params:
{
movieID:'123' (string)
}

Return:
{
  success: ...,
  code: ...,
  message: ...,
  data:Movie
}
```

4. 运行设计

4.1 运行模块的组合

本系统按照实现功能划分模块，每个模块不共享界面，相对独立。每个模块按照流程划分客户端界面，客户端脚本和后台服务器程序。各个模块之间不会共享界面，但共享数据库数据和搜索引擎，后台程序只共享数据库连接和搜索引擎。

4.2 运行控制

(1) 界面是用户直接与系统交互的部分，界面力求简洁而不简陋，能引导用户进行无碍操作。设计时，以在提供用户便捷操作的基础上增加美观度为基准。

(2) 运行控制的条件与限制

本项目的开发要求小组成员足够的参与度，能及时保质保量完成任务。且项目开发过程中可能会有技术上的难点和设备方面的欠缺，需要开发小组 合理利用现有设备和资源，积极查找资料解决问题，在完成项目开发的基础上，同时保证项目的可用性、安全性、可维护性等。

(3) 前台与后台的关系

前台主要展示搜索电影、地点、配乐相关信息、搜索结果内容等显示信息，后台主要负责业务流程，控制前台显示信息，负责与搜索引擎和数据库交互。

4.3 运行时间

用户在做搜索时候前端不断的向 ES 搜索引擎请求数据，会频繁与数据库交互以获取信息，会占用较多的数据库资源。

5. 系统数据结构设计

5.1 逻辑结构设计要点

5.1.1 MySQL 数据库设计：

Table: t_user

Field	Type	Null	Key	Default	Extra	Comment
id	bigint	not null	PRIMARY		auto_increment	用户唯一标志，主键
email	varchar	not null				用户邮箱

Table: t_search_action

Field	Type	Null	Key	Default	Extra	Comment
id	bigint	not null	PRIMARY		auto_in crement	用户行为标志，主键
uid	bigint	not null				用户表主键
content	varchar	not null		“ ”		用户搜索内容
type	int	not null		0		用户搜索类型
time	timestamp			current		记录创建时间

Table: t_click_action

Field	Type	Null	Key	Default	Extra	Comment
id	bigint	not null	PRIMARY		auto_inc rement	用户行为标志，主键
uid	bigint	not null				用户表主键
mid	varchar	not null		“ ”		用户选择的电影主键
genre	varchar	Not null		“ ”		电影分类

name	varchar			“ ”		电影名字
time	timestamp			current		记录创建时间

Table: t_recommend

Field	Type	Null	Key	Default	Extra	Comment
id	bigint	not null	PRIMARY		auto_incr ement	主键
uid	bigint	not null				用户表主键
movies	varchar	not null		“ ”		召回推荐给用户的电影

5.1.2 MongoDB 数据库设计:

Collection: Movie

Field	Type	Comment
_id	ObjectId	主键
name	String	电影名称
director	Array<String>	电影导演列表
country	String	电影出版国家
year	String	电影出版年份
genre	Array<String>	电影分类列表
rating	Number	电影评分
post	String	电影封面 URL

popularity	Number	电影热度（打分人数）
description	String	电影梗概
music	Array<Music>	电影配乐列表
location	Object<Location>	电影取景地
update_time	Timestamp	电影最后修改时间

Object: Music

Field	Type	Comment
name	String	配乐名称
composer	Array<String>	作曲家列表

Object: Location

Field	Type	Comment
images	Array<String>	配图列表
introduction	String	取景情况介绍
visit	Array<String>	取景地列表

5.2 物理结构设计要点

- 1) 数据表项存储在数据库中，通过 SQL 语句访问数据库获取。
- 2) 文件资源存储在磁盘中，通过 SQL 语句访问数据库获取存储位置，访问存储位置获取文件。

6. 系统出错设计

6.1 出错信息

输出信息形式	含义	处理方法
数据库连接失败	由于并发操作的用户数量很大，导致 ES 访问读写率降低；或者 ES 的节点配置不对，导致 ES 连接失败	修改 ES 节点配置 尝试重连
磁盘损坏	由于物理等因素，导致数据库中的数据发生丢失	定期对数据库中的数据 进行备份
数据库读取乱码或 汉字输出为'？'	客户端页面，数据库，数据库 读取过程编码不一致	统一各处的编码方式
词条内容为空	搜索引擎无法获得词条内容	手工检索

6.2 补救措施

(1) 系统备份

定期备份系统数据，当系统数据因不可抗力丢失时，可以启用备份数据。

(2) 分布式部署

将系统部署到不同计算机上，减小硬件损坏造成的数据丢失，削弱影响。

7. 系统维护设计

7.1 概述

对于程序中可能出现异常的部分，全组成员在程序设计伊始便应当设立相关处理方法，由于本程序在使用中需结合多模块进行综合调用，因此对于跨域访问，如前后端通信、数据库连接、神经网络训练、爬虫信息获取等不同模块间交互，应当给予相应的说明，防止因为意外出错导致系统崩溃。

在爬虫部分，由于涉及到两个网站的爬取和信息 Join 的操作，因此在切换爬取网站时，应当更新当前的全局变量，同时设立新的可区分变量值。若前者爬取失败，则本次爬取结束，后者应当接收到错误信息并直接跳过，控制台输出错误信息，同时开启下一次爬取工作。

连接数据库操作时，若数据库连接失败，则应当将错误捕捉并处理，同时尝试重连，防止第一时间导致系统崩溃，可采用的技术细节为 try-catch 语句。

对于系统出错信息，在后端设计时可以使用不同的出错码进行区分，如 401 表示邮箱格式错误、402 表示验证码错误、403 表示未登录、404 表示错误地址等信息，用于前端给出多样化反馈信息。

对于前端的设计，应当努力在满足网页端适配的基础上，拓展可能的多适配方案，如对于手机屏幕的适配等。

系统后期的维护工作后，维护人员应当留下完备可读的系统维护日志，并且程序在出错后也应当自动保存出错信息，供开发人员和其他维护人员查看。

7.2 检测点设计

用户登录功能

测试流程：

1. 进入登陆界面
2. 输入邮箱（合法/不合法）
3. 请求发送验证码
4. 输入验证码（正确/错误）
5. 点击“登录”按钮

用户搜索功能

测试流程：

1. 进入搜索界面
2. 输入搜索信息（电影名、城市名、音乐名、其他）
3. 点击“搜索”按钮
4. 查看搜索结果

查看词云功能

测试流程：

1. 进入主页
2. 查看系统搜索热词

电影推送功能

测试流程：

1. 进入主页
2. 查看系统推送（智能联想）

查看详情功能

测试流程：

1. 点击进入电影详情界面
2. 查看详细信息（封面、电影信息、取景地、配乐等信息）

上/下一篇功能

测试流程：

1. 进入电影详情界面
2. 点击“上一篇”按钮，页面跳转
3. 点击“下一篇”按钮，页面跳转
4. 观察在第一篇（最后一篇）中能否点击上一篇（下一篇）

7.3 系统维护设计

系统维护设计包括硬件资源和软件资源维护，其中软件资源可细分为前端、后端、爬虫、推荐算法、数据库资源。

硬件资源维护：定期清理主机和服务器硬盘垃圾和缓存情况，可根据实际需求更换更高性能的服务资源。

前端资源维护：定期更新所使用的前端框架，提高前端运行效率。

后端资源维护：根据模块耦合程度，提升模块化架构，定期更新所使用的后端框架，提高后端运行效率。

爬虫程序维护：定期爬取网站最新内容，补充信息源。

推荐算法维护：定期调整训练参数，以降低 loss，提高推荐准确性。

数据库资源维护：定期对数据库中重要信息进行备份，如用户信息和搜索记录等，同时检查数据库缓存使用情况。