目录

[1 项目简介 1](#_Toc28330873)

[2 功能性需求 1](#_Toc28330874)

[2.1 搜索电影模块需求 1](#_Toc28330875)

[2.1.1 模块用例图 1](#_Toc28330876)

[2.1.2 根据用户输入内容进行全字段分词检索 1](#_Toc28330877)

[2.1.2.1 功能描述 1](#_Toc28330878)

[2.1.2.2 用例规约 2](#_Toc28330879)

[2.1.3 根据关键词以及分类信息精确检索 2](#_Toc28330880)

[2.1.3.1 功能描述 2](#_Toc28330881)

[2.1.2.2 用例规约 3](#_Toc28330882)

[2.2推荐模块需求 4](#_Toc28330883)

[2.2.1 模块用例图 4](#_Toc28330884)

[2.2.2 时事热点新闻推荐 4](#_Toc28330885)

[2.2.2.1 功能描述 4](#_Toc28330886)

[2.2.2.2 用例规约 4](#_Toc28330887)

[2.2.2 根据用户历史点击数据推荐电影 5](#_Toc28330888)

[2.2.2.1 功能描述 5](#_Toc28330889)

[2.2.2.2 用例规约 5](#_Toc28330890)

[2.3 登陆注册模块需求 6](#_Toc28330891)

[2.3.1 模块用例图 6](#_Toc28330892)

[2.3.2 用户登陆 7](#_Toc28330893)

[2.3.2.1 功能描述 7](#_Toc28330894)

[2.3.2.2 用例规约 7](#_Toc28330895)

[2.3.3 用户注册 8](#_Toc28330896)

[2.3.3.1 功能描述 8](#_Toc28330897)

[2.3.3.2 用例规约 8](#_Toc28330898)

[2.3.4 找回密码 9](#_Toc28330899)

[2.3.4.1 功能描述 9](#_Toc28330900)

[2.3.4.2 用例规约 9](#_Toc28330901)

[2.4 订阅系统 10](#_Toc28330902)

[2.4.1 模块用例图 10](#_Toc28330903)

[2.4.2 将用户所订阅的频道存入数据库 10](#_Toc28330904)

[2.4.2.1 功能描述 10](#_Toc28330905)

[2.4.2.2 用例规约 10](#_Toc28330906)

[2.5频道推送模块需求 11](#_Toc28330907)

[2.5.1 用例图 11](#_Toc28330908)

[2.5.1.1 功能描述 11](#_Toc28330909)

[2.5.1.2 用例规约 12](#_Toc28330910)

[2.6 电影推荐弹窗模块需求 13](#_Toc28330911)

[2.6.1 模块用例图 13](#_Toc28330912)

[2.6.2随机显示推荐内容弹窗 13](#_Toc28330913)

[2.6.2.1 功能描述 13](#_Toc28330914)

[2.6.2.2 用例规约 14](#_Toc28330915)

[2.6.3 用户关闭后弹窗重置 14](#_Toc28330916)

[2.6.3.1 功能描述 14](#_Toc28330917)

[2.6.2.2 用例规约 14](#_Toc28330918)

[4 系统设计 15](#_Toc28330919)

[4.1 简述 15](#_Toc28330920)

[4.2 系统功能模块 15](#_Toc28330921)

[4.2.1 搜索系统 16](#_Toc28330922)

[4.2.2登陆系统 16](#_Toc28330923)

[4.2.3 订阅系统 17](#_Toc28330924)

[4.2.4 广告系统 18](#_Toc28330925)

[4.2.5 推荐系统 18](#_Toc28330926)

[5 系统架构 19](#_Toc28330927)

[5.1 搜索，推荐，计算系统结构 19](#_Toc28330928)

[5.2登陆系统架构 20](#_Toc28330929)

[5.3 订阅系统 21](#_Toc28330930)

[5.4 广告系统 21](#_Toc28330931)

[5.5 推荐系统 22](#_Toc28330932)

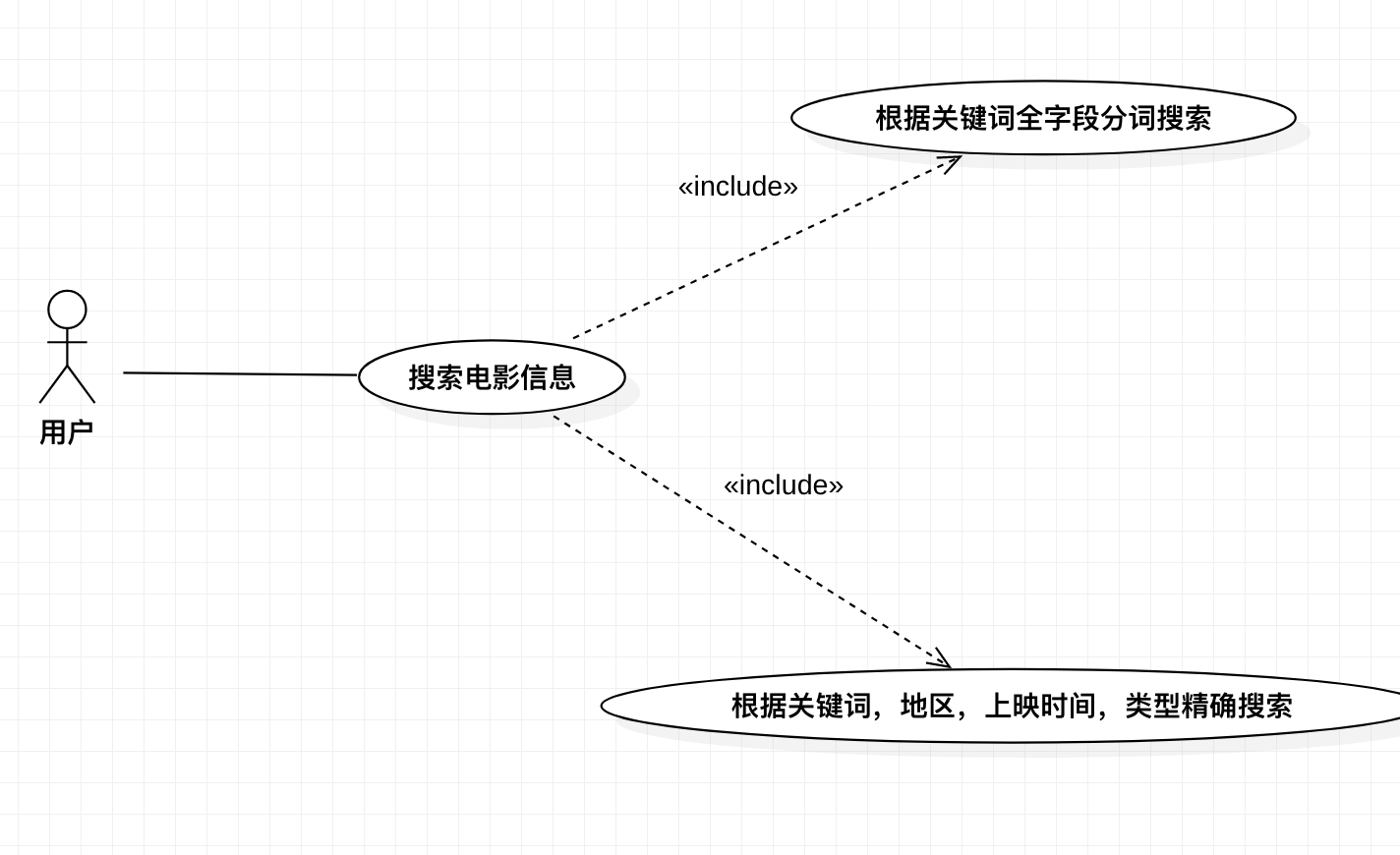
## 1 项目简介

本项目是一个具有企业级功能的电影网站,用户可以在网站上搜索任何关于电影的关键字(包括地区,演员,导演,台词等)然后得到电影的相关信息并且可以得到一些系统推荐的相似电影。用户也可以订阅一些电影的频道,然后在用户下一次登临本网站后可以查看系统的一些推送。当然用户也可以在网站上找到一些趣味相投的好友,并且可以尝试相互认识。作为一个网站,项目自身实现了一些广告的功能来起到一定的对用户的干扰功能。本项目基于分布式系统,需要四个服务器来搭建22个springboot项目的不同的服务。

2.功能性需求

2.1 搜索电影模块需求

## 2.1.1 模块用例图



## 2.1.2 根据用户输入内容进行全字段分词检索

## 2.1.2.1 功能描述

在主页中，用户通过输入想要查询电影的关键词(中文名称，外文名称，导演等)，来进行查询，系统会根据用户输入的关键词在电影的各个字段进行分词查询，如查询成功将会跳转分类查询的页面并且显示查询结果。

## 2.1.2.2 用例规约

|  |  |
| --- | --- |
| 用例名称 | 根据关键词全字段分词检索 |
| 执行者 | 系统用户，搜索系统 |
| 前置条件 | 用户登陆成功 |
| 后置条件 | 跳转页面并显示查询结果 |
| 基本流 | 1. 用户在文本框中输入想要查询电影的关键词 2. 用户点击查询按钮 3. 搜索系统查询相关电影信息 4. 系统跳转页面并显示查询结果 |
| 备选流 | 2a. 若用户没有输入关键词，则提示用户输入 |
| 质量需求 | 无 |
| 设计约束 | 无 |

## 2.1.3 根据关键词以及分类信息精确检索

## 2.1.3.1 功能描述

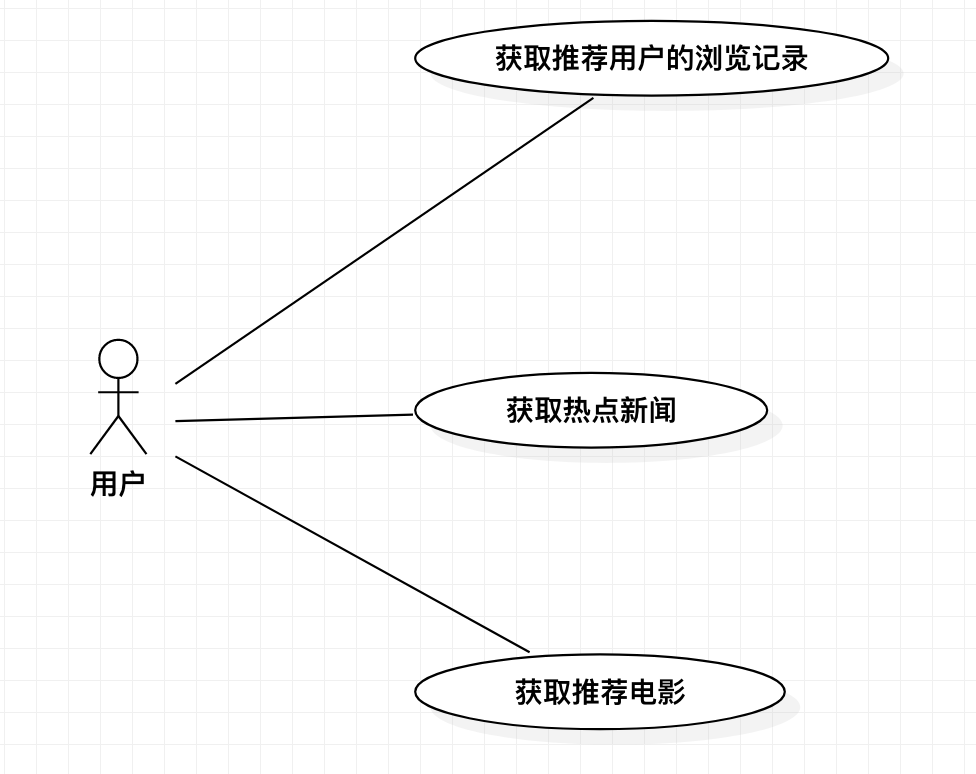
用户通过点击页面中提供的分类按钮来对电影的种类，地区以及上映时间进行选择，并且可以通过输入关键词来进一步精确检索，系统会根据用户的操作来精确搜索电影。

## 2.1.2.2 用例规约

|  |  |
| --- | --- |
| 用例名称 | 根据关键词以及分类信息精确检索 |
| 执行者 | 系统用户，搜索系统 |
| 前置条件 | 用户登陆成功 |
| 后置条件 | 显示查询结果 |
| 基本流 | 1. 用户在文本框中输入想要查询电影的关键词并通过点击分类按钮来选择分类信息 2. 用户点击查询按钮 3. 搜索系统查询相关电影信息 4. 当前页面显示查询结果 |
| 备选流 | 4a. 若数据库中没有相关电影信息，则在页面上显示没有相关信息来提示用户 |
| 质量需求 | 无 |
| 设计约束 | 无 |

## 2.2推荐模块需求

## 2.2.1 模块用例图



## 2.2.2 时事热点新闻推荐

## 2.2.2.1 功能描述

当用户处于分类查询电影的网页时，系统会调用相关的Api来获取当前的热点新闻，并定时更新热点新闻。

## 2.2.2.2 用例规约

|  |  |
| --- | --- |
| 用例名称 | 时事热点新闻推荐 |
| 执行者 | 系统用户，搜索系统 |
| 前置条件 | 用户登陆成功，用户处于分类查询电影页面 |
| 后置条件 | 显示查询结果 |
| 基本流 | 1. 系统调用外部Api来获取热点新闻 2. 网页定时更新热点新闻 3. 当前页面显示结果 |
| 备选流 | 1a 若Api异常，则在网页时显示相关错误信息(暂无热点新闻…) |
| 质量需求 | 无 |
| 设计约束 | 无 |

## 2.2.2 根据用户历史点击数据推荐电影

## 2.2.2.1 功能描述

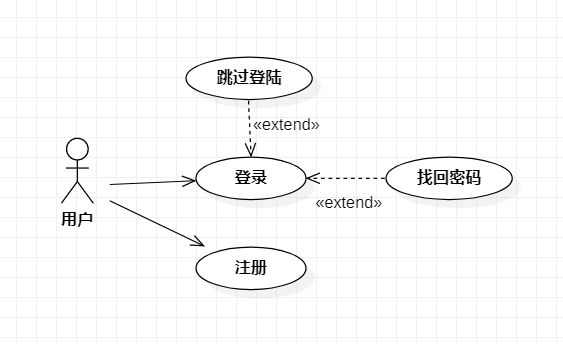
用户每次点击查询结果的标签时，页面都会将点击电影的类型保存在数组中，每隔一段时间都会将数据发送给后端接口，后端接口会对数据进行分析，然后计算得到相似度前9的电影存储在此用户的数据中，当用户下一次搜索时，会给用户推荐相似的电影。

## 2.2.2.2 用例规约

|  |  |
| --- | --- |
| 用例名称 | 根据用户历史数据推荐电影 |
| 执行者 | 系统用户，搜索系统 |
| 前置条件 | 用户登陆成功，用户处于分类查询电影页面 |
| 后置条件 | 显示查询结果 |
| 基本流 | 1. 用户点击查询结果 2. 页面记录用户点击电影的类型 3. 页面定时将保存点击记录的数组发送给推荐系统 4. 推荐系统分析计算得到相似度前9的电影 5. 将要给用户推荐电影的id存储到用户的数据库中 6. 用户点击搜索按钮后，页面拿到此用户数据库中推荐电影的数据，在页面中显示 |
| 备选流 | 6a 若数据库中暂无此用户的推荐记录，那么随即推荐9部电影 |
| 质量需求 | 无 |
| 设计约束 | 无 |

## 2.3 登陆注册模块需求

## 2.3.1 模块用例图



## 2.3.2 用户登陆

## 2.3.2.1 功能描述

用户在访问主页前需要先登陆。系统检测用户输入的用户名密码，如果用户名不存在，或密码错误，则拒绝用户登录。如果用户名和密码匹配，则为用户自动跳转到主页。如果用户直接访问主页，则用户处于游客状态，无法使用功能。

## 2.3.2.2 用例规约

|  |  |
| --- | --- |
| 用例名称 | 用户登陆 |
| 执行者 | 系统用户，登陆注册系统 |
| 前置条件 | 用户准备访问页面 |
| 后置条件 | 跳转到主页 |
| 基本流 | 1. 用户在文本框中输入用户名和密码 2. 用户点击登录按钮 3. 系统验证用户输入 4. 系统跳转到主页 |
| 备选流 | 3a. 若系统验证用户输入的用户名和密码不匹配，则拒绝用户登陆。  3b. 若系统验证用户输入的用户名不存在，则拒绝用户登录。  3c. 若系统验证用户帐号已被封禁，则拒绝用户登录。  3d. 若用户直接访问主页，则用户处于游客状态，无法使用各种功能。 |
| 质量需求 | 无 |
| 设计约束 | 无 |

## 2.3.3 用户注册

## 2.3.3.1 功能描述

用户若没有帐号，可以选择注册帐号，需要输入用户名密码进行注册。如果输入的用户名已存在则拒绝用户注册。

## 2.3.3.2 用例规约

|  |  |
| --- | --- |
| 用例名称 | 用户注册 |
| 执行者 | 系统用户，登录注册系统 |
| 前置条件 | 用户在登陆界面点击注册按钮 |
| 后置条件 | 跳转到登录界面 |
| 基本流 | 1. 用户在文本框中输入注册用的用户名和密码 2. 用户点击注册按钮 3. 系统验证用户输入 4. 页面显示用户注册成功的信息 |
| 备选流 | 3a. 若系统验证用户输入的用户名已存在，则拒绝用户注册。 |
| 质量需求 | 无 |
| 设计约束 | 无 |

## 2.3.4 找回密码

## 2.3.4.1 功能描述

用户若已有帐号但忘记密码，可以找回密码。用户需要输入用户名，系统返回该用户名的密码。如果输入的用户名不存在则找回密码无效。

## 2.3.4.2 用例规约

|  |  |
| --- | --- |
| 用例名称 | 找回密码 |
| 执行者 | 系统用户，登录注册系统 |
| 前置条件 | 用户在登陆界面点击找回密码按钮 |
| 后置条件 | 跳转到登录界面 |
| 基本流 | 1. 用户在文本框中输入要找回密码的用户名  2. 用户点击找回密码按钮  3. 系统验证用户输入  4. 页面为用户显示该用户名的密码 |
| 备选流 | 3a. 若系统验证用户输入的用户名不存在，则返回用户名不存在的错误信息。 |
| 质量需求 | 无 |
| 设计约束 | 无 |

## 2.4 订阅系统

## 2.4.1 模块用例图



## 2.4.2 将用户所订阅的频道存入数据库

## 2.4.2.1 功能描述

在页面中，用户选择下拉页框可以选择自己想要订阅的频道，点击提交之后，系统会将该频道和用户的id存入mysql数据库中。

## 2.4.2.2 用例规约

|  |  |
| --- | --- |
| 用例名称 | 用户订阅频道 |
| 执行者 | 系统用户，订阅系统 |
| 前置条件 | 用户登陆成功 |
| 后置条件 | 下次登陆时显示订阅推送 |
| 基本流 | 1. 用户下拉框中选择自己想要订阅的频道 2. 用户点击提交按钮 3. 订阅系统将数据存入数据库 4. 下次登陆进入页面时显示对应频道的订阅推送 |
| 备选流 | 无 |
| 质量需求 | 无 |
| 设计约束 | 无 |

## 2.5频道推送模块需求

## 2.5.1 用例图



## 2.5.1.1 功能描述

用户在登陆成功时会执行推送服务功能，将推送数据传给订阅系统，然后在前端渲染出来。

## 2.5.1.2 用例规约

|  |  |
| --- | --- |
| 用例名称 | 根据订阅频道调用api进行推送 |
| 执行者 | 系统用户，推送系统，订阅系统 |
| 前置条件 | 用户登陆成功 |
| 后置条件 | 显示推送结果 |
| 基本流 | 1. 用户之前已经进行了订阅操作 2. 用户登陆 3. 进入页面，显示推送结果 |
| 备选流 | 4a. 若之前用户没有订阅数据，则不显示推送数据 |
| 质量需求 | 无 |
| 设计约束 | 无 |

## 2.6 电影推荐弹窗模块需求

## 2.6.1 模块用例图



## 2.6.2随机显示推荐内容弹窗

## 2.6.2.1 功能描述

网站管理人希望间歇性推荐新电影给用户，以增加信息提供量和提供途径。为了适应不同用户的爱好需求，给弹窗内容附加随机性，来确保各类电影都有可能出现在弹窗中，增加其符合用户喜好的概率。

## 2.6.2.2 用例规约

|  |  |
| --- | --- |
| 用例名称 | 调用弹窗接口 |
| 执行者 | 网站管理 |
| 前置条件 | 页面各项设置完成 |
| 后置条件 | 弹窗及内容显示 |
| 基本流 | 1. 管理员设置前端代码和样式 2. 管理员编写ajax的post请求 3. 接收返回的json数据 4. 页面上显示弹窗及电影推荐内容 |
| 备选流 | 3a. 若返回失败，则控制台报错 |
| 质量需求 | 无 |
| 设计约束 | 无 |

## 2.6.3 用户关闭后弹窗重置

## 2.6.3.1 功能描述

用户可能不喜欢推荐的影片，并将其关闭后，根据管理员做的脚本设置，将在若干秒（由管理员决定）后重新发送请求，并再次显示弹窗及随机内容。

## 2.6.2.2 用例规约

|  |  |
| --- | --- |
| 用例名称 | 弹窗重置 |
| 执行者 | 系统用户 |
| 前置条件 | 用户关闭弹窗 |
| 后置条件 | 延迟一段时间后重新显示弹窗 |
| 基本流 | 1. 用户通过按钮关闭了弹窗 2. 前端自动发起请求 3. 接收重新获得的新数据 4. 弹窗及内容再次弹出 |
| 备选流 | 1a.若用户不关闭弹窗，则不做其他动作  3a.若接收失败，则在控制台报错 |
| 质量需求 | 无 |
| 设计约束 | 无 |

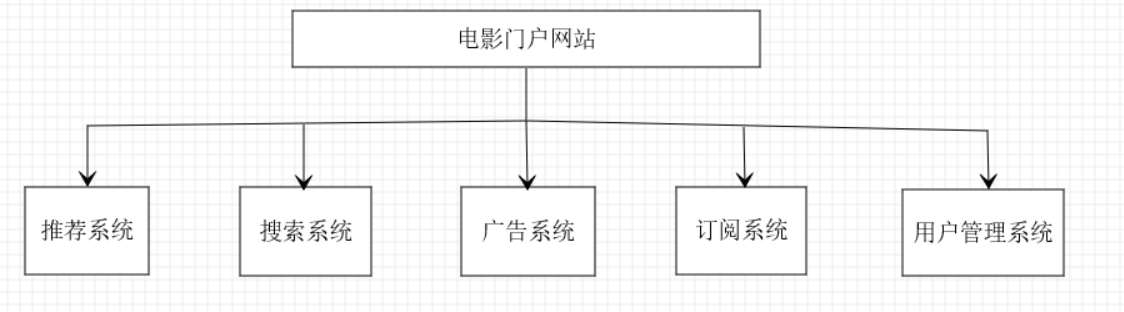
## 4 系统设计

## 简述

此电影信息门户网站主要由五个子系统组成，分别为推荐系统，搜索系统，广告系统，订阅系统以及用户管理系统。

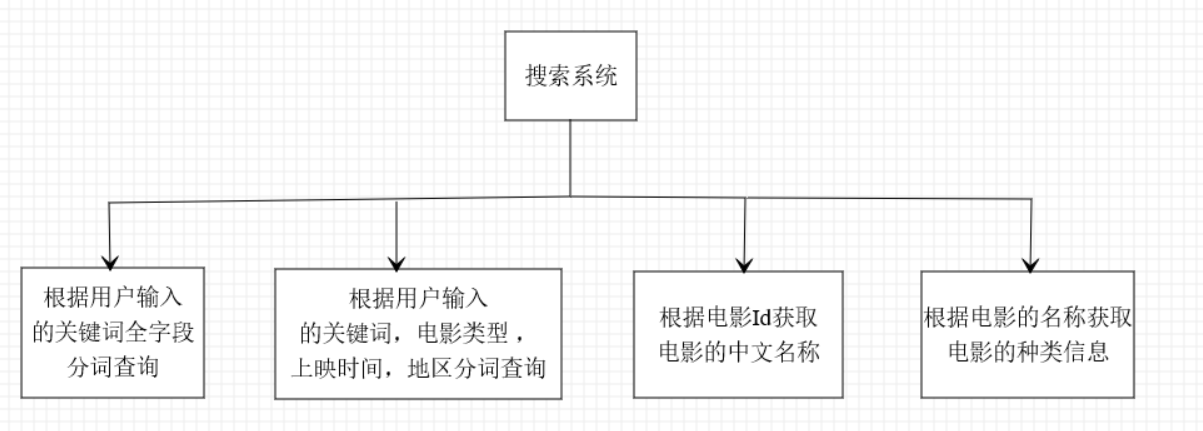
## 4.2 系统功能模块

系统功能模块如下图所示:



## 搜索系统

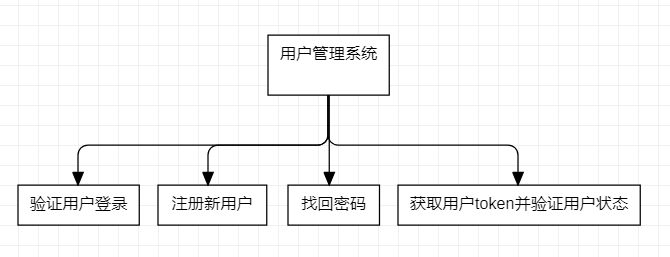
搜索系统功能如下图所示:



1. Elasticsearch根据用户输入的关键词全字段分词查询
2. Elasticsearch根据用户输入的关键词，电影类型，上映时间，地区精确查询电影信息
3. Elasticsearch根据电影Id获取电影的中文名称
4. Elasticsearch根据电影的名称获取电影的类型信息

## 4.2.2登陆系统

用户管理系统主要由两个子系统构成，分别为用户验证系统和控制访问系统。

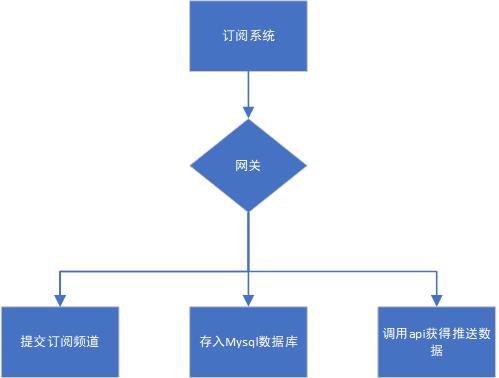


1. 用户验证系统控制用户的登陆注册，密码管理，将用户的相关数据存入elastic search；或从elastic search中查询用户数据。

2. 控制访问系统获取用户的登陆状态，根据用户名从elastic search中获取用户的最新token，并与当前token对比，根据token控制用户的访问。

## 4.2.3 订阅系统

订阅系统总共有五个服务组成，其中Eureca注册中心用于注册服务，网关用于将前端请求集合在一起，订阅服务介绍前端请求和返回页面，存入服务将数据存入数据库，推送服务调用api进行推送。



## 广告系统

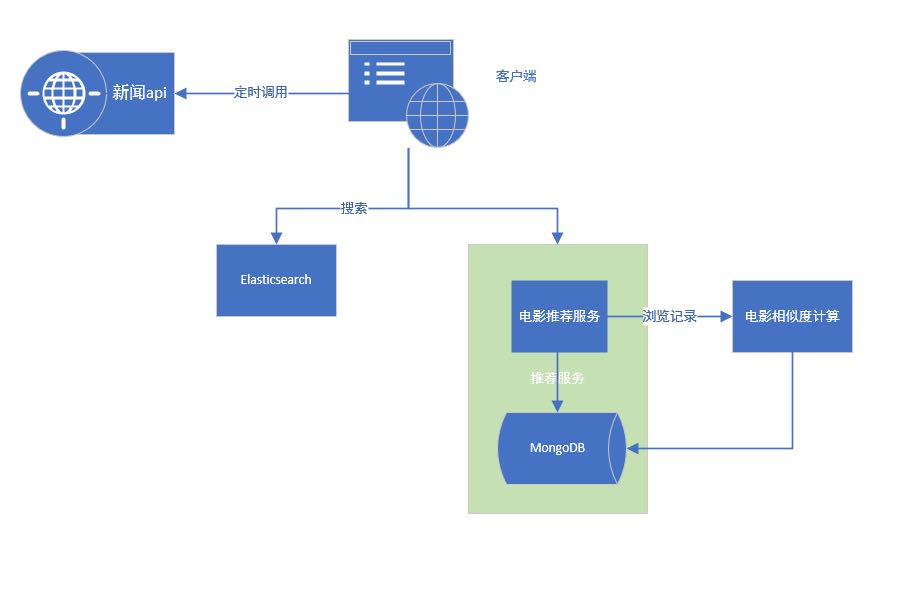
影片推荐弹窗系统主要由两部分组成，即获取随机数据API以及基于计时器的弹窗显示重置模块。该系统设计的初衷源于网站拥有者的需求，为用户提供更多的信息供给，通过弹窗模式，更直观地将新影片数据推送给用户。

## 推荐系统

推荐系统由8个服务组成,信息收集服务用于用户信息的收集,两个队列用于消息的发送,redis缓存服务用于消息缓存(降低系统的开销),mongodb和cassandra服务提供数据库的相关数据服务,hadoop服务用于关键词的计数排序,线上分析服务(爬虫服务,画像建立服务,NLP相关服务)用于相关数据的抓取与分析并且和graphql服务之间建立了两组rpc调用,好友推荐服务用于获取一个爱好相同的用户的用户名。

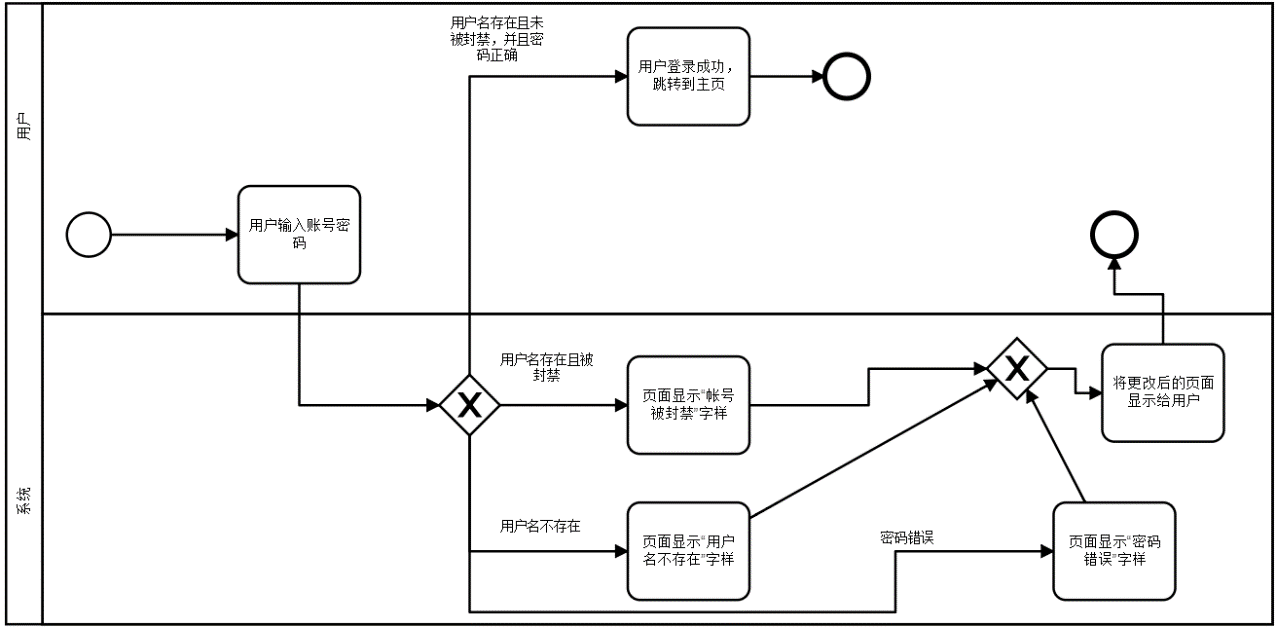
## 5 系统架构

## 5.1 搜索，推荐，计算系统结构



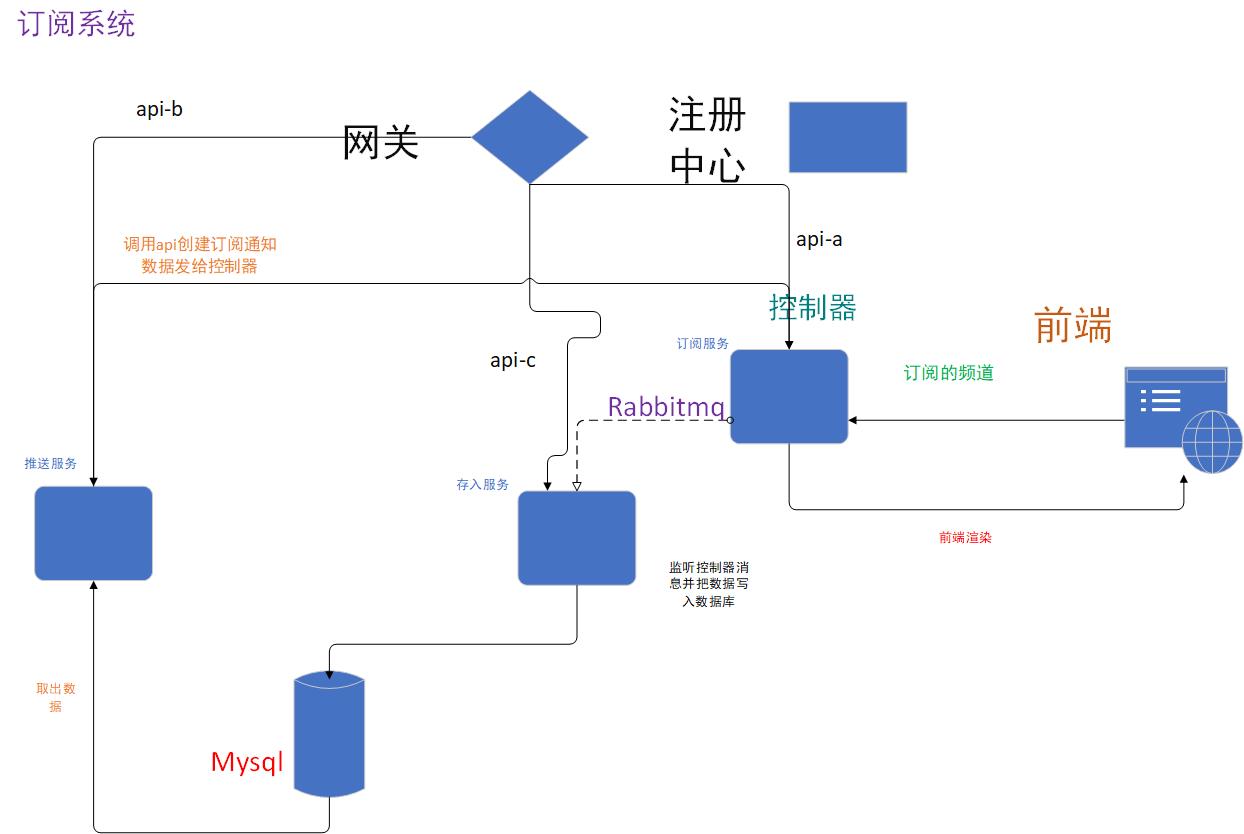
1. 客户端页面通过定时调用新闻Api来获取实时热点新闻并在页面上定时更新显示。
2. 用户通过在页面上输入电影的关键字来获取相关的电影信息，页面通过ajax请求向搜索服务请求数据，搜索服务通过在Elasticsearch搜索引擎中搜索来获取相关信息。
3. 用户在点击电影标签的时候，页面会存储该电影的种类以及名称并定时向推荐服务发送数据，推荐服务会调用电影相似度计算服务来获取为用户推荐的电影信息并存入MongoDB中，当下一次用户搜索电影时会给用户推荐相似度前9的电影信息。

## 5.2登陆系统架构



1. 用户登陆时根据用户输入与elastic search中存储的用户数据对比，并返回登陆结果。只有用户名与密码都存在并互相对应才可登录。
2. 用户注册时根据用户输入与elastic search中存储的用户数据对比，用户名不得重复。
3. 用户登陆后请求资源时要比较用户当前持有的token与elastic search中存储的最新用户token，若不相同则要求用户重新登录。

## 5.3 订阅系统



1.订阅服务主要是接受前端传递过来的id和订阅频道，将之放入rabbitmq消息中间件。

2.然后存入服务从rabbitmq中取出数据存入mysql数据库。

3.推送服务从mysql中取出数据，调用api进行浏览器伪装后获得推送数据，通过rabbitmq传给订阅服务，订阅服务返回订阅展示页面

## 5.4 广告系统

随机数据API（JAVA）

数据API

弹出显示控制

前端页面

Ajax调用

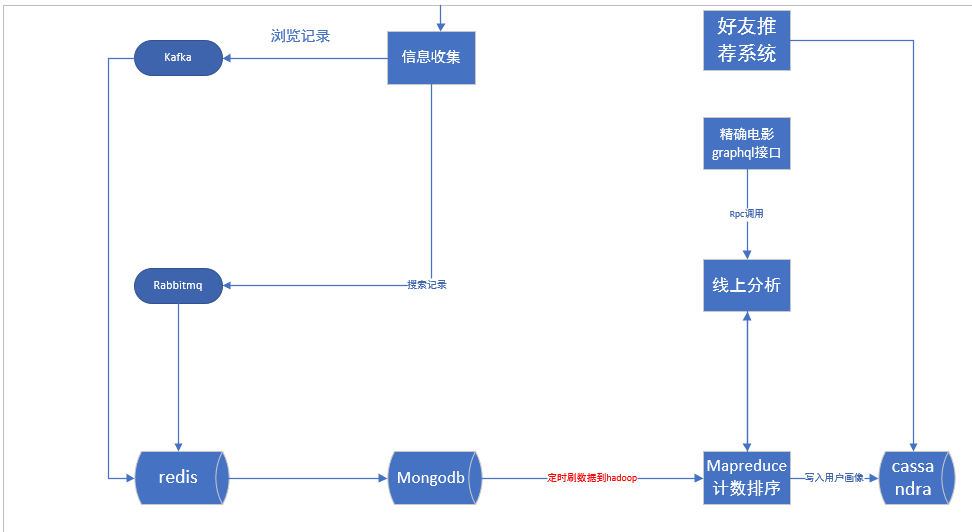
计时器

依赖

1. 客户端基于计时器进行等时间差的判断，当弹窗处于关闭状态时，将其重置显示。
2. 当页面首次加载或被关闭时，通过调用服务端的随机数据API获取随机返回数据，并渲染到弹窗。

## 5.5 推荐系统

系统架构图



信息收集服务只收集用户的搜索记录,这个记录是边缘化的,离散化的,redis缓存这些信息后,将信息组成列表结构写入mongodb。存储在mongodb中的数据是无规律的,纵横判若云泥的,需要先对每一个用户的每一条记录进行分词分析,去除停止词、动词等,并且如果两个词的相似度大于0.75需要去除其中的一个,然后用mapreduce计算框架进行计数排序,建立用户画像后写入cassandra。用户的推荐基于画像的余弦相似度,好友推荐系统会返回相似度大于0.8的所有比较者。