**电子科技大学计算机科学与工程学院**

**标 准 实 验 报 告**

**（实验）课程名称 数据库新技术**

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**实 验 报 告**

**工作方式：（个人/小组）（ 小组 ） 指导教师：牛新征**

**实验地点： 主楼A2-412 实验时间：2018.11.22**

**设计者：（组长） 王洋洋 ，学号： 201852081022**

**（组员） 李斌 ，学号： 201822080137**

**（组员） 何宗奎 ，学号： 201822080111**

**（组员） 杨俊 ，学号： 201822080119**

**（组员） 姜斯文 ，学号： 201722060131**

**一、实验室名称： 计算机学院实验示范中心**

**二、实验项目名称：数据库设计训练**

**三、实验学时：4学时**

目录

[1任务概述 1](#_Toc531261120)

[.1开发与运行环境 1](#_Toc531261121)

[1.1.1开发环境的配置 1](#_Toc531261122)

[1.1.2运行环境的配置 1](#_Toc531261123)

[1.1.3测试环境的配置 2](#_Toc531261124)

[1.2需求概述 2](#_Toc531261125)

[1.3条件与限制 2](#_Toc531261126)

[2总体设计 2](#_Toc531261127)

[2.1软件描述 2](#_Toc531261128)

[2.2数据流图 3](#_Toc531261129)

[2.3处理流程 4](#_Toc531261130)

[2.4功能结构设计 5](#_Toc531261131)

[2.4功能需求与程序代码的关系 5](#_Toc531261132)

[3接口设计 6](#_Toc531261133)

[3.1外部接口 6](#_Toc531261134)

[3.2内部接口 6](#_Toc531261135)

[4数据结构设计 8](#_Toc531261136)

[4.1逻辑结构设计 8](#_Toc531261137)

[4.2物理结构设计 8](#_Toc531261138)

[4.3数据结构与程序代码的关系 9](#_Toc531261139)

[5数据库设计 9](#_Toc531261140)

[6界面设计 10](#_Toc531261141)

# 1任务概述

系统的用户是普通电影爱好者以及需要对电影进行数据分析报的自媒体等，对用户来说主要需求如下：

票房最高的10部电影、每月电影上映数、每月电影票房、电影类型分类，并且对数据能够可视化感知，通过给出图表实现。另外用户在搜索一部电影后，系统能根据电影的类型，推荐相应的电影。

## .1开发与运行环境

1. 客户端的Web浏览器：

要求支持Javascript

2. 服务器端：

node.js，MongoDB

### 1.1.1开发环境的配置

|  |  |  |
| --- | --- | --- |
| 类别 | 标准配置 | 最低配置 |
| 计算机硬件 | CPU2.0G以上，内存1G以上 | CPU2.0G以上，内存1G以上 |
| 计算机软件 | node.js，MongoDB，python3.0，npm | node.js，MongoDB |
| 网络通信 | 无特殊需求 | 无特殊需求 |
| 其它 | 无特殊需求 | 无特殊需求 |

### 1.1.2运行环境的配置

|  |  |  |
| --- | --- | --- |
| 类别 | 标准配置 | 最低配置 |
| 计算机硬件 | CPU2.0G以上，内存1G以上 | CPU2.0G以上，内存1G以上 |
| 软件 | node.js，MongoDB | node.js，MongoDB |
| 网络通信 | 无特殊需求 | 无特殊需求 |
| 其它 | 无特殊需求 | 无特殊需求 |

### 1.1.3测试环境的配置

硬件：

CPU Intel Core 1.6 G，内存1.5G

软件：

Window10操作系统，node.js，MongoDB、python3.0，adminmongo（mongodb可视化工具）。

## 1.2需求概述

本系统主要通过分布式数据库结构实现数据的访问与存储。

实现的功能包括：对猫眼电影进行数据分析、数据可视化展示。

为降低开发难度，缩短开发时间，对电影数据管理系统的功能进行了精简，提炼了以下系统所必需实现的功能，具体功能如下：

1. 查询展示票房最高的10部电影、
2. 查询展示每月电影上映数
3. 查询展示每月电影票房
4. 查询展示电影类型分类

…….

## 1.3条件与限制

一个优秀的电影分析系统，应提供更强大的数据表现形式与分析功能，如更加丰富自由的数据输出，用户可以自由定义票房top20或者top50等，以及更加多样的电影信息如最佳影评、电影资源下载等功能。另外由于数据并没有严格的保密措施，安全性不能得到有效保障。

# 2总体设计

## 2.1软件描述

Web层

BLL层

Model层

Database

图1-1 系统结构

电影分析系统设计采用三层结构，如图1-1所示。说明如下：

1、Web层，爬取猫眼电影数据，展示数据结果

2、BBL层，即业务逻辑层，主要负责业务逻辑的处理；

3、Model层，即实体类，主要负责存储复杂的数据，封装各个类；

## 2.2数据流图

1.顶层数据流图

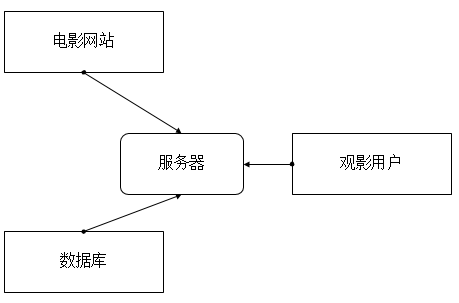


图1-2顶层数据流图

2.一层数据流图

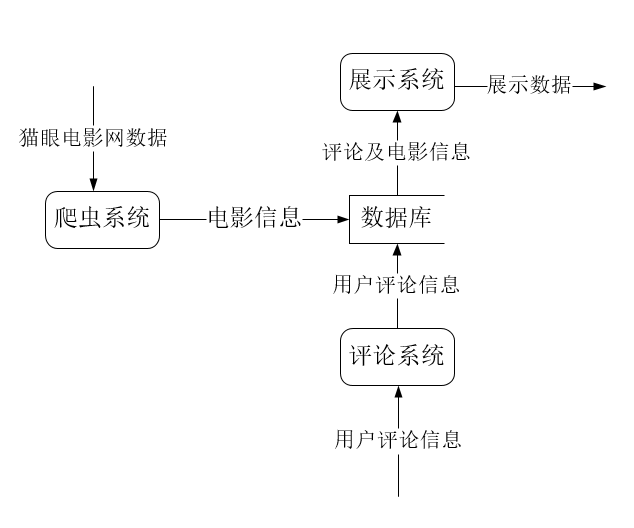


图1-3 一层数据流图

## 2.3处理流程

处理流程如图2-1所示

数据访问层

业务逻辑处理层处理

Web前端输入输出

MongoDB

用户表

电影数据表

……….

图2-1 处理流程

## 2.4功能结构设计

系统主要分为管理员和用户，两种角色都有各自的不同操作，具体见下系统框架结构图：

电影分析系统

用 户

管理员

每月票房

电影上映数

电影类型

安全维护

错误处理

数据查询

…

电影推荐

图2-2 功能结构

## 2.4功能需求与程序代码的关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 需求 代码 | 程序模块 1 | 程序模块 2 | … … | 程序模块 m |
| 功能需求 1 |  |  |  |  |
| 功能需求 2 |  |  |  |  |
| … … |  |  |  |  |
| 功能需求 n |  |  |  |  |

# 3接口设计

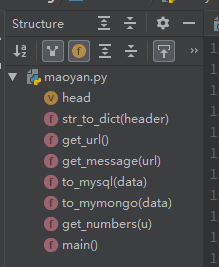
## 3.1外部接口

1．用户界面

按Web软件用户界面的规范来设计，在用户输入地址链接后，web界面将展示相应的电影分析数据。

## 3.2内部接口

1. 爬虫内部接口



1. 图3-1 爬虫内部接口
2. 数据库内部接口

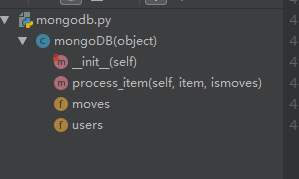


图3-2 数据库内部接口

3.服务器接口

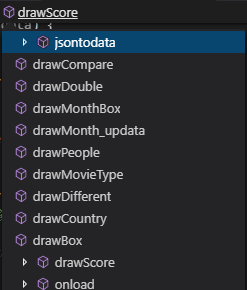
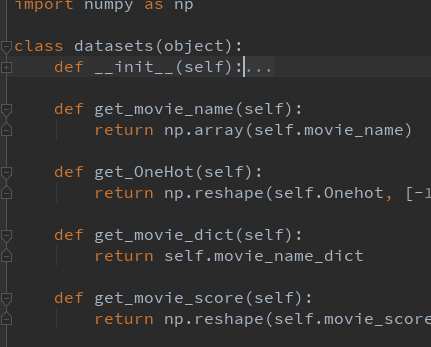
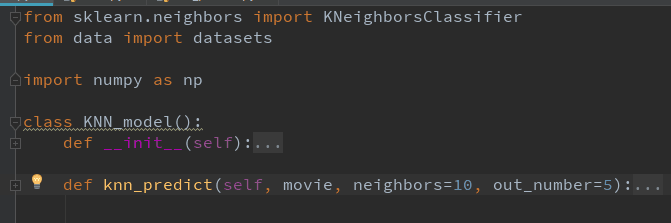


图3-3 服务器接口

4. kNN实现电影推接口





# 4数据结构设计

## 4.1逻辑结构设计

电影分类和电影的关系是一对多，一个电影类型有多个电影对应。

电影分类：分类ID（主键）、类型（喜剧，动作 ……）

电影：电影ID（主键）、分类ID（外键）、票房……)

类型

ID ID

电影分类

电 影

图 4-1 逻辑结构图

## 4.2物理结构设计

用户通过浏览器向服务器发出查询请求，服务器处理用户请求然后向数据库查询数据，数据库将从猫眼电影爬爬取的数据通过json数据格式返回给服务器，然后服务器将数据通过json数据格式将数据返回浏览器。如下图4-2所示。

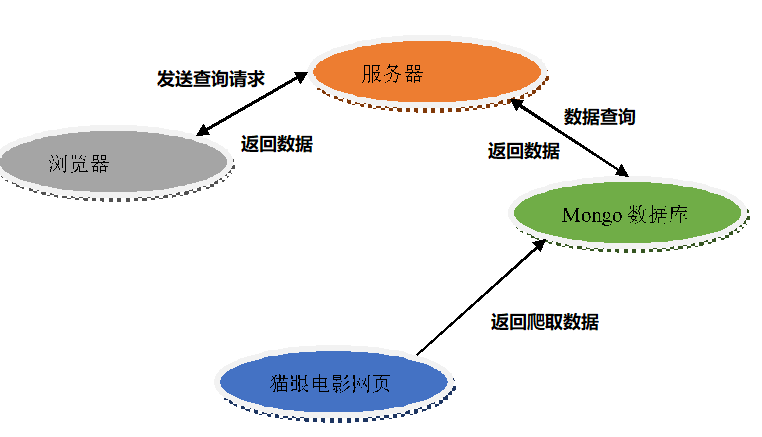


图 4-2 物理结构

## 4.3数据结构与程序代码的关系

# 5数据库设计

数据库的设计如下图5-1 所示。

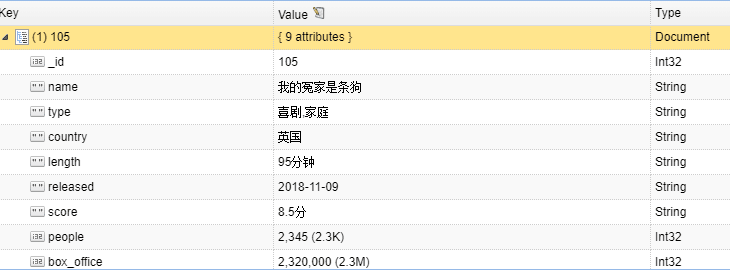


图5 -1 数据库设计

# 6界面设计

本系统界面的组织关系按照角色类别可以如下图6-1分类：

图6-1 界面设计

登陆页面：提供所有角色的登录操作。

管理员主界面：数据库查询与维护

用户主界面：查看电影数据的可视化分析