**华中科技大学计算机科学与技术学院**

**软件工程课程设计报告**

题目: 科研项目信息管理系统

专 业： 计算机科学与技术

班 级：计算机科学与技术（校交）1601

学 号： U201610136

姓 名： 朱晓光

成 绩：

指导教师： 朱建新

**完成日期：2018年10月14日**

目录

[目录 3](#_Toc530245632)

[1 引言 5](#_Toc530245633)

[2 系统分析 5](#_Toc530245634)

[2.1 系统功能分析 5](#_Toc530245635)

[2.1.1 基本信息的录入、修改和删除功能 5](#_Toc530245636)

[2.1.2 基本信息的查询功能 5](#_Toc530245637)

[2.1.3 数据统计功能 6](#_Toc530245638)

[2.1.4 数据存储功能 7](#_Toc530245639)

[2.2 数据分析 7](#_Toc530245640)

[2.2.1 数据流程图 7](#_Toc530245641)

[2.2.2 数据字典 7](#_Toc530245642)

[3 系统总体设计 10](#_Toc530245643)

[3.1 用户导航与文件读写 11](#_Toc530245644)

[3.2 数据维护 11](#_Toc530245645)

[3.3 数据查询 12](#_Toc530245646)

[3.4 数据统计 12](#_Toc530245647)

[4 系统详细设计 14](#_Toc530245648)

[4.1 基本数据结构 14](#_Toc530245649)

[4.1.1 院系数据结构 14](#_Toc530245650)

[4.1.2 团队数据结构 15](#_Toc530245651)

[4.1.3 项目数据结构 16](#_Toc530245652)

[4.1.4 用户指针数据结构 17](#_Toc530245653)

[4.1.5 头节点挂载点组数据结构 18](#_Toc530245654)

[4.1.6 教师人数查找条件数据结构 18](#_Toc530245655)

[4.2 基本算法 18](#_Toc530245656)

[4.2.1 用户导航与文件读写 18](#_Toc530245657)

[4.2.2 数据维护 22](#_Toc530245658)

[4.2.3 数据查询 23](#_Toc530245659)

[4.2.4 数据统计 24](#_Toc530245660)

[5 系统实现 25](#_Toc530245661)

[5.1 源程序文件结构 25](#_Toc530245662)

[6 测试报告 26](#_Toc530245663)

[6.1 数据维护模块测试 26](#_Toc530245664)

[6.1.1 数据添加测试 26](#_Toc530245665)

[6.1.2 数据修改 30](#_Toc530245666)

[6.1.3 数据删除 30](#_Toc530245667)

[6.2 数据查询模块测试 32](#_Toc530245668)

[6.2.1 按名称的全部或部分查询院系 32](#_Toc530245669)

[6.2.2 按教师人数查询团队 33](#_Toc530245670)

[6.2.3 按所属团队查询项目 34](#_Toc530245671)

[6.3 数据统计 35](#_Toc530245672)

[6.3.1 院系项目总数统计 35](#_Toc530245673)

[6 项目管理 36](#_Toc530245674)

[6.1 软件规模度量 36](#_Toc530245675)

[6.2 工作量估算 37](#_Toc530245676)

[6.3 软件开发时间估算 37](#_Toc530245677)

[6.4 工程进度计划 37](#_Toc530245678)

[7 课程学习总结 38](#_Toc530245679)

[7.1 课程学习体会 38](#_Toc530245680)

[7.2 课程教学建议 38](#_Toc530245681)

该课程设计项目在GitHub上的地址：

https://github.com/smdsbz/curriculum-design-1

1 引言

大学中为了方便对科研项目进行管理，各个院系都成立有独立的研究团队，每个团队由若干个教室组成。科研项目的立项、申报、研究、结题全过程都是以团队为单位来开展的。

但由于随着学校规模扩大以及越来越多新项目的立项，留存记录的科研项目数量剧增，科研项目管理的成本越来越高，人工管理不再是可以持续的解决方案，需要借助计算机来完成管理工作。通过计算机来管理科研项目，可以大大提高添加、查找、修改、删除等工作的效率，同时也可以降低在这些操作过程中出现人为疏忽的可能性，更使对科研项目的快速生成综合数据分析结果成为可能。

此次研制的科研信息管理系统的核心数据操作部分采用C语言实现，并由使用  
Python 3编码实现的Web服务端调用其操作接口。管理系统将在Linux环境下运行。

2 系统分析

2.1 系统功能分析

经过调研并对使用需求进行分析，确定了科研信息管理系统的基本功能和运行方式：数据管理员通过网页端向管理系统发送操作指令与数据内容，管理系统在对所管理数据进行操作后，向网页端传回对应操作结果的展示页面，并等待下一次指令输入。

下面给出该系统的具体功能要求。

2.1.1 基本信息的录入、修改和删除功能

系统应实现对基础数据信息的录入、修改和删除功能。在信息录入时，应简明地标识出所有用户应当提供的数据项，一次输入就能够完成对象的添加；同时，在添加对象的时候应该还提供信息校验功能，保证添加的数据项合理、正确，能够形成正确的层级、归属关系。

2.1.2 基本信息的查询功能

系统应该实现对基础数据信息的查询功能，提供按多种不同条件分别进行查询的方式，具体包括：

（1）按院系负责人姓名查询院系信息

（2）按院系名称的全部或一部分查询院系信息

（3）按团队名称的全部或一部分查询团队信息

（4）以教师人数为条件查询团队信息

（5）按项目编号查询项目信息

（6）按所属团队查询项目信息

上述查询结果中，如果有多条信息符合条件，则需逐条显示并提供进行选择的功能。

2.1.3 数据统计功能

在以上三种基础数据信息的基础上，提供多方面的数据统计功能，具体包括：

（1）统计各院系教师总数、研究生总数以及研究生与教师的人数比（保留两位小数），按学生教师人数比值降序排序后输出。

表2.3.1 院系学生教师人数统计表示例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **院系名称** | **教师总数** | **学生总数** | **学/教比值** |
| 1 | 计算机学院 | 78 | 799 | 10.24 |
| 2 | 物理学院 | 45 | 456 | 10.13 |
| ... |  |  |  |  |

（2）统计某年度各院系科研项目数、973项目数、863项目数以及科研总经费，按科研项目数降序排序后输出。

表2.3.2 院系科研项目数统计表示例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **院系名称** | **项目总数** | **973项目数** | **863项目数** | **科研经费** |
| 1 | 计算机学院 | 3 | 2 | 0 | 15.84 |
| 2 | 物理学院 | 1 | 0 | 1 | 12.30 |
| ... |  |  |  |  |  |

（3）统计历年来类别为国家自然科学基金的科研项目数最多的10个科研团队， 按项目数降序排序后输出科研团队名称、国家自然科学基金项目数以及项目经费总数。

表2.3.3 国家自然科学基金项目团队统计表示例

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **团队名称** | **NSFC项目数** | **项目经费** |
| 1 | 火箭队 | 1 | 2.30 |
| 2 | 银河队 | 3 | 12.30 |
| ... |  |  |  |

（4）统计科研项目数和教师人数的比值最高的5个科研团队，按比值（保留两位小数）降序排序后输出科研团队名称、教师人数、科研项目数、项目数与教师人数比值。

表2.3.4 团队教师分配统计表示例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **团队名称** | **教师人数** | **项目总数** | **项目/教师比值** |
| 1 | 火箭队 | 2 | 2 | 1.00 |
| 2 | 银河队 | 4 | 1 | 4.00 |
| ... |  |  |  |  |

（5）统计项目平均经费最多的5个院系，按平均经费降序排序后输出院系名称、项目总数、总科研经费、项目平均经费。

表2.3.5 院系平均项目经费统计表示例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **院系名称** | **项目总数** | **科研经费** | **项目平均经费** |
| 1 | 计算机学院 | 1 | 12.30 | 12.30 |
| 2 | 物理学院 | 3 | 15.84 | 5.28 |
| ... |  |  |  |  |

2.1.4 数据存储功能

系统基本信息在程序运行时，以可动态扩展的链表的形式存储在内存中；同时，在硬盘上以文件的形式实现长期保存。程序退出时会将内存中的数据与硬盘上的数据进行同步，从而保证数据的一致性以及安全性。

2.2 数据分析

科研信息管理系统中需要管理多项数据，这些数据之间可能相互关联，因此有必要在进行具体系统详细设计之前对系统所需要处理的数据进行整理分析，以厘清这些数据之间的层级关系。

2.2.1 数据流程图

根据2.1节中对系统功能的分析，发现系统中所有操作都是围绕院系、团队以及项目的数据进行的，且对三种数据类型的操作基本相同，故可以得到以下的通用数据流程图。

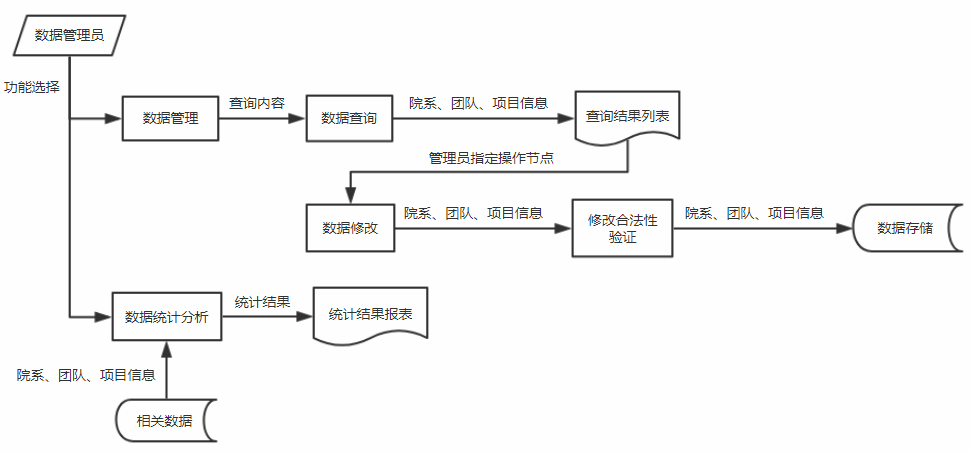


图2.1 科研信息管理系统数据流程图

2.2.2 数据字典

科研信息管理系统的基本信息主要包含院系基本信息、科研团队基本信息以及科研项目基本信息一共3类，系统中任何增、删、查、改以及统计的操作都是围绕这些信息进行的。下面给出这3类信息具体所需要包含的信息字段。

（1）院系基本信息

（1.1） 院系名称

· 简述：院系的名称

· 类型：字符串

· 长度：20

（1.2） 负责人姓名

· 简述：院系负责人的姓名

· 类型：字符串

· 长度：12

（1.3） 联系电话

· 简述：院系负责人的联系电话

· 类型：字符串

· 长度：15

（2）科研团队基本信息

（2.1） 团队名称

· 简述：科研团队的名称

· 类型：字符串

· 长度：30

（2.2） 负责人姓名

· 简述：团队负责人的姓名

· 类型：字符串

· 长度：12

（2.3） 教师人数

· 简述：团队中教师的人数

· 类型：正整数

（2.4） 研究生人数

· 简述：团队中研究生的人数

· 类型：正整数

（2.5） 团队所属院系

· 简述：团队所属院系的名称

· 类型：字符串

· 长度：20

（3）科研项目基本信息

（3.1） 项目编号

· 简述：科研项目的唯一编号

· 类型：字符串

· 长度：15

（3.2） 项目类别

· 简述：项目所属类别

· 类型：单字符

· 范围：‘1’-‘5’

项目类别代码由‘1’到‘5’分别为：“973计划项目”、“国家自然科学基金项目”、“863计划项目”、“国际合作项目”、“横向项目”。

（3.3） 起始时间

· 简述：项目立项时间

· 类型：字符串

· 长度：7

起始时间字符串遵循“yyyy/mm”的格式。

（3.4） 项目经费

· 简述：项目申请经费额度

· 类型：正实数

（3.5） 负责人姓名

· 简述：项目负责人姓名

· 类型：字符串

· 长度：12

（3.6） 项目所属团队

· 简述：项目负责团队的名称

· 类型：字符串

· 长度：30

为了进一步阐述这些数据之间的层级关系，这里给出如图2.2所示的系统三种基本数据的ER图。



图2.2 科研信息管理系统ER图

3 系统总体设计

根据系统需求分析结果，将整个系统在宏观上分解成了如图3.1所示的4个功能模块，图中每个模块的具体功能与设计如下所述。

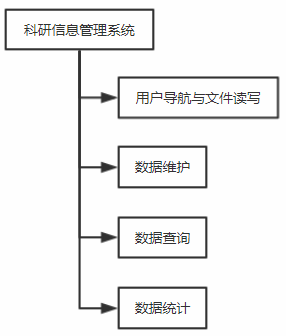


图3.1 科研信息管理系统总体结构

3.1 用户导航与文件读写

用户导航与文件读写模块包括一系列与系统启动时环境建立与系统结束运行时数据保存的操作。它进一步划分为如图3.2所示的3个子模块。

（1）数据加载子模块：用于将存储在硬盘中的数据读取到内存中，建立数据结构。

（2）数据保存子模块：在程序运行过程中，将内存中的数据以特定格式备份到硬盘上。

（3）用户导航子模块：用于在程序运行的各个阶段向用户展示当前操作结果与可用操作，使用户能够在系统的不同子功能模块中进行切换，并能够方便地导航到任意数据节点。

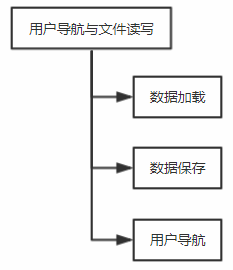


图3.1 用户导航与文件读写模块结构

3.2 数据维护

数据维护模块负责对三种基础数据信息的录入、修改和删除功能，需要保证数据的准确性、完整性和有效性。该模块按信息种类进一步划分为如图3.2所示的3个子模块。

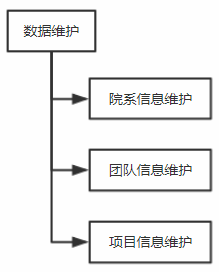


图3.2 数据维护模块结构

院系信息维护子模块、团队信息维护子模块和项目信息维护子模块：三个模块分别负责用户对三种基本信息的录入、修改、删除等操作。在对数据操作过程中，该模块确保基础数据的正确性，以保障用户在操作以上三种基础信息的时候不会破坏数据之间的依赖关系。

3.3 数据查询

数据查询模块提供对三种基础数据按多种条件进行查询的功能。该模块按信息种类分为如图3.3所示的3个子模块。

（1）院系信息查询子模块：用于提供两种方式查询院系基本信息——一种是按院系负责人姓名进行精确查询；另一种是按院系名称的全部或部分进行模糊查询。列出所有符合条件的查询结果后，用户能够选择其中一个或者退出查询。

（2）团队信息查询子模块：用于提供两种方式查询团队基本信息——一种是按团队名称的全部或一部分进行模糊查询；另一种是以教师人数为条件进行模糊查询。

（3）项目信息查询子模块：由于团队信息查询子模块可以定位到项目所属的团队，该模块只提供系统需求中另一种方式查询团队基本信息——按项目编号进行精确查询。

3.4 数据统计

数据统计模块提供对三种基础数据进行多方面统计的功能。该模块按统计项目划分为如图3.4所示的5个子模块。

（1）院系教师配置统计子模块：用于统计各院系的教师总数与学生总数，并由此计算学生与老师人数比（保留两位小数），按人数比降序排序后输出统计信息。

（2）院系科研项目统计子模块：用于统计某年度各院系科研项目总数、973项目数、863项目数以及科研总经费，按科研项目总数降序排序后输出统计信息。

（3）国家自然科学基金项目统计子模块：用于统计各团队国家自然科学基金项目总数与该类项目经费，按项目数降序排序后输出排名前10团队的统计信息。

（4）团队教师配置统计子模块：用于统计团队项目总数与教师总人数，并由此计算项目数与教师人数比值（保留两位小数），按比值降序排序后输出排名前5团队的统计信息。

（5）院系项目经费统计子模块：用于统计院系项目数与科研总经费，并由此计算项目的平均经费（保留两位小数），按比值降序排序后输出排名前5院系的统计信息。

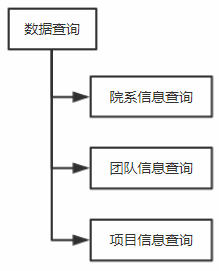


图3.3 数据查询模块的子模块划分

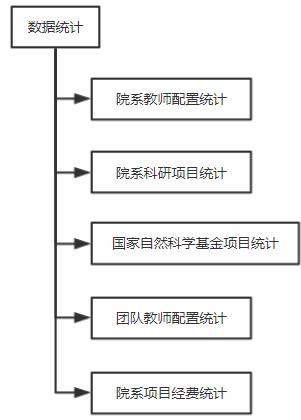


图3.4 数据统计模块的子模块划分

4 系统详细设计

根据上述对系统结构的拆分，可以确定各个模块中具体使用的数据结构与算法。

4.1 基本数据结构

按照要求，系统需要处理的基础信息有三种：院系基本信息、团队基本信息和项目基本信息。这三种信息之前存在这样的关联：院系下属有团队、团队开设项目。根据实现需要，每种基础数据结构有另外的包装结构，此外还增设了用户指针、头节点挂载点组、教师人数查找条件等数据结构。下面分别展示每种数据结构的设计。

4.1.1 院系数据结构

a） 数据文件存储结构

院系基础数据信息如表4.1所示。院系基础数据将以DepartData结构体的形式存储在“DEPART.DAT”文件中。

表4.1 院系基本信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构标识 | | DepartData | |
| **数据项** | **标识** | **类型** | **示例** |
| 院系名称 | name | char[20] | “Computer” |
| 负责人 | manager | char[12] | “Garry” |
| 联系电话 | mobile | char[15] | “13313371337” |

b） 内存存储结构

Depart结构体用于链接在内存中的所有院系数据，形成链表结构。

表4.2 院系节点存储用结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | Depart | |
| **数据项** | **标识** | **类型** |
| 院系数据域 | data | DepartData \* |
| 下一个院系节点 | next | Depart \* |
| 所属团队链头节点 | child\_team\_head | Team \* |
| 所属团队链尾节点 | child\_team\_tail | Team \* |

c） 搜索结果存储结构

DepartWrapper结构体用于返回只含符合搜索条件院系的链表。

表4.3 院系搜索用结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | DepartWrapper | |
| **数据项** | **标识** | **类型** |
| 院系节点 | depart | Depart \* |
| 下一个节点 | next | DepartWrapper \* |

d） 统计结果存储结构

DepartStatWrapper结构体用于返回按要求进行统计、排序后的院系链表。

表4.4 院系统计用结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | DepartStatData | |
| **数据项** | **标识** | **类型** |
| 学生总人数 | student\_num | int |
| 教师总人数 | teacher\_num | int |
| 项目总数 | project\_total | int |
| 973项目总数 | project\_973 | int |
| 863项目总数 | project\_863 | int |
| 科研总经费 | funding | float |

表4.5 院系统计用节点结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | DepartStatWrapper | |
| **数据项** | **标识** | **类型** |
| 院系节点 | depart | Depart \* |
| 统计数据 | stat | DepartStatData |
| 下一节点 | next | DepartStatWrapper \* |

4.1.2 团队数据结构

a） 数据文件存储结构

团队基础数据信息表4.6所示。团队基础数据将以TeamData结构体的形式存储在“TEAM.DAT”文件中。

表4.6 团队基本信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构标识 | TeamData | | |
| **数据项** | **标识** | **类型** | **示例** |
| 团队名称 | name | char[30] | “Rocket” |
| 负责人 | manager | char[12] | “Mark” |
| 教师人数 | teacher\_num | int | 13 |
| 学生人数 | student\_num | int | 25 |
| 所属院系名称 | faculty | char[20] | “Computer” |

b） 内存存储结构

Team结构体用于链接在内存中的所有团队数据成链表。

表4.7 团队节点存储用结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | Team | |
| **数据项** | **标识** | **类型** |
| 团队数据域 | data | TeamData \* |
| 下一节点 | next | Team \* |
| 开设项目链头节点 | child\_project\_head | Project \* |
| 开设项目链尾节点 | child\_project\_tail | Project \* |
| 所属院系节点 | parent\_depart | Depart \* |

c） 搜索结果存储结构

TeamWrapper结构体用于返回只含符合搜索条件团队的链表。

表4.8 团队搜索用结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | TeamWrapper | |
| **数据项** | **标识** | **类型** |
| 团队节点 | team | Team \* |
| 下一节点 | next | TeamWrapper \* |

d） 统计结果存储结构

TeamStatWrapper结构体用于返回按要求进行统计、排序后的团队链表。

表4.9 团队统计用数据结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | TeamStatData | |
| **数据项** | **标识** | **类型** |
| 项目总数 | project\_total | int |
| 国家自然科学基金项目数 | project\_NSFC | int |
| 总经费 | funding | float |

表4.10 团队统计用节点结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | TeamStatWrapper | |
| **数据项** | **标识** | **类型** |
| 团队节点 | team | Team \* |
| 统计数据 | stat | TeamStatData |
| 下一节点 | next | TeamStatWrapper |

4.1.3 项目数据结构

a） 数据文件存储结构

项目基础数据信息如表4.11所示。项目基础数据将以ProjectData结构体的形式存储在“PROJECT.DAT”文件中。

表4.11 项目基本信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构标识 | ProjectData | | |
| **数据项** | **标识** | **类型** | **示例** |
| 项目编号 | id | char[15] | “123684” |
| 项目类别 | type | char | ‘1’ |
| 起始时间 | start\_date | char[8] | “1970/01” |
| 项目经费 | funding | float | 13.2 |
| 负责人 | manager | char[12] | “Alice” |
| 所属团队名称 | team | char[30] | “Rocket” |

表4.12 基本项目信息项目类别代码表

|  |  |
| --- | --- |
| **代码** | **对应项目类别** |
| 1 | 973计划项目 |
| 2 | 国家自然科学基金项目 |
| 3 | 863计划项目 |
| 4 | 国际合作项目 |
| 5 | 横向项目 |

b） 内存存储结构

Project结构体用于链接在内存中的所有项目数据成链表。

表4.13 项目节点存储用结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | Project | |
| **数据项** | **标识** | **类型** |
| 项目数据域 | data | ProjectData \* |
| 下一节点 | next | Project \* |
| 开设团队节点 | parent\_team | Team \* |

c） 搜索结果存储结构

ProjectWrapper结构体用处返回只含符合搜索条件项目的链表。

表4.14 项目搜索用结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | ProjectWrapper | |
| **数据项** | **标识** | **类型** |
| 项目节点 | project | Project \* |
| 下一节点 | next | ProjectWrapper |

院系基本信息表、团队基本信息表与项目基本信息表在内存中构成类似于三向十字交叉链表的结构，如图4.1所示。

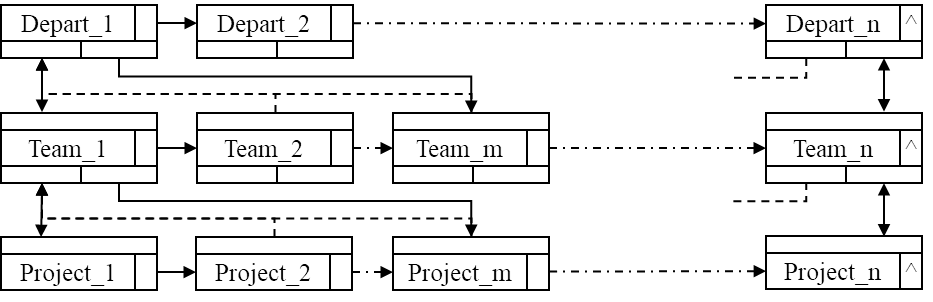


图4.1 基础数据存储链表结构示意图

4.1.4 用户指针数据结构

用户指针用于记录用户当前在上面展示的十字链表数据结构中的位置，方便用户导航功能的实现。

表4.15 用户指针结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | Cursor | |
| **数据项** | **标识** | **类型** |
| 指向对象类型 | type | int |
| 指向对象地址 | val | void \* |

表4.16 用户指针结构体指向对象类型代码表

|  |  |
| --- | --- |
| **代码** | **对应指向对象类型** |
| 1 | 院系 |
| 2 | 团队 |
| 3 | 项目 |
| 0 | 置空 |

4.1.5 头节点挂载点组数据结构

头节点挂载点组的设置旨在方便主程序管理院系链表、团队链表和项目链表的头节点。将该数据在全局申明，可以大大减少在具体实现中向视图函数传递头节点指针的次数。

表4.17 头节点挂载点组结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | MountPoint | |
| **数据项** | **标识** | **类型** |
| 院系链表头节点 | depart\_head | Depart \* |
| 团队链表头节点 | team\_head | Team \* |
| 项目链表投节点 | project\_head | Project \* |

4.1.6 教师人数查找条件数据结构

教师人数查找条件结构体的设置旨在向负责查询团队的函数传递查找条件的时候，减少参数个数，抽象查找条件并同一查询函数调用格式。

表4.18 教师人数查找条件结构体

|  |  |  |
| --- | --- | --- |
| 数据结构标识 | Where | |
| **数据项** | **标识** | **类型** |
| 查询方向 | direction | char[3] |
| 查询阈值 | value | int |

（其中查询方向可能的取值有“<”、“<=”、“=”、“>”、“>=”）

4.2 基本算法

基于对系统功能的分析以及设计，下面展示主体功能实现的算法。

图4.2展示了主程序运行流程。

下面详细展示载入数据部分以及系统功能的选择以及运行部分的设计。

4.2.1 用户导航与文件读写

（1）数据加载

数据加载功能又可以分为如图4.3所示的三个子过程进行：加载院系数据、加载团队数据和加载项目数据。加载三类数据的同时也要正确构建起数据之间的层级结构。

由于在存储的时候是有序的，所以加载时不需要考虑新数据插入的位置，按照“先进先读”的顺序读取即可。loadData()函数中关于用链表链接数据的部分均由另定义的函数（appendDepart()、appendTeam()和appendProject()）实现。

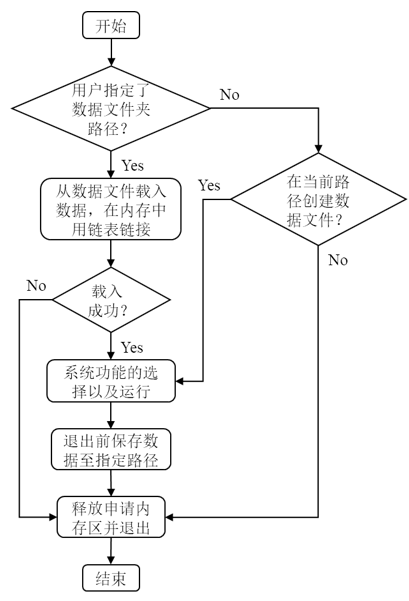


图4.2 主程序运行流程图

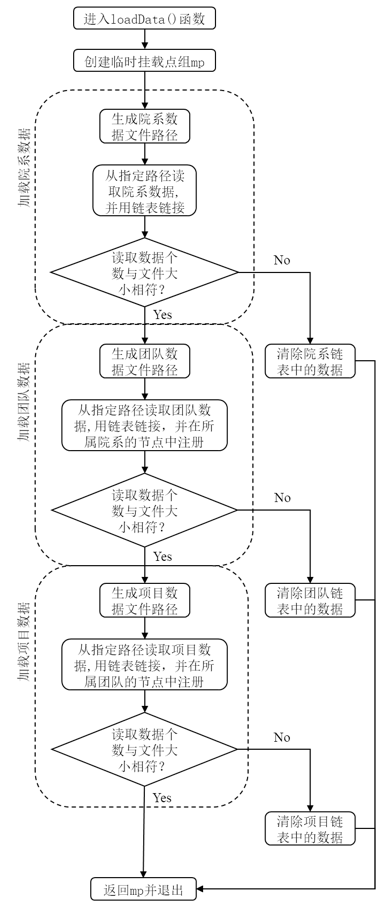


图4.3 数据加载子模块流程图

（2）用户导航

程序将以网页的形式向管理员的操作终端发送当前可用的操作，用户通过鼠标点击下拉列表选择其中的某一项来执行。图4.4展示了该软件的用户导航逻辑。

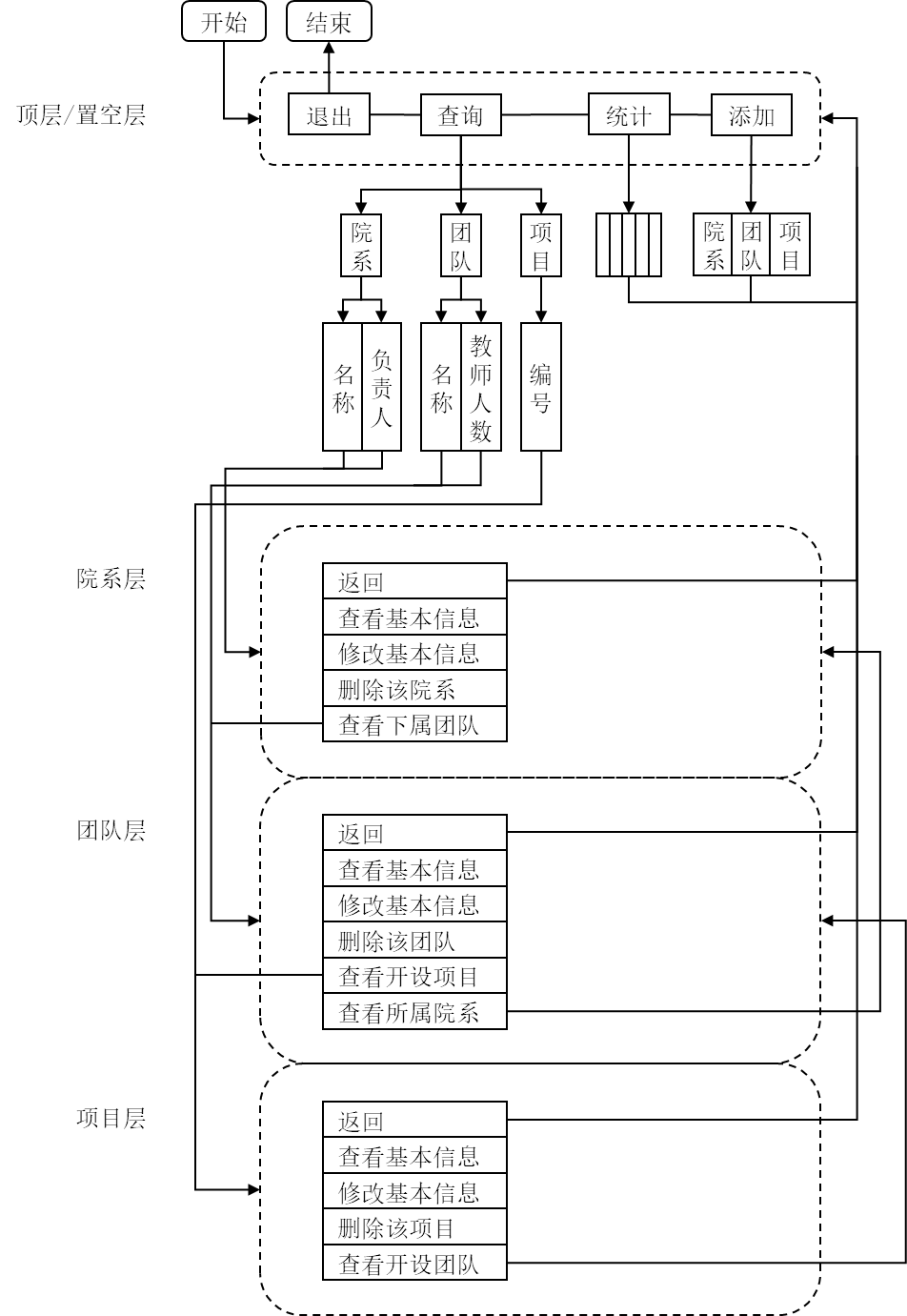


图4.4 用户导航逻辑示意图

（3）数据备份

该部分实现可以通过调用“数据保存”部分程序实现，其流程如图4.5所示。

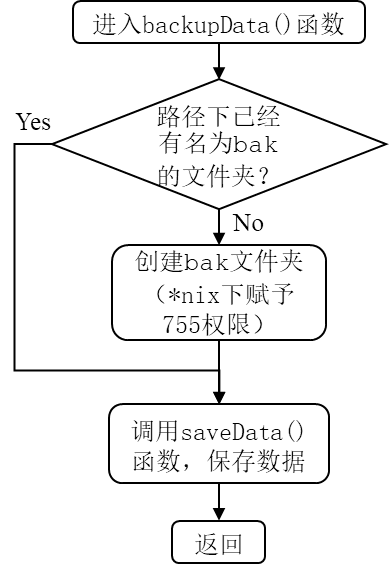


图4.5 数据备份流程图

（4）数据保存与退出系统

数据保存操作流程如图4.6所示。



图4.6 数据保存流程图

退出系统：退出系统部分主要负责释放程序运行中所有申请的内存空间。

4.2.2 数据维护

由于对于院系、团队、项目三种基础数据进行维护操作的重复度很高，在此只给出最具代表性、最复杂的对团队基础数据维护的详细流程。

（1）团队信息添加

为了不破坏已有链表结构的完整性，在新添加团队节点的时候不能只是简单地在团队链表末尾添加节点，而是要找到所属院系节点下的团队节点区块，插入到该区块的尾部。团队信息添加的流程如图4.7所示。

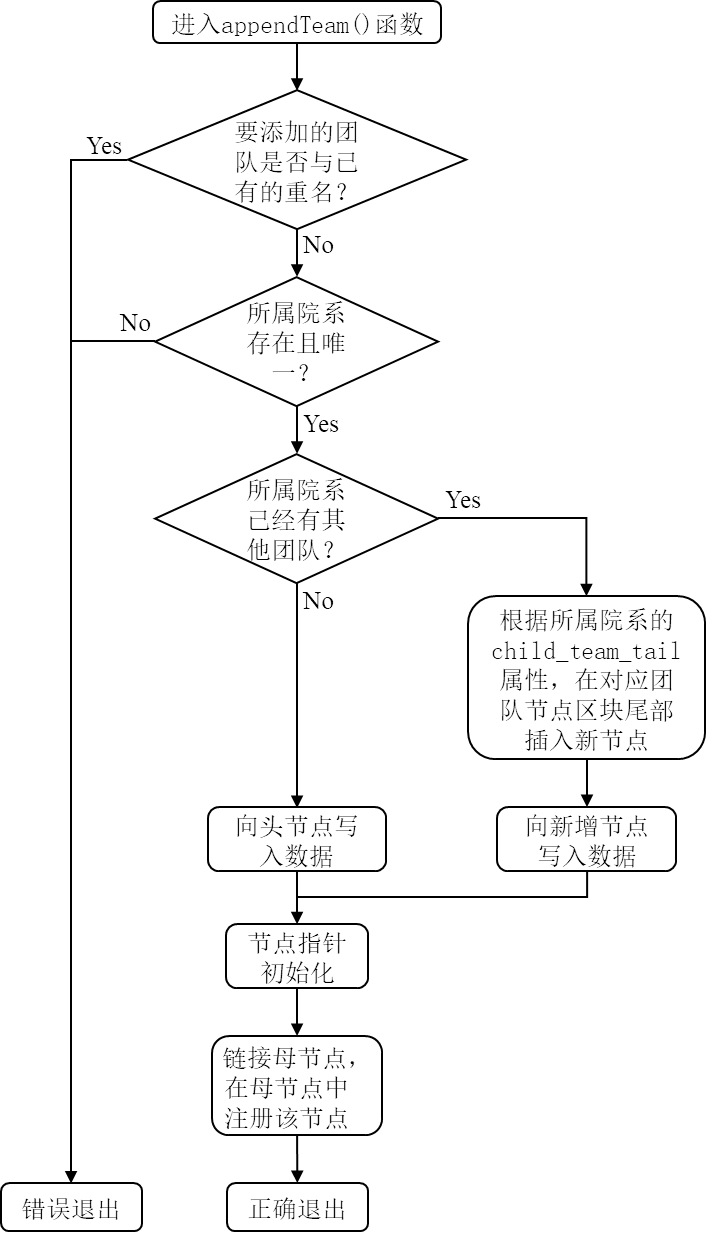


图4.7 团队信息添加流程图

（2）团队信息修改

系统向用户给出可供修改的数据项，读取用户提交的表单数据，判断数据合法性后即可修改内存中存储的内容。

（3）团队信息删除

删除节点的时候要注意不能够破坏链表结构的完整性。团队信息删除的流程如图4.8所示。

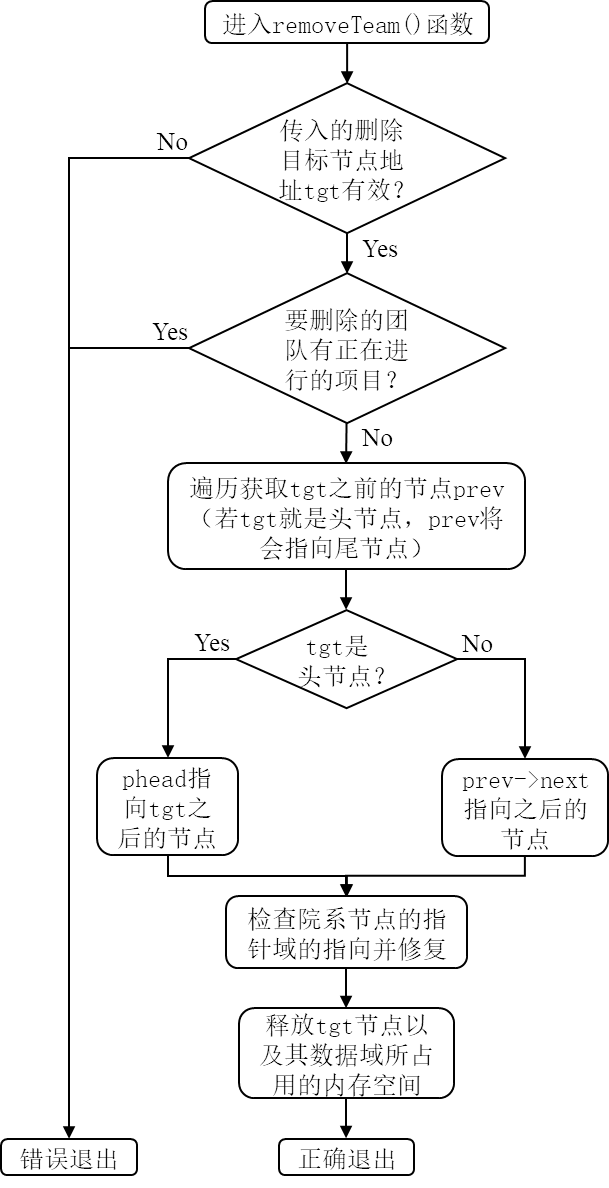


图4.8 团队信息删除流程图

4.2.3 数据查询

由于数据查询的实现过程只有条件判断的部分不同，故在此只给出按教师人数查找团队的详细流程。

对此例中给出的数据查询操作，由于查询方向比较多（大于、小于、等于等等），故设定了一个判断函数judger()以减少代码量，其本质为预置好的5个判断函数中一个的拷贝。其他数据查询函数由于查询方向单一，不需要设定判断函数。

其具体流程如图4.9所示。

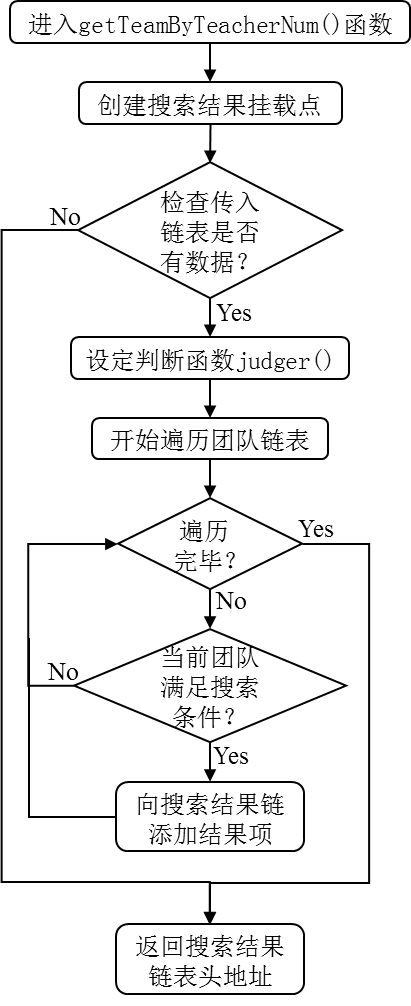


图4.9 按教师人数查找团队流程图

4.2.4 数据统计

数据统计实现方式在数据查询的基础上，多出了向包装结构体写入统计数据和对搜索结果链表进行排序的过程。这里只给出统计院系项目平均经费功能的详细流程。

第一步：统计所有院系的相关信息，包括项目总数与科研总经费。

若只是统计项目总数和科研经费，则可以不需要遍历团队链表，直接遍历项目链表即可。但是考虑到其他统计功能可能需要用到有关团队的统计信息，为了减少可能的代码量，在编写函数的时候还是遍历了团队链表。

第二步：对上一步中得到的院系统计结果链按科研总经费与项目总数的比值降序排序。这里采用冒泡排序，交换节点的时候交换院系统计节点指向的院系与相应的统计信息（depart指针与stat数据项）。

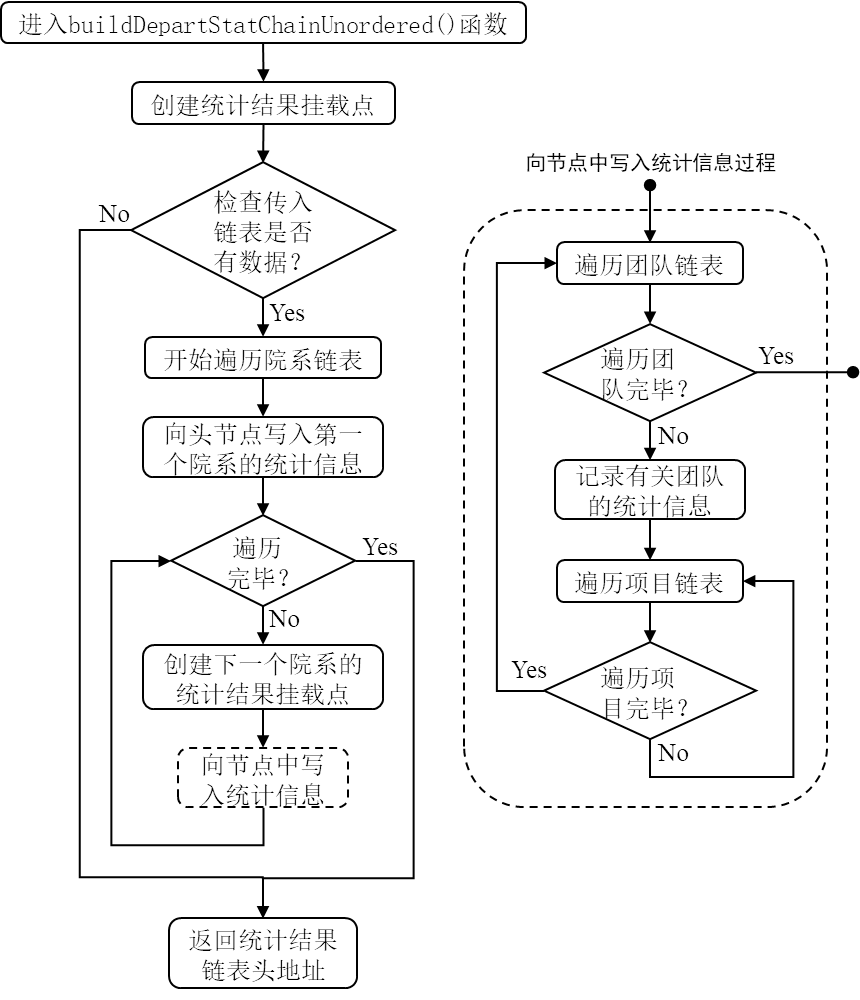


图4.4.1 建立所有院系统计信息表流程图

5 系统实现

本系统实现遵循结构化程序设计思想，将实现各种功能的函数按其关注的基本数据类型分在不同的模块中编码、测试。需要注意的是，在本系统中，数据处理部分均由C语言编写实现，Web服务后台部分由Python 3编写，Web后台通过接口调用C语言编写好的数据处理子程序，做到了数据处理与数据显示的完全分离，加强了系统的安全性。

5.1 源程序文件结构

下面给出本项目的源码文件结构。

./PROGRAM\_ROOT

+-- utils/

| +-- data/ // 测试用数据文件夹

| | +-- DEPART.DAT

| | +-- TEAM.DAT

| | +-- PROJECT.DAT

| +-- \_\_init\_\_.h // 所有数据处理函数的申明，方便引用

| +-- data\_structure.h // 数据结构

| +-- faculty\_functions.h // 院系函数

| +-- faculty\_functions.c

| +-- team\_functions.h // 团队函数

| +-- team\_functions.c

| +-- project\_functions.h // 项目函数

| +-- project\_functions.c

| +-- io\_functions.h // 数据载入、存储函数

| +-- io\_functions.c

| +-- makefile // 单元测试用

+-- doc\_strings.h // 文字界面用户帮助提示

+-- views.h // 文字界面视图函数申明

+-- main.c // 编译起点

+-- makefile // 完整编译用脚本

|

+-- py-extension/ // Web服务

+-- static/ // CSS, JS, 字体等资源

+-- templates/ // 页面模板

+-- build/ // C-Python共享对象（动态链接库）文件

+-- curr.c // C-Python对接API

+-- setup.py // 共享对象编译脚本

+-- py-extended-ver.py // 主程序

6 测试报告

以下测试开始时系统中已有如图6.1所示的部分数据。

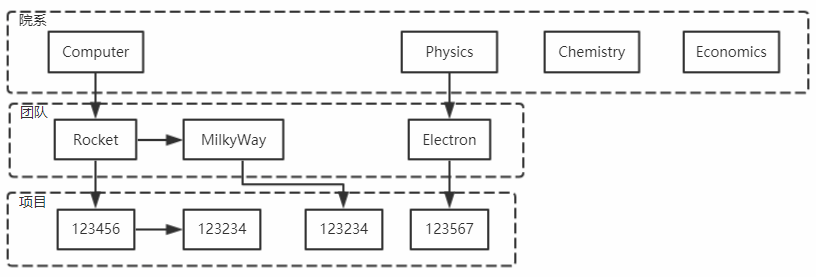


图6.1 测试时系统中基础数据

6.1 数据维护模块测试

6.1.1 数据添加测试

在下属“Physics”学院的“Electron”团队名下添加编号为“123321”的项目。应观察到“Electron”的项目列表中多出“123321”一项。

（1）运行程序Web服务，程序正确载入数据。

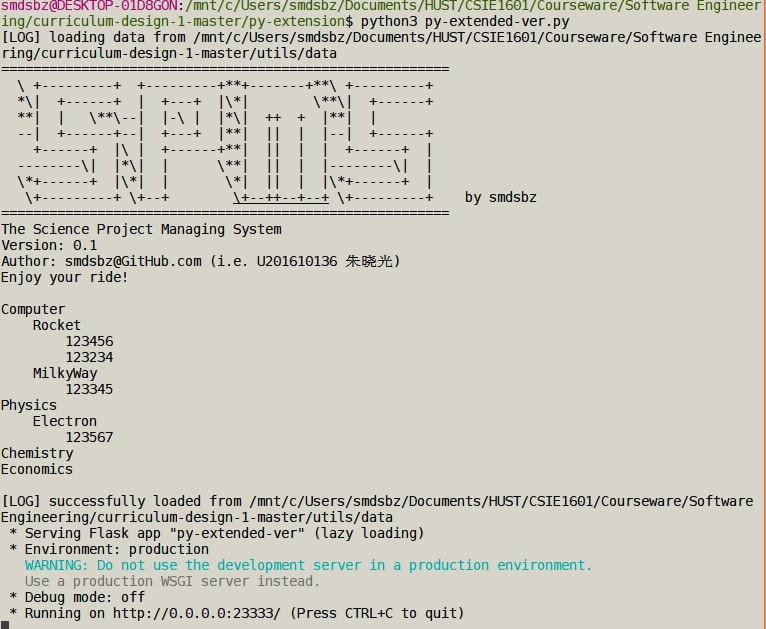


图6.2 程序成功载入数据截图

（2）在浏览器中输入URL，打开程序的操作界面。

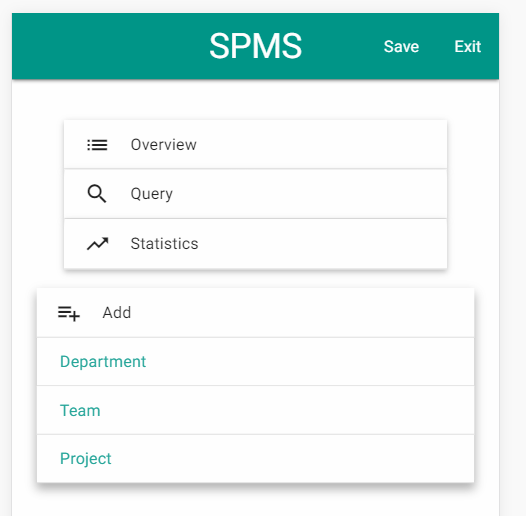


图6.3 系统主界面“Add”选单截图

（3）选择“Add/Project”，启用添加项目功能并填写信息。注意，此时将团队名称错误地填为“Electr”，以检测系统处理不合法输入的能力。

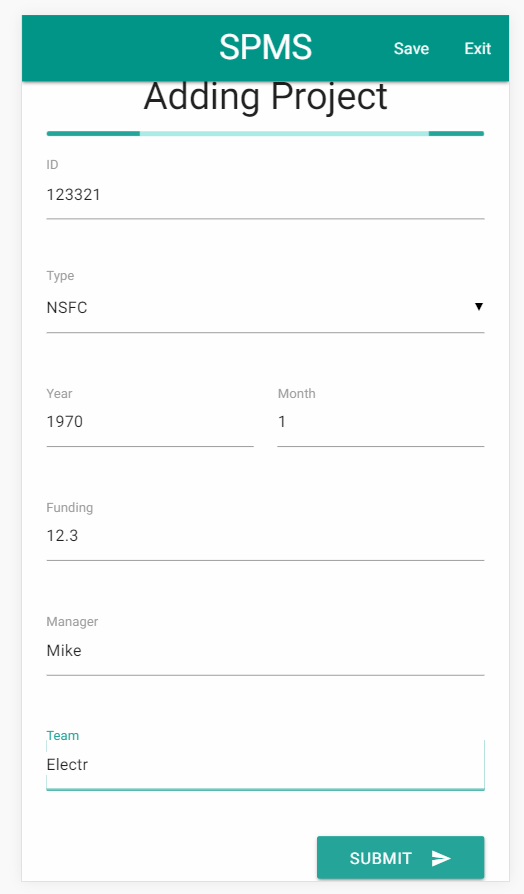


图6.4 错误表单截图

（4）单击“SUBMIT”按钮提交错误的项目信息，系统应提示项目添加失败并返回。

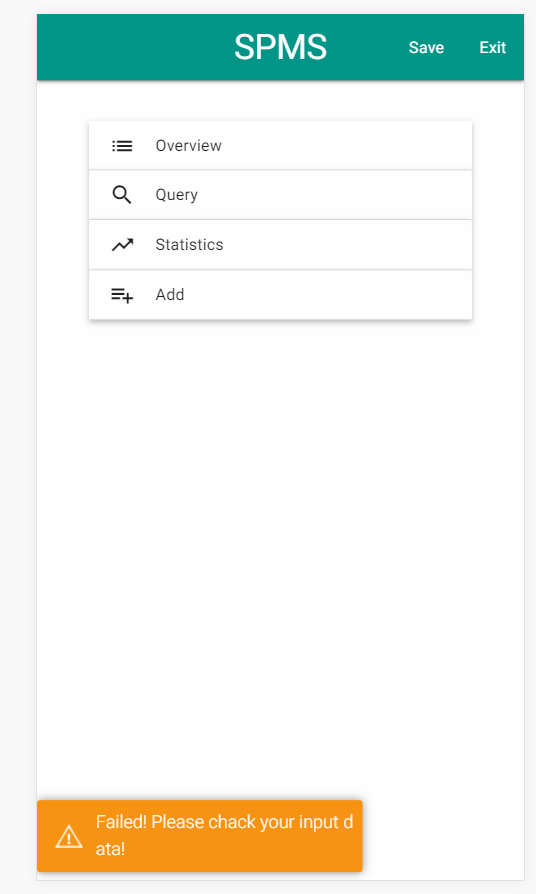


图6.5 提交错误表单截图

（5）再次填写正确的项目信息，点击提交，系统应提示操作成功并展示项目列表。

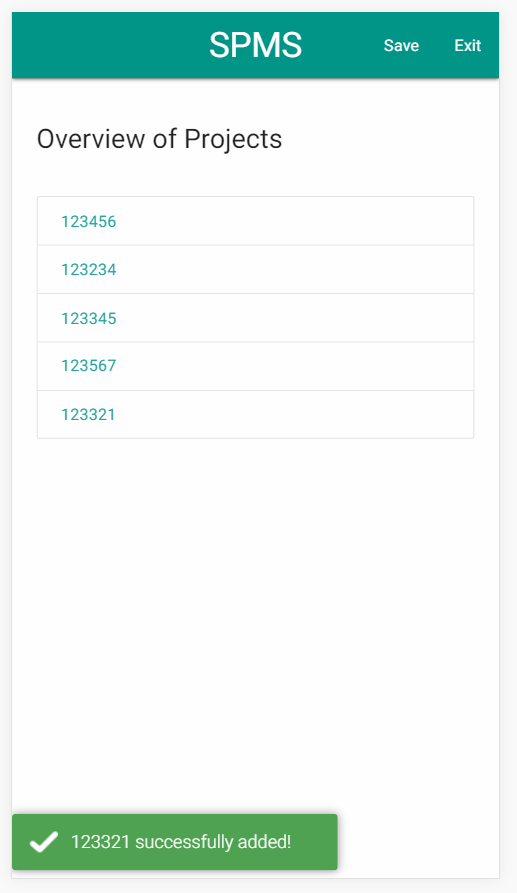


图6.6 成功添加项目截图

（6）选中“123321”项目，确认项目添加成功。

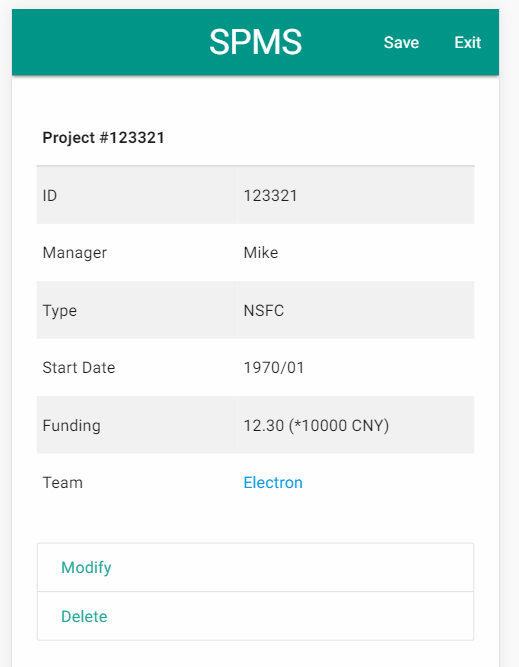


图6.7 “123321”项目基本信息截图

6.1.2 数据修改

接上一节的操作，修改项目“123321”的经费为13.3万元。项目的经费将从12.3万元变成13.3万元。

（1）选择“123321”项目的“Modify”功能，打开其信息修改窗口。

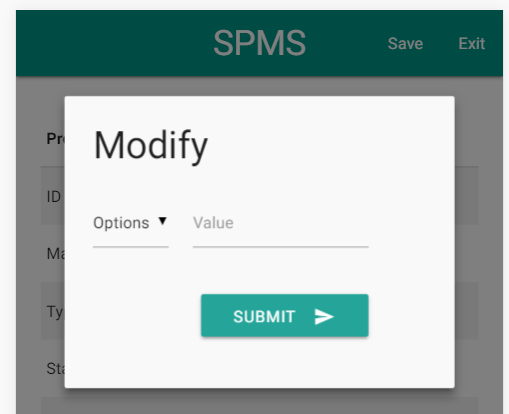


图6.8 “123321”项目信息修改窗口

（2）填写修改后信息。

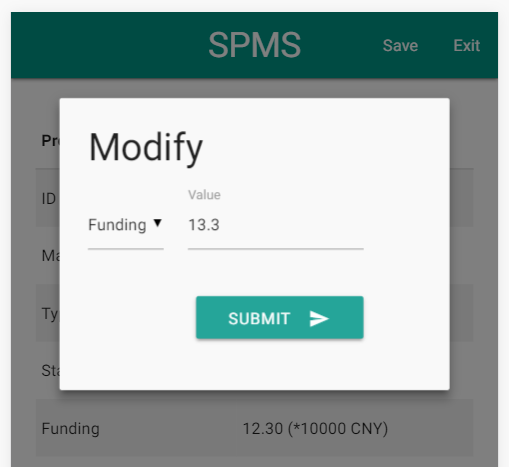


图6.9 填写修改后信息截图

（3）点击提交按钮。如图6.10所示，此时项目信息变更，新的项目经费已变为13.3万元。

6.1.3 数据删除

删除项目“123321”。所有记录的列表中将不再出现名为“123321”的项目。

（1）选择“Delete”删除当前项目。如图6.11所示，系统提示操作成功并返回项目列表。

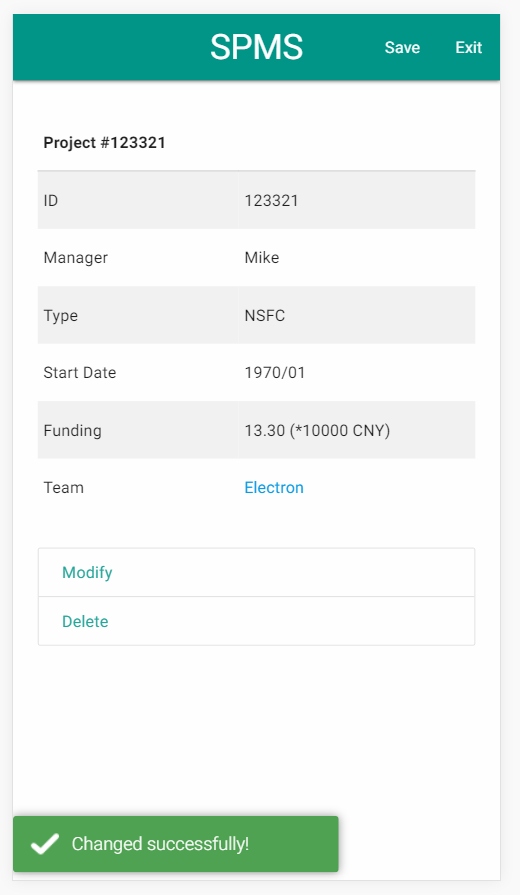


图6.10 成功提交修改后信息截图

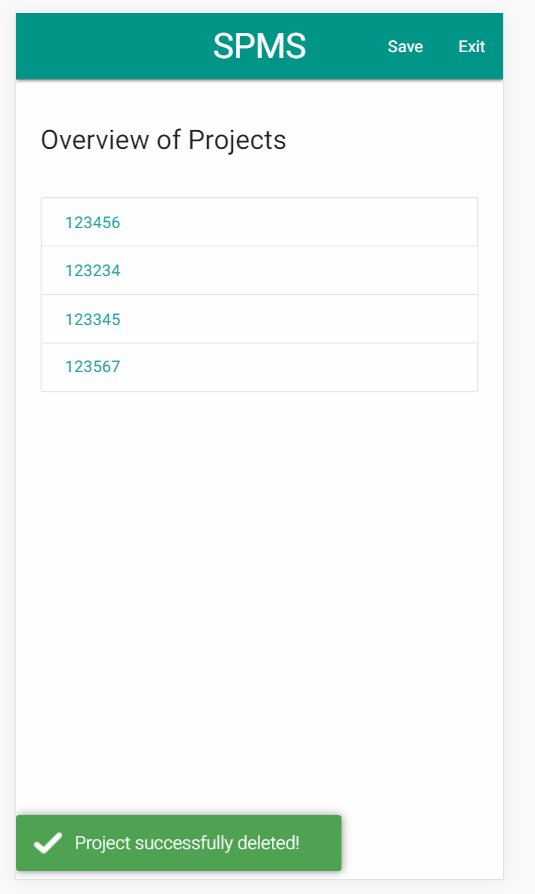


图6.11 数据删除截图

6.2 数据查询模块测试

这里只展示三种数据查询方式，及“按名称的全部或部分查询院系”、“按教师人数查询团队”、“按所属团队查询项目”

6.2.1 按名称的全部或部分查询院系

通过输入“Computer”学院名称的部分“Comp”来查找院系。

（1）在系统主页面中选择“Query/Department”项，进入院系查找功能。

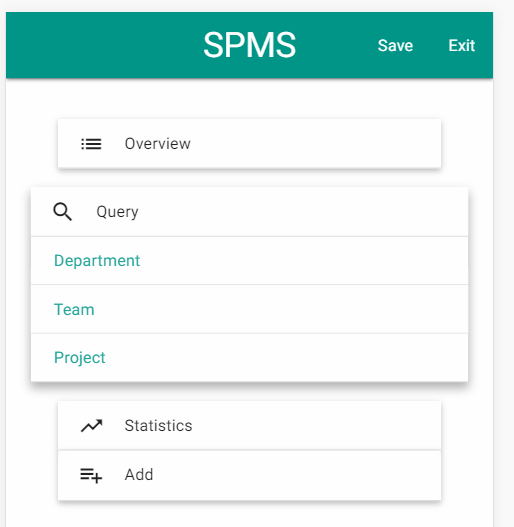


图6.12 系统主界面“Query”选单截图

（2）查找方式选择“by Name”，搜索框中输入“Comp”，点击放大镜图标或直接在搜索框中敲入回车。

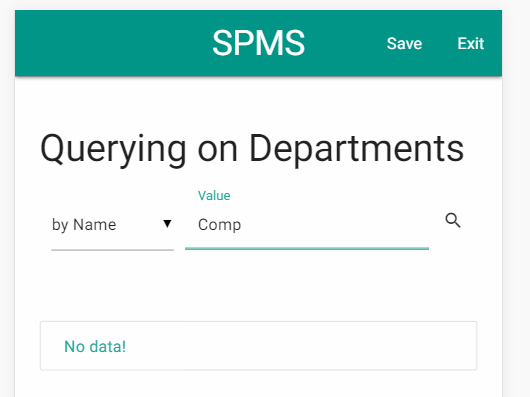


图6.13 院系查询内容截图

（3）如图6.14所示，搜索结果中出现“Computer”。

（4）点击“Computer”，可以查看到如图6.15所示的详细信息。

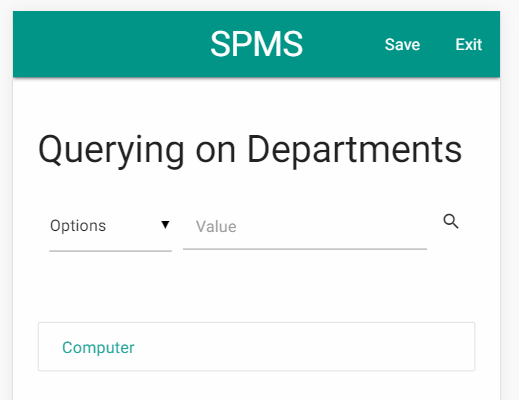


图6.14 院系查询结果截图

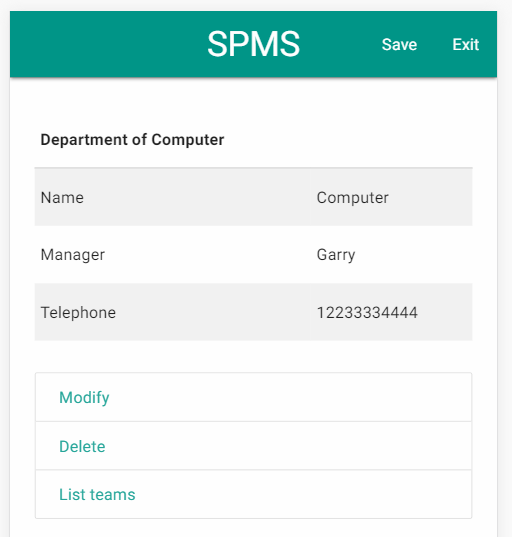


图6.15 “Computer”学院基本信息截图

6.2.2 按教师人数查询团队

通过团队查询功能，查询所有团队教师人数小于5的团队。

（1）选择“Query/Team”查询团队功能，键入查询条件。

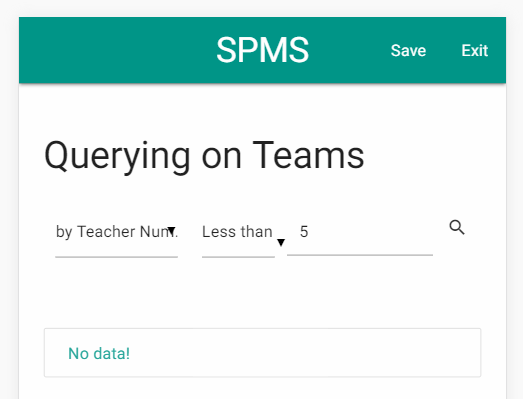


图6.16 团队查询条件截图

（2）点击放大镜图标，搜索结果如图6.17所示。



图6.17 团队查询结果截图

（3）随机点入一个团队，查看其教师人数是否小于5。

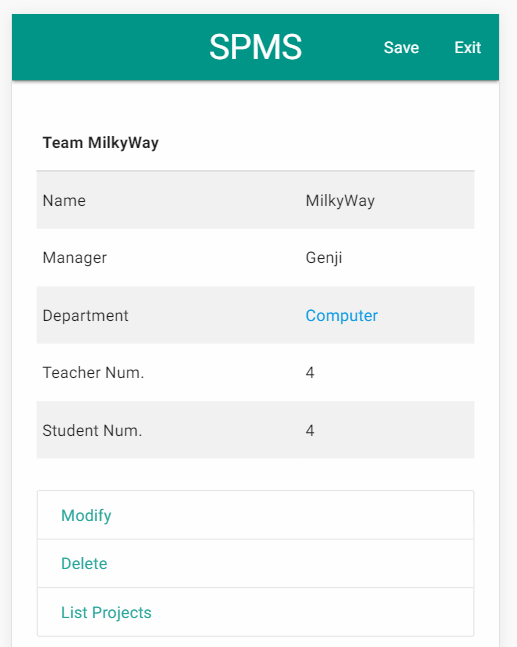


图6.18 团队教师数截图

6.2.3 按所属团队查询项目

在前面的例子中已经定位到“MilkyWay”团队，现在导航至其开设的项目“123345”。

（1）选择“List Projects”功能，查看该团队所有开设项目的列表。

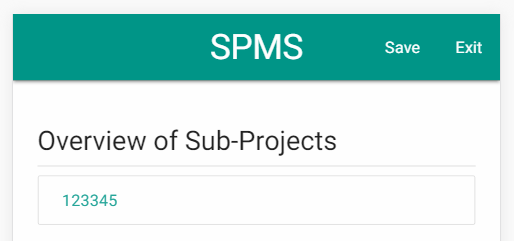


图6.18 团队开设项目列表截图

（2）选择“123345”项目，查看该项目的基本信息。

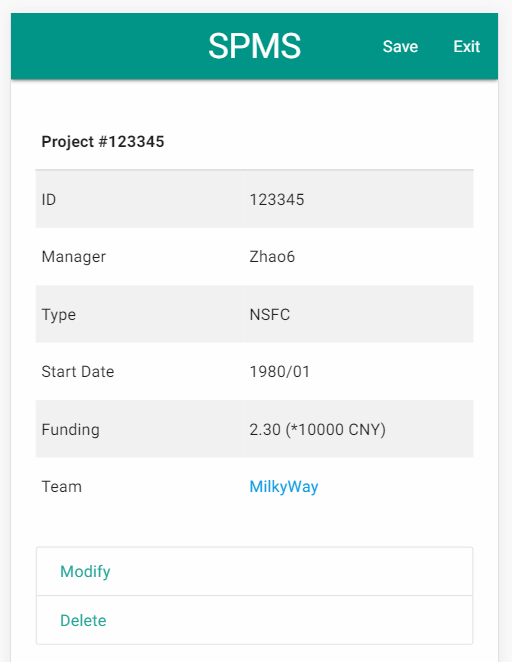


图6.19 项目基本信息截图

6.3 数据统计

这里只展示一种数据统计功能，其他统计功能操作类似。

6.3.1 院系项目总数统计

这里分别展示所有年度统计和指定1970年统计。

（1）在系统主界面选择“Statistics/2. Project Overview”功能。

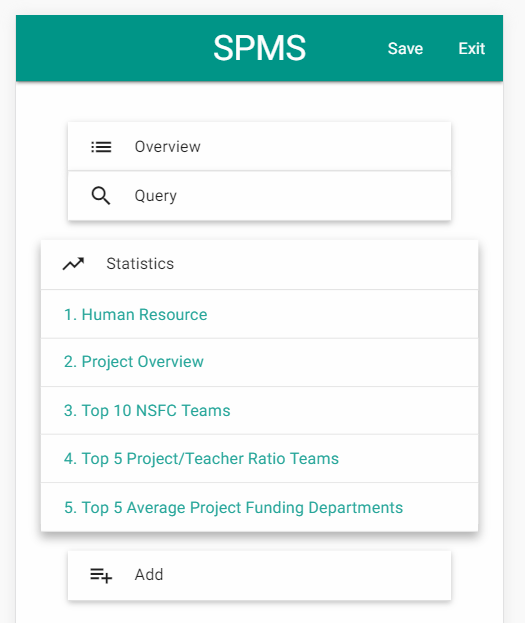


图6.20 系统主界面“Statistics”选单截图

（2）首先统计所有年度的数据，在输入框中输入“0”。

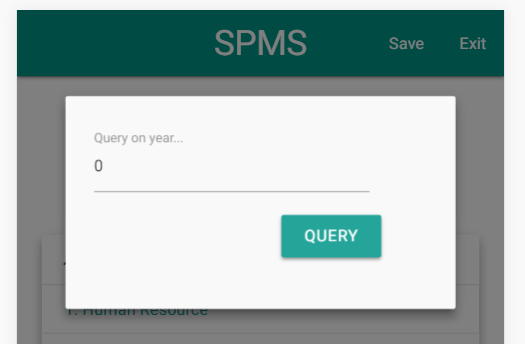


图6.21 统计所有年度输入截图

（3）按下“QUERY”按钮后，系统返回所有年度的统计数据。

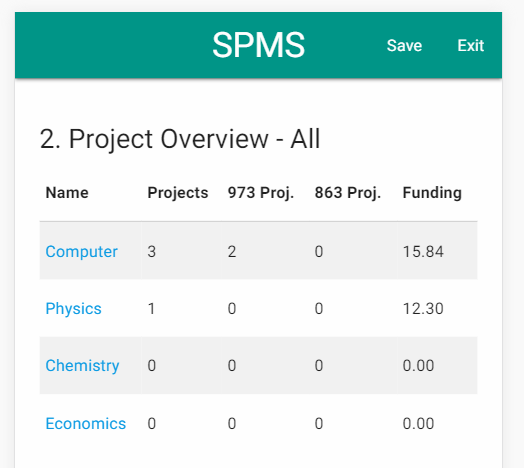


图6.22 所有年度统计结果截图

（4）点击页面顶部的“SPMS”返回主页面，重新选择该功能，统计1970年项目数。

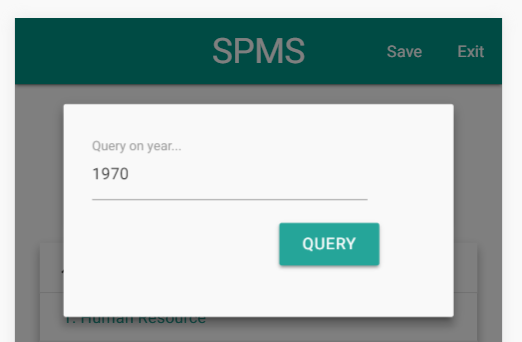


图6.23 统计1970年输入截图

（5）按下“QUERY”按钮后，系统返回如图6.24所示的统计结果。

**注：**若统计数据中涉及到除法计算，且除数是零时，除法结果将显示为“---”。

6 项目管理

6.1 软件规模度量

为了估算该系统的总工期以安排日程，需要在实际实现之前对该系统的最终完成规模有合理估计。

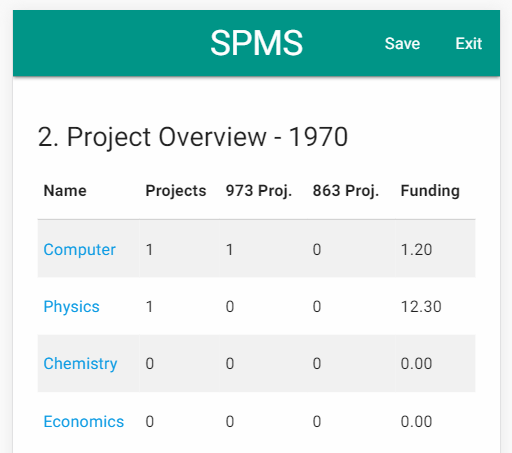
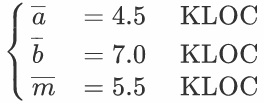


图6.24 1970年统计结果截图

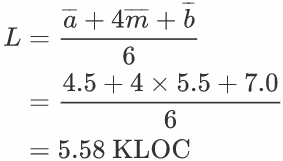
这里采用源代码行数来对系统复杂程度及规模进行评估。

经过统计，关于本系统实现所需代码行数的估计数据如下



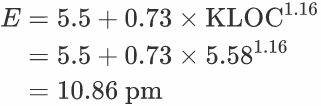
其中，a为最小规模估计均值，b为最大规模估计均值，m为最可能估计平均值。

由此可得系统程序规模的估计值为



6.2 工作量估算

这里使用Bailey-Basili模型估算完成该系统完成所需要的工作量。将上一节中计算所得的系统软件规模带入公式中可以得到



6.3 软件开发时间估算

考虑到其他各种成本因素，预期项目需要约6~7人来完成，预计工期为2个月。

6.4 工程进度计划

下面用如图6.1所示的Gantt图来展示本系统开发的工程进度计划。

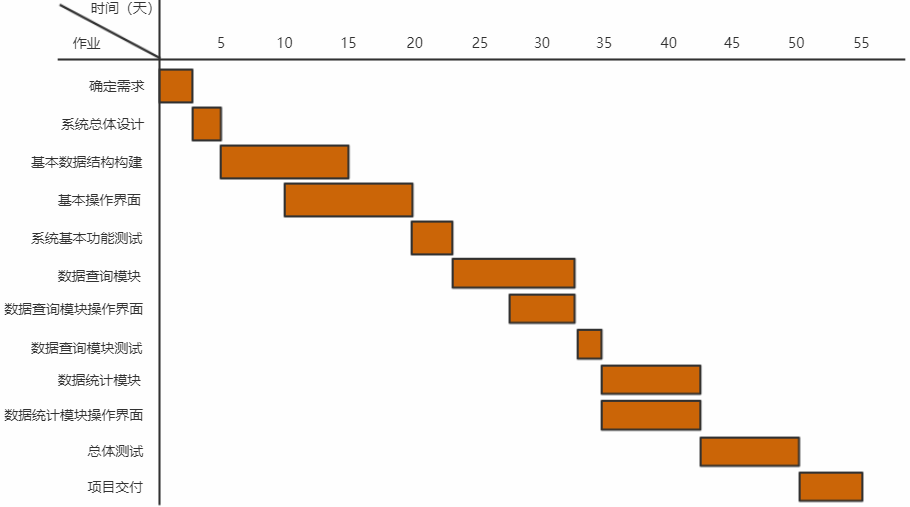


图6.1 项目工程进度计划Gantt图

7 课程学习总结

7.1 课程学习体会

通过本次软件工程课程设计，我了解了一个项目实际应有的开发流程，学习了项目文档编写与项目管理。

项目文档是帮助项目双方厘清项目需求重要工具，同时也是项目开发过程中的重要参考。在实现一个项目的时候，一定要在项目文档定稿后再开始着手实现，保证开发成果与项目需求契合。当一个项目有较大工作量时，需要小组协作完成，此时如何在小组内科学地分工就显得尤为重要，需要项目管理相关的知识来科学指导管理工作，以保证项目的开发工作能够高效地进行。

7.2 课程教学建议

希望允许课程设计以小组的形式共同完成，更贴近实际合作项目开发的场景。