# 概要设计说明书

**V1.0**

# 目录

# 概要设计说明书

# 1 引言

## 1.1 编写目的

本概要设计说明书编写的目的是说明程序模块的设计考虑，包括程序描述、输入/输出、算法和流程逻辑等，为软件编程和系统维护提供基础。本说明书的预期读者为系统设计人员、软件开发人员、软件测试人员和项目评审人员。

## 1.2 背景

a. 本项目名称：基于协同过滤的在线教育平台；
b. 本项目任务提出者：方涛；
   开发人员：方涛，胡恒昌，姜美羡
   目标用户：在校本科大学生，教师等人员；
   运行条件：腾讯云服务器。

## 1.3 定义

（1）在线教育：

或称远程教育、在线学习，现行概念中一般指的是指一种基于网络的学习行为，与网络培训概念相似。

（2）文本分类：

文本分类用电脑对文本集(或其他实体或物件)按照一定的分类体系或标准进行自动分类标记。

（3）网络爬虫（Reptilia）：

是一种自动获取网页内容的程序。是搜索引擎的重要组成部分，因此搜索引擎优化很大程度上就是针对爬虫而做出的优化。

（4）协同过滤推荐（Collaborative Filtering recommendation）：

协同过滤分析用户兴趣，在用户群中找到指定用户的相似（兴趣）用户，综合这些相似用户对某一信息的评价，形成系统对该指定用户对此信息的喜好程度预测。

（5）朴素贝叶斯算法（Naive Bayesian Model）：

朴素贝叶斯法是基于贝叶斯定理与特征条件独立假设的分类方法。

## 1.4 参考资料


[1]刘露,彭涛,左万利,戴耀康. 一种基于聚类的 PU 主动文本分类方法[J]. 软件学报,2013,11:2571-2583.

[2]平源. 基于支持向量机的聚类及文本分类研究[D].北京邮电大学,2012.

[3]杨杰明. 文本分类中文本表示模型和特征选择算法研究[D].吉林大学,2013.

[4]李荣陆. 文本分类及其相关技术研究[D].复旦大学,2005.

[5]王煜. 基于决策树和 K 最近邻算法的文本分类研究[D].天津大学,2006.

[6]苏金树,张博锋,徐昕. 基于机器学习的文本分类技术研究进展[J]. 软件学报,2006,09:1848-1859.

[7]周平红. 我国高等教育信息化水平测评与发展预测研究[D].华中师范大学，2012.

[8]范坤. 推进我国教育信息化建设进程的对策研究[D].华中师范大学，2004.

[9]牛龙飞. 基于Web的我国教育信息化公共服务平台的设计与实现[D].华中师范大学，2013.

[10] 艾瑞咨询2015中国在线教育行业发展报告


# 2 总体设计

## 2.1 需求规定

本系统主要分为前端（用户访问界面）和后端算法实现；

前端网页处理用户输入的访问需求，包括用户注册信息的提交、用户登录登出、数据库文档搜索、文献阅读等请求。功能性要求主要为处理用户注册登陆以及相关数据库信息访问、存储功能，性能上要求服务端对用户网页请求提交后响应时间不超过 5 秒钟。

后端算法主要分为协同过滤的推荐算法和朴素贝叶斯的分类算法。协同过滤算法输入用户 ID，书本 ID 和该用户对该书的评分情况，输出该用户可能喜欢的书本列表（或与该用户相似的用户 ID），性能上要求系统生成用户推荐列表处理时间不超过 3 秒钟。

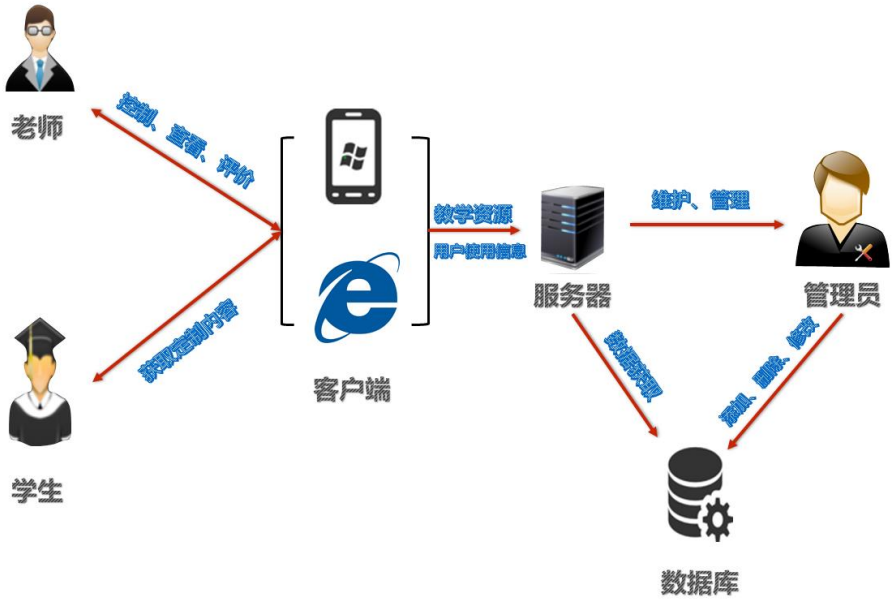
朴素贝叶斯分类算法输入书本 ID，书本内容，输出该书本可能的分类标签列表并写入数据库中，性能要求上对一本书的分类处理时间不超过 5 秒钟。

## 2.2 运行环境

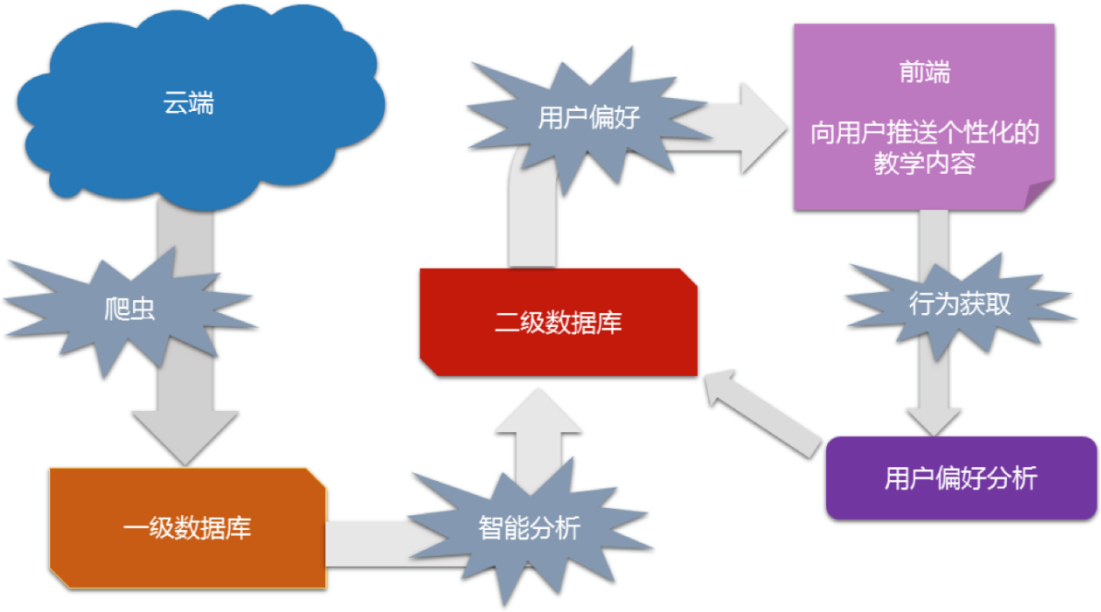
| 服务端 | | |
|---|---|---|
| 软件环境 | 操作系统 | CentOS 7 |
| | 编译语言支持 | Python2.7 |
| | 支持库 | Django 1.10.2 |
| | | MysqlClient |
| | | Numpy |
| | 数据库 | Mysql |
| 硬件环境 | CPU | 1GHZ |
| | 内存 | 1GB |
| | 带宽 | 1MB/s |
| | 硬盘 | 50GB |

## 2.3 基本设计概念和处理流程

（a）基本设计概念



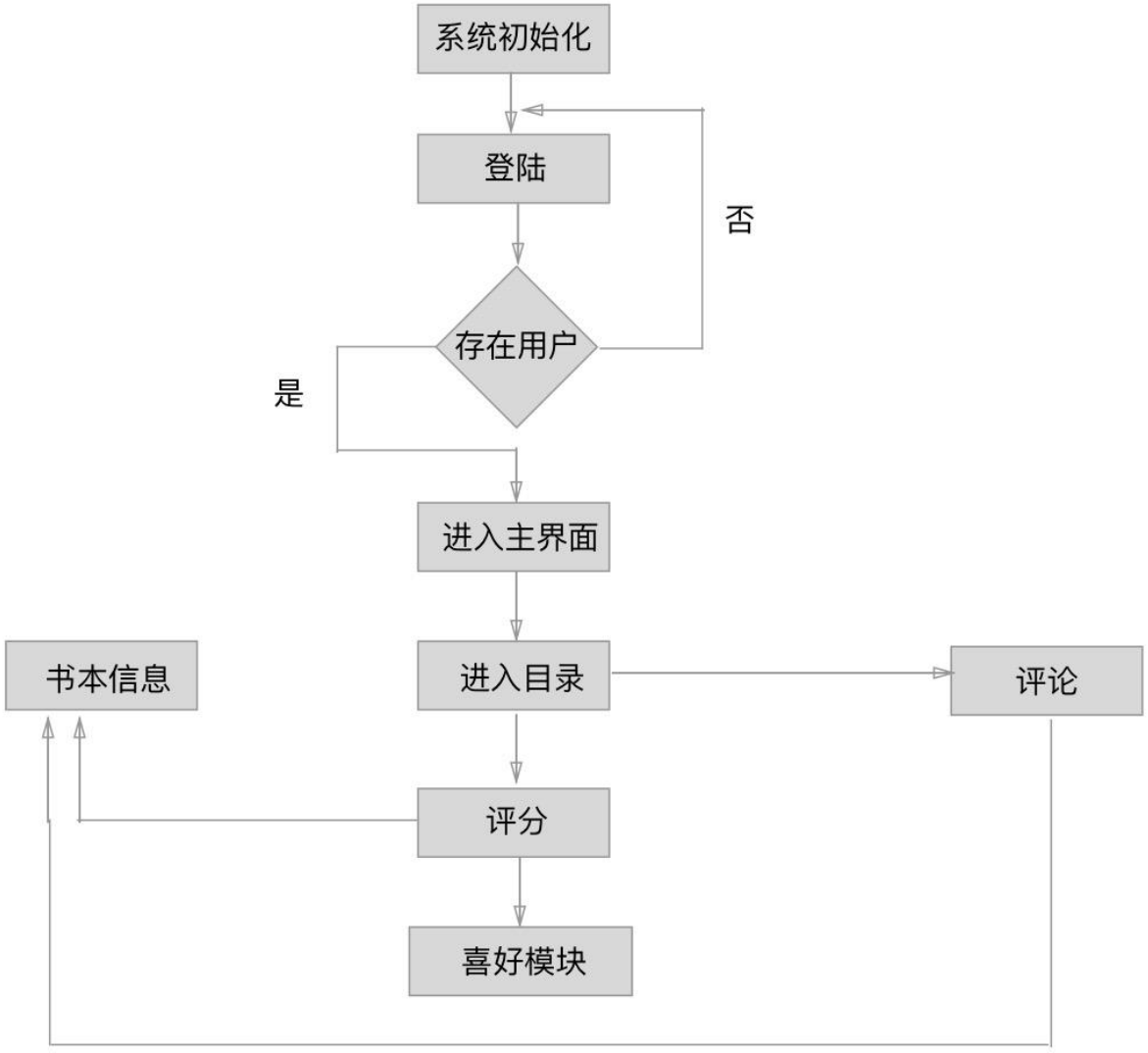

（b）数据处理流程




## 2.4 结构

| 系统元素名称 | 元素功能 |
| --- | --- |
| 推荐系统 | 系统需要根据用户自身的偏好信息定向提供相符的资源。先收集用户偏好，即找到相似用户和物品，计算用户间以及物品间的相似度，最终找到相似的用户或物品。 |
| 文本分类系统 | 本项目采用朴素贝叶斯算法实现。服务器接收书籍简介后，对文本进行分词并建立单词向量，输入贝叶斯分类算法中计算与各个类别的相似度，生成标签矩阵，并根据设定的阈值划分可能的标签，最后将该标签集合和书本 ID 存入数据库。 |
| 数据管理模块 | 管理用户数据的注册、提取、校验、登陆，以及用户浏览历史记录的存储调取，用户喜好标签的存储、修改、调取。 |
| 用户界面模块 | 提供与用户交互接口，允许用户在系统进行登陆、注册、浏览、信息修改等操作。 |
| 爬虫模块 | 采用开源爬虫框架 scrapy 进行分布式爬虫集群操作。采用超文本分类和随机森林聚类算法进行分类。根据网页链接网页的相关类型对网页进行分类，依靠相关联的网页推测该网页的类型。一个自动下载网页的程序，它根据既定的抓取目标，有选择的访问万维网 |

| | |
|---|---|
| | 上的网页与相关的链接，获取所需要的信息。用深度优先策略，其基本方法是对于每个目标网站，按照深度由低到高的顺序，依次访问下一级网页链接，直到不能再深入为止（即与该目标信息网页相关信息全部爬取完毕）。 |
| 搜索模块 | 获取输入的数据，如果有书名号则用正则表达式去除，然后在数据库中匹配每条记录的 name 字段，匹配成功则返回，否则返回缺失。最终生成搜索结果列表返回。 |




## 2.5 功能器求与程序的关系

本条用一张如下的矩阵图说明各项功能需求的实现同各块程序的分配关系：

| | 用户界面（WEB） | 用户请求处理（服务端） | 数据库管理 | 搜索模块 | 推荐算法程序 | 分类算法程序 |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| 用户登录 | √ | √ | √ | | | |
| 用户注册 | √ | √ | √ | | | |
| 用户登出 | √ | √ | | | | |
| 浏览书籍 | √ | | | | | |
| 搜索书籍 | √ | | | √ | | |
| 生成推荐书籍列表 | | √ | √ | | √ | |
| 书籍标签分类 | | | √ | | | √ |

## 2.6 人工处理过程

（a） 标签定义

本平台对初始书籍分类标签的定义难以从网上直接获取，需要人工定义需要显示的标签。

（b） 启动爬虫与分类程序

本教育系统运行时需要管理人员人工选择所需爬取的网页以及网页内容，并且人工地在合适的时候开启爬虫程序获取网络资源并分类。

（c） 初始用户数据

协同过滤算法实现推荐功能需要大量用户偏好数据作为样本才能启动，因此系统启动前期需要人工搜集大量用户数据。

## 2.7 尚未问决的问题

（a）协同过滤算法前期实现推荐所需用户数据的来源问题；
（b）爬虫爬取目标网站的选取问题；
（c）自动化获取信息并自动更新问题；
（d）服务端对用户请求响应过慢的问题。

# 3 接口设计

## 3.1 用户接口

（a） 用户注册接口

在 emousica.com/signup 网址下用户输入相关注册信息（包括注册邮箱、密码、用户昵称以及喜好标签），提交表单完成注册功能。

注册成功，系统则返回注册成功提示并跳转至主页面。

（b） 用户登陆接口

在 emousica.com 网址下用户输入相关登陆信息；

登陆成功，系统自动跳转至主页面并显示用户在线。

（c） 用户图书搜索接口

用户在 emousica.com/index 网址下，在搜索框中输入图书相关信息，点选搜索后系统返回数据库中所存储的相关图书索引列表。

（d） 用户评论接口

用户在 emousica.com/index/content 网址下，在评论页面输入相关评论，点选发表评论后系统刷新页面并返回提示评论成功，显示用户评论信息。

（e） 用户打分接口

用户在 emousica.com/index/content 网址下，在书本内容页面输入评分分数，点选发表评分后系统刷新页面并返回提示评分成功，显示用户评分信息。

## 3.2 外部接口

其他支持软件：

（a） 结巴分词：

输入文本内容，该软件处理后返回分词后的列表结果。

（b） Azure 文本关键词提取：

输入文本内容，该软件处理后返回提取关键词的列表结果。

## 3.3 内部接口

说明本系统之内的各个系统元素之间的接口的安排。

（a） 爬虫模块：

Public void getBookSources( int start , int end ,String keyWords)

调用该模块时输入开始爬取的网页数，爬取结束时的网页数，以及关键词，该模块将从目标网页按照关键词搜索后从开始到结束页数爬取书本关键信息，包括书名、作者、图片（URL）以及书本内容等。爬取成功之后自动存入服务器上的数据库中。

（b） 文本分类模块

Public String[] classification( String text)

调用该模块时输入需要分类文本的内容，该模块调用结巴分词自动将文本内容分词成数组并使用贝叶斯分类算法进行分类，最后返回可能的标签的数组。

（c） 用户推荐模块：

Public String[] recommender( int userID)

调用该模块时输入目标用户 ID，该模块将从数据库中自动调用所有用户的历史打分记录，并结合该用户的历史打分记录对该用户进行喜好推荐，返回可能喜好书目的列表。

（d） 用户注册模块：

Public boolean sign_up(int password , String mailbox , String name,String[] labels)

调用该模块时输入用户注册时输入的邮箱地址、用户昵称、密码和偏好标签，该模块自动匹配用户是否在数据库中已经存在，若不存在则写入数据库中。以布尔形式返回是否注册成功。

（e） 用户登陆模块：

Public boolean login(int password , String mailbox)
调用该模块时输入用户注册时输入的邮箱地址、密码，该模块自动匹配用户是否在数据库中已经存在，并匹配用户名和密码是否匹配。以布尔形式返回是否登陆成功。

# 4 运行设计

## 4.1 运行模块组合

（a） 获取数据库书籍资源：
先调用爬虫模块，该模块将从目标网页按照关键词搜索后从开始到结束页数爬取书本关键信息，包括书名、作者、图片（URL）以及书本内容等，按表存入数据库中，之后调用文本分类模块，向该模块中传入爬取的文本的内容，并使用结巴分词模块对文本内容进行分词，之后使用贝叶斯算法进行分类，最终将生成的文本分类标签存入数据库中。

（b） 生成用户推荐列表：
平时用户使用打分评论模块，该模块存储用户 ID，书本 ID 和用户对该书的评分在数据库中，之后生成用户推荐列表的时候调用推荐模块，输入用户 ID，该模块从数据库表中调取用户评分记录，使用协同过滤算法生成用户推荐列表并返回。

（c） 用户完成注册：
用户填写表单后发送提交表单请求，系统先调用注册模块对用户信息是否符合格式进行检查，之后调用查重模块遍历数据库检查用户数据是否已经存在，最终返回注册结果。

## 4.2 运行控制

（a） 运行爬虫与分类并存储：
管理人员在服务端运行 bookSources.py 脚本，输入开始结束页面数以及关键词，等待程序运行结束提示。

（b） 运行用户推荐模块：
用户发送请求后根据用户传入的用户 ID，服务端运行 recommender.py 脚本，运行完毕后返回用户推荐列表。

（d） 注册功能：
用户填写表单后发送提交表单请求，系统运行注册模块检查输入信息是否合理后存储用户信息。

## 4.3 运行时间

说明每种运行模块组合将占用各种资源的时间。

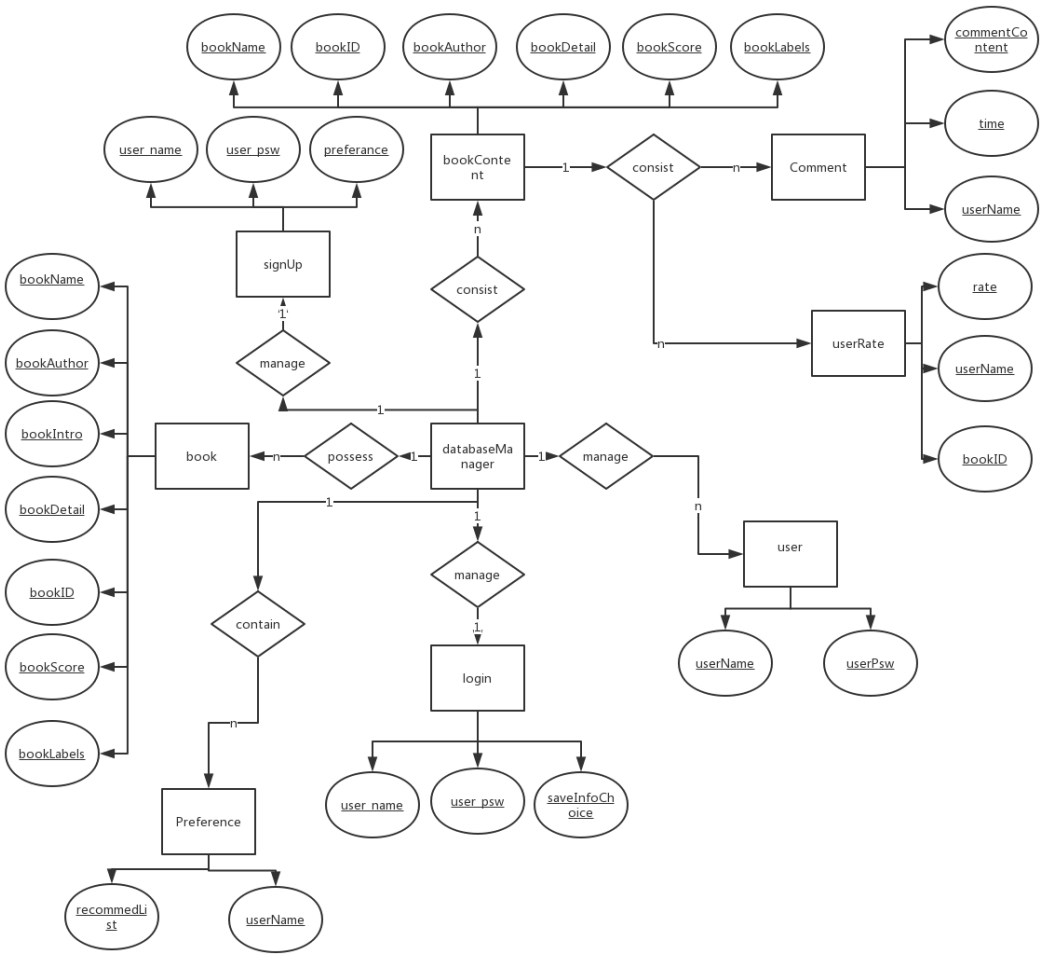
（a） 运行爬虫与分类并存储：
运行期间占用带宽与内存；
结束后写入数据库，运行时间与爬取资源数量有关；
平均爬取一页运行时间：10min。

（b） 运行用户推荐模块：
占用内存，使用带宽，
平均每个用户生成推荐列表处理时间不超过 10s。

（c） 注册功能：
写入数据库，
平均完成一个用户信息注册时间不超过 1S.。

# 5 系统数据结构设计

## 5.1 逻辑结构设计要点




本程序逻辑结构设计要点在于数据结构模块化划分的清晰，一级逻辑数据模块包括：login、preference、book、signup、bookContent、Comment、UserRate、user，这些模块彼此相对功能独立，但是也有千丝万缕的联系。

Login 是本平台的登入的第一个模块，其包括 user_name(用户名)、user_psw（用户密码）、saveInfoChoice（保存个人化信息选项），这些二层数据对接下来饿模块也具有影响，比如对用户的 user 数据库操作的时候，需要将 user_name 保存下来然后对服务器进行相关信息的提交，做个人化信息、书籍的推出的时候，也需要用到 saveInfoChoice。

Book 包括 bookName（书籍名字）、bookAuthor（书籍作者）、bookIntro（书籍简介）、bookDetail（书籍内容）、bookID（书籍编码）、bookScore（书籍评分）、bookLabel（书籍标签），这些书籍也与 bookContent、userRate 里的而成逻辑数据有着紧密的联系。

## 5.2 物理结构设计要点

存取方法：调用外部依赖库 mysqlclient，使用 Django 的 model 模版创建表结构，并且在 view 里进行相关操作。

存取单元结构：根据逻辑结构的指标以及 DBMS 支持的数据类型，所确定的数据项的存储类型和长度以及元组的存储结构等，即：数据文件及其数据项在介质上的具体存储结构。

存放位置；指根数据库文件和索引文件等在介质上的具体存储位置。

存储介质： 用于存储文件的物理存储设备包括磁盘、磁带、光盘、磁盘阵列、磁带库、光盘阵列，具体包括：介质容量的大小、存取速度与费用

## 5.3 数据结构与程序的关系

服务器程序在对用户进行增删改、书籍进行增删改、评分进行增删改、喜好推荐表进行修改的过程中需要对数据库数据结构也就是数据表进行查询和修改，在查询用户、书籍等操作中都需要对数据库中的所有表，进行联合查询、修改。

物理数据结构主要用于各模版之间函数的信息传递，接口传递的信息将是以数据结构封装了的数据，以参数传递或返回值的形式在各模版间传输。出错信息将送入显示模版中，一些数据的测试模块则送入准备模块中准备打印格式。

我们程序的数据结构在程序上的操作时线性的操作，从表的一段开始，向另外一端逐个按给定值与关键码进行比较，若找到，则是查找成功，并给出数据元素在表中俄位置，若整个表监测完，未找到相同的关键码，则查找失败，给出失败信息。从数据结构的逻辑关系层面考虑，顺序查找的方向是可以从左到右，也可以是从右到左。但是如果进一步考虑存储结构，该结论就不一定正确，比如单链表只能从左到右，如果决定使用链表，又要考虑从右到左的查找，显然必须启用双向链表，为了操作方便性而付出空间代价。

# 6 系统出错处理设计

## 6.1 出错信息

| 功能模块 | 异常说明 | 输出信息 | 处理方法 |
| --- | --- | --- | --- |
| 登陆模块 | 网络状况异常，信息通讯交换不稳定 | 无法获得正常返回信息 | 对程序中网络信息传输的稳定加保护 |
| | 输入值非法：长度溢出 | 可能造成 IndexOutOfBoundsException 溢出 | 对输入值的长度进行限制 |
| | 输入值非法：SQL 语言注入 | 非常规登入 | 加入登录防注入 |
| 查阅书籍模块 | 带有 href 属性的元素的值未在 Django 注册表单里登记 | 无法寻找到链接 | 对每一个外链 url 进行定义 |

| | | | |
|---|---|---|---|
| | 查询的字符串非法 | 未能搜索成功 | 对字符串合法监测 |
| | 网络状况异常，信息通讯交换不稳定 | 无法获得正常返回信息 | 对程序中网络信息传输的稳定加保护 |
| 评论模块 | 输入值非法 | 未能评论成功 | 对字符串合法监测 |
| | Decode 方法未正确定义 | 评论显示出问题 | Decode 正确定义 |

## 6.2 补救措施

我们的系统除了在数据库上，在本地也有备份，可以放置数据库的系列错误导致信息不可恢复的结果。在操作数据的过程中，如果因为软件错误或失误，则会进行人为就诊和修改错误代码。并且会及时更新系统。

后备技术说明准备采用的后备技术，当原始系统数据万一丢失时启用的副本的建立和启动的技术，例如周期性地把磁盘信息记录到磁带上去就是对于磁盘媒体的一种后备技术；

降效技术说明准备采用的后备技术，使用另一个效率稍低的系统或方法来求得所需结果的某些部分，例如一个自动系统的降效技术可以是手工操作和数据的人工记录；

恢复及再启动技术说明将使用的恢复再启动技术，使软件从故障点恢复执行或使软件从头开始重新运行的方法。

## 6.3 系统维护设计

系统大小：系统越大，理解掌握起来越困难，所执行功能越复杂。因而需要更多的维护工作量。

程序设计语言：语言的功能越强，生成程序所需的指令数就越少；语言的功能越弱，实现同样功能所需语句就越多，程序就越大。有许多软件是用较老的程序设计语言书写的，程序逻辑复杂而混乱，没有做到模块化和结构化，直接影响到程序的可读性。

系统年龄：老系统随着不断的修改，结构越来越乱；由于维护人员经常更换，程序又变得越来越难于理解。而且许多老系统在当初并未按照软件工程的要求进行开发，因而没有文档或文档太少，在长期的维护过程中文档在许多地方与程序实现变得不一致，这样在维护时就会遇到很大困难。

数据库技术的应用：使用数据库，可以简单而有效地管理和存储用户程序中的数据，还可以减少生成用户报表应用软件的维护工作量。

先进的软件开发技术：在软件开发时，若使用能使软件结构比较稳定的分析与设计技术及程序设计技术，如面向对象技术、复用技术等，可减少大量的工作量。