北工大教材资源整合平台TextbookGenius

软件设计说明书

团队编号：1

团队成员：张嘉茵、王林诺、彭叶、

李锶、李懿璇、张瑞涵

完成日期：

目录

[1．引言 3](#_Toc43487745)

[1.1编写目的 3](#_Toc43487746)

[1.2项目背景 3](#_Toc43487747)

[1.3定义 3](#_Toc43487748)

[1.4参考资料 3](#_Toc43487749)

[2．概要设计 3](#_Toc43487750)

[2.1系统模块划分 3](#_Toc43487751)

[2.2系统结构设计 3](#_Toc43487752)

[2.3处理流程设计 3](#_Toc43487753)

[3．数据库设计 4](#_Toc43487754)

[4．接口设计 4](#_Toc43487755)

[4.1外部接口 4](#_Toc43487756)

[4.2内部接口 4](#_Toc43487757)

[5．模块详细设计 4](#_Toc43487758)

[6. 界面设计 4](#_Toc43487759)

[6.1界面样式设计 4](#_Toc43487760)

[6.2界面交互设计 4](#_Toc43487761)

[7．出错处理设计 5](#_Toc43487762)

[7.1错误类型及出错处理对策 5](#_Toc43487763)

# 1．引言

## 1.1编写目的

【阐明编写本设计说明书的目的，并指明读者对象。】

目的：思考、梳理软件各部分的设计思路，在开发过程中为开发人员提供具体的实现指导。

读者对象：开发人员

## 1.2项目背景

【应包括：

1. 项目的委托单位、开发单位和主管部门；
2. 该软件系统与其他系统的关系。】

本组成员在过去的学期中经常出现以下问题：购买教材时因教材使用情况无处得知，购买后教材在课程中完全不被使用、在第三方网站购买教材远比在学校购买便宜，因此带来经济上的损失。通过调查问卷可知，这是同学们共同的疑惑与困扰，参与本次调查文件共29人，全部参与者出现购买教材后不使用的情况。



正逢软件工程课设，我们针对教材使用情况无处得知的痛点，考虑到老师或许存在利益相关问题，因此准备开发一个基于北工大学生对教材使用情况进行评价的资源整合平台。

考虑信息时代web开发的发展与繁荣、web开发涉及技术的广泛应用，本组成员认为进行web开发可以将学习到的知识最大化，因此选择web应用作为开发目标，学习并使用当前流行的技术进行开发。

系统将调用豆瓣、淘宝等系统的API，获得教材基本信息及价格等信息，并通过评论区收集用户（北工大学生）对教材的评价。

## 1.3定义

【列出本文档中所用到的专门术语的定义和缩写词的原文。】

API, Application Programming Interface，即应用程序编程接口。

React, 是一个用于构建用户界面的 Javascript 库，主要用于构建 UI.

Django，Django 是一个高级的 Python 网络框架，可以快速开发安全和可维护的网站。由经验丰富的开发者构建，Django 负责处理网站开发中麻烦的部分，因此开发者可以专注于编写应用程序，而无需重新开发。

Material UI，是React的一个UI框架，它提供了各种现成的组件，可以在React项目中使用。

## 1.4参考资料

【列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源】

Django官方文档：<https://docs.djangoproject.com/zh-hans/4.2/>

Tim的Music controller项目：

<https://github.com/techwithtim/Music-Controller-Web-App-Tutorial/tree/main>

# 2．概要设计

## 2.1系统模块划分

【描述系统模块划分、说明每个模块所包含的功能、以及该功能与需求（参考需求规格说明）的对应关系】

1. 用户信息管理模块
2. 用户注册、登录
3. 用户修改个人信息
4. 用户选择想向他人展示的信息
5. 显示用户收藏的教材
6. 教材信息管理模块

（1）显示教材基本信息（书名、作者、出版社、版本、ISBN号、使用课程、使用教师等）

（2）显示电子书获取途径

（3）显示实体书获取途径

（4）添加/删除教材

（5）修改教材信息

（6）上传教材/课程资源

（7）显示共享教材资料（笔记、其他学习资料等）

1. 学生互动模块
2. 关于教材/课程是否有用的投票及打分
3. 发表评论
4. 点赞他人的评论内容，面对无效/有害内容可点踩
5. 常用关键词搜索、筛选评论
6. 显示用户的年级、专业信息和信用分，保证用户发言的可靠性
7. 计算用户信用分
8. 教材检索模块
9. 通过关键词搜索教材
10. 显示用户搜索历史

## 2.2系统结构设计

【描述系统结构（系统结构图）】



本项目选择C/S的分层系统结构，表示层使用React框架与Material UI进行构件，通过HTTP请求与Django后端通信。

应用处理层实现系统的主要功能。

数据管理层使用Django的数据管理功能及Django内置SQLite数据库。

## 2.3处理流程设计

【描述各模块的主要操作流程（流程图）】

# 3．数据库设计

【数据库表设计，外键关系设计等】

1. Book

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Field | Type | Null | Key | Default | 解释 |
|  | isbn | varchar(13) | NO | 主键 | NULL | isbn号 |
|  | title | varchar(50) | NO |  | "" | 书名 |
|  | author | json | NO |  | "" | 作者列表 |
|  | publisher | varchar(50) | NO |  | "" | 出版社 |
|  | pubdate | varchar(10) | NO |  | NULL | 出版日期 |
|  | cover | varchar(100) | NO |  | "" | 封面图片地址 |
|  | douban\_url | varchar(50) | NO |  | "" | 豆瓣地址 |

1. Comment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Field | Type | Null | Key | Default | 解释 |
|  | com\_id | varchar(50) | NO | 主键 | NULL | 评论编号 |
|  | info | varchar(50) | NO |  | "" | 内容 |
|  | user\_id | varchar(50) | NO |  | "" | 发表用户id |
|  | date | date | NO |  | NULL | 评论日期 |

1. User

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Field | Type | Null | Key | Default | 解释 |
|  | id | varchar(50) | NO | 主键 | NULL | id编号 |
|  | name | varchar(50) | NO |  | "" | 昵称 |
|  | password | varchar(20) | NO |  | "" | 密码 |
|  | pic | varchar(100) | NO |  | "" | 头像地址 |
|  | class | int | NO |  | "" | 年级 |
|  | major | varchar(50) | NO |  | "" | 专业 |
|  | email | varchar(100) | NO |  | "" | 邮箱 |
|  | upload\_book | json | NO |  | "" | 上传信息记录 |
|  | comment | json | NO |  | "" | 评论记录 |
|  | star | json | NO |  | "" | 收藏记录 |
|  | credit\_score | int | NO |  | 100 | 信誉分 |

# 4．接口设计

## 4.1外部接口

【包括软件接口与硬件接口】

**1. 获得电子书、实体书获取途径及信息**

功能：获得Z-Library，Library Genius，Anna’s Archive等电子书获取途径，实体书的淘宝、当当、京东、孔子旧书网等网站对应书记的信息

接口：

1. GET /api/get-douban-book
2. GET /api/get-anna-book
3. GET /api/get-taobao-info
4. GET /api/get-dangdang-info
5. GET /api/get-jingdong-info
6. GET /api/get-kong-info

## 4.2内部接口

【模块之间的接口】

## 用户信息管理

功能：对用户信息进行增删改查等管理功能

接口：GET /api/user

**1.1 注册**

功能：用户填写必要信息后可成为网站注册用户，可以进行评论

接口：POST /api/user-register

**1.2 登录**

功能：已注册用户登录

接口：POST /api/user-login

**1.3 修改用户信息**

功能：修改个人信息如昵称、学部、专业、年级、头像，编辑部分信息是否向他人展示

接口：POST /api/update-user/:userId

**1.5 获得用户信息**

功能：获得其他用户的昵称、信誉分、学部、专业、年级、历史评论记录等必要基本信息和希望展示的信息，及用户收藏的内容

接口：GET /api/user/:userId

## 教材信息管理

功能：对教材相关信息进行创建、删除、修改、删除等管理功能

接口：GET /api/book

**2.1 获得教材基本信息**

功能：获得书名、作者、出版社、版本、ISBN号、使用课程、使用教师等信息

接口：GET /api/book/:bookId

**2.4 增加教材**

功能：用户和管理者自主增加对应课程、教师使用教材的信息

接口：POST /api/create-book

**2.5 删除教材**

功能：用户、网站管理者及开发人员删除教材信息

接口：DELETE /api/delete-book

**2.6 修改教材信息**

功能：用户、网站管理者及开发人员修改教材信息

接口：PUT /api/update-book

## 基于课程的论坛模块

功能：用户互动区，包括对教材实用性进行评分、发表评论、点赞/点踩和回复评论功能

接口：GET /api/forum

**3.1 获得评论信息**

功能：获得用户发表的评论、点赞数量和针对教材对应于某课程是否实用的评分结果。

接口：

1. 教材实用性评分、评论列表、点赞信息获取：GET/api/forum
2. 评论详情获取：GET/api/forum/:commentId

**3.2 用户互动**

功能：用户对教材实用性打分、发表评论、点赞或回复评论，面对负面无效信息可以点踩。

接口：

1）针对教材实用性打分：POST/api/forum/score

2）发表评论：POST/api/forum/commentThread

3）点赞评论：POST/api/forum/like

4）点踩评论：POST/api/forum/dislike

**3.3 常用关键词搜索，精选评论**

功能：评论上方展示一些评论高频词tag并设置搜索框，用户可通过选择tag或输入关键词来筛选评论。

接口：

1）关键词搜索：GET/api/forum/search?q=:query

2）高频词tag搜索：GET/api/forum/search?q=:query

**3.4 上传课程资源**

功能：上传用户上传的文件作为课程资源

接口：POST /api/upload-src

**3.4 删除课程资源**

功能：删除已存在的用户分享的电子教材、课程笔记等资源

接口：PUT /api/delete-src

**3.5 获得共享课程资料信息**

功能：获得由用户上传的教材、课程资料

接口：GET /api/course/:courseId

## 教材检索页面

功能：根据条件对书籍、课程进行检索

接口：GET /api/search

**4.1关键词搜索**

功能：用户通过输入关键词搜索书籍

接口：GET/api/search?q=:Wd

## 信誉系统

功能：对用户信誉进行实时计算并获取

接口：GET /api/rep

**5.1 获得用户信誉等级**

功能：获得用户年级和专业信息、信用分

接口：GET /api/get-rep

**5.2 计算用户信用分**

功能：设置适当计算方法，如1.评论被点赞多少次就增加一定的信用分；2.被举报或存在违规行为时降低信用分；3.用户信息越完备，信用分越高；4.一定时期内无违规行为，给予一定的信用分作为奖励；5.创建用户时基础信用分一致，信用分存在上限，超出时不再累计。

接口：POST /api/update-rep

# 5．模块详细设计

【描述系统关键模块的静态结构（类图），其中关键功能/方法应给出伪码或流程图】

# 6. 界面设计

## 6.1界面样式设计

【程序或原型的主要页面截图】

## 6.2界面交互设计

【页面导航，界面元素和布局设计等】

# 7．出错处理设计

## 7.1错误类型及出错处理对策

【错误类型包含各个模块中的用户输入错误，程序执行错误，硬件故障等】

* + 1. 接口请求错误处理
       1. GET请求资源请求
       2. POST创建资源请求
       3. DELETE删除资源请求