**Code-tagger代码在线编辑系统**

**[软件设计说明书]**

目录

[1.引言 1](#_Toc124176619)

[1.1 编写目的 1](#_Toc124176620)

[1.2 相关文档 1](#_Toc124176621)

[1.3 版本更新记录 1](#_Toc124176622)

[2.总体设计 2](#_Toc124176623)

[2.1 运行环境 2](#_Toc124176624)

[2.2 子系统清单 2](#_Toc124176625)

[2.3 功能模块清单 2](#_Toc124176626)

[2.4 类图设计 3](#_Toc124176627)

[3.数据库设计 4](#_Toc124176628)

[3.1 数据库中表名列表 4](#_Toc124176629)

[3.2 数据库表的详细清单 4](#_Toc124176630)

[4.前端页面设计 6](#_Toc124176631)

[4.1 登录界面 6](#_Toc124176632)

[4.2 注册界面 6](#_Toc124176633)

[4.3 用户主界面 7](#_Toc124176634)

[4.4 标签库主界面 7](#_Toc124176635)

[4.5 代码库主界面 8](#_Toc124176636)

[4.6代码编辑器界面 8](#_Toc124176637)

[5.后端功能函数设计 8](#_Toc124176638)

[5.1 login () 9](#_Toc124176639)

[5.2 resist() 9](#_Toc124176640)

[5.3 queryRecommendBookList() 9](#_Toc124176641)

[5.4 queryBookByID() 9](#_Toc124176642)

[5.5 addBook() 9](#_Toc124176643)

[5.6 updateBook() 10](#_Toc124176644)

[5.7 deleteBook() 10](#_Toc124176645)

[6.接口设计 11](#_Toc124176646)

[6.1 用户接口 11](#_Toc124176647)

[6.2 外部接口 11](#_Toc124176648)

[6.3 内部接口 11](#_Toc124176649)

[7.系统错误处理 12](#_Toc124176650)

[7.1出错信息 12](#_Toc124176651)

[7.2故障预防与补救 12](#_Toc124176652)

[7.3系统维护设计 12](#_Toc124176653)

[8.测试计划 13](#_Toc124176654)

# 1.引言

## 1.1 编写目的

在需求分析的基础上，我们依据《需求规格说明书》对该系统进行了设计。为了使系统的设计文档化，我们编写了此文档。

## 1.2 相关文档

[1] 软件开发计划书

[2] 需求规格说明书

[3] 部署文档

[4] 测试报告

[5] 用户使用说明书

## 1.3 版本更新记录

表格 1 版本更新记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本号** | **创建者** | **创建日期** | **维护者** | **维护日期** | **维护纪要** |
| V1.0 | 陆清媛 | 2022.10.31 | —— | —— | —— |
| V1.0.1 | —— | —— | 秦昊 | 2022.11.14 | 为用户注册增加安全性保障 |
| V1.0.2 | —— | —— | 袁欣怡 | 2022.12.20 | 修改了部分页面设计 |
| V1.0.3 | —— | —— | 杨睿 | 2022.1.1 | 修改了部分接口 |
| V1.0.4 | —— | —— | 向雨新 | 2023.1.8 | 增加管理员功能 |

# 2.总体设计

## 2.1 运行环境

（1）软件形式：web应用

（2）Web服务器：阿里云ECS云服务器

（3）数据库：sqlite 3.0

（4）浏览器：IE8以上

## 2.2 子系统清单

表格 2 子系统清单

|  |  |  |
| --- | --- | --- |
| **编号** | **子系统名称** | **子系统功能描述** |
| 1 | 个人信息管理系统 | （1）登陆注册功能的实现；  （2）密码安全性保障 |
| 2 | 代码编辑系统 | （1）存储代码文件；  （2）代码在线编辑保存；  （3）部分编程语言的代码高亮显示 |
| 3 | 标签标注系统 | （1）用户标签库存储标签；  （2）标签信息的增删改查；  （3）对代码文件的特定代码段进行标注 |

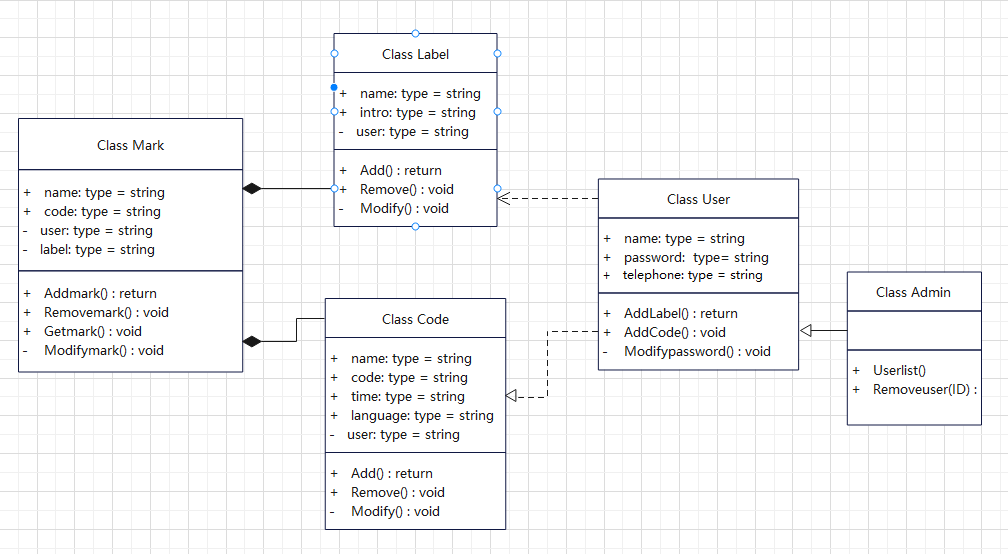
## 2.3 功能模块清单

表格 3 功能模块清单

|  |  |  |
| --- | --- | --- |
| **编号** | **名称** | **功能描述** |
| 101 | 注册 | 游客注册成为已注册用户 |
| 102 | 登录 | 已注册用户登录系统 |
| 103 | 修改密码 | 已注册用户修改密码 |
| 104 | 注销 | 已注册用户登录后进行注销回到未登录状态 |
| 201 | 添加标签 | 用户在本地标签库中添加一个标签 |
| 202 | 查找标签 | 用户查找一个标签的具体内容 |
| 203 | 修改标签 | 用户修改标签名称或简介 |
| 204 | 获取标签列表 | 获取一个用户的全部标签信息 |
| 204 | 删除标签 | 用户删除一个标签 |
| 205 | 查看标签使用 | 用户查看一个标签的全部标注 |
| 301 | 添加文件 | 用户在本地代码库中添加一个代码文件 |
| 302 | 编辑文件 | 用户修改代码文件并保存修改 |
| 303 | 删除文件 | 用户删除一个代码文件 |
| 304 | 标注代码 | 用户文件中的一段代码添加标签和注释 |
| 305 | 查看标注信息 | 用户查看一个标注的具体信息 |
| 306 | 编辑标注信息 | 用户修改标注的注释，标签或范围 |
| 307 | 删除标注关系 | 用户删除一个标注关系 |
| 308 | 获取标注内容 | 用户查看一个标注的介绍和对应代码 |
| 309 | 获取文件标注 | 用户查看一个代码文件内的全部标注关系 |
| 310 | 更改编程语言 | 用户更改文件的编程语言和高亮显示方法 |

## 2.4 类图设计

图 1 系统类图



# 3.数据库设计

## 3.1 数据库中表名列表

表格 4 数据库中的表名列表

|  |  |  |
| --- | --- | --- |
| **编号** | **表名** | **说明** |
| 1 | User | 用户信息表 |
| 2 | Label | 标签表 |
| 3 | Code | 代码文件表 |
| 4 | Mark | 标注关系表 |

## 3.2 数据库表的详细清单

表格 5 User 用户信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 内容 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | \_id | 自动生成 | int | 否 | 主键 |
| 2 | Passwd | 密码 | varchar(20) | 否 |  |
| 3 | Username | 用户名 | varchar(20) | 否 |  |
| 4 | Telephone | 电话号码 | varchar(20) | 否 |  |

表格 6 Label标签表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | \_id | 自动生成 | int | 否 | 主键 |
| 2 | UserID | 用户名称 | varchar(20) | 否 | 外键 |
| 3 | Labelname | 标签名称 | varchar(20) | 否 |  |
| 4 | Labelintro | 标签简介 | varchar(100) | 是 |  |

表格 7 Code代码文件表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | \_id | 自动生成 | int | 否 | 主键 |
| 2 | UserID | 用户名称 | varchar(20) | 否 | 外键 |
| 3 | Codename | 文件名称 | varchar(20) | 否 |  |
| 4 | Code | 代码内容 | mediumtext | 是 |  |
| 5 | Create\_time | 修改时间 | Time | 否 |  |
| 6 | Language | 编程语言 | varchar(10) | 否 |  |

表格 8 Mark标注关系表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | \_id | 自动生成 | int | 否 | 主键 |
| 2 | Username | 用户名称 | varchar(20) | 否 | 外键 |
| 3 | Codename | 文件名称 | varchar(50) | 否 | 外键 |
| 4 | Labelname | 标签名称 | varchar(20) | 否 | 外键 |
| 5 | Code | 标注代码段 | varchar(1000) | 否 |  |

# 4.前端页面设计

后端前端采用Vue框架，主要使用JavaScript编程语言，这一部分将对主要界面的功能和使用做一个大致的介绍

## 4.1 登录界面

调用背景：

(1)当用户需要进入该网站时，跳转到该页面；

(2)当用户从注册界面完成注册后，跳转到此页面；

(3)当用户注销登录后，回到此页面；

页面组成：

一个登录框组成。登录框包含用户名和密码两个文本框，两个按钮：一个登录按钮，一个注册按钮。

调用描述：

(1)当用户未登录时，在用户名和密码填入后，点击“登录”。在数据库中进行查询，若存在此记录，若没有此记录，则在登录框中提示登录失败信息，登录成功后跳转到用户主页界面；

(2)点击“注册”按钮，则页面跳转到注册界面，用户可以使用手机号进行注册，跳转到注册界面

## 4.2 注册界面

调用背景：

(1)当用户从登录界面点击注册后，回到此页面；

页面组成：

一个注册组成。注册包含若干个文本框，如用户名，密码，确认密码，手机号等，和注册按钮。注册后会首先查询已有用户是否有相同手机号或重名，以及手机号是否合法，如若不然，返回注册失败信息，否则返回注册成功，回到登录界面。

调用描述：

(1)用户填入相关信息，点击注册按钮，完成注册（或返回注册失败及注册失败原因）

## 4.3 用户主界面

调用背景：

1. 当用户登录成功后，调用此界面；
2. 当用户在代码库和标签库点击返回时，回到此界面。

页面组成：

一个主体界面，右上导航栏可以选择代码库，标签库来查看用户存储的代码文件和标签内容。

调用描述：

(1)用户点击标签库，进入标签库主界面，用户点击代码库标签，进入代码库主界面。

## 4.4 标签库主界面

调用背景：

1. 当用户在主界面点击标签库后，进入此界面；
2. 当用户查看标签信息点击返回时，回到此界面。

页面组成：

一个列表，列出用户存储在库中的各个标签，以及标签介绍等内容。

调用描述：

(1)用户点击标签，查看标签的详细信息，以及用户使用该标签做了哪些标注；

(2)用户点击标签对应的标注，进入该代码文件的编辑器界面。

## 4.5 代码库主界面

调用背景：

1. 当用户在主界面点击代码库后，进入此界面；
2. 当用户查在代码编辑界面点击返回时，回到此界面。

页面组成：

一个列表，列出用户存储在库中的各个代码文件，以及代码文件的名称，修改时间，编程语言等信息。

调用描述：

(1)用户点击代码文件，进入代码编辑器界面；

## 4.6代码编辑器界面

调用背景：

1. 当用户在代码库界面点击代码文件后，进入此界面；
2. 当用户查看标签信息点击标签对应的代码标注时，进入此界面。

页面组成：

一个代码编辑界面，一个选项框用于更改编程语言信息，一个按钮用于提交保存，一个侧边栏列表，用于显示这个代码文件中的每个标注信息：

(1)用户在编辑器中修改代码，点击保存按钮将改动保存到代码库中；

(2)用户通过选择器修改编程语言并自动保存；

(3)用户点击表格中的标注，查看这次标注的详细信息，包含标签简介，标注的代码段等，也可以点击删除删除此次标注；

(4)用户选择一段代码，点击标注，选择标签库内已有标签或新建一个标签进行一次代码标注。

# 5.后端功能函数设计

后端采用flask框架，主要使用Python编程语言，Python提供了良好的接口连接数据库进行与数据库的交互，由于函数众多，这里选择部分有代表性的功能函数介绍其功能和参数，对于label的增删改查和对于code文件的操作基本逻辑相同，故省略了部分对于文件的操作。

## 5.1 login ()

功能描述：用户登录

入口参数：telephone，username，password

出口参数：state:是否登录成功

## 5.2 resist()

功能描述：用户注册

入口参数：telephone，username，password

出口参数：state:是否注册成功

## 5.3 exit()

功能描述：用户退出登录

入口参数：无

出口参数：无

## 5.4 addLabel()

功能描述：用户在标签库中添加一个标签

入口参数：userId，labelId，label\_intro

出口参数：state:是否添加成功

## 5.5 getLabel()

功能描述：用户获取自己的所有label

入口参数：userId,labelId

出口参数：list<label>

## 5.6 modifyLabel ()

功能描述：用户更改一个label的信息

入口参数：userId，label

出口参数：state:是否修改成功

## 5.7 addMark ()

功能描述：用户对一段代码进行标注

入口参数：userId，labelId，codeId，code(需要标注的代码段落)，new(是新建标签，还是从标签库中选择)

出口参数：state:是否添加成功

## 5.8 getCodeMark ()

功能描述：用户获取一个文档中的全部标注

入口参数：userId，codeId

出口参数：list<mark>

## 5.9 getUsers ()

功能描述：管理员获取用户列表

入口参数：无

出口参数：list<user>

## 5.10 removeUser ()

功能描述：管理员删除一个用户

入口参数：userId

出口参数：无

# 6.接口设计

## 6.1 用户接口

用Vue搭建，提供清晰简洁易用且美观的用户界面。

## 6.2 外部接口

后端采用flask框架，通过axios模板引擎进行前后端交互。

## 6.3 内部接口

本系统后端采用flask框架，通过建立db类，完成数据库接口的编写，使用Python提供的库与sqlite数据库进行交互。

# 7.系统错误处理

## 7.1出错信息

（1）对用户输入的各项内容进行有效性、安全性检查，减少错误发生的几率；

（2）对程序运行中的异常均进行捕获，按统一的方式将出错提示信息提供给用户；

## 7.2故障预防与补救

以统一的机制进行网站权限的控制，对程序中用到的数据尽量进行加密，以减少黑客攻击的危害。定期对数据库中的数据进行海量备份及增量备份。

## 7.3系统维护设计

（1）编码实现时应采用模块化和分层的思想，尽量满足模块设计的高内聚、低耦合性，使系统逻辑结构清晰，从而增强可读性和可维护性。

（2）面向数据与面向对象相结合，模块划分符合面向对象思想。

（3）在编码过程中注意标识符命名的意义，添加适量注释。

# 8.测试计划

测试计划作为本系统的测试指导，测试成员按计划和规定进行测试。本系统的测试将从单元测试、集成测试、系统测试3个方面来设计，具体测试将在测试报告中详细说明。