

## 1. 引言

### 1.1. 概要设计依据

1. “财富引擎”需求规约
2. “财富引擎”需求分析规约
3. 概要设计基本原则

### 1.2. 参考资料

1. 《SpringBoot 更好的开发》
2. “财富引擎”需求规约
3. “财富引擎”需求分析规约

### 1.3. 假定和约束

同需求分析规约中的假定和约束。

## 2. 概要设计

### 2.1. 系统总体架构设计

项目的总体架构为典型的三层架构。

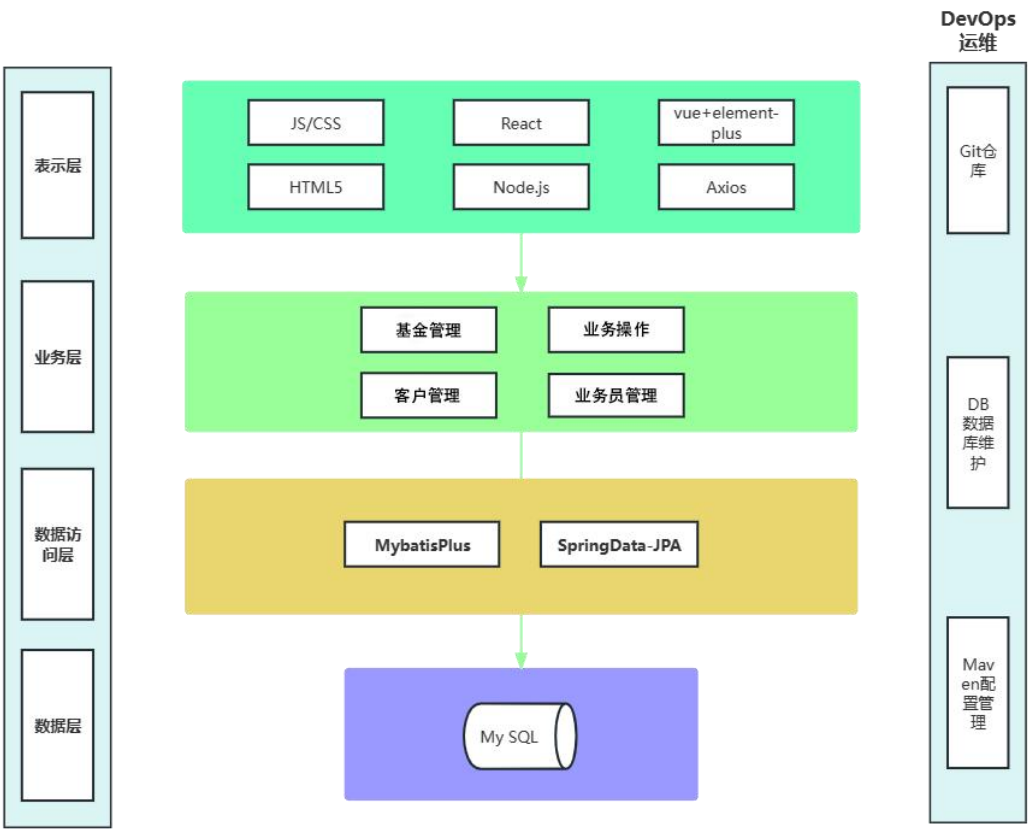
其中，表示层是用户直接与之交互的界面，它包括用户界面元素、UI 控件、用户输入、显示和呈现数据等。表示层主要用的技术为：vue+element-plus, Axios 等。

业务层包含应用程序的核心逻辑和规则，主要用于完成客户管理、业务操作、基金管理和业务员管理功能。

数据访问层负责与数据存储系统进行交互，执行数据的检索、存储、更新和删除操作，主要使用的技术是 MybatisPlus、SpringData-JPA，更加方便的和数据库进行交互。

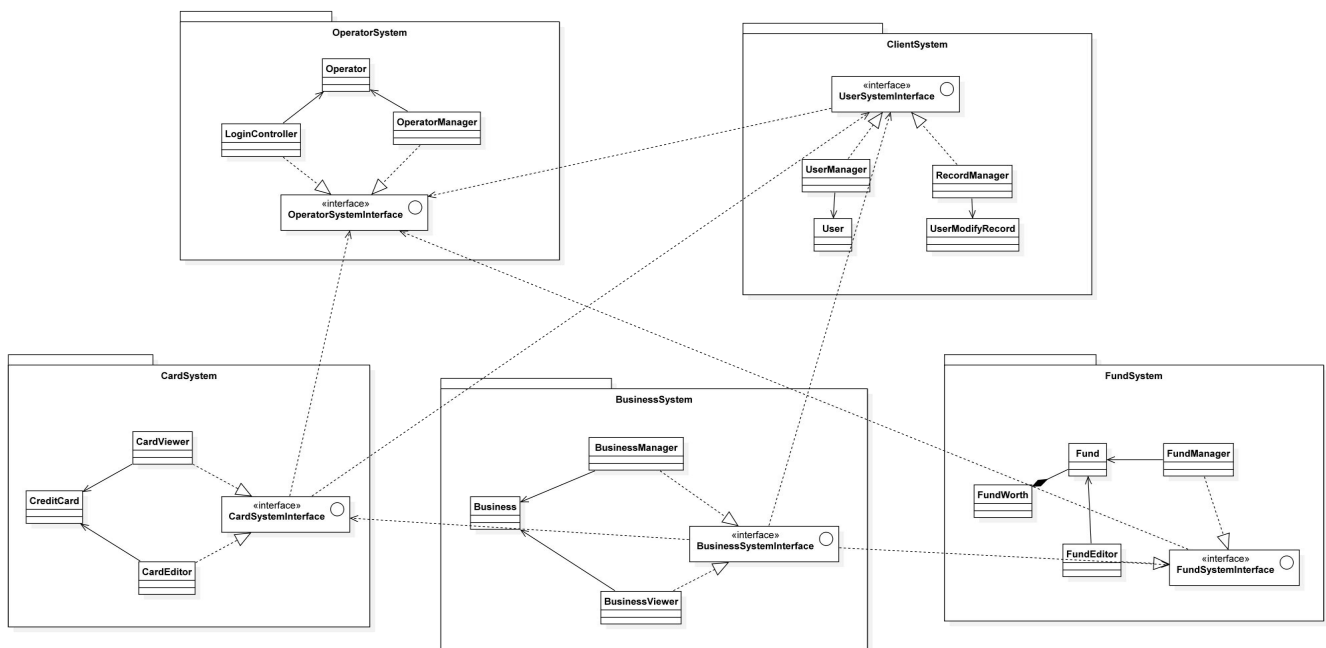
数据库的设计上，大部分的数据存储在 Mysql 数据库中，在本地进行数据规模较小的数据存储；考虑到图片上传的速度和内存问题，将图片单独存储在阿里云的服务器上，从而更加方便的获取数据，Mysql 数据库中仅需存储对应图片的 url 即可访问。

详情如下图所示：



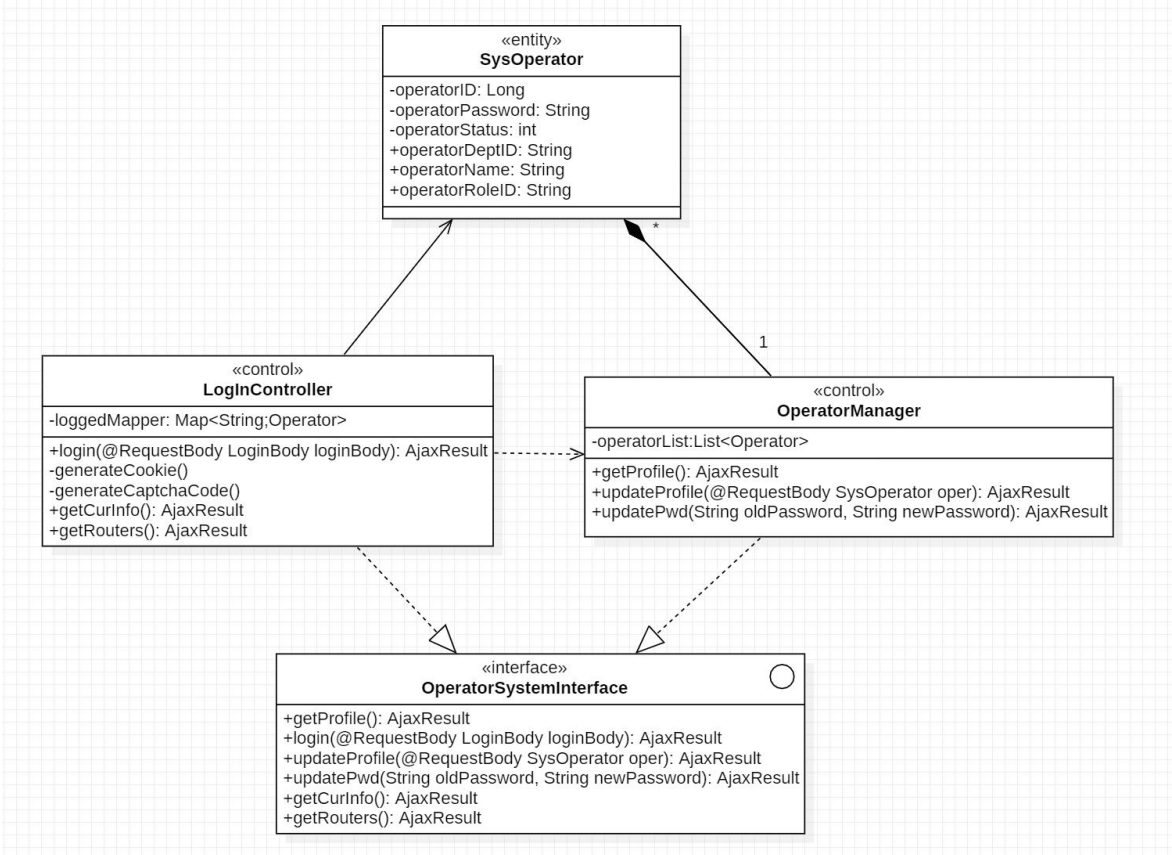
## 2.2. 系统软件结构设计

项目可分为五个子系统，分别为业务员管理系统、客户管理系统、银行卡管理系统、基金管理系统和业务操作系统。



## 2.3. 接口设计

### 2.3.1. 业务员管理子系统



### getProfile

方法签名: `public AjaxResult getProfile()`

描述: 该方法用于获取用户个人信息。

返回值：返回 AjaxResult 对象，包含用户个人信息、角色等数据。

## updateProfile

方法签名：public AjaxResult updateProfile(@RequestBody SysUser user)

描述：该方法用于修改用户个人信息。

参数：

user：SysUser 对象，包含待修改的用户个人信息。

返回值：返回 AjaxResult 对象，包含修改结果信息。

## updatePwd

方法签名：public AjaxResult updatePwd(String oldPassword, String newPassword)

描述：该方法用于重置用户密码。

参数：

oldPassword：String，旧密码。

newPassword：String，新密码。

返回值：返回 AjaxResult 对象，包含重置密码的结果信息。

## login

方法签名：public AjaxResult login(@RequestBody LoginBody loginBody)

描述：该方法用于用户登录验证。

参数：

loginBody：LoginBody 对象，包含登录信息（用户名、密码、验证码、UUID 等）。

返回值：返回 AjaxResult 对象，包含登录成功后生成的令牌。

## getInfo

方法签名：public AjaxResult getInfo()

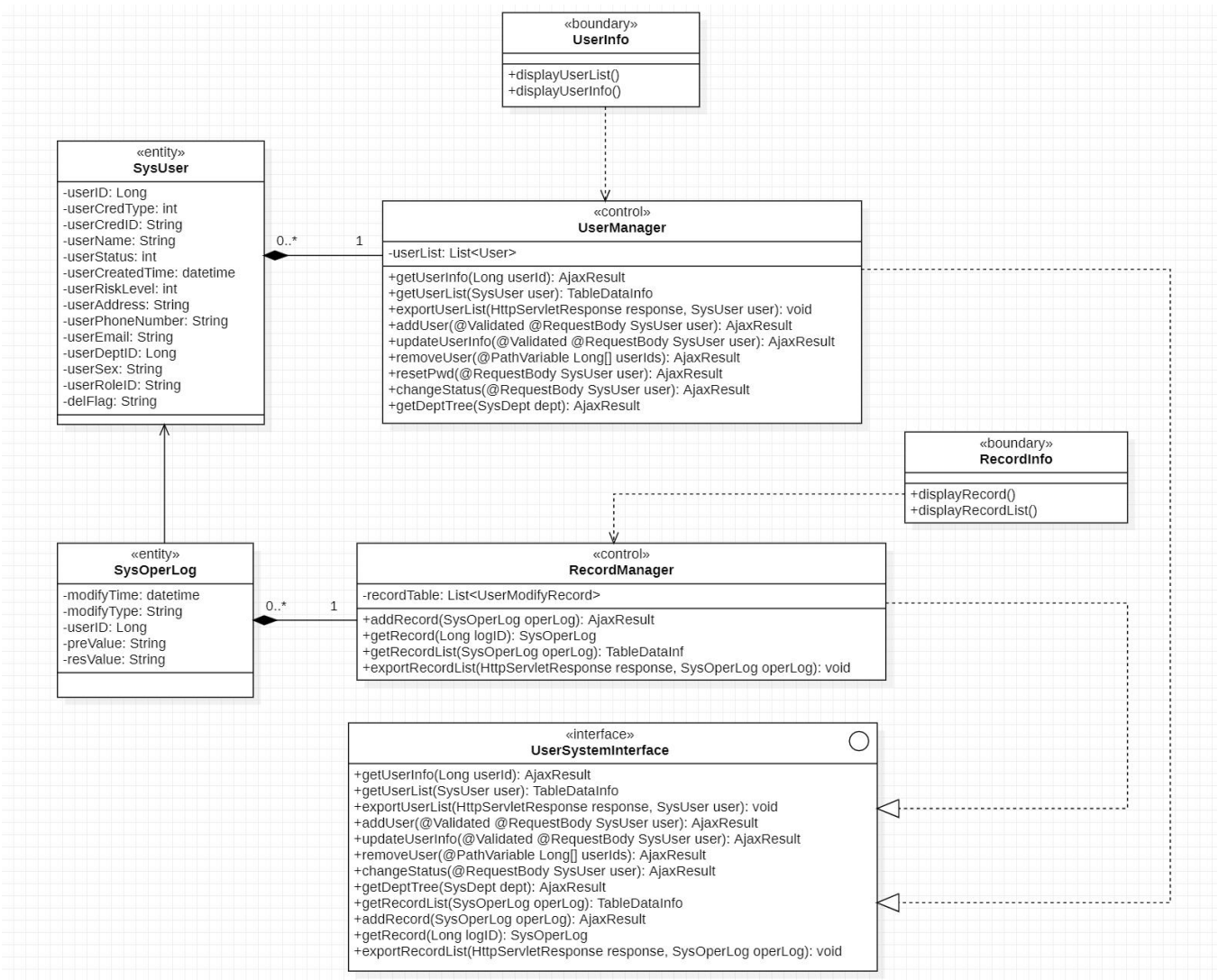
描述：该方法用于获取当前登录用户的信息，包括用户基本信息、角色集合、权限集合等。

返回值：返回 AjaxResult 对象，包含用户信息、角色集合、权限集合。

# getRouters

方法签名：public AjaxResult getRouters()  
描述：该方法用于获取当前登录用户的路由信息，返回用户可访问的菜单路由信息。  
返回值：返回 AjaxResult 对象，包含路由信息。

## 2.3.2. 客户管理子系统



# getUserLlist

方法签名: `public TableDataInfo getUserList(SysUser user)`

描述: 该方法用于获取用户列表。

参数:

user: SysUser 对象, 包含用户查询条件。

返回值: 返回 TableDataInfo 对象, 包含分页信息和用户列表数据。

## exportUserList

方法签名: `public void exportUserList(HttpServletResponse response, SysUser user)`

描述: 该方法用于导出用户数据。

参数:

response: HttpServletResponse 对象, 用于响应导出文件。

user: SysUser 对象, 包含用户查询条件

返回值: void

## getUserInfo

方法签名: `public AjaxResult getUserInfo(Long userId)`

描述: 该方法用于根据用户编号获取详细信息。

参数:

userId: Long 型, 表示用户编号。

返回值: 返回 AjaxResult 对象, 包含用户详细信息。

## addUser

方法签名: `public AjaxResult addUser(@Validated @RequestBody SysUser user)`

描述: 该方法用于新增用户。

参数:

user: SysUser 对象, 包含待新增的用户信息。

返回值: 返回 AjaxResult 对象, 包含新增结果信息。

## updateUserInfo

方法签名: `public AjaxResult updateUserInfo(@Validated @RequestBody SysUser user)`

描述：该方法用于修改用户信息。

参数：

user: SysUser 对象，包含待修改的用户信息。

返回值：返回 AjaxResult 对象，包含修改结果信息。

## removeUser

方法签名：public AjaxResult removeUser(@PathVariable Long[] userIds)

描述：该方法用于删除用户。

参数：

userIds: Long 数组，包含待删除的用户编号。

返回值：返回 AjaxResult 对象，包含删除结果信息。

## resetPwd

方法签名：public AjaxResult resetPwd(@RequestBody SysUser user)

描述：该方法用于重置用户密码。

参数：

user: SysUser 对象，包含待重置密码的用户信息。

返回值：返回 AjaxResult 对象，包含重置密码结果信息。

## changeStatus

方法签名：public AjaxResult changeStatus(@RequestBody SysUser user)

描述：该方法用于修改用户状态。

参数：

user: SysUser 对象，包含待修改状态的用户信息。

返回值：返回 AjaxResult 对象，包含修改状态结果信息。

## getDeptTree

方法签名：public AjaxResult getDeptTree(SysDept dept)

描述：该方法用于获取部门树列表。

参数：

dept: SysDept 对象，包含部门查询条件。



返回值：返回 AjaxResult 对象，包含部门树列表信息。

## **getRecordList**

方法签名：public TableDataInfo getRecordList(SysOperLog operLog)

描述：该方法用于获取操作日志列表。

参数：

operLog: SysOperLog 对象，包含操作日志查询条件。

返回值：返回 TableDataInfo 对象，包含分页信息和操作日志列表数据。

## **getRecord**

方法签名：public SysOperLog getRecord(Long logID)

描述：该方法用于获取操作日志信息。

参数：

logID: Long 型，操作记录的 ID。

返回值：返回 SysOperLog 对象，包含分页信息和操作日志列表数据。

## **addRecord**

方法签名：public AjaxResult addRecord(SysOperLog operLog)

描述：该方法用于添加日志记录。

参数：

operLog: SysOperLog 对象，包含操作日志内容。

返回值：返回 AjaxResult 对象，包含添加结果。

## **exportRecordList**

方法签名：public void exportRecordList(HttpServletResponse response,  
SysOperLog operLog)

描述：该方法用于导出操作日志数据。

参数：

response: HttpServletResponse 对象，用于响应导出文件。

operLog: SysOperLog 对象，包含操作日志查询条件。

返回值：void

### 2.3.3. 其余子系统

## 2.4. 界面设计

### 2.4.1. 登录界面设计



The image shows the login interface of the TWWS Fund Management System. The title 'TWWS基金管理系统' is centered at the top. Below it are three input fields: a username field with 'admin' entered, a password field with masked characters, and a verification code field. To the right of the verification code field is a small box displaying the math problem '9+8=?'. Below these fields is a checkbox labeled '记住密码' (Remember Password). At the bottom is a large blue button labeled '登录' (Login).

TWWS基金管理系统

admin

.....

验证码

9+8=?

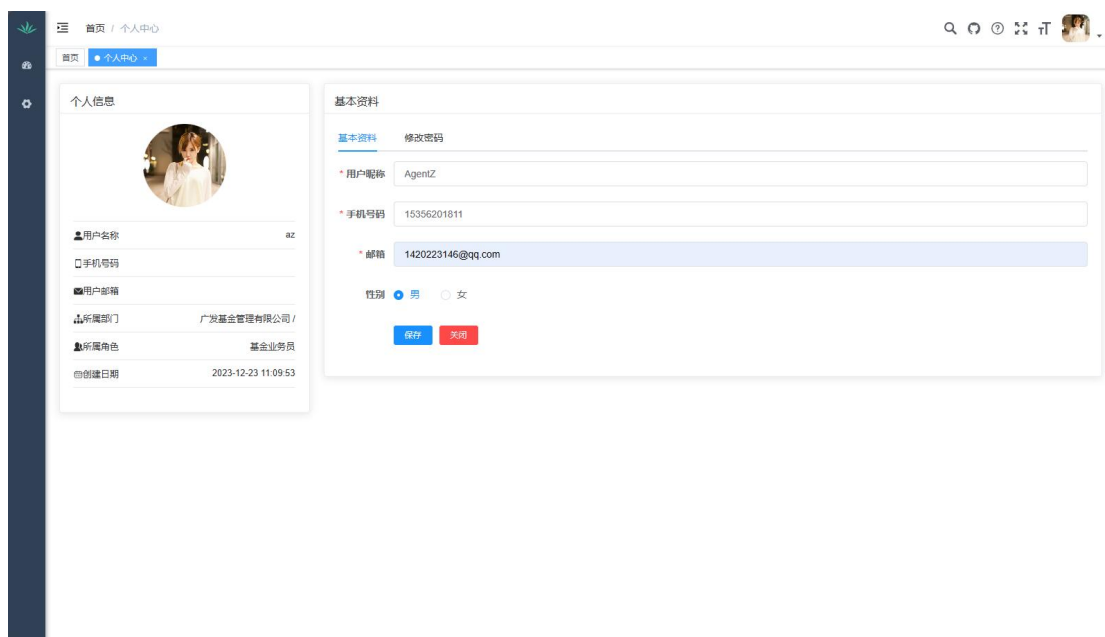
☐ 记住密码

登录

### 2.4.2. 主页设计



### 2.4.3. 个人中心



### 2.4.4. 修改信息界面

基本资料

基本资料

修改密码

\* 用户昵称

AgentZ

\* 手机号码

15356201811

\* 邮箱

1420223146@qq.com

性别

☒ 男

☐ 女

保存

关闭

2.4.5. 修改密码界面

基本资料

基本资料

修改密码

\* 旧密码

请输入旧密码

\* 新密码

请输入新密码

\* 确认密码

请确认新密码

保存

关闭

2.4.6. 用户管理界面

用户管理

请输入支行名称

请输入用户名称

请输入手机号码

用户状态

广发基金管理有限公司  
京东金融

创建时间

开始日期

结束日期

风险类型

用户风险类型

搜索

重置

+新增

修改

删除

导出

	用户编号	用户名称	所属基金公司/支行	手机号码	状态	创建时间	风险类型	操作
<input type="checkbox"/>	108	郑先生	京东金融	15356201810	<input checked="" type="checkbox"/>	2023-12-23 16:40:50	稳健型	<a href="#">详情</a> <a href="#">删除</a>
<input type="checkbox"/>	110	pz	京东金融	15356101111	<input checked="" type="checkbox"/>	2023-12-24 07:01:53	保守型	<a href="#">详情</a> <a href="#">删除</a>

共 4 条

10条/页

<

1

>

前往 1 页

## 2.4.7. 开户界面

添加用户

\* 用户昵称

请输入用户昵称

\* 所属支行

请选择归属支行

\* 手机号码

请输入手机号码

邮箱

请输入邮箱

\* 证件类型

请选择证件

\* 证件号码

请输入证件号码

\* 用户名称

请输入用户名称

\* 用户密码

.....

用户性别

请选择性别

状态

☒ 正常 ☐ 停用

\* 风险等级

请选择风险等级

\* 权限

请选择权限

备注

请输入内容

确定

取消

## 2.4.8. 用户信息详情/修改信息界面

用户详情

\* 用户昵称

郑先生

\* 所属支行

京东金融

\* 手机号码

15356201810

邮箱

请输入邮箱

\* 证件类型

身份证

\* 证件号码

111

用户性别

男

状态

正常

停用

\* 风险等级

稳健型

\* 权限

客户

备注

请输入内容

确定

取消

2.4.9. 基金管理界面

基金名称 请输入基金名称 基金规模 基金经理 请输入基金经理 成立日 请选择基金成立日 所属公司 请选择归属部门

基金评级 风险评级 基本概况 请输入基金基本概况

新增 修改 删除 导出

基金代码	基金名称	基金类型	基金规模	基金经理	成立日	所属公司	基金评级	风险评级	基本概况	基金状态	操作
1	1234		10-30亿	test	2023-12-24	105	中等	高风险	test01		修改 删除
2	普		10-30亿		2023-12-05	104	中等	中风险			修改 删除
3	混合货币C		100亿以上	zwc	2023-12-28	105	极佳	低风险			修改 删除
4	社保投资A		10-30亿	zk	2023-12-28	107	合格	中风险			修改 删除
5	同济大学科技园		30-100亿	小郑	2024-01-01	104	极佳	高风险			修改 删除

共 5 条 10条/页 1 前往 1 页

2.4.10. 新增基金界面

添加基金管理

×

基金名称

请输入基金名称

基金规模

☆☆☆☆☆

基金经理

请输入基金经理

成立日

目 请选择基金成立日

所属公司

请选择归属部门

基金评级

☆☆☆☆☆

风险评级

☆☆☆

基本概况

请输入基金基本概况

确定

取消

2.4.11. 修改基金信息界面

修改基金管理

基金名称

同济大学科技园

基金规模

★★★★☆ 30-100亿

基金经理

小郑

成立日

📅 2024-01-01

所属公司

广发基金管理有限公司

基金评级

★★★★★ 极佳

风险评级

★★★ 高风险

基本概况

请输入基金基本概况

确定

取消

2.4.12. 银行卡管理界面

首页 / 系统管理 / 银行卡管理

银行卡管理

银行卡号

请输入银行卡号

持卡人ID

请输入持卡人ID

搜索

重置

+新增

修改

删除

导出

	银行卡号	持卡人ID	金额	卡状态	操作
<input type="checkbox"/>	114514	106	100		修改 删除
<input type="checkbox"/>	19999999	107	103		修改 删除
<input type="checkbox"/>	2154101	101	2500		修改 删除
<input type="checkbox"/>	2154314	101	24500		修改 删除

共 4 条

10条/页

< 1 >

前往 1 页



2.4.13. 添加银行卡界面

绑定银行卡

×

银行卡ID

请输入银行卡ID

持卡人ID

请输入持卡人ID

金额

请输入金额

确定

取消

2.4.14. 修改银行卡信息界面

修改银行卡信息

×

银行卡ID

2154101

持卡人ID

101

金额

2500

确定

取消

2.4.15. 业务操作界面

首页 / 系统管理 / 业务管理

基金管理 银行卡管理 业务管理

用户ID 请输入用户ID 银行卡号 请输入银行卡号 基金代码 请输入基金代码 交易时间 请选择交易时间 搜索 重置

+ 添加业务 业务清算

交易流水编号	用户ID	银行卡号	基金代码	交易时间	交易金额	交易状态	交易类型	操作
1	106	114514	1			已撤回	申购业务	自撤销业务
2	106	114514	1	2023-12-26	6	已清算	申购业务	自撤销业务
3	106	114514	2	2023-12-26	10	已清算	赎回业务	自撤销业务
4	106	114514	2	2023-12-26	10	已清算	赎回业务	自撤销业务
5	106	114514	1	2023-12-26	1	已清算	赎回业务	自撤销业务
6	106	114514	1	2023-12-26	200	已清算	申购业务	自撤销业务
7	106	114514	2	2023-12-27	100	已撤回	申购业务	自撤销业务
8	101	2154314	1	2023-12-27	200	已清算	申购业务	自撤销业务
9	101	2154314	1	2023-12-27	300	已清算	申购业务	自撤销业务

共 9 条 10条/页 < 1 > 前往 1 页

2.4.16. 添加业务界面

添加业务

\* 用户ID

请输入用户ID

\* 银行卡号

请输入银行卡号

\* 基金代码

请输入基金代码

\* 交易金额

请输入交易金额

\* 交易类型

请选择交易类型

确定

取消

2.4.17. 清算界面

首页 / 系统管理 / 业务管理

无需清算!

业务管理

用户ID: 请输入用户ID 银行卡号: 请输入银行卡号 基金代码: 请输入基金代码 交易时间: 请选择交易时间 搜索 重置

+ 添加业务 业务清算

2.4.18. 通告界面

首页 / 系统管理 / 通知公告

通知公告

公告标题: 请输入公告标题 操作人员: 请输入操作人员 类型: 公告类型 搜索 重置

序号	公告标题	公告类型	状态	创建者	创建时间	操作
10	关于基金修改Bug修复	公告	正常	ZWC	2023-12-27	
11	维护通知	公告	正常	ZWC	2023-12-27	

共 2 条 10条/页 1 前往 1 页

2.4.19. 日志查看界面

首页 / 系统管理 / 操作日志

操作日志

操作地址: 请输入操作地址 系统模块: 请输入系统模块 操作人员: 请输入操作人员 类型: 操作类型

状态: 操作状态 操作时间: 开始日期 - 结束日期 搜索 重置

删除 清空 导出

日志编号	系统模块	操作类型	操作人员	操作地址	操作地点	操作状态	操作日期	消耗时间	操作
377	基金管理	新增	az	127.0.0.1	内网IP	成功	2024-01-11 14:21:43	45毫秒	详情
376	个人信息	修改	admin	127.0.0.1	内网IP	成功	2024-01-07 06:24:29	78毫秒	详情
375	银行卡管理	修改	admin	127.0.0.1	内网IP	成功	2023-12-28 07:20:54	43毫秒	详情
374	trans	修改	admin	127.0.0.1	内网IP	成功	2023-12-28 07:20:54	65毫秒	详情
373	trans	新增	az	127.0.0.1	内网IP	成功	2023-12-28 07:06:48	51毫秒	详情
372	用户管理	新增	Honglei Fan	127.0.0.1	内网IP	成功	2023-12-28 07:02:13	307毫秒	详情
371	角色管理	修改	admin	127.0.0.1	内网IP	成功	2023-12-28 06:41:19	225毫秒	详情
370	银行卡管理	修改	admin	127.0.0.1	内网IP	成功	2023-12-28 06:37:52	38毫秒	详情
369	trans	修改	admin	127.0.0.1	内网IP	成功	2023-12-28 06:37:52	41毫秒	详情
368	trans	新增	admin	127.0.0.1	内网IP	成功	2023-12-28 06:37:36	36毫秒	详情

共 278 条 10条/页 1 2 3 4 5 6 ... 28 前往 1 页

## 2.5 数据库设计

### 2.5.1. 概述

在本基金业务管理系统中，最核心的功能由业务员执行，其职责是为特定的客户进行基金交易相关的操作。

我们基于项目的需求分析，围绕基金管理业务设计了若干实体类：

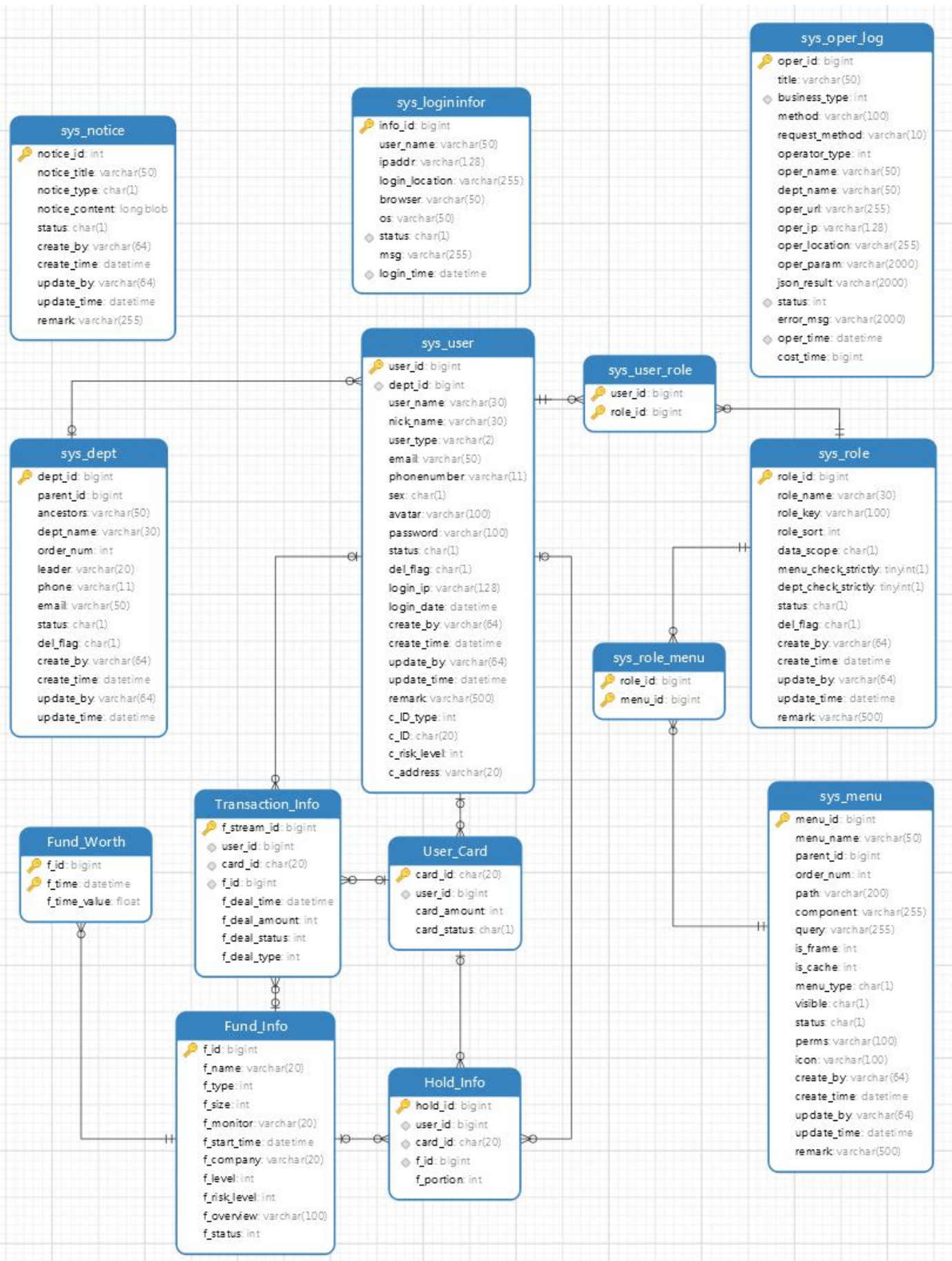
- 用户（包括客户和业务员）
- 部门（每个用户属于一个部门）
- 银行卡（用户持有）
- 基金基本信息
- 基金的估值信息（随时间变动）
- 交易记录（包括申购与赎回）
- 持仓记录（客户目前持有的基金）

此外还有一些与系统相关的信息：

- 菜单视图
- 登录与操作日志
- 用户的角色与权限
- 用户发布的通知或公告

因此，我们在数据库中设计了如下 14 张表格

（下图展示了各表的基本属性和它们之间的基数映射关系）



### 2.5.2. 表设计

#### 2.5.2.1. 用户信息(sys\_user)

同时存储了业务员与客户的信息

序号	字段名	数据类型	字段说明	备注
1	user_id	bigint	用户内部编码	PK
2	c_ID_type	int	用户证件类型	数字对应类型（身份证=0，护照=1）
3	c_ID	char(20)	用户证件号码	
4	user_name	varchar(30)	用户姓名	也是账户名
5	status	char(1)	用户状态	正常=0，停用=1
6	create_time	datetime	用户注册时间	
7	c_risk_level	int	用户风险等级	0、1、2 分别对应低、中、高
8	c_address	char(20)	用户地址	
9	email	varchar(50)	用户邮箱	
10	phonenum	varchar(11)	用户电话	
11	dept_id	bigint	部门 ID	FK from sys_dept(dept_id)
12	user_type	varchar(2)	用户类型	业务员=0，客户=1

#### 2.5.2.2. 登录信息 (sys\_logininfor)

存储每个用户的登录历史记录

序号	字段名	数据类型	字段说明	备注
1	info_id	bigint	访问记录 id	PK
2	user_name	varchar (30)	用户姓名	也是账户名
3	ipaddr	char (128)	访问者 IP 地址	
4	login_location	varchar (255)	登录地点	
5	browser	varchar (50)	浏览器类型	
6	os	varchar (50)	操作系统	
7	status	char (1)	登录状态	成功=0，失败=1
8	msg	varchar (255)	提示消息	
9	logintime	datetime	访问时间	

#### 2.5.2.3. 角色信息(sys\_role)

存储了系统用户的所有角色

序号	字段名	数据类型	字段说明	备注
1	role_id	bigint	角色 id	PK

2	role_name	varchar(30)	角色姓名	
3	role_key	char(100)	用户权限字符串	英文单词
4	role_sort	int	排序优先级	
5	data_scope	char(1)	角色访问权限	1: 全部数据权限 2: 自定数据权限 3: 本部门数据权限 4: 本部门及以下数据权限
6	status	char(1)	角色状态	正常=0, 停用=1
7	create_by	varchar(64)	创建者	
8	create_time	datetime	创建时间	

#### 2.5.2.4. 用户角色信息(sys\_user\_role)

用户信息表与角色信息表之间的关系表

序号	字段名	数据类型	字段说明	备注
1	user_id	bigint	用户 id	PK, FK from sys_user(user_id)
2	role_id	bigint	角色 id	PK, FK from sys_role(role_id)

#### 2.5.2.5. 菜单信息(sys\_menu)

存储了系统的菜单结构，主要服务于前端界面

序号	字段名	数据类型	字段说明	备注



1	menu_id	bigint	菜单 id	PK
2	menu_name	varchar (50)	菜单名称	
3	parent_id	bigint	父菜单 id	
4	order_num	int	排序优先级	
5	path	char (200)	路由地址	
6	component	char (255)	组件路径	
7	query	char (255)	路由参数	
8	menu_type	char (1)	菜单类型	M 目录 C 菜单 F 按钮
9	visible	char (1)	显示状态	显示=0, 隐藏=1
10	status	char (1)	菜单状态	正常=0, 停用=1
11	create_by	varchar (64)	创建者	
12	create_time	datetime	创建时间	

2.5.2.6. 用户角色信息(sys\_role\_menu)

角色信息表和菜单信息表之间的关系表

即角色对菜单的访问权限。

序号	字段名	数据类型	字段说明	备注
----	-----	------	------	----

1	role_id	bigint	角色 id	PK, FK from sys_role(role_id)
2	menu_id	bigint	菜单 id	PK, FK from sys_menu(menu_id)

#### 2.5.2.7. 部门信息 (sys\_dept)

部门实际上是树状结构，该表对其进行了扁平化处理

序号	字段名	数据类型	字段说明	备注
1	dept_id	bigint	部门编码	PK
2	parent_id	bigint	父部门 id	
3	ancestors	varchar (50)	祖级列表	parent_id 组成的列表，由逗号分隔
4	dept_name	varchar (30)	部门名称	
5	order_num	int	显示顺序	排序优先级
6	leader	varchar (20)	负责人	
7	phone	varchar (11)	联系电话	
8	email	varchar (50)	邮箱	
9	status	char (1)	部门状态	正常=0，停用=1
10	create_by	varchar (64)	创建者	
11	create_time	datetime	创建时间	

#### 2.5.2.8. 操作信息 (sys\_oper\_log)

存储每个用户的操作日志

序号	字段名	数据类型	字段说明	备注
1	oper_id	bigint	日志 id	PK
2	title	varchar(30)	模块标题	也是账户名
3	business_type	int	业务类型	其它=0, 新增=1, 修改=2, 删除=3
4	method	varchar(100)	方法名称	
5	request_method	varchar(10)	请求方式	
6	operator_type	int	操作类别	其它 0, 后台用户=1, 手机端用户=2
7	oper_name	char(50)	操作人员	
8	dept_name	char(50)	部门名称	
9	oper_url	varchar(255)	请求 URL	
10	oper_ip	char(128)	操作者地址	
11	oper_location	varchar(255)	操作者地点	
12	status	char(1)	操作状态	正常=0, 失败=1
13	error_msg	varchar(2000)	错误信息	

14	oper_time	datetime	操作时间	
----	-----------	----------	------	--

#### 2.5.2.9. 通知信息 (sys\_notice)

用户（通常为管理员）在系统中发布的公告

序号	字段名	数据类型	字段说明	备注
1	notice_id	int	通知 ID	PK
2	notice_title	varchar (50)	通知标题	
3	notice_type	char (1)	公告类型	通知=1，公告=2
4	notice_content	longblob	公告内容	
5	status	char (1)	公告状态	正常=0，停用=1
6	create_by	varchar (64)	创建者	
7	create_time	datetime	创建时间	

#### 2.5.2.10. 银行卡 (User\_Card)

每位用户可以绑定多张银行卡，但同一笔交易只能使用其中一张

序号	字段名	数据类型	字段说明	备注
1	card_id	char (20)	卡号	PK
2	user_id	bigint	用户内部编码	FK from sys_user (user_id)

3	card_amount	int	金额	
4	card_status	char(1)	卡状态	

2.5.2.11. 基金信息 (Fund\_Info)

本系统的核心数据信息

序号	字段名	数据类型	字段说明	备注
1	f_id	varchar(6)	基金代码	PK
2	f_name	varchar(20)	基金名称	
3	f_type	int	基金类型	数字对应类型，详情见“约束条件”
4	f_size	int	基金规模	
5	f_monitor	varchar(20)	基金经理	
6	f_start_time	datetime	基金成立日	
7	f_company	varchar(20)	基金所属公司	
8	f_level	int	基金评级	数字对应类型，详情见“约束条件”
9	f_risk_level	int	基金风险评级	数字对应类型，详情见“约束条件”
10	f_overview	varchar(100)	基金基本概况	
11	f_status	int	基金状态	数字对应类型，详情见“约束条件”

#### 2.5.2.12. 基金估值 (Fund\_Worth)

实际上是基金的多值属性，因为估值随时间变动

序号	字段名	数据类型	字段说明	备注
1	f_id	varchar (20)	基金代码	PK, FK from Fund_Info(f_id)
2	f_time	datetime	估值时间	PK
3	f_time_value	float	净值	

#### 2.5.2.13. 交易记录表 (Transaction\_Info)

同时记录了申购与赎回的操作，具体类型由字段 f\_deal\_type 区分

序号	字段名	数据类型	字段说明	备注
1	f_stream_id	bigint	交易流水编号	PK
2	user_id	bigint	用户内部代码	FK from sys_user(user_id)
3	card_id	char (20)	使用的银行卡号	FK from User_Card(card_id)
4	f_id	varchar (20)	基金代码	FK from Fund_Info(f_id)
5	f_deal_date	datetime	交易时间	
6	f_deal_amount	int	交易金额	

7	f_deal_status	int	交易状态	数字对应类型，详情见“约束条件”
8	f_deal_type	int	交易类型	0 申购 1 赎回

#### 2.5.2.14. 持仓记录(Hold\_Info)

客户当前持有的基金记录，随着基金的申购和赎回而变动

序号	字段名	数据类型	字段说明	备注
1	hold_id	bigint	持仓记录的 id	PK
1	user_id	bigint	用户内部代码	FK from sys_user(user_id)
2	card_id	char(20)	绑定银行卡号	FK from User_Card(card_id)
3	f_id	varchar(20)	基金编号	FK from Fund_Info(f_id)
4	f_portion	int	基金份额	

## 2.6 系统出错处理设计

### 2.6.1 出错信息

本系统高度设计了涉及用户个人隐私、财产和支付等关键信息，因此需要得到较好保护，需要采用了严密的出错处理机制和系统故障后的补救措施，以确保系统的可用性和用户体验。系统在处理错误时，区分了软错误和硬错误，并针对性地制定了详细的处理策略。

对于软错误，我们实施了一系列即时响应和用户友好的处理机制。当用户输入数据格式不符合规范时，系统会立即在前端展示详细的错误提示，并提供可能的修正建议，使用户能够及时发现并纠正输入错误，提高了系统的用户友好性。同时，对于表单信息不完整的情况，系统也会在前端显示相应的错误信息，并引导用户完善必填项，以防止遗漏信息而导致错误。此外，当用户的输入中包含非法字符时，系统会立即提醒用户，防范潜在的安全风险，并清晰地指示包含非法字符的位置，帮助用户纠正输入。

针对硬错误，我们关注了一些关键的处理场景。在数据库连接失败的情况下，系统将尝试自动重新连接，并在前端显示错误提示，提示用户稍后再试。同时，系统会记录相关的日志，以便于后续的故障排查和修复。

以下以一览表的形式说明每种可能出现的软错误和硬错误发生时，系统输出信息的形式，含义及处理方法：



错误类型	错误信息	处理方法	系统输出信息
数据库错误	数据库连接失败	尝试重连	前端输出错误提示信息
数据库错误	数据库操作失败	事务回滚，输出错误信息	前端输出更新失败信息
数据库错误	数据库溢出	事务回滚，输出错误信息	前端输出更新失败信息
TCP 连接错误	连接超时或断开	尝试重新连接，特定时间后输出错误信息	尝试失败后输出对应的网络错误信息
系统自定义错误	用户名密码错误	不予登录	系统输出输入错误提示信息，登录失败
系统自定义错误	身份证输入错误	不予注册	系统输出错误的身份证提示信息，登录失败
系统自定义错误	验证码错误/为空	不予登录，切换验证码	系统输出验证码错误提示信息，登录失败
系统自定义错误	修改个人信息错误	不予修改	系统输出不满足条件的错误修改提示信息，修改失败
系统自定义错误	修改客户信息错误	不予修改	系统输出不满足条件的错误修改提示信息，修改失败
系统自定义错误	修改基金信息错误	不予修改	系统输出不满足条件的错误修改提示信息，修改失败
系统自定义错误	支付错误	系统回滚事务	系统输出支付失败信息并跳转回待支付界面
链接错误	图片链接错误	抛出异常	系统展示占位图片信息，提示图片展示失败
链接错误	外部链接错误	抛出异常	系统提示外部链接错误信息
程序错误	空指针错误	服务停止重启	系统提示内部错误并重启服务

错误类型	错误信息	处理方法	系统输出信息
程序错误	内部程序错误	服务重启	系统提示内部错误并重启服务
其他错误	服务器故障	无	系统提示服务器故障信息并等待人为修复重启

2.6.2. 补救措施

在系统设计中，为了应对各种可能的错误和故障，制定了综合的补救措施，旨在最小化潜在损失、确保用户体验和系统的稳定性。以下是在系统中实施的主要补救措施：

错误信息记录与监控：对于系统产生的错误和异常情况，建立了完善的日志记录系统。这包括软件错误、硬件故障、数据库连接问题等。通过对日志进行监控和分析，能够及时发现并追踪系统中出现的问题，有助于快速定位并解决潜在的故障点。

自动恢复机制：针对一些可能的软错误和临时性硬错误，系统具备自动恢复机制。例如，在数据库连接失败后，尝试自动重新连接，降低了因短暂故障而导致的影响。

用户交互确认机制：针对用户可能导致的操作误差，引入了用户交互确认机制。对于关键操作，如删除操作，系统会在用户执行前进行二次确认，避免用户误删或误改关键信息。

事务回滚与数据备份：在发生数据库操作失败的情况下，系统采用事务回滚的方式，确保数据的完整性。同时，定期进行数据备份，以防止数据丢失，并能够在需要时迅速进行恢复。

系统状态监测与预警：实施了系统状态监测与预警机制，通过监控关键性能指标和系统运行状态，及时发现潜在的问题。一旦系统出现异常，能够迅速采取相应措施，防止问题扩大化。

安全性措施：针对用户隐私和重要信息，实施了严格的安全措施，包括加密存储、访问控制等。在发生安全性问题时，系统会自动启动相应的应急措施，防范潜在的风险。

### 2.6.3. 系统维护设计

模块化设计：系统采用了模块化的架构，将功能划分为独立的模块。每个模块都有清晰的接口和功能，使得在维护时可以更容易地定位、理解和修复问题。这有助于提高维护效率和降低潜在的错误传播风险。

文档化：我们重视系统文档的编写和更新。详细的技术文档包括系统结构、模块功能、接口说明、数据库设计等方面的信息，为维护人员提供了全面的了解系统的参考资料。文档的及时更新也确保了维护人员获取最新信息的能力。

版本控制：引入版本控制系统，例如 Git，以跟踪系统代码的变化。通过版本控制，我们能够追溯每个版本的变更记录，方便排查问题和回滚到稳定的版本。这有助于降低系统在维护过程中的风险。

自动化测试：配置了自动化测试框架，覆盖了系统的关键功能和模块。在维护前，通过自动化测试可以及时发现潜在的问题，保障系统在更新后的稳定性。同时，自动化测试也减轻了维护人员的工作负担。

远程监控和告警系统：引入远程监控和告警系统，对系统关键性能指标进行实时监控。一旦系统出现异常，系统会发出告警通知，并提供详细的错误信息。这使得维护人员可以迅速响应问题，缩短故障修复时间。

定期巡检和维护：制定了定期巡检和维护计划，包括数据库优化、系统性能调整、安全漏洞修复等。定期的系统维护可以有效地预防潜在问题，提高系统的可用性和稳定性。