

## 软件工程课程设计案例

### 问题描述：

在金融交易和融资领域，违约是指融资主体、发行人或交易对手客户违反交易规则、合同约定的情形，包括未在规定期限内了结债务、未能按合约规定支付或延期支付利息或本金的行为。违约行为可能导致严重的经济损失，因此对于金融机构和投资者来说，识别违约风险并及时做出应对措施是至关重要的。为了提高对违约风险的识别和管理能力，我们计划开发一个违约认定系统。该系统旨在预先识别潜在违约客户，并根据可能发生的违约原因进行人工认定申请。本项目将重点实现系统中的人工认定维护功能，通过该功能，相关工作人员可以进行违约认定的处理和查询，根据特定情况，对某些客户进行手动提交认定申请，以便进行更加精准的违约风险判断。

项目分工：系统分析：22050233 郑方昊， 22050230 吴博涵

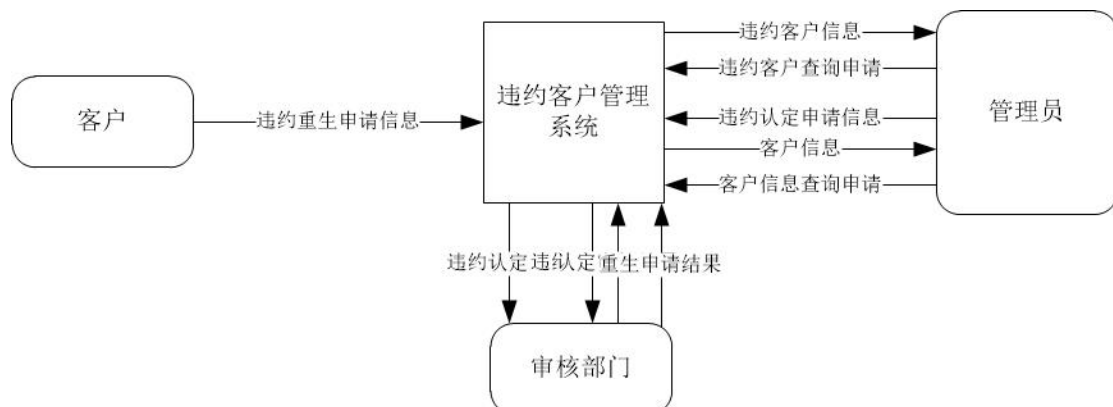
系统设计：22050233 郑方昊， 22050230 吴博涵

系统实现：22050233 郑方昊， 22050230 吴博涵

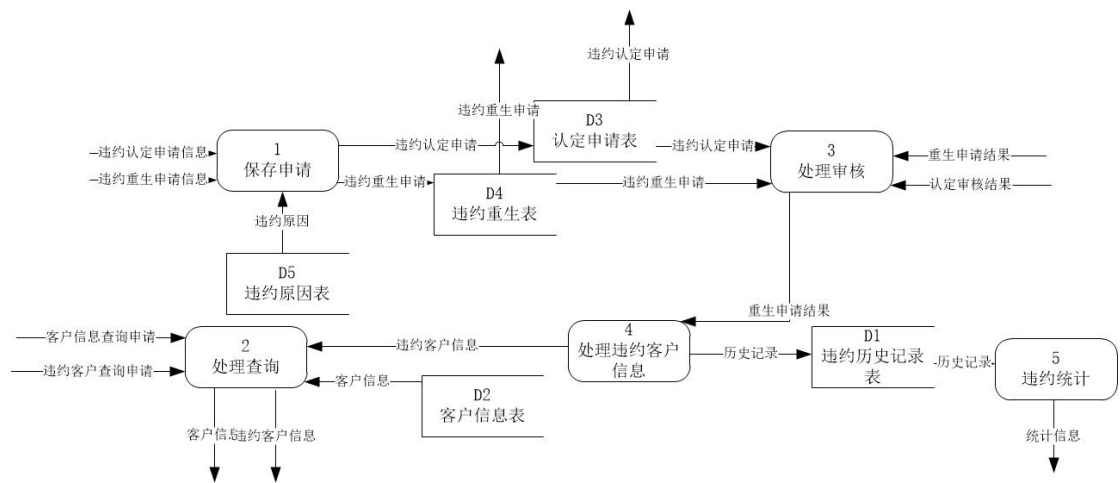
## 1. 可行性研究

### 1.1 数据流图

#### (1) 顶层数据流图



(2) 第一层数据流图



(3) 第二层数据流图

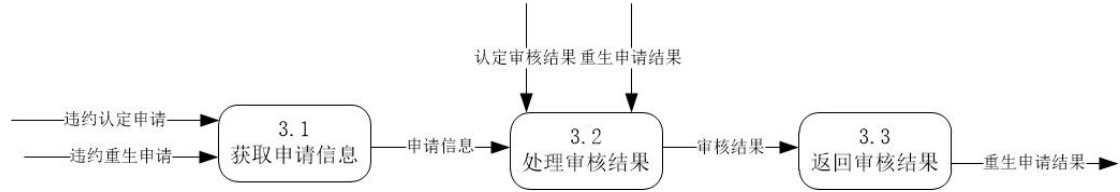
1) “1 保存申请”加工处理



2) “2 处理查询”加工处理



3) “3 处理审核”加工处理



1.2 数据字典

违约认定申请信息=客户名称+最新外部等级+违约原因+违约严重性+备注信息  
+附件

违约重生申请信息=客户名称+最新外部等级+违约原因+违约严重性+认定人+  
重生原因

客户名称=0[字符]30

最新外部等级=[0|1|`"|10]

认定人=0[字符]30

违约严重性=[低|中|高]

查询申请=[客户信息查询申请|违约客户查询申请]

客户信息查询申请=客户名+客户查询请求

违约客户查询申请=违约客户查询请求

违约认定申请=违约认定申请信息+审核结果+操作

违约重生申请=违约重生申请信息+审核结果+操作

客户信息=客户名+行业+地区+是否违约

是否违约=[是|否]

违约客户信息=认定违约原因+严重程度+认定人+认定申请时间

认定申请时间=年+月+日+时+分+秒

认定违约原因=[违约原因 1|违约原因 2|违约原因 3|`"""]

## 2. 需求分析

### 2.1 数据分析

#### (1) 实体

客户表: 客户名, 行业, 地区

认定申请表: 客户名称, 最新外部等级, 违约原因, 违约严重性, 备注信息, 审核状态, 审核结果, 附件地址, 认定人, 认定申请时间

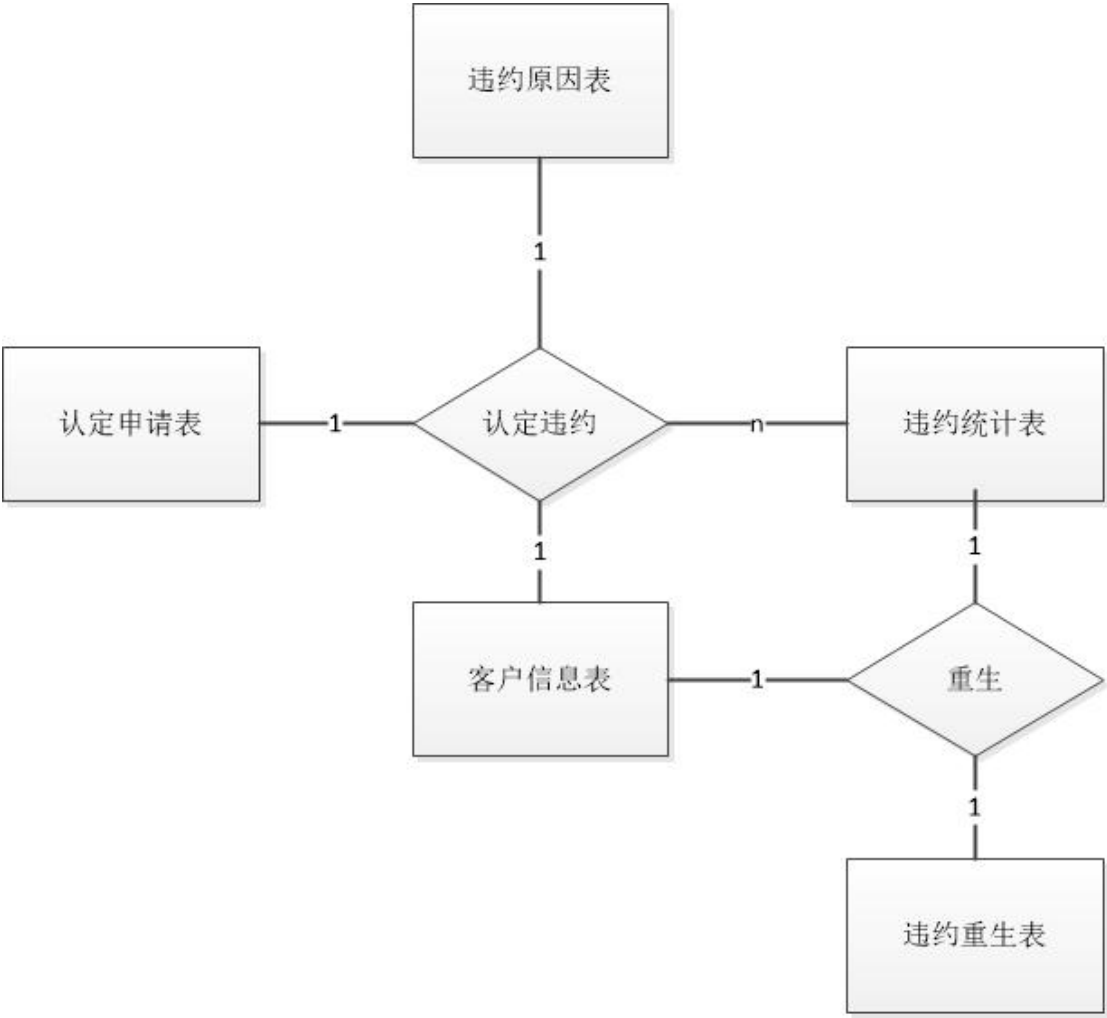
违约重生表: 违约客户, 审核状态, 认定审核原因, 严重程度, 认定人, 认定申请时间, 认定审核时间, 最新外部等级, 审核结果, 重生原因

历史数据表：地区，行业，时间

(2) 联系

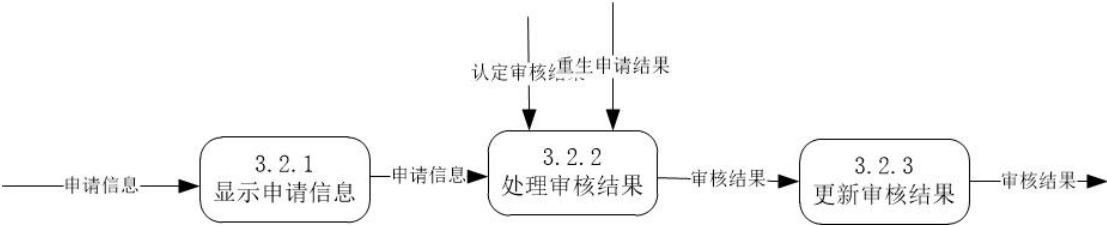
违约原因表：违约原因序号，违约原因

(3) 实体联系

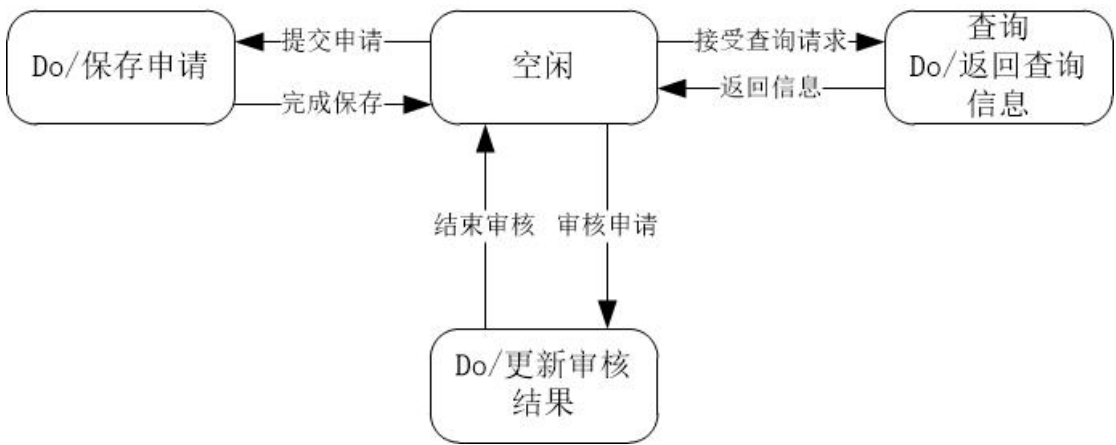


2.2 功能分析

(1) “3.2 处理审核结果”的细化



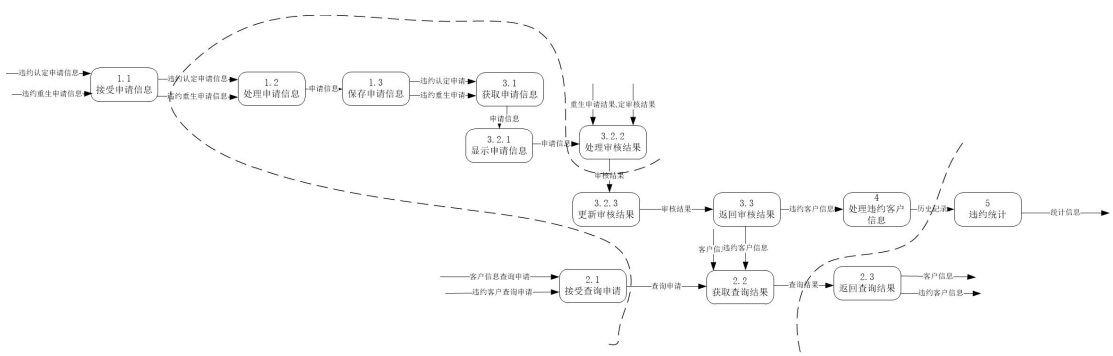
2.3 行为分析



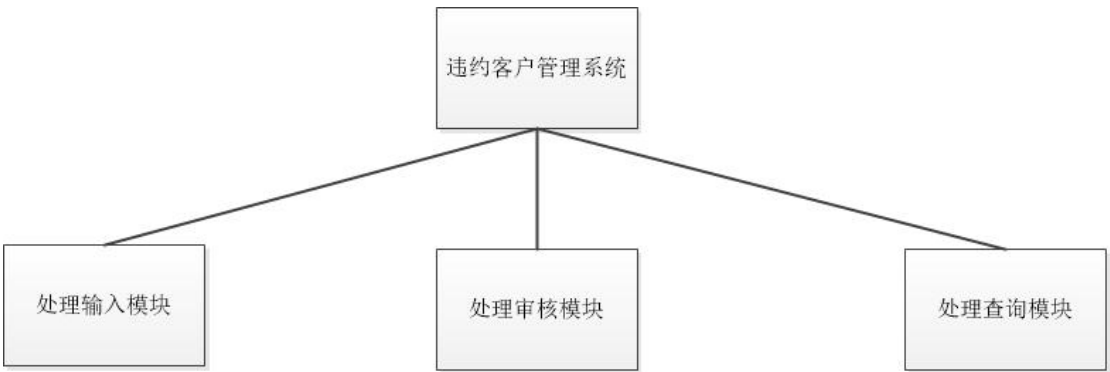
3.总体设计

3.1 软件结构设计

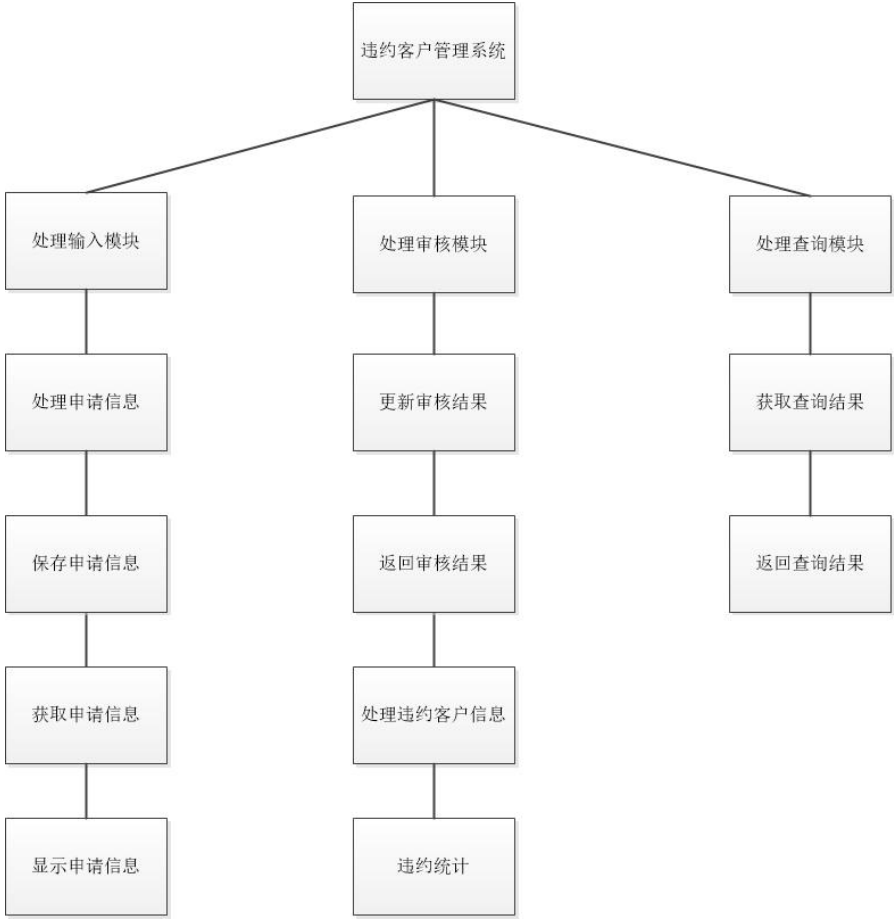
(1) 精华数据流图



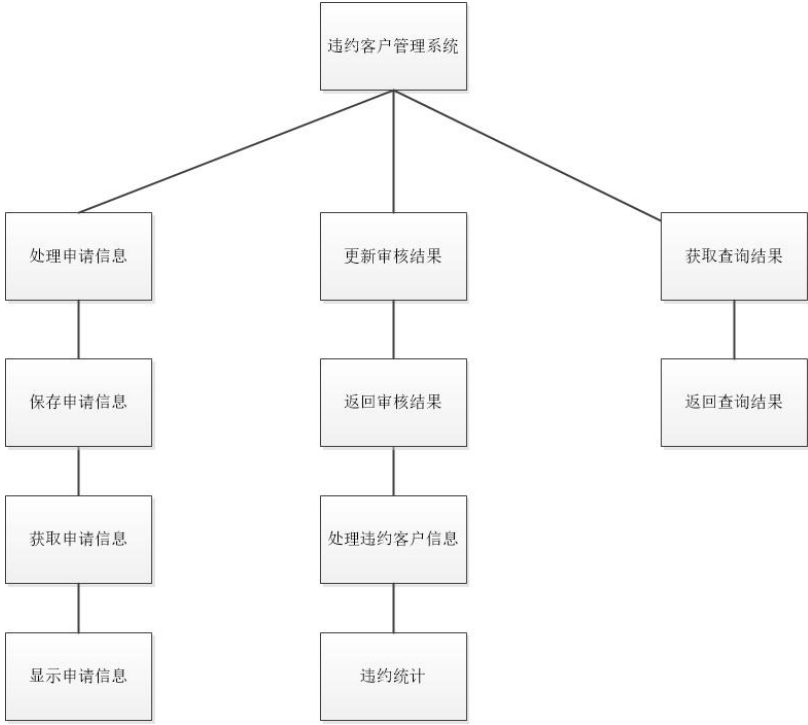
(2) 一级分解



(3) 二级分解



(4) 优化软件结构



### 3.2 数据库设计

(1) 数据表 1: 违约历史记录(历史数据表)

字段名	类型	取值范围	关键字	备注
地区	字符型	长度 0--50	否	
行业	字符型	长度 0--50	否	
时间	字符型	长度 0--50	否	

(2) 数据表 2: CustomerInfo(客户表)

字段名	类型	取值范围	关键字	备注
客户名	字符型	长度 0--50	是	
行业	字符型	长度 0--50	否	
地区	字符型	长度 0--50	否	

(3) 数据表 3: rdsh(认定申请表)

字段名	类型	取值范围	关键字	备注
客户名称	字符型	长度 0--50	是	
最新外部等级	字符型	长度 0--50	否	
违约原因	字符型	长度 0--50	否	
违约严重性	字符型	长度 0--50	否	
备注信息	字符型	长度 0--50	否	
审核状态	字符型	长度 0--50	否	
审核结果	字符型	长度 0--50	否	
附件地址	字符型	长度 0--50	否	
认定人	字符型	长度 0--50	否	

认定申请时间	Datetime		否	
--------	----------	--	---	--

(4) 数据表 4: Wycs(违约重生表)

字段名	类型	取值范围	关键字	备注
违约客户	字符型	长度 0--50	是	
审核状态	字符型	长度 0--50	否	
认定审核原因	字符型	长度 0--50	否	
严重程度	字符型	长度 0--50		
认定人	字符型	长度 0--50		
认定申请时间	Datetime	长度 0--50		
认定审核时间	Datetime	长度 0--50		
最新外部等级	字符型	长度 0--50		
审核结果	字符型	长度 0--50		
重生原因	字符型	长度 0--50		

(5) 数据表 5: Wyyy(违约原因表)

字段名	类型	取值范围	关键字	备注
违约原因序号	字符型	长度 0--50	是	
违约原因	字符型	长度 0--50	否	

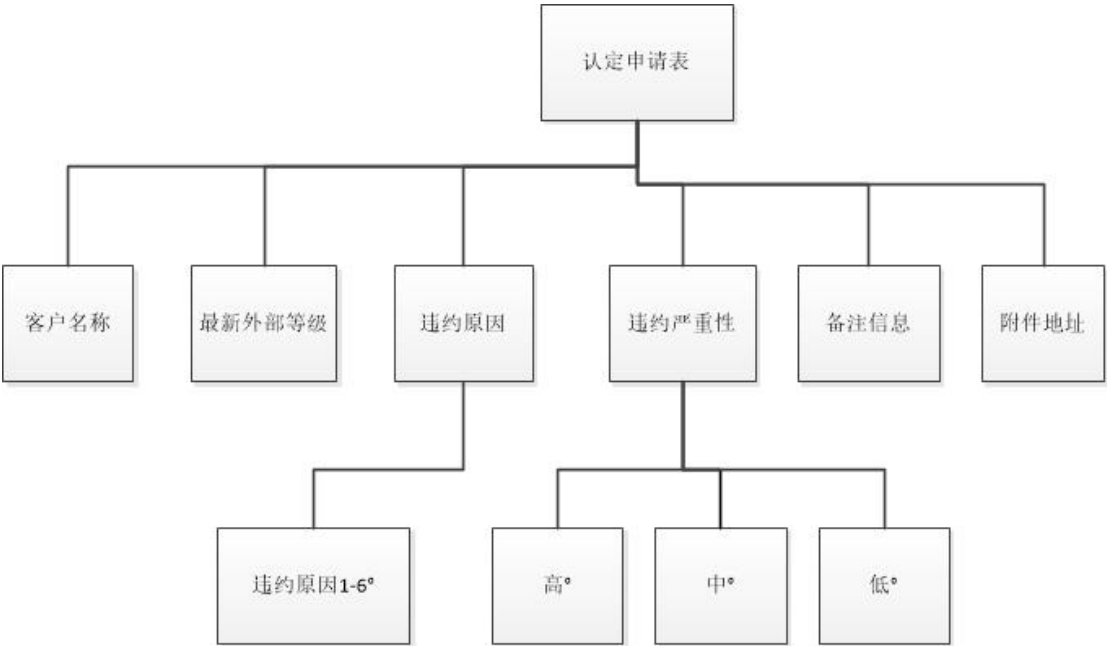


4.详细设计

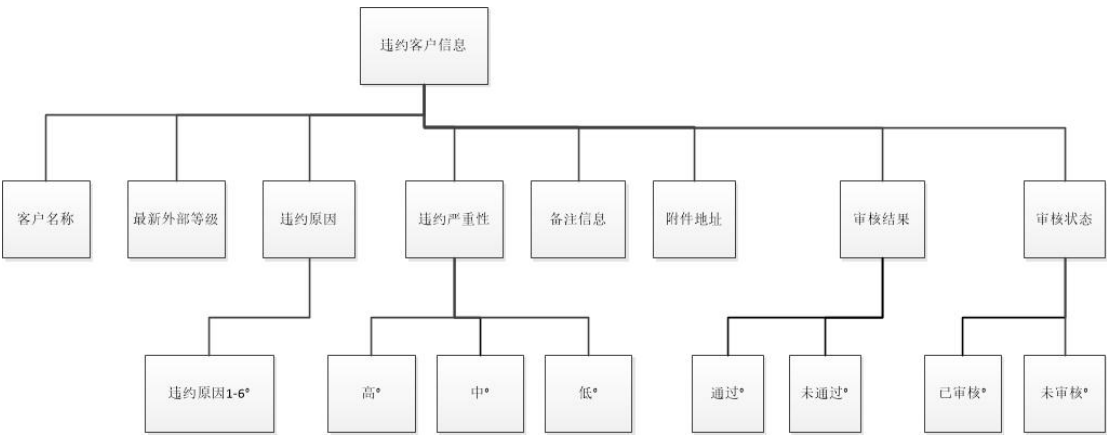
(1) 模块：违约认定审核，输入：认定申请表，输出：违约客户信息

算法设计：

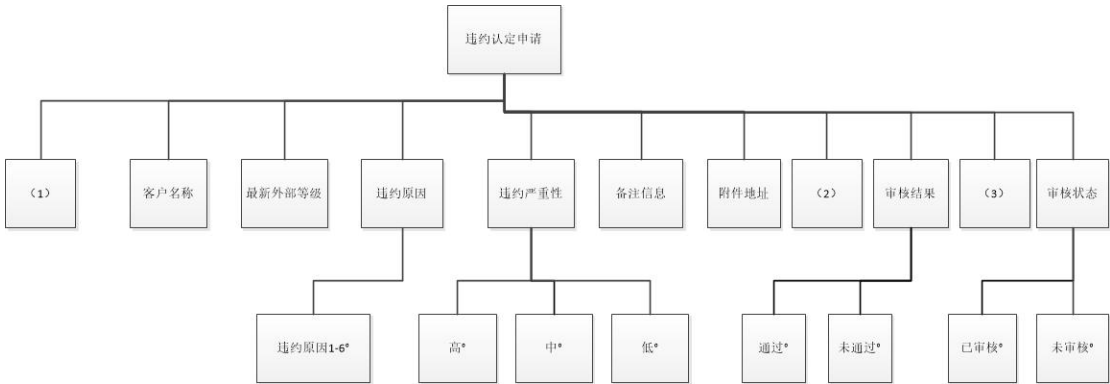
输入数据结构：



输出数据结构：



程序结构：



伪码描述：

Begin

    导入认定申请信息(1);

    输入客户名称;

    输入最新外部等级;

    If 违约原因=="违约原因 1" Then

        输入 6 个月内，交易对手技术性或资金等原因，给当天结算带来头寸缺口 2 次以上;

    Elseif 违约原因=="违约原因 2" Then

        输入 6 个月内因各种原因导致成交后撤单 2 次以上;

    Elseif 违约原因=="违约原因 3" Then

        输入未能按照合约规定支付或延期支付利息，本金或其他交付义务（不包括在宽限期内延期支付）；

    Elseif 违约原因=="违约原因 4" Then

        输入关联违约：如果集团（内部联系较紧密的集团）或集团内任一公司

(较重要的子公司，一旦发生违约会对整个集团造成较大影响的) 发生违约，可  
视情况作为集团内所有成员违约的触发条件;

Elseif 违约原因=="违约原因 5" Then

输入发生消极债务置换：债务人提供给债权人新的或重组的债务，或新的  
证券组合、现金或资产低于原有金融义务；或为了债务人未来避免发生破产或  
拖欠还款而进行的展期或重组;

Elseif 违约原因=="违约原因 6" Then

输入申请破产保护，发生法律接管，或者处于类似的破产保护状态;

EndIf

If 违约严重性=="低" Then

输入低;

Elseif 违约严重性=="中" Then

输入中;

Else

输入高;

输入备注信息;

输入附件地址;

审核是否通过 (2) ;

If 审核结果=="通过" Then

输入通过;

Else

输入未通过;

检查是否已审核（3）；

If 审核状态=="已审核" Then

    输入已审核;

Else

    输入未审核;

界面设计：

一级选项1

二级选项1

二级选项2

二级选项3

一级选项2

一级选项3

认定申请

文本

请输入文本

文本

文本

下拉菜单

文本

下拉菜单

文本

请输入文本

文本

选择文件

未选择任何文件

按钮

一级选项1

二级选项1

二级选项2

二级选项3

一级选项2

一级选项3

认定审核

文本

请输入文本

文本

下拉菜单

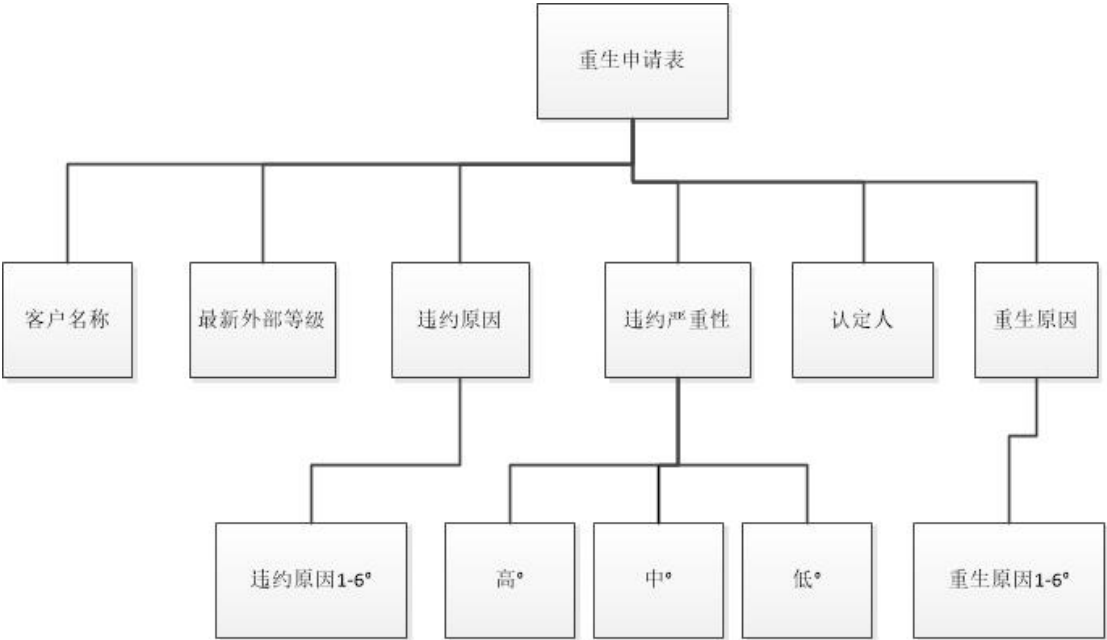
按钮

查询结果

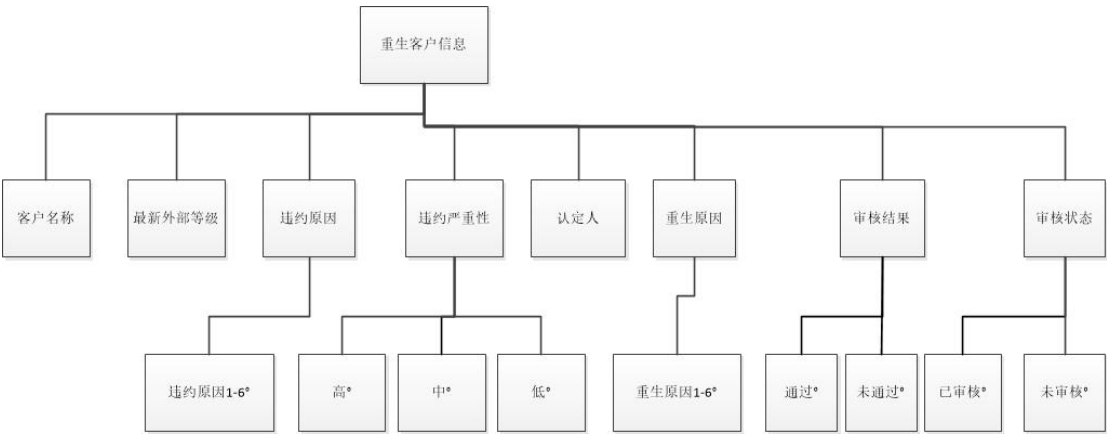
(2) 模块：违约重生审核，输入：重生申请表，输出：重生客户信息

算法设计：

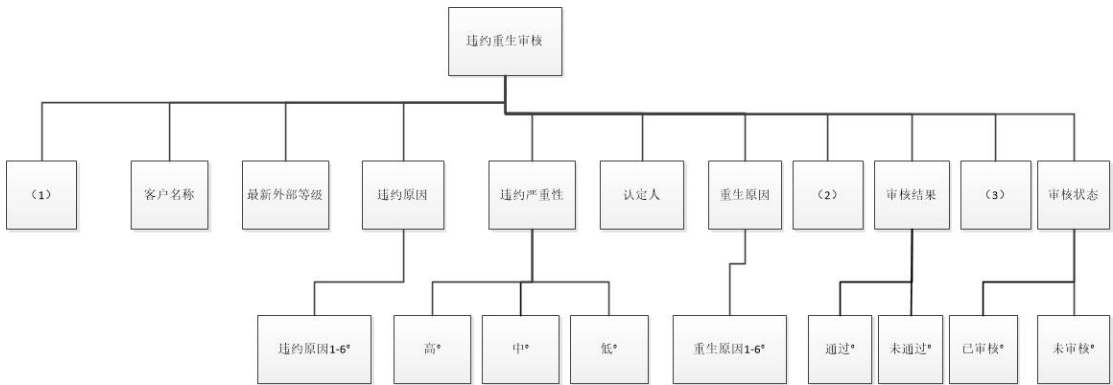
输入数据结构：



输出数据结构：



程序结构：



伪码描述：

Begin

    导入重生申请信息(1);

    输入客户名称;

    输入最新外部等级;

    If 违约原因=="违约原因 1" Then

        输入 6 个月内，交易对手技术性或资金等原因，给当天结算带来头寸缺口 2 次以上;

    Elseif 违约原因=="违约原因 2" Then

        输入 6 个月内因各种原因导致成交后撤单 2 次以上;

    Elseif 违约原因=="违约原因 3" Then

        输入未能按照合约规定支付或延期支付利息，本金或其他交付义务（不包括在宽限期内延期支付）；

    Elseif 违约原因=="违约原因 4" Then

        输入关联违约：如果集团（内部联系较紧密的集团）或集团内任一公司（较重要的子公司，一旦发生违约会对整个集团造成较大影响的）发生违约，可

视情况作为集团内所有成员违约的触发条件;

Elseif 违约原因=="违约原因 5" Then

输入发生消极债务置换: 债务人提供给债权人新的或重组的债务, 或新的证券组合、现金或资产低于原有金融义务; 或为了债务人未来避免发生破产或拖欠还款而进行的展期或重组;

Elseif 违约原因=="违约原因 6" Then

输入申请破产保护, 发生法律接管, 或者处于类似的破产保护状态;

EndIf

If 违约严重性=="低" Then

输入低;

Elseif 违约严重性=="中" Then

输入中;

Else

输入高;

EndIf

输入认定人;

If 重生原因=="重生原因 1" Then

输入正常结算后解除;

Elseif 重生原因=="重生原因 2" Then

输入在其他金融机构违约解除, 或外部评级显示为非违约级别;

Elseif 重生原因=="重生原因 3" Then

输入计提比例小于设置界限;

Elseif 重生原因=="重生原因 4" Then

输入连续 12 个月内按时支付本金和利息;

Elseif 重生原因=="重生原因 5" Then

输入客户的还款意愿和能力明显好转, 已偿付逾期本金和利息等费用,  
且连续 12 个月内按时支付本金、利息;

Elseif 重生原因=="重生原因 6" Then

输入导致违约的关联集团内其他违约客户已经违约重生, 解除关联成员  
的违约设定;

EndIf

审核是否通过 (2) ;

If 审核结果=="通过" Then

输入通过;

Else

输入未通过;

检查是否已审核 (3) ;

If 审核状态=="已审核" Then

输入已审核;

Else

输入未审核;



界面设计：

页面 1

一级选项1

二级选项1

二级选项2

二级选项3

一级选项2

一级选项3

违约重生

文本

请输入文本

文本

文本

下拉菜单

文本

下拉菜单

文本

请输入文本

文本

选择文件

 未选择任何文件

按钮

页面 1

一级选项1

二级选项1

二级选项2

二级选项3

一级选项2

一级选项3

违约重生审核

文本

请输入文本

文本

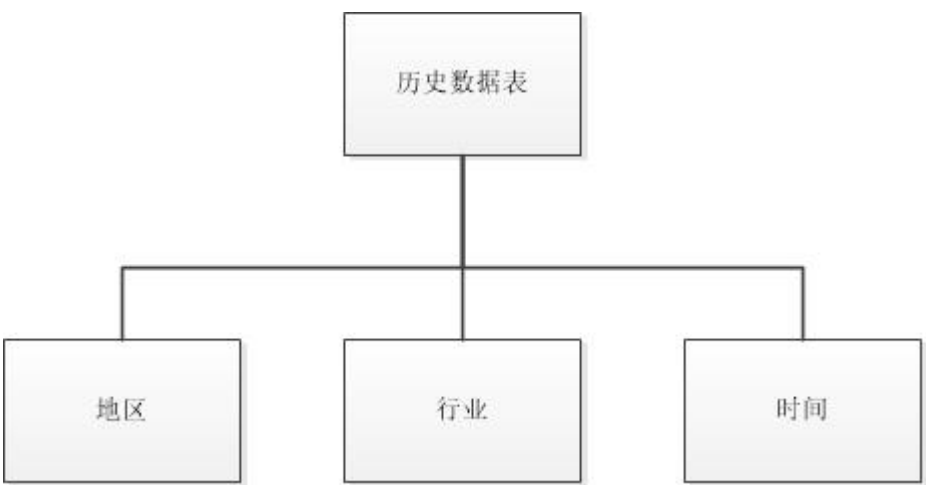
下拉菜单

按钮

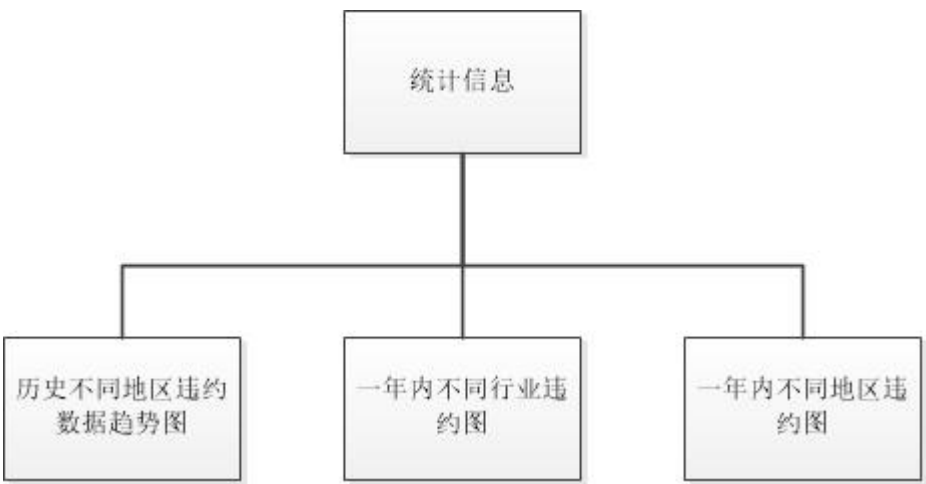
请输入文本

(3) 模块：统计信息 输入：历史数据表 输出：统计信息

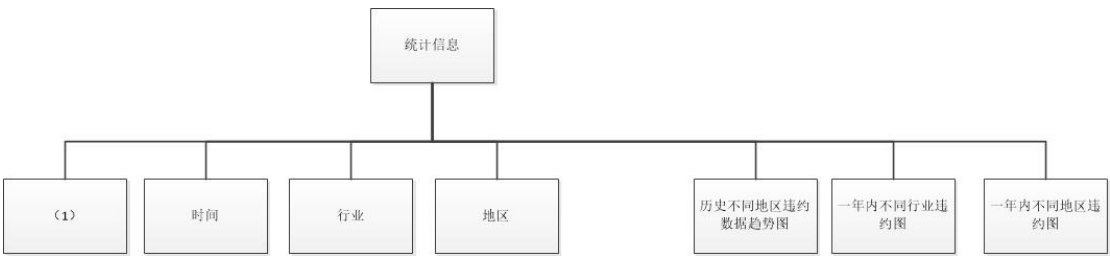
输入数据结构：



输出数据结构：



程序结构：



5.软件实现

5.1 系统运行界面

“违约认定审核”实现界面：



导航栏

查询客户信息

违约客户

违约认定申请

认定审核

违约重生申请

重生审核

数据统计

退出登录



认定申请

客户名称 \*

最新外部等级

0

违约原因 \*

请选择

违约严重性

高

备注信息

附件

选择文件 未选择文件

提交申请



导航栏

查询客户信息

违约客户

违约认定申请

认定审核

违约重生申请

重生审核

数据统计

退出登录



认定审核

客户名称

审核状态

全部

查询

客户名称	最新外部等级	违约原因	违约严重性	备注信息	审核状态	审核结果	操作
bbb	0	6	高	3	未审核		<div>通过</div> <div>不通过</div> <div>撤回</div>
2	0	4	高	3	已审核	通过	<div>通过</div> <div>不通过</div> <div>撤回</div>
ccc	0	4	高	23	已审核	不通过	<div>通过</div> <div>不通过</div> <div>撤回</div>
e	1	3	中	22	未审核		<div>通过</div> <div>不通过</div> <div>撤回</div>
f	0	2	高		已审核	不通过	<div>通过</div> <div>不通过</div> <div>撤回</div>

“违约重生审核”实现界面：



导航栏

查询客户信息

违约客户

违约认定申请

认定审核

违约重生申请

重生审核

数据统计

退出登录



违约重生

违约客户

认定违约原因

6 个月内，交易对手技术性或资金等原因，给当天结算带来头寸缺口 2 次以上

严重程度

高

认定人

最新外部等级

0

重生原因

请选择重生原因

提交重生申请



导航栏

查询客户信息

违约客户

违约认定申请

认定审核

违约重生申请

重生审核

数据统计

退出登录



重生审核

客户名称

审核状态

全部

查询

客户名称	最新外部等级	重生原因	审核状态	审核结果	操作
2	null	null	未审核		<div>通过</div> <div>不通过</div> <div>撤回</div>
33	0	在其他金融机构违约解除	已审核	通过	<div>通过</div> <div>不通过</div> <div>撤回</div>
1	0	还款意愿好转	已审核	不通过	<div>通过</div> <div>不通过</div> <div>撤回</div>
55	1	在其他金融机构违约解除	未审核		<div>通过</div> <div>不通过</div> <div>撤回</div>

统计信息实现界面：



## 5.2 单元测试

### (1) “认定审核”模块算法过程测试

语句覆盖测试用例 1：客户名称=NULL；审核状态=全部

语句覆盖测试用例 2：客户名称=NULL；审核状态=已审核

语句覆盖测试用例 3：客户名称=NULL；审核状态=未审核

语句覆盖测试用例 4：客户名称=张飞；审核状态=全部

语句覆盖测试用例 5：客户名称=张飞；审核状态=已审核

语句覆盖测试用例 6：客户名称=张飞；审核状态=未审核

实际测试结果：

语句覆盖测试用例 1 测试结果：全部记录

语句覆盖测试用例 2 测试结果：已审核记录

语句覆盖测试用例 3 测试结果：未审核记录

语句覆盖测试用例 4 测试结果：客户名为张飞的记录

语句覆盖测试用例 5 测试结果：客户名为张飞的已审核记录

语句覆盖测试用例 6 测试结果：客户名为张飞的未审核记录

### 5.3 集成测试

#### (1) “违约认定申请”模块算法过程测试

测试用例设计：

有效类测试用例 1：客户名称=张飞；最新外部等级=1；违约原因==

[违约原因 1]，违约严重性=低 备注信息=NULL 附件=NULL

无效类测试用例 2：客户名称=NULL；最新外部等级=1；违约原因==

[违约原因 1]，违约严重性=低 备注信息=NULL 附件=NULL

无效类测试用例 3：客户名称=张飞；最新外部等级=1；违约原因=

NULL，违约严重性=低 备注信息=1 附件=1.txt

实际测试结果：

语句覆盖测试用例 1 测试结果：提交成功

语句覆盖测试用例 2 测试结果：提交失败

语句覆盖测试用例 3 测试结果：提交失败

## 6.课程思政报告

### (1) 22050233 郑方昊课程思政报告

在这学期的计算机系统能力大赛中，我有幸基于 OpenHarmony 完成了一个应用的开发，整个过程让我受益匪浅。我不仅遇到了各种技术上的挑战，还学到了许多新知识，尤其是在操作系统、开源软件以及开发流程方面的深刻体会。接下来，我将从 OpenHarmony 的介绍、开发过程中的心得、遇到的困难与解决方法等方面来谈谈我在此次项目中的感悟。

#### 一、OpenHarmony 的简介与背景

OpenHarmony，作为一个开源项目，由开放原子开源基金会（OpenAtom Foundation）孵化及运营，旨在面向全场景、全连接、全智能的时代，提供一个跨设备、跨场景的操作系统平台。它的目标是促进智能终端设备的生态建设，推动万物互联产业的发展。OpenHarmony 作为一个开源系统，不仅有着强大的技术背景，还凝聚了全球开发者的智慧，使得它能够快速发展并适应各种设备的需求。

与鸿蒙 OS（HarmonyOS）相比，OpenHarmony 是鸿蒙 OS 的开源版，鸿蒙 OS 2 是华为基于 OpenHarmony 2.0 版本开发的面向商用市场的系统。鸿蒙 OS 作为华为的核心操作系统，除了具备传统操作系统的功能外，还具有更加深入的多设备互联能力。OpenHarmony 作为开源版本，它能够让开发者参与到系统的开发与优化中去，也能帮助其他厂商与开发者将其用于智能设备的开发中。通过 OpenHarmony 的开源，我们能够更自由地定制、修改操作系统，为更多智能设备提供支持。

#### 二、OpenHarmony 的技术优势

OpenHarmony 具备诸多技术优势，特别是在全场景智能终端和跨设备协同方面。作为一个轻量级、模块化的操作系统，它的架构可以根据设备的不同需求进行灵活的调整与裁剪。因此，OpenHarmony 能够支持从智能手机、智能家居设备到可穿戴设备等多种类型的终端。此外，OpenHarmony 还采用了分布式技术，这使得不同设备之间能够实现无缝的协同与交互，极大提升了用户的体验感。

### 三、基于 OpenHarmony 的开发过程

在这次基于 OpenHarmony 的应用开发过程中，我首先面临的挑战就是如何选择合适的开发工具和框架。由于 OpenHarmony 支持多种编程语言与开发框架，我最终选择了 ArkTS（OpenHarmony 生态的应用开发语言）与 ArkUI（OpenHarmony 的原生 UI 框架）进行开发。ArkTS 具备强大的性能和扩展性，同时还支持面向对象的编程方式，极大提高了开发效率。ArkUI 则提供了声明式的 UI 编程方式，使得界面的开发更加灵活和易于管理。

### 四、遇到的困难与解决方法

开发环境的配置问题：在最初的阶段，我遇到了一些关于开发环境配置的问题。特别是在理解 OpenHarmony 与鸿蒙 OS 之间的差异时，我曾产生过一些困惑。比如，华为的 DevEco Studio 并不直接支持 OpenHarmony 开发，我需要手动修改相关的配置文件才能使其支持 OpenHarmony 的开发。虽然这项工作花费了不少时间，但通过查阅相关资料并进行多次调试，最终顺利解决了问题。

功能实现的难点：在开发过程中，我实现了多个功能模块，如用户登录、菜谱查找、购物清单生成等，其中最具挑战性的模块是营养成分分析。由于营养成分的算法较为复杂，需要对用户的个人信息（如体重、性别、健康目标等）进行个性化分析，这就要求系统能够实时处理大量的数据并提供准确的建议。这个模



块的开发过程涉及了大量的数据处理与算法设计，我通过查阅文献和参考已有的开源项目，最终实现了该功能。

## 五、从开发中学到的新知识

通过此次基于 OpenHarmony 的开发，我学到了很多新的技术和知识。首先，我对操作系统的内部机制有了更深入的了解，尤其是对于 OpenHarmony 这样一个开源项目，它的灵活性和定制性让我意识到，开发者不仅可以使用它提供的工具和框架，还能够参与到系统的优化和改进中，推动技术的创新与进步。

其次，我在开发过程中进一步掌握了前端开发技能，特别是在使用 ArkUI 进行界面设计时，我体会到了声明式 UI 编程的优势，它使得 UI 的开发和管理变得更加高效。同时，通过处理各种复杂的用户输入和数据交互，我也加强了自己的后端开发能力，尤其是在数据库设计和算法实现方面。

## 六、总结

通过参与计算机系统能力大赛，基于 OpenHarmony 的应用开发让我不仅学到了新的技术，还对开源精神、团队合作和技术挑战有了更深刻的认识。尽管过程中有很多困难和挑战，但通过不断地尝试和学习，我最终成功地完成了这个项目。

### (2) 22050230 吴博涵课程思政报告

在本学期的软件工程实践课程中，我通过实际参与软件开发项目，深入理解了软件工程的理论知识和实践操作。课程内容不仅帮助我掌握了软件开发的核心技能，也让我更加意识到团队合作、需求分析、项目管理和软件质量等方面的重要性。

软件工程课程是计算机专业的一门核心课程，内容丰富且系统。通过课程的学习，我深入理解了软件开发过程中各个环节的关键知识点。例如，需求分析帮助我学会了如何从用户需求出发，理清问题的本质；架构设计和模式使我更加清晰地理解了系统的结构与组织，避免了盲目开发带来的不合理设计；测试与维护则让我认识到，软件开发不仅仅是写代码，还需要考虑如何保证软件质量和持续改进。软件工程实践不仅是一门实践课程，更是一次全面的职业素养培养。它让我真正理解了软件开发的各个环节，也让我体会到团队合作、需求分析、系统设计和质量保证等方面的挑战和乐趣。

在未来的学习和工作中，我将继续秉持在实践课程中学到的知识和经验，不断提升自己的技术能力，培养团队协作和项目管理的能力，力争成为一名合格的软件工程师，为社会的发展贡献自己的力量。