**“银行业务管理系统”**

系统设计与实现报告

**姓名：龚小航**

**学号：PB18151866**

**计算机科学与技术学院**

**中国科学技术大学**

2021年6月

**目 录**

1 概述 1

1.1 系统目标 1

1.2 需求说明 1

1.3 本报告的主要贡献 1

2 总体设计 2

2.1 系统模块结构 2

2.2 系统工作流程 2

2.3 数据库设计 3

3 详细设计 4

3.1 Main 模块 4

3.2 Welcome与Login模块 4

3.3 Function模块 5

4 实现与测试 7

4.1 实现结果 7

4.2 测试结果 13

4.3 实现中的难点问题及解决 15

5 总结与讨论 15

# 

# 1 概述

## 系统目标

为银行开发一套业务管理系统。

本次实验采用python 3.9与QtDesigner完成前端设计与开发，基于客户/服务器结构。实现中主要调用图形包PyQt5完成前端消息的获取与传递。后台DBMS采用MySQL

## 1.2 需求说明

逻辑关系与开发需求：

银行有多个支行。各个支行位于某个城市，每个支行有唯一的名字。银行要监控每个支行的资产。

银行的客户通过其身份证号来标识。银行存储每个客户的姓名、联系电话以及家庭住址。为了安全起见，银行还要求客户提供一位联系人的信息，包括联系人姓名、手机号、Email 以及与客户的关系。客户可以有帐户，并且可以贷款。客户可能和某个银行员工发生联系，该员工是此客户的贷款负责人或银行帐户负责人。银行员工也通过身份证号来标识。员工分为部门经理和普通员工，每个部门经理都负责领导其所在部门的员工，并且每个员工只允许在一个部门内工作。每个支行的管理机构存储每个员工的姓名、电话号码、家庭地址、所在的部门号、部门名称、部门类型及部门经理的身份证号。银行还需知道每个员工开始工作的日期，由此日期可以推知员工的雇佣期。

银行提供两类帐户——储蓄帐户和支票帐户。帐户可以由多个客户所共有，一个客户也可开设多个账户，但在一个支行内最多只能开设一个储蓄账户和一个支票账户。每个帐户被赋以唯一的帐户号。银行记录每个帐户的余额、开户日期、开户的支行名以及每个帐户所有者访问该帐户的最近日期。另外，每个储蓄帐户有利率和货币类型，且每个支票帐户有透支额。

每笔贷款由某个分支机构发放，能被一个或多个客户所共有。每笔贷款用唯一的贷款号标识。银行需要知道每笔贷款所贷金额以及逐次支付的情况（银行将贷款分几次付给客户）。虽然贷款号不能唯一标识银行所有为贷款所付的款项，但可以唯一标识为某贷款所付的款项。对每次的付款需要记录日期和金额。

本次实验需要开发的系统具体有以下的要求：

• 客户管理：提供客户所有信息的增、删、改、查功能；如果客户存在着关联账户或者贷款记录，则不允许删除；

• 账户管理：提供账户开户、销户、修改、查询功能，包括储蓄账户和支票账户；账户号不允许修改；

• 贷款管理：提供贷款信息的增、删、查功能，提供贷款发放功能；贷款信息一旦添加成功后不允许修改；要求能查询每笔贷款的当前状态（未开始发放、发放中、已全部发放）；处于发放中状态的贷款记录不允许删除；

• 业务统计：按业务分类（储蓄、贷款）和时间（月、季、年）统计各个支行的业务总金额和用户数，统计的结果以表格形式展示。

## 1.3 本报告的主要贡献

本报告阐述了银行管理系统的具体需求，整个系统的框架设计，模块之间的组织方式，各模块的具体实现以及该系统的完整性与正确性验证。

# 2 总体设计

## 2.1 系统模块结构

整个系统的组成结构如下所示：

图形用户界面, 应用程序

描述已自动生成

## 2.2 系统工作流程

系统从main模块开始执行，以下为main模块的实现：

图形用户界面, 文本, 应用程序

描述已自动生成

从总体上看，程序始终在运行两个Window。 Main模块中例化了QMainWindow类型的实例mainform以及QWidget类型的实例funform，分别对应主界面窗口以及功能界面窗口。在main模块中只需要管理它们之间的切换逻辑以及数据消息的传递。即在主窗口获取到了从对话窗口Login传回的有效db连接时将当前窗口切换为能对数据库操作的功能窗口，并将db数据传递下去；而在功能窗口中点击退出登录时需要切换回主窗口，此时断开db连接并实现窗口切换，系统回到初始状态。

各模块的详细工作流程在“详细设计”中阐述。

## 2.3 数据库设计

ER图：

图示, 示意图

描述已自动生成

CDM设计图：

图示

描述已自动生成

SQL数据库具体结构：（外键等定义由PowerDesigner自动生成Alter添加）

表格

描述已自动生成文本

描述已自动生成文本

描述已自动生成

图片包含 图形用户界面

描述已自动生成文本

描述已自动生成表格

中度可信度描述已自动生成文本

描述已自动生成图片包含 表格

描述已自动生成图片包含 图形用户界面

描述已自动生成表格

中度可信度描述已自动生成文本

中度可信度描述已自动生成

文本

中度可信度描述已自动生成文本

描述已自动生成

图形用户界面, 应用程序

中度可信度描述已自动生成其中部门-员工类信息不要求维护与实现，这些信息已经预插入表中。

发款表增加了一个自增字段，每次插入新纪录时+1保证主键不重复。

# 3 详细设计

## 3.1 main模块

具体实现已经在上方给出。Main例化qt主窗口类以及窗口类，实现程序主循环并通过信号槽触发实现窗口切换。

## 3.2 Welcome以及Login模块

WelcomeView是程序的主窗口。运行界面如下所示：

图表, 表面图

描述已自动生成

通过上方菜单栏选项可以登录系统。选择Login后弹出登录对话框：

图形用户界面

描述已自动生成

输入正确的连接属性后点击Login，主窗口从dialog得到数据库的连接，触发窗口切换事件，进入程序功能窗口Function；若输入错误，则会弹出“登录失败”的提示框。

主窗口的实现如下所示：

**from** PyQt5**.**QtWidgets **import** QMainWindow

**from** QtDesignerFiles **import** WelcomeView

**from** LoginFile **import** LoginDialog

**from** PyQt5 **import** QtCore

**class** **WelcomeWindow(**QMainWindow**):**

show\_funcform\_signal **=** QtCore**.**pyqtSignal**()**

" The Entrance of the Main window"

**def** \_\_init\_\_**(**self**):**

#初始化入口，以下代码在例化时自动执行

**super().**\_\_init\_\_**()**

self**.**dialog **=** **None**

self**.**db **=** **None**

self**.**dbname **=** ''

#WelcomeView的UI实例

self**.**ui **=** WelcomeView**.**Ui\_WelcomeView**()**

#调用Ui\_WelcomeView类中的方法,向主窗口中添加控件

self**.**ui**.**setupUi**(**self**)**

self**.**initBinding**()**

**def** initBinding**(**self**):** #将welcome页面上的动作绑定到实现函数上，仅含登录操作。

self**.**ui**.**actionLogin**.**triggered**.**connect**(**self**.**Login**)**

**def** Login**(**self**):**

**print(**"login"**)**

self**.**dialog **=** LoginDialog**(**self**)**

self**.**dialog**.**exec\_**()**

**if** self**.**db **!=** **None:**

self**.**jumpToFunform**()**

**def** jumpToFunform**(**self**):**

self**.**show\_funcform\_signal**.**emit**()**

## 3.3 Function模块

Function模块的输入是从主窗口传入的db连接信息，而窗口激活信号是登录成功后发出的槽触发信号。对整个程序来说Function模块的输出是“退出登录”信号，点击退出以后激发槽信号触发器，使窗口发生切换并将主窗口、本窗口的db连接置空。

Function界面的UI设计如下所示：

图形用户界面, 应用程序

描述已自动生成

从顶层向下层说明：总体frame与form之间是垂直布局；frame与其内容之间是垂直布局；最上方的三个功能钮与下方的整体QTabWidget是垂直布局；左侧与右侧的子QTabWidget是水平布局；左侧的功能区为栅格布局，右侧子QTabWidget内部是垂直布局。每一行是水平布局。

窗口的伸缩属性为Preferred，默认最佳大小为1920×1080；窗口内控件布局分配了占位符，伸缩策略也调整为伸缩占位符从而保证在放大窗口时有效控件不变型。

这样的UI设计保证了整体的可伸缩性与不变型的特点，同时界面按钮与输入都考虑到了实际需求。

图形用户界面, 应用程序

描述已自动生成

通过上方的标签可选择不同的业务类型，每一类左侧有对应的操作类型，右侧上方为输入框，下方为显示结果的表格。

一般的输入框LineEdit均做了正则表达式检验，不符合类型的数据无法键入输入框。

图形用户界面, 文本, 应用程序

描述已自动生成

Function模块的缩略具体实现如下：

图形用户界面, 文本, 应用程序

描述已自动生成

后面均为各功能的函数实现。通过将按钮点击事件绑定到函数调用各种功能。

本次实验python端的作用主要是前端与后端DBMS之间的消息协调与传递。

代码实现以删除一条客户信息为例：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

只需要对必要的字段预先处理，再用游标执行目标SQL语句输出结果即可。本次实验所有SQL都预先在WorkBench中运行验证，直接得到期望输出的结果。因此在python程序中只需要将这些sql语句执行一遍即可得到需要输出的数据。

其他功能模块类似，具体实现见附件源码。

业务管理的SQL实现：

#按月统计发**款**

**select** **支**行名**,DATE\_FORMAT(**发**款**日期**,**'%Y-%m'**)** 月份**,count(**发**款**日期**)** 发**款次**数 **,SUM(**发**款**金额**)** 发**款总**额

**from** **(select** **支**行名**,**贷**款**号**,**发**款**日期**,**发**款**金额 **from** **(支**行 **left** **join** 贷**款** **Using(支**行名**))** **left** **join** 发**款** **Using(**贷**款**号**))** C1

**group** **by** **支**行名**,**月份**;**

#按季度统计发**款**

**select** **支**行名**,** **DATE\_FORMAT(**发**款**日期**,**'%Y'**)**年度**,** QUARTER**(**发**款**日期**)**季度**,** **count(**发**款**日期**)** 发**款次**数 **,SUM(**发**款**金额**)** 发**款总**额

**from** **(select** **支**行名**,**贷**款**号**,**发**款**日期**,**发**款**金额 **from** **(支**行 **left** **join** 贷**款** **Using(支**行名**))** **left** **join** 发**款** **Using(**贷**款**号**))** C1

**group** **by** **支**行名**,**年度**,**季度**;**

#按年统计发**款**

**select** **支**行名**,DATE\_FORMAT(**发**款**日期**,**'%Y'**)**年度**,count(**发**款**日期**)** 发**款次**数 **,SUM(**发**款**金额**)** 发**款总**额

**from** **(select** **支**行名**,**贷**款**号**,**发**款**日期**,**发**款**金额 **from** **(支**行 **left** **join** 贷**款** **Using(支**行名**))** **left** **join** 发**款** **Using(**贷**款**号**))** C1

**group** **by** **支**行名**,**年度**;**

#按月统计银行储蓄**总**额以及客户人数

**select** S1**.支**行名**,** S1**.**月份**,** 新增客户人数**,** 存储**总**额

**from**

**(select** **支**行名**,** **DATE\_FORMAT(**开户日期**,**'%Y-%m'**)** 月份**,** **SUM(账**户余额**)** 存储**总**额

**from** **((select** **distinct** **支**行名**,账**户号**,账**户余额**,**开户日期

**from** **支**行 **left** **join** **((**储蓄**账**户 **left** **join** 储蓄**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**))**

**Union**

**(select** **distinct** **支**行名**,账**户号**,账**户余额**,**开户日期

**from** **支**行 **left** **join** **((支**票**账**户 **left** **join** **支**票**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**)))** C1

**group** **by** **支**行名**,**月份**)** S1**,**

**(**

**select** **支**行名**,DATE\_FORMAT(**开户日期**,**'%Y-%m'**)** 月份**,** **count(distinct** 客户身份证号**)** 新增客户人数

**from** **((select** **支**行名**,账**户号**,账**户余额**,**开户日期**,**客户身份证号

**from** **支**行 **left** **join** **((**储蓄**账**户 **left** **join** 储蓄**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**))**

**Union**

**(select** **支**行名**,账**户号**,账**户余额**,**开户日期**,**客户身份证号

**from** **支**行 **left** **join** **((支**票**账**户 **left** **join** **支**票**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**)))** C1

**group** **by** **支**行名**,**月份**)** S2

**where** S1**.支**行名 **=** S2**.支**行名 **and** S1**.**月份 **=** S2**.**月份**;**

#按季度统计银行储蓄**总**额以及客户人数

**select** S1**.支**行名**,** S1**.**年度**,** S1**.**季度**,** 新增客户人数**,** 存储**总**额

**from**

**(select** **支**行名**,** **DATE\_FORMAT(**开户日期**,**'%Y'**)** 年度**,** QUARTER**(**开户日期**)**季度**,** **SUM(账**户余额**)** 存储**总**额

**from** **((select** **distinct** **支**行名**,账**户号**,账**户余额**,**开户日期

**from** **支**行 **left** **join** **((**储蓄**账**户 **left** **join** 储蓄**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**))**

**Union**

**(select** **distinct** **支**行名**,账**户号**,账**户余额**,**开户日期

**from** **支**行 **left** **join** **((支**票**账**户 **left** **join** **支**票**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**)))** C1

**group** **by** **支**行名**,**年度**,**季度**)** S1**,**

**(**

**select** **支**行名**,DATE\_FORMAT(**开户日期**,**'%Y'**)** 年度**,** QUARTER**(**开户日期**)**季度**,** **count(distinct** 客户身份证号**)** 新增客户人数

**from** **((select** **支**行名**,账**户号**,账**户余额**,**开户日期**,**客户身份证号

**from** **支**行 **left** **join** **((**储蓄**账**户 **left** **join** 储蓄**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**))**

**Union**

**(select** **支**行名**,账**户号**,账**户余额**,**开户日期**,**客户身份证号

**from** **支**行 **left** **join** **((支**票**账**户 **left** **join** **支**票**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**)))** C1

**group** **by** **支**行名**,**年度**,**季度**)** S2

**where** S1**.支**行名 **=** S2**.支**行名 **and** S1**.**年度 **=** S2**.**年度 **and** S1**.**季度 **=** S2**.**季度**;**

#按年统计银行储蓄**总**额以及客户人数

**select** S1**.支**行名**,** S1**.**年度**,** 新增客户人数**,** 存储**总**额

**from**

**(select** **支**行名**,** **DATE\_FORMAT(**开户日期**,**'%Y'**)** 年度**,** **SUM(账**户余额**)** 存储**总**额

**from** **((select** **distinct** **支**行名**,账**户号**,账**户余额**,**开户日期

**from** **支**行 **left** **join** **((**储蓄**账**户 **left** **join** 储蓄**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**))**

**Union**

**(select** **distinct** **支**行名**,账**户号**,账**户余额**,**开户日期

**from** **支**行 **left** **join** **((支**票**账**户 **left** **join** **支**票**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**)))** C1

**group** **by** **支**行名**,**年度**)** S1**,**

**(**

**select** **支**行名**,DATE\_FORMAT(**开户日期**,**'%Y'**)** 年度**,** **count(distinct** 客户身份证号**)** 新增客户人数

**from** **((select** **支**行名**,账**户号**,账**户余额**,**开户日期**,**客户身份证号

**from** **支**行 **left** **join** **((**储蓄**账**户 **left** **join** 储蓄**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**))**

**Union**

**(select** **支**行名**,账**户号**,账**户余额**,**开户日期**,**客户身份证号

**from** **支**行 **left** **join** **((支**票**账**户 **left** **join** **支**票**账**户访问时间 **Using(账**户号**)))** **Using(支**行名**)))** C1

**group** **by** **支**行名**,**年度**)** S2

**where** S1**.支**行名 **=** S2**.支**行名 **and** S1**.**年度 **=** S2**.**年度**;**

# 4 实现与测试

## 4.1 实现结果

1、客户管理业务：

客户所有信息展示：

表格

描述已自动生成

新增一个客户及其联系人：

图形用户界面, 应用程序, 表格

描述已自动生成

表格

描述已自动生成

模糊查询客户信息：

图形用户界面

描述已自动生成

修改客户身份证号（主键）：（在其他表内将级联更新）

图形用户界面, 表格

描述已自动生成

表格

描述已自动生成

修改同一ID的多个其他属性：

表格

描述已自动生成

表格

描述已自动生成

删除记录：

图形用户界面, 应用程序, 表格

描述已自动生成

表格

描述已自动生成

查询所有客户拥有的账户信息：

表格

描述已自动生成

查询所有客户的贷款信息：

表格

描述已自动生成

2、账户管理业务：（以储蓄账户为例，支票账户完全相同）

显示全部记录，允许多个用户共享一个账号：

表格

描述已自动生成

添加账户：

表格

描述已自动生成

表格

描述已自动生成

增加账户所有者：

表格

描述已自动生成

表格

描述已自动生成

更新账户号、所有者以外的信息：

表格

描述已自动生成

表格

描述已自动生成

删除某个账户所有者：

表格

描述已自动生成

表格

描述已自动生成

销户：

表格

描述已自动生成

表格

描述已自动生成

3、贷款管理业务：

显示所有贷款信息，允许多个客户负担一笔贷款：

表格

描述已自动生成

按输入模糊查询：

表格

描述已自动生成

添加新的贷款记录：

表格

中度可信度描述已自动生成

表格

描述已自动生成

为已有贷款添加新的贷款人：

图形用户界面, 表格

描述已自动生成

表格

描述已自动生成

发放一笔款项：

图形用户界面, 表格

描述已自动生成

表格

描述已自动生成

删除已经发放完成或未发放的贷款：（先将DK09发放完成）

表格

描述已自动生成

表格

描述已自动生成

表格

描述已自动生成

4、业务统计：

根据六种统计方式分别设计sql语句调用展示即可：

根据月份统计发款信息：

表格

描述已自动生成

按季度统计发款信息：

表格

描述已自动生成

按年统计发款信息：

表格

描述已自动生成

按月统计储蓄信息以及用户注册数：

表格

描述已自动生成

按季度统计储蓄以及用户注册数：

图形用户界面, 表格

描述已自动生成

按年统计储蓄信息以及用户注册数：

表格

描述已自动生成

## 4.2 测试结果

测试一些常见错误情况

1、客户管理业务：

插入时数据完整性不满足：(添加，修改，删除需要ID)

图形用户界面, 应用程序, 表格

描述已自动生成

插入客户名、联系人名带有特殊字符：

表格

描述已自动生成

2、账户管理业务：

同一个客户在一个银行最多拥有一个储蓄账户和一个支票账户：

将CX01的拥有者C2改为C6：

表格

描述已自动生成

3、贷款管理业务：

删除一笔正在发放中的贷款：

表格

描述已自动生成

发放一笔超过未发余额的发款：

图形用户界面, 表格

描述已自动生成

发放一笔比上一次发放时间更早的发款：

表格

描述已自动生成

## 4.3 实现中的难点问题及解决

·窗口切换：窗口的实例全部定义在main中，因而窗口切换需要在main中执行hide(), show()操作。而在其他模块import main会造成循环调用问题；而main中执行过前面的代码后会一直在sys.exit(app.exec\_())中循环执行，不会再主动执行外部的代码。因此采用信号槽机制，主窗口设置触发条件，在main中执行一段特定的代码实现窗口切换。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

# 5 总结与讨论

本次实验实现的是一个工程上的应用问题。在完成实验的过程中，我完整的体验到软件开发的各个部分需要处理哪些内容，负责哪些工作。本次实验让我对软件工程的开发标准也有了一定的理解，良好的开发习惯以及编程规范能让程序开发事半功倍。