银行存取系统

<开发计划书>

姓名：敬舒舒
学号：161220056
班级：2018-金融软件工程
邮箱：1151545191@qq.com

一、 项目背景

1.1 需求背景

随着信息技术的发展，办公自动化的普及，如何快速，高效，便捷的为银行用户服务、管理用户的账户受到了高度的关注。我们设计这个系统使用户更方便的管理个人账户，使银行对账目资金的管理更加高效便捷。

在传统的银行账户管理中，其过程往往是很复杂的，繁琐的，账户管理以入账和出账两项内容为核心，在此过程中又需要经过若干道手续，因为整个过程都需要手工操作，效率十分低下，且由于他们之间关联复杂，统计和查询的方式各不相同，且会出现信息的重复传递问题，因此该过程需要进行信息化，以利用计算机进行账目管理

1.2 项目背景

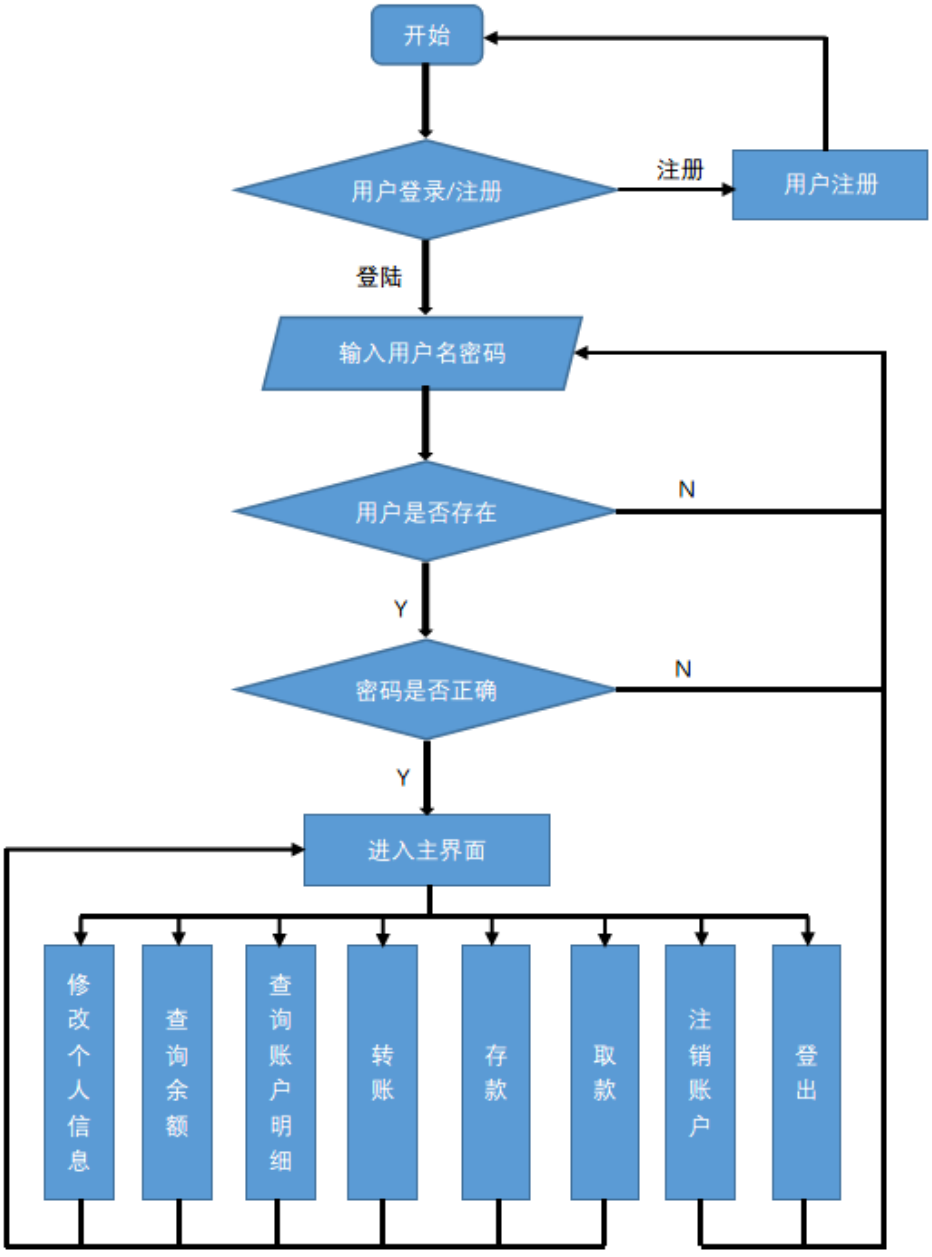
A．此项目为金融软件工程课程设计

B．目前市面上已有成熟的类似系统，将以此成熟的产品为参考

二、 项目目标

本银行系统面向个人用户设计，用户可自主注册/注销账号，进行登陆、登出、查询、存款、转账、取款、修改个人信息等操作。

用户功能最终完成的流程图如下:



除了用户账户外，还设立拥有系统管理权限的管理员账户用以对系统进行管理，功能包括：导出用户数据，修改数据，监视异常等

本系统的管理员账号为备选功能，视项目完成情况决定是否实现。

三、 概要设计

3.1 页面与功能

本项目的银行存取系统需要在网页上实现，包括 9 个页面：

1. 用户登陆/注册页面

   包含用户名、密码、验证码输入框

2. 注册页面

   用户信息填写，在数据库生成对应用户账户

3. 主界面

   功能分栏：个人信息、账户明细、转账、存款、取款

   功能按钮：注销、登出

4. 个人信息页面

   显示用户的个人信息，并具有修改功能

5. 余额与账户明细页面

   显示余额、用户历史操作、账户资金改变记录

6. 转账页面

   填写目标账户用户名及转账金额，输入密码以确定转账

7. 存款页面

   填写存款金额，增加账户余额

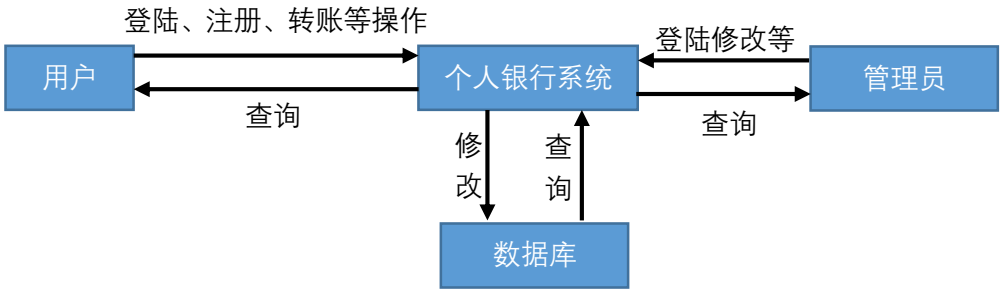
8. 取款页面

   填写取款金额，输入密码以确定取款，减少账户余额

3.2 数据与管理

编写数据库生成用户信息表以储存用户的账户信息与操作记录

数据流走向：



数据流权限：

用户只能访问自己的数据，不能访问/修改其他用户的数据。

管理员可以访问所有的用户数据，并查询修改所有的数据。

对所有的数据操作都必须先通过权限验证。

3.3 系统特性

稳定性：本系统为银行存取管理系统，多为查询、修改、删除、添加数据等操作，要求数据的稳定性好。

可靠性：个人财务信息十分重要，要求系统数据能够较好存储，数据尽量最少出现错误。

安全性：登录系统需要输入密码，可保证个人信息安全。若在一定时间内未操作则自行登出。

四、 详细设计说明书

模块实现设计

4.1 数据库模块设计

模块名称：db_operation

模块功能：对数据库的初始化、连接、读取、修改、关闭

模块调用方法：

```
db_init()                    //初始化数据库

@app.cli.command('initdb')
def initdb_command():        //通过命令行 flask initdb 来调用 init_db()

db_connect()     //连接数据库

db_get()         //获取数据库，为了减少反复数据库的开销。且本实验数据量
                 并不大，我们可以将数据存在 flask 提供全局变量 g 中

@app.teardown_appcontext
db_close()       //关闭数据库，应用关闭时会自动被调用，以避免数据的泄露

db_read_information()        //读取用户个人信息
db_read_login()              //读取用户登陆密码

db_modify_information ()     //修改用户个人信息，检验修改是否合法
db_modify_code()             //修改用户密码，检验修改是否合法
db_modify_balance()          //修改账户余额

db_record_operation()        //记录用户操作（转账、存款、取款）
db_record_information()      //记录用户注册时的个人信息
db_record_code()             //记录用户登陆密码

db_create_account()          //创建账号（调用 db_record_information 和
                             db_record_code）
db_ delete _code()           //删除用户登陆密码
db_delete_accout()           //注销账号，删除账号信息

db_check_accout()            //检查该账号是否存在（注册、转账时）
db_check_balance()           //检查账户余额
db_check_code()              //通过密码确认操作，所有函数中需要密码来确
```

认操作的函数都会调用此函数来（比如转账、取款、注销等）

4.2 登陆/登出模块设计

模块名称：login_&_logout

模块功能：用户登陆和登出

模块调用方法：

signin()　　//解析页面发回的表单，调用 db_check_code()验证登陆信息
若用户名密码错误，则返回错误信息，否则跳转至用户界面
signin_form()　　//显示登陆页面
signout()　　//清楚登陆状态，返回未登录页面
register()　　//处理注册信息，调用数据库操作 db_record_code 和 db_create_account 来创建账号，账号创建完成后返回登陆页面
register_form　　//显示注册页面

4.3 用户功能模块设计

模块名称：user_function

模块功能：处理与用户相关的的功能跳转，实现相关功能

模块调用方法：

user_select_function()　　//处理用户选择，跳转至相关页面

user_information()　　//用户个人信息页面，可选择修改
user_transfer_accout()　　//用户转账页面，检查对方账号 db_check_accout()
检查密码确认 db_check_code()
检查余额够否 db_check_balance()
修改自身余额 db_modify_balance()
修改对方余额 db_modify_balance()
记录双方操作 db_record_operation()
user_deposit()　　//用户存款
user_cash()　　//用户取款
user_ inquire_account()　　//用户查询余额
user_inquire_history()　　//用户查询历史操作

user_delete_account()　　//用户注销账号，检查账户余额 db_check_balance()
输入密码确认 db_check_code()
删除账户信息 db_ delete _code()
删除用户密码 db_delete_accou

```
user_form_main()                    //显示用户主界面
uesr_form_ transfer_accout ()       //显示转账页面
user_form_information()             //显示用户个人信息页面
user_form_ deposit()                //显示用户存款页面
user_form_cash()                    //显示用户取款页面
user_form_ inquire_account()        //显示用户查询界面
user_form_ inquire_ history ()      //显示用户历史操作界面
user_f
```

五、 开发方案

我们的系统主要由一个一个不同的功能组成，采用迭代-递增生命周期模型。第一步完成登陆页面和主界面，接下来依次完成每个功能对应的页面及其功能。在每一步保证系统的正常运行。

因为部分功能需要建立在一些特定功能的基础上，我们也需要将功能页面的开发划分为几个阶段，完成顺序为：登陆页面、主界面 -> 建设数据库 -> 余额与账户明细页面 –> 存款、取款、转账页面 -> 个人信息、注册页面

时间安排（代码编写）：

登陆页面、主界面 -------- 3 天

建设数据库 -------- 2 天

余额与账户明细页面 -------- 2 天

存款、取款、转账页面------- 2 天

个人信息、注册页面 -------- 2 天

开发语言：网页开发-----Javascript

数据库 ------SQL server2017

在代码编写的每个阶段都需进行测试保证已完成的功能没有问题，且在前期开发的过程中应充分考虑后续开发的要求。

由于此系统为独立开发，对系统开发的控制力较强，此递增迭代周期的可行性较高。但由于日程将会是开发过程不连续，因此对开发进程需要详细记录。

六、 可行性分析

5.1 系统基本要求

5.1.1 采用架构

本项目采用浏览器/服务器（B/S）架构：浏览器可用 IE8.0 或其他等同浏览器，服务器端使用 Tomcat8.0，数据库采用 MYSQL5.7.1。

5.1.2 主要功能要求

主要分为三大模块功能：个人账户浏览，个人账户管理、（可选）管理员管理系统

（1） 个人账户浏览包括：注册、查询余额、查看个人信息等

（2） 个人账户管理包括：登陆、转账、取款、修改个人信息、注销、登出等

（3） （可选）管理员管理系统包括：用户管理、异常监控、修改用户信息等

5.1.3 系统运行环境

服务器运行采用 Linux CentOS，浏览器端使用装有基本浏览器的操作系统即可，一般是 Window 7 及以上。

5.2 系统开发要求

5.2.1 网站实现

开发周期：文档编写 2 周，代码编写 2 周，软件测试 1 周。

5.2.2 环境搭建

服务器端采用云服务提供商的云主机，此处我们只考虑 windows 系统

5.2.3 技术可行性

此银行存取系统是一个功能比较简单的面向用户的系统，需要实现的都是基本且简单的。网页的数目不多，数据的数量也不大，个人电脑便能满足这个系统的运行要求。

银行存取系统需要使用的技术/语言有：网页开发（JavaScript）、数据库开发（SQL），这两门语言已经成熟且已经有了无数成功开发产品的案例，在开发时我们可以参考这些样例进行开发。

5.2.4 经济可行性（费用开支与效益分析）

（1）租用学生云主机 1 元/月，环境搭建采用开源免费软件。

（2）代码开发，因项目开发为个人银行系统，所需开发与维护人员为 1。

综上，各项费用开支基本为 0，个人银行系统为课程设计网站，纯经济效益为 0。

5.3. 可行性分析总结

上述可行性分析，参考现有开发资料、文档等资源，个人银行系统的体系结构比较完善，开发要求相比于复杂的系统较低，具备进一步进行需求分析与后续开发的条件。

故，此项目是可行的。

以上为项目的可行性分析，个人银行系统的需求分析之前的项目内容中已经分析过。

七、 工作基础

使用 C++编写过酒店信息管理系统（包括预定、查询、取消等）与此有相似之处。

但并未写过网页和数据库，在进行此系统开发前需要学习这两门语言进行知识储备。