



**QG工作室项目报告**

学 院 计算机学院

专 业 计算机类

班 级 12班

组 别 后台

姓 名 陈明楷

学 号 3123004606

2023年 月 日

广东工业大学计算机学院制

目录

[1. 项目简介 1](#_Toc30766)

[2. 设计思路 1](#_Toc24252)

[3. 详细功能设计 1](#_Toc20914)

[4. 程序测试 1](#_Toc7265)

[5. 项目亮点 1](#_Toc19276)

[6. 心得体会 1](#_Toc25747)

# 项目简介

资金管理系统，具有登录，注册功能，且登录后进入主界面可以拥有查看个人信息，修改个人信息的功能，转账，查看收款申请，查看流水，查看自己所属企业，申请企业，查看申请企业情况，并且可以根据企业名称查找企业，并且可以查看公开企业的详情

企业方面，可以加入企业，退出企业，负责人可以，分配企业资金，查看企业流水，邀请用户，查看企业申请，申请解封企业，注销企业

转账方面，可以选择转账来源是用户还是企业，可以选择转账给用户还是企业，使用企业分配资金后还可以申请上传报销文件

转账后的资金保存到数据库中，待定即冻结该资金，已扣除付款方的资金

# 设计思路

首先做了思维导图，优先完成一个登录界面，并完成注册界面，登录后，实现展示公开企业的功能，通过ajax异步获取，因为被封禁的用户无权查看，ajax发送异步请求判断用户是否登录，给予用户不同按钮，若登录则，有退出登录按钮，且会显示该用户的用户名在界面上，并且拥有多种功能可以使用，例如:查看个人信息，修改个人信息的功能，转账，查看收款申请，查看流水，查看自己所属企业，申请企业，查看申请企业情况，并且可以根据企业名称查找企业，并且可以查看公开企业的详情。

实现企业界面，展示企业名称等，但是不展示企业的资金，因为资金过于私密，不应当被任何人看到，进入企业后判断用户是否该企业的给予不同的按钮，非企业成员，拥有申请加入企业和返回主界面功能，企业普通成员具有申请成为企业负责人、退出企业和返回主界面功能，而企业负责人则拥有邀请用户，分配企业资金，查看企业流水，查看申请信息，设置企业基本信息，充值功能，还有注销企业的功能，还有补充一个查看报销文件以及下载功能（并不完善）。当企业被封禁时，非企业成员无法查看该企业，企业负责人失去全部功能，只剩下申请解封和返回的按钮。

转账功能中可以查看尚未处理的转账申请，以及申请后尚未通知的转账申请，会展示用户的资金，用户选择企业后也会展示企业分配资金，选择企业后可以选择是否上传文件，且一些按钮数据都是层层出现，而非一次性全部产生，即确认选择资金后再显示转账用户，确认转账用户后再输入转账密码，层层递进，而不是一次性全部展示

修改个人信息，用户本身会有个头像，可以更改头像，更改其他个人信息，以及更改支付密码，但是需要输入原支付密码（防止被盗号的情况）才能修改支付密码。

查看流水信息，用户可以对转账状态为失败的转账进行再次转账，而成功的和待处理的则无需重新处理转账。

对于转账功能的实现，我的想法是，用户发起转账，后扣除用户资金，并将这部分资金保存到转账订单中，而这个过程是一个事务，成功就提交，不成功就回滚，防止资金的数据异常，导致整个系统的资金数据的不稳定，然后设计的转账功能是需要收款方确认收款的，类似于微信收款的方式，当收款方不确认时，这部分资金就保存到数据库中冻结，付款方也无法对这部分资金重新取出，收款方接受收款请求后，更改付款订单信息，并增添资金到用户中，这部分也是通过事务控制，防止资金异常。

# 详细功能设计

登录功能的完成通过ajax发送异步请求，判断用户输入的账户密码是否存在于数据库中，即通过两个条件判断是否存在该数据，不存在则返回失败，在这部分必须放在sql注入，防止账户被非法入侵，同时使用了验证码功能，且会先判断用户输入的账户密码是否正确再判断二维码是否输入正确，若是账号密码输入错误，验证码也会刷新、验证码输入错误，验证码也会刷新，验证码的值通过session保存

注册功能，首先id是系统自动分配的，用户名，昵称，密码，手机号等用户自己输入，同时输入中全部使用正则表达式控制输入的数据，若不符合正则表达式则会提示错误，且用户无法提交注册内容，其中用户名通过正则后还会通过ajax发送异步判断用户名是否存在，防止用户名重复导致数据混乱，并提示用户名已存在，同样的手机号码也会通过正则表达式控制，只能输入11位纯数字，通过正则表达式后还会判断手机号是否已经注册过，防止同一手机号码重复注册，导致非法人员通过一个手机号注册很多个账户导致数据库崩溃。同时，再注册界面会使用验证码验证，同样是为了防止非法用户尝试通过脚本轰炸数据库所使用的功能，且验证码输入错误后也会重新刷新，业务逻辑相对正常。

在主界面，首先通过ajax判断用户是否登录。若未登录，则告知用户，同时给予登录按钮，并且只能查看企业的信息，无法申请加入企业，只能查看公开的企业，封禁的企业点击查看详情会被强制返回主界面；若登录了，则会判断用户是否被封禁，封禁则只有申请解封和退出登录的按钮，且主界面不会展示企业的信息；若未被封禁，则判断是否未网站负责人，若是普通用户则拥有基础功能，例如：修改个人信息，查看流水，转账，申请企业，查看申请企业情况，查看我的企业（可以查看到私有企业），以及查看公开企业的相关信息，若是网站管理员，则判断可以查看企业信息，查看所有用户信息，查看解封申请，查看所有用户流水（这里未能实现查看异常转账流水的功能）。同时在主界面使用计时器，实时发送ajax给后台判断用户是否被封禁，若是被封禁则会直接停止用户的一切行为，并重新跳转到主界面，只剩下被封禁用户的基本功能。其中判断用户登录后的用户信息，是在后台使用cookie保存到本地浏览器的方式储存用户信息（这是通过试验b站网站的登录功能得出的灵感）。

查看企业信息的功能，在进入该企业信息后会通过ajax发送异步请求判断用户是否被封禁，若是被封禁则直接返回主界面，若是没有被封禁则通过ajax发送异步请求获取信息，展示企业的相关信息以及通过ajax发送异步请求判断该用户是否为网站管理员，若是网站管理员则可以对未封禁企业进行封禁，对封禁企业进行解封，若是普通用户，则再次通过ajax发送异步请求判断该用户是否未企业负责人，若是普通用户则判断该用户是否为企业成员，若是企业成员，则给予申请成为企业负责人的按钮，申请发送是否成功都会弹出提示，还可以退出企业以及返回主界面；若是企业负责人，则可以邀请非企业成员进入企业（输入用户名称时会判断该用户是否为企业成员，防止重复邀请）、可以分配企业资金，查看企业的收款申请，修改企业基本信息，查看企业信息，查看企业流水，查看企业申请信息（即用户申请加入企业，以及申请成为企业负责人的申请），查看报销文件（可以下载文件），注销企业（会用ajax判断企业剩余分配资金是否为0，不为0则不允许注销，为0，则根据企业分配资金的情况，将资金发送给每一个企业成员，并删除所有企业相关信息），也可以退出企业.

转账功能，进入该功能后，会先用ajax判断用户是否被封禁，然后展示用户剩余资金（实时更新），当用户选择使用用户资金的时候，直接进入转账界面，当用户更换使用的资金时候，转账界面会清空，当用户选择企业资金后会要求用户选择使用哪个企业的资金，选择后会判断企业是否已经被封禁，被封禁则告知用户企业被封禁，并且无法进入转账界面，未被封禁则展示企业分配资金（实时更新）,并进入转账界面，转账界面先是让用户选择转账对象类型，企业或者用户，选择后再显示转账对象的界面，对于转账对象都会用ajax判断该用户或者企业是否存在，存在才能发起转账，不存在会提示用户，并且无法发起转账，输入转账金额，使用正则表达式，只可输入正数小数，否则会无法通过正则表达式，输入金额合法后，还会判断金额是否溢出，溢出则无法发起转账，不溢出则可以发起转账，发起转账后，要求用户输入转账密码（若是动用企业资金则会显示一个提交文件的功能，用户可以选择是否提交文件，提交文件的话则会增添一个提交报销文件的数据），会先用正则表达式判断用户输入的密码是否为6位纯数字，若不满足则会提醒用户，若为纯数字，则通过ajax判断该用户对应的密码是否正确，正确则发起转账，发起转账后，后台先是设置一个事务，并且锁住转账用户的资金，防止被其他人再次查询，然后设置转账订单信息（即冻结该资金），再扣除用户数据库的资金，当这两步功能同时成功后，提交事务，反之则回滚事务，成功后会告知用户发起转账成功并返回主界面，失败也会告知用户发起转账失败。其中转账中的业务逻辑通过实现一个接口，即第三方，可以对用户的资金进行控制，以及订单的生成，这部分逻辑都是第三方实现的。

收款功能，用户可以对于尚未处理的收款申请进行处理，即接受收款功能或者拒绝收款功能，若是接受收款，则发送ajax异步请求，给服务器第三方，第三方接受请求，若用户接受收款，则设置事务，将用户资金数据上锁，更改用户资金数据，并更改订单的处理情况，当两步都正常完成后，提交事务，若用户掘金收款，则第三方涉资事务则根据订单信息返还资金跟付款方（若转账主体为用户，则锁住付款方用户的资金，并返还资金给用户，若主体为企业，则锁住企业资金和企业分配资金，返还给企业和企业分配给对应用户的资金），然后更改转账订单的处理情况为拒绝，并提交事务，成功则告知用户并刷新页面，失败也告知用户但是用户可以重复处理。且对于同一用户的多个使用者或者同一企业的企业管理员对于同一申请在短时间内同时处理，则会进行甄别，即判断是否已经被处理了，防止订单已经被处理导致出现了二次处理影响资金流通的稳定性。

查看流水方面，会对于已经完成的转账情况，但是转账方尚未得知的情况进行告知，并且转账的列表是从距今时间的长短排序的，距今时间短的在上面，依次类推，这是仿照微信的查看流水情况的功能进行使用的，且在查看流水的时候，会对于自身账户进行加粗，有利于用户甄别流水的转账方和收款方是不是自己，但是未完成打印流水的技术

在修改个人数据的功能中，头像使用url的方式保存，修改用户名或者手机号等信息都会通过ajax发送异步请求判断是否已经存在该信息，防止信息的重复，用户修改支付密码时，需要用户输入原支付密码以甄别是否被盗号的情况发生。

在企业分配资金的功能中，本身是设计希望只能由一个管理员进行分配，但是出现了死锁的情况发生，因此打算另辟蹊径，即分配企业资金后确认时候再锁住企业的总资金，且分配完后会判断分配后的资金是否溢出，溢出则告知用户，不可溢出，请重新输入，不溢出则发异步请求实现业务，使用锁的机制，完成业务逻辑，防止资金的不正确性发生，同时为了方便企业负责人对于资金的分配，会在界面上告知企业负责人企业总资金，以及剩余可分配的企业资金为多少，且会进行实时更新。

企业处理申请功能，企业负责人可以对于申请进行处理，且会判断是否已经被处理了防止二次处理，导致数据异常，通过对申请描述的判断实现对应的代码逻辑，即申请加入企业则添加企业关系表，若是申请成为企业负责人则更改企业关系表leader为yes，即设置申请用户为企业负责人。

企业负责人查看报销文件，可以观察到用户的对于企业的资金的动用时间和金额以及可以下载报销文件以查看是否合理报销，但是下载功能尚未完善，即需要重启服务器才能下载到指定文件，否则无法下载到指定文件

网站管理员，可以查看所有用户，可以对未被封禁的用户进行封禁，也可以对于封禁的用户进行解封，同理对于企业而言也是如此，并且网络管理员是没有转账功能的，只有查看用户信息，查看所有流水，查看企业，处理解封申请的功能，处理解封申请是会将未处理的展示给网站负责人并使得他进行审批，但是同时为了防止同一申请短时间内被多个管理员审批，还是进行了甄别处理，防止了申请被二次处理，且处理完后会记录审批人的用户名。

用户在发起申请加入企业以及申请成为企业负责人和申请解封的功能都会进行判断是否以及发起过一个尚未处理的申请，防止同一用户多次申请相同的内容导致了数据库的崩溃，且影响到网站管理员对于申请的处理的简便化，因此为了简便网络管理员的工作，且防止数据库被轰炸，要对插入数据库的数据进行甄别处理。

# 程序测试

登录界面展示

图片包含 图形用户界面

描述已自动生成

成功登录

图形用户界面, 应用程序

描述已自动生成

普通用户主界面

电脑萤幕的截图

描述已自动生成

网站管理员主界面

图形用户界面

中度可信度描述已自动生成

封禁用户主界面

图形用户界面

中度可信度描述已自动生成

用户查看个人信息功能（头像输入头像的url即可更改）

图形用户界面, 文本, 应用程序, 表格, Excel

描述已自动生成

更改用户信息功能方面

文本

中度可信度描述已自动生成

修改支付密码

图形用户界面, 文本

描述已自动生成

转账功能的实现

图形用户界面

描述已自动生成

查看收取转账的功能

应用程序, 表格

描述已自动生成

对转账进行处理，注意点击了接受按钮后，该按钮会被禁用

图形用户界面

描述已自动生成

查看我的企业的功能（可以查看到私有的企业）

图形用户界面, 应用程序, Word

描述已自动生成

查看企业详情的功能，普通用户

图形用户界面, 文本

描述已自动生成

普通用户可以申请加入企业

图形用户界面, 应用程序

描述已自动生成

申请成功后并成为企业负责人

图形用户界面, 文本, 应用程序

描述已自动生成

可以邀请用户进入企业

图形用户界面, 文本

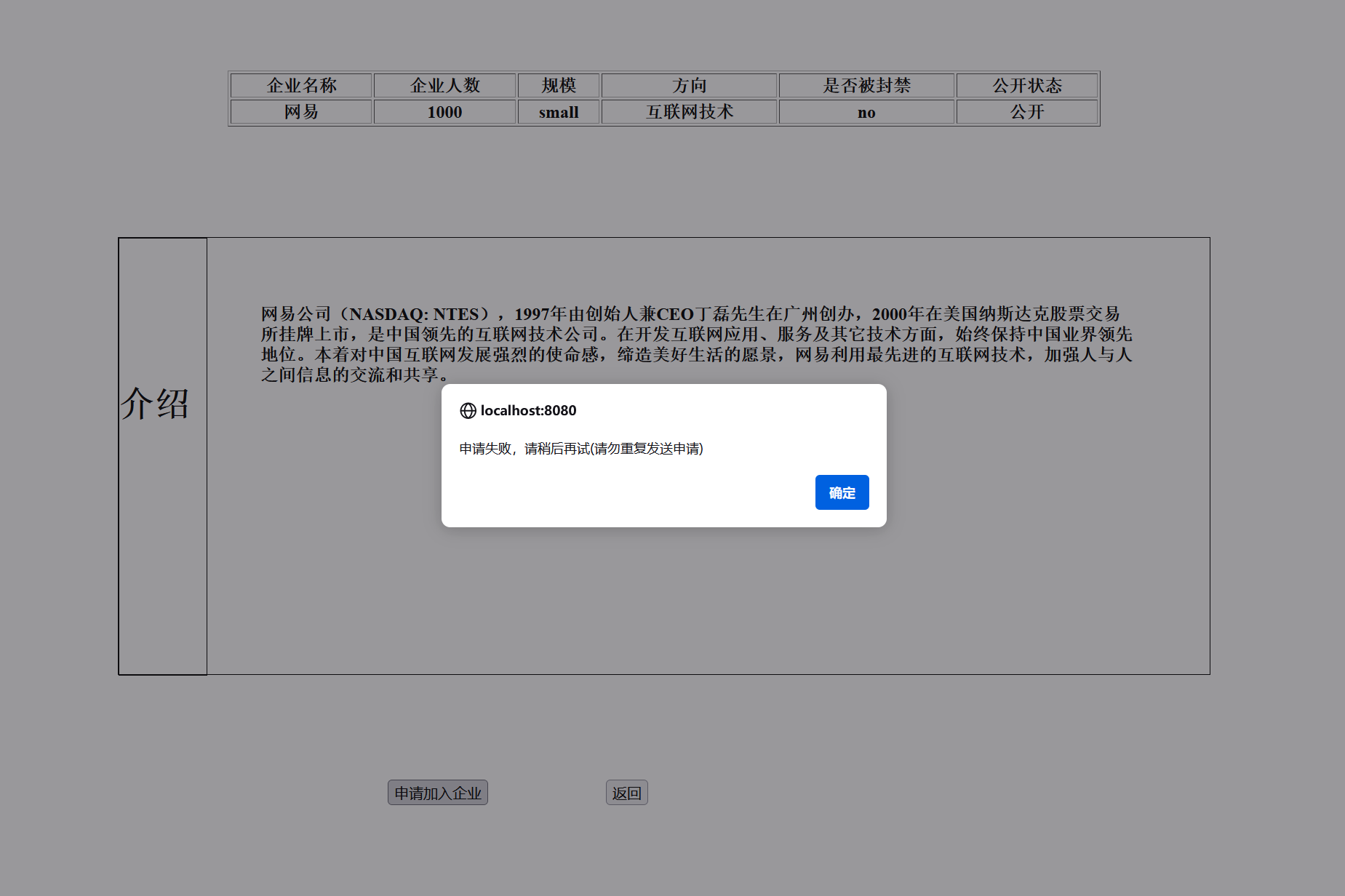
中度可信度描述已自动生成

申请成为企业负责人的功能

图形用户界面, 应用程序

描述已自动生成

禁止用户重复申请



企业负责人审批申请

图形用户界面, 应用程序

描述已自动生成

企业负责人可以修改企业信息

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本

描述已自动生成

修改完成界面

图形用户界面, 文本, 应用程序

描述已自动生成

企业负责人可以查看企业流水，企业信息加粗显示，便于负责人查看

表格

描述已自动生成

企业负责人可以对于企业进行充值操作

图形用户界面, 文本, 应用程序

描述已自动生成

企业成员可以退出企业

图形用户界面, 文本, 应用程序

描述已自动生成

同样的企业负责人可以注销企业，会进行资金判断，若剩余分配资金不为0则



若剩余分配资金为0则会弹出提示框，防止误触导致不小心注销了企业

图示

中度可信度描述已自动生成

确认注销后后台会进行业务实现，成功后会通知用户注销成功，失败也会通知用户注销失败了，且注销完成后，会根据企业资金分配情况将企业资金分配给每一个企业成员，防止这部分资金凭空消失。

转账功能展示(若是只用企业资金，则可以让用户选择是否上传报销文件，若是使用个人资金则无需上传报销文件，且上传报销文件有可能会失败，会提醒用户是否重新上传)

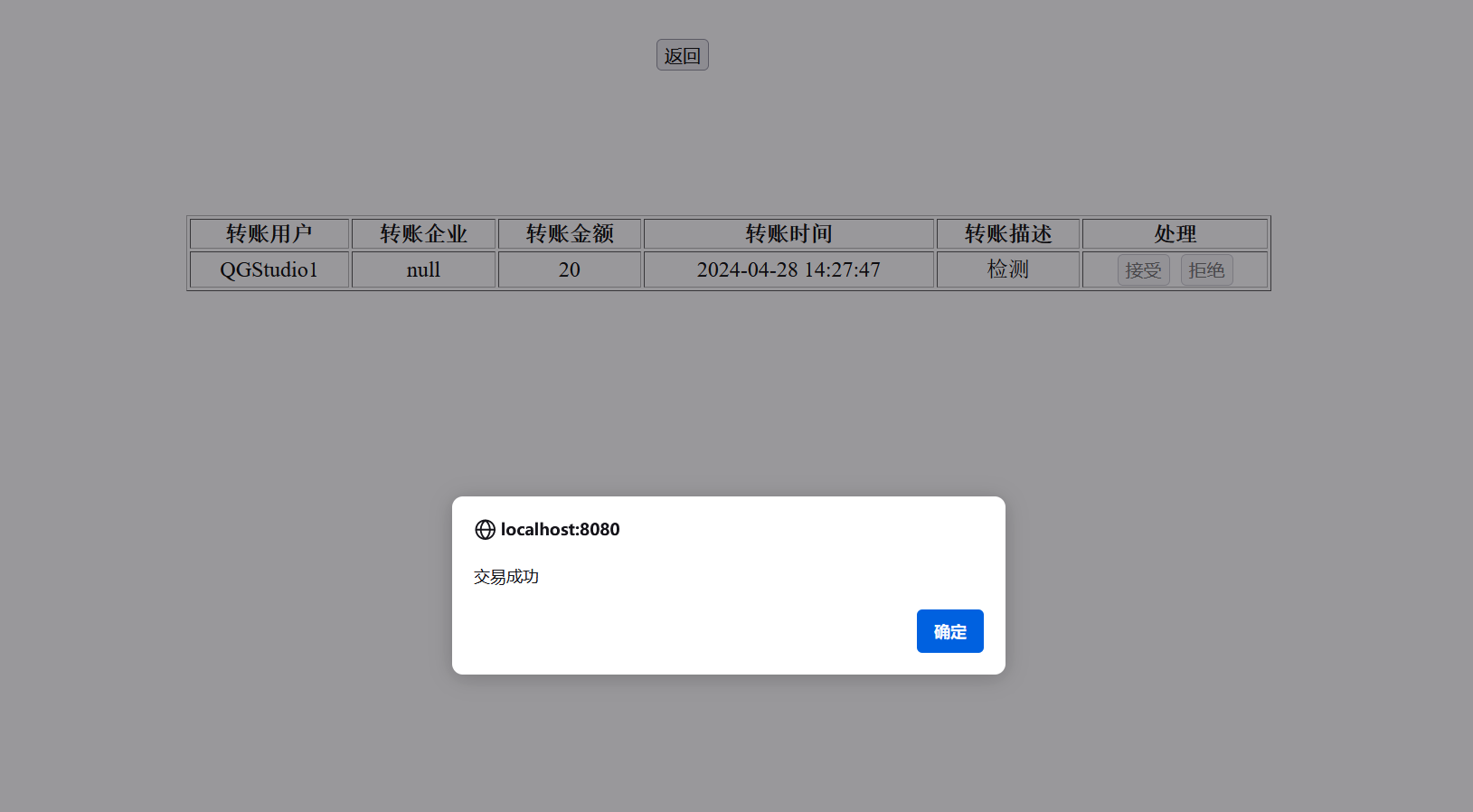
使用个人资金



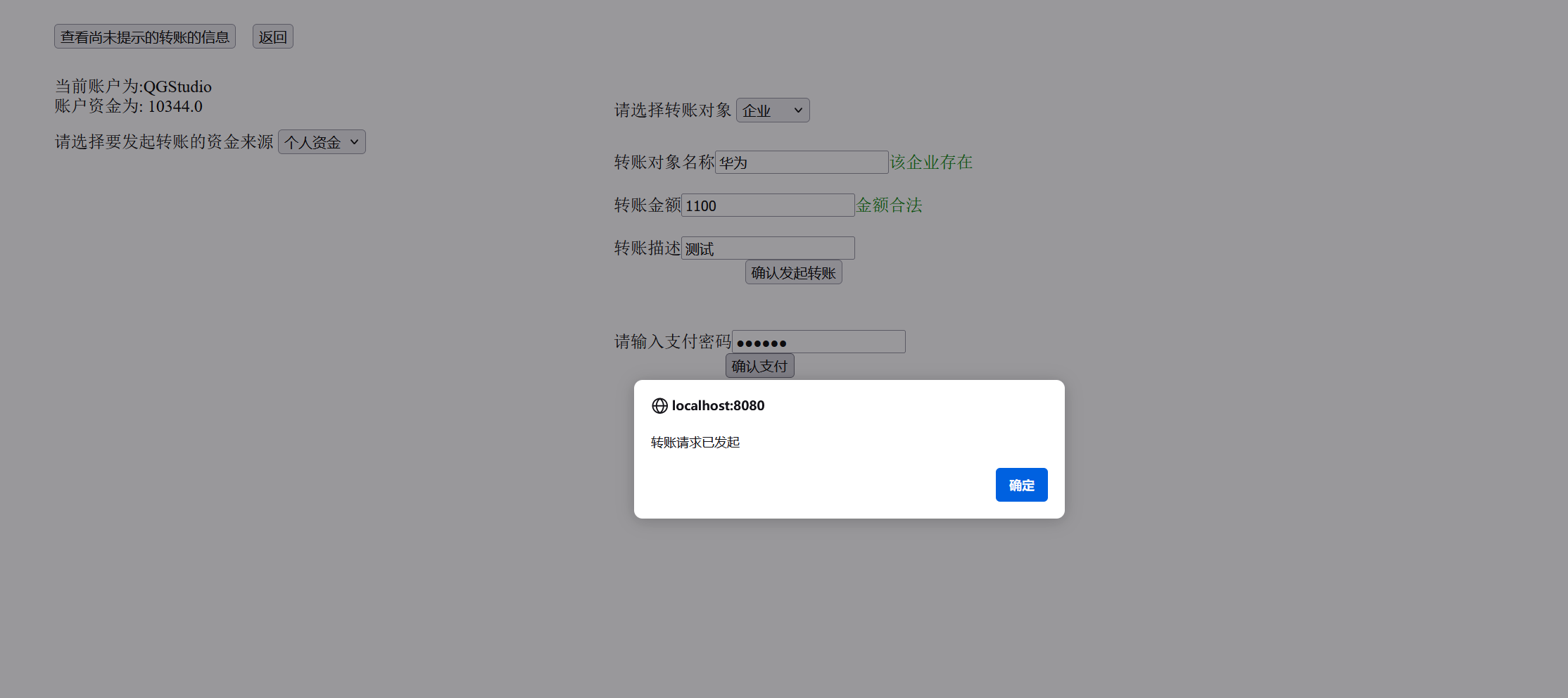
使用企业资金



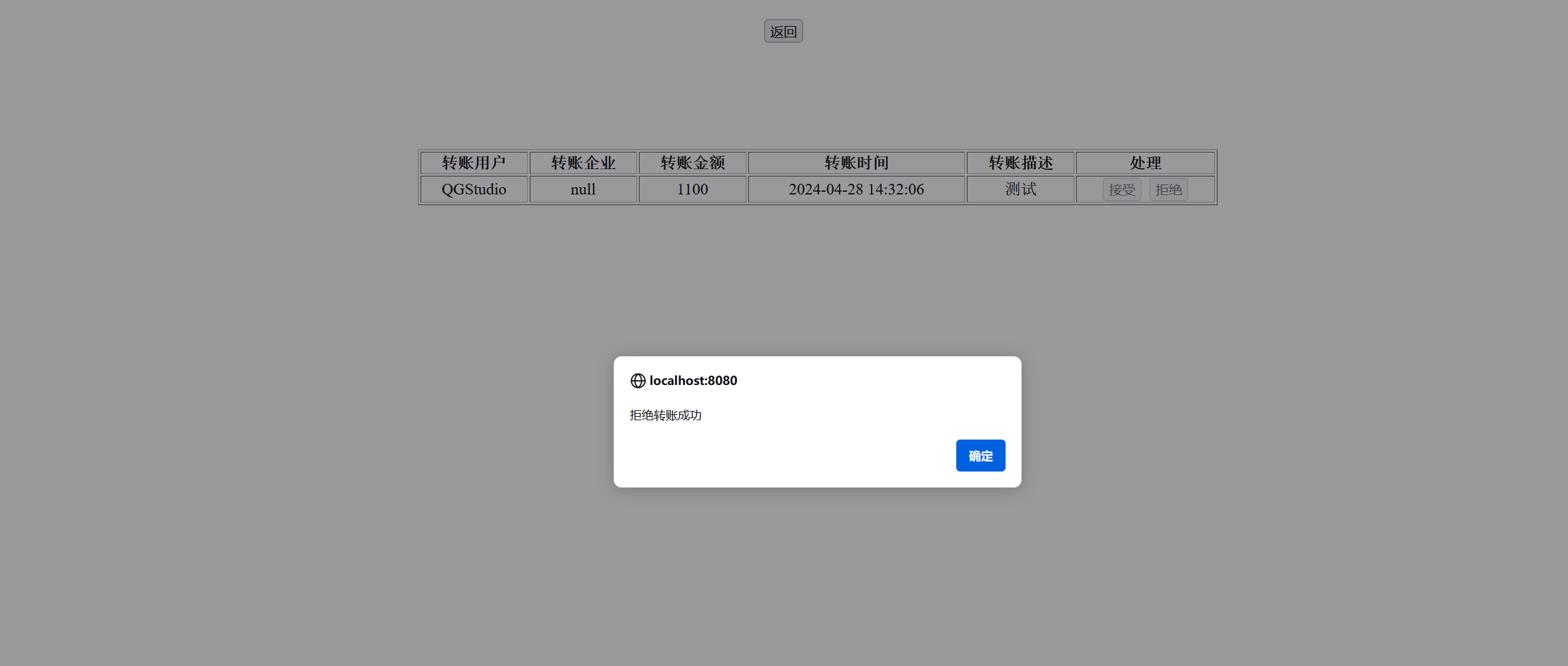
用户的收款功能，用户可以选择接受，也可以选择拒绝（点击完接受或者拒绝按钮，这两个按钮会同时被禁用，防止用户一次性多次点击该申请）；



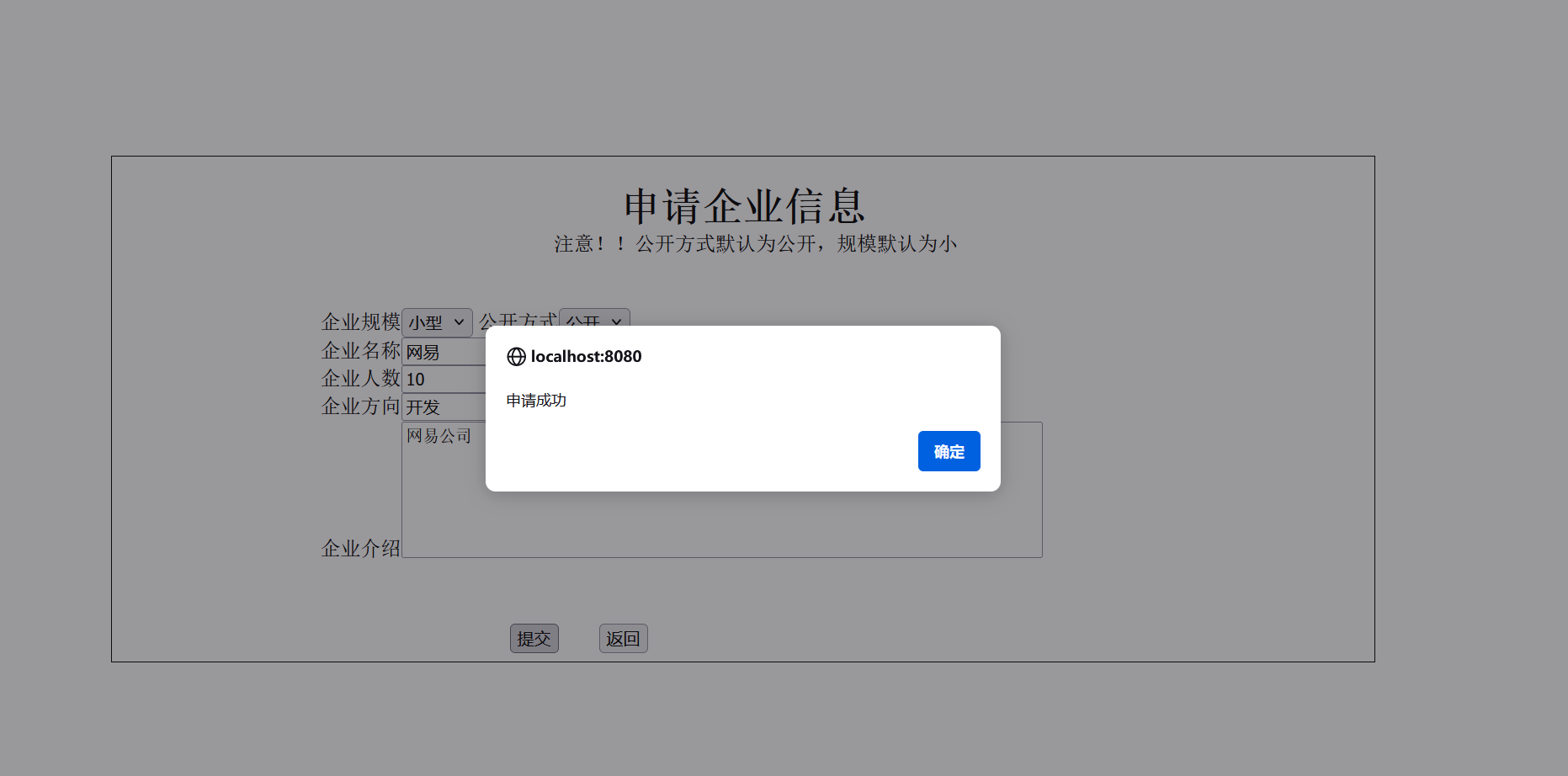
用户向企业发送转账申请



企业对于转账申请进行处理（可以接受转账也可以拒绝转账）



用户申请企业功能



用户查看申请企业的结果，且可以知道由谁处理的申请

图片包含 图示

描述已自动生成

用户查看流水后会先前由待处理的转账申请转化为结果的申请进行提示

图形用户界面, 文本, 应用程序, 聊天或短信

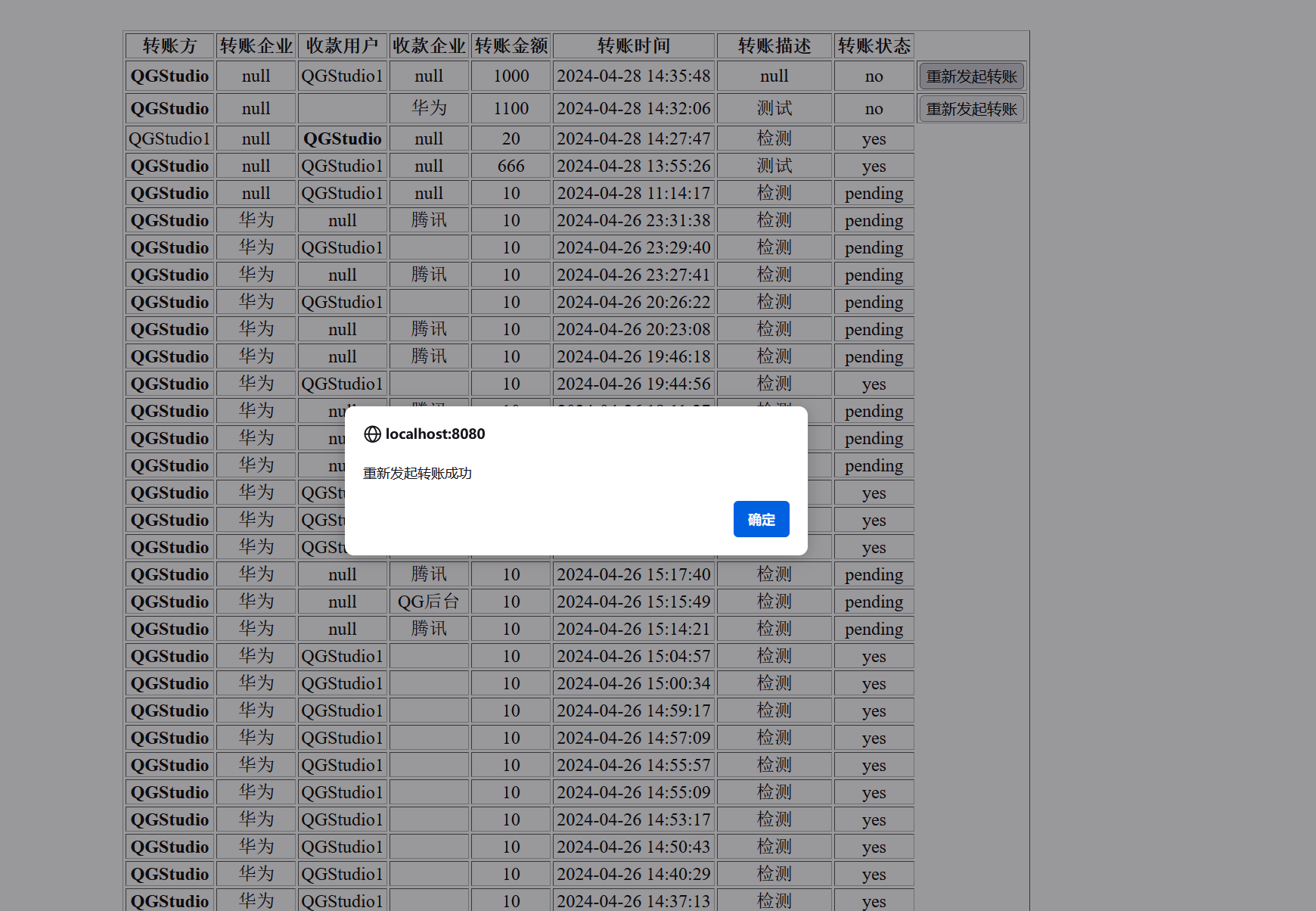
描述已自动生成

查看流水

表格

描述已自动生成

同时可以对于失败的转账申请进行重新申请



网站管理员可以对于企业申请进行审批

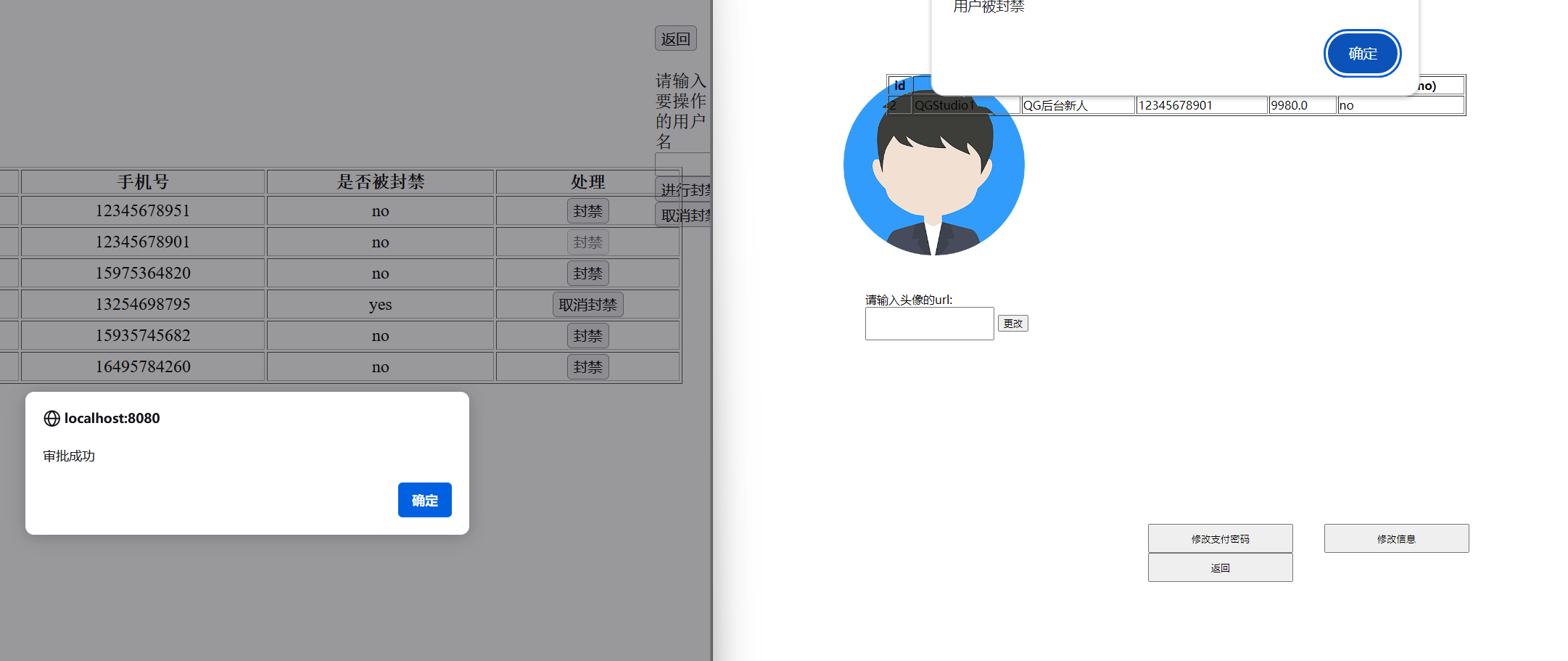


网站管理员查看用户信息

表格

描述已自动生成

网站管理员可以对于被封禁的用户进行解封也可以对于未封禁的用户进行封禁，也可以根据用户名进行精准封禁（且会对于用户的一些操作进行实时的封禁处理）



若是解封未被封禁的用户则会发出提示

表格

中度可信度描述已自动生成若是封禁已经封禁的用户也会有提示

表格

描述已自动生成

以下是正常情况下的操作

表格

描述已自动生成

被封禁用户可以申请解封

图形用户界面, 文本, 应用程序

描述已自动生成

网站管理员可以对解封申请进行审批

表格

描述已自动生成

网站管理员查看企业信息（可以对于未被封禁的企业进行封禁）

图形用户界面, 文本

描述已自动生成

封禁企业后，也会进行实时提醒，可以冻结正在进行的操作

图形用户界面, 图示

描述已自动生成

可以对于封禁的企业进行解封

文本

描述已自动生成

当企业被封禁时，企业负责人可以申请解封企业

文本

描述已自动生成

同样的网站负责人也可以处理该申请

图形用户界面, 应用程序

描述已自动生成

企业负责人查看报销文件并下载。

表格

描述已自动生成

# 项目亮点

首先，在注册和登录界面实现了验证码的功能实现，用户可以通过点击验证码进行更改验证码的图片，同时用户输错验证码后会自动更改验证码的信息，同时输错其他信息也会导致验证码刷新，在注册和登录界面均可以制止脚本注入数据，因为验证码是动态变化的。

其次，实现了对用户和企业封禁的实时检测，即企业被封禁后，在企业界面中的用户都会受到提示，并进行分辨，普通用户、游客和企业普通成员都会直接退出企业界面，而企业负责人则会返还企业主界面且只剩下申请解封企业的按钮，而企业负责人则是控制企业是否被封禁；用户被封禁后会在任何界面都告知被封禁了，并强行拉回主界面（主界面没有进行实时检测，因为用户无论进入任何界面都会被检测到是被封禁了，相当于失去了所有的功能）。

实现了对于敏感信息的加密，对于用户密码使用哈希算法进行加密处理，再保存到数据库中，同样的用户支付密码也是使用哈希算法进行处理的，处理完再保存到数据库中，但是两个密码使用的哈希算法并不相同，这样会更加严密，即数据库被入侵了，也无法知道密码的格式是什么，因为哈希算法加密后是无法反向解密的，对于敏感信息有了安全的保障.

在实现转账功能中，对于业务逻辑相对严谨，即虽然进入转账界面后用户可以重新更改使用资金的选择，但是会将转账界面刷新，防止当企业资金足够时，用户突然转为个人资金但是个人资金其实是不够的，躲开的正则和ajax的判断，这样实际上是不太合理的，因此为了防止这种情况的发生，前端在判断用户更改后会刷新掉转账界面防止出现上述情况导致用户或者企业分配资金为负数的情况，这显然不符合实际，同样的在发起转账时候会判断资金是否合理，并且发送异步请求判断用户是否存在，发起转账后同样用户可以更改资金等信息但是当用户想要通过这种方式溢出转账的时候，点击支付按钮并无法成功因为在设置转账等地方控制了失去焦点的事件，即将付款界面清空，这样能够有效防止转漏洞的情况发生，同时也可以保证资金流动的稳定性。

在实现对转账申请的处理的功能，由于类似于微信转账一样的功能，若是出现一次性多次点击的确会导致资金异常，因此当用户点击接受或者拒绝任何一个按钮都会将两个按钮都禁用，即用户只有可能点击一次，也就是防止服务器卡住后用户收款多次或者退款多次的情况发生，因此这种禁用按钮的方式可以防止资金的异常，直到服务器响应了结果给前端后，前端进行判断处理成功则刷新页面，处理失败则恢复两个按钮，并使得用户可以重新处理申请，并且防止了服务器卡顿后用户的多次点击的可能性，从根源上遏制了多次请求的可能，即便有能够转漏洞的也可以在服务器对申请进行判断，若非待处理状态，则返回错误，并告知用户。

对于网络管理员观察网站总资金是实时更新的，而且转账功能中的个人资金和企业资金同样是实时更新的，防止了资金的变动后，前端并不知晓导致了资金异常的可能，因此资金的显示是实时更新的，给网站管理员实时更新的缘故是利于观察网站资金是否发生异常，正常情况下网站总资金是不会发生改变的，倘若发生了改变，则说明了有人利用bug刷金额了，这时候网站管理员可以查看网站流水，并查看转账用户，对转账或者收款用户进行封禁处理，而转账功能中的则是防止了资金出现负数的情况发生以及资金凭空出现的可能，这样会导致资金的不稳定性，因此这里是需要实时更新的，同时也是因为这里的判断资金是否溢出的来源是网站上展示的用户资金和企业资金，这样也会导致资金溢出判断的失误，因此这并不合理。

并且实现了找回密码的业务逻辑，即用户可以通过注册时候的用户名和手机号（手机号码）实现找回密码的功能，因为没能实现发送手机短信的功能，只能是查找用户手机号码了，同时在这里用户更改用户名后，找回密码界面会再次刷新防止转漏洞通过已知的号码去更改未知的号码的功能，即防止了账户被入侵的业务。

收款功能中，对于高并发的情况，即对于一个收款请求的多次处理，通过锁的机制以及判断该订单是否以及被处理的功能，保证了即使在高并发的情况下，同一收款请求并不会被多次处理，这有效的防止了高并发问题，因为如若出现这类高并发问题，会影响到资金的传递的稳定性，容易平白无故网站多了一笔资金，这并不符合转账功能的逻辑，因此通过这种判断方式可以抑制住高并发的情况导致的多个请求对于同一申请的处理，这不仅仅体现在收款方面，企业处理申请，网络负责人处理申请都运用到了这个功能，同样都避免的高并发的情况导致的数据异常.

# 心得体会

在实现项目的过程中，首先我对于我先前的mvc架构的理解有了更深的理解，即我先前学习的时候，Dao层的代码其实写的并不是特别好，即复用性极差，一个servlet对应一个dao层的方法，但是因为我将从请求或者cookie中获取数据的部分给到了dao层的方法，这就导致了其实servlet层实际上并没有多少代码可以写，但是实际上开发并不是这样子的，应该由servlet层获取请求中的数据，然后将这些数据调用dao层的方法去实现对数据库的CRUD，但是我先前写的代码并没有按照这个mvc架构的方式实现，直到我后面需要调用某个类似的sql语句的时候，我发现了dao层代码复用性极差，无法满足所有servlet层的使用，即一个dao层方法只能被一个sevlet的方法使用，这显然并不符合dao层代码的需求，因此我在发现了我这个代码复用性并不好，因此我在后续完成项目的过程中，已经有意识的更改相应的代码，使得dao层方法更具有复用性，而不是一个dao方法只能给一个servlet方法使用，这个经验是亲身经历后才能理解的，先前虽然有听说过，但是当自己亲身经历，亲身面对这种情况的时候，我对mvc架构的理解有了进一步的认知。

其次，我在实现项目的过程中，懂得了要以使用者的角度去面对我的项目，因为用户并不会一定按照我们的想法去做某些事情，我不能只以开发者的角度去实现我的业务，否则用户倘若不按套路出牌，反其道而行，反而会导致服务器的崩溃等问题，因此是需要对用户等信息进行甄别，即同一个页面，对于不同的身份用户来说，界面的展示不应该是完全相同的，而是有所不同，即用户有用户的界面，游客有游客的界面，企业负责人同样有企业负责人的界面，网站管理员有网站管理员的界面，这都是不一样的，是需要开发人员进行甄别的，我们不可能要求用户只能按某个按键，我们应该是将用户可以使用的功能全部展示给用户，而用户无法使用的功能自然不展示，这是十分有必要的，否则用户体验感极差，每次使用功能前都要先去查看自己是什么身份能使用什么功能，这样对于开发人员来说是轻松了，但是用户的体验确实非常的不好。因此我也懂得了实现项目的时候，要考虑到用户的使用，而用户是一个非常不确定的因素，我们不知道用户会如何使用我的项目，因此我必须使得用户能够比较舒适的使用项目，而不是要求用户这里该怎么做，那里该怎么做，用户不一定会听我的要求，但是我可以控制用户不得不听我的要求，否则将无法实现想要的功能，这样对于用户来说也更加合理。

在实现项目的时候，我同样意识到了自己技术的不足，在实现实时监控用户是否被封禁的过程中以及实现实时展示资金的功能中，我意识到需要使用多线程的方式去实行，因为使用单一线程的方式并不是最好的方法，因为这样的话当用户多起来，容易使得主线程几乎全部都是对于用户的监控以及资金的展示的业务了，这样容易使得其他业务十分缓慢，卡顿，因主线程已经被爆满了，另外开辟一个线程更能有利于主要业务的实现，但是由于线程方面并未很好的掌握，且由于时间关系，并无法更改实时监控的demo，因此导致了目前线程的容易被非主要业务占据，但是因为时间关系以及对线程的掌握并不深刻，导致这个问题并没有得到解决，因此这个项目主要还是单线程的方式去实现的，但是其实我认为应该使用多线程的方式去实现这些实时检测的业务逻辑，他们不应该占据主线程的大部分空间，否则会十分的影响服务器的流畅度。用户一多，服务器卡顿，十分影响用户的使用体验感，这个问题也映射了我对于多线程方面的知识并未很好掌握，后续还需加强补充学习这方面的知识。

在设计前端界面中，我由于对于vue等框架掌握并不是很好，因此前端界面我并没有使用vue等框架，仅仅使用html，js等实现界面的展示，因为框架的学习并不多，因此不能熟练的使用框架，这对于短时间内实现项目来说是十分影响的，所以我在实现项目中并没有使用前端框架，只是用html简单展示了一些应该有的界面，即简单展示出应该有的数据，并没有使用框架展现一个非常绚丽的前端界面，这也是我对于前端框架的学习不够导致的。

在中期进度检查当中，看到别人的项目使用的技术基本没听说过也是大为震惊，也意识到了自己技术有所欠缺，并没有什么突出技术，技术的学习仍需努力，同时也从导师口中得知了，作为开发人员，是要懂得对于甲方提供的要求进行深度剖析的，并不会像导师一样写的如此详细，懂得了开发项目这一任务并不只是写代码这么简单，要分析甲方的业务需求，要和前端的进行交流，一个项目一般来说不可能只是一个后台完成这么简单（除非是全栈），后台是需要跟前端进行数据交互的，要有数据交流的，前后端是紧密相连的，虽然现在前后端分离，但是后台依旧需要学习一部分前端知识，我也明白了原因，学习了前端知识，后台才知道应该如何传输给前端数据，以及这样传输是否是正确的，前端是不是可以拿到这部分的数据，在学习前端知识后，才能自己去实践，去体验数据交互的过程是不是如何的，前端又是如何发送数据给后台的是post还是get这都是不一样的，后台要获取这部分的数据使用的方法都是不一样的，这都是在写项目中体会到的，理解的。

在完成整个项目的过程中，我切切实实的体验了一把作为开发人员的日子是如何的，整天坐在电脑前面敲代码，终日面对着demo，确实很难，也很苦，没有其他舍友那么轻松，无忧无虑，这种日子确实很难熬，不过每次完成每个功能后都会有种莫名的开心，因为自己的努力并没有白费，虽然说这是我第一次写一个java项目，我并不觉得我完成的有多么出色，多么突出，但是我知道这是我日日夜夜努力的结果，在写项目的时候确实很累，但是完成项目后有着实实在在的惬意感。

在跟着qg训练营的这段日子里，即寒假训练营和开学是训练营，到最终考核，加起来得有接近三个月吧，这段日子跟着训练营学了很多东西，也更加理解后台开发的任务是什么，使得我在学习java这方面有了明确的目标、同时训练营也教会我们要提前学习数据结构与算法，学习了，也会写demo但是目前还不知道应该如何去使用这些数据结构与算法，可能后续慢慢就会有所了解吧。跟着训练营的日子，也让我感受到了，我不是一个人在努力，还有很多人在努力，虽然很多人最后都坚持不下去了，但是好在我坚持下来了，我可能技术并没有学习到多好，但是这个过程中磨练了我坚持不懈的意志，几近崩溃的边缘，是重新站起来的意志支撑着我，这段经历很难熬，但是对我来说一定是一段难忘的经历。