****

**成人高等教育毕业论文(设计)**

基于Golang的项目回款管理小程序的设计与实现

**学生姓名 张健 学号 2018310500**

**指导教师 谷雨 学习形式 业余**

**院、系、站点 合肥工业大学继续教育学院**

**专业年级 2018级**

**完成日期 2020 年 03 月 31 日**

**合肥工业大学继续教育学院**

基于Golang的项目回款管理小程序的设计与实现

摘##要

当今世界，随着“互联网+”经济的兴起，互联网的服务被广泛应用，人们逐渐感受到互联网所带来的便利，从这几年出现的各种记账APP，人们享受互联网提供的便捷服务。

基于微信是社交工具领域中的霸主，微信小程序的轻量级等特点上手更容易，操作简单。所以本文以团队回款为使用场景，以微信小程序为载体，基于云端模式，设计并开发出一款适合小型创业团队、工作室、自由职业者等人群使用的辅助工具，方便团队管理者轻松管理账目。

项目回款管理系统是基于Golang、MySQL、Redis、微信小程序MPVUE框架，微信原生API调用、报表分析，并通过总体设计、详细设计对项目进行有效管理，极大解决项目回款，提高项目回款管理便捷性。

关键词：项目回款；微信小程序；MPVUE；Golang

Design and implementation of project return management applet based on Golang

**Abstract**

In today's world, with the rise of the "Internet +" economy, Internet services are widely used, and people gradually feel the convenience brought by the Internet. From the various accounting apps that have appeared in recent years, people are enjoying the benefits brought by the Internet. Convenient service.

Based on WeChat is the dominant player in the field of social tools, the lightweight features of WeChat Mini Program are easier to use and simple to operate. Therefore, this article uses the team's cash back as the use scenario, uses WeChat mini-program as the carrier, and based on the cloud model, designs and develops an auxiliary tool suitable for small entrepreneurial teams, studios, freelancers and other people, Easy for team managers to manage accounts.

The project payment management system is based on the Golang, MySQL, Redis, WeChat applet MPVUE framework, WeChat native API calls, report analysis, and effective management of the project through the overall design and detailed design, which greatly solves the project payment and improves the project payment Convenience of money management.

Keywords: project payment; WeChat applet; MPVUE; Golang

**目##录**

[基于Golang的项目回款管理小程序的设计与实现 I](#_Toc36914238)

[Abstract III](#_Toc36914239)

[1. 绪 论 1](#_Toc36914240)

[1.1 背景介绍 1](#_Toc36914241)

[1.2 研究意义 1](#_Toc36914242)

[1.3 使用开发技术 1](#_Toc36914243)

[1.3.1 JetBrains Goland平台特点 2](#_Toc36914244)

[1.3.2 Golang 2](#_Toc36914245)

[1.3.3 Mysql数据库 2](#_Toc36914246)

[1.3.4 微信小程序 2](#_Toc36914247)

[1.3.5 MPVUE框架 2](#_Toc36914248)

[1.4 微信小程序开发流程 2](#_Toc36914249)

[1.4.1 获取微信小程序AppID 2](#_Toc36914250)

[1.4.2 安装微信开发者工具 2](#_Toc36914251)

[1.4.3 创建项目 3](#_Toc36914252)

[1.4.3 编写代码 3](#_Toc36914253)

[1.4.4 创建页面 3](#_Toc36914254)

[1.4.5 预览页面 3](#_Toc36914255)

[2. 需求分析 4](#_Toc36914256)

[2.1可行性分析 4](#_Toc36914257)

[2.1.1经济可行性 4](#_Toc36914258)

[2.1.2技术可行性 4](#_Toc36914259)

[2.1.3社会可行性 5](#_Toc36914260)

[3. 项目回款系统设计 6](#_Toc36914261)

[3.1系统总体设计 6](#_Toc36914262)

[3.1.1 系统总体架构设计技术可行性 6](#_Toc36914263)

[3.1.2 系统部署方案设计 7](#_Toc36914264)

[3.1.3 系统功能结构设计 7](#_Toc36914265)

[3.3数据库设计 8](#_Toc36914266)

[3.3.1数据库E-R图设计 8](#_Toc36914267)

[3.3.2数据库物理结构设计 9](#_Toc36914268)

[4. 项目回款系统的实现 12](#_Toc36914269)

[4.1系统页面实现 12](#_Toc36914270)

[4.2用户登录模块实现 13](#_Toc36914271)

[4.3客户页面 14](#_Toc36914272)

[4.3项目页面 18](#_Toc36914273)

[4.3数据统计页面 21](#_Toc36914274)

[4.5系统框架配置 22](#_Toc36914275)

[4.6数据库表创建 24](#_Toc36914276)

[5. 系统测试与运行 25](#_Toc36914277)

[5.1测试环境介绍 25](#_Toc36914278)

[5.1登录界面测试 26](#_Toc36914279)

[5.2项目测试 26](#_Toc36914280)

[5.2.1 创建项目测试 26](#_Toc36914281)

[5.2.2 项目删除测试 27](#_Toc36914282)

[5.2.3 创建项目回款任务测试 27](#_Toc36914283)

[5.2.4 项目回款任务状态变更测试 28](#_Toc36914284)

[5.2.5 客户新增测试 29](#_Toc36914285)

[5.2.6 统计测试 29](#_Toc36914286)

[6. 结束语 30](#_Toc36914287)

[参考文献 31](#_Toc36914288)

[致 谢 32](#_Toc36914289)

# 1. 绪 论

## 1.1 背景介绍

当今世界，随着“互联网+”经济的兴起，计算机技术不断创新，不断迭代，已然日新月异，服务被广泛应用，采用计算机来管理我们的信息已经是当今社会发展的趋势，而网络信息自动化管理也逐渐取代传统表格制作的记账方式。手工记账会存在诸多缺陷，比如：忘记记账、资料丢失、信息维护困难等等。因此很多小型创业团队、工作室、自由职业者等人群希望能借助管理系统来减轻零碎的工作。基于Golang的项目回款管理小程序就是针对这个场景设计的，它采用目前B/S架构技术，操作简单，可维护性高。微信小程序(WeChat Mini Program)于 2017 年 1 月 9 日正式发布上线，是由腾讯公司开发的一 款嵌于移动端社交软件微信内部的应用，使用者可以在微信里直接搜小程序名字，或者扫描二维码就可以找到对应的小程序应用，不需要额外安装 App，在微信内部使用完小程序退出即可，微信小程序最大的亮点就是无需安装，用完即走，是一种去客户端化的轻量级应用。并且微信小程序对C端用户的隐私非常重视，授权确认减少用户信息曝露，强制使用 HTTPS 安全链接，通过加密TCP通信，确保将数据加密后完整的正确的发送到后端服务器，可以有效防止被窃取并修改发送的数据，不需要担心用户数据被更改。

## 1.2 研究意义

项目回款管理系统可以很方便对每一笔回款信息进行管理，该系统可以减少账目管理员的工作量以及 所以此课题将以团队回款为使用场景，以微信小程序为载体，基于云端模式，设计并开发出一款适合小型创业团队、工作室、自由职业者等人群使用的辅助工具，方便团队管理者轻松管理账目。

## 1.3 使用开发技术

基于golang的项目回款管理小程序的设计与实现采用了B/S 模式进行设计。使用Golang语言构建后端开发环境，使用微信小程序开发者工具搭建前端开发环境。后台接口采用MVC框架设计，MySQL数据库进行提供数据支持，前端采用 AJAX技术+微信小程序开发客户端界面。

### 1.3.1 JetBrains Goland平台特点

GoLand 2020.1是JetBrains 2020年最新迭代研发的商业开发软件。 Goland是一款复核人体工程学的新型IDE。这款IDE内置了Golang的代码编写辅助工具。

### 1.3.2 Golang

Golang是一种新晋热门语言，有C基础，学Golang非常简单轻松，同步发展方式可以轻松进行实现高并发，代码设计简洁，格式统一，阅读教学方便，性能强劲的同时，开发工作效率又不差于Python等动态管理语言。从工程的角度，对于大多数应用场景的背景下，选择Golang是非常明智的选择。

### 1.3.3 xorm 和Mysql

MySQL是优秀的关系型数据库，xorm是一个优秀的操作MySQL等数据库的ORM库. 通过xorm可以很方便的操作Mysql数据库。并且可以让Struct与SQL互相映射，并可以通过sync2函数开启进行自动同步; 支持调用MySQL数据库方法自动生成xorm的Struct；并且通过EngineGroup可以支持读写分离和负载均衡。

### 1.3.4 微信小程序

微信小程序是市场已知小程序，具有无需安装、跨平台、不占内存、入口多，流量触手可达等几种有点，具有APP的优点，也具有网页传播快的优势。是一种新型的易传播的Native APP。

### 1.3.5 MPVUE框架

MPVUE框架是最近热门的基于 Vue.js的小程序框架。MPVUE 修改了vue的事件代理机制和数据同步机制，vue.js中 数据变更后，最终编译成小程序代码。

## 1.4 微信小程序开发流程

### 1.4.1 获取微信小程序AppID

注册小程序账号并登录微信公众平台，在菜单 “开发”-“开发设置”-“开发者ID” 看到小程序的 AppID(小程序ID)，根据指引填写信息和提交相应的小程序基本资料。

### 1.4.2 安装微信开发者工具

打开微信的开发者工具下载页面 ，下载MAC版本安装包，并下载完成后进行安装。

### 1.4.3 创建项目

这里我们采用MPVUE CLI来创建项目，选择好目录运行CLI 命令生成基础项目，并填入申请好的AppID等相关参数，并打开MAC端开发者工具，打开并导入上述项目文件目录，这样项目就创建好了，在项目目录运行yarn dev命令就能编译并运行项目了，此时可以再开发者工具中预览刚刚初始化的小程序。

### 1.4.3 编写代码

这里我们采用MPVUE来创建项目，并编译成复核小程序规范的代码，小程序主体仍然是app.js文件，页面在src/pages目录创建。

### 1.4.4 创建页面

小程序的每一个页面都在src/pages里创建，但是src/router.js中新建路由以及页面配置信息。这样才能在生成预览页面。MPVUE会把.vue文件编译成4个文件，分别对应小程序的4种类型源文件

### 1.4.5 预览页面

在项目目录运行yarn dev命令，把源码编译成小程序原生源码，点击开发者工具的预览，会生成有效期15分钟的预览二维码，用测试手机扫一扫即可预览启动的项目。

# 2. 需求分析

## 2.1可行性分析

项目回款系统的开发离不开可行性分析，可行性分析包含需求分析，经济分析、技术分析等。可行性分析是鉴定是否可以用最小的成本尽快解决问题，避免因为技术问题、经济问题或者社会问题导致开发不能继续，造成资源浪费。在分析过程中，对项目回款系统在开发中遇到的一系列问题进行前置预估，分析出项目回款系统小程序在在开发之前需要的开发环境，在开发中遇到的难点，及时制定相关策略，避开疑难杂症，保障项目顺利上线。本章节通过对技术、经济、社会等进行多方位可靠的分析来阐述开发本系统的意义。

### 2.1.1经济可行性

经济可行性分析是通过分析本系统开发前的成本以及项目上线之后的收益进行预估，只要预期收益高于研发成本，则认为本系统风险低，收益高。

本项目回款系统小程序，采用小程序作为平台载体，抛弃原生APP，这样的选择很大的优势，不但节省了多平台的研发成本，更接住了微信APP引流的优势，可以更快更省力的吸引用户入住。并且使用了MPVUE进一步降低小程序研发的平台壁垒，让研发成本进一步降低，然而小程序的优势缺依然保留。

经过上述分析可得本系统开发成本与未来收益对比，收益远大于研发成本，因此经济上也是可行的。

2.1.2技术可行性

如何利用手头已有的设备、软件和技术人员，并达成目标，也是需要考虑的问题。

以开发时间成本的角度分析，本项目回款系统采用微信小程序为开发平台，并采用MPVUE作为跨平台开发方案，解决人力成本高，并解决小程序跨平台开发方案，一次开发，多端通用，节省大量时间。并且能利用微信APP提供的API实现调用原生功能。

Golang是一种新晋热门语言，也是当今流行的十大语言之一。可以轻松进行实现高并发，代码设计简洁，格式统一，阅读教学方便，性能强劲的同时，开发工作效率又不差于Python等动态管理语言。从工程的角度，对于大多数应用场景的背景下，选择Golang是非常明智的选择。

免费并且强大的开源的MySQL数据库也是作为数据存储的首选。开源的Nginx服务器也是部署系统的首选。

因此认为技术上也是可行的。

### 2.1.3社会可行性

随着计算机网络技术的发展，采用计算机来管理我们的账单已经逐渐取代人工表格管理，所以开发本回款小程序系统的目标不仅可以提高工作效率，减轻记账的繁琐，还可以提高准确率。长远来说具有非凡的意义。

最后，从小程序出现至今，已经超过10亿日活，已经成为互联网新型的流量蓝海，小程序的商业、用户、流量等价值不言而喻，维信小程序已经成为社交电商的新霸主，依托微信已经诞生了多种交互场景.

因为对项目回款系统在经济可行性、技术可行性、社会可行性三方面进行了分析，综合评估认为此系统的可行性较高，可以正式投入生产使用。

# **3. 项目回款系统设计**

## 3.1系统总体设计

系统设计是对本项目回款小程序从前端到后端到运维的架构概念图。将本项目回款系统的架构进行设计，并针对运维和系统功能进行设计。系统架构图是支撑系统开发的基础。

### 3.1.1 系统总体架构设计技术可行性

根据项目回款管理小程序的使用场景、业务流程以及项目架构、服务器架构，本项目回款系统的总体架构如图3.1所示。

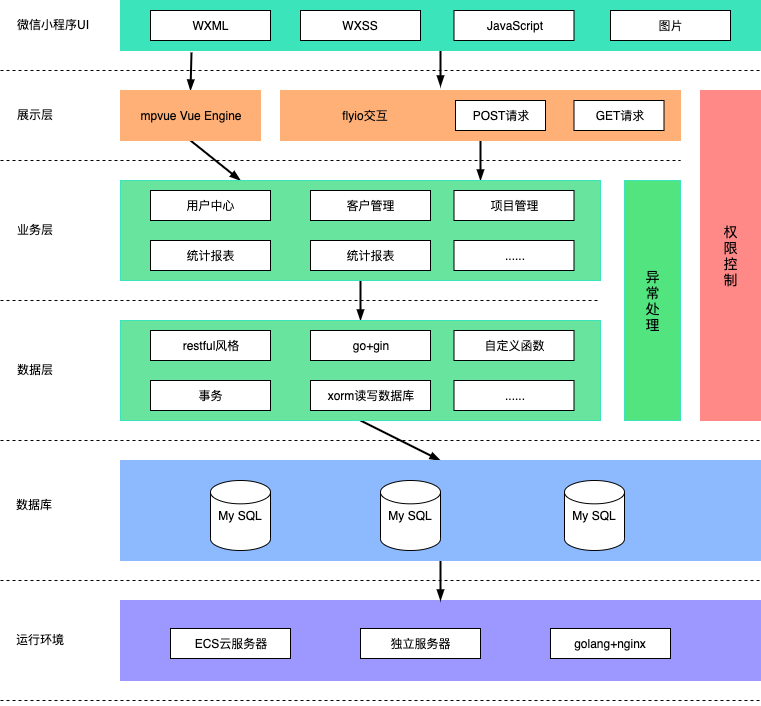


图3.1 系统架构图

如图所示，本项目回款系统架构为MPVUE通过flyio将请求发送给nginx负载均衡服务器，通过nginx权重分配到几台部署了服务API的服务器上，各服务器分别根据JSON Web Token中间件进行TOKEN鉴权，Gin Web服务器 会根据MySQL数据对请求进行处理。

### 3.1.2 系统部署方案设计

项目回款系统一共使用2 台华为云服务器进行部署。华为云服务器可以利用免费SSL证书部署https,与微信开发者工具打通。数据库采用自建数据库，并使用主从分离机制。系统为了实现高并发、高可用，使用用nginx 负载均衡权重分配方案，让请求根据服务器配置性能的权重来分配到2台华为云服务器上，这样可以最大化利用服务器的性能， 并使用Redis进行数据缓存处理。

### 3.1.3 系统功能结构设计

本项目回款系统包含四大功能，如下：项目模块、客户模块、报表模块、用户模块。

针对用户登陆进行登录校验，登录成功后分发TOKEN， 后续所有接口采用JWT校验TOKEN有效性，假设登录失败，会弹出提示信息并停留在登录页，用户仍然需要重新登录。用户登录流程图如图3.2所示。

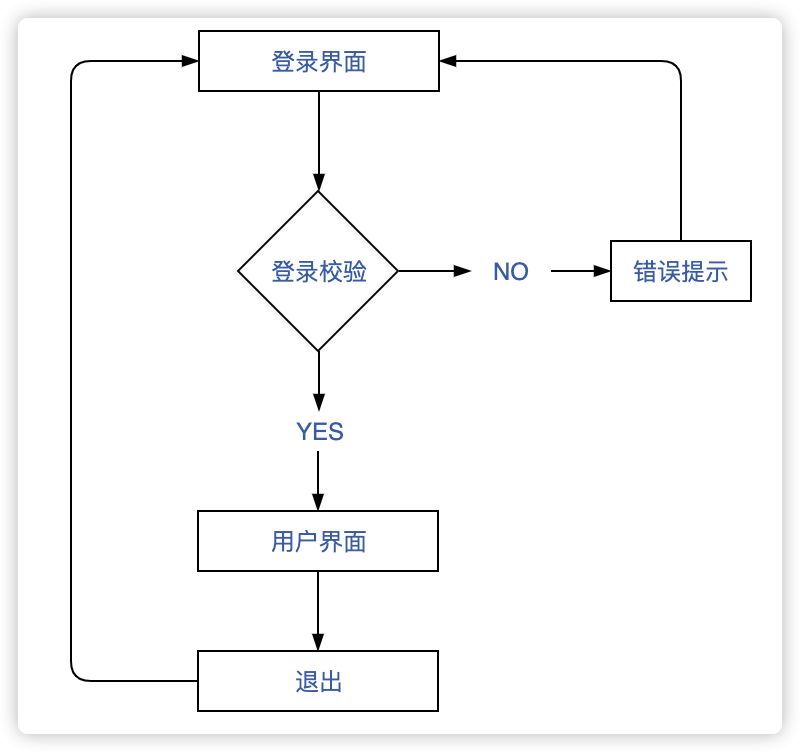


图3.2 用户登录流程图

## 3.3数据库设计

数据库采用MySQL 和Redis 对项目回款系统里面的数据进行存储，是整个系统的重中之重，数据库设计的健壮性和可靠性，对系统起着至关重要的作用，他不仅把不同数据进行关联，也把数据进行缓存。

本项目回款系统采用MySQL数据库进行服务器端数据的管理，MySQL 优势大，性能块，有标准的语法规范，并且加密传输，是一个优秀的数据库管理工具。使用Redis进行数据缓存，让服务高可用。

### 3.3.1数据库E-R图设计

对于原型系统的分析，本项目回款工具决定使用E-R图用来建立数据模型，本项目回款系统设计了如下几张数据库表：用户表、客户表、项目表、项目回款任务表，并设计各自的属性以及关联。本项目回款系统E-R图如3.3所示。

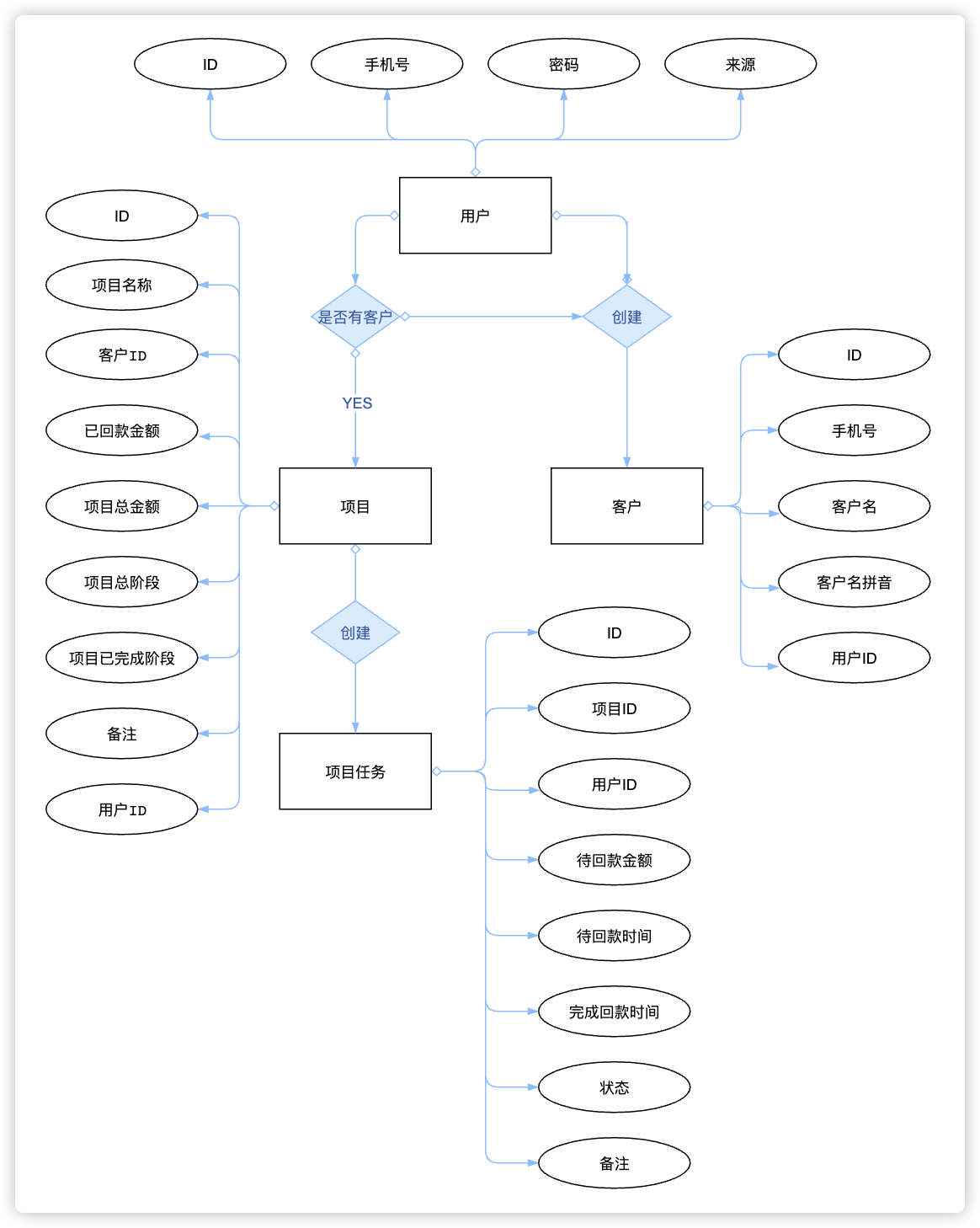


图3-3 系统E-R图

### 3.3.2数据库物理结构设计

物理结构设计就是把在数据库概念设计阶段产生的实体—联系图(E-R图)进一步转成与数据结构相符合的物理逻辑结构。在此次毕业设计中，我采用的是MySQL5.7版本数据库进行数据的存储，本项目回款系统数据库表设计结构如下所示。

将上一节中设计的E-R图转换为MySQL支持的物理逻辑结构，以供数据模型检查。数据库名称为kuan.db,包含4个表：用户表User、客户表Customer、项目表Project、项目任务表ProjectTask。

1. 用户表（User）

用户表的设计尤为重要，如果设计的不好后期可能会返工，导致工期延长，所以经过思考，定下如下字段：unionid（针对 QQ / 微信等）、手机号、密码、姓名等字段

表3-1 用户表（User）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段类型 | 为空 | 字段注释 | 备注 |
| id | int(10) | 否 | 索引 | 主键、唯一 |
| real\_name | varchar(64) | 否 | 姓名 |  |
| mobile | varchar(11) | 否 | 手机号 |  |
| password | varchar(255) | 否 | 密码 |  |
| unionid | varchar(64) | 是 | unionid |  |
| token | varchar(255) | 否 | Token |  |

1. 客户表（Customer）

客户表用于记录使用本系统客户的姓名、手机号等信息。如表3.2所示。该表主要包括id、姓名、来源等相关字段。

表3-2 客户表（Customer）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段类型 | 为空 | 字段注释 | 备注 |
| id | int(10) | 否 | 索引 | 主键、唯一 |
| real\_name | varchar(64) | 否 | 姓名 |  |
| mobile | varchar(11) | 否 | 手机号 |  |
| pin\_yin | varchar(255) | 否 | 客户名拼音 |  |
| source | varchar(10) | 否 | 来源 |  |
| user\_id | int(10) | 否 | 用户ID |  |

1. 项目表（Project）

项目表于记录使用本系统项目的回款金额、回款阶段等信息。如表3.3所示。该表主要包括id、用户id、客户id等相关字段。

表3-3 项目表（Project）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段类型 | 为空 | 字段注释 | 备注 |
| id | int(10) | 否 | 索引 | 主键、唯一 |
| Name | varchar(64) | 否 | 姓名 |  |
| bad\_amount | bigint(20) | 否 | 项目坏账 |  |
| done\_amount | bigint(20) | 否 | 已回款金额 |  |
| total\_amount | bigint(20) | 否 | 项目总金额 |  |
| total\_task | int(10) | 否 | 项目总阶段 |  |
| done\_task | int(10) | 否 | 项目已完成阶段 |  |
| remark | varchar(64) | 是 | 备注 |  |
| user\_id | int(10) | 否 | 用户ID |  |
| customer\_id | int(10) | 否 | 客户ID |  |

1. 项目任务计划表（ProjectTask）

项目任务计划表于记录使用本系统项目的回款金额、回款阶段等信息。如表3.4所示。该表主要包括id、用户id、客户id等相关字段。

表3-4项目任务计划表（ProjectTask）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段类型 | 为空 | 字段注释 | 备注 |
| id | int(10) | 否 | 索引 | 主键、唯一 |
| Name | varchar(64) | 否 | 姓名 |  |
| amount | bigint(20) | 否 | 待回款金额 |  |
| waiting\_date | varchar(10) | 否 | 待回款时间 |  |
| done\_date | varchar(255) | 否 | 完成回款时间 |  |
| status | int(1) | 否 | 状态 |  |
| remark | varchar(64) | 是 | 备注 |  |
| user\_id | int(10) | 否 | 用户ID |  |
| project\_id | int(10) | 否 | 项目ID |  |

# 4. 项目回款系统的实现

## 4.1系统页面实现

本项目回款系统前端开发使用MPVUE框架，在src/router.js中新建路由以及页面配置信息，并在src/pages/中新建对应的文件。pages/project/project-list项目列表页面，pages/project/project-detail项目详情页面，pages/project/project-add创建项目页面，pages/project/project-task-list项目任务列表页面，pages/project/project-task-add创建项目任务页面，pages/customer/customer-list客户列表页面，pages/customer/customer-add创建客户页面，pages/statistics/index统计页面，pages/user/index我的页面，pages/user/login登录页面，pages/user/reg注册页面。页面建立如图4.1所示。

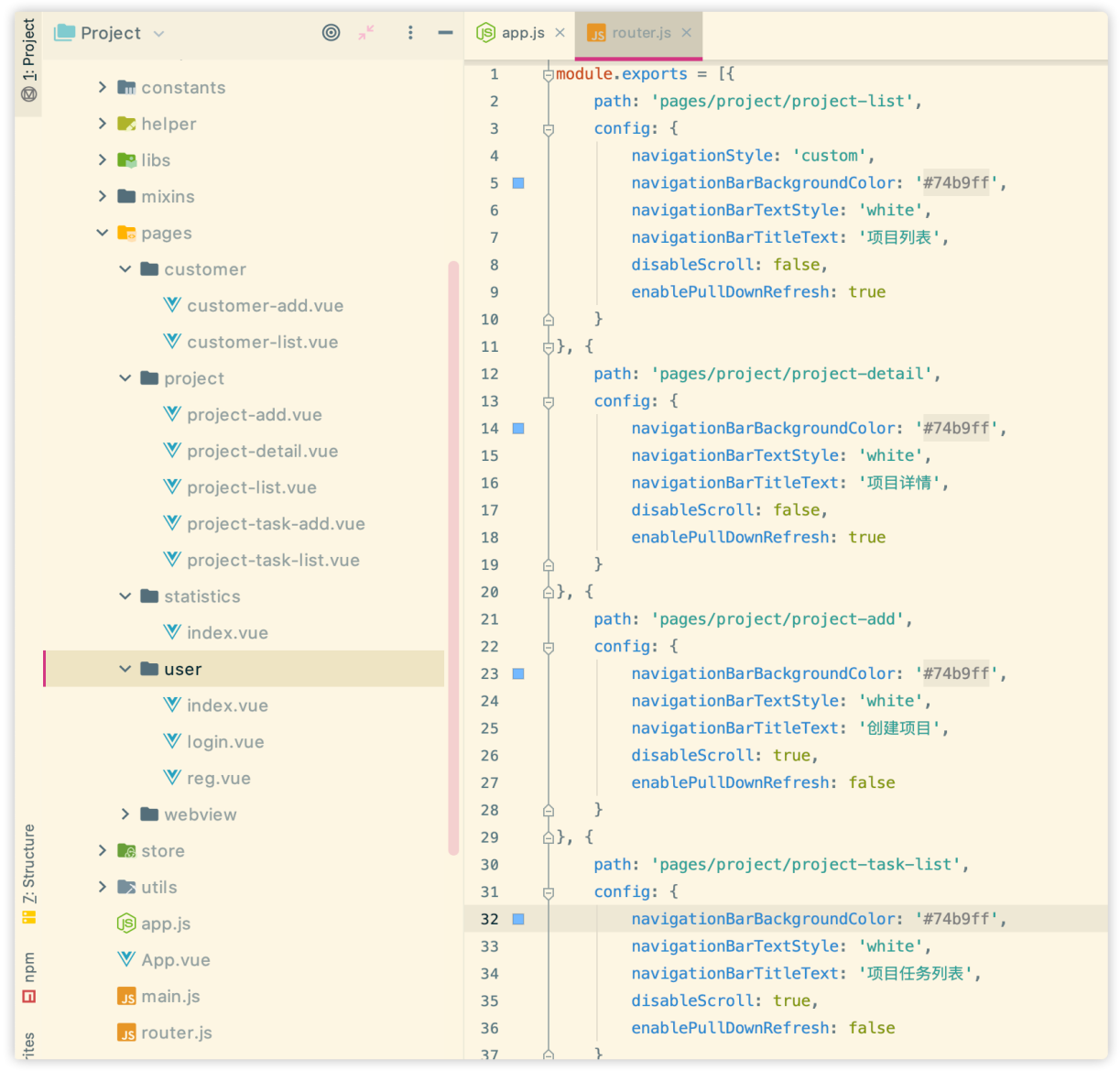


图4-1 页面建立如图

## 4.2用户登录模块实现

功能描述：登录系统看到的第一个界面就是登录界面如下图所示，用户输入手机号码和密码，然后单击登录按钮，如果登录失败则会重新回到登录界面并会提示“手机号或者密码错误”信息，登录成功会跳转到项目页面。



图4-2登录界面

主要算法思想：登录检验主要是用到代码查询，用户输入手机号码和密码后，调用接口进行手机号以及密码加密后查询数据库内是否有匹配用户，校验成功则生成TOKEN 并返回用户信息。

部分代码：

func (uc UserController) Login(c \*gin.Context) {

mobile := c.PostForm("mobile")

if utils.IsEmpty(mobile) {

SendResponseError(c, ecode.Error, fmt.Errorf("请输入手机号"))

return

}

password := c.PostForm("password")

if utils.IsEmpty(password) {

SendResponseError(c, ecode.Error, fmt.Errorf("请输入密码"))

return

}

item, err := UserService.GetByMobile(mobile)

if err != nil {

SendResponseError(c, ecode.Error, err)

return

}

if pwdMd5 := utils.MD5([]byte(password)); pwdMd5 != item.Password {

SendResponseError(c, ecode.Error, fmt.Errorf("密码不正确"))

return

}

utils.Int642Str(time.Now().Unix()))

token, err := jwt.GenerateToken(item.Mobile, item.Password)

if err != nil {

SendResponseError(c, ecode.Error, fmt.Errorf("token生成失败"))

return

}

item.Token = token

UserService.UpdateToken(item.Id, item.Token)

SendResponse(c, item)

}

## 4.3客户页面

功能描述：用户创建项目的前置条件就是需要创建客户信息。客户模块是整个项目回款系统中的尤为关键的部分。客户模块包括客户姓名、客户来源、客户渠道、手机号等可追溯条件。



图4-3 客户页面

主要算法思想：用户登录回款系统后可以查询有哪些客户，可以查看客户入了多少款项。

部分代码：

//客户添加接口

func (this CustomerController) CreateCustomer(c \*gin.Context) {

type CustomerFormData struct {

//真实姓名

RealName string `json:"realName" form:"realName" validate:"required"`

//手机号

Mobile string `json:"mobile" form:"mobile"`

//邮箱

Email string `json:"email" form:"email"`

//性别

Gender string `json:"gender" form:"gender"`

//生日

Birthday string `json:"birthday" form:"birthday"`

//客户来源

Source string `json:"source" form:"source"`

//客户渠道

Channel string `json:"channel" form:"channel"`

// 创建人ID

UserId int64 `json:"createUserId" form:"createUserId" validate:"required"`

//备注

Remark string `json:"remark" form:"remark"`

}

formData := new(CustomerFormData)

if err := ValidateRequest(c, formData); err != nil {

SendResponseError(c, ecode.Error, err)

return

}

pinyin := utils.Hans2Pinyin(formData.RealName)

item := models.Customer{

RealName: formData.RealName,

PinYin: formData.RealName,

Mobile: formData.Mobile,

Email: formData.Email,

Gender: formData.Gender,

Birthday: formData.Birthday,

Source: formData.Source,

Channel: formData.Channel,

UserId: formData.UserId,

Remark: formData.Remark,

Block: "0",

}

if pinyin != "" {

item.PinYin = pinyin

}

itemData, err := CustomerService.CreateCustomer(item)

if err != nil {

SendResponseError(c, ecode.Error, err)

return

}

SendResponse(c, itemData)

}

// 客户列表查询接口

func (this CustomerController) GetCustomerList(c \*gin.Context) {

var ListPage common.ListPage

page, \_ := strconv.Atoi(c.DefaultPostForm("page", "1"))

pageSize, \_ := strconv.Atoi(c.DefaultPostForm("pageSize", "2"))

ListPage.Page = page

ListPage.PageSize = pageSize

listData, total, err := CustomerService.GetCustomerList(ListPage)

if err != nil {

SendResponseError(c, ecode.Error, err)

return

}

ListPage.Total = total

ListPage.TotalPage = ListPage.Total / ListPage.PageSize

SendResponse(c, gin.H{

"result": listData,

"pagination": ListPage,

})

}

## 4.3项目页面

功能描述：管理您需要完成的项目，添加项目下的回款任务，可以统计当前项目已回款金额、待回款金额、坏账金额。每个项目可包含多个回款任务，回款任务由金额、预计回款时间组成，在项目回款系统中可以自由的更改每一条回款任务的任务状态：待回款、已回款、坏账

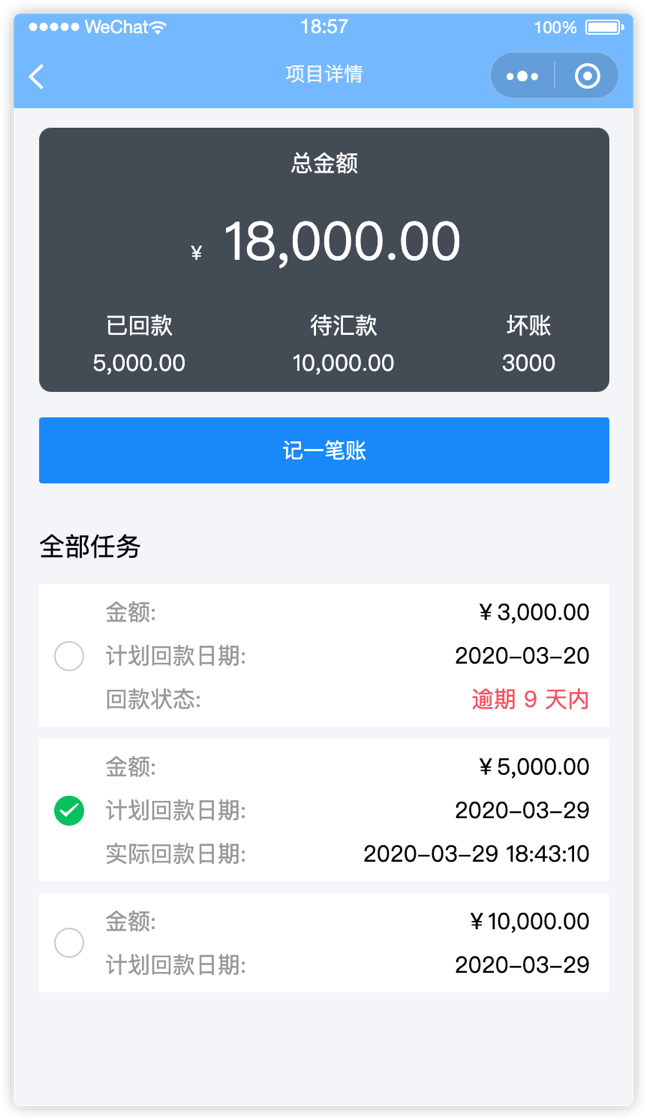




图4-3 项目页面

主要算法思想：根据每一条项目任务计算项目回款阶段，项目回款金额、待回款金额、坏账金额。针对不同状态做颜色异常处理，并红色提示坏账。

//项目详情

func (this ProjectController) GetProjectById(c \*gin.Context) {

formData := new(ProjectFormDataById)

if err := ValidateRequest(c, formData); err != nil {

SendResponseError(c, ecode.Error, err)

return

}

// 校验参数

item := models.Project{

Id: formData.Id,

}

itemData, err := ProjectService.GetProjectById(item.Id)

if err != nil {

//panic(err)

SendResponseError(c, ecode.Error, err)

return

}

SendResponse(c, itemData)

}

// 更新总回款金额、待回款金额、待回款金额

func (this \*ProjectService) UpdateProjectAmount(ProjectId int64) error {

if ProjectId == 0 {

return errors.New("ProjectId 为空")

}

taskList, total, err := this.ProjectTaskDao.GetProjectTaskListByProjectIdNoPage(ProjectId)

logger.Debugf("taskList", taskList)

if err != nil {

return err

}

projectItem := &models.Project{

Id: ProjectId,

//已回款金额

DoneAmount: 0,

//项目总金额

TotalAmount: 0,

//项目总金额

BadAmount: 0,

//项目总阶段

TotalTask: total,

//项目已完成阶段

DoneTask: 0,

}

for \_, value := range taskList {

projectItem.TotalAmount = projectItem.TotalAmount + value.Amount

if value.Status == "1" {

projectItem.DoneAmount = projectItem.DoneAmount + value.Amount

projectItem.DoneTask = projectItem.DoneTask + 1

} else {

// 当前时间戳 > 预定时间 + ' 23:59:59'

taskTime, \_ := time.Parse("2006-01-02 15:04:05", value.WaitingDate+" 23:59:59")

logger.Debugf("projectItem.taskTime", value.WaitingDate, taskTime)

if time.Now().UnixNano() > taskTime.UnixNano() {

projectItem.BadAmount = projectItem.BadAmount + value.Amount

}

}

}

logger.Debugf("projectItem.Id", projectItem.BadAmount)

err = this.ProjectDao.UpdateProjectAmount(\*projectItem)

if err != nil {

return err

}

return nil

}

## 4.3数据统计页面

功能描述：通过时间段筛选，可以统计出此时间段内实际回款、待收回款、坏账，以及图表展示实际回款中的每个客户、项目占比



图4-3 客户页面

主要算法思想：用户登录回款系统后可以查询实际回款中各项目占比，以及实际回款中个客户占比。

## 4.5系统框架配置

func main() {

// 注册日志

logger.Register(&logger.ZapConfig{

Path: "./log/zap",

Level: "debug",

TimeUnit: logger.Day,

MaxFileSize: 1,

MaxBackups: 7,

MaxAge: 1,

Compress: true,

Caller: true,

JsonFormat: false,

LogInConsole: true,

})

cron.Register()

config.Register()

// 注册redis

err1 := redis.Instance().Register(&config.RedisConfig{

Host: viper.GetString("redis.host"),

Password: viper.GetString("redis.password"),

Port: viper.GetString("redis.port"),

DB: viper.GetInt("redis.db"),

})

if err1 != nil {

logger.Error("err", err1)

}

// 注册 db

dao.Register(&config.DatabaseConfig{

Host: viper.GetString("database.host"),

User: viper.GetString("database.user"),

Password: viper.GetString("database.password"),

Database: viper.GetString("database.database"),

Port: viper.GetString("database.port"),

Charset: viper.GetString("database.charset"),

})

defer dao.Close()

// 启动服务

err := startServer(&config.ServerConfig{

Mode: viper.GetString("RUN\_MODE"),

Port: viper.GetString("server.port"),

})

if err != nil {

logger.Error("Start server: %+v", err)

}

}

## 4.6数据库表创建

// 使用xorm工具同步数据库表字段

func dbMigrateSync() {

err := x.Sync2(

new(models.Customer),

new(models.Project),

new(models.ProjectTask),

new(models.User),

new(models.Options),

)

if err != nil {

logger.Warnf("Fail to sync2 database: %v\n", err)

}

}

# 5. 系统测试与运行

测试是保障功能开发后无BUG上线至关重要的一步，测试可以及时的发现问题，纠正问题，并且回归问题本身，并梳理清楚各模块之间的逻辑是否正确，保障功能完善。测试分为单元测试、集成测试等，下面我们进行的就是单元测试。通过测验系统内最小单位功能，来保障系统的完善性。

## 5.1测试环境介绍

本项目回款系统基于小程序平台开发，因此要充分测试各平台以及各系统版本的兼容性，固准备如下2台机器用来测试。详细的手机硬件信息、服务器系统信息如下所示：

苹果系统:

* 设备名称：苹果
* 设备型号：iphone11
* iOS版本：iOS13.4
* CPU：A13 + 第三代神经网络引擎
* RAM：4 GB
* 微信版本：7.0.12

安卓系统：

设备名称:小米6

* 设备型号：MI 6
* 系统版本：MIUI 11.0.2
* Android版本：9
* CPU：高通骁龙835 八核 2.5GMHz
* RAM：6 GB
* 微信版本：7.0.12

服务器信息:

* CPU：Intel(R) Xeon(R) Gold 6161 CPU @ 2.20GHz
* 系统：CentOS Linux release 7.7.1908 (Core)
* 内存：4GB
* 内网带宽：1Gbps
* 公网带宽：1M

## 5.1登录界面测试

当用户输入账号密码后，系统会对账户密码进行校验，如果成功则进去主页，失败则根据失败类型进行提示并停留在登录页面。

图5-1 登录失败提示

## 5.2项目测试

### 5.2.1 创建项目测试

用户可以创建自己想要的项目，并且维护改项目，同时生成一条回款任务，切任务为待回款状态。如图5-2所示。

图5-2 创建项目

### 5.2.2 项目删除测试

当项目已经废弃或者重复创建，可以对回款项目进行删除，会对用户进行提示操作。如图5-3所示。



图5-3 项目删除测试

### 5.2.3 创建项目回款任务测试

用户在该项目下，可以新增项目回款任务，新增后项目总金额，已回款金额，待回款金额，会发生相应的变更。如图5-4所示。

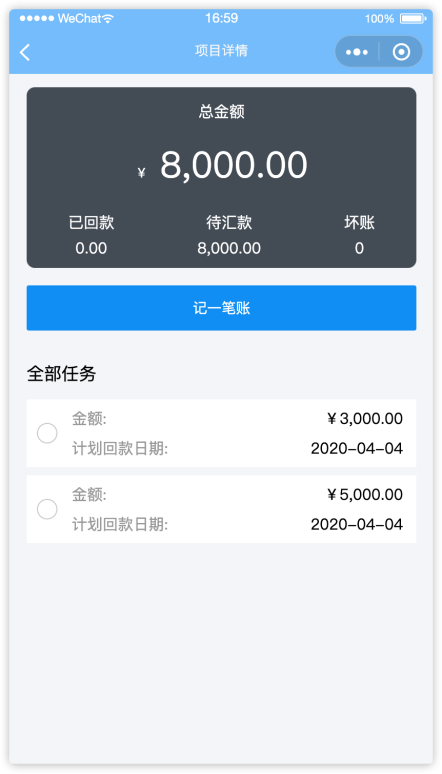
  

图5-4 创建回款任务

### 5.2.4 项目回款任务状态变更测试

用户变更项目回款任务状态，随后项目总金额，已回款金额，待回款金额，会发生相应的变更。如图5-5所示。



图5-5 项目回款任务状态变更测试

### 5.2.5 客户新增测试

用户新增后，在列表呈现。并可以对无用客户进行删除。如图5-6所示。

图5-6 用户修改项目回款任务状态

### 5.2.6 统计测试

用户项目汇款后，会在统计模块进行汇总统计，有如下2种统计：1.实际回款中各项目占比2.实际回款中个客户占比。如图5-7所示。

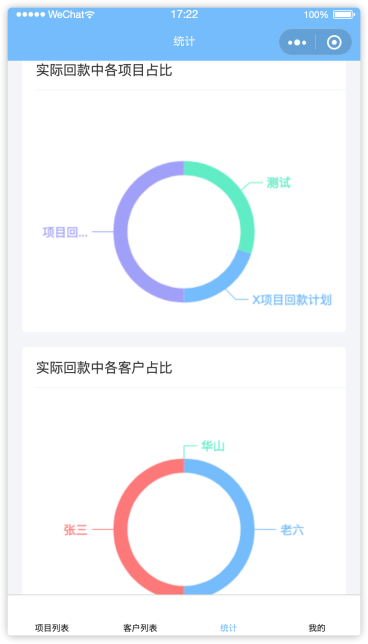


图5-7 用户修改项目回款任务状态

# 6. 结束语

本系统借助当下互联网+的热潮，利用具有社交统治地位的微信小程序作为载体，以小团队记账为使用场景，为小团队提供项目回款服务，基于上文的研究分析与设计，本系统已经按照设计基本实现了各模块功能，并充分进行了系统测试。在完成项目的期间，增强了自己的分析能力，巩固了专业知识，收获颇丰，但是因为时间关系，本系统仍然有需要改进的余地。

互联网信息化是在这个时代十分提倡的一种信息化记账方式，在互联网+的热潮下，利用小程序作为载体给小型创业团队、工作室、自由职业者等人群带来便利。

本系统对小型团队为使用场景，为小型创业团队、工作室、自由职业者设计并实现了一款项目回款管理辅助小程序。随着智能手机的发展，以小程序作为载体，无疑是是最适合人们的工具，可以在任何地方、任何时间自由的添加项目回款以及项目回款任务。项目回款系统中前端使用 MPVUE 框架并编译为小程序源码，后端使用基于Go的Gin+xorm的MVC框架，数据持久化工具使用MySQL和Redis，设计本系统所包含的所有数据字段，通过自主购入2台华为云服务器实现数据云端服务器的负载均衡。本文在充分阅读微信小程序的在线文档的前提下，结合小团队的记账现状，通过用户需求分析，流程图、时序图完成系统的整体设计，依次通过项目基础背调、需求可行性分析、技术可行性分析、完整项目的设计与实现以及项目整体流程测试，进一步强化了自己的能力。

在这次项目回款系统开发中我遇到如下的问题：

1. 界面不够美观
2. 因时间有限，没有开发出多人共同维护一个回款项目的功能

# 参考文献

[1] 王珊. 数据库系统概论（第四版）[M].高等教育出版社. 2016.

[2] 艾伦 A. A. 多诺万. Go程序设计语言. 机械工业出版社. 2017.

[3] 黄梯云.管理信息系统（第三版）[M].高等教育出版社，2004

[4] 胡学钢，王浩.计算机科学与技术专业软件系列课程实践教程[M].合肥工业大学出版社，2003

[5] Baron Schwartz. 高性能 MySQL（第 3 版）电子工业出版社出版. 2013.

[6] 段华斌. [基于工作流的毕业设计过程管理系统的设计与实现](http://www.cnki.net/kcms/detail/detail.aspx?filename=2009237864.nh&dbcode=CMFD&dbname=CMFD2010&v=)[D]. 中南大学. 2008.

[7] 张海藩. 软件工程导论第四版[M].北京:清华大学出版社，2003.

[8] 王珊,萨师煊.数据库系统概论（第四版）[M].北京:高等教育出版社,2006. [5]

[9] 恒兆文化编辑部.家庭记账簿[M].机械工业出版社，2009

# 致 谢

在这一段时间里，通过对本系统和论文的编写让我学到了很多知识，同时也让我把大学的课程串联到一起。在这段时间如果没有老师、朋友的帮助，相信完成这篇论文，还是非常困难的，在此我要感谢我的老师和朋友们。

首先，我要衷心感谢我的导师谷雨老师，老师在此次论文以及毕业设计中给了我非常大的帮助，在毕业设计和论文的编写中给与多次指导，帮我解答疑难，正因为有了老师的教导和指引，才没有让我的论文偏题，并顺利完成。借此机会，向谷雨老师表示诚挚的谢意。

此外还要感谢吕增威老师，是老师给我锻炼的机会，帮我解答疑难，给我指明方向。在本科这几年生活中，吕老师给我学习上的教导和指引，为我学习和为人处事方面带来了意义深远的影响。在这里我非常感谢吕增威老师。

我还要感谢我的同学，是他们陪我度过了大学这段美好时光。是他们在我遇到不懂的问题时，能互相探讨，相互请教。时光如梭，眨眼间我们即将毕业，感谢有你们的时光。时光荏苒，遇见你们真好。祝福你们再将来的日子里继续前进。

最后感谢在百忙之中抽出宝贵的时间参与此论文评审的专家和老师们。