基于Go实现的后端租房评价系统

摘要：本论文主要设计的租房评价系统，在单纯的租赁交易中加入承租者与房东的互相评价与承租者对所租住房屋的评价，实现了为后来的承租者提供更多有效的参考，而且其信息参考价值与房屋所租赁次数的增加呈正相关。系统采用的实现语言是Go语言，数据库采用的是PostgreSQL，在框架的选择上采用的是业界最流行的开源框架，主要有处理Web请求和路由管理的Gin框架，处理数据库操作的GORM框架，权限管理的Casbin框架以及打印日志的Logrus框架。

关键词：租房；评价系统；Go；PostgreSQL；GORM

**Backend rental evaluation system based on Go**

**Abstract:** In this paper, the main design of the rental evaluation system, in a simple rental transaction to add the mutual evaluation of tenants and landlords and the evaluation of tenants on the rented housing, to achieve the later tenants to provide more effective reference, and its information reference value is positively correlated with the increase of the number of rental housing. The implementation language adopted by the system is Go language, the database adopts PostgreSQL, and the most popular open source framework in the industry is adopted in the choice of framework. There are mainly Gin framework for processing web requests and routing management, GORM framework for processing database operations, permissions Casbin framework for management and Logrus framework for printing logs.

**Key words:** Renting; Evaluation System; Go; PostgreSQL; GORM

用于解析和下载项目依赖的文件，完成项目的初始化工作[12]。项目其他目录的说明，如表3.1所示。

表3.1 目录结构说明

|  |  |
| --- | --- |
| 目录名称 | 解释 |
| api | 用于处理相关请求的方法 |
| conf | 项目配置 |
| modules | 模型定义 |
| router | 定义请求的路由规则 |
| service | 复杂逻辑处理 |
| test | 测试代码 |
| util | 工具文件，定义经常调用的处理方法 |

## 3.2 程序启动流程

首先要在CentOS系统中启动基础的服务，基础服务包括Nginx服务、Redis服务和PostgreSQL服务，然后再去启动项目程序，如图3.2所示。

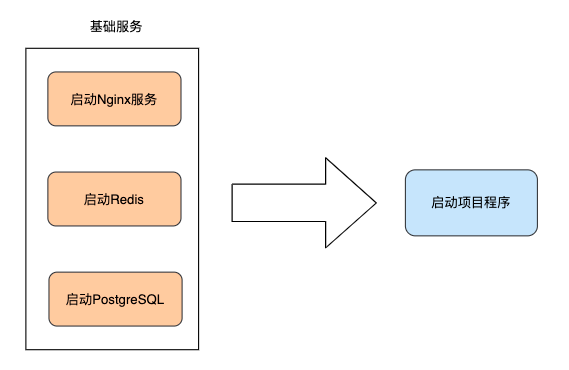


图3.2 程序启动流程

项目程序在启动时首先加载配置文件，配置文件定义的配置信息会和程序中默认的配置信息做对比，配置文件中缺少的配置信息以程序中定义的为准。配置

文件加载完成后，会根据配置文件创建数据库连接池和Redis连接池，然后通过一个for循环检查连接创建情况[13]，如果创建失败，则根据配置文件中指定的

在入住前期很难察觉，等到入住后如果接受不了，那往往都是承租者承担毁约造成的损失，这样的情况，显然对承租者是不公平的。

因为评价信息中对房屋的评价比对人的评价多很多各维度，所以在表结构设计上为了减少数据库表中的字段，采用了分表设计，这样设计的另一个优点是可以更方便的对房屋评价信息做权限控制，因为对于房屋的评价信息中可能会涉及到房屋地理位置等隐私信息。系统在设计上默认房屋的评价信息对承租者是不可见的，但承租者可以看到所有前租户对它的评价得分情况。如果承租者需要了解房子的评价信息，需要得到房东的授权。

评论模块具有的操作主要有下面这几点：

1. 对于人的评论和人进行绑定，无论房东合适承租者都会有多条评价信息，每条信息都代表这一次完成的交易；
2. 对于房屋的评价会有多个维度，也会有多个维度的打分情况，便于更真是有效的描述房屋信息；
3. 评价是不能由用户删除的，每一位用户都是基于身份证进行验证，不存在多账号同一人的行为，所以，参与评价的用户系统默认都是具有良好品德，能够实事求是发表意见的用户。

### 3.3.4 签约模块设计

签约模块的存在为前面所提到的关系提供了真实性、有效性的保障。签约模块主要描述了合同的一些相关内容，这些内容主要包括双方的基本信息，签约前房内物品的核对，双方在合同有效期间应尽的责任与义务，约定租金支付时间与金，商议出现纠纷时的处理方案。

为了租赁过程的透明性和有效性，系统支持在线签约，签约成功后双方可以随时调阅，并且能够导出PDF版本使用。签约模块如图3.7所示。

# 第2章 系统的技术选型

技术选型在一个系统设计之初尤为重要，这就像一所高楼的地基，决定了这套系统的天花板。所以，在相关技术的选择上，深刻分析了其优势点和弱势点，也对比了业界相似的技术，最后在性能、资源消耗、开发效率这三个因素下，确定了此套系统的实现技术。

## 2.1 开发语言

本项目的开发语言采用的是Go语言，Go语言被称为是“21世纪的编程语言”，它在设计之初就是为了弥补传统语言的缺点[3]。传统语言比如C++、Java等静态语言的执行速度很快，但是开发的过程中相对缓慢，实现一个功能往往需要编写大量的代码，而Python、PHP等动态语言开发效率比较高，却可以用很少的代码实现一个功能，但是执行速度相比起静态语言可谓非常慢。Go语言的设计既兼顾了静态语言的速度，又保持了动态语言的简洁[4]。这门语言还有一些其他优点，主要有下面几个：

1. 原生支持并发编程，能够很简洁的写出支持高并发的代码；
2. 能很好的应对大数据、微服务场景；
3. 语言本身要求编写规范，不需要第三方指定规范，实现了编码风格的统一;
4. 语法简洁优雅，关键字少，可以让编码更加专注于功能的实现；
5. 自带垃圾回收机制，无需手动干预内存，保证了内存安全；

## 2.2 数据存储与缓存

数据库采用的是PostgreSQL，它是一个开源的对象-关系型数据库管理系统，它完全支持SQL标准，并且具有很多现代特性，比如事物完整性、触发器、视图、多版本并发控制等[5]。在数据库选择的时候，最初考虑到的是MySQL，它的特性也基本可以满足本系统的使用，但是在开源许可协议上的开放度、数据库内部数据类型的限制以及对于多版本并发控制上，还是逊色于PostgreSQL，所以最终采用的是PostgreSQL[6]。

缓存采用的是Redis，Redis是一种Key-Value型数据库[7]，它存储的数据是在

内存中，所以对数据的增删改查非常的快速。由于系统跟踪用户状态使用的技