

设计文档

1. 引言

1.1 编写目的

本设计文档用于对本次课程设计“宠物寄养平台”的系统设计方案进行统一说明与规范描述，明确系统的总体结构、模块划分、业务流程以及数据库设计等关键内容，从而为后续的实现、测试与验收提供依据。在文档中会将需求的具体实现方案进行讲解，能够通过文档清楚知道我们项目的技术路线。

1.2 系统目标

1.2.1 功能目标

本系统的总体目标是实现一个具备较为完整的宠物寄养页面，使用户能够完成从浏览到预约的主要操作，并具备清晰的信息结构与可扩展的数据设计。以卡片列表形式展示多条寄养服务信息，包括服务的名称与项目的收费标准。支持标签的检索功能，实现高效率检索。提供统一表单让用户填写基本信息与服务需求，并完成提交反馈，同时支持订单的支付页面。

1.2.2 设计目标

对页面功能进行合理模块划分，包括列表、筛选、详情等模块，并给出模块图说明各个模块的职责与关系。给出用户操作流程图与后台管理业务流程图，确保交互逻辑明确、可实现。设计数据库，明确字段含义、约束关系以及主外键，保证数据存储简洁实用。

2. 系统总体设计

2.1 系统功能概述

整个系统分为三个部分，后端数据库设计，用来存储操作中会涉及到的所有数据；客户端（前台）页面，用于顾客了解服务详情、添加宠物信息、添加服务订单与联系工作人员等；管理（后台）页面，用于员工和管理员管理订单与员工信息等。

2.2 系统架构设计

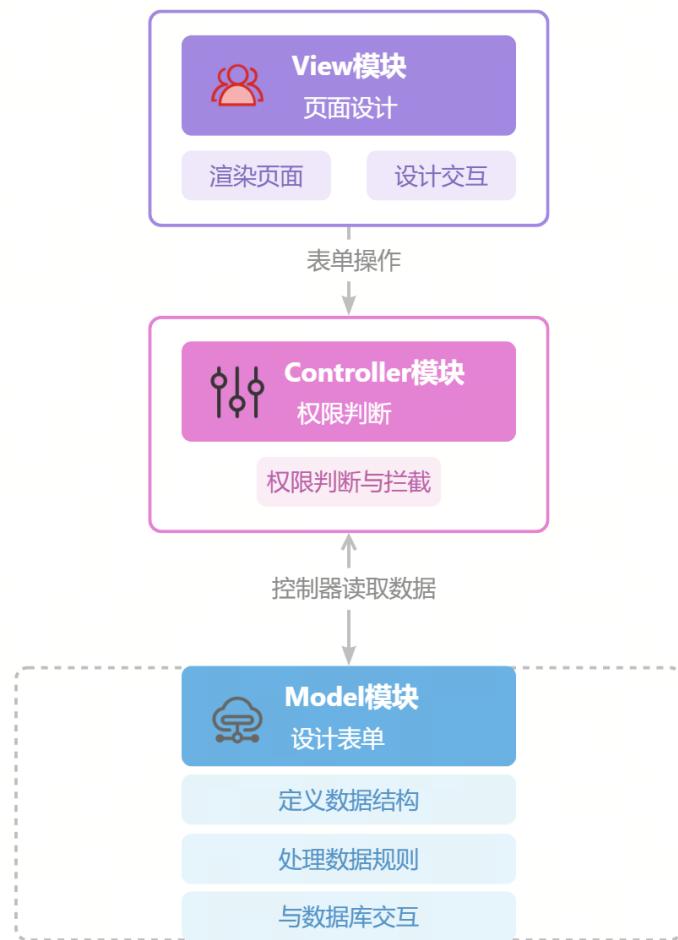
本系统选择使用**Yii2**实现前端页面的整体构建，包括Model的生成与增删改查权限的设计；使用**Apache作为本地Web服务器**，连接到后端数据库，**数据库使用MySQL**。脚本运行环境为PHP，使用**phpMyAdmin**进行数据库页表与信息管理。

3. 模块设计

3.1 模块划分说明

- Model模块：定义数据结构，设计存储字段；处理数据逻辑和规则，筛选需要的数据；从数据库读写数据。
- Controller模块：接收用户操作，做权限的判断与拦截，调用Model处理数据，再让View模块更新界面。
- View模块：渲染页面，展示数据，设计交互功能，把接收的事件交给Controller模块。

3.2 模块图



3.3 关键模块详细设计

1. View层

这一层是最顶层的模块，在这个模块中会对页面进行渲染，实现所有功能的可视化。在这一层，会实现前台和后台的登陆页面设计，以及客户端首页的设计和后台管理的设计。前端的交互功能也会在这一层进行实现，设计交互入口，以及交互逻辑等。表单数据的可视化也会在这一层进行实现，这一层中会把表单的数据可视地显示出来，可以通过交互进行表单的特定操作。

2. Controller层

这一层负责前端和后端的交互功能，所有的权限设置都在这一层考虑，前端的交互会经过这一层的判断进行限制。被限制的操作会显示不可进行，所有表单的增删改查会在这一层考虑到。

3. Model层

这一层是最底层的设计，这一层就需要进行数据库的设计了，需要设计有哪些表，以及表中要有哪些字段。每一个表单都会生成一个Model，会在Model的控制器中限制表单的权限。

4. 流程设计

4.1 业务流程总览

业务流程主要分为三个部分，一个是顾客的浏览流程，一个是员工的管理流程，还有一个是管理员的流程。我们这里把前台和后台单独分开进行考虑。顾客会进入前台的浏览流程，员工和管理员都会进入后台的管理流程。下面会分别介绍不同身份的业务流程。

4.1.1 顾客流程

对于顾客来说，主要的需求就是如何下单。在我们设计的系统中，顾客会先进入客户端的注册页面，填写个人信息进行账号的注册；接着会跳转到客户端的登录页面，输入用户名和密码登录账号；登录之后会进入到客户端的前台页面，在这个页面中顾客可以查看相关的介绍。

主页中有四个功能按钮，点击可以进行不同的操作，分别为个人信息、宠物信息、订单信息以及寄养服务。用户接下来可以点进个人信息页面，查看自己的账号编号、会员等级等，还可以修改个人信息，包括用户名、性别、联系方式和地址；接着可以点击宠物信息，添加自己的宠物相关信息，包括宠物品种、宠物名字和性别等；接着可以点击寄养服务，查看平台提供的服务以及各个服务的价格；根据自己的需求在订单信息页面创建新的订单，在这里可以选择负责的员工，还可以直接进行支付操作。

4.1.2 员工流程

员工的主要职责就是负责好分配给自己的订单，员工要先在员工注册界面进行账号的注册，接着会跳转到后台管理登录页面，输入信息登录之后就会进入员工专属的后台首页。员工管理页面分为五个部分，分别为宠物管理、个人信息、顾客管理、订单管理和服务管理。

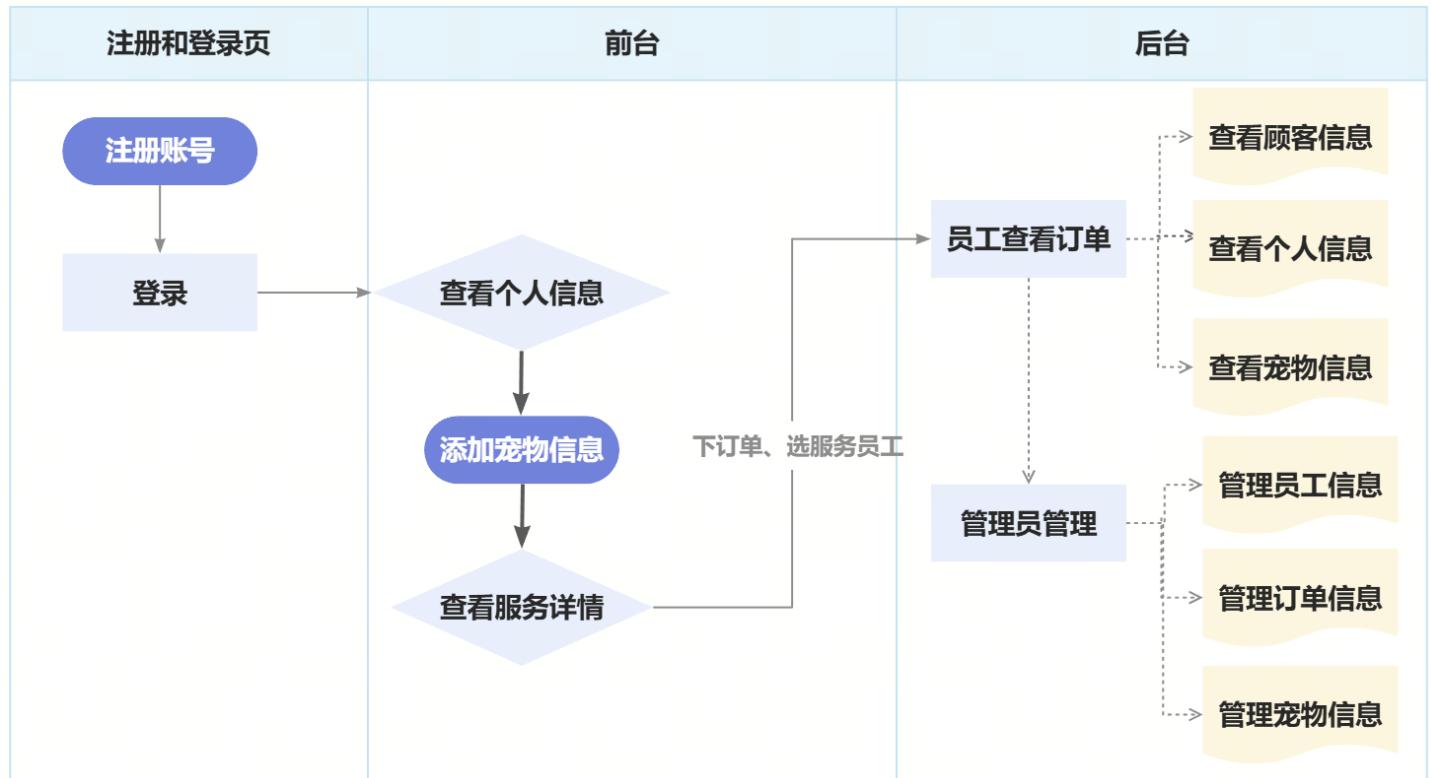
在宠物管理部分，员工可以查看自己的订单的宠物相关信息；在个人信息部分，员工只能查看自己的相关信息，还可以修改自己的联系方式；在顾客管理部分，可以查看属于自己的订单的顾客信息；订单管理部分，查看给自己分配的订单要求；服务管理部分，只能够查看现在平台提供的服务和价格。

4.1.3 管理员流程

管理员账号已经提前在数据库中添加好了，这里可以直接在后台管理登录页面进行管理员的登录。登录之后会进入后台管理员管理页面，管理员页面和员工页面是一样的，但是权限不同。管理员页面也有五个部分，分别为宠物管理、员工管理、顾客管理、订单管理和服务管理。

宠物管理部分可以查看平台所有的宠物信息，还可以直接增加宠物信息，分为增加猫和增加狗；员工管理部分可以查看所有员工的信息，也可以删除或者改员工的信息；顾客管理部分可以查看平台所有的订单信息，并且可以增加或者删除平台的订单；服务管理部分只能够进入查看平台的服务和价格。

4.2 操作流程图



5. 数据库设计

5.1 字段设计

这里主要设计了五张表和两张子表，主表分别为**employee**员工表、**user**用户账号表、**customer**顾客表、**fosterorder**订单表、**fosterservice**服务表、**pet**宠物信息；两张pet的子表**cat**猫表和**dog**狗表，还会产生一张中间表**order_employee**。

5.1.1 employee表

这张表设计了七个字段，分别为**EmployeeID**、**user_id**、**Name**、**Gender**、**Position**、**Contact**和**HireDate**。主键为**EmployeeID**员工ID，每一个员工都会有一个不能重复的ID号来代表身份，这个字段在注册的时候会自动分配；**user_id**为外键，关联到**user**表，每一个注册过的账号信息都会在**user**表中存储；**Name**是员工姓名，**Gender**是员工的性别，我们在这里设计了一个枚举值，只能选择“男”或者“女”；**Position**是员工职务，**Contact**是联系方式，**HireDate**为入职时间，在注册账号时会直接在日历表上选择时间，类型为date型。

5.1.2 customer表

这张表也设计了七个字段，分别为**CustomerID**、**user_id**、**Name**、**Gender**、**Contact**、**Address**和**MemberLevel**。主键为CustomerID顾客ID，每一个顾客都会有一个不能重复的ID来表示身份，注册的时候也会自动分配；user_id为外键，也是user表的唯一代号值；Name为用户名，Gender也是一个枚举值，选择“男”或“女”；Contact为联系方式、Address为家庭住址；MemberLevel为顾客身份，也是一个枚举值，可以选择“普通会员”、“银卡会员”或者“金卡会员”。

5.1.3 user表

这张表设计了十一个字段，分别为**id**、**username**、**auth_key**、**password_hash**、**password_reset_token**、**email**、**status**、**role**、**created_at**、**updated_at**、**verification_token**。主键为id，外键为password_reset_token和email，这个会对应到其他的表中；所有账号的密码会在这里通过哈希函数进行加密，加密后存储到user表中，这里不能反向算出明文密码；这里的role记录了三个身份，分别为employee员工、admin管理员和customer顾客，管理员只有一个，一开始就在表里加好了数据。

5.1.4 fosterorder表

这张表设计了八个字段，分别为**OrderID**、**CustomerID**、**PetID**、**ServiceID**、**StartTime**、**EndTime**、**OrderStatus**和**PaymentAmount**。主键为OrderID，记录订单的唯一值；外键为CustomerID、PetID和ServiceID，对应到顾客表、宠物表和服务表；还会记录订单的开始时间和结束时间，类型都为datetime；OrderStatus记录支付情况，是一个枚举值，可以选择“未支付”和“已支付”；PaymentAmount记录订单的价格，这里会根据天数自动计算价格。

5.1.5 fosterservice表

这张表设计了五个字段，分别为**ServiceID**、**ServiceType**、**PetCategory**、**Price**和**Duration**。主键为ServiceID，没有外键；ServiceType记录服务的类型，这里设计了一个枚举值，可以选择“普通寄养”或者“豪华寄养”；PetCategory记录宠物的外观情况，设计了枚举值，选择“猫”、“小型犬”、“中型犬”和“大型犬”；Price记录每一种服务的价格；Duration记录订单的持续时间。

5.1.6 pet表

这张表设计了七个字段和两个子表，主表记录了宠物的基本信息，两个子表分别为猫表和狗表，记录宠物的特点信息。主表字段分别为**PetID**、**CustomerID**、**PetName**、**Gender**、**AgeYears**、**AgeMonths**和**HealthStatus**。主键为PetID，外键为CustomerID，对应着哪个顾客的宠物；Gender是一个枚举值，可以选择“公”或者“母”；AgeYears和AgeMonths记录宠物的年龄和月份；HealthStatus可以填写宠物的健康情况。

5.1.6.1 cat表

这个表是pet表的子表，记录的都是猫的情况，设计了三个字段，分别为**PetID**、**FurLength**和**Personality**。继承了主表的主键PetID；新加了外貌字段FurLength，这是一个枚举值，可以选择“短毛”、“中毛”或者“长毛”；还新加了性格字段，可以填写猫的性格信息。

5.1.6.2 dog表

这个表也是pet表的子表，记录的都是狗的情况，设计了三个字段，分别为**PetID**、**DogBreedType**和**TrainingLevel**。PetID也是从主表继承过来的主键，其他的两个字段都是枚举值，DogBreedType记录狗的大小，可以选择“大型犬”、“中型犬”和“小型犬”，TrainingLevel可以选择“未训练”、“基础训练”或者“高级训练”。

5.2 权限设计

- 三类人员：管理员、员工、客户。
- 所有人不能修改主外键，包括所有的ID值。

5.2.1 employee 表

- 管理员：有权限删改查，增应该去职员注册页面自行注册。
- 员工：无权限增、删，改只能修改自己的Contact，查只能查自己的全部表项。
- 客户：无权增删改查，不可以查看员工任何信息。

5.2.2 costomer 表

- 管理员：有权限删改查，不能查看密码，增应该去客户注册页面自行注册。
- 员工：无权限增、删、改，查只能查看自己匹配的用户信息。
- 客户：无权增、删，改不能修改自己的ID+MemberLevel，可以查自己的所有信息。

5.2.3 Pet 表

- 管理员：有权增删改查。
- 员工：无权增、删、改，有权查看自己匹配的宠物信息。
- 客户：增、删只能删自己的ID的，改只能修改自己ID的非主键，只能查看自己ID对应的宠物信息。

5.2.4 子表Cat、Dog

- 管理员：有权增删改查。
- 员工：无权增、删、改，有权查看自己匹配的宠物信息。
- 客户：增、删只能删自己的ID的，改只能修改自己ID的非主键，只能查看自己ID的。

5.2.5 fosterservice表

- 管理员：增删改查都可以做。
- 员工：不能增删改，只能查。
- 客户：不能增删改，只能查。

5.2.6 fosterorder

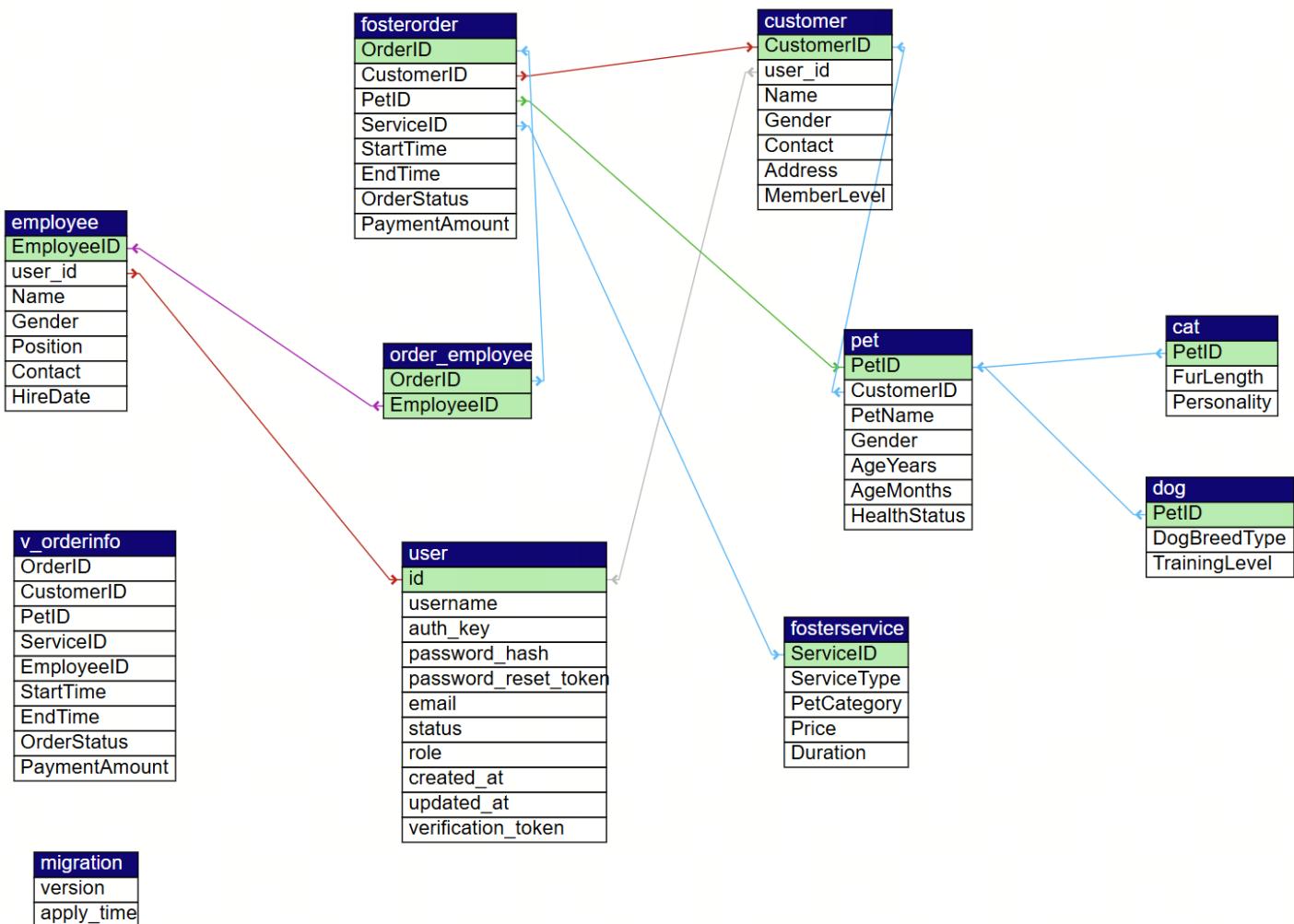
- 管理员：增删查都可以做，不能改。

- 员工：不能增删改，只能查匹配信息。
- 客户：增可以自己下单；不能删，只能通过支付码改支付状态，只能查自己订单。

5.2.7 Order_employee

- 管理员：增删改查都可以。
- 员工：无权增删改查，因为是中间表。
- 客户：无权增删改查。

5.3 概念结构设计 (E-R 图)



这里主要设计了五张表和两张子表以及一张中间表，主表分别为employee员工表、user用户账号表、customer顾客表、fosterorder订单表、fosterservice服务表、pet宠物信息；两张pet的子表cat猫表和dog狗表，以及一张中间表order_employee订单和员工表。v_orderinfo是视图表，不属于自己设计的范围；migration表是迁移的时候产生的表，用来记录数据库的版本和迁移时间。

5.3.1 表与表关系设计

5.3.1.1 User 与 Customer和Employee (账户与角色扩展)

- customer.user_id → user.id，每个 Customer 本质上是一个 user 账号的扩展信息。

- employee.user_id → user.id，每个 Employee 也是一个 user 账号的扩展信息。

5.3.1.2 Customer 与 Pet (客户与宠物)

- pet.CustomerID → customer.CustomerID 关系为**1对多**，一个客户 (Customer) 可以有多只宠物 (Pet)，每只宠物只属于一个客户。

5.3.1.3 Pet 与 Cat和Dog (宠物的继承与分类)

- cat.PetID → pet.PetID 关系为**1 对 1 (子表)**，如果这只宠物是猫，就会在 `cat` 表里补充猫的特有属性。
- dog.PetID → pet.PetID 关系为**1 对 1 (子表)**，如果这只宠物是狗，就会在 `dog` 表里补充狗的特有属性。

5.3.1.4 FosterOrder 与 Customer、Pet 和 FosterService (订单关联谁、什么宠物、什么服务)

- fosterorder.CustomerID → customer.CustomerID 关系为**多对1** (一个客户对多个订单)，一个客户可以下很多单，每个订单属于一个客户。
- fosterorder.PetID → pet.PetID 关系为**多对1** (一个宠物对多个订单)，一只宠物可以产生多次寄养订单，每个订单对应一只宠物。
- fosterorder.ServiceID → fosterservice.ServiceID 关系为**多对1** (一个服务对多个订单)，一种寄养服务可被多次下单，每个订单选择一种服务。

5.3.1.5 FosterOrder 与 Employee (通过中间表实现多对多)

- `order_employee.OrderID` → `fosterorder.OrderID`
- `order_employee.EmployeeID` → `employee.EmployeeID`

整体关系都是**订单和员工的多对多**，一个订单可以由多个员工负责，一个员工也可以处理多个订单，`order_employee` 就是专门用来“绑定订单和员工”的关联表。

5.3.1.6 v_orderinfo (视图 View)

- **v_orderinfo 不是实体表，是视图**，它把订单相关字段整合起来，方便查询展示。

5.3.1.7 migration (迁移记录表)

- `migration` 用来记录数据库版本/迁移时间，**不参与业务关系**。

6. 页面结构说明

页面主要分为**注册页面、登录页面、平台主页、表单操作页面**。注册页面分为顾客进入的顾客注册页面和员工进入的员工注册页面，两个身份注册时需要填写的信息不同，员工需要填写邀请码才能成功注册。登录页面也分为顾客登录页面、员工和管理员登录页面，只有在符合自己身份的页面进行登录才能成功跳转到主页。

平台主页分为前台和后台页面，前台页面提供给顾客，顾客在这个页面进行了解平台信息、下订单等操作；后台页面提供给管理员和员工进行管理，管理员和员工可以在这个页面管理表单信息。通过平台主页的按钮，可以跳转到**表单操作页面**，在这个页面可以对不同的表单进行增删改查等操作。