

吃在科大 · 餐饮点评系统

李睿哲 PB22061168 王文彦 PB22611935
人工智能与数据科学学院 2022 级
2024. 05. 11–2022. 11. 16

目录

| | |
|-------------------------|----|
| 1. 绪论 | 4 |
| 1.1 分工 | 4 |
| 1.2 课程设计环境及技术 | 5 |
| 2. 需求分析 | 5 |
| 2.1 信息需求 | 5 |
| 2.2 处理需求 | 6 |
| 3. 数据库设计 | 8 |
| 3.1 数据库概念结构设计 | 8 |
| 3.2 数据库逻辑结构设计 | 9 |
| 3.3 数据库物理结构设计 | 11 |
| 4. 功能详细设计与实现 | 12 |
| 4.1 实现的全部功能表 | 12 |
| 4.2 登陆与注册功能 | 13 |
| 4.3 餐厅创建功能（顾客/经理） | 15 |
| 4.4 餐厅认领与管理功能（经理） | 17 |
| 4.5 餐厅检索、排序功能（顾客） | 18 |
| 4.7 查看餐厅/菜品详情的功能 | 21 |
| 4.8 修改餐厅/菜品信息功能 | 22 |
| 4.9 餐厅评分/评论/回复功能 | 24 |
| 4.10 餐厅删除功能 | 25 |
| 4.11 完整性概述 | 25 |
| 4.12 安全性概述 | 26 |
| 5. 项目测试 | 27 |
| 5.1 功能测试 | 27 |
| 5.2 完整性测试 | 43 |
| 5.3 安全性测试 | 45 |

1. 緒論

1.1 分工

| 大类 | 详细 | 李睿哲 PB22061168 | 王文彥 PB22611935 |
|------|-----------|----------------|----------------|
| 前期准备 | 信息搜集 | | |
| | 需求分析 | | |
| | 概念结构设计 | | |
| | 逻辑结构设计 | | |
| 后端设计 | 用户注册与登录 | | |
| | 增加餐厅 | | |
| | 餐厅/菜品筛选 | | |
| | 餐厅/菜品排序 | | |
| | 餐厅/菜品搜索 | | |
| | 修改餐厅/菜品信息 | | |
| | 餐厅相关信息检索 | | |
| | 评论 | | |
| | 回复 | | |
| | 餐厅/菜品均分计算 | | |
| | 经理认领餐厅 | | |
| | 经理拥有餐厅检索 | | |
| | 删除餐厅申请 | | |
| 前端设计 | 注册/登录界面 | | |
| | 餐厅/菜品列表页 | | |
| | 餐厅/菜品详情页 | | |
| | 经理主页 | | |
| | 各种表单 | | |
| | CSS 设计 | | |
| 运行维护 | 调试 | | |
| 材料撰写 | E-R 图绘制 | | |
| | 需求分析文档撰写 | | |
| | 实验报告 | | |
| | README 编写 | | |

*颜色越深代表贡献越大

1.2 课程设计环境及技术

在本次数据库实验中，我们主要使用了以下环境与技术：

1. 编程语言：

- **Python (3.11.4)**：作为本次实验的主要编程语言，Python 以其简洁和易读性使得开发过程更加高效。我们利用 Python 的强大生态系统和丰富的库来实现数据库操作和处理。

2. Web 框架：

- **Django (4.1)**：Django 是一个高层次的 Python Web 框架，鼓励快速开发和干净、实用的设计。本次实验使用 Django 框架来搭建 Web 应用，实现了 MVC 模式下的数据库交互，提供了灵活的视图控制和强大的后台管理功能。

3. 前端技术：

- **HTML**：超文本标记语言 (HTML) 是构建 Web 页面的基础。本实验中，HTML 用于定义 Web 页面的结构和内容。
- **CSS**：层叠样式表 (CSS) 用于描述 HTML 文档的显示样式。在本次实验中，我们使用 CSS 来美化 Web 页面，提高用户体验。

4. 数据库管理系统：

- **MySQL**：MySQL 是一个广泛使用的关系型数据库管理系统 (RDBMS)，以其性能高、可靠性强和易用性著称。在本实验中，MySQL 用于存储和管理应用程序所需的数据，并通过 Django ORM (对象关系映射) 进行数据库操作。

2. 需求分析

2.1 信息需求

该系统需要存储以下信息：

(一) 餐厅信息

包括餐厅名称、区域、位置、营业时间、菜系/口味、均价、均分、是否正常营业、餐厅经理相关信息和餐厅图片。

(二) 菜品信息

包括菜品所属的餐厅、菜品名称、口味、价格、是否在售、菜品图片。

(三) 用户信息

该系统共有 3 类用户，餐厅经理、注册顾客与未注册顾客，其中未注册用户不需要存储任何信息，餐厅经理与注册用户需存储的信息如下：

1. 餐厅经理

包括账号、密码、邮箱。

2. 注册用户

包括账号、密码。

(四) 点评信息

点评共分为 2 类，对餐厅的点评与回复，其分别需存储的信息如下：

1. 对餐厅的点评

包括点评的用户、点评的餐厅、点评的菜品（可以为空，代表对餐厅整体的点评）、打分、点评内容。

2. 回复*

包括给出回复的用户、被回复的评论、回复内容。

(五) 待处理请求

考虑到部分操作需要的权限较高，用户发出请求需要向超级管理员申请，具体包括：

1. 餐厅的删除

需要存储：删除的餐厅，餐厅倒闭的证据（图片）。

2. 经理对餐厅的接管

需要存储：经理，餐厅，经理对餐厅的所有权证据（图片）。

2.2 处理需求

2.2.1 角色与权限

该系统共有 4 种数据库角色，不同角色具有的权限如下表所示：

| 对象 | 操作 | 未注册用户 | 注册用户 | 餐厅经理 | 超级管理员 |
|------|----|-------|--------|-------|-------|
| 餐厅信息 | 增 | | | 自己的餐厅 | |
| | 删 | | | 自己的餐厅 | |
| | 改 | | 经理未入驻时 | 自己的餐厅 | |
| | 查 | | | | |
| 菜品信息 | 增 | | 经理未入驻时 | 自己的餐厅 | |
| | 删 | | 经理未入驻时 | 自己的餐厅 | |
| | 改 | | 经理未入驻时 | 自己的餐厅 | |
| | 查 | | | | |
| 评论信息 | 增 | | | | |
| | 删 | | | | |
| | 改 | | | | |
| | 查 | | | | |
| 用户信息 | 增 | | | | |
| | 删 | | | | |
| | 改 | | | | |
| | 查 | | | | |

2.2.2 不同角色的处理需求

1. 未注册用户

1. 筛选、排序特定餐厅

根据区域、营业时间、菜系/口味筛选特定餐厅，并按均分进行排序。

2. 搜索特定餐厅/菜品

根据餐厅名称搜索特定餐厅/菜品。

2. 注册用户

1. 注册用户具有未注册用户所具有的所有功能

2. 增加餐厅*

若系统中暂时没有某餐厅信息，注册用户可自行添加餐厅。

3. 修改餐厅信息

当对应餐厅的经理未入驻时，注册用户有权编辑餐厅信息。

4. 申请删除餐厅

当餐厅倒闭时，注册用户可向超级管理员申请删除餐厅。

5. 增加、修改、删除菜品

当对应餐厅的经理未入驻时，注册用户有权增加、修改、删除该餐厅的菜品。

6. 评价

注册用户有权对餐厅进行评价（包括打分与评论），可针对某一道菜或餐厅整体。且注册用户可回复他人的评价*。

3. 餐厅经理

1. 增加/认领餐厅*

经理可向超级管理员提交证据，添加/认领自己的餐厅。

2. 修改、删除餐厅

经理有权修改自己的餐厅信息或删除自己的餐厅。

3. 增加、修改、删除菜品

经理有权增加、修改、删除自己餐厅的菜品。

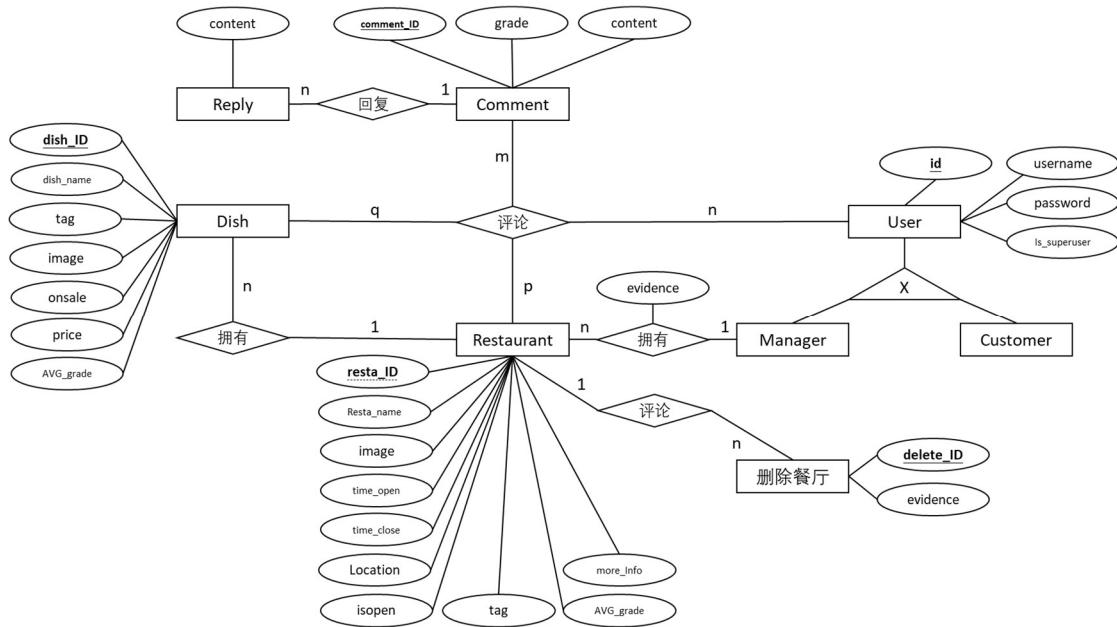
4. 回复

经理有权回复他人对餐厅的评价。

3. 数据库设计

3.1 数据库概念结构设计

3.1.1 E-R 图



3.1.2 概念结构描述

(一) 实体与关系的描述

1. User (用户) 分为 Customer (顾客) 和 Manager (经理)
2. 一个经理拥有多个餐厅 (餐厅可以没有经理)
3. 一个餐厅拥有多个菜品
4. 一条评论对应一个用户、一家餐厅、一道菜品 (仅菜品可以为空), 每个餐厅、菜品、用户可以有多条评论
5. 一条评论可以有多条回复
6. 一个餐厅可以有多条“删除餐厅” (即倒闭的证据), 供超级管理员审核

(二) 部分属性的描述

1. Restaurant
 - Isopen**: 考虑到餐厅可能因为某些原因临时关闭一段时间, 这是直接删除餐厅是不合理的, 故加入 isopen 布尔属性, 用以判断餐厅是否正常营业。若有餐厅经理, 则只有经理有修改此属性的权限; 若无餐厅经理, 则所有登录用户都有权限修改。
 - Location、Tag**: 均只含固定选项, 填写或修改时在特定选项里选择。
 - 其余属性均为字面意思, 不做过多解释.

2. Dish

onsale: 类似 Restaurant.isopen

tag: 同 Restaurant.tag

其余属性均为字面意思，不做过多解释。

3. Evidence (包括经理与餐厅关系的 evidence 和删除餐厅的 evidence)

认领餐厅及餐厅倒闭的证据，以图片的形式存储

其余属性均为字面意思，不做过多解释。

3.2 数据库逻辑结构设计

3.2.1 由 ER 图向关系模型的转换

根据得到的 ER 图，按照以下规则进行关系模式的转换：

一对联系：若双方部分的参与，则将联系定义为一个新的关系，属性为参与双方的码，

若一方全部参与，则将联系另一方的码作为全部参与一方的属性；

一对多联系：将单方参与实体的码作为多方参与实体的属性；

多对多联系：将联系定为新的关系，属性为参与双方的码。

具体得到的关系模型如下：

| auth_user | | | | |
|---------------------|--------------|--------|-----|-------|
| 数据项名 | 数据类型 | 含义 | 可否空 | 主码/外码 |
| id | INT | 用户 ID | 否 | 主码 |
| password | VARCHAR(128) | 密码 | 否 | |
| last_login | DATETIME | 最后登录时间 | 是 | |
| is_superuser | TINYINT(1) | 超级管理员 | 否 | |
| username | VARCHAR(150) | 用户名 | 否 | |
| first_name | VARCHAR(150) | 名 | 是 | |
| last_name | VARCHAR(150) | 姓 | 是 | |
| email | VARCHAR(254) | 电子邮件 | 是 | |
| is_staff | TINYINT(1) | 是否工作人员 | 否 | |
| is_active | TINYINT(1) | 是否活跃 | 否 | |
| date_joined | DATETIME | 加入时间 | 否 | |

| RESTAURANT | | | | |
|------------|--------------|-------|-----|------------------|
| 数据项名 | 数据类型 | 含义 | 可否空 | 主码/外码 |
| resta_ID | INT | 餐厅 ID | 否 | 主码 |
| resta_name | VARCHAR(20) | 餐厅名称 | 否 | |
| location | CHAR(1) | 位置 | 否 | |
| time_open | TIME | 开店时间 | 是 | |
| time_close | TIME | 关店时间 | 是 | |
| manager | INT | 经理 | 是 | 外码(auth_user.id) |
| tag | CHAR(4) | 标签 | 是 | |
| AVG_grade | FLOAT | 平均评分 | 是 | |
| more_Info | VARCHAR(200) | 更多信息 | 是 | |
| image | VARCHAR(100) | 图片 | 否 | |
| isopen | TINYINT(1) | 是否营业 | 否 | |

| DISH | | | | |
|-----------|--------------|-------|-----|-------------------------|
| 数据项名 | 数据类型 | 含义 | 可否空 | 主码/外码 |
| dish_ID | INT | 菜品 ID | 否 | 主码 |
| resta_ID | INT | 餐厅 ID | 否 | 外码(RESTAURANT.resta_ID) |
| dish_name | VARCHAR(20) | 菜品名称 | 否 | |
| tag | CHAR(4) | 标签 | 是 | |
| price | FLOAT | 价格 | 是 | |
| image | VARCHAR(100) | 图片 | 否 | |
| onsale | TINYINT(1) | 是否在售 | 否 | |
| more_Info | VARCHAR(200) | 更多信息 | 是 | |
| AVG_grade | FLOAT | 平均评分 | 是 | |

| COMMENT | | | | |
|----------|--------------|-------|-----|-------------------------|
| 数据项名 | 数据类型 | 含义 | 可否空 | 主码/外码 |
| comm_ID | INT | 评论 ID | 否 | 主码 |
| user_ID | INT | 用户 ID | 否 | 外码(auth_user.id) |
| resta_ID | INT | 餐厅 ID | 否 | 外码(RESTAURANT.resta_ID) |
| dish_ID | INT | 菜品 ID | 是 | 外码(DISH.dish_ID) |
| grade | FLOAT | 评分 | 否 | |
| content | VARCHAR(200) | 内容 | 否 | |

| REPLY | | | | |
|----------|--------------|-------|-----|---------------------|
| 数据项名 | 数据类型 | 含义 | 可否空 | 主码/外码 |
| reply_ID | INT | 回复 ID | 否 | 主码 |
| user_ID | INT | 用户 ID | 否 | 外码(auth_user.id) |
| comm_ID | INT | 评论 ID | 否 | 外码(COMMENT.comm_ID) |
| content | VARCHAR(200) | 内容 | 否 | |

| DELETE_RESTA | | | | |
|--------------|--------------|-------|-----|-------------------------|
| 数据项名 | 数据类型 | 含义 | 可否空 | 主码/外码 |
| delete_ID | INT | 删除 ID | 否 | 主码 |
| resta_ID | INT | 餐厅 ID | 否 | 外码(RESTAURANT.resta_ID) |
| evidence | VARCHAR(100) | 证据 | 否 | |

| MANAGER_REG | | | | |
|-------------|--------------|-------|-----|-------------------------|
| 数据项名 | 数据类型 | 含义 | 可否空 | 主码/外码 |
| reg_ID | INT | 注册 ID | 否 | 主码 |
| user_ID | INT | 用户 ID | 否 | 外码(auth_user.id) |
| resta_ID | INT | 餐厅 ID | 否 | 外码(RESTAURANT.resta_ID) |
| evidence | VARCHAR(100) | 证据 | 否 | |

3.2.2 关系模型的范式分析

要分析每个表满足的范式，我们需要检查以下几点：

第一范式 (1NF)：表中每一列都只包含原子值（即不可再分的值）。

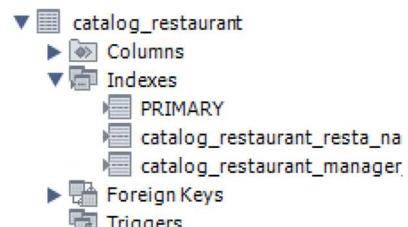
第二范式 (2NF)：表满足 1NF，且表中的非主属性完全依赖于主键。

第三范式 (3NF)：表满足 2NF，且表中的非主属性不传递依赖于主键。

我们逐一分析每个表，不难得出，每一个基本表均至少满足 3NF。

3.3 数据库物理结构设计

数据库的物理设计通常是确定数据的相关存储过程，它一般是依赖于相应的数据库管理系统，几乎是系统进行自主完成的任务，不用过多的人去考虑。例如，如图，我们发现 Django 已经自动为我们的基本表建立了索引。



4. 功能详细设计与实现

4.1 实现的全部功能表

| 功能 | 详细说明 |
|---------|---|
| 用户注册 | 用户可以注册成为普通用户或餐厅经理 |
| 用户登录 | 用户可以登录系统，分为普通用户和经理登录 |
| 餐厅创建 | 用户可以创建新的餐厅，添加餐厅的基本信息和图片 |
| 餐厅认领 | 餐厅经理可以认领未被认领的餐厅，提交认领申请并等待管理员审核 |
| 餐厅管理 | 经理可以管理已认领的餐厅，包括修改餐厅信息和上传图片 |
| 餐厅删除申请 | 用户可以提交餐厅删除申请，并上传餐厅倒闭的证据 |
| 餐厅信息修改 | 经理或注册用户可以修改餐厅的基本信息 |
| 菜品创建 | 用户可以为餐厅添加新的菜品，输入菜品信息和上传图片 |
| 菜品信息修改 | 经理或注册用户可以修改菜品信息 |
| 餐厅检索和排序 | 用户可以根据不同条件检索和排序餐厅，包括评分、位置、标签等 |
| 餐厅详情查看 | 用户可以查看餐厅的详细信息，包括菜品和评论 |
| 菜品详情查看 | 用户可以查看菜品的详细信息和评论 |
| 餐厅评分 | 用户可以对餐厅进行评分，评分将影响餐厅的平均评分 |
| 评论和回复 | 用户可以对餐厅和菜品进行评论，并可以回复其他用户的评论 |
| 完整性检查 | 系统对数据的完整性进行检查，包括实体完整性、参照完整性和用户定义的完整性 |
| 安全性检查 | 系统对用户访问权限进行安全性检查，确保只有经过认证的用户才能访问系统的特定功能 |

*具体以其余小节为准

4.2 登陆与注册功能

4.2.1 顾客注册

顾客注册功能由 `register` 函数实现，该函数处理用户的注册请求。简要代码与分析如下：

```
def register(request):
    if request.method == 'GET':
        return render(request, 'registration/register.html')
    elif request.method == 'POST':
        user_name = request.POST.get('username')
        pwd = request.POST.get('password')
        if not user_name or not pwd: # 信息不完整
            return render(request, 'registration/register.html', {'errors': ['xxx']})
        if User.objects.filter(username=user_name): # 用户名重复
            return render(request, 'registration/register.html', {'errors': ['xxx']})
        else:
            user = User.objects.create_user(xxx)
            user.save()
            return redirect('restaurants')
    else:
        return render(request, 'registration/register.html')
```

实现步骤

1. 检查请求方法：
 - 如果请求方法是 GET，则返回注册页面。
 - 如果请求方法是 POST，则处理注册表单提交的数据。
2. 获取表单数据：
 - 从请求中获取用户名 (username) 和密码 (password)。
3. 验证表单数据：
 - 如果用户名或密码为空，返回注册页面并提示用户填写完整信息。
 - 如果用户名已存在于数据库中，返回注册页面并提示用户换一个用户名。
4. 创建新用户：
 - 使用 `User.objects.create_user` 方法创建一个新用户，设置用户名、密码、电子邮件等信息。注意，对于顾客的注册而言，`is_stuff` 设置为 False。
 - 调用 `user.save()` 方法将新用户保存到数据库中。
5. 重定向：
 - 用户注册成功后，重定向到餐厅列表页面（即顾客主页）。

4.2.2 经理注册

经理注册功能由 `register_manager` 函数实现，基本实现思路与 4.2.1 中 `register` 相同。不同在于 `is_stuff` 设置为 True。

4.2.3 顾客登录

顾客登录功能直接由 Django 框架中提供的的 login 相关功能实现，在此不再赘述。登陆后重定向到餐厅列表页面（即顾客主页）。

4.2.4 经理登录

经理登录功能由 managerlogin 函数实现，该函数处理经理的登录请求。简要代码与分析如下：

```
def managerlogin(request):
    if request.method == 'GET':
        return render(request, 'registration/managerlogin.html')
    elif request.method == 'POST':
        user_name = request.POST.get('username')
        pwd = request.POST.get('password')
        if not user_name or not pwd: # 信息不完整
            return render(request, 'registration/managerlogin.html', {'errors': ['xxx']})
        if not User.objects.filter(username=user_name): # 用户不存在
            return render(request, 'registration/managerlogin.html', {'errors': ['xxx']})
        for auser in User.objects.filter(username=user_name):
            if auser.is_staff == 0: # 用户存在但不是经理
                return render(request, 'registration/managerlogin.html', {'errors': ['xxx']})
        user = authenticate(request, username=user_name, password=pwd)
        if user is not None:
            login(request, user)
            return redirect('manager', pk=user.pk)
        else: # 信息错误
            return render(request, 'registration/managerlogin.html', {'errors': ['xxx']})
    else:
        return render(request, "login_fail.html")
```

实现步骤

1. 检查请求方法：
 - 如果请求方法是 GET，则返回登录页面。
 - 如果请求方法是 POST，则处理登录表单提交的数据。
2. 获取表单数据：
 - 从请求中获取用户名 (username) 和密码 (password)。
3. 验证表单数据：
 - 如果用户名或密码为空，返回登录页面并提示用户填写完整信息。
 - 如果用户名不存在于数据库中，返回登录页面并提示用户用户名不存在。
 - 检查用户是否为经理（即 is_staff 属性是否为 1），如果不是经理，返回登录页面并提示用户不是经理。

4. 用户认证:
 - 使用 authenticate 方法验证用户名和密码。
 - 如果认证成功, 调用 login 方法将用户登录。
 - 登录成功后, 重定向到经理的管理页面 (即经理主页)。
5. 重定向:
 - 如果用户名和密码验证失败, 返回登录页面并提示密码错误。

4.3 餐厅创建功能 (顾客/经理)

4.3.1 功能描述

该功能允许顾客或餐厅经理在列表中未找到所需餐厅时, 自行创建新的餐厅信息。具体功能包括:

1. 用户权限
 - 所有注册用户都可以创建餐厅
2. 信息录入
 - 用户可以输入餐厅的基本信息, 如名称、地址、营业时间等。
 - 提供详细的描述字段, 允许用户添加餐厅的特色、菜品类型、服务特点等。
3. 图片上传
 - 支持上传餐厅的图片, 展示餐厅环境和特色菜品, 提升用户体验。
4. 分类标签
 - 用户可以为餐厅添加分类标签, 如菜系类型 (中餐、西餐等)、适合场景 (家庭聚餐、商务宴请等), 方便其他用户检索。

通过这些功能, 用户不仅可以轻松创建和管理餐厅信息, 还能为其他用户提供更全面的餐厅选择和参考。这样不仅提升了用户体验, 也促进了平台内容的丰富和多样化。

4.3.2 代码实现

urlpattern

```
path('addrestaurant/', views.addrestaurant, name='addrestaurant')
```

视图函数

当 url 匹配到 'addrestaurant/' 时, 调用 addrestaurant 视图函数。

如果是 GET 请求, 则返回 AddRestaurantForm 表单, 如果是 POST 请求, 如果表单时有效的, 则保存到数据库中去。

```
def addrestaurant(request):  
    if request.method == 'POST':  
        form = AddRestaurantForm(request.POST, request.FILES)  
        if form.is_valid():  
            form.save() # 保存数据到数据库  
            return redirect('index')
```

```
else:  
    form = AddRestaurantForm  
    return render(request, 'catalog/addrestaurant.html', {'form': form})
```

form 的定义

- `Meta`: 类用于指定表单的元数据。
- `model = RESTAURANT`: 指定表单对应的模型是 `RESTAURANT`。
- `fields = '__all__'`: 表示表单将包含模型的所有字段。
- `widgets`: 定义了每个字段的显示小部件及其 HTML 属性。

```
class AddRestaurantForm(forms.ModelForm):  
    class Meta:  
        model = RESTAURANT  
        fields = '__all__'  
        widgets = {  
            'resta_name': forms.TextInput(attrs={'class': 'form-control'}),  
            ...  
            'manager': forms.HiddenInput(), # 设置为隐藏字段  
            'AVG_grade': forms.HiddenInput(), # 设置为隐藏字段  
        }  
  
    def __init__(self, *args, **kwargs):  
        super().__init__(*args, **kwargs)  
        self.fields['manager'].required = False # 设置为非必填  
        self.fields['manager'].initial = None # 设置初始值为空  
        ...
```

这里将 manager 和 AVG_grade 的值设置为空值，并且不再网页上显示

html 代码

这段使得只有已经登录的用户才能看到增加餐厅的按钮

```
{% if user.is_authenticated%}  
    <a href="{% url 'addrestaurant' %}">  
        <button type="button" class="button">增加餐厅</button></a>  
{% endif %}
```

4.4 餐厅认领与管理功能（经理）

4.4.1 功能描述

餐厅经理在登录后，可以进入管理餐厅页面，查看并管理自己所拥有的餐厅。此外，餐厅经理还可以申请认领其他餐厅，以便将其纳入自己的管理范围。具体功能包括：

1. 查看餐厅信息

- 餐厅经理可以查看自己所拥有餐厅的详细信息，包括餐厅名称、地址、营业时间、评分、标签等。

2. 编辑餐厅信息

- 餐厅经理可以编辑和更新餐厅的基本信息，如名称、地址、营业时间、特色标签等，确保信息的准确性和实时性。

3. 上传餐厅图片

- 支持上传和更新餐厅的图片，展示餐厅环境和特色菜品，提升用户体验。

4. 申请认领餐厅

- 餐厅经理可以通过系统提交申请，认领尚未被管理的餐厅。申请通过后，该餐厅将被添加到经理的管理列表中。

5. 审核机制

- 系统管理员将审核餐厅经理的认领申请，确保申请的合法性和合理性。审核通过后，餐厅经理将获得该餐厅的管理权限。

4.4.2 代码实现

urlpattern

```
path('manager/<int:pk>', views.manager, name='manager')
```

视图函数

```
@login_required
#检查登录情况
def manager(request, pk):
    manager = get_object_or_404(User, pk=pk)
    # 检查当前用户是否是指定的 manager
    if request.user != manager:
        # 跳转至 403 页面
        return render(request, 'catalog/403.html')
    restaurants = RESTAURANT.objects.filter(manager=manager)
    #筛选出 manager 字段为当前用户的 restaurant
    context = {'manager': manager, 'restaurants': restaurants}
    return render(request, 'catalog/manager.html', context)
```

认领餐厅的表单页面的视图函数中的一部分代码，这里将表单的 manager_reg.user_ID 设置为当前登录用户，就不需要用户自己进行选择，更加方便，也提升了安全性。

```
if form.is_valid():
    manager_reg = form.save(commit=False)
    manager_reg.user_ID = request.user # 设置 user_ID 为当前登录用户
    manager_reg.save()
    return redirect('index')
```

form 表单

```
class ApplyRestaurantForm(forms.ModelForm):
    resta_ID = forms.ModelChoiceField(
        queryset=RESTAURANT.objects.filter(manager__isnull=True),
        #先过滤掉 manager 字段不为空的的餐馆 (即已经被认领的餐馆)，剩下的选择为
        #未被认领的餐馆
        label="Restaurant"
    )
    class Meta:
        model = MANAGER_REG
        fields = ['resta_ID', 'evidence'] # 只包含需要填写的字段
```

4.5 餐厅检索、排序功能（顾客）

4.5.1 功能描述

该部分实现对餐厅和相关产品的筛选，对餐厅和产品实现一些检索和排序的功能，主要功能有以下几点

1. 关键词搜索

- 用户可以通过输入餐厅名称、菜品、或其他关键词（more_info 字段）来搜索餐厅。
- 若用户在搜索栏输入的是菜品名称(或菜品名称的一部分)，则会筛选出包含该菜品的餐厅以及该餐厅包含了搜索信息的菜品
- 若用户在搜索栏输入的是餐厅的名称(或餐厅名称的一部分)，则会筛选出对应的餐厅，并显示该餐厅的部分菜品
- 输入的信息也会更具餐馆的 more_info 字段进行匹配，如果输入的是一些其他信息，则也会输出 more_info 中包含了该信息的餐馆，并展示部分菜品

2. 位置搜索：

- 根据用户选择的位置，显示位于该位置附近的餐厅。

3. 评分排序

- 默认根据用户评价或评分从高到低对餐厅进行排序，帮助用户找到高评分的餐厅。

4. 营业时间筛选

- 根据餐厅的营业时间进行筛选，确保用户选择的餐厅在访问时是营业状态。
- 若当前时间不在餐厅的营业时间内，则列表中不会显示该餐厅的信息

5. 餐厅是否营业筛选

- 餐厅还有一个字段为 `isopen` 字段，在增加餐厅时 `isopen` 字段的值默认为 `True`
- 因为有些餐厅可能之前营业，但是现在可能倒闭了或者最近期间店主有事情导致最近不开张，则餐馆的经理可以将 `isopen` 的字段值设置为 `false`。
- 餐馆列表页面根据 `isopen` 进行筛选，如果 `isopen` 的值为 `false`，则不会显示在列表页面

6. 特色标签

- 根据餐厅一些标签（如 `drink`、`fastfood`、`spicy` 等）进行筛选。

4.5.2 代码实现

urlpattern

```
path('restaurants/', views.RestaurantListView.as_view(), name='restaurants'),
```

视图函数

- `get_queryset`: 获取并过滤查询集。
- `search_query`、`location`、`tag`: 从请求中获取搜索和筛选参数。
- `shanghai_timezone`: 获取上海时区。时区设置与筛选，因为要实现对营业时间的筛选，所以先设置时区为上海时区，再利用 `filter` 方法进行对餐馆的筛选
- `current_time`: 获取当前时间并转换为上海时区。
- **时间过滤**: 过滤餐馆的营业时间在当前时间范围内，或没有设置营业时间的餐馆。
- `isopen` **过滤**: 只显示营业中的餐馆。
- **搜索过滤**: 根据餐馆名称、菜品名称和更多信息进行搜索。
- **排序**: 按评分降序排序。
- **位置和标签过滤**: 根据位置和标签进行过滤。
- `get_context_data`: 获取并设置上下文数据。
- `search_query`、`location`、`tag`: 将搜索和筛选参数添加到上下文中。
- `location_choices`、`tag_choices`: 将位置和标签选项添加到上下文中。
- **菜品过滤**: 根据搜索内容过滤餐馆的菜品，默认显示最多 5 个在售菜品。

1. 过滤查询集:

```
queryset = queryset.filter(  
    Q(resta_name__icontains=search_query) |  
    Q(dish_dish_name__icontains=search_query) |  
    Q(more_info__icontains=search_query)  
).distinct()
```

这段代码使用 Django 的 `Q` 对象和 `filter` 方法来过滤查询集 `queryset`。具体过滤条件如下：

- `Q(resta_name__icontains=search_query)`: 检查餐馆名称 `resta_name` 是否包含搜索查询（不区分大小写）。
- `Q(dish_dish_name__icontains=search_query)`: 检查菜品名称 `dish_name` 是否包含搜索查询（不区分大小写）。这里假设 `dish` 是一个关联模型（如外键）。

- `Q(more_Info__icontains=search_query)`：检查餐馆的更多信息`more_Info`是否包含搜索查询（不区分大小写）。

这些条件之间使用`|`（逻辑或）运算符连接，表示只要满足其中一个条件，就会包含在过滤结果中。

2. 去重：

.distinct()

这行代码确保返回的结果是唯一的，即使某个餐馆同时满足多个条件，也只会在结果中出现一次。

全部代码概览：

```
class RestaurantListView(generic.ListView):
    .....
    paginate_by = 3#设置每页的餐厅个数
    def get_queryset(self):
        .....
        # 使用 ZoneInfo 获取上海时区
        shanghai_timezone = ZoneInfo('Asia/Shanghai')
        current_time = timezone.now().astimezone(shanghai_timezone)
        # 根据 time_open 和 time_close 内的当前时间进行筛选
        queryset = queryset.filter(
            Q(time_open__lte=current_time, time_close__gte=current_time) |
            Q(time_open__isnull=True) | Q(time_close__isnull=True)
        )
        queryset = queryset.filter(isopen=True)#筛选掉未营业的餐厅
        if search_query:
            queryset = queryset.filter(
                Q(resta_name__icontains=search_query) |
                Q(dish_dish_name__icontains=search_query) |
                Q(more_Info__icontains=search_query)
            ).distinct()
        #根据平均分进行排序
        queryset = queryset.order_by('-AVG_grade')
        .....
    def get_context_data(self, **kwargs):
        .....
        if search_query:
            for restaurant in context['restaurants']:
                if search_query.lower() in restaurant.resta_name.lower() or
search_query.lower() in restaurant.more_Info.lower():
                    # 搜索内容是餐厅名称时，显示最多 5 个菜品
                    restaurant.filtered_dishes = restaurant.dish_set.filter(onsale=True)[:5]
            else:
                # 否则，显示匹配搜索内容的菜品
                restaurant.filtered_dishes =
```

```
restaurant.dish_set.filter(dish_name__icontains=search_query, onsale=True)[:5]
else:
    # 没有搜索内容时，显示在售的菜品（最多五个）与上面代码类似
....
```

4.7 查看餐厅/菜品详情的功能

4.7.1 查看餐厅详情功能概述

顾客或经理可以在餐厅的详情页查看餐厅的相关信息，包括：餐厅的基本信息、餐厅拥有的菜品、针对该餐厅的评论、评论下的回复等。当餐厅已经被经理认领时，只有经理有权限编辑餐厅信息；当餐厅未被认领时，登录用户均有权限编辑餐厅信息。对于评论与回复板块，只有登录用户有权限发表评论与回复，而未登录用户没有。

4.7.2 查看餐厅详情具体实现

查看餐厅详情的功能由 `RestaurantDetailView` 视图类实现，该类基于 Django 的 `generic.DetailView`，并自定义了一些方法和属性来满足具体需求。以下是对 `RestaurantDetailView` 视图类的详细分析：

```
class RestaurantDetailView(generic.DetailView):
    model = RESTAURANT

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['restaurant_dishes'] = DISH.objects.filter(resta_ID=self.object.resta_ID, onsale=True)
        context['restaurant_comments_and_replies'] = COMMENT.objects.filter(resta_ID = self.object.resta_ID).prefetch_related('replies')
        restaurant = self.get_object()
        context['can_edit'] = restaurant.manager is None or self.request.user == restaurant.manager
        return context

    def test_func(self):
        restaurant = self.get_object()
        if restaurant.manager is None:
            return self.request.user.is_authenticated
        return self.request.user == restaurant.manager
```

实现步骤：

1. 继承 `generic.DetailView`

`RestaurantDetailView` 继承自 Django 的 `generic.DetailView`，指定了 `model` 为 `RESTAURANT`，这意味着该视图类用于显示单个 `RESTAURANT` 对象的详细信息。

2. 重写 `get_context_data` 方法

`get_context_data` 方法用于向模板上下文添加额外的数据，以便在模板中使用。

- 获取餐厅的菜品信息：

```
context['restaurant_dishes']  
= DISH.objects.filter(resta_ID=self.object.resta_ID, onsale=True)
```

通过过滤 `DISH` 模型中 `resta_ID` 与当前餐厅 ID 匹配且 `onsale` 为 `True` 的菜品，获取该餐厅在售的所有菜品，并添加到上下文中。

- 获取餐厅的评论及回复信息：

```
context['restaurant_comments_and_replies']  
= COMMENT.objects.filter(resta_ID = self.object.resta_ID).prefetch_related('replies')
```

通过过滤 `COMMENT` 模型中 `resta_ID` 与当前餐厅 ID 匹配的评论，获取该餐厅的所有评论，并预取相关联的回复信息，添加到上下文中。

- 添加用户权限信息到上下文：

```
context['can_edit']  
= restaurant.manager is None or self.request.user == restaurant.manager
```

获取当前餐厅对象，检查当前餐厅是否有经理。如果没有经理或当前用户是该餐厅的经理，则设置 `can_edit` 为 `True`，表示用户有权限编辑餐厅信息。

3. 自定义 `test_func` 方法

`test_func` 方法用于检查用户是否有权限查看或编辑餐厅信息。

- 检查权限：

```
if restaurant.manager is None:  
    return self.request.user.is_authenticated  
return self.request.user == restaurant.manager
```

如果餐厅没有经理，则只要用户已登录，就有权限查看或编辑餐厅信息。如果餐厅有经理，则只有当用户是该餐厅的经理时，才有权限查看或编辑餐厅信息。

4.7.3 查看菜品详情

功能与实现思路与 4.7.1, 4.7.2 几乎一致，故不再赘述

4.8 修改餐厅/菜品信息功能

4.8.1 修改餐厅信息

修改餐厅基本信息的功能由 `editrestaurants` 视图函数实现。以下是对 `editrestaurants` 视图函数的详细分析：

```
@login_required  
def editrestaurants(request, pk):  
    restaurant = get_object_or_404(RESTAURANT, pk=pk)  
  
    if request.method == 'POST':
```

```
resta_name = request.POST.get('resta_name')
xxx = request.POST.get('xxx')

...
restaurant = RESTAURANT.objects.get(resta_ID=pk)

if restaurant.manager is not None and request.user != restaurant.manager: # 权限检查
    raise PermissionDenied
restaurant.resta_name = resta_name
xxx = xxx
...

try:
    restaurant.full_clean() # 验证字段
    restaurant.save()
    return redirect('restaurant-detail', pk=restaurant.resta_ID)
except ...

return render(request, 'catalog/edit_restaurants.html', {'restaurant': restaurant})
```

实现步骤

1. 使用 @login_required

@login_required 确保只有已登录的用户才能访问此函数。如果用户未登录, Django 会自动重定向到登录页面。

2. 获取餐厅对象

使用 get_object_or_404 函数根据主键 (pk) 获取 RESTAURANT 对象。如果餐厅对象不存在, 会返回 404 错误。

3. 处理 POST 请求

检查请求方法是否为 POST, 如果是, 则处理表单提交的数据。

4. 获取表单数据

从 POST 请求中获取表单数据, 并根据键获取相应的值。

5. 权限检查

检查当前用户是否为餐厅的经理。如果餐厅已经有经理且当前用户不是经理, 则抛出 PermissionDenied 异常。

6. 更新餐厅信息

将获取的表单数据赋值给餐厅对象的相应字段。如果有上传图片, 则更新餐厅的图片字段。另外, 值得提的一点是, 前端表单中的默认值就是修改前的值, 故若用户没有进行任何修改, 直接提交表单, 数据库并不会因此发生变化。

7. 验证并保存餐厅对象

调用 full_clean 方法验证字段, 捕获可能的 ValidationError 异常。然后调用 save 方法将更新后的餐厅对象保存到数据库。

8. 处理 GET 请求

如果请求方法不是 POST, 则返回包含餐厅信息的编辑页面。

另外, 本系统中的约半数表单页均与该表单页有着相似的处理, 故其它类似的表单页不再赘述。

4.8.2 修改菜品信息

功能与实现思路与 4.8.1 几乎一致，故不再赘述

4.9 餐厅评分/评论/回复功能

4.9.1 用户评分/评论

同样使用表单进行，实现思路与 4.8.1 几乎一致，故不再赘述

4.9.2 餐厅/菜品综合评分计算

使用触发器实现对餐厅/菜品综合评分计算，具体代码如下：

```
DELIMITER $  
CREATE TRIGGER UPDATE_GRADE  
AFTER INSERT ON catalog_comment  
FOR EACH ROW  
BEGIN  
    DECLARE RESTA_AVG_GRADE FLOAT;  
    DECLARE DISH_AVG_GRADE FLOAT;  
    SELECT AVG(grade) INTO RESTA_AVG_GRADE FROM catalog_comment WHERE catalog_comment.resta_ID_id = NEW.resto_ID_id;  
    UPDATE catalog_restaurant SET AVG_grade = RESTA_AVG_GRADE WHERE resto_ID = NEW.resto_ID_id;  
    IF NEW.dish_ID_id != NULL THEN  
        SELECT AVG(grade) INTO DISH_AVG_GRADE FROM catalog_comment WHERE catalog_comment.dish_ID_id = NEW.dish_ID_id;  
        UPDATE catalog_dish SET AVG_grade = DISH_AVG_GRADE WHERE dish_ID = NEW.dish_ID_id;  
    END IF;  
END  
$ DELIMITER ;
```

每当有新的评论（内容中评分非空）插入，触发器都会重新计算涉及到餐厅/菜品的均分，然后对餐厅/菜品表进行更新。

4.9.3 回复功能

使用表单进行，实现思路与 4.8.1 几乎一致，故不再赘述

4.10 餐厅删除功能

考虑到在某些时候可能会出现餐厅老板卷钱跑路等等特殊情况，删除餐厅不能仅由餐厅经理进行，故在我们的系统中，普通的登录用户可以上传相关证据，联系超级管理员进行删除。

具体实现同样使用表单进行，实现思路与 4.8.1 几乎一致，故不再赘述

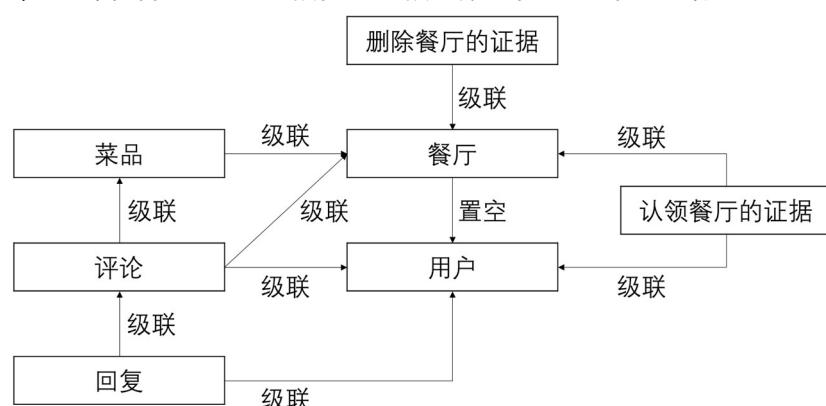
4.11 完整性概述

4.11.1 实体完整性

数据库的实体完整性检查由 Django 框架与 MySQL 自行完成。

4.11.2 参照完整性

各个基本表之间的参照关系与被参照表被删除后参照表的应对措施如图所示：



4.11.2 用户定义的完整性

1. 唯一性约束

○ RESTAURANT 模型

```
resta_name = models.CharField(max_length = 20, unique = True)
```

resta_name 字段必须在整个表中唯一，防止重复的餐厅名称出现。

○ DISH 模型

```
class Meta:  
    constraints = [  
        models.UniqueConstraint(fields=['resta_ID', 'dish_name'], name='unique_dish')  
    ]
```

在 resta_ID 和 dish_name 字段上设置了复合的唯一约束，确保同一餐厅内的菜品名称是唯一的。

2. 非空约束

见 3.2.1 节

4.12 安全性概述

4.12.1 用户认证

使用 Django 自带的用户认证系统，确保只有经过认证的用户才能访问系统中的某些功能和数据。视图函数中使用 `@login_required` 装饰器限制未认证用户的访问。

```
from django.contrib.auth.decorators import login_required

@login_required
def xxx(request):
    xxx
```

4.12.2 存取控制

通过权限控制来限制用户可以执行的操作。系统中的不同用户角色（如普通用户、餐厅经理、管理员）具有不同的权限。

```
from django.contrib.auth.models import User

def test_func(self):
    restaurant = self.get_object()
    if restaurant.manager is None:
        return self.request.user.is_authenticated
    return self.request.user == restaurant.manager
```

4.12.3 数据加密

Django 使用 PBKDF2 加密算法对用户密码进行哈希处理并加盐存储，增强密码的安全性。如图，用户密码不会以明文形式存储在数据库中。

| id | password |
|------|---|
| 1 | pbkdf2_sha256\$390000\$WK9172nzXdmTjMjFVc33um\$YTK0... |
| 2 | pbkdf2_sha256\$390000\$Qr8t82rnduZgGgsP0kWTjf\$gjdpER... |
| 3 | pbkdf2_sha256\$390000\$sIhfnmCvAgrC76RkQN2kVp\$gfjzw... |
| 4 | pbkdf2_sha256\$390000\$uoMXIQUWY3uhSTPw26p8mH\$CuP... |
| 5 | pbkdf2_sha256\$390000\$tijsa2pk19f201R51F1l5Z\$VRvCIws... |
| NULL | NULL |

5. 项目测试

5.1 功能测试

5.1.1 注册

1. 若昵称重复，则无法注册

The screenshot shows a registration form titled "Register 注册". It has two input fields: "Username:" containing "Harry" and "Password:" containing "*****". A red rectangular box at the top of the page contains the text "该昵称已存在，请换一个昵称" (The nickname already exists, please choose another nickname). At the bottom of the form is a red "Submit 提交" button.

2. 信息不完整，无法注册

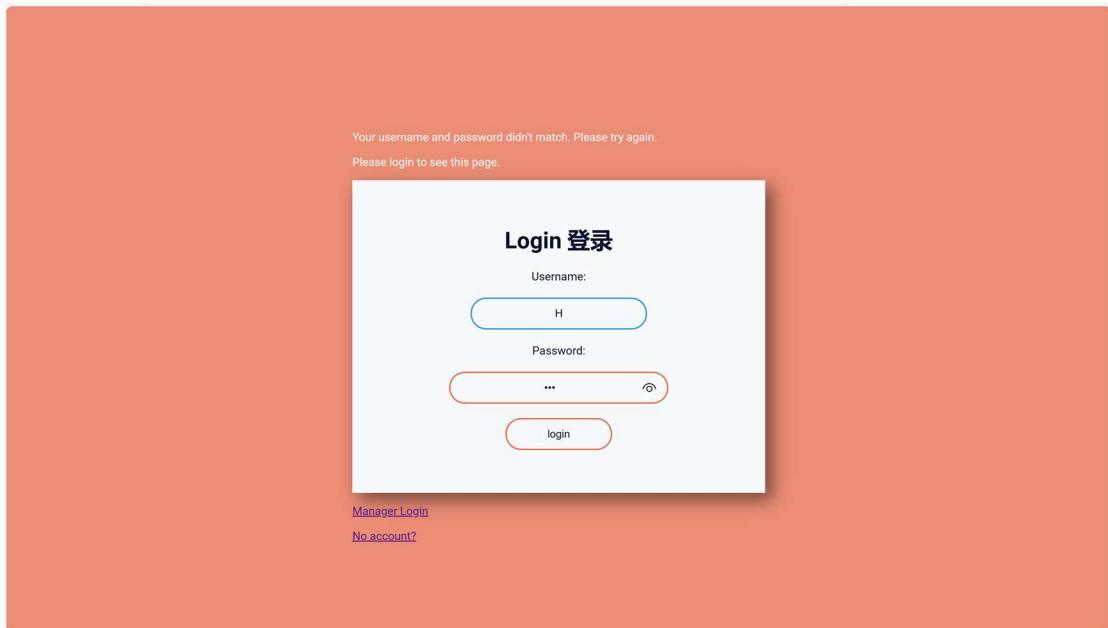
The screenshot shows a registration form titled "Register 注册". It has two input fields: "Username:" containing "12345" and "Password:" containing "Passwdrd 密码". A red rectangular box covers the "Password:" field, with an orange exclamation mark icon and the text "请填写此字段。" (Please fill in this field.). At the bottom of the form is a red "Submit 提交" button.

3. 注册成功，相关信息存入数据库

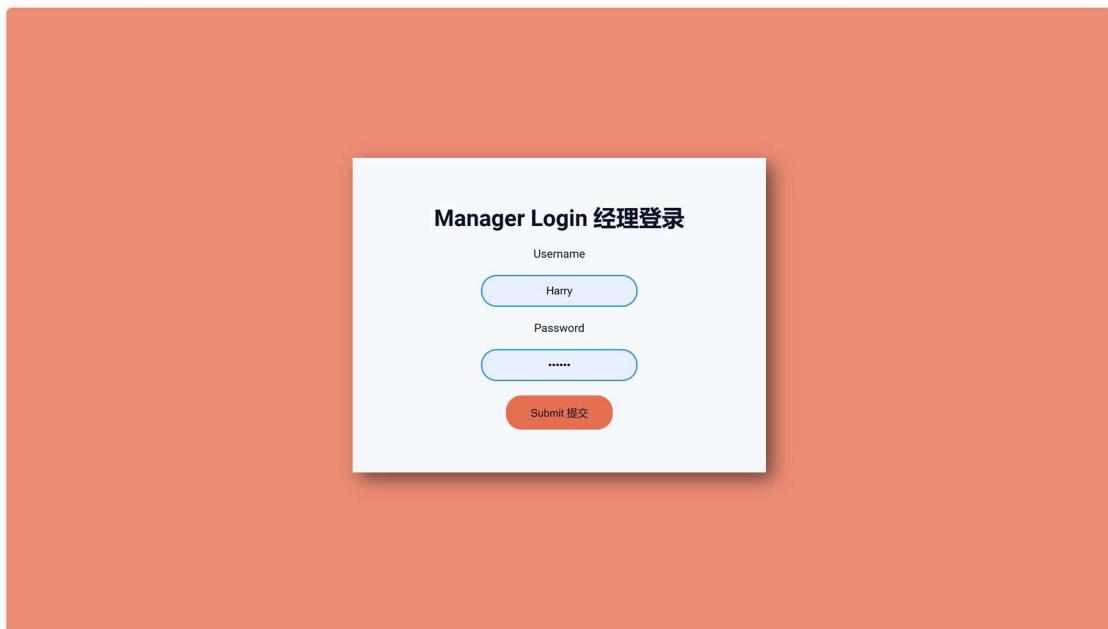
| id | password | last_login | is_superuser | username |
|----|---|----------------------------|--------------|------------|
| 1 | pbkdf2_sha256\$390000\$WK9172nzXdmTjMjFVc33um\$YTK0... | 2024-06-19 09:24:07.987282 | 1 | imlrz |
| 2 | pbkdf2_sha256\$390000\$Qr8t82rnduZgGgsP0kWTjf\$gjdpER... | 2024-05-17 13:48:10.438045 | 0 | wangwenyan |
| 3 | pbkdf2_sha256\$390000\$slhfrmCvAgrC76RkQN2kVp\$gfjzw... | 2024-07-15 07:14:51.277686 | 0 | Harry |
| 4 | pbkdf2_sha256\$390000\$uoMXIQUWY3uhSTPw26p8mH\$CuP... | HULL | 0 | 你说得对 |
| 5 | pbkdf2_sha256\$390000\$tijea2pk19f201R51F1J5Z\$VRvCIws... | 2024-06-19 09:22:51.857921 | 0 | m |
| 6 | pbkdf2_sha256\$390000\$xkrb9HcpX5pYcTL15UjhJW\$irMnIZo... | HULL | 0 | 12345 |

5.1.2 登录

1. 账号/密码错误，无法登录



2. 普通用户无法作为经理登录



3. 顾客登陆成功，跳转至顾客首页，左侧显示登录信息

The screenshot shows a customer homepage with a sidebar on the left containing 'Home', 'User: Harry', and 'Logout' links. The main content area is titled 'Restaurant Search'. It features a search bar with placeholder 'Search for restaurants or dishes...', dropdown menus for 'All Locations' and 'All Tag', and two buttons: 'Search' and '增加餐厅' (Add Restaurant). Below the search bar is a card for '麦当劳 (合肥曙光餐厅)'. The card includes a thumbnail image of the restaurant, its name, address ('Address: East Campus'), average grade ('Average Grade: 2.16'), and more info ('More Info: 金寨路136号'). A section labeled 'Dishes:' shows three small images of food items: a sandwich, a burger, and another sandwich. Below these images are their names and prices: '123 - \$123.0' for the sandwich, '随心配1+1超值套餐 - \$13.9' for the burger, and '麦辣鸡腿堡中套餐 - \$33.5' for the other sandwich. At the bottom of the page is a footer note 'Page 1 of 1.'

4. 经理登陆成功，跳转至经理首页，显示拥有的餐厅

The screenshot shows a manager homepage with a sidebar on the left containing 'Home', 'User: m', and 'Logout' links. The main content area is titled 'm's Restaurants'. It features a card for '麦当劳 (合肥曙光餐厅)'. The card includes a thumbnail image of the restaurant, its name, address ('Address: East Campus'), average grade ('Average Grade: 2.16'), and more info ('More Info: 金寨路136号'). To the right of the card is a dark button labeled '申请餐厅' (Apply for Restaurant).

5.1.3 餐厅检索

在页面上方是相关的检索栏以及筛选条件，用户可以根据需要进行筛选，界面的主要部分是餐馆列表，能够查看餐馆的大致信息以及所包含的产品菜品，且默认按照评分对餐馆进行排序，在没有进行筛选前回显示所有的餐馆信息，并进行了分页，点击 next 和 last 可以跳转至对应得页数。并且每个餐馆默认展示部分菜品的信息（最大数量为 5）

这些是目前数据库中拥有的餐厅的一些信息

| resta_ID | resta_name | location | time_open | time_close | tag | more_Info | manager_id | Avg_grade | image | isopen |
|----------|--------------|----------|-----------------|-----------------|------|---------------------|------------|-----------|-------------------------------|--------|
| 2 | KFC | J | 08:00:00.000000 | 23:00:00.000000 | 1 | 合肥市蜀山区金寨路鸿基商务大... | NULL | 4.5 | restaurants/KFC.jpg | 1 |
| 3 | 卡旺卡（宏基广场网店店） | J | 10:00:00.000000 | 21:30:00.000000 | 1 | 合肥市蜀山区金寨路鸿基商务大... | NULL | 1.5 | restaurants/卡旺卡.jpg | 1 |
| 4 | 米雪冰城 | E | 08:00:00.000000 | 23:00:00.000000 | 1 | 黄山路441号百盛16幢商铺101 | 1 | 5 | restaurants/蜜雪冰城.jpg | 1 |
| 5 | 瑞幸咖啡 | J | 07:30:00.000000 | 20:00:00.000000 | C | 合肥市蜀山区金寨路91号立基... | NULL | 4 | restaurants/瑞幸.jpg | 1 |
| 6 | 凤渝祥烤鱼重庆鸡公煲 | H | 09:00:00.000000 | 22:00:00.000000 | S | 蜀山区黄山路439号综合楼1-2... | NULL | NULL | restaurants/24ccbdbd2a2944... | 1 |
| 7 | 麦当劳合肥曙光餐厅 | J | 00:00:00.000000 | 23:59:00.000000 | F | 金寨路136号 | 1 | NULL | restaurants/MacDonald.jpg | 1 |
| 8 | 某未营业烤肉点 | W | 00:00:00.000000 | 23:59:00.000000 | S | 肥西路 | NULL | NULL | restaurants/屏幕截图_2023-12-... | 0 |
| 9 | 米雪冰 | O | NULL | NULL | NULL | NULL | NULL | NULL | restaurants/404.png | 1 |

餐厅列表分页显示出位于营业时间的并且目前在在营业(is_open=1, 因为可能餐厅老板因为家里的或者其他事情导致最近几天不营业)的餐厅, 可以看到这里共显示了七个餐厅,而没有显示“某未营业烤肉点”, 截图时间未 13: 58, 其他七个餐厅均处于营业时间, 所以都能显示

The screenshot shows a user profile on the left with "User: 86186" and "Logout" options. The main area displays three restaurant cards:

- Luckin Coffee**: Address: East Campus, Average Grade: 5.0, More Info: 黄山路441号百盛16幢商铺101. Dishes: (Image of a coffee cup).
- KFC**: Address: Jinzhai Road, Average Grade: 4.5, More Info: 合肥市蜀山区金寨路鸿基商务大楼1幢1层. Dishes: (Image of a meal).
- Mcdonald's**: Address: Jinzhai Road, Average Grade: None, More Info: 金寨路136号. Dishes: (Image of a meal).

At the bottom right, there are "next" and "last" navigation buttons.

The screenshot shows a user profile on the left with "User: 86186" and "Logout" options. The main area displays three restaurant cards:

- KFC**: Address: Jinzhai Road, Average Grade: 4.5, More Info: 合肥市蜀山区金寨路鸿基商务大楼1幢1层. Dishes: (Image of a meal).
- 凤渝祥烤鱼重庆鸡公煲**: Address: Huangshan Road, Average Grade: None, More Info: 蜀山区黄山路439号综合楼1-2层3号. Dishes: (Image of a meal).
- 麦当劳合肥曙光餐厅**: Address: Jinzhai Road, Average Grade: None, More Info: 金寨路136号. Dishes: (Image of a meal).

The screenshot shows a user profile on the left with "User: 86186" and "Logout" options. The main area displays one restaurant card:

- 米雪冰**: Address: Other places, Average Grade: None, More Info: None. Dishes: (Image of a meal).

At the bottom right, there are "first", "previous", and "next" navigation buttons.

若此时将时区改为洛杉矶时区

```
shanghai_timezone = ZoneInfo('America/Los_Angeles')#('Asia/Shanghai')
current_time = timezone.now().astimezone(shanghai_timezone)
```

现在的洛杉矶州洛杉矶时间

23:33:43

2024年7月20日星期六，第29周

International Chess Day / International Ride MTB Day

太阳： + 05:55 ± 20:04 (1小时 8分钟) - 更多信息 - 将洛杉矶时间设为默认 - 取消关注此位置

| | | | | |
|--------------|-------------|-------------|-------------|-------------|
| 洛杉矶 23:33 | 伦敦 07:33 | 巴黎 08:33 | 北京 14:33 | 东京 15:33 |
|--------------|-------------|-------------|-------------|-------------|

那么只会显示如下两个餐厅

The screenshot shows a user interface for a restaurant search. On the left, there is a sidebar with 'Home', 'User: 86186', and 'Logout' buttons. The main area has a title 'Restaurant Search' and a search bar with placeholder 'Search for restaurants or dishes...'. Below the search bar are dropdown menus for 'All Locations' and 'All Tag', and buttons for 'Search', '管理餐厅' (Manage Restaurants), and '增加餐厅' (Add Restaurant). The search results are displayed in two cards:

- 麦当劳合肥曙光餐厅**
Address: Jinzhai Road
Average Grade: None
More Info: 金寨路136号
Dishes:
- 米雪冰**
Address: Other places
Average Grade: None
More Info: None
Dishes:

At the bottom right of the main area, it says 'Page 1 of 1.'

餐厅默认按照评分进行排序，若对餐厅进行搜索或者筛选，得到的结果也是按照评分进行排序的

This screenshot shows the same user interface as the previous one, but the search results are now sorted by rating. The cards for the three restaurants are:

- 雪冰城**
Address: East Campus
Average Grade: 5.0
More Info: 黄山路441号百盛苑10幢商铺101
Dishes:
- KFC**
Address: Jinzhai Road
Average Grade: 4.5
More Info: 合肥市蜀山区金寨路鸿基商务大楼1幢1层
Dishes:
- 瑞幸咖啡**
Address: Jinzhai Road
Average Grade: 4.0
More Info: 合肥市蜀山区金寨路91号立基大厦底
Dishes:

若按照餐厅名字进行搜索，则会筛选出对应的餐厅，并显示最多五个该餐厅拥有的菜品

The screenshot shows the 'Restaurant Search' page. On the left, there's a sidebar with 'Home', 'User: 86186', and 'Logout'. The main area has a search bar with '卡旺卡' entered, and dropdowns for 'All Locations' and 'All Tag'. Buttons for 'Search', '管理餐厅' (Manage Restaurant), and '增加餐厅' (Add Restaurant) are at the bottom. Below the search bar, a card for '卡旺卡 (宏基广场网通店)' is displayed, showing its address, average grade, and more info. A section titled 'Dishes:' shows five small images of dishes: 手剥水蜜桃, 水蜜桃酸奶奶昔, 芒果酸奶奶昔, 芒芒云顶, and 芒芒冰椰. Below these images are their names and prices: \$10.0, \$19.0, \$19.0, \$19.0, and \$18.0 respectively. At the bottom of the card, it says 'Page 1 of 1.'

若按照菜品进行搜索，则会显示含有该菜品的餐厅，并且显示该餐厅拥有的对应得菜品

The screenshot shows the 'Restaurant Search' page. On the left, there's a sidebar with 'Home', 'User: 86186', and 'Logout'. The main area has a search bar with '茉莉' entered, and dropdowns for 'All Locations' and 'All Tag'. Buttons for 'Search', '管理餐厅' (Manage Restaurant), and '增加餐厅' (Add Restaurant) are at the bottom. Below the search bar, a card for 'KFC' is displayed, showing its address, average grade, and more info. A section titled 'Dishes:' shows one small image of a dish: 茉莉绿雪, with the price \$5.0. Below the image, it says '茉莉绿雪 - \$5.0'. Another card for '卡旺卡 (宏基广场网通店)' is shown below, with its details and a small image of a dish: 茉莉绿雪, with the price \$5.0.

这里只会搜索到目前正在销售的菜品 (onsale=1), 比如搜索芒果, 由于芒果爽目前是不上架 (onsale=0), 所以不会在 list 页面显示

| Result Grid | | | | | | | | |
|-------------|---------|-----------|-----|-------|--------------------------|--------|-----------|-------------|
| | dish_ID | dish_name | tag | price | image | onsale | more_Info | resta_ID_id |
| ▶ | 1 | 茉莉绿雪 | 1 | 5 | dishes/茉莉绿雪.jpg | 1 | 好喝 | 2 |
| | 2 | 茉莉绿雪 | 1 | 5 | dishes/茉莉绿雪_UYusiXe.jpg | 1 | 好喝 | 3 |
| | 3 | 芒果爽 | 1 | 5 | dishes/芒果爽.jpg | 0 | 好喝 | 3 |
| | 4 | 水蜜桃酸奶奶昔 | D | 19 | dishes/水蜜桃酸奶奶昔.png | 1 | NULL | 3 |
| | 5 | 手剥水蜜桃 | D | 18 | dishes/手剥水蜜桃.png | 1 | NULL | 3 |
| | 6 | 芒芒冰椰 | D | 18 | dishes/手剥水蜜桃_Ty9vAzW.png | 1 | NULL | 3 |
| | 7 | 芒芒云顶 | D | 19 | dishes/芒芒云顶.png | 1 | NULL | 3 |
| | 8 | 芒果酸奶奶昔 | D | 19 | dishes/芒果酸奶奶昔.png | 1 | NULL | 3 |

The screenshot shows the 'Restaurant Search' page. On the left, there's a sidebar with 'Home' and 'Login'. The main area has a search bar with '芒果' entered, and dropdowns for 'All Locations' and 'All Tag'. Buttons for 'Search', '管理餐厅' (Manage Restaurant), and '增加餐厅' (Add Restaurant) are at the bottom. Below the search bar, a card for '卡旺卡 (宏基广场网通店)' is displayed, showing its address, average grade, and more info. A section titled 'Dishes:' shows one small image of a dish: 芒果酸奶奶昔, with the price \$19.0. Below the image, it says '芒果酸奶奶昔 - \$19.0'. At the bottom of the card, it says 'Page 1 of 1.'

同时餐厅也有一个 **more_info** 字段，也可以根据该字段所拥有一些信息进行搜索

The screenshot shows a user interface for searching restaurants. On the left is a sidebar with 'Home', 'User: 86186', and 'Logout'. The main area has tabs for 'Jiushan District' and 'All Locations'. A search bar shows 'All Tag'. Buttons for 'Search', 'Manage Restaurant', and 'Add Restaurant' are at the top right. Three restaurant cards are displayed:

- KFC**: Address: Jinzhai Road, Average Grade: 4.5, More Info: 合肥市蜀山区金寨路鸿基商务大楼1幢1层. Dishes: 茉莉绿雪 - \$5.0 (with a small image)
- luckin coffee**: Address: Jinzhai Road, Average Grade: 4.0, More Info: 合肥市蜀山区金寨路91号立基大厦底层. Dishes: (empty)
- 卡旺卡 (宏基广场网通店)**: Address: Jinzhai Road, Average Grade: 1.5, More Info: 合肥市蜀山区金寨路鸿基商务大楼1幢1层. Dishes: (empty)

按 **location** 进行筛选，比如查找黄山路的餐厅

The screenshot shows a user interface for searching restaurants. On the left is a sidebar with 'Home', 'User: 86186', and 'Logout'. The main area has tabs for 'Restaurant Search' and 'Huangshan Road'. A search bar shows 'Huangshan Road' and 'All Tag'. Buttons for 'Search', 'Manage Restaurant', and 'Add Restaurant' are at the top right. One restaurant card is displayed:

- 凤渝样烤鱼重庆鸡公煲**: Address: Huangshan Road, Average Grade: None, More Info: 蜀山区黄山路439号综合楼1-2层3号. Dishes: 烤鱼 - \$54.0, 鸡公煲 - \$20.0 (with small images)

按照**标签 tag** 进行筛选

The screenshot shows a user interface for searching restaurants. On the left is a sidebar with 'Home', 'User: 86186', and 'Logout'. The main area has tabs for 'Restaurant Search' and 'Good drink'. A search bar shows 'Good drink' and 'All Locations'. Buttons for 'Search', 'Manage Restaurant', and 'Add Restaurant' are at the top right. Three restaurant cards are displayed:

- 米雪冰城**: Address: East Campus, Average Grade: 5.0, More Info: 黄山路441号百盛苑16幢商铺101. Dishes: (empty)
- KFC**: Address: Jinzhai Road, Average Grade: 4.5, More Info: 合肥市蜀山区金寨路鸿基商务大楼1幢1层. Dishes: 茉莉绿雪 - \$5.0 (with a small image)
- 卡旺卡 (宏基广场网通店)**: Address: Jinzhai Road, Average Grade: 1.5, More Info: 合肥市蜀山区金寨路鸿基商务大楼1幢1层. Dishes: (empty)

这三种搜索方法也可以一起选，会筛选出满足三种条件的餐厅

The screenshot shows a search interface with three filters selected: 'Jinzhai Road', 'Good drink', and 'KFC'. The results page displays two restaurants: KFC and 卡旺卡 (宏基广场网通店). For each restaurant, it shows the logo, address, average grade, more info, dishes, and their prices.

| Dish | Price |
|------------|--------|
| KFC 茉莉绿雪 | \$5.0 |
| 卡旺卡 手剥水蜜桃 | \$18.0 |
| 卡旺卡 水蜜桃酸奶昔 | \$19.0 |
| 卡旺卡 芒果酸奶奶昔 | \$19.0 |
| 卡旺卡 芒芒云顶 | \$19.0 |
| 卡旺卡 芒芒冰椰 | \$18.0 |

5.1.5 餐厅详情页

- 对于未注册用户，只显示基本信息、餐厅拥有的菜品和评论回复。点击相应的菜品可跳转至菜品详情页。

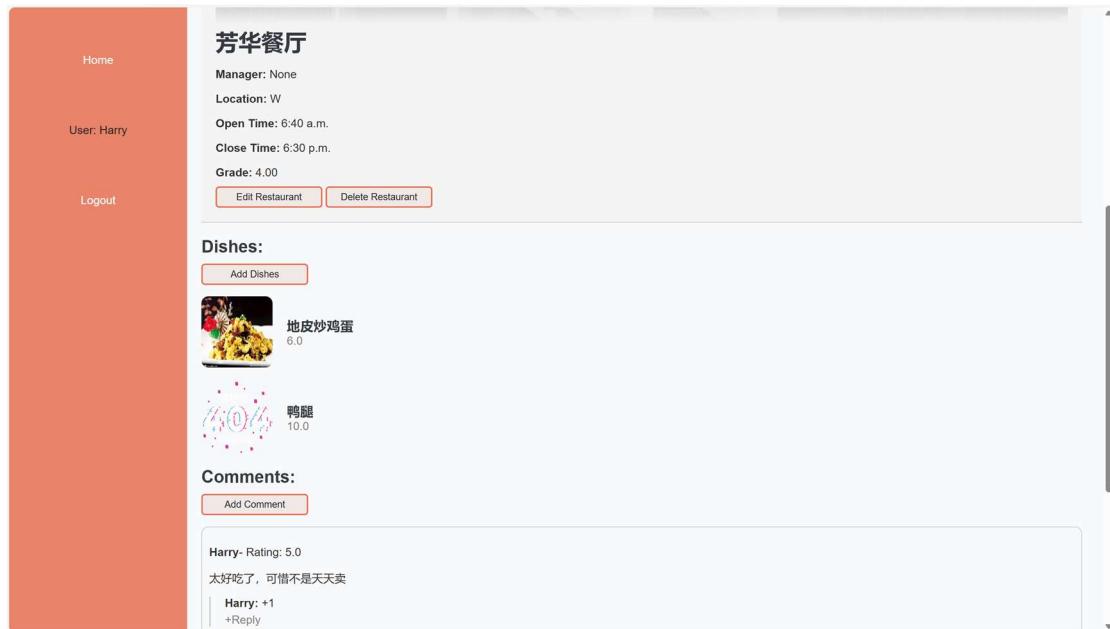
The screenshot shows a detailed view of a McDonald's restaurant. It includes a large image of the exterior, basic information like manager name and location, a list of dishes with small images and prices, and a comments section with user reviews and ratings.

| Dish | Price |
|------------|-------|
| 123 | 123.0 |
| 随心配1+1超值套餐 | 13.9 |
| 麦辣鸡腿堡中套餐 | 33.5 |

Comments:

Harry- Rating: 5.0
穷鬼套餐YYDS！！！
你说得对：你说得对
Harry: 123

2. 对于注册用户，页面中会显示删除餐厅按钮、编辑餐厅信息（无经理时）、添加菜品按钮（无经理时）和评论回复按钮



5.1.6 菜品详情页

1. 对于未注册用户，只显示基本信息和评论回复。点击餐厅名称可跳转至餐厅详情页。



2. 对于注册用户，页面中会显示删除餐厅按钮（无经理时）、编辑餐厅信息（无经理时）和评论回复按钮

The screenshot shows a restaurant detail page. On the left, a sidebar for user 'Harry' includes 'Home', 'User: Harry', and 'Logout'. The main content area features a large image of a dish, with the name 'Name: 地皮炒鸡蛋' displayed below it. Below the name are details: 'Restaurant: 芳华餐厅', 'Tag: None', 'Price: 6.0', and 'more_Info: 地皮：一种菌'. There are 'Edit Dish' and 'Delete Dish' buttons. A 'Comments:' section follows, with a 'Add Comment' button. It contains two comments: one from 'Harry' rating 5.0 and another from 'imlrz' rating 3.0, each with a '+Reply' link.

5.1.7 创建餐厅

在打开网站后，在搜索栏处有一个“增加餐厅”的按钮，点击后能够到达下面的表单页面

The screenshot shows a 'Restaurant Search' interface. The sidebar for user '86186' includes 'Home', 'User: 86186', and 'Logout'. The main area has a search bar with placeholder 'Search for restaurants or dishes...', dropdowns for 'All Locations' and 'All Tag', and 'Search', '管理餐厅', and '增加餐厅' buttons. Two restaurant entries are listed: '米雪冰城' (with a snowman logo) and 'KFC' (with the KFC logo). Each entry shows address, average grade, and more info. Below each entry is a 'Dishes:' section with a thumbnail image and a price.

Add Restaurant

Resta name:

Location: Other places

Time open:

Time close:

Tag:

More Info:

Image:

Isopen:

这个表单引用了 class RESTAURANT(models.Model)中的字段。登录以后才能看到“增加餐厅”的按钮。

Restaurant Search

Search for restaurants or dishes... All Locations All Tag

米雪冰城
Address: East Campus
Average Grade: 5.0
More Info: 黄山路441号百盛苑16幢商铺101
Dishes:

Restaurant Search

Search for restaurants or dishes... All Locations All Tag 增加餐厅

米雪冰城
Address: East Campus
Average Grade: 5.0
More Info: 黄山路441号百盛苑16幢商铺101
Dishes:

填写相关信息后，点击“Upload”就会跳转到 Home 页面，可以看到餐馆正确显示了

Add Restaurant

Resta name: 凤渝鲜烤鱼重庆公煲

Location: Huangshan Road

Time open: 09:00

Time close: 22:00

Tag: Spicy

More Info: 蜀山区黄山路439号综合楼1-2层3号

Image: 24cdcbdd2a294457af02d20fb065c.jpg

Isopen:

The screenshot shows a user interface for searching restaurants or dishes. On the left, there's a sidebar with 'Home', 'User: 86186', and 'Logout'. The main area has a search bar with placeholder text 'Search for restaurants or dishes...', dropdown menus for 'All Locations' and 'All Tag', a 'Search' button, and two buttons for '管理餐厅' (Manage Restaurant) and '增加餐厅' (Add Restaurant). Two restaurant cards are displayed:

- 卡旺卡 (宏基广场网通店)**
Address: Jinzhai Road
Average Grade: 1.5
More Info: 合肥市蜀山区金寨路鸿基商务大楼1幢1层
Dishes:

茉莉绿雪 - \$5.0
- 凤渝祥烤鱼重庆鸡公煲**
Address: Huangshan Road
Average Grade: None
More Info: 蜀山区黄山路439号综合楼1-2层3号
Dishes:

餐馆的名字是 primary key，所以如果重复提交则会显示下面的报错信息

- Restaurant with this Resta name already exists.

5.1.8 认领餐厅

在登录后，若用户为某个餐厅的经理，则会自动跳转到到管理餐厅的页面

The screenshot shows a user profile for '86186's Restaurants'. On the left, there's a sidebar with 'Home', 'User: 86186', and 'Logout'. The main area lists managed restaurants:

- 米雪冰城**
Location: East Campus
Open: 8 a.m. - Close: 11 p.m.
Average Grade: 5.0
More Info: 黄山路441号百盛苑16幢商铺101
Is Open: True

- 麦当劳合肥曙光餐厅**
Location: Jinzhai Road
Open: midnight - Close: 11:59 p.m.
Average Grade: None

若要对某个餐厅进行修改，可以点击要修改的餐厅，具体操作可以查看“餐厅/菜品信息修改功能”

The screenshot shows a user profile on the left with "User: 86186" and "Logout" options. The main content area displays two restaurant entries:

- 米雪冰城**
Location: East Campus
Open: 8 a.m. - Close: 11 p.m.
Average Grade: 5.0
More Info: 黄山路441号百盛苑16幢商铺101
Is Open: True
Image: A cartoon snowman holding an ice cream cone.
- 麦当劳合肥曙光餐厅**
Location: Jinzhai Road
Open: midnight - Close: 11:59 p.m.
Average Grade: None

这里的选项为未被认领的餐厅，user_id 会不自动填充未当前的用户，因此不需要填

The screenshots show the "认领餐厅" (Claim Restaurant) feature. On the left is a sidebar with "Home", "User: 86186", and "Logout".

The top screenshot shows a dropdown menu titled "Restaurant:" with several options listed, including KFC, 卡旺卡 (宏基广场网酒店), 瑞幸咖啡, 凤渝拌海鱼重庆鸡公煲, 茶香营业纯内点, and 米雪冰. An "Upload" button is at the bottom right of the dropdown.

The bottom screenshot shows a form titled "认领餐厅" with fields for "Restaurant:" (set to "凤渝拌海鱼重庆鸡公煲") and "Evidence:" (with a file input field containing "选择文件 屏幕截图 2023-05-13 123303.png"). An "Upload" button is at the bottom right.

点击 upload 之后，就会存储在数据库中了

| | reg_ID | evidence | resta_ID_id | user_ID_id |
|---|--------|----------------------------------|-------------|------------|
| ▶ | 1 | evidences/restaurants/Saul_Go... | 2 | 1 |
| | 5 | evidences/restaurants/1_7oFzh... | 7 | 1 |
| | 8 | evidences/restaurants/屏幕截图... | 3 | 1 |
| * | 9 | evidences/restaurants/屏幕截图... | 6 | 1 |
| | NULL | NULL | NULL | NULL |

超级管理员在审核后，可以进行授权，将 manager 字段授权给用户 86186 后，可以看到在所管理的餐厅中就能显示该餐厅了



5.1.9 编辑餐厅信息

餐厅经理或已注册用户（无经理时）可以编辑餐厅的相关信息通过填写表单完成

1. 填写信息不完整时，会给出提示信息

The screenshot shows a 'Edit Restaurant' form titled '修改餐厅信息'. The form has several input fields:

- 'Restaurant Name:' field containing '苏华餐厅'.
- 'Location:' dropdown set to 'West Campus'.
- 'Time Open:' field with two time inputs. The first input has a red border and contains '0:--' with a warning message '请填写此字段.' above it. The second input contains '--:--'.
- 'Tag:' dropdown set to 'Spicy'.
- 'More Info:' text input field containing 'None'.
- 'Upload Image:' button with placeholder '选择文件 未选择文件'.
- 'Is Open:' checkbox which is checked.

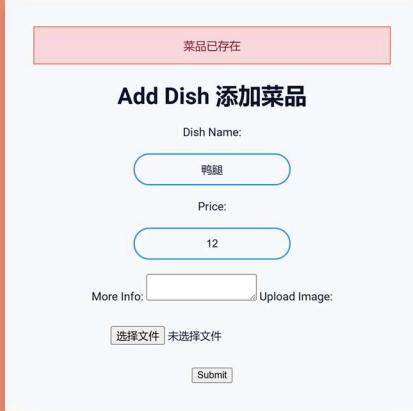
The 'Submit' button is located at the bottom right of the form.

2. 正确填写后跳转至餐厅详情页，信息已正确修改

5.1.10 添加菜品/编辑菜品

餐厅经理或已注册用户（无经理时）可以在餐厅详情页中通过填写表单添加菜品

- 若菜品名称重复（在该餐厅范围内），则无法添加（这是我们自行定义的完整性规则，见4.11.2）



The screenshot shows a 'Add Dish' form with a red header bar containing the text '菜品已存在' (Dish already exists). Below the header, the form has fields for 'Dish Name:' (输入框内为 '脆腿') and 'Price:' (输入框内为 '12'). There is also a 'More Info:' field with a file input and an 'Upload Image:' button. A 'Submit' button is at the bottom.

- 若信息合法，则填写完成后跳转至餐厅详情页，菜品此时已经被添加

5.1.11 评论

通过填写表单进行，可以选择针对菜品或针对整个餐厅



The screenshot shows a 'Add Comment for 麦当劳 (合肥曙光餐厅)' form. It includes a 'Grade:' field with a value of '4', a 'Content:' text area containing '12345', a 'Dish (optional)' dropdown menu with options like 'None', 'None', '123', '随心配1+1超值套餐', and '麦辣鸡腿堡中套餐', and a 'Submit' button.

5.1.12 回复



5.1.13 删除餐厅

任何注册用户可以通过提交图片证据，联系超级管理员删除餐厅



超级管理员可以在管理界面看到相关信息

Select delet e_restas to change

Action: Go 0 of 3 selected

- [DELETE E_RESTA](#)
- [3-test](#)
- [2-麦当劳 \(合肥曙光餐厅\)](#)
- [1-麦当劳 \(合肥曙光餐厅\)](#)

3 delet e_restas

5.2 完整性测试

5.2.1 实体完整性

实体的 id 由 django 自行编号, 实体完整性可以自动满足。以 restaurant 为例, 如下图所示。

| | resta_ID | resta_name |
|---|----------|--------------|
| ▶ | 1 | 麦当劳 (合肥曙光餐厅) |
| | 2 | 芳华餐厅 |
| | 3 | test |
| * | NULL | NULL |

5.2.2 参照完整性

在 models.py 中定义了各被参照关系被删除时, 参照关系采取的动作。具体定义见 4.11.2 以删除 Dish 和 Manager 为例:

1. 删除 Dish

删除前:

| | dish_ID | dish_name | | comm_ID | grade | content | dish_ID_id |
|---|---------|------------|---|---------|-------|--------------------------|------------|
| ▶ | 1 | 麦辣鸡腿堡中套餐 | ▶ | 1 | 5 | 穷鬼套餐YYDS!!! | 3 |
| | 3 | 随心配1+1超值套餐 | | 2 | 2 | 不裹鳞片的炸鸡不是好炸鸡:(转粉KFC了 | NULL |
| | 4 | 地皮炒鸡蛋 | | 3 | 5 | 太好吃了, 可惜不是天天卖 | 4 |
| | 5 | 鸭腿 | | 4 | 4 | 众筹开一个鲜榨果汁窗口(参考东区美广) | NULL |
| | 8 | 123 | | 5 | 0.8 | WOW | 3 |
| | 9 | 12 | | 6 | 0 | 麦当劳不如肯德基(5毛1条, 记得把括号删... | NULL |
| | 10 | 123 | | 7 | 3 | 可是好贵 | 4 |
| | 11 | 321 | | 8 | 3 | 123 | NULL |
| * | NULL | NULL | * | NULL | NULL | NULL | NULL |

删除 dish_ID=4 后:

| | dish_ID | dish_name | | comm_ID | grade | content | dish_ID_id |
|---|---------|------------|---|---------|-------|--------------------------|------------|
| ▶ | 1 | 麦辣鸡腿堡中套餐 | ▶ | 1 | 5 | 穷鬼套餐YYDS!!! | 3 |
| | 3 | 随心配1+1超值套餐 | | 2 | 2 | 不裹鳞片的炸鸡不是好炸鸡:(转粉KFC了 | NULL |
| | 5 | 鸭腿 | | 4 | 4 | 众筹开一个鲜榨果汁窗口(参考东区美广) | NULL |
| | 8 | 123 | | 5 | 0.8 | WOW | 3 |
| | 9 | 12 | | 6 | 0 | 麦当劳不如肯德基(5毛1条, 记得把括号删... | NULL |
| | 10 | 123 | | 7 | 3 | 123 | NULL |
| | 11 | 321 | | 8 | 3 | 123 | NULL |
| * | NULL | NULL | * | NULL | NULL | NULL | NULL |

相关评论被级联删除

2. 删除 manager

删除前：

| | id | password | last_login | is_superuser | username |
|---|--------|---|----------------------------|--------------|------------|
| ▶ | 1 | pbkdf2_sha256\$390000\$WK9172nzXdmTjMjFVc33Um\$YTK0... | 2024-07-20 09:46:18.093904 | 1 | imlrz |
| | 2 | pbkdf2_sha256\$390000\$Qr8t82rnduZgGgsP0kWTjf\$gjdpER... | 2024-05-17 13:48:10.438045 | 0 | wangwenyan |
| | 3 | pbkdf2_sha256\$390000\$sIhfnmCvAgrC76RkQN2kVp\$gfjzw... | 2024-07-20 09:44:19.717918 | 0 | Harry |
| | 4 | pbkdf2_sha256\$390000\$uoMXIQUWY3uhSTPw26p8mH\$CuP... | NULL | 0 | 你说得对 |
| | 5 | pbkdf2_sha256\$390000\$tijea2pk19f201R51F1l5Z\$VRvCIws... | 2024-07-17 13:39:44.738628 | 0 | m |
| * | 6 | pbkdf2_sha256\$390000\$xkrb9HcpX5pYcTL15UjhJW\$irMnIZO... | NULL | 0 | 12345 |
| | * NULL | NULL | NULL | NULL | NULL |

| | resta_ID | resta_name | location | time_open | time_close | tag | more_Info | manager_id |
|---|----------|--------------|----------|-----------------|-----------------|------|-----------|------------|
| ▶ | 1 | 麦当劳 (合肥曙光餐厅) | E | 00:01:00.000000 | 23:59:00.000000 | 0 | 金寨路136号 | 5 |
| | 2 | 芳华餐厅 | W | 06:18:00.000000 | 21:21:00.000000 | SW | None | NULL |
| | 3 | test | O | 15:30:00.000000 | 18:00:00.000000 | NULL | NULL | NULL |
| * | HULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

删除 id=5 的用户（麦当劳经理）后：

| | id | password | last_login | is_superuser | username |
|---|--------|---|----------------------------|--------------|------------|
| ▶ | 1 | pbkdf2_sha256\$390000\$WK9172nzXdmTjMjFVc33Um\$YTK0... | 2024-07-20 09:46:18.093904 | 1 | imlrz |
| | 2 | pbkdf2_sha256\$390000\$Qr8t82rnduZgGgsP0kWTjf\$gjdpER... | 2024-05-17 13:48:10.438045 | 0 | wangwenyan |
| | 3 | pbkdf2_sha256\$390000\$sIhfnmCvAgrC76RkQN2kVp\$gfjzw... | 2024-07-20 09:44:19.717918 | 0 | Harry |
| | 4 | pbkdf2_sha256\$390000\$uoMXIQUWY3uhSTPw26p8mH\$CuP... | NULL | 0 | 你说得对 |
| * | 6 | pbkdf2_sha256\$390000\$xkrb9HcpX5pYcTL15UjhJW\$irMnIZO... | NULL | 0 | 12345 |
| | * NULL | NULL | NULL | NULL | NULL |

| | resta_ID | resta_name | location | time_open | time_close | tag | more_Info | manager_id |
|---|----------|--------------|----------|-----------------|-----------------|------|-----------|------------|
| ▶ | 1 | 麦当劳 (合肥曙光餐厅) | E | 00:01:00.000000 | 23:59:00.000000 | 0 | 金寨路136号 | NULL |
| | 2 | 芳华餐厅 | W | 06:18:00.000000 | 21:21:00.000000 | SW | None | NULL |
| | 3 | test | O | 15:30:00.000000 | 18:00:00.000000 | NULL | NULL | NULL |
| * | HULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

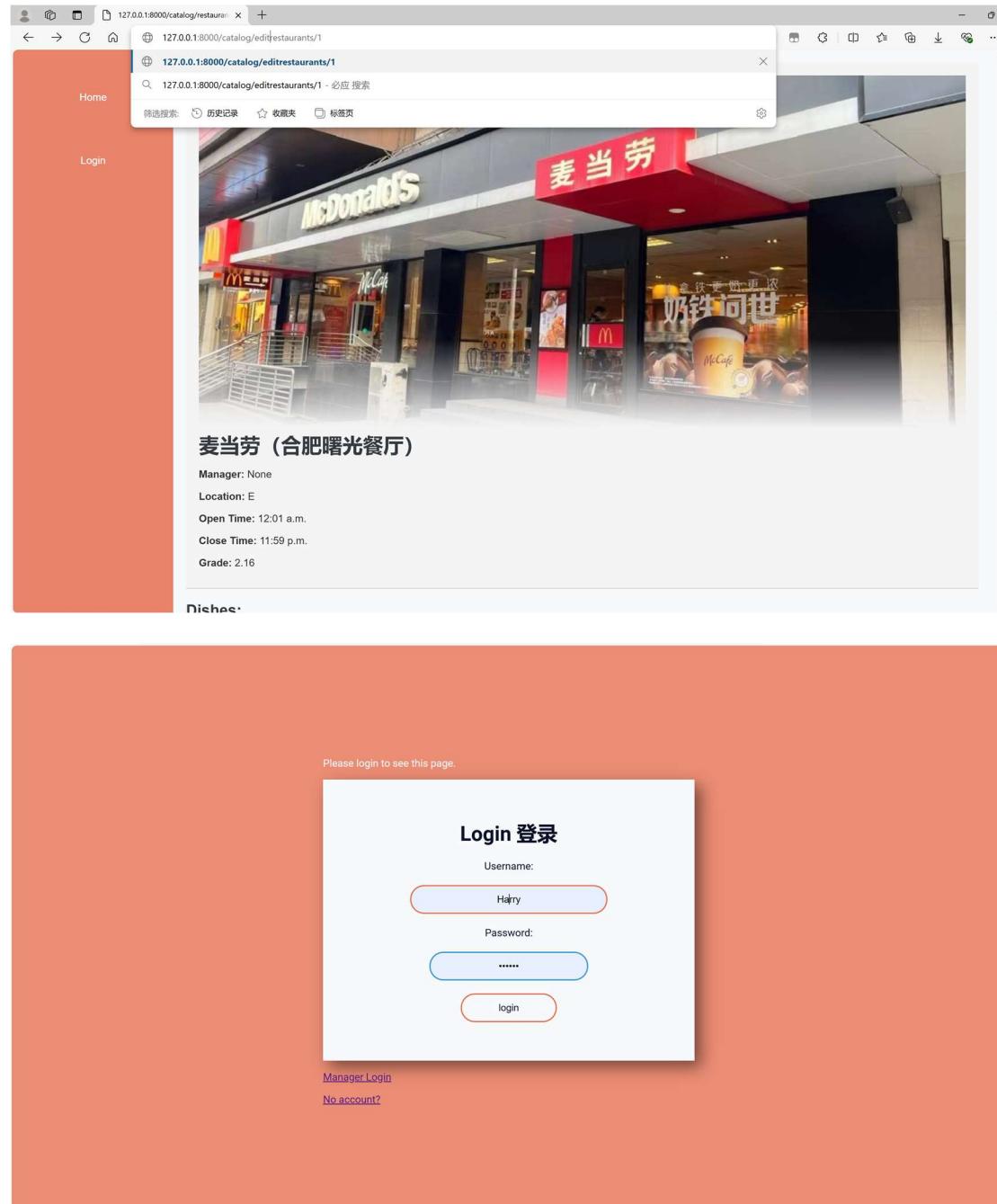
麦当劳 manager 被置空

5.2.3 用户定义的完整性

涉及到一些唯一性以及非空规则，具体的测试在前文相应的表单中已有所体现，不再赘述。

5.3 安全性测试

- 当用户未登录，却仍试图访问需登录才能访问的界面时，会自动定向至登陆界面

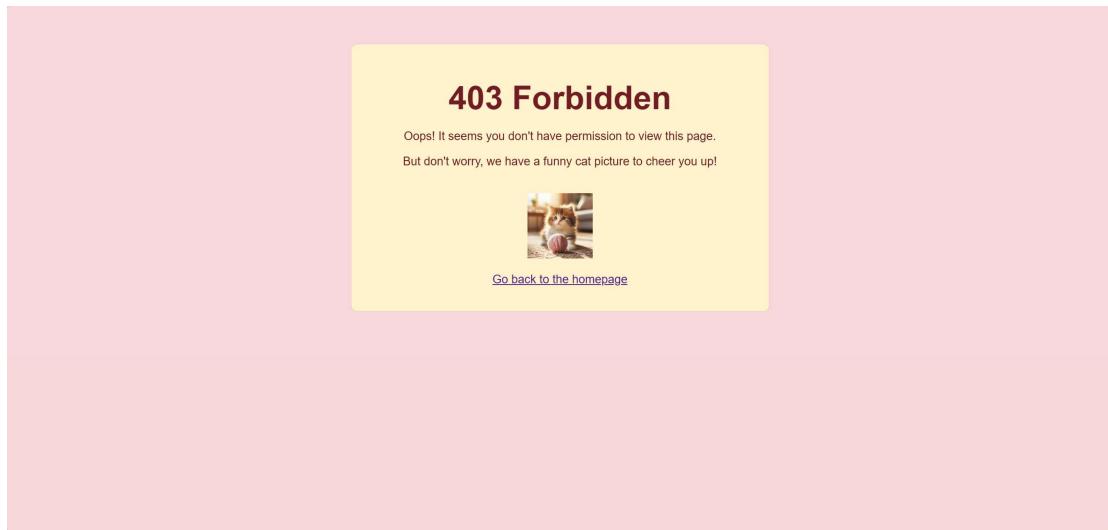


- 当非经理用户试图修改有经理餐厅的信息时：

经理登录后能够之间跳转到管理餐厅页面，而未登录和普通用户不会跳转到管理餐厅的页面，即使为登录用户和普通用户希望通过 url 进行直接的跳转，对应得视图函数也会进行登录认证和身份认证，有一个不满足则会无法进入管理页面，从而保证了对餐厅管理得安全性

对于不是经理的用户，若尝试通过输入 url 到达管理的页面，比如对于未登录的用户输入“127.0.0.1: 8000/catalog/manager/1”尝试查看某个经理的管理页面，则会跳转到下面的页面要求先登录

即使点击登录，但若不是经理也会跳转至下面的信息，告知用户没有权限访问该页面



若用户强制访问有经理餐厅 x 的“127.0.0.1: 8000/catalog/editrestaurants/x”，即使编辑信息点击提交后，仍然会显示：

