



» 高共享食堂外卖平台 “航送”

简介

我们的高共享性食堂外卖系统“航送”，开发初衷是通过食堂的外送服务将同学连接在一起。平台最大的特点是顶送一体，同学可以点餐，也可以在自己顺手的时候抢单，把相应的订单配送，这样可以极大地方便同学们的生活，解决一些时间问题带来的窘迫。相应的理念还可以套用在其他平台上，例如，代取快递等等。接下来是我们具体的细节展示。

目录

前端界面展示

后端设计

深度学习部分

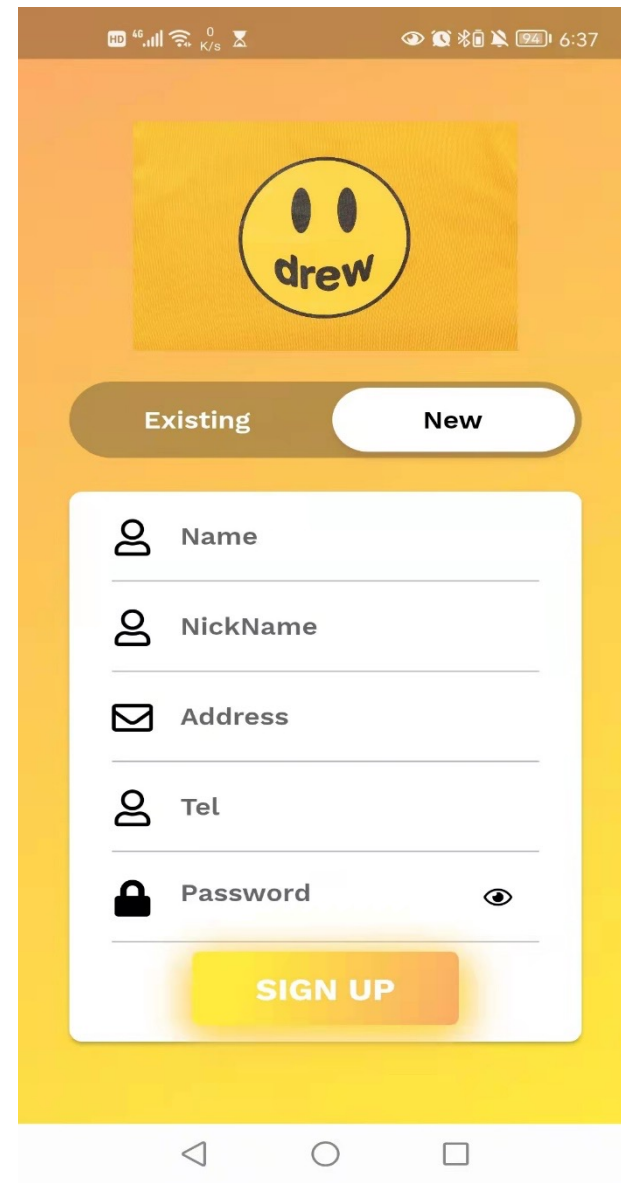
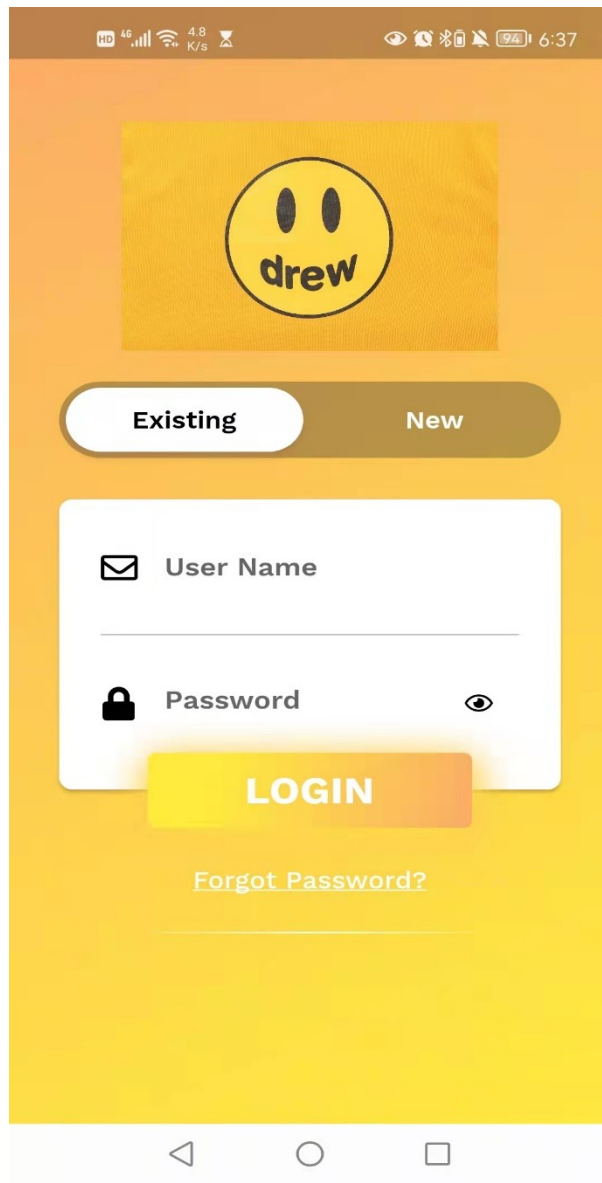
» 01 抢单、用户界面

登陆注册

主要是实现了登陆和注册的功能。

亮点是通过有状态的Widget和controller实现了上方的滑条。

同时，登陆时密码错误和注册时两次输入密码不一致都会进行报错的提醒。



个人主页

个人界面主要是进行一个展示功能，提供跳转到其他界面的按钮。

Badge组件：

可以根据订单或者收藏的数量进行按钮右上角数字的显示（数量为0的时候不会进行显示）。

更改个人信息：用户点击“锁”的图标，可以跳转到相应的更改信息的界面（也是通过controller实现）。



送餐界面

送餐界面显示用户所接单的信息，与送餐历史相似。

只有当下单者按下已收货按钮后，订餐才会在接单者的送餐页面消失，表示此订单已送达；并实时显示在送餐历史中

显示订单的详细信息
(便于与下单者核对)

显示订单的收货人
收货地址等信息



待收餐界面

显示用户待收餐的订单信息。

收到餐品后，点击收货按钮，此订单就会在带收餐界面消失，在收餐历史显示。

若无骑手接单，收获按钮则标灰，不可按



订单信息可收缩
(便于接单者快速浏览)

预测时间
(根据收餐地点、餐厅地点、是否处于高峰期预测出来)



按照时间逆序显示



收货按钮
(点击之后接单，此订单在抢单界面消失)

历史记录

包括订餐、送餐的历史记录。

订单历史：

可以收缩显示之前的订餐信息，
和骑手的姓名及其电话等。

送餐历史：显示收货人昵称、收货地点、所买餐品等信息。



抢单界面

抢单界面是我们app的一大特色，它允许每一个客户对所有待配送的订单进行抢单。用户可以根据订单的具体信息，如餐厅地点、收货地点来自行选取。



订单信息可收缩
(便于接单者快速浏览)

显示订单信息
(便于接单者选择)



按照时间正序显示
(早下单，早被接单)

接单按钮
(点击之后接单，此订单在抢单界面消失)

» 02

食堂、菜品页面

首页 (热菜推荐)

登陆成功后，首先展示的页面为Home首页，即热菜推荐页面。该页面会按收藏数量将所有菜品进行排序展示，并对菜品的图片、名称、食堂进行展示。用户可以根据此点击喜好的菜品进行选择下单。



菜品展示卡片，
点击可以进入对应菜品信息界面

显示菜品信息
(向用户展示)



按照收藏量
正序显示



收藏按钮
(初始状态为是否
收藏了该菜品，点
击后会更改收藏状
态，红心样式改变)



分食堂展示页面

点击下边栏Category进入食堂展示页面。该页面向用户展示全部食堂窗口，供用户分食堂进行选单、信息查询。



食堂展示卡片，
点击可以进入
对应食堂窗口
信息界面



显示食堂信息
(向用户展示)

菜品信息展示 页面

点击菜品卡片后进入菜品信息展示页面。该页面向用户展示选中菜品的基础信息，与评价信息，包括平均评分、评论等。

另外，提供了评分接口、评论接口可以供用户进行评价；

提供了下单按钮，可以将当前菜品加入购物车；

提供了购物车按钮可以进入购物车进行查看。

菜品信息展示：
图片、名称、
食堂、价格

菜品评价接口、评论
添加框



菜品添加至
购物车按钮

菜品评价展示：
平均评分、大
众评论

进入购物车
页面按钮

食堂信息展示 页面

点击食堂卡片后进入食堂信息展示页面。该页面向用户展示选中食堂的基础信息、以及食堂所包含的全部菜品信息，点击卡片同样可以进入对应菜品信息展示页面。

另外，提供了购物车按钮可以进入购物车进行查看。



食堂信息展示：
图片、名称、
地址、电话

食堂窗口全
部菜品展示



进入购物车
页面按钮

购物车商品展示页面

点击进入购物车按钮后进入购物车商品展示页面。该页面向用户展示当前已下单的菜品，并可以进行信息查阅核对。

另外，提供了下单按钮可以进入购物车进行查看。



购物车全部菜品信息展示：
图片、名称、
价格、地址



下单按钮

» 03

后端架构

框架



Python下有许多款不同的 Web 框架。Django是重量级选手中最有代表性的一位。许多成功的网站和APP都基于Django。

Django 是一个开放源代码的 Web 应用框架，由 Python 写成。

Django 遵守 BSD 版权，初次发布于 2005 年 7 月，并于 2008 年 9 月发布了第一个正式版本 1.0 。

Django 采用了 MVT 的软件设计模式，即模型（Model），视图（View）和模板（Template）。



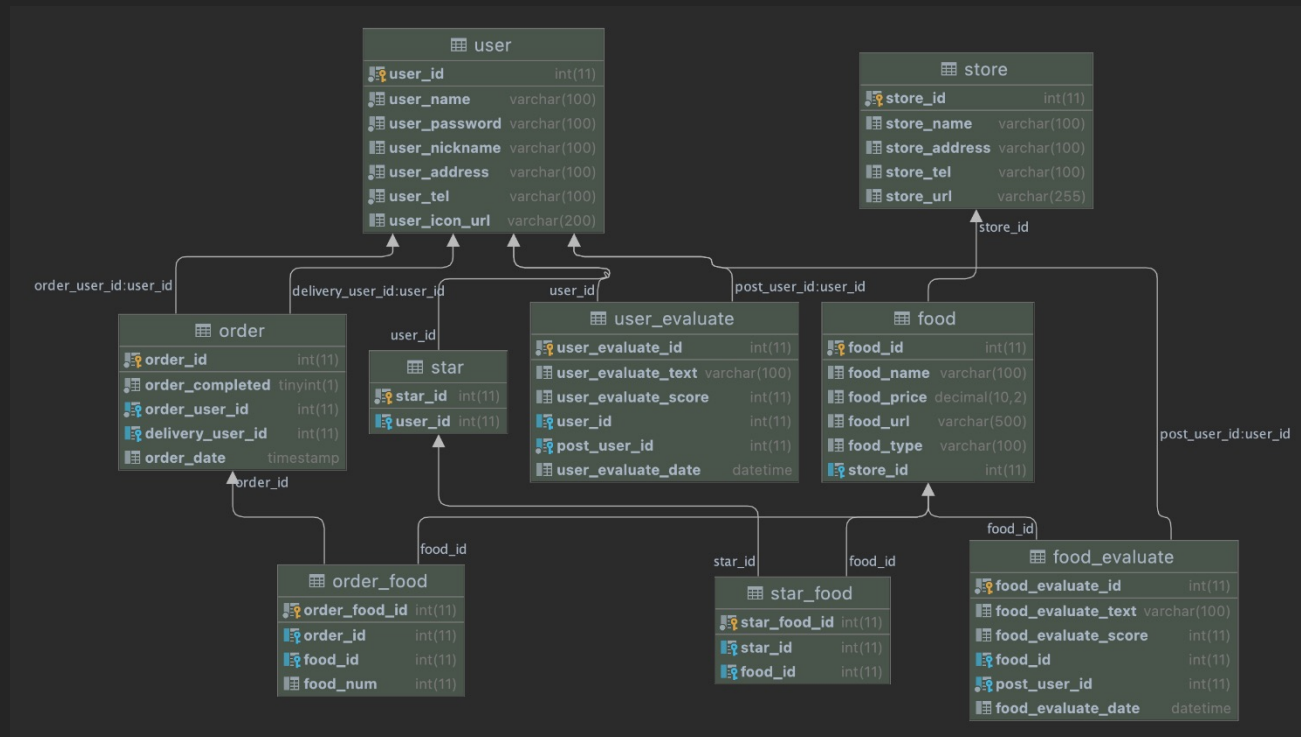
经典关系数据库

我们将Django后端部署腾讯云上，前端只需要访问API即可获得数据信息。

MySQL

我们基于业务需求进行了数据库设计，包含了概念设计（画ER图）、逻辑结构设计以及数据库实现三个阶段，最终得到的数据库包括了7个实体，且得到的关系模式规范到4NF。

我们为所有存在关系的对象增加了外键约束，并且加入了 MySQL on delete cascade 的删除规则，在对象删除时删除与其有外键约束的对象，保证数据完整性。

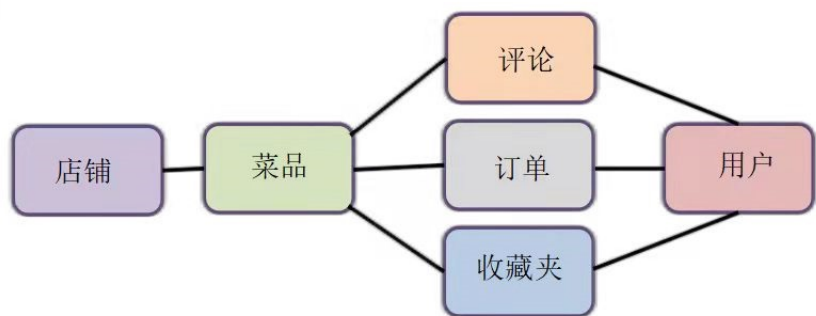


数据库ER图

我们将MySQL部署在云端服务器上，后端通过访问数据库3306端口获得数据，我们的数据库环境如下：


服务器系统版本：CentOS 7

MySQL版本：5.5.62



密码加密

我们对数据库的密码进行了加密，通过Django的 `make_password()` 和 `check_password()` 函数在传入和验证密码时候进行了加密解密操作。



user_id	user_name	user_password
1	zzh	pbkdf2_sha256\$2600000\$9nWwzBNSHZ4NUbpQ34QDcp\$0z...
2	charlot	pbkdf2_sha256\$2600000\$aFbsx49mui99DBgieSliNa\$b5...

```
if user.user_password == password:
    user.user_password = make_password(password)
    user.save()
    return JsonResponse({"success": True, "message": "登录成功，已为您的密码加密", "userID": user.user_id})
elif check_password(password, user.user_password):
    return JsonResponse({"success": True, "message": "登录成功", "userID": user.user_id})
```

» 04

机器学习应用

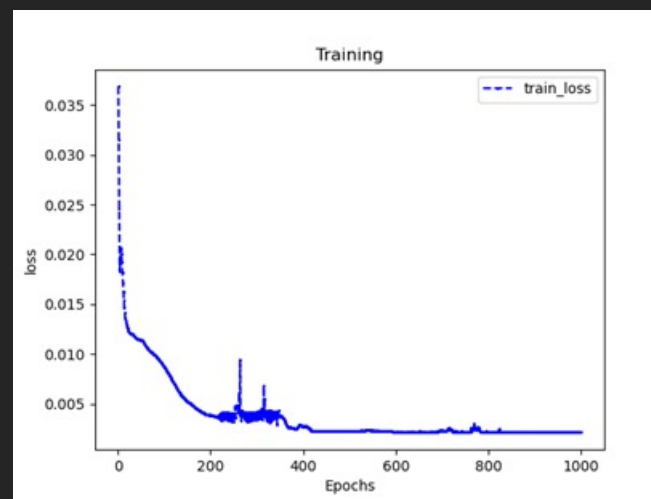
机器学习

我们根据订单的配送时间是星期几，餐厅，配送地等信息通过深度学习预测订单送达时间 我们通过手动构造数据集，训练了一个模型存到服务器，并在每次前端发送新的请求后对订单进行预计送达时间的计算

forecast.py
model_parameter.pkl

```
def predict(model, dl):  
    model.eval()  
    with torch.no_grad():  
        result = model(torch.FloatTensor(dl))  
    return(result.item()*65.0)
```

py



将**Loss**值降到了**0.005**以下