# 项目介绍

* 1. 项目描述简介

通过完成一个仿饿了么的在线订餐平台，让大家掌握企业开发特点，以及解决问题的能力。

该项目会涉及以后工作中的一些非常有代表性的功能。

用户端采用前后端分离开发模式，前端通过调用后端API接口来获取和提交数据。掌握项目中前后端分离的开发模式。

我们使用git作为项目的版本控制软件。

在项目中会使用很多其他的知识，比如系统架构、维护、优化等等。

* 1. 主要功能模块
  2. 开发环境和技术

|  |  |
| --- | --- |
| 开发环境 | Window |
| 开发工具 |  |
| 相关技术 |  |

* 1. 项目人员组成周期成本
     1. 人员组成

|  |  |  |
| --- | --- | --- |
| 职位 | 人数 | 备注 |
| 项目经理和组长 | 1 | 一般小公司由项目经理负责管理，中大型公司项目由项目经理或组长负责管理 |
| 开发人员 | 2~3 |  |
| UI设计人员 | 1 |  |
| 前端开发人员 | 1 | 专业前端不是必须的，所以前端开发和UI设计人员可以同一个人 |
| 测试人员 | 1~3 | 有些公司并未有专门的测试人员，测试人员可能由开发人员完成测试。  公司有测试部，测试部负责所有项目的测试。  项目测试由产品经理进行业务测试。 |

* + 1. 项目周期成本

|  |  |  |
| --- | --- | --- |
| 人数 | 周期 | 备注 |
| 3 | 四个月  第一个月需求设计  第二三个月编码实现  第四个月编码实现、测试 |  |

1. 功能模块

2.1用户功能模块

### 2.1.1需求

1. 用户填写电话号码，并设置6-16位密码，进行注册
2. 后台验证用户提交的电话号码是否合理以及是否已经注册，验证成功跳转至登录页面，若验证电话号码已经存在则显示号码已经注册，请登录。
3. 用户点击登录，填写电话号码或用户名以及密码，进行登录。
4. 后台验证用户名及密码，如果用户名和密码都正确则返回个人中心页面，如果验证错误则提示相应的错误信息。
5. 用户点击忘记密码，则返回找回密码页面，用户填写电话号码，接收验证码，填写正确的验证码并重新设置密码。
6. 当用户进入个人中心，则显示用户头像，电话号码。当用户点击个人资料时，进入个人资料，填写个人资料，并提交。
7. 当用户进入安全设置，则返回密码重设，用户填写旧密码及重设密码。

### 2.1.2流程

1. 当用户发出注册请求时，显示注册页面，让用户填写电话号码以及两次填写密码
2. 当用户点击注册按钮时，获取用户填写的数据，并进行表单验证，验证数据的合理性：电话必须为11位未注册过的电话号码，密码必须为6-16个字符，两次输入的密码必须一致
3. 若用户提交的数据合理，则将用户提交的密码进行哈希加密，并将电话号码和密码保存到数据库中，并跳转到登录页面，让用户登录
4. 若用户提交的数据不合理，则提示错误信息：电话号码已存在，电话号码格式错误，密码必须为6-16个字符；并返回注册页面让用户重新填写。

### 2.1.3设计要点（数据库和页面交互）

1. model设计：

1.1数据库的名字：users

1.2数据库的基本字段：

ID 主键，自动生成

Mobile 手机号

字段类型：Charfiled,设置正则匹配

username昵称

字段类型：Charfiled，设置最大长度20以及最小长度1，默认为空

password密码

字段类型：Charfiled，设置最大长度20以及最小长度1，必须填写

sex性别,字段类型：Charfiled，设置选项：1.男，2.女，默认为男

school学校,字段类型：Charfiled

字段类型：Charfiled，设置最大长度50，默认为空

loction 地址,字段类型：Charfiled

字段类型：Charfiled，设置最大长度50，默认为空

home老家,字段类型：Charfiled

字段类型：Charfiled，设置最大长度50，默认为空

add\_time添加时间

字段类型：Datefiled，设置为auto\_time\_now

update\_time修改时间

字段类型：Datefiled ，设置为auto\_time

Is\_delete是否删除

字段类型：Booleanfiled，默认为False

2. 后端页面对数据的校验form表单设计

2.1.对手机号码进行正则判断，用户如果输入了格式不正确的手机号，提示用户输入正确的手机号；对密码进行验证如果密码长度不对，提示用户密码长度应为6-12位。

2.2对手机号码进行单一验证

从数据库中获取mobile数据与用户提交的数据一样的数据，如果存在则提示用户：账号已存在。

如果不存在则进行密码验证。

2.3对密码进行综合验证，验证用户两次输入的密码是否一致，如果不一致则提示两次密码不一致，如果一致则返回清洗后的数据。

### 2.1.4难点及解决方案

### 难点：

### form表单验证难

### 验证码的获取

粗心错误：

Render和redirect要区分开来。

单词错误。

在写代码的过程中注意变量的使用变量名不需要加引号。

表单提交的时候需要有name属性才能提交。

把提交框写在了form表单的外面。

基础知识错误:

1表单验证中：单一验证抛出错误时不能指定抛错的字段（谨记）。

2数据库操作中filter方法获取的是查询集而不是某个单独的字段。

3当模型中字段设置了唯一键时，不能再次提交相同的值，所以在更改的时候验证表单时创建对象。

form = InforModelForm(data, instance=Users.objects.get(pk=user\_id))

4当html 表单元素在提交时设定了value属性会影响后台表单验证，使字符数目变大。

5在单一验证时的返回值只能是验证的数据。

1. 2.2商品功能模块
   1. 2.2.1需求
2. 从数据库把数据查询出来展示到首页页面
3. 从数据库把页面查询出来展示在超市页面
4. 从数据库将数据查询出来展示在活动区域
5. 从数据库把数据查询出来展示在详情页面
   1. 2.2.2流程
6. 当用户访问127.0.0.1:8000时，进入首页，从数据库查询ActiveZone表和Goods\_sku表，将活动商品信息展示出来，当用户点击活动商品时，根据商品id从数据库查询Goods\_sku以及Goods\_spu表里的信息，展示在相应的位置。
7. 当用户点击超市时，返回超市页面，并从数据库把商品信息以及分类信息查询并展示在页面。当用户查看商品分类信息以及商品信息时，根据id从数据库查询Goods\_sku以及Goods\_spu表里的信息，展示在相应的位置。
   1. 2.2.3设计要点（数据库和页面交互）
      * 1. model设计原则：先建一的一方的表格，再建多的一方的表格
8. Category 模型

cate\_name，分类名称,字段类型：Charfiled

Brif:简介，字段类型：Charfiled

order:排序,字段类型：SmallIntegerField(该字段用于后续排序)

1. Unit 模型

Name：单位名称，字段类型：Charfiled

1. GoodsSPU 模型

spu\_name：spu名称，字段类型：Charfield

Content:商品详情，字段类型：RichTextFileField(这里必须为这个类型)

1. GoodsSKU 模型

Sku\_name:sku名称，字段类型：Charfield

Brif:商品简介，字段类型：Charfield

Price:商品简介，字段类型：Decimalfield(这里价格的字段类型只能为这个类型)

unit：商品单位,字段类型：Charfield

Stock:商品库存，字段类型：IntegerField

sale\_num：商品销量，字段类型：IntegerField

logo：商品图片，字段类型：ImageField

is\_on\_sale：是否售卖，字段类型：Booleanfield

Category:外键

goods\_spu：外键

1. Gallery 模型

img\_url：图片地址，字段类型：ImageField（用于保存商品图片地址）

goods\_sku：外键

1. Banner 首页轮播模型

name：轮播活动名，字段类型：Charfield

img\_url：图片地址，字段类型：ImageField（用于保存商品图片地址）

order:排序,字段类型：SmallIntegerField(该字段用于排序）

goods\_sku：外键

1. Activity 活动首页模型

Title:活动名，字段类型：Charfield

img\_url：图片地址，字段类型：ImageField（用于保存商品图片地址

url\_site：活动地址，字段类型：URLField

1. ActivityZone 活动专区模型

Title:活动名，字段类型：Charfield

Brif:活动简介，字段类型：Charfield

* + - 1. 一定要把所有的models加入后台管理
  1. 2.2.4要点难点及解决方案
  2. 要点：模型要设计好
  3. 难点：地图

1. 订单功能模块
   1. 需求
   2. 流程
   3. 设计要点（数据库和页面交互）
   4. 要点难点及解决方案
2. 购物车功能模块
   1. 需求
   2. 流程
   3. 设计要点（数据库和页面交互）
   4. 要点难点及解决方案
3. 常见面试问题