**“拜托了快递”快递代拿系统**

**项目详细设计说明书**

**版本<7.0>**

|  |  |  |  |
| --- | --- | --- | --- |
| **拟制** |  | **日期** |  |
| **审核** |  | **日期** |  |
| **批准** |  | **日期** |  |

**XXXX单位/公司**

**修改记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 修改日期 | 修改原因 | 版本号 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 引言 4](#_Toc40610545)

[1.1 编写目的和范围 4](#_Toc40610546)

[1.2 术语表 4](#_Toc40610547)

[1.3 参考资料 4](#_Toc40610548)

[2. 全局说明 4](#_Toc40610549)

[2.1 全局流程说明 4](#_Toc40610550)

[2.2 常量变量说明 5](#_Toc40610551)

[2.3 数据结构和算法 5](#_Toc40610552)

[3. 用户认证模块设计 5](#_Toc40610553)

[3.1 用例图 5](#_Toc40610554)

[3.2 流程图 5](#_Toc40610555)

[3.3 功能设计说明 6](#_Toc40610556)

[3.3.1 微信授权登录功能 6](#_Toc40610557)

[3.3.2 身份认证功能 7](#_Toc40610558)

[3.4 数据存储设计 7](#_Toc40610559)

[4. 地址管理模块设计 8](#_Toc40610560)

[4.1 用例图 8](#_Toc40610561)

[4.2 流程图 8](#_Toc40610562)

[4.3 功能设计说明 9](#_Toc40610563)

[4.3.1 添加地址功能 9](#_Toc40610564)

[4.3.2 删除地址功能 10](#_Toc40610565)

[4.4 数据存储设计 10](#_Toc40610566)

[5. 订单管理模块设计 11](#_Toc40610567)

[5.1 用例图 11](#_Toc40610568)

[5.2 流程图 11](#_Toc40610569)

[5.3 功能设计说明 11](#_Toc40610570)

[5.3.1 发布订单功能 11](#_Toc40610571)

[5.3.2 接收订单功能 12](#_Toc40610572)

[5.3.3 确认送达功能 12](#_Toc40610573)

[5.3.4 系统自动删除订单功能 12](#_Toc40610574)

[5.3.5 用户删除订单功能 12](#_Toc40610575)

[5.4 数据存储设计 13](#_Toc40610576)

[6. 信息反馈模块设计 13](#_Toc40610577)

[6.1 用例图 13](#_Toc40610578)

[6.2 流程图 14](#_Toc40610579)

[6.3 功能设计说明 14](#_Toc40610580)

[6.3.1 举报功能 14](#_Toc40610581)

[6.3.2 评价功能 14](#_Toc40610582)

[6.4 数据存储设计 15](#_Toc40610583)

[7. 系统性能设计 15](#_Toc40610584)

[8. 系统出错处理 15](#_Toc40610585)

[9. 项目风险 16](#_Toc40610586)

[10. 当前进度 16](#_Toc40610587)

# 引言

## 编写目的和范围

编写详细设计说明书的目的是为了使开发人员在完成概要设计说明书的基础上，完成概要设计的各项功能规定；为程序员写出实际的程序代码提供依据；说明程序模块的设计考虑,包括程序描述、输入/输出、算法和流程逻辑等，为软件编程和系统维护提供基础。它是软件设计阶段所有任务和所有相关人员所需的参考资料。 预期的读者为系统的开发者、使用者和指导老师。

## 术语表

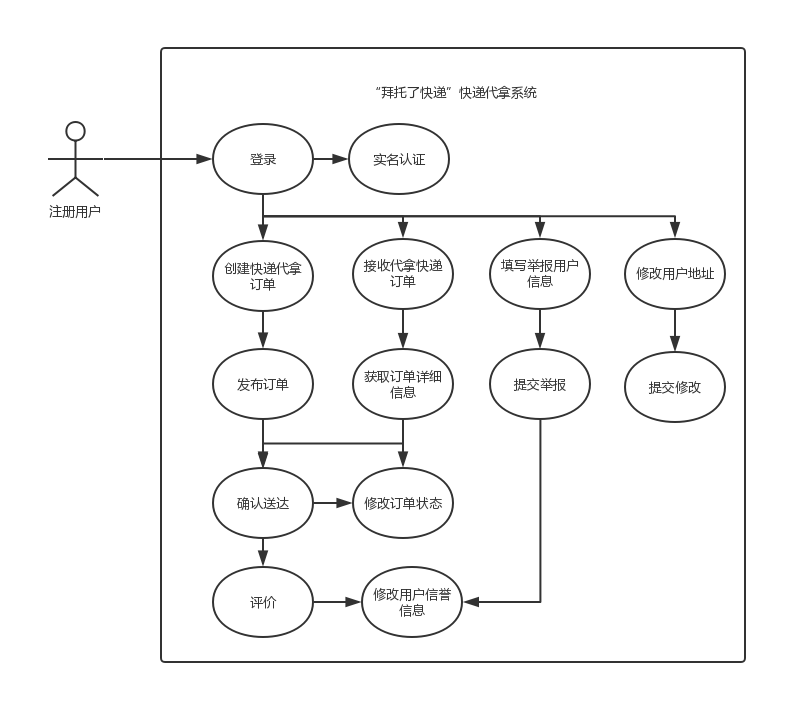
|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| 发布者 | 指在快递代拿系统中发布快递等信息，请求其他用户为自己提供代拿快递服务的用户 |
| 接单者 | 指在快递代拿系统中接收某一用户的快递代拿订单的用户 |
| 接收者 | 同上 |

## 参考资料

1. 本小组成员撰写的《可行性分说明书》；
2. 本小组成员撰写的《需求分析说明书》；
3. 本小组成员撰写的《概要设计说明书》。

# 全局说明

## 全局流程说明



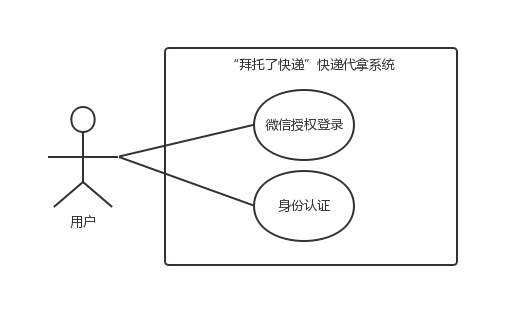
## 常量变量说明

后端Django中的数据库配置

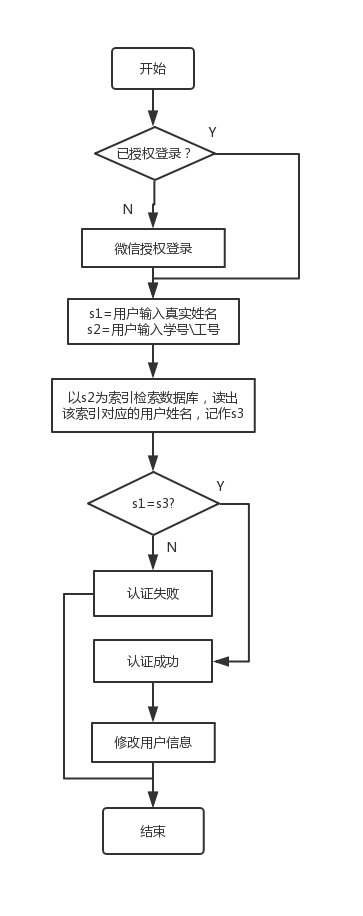
DATABASES = {  
 **'default'**: {  
 **'ENGINE'**: **'django.db.backends.mysql'**,  
 **'NAME'**: **'go\_express'**,  
 **'USER'**:**'root'**,  
 **'PASSWORD'**:**'\*\*\*\*\*\*\*\*'**,  
 **'HOST'**:**''**,  
 **'PORT'**:**''**,  
 }  
}

# 用户认证模块设计

## 用例图



## 流程图



## 功能设计说明

### 微信授权登录功能

后端通过以下程序判断该用户是否已授权登录，

wx.getSetting({

success: (res) => {

/\*

\* res.authSetting = {

\* "scope.userInfo": true,

\* "scope.userLocation": true

\* }

\*/

}})

如果未授权登录，则提示用户“请在个人中心中授权登录”。当用户授权登录时，系统获取到用户的openid以及微信名称并写入数据库中的用户信息表中。若用户已完成授权登录，则系统不再授予该用户“授权登录”的选项。

### 身份认证功能

前端个人中心中存在“绑定信息”入口，用户可点击“绑定信息”进入身份认证界面，后端通过以下程序判断该用户是否已授权登录，

wx.getSetting({

success: (res) => {

/\*

\* res.authSetting = {

\* "scope.userInfo": true,

\* "scope.userLocation": true

\* }

\*/

}})

如果未授权登录，则提示用户“请在个人中心中授权登录”，如果已授权登录，后端继续判断用户是否已认证（绑定个人信息），若未绑定个人信息，则提示用户“请在个人中心中绑定信息”。用户点击“绑定信息”，后端读入用户在前端输入的姓名及学号，随后系统以学号作为索引查找数据库中的学生信息表，并读取到该索引在数据库中对应的用户真实姓名，再将该姓名与前端读入的姓名作比较。如若二者完全一致，则系统认为该用户身份认证成功，并将其学号及姓名存储到数据库中对应的用户信息表上。如若二者不一致，则系统认为该用户身份认证失败，不予受理该用户真实身份。若用户已完成个人信息的绑定，则系统不在授予该用户“绑定信息”功能，默认该用户为系统上一次所认证到的身份。

## 数据存储设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 字段约束 | 说明 |
| stuname | varchar | 10 | Not NULL | 学生姓名 |
| stuid | varchar | 10 | Not NULL | 学生学号 |

# 地址管理模块设计

## 用例图

**

## 流程图



## 功能设计说明

### 添加地址功能

前端个人中心中存在“我的地址”入口，用户可点击“我的地址”进入地址管理界面，后端通过以下程序判断该用户是否已授权登录，

wx.getSetting({

success: (res) => {

/\*

\* res.authSetting = {

\* "scope.userInfo": true,

\* "scope.userLocation": true

\* }

\*/

}})

如果未授权登录，则提示用户“请在个人中心中授权登录”，如果已授权登录，后端继续判断用户是否已认证（绑定个人信息），若未绑定个人信息，则提示用户“请在个人中心中绑定信息”。已绑定信息或绑定信息后，用户点击“我的地址”，后端通过用户openid，在数据库中查找到该用户已添加的所有地址信息并传给前端进行显示，用户在地址信息末尾点击“添加地址”进入添加地址信息界面，填入地址，点击“添加地址”，前端获取到地址信息后返回给后端，后端判断地址信息的三个属性是否存在有空值，根据判断结果给出提示信息，添加成功后返回地址管理界面，显示已有地址信息和刚添加的地址信息。

### 删除地址功能

与添加地址功能相似，前端个人中心中存在“我的地址”入口，用户可点击“我的地址”进入地址管理界面，后端通过以下程序判断该用户是否已授权登录，

wx.getSetting({

success: (res) => {

/\*

\* res.authSetting = {

\* "scope.userInfo": true,

\* "scope.userLocation": true

\* }

\*/

}})

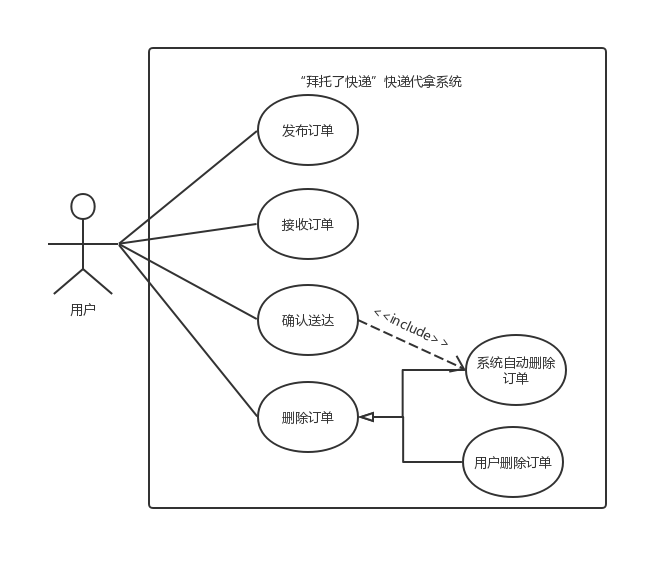
如果未授权登录，则提示用户“请在个人中心中授权登录”，如果已授权登录，后端继续判断用户是否已认证（绑定个人信息），若未绑定个人信息，则提示用户“请在个人中心中绑定信息”，已绑定信息或绑定信息后，用户点击“我的地址”，后端通过用户openid，在数据库中查找到该用户已添加的所有地址信息并传给前端进行显示。用户可对已有地址信息进行删除操作，前端获取待删除的地址信息和用户的openid返回给后端，后端在数据库中的地址信息表里找到该条地址信息进行删除，删除成功后返回地址管理界面，显示已有地址信息。

## 数据存储设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 字段约束 | 说明 |
| openid | varchar | 50 | Not NULL | 用户在微信小程序中的唯一标示 |
| useraddress | varchar | 50 | Not NULL | 地址 |

# 订单管理模块设计

## 用例图



## 流程图



## 功能设计说明

### 发布订单功能

前端存在“发布”入口，用户可点击“发布”进入发布订单界面，后端通过以下程序判断该用户是否已授权登录，

wx.getSetting({

success: (res) => {

/\*

\* res.authSetting = {

\* "scope.userInfo": true,

\* "scope.userLocation": true

\* }

\*/

}})

如果未授权登录，则提示用户“请在个人中心中授权登录”，如果已授权登录，后端继续判断用户是否已认证（绑定个人信息），若未绑定个人信息，则提示用户“请在个人中心中绑定信息”。已绑定信息或绑定信息后，用户输入发布订单的信息，包括跑腿费、快递大小、快递站点、截至时间等信息，系统后端获取这些订单信息存入订单信息表中，点击发布订单按钮之后系统将获取的订单信息在首页上发布，将订单信息公开显示出来。

### 接收订单功能

在前端的首页中，用户可点击“我的订单”进入订单管理界面，后端通过以下程序判断该用户是否已授权登录，

wx.getSetting({

success: (res) => {

/\*

\* res.authSetting = {

\* "scope.userInfo": true,

\* "scope.userLocation": true

\* }

\*/

}})

如果未授权登录，则提示用户“请在个人中心中授权登录”，如果已授权登录，后端继续判断用户是否已认证（绑定个人信息），若未绑定个人信息，则提示用户“请在个人中心中绑定信息”。已绑定信息或绑定信息后，用户点击“我要代拿”并且确定接单之后，系统后端修改订单信息表中该订单的状态，将订单状态改成“已接单”。

### 确认送达功能

前端个人中心中存在“我的订单”入口，用户可点击“待收货”进入订单管理界面，后端通过以下程序判断该用户是否已授权登录，

wx.getSetting({

success: (res) => {

/\*

\* res.authSetting = {

\* "scope.userInfo": true,

\* "scope.userLocation": true

\* }

\*/

}})

如果未授权登录，则提示用户“请在个人中心中授权登录”，如果已授权登录，后端继续判断用户是否已认证（绑定个人信息），若未绑定个人信息，则提示用户“请在个人中心中绑定信息”。已绑定信息或绑定信息后，若接收者和发布者均点击确认送达，系统从前端获取信息，后端将查找到该订单信息，将其订单状态修改为“已送达”。

### 系统自动删除订单功能

前端个人中心中存在“我的订单”入口，用户可点击“我的订单”进入订单管理界面，后端通过以下程序判断该用户是否已授权登录，

wx.getSetting({

success: (res) => {

/\*

\* res.authSetting = {

\* "scope.userInfo": true,

\* "scope.userLocation": true

\* }

\*/

}})

如果未授权登录，则提示用户“请在个人中心中授权登录”，如果已授权登录，后端继续判断用户是否已认证（绑定个人信息），若未绑定个人信息，则提示用户“请在个人中心中绑定信息”。已绑定信息或绑定信息后，系统取到当前用户全部“已发布订单”的订单编号。随后系统以这些编号作为索引，在数据库中的订单信息表中查找该索引对应的订单记录，并读取出这些索引对应的latesttime，也即最晚接单时间，然后判断当前系统时间是否已超过最晚接单时间。若已超过，则系统删除该索引在数据库中对应的订单记录。若还未超过，则系统不对该订单做任何处理。

### 用户删除订单功能

前端个人中心中存在“我的订单”入口，用户可点击“我的订单”进入订单管理界面，后端通过以下程序判断该用户是否已授权登录，

wx.getSetting({

success: (res) => {

/\*

\* res.authSetting = {

\* "scope.userInfo": true,

\* "scope.userLocation": true

\* }

\*/

}})

如果未授权登录，则提示用户“请在个人中心中授权登录”，如果已授权登录，后端继续判断用户是否已认证（绑定个人信息），若未绑定个人信息，则提示用户“请在个人中心中绑定信息”。已绑定信息或绑定信息后，用户点击“我的订单”中的“已发布订单”可查看到自己已发布的所有快递代拿订单。当用户向系统发出删除某一订单的请求时，系统从前端读入该订单对应的订单编号。随后系统以该编号作为索引，在数据库中的订单信息表中查找该索引对应的订单记录，并删除这些记录。

## 数据存储设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 字段约束 | 说明 |
| orderid | int | 10 | Not NULL | 订单编号 |
| pubWXname | varchar | 20 | Not NULL | 发布者微信名 |
| pubid | varchar | 10 | Not NULL | 发布者编号 |
| pubtime | varchar | 15 | Not NULL | 发布时间 |
| takeadd | varchar | 20 | Not NULL | 取件地址 |
| takeid | varchar | 5 | Not NULL | 取件码 |
| rectel | varchar | 15 | Not NULL | 收件人手机号 |
| recadd | varchar | 20 | Not NULL | 收货地址 |
| reward | double | - | Not NULL | 跑腿费 |
| weight | Int | - | Not NULL | 快递重量 |
| expecttime | varchar | 15 | Not NULL | 期望送达时间 |
| latesttime | varchar | 15 | Not NULL | 最晚接单时间 |
| ordstatus | varchar | 1 | Not NULL | 订单状态 |
| note | varchar | 100 | - | 备注 |
| completetime | varchar | 15 | - | 完成时间 |
| rectime | varchar | 15 | - | 接单时间 |
| recid | varchar | 10 | - | 接单者编号 |
| recWXname | varchar | 20 | - | 接单者微信名 |

# 信息反馈模块设计

## 用例图



## 流程图



## 功能设计说明

### 举报功能

前端个人中心中存在“联系客服”入口，用户可点击“联系客服”进入举报信息界面，该界面显示了客服的QQ号以及邮箱，以及举报的流程，用户可自己通过联系方式联系客服。进行反馈以及举报。

### 评价功能

前端个人中心中存在“我的订单”入口，用户可点击“待评价”进入评价界面，后端通过以下程序判断该用户是否已授权登录。

wx.getSetting({

success: (res) => {

/\*

\* res.authSetting = {

\* "scope.userInfo": true,

\* "scope.userLocation": true

\* }

\*/

}})

如果未授权登录，则提示用户“请在个人中心中授权登录”，如果已授权登录，后端继续判断用户是否已认证（绑定个人信息），若未绑定个人信息，则提示用户“请在个人中心中绑定信息”。已绑定信息或绑定信息后，用户点击“我的地址”，后端通过用户openid，在数据库中查找到该用户已完成且未评价的订单信息并传给前端进行显示，用户在订单信息处点击“评价”按钮进入评价界面，点击“好评”按钮或者“差评”按钮即可完成对该订单的评价，前端获取到评价信息后返回给后端，后端通过数据库查询到该订单的接单者，若是好评且接单者信用值不为100时，增加10点信用值，若是差评且信用值不为0时，减少10点信用值，评价完成之后，返回评价界面。

## 数据存储设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 字段约束 | 说明 |
| openid | varchar | 50 | Not NULL | 用户在微信小程序中的唯一标示 |
| creditrating | int | 10 | Not NULL | 用户的信用值等级 |
| ordstatus | varchar | 11 | Not NULL | 订单状态 |
| evaluateinfo | varchar | 50 | - | 评价信息 |

# 系统出错处理

## 7.1出错信息

|  |  |  |
| --- | --- | --- |
| 错误类型 | 子项 | 错误原因 |
| 数据库错误 | 连接 | 连接超时 |
| 连接断开 |
| 数据库本身 | 数据库代码错误 |
| 数据库溢出 |
| TCP连接错误 | 连接 | 连接超时 |
| 连接断开 |
| 其他TCP错误 | Socket自身错误 |
| 系统部分自定义错误 | 权限错误 | 管理员权限设置故障 |
| 输入错误 | 用户ID错误/为空 |
| 用户密码错误/为空 |
| 验证码错误/为空 |
| 查找错误 | 为查找到符合要求的记录 |
| 链接错误 | 内部连接错误 | 网页链接错误 |
| 文件/图片链接错误 |
| 外部链接错误 | 页面休整或者其他错误无法打开 |
| 友情连接网址维护或其他错误无法打开 |

## 7.2补救措施

说明故障出现后可能采取的变通措施，主要包括：

1. 对于软错误，需要在添加/修改操作中及时对输入数据进行验证，分析错误的类型，并且给出相应的错误提示语句，传送到客户端界面上；
2. 对于硬错误，错误类型不较少而且比较明确，所以可以在可能出错的地方中输出相应的出错语句，并将程序重置，最后返回输入阶段；
3. 后备技术说明准备采用的后备技术，当原始数据丢失时启用的副本的建立和启动的技术，例如周期性的把磁盘信息记录在案；
4. 恢复及在启动技术说明将使用的恢复再启动技术，是软件从故障点恢复执行或是软件从头开始重新开始运行的方法