

# 项目测试文档

## 1 功能

### 1.1 基本功能

用户可以注册账号，登录账号，退出登录。  
用户可以查看商家的详细信息，包括商家名称，菜单，菜品价格。  
用户可以将菜品添加到购物车，从购物车移除等。  
用户可以在确认订单页面，修改或新增收货地址信息并选择。  
用户可以添加多个收货信息，包括姓名，手机号，和地址。  
用户可以选择支付或取消支付。  
用户可以成功完成订单支付过程。  
用户可以查看历史订单信息，包括未支付订单和已支付订单。  
用户可以在未支付订单的支付入口进行支付。

### 1.2 特色功能

用户可以查看购物车中商品的名称，单价和数量。  
用户可以编辑个人信息包括修改头像，修改密码。  
用户可以退出登录。  
用户可以选择支付或取消支付。

## 2 前端项目设计

vue2 升级到 vue3：升级 node.js 支持 vue cli3，升级 vue.js 并更新相关依赖，安装运行迁移工具，逐一修改迁移中的警告，迁移全局和内部 API，逐步替换 vue2 的语法、组件等等。

在配置 index.js 文件时，由于使用的初始代码为 vue2 版本，但是最新下载的 vue-router 版本为 3，故出现报错，因此需要修改代码。如图 2-1 和 2-2。

```
import { createApp } from 'vue';
import App from '../App.vue';
//import router from './router';
import { createRouter, createWebHistory } from 'vue-router';
```

图 2-1

```
]const router = createRouter({
  history: createWebHistory(process.env.BASE_URL),
  routes
});

createApp(App).use(router).mount('#app');
```

图 2-2

添加新的收货地址的问题。修改前在新增收获地址或修改地址时会出现显示的用户信息与新增的不同的情况。

前端订单页面会出现多位小数点的情况，对于该问题使用toFixed函数将数字转换为指定小数位数的字符串。如图 3-3 所示。

```
<ul class="order-detailet" v-show="item.isShowDetailet">
  <li v-for="odItem in item.list">
    <p>{{ odItem.food.foodName }} x {{ odItem.quantity }}</p>
    <p>{{ (odItem.food.foodPrice * odItem.quantity).toFixed(2) }}</p>
  </li>
  <li>
    <p>配送费</p>
    <p>{{item.business.deliveryPrice}}</p>
  </li>
</ul>
```

图 2-3

对于用户一路点到底才能退出的问题，首页之后的页面增加了返回键。

增加了弹出层的组件(src/components/CartPopup.vue)，以便用户访问购物车并查看已添加的商品信息。

前端版本代码均存在商家列表/食品列表不能拉到底的问题，最底部会被底菜单栏遮挡。为解决该问题修改了商家列表部分的格式，即将margin改成了padding。如图 3-4 所示。

```
.wrapper .business {
  width: 100%;
  margin-top: 12vw;
  padding-bottom: 14vw;
}
```

图 2-4

### 3 后端项目设计

增加了订单业务逻辑对订单和购物车是否为空的验证  
com/neusoft/elmboot/service/impl/OrdersServiceImpl.java

用户修改密码

```
@RequestMapping("/modifyPassword")
public int modifyPassword(UserPassword userPassword) throws Exception {
    return userService.modifyPassword(userPassword);
}
```

用户修改头像

```
@RequestMapping("/changeUserPortrait")//base64
public int changeUserPortrait( UserImg userImg) throws Exception {
    return userService.changeUserPortrait(userImg);
}
```

删除用户

```
@RequestMapping("/removeUser")
public int removeUser(User user) throws Exception{
    return userService.removeUser(user);
}
```

更新用户

```
@RequestMapping("/updateUserById")
public int updateUserById(User user) throws Exception{
    return userService.updateUserById(user);
}
```

列出全部商家

```
@RequestMapping("/listAllBusiness")
public List<Business> listAllBusiness() throws Exception {
    return businessService.listAllBusiness();
}
```

订单的支付

```
@RequestMapping("/pay")
public void pay(Integer orderId) throws Exception{
    ordersService.pay(orderId);
}
```

更新支付状态

```
@RequestMapping("/turnOrderState")
public int turnOrderState(Orders orders) throws Exception {
    return ordersService.turnOrderState(orders.getOrderId());
}
```

修改了跨域处理方案，在允许本地进行访问的同时，部署上线。

com/neusoft/elmboot/WebMvcConfig\_01.java

```
src/main/java/com/neusoft/elmboot/WebMvcConfig_01.java
```