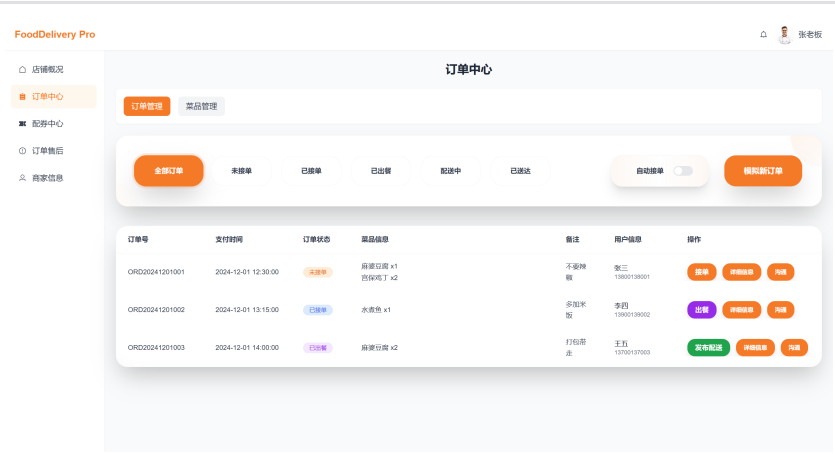
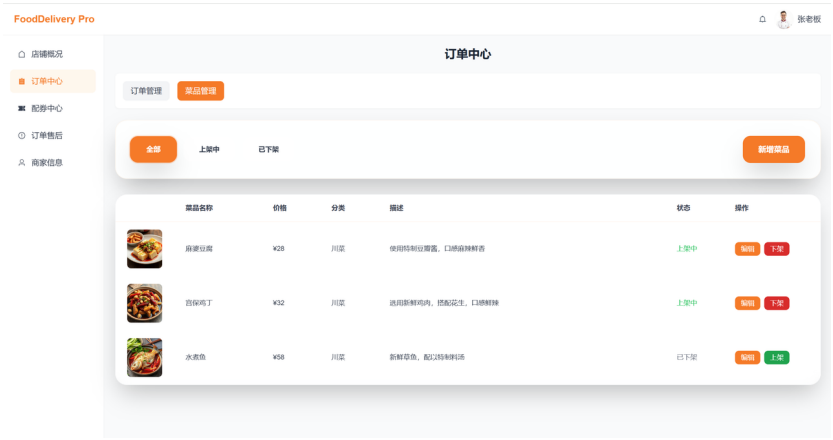
商家：订单中心

商家：订单中心

一、原型图及需求说明

板块	说明	截图
订单管理	● 订单状态筛选 提供全部订单、未接单、已接单、已出餐、配送中、已送达六个状态筛选按钮 ● 自动接单开关 开启后，新订单自动接单（状态从未接单变为已接单） ● 模拟新订单按钮 用于测试新订单流程 ● 订单列表 展示订单号、支付时间、订单状态、菜品信息、备注、用户信息、操作列 ● 操作按钮 根据订单状态显示不同操作按钮： - 未接单：显示"接单"按钮 - 已接单：显示"出餐"按钮 - 已出餐：显示"发布配送"按钮 - 配送中/已送达：仅显示状态文本 每个订单还有"详细信息"和"沟通"按钮	
菜品管理	● 菜品状态筛选	

	提供全部、上架中、已下架三个状态筛选按钮 ● 新增菜品按钮 点击弹出新增菜品表单 ● 菜品列表 展示菜品图片、名称、价格、分类、描述、状态、操作列 ● 操作按钮 编辑：弹出编辑表单 上架/下架：切换菜品状态	
新增/编辑菜品表单	● 表单字段 菜品名称、价格（数字）、分类（两级选择：先选大菜系，再选具体分类）、菜品图片（支持拖拽或点击上传）、描述 ● 图片上传 支持拖拽或点击上传，限制5MB以内，支持JPG/PNG ● 表单按钮 取消：关闭表单 创建/保存：提交表单	
订单详情对话框	● 客户信息 姓名、电话 ● 订单信息 下单时间、订单状态、送货地址 ● 商品明细 表格展示商品名称、规格、数量、单价、小计 ● 费用明细 商品原价、打包费、配送费、优惠券信息、优惠金额、实付金额 ● 底部按钮 关闭、打印订单	

沟通对话框

● 头部

显示用户信息（姓名、电话）和操作按钮（打电话、关闭）

● 消息列表

区分用户和商家消息，显示头像、消息内容、时间

● 输入区域

支持输入文字，可插入表情和常用语

● 表情面板

点击表情按钮弹出，选择后插入输入框

● 常用语

展示常用语按钮，点击插入输入框

● 发送按钮

三、数据需求

订单管理

变量名	类型	说明
`autoAcceptOrders`	boolean	自动接单开关状态
`selectedOrderStatus`	string	当前选中的订单状态（all, 未接单, 已接单, 已出餐, 配送中, 已送达）
`orders`	Order[]	订单列表
`filteredOrders`	Order[]	根据状态筛选后的订单列表
`orderStatuses`	Array<{value: string, label: string}>	订单状态筛选选项

订单类型定义：

代码块

```
interface Order {
  orderNo: string;
  payTime: string;
  status: '未接单' | '已接单' | '已出餐' | '配送中' | '已送达';
  remark: string;
  userInfo: UserInfo;
  dishes: OrderDish[];
}

interface UserInfo {
  name: string;
  phone: string;
  avatar?: string;
}

interface OrderDish {
```

```
    name: string;
    price: number;
    quantity: number;
  }
```

菜品管理

变量名	类型	说明
`selectedDishStatus`	string	当前选中的菜品状态（all, 上架中, 已下架）
`dishes`	Dish[]	菜品列表
`filteredDishes`	Dish[]	根据状态筛选后的菜品列表
`dishStatuses`	Array<{value: string, label: string}>	菜品状态筛选选项
`categoryMap`	Record<string, string[]>	菜品分类映射（大菜系到具体分类）
`selectedMainCategory`	string	新增菜品时选择的大菜系
`subCategories`	string[]	新增菜品时大菜系对应的具体分类列表
`selectedEditMainCategory`	string	编辑菜品时选择的大菜系
`editSubCategories`	string[]	编辑菜品时大菜系对应的具体分类列表
`newDish`	NewDishData	新增菜品表单数据
`editingDish`	Dish & { price: string, imagePreview: string }	编辑菜品表单数据（注意：价格用字符串便于输入，图片预览为base64）

菜品类型定义：

代码块

```
interface Dish {
  id: number;
  name: string;
  price: number;
  category: string;
  description: string;
  status: '上架中' | '已下架';
  image: string;
}

interface NewDishData {
  name: string;
  price: string; // 注意：表单中使用字符串类型
  category: string;
  description: string;
  imagePreview?: string; // 图片预览（base64）
}
```

订单详情对话框

变量名	类型	说明
`showOrderDetailsDialog`	boolean	是否显示订单详情对话框
`selectedOrder`	OrderDetail | null	当前选中的订单详情数据

订单详情类型定义：

代码块

```
interface OrderDetail extends Order {
  deliveryAddress: string;
  fees: {
    originalPrice: number;
    packingFee: number;
    deliveryFee: number;
    couponDiscount: number;
    finalPrice: number;
```

```
    couponName?: string;
  };
}
```

沟通对话框

变量名	类型	说明
`showChatDialog`	boolean	是否显示沟通对话框
`currentChatUser`	UserInfo | null	当前聊天的用户信息
`chatMessages`	ChatMessage[]	聊天消息列表
`newMessage`	string	新消息内容
`showEmojiPanel`	boolean	是否显示表情面板
`emojiList`	string[]	表情列表
`quickPhrases`	string[]	常用语列表

聊天消息类型定义：

代码块

```
interface ChatMessage {
  sender: 'user' | 'merchant';
  content: string;
  time: string;
}
```

四、接口说明

订单管理相关接口

1. **获取订单列表**
   - 接口：`getOrders(params?: { status?: 'all' | '未接单' | '已接单' | '已出餐' | '配送中' | '已送达' })`
   - 后端接口：`GET /orders`
   - 请求参数：`status`（可选，不传则获取全部）
   - 返回数据：`Order[]`

2. **获取订单详情**
   - 接口：`getOrderDetail(orderNo: string)`
   - 后端接口：`GET /orders/{orderNo}`
   - 请求参数：`orderNo`（订单号）
   - 返回数据：`OrderDetail`

3. **更新订单状态**
   - 接口：`updateOrderStatus(orderNo: string, status: '已接单' | '已出餐' | '配送中' | '已送达')`
   - 后端接口：`PATCH /orders/{orderNo}/status`
   - 请求体：`{ status }`
   - 返回数据：操作结果（通用结构）

4. **发布配送任务**
   - 接口：`publishDeliveryTask(orderNo: string)`
   - 后端接口：`POST /orders/{orderNo}/publish-delivery`
   - 请求体：无
   - 返回数据：操作结果（通用结构）

5. **获取自动接单状态**
   - 接口：`getAutoAcceptStatus()`
   - 后端接口：`GET /shop/auto-accept-status`
   - 返回数据：`{ autoAcceptOrders: boolean }`

6. **切换自动接单状态**
   - 接口：`toggleAutoAcceptOrders(enabled: boolean)`
   - 后端接口：`PATCH /shop/auto-accept`
   - 请求体：`{ enabled }`
   - 返回数据：操作结果（通用结构）

7. **模拟添加新订单**
   - 接口：`addNewOrder()`
   - 后端接口：`POST /orders/simulate-new`
   - 返回数据：`Order`（新创建的订单）

菜品管理相关接口

1. **获取菜品列表**
   - 接口：`getDishes(params?: { status?: 'all' | '上架中' | '已下架' })`
   - 后端接口：`GET /dishes`
   - 请求参数：`status`（可选，不传则获取全部）
   - 返回数据：`Dish[]`

2. **创建菜品**
   - 接口：`createDish(dishData: NewDishData)`
   - 后端接口：`POST /dishes`
   - 请求体：`{ name, price: number, category, description, status: '上架中', image }`
   - 返回数据：`Dish`（新创建的菜品）

3. **更新菜品信息**
   - 接口：`updateDish(dishId: number, dishData: { name?: string; price?: number; category?: string; description?: string; image?: string })`

- 后端接口：`PATCH /dishes/{dishId}`
- 请求体：部分菜品字段
- 返回数据：`Dish`（更新后的菜品）

4. **切换菜品上下架状态**
   - 接口：`toggleDishStatus(dishId: number, status: '上架中' | '已下架')`
   - 后端接口：`PATCH /dishes/{dishId}/toggle-status`
   - 请求体：`{ status }`
   - 返回数据：操作结果（通用结构）

5. **获取菜品分类数据**
   - 接口：`getDishCategories()`
   - 后端接口：`GET /dishes/categories`
   - 返回数据：`{ [key: string]: string[] }`（大分类到具体分类数组的映射）

6. **上传菜品图片**
   - 接口：`uploadDishImage(file: File)`
   - 后端接口：`POST /upload/dish-image`
   - 请求体：`FormData`（包含图片文件）
   - 返回数据：`{ imageUrl: string }`

沟通相关接口

1. **获取聊天记录**
   - 接口：`getChatMessages(orderNo: string)`
   - 后端接口：`GET /chat/order/{orderNo}/messages`
   - 返回数据：`ChatMessage[]`

2. **发送消息**
   - 接口：`sendMessage(orderNo: string, content: string)`

- 后端接口：`POST /chat/order/{orderNo}/send`
- 请求体：`{ content, sender: 'merchant', time: string }`
- 返回数据：`ChatMessage`（发送的消息）

3. **模拟用户拨号**
   - 接口：`callUser(phone: string)`
   - 后端接口：`POST /communication/call`
   - 请求体：`{ phone }`
   - 返回数据：操作结果（通用结构）

通用接口

1. **错误处理**
   - 方法：`handleApiError(error: any)`
   - 功能：将API错误转换为友好的错误消息字符串

请求和响应拦截器

- 请求拦截器：自动添加认证token（从localStorage的'merchant_token'获取）
- 响应拦截器：统一处理错误，特别是401错误（清除token并跳转登录页）

注：所有接口返回的数据结构外层没有固定包裹（如`{ data: ... }`），直接返回数据对象或数组。错误处理由响应拦截器统一处理，并抛出带有友好错误消息的异常。