

项目方案设计书

项目需求：开发一个在线订餐 SaaS 系统，支持扫码点餐、菜单管理、在线支付、订单流转与基础经营报表，面向中小餐饮门店。

说明：本文档为 AI 团队产出的**方案设计书**示例，用于展示 AgentHub 的产物形态与深度；正式使用时由 10+ 位 AI Agent 依据你的真实需求协作生成。

● 售前顾问 需求澄清与项目定位

项目概述

为中小餐饮门店提供一套轻量的在线订餐 SaaS。顾客扫桌台二维码即可浏览菜单、下单并在线支付；商家在后台管理菜单、接收订单、查看经营数据。目标是降低门店点餐人力成本、提升翻台效率。

关键澄清结论

- 交付目标：可运行 Web 应用（响应式，顾客端 H5 + 商家后台）。
- 规模假设：单店日均 300-800 单，初期支持百级门店多租户。
- 支付方式：接入微信支付（JSAPI），预留支付宝扩展。
- 边界（MVP 不含）：会员积分、营销满减、外卖配送对接，列入二期。

● 产品经理 功能规格

核心功能模块

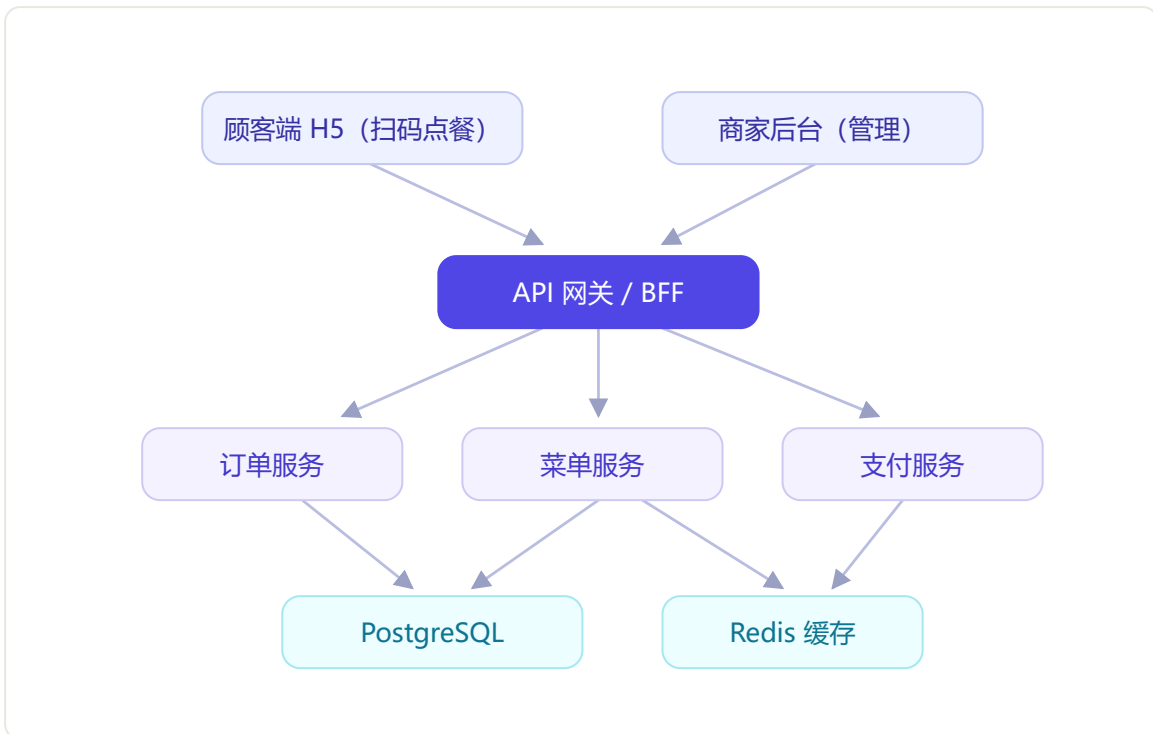
模块	顾客端	商家后台
菜单	分类浏览、规格选择、加购	菜品增删改、上下架、库存
订单	下单、支付、订单状态查询	接单、出餐、订单流转
桌台	扫码绑定桌号	桌台二维码生成与管理
报表	—	营业额、热销菜品、时段分析

关键用户流程

扫码 → 选桌 → 浏览菜单 → 加购 → 提交订单 → 微信支付 → 后厨接单出餐 → 顾客查看进度。

● 架构师 系统架构

整体架构



技术选型

层	选型	理由
前端	Next.js 14 + Tailwind	SSR 利于 H5 首屏与 SEO, 一套代码响应式
后端	Node.js + NestJS	模块化清晰, TypeScript 全栈一致
数据库	PostgreSQL + Redis	事务可靠 + 热点缓存 (菜单/桌台)
支付	微信支付 JSAPI	覆盖目标场景, 预留多渠道抽象
部署	Docker + Nginx	一键编排, 便于上云

● 后端工程师 接口与服务设计

核心接口 (节选)

POST /api/orders	创建订单（桌号、菜品、规格）
GET /api/orders/:id	查询订单状态
POST /api/pay/wechat	发起微信支付（返回 JSAPI 参数）
POST /api/pay/callback	支付结果回调（验签 + 幂等）
GET /api/menu?storeId=	获取门店菜单（Redis 缓存）
PUT /api/menu/:dishId	上下架 / 改价（商家鉴权）

关键约定

- 支付回调验签 + 订单状态机幂等，杜绝重复发货。
- 多租户隔离：所有查询强制带 storeId，行级隔离。

● 数据库设计师 数据模型

表	关键字段	说明
stores	id, name, owner_id	门店（租户）
tables	id, store_id, no, qr_token	桌台与二维码
dishes	id, store_id, name, price, status	菜品
orders	id, store_id, table_id, status, amount	订单主表
order_items	order_id, dish_id, qty, spec	订单明细
payments	order_id, channel, trade_no, paid_at	支付流水

● 测试工程师 测试策略

- 单元：金额计算、订单状态机、支付验签。
- 集成：下单→支付→回调全链路（含重复回调幂等）。
- 压测：单店高峰并发下单与菜单缓存命中。

● 项目统筹 一页纸交付总结

交付清单

- 可运行项目代码（顾客端 H5 + 商家后台 + 后端服务）。
- PostgreSQL 建表脚本与种子数据。
- QUICKSTART 启动指南（本地一键跑通）+ 可落地性体检报告。

落地路线

一期 4-6 周交付 MVP (扫码点餐 + 支付 + 后台) ; 二期补会员/营销/外卖对接。

由 AgentHub 生成 · 10+ 位 AI Agent 协作 · 方案设计书 (示例)