# 微信外卖聚合平台小程序

## 一、需求分析

项目名称：校园外卖小程序聚合平台  
作者：丁宇涵  
时间：2025年6月8日

**1.项目背景**

当前高校外卖市场活跃，但大多数校内商家各自开发了独立的微信小程序，导致同学们在查找、下单时需要反复切换多个小程序。信息分散不仅影响用户体验，也降低了校内商家的曝光度。因此，本项目旨在开发一款聚合各类外卖小程序链接的微信小程序，统一入口、简化查找流程，提升用户点餐效率。

**2.项目目标**

开发一款微信小程序，聚合校内所有支持微信点餐的商家小程序链接，用户可在本平台浏览、搜索并点击跳转原商家小程序进行下单，提升用户体验，方便校内生活。

**3.用户需求（以目标用户视角描述）**

本项目面向的用户为校内学生群体。经过模拟访谈与调研，用户提出如下核心需求：

能够在一个地方查看所有校内外卖商家；

能按分类（如奶茶、烧烤、快餐等）浏览商家；

能搜索关键词查找目标商家；

商家信息一目了然（包括图标、营业时间、简介等）；

点进商家可直接跳转其原有小程序；

页面简洁美观，操作流畅；

不需要强制登录或授权；

后期可扩展收藏、热门推荐、反馈入口等功能。

**4.功能性需求（Functional Requirements）**

| 编号 | 名称 | 描述 |
| --- | --- | --- |
| F1 | 首页商家展示 | 首页展示所有已录入商家卡片，含 logo、名称、起送价、状态等 |
| F2 | 分类浏览 | 用户可通过“全部 / 奶茶 / 小吃 / 快餐”等分类筛选 |
| F3 | 商家跳转 | 点击任一商家卡片，可通过 wx.navigateToMiniProgram 跳转其原小程序 |
| F4 | 搜索功能 | 输入关键字模糊匹配商家名称与标签 |
| F5 | 营业状态提醒 | 商家打烊/暂停营业时卡片置灰或加提示文案 |
| F6 | 自适应展示 | 页面适配不同手机尺寸，支持上下滑动加载更多商家 |

**5.非功能性需求（Non-Functional Requirements）**

| 编号 | 名称 | 描述 |
| --- | --- | --- |
| N1 | 响应性能 | 页面跳转与加载时间在 2 秒以内 |
| N2 | UI风格 | 界面风格简约，颜色统一，参考 Vant Weapp 风格 |
| N3 | 兼容性 | 兼容 iOS 和 Android 微信客户端 |
| N4 | 操作简便 | 所有功能无需学习成本，1-2 步完成查找与跳转 |
| N5 | 用户隐私 | 无需强制用户授权登录即可使用基础功能 |

**6.可选扩展需求（Optional Requirements）**

| 编号 | 名称 | 描述 |
| --- | --- | --- |
| O1 | 收藏功能 | 用户可收藏常用商家 |
| O2 | 热门推荐 | 首页展示“今日热店” |
| O3 | 商家提交入口 | 提供商家自行提交信息的表单 |
| O4 | 管理后台 | 后台支持手动维护商家列表与数据更新 |

**7.项目边界**

本项目不包含用户下单系统（由原商家小程序完成）

本平台仅做跳转与聚合展示

商家数据将由开发者人工录入或调用数据库维护

**8.验收标准（Acceptance Criteria）**

| 编号 | 条件 |
| --- | --- |
| A1 | 能成功展示不少于10个商家卡片 |
| A2 | 能根据分类和搜索准确筛选商家 |
| A3 | 商家点击后可成功跳转至其微信小程序 |
| A4 | 页面布局简洁，逻辑清晰，能流畅使用 |
| A5 | 页面适配不同手机，兼容主流微信版本 |

## 二、总体设计

**2.1 系统架构**

本系统采用微信小程序前端 + 微信云开发后端的架构设计，系统结构清晰，易于维护与扩展。

微信小程序前端：负责页面展示、用户交互、调用云函数和跳转接口，实现用户友好的操作体验。

微信云开发：利用微信云数据库存储商家信息，云函数处理业务逻辑和数据请求，无需自行搭建服务器，降低运维成本。

微信API接口：利用wx.navigateToMiniProgram实现无缝跳转至目标外卖小程序。

**2.2 模块划分及功能说明**

**2.2.1 首页模块**

功能：

1加载并展示所有入驻的外卖商家卡片

2支持商家分类标签筛选（如奶茶、快餐、零食等）

3搜索框支持关键字查询商家

4显示商家名称、Logo、起送价、营业状态等信息

技术实现：

1页面渲染采用WXML+WXSS+JavaScript

2通过云函数从云数据库拉取数据，前端分页加载优化性能

3使用微信小程序组件实现列表和筛选控件

**2.2.2 商家列表模块**

功能：

1根据首页筛选条件，展示对应商家详细列表

2支持分页加载，避免一次加载大量数据导致卡顿

3实时显示商家当前营业状态（营业/休息）

技术实现：

1云数据库查询带条件分页（limit、skip）

2状态字段实时更新，前端定时或手动刷新支持

**2.2.3 跳转模块**

功能：

1用户点击商家卡片后，调用微信官方跳转接口，打开对应商家的外卖小程序

2保证跳转体验流畅，快速打开目标小程序

技术实现：

1使用 wx.navigateToMiniProgram({ appId: 商家AppID }) 实现跳转

2跳转参数校验，避免无效跳转

**2.2.4 后台管理模块（未来可扩展）**

功能：

1由管理员维护商家信息，支持增删改查操作

2保障商家数据的准确性和及时更新

技术实现：

1可基于微信小程序开发管理端，或使用简单的Web后台

2利用云函数实现数据管理接口

**2.3 数据设计与存储**

商家数据结构示例

| 字段名 | 类型 | 说明 |
| --- | --- | --- |
| id | String | 唯一标识（自动生成） |
| name | String | 商家名称 |
| logoUrl | String | 商家Logo图片URL |
| appId | String | 商家小程序AppID |
| category | String | 商家类别（奶茶、快餐等） |
| deliveryStart | Number | 起送价 |
| isOpen | Boolean | 当前是否营业 |
| description | String | 简短介绍 |
| lastUpdated | Timestamp | 信息最后更新时间 |

**2.4 核心业务流程**

1用户打开小程序，首页通过云函数拉取商家数据并渲染列表。

2用户选择分类或输入搜索关键词，小程序向云函数请求筛选后的数据。

3用户浏览商家列表，可查看商家营业状态及起送价信息。

4点击某个商家卡片，调用 wx.navigateToMiniProgram 跳转至对应商家外卖小程序。

5用户完成点餐后返回，继续浏览或退出。

**2.5 创新点与优势**

聚合统一入口：解决校园外卖小程序分散难找问题，一站式访问所有商家，提高便捷度和效率。

智能筛选与搜索：分类和搜索功能让用户快速定位所需商家，节省时间。

动态状态展示：商家营业状态实时显示，避免无效点击。

无缝跳转体验：利用微信官方API，确保跳转快速稳定，用户体验极佳。

轻量云端架构：无需复杂服务器，快速开发上线，降低维护成本，适合校园场景。

易扩展后台管理：方便管理员维护商家数据，支持未来功能升级。

## 三、详细设计

**3.1 页面设计**

**3.1.1 首页页面设计**

功能

1显示商家分类标签（如奶茶、快餐、甜点等）

2显示热门商家推荐区

3商家列表展示，卡片样式包含商家logo、名称、起送价和营业状态

4顶部搜索栏，支持输入关键词模糊搜索商家

布局说明

1顶部搜索栏固定位置，方便随时搜索

2分类标签横向滚动条，点击筛选对应分类商家

3商家卡片列表垂直排列，支持分页加载

4卡片颜色或图标区分商家营业状态（绿色为营业，灰色为休息）

交互设计

1点击分类标签时刷新列表，展示对应分类商家

2输入搜索关键词时触发搜索请求，动态更新列表

3点击商家卡片触发跳转模块，打开商家小程序

**3.1.2 商家列表页设计（可作为首页内容的延展）**

功能

1显示符合筛选条件的全部商家

2支持分页或下拉加载更多

3显示商家详细信息简介

布局说明

1标题栏显示当前分类或搜索关键词

2列表项显示商家logo、名称、起送价、营业状态及简介

3列表底部显示加载状态提示

交互设计

1上拉加载触发下一页数据请求

2点击列表项调用跳转模块

**3.2 跳转实现**

技术细节

利用微信小程序官方API wx.navigateToMiniProgram 实现跳转，代码示例如下：

wx.navigateToMiniProgram({

appId: targetAppId, // 目标商家小程序AppID，从数据库获取

path: '', // 可选，跳转目标小程序的具体页面路径

extraData: {}, // 可选，传递给目标小程序的数据

envVersion: 'release', // 打开正式版

success(res) {

console.log('跳转成功');

},

fail(err) {

console.error('跳转失败', err);

}

});

跳转流程

1用户点击商家卡片

2读取该商家的 appId

3调用上述API完成跳转

4跳转失败时给出提示

异常处理

1若 appId 为空或格式错误，提示“暂时无法打开该商家”

2跳转失败后可提示“请检查微信版本或网络环境”

**3.3 数据结构设计**

数据库采用微信云开发的云数据库，设计“商家”集合（Collection），字段示例如下：

| 字段名 | 类型 | 说明 |
| --- | --- | --- |
| \_id | String | 云数据库自动生成的唯一ID |
| name | String | 商家名称 |
| logoUrl | String | 商家Logo图片链接 |
| appId | String | 商家小程序AppID |
| category | String | 商家类别，如“奶茶”、“快餐” |
| deliveryStart | Number | 起送价格 |
| isOpen | Boolean | 当前是否营业 |
| description | String | 商家简介 |
| updateTime | Date | 信息更新时间 |

**3.4 交互流程设计**

用户打开首页

1页面加载时调用云函数拉取商家列表数据

2显示默认分类（如“全部”）商家

3用户操作分类筛选

4点击分类标签触发页面刷新，调用云函数带参数查询对应分类商家

用户使用搜索功能

1输入关键词，触发防抖后调用云函数，返回模糊匹配的商家列表

用户点击商家卡片

1触发跳转模块，调用 wx.navigateToMiniProgram 跳转

异常或错误提示

1跳转失败时弹出toast提示

2网络异常时显示重试按钮

**3.5 其他设计细节**

分页加载设计

每次拉取20条数据，支持滚动到底部自动加载下一页

采用云数据库limit和skip实现分页查询

数据缓存优化

首页数据可做本地缓存，减少网络请求

切换分类时清理缓存，保持数据新鲜度

响应式布局

兼容不同屏幕尺寸，保证良好展示效果