# 同城美食分享小程序

# 需求分析与概要设计

## 项目说明

## 项目目标：

开发一款同城美食分享小程序，用户可以浏览附近的美食店铺信息、查看具体菜品、分享美食体验并进行评分。系统将通过爬虫技术自动获取主流美食平台（如大众点评、美团）的店铺名称、具体美食、价格、评分等信息，并结合用户生成内容(UGC)构建完整的美食数据库。

## 软硬件环境需求

**软件环境：**

* 前端：微信小程序框架
* 后端：Node.js/Python (Django/Flask)
* 数据库：MySQL/MongoDB
* 爬虫服务器：Python (Scrapy/BeautifulSoup)
* 云服务：阿里云/腾讯云

**硬件环境：**

* 服务器：2核4G以上配置
* 爬虫服务器：独立IP池，反爬虫绕过能力

## 使用的关键技术

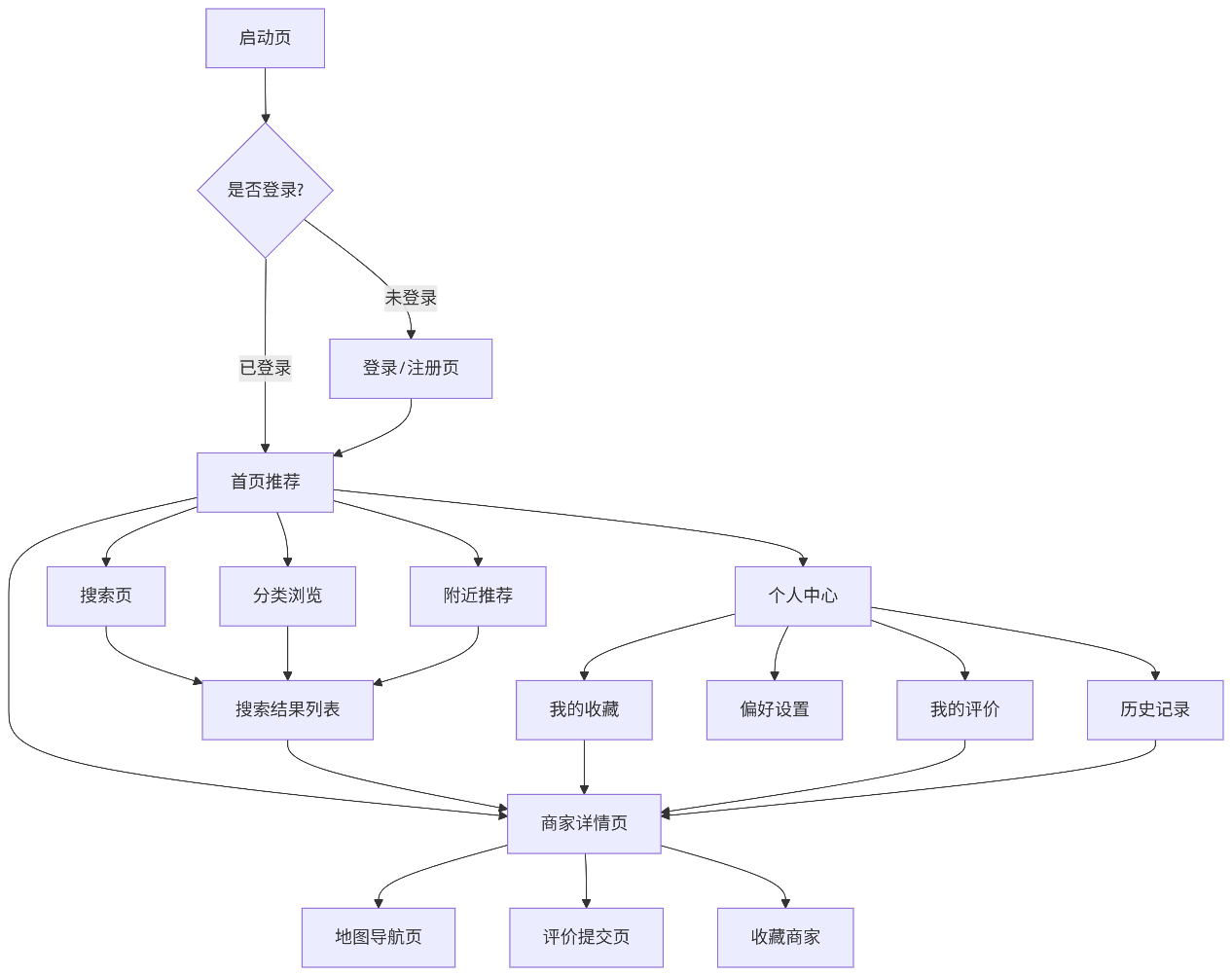
1. **网络爬虫技术**：
   * 使用Scrapy框架构建分布式爬虫系统
   * 动态页面渲染处理(Puppeteer/Selenium)
   * 反爬虫绕过机制(IP轮换、请求头模拟、验证码识别)
2. **小程序开发**：
   * 微信小程序原生开发或Taro跨平台框架
   * 地图API集成(腾讯地图/高德地图)
3. **数据处理**：
   * 数据清洗与去重
   * 情感分析(用户评论处理)
   * 推荐算法(基于用户行为的协同过滤)

## 需求分析

## 系统用例

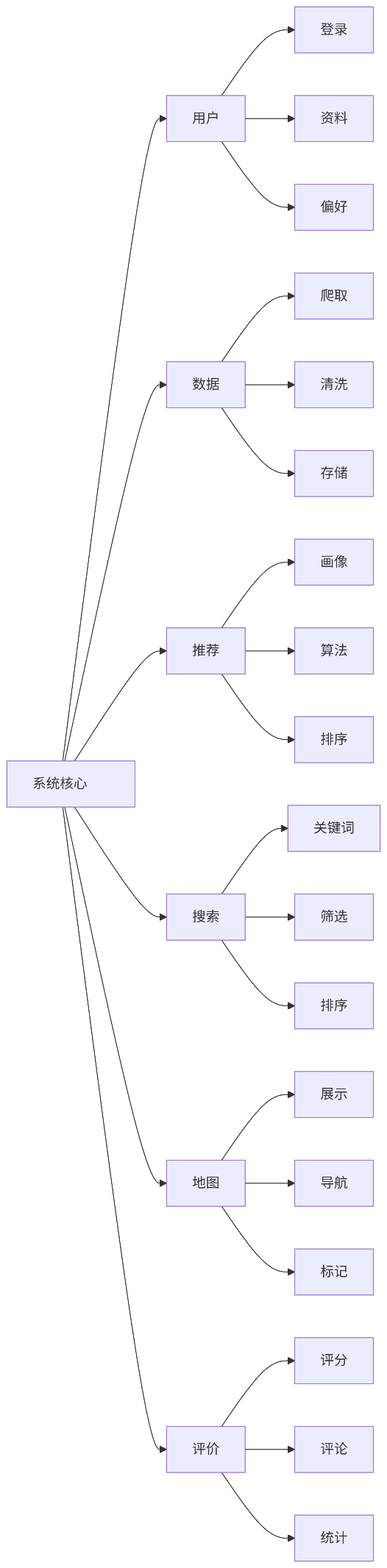
1. **浏览美食信息**
   * **参与者**：用户
   * **基本事件流**：用户打开小程序，系统显示附近美食店铺列表，用户可点击查看店铺详情(包括菜品、价格、评分等)
2. **搜索美食**
   * **参与者**：用户
   * **基本事件流**：用户输入关键词(如"火锅")，系统返回匹配的店铺和菜品列表
3. **分享美食体验**
   * **参与者**：注册用户
   * **基本事件流**：用户对已消费的店铺进行评价、上传照片、打分
4. **管理爬虫任务**
   * **参与者**：管理员
   * **基本事件流**：管理员配置爬虫目标网站、爬取频率、数据字段等参数
5. **数据清洗与整合**
   * **参与者**：系统
   * **基本事件流**：系统自动对爬取的数据进行去重、格式标准化、与现有数据合并

## 业务流程



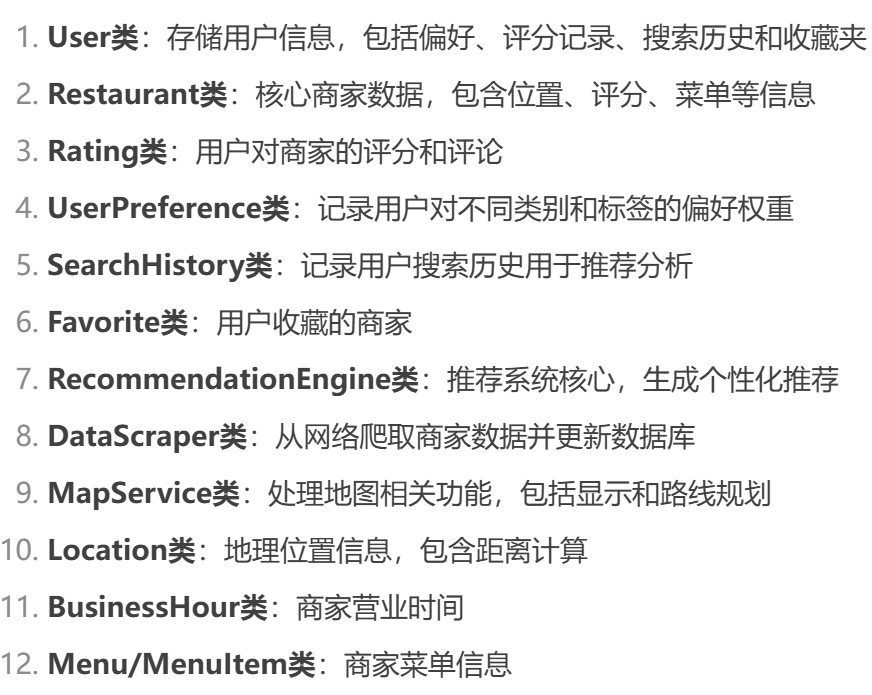
## 概要设计

## 功能模块设计



1. **爬虫管理模块**
   * **输入**：目标网站URL、爬取规则配置
   * **输出**：结构化美食数据
   * **功能概述**：负责从各美食平台抓取店铺信息、菜单、评分等数据
2. **数据处理模块**
   * **输入**：原始爬取数据
   * **输出**：清洗后的标准化数据
   * **功能概述**：数据去重、字段标准化、异常值处理
3. **美食展示模块**
   * **输入**：用户位置、筛选条件
   * **输出**：美食列表和详情
   * **功能概述**：基于地理位置展示附近美食，支持多种排序和筛选
4. **用户互动模块**
   * **输入**：用户评价、评分、照片
   * **输出**：整合后的店铺评分和评论
   * **功能概述**：管理用户生成内容，计算综合评分
5. **推荐模块**
   * **输入**：用户行为数据
   * **输出**：个性化推荐列表
   * **功能概述**：基于用户历史行为和偏好推荐美食

## 核心类图



## 界面设计

**界面设计**

1. **首页**：
   * 地图模式展示附近美食
   * 列表模式展示热门店铺
   * 搜索框和筛选条件
2. **店铺详情页**：
   * 店铺基本信息(名称、地址、评分)
   * 菜单展示
   * 用户评价列表
   * 分享按钮
3. **个人中心**：
   * 我的收藏
   * 我的评价
   * 浏览历史
4. **管理员后台**：
   * 爬虫任务管理
   * 数据质量监控
   * 用户管理