Document-CaseA-LLM-03

组别：大模型

第一章 引言

1.1 目的

本文档旨在为 QuickEats 餐饮及外卖软件应用的功能需求提供全面的描述，明确系统设计的目标、范围和具体实现需求。通过本文档，开发团队、测试人员及相关利益方能够清晰了解 QuickEats 的核心功能、技术架构以及用户体验设计原则，从而确保项目顺利实施并满足用户需求。

1.2 范围

本文档覆盖了 QuickEats 的所有主要功能模块，包括餐厅与菜品浏览、购物车管理、订单处理、支付集成、实时订单跟踪以及评价反馈系统。此外，文档还涉及系统的非功能性需求（如性能、安全性和可用性）以及与其他第三方服务的集成方式。本文档适用于产品开发、测试、运维及相关业务团队。

1.3 定义、缩写与缩略语

- QuickEats：一款专注于速度与便利性的餐饮及外卖软件。

- API：应用程序编程接口，用于不同服务之间的通信。

- GPS：全球定位系统，用于实时跟踪骑手位置。

- UI/UX：用户界面/用户体验，指软件的视觉设计与交互流程。

- WebSocket：一种支持全双工通信的网络协议，用于实现实时数据传输。

1.4 参考文献

- IEEE 830-1998: 软件需求规格说明书标准

- 《用户体验要素》 by Jesse James Garrett

- 《微服务架构设计模式》 by Chris Richardson

---

第二章 总体概述

2.1 产品视角

QuickEats 是一款基于分布式微服务架构的餐饮及外卖平台，致力于为用户提供快速、便捷的点餐体验。系统从前端到后端均采用模块化设计，支持跨平台访问（iOS、Android 和 Web）。通过整合第三方服务（如地图导航和支付网关），QuickEats 实现了从餐厅选择到订单完成的全流程闭环服务。未来，QuickEats 还计划引入人工智能推荐引擎和大数据分析工具，以进一步提升个性化服务水平。

2.2 产品功能

QuickEats 的主要功能模块包括：

1. 餐厅与菜品浏览：支持分类筛选、关键词搜索和智能推荐。

2. 购物车与订单管理：允许用户添加、编辑和提交订单，并选择自取或外卖模式。

3. 实时订单跟踪：通过 GPS 技术展示骑手位置和预计送达时间。

4. 支付与结算：集成多种支付方式，保障交易安全。

5. 评价与反馈：用户可对餐厅和服务进行评分，帮助优化服务质量。

6. 通知系统：通过推送消息提醒用户订单状态更新。

2.3 用户特征

QuickEats 的目标用户群体具有以下特征：

- 白领：工作繁忙，注重效率，倾向于使用手机快速完成点餐。

- 学生：预算有限，偏好性价比高的餐品，同时需要灵活的送餐时间。

- 家庭用户：关注食品卫生与营养搭配，通常订购多人份餐品。

- 餐厅经营者：希望通过数字化工具提高订单管理效率，并扩大客户群。

2.4 竞品分析

当前市场上已有多个成熟的餐饮及外卖平台（如美团、饿了么），它们在功能上较为完善，但在用户体验和智能化方面仍有改进空间。QuickEats 的差异化优势在于：

- 更加简洁直观的 UI/UX 设计，降低用户学习成本。

- 基于 AI 的个性化推荐算法，精准匹配用户口味。

- 高效的实时订单跟踪系统，减少用户等待焦虑。

- 强调数据隐私保护，增强用户信任感。

2.5 设计约束

- 性能要求：系统需支持每秒数千次并发请求，保证高峰期稳定运行。

- 兼容性：需适配主流操作系统（iOS、Android）及浏览器版本。

- 安全性：所有用户数据必须加密存储，支付环节需符合 PCI-DSS 标准。

- 法律合规：遵守中国相关法律法规，特别是关于数据隐私和食品安全的规定。

2.6 假设与依赖关系

- 假设条件：

- 用户设备具备基本的硬件配置（如 GPS 模块和互联网连接）。

- 第三方服务（如支付网关和地图 API）始终可用且响应及时。

- 依赖关系：

- 开发团队需熟练掌握微服务架构和前端框架（如 React Native 或 Flutter）。

- 测试环境需模拟真实用户行为，以验证系统的稳定性和性能表现。

- 数据中心需提供高可用性和容灾能力，以应对突发流量高峰。

第三章 功能需求

本章详细描述 QuickEats 的核心功能，每个功能点均包含功能描述、输入和输出。功能粒度控制为单一功能，确保清晰且可实现。

---

3.1 餐厅与菜品浏览

- 描述：用户可以通过分类筛选、关键词搜索或推荐列表快速找到附近的餐厅和热门菜品。

- 输入：

- 用户输入的搜索关键词（如餐厅名称、菜品类别）。

- 系统定位信息（GPS 坐标）。

- 用户偏好数据（历史行为、收藏记录等）。

- 输出：

- 按优先级排序的餐厅列表（包括餐厅名称、评分、距离、预计送达时间）。

- 菜品详情（图片、价格、描述、用户评价）。

---

3.2 餐厅筛选与排序

- 描述：用户可根据多种条件（如配送时间、评分、价格区间）对餐厅进行筛选和排序。

- 输入：

- 用户选择的筛选条件（如“免费配送”、“评分 >4.5”）。

- 用户选择的排序方式（如按配送时间、评分、价格）。

- 输出：

- 符合筛选条件并按指定规则排序的餐厅列表。

---

3.3 购物车添加菜品

- 描述：用户可以将选中的菜品添加到购物车，并查看购物车中的内容。

- 输入：

- 用户点击的菜品 ID 和数量。

- 输出：

- 更新后的购物车内容（包括菜品名称、单价、数量、小计金额）。

---

3.4 购物车编辑

- 描述：用户可以在购物车中调整菜品的数量或移除菜品。

- 输入：

- 用户的操作类型（增加、减少、删除）。

- 目标菜品 ID 和操作后的数量。

- 输出：

- 更新后的购物车内容（包括总金额）。

---

3.5 订单提交

- 描述：用户确认购物车内容后提交订单，并选择配送方式（自取或外卖）。

- 输入：

- 用户选择的配送方式（自取/外卖）。

- 用户填写的送餐地址（仅限外卖模式）。

- 用户选择的支付方式。

- 输出：

- 生成的订单编号及订单详情（包括餐厅信息、菜品清单、总价）。

---

3.6 支付处理

- 描述：用户通过系统集成的支付网关完成订单支付。

- 输入：

- 用户选择的支付方式（在线支付/货到付款）。

- 支付金额及订单编号。

- 输出：

- 支付结果（成功/失败）及交易凭证（如支付流水号）。

---

3.7 实时订单跟踪

- 描述：用户可在应用中实时查看骑手位置和预计送达时间。

- 输入：

- 订单编号。

- 骑手的 GPS 定位信息（由第三方服务提供）。

- 输出：

- 骑手的实时位置（地图标记）及预计送达时间。

---

3.8 订单状态更新

- 描述：系统根据订单处理进度自动更新订单状态，并通知用户。

- 输入：

- 订单编号。

- 当前订单状态（如“已接单”、“配送中”、“已完成”）。

- 输出：

- 更新后的订单状态信息，并通过推送通知告知用户。

---

3.9 评价与反馈

- 描述：用户在订单完成后可对餐厅和服务进行评分和评论。

- 输入：

- 用户输入的评分（1-5 星）。

- 用户填写的评论内容（可选）。

- 订单编号。

- 输出：

- 存储至数据库的评价记录，并更新餐厅的综合评分。

---

3.10 推荐系统

- 描述：系统基于用户的历史行为和偏好，智能推荐个性化的餐厅和菜品。

- 输入：

- 用户的历史订单记录、浏览行为、收藏记录。

- 输出：

- 推荐的餐厅列表或菜品列表，附带推荐理由（如“您常点的菜系”）。

---

3.11 通知系统

- 描述：系统通过推送消息提醒用户订单状态更新或其他重要信息。

- 输入：

- 通知类型（如订单状态更新、促销活动提醒）。

- 用户 ID 或设备标识符。

- 输出：

- 用户收到的推送通知，显示具体内容（如“您的订单已配送中”）。

在之前的回答中，我已经详细列出了 QuickEats 的核心功能需求，并确保每个功能点的输入、输出和描述清晰明确。为了进一步确保功能需求的完整性，我将从以下几个方面进行补充和完善：

3.12 用户注册与登录

- 描述：用户可以通过手机号、邮箱或第三方账号（如微信、支付宝）完成注册和登录。

- 输入：

- 用户提供的手机号或邮箱地址。

- 用户设置的密码（仅限手机号/邮箱注册）。

- 第三方授权信息（如微信 OpenID）。

- 输出：

- 注册成功后生成的用户唯一标识符（UserID）。

- 登录成功后的会话令牌（Session Token），用于后续身份验证。

第四章 外部接口

本章描述 QuickEats 系统与外部交互的接口设计，涵盖用户接口、硬件接口、软件接口和通信接口，确保系统能够无缝对接各类外部实体。

---

4.1 用户接口

系统通过直观的用户界面与用户进行交互，包括餐厅浏览、菜品选择、订单跟踪等功能。用户可通过屏幕查看餐厅详情、购物车内容及订单状态，并接收语音提示或打印小票等辅助信息。界面设计注重用户体验，确保操作流畅且信息清晰。

---

4.2 硬件接口

系统支持与多种外部硬件设备的交互，如热敏打印机、GPS 模块等。

通过标准通信协议（如 USB、蓝牙、Wi-Fi），系统可向打印机发送订单信息生成小票，并从 GPS 设备获取骑手实时位置数据，从而实现配送过程的精准管理。

---

4.3 软件接口

系统与其他软件平台和服务紧密集成，包括支付网关、数据分析工具和第三方推送服务。

通过标准化的 API 接口，系统能够完成支付处理、数据导出以及用户行为分析等功能，

同时兼容主流支付平台（如微信支付、支付宝）和消息推送服务（如 FCM、APNs）。

---

4.4 通信接口

系统利用多种通信方式与用户和其他系统保持高效连接。通过邮件、短信和推送通知，系统能够在关键节点（如下单成功、配送到达）及时告知用户最新动态。

此外，系统通过 WebSocket 实现骑手位置的实时更新，确保配送过程透明且可控。

第五章 非功能需求

本章概述 QuickEats 系统在性能、安全性、可用性等方面的非功能需求，确保系统稳定高效运行。

---

5.1 性能需求

- 系统响应迅速，关键操作（如搜索、下单）不超过 2 秒，骑手位置更新延迟控制在 1 秒内。

- 支持高并发，可同时处理 10,000 用户在线和每秒 500 笔订单。

---

5.2 安全性需求

- 敏感数据加密存储与传输，使用 HTTPS 协议。

- 用户登录需多因素认证，角色权限严格控制。

- 防范攻击，定期扫描漏洞并修复隐患。

---

5.3 可用性需求

- 系统全年可用性达 99.9%，核心服务支持故障自动切换。

- 数据分布式存储，灾难后 30 分钟内恢复核心功能。

---

5.4 可扩展性需求

- 模块化设计，功能独立部署，便于扩展和维护。

- 动态扩容应对流量高峰，API 接口保持向后兼容。

---

5.5 用户体验需求

- 界面简洁直观，支持多语言切换和无障碍设计。

- 提供实时反馈（如加载动画），提升用户操作感知。

---

5.6 法规与合规需求

- 符合隐私保护法规，支付流程安全合规。

- 自动审核用户生成内容，过滤违规信息。

---

5.7 可维护性需求

- 记录关键操作日志至少 6 个月，集成监控工具实时跟踪系统状态。

- 异常时及时报警，提供详尽文档支持运维和开发。

以下是 QuickEats 系统的几个核心用例及其详细描述，包括用例名称、编号、参与者、前置条件、后置条件、主事件流和异常事件流。

---

用例 1：用户登录

- 用例名称：用户登录

- 用例编号：UC-001

- 参与者：普通用户

- 前置条件：

- 用户已注册账号。

- 系统网络连接正常。

- 后置条件：

- 用户成功登录系统并进入主界面。

- 主事件流：

1. 用户打开应用并进入登录页面。

2. 用户输入手机号或邮箱及密码。

3. 系统验证用户输入信息是否正确。

4. 验证通过后，系统跳转至主界面。

- 异常事件流：

- A1：用户输入的账号或密码错误。

- 系统提示“账号或密码错误，请重试”。

- 用户重新输入信息。

- A2：网络连接中断。

- 系统提示“网络连接失败，请检查网络设置”。

- 用户检查网络后重试。

---

用例 2：下单点餐

- 用例名称：下单点餐

- 用例编号：UC-002

- 参与者：普通用户、餐厅系统

- 前置条件：

- 用户已登录系统。

- 餐厅处于营业状态且有可选菜品。

- 后置条件：

- 订单生成成功，用户收到订单确认信息。

- 主事件流：

1. 用户浏览餐厅菜单并选择菜品。

2. 用户将菜品加入购物车并确认订单详情。

3. 用户选择支付方式并完成支付。

4. 系统生成订单并向餐厅发送通知。

5. 用户收到订单确认信息（如订单号、预计送达时间）。

- 异常事件流：

- A1：用户余额不足或支付失败。

- 系统提示“支付失败，请更换支付方式或充值”。

- 用户重新选择支付方式。

- A2：餐厅库存不足。

- 系统提示“所选菜品已售罄，请重新选择”。

- 用户修改订单内容。

---

用例 3：骑手接单

- 用例名称：骑手接单

- 用例编号：UC-003

- 参与者：骑手、系统

- 前置条件：

- 骑手已登录系统并处于“空闲”状态。

- 系统中有待分配订单。

- 后置条件：

- 骑手成功接单，订单状态更新为“配送中”。

- 主事件流：

1. 系统向附近骑手推送新订单信息。

2. 骑手查看订单详情并点击“接单”。

3. 系统确认骑手接单，并更新订单状态。

4. 骑手开始前往餐厅取餐。

- 异常事件流：

- A1：骑手未及时响应接单请求。

- 系统将订单重新分配给其他骑手。

- A2：骑手无法完成配送（如车辆故障）。

- 骑手在系统中申请取消接单。

- 系统重新分配订单并通知用户。

---

用例 4：用户评价订单

- 用例名称：用户评价订单

- 用例编号：UC-004

- 参与者：普通用户、系统

- 前置条件：

- 用户已完成订单且订单状态为“已送达”。

- 后置条件：

- 用户评价提交成功，餐厅评分更新。

- 主事件流：

1. 用户进入订单详情页面，点击“评价订单”。

2. 用户填写评分（如 1-5 星）并添加文字评论。

3. 用户提交评价。

4. 系统保存评价并更新餐厅综合评分。

- 异常事件流：

- A1：用户未填写评分或评论直接提交。

- 系统提示“请填写评价内容后再提交”。

- 用户补充内容后重新提交。

- A2：系统未能保存评价（如网络问题）。

- 系统提示“评价提交失败，请稍后重试”。

- 用户重新提交评价。

---

用例 5：管理员审核餐厅入驻

- 用例名称：管理员审核餐厅入驻

- 用例编号：UC-005

- 参与者：管理员、系统

- 前置条件：

- 餐厅已提交入驻申请并上传相关资质文件。

- 后置条件：

- 审核结果记录在系统中，餐厅状态更新为“已通过”或“未通过”。

- 主事件流：

1. 管理员登录后台管理系统。

2. 管理员查看餐厅入驻申请及资质文件。

3. 管理员审核资质文件并决定是否通过。

4. 系统记录审核结果并通知餐厅。

- 异常事件流：

- A1：资质文件不完整或不符合要求。

- 管理员驳回申请并注明原因。

- 系统通知餐厅补充资料。

- A2：系统未能保存审核结果（如数据库故障）。

- 系统提示“操作失败，请稍后重试”。

- 管理员重新提交审核结果。