项目文档

# Introduction

第一章 引言  
  
 背景  
  
在当今日新月异的科技和社会背景下，餐饮行业尤其外卖市场的创新需求日益提高。随着人们生活节奏的加快，点餐方式已经从传统的到店消费逐渐转变为通过应用进行在线点餐。为追求速食生活方式的年轻消费群体提供便利的同时，如何有效地整合周边的餐饮资源，并为用户提供一站式的点餐、支付、配送服务成为关键环节。针对此，QuickEats（快速享用）软件应用应运而生。  
  
QuickEats是一款专为手机和平板等移动设备设计的餐饮外卖应用，主打速度与便捷，其核心特点不言而喻。除了致力于提供快速便捷的点餐体验外，还结合现代用户的时尚追求，融合实时信息推送和社交分享元素，为用户创造个性化用餐体验的同时，也为餐厅带来高效运作空间。  
  
 业务目标  
  
本文档旨在明确QuickEats软件应用的所有功能需求、非功能需求，以及相关运营及开发中的约束条件。着重定位QuickEats如何立足市场，提高自身品牌核心竞争力并满足快速增长的外卖订单需求。文档的具体目标可总结为：  
  
1. 为开发、设计团队清晰勾勒出软件的功能架构蓝图，确保产品经理、技术开发者能够基于共同的理解进行合作，依照此文档设计并实现系统的各项功能模块。  
  
2. 提升用户体验: 深入挖掘用户需求，构建友好的用户界面，适应目标用户习惯的同时提供个性化的信息推送，以及全景的餐品浏览体验。  
  
3. 数据驱动运营: 通过大数据分析，为餐厅提供精准用户口味偏好、购物车添加率等数据，以便进行菜品调整及风险预警，尽量减少餐厅与用户的摩擦点。  
  
4. 动态适应市场变化: 动态调整软件功能以响应餐饮外卖市场最新变化，持续集成迭代，确保满足快速变化的市场需求以及用户偏好的更迭。  
  
 范围  
  
本文档范围覆盖QuickEats应用所有功能需求，包括但不限于用户管理、订单创建与管理、购物车管理、支付管理等重点领域的功能规范。还涵盖了非功能需求，例如用户体验设计、性能、安全性、互操作性和扩展性等。各项内容明确地帮助团队成员了解整个项目的背景、目标及其相关内容边界，简化沟通壁垒，优化项目实施流程，以确保项目任务按期完成，顺利上线运行。  
  
 定义、缩写与缩略语  
  
在本文档中，为了确保明晰性，特意对常见术语、概念进行了界定与说明，期望能够对读者理解文档内容进行一定指导与帮助。  
  
- 用户(User): 使用QuickEats平台进行日常操作，如创建账户、浏览并订购餐品等的用户群体。  
- 餐厅(Restaurant): 通过QuickEats平台为用户供应餐品的实体商家。  
- 菜品(MenuItem): 餐厅通过平台向用户展示并售卖的具体餐品项目，包括名字、价格、类型等信息。  
- 订单(Order): 用户通过平台下单购买餐品的一次性消费行为，从下订单、付款、配送到确认收货均为其组成部分。  
- 购物车(ShoppingCart): 用户在浏览餐品时用于临时存储多个餐品项目的容器，形成最终订单前的过渡状态。  
- 支付（Payment）: 用户在订购餐品后，选定支付方法并完成交易的过程，实现资金转移和订单确认。  
- 评论（Comment）: 用户在利用平台享用餐品后对其评价，包括文字描述和评分两方面内容。  
- 管理员（Administrator）: 平台的后台管理者，负责监管用户信息、商品信息、订单状态等，涉及平台整体运营相关工作。  
- 订单详情（OrderDetail）: 记录每个订单包含的具体餐品及其数量信息，由成团的ListItem组成，且链接至订单。  
- 用户管理记录（UserManagementRecord）: 记录了管理员对于用户操作的所有记录，包含创建、修改和删除等，方便追踪操作历史。  
- 餐厅管理（RestaurantManagement）: 管理餐厅及其商品的服务范畴，记录了所有对餐厅及其交易相关操作，从而维护餐厅状态。  
- 菜品管理（MenuItemManagement）: 记录所有商品，包括创建新增、删除、修改创建日期及相关操作历史，保证菜品信息的真实性和准确性。  
  
 参考标准  
  
- ISO/IEC 9126: 软件产品的度量标准，主要用于量化软件的属性如功能性、可靠性、易用性等。  
- ISO/IEC 25010, 25011, 25012: 家庭标准，主要针对软件质量模型，包括量化软件质量和性能测试的方法。  
- GB/T 16260-2006 信息技术 质量标准: 中国国家标准，定义了测试软件质量的准则，是制定国家标准的一个重要框架。  
- 《快速享用 (QuickEats) 产品设计文档》: 遵循快速享用产品团队的现有文档进行需求说明，涵盖了初始设计的理念和目标。  
- 《移动应用开发的建议标准和最佳实践》: 提供针对移动应用开发的实际建议及操作方法，确保软件开发的质量。  
  
除以上内容，对于项目实施期间必要的标准、规则将同步列入本文档参考范围。目的在于明确了解最新的市场需求趋势及技术方案，保障最终软件产品的质量与实际功能符合要求并满足各类用户群体的期待。

# Overall Description

1. 总体概述  
     
    1. 产品视角  
     
   QuickEats是一款旨在提供便捷且高效的餐饮订购与管理系统的软件应用。愿景是通过无缝的用户界面和强大的后台支持，为用户提供便捷的在线餐饮服务，同时为餐厅提供有效的管理工具。其目标用户群体是城市繁忙的白领和学生，他们倾向于订购外送和外卖服务，希望通过简单的操作，以合理的价格获得美味可口的餐饮服务。预期市场定位为面向各类型餐馆和外界用户，提供一种高效、安全、便捷的网上订餐解决方案。本软件将支持全机型移动设备，覆盖iOS与Android系统的机型，便于各类用户使用，同时支持PC端进行餐厅及菜品管理。  
     
    2. 产品功能  
     
    2.1 用户相关功能  
   - 账户管理：用户可以创建、登录、修改个人资料以及密码管理等，注册时验证用户名的唯一性，登录时验证账户的有效性。  
   - 订单管理：创建、查看、修改订单，使用户能在系统中全程追踪订单状态。确保每个步骤都是为用户提供个性化定制的简洁体验。  
   - 购物车管理：管理购物车中的菜品，支持添加多个菜品并查看合计价格；使用时需要进行库存与支付状态的验证与对应处理。  
     
    2.2 餐厅相关功能  
   - 餐厅管理：添加新的餐厅信息、编辑和删除现有的餐厅，允许用户查询各类食品信息，支持查询对应信息的功能。  
     
    2.3 菜品相关功能  
   - 菜品管理：添加、编辑和删除菜品信息，查看餐厅的菜单细节，库存管理和菜品信息更新同步，根据餐厅侧需求对菜品价格等必要元素进行适时调整。  
     
    2.4 支付相关功能  
   - 支付流程管理：发起支付操作，预览支付详情并确认支付；操作中支持多种支付方式，如在线即时支付和后续支付安排。  
     
    2.5 管理员相关功能  
   - 管理员管理：管理多个管理员账户，对不同的管理员设置不同权限等级，管理用户与餐厅的功能操作权限。  
   - 评论管理：监督和管理用户的评论及评分，必要时对其进行删除或编辑，促进高质量、真实的用户体验共享。  
   - 订单与支付记录管理：定期审查和导出交易记录，提供行政机构所需的数据，保障商家财务收入透明度。  
     
    3. 用户特征  
     
   本软件综合定位群体广泛，主要聚焦于工作繁忙的白领阶级和高校在校生，特别是哪些对时间安排高效率、对操作界面友好度有较高要求的用户。这些用户偏好生活便利性、界面简洁美观、功能操作简单以及响应速度快的产品，他们期望在购买任意餐类过程中获得流畅的使用体验，同时倾向于频繁使用基于移动网络的应用。  
     
    4. 竞品分析  
     
   在餐饮订餐服务领域已有很多具有影响力的竞品，例如美团、饿了么等。这些竞品一方面通过长期的市场深耕积累了大量的用户群体和使用数据，另一方面其服务质量的完善性也为类似软件设立了行业标准。然而，这些竞品也存在一些满足不了的“痛点”：如用户信息归档复杂、餐厅资料审核严格但最终呈现信息更新速度慢等问题。QuickEats则致力于通过强大的技术支持和优化的操作流程，为用户提供更为人性化和便利的服务，以为其构建差异化优势。  
     
    5. 业务风险与机遇  
     
   业务风险包括可能出现的法律问题（例如用户数据保护）、竞争者优势领先（依托已有用户基础争取新用户）和市场过饱和导致用户端接受度低；同样存在的发展机遇决策支持层面的决定性数据记录将促进QuickEats快速地以及成功地走向市场前台，为行政机构提供可靠的统计数据支持，从而带动餐饮行业的整体发展。该系统帮助商家提升业绩，增加订单变现率，帮助国家财务相关部门的提升税收，为国内第三产业发展提升生产力、为整个餐饮形式添加新动能。  
     
    6. 设计约束  
     
   - 技术栈：支持跨平台操作，前端采用React Native等框架；后端采用Spring Boot等Java后端框架，数据持久层应考虑使用MySQL等高效稳定的数据库。  
   - 硬件平台：应用应具有良好的兼容性，支持常见的智能手机（包括iOS和Android）。  
   - 法规要求：应用程序应遵守全球的主要数据保护法规，确保用户的隐私在操作过程中得到保护，产生异议时，需预留专门渠道和实时固定优先级的解决机构。  
     
    7. 假设与依赖关系  
     
   - 假设：用户具有一定程度的互联网使用常识，并且对于移动设备有基础的操作技术；用户对于支付服务的接受度以及愿意于新系统尝试新的购餐方式。  
   - 依赖关系：系统依赖于互联网连接以保证顺畅的在线服务体验，并依赖第三方支付平台以完成交易。  
      
   总的来说, QuickEats旨在通过合理的设计和强大的后台支持，提供给用户高质量的服务体验，赢得市场与用户的选择。

# Functional Requirement

以下是基于上述用例描述生成功能需求的汇总，每个功能需求包括输入、输出和功能描述：

---

功能需求 1：用户注册

- 输入：

- 用户名、密码、联系方式等注册信息。

- 输出：

- 注册成功提示或错误提示（如用户名重复、数据非法）。

- 功能描述：

系统提供用户注册功能，用户需填写注册表单并提交。系统验证输入数据的合法性后保存用户信息，并返回注册结果。

---

功能需求 2：删除用户

- 输入：

- 管理员选择的目标用户账户。

- 输出：

- 删除成功提示或错误提示（如删除失败）。

- 功能描述：

管理员可通过用户管理界面删除目标用户账户。系统在管理员确认后删除用户信息，并返回操作结果。

---

功能需求 3：添加餐厅

- 输入：

- 餐厅名称、地址、营业时间等信息。

- 输出：

- 添加成功提示或错误提示（如必填项为空）。

- 功能描述：

管理员可通过餐厅管理界面添加新餐厅。系统验证表单数据合法性后保存餐厅信息，并返回操作结果。

---

功能需求 4：删除餐厅

- 输入：

- 管理员选择的目标餐厅。

- 输出：

- 删除成功提示或错误提示（如餐厅有未完成订单）。

- 功能描述：

管理员可通过餐厅管理界面删除目标餐厅。系统在确认无未完成订单后删除餐厅信息，并返回操作结果。

---

功能需求 5：管理餐厅

- 输入：

- 修改后的餐厅信息（如地址、营业时间）。

- 输出：

- 修改成功提示或错误提示（如地址格式错误）。

- 功能描述：

管理员可通过餐厅管理界面编辑餐厅信息。系统验证修改内容的合法性后保存更新，并返回操作结果。

---

功能需求 6：添加菜品

- 输入：

- 菜品名称、价格、图片等信息。

- 输出：

- 添加成功提示或错误提示（如价格为负数）。

- 功能描述：

餐厅管理员可通过菜品管理界面添加新菜品。系统验证表单数据合法性后保存菜品信息，并返回操作结果。

---

功能需求 7：删除菜品

- 输入：

- 餐厅管理员选择的目标菜品。

- 输出：

- 删除成功提示或错误提示（如菜品有未完成订单）。

- 功能描述：

餐厅管理员可通过菜品管理界面删除目标菜品。系统在确认无未完成订单后删除菜品信息，并返回操作结果。

---

功能需求 8：修改菜品信息

- 输入：

- 修改后的菜品信息（如价格、描述）。

- 输出：

- 修改成功提示或错误提示（如价格为负数）。

- 功能描述：

餐厅管理员可通过菜品管理界面编辑菜品信息。系统验证修改内容的合法性后保存更新，并返回操作结果。

---

功能需求 9：查看菜品

- 输入：

- 用户选择的目标菜品。

- 输出：

- 菜品详情（如图片、价格、描述）或错误提示（如菜品未找到）。

- 功能描述：

用户可通过餐厅页面或搜索功能查看目标菜品的详细信息。系统加载并显示菜品详情。

---

功能需求 10：创建订单

- 输入：

- 用户购物车中的菜品及总价信息。

- 输出：

- 订单创建成功提示或错误提示（如购物车为空）。

- 功能描述：

用户可通过购物车页面创建订单。系统生成订单并返回操作结果。

---

功能需求 11：删除订单

- 输入：

- 用户选择的目标订单。

- 输出：

- 删除成功提示或错误提示（如订单状态不为“待支付”）。

- 功能描述：

用户可通过订单管理界面删除目标订单。系统在确认订单状态后删除订单，并返回操作结果。

---

功能需求 12：查看订单

- 输入：

- 用户选择的目标订单。

- 输出：

- 订单详情（如菜品列表、总价、状态）或错误提示（如订单未找到）。

- 功能描述：

用户可通过订单管理界面查看目标订单的详细信息。系统加载并显示订单详情。

---

功能需求 13：修改订单信息

- 输入：

- 修改后的配送地址或备注信息。

- 输出：

- 修改成功提示或错误提示（如订单状态不为“待支付”）。

- 功能描述：

用户可通过订单管理界面修改目标订单信息。系统验证订单状态后保存更新，并返回操作结果。

---

功能需求 14：添加菜品到购物车

- 输入：

- 用户选择的目标菜品。

- 输出：

- 添加成功提示或错误提示（如库存不足）。

- 功能描述：

用户可通过菜品详情页面将目标菜品添加到购物车。系统更新购物车并返回操作结果。

---

功能需求 15：从购物车移除菜品

- 输入：

- 用户选择的目标菜品。

- 输出：

- 移除成功提示或错误提示（如菜品未找到）。

- 功能描述：

用户可通过购物车页面移除目标菜品。系统更新购物车并返回操作结果。

---

功能需求 16：管理购物车

- 输入：

- 用户对购物车的操作（如增减菜品、清除购物车）。

- 输出：

- 更新成功提示或错误提示（如购物车为空）。

- 功能描述：

用户可通过购物车页面管理购物车内容。系统实时更新购物车并返回操作结果。

---

功能需求 17：发起支付

- 输入：

- 用户选择的目标订单及支付方式。

- 输出：

- 支付成功提示或错误提示（如余额不足、网络中断）。

- 功能描述：

用户可通过订单详情页面发起支付。系统调用支付接口并更新订单状态，返回操作结果。

---

功能需求 18：取消支付

- 输入：

- 用户选择的目标订单。

- 输出：

- 取消成功提示或错误提示（如支付已完成）。

- 功能描述：

用户可在支付页面取消支付。系统终止支付流程并恢复订单状态，返回操作结果。

---

功能需求 19：发布评论

- 输入：

- 用户填写的评分和文字内容。

- 输出：

- 发布成功提示或错误提示（如内容为空）。

- 功能描述：

用户可通过订单或餐厅页面发布评论。系统验证评论内容后保存信息，返回操作结果。

---

功能需求 20：管理评论

- 输入：

- 管理员选择的目标评论及操作（编辑或删除）。

- 输出：

- 操作成功提示或错误提示（如包含敏感词）。

- 功能描述：

管理员可通过评论管理页面编辑或删除评论。系统执行操作并返回结果。

---

功能需求 21：管理支付记录

- 输入：

- 管理员选择的目标支付记录及操作（查看、修改、导出）。

- 输出：

- 操作成功提示或错误提示（如记录未找到）。

- 功能描述：

管理员可通过支付记录管理页面查询、修改或导出支付记录，返回操作结果。

---

功能需求 22：创建管理员

- 输入：

- 管理员用户名、密码、权限等级等信息。

- 输出：

- 创建成功提示或错误提示（如用户名重复）。

- 功能描述：

超级管理员可通过管理员管理界面创建新管理员。系统验证数据合法性后保存信息，返回操作结果。

---

功能需求 23：删除管理员

- 输入：

- 超级管理员选择的目标管理员。

- 输出：

- 删除成功提示或错误提示（如删除失败）。

- 功能描述：

超级管理员可通过管理员管理界面删除目标管理员。系统执行删除操作并返回结果。

---

# External Description

第四章 外部接口  
  
 4.1 中央管理系统接口  
  
- 接口概述：中央管理系统（CMS）负责统一管理和维护所有的餐厅、菜品及订单信息，为用户提供最新的数据支持。  
  
- 接口类型：RESTful API  
  
- 通信协议：HTTP/HTTPS  
  
- 数据格式：  
 - 请求消息：JSON  
 - 响应消息：JSON  
  
 - 接口的URL样例：  
 - [POST] /api/restaurants/add (添加餐厅)  
 - [PUT] /api/restaurants/edit (编辑餐厅)  
 - [DELETE] /api/restaurants/remove (删除餐厅)  
 - [POST] /api/menu-items/add (添加菜品)  
 - [PUT] /api/menu-items/edit (编辑菜品)  
 - [DELETE] /api/menu-items/remove (删除菜品)  
 - [POST] /api/orders/create (创建订单)  
 - [PUT] /api/orders/edit (编辑订单)  
 - [DELETE] /api/orders/remove (删除订单)  
  
 4.2 第三方支付系统接口  
  
- 接口概述：支持与国内主流在线支付系统的集成，确保用户的订餐支付流畅进行。  
  
- 接口类型：支付网关接口  
  
- 通信协议：HTTPS  
  
- 数据格式：  
 - 请求消息格式：JSON、Form Data  
 - 响应消息格式：JSON  
   
 - 接口的URL示例：  
 - [POST] /api/payment/initiate (发起支付)  
 - [POST] /api/payment/complete (支付完成回调)  
 - [POST] /api/payment/cancel (取消支付)  
 - [PUT] /api/payment/update (更新支付状态)  
  
 4.3 第三方用户认证与登录系统接口  
  
- 接口概述：集成第三方登录服务，方便用户使用常见社交账号或邮箱密码直接登录。  
  
- 接口类型：OAuth 2.0 接口  
  
- 通信协议：HTTPS  
  
- 数据格式：  
 - 请求消息：Form Data  
 - 响应消息：JSON  
   
 - 接口的URL示例：  
 - [GET] /third-party/oauth/authorize[]  
 - [POST] /third-party/oauth/token  
 - [GET] /third-party/oauth/user-info  
  
 4.4 数据统计与分析系统接口  
  
- 接口概述：确保能够接入大数据分析工具，利用用户行为数据优化产品的动态资源配置和用户体验设计。  
  
- 接口类型：RESTful API  
  
- 通信协议：HTTPS  
  
- 数据格式：  
 - 请求消息：JSON  
 - 响应消息：JSON  
   
 - 接口的URL示例：  
 - [POST] /api/analytics/submit (提交分析数据)  
 - [GET] /api/analytics/report (获取分析报告)  
  
 4.5 用户消息通知系统接口  
  
- 接口概述：向用户发送包括订单状态更新、促销信息等在内的各类通知消息。  
  
- 接口类型：HTTP(S)、WebSocket  
  
- 通信协议：HTTP/HTTPS  
  
- 数据格式：  
 - 请求消息：JSON  
 - 响应消息：JSON  
  
 - 接口的URL示例：  
 - [POST] /api/notifications/send (发送消息通知)  
  
 4.6 系统日志及监控系统接口  
  
- 接口概述：对接系统日志服务，确保系统的稳定运行并能够快速响应问题。  
  
- 接口类型：API log日志接口  
  
- 通信协议：HTTPS  
  
- 数据格式：  
 - 请求消息：JSON  
 - 响应消息：JSON  
  
 - 接口的URL示例：  
 - [POST] /api/log/error  
 - [POST] /api/log/info  
  
以上接口需求将在软件实现过程中，需要严格遵守定义的技术规范和通信协议，保证外部系统的数据能顺利、准确地进行交互，同时确保系统的安全性与稳定性。

# Non-Functional Requirement

第五章 非功能需求  
  
非功能需求主要涉及软件的一些特性要求，这些需求通常与软件的功能性无关，但对提高软件的用户体验和可用性具有重大影响。QuickEats的应用将遵循以下非功能需求：  
  
 5.1 性能要求  
  
 5.1.1 响应时间  
- 定义：响应时间是指系统从用户请求发起到接收响应完成的时间。响应时间包括了数据处理、网络传输及数据返回各个过程。  
- 目标：QuickEats的目标是保持用户体验流畅，具体细化为：  
 - 页面请求的响应时间不大于2秒。对于页面内容较多，如历史订单页面等可能稍长些。  
 - 数据提交后，系统应于1秒内完成处理并反馈结果，这对用户体验是必要因素。  
- 衡量标准：使用Web performance测试工具，如Chrome DevTools或者Selenium等进行测试，在理想的测试环境下，确保95%的请求响应时间不超过2秒，99%的响应时间不超过3秒。  
  
 5.1.2 吞吐量  
- 定义：吞吐量表示单位时间内系统处理的事务数量，反映了系统的处理能力。  
- 目标：QuickEats需要保证能同时处理大量并发请求，减少服务器压力以及数据等待时间。  
- 衡量标准：使用像LoadRunner或Apache JMeter这样的负载测试工具来模拟特定用户行为模式。目标设定是每秒钟处理至少1000个交易/请求，稳定运行不小于24小时，其间吞吐量波动不超过10%。  
  
 5.1.3 并发用户数  
- 定义：并发用户是指同一时间内访问并操作系统的用户数量。  
- 目标：系统应能够稳定处理大量并发用户（例如，高峰时段需要不少于50000个并发用户），并且用户操作体验不受影响。  
- 衡量标准：对于高峰时段的并发测试，采用实时负载生成器，在保持每秒处理超过1000个请求的前提下测试最大可以承载的并发接入人数。  
  
 5.1.4 高峰使用情况  
- 定义：高峰时段是指用户访问平台的密度较大，流量急剧上升的时间段。  
- 目标：平台在高峰时段应能保持稳定的服务提供，避免由于高并发造成的服务器负载过重、处理能力下降等问题。  
- 衡量标准：高并发测试可以在高峰时段模拟，目标是保证峰值时90%以上的请求可以在2秒内得到响应，99%的请求不超过5秒响应。  
  
 5.2 软件质量属性  
  
 5.2.1 可靠性  
- 定义：可靠性是软件不受软件错误影响的能力，软件在规定条件下和时间内实现其预期功能的性能。  
- 目标：QuickEats的软件设计需保证服务的高可靠性，减少系统失败导致的不可预期损失，确保724小时可用。  
- 衡量标准：在线故障恢复时间不超过30秒，平均故障间隔时间（MTBF）至少为每年10万小时，错误发生后应不超过60秒内自动恢复，系统可靠性大于99.99%。  
  
 5.2.2 可用性  
- 定义：可用性是指系统及组件根据其不失败的操作指南用于指定的功能的性能，这不仅包括性能问题，还包括整体系统的无障碍使用情况。  
- 目标：对于用户和管理员而言，系统应始终保持可用，避免任何形式的停机时间。  
- 衡量标准：应用可用性需达到99.99%，应用具备多区域高可用，具有故障自动转移机制，在网络中断和服务器宕机等异常情况时仍然可以确保基础服务不中断提供。  
  
 5.2.3 可维护性  
- 定义：可维护性意味着软件可以在未来持续开发、运行和维护，而不会给团队带来过多负担。  
- 目标：为了快速解决问题和持续改进，软件应易于更新和维护更新越快，客户满意度越高。  
- 衡量标准：新开发的功能模块其维护成本低于总体开发成本的30%，关键新特性可以从设计到测试完成全部流程需少于60天，且包含完整的测试用例。  
   
 5.2.4 可移植性  
- 定义：可移植性是对不同操作系统和硬件环境兼容的度量，小化对公司现有基础设施的影响。  
- 目标：确保QuickEats能够在最新的多平台和操作系统环境中无差错运行，支持包括iOS、Android等主流操作系统及平台版本。  
- 衡量标准：支持包括但不限于最新的主流操作系统版本，任何一种平台的软件版本更新在一个月内完成且其测试用例全部通过，就算作是完全准确的版本。  
  
 5.2.5 性能属性  
- 定义：性能属性一般指学习性、熟悉革新、接口设计等非功能性特性。  
- 目标：提升用户体验与技巧快速掌握应用对性能有一定的要求。  
- 衡量标准：用户80%能够在20分钟内快速学习使用QuickEats，新的功能递进式引入，测试显示新功能理解和使用时间会逐步减少新功能用户大幅度学会。  
  
 5.3 数据保护和隐私  
- 定义：平台需合规要求，遵循国家和全球数据保护法，确保数据处理手段合法，注重用户非个人信息相关内容安全。  
- 目标：从保护用户隐私角度出发，加强系统安全，防止数据泄露。  
- 衡量标准：遵守所有联邦数据保护规定——包括不限于GDPR，确保最新数据进行加密保存与传输，并定期对其进行安全检查、审计和认证。在发生数据泄露事故后，应在几小时内对外公告，并采取有效的补救措施消除威胁。  
  
 5.4 系统安全性  
- 定义：安全性是系统抵御外部攻击以及内部破坏的能力，在实现服务或进行升级时，应考虑到安全性因素。  
- 目标：保障用户账户、交易信息不被非法访问，对用户和餐厅的数据采取严密的安全措施。  
- 衡量标准：防火墙应有效阻止具有极高威胁的攻击（如SQL注入、DDoS攻击），被置于合法用户仅有交易或管理权限。至少通过ISO27001标准。  
  
 5.5 标准和法规  
- 定义：标准和规则遵守是由国家或国际组织制定的所有相关依法的良好规范条例。  
- 目标：确保开发和运维过程符合标准以应对更新改版新问题，同时赢得合法声誉。  
- 衡量标准：项目应根据ISO/IEC 9126、ISO/IEC 25010、25011、25012评估软件品质，遵守地区法律文件，如GA776-2014实行移动APP安全技术规范，更新跨境数据传输管理制度，使之保持合规。  
  
 5.6 用户界面  
- 定义：用户界面指的是人机交互中呈现的所有内容，包括图形界面、文本界面，等等。  
- 目标：塑造出良好的体验界面，打造舒适的使用环境。  
- 衡量标准：评估界面可用性的标准是易用性要达到项目客户接受的行业平均水平，由用户满意度和用户反馈测评，满足业界UI/UX主流模型，90%以上用户反馈为良好使用体验，最大程度优化受众人群达到预期的视觉效果。  
  
 总结  
以上这些非功能需求都是为了确保QuickEats能够提供最好的服务体验，这些具体的要求不仅增强了用户的满意度，还能加强整体的应用稳定性与安全性，提高了系统的维护性和易于升级、优化性。符合所有标准，包括国际和国内的数据保护标准，有助于确保QuickEats符合法规要求，提供用户在使用时所需的安全保证。这为保证软件的成功运作设定了一个清晰的边界。

# Use Case

用例描述1：用户注册  
- 用例名称：用户注册   
- 用例编号：UC-001   
- 参与者：用户、系统   
- 前置条件：用户未注册账户，且已打开QuickEats应用。   
- 后置条件：用户账户成功创建，系统保存用户信息。   
- 主事件流：   
 1. 用户点击“注册”按钮。   
 2. 系统显示注册表单，要求输入用户名、密码、联系方式等信息。   
 3. 用户填写表单并提交。   
 4. 系统验证输入数据的合法性（如检查用户名是否唯一）。   
 5. 系统保存用户信息，并提示注册成功。   
- 异常事件流：   
 - A1: 用户名已存在，系统提示错误并要求重新输入用户名。   
 - A2: 输入数据非法（如密码格式不正确），系统提示错误并要求重新输入。   
  
---  
  
 用例描述2：删除用户   
- 用例名称：删除用户   
- 用例编号：UC-002   
- 参与者：管理员、系统   
- 前置条件：管理员已登录系统，目标用户账户存在。   
- 后置条件：目标用户账户被删除，相关数据从系统中移除。   
- 主事件流：   
 1. 管理员进入用户管理界面。   
 2. 管理员选择目标用户并点击“删除”按钮。   
 3. 系统弹出确认对话框。   
 4. 管理员确认删除操作。   
 5. 系统删除用户信息，并提示删除成功。   
- 异常事件流：   
 - A1: 删除失败（如数据库连接问题），系统提示错误并终止操作。   
  
---  
  
 用例描述3：添加餐厅   
- 用例名称：添加餐厅   
- 用例编号：UC-003   
- 参与者：管理员、系统   
- 前置条件：管理员已登录系统，且具备添加餐厅权限。   
- 后置条件：新餐厅信息被添加到系统中，可供用户浏览。   
- 主事件流：   
 1. 管理员进入餐厅管理界面。   
 2. 管理员点击“添加餐厅”按钮。   
 3. 系统显示餐厅信息表单（包括名称、地址、营业时间等）。   
 4. 管理员填写表单并提交。   
 5. 系统验证表单数据的合法性。   
 6. 系统保存餐厅信息，并提示添加成功。   
- 异常事件流：   
 - A1: 表单数据非法（如必填项为空），系统提示错误并要求重新填写。   
  
---  
  
 用例描述4：删除餐厅   
- 用例名称：删除餐厅   
- 用例编号：UC-004   
- 参与者：管理员、系统   
- 前置条件：管理员已登录系统，目标餐厅存在且无未完成订单。   
- 后置条件：目标餐厅信息从系统中移除，用户无法再浏览该餐厅。   
- 主事件流：   
 1. 管理员进入餐厅管理界面。   
 2. 管理员选择目标餐厅并点击“删除”按钮。   
 3. 系统弹出确认对话框。   
 4. 管理员确认删除操作。   
 5. 系统删除餐厅信息，并提示删除成功。   
- 异常事件流：   
 - A1: 餐厅有未完成订单，系统提示无法删除并终止操作。   
  
---  
  
 用例描述5：管理餐厅   
- 用例名称：管理餐厅   
- 用例编号：UC-005   
- 参与者：管理员、系统   
- 前置条件：管理员已登录系统，目标餐厅存在。   
- 后置条件：餐厅信息更新或状态调整完成。   
- 主事件流：   
 1. 管理员进入餐厅管理界面。   
 2. 管理员选择目标餐厅并点击“编辑”按钮。   
 3. 系统显示餐厅信息表单。   
 4. 管理员修改表单内容并提交。   
 5. 系统保存更新后的餐厅信息，并提示修改成功。   
- 异常事件流：   
 - A1: 修改内容非法（如地址格式错误），系统提示错误并要求重新输入。   
  
---  
  
 用例描述6：添加菜品   
- 用例名称：添加菜品   
- 用例编号：UC-006   
- 参与者：餐厅管理员、系统   
- 前置条件：餐厅管理员已登录系统，目标餐厅存在。   
- 后置条件：新菜品信息被添加到系统中，可供用户浏览。   
- 主事件流：   
 1. 餐厅管理员进入菜品管理界面。   
 2. 餐厅管理员点击“添加菜品”按钮。   
 3. 系统显示菜品信息表单（包括名称、价格、图片等）。   
 4. 餐厅管理员填写表单并提交。   
 5. 系统验证表单数据的合法性。   
 6. 系统保存菜品信息，并提示添加成功。   
- 异常事件流：   
 - A1: 表单数据非法（如价格为负数），系统提示错误并要求重新填写。   
  
---  
  
 用例描述7：删除菜品   
- 用例名称：删除菜品   
- 用例编号：UC-007   
- 参与者：餐厅管理员、系统   
- 前置条件：餐厅管理员已登录系统，目标菜品存在且无未完成订单。   
- 后置条件：目标菜品信息从系统中移除，用户无法再浏览该菜品。   
- 主事件流：   
 1. 餐厅管理员进入菜品管理界面。   
 2. 餐厅管理员选择目标菜品并点击“删除”按钮。   
 3. 系统弹出确认对话框。   
 4. 餐厅管理员确认删除操作。   
 5. 系统删除菜品信息，并提示删除成功。   
- 异常事件流：   
 - A1: 菜品有未完成订单，系统提示无法删除并终止操作。   
  
---  
  
 用例描述8：修改菜品信息   
- 用例名称：修改菜品信息   
- 用例编号：UC-008   
- 参与者：餐厅管理员、系统   
- 前置条件：餐厅管理员已登录系统，目标菜品存在。   
- 后置条件：菜品信息更新完成。   
- 主事件流：   
 1. 餐厅管理员进入菜品管理界面。   
 2. 餐厅管理员选择目标菜品并点击“编辑”按钮。   
 3. 系统显示菜品信息表单。   
 4. 餐厅管理员修改表单内容并提交。   
 5. 系统保存更新后的菜品信息，并提示修改成功。   
- 异常事件流：   
 - A1: 修改内容非法（如价格为负数），系统提示错误并要求重新输入。   
  
---  
  
 用例描述9：查看菜品   
- 用例名称：查看菜品   
- 用例编号：UC-009   
- 参与者：用户、系统   
- 前置条件：用户已登录或未登录系统，目标餐厅和菜品存在。   
- 后置条件：用户成功浏览到目标菜品的详细信息。   
- 主事件流：   
 1. 用户进入餐厅页面或搜索菜品。   
 2. 用户点击目标菜品。   
 3. 系统加载并显示菜品详情（包括图片、价格、描述等）。   
 4. 用户浏览菜品信息。   
- 异常事件流：   
 - A1: 目标菜品不存在，系统提示“菜品未找到”。   
  
---  
  
 用例描述10：创建订单   
- 用例名称：创建订单   
- 用例编号：UC-010   
- 参与者：用户、系统   
- 前置条件：用户已登录系统，购物车中有至少一件菜品。   
- 后置条件：新订单生成，状态为“待支付”。   
- 主事件流：   
 1. 用户进入购物车页面。   
 2. 用户点击“结算”按钮。   
 3. 系统加载订单确认页面，显示购物车中的菜品及总价。   
 4. 用户确认订单信息并提交。   
 5. 系统生成订单，并提示订单创建成功。   
- 异常事件流：   
 - A1: 购物车为空，系统提示“请先添加菜品到购物车”。   
 - A2: 网络连接失败，系统提示“操作失败，请稍后重试”。   
  
---  
  
 用例描述11：删除订单   
- 用例名称：删除订单   
- 用例编号：UC-011   
- 参与者：用户、系统   
- 前置条件：用户已登录系统，目标订单处于“待支付”状态。   
- 后置条件：目标订单从系统中移除。   
- 主事件流：   
 1. 用户进入订单管理页面。   
 2. 用户选择目标订单并点击“删除”按钮。   
 3. 系统弹出确认对话框。   
 4. 用户确认删除操作。   
 5. 系统删除订单，并提示删除成功。   
- 异常事件流：   
 - A1: 订单状态不为“待支付”，系统提示“无法删除已完成或已支付订单”。   
  
---  
  
 用例描述12：查看订单   
- 用例名称：查看订单   
- 用例编号：UC-012   
- 参与者：用户、系统   
- 前置条件：用户已登录系统，目标订单存在。   
- 后置条件：用户成功浏览到目标订单的详细信息。   
- 主事件流：   
 1. 用户进入订单管理页面。   
 2. 用户点击目标订单。   
 3. 系统加载并显示订单详情（包括菜品列表、总价、状态等）。   
 4. 用户浏览订单信息。   
- 异常事件流：   
 - A1: 目标订单不存在，系统提示“订单未找到”。   
  
---  
  
 用例描述13：修改订单信息   
- 用例名称：修改订单信息   
- 用例编号：UC-013   
- 参与者：用户、系统   
- 前置条件：用户已登录系统，目标订单处于“待支付”状态。   
- 后置条件：订单信息更新完成。   
- 主事件流：   
 1. 用户进入订单管理页面。   
 2. 用户选择目标订单并点击“编辑”按钮。   
 3. 系统加载订单编辑页面，允许修改配送地址或备注信息。   
 4. 用户修改信息并提交。   
 5. 系统保存更新后的订单信息，并提示修改成功。   
- 异常事件流：   
 - A1: 订单状态不为“待支付”，系统提示“无法修改已完成或已支付订单”。   
  
---  
  
 用例描述14：添加菜品到购物车   
- 用例名称：添加菜品到购物车   
- 用例编号：UC-014   
- 参与者：用户、系统   
- 前置条件：用户已登录或未登录系统，目标菜品存在且可购买。   
- 后置条件：目标菜品被添加到用户的购物车中。   
- 主事件流：   
 1. 用户进入菜品详情页面。   
 2. 用户点击“加入购物车”按钮。   
 3. 系统将菜品添加到购物车，并更新购物车数量。   
 4. 系统提示“已成功添加到购物车”。   
- 异常事件流：   
 - A1: 菜品库存不足，系统提示“该菜品暂时不可购买”。   
  
---  
  
 用例描述15：从购物车移除菜品   
- 用例名称：从购物车移除菜品   
- 用例编号：UC-015   
- 参与者：用户、系统   
- 前置条件：用户已登录或未登录系统，目标菜品存在于购物车中。   
- 后置条件：目标菜品从购物车中移除。   
- 主事件流：   
 1. 用户进入购物车页面。   
 2. 用户选择目标菜品并点击“移除”按钮。   
 3. 系统从购物车中移除菜品，并更新购物车数量。   
 4. 系统提示“已成功移除菜品”。   
- 异常事件流：   
 - A1: 购物车中无目标菜品，系统提示“菜品未找到”。   
  
---  
  
 用例描述16：管理购物车   
- 用例名称：管理购物车   
- 用例编号：UC-016   
- 参与者：用户、系统   
- 前置条件：用户已登录或未登录系统，购物车中有至少一件菜品。   
- 后置条件：购物车内容更新完成。   
- 主事件流：   
 1. 用户进入购物车页面。   
 2. 用户对购物车中的菜品进行增减或清除操作。   
 3. 系统实时更新购物车内容和总价。   
 4. 系统提示“购物车已更新”。   
- 异常事件流：   
 - A1: 购物车为空，系统提示“购物车中无任何菜品”。   
  
---  
  
 用例描述17：发起支付   
- 用例名称：发起支付   
- 用例编号：UC-017   
- 参与者：用户、系统   
- 前置条件：用户已登录系统，目标订单处于“待支付”状态。   
- 后置条件：支付流程启动，订单状态更新为“支付中”。   
- 主事件流：   
 1. 用户进入订单详情页面。   
 2. 用户点击“支付”按钮。   
 3. 系统加载支付页面，展示支付方式选项（如微信、支付宝等）。   
 4. 用户选择支付方式并确认支付。   
 5. 系统调用支付接口完成支付，并更新订单状态为“已支付”。   
 6. 系统提示“支付成功”。   
- 异常事件流：   
 - A1: 支付失败（如余额不足），系统提示“支付失败，请稍后重试”。   
 - A2: 网络连接中断，系统提示“支付超时，请重新尝试”。   
  
---  
 用例描述18：取消支付   
- 用例名称：取消支付   
- 用例编号：UC-018   
- 参与者：用户、系统   
- 前置条件：用户已登录系统，目标订单处于“支付中”状态。   
- 后置条件：支付流程终止，订单状态恢复为“待支付”。   
- 主事件流：   
 1. 用户进入支付页面。   
 2. 用户点击“取消支付”按钮。   
 3. 系统弹出确认对话框。   
 4. 用户确认取消操作。   
 5. 系统终止支付流程，并将订单状态更新为“待支付”。   
 6. 系统提示“支付已取消”。   
- 异常事件流：   
 - A1: 支付已完成，系统提示“支付已完成，无法取消”。   
  
---  
  
 用例描述19：发布评论   
- 用例名称：发布评论   
- 用例编号：UC-019   
- 参与者：用户、系统   
- 前置条件：用户已登录系统，目标订单已完成且未评论。   
- 后置条件：评论成功发布并关联到目标订单或餐厅。   
- 主事件流：   
 1. 用户进入订单详情页面或餐厅页面。   
 2. 用户点击“发表评论”按钮。   
 3. 系统显示评论表单（包括评分和文字内容）。   
 4. 用户填写表单并提交。   
 5. 系统验证评论内容的合法性。   
 6. 系统保存评论信息，并提示“评论已发布”。   
- 异常事件流：   
 - A1: 评论内容为空或非法，系统提示“请填写有效评论”。   
  
---  
  
 用例描述20：管理评论   
- 用例名称：管理评论   
- 用例编号：UC-020   
- 参与者：管理员、系统   
- 前置条件：管理员已登录系统，目标评论存在。   
- 后置条件：评论被修改或删除。   
- 主事件流：   
 1. 管理员进入评论管理页面。   
 2. 管理员选择目标评论并点击“编辑”或“删除”按钮。   
 3. 若选择“编辑”，系统加载评论编辑页面，管理员修改内容并提交。   
 4. 若选择“删除”，系统弹出确认对话框，管理员确认删除操作。   
 5. 系统保存修改或删除评论，并提示操作成功。   
- 异常事件流：   
 - A1: 修改内容非法（如包含敏感词），系统提示“评论内容不符合规范”。   
  
---  
  
 用例描述21：管理支付记录   
- 用例名称：管理支付记录   
- 用例编号：UC-021   
- 参与者：管理员、系统   
- 前置条件：管理员已登录系统，支付记录存在。   
- 后置条件：支付记录被查询、修改或导出。   
- 主事件流：   
 1. 管理员进入支付记录管理页面。   
 2. 管理员选择目标支付记录并进行操作（如查看详细信息、修改备注、导出报表等）。   
 3. 系统执行相应操作并反馈结果。   
 4. 系统提示操作成功。   
- 异常事件流：   
 - A1: 支付记录不存在，系统提示“未找到相关记录”。   
  
---  
  
 用例描述22：创建管理员   
- 用例名称：创建管理员   
- 用例编号：UC-022   
- 参与者：超级管理员、系统   
- 前置条件：超级管理员已登录系统，具备创建管理员权限。   
- 后置条件：新管理员账户成功创建，系统保存管理员信息。   
- 主事件流：   
 1. 超级管理员进入管理员管理页面。   
 2. 超级管理员点击“添加管理员”按钮。   
 3. 系统显示管理员信息表单（包括用户名、密码、权限等级等）。   
 4. 超级管理员填写表单并提交。   
 5. 系统验证输入数据的合法性。   
 6. 系统保存管理员信息，并提示“管理员已创建”。   
- 异常事件流：   
 - A1: 用户名已存在，系统提示“用户名重复，请重新输入”。   
  
---  
  
 用例描述23：删除管理员   
- 用例名称：删除管理员   
- 用例编号：UC-023   
- 参与者：超级管理员、系统   
- 前置条件：超级管理员已登录系统，目标管理员账户存在。   
- 后置条件：目标管理员账户从系统中移除。   
- 主事件流：   
 1. 超级管理员进入管理员管理页面。   
 2. 超级管理员选择目标管理员并点击“删除”按钮。   
 3. 系统弹出确认对话框。   
 4. 超级管理员确认删除操作。   
 5. 系统删除管理员信息，并提示“管理员已删除”。   
- 异常事件流：   
 - A1: 删除失败（如数据库连接问题），系统提示“操作失败，请稍后重试”。   
---