项目文档

# Introduction

\*\*第一章 引言\*\*  
  
1.1 背景  
  
随着外卖市场增长迅速，食品便捷订购平台也成为了现代城市生活中不可或缺的部分。QuickEats正是作为一种满足用户快速便捷点餐需求的软件应用，致力于为用户提供从附近餐厅快速选取人气菜品的服务。通过该平台用户可以轻松完成自取或是外卖订单，所有操作仅需几次点击，同时平台提供实时订单追踪以及无缝支付服务，这一切旨在为用户提供更好的整体体验。为此，编写本需求规约说明书旨在明确系统各部分的功能要求、非功能要求及相关约束，以指导后期的开发工作。  
  
1.2 业务目标  
  
本文档旨在明确"QuickEats"的软件需求，为开发团队提供明确的指导与方向，确保系统开发过程中严格按照业务需求进行，确保最终交付的产品既能满足用户对于快速便捷点餐的需求，也能提供良好的使用体验，进一步提升用户的整体满意度。  
  
1.3 范围  
  
本文档覆盖"QuickEats"软件的所有功能需求、非功能需求以及相关的约束条件，包括但不限于用户账户管理、餐厅查看与搜索、菜品浏览与下单、订单创建与支付、订单取消与状态追踪、评价管理等功能。针对软件中每个关键模块将从用户界面、数据模型以及系统交互等多个角度详细描述。  
  
1.4 定义、缩写与缩略语  
  
- 用户 (User): 平台使用端个体，进行账户创建、管理个人资料、浏览餐厅信息、购物车操作、订购与支付过程以及查看订单详情等操作。  
- 餐厅 (Restaurant): 提供餐饮服务并展示于平台的商业实体，向用户提供点餐服务及菜品信息。  
- 菜品 (Dish): 餐厅提供的具体菜单项，用户可选择添加到自己的购物车中。  
- 购物车 (Cart): 用户临时存放所选购菜品的虚拟容器。  
- 购物车记录 (Cart Record): 记载用户添加至购物车的菜品信息。  
- 订单 (Order): 用户下单后形成的交易记录。  
- 订单状态 (Order Status): 订单在其生命周期内状态的变更记录，包括已下单、待支付、已完成等。  
- 支付 (Payment): 进行订单交易时的资金转账过程。  
- 跟踪记录 (Tracking Record): 记录外卖配送过程中订单状态的变动及时间戳。  
- 评价记录 (Review Record): 用户对所购菜品的评价与反馈记录。  
- 管理员 (Administrator): 负责系统维护与管理，包括但不限于用户账户审核、订单状态调整及评价与跟踪记录的管理。  
  
1.5 参考标准  
  
本需求规约说明书遵循IEEE-830标准撰写，以确保其结构合理、条理清晰。此外，本文档还参考了《软件需求规格说明》编写指南及行业内相关软件开发文档的标准。这些标准和指南提供了编写高质量需求文档的基础，以帮助确保最终软件产品的质量和使用满意度。

# Overall Description

### 2 总体概述  
  
#### 2.1 产品视角  
  
产品愿景：快速餐饮配送软件旨在为用户提供便捷的在线餐饮服务，支持用户浏览餐厅、选购菜品、完成购物车操作、创建订单及支付，同时支持管理用户的评价记录和订单跟踪。通过改进用户交互和简化操作流程，确保用户能高效地享受到餐饮服务。  
  
目标用户群体：目标用户群体主要是快节奏生活中的个人或小团队，对餐饮配送有着高效快捷的需求。使用高校学生、上班族和餐饮爱好者等作为主要目标市场。  
  
预期市场定位：在全球线上餐饮服务市场中竞争，通过提供高效、用户体验优化的服务来提高用户满意度，从而在市场中建立长期的位置。  
  
#### 2.2 产品功能  
  
主要功能模块包括用户账户管理、餐厅查询及浏览、购物车管理、订单创建及支付处理、浏览支付历史、订单跟踪、评价管理等。各模块间的关联主要体现在用户身份贯穿整个系统，用户需通过身份验证来执行项目各模块的功能。例如，用户身份保障了在浏览餐厅、购物车编辑、订单以及支付过程中的行为权限。  
  
#### 2.3 用户特征  
  
目标用户具备一定的互联网使用经验，且对于移动设备操作较为熟悉。不同年龄段的用户可能存在操作习惯上的差异，同时介意菜品多样性及准时交付。用户通常关注服务质量、订单状态的透明以及食品的安全卫生等问题。  
  
#### 2.4 竞品分析  
  
当前市场中已有如美团、饿了么等知名的快速餐饮配送服务软件，用户基数大、市场渗透率高。据某第三方研究机构数据统计，这些平台月活跃用户数达到千万级别，且技术迭代频繁、场景丰富多样。  
  
优势方面，本产品拟通过提升用户体验、优化用户界面设计、改进订单跟踪功能和提供个性化购物车服务等方式，来减少用户等待时间、增加用户满意度。同时，与本地小型餐厅的合作也能提高菜品的丰富度及本地感。  
  
#### 2.5 业务风险与机遇  
  
风险：市场竞争使得新加入者难以获取足够市场份额，外卖配送的冷链物流是链路的痛点；商业模式中需有足够赢利点来支撑团队的长期发展。  
  
机遇：在餐饮数字化普及的大背景下，本地小餐厅芝 STDOUT TRUNCATED

# Functional Requirement

### 3.1 创建用户  
功能编号：FR-01  
描述：管理员可以创建一个新用户信息。  
输入：管理员输入新用户的个人信息，包括UserID, UserName, Email, PhoneNumber, Address, PasswordHash, RegisterDate。  
输出：系统创建一个新用户，并将用户信息存储于用户(User)实体中。  
  
### 3.2 管理用户信息  
功能编号：FR-02  
描述：管理员可以更新用户的信息。  
输入：管理员输入更新后的用户个人信息，如UserName, Email, PhoneNumber, Address。  
输出：系统更新用户信息，并将更新后的用户信息存储于用户(User)实体中。  
  
### 3.3 删除用户  
功能编号：FR-03  
描述：管理员可以删除一个用户。  
输入：管理员选择需要删除的用户信息，如UserID。  
输出：系统删除用户信息及相关联系实体的数据，并向管理员反馈操作结果。  
  
### 3.4 浏览餐厅  
功能编号：FR-04  
描述：用户可以浏览餐厅信息。  
输入：用户的浏览请求。  
输出：展示餐厅的列表或特定餐厅的详细信息。  
  
### 3.5 搜索餐厅  
功能编号：FR-05  
描述：用户可以根据查询条件搜索餐厅。  
输入：用户输入搜索餐厅的条件，如餐厅名称、菜品类型。  
输出：返回符合条件的餐厅的列表。  
  
### 3.6 查看餐厅详情  
功能编号：FR-06  
描述：用户可以查看餐厅的各项详情。  
输入：用户选择一个特定的餐厅。  
输出：展示选定餐厅的详细信息。  
  
### 3.7 添加菜品到购物车  
功能编号：FR-07  
描述：用户可以将菜品加入购物车。  
输入：用户选择要添加的菜品信息，如DishID, CartID, Quantity。  
输出：系统创建一个购物车记录(CartRecord)并将菜品信息存入。  
  
### 3.8 从购物车移除菜品  
功能编号：FR-08  
描述：用户可以移除购物车中的菜品。  
输入：用户选择需要从购物车中移除的菜品信息，如RecordID。  
输出：购物车记录(CartRecord)被删除，并返回移除操作结果。  
  
### 3.9 清空购物车  
功能编号：FR-09  
描述：用户可以清空购物车。  
输入：用户提交清空购物车的请求。  
输出：购物车中的所有购物车记录(CartRecord)被删除，并返回清空操作结果。  
  
### 3.10 创建订单  
功能编号：FR-10  
描述：用户可以创建订单。  
输入：用户提交创建订单的请求，包括CartID及选中的支付方式。  
输出：系统根据购物车创建一个订单(Order)，并返回订单创建的结果。  
  
### 3.11 取消订单  
功能编号：FR-11  
描述：用户可以取消订单。  
输入：用户选择需要取消的订单信息，如OrderID。  
输出：系统更新订单状态为已取消，并返回取消操作结果。  
  
### 3.12 查看订单详情  
功能编号：FR-12  
描述：用户可以查看订单的详细信息。  
输入：用户选择一个订单。  
输出：展示该订单的详细信息，包含订单状态、总价等。  
  
### 3.13 支付订单  
功能编号：FR-13  
描述：用户可以对订单进行支付。  
输入：用户选择一个订单并选择支付方式。  
输出：系统创建支付记录(Payment)，更新订单状态为已支付，并返回支付结果。  
  
### 3.14 取消支付  
功能编号：FR-14  
描述：用户可以取消支付操作。  
输入：用户选择一个订单并提交取消支付请求。  
输出：系统取消支付记录(Payment)，并恢复订单到支付前的状态。  
  
### 3.15 查看支付历史  
功能编号：FR-15  
描述：用户可以查看支付历史记录。  
输入：用户提交查询请求。  
输出：返回用户所有的支付历史记录，包括支付时间、支付金额等信息。  
  
### 3.16 管理评价记录  
功能编号：FR-16  
描述：管理员可以查看和更新餐厅的评价记录。  
输入：管理员选择或输入需要操作的评价信息。  
输出：评价记录(ReviewRecord)被更新或删除，返回操作结果。  
  
### 3.17 管理跟踪记录  
功能编号：FR-17  
描述：管理员可以查看、更新和删除订单的跟踪记录。  
输入：管理员选择或输入需要操作的跟踪信息。  
输出：跟踪记录(TrackingRecord)被更新或删除，返回操作结果。  
  
### 3.18 管理购物车记录  
功能编号：FR-18  
描述：管理员可以查看、更新和删除用户的购物车记录。  
输入：管理员选择或输入需要操作的购物车信息。  
输出：购物车记录(CartRecord)被更新或删除，返回操作结果。  
  
### 3.19 查看购物车记录  
功能编号：FR-19  
描述：用户可以查看他们的购物车记录。  
输入：用户提交查看购物车记录的请求。  
输出：返回用户购物车中的商品记录，包含商品名称、数量、价格等信息。  
  
### 3.20 查看跟踪记录  
功能编号：FR-20  
描述：用户可以查看他们的订单跟踪记录。  
输入：用户提交查看跟踪记录的请求。  
输出：返回用户订单的跟踪信息，包含订单号、物流状态、更新时间等信息。  
  
### 3.21 查看评价记录  
功能编号：FR-21  
描述：用户可以查看他们提交的评价记录。  
输入：用户提交查看评价记录的请求。  
输出：返回用户评价的记录，包括商品名称、评分、评论内容、发布时间等信息。

# External Description

### 4 外部接口部分  
  
针对"QuickEats"系统功能实现过程中涉及的外部数据源以及系统外部交互接口部分，以下详细列出并分类提供了各个外部接口描述及其作用说明。这些接口对应的需求来自于各个模块的功能描述与用例沟通，它们是系统与外界数据交换的重要媒介，遵照上述的形式进行分类描述与定义。  
  
#### 4.1 用户接口  
  
用户接口主要用于处理用户的指令输入与系统信息提供，核心在于实现用户界面的友好性与交互便捷性，确保用户可以较为直观、高效地完成各自的使用需求。  
  
- \*\*用户浏览\*\*：响应用户的选择，返回餐厅列表或详细的餐厅信息，支持用户浏览功能。可通过下拉列表、按钮选择等方式和用户进行交互。  
- \*\*用户购物车操作\*\*：处理用户对购物车的添加、删除、清空等操作，支持购物车记录的创建、更改与删除。  
- \*\*查看订单及订单相关的操作\*\*：展示用户的订单信息，支持用户查看及操作如取消、支付订单。  
- \*\*支付操作\*\*：提供多种支付方式的展示和选择，以完成用户的支付操作。确保支付方式界面友好，且具有消息确认机制。  
- \*\*查看支付历史、评价记录及跟踪记录\*\*：为用户提供查看过去订单详情、支付历史、用餐评价及订单跟踪的服务内容，包括订单号、物品详情、支付金额、时间戳等信息。  
- \*\*用户账户管理\*\*：涉及用户登录验证、密码更改、个人信息查看与编辑等。为用户提供界面操作，直观展示账户信息。  
  
#### 4.2 硬件接口  
  
由于这是一个软件性质的服务，硬件接口主要是通过与终端连接，比如用户设备（手机、电脑等）与服务器的通信，具体细节取决于软件的客户端开发，因此重点在于：  
  
- 设备类型兼容性：系统会支持多种操作系统及设备类型（如 iOS, Android, Windows），确保无缝接入。  
- 网络通信协议：包括HTTP，HTTPS等标准通信协议。  
  
#### 4.3 软件接口  
  
- \*\*数据库操作接口\*\*：用于与数据库进行数据交互，包括用户信息、用户订单、购物车、评价、跟踪等数据。  
 - 创建用户时需要操作用户数据库，验证是否已存在相同用户信息，系统创建新用户并保存后需要操作数据库保存用户信息。  
 - 用户对订单、购物车操作，以及评价餐厅时的数据修改，都需要通过接口与数据库进行交互。  
 - 管理员管理功能需要与数据库进行交互，获取用户的订单、购物车，评价和跟踪信息。  
   
- \*\*外部支付模块接口\*\*：提供安全、付费的支付渠道，通过友好的支付界面支持用户进行安全交易。  
  
#### 4.4 通信接口  
  
- \*\*邮件通知接口\*\*：处理注册确认、忘记密码、订单确认、支付成功的系统邮件发送。邮件模板设计，确保信息明了易懂。  
- \*\*推送通知接口\*\*：支持会员消息推送服务，包括但不限于状态更新通知、评价反馈等，通过短信、推送消息等形式实现，保证用户的及时获取信息。  
  
以上四个部分，即用户接口、硬件接口、软件接口以及通信接口，覆盖了"QuickEats"所有可能涉及到的外部接口，所有与外部的数据交换都在这四个大类中得到明确说明，以保障系统开发与实现过程中，团队成员能正确理解接口需求。

# Non-Functional Requirement

根据你提供的概要和结构，以下是根据上述规则和输出格式整理的非功能需求：  
  
### 5 非功能需求  
  
#### 5.1 性能需求  
1. \*\*响应时间\*\*：  
 - 对于用户管理相关的操作，如查看、修改信息、添加等，响应时间应不大于2秒。  
 - 浏览餐厅信息，响应时间应不大于3秒。  
 - 搜索餐厅信息功能，系统应在接收请求的3秒内返回结果。  
 - 创建订单、支付处理应确保在3秒内完成主要操作，并在10秒内完成整个流程。  
  
2. \*\*并发处理能力\*\*：  
 - 系统应能够同时支持1000个在线用户进行常规操作而不影响用户体验。  
 - 系统需能够支持用户高峰时间段内多个并发的操作（如登入、支付、创建订单等）并且性能不应低于上述响应时间标准。  
  
#### 5.2 安全需求  
1. \*\*账号安全\*\*：  
 - 用户账户应经过多步骤验证（包括密码强度检查），避免基础暴力攻击。  
 - 登录需启用验证码，防机器人自动进行登录尝试。  
  
2. \*\*数据传输安全\*\*：  
 - 所有的数据通信应使用HTTPS协议以保障信息安全。  
 - 敏感信息如支付细节、用户账户密码使用加密处理，确保在传输过中的安全。  
  
3. \*\*数据保护和隐私\*\*：  
 - 用户数据需符合GDPR或当地法律法规，保护用户隐私。  
 - 须有机制以匿名方式处理用户反馈和数据，避免泄漏个人信息。  
  
4. \*\*安全措施\*\*：  
 - 提供数据备份与恢复机制，以防重要数据丢失。  
 - 系统应有自动检测异常登录尝试、频率异常的交易操作等安全机制。  
  
#### 5.3 其它质量需求  
1. \*\*可扩展性\*\*：  
 - 系统设计应具备一定的组件化和模块化，以促进未来系统的扩展，例如增加新的支付方式，支持多语言等。  
   
2. \*\*可移植性\*\*：  
 - 系统应能在不同操作系统和版本上正常运行，支持跨平台部署。  
  
3. \*\*可用性\*\*：  
 - 系统应设计智能化的错误反馈系统，确保用户能快速识别问题并获得有效解决方案。  
 - 应具备7 x 24小时的服务能力，对关键系统执行定期健康检查。  
  
4. \*\*可靠性\*\*：  
 - 系统的关键操作，如计算机、网络、数据库等需要有备份方案来保障系统的持续运行能力。  
 - 软件应有容错处理机制，能在异常情况下确保系统稳定运营。  
  
以上非功能需求为基本的构建，开发过程中还需要具体情境和客户反馈不断调整和完善。

# Use Case

用例名称：创建用户(Create User)  
用例编号：UC-01  
参与者：管理员  
前置条件：管理员已成功登录系统。  
后置条件：新用户信息创建成功，系统状态保持一致。  
主事件流：  
1. 管理员登录到系统。  
2. 管理员选择“创建用户”功能。  
3. 系统展示创建用户界面。  
4. 管理员输入新用户的相关信息。  
5. 系统验证输入信息的有效性。  
6. 系统创建新用户并保存信息。  
7. 系统返回用户创建成功的消息给管理员。  
后置条件达成：新用户信息创建成功，系统状态保持一致。  
异常事件流：  
A1. 如果管理员输入的信息无效或有误（如邮箱格式不正确），系统显示错误消息并要求重新输入。  
A2. 如果系统在创建用户过程中遇到错误（如数据库操作失败），系统显示错误消息并终止操作。  
--------------------------------------------------------------------------------------------  
用例名称：管理用户信息(Manage User Information)  
用例编号：UC-02  
参与者：管理员  
前置条件：管理员已成功登录系统。  
后置条件：用户信息被更新，系统状态保持一致。  
主事件流：  
1. 管理员登录到系统。  
2. 管理员选择“管理用户信息”功能。  
3. 系统展示用户信息管理界面。  
4. 管理员选择需要更新的用户记录。  
5. 管理员输入更新后的用户信息。  
6. 系统验证更新信息的有效性。  
7. 系统更新用户记录。  
8. 系统返回操作成功的消息给管理员。  
后置条件达成：用户信息被更新，系统状态保持一致。  
异常事件流：  
A1. 如果管理员输入的更新信息有误，系统显示错误消息并要求重新输入。  
A2. 如果系统在更新用户信息过程中遇到错误（如数据库操作失败），系统显示错误消息并终止操作。  
-------------------------------------------------------------------------------------------------  
用例名称：删除用户(Delete User)  
用例编号：UC-03  
参与者：管理员  
前置条件：管理员已成功登录系统。  
后置条件：对应的用户信息被删除，系统状态保持一致。  
主事件流：  
1. 管理员登录到系统。  
2. 管理员选择“删除用户”功能。  
3. 系统展示用户列表或搜索界面。  
4. 管理员选择需要删除的用户记录。  
5. 系统确认删除操作。  
6. 系统删除用户记录。  
7. 系统返回操作成功的消息给管理员。  
后置条件达成：对应的用户信息被删除，系统状态保持一致。  
异常事件流：  
A1. 如果管理员选择删除用户时，系统中不存在此用户，系统显示错误消息并终止操作。  
A2. 如果系统在执行删除操作时遇到错误（如数据库操作失败），系统显示错误消息并终止操作。  
-------------------------------------------------------------------------------------------------------  
用例名称：浏览餐厅(Browse Restaurant)  
用例编号：UC-04  
参与者：用户  
前置条件：用户已成功登录系统。  
后置条件：获得餐厅列表或详细信息。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“浏览餐厅”功能。  
3. 系统展示用户可浏览的餐厅列表或搜索界面。  
4. 用户选择感兴趣的餐厅进行浏览。  
5. 系统展示对应餐厅的详细信息或未进一步操作的列表。  
后置条件达成：获得餐厅列表或详细信息。  
异常事件流：  
A1. 如果用户尝试访问不存在的餐厅信息，系统显示错误消息并终止操作。  
A2. 如果系统在获取餐厅信息时遇到错误（如网络连接问题），系统显示错误消息并终止操作。  
-----------------------------------------------------------------------------------------------------  
用例名称：搜索餐厅(Search Restaurant)  
用例编号：UC-05  
参与者：用户  
前置条件：用户已成功登录系统。  
后置条件：返回与指定搜索条件匹配的餐厅列表。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“搜索餐厅”功能。  
3. 用户输入搜索条件（如餐厅名称、菜品类型）。  
4. 系统根据输入的搜索条件搜索餐厅。  
5. 系统返回与搜索条件匹配的餐厅列表。  
后置条件达成：返回与指定搜索条件匹配的餐厅列表。  
异常事件流：  
A1. 如果搜索条件为空或无效，系统提示用户重新输入。  
A2. 如果系统在处理搜索请求过程中遇到错误（如服务器响应时间超时），系统显示错误消息并终止操作。  
--------------------------------------------------------------------------------------------------  
用例名称：查看餐厅详情(View Restaurant Detail)  
用例编号：UC-06  
参与者：用户  
前置条件：用户已成功登录系统。  
后置条件：用户能够获取到餐厅的详细信息。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“查看餐厅详情”功能。  
3. 用户选择一个餐厅。  
4. 系统展示该餐厅的详细信息。  
后置条件达成：用户能够获取到餐厅的详细信息。  
异常事件流：  
A1. 如果选择的餐厅不存在，系统提示用户该餐厅不存在并终止操作。  
A2. 如果系统在获取餐厅详细信息时遇到错误（如数据库连接失败），系统显示错误消息并终止操作。  
---------------------------------------------------------------------------------------  
用例名称：添加菜品到购物车(Add Dish to Shopping Cart)  
用例编号：UC-07  
参与者：用户  
前置条件：用户已成功登录系统且已经选择餐厅。  
后置条件：所选菜品已经添加到用户购物车内。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“添加菜品到购物车”功能。  
3. 用户选择一个想要添加到购物车的菜品。  
4. 系统验证菜品的有效性。  
5. 增加菜品到购物车内并显示确认信息。  
后置条件达成：所选菜品已经添加到用户购物车内。  
异常事件流：  
A1. 如果用户尝试添加一个无效或不存在的菜品，系统显示错误信息并要求重新选择。  
A2. 如果出现系统错误（如数据库更新操作失败），系统提示错误信息并中断操作。  
----------------------------------------------------------------------------------------------------------  
用例名称：从购物车移除菜品(Remove Dish from Shopping Cart)  
用例编号：UC-08  
参与者：用户  
前置条件：用户已成功登录系统且购物车内存在菜品。  
后置条件：目标菜品已从购物车中移除。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“从购物车移除菜品”功能。  
3. 用户选择购物车中的菜品。  
4. 用户确认菜品移除操作。  
5. 系统执行从购物车移除的操作。  
6. 系统返回移除操作完成的信息。  
后置条件达成：目标菜品已从购物车中移除。  
异常事件流：  
A1. 如果操作失误或菜单不存在，系统显示错误信息并中止移除操作。  
A2. 如果在操作过程中出现系统错误（如更新失败），系统提示错误信息并中止操作。  
--------------------------------------------------------------------------------------------------------------  
用例名称：清空购物车(Clear Shopping Cart)  
用例编号：UC-09  
参与者：用户  
前置条件：用户已成功登录系统且购物车内存在菜品。  
后置条件：用户购物车被清空。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“清空购物车”功能。  
3. 系统验证用户是否有菜品在购物车内。  
4. 用户确认清空购物车操作。  
5. 系统执行购物车的清空。  
6. 系统返回清空购物车完成的信息。  
后置条件达成：用户购物车被清空。  
异常事件流：  
A1. 如果购物车为空，系统显示相关信息并中止操作。  
A2. 如果系统在执行删除操作时出现错误（如数据库问题），系统显示错误信息并中止操作。  
-----------------------------------------------------------------------------------------------------------  
用例名称：创建订单(Create Order)  
用例编号：UC-10  
参与者：用户  
前置条件：用户已成功登录系统且购物车内有拟下单的菜品。  
后置条件：用户能成功创建一个订单。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“创建订单”功能。  
3. 系统展示用户的购物车信息。  
4. 用户确认订单信息。  
5. 系统执行订单创建操作。  
6. 系统返回订单创建成功的消息。  
后置条件达成：用户能成功创建一个订单。  
异常事件流：  
A1. 如果购物车为空或无效物品在购物车，系统显示错误信息并中止操作。  
A2. 如果在创建订单过程中遇到系统错误（如数据库连接失败），系统显示错误信息并终止操作。  
-----------------------------------------------------------------------------------------------  
用例名称：取消订单(Cancel Order)  
用例编号：UC-11  
参与者：用户  
前置条件：用户已成功登录系统且存在未完成的订单。  
后置条件：被取消的订单状态已更改。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“取消订单”功能。  
3. 用户选择要取消的订单。  
4. 用户确认订单取消操作。  
5. 系统执行订单取消操作。  
6. 系统返回订单取消成功的消息。  
后置条件达成：被取消的订单状态已更改。  
异常事件流：  
A1. 如果选择取消的订单不存在或状态不适用取消，系统显示错误信息并终止操作。  
A2. 如果在取消订单过程中遇到系统错误（如服务器响应失败）, 系统显示错误信息并终止操作。  
------------------------------------------------------------------------------------------------  
用例名称：查看订单详情(View Order Detail)  
用例编号：UC-12  
参与者：用户  
前置条件：用户已成功登录系统且拥有至少一项订单。  
后置条件：用户获得订单的详细信息。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“查看订单详情”功能。  
3. 用户选择一个订单。  
4. 系统展示该订单的详细信息。  
后置条件达成：用户获得订单的详细信息。  
异常事件流：  
A1. 如果选择的订单不存在，系统提示该订单不存在并终止操作。  
A2. 如果系统在获取订单详细信息时遇到错误（如数据库连接失败），系统显示错误信息并终止操作。  
------------------------------------------------------------------------------------------------------  
用例名称：支付订单(Pay Order)  
用例编号：UC-13  
参与者：用户  
前置条件：用户已成功登录系统且拥有未支付的订单。  
后置条件：用户的订单已被支付。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“支付订单”功能。  
3. 用户选择一个订单进行支付。  
4. 系统展示支付界面及支付方式。  
5. 用户选择支付方式并完成支付过程。  
6. 系统验证支付结果。  
7. 系统更新订单状态为已支付。  
8. 系统返回支付成功的消息。  
后置条件达成：用户的订单已被支付。  
异常事件流：  
A1. 如果在支付过程中遇到无效支付信息，系统显示错误信息并要求重新输入。  
A2. 如果在支付过程中出现系统错误（如银行或者支付网关的API失败），系统显示错误信息并终止操作。  
-----------------------------------------------------------------------------------------------------------  
用例名称：取消支付(Cancel Payment)  
用例编号：UC-14  
参与者：用户  
前置条件：用户已成功登录，订单处于支付中状态。  
后置条件：订单恢复到支付前状态或被撤销。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“取消支付”功能。  
3. 用户确认取消操作。  
4. 系统执行订单支付取消操作。  
5. 系统返回支付取消成功的消息。  
后置条件达成：订单恢复到支付前状态或被撤销。  
异常事件流：  
A1. 如果系统取消支付过程中出现错误（如操作超时），系统显示错误信息并终止操作。  
A2. 如果支付状态已更改无法取消，系统显示相关信息并中断操作。  
-------------------------------------------------------------------------------------------------  
用例名称：查看支付历史(View Payment History)  
用例编号：UC-15  
参与者：用户  
前置条件：用户已成功登录系统。  
后置条件：用户能够查看到所有的支付历史记录。  
主事件流：  
1. 用户登录到系统。  
2. 用户选择“查看支付历史”功能。  
3. 系统展示所有的支付历史记录。  
后置条件达成：用户能够查看到所有的支付历史记录。  
异常事件流：  
A1. 如果支付历史记录为空，系统提示没有支付历史并终止操作。  
A2. 如果系统在获取历史记录过程中遇到错误（如数据查询失败），系统显示错误信息并终止操作。  
------------------------------------------------------------------------------------------------------  
用例名称：管理评价记录(Manage Rating Record)  
用例编号：UC-16  
参与者：管理员  
前置条件：管理员已成功登录系统。  
后置条件：评价信息被更新或删除，系统状态保持一致。  
主事件流：  
1. 管理员登录到系统。  
2. 管理员选择“管理评价记录”功能。  
3. 系统展示评价记录列表或搜索功能。  
4. 管理员选择需要更新或删除的评价记录。  
5. 管理员操作评价记录更新或删除。  
6. 系统验证更新或删除操作的有效性。  
7. 系统更新或删除评价记录。  
8. 系统返回操作成功的消息给管理员。  
后置条件达成：评价信息被更新或删除，系统状态保持一致。  
异常事件流：  
A1. 如果管理员操作不合理如删除评分记录无效，系统显示错误信息。  
A2. 如果在操作过程中系统遇到错误例如无法更新数据库，系统显示错误信息并终止操作。  
-----------------------------------------------------------------------------------------------------  
用例名称：管理跟踪记录(Manage Tracking Record)  
用例编号：UC-17  
参与者：管理员  
前置条件：管理员已成功登录系统。  
后置条件：跟踪信息被更新或删除，系统状态保持一致。  
主事件流：  
1. 管理员登录到系统。  
2. 管理员选择“管理跟踪记录”功能。  
3. 系统展示跟踪记录列表或搜索界面。  
4. 管理员选择需要更新或删除的跟踪记录。  
5. 管理员执行记录的更新或删除操作。  
6. 系统验证更新或删除操作的有效性。  
7. 系统更新或删除跟踪记录。  
8. 系统返回操作成功的消息给管理员。  
后置条件达成：跟踪信息被更新或删除，系统状态保持一致。  
异常事件流：  
A1. 如果跟踪记录不存在，系统显示错误信息并终止操作。  
A2. 如果在操作过程中系统出现错误（如数据库失败），系统显示错误信息并终止操作。  
--------------------------------------------------------------------------------------------------------------  
用例名称：管理购物车记录(Manage Shopping Cart Record)  
用例编号：UC-18  
参与者：管理员  
前置条件：管理员已成功登录系统。  
后置条件：购物车信息被更新或删除，系统状态保持一致。  
主事件流：  
1. 管理员登录到系统。  
2. 管理员选择“管理购物车记录”功能。  
3. 系统展示所有购物车记录或搜索功能。  
4. 管理员选择需要更新或删除的购物车记录。  
5. 管理员执行记录的更新或删除操作。  
6. 系统验证更新或删除操作的有效性。  
7. 系统更新或删除购物车记录。  
8. 系统返回操作成功的消息给管理员。  
后置条件达成：购物车信息被更新或删除，系统状态保持一致。  
异常事件流：  
A1. 如果购物车记录不存在，系统显示错误信息并终止操作。  
A2. 如果在操作过程中系统出现错误（如数据库操作失败），系统显示错误信息并终止操作。  
  
---  
  
### 用例名称：查看购物车记录(View Shopping Cart Record)   
\*\*用例编号\*\*：UC-19   
\*\*参与者\*\*：用户   
\*\*前置条件\*\*：用户已成功登录系统。   
\*\*后置条件\*\*：购物车记录被正确展示，系统状态保持一致。   
  
\*\*主事件流\*\*：   
1. 用户登录到系统。   
2. 用户选择“查看购物车记录”功能。   
3. 系统验证用户权限并查询该用户的购物车记录。   
4. 系统将购物车记录以列表形式展示给用户（包括商品名称、数量、价格等信息）。   
5. 用户浏览购物车记录。   
6. 系统保持购物车记录的显示状态，等待用户进一步操作。   
\*\*后置条件达成\*\*：购物车记录被正确展示，系统状态保持一致。   
  
\*\*异常事件流\*\*：   
A1. 如果购物车记录为空，系统显示提示信息“购物车为空”，并终止操作。   
A2. 如果在操作过程中系统出现错误（如数据库查询失败），系统显示错误信息并终止操作。   
  
---  
  
### 用例名称：查看跟踪记录(View Tracking Record)   
\*\*用例编号\*\*：UC-20   
\*\*参与者\*\*：用户   
\*\*前置条件\*\*：用户已成功登录系统且存在有效的跟踪记录。   
\*\*后置条件\*\*：跟踪记录被正确展示，系统状态保持一致。   
  
\*\*主事件流\*\*：   
1. 用户登录到系统。   
2. 用户选择“查看跟踪记录”功能。   
3. 系统验证用户权限并查询与该用户相关的跟踪记录。   
4. 系统将跟踪记录以列表形式展示给用户（包括订单号、物流状态、更新时间等信息）。   
5. 用户浏览跟踪记录。   
6. 系统保持跟踪记录的显示状态，等待用户进一步操作。   
\*\*后置条件达成\*\*：跟踪记录被正确展示，系统状态保持一致。   
  
\*\*异常事件流\*\*：   
A1. 如果跟踪记录不存在，系统显示提示信息“暂无跟踪记录”，并终止操作。   
A2. 如果在操作过程中系统出现错误（如数据库查询失败），系统显示错误信息并终止操作。   
  
---  
  
### 用例名称：查看评价(View Reviews)   
\*\*用例编号\*\*：UC-21   
\*\*参与者\*\*：用户   
\*\*前置条件\*\*：用户已成功登录系统且存在有效的评价记录。   
\*\*后置条件\*\*：评价记录被正确展示，系统状态保持一致。   
  
\*\*主事件流\*\*：   
1. 用户登录到系统。   
2. 用户选择“查看评价”功能。   
3. 系统验证用户权限并查询与该用户相关的评价记录。   
4. 系统将评价记录以列表形式展示给用户（包括商品名称、评分、评论内容、发布时间等信息）。   
5. 用户浏览评价记录。   
6. 系统保持评价记录的显示状态，等待用户进一步操作。   
\*\*后置条件达成\*\*：评价记录被正确展示，系统状态保持一致。   
  
\*\*异常事件流\*\*：   
A1. 如果评价记录不存在，系统显示提示信息“暂无评价记录”，并终止操作。   
A2. 如果在操作过程中系统出现错误（如数据库查询失败），系统显示错误信息并终止操作。