# 一、功能性需求评审

F1 用户注册、登录和注销账户（高优先级）

- 评审意见：需要考虑账户安全性，如密码强度校验、验证码、防止暴力破解等安全措施。

- 建议：明确支持的第三方登录方式（如微信、支付宝），并制定相应的流程。

F2 用户查看餐厅列表和菜品信息（高优先级）

- 评审意见：需要确保餐厅和菜品信息的实时性和准确性。

- 建议：考虑增加按距离、评分或菜系等条件的筛选和排序功能，提升用户体验。

F3 用户将菜品加入购物车（高优先级）

- 评审意见：需要处理购物车的有效期，以及跨设备同步的问题。

- 建议：对于未登录用户，需考虑购物车数据的临时存储方式。

F4 用户下单并选择配送地址（高优先级）

- 评审意见：需要确保用户填写的配送地址准确，并提供地址管理功能。

- 建议：集成地址自动填充或地图选点功能，提升输入效率。

F5 用户选择在线支付或货到付款方式（中优先级）

- 评审意见：支付方式需要符合相关法律法规，确保资金安全。

- 建议：明确支持的支付渠道（如支付宝、微信支付），并提供支付结果的反馈。

F6 商家管理菜品、设置价格和库存（高优先级）

- 评审意见：需要提供友好的商家后台，支持批量操作和数据统计。

- 建议：增加菜品的分类管理和促销活动设置功能。

F7 商家查看订单并确认接单（高优先级）

- 评审意见：订单处理流程需要清晰，避免漏单或重复接单。

- 建议：提供订单提醒和自动接单功能，提高效率。

F8 配送员查看并接收待配送订单（中优先级）

- 评审意见：需要规划合理的派单机制，防止订单被多次接收。

- 建议：实现定位功能，帮助配送员规划最优配送路线。

F9 用户对订单进行评价和打分（低优先级）

- 评审意见：评价功能有助于提升服务质量，但需注意内容审核。

- 建议：设立举报机制，防止虚假或不良评价。

F10 用户查看历史订单及其状态（中优先级）

- 评审意见：历史订单数据需要进行存储和隐私保护。

- 建议：提供订单搜索和筛选功能，方便用户查找。

F11 系统提供订单状态的实时更新（高优先级）

- 评审意见：需要技术支持，实现订单状态的实时推送或刷新。

- 建议：使用WebSocket或长轮询技术，实现实时性。

F12 管理员管理用户、商家和配送员账号（低优先级）

- 评审意见：需要明确管理员的权限范围，防止权限滥用。

- 建议：实施权限分级管理，增加操作日志记录。

# 二、非功能性需求评审

NF1 系统应在3秒内加载主要页面（高优先级）

- 评审意见：需要优化前端性能，减少不必要的资源加载。

- 建议：采用懒加载、资源压缩等技术，提升页面加载速度。

NF2 系统应支持至少1000个并发用户访问（中优先级）

- 评审意见：需要进行压力测试，验证系统能承受的最大并发量。

- 建议：考虑使用负载均衡、缓存等优化手段，提升并发处理能力。

NF3 用户数据应进行加密存储以保障隐私（高优先级）

- 评审意见：必须遵循���关的数据安全法规，如GDPR等。

- 建议：采用加密算法存储敏感信息，定期进行安全审计。

NF4 系统应在故障时提供自动恢复机制（低优先级）

- 评审意见：需要设计高可用架构，防止单点故障。

- 建议：实现数据备份和自动切换机制，确保服务持续性。

NF5 系统应具备友好的用户界面，操作简便（中优先级）

- 评审意见：用户体验直接影响用户留存率。

- 建议：遵循UI/UX设计原则，定期进行用户测试和反馈收集。

NF6 支持移动端适配，兼容iOS和Android平台（高优先级）

- 评审意见：移动端用户比例较高，必须确保兼容性。

- 建议：采用响应式设计，或开发原生移动应用。

NF7 系统应具备良好的扩展性，便于后续功能的增加（中优先级）

- 评审意见：需要在架构设计阶段考虑扩展性。

- 建议：采用微服务架构，模块化设计，方便后期维护和升级。

# 三、需求跟踪矩阵评审

- 实现状态需动态更新：建议在项目开发过程中，定期更新需求的实现状态，确保需求跟踪的准确性。

- 备注信息的完善：在备注中记录任何需求变更、决策和特殊情况，保持信息透明。

# 四、其他建议

- 明确需求优先级：对于中、低优先级的需求，需根据项目时间和资源进行取舍，确保核心功能按时交付。

- 风险识别与管理：针对高风险需求，制定应急预案，避免项目延期。

- 法律和合规性：确保系统符合相关的法律法规，如数据保护法、电商法等。