可行性分析(研究)报告(FAR)

说明：

l.《可行性分析(研究)报告》(FAR)是项目初期策划的结果，它分析了项目的要求、目标和环境；提出了几种可供选择的方案；并从技术、经济和法律各方面进行了可行性分析。可作为项目决策的依据。

2.FAR也可以作为项目建议书、投标书等文件的基础。

目录

[可行性分析(研究)报告(FAR) 1](#_Toc235938394)

[1引言 4](#_Toc235938395)

[1.1标识 4](#_Toc235938396)

[1.2背景 4](#_Toc235938397)

[1.3项目概述 4](#_Toc235938398)

[1.4文档概述 4](#_Toc235938399)

[2引用文件 4](#_Toc235938400)

[3可行性分析的前提 5](#_Toc235938401)

[3.1项目的要求 5](#_Toc235938402)

[3.2项目的目标 5](#_Toc235938403)

[3.3项目的环境、条件、假定和限制 5](#_Toc235938404)

[3.4进行可行性分析的方法 5](#_Toc235938405)

[4可选的方案 5](#_Toc235938406)

[4.1原有方案的优缺点、局限性及存在的问题 5](#_Toc235938407)

[4.2可重用的系统，与要求之间的差距 5](#_Toc235938408)

[4.3可选择的系统方案1 5](#_Toc235938409)

[4.4可选择的系统方案2 5](#_Toc235938410)

[4.5选择最终方案的准则 5](#_Toc235938411)

[5所建议的系统 5](#_Toc235938412)

[5.1对所建议的系统的说明 5](#_Toc235938413)

[5.2数据流程和处理流程 5](#_Toc235938414)

[5.3与原系统的比较(若有原系统) 5](#_Toc235938415)

[5.4影响(或要求) 6](#_Toc235938416)

[5.4.1设备 6](#_Toc235938417)

[5.4.2软件 6](#_Toc235938418)

[5.4.3运行 6](#_Toc235938419)

[5.4.4开发 6](#_Toc235938420)

[5.4.5环境 6](#_Toc235938421)

[5.4.6经费 6](#_Toc235938422)

[5.5局限性 6](#_Toc235938423)

[6经济可行性(成本----效益分析) 6](#_Toc235938424)

[6.1投资 6](#_Toc235938425)

[6.2预期的经济效益 7](#_Toc235938426)

[6.2.1一次性收益 7](#_Toc235938427)

[6.2.2非一次性收益 7](#_Toc235938428)

[6.2.3不可定量的收益 7](#_Toc235938429)

[6.2.4收益/投资比 7](#_Toc235938430)

[6.2.5投资回收周期 7](#_Toc235938431)

[6.3市场预测 7](#_Toc235938432)

[7技术可行性(技术风险评价) 7](#_Toc235938433)

[8法律可行性 7](#_Toc235938434)

[9用户使用可行性 7](#_Toc235938435)

[10其他与项目有关的问题 7](#_Toc235938436)

[11注解 8](#_Toc235938437)

[附录 8](#_Toc235938438)

# 1引言

## **1.1标识**

## **1.1.1 系统和软件标识**

## 本文档适用于以下系统和软件：

## 系统软件： Windows10、Windows11、MacOS、Linux 等操作系统

## 软件软件： Microsoft Word（office2019 及以上） 、WPS（11.1.0.13703 版本及以上）、飞书（V3.18.1 版本及以上）等文字处理器应用程序

## **1.1.2 文档标识**

## 标识号：A0001

## 标题：外卖系统（Online Food Delivery System）

## 缩略词语：OFDS（Online Food Delivery System）

## 版本号： 2024.1.0.0

## 发行号： 2024

## **1.1.3 修订历史**

## 版本号 修订日期 修订说明 作者

## v1.0 [2024/3/11] [编写文档各项内容] [小组各成员]

## v1.1 [2024/3/15] [编写文档各项内容] [小组各成员]

## **1.1.4 关于本文档**

## 本文档提供了关于系统和软件的详细信息，包括标识、版本历史和修订记录。请查阅相关部分以获取所需信息。

## **1.1.5 注意事项**

## 在使用本文档时，请确保查阅最新版本的文档，以获取最准确和最新的信息。若有任何疑问或需要进一步解释，请联系本软件支持团队

## **1.2背景**

**1.2.1提出条件**

（1）市场需求-近年来，随着人们生活节奏的加快和移动互联网的普及，外卖服务需求不断增加。（2）技术先进性-新一代技术的发展为在线外卖系统提供了更广阔的发展空间，包括智能化订餐、实时物流追踪等。

**1.2.2提出者的要求、目标**

（1）提高用户体验- 创建一个简单、直观、高效的在线外卖平台，提高用户在订餐和外卖过程中的体验。（2）拓展市场份额-通过与各类餐饮企业的合作，拓展在线外卖服务的市场份额，成为该领域的领导者。

**1.2.3技术创新**

推动技术在餐饮行业的应用，实现更智能、更便捷的服务。

**1.2.4实现环境**

互联网普及相关网络技术支持。

**1.2.5限制条件**

（1）法规合规性-在线外卖服务需符合当地和国家相关法规。（2）应保障数据安全-系统需具备高水平的数据安全保障，防止用户隐私数据受到侵犯。

## **1.3项目概述**

**1.3.1 一般特性**

外卖系统（java项目）是一款面向现代社会需求的在线外卖平台。其一般特性包括：

智能化订餐： 提供用户友好的订餐界面，支持个性化推荐和智能搜索功能。

实时物流追踪： 提供订单实时追踪，确保用户清晰了解外卖送达时间。

多样支付方式： 集成多种支付方式，包括在线支付、货到付款等，以满足用户不同支付需求。

多平台支持： 提供Web和移动应用版本，覆盖多平台，确保用户可以在任何设备上方便地使用。

**1.3.2. 历史**：

外卖系统的开发源于对市场需求的深刻洞察和对技术创新的不断追求。历经几轮迭代和优化，系统在功能和性能上不断完善，经过长期的运营积累了丰富的用户体验和技术实践。

**1.3.3. 项目参与方**

投资方： 由瑞吉集团投资支持，致力于打造领先的在线外卖平台。

需方： 消费者作为系统的主要需方，期望通过平台便捷地获取美味食品。

用户： 涵盖各类消费者，包括个人用户、企业订餐和特殊需求群体。

开发方： 由一支经验丰富、技术过硬的开发团队负责系统的设计、开发和维护。

支持机构： 设立专业的客户服务团队，以确保及时响应用户问题并提供技术支持。

**1.3.4. 运行现场**

当前，外卖系统已经在青岛、济南等地投入运营，覆盖了广泛的餐饮商户。计划未来拓展至更多城市，提供更广泛的外卖服务。

**1.3.5. 文档要求**

用户手册： 提供详细的用户使用手册，指导用户注册、订餐、支付等操作步骤。

开发文档： 包括系统架构、数据库设计、接口文档等，为开发人员提供系统维护和二次开发的依据。

运维手册： 指导运维人员进行系统的监控、维护和故障排查。

安全策略文档： 描述系统的安全策略，包括数据加密、用户隐私保护等。

通过综合考虑上述方面，瑞吉外卖系统致力于为用户提供便捷、高效、安全的在线外卖服务，实现消费者、商户和平台的共赢局面。

## **1.4文档概述**

本文档旨在提供对外卖系统的全面了解，包括系统的用途、功能特性以及相关的保密性和私密性要求。通过详细描述系统的设计、实施和维护方面的信息，本文档为各方利益相关者提供了关键性的指导和参考。

**1.4.1用途和内容：**

系统用途： 本文档主要介绍外卖系统的设计目的和预期效果。系统的主要用途包括提供用户友好的订餐平台、实现实时物流追踪、支持多种支付方式等，旨在满足广泛用户群体的需求。

**1.4.2功能特性：** 详细描述系统的功能特性，包括但不限于用户注册、订餐流程、支付系统、订单跟踪、商户管理等方面。提供读者对系统功能的深入理解，以便更好地使用和维护系统。

保密性和私密性要求：

为确保系统的正常运行和用户信息的安全性，本文档强调以下保密性和私密性要求：

用户信息保密： 用户的个人信息、支付信息等属于敏感数据，要求系统在处理和存储这些信息时遵循高标准的保密性要求，确保用户隐私得到妥善保护。

系统配置和接口保密： 文档中包含了系统的配置信息、接口设计等关键性内容，这些信息的保密性对于系统的稳定运行至关重要。只有授权人员可以访问这些敏感信息。

安全策略和控制： 文档概述了系统的安全策略，包括数据加密、访问控制等方面的措施。这些安全策略的保密性要求确保系统不易受到未授权访问和恶意攻击。

通过仔细阅读本文档，用户、开发人员、维护人员等各方利益相关者将更好地理解外卖系统的设计原理、功能特性以及保密性和私密性要求。本文档将为系统的使用、维护和安全性提供全方位的指导，确保系统在保持高效运行的同时，保护用户和业务信息的隐私安全。

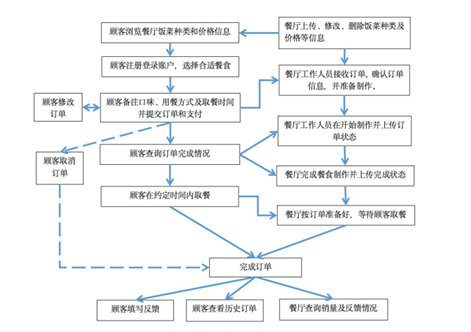
# **2引用文件**

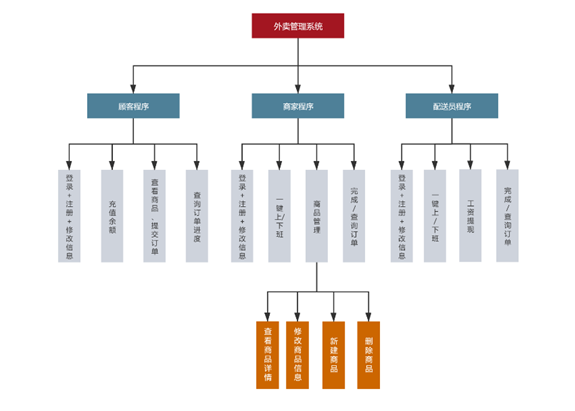
本文档（FAR）的内容没有引用其他文档，由不错不错小组各成员自行编写完成。

# 3可行性分析的前提

## 3.1项目的要求

1. **用户管理**：
   1. 提供客户、商家和骑手的注册和登录功能。
   2. 允许用户管理其个人资料和账户设置。
2. **商家管理**：
   1. 商家能够添加和管理菜单、更新库存和价格。
   2. 提供订单管理功能，包括接受订单、处理订单、查看订单历史等。
3. **菜单浏览和订购**：
   1. 客户能够浏览不同餐厅的菜单、查看菜品详情和价格。
   2. 提供购物车功能，客户可以将菜品加入购物车，并进行支付。
4. **订单管理**：
   1. 客户能够下订单、选择配送地址和支付方式。
   2. 骑手能够接受订单、查看订单详情和配送地址，并更新配送状态。
5. **配送功能**：
   1. 提供订单分配和调度功能，将订单分配给可用的骑手。
   2. 骑手能够通过导航功能找到配送地址，并实时更新配送状态。
6. **支付和结算**：
   1. 提供多种支付方式，如在线支付、货到付款等。
   2. 商家能够查看销售统计和订单收入，并申请提现。
7. **评价和反馈**：
   1. 客户能够对菜品和服务进行评价和反馈。
   2. 商家能够查看评价和反馈，并做出相应的改进。
8. **通知和提醒**：
   1. 提供订单状态更新通知，如订单确认、配送中、已送达等。
   2. 商家和骑手能够接收到新订单和重要提醒的通知。
9. **管理后台**：
   1. 管理员能够对用户、商家和骑手进行管理，包括审核、封禁和解封账户。
   2. 提供数据统计和报表功能，以便分析业务情况和用户行为。





## 3.2项目的目标

* 实现一个功能完善、用户体验良好的外卖系统软件，满足客户、商家和骑手的需求。
* 提供高效的订单管理和配送服务，确保订单准时送达。
* 建立稳定的商业模式，实现长期盈利和市场占有率的增长。

## 3.3项目的环境、条件、假定和限制

#### 环境

##### **市场环境**

* 外卖服务市场竞争激烈，但需求旺盛，用户对于快速配送和优质服务的期望较高。

##### **构建环境**

1. 该项目是外卖系统的软件开发项目。
2. 项目旨在开发一个涵盖外卖服务各个环节的综合系统，包括用户管理、商家管理、订单管理、配送管理、支付管理、评价管理等。
3. 该项目涉及多个利益相关者，包括开发商、项目经理、技术团队和最终用户。
4. 该项目需要不同团队和部门之间的高度协作和协调。

#### 条件

1. 该项目必须在特定的时间范围和预算内完成，以满足市场需求。
2. 项目必须满足质量标准和用户要求，确保软件稳定可靠、用户体验良好。
3. 该项目必须符合法律和法规要求，包括数据隐私、支付安全和个人信息保护。
4. 该项目必须具有可扩展性和灵活性，以适应未来的变化和更新，例如增加新功能或适应用户增长。

#### 假设

1. 用户需求和系统规范被准确识别和记录，以确保开发的软件能够满足用户期望。
2. 开发团队具备开发外卖系统所需的技能和专业知识，能够有效地进行软件开发和维护。
3. 该系统可以与现有的外卖系统和基础设施集成，以确保顺利的数据交换和业务流程。
4. 最终用户将采用并有效地利用该系统，为商家提供订单并为骑手提供配送任务。

#### 限制

* 外卖服务市场受到法律法规的严格限制，包括食品安全、支付安全和个人信息保护等方面，需要严格遵守以避免法律风险。

## 3.4进行可行性分析的方法

* 市场调研：调查外卖服务市场的需求和竞争情况，评估市场潜力和定位。
* 技术评估：评估可行的技术方案，包括开发工具、框架和数据库等。
* 商业模式分析：分析不同的盈利模式，评估收入和成本，确定最合适的商业模式。
* 法律和合规性检查：确保软件符合当地的法律法规，避免法律风险。

# 4可选的方案

## 4.1原有方案的优缺点、局限性及存在的问题

**优点：**

* 开发成本低。
* 简单易操作，适用于简单的数据处理。

**缺点和局限性：**

* 缺乏前端界面，用户体验差。
* 不具备良好的可扩展性和可重用性，无法满足外卖系统复杂的业务需求。
* 无法支持多用户同时操作，效率低下。
* 数据易丢失，安全性低。

## 4.2可重用的系统，与要求之间的差距

在了解原有方案的优缺点之后，我们决定选择可重用的系统。然而，在现有的系统和我们外卖系统的需求之间仍然存在一些差距，包括但不限于以下方面：

* 原有系统缺少前端界面和用户交互。
* 原有系统需要具备良好的可扩展性，使用更先进的技术，避免各模块之间的过耦合性。
* 需要系统具备稳定的性能和安全性，可以保障数据的完整性和机密性。
* 原有系统的功能和外卖系统要求之间存在一定的差距，需要进行改进和扩展。

## 4.3可选择的系统方案1

针对以上差距，我们决定选择一个基于现有系统进行二次开发和扩展的方案。具体来说，该方案采用 C/S 架构，为用户提供客户端应用进行交互，该客户端应用提供一个用户友好的交互界面。这个方案包括以下特点：

**优点：**

* 基于现有系统进行二次开发，节省开发成本。
* 具备客户端前端界面和用户交互功能，提高用户体验和操作效率。
* 可以根据需求进行定制开发和扩展。
* 容易进行数据的备份，数据安全性较高。

**缺点和局限性：**

* 需要投入一定的开发资源和时间。
* 可能存在一定的技术风险。
* 系统耦合性过大且不稳定，不利于客户端的更新维护，用户负担较重。

## 4.4可选择的系统方案2

除了方案 1 之外，我们还考虑了一个全新开发的 Web C/S 方案。该方案采用了前后端分离的 C/S 架构，可以复用原方案的后端系统，为前端暴露出相应的接口，保证了数据的安全性和系统的可维护性；采用了 Web 前端界面，使得用户负担轻量化，优化了用户体验。该方案的主要特点包括：

**优点：**

* 具备 Web 前端界面和用户交互功能，可以进一步提高用户体验。
* 可以完全根据需求进行定制开发。
* 可以根据业务发展需要进行模块化的扩展和升级。
* 保证了数据与系统的分离，可以定期进行数据的备份，保证了用户数据安全性、鲁棒性。
* 采用前后端分离，模块间耦合性低，便于系统调试、维护和更新。

**缺点和局限性：**

* 开发成本较高。
* 技术风险较大。

## 4.5选择最终方案的准则

* 技术可行性：选定方案是否具备开发和维护的能力。
* 商业可行性：方案是否能够实现长期盈利和市场竞争优势。
* 用户体验：方案是否能够提供良好的用户体验，满足用户的需求和期望。
* 合规性：方案是否符合当地法律法规，避免法律风险。
* 成本效益：方案的开发和运营成本是否可控，是否能够带来良好的投资回报率。

# 5所建议的系统

## 5.1对所建议的系统的说明

本项目是为餐饮企业定制的一款软件产品，包括系统管理后台和移动端应用两部分，其中系统管理后台主要提供给餐饮企业内部员工使用，可以对餐厅的分类，菜品，套餐，订单，员工等进行管理维护。移动端应用主要提供给消费者使用，可以在线浏览菜品，添加购物车，下单等。

编程语言：Java

数据库：Mysql

## 5.2数据流程和处理流程

**数据流程：**

1. 用户数据：当用户注册或登录外卖平台时，系统会收集并储存用户的个人信息，如：姓名、地址、电话等。这些数据用于后续的订单处理，配送及客户服务。
2. 订单数据：当用户下单时，系统会记录订单内容，包括菜品、数量、价格、配送地址、支付方式等，这些订单数据是后续处理流程的重要依据。
3. 餐厅数据：平台上的餐厅需要提供其菜单、价格、营业时间等信息，这些数据用于用户在平台上浏览和选择菜品。
4. 配送员数据：配送员的信息，如姓名、联系方式、当前位置等，会被系统实时更新和记录，以便进行订单分配和跟踪。

**处理流程：**

1. 订单接收与确认：当用户提交订单后，系统会即时接收并确认订单信息。同时，系统会根据订单内容，配送地址等因素自动分配给餐厅和最合适的配送员。
2. 餐厅接单和备餐：餐厅收到订单后，会确认接单并开始备餐，系统会实时跟踪备餐进度，确保菜品能够按时制作完成。
3. 配送员接单与配送：系统根据订单信息和配送员位置，自动分配给最佳的配送员，配送员通过系统接收订单，并按照导航前往餐厅取餐，然后在规定的时间内送达用户指定的地址。

## 5.3与原系统的比较(若有原系统)

**数据流程**：

1. 数据采集与储存：利用更先进的技术，如大数据和云计算，来实时收集，处理并存储海量数据，包括用户行为，消费习惯，餐厅销售情况等。
2. 数据处理与分析：运用更高级的算法和模型，进行深度的数据挖掘和分析，从而优化业务的处理。

**处理流程**：

1. 订单处理效率：通过智能化和自动化的手段，如智能调度算法，实现订单的快速分配和跟踪，大大提高了处理效率。
2. 用户体验优化：通过个性化算法，为用户提供更加精准的菜品推荐，通过实时反馈和评价系统，让用户能够更便捷地表达意见和建议，从而不断提高服务质量。

## 5.4影响(或要求)

### 5.4.1设备

1. 系统需要快速、稳定地运行，以确保用户在使用过程中的流畅体验。这要求设备具备较高的处理器性能、充足的内存和存储空间，以及良好的网络连接能力
2. 系统需要保护用户数据的安全，防止数据泄露和被盗用。因此，设备需要具备一定的安全防护能力，如数据加密、防病毒攻击等。
3. 由于不同用户可能使用不同品牌和型号的设备，因此外卖系统需要具备良好的跨平台兼容性，能够在多种操作系统和设备上正常运行。

### 5.4.2软件

1. 外卖软件需要为用户提供直观、简洁且易于操作的用户界面。这包括清晰的菜单展示、快速的搜索和筛选功能，以及顺畅的订单流程。同时，软件还需要考虑到不同设备和平台的显示效果，确保在各种环境下都能提供良好的用户体验。
2. 外卖系统需要处理大量的用户请求和订单数据，因此软件必须具备高性能和稳定性。这包括快速的响应速度、低延迟、高并发处理能力以及良好的容错机制，确保系统能够稳定、高效地运行。
3. 保护用户数据的安全是外卖软件的重要任务。软件需要采用加密技术、安全认证等手段来保护用户信息，防止数据泄露和非法访问。同时，软件还需要定期进行安全检查和更新，及时修复潜在的安全漏洞
4. 外卖软件需要具备多种功能以满足用户和商家的需求。例如，用户需要能够在线浏览菜单、下单、支付、查看订单状态和评价商家；商家则需要能够管理订单、库存、财务和客户信息等。这些功能需要设计得既实用又易于操作。
5. 通过收集和分析用户数据，外卖软件可以提供更个性化的服务，如智能推荐菜品、优惠活动等。这不仅可以提升用户体验，还可以帮助商家更好地了解用户需求，优化经营策略。

### 5.4.3运行

1. **系统稳定性**：外卖系统的稳定性对于其正常运行至关重要。系统必须能够持续、稳定地提供服务，避免出现崩溃、卡顿或数据丢失等问题。这要求系统架构合理、代码质量高，并具备容错和灾备能力，以确保在各种情况下都能保持稳定运行。
2. **安全性保障**：外卖系统涉及用户的个人信息、支付信息等敏感数据，因此安全性是运行过程中的重要考虑因素。系统必须采用加密技术、安全认证等手段来保护数据的安全，防止数据泄露和非法访问。同时，系统还需要建立严格的安全管理制度和应急响应机制，以应对可能的安全威胁和攻击。
3. **可扩展性与灵活性**：随着外卖市场的不断扩大和用户需求的变化，外卖系统需要具备可扩展性和灵活性。系统需要能够方便地添加新功能、调整业务流程，以适应市场的变化和用户的需求。同时，系统还需要支持横向和纵向的扩展，以满足不断增长的用户和订单量。
4. **性能优化**：外卖系统需要处理大量的用户请求和订单数据，因此对性能的要求非常高。系统必须能够快速响应用户操作，及时处理订单，并避免延迟和卡顿。为了实现性能优化，系统需要采用高效的算法和数据结构，进行负载均衡和缓存策略的设计，以及定期的性能调优和监控。
5. **实时监控与故障排查**：外卖系统需要建立实时监控机制，对系统的运行状态、性能指标和异常情况进行实时监控和预警。这有助于及时发现和解决潜在问题，确保系统的稳定运行。同时，系统还需要具备故障排查和恢复能力，以便在出现故障时能够迅速定位原因并进行修复。

### 5.4.4开发

1. **技术选型与架构设计**：开发外卖系统需要选择适合的技术栈和架构设计。这包括前端技术、后端技术、以及数据库技术。同时，系统架构需要考虑到高可用性、可扩展性和安全性，确保系统能够应对未来的业务增长和变化。
2. **功能需求实现**：外卖系统需要实现一系列核心功能，如用户注册登录、菜品浏览与搜索、购物车管理、订单生成与支付、配送跟踪等。开发人员需要深入理解业务需求，将功能需求转化为具体的代码实现，并确保功能的准确性和稳定性。
3. **用户体验优化**：用户体验是外卖系统开发中不可忽视的因素。开发人员需要关注界面的美观性、操作的便捷性以及系统的响应速度。通过优化界面设计、提升交互体验以及减少加载时间等方式，提高用户的使用满意度。
4. **数据安全与隐私保护**：外卖系统涉及用户的个人信息和支付数据，因此数据安全与隐私保护至关重要。开发人员需要采取加密措施、访问控制等手段来保护数据的安全，并遵循相关法律法规和用户隐私政策，确保用户数据不被滥用或泄露。
5. **性能调优与监控**：外卖系统需要处理大量的并发请求和数据，因此性能调优是开发过程中的关键任务。开发人员需要关注系统的响应时间、吞吐量等性能指标，通过优化算法、数据库操作等方式提升系统性能。同时，建立监控机制，实时监控系统的运行状态，及时发现并解决潜在问题。
6. **测试与部署**：在开发过程中，测试是必不可少的环节。开发人员需要进行单元测试、集成测试和系统测试，确保代码的质量和系统的稳定性。完成测试后，还需要进行部署和上线工作，确保系统能够正常运行并为用户提供服务。

### 5.4.5环境

1. **包装垃圾的产生**：外卖订单通常伴随着大量的包装材料，如塑料袋、快餐盒、餐具和包装盒等。这些包装材料在消费高峰期间会大量产生，进而转化为大量的垃圾。这些包装垃圾如果处理不当，会对环境造成严重的污染。2.
2. **土壤和水体污染**：部分外卖包装材料，如塑料袋和餐盒，很难自然降解。如果填埋处理，会破坏土壤结构和微生物平衡，导致土壤功能退化。同时，这些包装材料中的有毒物质可能通过填埋或倾倒的方式深入地下水或直接污染河流、湖泊和海洋，对水体造成污染。
3. **大气污染**：在垃圾处理过程中，如果采用焚烧方式，外卖包装垃圾在燃烧时会释放出氨、硫化物等有毒气体，对大气造成污染。

### 5.4.6经费

1. 首先，开发外卖系统需要大量的初始投入。这包括人工成本、开发工具费用、开发设备费用以及软件许可证费用等。人工成本是其中的主要部分，涵盖了产品经理、设计师、研发工程师、测试工程师和运维工程师等多个岗位的薪资支出。此外，购买开发工具、开发设备以及获取软件许可证也是必不可少的支出。
2. 其次，外卖系统的运营和维护也需要持续的经费支持。这包括服务器的租赁和维护费用、带宽费用、数据库费用等。为了吸引用户并保持用户活跃度，还需要投入资金进行品牌推广和市场营销活动，如线上广告投放、线下活动推广等。
3. 此外，随着外卖业务的扩展和市场的竞争，外卖系统可能需要进行定期的升级和改造，以适应市场的变化和满足用户的需求。这同样需要额外的经费支持。
4. 综上所述，外卖系统对经费的要求较高，并且需要持续投入。因此，在规划外卖系统时，需要充分考虑经费问题，制定合理的预算和投入计划，以确保外卖系统的稳定运行和持续发展。同时，也需要通过合理的运营策略和市场推广手段，提高外卖系统的盈利能力和市场竞争力，从而实现经费的有效利用和回报。

### 5.5局限性

1. **商家与外卖平台的整合问题**：商家在使用外卖系统时，需要同时处理堂食和线上订单，这可能导致接单时长的延长和总接单率的下滑，尤其是在客流量大的时候。此外，由于平台众多，人工统计的步骤繁琐，容易出现外卖数据与门店系统对接不准确的情况，导致商家对成本、营销费用等方面的分析出现偏差。
2. **人力成本问题**：外卖订单量增加时，商家通常需要增派接单人手，这增加了人力成本。同时，每天需要查看各个平台的账单报表，再人工汇总核算营业额、平台扣款等数据，这也增加了人力负担。
3. **会员管理问题**：由于商家入驻的是其他外卖平台，顾客很难成为商家自己店铺的线上会员，这直接影响了商家与顾客之间的直接沟通互动，也不利于将线上消费引导至线下。
4. **技术安全与稳定性问题**：对于免费或低成本的外卖系统，可能存在安全隐患，如窃取用户信息、转移账户资金等。同时，系统的稳定性和便利性也可能受到影响，因为系统的维护和服务器都需要成本，如果系统无法持续更新和升级，可能会影响到其稳定性和功能完整性。
5. **依赖性问题**：外卖系统过于依赖线上平台和电子设备，一旦出现网络故障或设备问题，整个外卖业务可能受到影响，甚至停滞。

# 6经济可行性(成本----效益分析)

## 6.1投资

包括基本建设投资(如开发环境、设备、软件和资料等)，其他一次性和非一次性投资(如技术管理费、培训费、管理费、人员工资、奖金和差旅费等)。

1. **基本建设投资：**

（1）**平台开发和技术费用**：这包括软件开发、前后端技术支持和服务器租用等费用。

（2）**市场调研和定位费用**：在搭建外卖平台之前，进行市场调研和定位是非常重要的，以了解目标用户群、竞争对手和市场趋势，从而制定合适的战略。

（3）**品牌和推广费用**：为了吸引用户和合作伙伴，建立品牌知名度和推广平台是必要的，这包括线上线下的广告宣传、促销活动和市场营销等费用。

（4）**经营和运营费用**：外卖平台需要雇佣员工来处理订单、客户服务和运营管理等任务，这将涉及员工薪资和办公室租金等费用。

（5）**物流和配送费用**：外卖平台需要与供应商和餐厅建立合作关系，并有自己的物流和配送系统，这包括物流设备、人员和车辆等费用。

1. **其他一次性投资：**

（1）**硬件设备采购**：为支持外卖平台的运营，可能需要购买一些硬件设备，如用于处理订单的服务器等。这些设备的购置费用是一次性的，但可以为平台提供稳定的运行环境和高效的订单处理能力。

（2）**支付系统对接费用**：如果外卖平台需要接入第三方支付系统以提供便捷的支付服务，可能需要支付一定的对接费用。这包括与支付机构的技术对接、安全验证和费用结算等方面的费用。

（3）**专利和版权费用**：如果外卖平台使用了某些特定的技术或设计，并且这些技术或设计受到专利或版权的保护，那么可能需要支付相应的专利或版权费用。

（4）**法律咨询和合规费用**：为确保外卖平台的合法性和合规性，可能需要聘请专业律师进行法律咨询和合规审查，这也会产生一次性费用。

（5）**培训和培训材料费用**：在平台启动初期，可能需要对员工进行培训以确保他们熟悉平台操作和服务流程。这包括培训材料的制作和培训课程的实施等费用。

1. **非一次性投资：**

（1）**运营成本**：这包括员工薪资、办公室租金、水电费、日常办公用品等日常开支。为了维持平台的正常运营，需要定期支付这些费用。

（2）**系统维护和升级费用**：外卖系统需要定期进行维护和升级，以确保平台的稳定性和安全性。这包括修复潜在的安全漏洞、更新软件版本、优化系统性能等。这些工作通常需要专业的技术人员来完成，因此会产生相应的费用。

（3）**市场推广费用**：为了吸引更多的用户和合作伙伴，提升平台的知名度和影响力，需要投入一定的市场推广费用。这包括线上广告投放、线下宣传活动、合作推广等。市场推广费用可以根据平台的推广策略和预算进行调整。

（4）**客户服务与售后支持**：提供优质的客户服务和售后支持是保持用户满意度和忠诚度的关键。这包括处理用户咨询、解决用户问题、收集用户反馈等。为了提供高效的客户服务，需要投入相应的人力和物力资源。

（5）**数据安全与合规性投入**：随着数据保护法规的日益严格，外卖平台需要投入资金来确保用户数据的安全和合规性。这包括数据加密、数据备份、合规性审查等方面的投入。

## 6.2预期的经济效益

### 6.2.1一次性收益

1. **系统销售或授权费用**：如果外卖系统作为产品进行销售，或者将其使用权授权给其他企业使用，那么销售或授权费用可以被视为一次性收益。
2. **初始加盟费或合作费**：外卖系统可能采用加盟或合作模式进行扩张，加盟方或合作方在加入时需要支付一定的初始费用，这也是一种形式的一次性收益。
3. **广告收入**：在系统平台上投放的广告可能带来一次性的广告收入。

### 6.2.2非一次性收益

1. **订单处理费用**：每当用户在平台上完成一笔订单，平台都会按照一定比例收取订单处理费用或服务费。这是外卖系统最直接的收益来源之一，随着订单量的增加，收益也会相应增长。
2. **广告和推广收入**：平台可以在其界面上展示广告，向商家提供推广服务，从而获得广告费用和推广收入。这包括首页推荐、商家排名提升、精准推送等广告形式。
3. **会员制度和增值服务**：外卖系统可以通过推出会员制度，为会员提供专属优惠、快速配送等增值服务，并收取会员费或增值服务费。
4. **数据分析和营销服务**：外卖系统积累了大量的用户数据、订单数据等，可以对这些数据进行分析，为商家提供营销策略、用户画像等服务，进而获得相应的服务费用。
5. **合作伙伴分成**：外卖系统通常会与商家、物流公司、支付机构等合作伙伴进行合作，通过分成模式实现收益。例如，与商家合作时，可以按照订单金额的一定比例进行分成。

### 6.2.3不可定量的收益

1. **用户活跃度与留存率**：用户的活跃度和留存率直接影响了外卖系统的订单量和收益。当用户更频繁地使用平台并长期留存时，平台可以获得更多的订单处理费用、广告展示机会以及可能的增值服务收益。然而，用户的活跃度和留存率受到多种因素的影响，如平台的用户体验、服务质量、市场竞争等，因此这部分收益难以精确预测。
2. **商家合作与拓展**：外卖系统与商家的合作关系以及商家的拓展情况也会影响收益。当平台能够吸引更多优质商家入驻，并与商家建立稳固的合作关系时，可以获得更多的订单分成、广告推广机会等。然而，商家的合作意愿和拓展能力受到市场变化、竞争态势等多种因素的影响，因此这部分收益也是不可定量的。
3. **市场活动与推广效果**：外卖系统通常会进行各种市场活动和推广活动，以吸引新用户和提升用户活跃度。这些活动的效果直接影响到平台的订单量和收益。然而，市场活动的成功与否受到多种因素的影响，如活动的创意、执行力度、用户参与度等，因此其带来的收益也是不可定量的。
4. **技术创新与业务拓展**：外卖系统通过技术创新和业务拓展，可以开发新的收益来源。例如，引入智能推荐算法提升用户体验等多元化业务。然而，这些创新和拓展的效果难以预测，因此其带来的收益也是不可定量的。

### 6.2.4收益/投资比

外卖系统的收益与投资比涉及到多个变量，如平台的规模、功能、市场定位、竞争环境、运营策略等。这些因素都会影响外卖系统的投资成本以及潜在收益。

在投资方面，建立一个外卖平台需要考虑多个方面的成本，包括开发和设计网站或应用程序的费用、与餐馆签订合作协议的成本、建立物流系统的投入，以及市场推广和营销的费用。这些成本会根据平台的规模和功能的增加而变化。例如，一个功能更为复杂的平台可能需要更高的开发成本，而建立品牌知名度和用户基数更大的平台也需要更多的市场推广投入。

在收益方面，外卖系统可以通过多种方式获得收入，如商家抽佣、平台配送费、广告收入、会员制度收益、数据变现收益以及合作伙伴收益等。这些收益来源的具体表现也会受到市场竞争、用户需求、服务质量等多种因素的影响。

### 6.2.5投资回收周期

外卖系统的投资回报周期因多种因素而异，包括但不限于平台规模、市场竞争、运营策略以及投资规模等。

一般而言，外卖系统在初期需要进行大量的投资，包括平台开发、市场推广、运营维护等方面的费用。随着平台的逐渐成熟和用户基数的扩大，收益也会逐渐增加。然而，具体的投资回报周期取决于平台的运营情况和市场环境。在运营情况良好的情况下，外卖系统可能在较短的时间内实现投资回报。但如果市场竞争激烈，或者运营策略不当，投资回报周期可能会延长。

## 6.3市场预测

外卖系统未来发展趋势受到多重因素的影响，包括技术进步、消费者习惯、政策环境以及市场竞争等。

首先，从技术进步的角度来看，随着人工智能、大数据、物联网等技术的不断发展，外卖系统有望实现更高效、更精准的运营。例如，通过智能算法优化配送路线，降低配送成本；通过大数据分析消费者需求，提供更个性化的服务；通过物联网技术实现对餐饮生产流程的监控和管理，提高食品安全水平。

其次，消费者习惯也是影响外卖系统市场的重要因素。随着人们生活节奏的加快，越来越多的消费者选择通过外卖平台解决餐饮需求。同时，消费者对餐饮品质、服务体验的要求也在不断提高，这将推动外卖系统不断提升服务质量，满足消费者的多样化需求。

此外，政策环境也会对外卖系统市场产生影响。政府对于食品安全、环保等方面的监管力度不断加强，这将要求外卖系统加强自律，提高合规性。同时，政府对于数字经济、新兴产业的支持政策也将为外卖系统的发展提供有利条件。

最后，市场竞争也是不可忽视的因素。随着外卖市场的不断发展，越来越多的企业进入这个领域，竞争日益激烈。为了在市场中脱颖而出，外卖系统需要不断创新，提升核心竞争力。

综合以上因素，我们可以预测，未来外卖系统市场仍将保持快速增长的态势。但是，具体的增长速度和市场份额将受到技术进步、消费者习惯、政策环境以及市场竞争等多重因素的影响。

# 7技术可行性(技术风险评价)

本公司现有资源(如人员、环境、设备和技术条件等)能否满足此工程和项目实施要求，若不满足，应考虑补救措施(如需要分承包方参与、增加人员、投资和设备等)，涉及经济问题应进行投资、成本和效益可行性分析，最后确定此工程和项目是否具备技术可行性。

1.平台架构和功能需求：

- 外卖系统有一个响应迅速的用户界面，方便用户浏览菜单、下单和支付。

- 后端服务器应能处理高峰时段的订单量，数据库需具备高可靠性和性能。

2. 性能考虑：

- 系统应具备水平扩展能力，以应对用户量的突然增加。

- 考虑采用云服务，如AWS或Azure，来确保系统的稳定性和弹性

3. 数据安全和隐私：

- 外卖系统采取加密措施保护用户数据，包括个人信息和支付信息。

- 数据备份和恢复机制也是必要的，以应对意外数据丢失的情况。

4. 支付和订单管理：

- 集成安全的第三方支付系统，如PayPal或Stripe，保障用户支付信息的安全。

- 实时更新订单状态，确保订单信息的及时性和准确性。

# 8法律可行性

1. 数据隐私保护：

- 遵守当地数据隐私法规，如欧洲的GDPR，保护用户数据安全和隐私。

- 明确用户数据使用和存储政策，取得用户同意后才能使用其数据。

2. 支付安全合规：

- 遵守支付行业标准，确保用户支付信息的保密性和安全性。

- 考虑采用双重认证等安全措施来保护用户支付信息。

3.食品安全和卫生：

- 合作商家需符合当地食品安全和卫生标准，确保食品质量和安全。

- 外卖系统需建立食品安全检查机制，保障消费者健康。

4. 合同和责任规定：

- 明确用户和平台的责任与权利，建立用户协议和服务条款。

- 设立客户服务团队，处理用户投诉和退款事宜，遵守相关消费者权益法规。

# 9用户使用可行性

用户单位的行政管理和工作制度：

行政管理：

1. 确定用户单位对系统使用的政策和程序
2. 建立清晰的角色和职责，包括系统访问权限和数据管理责任
3. 提供持续的指导支持，以确保用户有效使用系统

工作制度：

1. 分析当前的工作流程，并确定系统如何融入和改善流程
2. 制定新的工作制度或修改现有制度，以适应系统的使用
3. 确保工作制度支持系统功能的充分利用

使用人员的素质和培训要求：

人员素质：

1. 拥有基本计算机素养并对相关软件熟悉
2. 对系统和工作流程有较好的理解
3. 良好的沟通和人际交往能力

培训要求：

1. 全面的系统培训，涵盖功能、流程和实践
2. 定期进行培训，以跟上系统的更新和新功能的应用
3. 根据用户角色和职责提供定制培训
4. 提供持续的指导，以解决用户问题和提高熟练度

# 10其他与项目有关的问题

技术进步：

1. 无人机送货：无人机技术的发展可能会使无人机送货成为一种可行的选择，扩大外卖系统的覆盖范围和效率。
2. 自动驾驶汽车：自动驾驶汽车的出现可以自动化送货流程，减少对人工司机的依赖。
3. 虚拟现实（VR）和增强现实（AR）：VR 和 AR 技术可以增强客户体验，例如允许他们虚拟查看菜单或跟踪订单状态。

市场趋势：

1. 健康饮食选择：对健康饮食选择的需求不断增长，外卖系统需要适应以提供更多健康的选择。
2. 可持续性：消费者对可持续发展的关注度越来越高，外卖系统需要探索减少包装浪费和追踪碳足迹的方法。
3. 个性化：客户期望个性化的体验，外卖系统需要提供个性化的菜单推荐和促销活动。

监管变化：

1. 劳动力法规：监管变化可能会影响送货司机的工资和工作条件，从而影响外卖系统的运营成本。
2. 食品安全法规：不断变化的食品安全法规可能会影响外卖系统的食品处理和送货程序。
3. 数据隐私法规：日益严格的数据隐私法规可能会影响外卖系统收集和使用客户数据的方式。

竞争格局：

1. 新进入者：新进入者的出现可能会加剧外卖市场的竞争，迫使现有系统创新和差异化。
2. 合并和收购：外卖行业可能出现合并和收购，导致市场格局发生变化。
3. 与其他行业的合作：外卖系统可能会与其他行业（例如杂货店或便利店）合作，提供更全面的服务。

分析这些变化对项目的影响：

1. 机会：技术进步和市场趋势为外卖系统项目创造了机会，例如通过无人机送货和个性化体验来提高效率和客户满意度。
2. 挑战：监管变化和竞争格局的变化可能会给项目带来挑战，例如遵守新的法规和应对新竞争对手。
3. 缓解措施：通过持续监控行业趋势、适应监管变化和投资于创新，可以缓解这些挑战并利用机会。

通过考虑这些未来变化，外卖系统项目团队可以主动规划和调整其策略，以确保项目在不断变化的环境中取得成功。

# 11注解

Java：是一种高性能、跨平台的面向对象编程语言，由Sun Microsystems公司（现在为Oracle Corporation）的James Gosling和Mike Sheridan于1991年开发。Java的设计宗旨是“Write Once, Run Anywhere”（一次编写，到处运行），意味着一次编写的Java程序可以在各种平台上运行，而无需重新编译。具有面向对象、跨平台性等特性。

SpringBoot： 快速构建Spring项目, 采用 "约定优于配置" 的思想, 简化Spring项目的配置开发。

Spring: 统一管理项目中的各种资源(bean), 在web开发的各层中都会用到。

SpringMVC：SpringMVC是spring框架的一个模块，springmvc和spring无需通过中间整合层进

行整合，可以无缝集成。

SpringSession: 主要解决在集群环境下的Session共享问题。

lombok：能以简单的注解形式来简化java代码，提高开发人员的开发效率。例如开发中经常需要写的

javabean，都需要花时间去添加相应的getter/setter，也许还要去写构造器、equals等方法。

Swagger： 可以自动的帮助开发人员生成接口文档，并对接口进行测试。

MySQL： 关系型数据库, 本项目的核心业务数据都会采用MySQL进行存储。

MybatisPlus： 本项目持久层将会使用MybatisPlus来简化开发, 基本的单表增删改查直接调用框架提供的方法即可。

Redis： 基于key-value格式存储的内存数据库, 访问速度快, 经常使用它做缓存(降低数据库访问、压力, 提供访问效率), 在后面的性能优化中会使用。

git: 版本控制工具, 在团队协作中, 使用该工具对项目中的代码进行管理。

maven: 项目构建工具。

junit：单元测试工具，开发人员功能实现完毕后，需要通过junit对功能进行单元测试。

# 附录

附录A

本外卖系统功能架构为

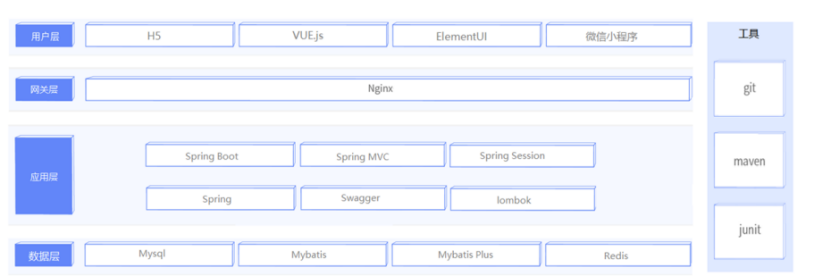


1). 移动端前台功能包括手机号登录 , 微信登录 , 收件人地址管理 , 用户历史订单查询 , 菜品规格查询 , 购物车功能, 下单 , 分类及菜品浏览。

2). 系统管理后台功能包括员工登录/退出 , 员工信息管理 , 分类管理 , 菜品管理 , 套餐管理 , 菜品口味管理 , 订单管理。

附录B

关于本项目的技术选型, 我们将会从 用户层、网关层、应用层、数据层 这几个方面进行介绍，而对于我们服务端开发工程师来说，在项目开发过程中，我们主要关注应用层及数据层技术的应用。



1). 用户层

本项目中在构建系统管理后台的前端页面，我们会用到H5、Vue.js、ElementUI等技术。而在构建移动端应用时，我们会使用到微信小程序。

2). 网关层

Nginx是一个服务器，主要用来作为Http服务器，部署静态资源，访问性能高。在Nginx中还有两个比较重要的作用： 反向代理和负载均衡， 在进行项目部署时，要实现Tomcat的负载均衡，就可以通过Nginx来实现。

3). 应用层

使用SpringBoot、 Spring、SpringMVC、SpringSession、lombok、javabean、Swagger等技术

4). 数据层

使用MySQL、MybatisPlus、Redis等技术

5).决定使用的CASE工具

使用git、maven、junit等

**Git**

是一个分布式版本控制软件，最初由林纳斯·托瓦兹创作，于2005年以GPL许可协议发布。最初目的是为了更好地管理Linux内核开发而设计。应注意的是，这与GNU Interactive Tools（一个类似Norton Commander界面的文件管理器）不同。

git最初的开发动力来自于BitKeeper和Monotone。git最初只是作为一个可以被其他前端（比如Cogito或Stgit）包装的后端而开发的，但后来git内核已经成熟到可以独立地用作版本控制。很多被广泛使用的软件项目都使用 git 进行版本控制，其中包括 Linux 内核、X.Org服务器和OLPC内核等项目的开发流程。

**Maven**

　Maven 是一个项目管理工具，它包含了一个项目对象模 型 (POM：Project Object Model)，一组标准集合，一个项目生命周期(Project Lifecycle)，一个依赖管 理系统(Dependency Management System)，和用来运行定义在生命周期阶段(phase)中插件(plugin)目标 (goal)的逻辑。

**Junit**

JUnit是一个Java语言的单元测试框架。它由Kent Beck和Erich Gamma建立，逐渐成为源于Kent Beck的sUnit的xUnit家族中最为成功的一个。 JUnit有它自己的JUnit扩展生态圈。多数Java的开发环境都已经集成了JUnit作为单元测试的工具。