

# 分析设计说明书模板

## XX分析设计说明书模板

- XX分析设计说明书模板
  - 文档历史
  - 摘要
    - 编写目的
    - 项目背景
    - 任务概述
    - 目标人员
    - 规范与约定
    - 术语与缩略语
    - 参考资料
  - 系统分析设计
    - 系统设计目标
      - 容量评估
    - 总体架构分析
      - 调用链路分析
    - 领域模型分析
      - 状态机
    - 用例分析
      - 业务用例图
      - 系统用例边界
    - 核心业务规则
      - 业务规则1
    - 外部依赖系统影响分析
      - 外部系统1
    - 部署架构（可选）
      - 网络拓扑图
      - 资源规划
      - 配置&配置规约
      - 部署目录结构
  - 中间件设计（可选）
    - 数据库
    - 缓存
    - 消息队列
    - 对象存储
  - 非功能性特性设计
    - DR（Disaster Recovery）兼容性
    - 可靠性
    - 可扩展性
    - 可运维
    - 安全性
    - 可测试性
    - 可维护性
    - 监控
  - 其他
    - 灰度发布分析
    - 发布回滚分析
    - 数据迁移分析

### 文档历史

修订日期	修订内容	修订版本	修订人
2022.6.1	非功能特性设计新增DR兼容性的分析	v22.6.1	Emin Xu

2022.6.16	上下游影响范围分析，增加  1. 目标人员增加依赖系统的相关人员，并增加方案评审通过后，终稿需同步给依赖系统相关人员的建议 2. 现有系统影响分析，增加对上下游系统影响的建议分析项目，增加对上下游分析项的分析结果，与对应系统PIC确认分析结果的建议	v22.6.16	Kairui Lu

摘要

编写目的

概述本文档编写目的，如：此分析设计说明书是对渠道介入提供技术设计方案，功能分配，模块划分，系统总体结构、运行设计以及出错设计等方面做了全面的概括性说明，为后期开发、测试人员进行编码、功能测试提供指导和帮助。

项目背景

概述本文档相关的项目背景，例如时间，项目起因，预期的业务结果等。

任务概述

概述本文档项目相关任务，如：完成物流费用对账，主要包含账单倒入，费用批次对账，差错处理等。

目标人员

指明文档阅读人员。包括架构师、产品经理、开发人员、测试人员、运维等。  
  
特别的，目标人员需指出本需求涉及的外部依赖系统的相关人员(需同步方案评审结论给相关人员)

规范与约定

概述本文档相关的规范与约定，如：  
  
package结构：如model包、manager包等；  
  
环境配置、其他常量约定等；

术语与缩略语

描述与本文档相关的业务或技术上的术语，如：

缩略语/术语	全称	说明
FM	First Mile	从发件端到中转仓之间的物流服务商类型

参考资料

列出参考资料名称以及出处。

系统分析设计

系统设计目标

描述系统设计的目标，系统分析设计将以此作为目标展开。

容量评估

容量评估、pv/uv、tps、预估数据量，并发等；特别是容量评估，需要重点分析

总体架构分析

描述项目涉及的系统以及应用的总体架构分析，比如新的应用架构，新的系统，全新的系统交互以及通信方式等，并且说明其设计的背景原因和必要性，如评估渠道流量压力，设计合适的系统架构。

## 调用链路分析

新增或改动现有接口调用时，需要重点分析调用场景和链路，及时发现性能隐患

## 领域模型分析

不同于数据库表结构，领域模型适用于业务层，是对数据库数据在业务上的加工，比如商品描述和商品基本信息是属于一个领域模型，但在数据库中是两个表，因为商品描述通常是`text`类型，数据量比较大，和主表在一张表会拖慢主表的查询。请使用`um`图中的类图来描述领域模型。

## 状态机

描述领域模型的状态转换和生命周期，以及状态转换的条件。从满足业务的角度出发，论证状态存在和转换的必要性。请使用`um`图中的`statemachine`图来进行描述。

## 用例分析

用例分析分别从业务和系统维度描述业务以及系统实现，方便项目成员以及团队其他成员一目了然的了解项目的整体业务，请使用`um`工具用例图来进行描述。

## 业务用例图

业务用例图提供全局业务视图，是对项目整体的任务分解。需求经过分析后，细化到每个任务都是可评估工作量的程度。每个用例都应该有相应的操作者(`actor`)。一个完整的业务用例图，能够让项目成员更容易的天然的可以用于项目任务分解。

## 系统用例边界

系统用例主要描述全局的业务用例在各个系统间的分布，以明确业务在系统间的流转，以及确认系统边界。

## 核心业务规则

核心业务规则用于描述项目涉及的业务流程以及约束，说明在技术实现上的设计细节，阐述设计的合理性。由于规则和模型之间本身存在业务以及技术上的关联性，有可能在领域模型中已经涉及到了核心业务规则的设计说明，但重复描述不会降低文档的可理解性。请使用`um`图中的流程图辅助描述。

## 业务规则1

## 外部依赖系统影响分析

列出本项目涉及的对现有外部的哪些系统、哪些业务的影响分析。需要涉及配合改造或者部署有变更的才是有影响，比如依赖的`package`升级或者`go`第三方库不兼容，对接的地方发需要外围系统配合改造，或者某个服务实现增加了判断，结果码有增加，需要外围系统配置文案，或者变更业务判断等。

需要配合修改的依赖系统的影响，需要对应依赖系统的PIC，确认影响范围评估结论的正确性

## 外部系统1

列出此项目对外部系统1的容量、QPS、响应时长、交互方式、依赖关系造成的影响，此部分内容需要对应外部依赖系统1的PIC确认

列出为支持此项目的发布上线，被依赖的系统配合修改，此部分修改造成的影响，此部分内容需要对应外部依赖系统1的PIC确认

## 部署架构（可选）

## 网络拓扑图

说明项目设计的应用系统、中间件、`db`以及网络设备的部署方案，并且描述其合理性。

## 资源规划

根据性能和容量目标描述资源计划和列表。

## 配置&配置规约

描述系统需要的配置，以及配置方面的约定，如无则忽略。

## 部署目录结构

明确应用部署目录结构，如无新增或者变更，可忽略。

## 中间件设计（可选）

### 数据库

注意此处不是表结构设计，而是对采用何种数据库、分库分表的选择、数据增长、事务使用及对应的高可用分析

### 缓存

### 消息队列

### 对象存储

## 非功能性特性设计

### DR (Disaster Recovery) 兼容性

描述系统在目前DR方案上适配程度，是否按买家市场做单元化，是否存在跨AZ调用，是否需要支持多活。

### 可靠性

描述系统在可靠性上的分析和设计，系统在容错性上是如何处理的，故障恢复、服务降级、熔断等的处理机制，以及在数据可靠性方面的考虑等。

### 可扩展性

可扩展性涵盖了系统的编译期和运行期的设计，运行期如何具备水平扩展能力来提升系统性能，编译期如何设计良好的代码结构来适应业务在未来可预见时间内的变化。这里需要描述系统在可扩展性上的设计思路。

### 可运维

描述在提升系统运维方面的分析及设计，好的系统应该是尽可能自动化的完成业务，自身应该具备相当能力的容错处理，不需要运维人员介入处理；同时也尽可能是标准化的，方便运维人员部署，或者提供可视化的运维操作。

### 安全性

配合安全要求，阐述系统安全方面（例如XSS，SQL注入，DDOS，数据安全等）的设计。

### 可测试性

描述系统各个业务在可测试性上的分析，发布应该满足灰度要求，系统应该具备在生产环境的可测试性。必须等到某个时间点、修改操作系统时间，或者直接不可测，都不应该发生。

### 可维护性

可维护性包含可扩展性在编译期的设计，可维护性的系统必然具备可扩展性，此外，可读性、命名规范、包路径约定等也是可维护性的考量。

### 监控

描述需要监控的业务功能点，浮动/绝对值等。

### 其他

描述其他关注点的分析，例如灰度发布，回滚方案，数据迁移方案等对系统设计的影响，系统设计如何处理才更好的支持。

## 灰度发布分析

这里需要描述的是系统如何设计才能满足灰度发布原则，而不是描述如何进行灰度方案。

## 发布回滚分析

这里需要描述的是系统如何设计才能更好的进行发布回滚，而不是描述回滚方案，设计合理可以确保回滚更易于处理。

## 数据迁移分析

和发布回滚类似，这里也是需要描述系统如何设计才能更好的进行数据迁移，而不是说明迁移方案。