

文档名称	修订记录 [创建 更新 归档]	备注	作者/记录人
设计文档模板	2018-03-26 创建		

1项目/子系统 概述

必要项：

- 解决了什么问题（是什么）
- 存在目的（为什么）
- 使用对象（外部调用者）

写几个段落来描述你的项目或者子系统做了什么它试图解决的问题是什么？为什么它需要存在？谁将会使用它？通过回答这些问题，你建立起设计的范围。如果你发现在这一节中要写出几个段落很困难，那你可能还没有将问题域理解到你需要的程度。如果你不能将你的描述在几个段落之内表达出来，那可能这个设计的范围太大了。将这一章节作为一个工具，来验证你的设计范围是合理的。

2概要设计：

- 功能模块说明（分别有有哪些模块）
- 总览图
- 模块概要说明（每个模块——文字说明）
 - 模块做什么
 - 使用对象
与之交互的对象
 - 使用的技术
用了什么语言，什么版本等等

3详细设计

每一个模块的具体设计

模块说明

（说明编写者对这个模块的理解，在设计模块之前先写自己的理解，保证和项目经理的要求是一致的）

模块框架图

使用对象

（一般是该模块的接口）在一个段落中描述对象是如何使用的，它服务于什么功能

实现方式：

- 流程图
内部数据流向方式
- 时间交互图
如果有复杂的状态需要以交互图来表现状态的切换，一个交互图会展示对象或实体集在完成一个复杂任务时如何相互交流，它解释了完成复杂任务时的交流顺序。在你想用图表展示你系统中的一个对象如何同其它子系统中的对象交流，交互图是最有用的。这一类图表可以让其它开发者核实交互是否正确
- 重要的类或者结构定义
重要的类或者结构体的设计
- 关键的技术点说明（特定的算法或者类关系模式）

以上不一定要全部写，看具体情况

- 设计的优缺点，风险
列举改设计的有点，缺点，以及使用场景，超过了使用场景会出现的潜在问题

4优点，风险/问题

一个列表列举所有已知的风险/问题，还有一个列表列举所有假定。其中一些可能只是重复你在之前章节中已经写过的。重要的是要将这些条目都放在一个章节中，这样如果读者想了解有哪些好处，风险和假定，就不需要去阅读整个文档。永远不要移除此章节中的任何内容！如果风险变成非风险，记录下它们现在不再是风险以及变化原因。不要简单地擦除它们。同样的原则也适用于假定。你应该能够一看本章节就立刻了解当前设计中的风险

结语

写一篇文档最困难的部分跟“写作”没什么关系。难的是你要在开始编码之前完成一个恰当合理的设计。一旦能真正理解问题的场景，要解决什么问题以及打算怎么做，以上都不模糊的情况下，书写细节是简单的,如果没理解清楚，自然就写不清楚，这个时候不是逃避写文档，而是应该要再花时间理解。另外，完成这一任务应该不需要编辑器和一个简单的绘图软件外的其它工具。在这一任务上花费时间所带来的积极效应，会在项目结束时带来难以想象的回报,文档即能检验你自己是否理解清楚，还能帮助别人理解清楚，不同人之间的认知差别不知道有多大。简单来说：“如果你在做计划上失败了，那么你就是在计划着去失败”。