

关于软件工程设计提交文件的说明和解释

1. 提交文件要求

首先按照自己的个人兴趣在题目列表选择一个题目或者自拟题目，然后根据软件工程开发流程，完成这个题目从需求分析到系统测试的各个阶段环节目标，并按照附件里面给出的各种文档格式，撰写相关文档。

请注意本课程提交的内容，应该包括软件工程中各个环节的文档以及源代码实现。如果仅提交一份软件工程报告或者一篇论文将做不及格处理!!!

2. 文档包中各个文件的解释及说明

按照软件工程的流程，文档包中包含以下几类文档：需求文档、体系结构设计文档、软件概要设计文档和系统测试文档。这与学过的软件开发模型是对于的。

2.1 需求文档的撰写

在软件工程课程中已经描述了需求阶段的目标，而其中需求文档是系统设计人员与业务人员交流的最好中间媒介。**需求文档撰写的主要目的是使得系统设计和开发人员能更好的理解系统特性、功能和业务流程等方面内容。**需求文档撰写过程中，要最大程度还原用户对系统的要求；使用精简的语言来撰写相关业务逻辑，尽量减少系统需求人员与用户和系统设计人员之间的理解歧义。

那么怎么才能撰写一份合格的需求分析文档呢？

如果你选择了一个题目。比如，***图书管理系统。在需求文档中主要包括两大方面内容。一方面是功能性描述；另一方面是非功能性描述。

在功能性描述中，首先要说明这个系统包含什么功能，比如图书查询功能，然后在每个功能使用**用例**的方法来进行业务逻辑实现的详细描述。比如针对图书关系系统里面的一个功能：登陆。你需要进行如下的描述。

登陆功能主要包括普通用户和系统管理员用户登陆。登陆页面支持用户进行身份类型选择（普通和管理员）。针对普通用户登陆，系统需要根据用户提交的用户名和密码，对其身份进行验证，具体业务流程见普通用户登陆用例。针对系统用户登陆，*****。

用例名称：普通用户登陆

用例执行的前置条件：登陆页面正常打开

用例流程描述：

1. 校验用户输入的用户名、密码是否合法（字符长度是否过长、是否有非法字符串等）。如合法将转入 2，不合法将在界面中进行提示。
2. 将用户名和密码与系统库中相应的字段进行比对，如一致，则转入 3；如不一致，则提示当前用户的身份与密码不一致的信息
3. 向系统日志中写入成功登陆的用户名及登陆时间
4. 系统跳转到针对普通用户功能主页上。

用例执行的后置条件：转入正常的页面或提示用户登陆不成功的信息。

在非功能性描述中要说明为实现系统的某些性能、安全性、可靠性等方面要求（具体在需求文档中有）。比如针对登陆功能，有以下性能、安全性方面的要求。

系统登陆能够支持 20000 用户同时在线；每个用户登陆时间不大于 0.3 秒系统有相应；用户的密码验证使用 MDS 加密方式进行；*****。

2.2 体系结构文档的撰写。

体系结构文档是架构师或系统设计人员根据需求文档，从以下几个方面来考虑如何实现系统（需要注意的是体系结构一般关注的粒度为系统模块）。包括逻辑视图、开发视图、进程视图、物理视图和场景视图等部分。这些内容如何撰写要求请查阅软件体系结构 4+1 视图模型的相关材料。

2.3 软件设计文档的撰写

软件设计文档主要关注软件模块的接口、关键数据结构和算法。在对软件模块设计时还需要考虑如果来实现需求文档中非功能性要求的内容。如果某个模块实现复杂，需要写入描述流程或算法的伪代码，并辅以相关说明来描述每个模块接口如何进行实现。

除此之外，在有些情况下，软件设计文档中还需求包括数据库表的逻辑和物理设计，以及界面的具体实现设计。

2.4 软件测试文档

根据需求文档，编写测试用例，以指导测试人员进行测试。并将测试结果和过程写入文档相应的部分。比如针对在需求文档中的普通用户登陆功能，需要设计以下测试用例。

测试用例名称：用户名参数合法性测试

测试内容： 1. 系统能否超长字符串的用户名做出正常反应
2. 系统是否能够检测包含非法字符的用户名

测试用例名称：密码参数合法性测试

测试内容： 1. 系统能否对超长密码字符串做出正常反应
2. ****

测试用例名称：登陆成功测试

测试内容： 1. 登陆成功后系统是否在日志上记录了成功登陆用户的信息
2. 系统是否能正常转到正确的页面中。
3*****

测试用例名称：登陆功能并发用户性能测试

测试内容： 1. 测试系统是否能够支持同时 10000 人登陆
2. 是否对每个用户的相应时间小于 0.3 秒钟。

测试方法： 1. 通过***自动测试工具，来仿真***用户同时进行用户登陆页面调用。并记载每个用户获得登陆页面的系统响应时间。
2. *****

3. 文档内容的逻辑完备体系

软件工程中这些文档包括实现代码是逻辑一致和完备的，也是相互可追溯和验证的。比如，需求文档中出现了登陆的功能用例和相关的非功能要求，那么在体系结构文档和软件设计文档中一定要有对其设计实现相关的内容，在测试文档中一定要求对其进行测试的考量，在代码中一定要有其对应的实现部分。这些是紧密相关的。如果需求发送变化，那么相应的文档内容也要作出相应的调整。

软件工程中的这些文档是参与软件设计过程中不同人员交流和合作的基础。只有完备的这些文档存在，才能保证软件设计的正确性、可追溯性。能够减少软件代码的维护成本。