**软件需求规格说明书**

[1. 文档概述 1](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261866)

[1.1目的 1](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261867)

[1.2范围 1](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261868)

[1.3 定义、首字母缩写词和缩略语 1](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261869)

[1.4参考资料 2](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261870)

[1.5 概述 2](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261871)

[2. 整体说明 2](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261872)

[2.1用例模型 2](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261873)

[2.2 假设与依赖关系 2](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261874)

[3. 具体需求 2](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261875)

[3.1用例描述 2](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261876)

[3.2补充需求 3](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261877)

[4.支持信息 3](file:///E:\GitHub\easyProject\软件需求规格说明书(RUP版).doc#_Toc307261878)

1. 文档概述

[该部分主要是对软件需求规格说明书文档进行基本的描述，包括该文档的目的、范围、术语定义、参考资料以及概要。]

[软件需求规格说明书用来系统、完整地记录系统的软件需求。该软件需求说明书的基础是用例分析技术。因此该文档中应包括用例模型、补充规约等内容。]

1.1目的

[在此小节中，主要对软件需求规格说明书的目的做一概要性说明，通常软件需求规格说明书应详细地说明应用程序、子系统的外部行为，还要说明非功能性需求、设计约束，以及其它的相关因素。]

本项目——Easyproject，用于有计划的控制软件开发流程，从立项开始至最终的提交任务，提供包括但不限于，管理人员、管理任务、管理里程碑、管理任务日历功能。

1.2范围

[系统是有范围的，而不是无限扩展的，对于无限扩展的需求是无法进行描述的。因此，在本小节应该对该说明书所涉及的项目范围进行清晰的界定。指定该规格说明书适用的软件应用程序、特性或者其它子系统分组、其相关的用例模型。当然在此也需要列出会受到该文档影响的其它文档。]

本项目适用于

1.3 定义、首字母缩写词和缩略语

[与其它文档一样，该文档也需要将本文档中所涉及的所有术语、缩略语进行详细的定义。还有一种可简明的做法，就是维护在一个项目词汇表中，这样就可以避免在每个文档中都重复很多内容。]

1.4参考资料

[在这一小节中，应完整地列出该文档引用的所有文档。对于每个引用的文档都应该给出标题、标识号、日期以及来源，为阅读者查找这些文档提供足够详细的信息。]

1.5 概述

[在本小节中，主要是说明软件需求规格说明书各个部分所包含的主要内容，就像一个文章摘要一样。同时也应该对文档的组织方式进行解释。]

2. 整体说明

[在本节中，将对整个软件需求进行总体性的描述，以期让读者对整个软件系统的需求有一个框架性的认识。也就是说，该节中主要包括影响产品及其需求的一般因素，而不列举 具体的需求。主要包括产品总体效果、产品功能、用户特征、约束、假设与依赖关系、需求子集等方面的内容。]

2.1用例模型

[在本小节中，将列出该软件需求的用例模型，该模型处于系统级，对系统的特性进行宏观的描述。在此应该列出所有的用例和Actor的名称列表，并且对其做出简要的说明，以及在图中的各种关系。]

2.2 假设与依赖关系

[在软件系统的开发过程中，存在许多假设和依赖关系。在本小节中应列举出所有的重要的技术可行性假设、子系统或构件可用性假设，以及一些可行性的假设。]

3. 具体需求

[如果说第二章节是框架，那么本节就是血肉。在本节中，应该详细列出所有的软件需求，其详细程序应使设计人员能够充分理解并且进行设计的要求，同时也应该给予测试人员足够的信息，以帮助他们来验证系统是否满足了这些需求。整个需求的组织可以采用用例描述进行。]

3.1用例描述

[如果你使用用例建模技术，那么你已经通过用例定义了系统的大部分功能性需求和一些非功能性需求。因此，在软件需求规格说明书只需将这些具体的用例描述，整理在一起，全部放在该小节之中。当然也可以将用例描述做为附件，在此列出引用，只是这样做并不利于阅读。建议在组织形式上采用以“软件需求”为线索，在每个需求中，填入对应的1个或几个用例描述。]

3.2补充需求

[由于用例毕竟主要针对功能性需求，因此还会有一些其它的补充需求遗漏，因此在本小节中就是将这些东西补充出来。这些补充需求大部分集中在非功能需求之上，包括以下几个方面的内容：]

1） 易用性：例如指出普通用户和高级用户要高效地执行某个特定操作所需的培训时间；指出典型任务的可评测任务次数；或者指出需要满足的可用性标准（如IBM的CUA标准、Microsoft的GUI标准。

2） 可靠性：包括系统可用性（可用时间百分比、使用小时数、维护访问权、降纸模式操作等）；平均故障间隔时间（MTBF，通常表示为小时数，但也可表示为天数、月数或年数）；平均修复时间（MTTR，系统在发生故障后可以暂停运行的时间）；精确度（指出系统输出要求具备的精密度、分辨率和精确度）；最高错误或缺陷率（通常表示为bugs/KLOC，即每千行代码的错误数目或 bugs/function-point，即每个功能点的错误数目）；错误或缺陷率（按照小错误、大错误和严重错误来分类：需求中必须对“严重”错误进行界定，例如：数据完全丢失或完全不能使用系统的某部分功能）。

3） 性能：包括对事务的响应时间（平均、最长）；吞吐量（例如每秒处理的事务数）；容量（例如系统可以容纳的客户或事务数）；降级模式（当系统以某种形式降级时可接受的运行模式）；资源利用情况：内存、磁盘、通信等。

4） 其它：包括用户界面要求、联机帮助系统要求、法律许可、外购构件，以及操作系统、开发工具、数据库系统等设计约束。

4.支持信息

[支持信息用于使软件需求规格说明书更易于使用。它包括：目录、索引、附录等。]