**Tally记账软件项目设计规格说明书评审报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **项目名称** | | | Tally记账软件项目 | |
| **检查人** | | | 金童 | |
| **检查日期** | | | 2019/11/01 | |
| **#** | **检查项** | | **是**/否/不适用 | |
| 清晰性、完整性 | | | | |
| 1 | 是否清晰的描述了单元设计信息，包括数据流程、控制流程、接口？ | | 是 | |
| 2 | 文档结构是否清晰、组织是否合理？ | | 是 | |
| 3 | 文档结构是否便于维护和修改？ | | 是 | |
| 4 | 设计是否易于理解？ | | 是 | |
| 5 | 每个单元模块是否都有相应的标识? | | 是 | |
| 6 | 是否对单元模块的目的和功能进行了描述？ | | 是 | |
| 7 | 每个单元模块的输入/输出是否进行了描述? | | 是 | |
| 9 | 是否提供了一致的错误处理机制? | | 否 | |
| 10 | 系统结构是否合理、清晰？ | | 是 | |
| 11 | 各子系统、模块之间的关系是否描述得清楚? | | 是 | |
| 12 | 系统的设计是否考虑了系统的可扩展性? | | 未涉及 | |
| 13 | 设计是否使用标准的技术，避免使用怪异的、不易理解的方式和方法? | | 是 | |
| 14 | 是否列出了所有的调用？ | | 是 | |
| 15 | 对变量、指针和常量进行了定义和初始化吗？ | | 否 | |
| 16 | 设计能实现特定的需求和目标吗？ | | 是 | |
| 17 | 是否对程序的注释进行了设计？ | | 否 | |
| 18 | 是否对程序的限制和约束进行了说明？ | | 是 | |
| 19 | 所有设计是否是可测试的？ | | 是 | |
| 一致性、正确性 | | | | |
| 20 | 文档是否符合项目标准？ | | 是 | |
| 21 | 是否用要求的方法或工具进行设计的？ | | 是 | |
| 22 | 数据元素的名称在整个单元中保持一致吗？ | | 是 | |
| 23 | 所有的设计接口相互间是一致的吗？ | | 是 | |
| 24 | 是否存在逻辑上的问题？ | | 是 | |
| 25 | 是否对各种情况都进行了处理？（如大于、等于、小于0，switch/case情况） | | 是 | |
| 26 | 是否为开发和维护代码提供了充分的基础？ | | 是 | |
| 27 | 所有的设计单元都可追溯回需求吗？ | | 是 | |
| 接口 | | | | |
| 28 | 参数的数量、类型和顺序是否匹配？ | | 是 | |
| 29 | 是否正确的定义了输入输出数据？ | | 否 | |
| 30 | 是否清晰的描述了传递参数的顺序？ | | 否 | |
| 31 | 是否识别了传递参数的机制？ | | 否 | |
| 可维护性、可靠性 | | | | |
| 32 | 设计单元是否具有高内聚度低耦合度？（即该单元的变化不会对本单元造成不可预料的影响，对其他单元的影响达到最小） | | 未涉及 | |
| 33 | 设计的复杂度已经最小了吗？ | | 未涉及 | |
| 34 | 是否具有清晰性、可读性、可修改性，满足维护需求？ | | 未涉及 | |
| 35 | 是否对输入、输出、接口和结果进行了错误检查？ | | 否 | |
| 36 | 是否对错误情况给出了有意义的信息提示？ | | 否 | |
| 37 | 是否考虑了意外情况？ | | 否 | |
| 38 | 是否符合相关的法律法规 | | 是 | |

**评审说明**

# 概述

本次评审对象为《Tally软件设计规格说明书V2.0》

在评审报告中，主要针对上述检查项进行了评审。内容包括清晰性、完整性、一致性、正确性、接口、可维护性和可靠性，以及在《需求规格说明书》中表达不明确的地方。

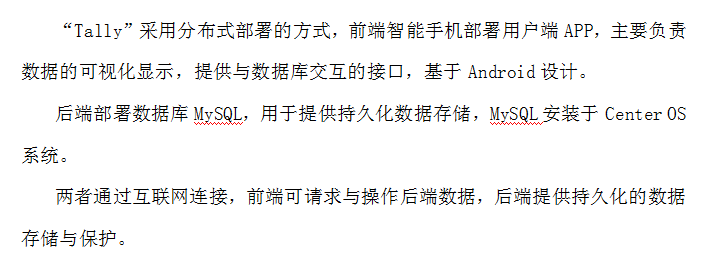
# 评审意见

本次《软件设计规格说明书》对于运行环境基本已经明确，并且考虑到了数据的类型及长度。系统的安全性、私密性等需求也有所概述。对软件的界面描述清晰。以下我只针对《说明书》中一些可能存在歧义和问题的地方进行说明。

▽清晰性、完整性：

如表格中第9、15和17条所述，《软件设计规格书》中未提及对错误处理机制，包括但不仅限于出现404、400、500等常规错误。未对变量、指针和常量进行了定义和初始化，甚至未提及指针方面。未对程序的注释进行了设计，可能会导致修改代码时出现理解性问题。

▽一致性、正确性：

如表格中第24条所述，《软件设计规格书》中存在逻辑上的问题，在文档中“3.6 部署设计”的最后，如下图所示：

不应是由前端负责提供与数据库交互的接口，而是应由后端提供。与编码逻辑及讨论结果相违背。

▽接口：

如表格中第29、30及31条所述，《软件设计规格说明书》中未对接口有详细的设计，包括定义输入输出数据、描述传递参数的顺序和识别传递参数的机制。

▽可维护性和可靠性：

如表格中第35、36及37条所述，《软件设计规格说明书》中未对输入、输出、接口和结果进行了错误检查，未对错误情况给出了有意义的信息提示，未考虑意外情况。包括但不仅限于具体给出用户输入错误的用户名、密码会出现怎样的提示，用户未联网情况记账会给出怎样的提示等。