**武汉大学计算机学院**

**本科生课程报告**

**软件工程小组实践项目——需求规格评审表**

专 业 名 称 ：计算机科学与技术

课 程 名 称 ：软件工程

指 导 教 师 ：伍春香 副教授

学 生 学 号 ：2020302131035

2020300004052

2020302021170

　 　　2020302111017

　　　　　 　　2020302111300

学 生 姓 名 ：施嘉乐

　 　李哲

　滕天琦

　 　吴彬

　张怡宁

二○二三年四月

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 需求规格评审表 | | | | | | |
| 序号 | 需求特性 | 检查内容 | 合格 | 不合格 | 不适用 | 具体描述 |
| 1 | 规范性 | 数据流图-不同部分所用符号是否正确 |  | √ |  | 数据流应统一使用实线 |
| 2 | 数据流图-外部实体之间是否存在数据流 | √ |  |  | 外部实体间不存在数据流 |
| 3 | 数据流图-是否对每个数据流命名和解释 |  | √ |  | 拆分-修改内容，拆分-调整编辑区域，序列化与反序列化-文档结构树，调整编辑区域-当前编辑区域，文档结构树-转化为渲染树没有对数据流做说明 |
| 4 | 数据流图-是否简洁清晰，包含所有必要信息而不包含无关信息 | √ |  |  | 简洁清晰，包含必要信息 |
| 5 | ER图-实体和属性的命名应该遵循一致的命名规范 | √ |  |  | 遵循一致命名规范 |
| 6 | ER图-实体之间的关系应该清晰明确，避免冗余或者遗漏 |  | √ |  | 涉及实体和属性过于简单，未处理和文件创建人等之间的关系 |
| 7 | ER图-所有实体都应该有唯一标识符，以保证数据的唯一性 | √ |  |  | 有唯一标识 |
| 8 | ER图-实体和属性的定义应该具有可扩展性，能够满足未来可能出现的需求 | √ |  |  | 具有可扩展性 |
| 9 | ER图-是否避免了数据的冗余存储，是否符合最佳实践 | √ |  |  | 避免了数据冗余 |
| 10 | 状态图刻画的对象实体是否与UML类建模是否保持一致性 |  |  | √ | 未完成面向对象状态图 |
| 11 | 状态图初始状态、终止状态、中间状态、迁移等符号表示是否准确 |  | √ |  | 出现箭头指向箭头并不合法 |
| 12 | 状态图是否简洁清晰，包含所有必要信息而不包含无关信息 | √ |  |  | 简洁清晰，包含必要信息 |
| 13 | 顺序图-对象、生命线、激活符号等符号和标记是否符合规范 |  |  | √ | 未包含顺序图 |
| 14 | 顺序图-是否存在任何错误的UML符号和标记使用 |  |  | √ | 未包含顺序图 |
| 15 | 顺序图-是否使用了相同的符号来表示相同的对象或消息 |  |  | √ | 未包含顺序图 |
| 16 | 顺序图-是否按照良好的视觉设计和排版规则进行排版 |  |  | √ | 未包含顺序图 |
| 17 | 顺序图-顺序刻画的对象实体是否与UML类建模是否保持一致性 |  |  | √ | 未包含顺序图 |
| 18 | 类图-类名、属性名、方法名等命名是否规范，易于理解 | √ |  |  | 符合规范，易于理解 |
| 19 | 类图-类之间的关系是否清晰明了，是否出现多余的关系 | √ |  |  | 清晰明了，没有多于关系 |
| 20 | 类图-类之间的关系是否符合实际情况，是否出现不合理的关系。 | √ |  |  | 符合实际情况，无不合理关系 |
| 21 | 类图-UML类图中的关系箭头和标识是否符合规范 | √ |  |  | 符合规范 |
| 22 | 类图-类之间的关系是否易于维护和修改 | √ |  |  | 易于维护和修改 |
| 23 | 正确性 | 数据流图-是否确定了系统的输入输出 | √ |  |  | 确定了系统的输入和输出 |
| 24 | 数据流图-是否准确刻画了数据的流向、系统的工作方式 | √ |  |  | 准确刻画了数据流向和系统工作方式 |
| 25 | 数据流图-父图与子图（如果有多层）的输入输出数据流是否一致 | √ |  |  | 一致 |
| 26 | 数据流图-每个数据加工所需数据是否都能从其输入数据流中直接获得 | √ |  |  | 能从其输入数据流中直接获得 |
| 27 | 数据流图-每个加工是否既有输入数据流又有输出数据流 | √ |  |  | 既有输入数据流又有输出数据流 |
| 28 | 数据流图-每个加工是否都对数据进行了处理（而非完全不变） | √ |  |  | 都对数据进行了处理 |
| 29 | ER图-是否包括所有必要的实体、关系和属性，是否存在缺失 |  | √ |  | 不包含，存在缺失 |
| 30 | ER图-实体和关系是否完整，是否遗漏任何重要的信息 |  | √ |  | 不完整，有遗漏 |
| 31 | ER图-实体和关系之间的连接是否正确，是否存在不合逻辑的连接 | √ |  |  | 正确 |
| 32 | ER图-实体和关系是否符合数据模型的要求，例如，是否存在环路、孤立实体等 | √ |  |  | 符合模型要求 |
| 33 | ER图-实体和关系是否能够正确地映射到数据库表和列 | √ |  |  | 能正确映射 |
| 34 | ER图-是否存在错误、歧义或不一致的部分 | √ |  |  | 不存在错误、歧义或不一致 |
| 35 | ER图-是否被正确地文档化和记录，是否有必要的注释和说明 |  | √ |  | 没有必要的注释 |
| 36 | 状态图-是否正确刻画了对象实体的各种状态，是否考虑全面且不存在冗余状态 |  | √ |  | 没有明确地用文字说明状态图刻画的是什么对象或过程的状态 |
| 37 | 状态图-各个状态间的跃迁条件是否准确 | √ |  |  | 准确 |
| 38 | 状态图-是否正确区分事件、活动、动作的差异性，并在状态图中用正确的方式表示 | √ |  |  | 正确区分事件、活动、动作的差异性，并在状态图中用正确的方式表示 |
| 39 | 状态图-是否准确刻画各个状态间的“与\或”关系，是否正确处理好状态的并行与串行关系 | √ |  |  | 正确刻画、正确处理 |
| 40 | 顺序图-是否准确刻画了对象的生命周期 |  |  | √ | 未包含顺序图 |
| 41 | 顺序图-是否准确刻画了消息的交互过程 |  |  | √ | 未包含顺序图 |
| 42 | 顺序图-是否考虑了消息的时间顺序 |  |  | √ | 未包含顺序图 |
| 43 | 顺序图-是否定义了正确的输入输出 |  |  | √ | 未包含顺序图 |
| 44 | 顺序图-是否存在任何重复定义的对象或消息 |  |  | √ | 未包含顺序图 |
| 45 | 顺序图-是否有任何未定义的对象或消息 |  |  | √ | 未包含顺序图 |
| 46 | 顺序图-是否考虑了所有的业务需求，不存在遗漏 |  |  | √ | 未包含顺序图 |
| 47 | 类图-是否准确刻画了整个系统的结构和组成 | √ |  |  | 准确刻画 |
| 48 | 类图-是否准确刻画了各类之间的关系 | √ |  |  | 准确刻画 |
| 49 | 类图-是否准确全面地定义了各类中的属性和方法 | √ |  |  | 准确全面 |
| 50 | 类图-是否准确定义了全部用到的接口 |  | √ |  | 准确定义 |
| 51 | 类图-是否存在任何重复定义的属性，方法或关系 | √ |  |  | 不存在重复定义 |
| 52 | 类图-是否准确描述了各属性的类型和方法的功能 | √ |  |  | 准确描述 |
| 53 | 类图-是否考虑了所有的业务需求，不存在遗漏 | √ |  |  | 考虑了所有业务需求 |
| 54 | 一致性 | 各个需求之间是否一致是否有冲突和矛盾 | √ |  |  | 一致，没有冲突 |
| 55 | 所规定的模型、算法和数值方法是否相容 | √ |  |  | 相容 |
| 56 | 是否使用了标准的术语和定义形式 | √ |  |  | 使用了标准的术语和定义形式 |
| 57 | 需求是否与其软硬件操作环境相容 | √ |  |  | 相容 |
| 58 | 是否说明了软件对其系统和环境的影响 | √ |  |  | 说明了软件对其系统和环境的影响 |
| 59 | 是否说明了环境对软件的影响 | √ |  |  | 说明了环境对软件的影响 |
| 60 | 所采用的技术是否与用户要求的技术一致 | √ |  |  | 与用户要求的技术一致 |
| 61 | 是否确保了需求与其他文档的一致性 | √ |  |  | 确保了需求与其他文档的一致性 |
| 62 | 易跟踪性 | 是否规定了所有必要且唯一的的需求编号和版本控制 |  | √ |  | 未规定唯一的需求编号和版本控制 |
| 63 | 是否确保了每个需求都可以追溯到其来源 | √ |  |  | 确保了每个需求都可以追溯到其来源 |
| 64 | 是否可从上一阶段的文档中找到需求定义中的相应内容 | √ |  |  | 可从上一阶段的文档中找到需求定义中的相应内容 |
| 65 | 需求定义是否明确地表明前阶段中提出的有关需求和设计限制是否都已被覆盖 | √ |  |  | 明确地表明前阶段中提出的有关需求和设计限制都已被覆盖 |
| 66 | 需求定义是否便于向后继开发阶段查找信息 | √ |  |  | 便于向后继开发阶段查找信息 |
| 67 | 是否规定了所有必要的需求跟踪和变更管理 |  | √ |  | 未规定需求跟踪和变更管理 |
| 68 | 文档完整性 | 是否包含所有必要的需求内容 |  |  | √ | 未完成面向对象方法动态建模 |
| 69 | 是否完整准确地描述了所有相关的功能和系统行为 | √ |  |  | 完整准确 |
| 70 | 是否描述了与安全、性能、稳定性和可维护性等相关的非功能需求 | √ |  |  | 描述了与安全、性能、稳定性和可维护性等相关的非功能需求 |
| 71 | 是否定义了与系统交互相关的所有必要接口和协议 |  | √ |  | 未定义明确接口 |
| 72 | 是否包括了项目的范围和限制 | √ |  |  | 包括了项目的范围和限制 |
| 73 | 是否包括了质量要求和标准 | √ |  |  | 包括了质量要求和标准 |
| 74 | 是否包括了必要的参考文献和附录 |  | √ |  | 没有定义文档的参考文献和附录 |
| 75 | 是否存在明显的错误或者矛盾，需要进行修改或者进一步确认 | √ |  |  | 没有明显的错误 |
| 76 | 是否涵盖了所有的必要的附加信息，例如用例、用户界面等 | √ |  |  | 涵盖了所有的必要的附加信息 |
| 77 | 文档是否符合公司或行业的标准模板或格式要求 | √ |  |  | 符合公司或行业的标准模板或格式要求 |
| 78 | 文档可读性 | 文档中需求的描述是否清晰明了，易于理解 | √ |  |  | 清晰明了，易于理解 |
| 79 | 文档中的缩写、术语是否清晰明了，并且在文档的一开始进行了定义 | √ |  |  | 清晰明了 |
| 80 | 文档中的图表、表格是否清晰易懂，能够清晰地呈现需求内容 | √ |  |  | 清晰易懂 |
| 81 | 文档中是否存在歧义或者模糊的表述，需要进行进一步澄清 |  | √ |  | “功能简约易用”说法笼统，有点模糊 |
| 82 | 是否具有清晰的逻辑结构和章节划分 | √ |  |  | 具有清晰的逻辑结构和章节划分 |
| 83 | 是否使用了易于阅读和识别的字体、颜色和排版 | √ |  |  | 使用了易于阅读和识别的字体、颜色和排版 |
| 84 | 是否包括了易于查找的索引和导航 | √ |  |  | 包括了易于查找的索引和导航 |
| 85 | 是否包括了足够的摘要或概述，以帮助读者快速理解文档内容 |  | √ |  | 没有摘要和概述 |
| 86 | 是否包括了足够的注释和说明，以帮助读者更好地理解和使用文档 | √ |  |  | 包括了足够的注释和说明 |
| 87 | 可靠性 | 是否为每个需求指定了软件失效的结果 | √ |  |  | 为每个需求指定了软件失效的结果 |
| 88 | 是否指定了特定失效的保护信息 | √ |  |  | 指定了特定失效的保护信息 |
| 89 | 是否指定了特定的错误检测策略 | √ |  |  | 指定了特定的错误检测策略 |
| 90 | 是否指定了错误纠正策略 | √ |  |  | 指定了错误纠正策略 |
| 91 | 其他 | 是否考虑了人机交互等人性化需求 | √ |  |  | 考虑了人机交互等人性化需求 |
| 92 | 是否考虑了性能、安全性、可靠性等非功能需求 | √ |  |  | 考虑了性能、安全性、可靠性等非功能需求 |
| 93 | 是否考虑了未来的扩展和修改需求 |  |  |  | 考虑了未来的扩展和修改需求 |
| 94 | 是否考虑了国际化、本地化等特殊需求 |  |  |  | 考虑了国际化、本地化等特殊需求 |