**检查单：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 分类 | 检查点 | 检查判断 | 检查内容 |
| 1 | 可追溯性 | 软件设计是否覆盖所有已确定的软件需求 | 否 | 管理员模块未完成、缺陷预测模块中的结果导出功能未实现。 |
| 2 | 软件每一成分是否可追溯到某一项需求 | 是 | 项目完成的每一成分均来自于需求文件中的登录、注册、缺陷验证、用户中心、模型管理、数据库管理、用户管理模块。 |
| 3 | 接口 | 分析软件各部分之间的联系，确认软件的内部接口与外部接口是否已经明确定义 | 是 | 用规范的命名法命名前后端交互的url地址。将每一个接口作用明确定义并且规范注释。 |
| 4 | 模块是否满足高内聚低耦合 | 否 | 不满足低耦合，用户管理模块以及用户中心模块功能重复。 |
| 5 | 模块的作用范围是否在其控制范围内 | 否 | 用户中心模块的部分功能并不受用户中心模块控制，而是受用户管理模块控制。 |
| 6 | 风险 | 确定软件设计是否在现有技术条件和预算范围内能按时实现 | 否 | 未在预算范围内实现软件的需求。 |
| 7 | 可维护性 | 从软件维护的角度出发，确认软件设计是否考虑方便未来的维护 | 否 | 没有过多考虑项目未来的维护。 |
| 8 | 一致性 | 软件功能是否与软件需求规格说明书保持一致，并完整体现 | 否 | 管理员模块未完成、缺陷预测模块中的结果导出功能未实现。 |
| 9 | 正确性 | 流程逻辑是否正确、合理 | 否 | 算法的调用被分解，部分位于算法内部，部分位于调用算法的service层，逻辑不清。 |
| 10 | 算法是否合适、有效 | 否 | 算法准确率最高86%，未能超过90%，未能达到理想情况。 |
| 11 | 设计是否达到性能最佳 | 否 | 否，项目设计未考虑性能问题。 |
| 12 | 设计中是否考虑处理故障和避免失效 | 否 | 设计中没有考虑处理故障和避免失效。 |
| 13 | 用户界面设计是否正确反应功能实现 | 是 | 用户界面功能含义简单易懂，正确反应功能实现。 |
| 14 | 易理解性 | 模块结构是否良好、清晰，易于理解 | 是 | 登录、注册、缺陷验证、用户中心、模型管理、数据库管理、用户管理功能模块功能简单易懂，模块结构清晰，易于理解。 |
| 15 | 用户界面是否简洁、明了，具有功能指导性 | 是 | 用户界面简洁、明了，没有多余冗杂功能，能够清楚地指导每一个功能。 |
| 16 | 复用性 | 设计是否可复用 | 是 | 登录、注册等基础功能具有很强的复用性，可以在其他项目中进行快速地服用。 |
| 17 | 是否复用其他项目的部件 | 是 | 邮箱注册功能复用自其他项目。 |