Project Phase 1

M. Saad Khan (22-11006)

Mir Hassan Siddiqui (22-11007)

The Checklists that we have chosen are Design Inspection Checklist and Inspection Check List

Design Checklist

|  |  |  |
| --- | --- | --- |
| Items to be inspected | Pass/Fail | Comments |
| 1. Is the design consistent with the requirements | Yes | The Design is simple but it meets our requirements |
| 1. Are deviations from the requirements documented and approved? | Partial | There are no deviations |
| 1. Are all assumptions documented? | Yes | The use cases mention all assumptions |
| 1. Have major design decisions been documented? | No | There were no design decisions documented. |
| 1. Is the design consistent with these decisions? | No |  |
| 1. Does the design adequately address the following?   real-time requirements  performance issues (memory and timing)  spare capacity (CPU and memory)  maintainability  understandability  database requirements  loading and initialization  error handling and recovery  user interface issues  software upgrades | No | It’s a simple design which is very basic. It does not have these features. |
| 1. Is the P-spec for each process accurate and complete? | Yes | The performance specs are accurate |
| 1. Are dependencies on other functions, Operating system kernel, hardware, etc., identified and documented | No | Simple program that does simple calculations |
| 1. Are human factors considerations properly addressed in those functions that provide the user interface? | No | They are not |
| 1. Are design constraints, such as memory and timing budgets, specified where appropriate? | No | There are no such design constraints. As said it is a simple program that does basic match calculations |
| 1. Are requirements for error checking, error handling and recovery specified where needed? | No | There are supposed to be no errors. It uses simple add function |
| 1. Are interfaces consistent with module usage? Missing interfaces? Extra interfaces? | Yes | The interface is consistent. No missing or Extra Interface |
| 1. Are the interfaces specified to a sufficient level of detail that allows them to be verified? | Yes | Yes, it is. |
| 1. Does the design follow the established notation conventions? | Yes | Yes, it does. |
| 1. Does the detailed design of this module or interface fulfill its part of the requirements? | Yes |  |
| 1. Has the inspection of this module or interface identified problems in the SRS? For example, missing requirements, ambiguous requirements, conflicting requirements, untestable requirements, implied requirements? | No |  |
| 1. Does the detailed design of this module or interface meet its high level design requirements? | Yes |  |
| 1. Has the inspection of the detailed design identified problems in the high level design? | No | There were no problems identified. |
| 1. Are all functions completely and accurately described in sufficient detail? | Yes | The documentation includes all the details. |
| 1. Are all interfaces completely and accurately described, including keyword or positional parameters, field descriptors, attributes, ranges, and limits? | Yes | The documentation includes all details |
| 1. At a system and subsystem level, have all components or modules been identified on a System Architecture Model? | Yes |  |
| 1. Is the level of decomposition sufficient to identify all modules? | Yes |  |
| 1. Will further decomposition result in identifying more modules? | Yes | Yes, Further decomposition might lead to more modules |
| 1. Are modules performing more than one specific function? | No | Each module/function performs a specific function |
| 1. Are there logic errors? | No | There are no errors |
| 1. Are...   all unique values tested?  all positional values tested?  increment and loop counters properly initialized? variables and data areas initialized before use? | Yes | Everything mentioned is tested |
| 1. Has the module been inspected for...   correct begin and end of table processing?  correct processing of queues across interrupts? correct decision table logic?  correct precision/accuracy of calculations? | Yes |  |
| 1. Are message priorities allocated properly to ensure the correct execution of code? | No | There are no message priorities |
| 1. . Is the message processing sequence correct? | No |  |
| 1. Are there errors in handling data, data buffers, or tables, incorrect field updated, conflicting use of data areas, incomplete initialization or update, inconsistent or invalid data attributes? | Partial | There might be errors in tables if a lot of items are used but the chances of that are very low |
| 1. Are memory and timing budgets reasonable and achievable? | Yes |  |
| 1. Is there adequate error condition testing? | Yes |  |
| 1. Are return codes documented? | Yes | All errors are documented in the document |
| 1. Is the design...   understandable (i.e., easy to read, follow logic)?  maintainable (i.e., no obscure logic...)?  testable (can be tested with a reasonable number of tests? | Yes | It’s a simple and understandable design |

Inspection Checklist

|  |  |  |  |
| --- | --- | --- | --- |
| Items to be inspected | Pass/Fail | Comments | |
| 1. Are all sources of input identified | Yes | |  | |
| 1. Is there a need for robustness across the entire input space? | No | |  | |
| 1. Are there any timing constraints on the inputs? | No | |  | |
| 1. Are all types of outputs identified | Yes | |  | |
| 1. \*Are there any timing constraints on the outputs? | No | |  | |
| 1. Is the user interface clearly defined? | Yes | |  | |
| 1. Are assumptions clearly identified | Yes | |  | |
| 1. Does SRS include all user requirements? | Yes | |  | |
| 1. Do the functional requirements cover all abnormal situations? | Yes | |  | |
| 1. Does SRS define those requirements for which future changes are anticipated? | No | |  | |
| 1. Have the temporal aspects of all functions been considered? | No | |  | |
| 1. Are the environmental conditions specified for all operating modes (e.g., normal, abnormal, disturbed)? | Yes | |  | |
| 1. Is there any internal inconsistency between the software requirements | No | |  | |
| 1. Has the impact of the environment on the software been specified? | No | |  | |
| 1. Does SRS use standard terminology and definitions throughout | No | |  | |
| 1. Does the SRS conform to SRS standards | No | |  | |
| 1. Does the SRS identify external interfaces in terms of input and output mathematical variables? Is there justification for the design/implementation constraints? | Yes | |  | |
| 1. \*Are requirements organized to allow for modifications | Yes | |  | |
| 1. Is each unique requirement defined more than once? Are there any redundant statements | No | |  | |
| 1. Does the SRS show explicitly the mapping and complete coverage of relevant requirements and design constraints defined in the concept phase? | No | |  | |
| 1. Is SRS traceable forward through successive development phases (e.g., into the design, code, and test documentation)? | No | |  | |
| 1. Does the SRS contain only necessary implementation details and no unnecessary details? IS it over specified? | Partial | | It contains only necessary information and is not over specified. | |
| 1. Does the documentation follow MES portfolios? | No | |  | |
| 1. Are the requirements verifiable | Yes | |  | |
| 1. Is the documentation clear and unambiguous | Yes | |  | |