Checklist for the LOAFR System:

By: Wil Bishop, Anakin Nolette, Trent Young, Ashlyn Pietrowski

---

NOTE: *All of the check boxes below have been checked off throughout the course of this project. They were unchecked for readability purposes.*

General Checklist:

This checklist was applied to all of the sections below. It was copied and checked off as needed as we were making our edits for the project.

- ☐ Organization and Completeness
  - ☐ Are all internal cross-references to other requirements correct?
  - ☐ Are all sections written at a consistent and appropriate level of detail?
  - ☐ Are all requirements sharing the same voice and style?
- ☐ Correctness
  - ☐ Do any requirements conflict with or duplicate other requirements?
  - ☐ Is each requirement written in clear, concise, unambiguous language?
  - ☐ Is each requirement verifiable by testing, demonstration, review, or analysis?
  - ☐ Is each requirement in scope for the project?
  - ☐ Is each requirement free from content and grammatical errors?
  - ☐ Is any necessary information missing from a requirement?
  - ☐ If so, is it identified as TBD? Can all of the requirements be implemented within known constraints?
- ☐ Traceability
  - ☐ Is each requirement uniquely and correctly identified?
- ☐ Special Issues
  - ☐ Are all requirements actually requirements, not design or implementation solutions?

Section 1 - Introduction:

- ☐ Organization and Completeness
  - ☐ Does the product scope cover the most important goals of the system?

Section 2 - Overall Description:

- ☐ Organization and Completeness
  - ☐ Do the requirements provide an adequate basis for design?
- ☐ Quality Attributes
  - ☐ Are all performance objectives properly specified?

—----------

Section 3 - External Interface Requirements:

- ☐ Organization and Completeness
    - ☐ Are all external hardware, software, and communication interfaces defined?
    - ☐ Are all interactions between interfaces time bound
    - ☐ Is all memory usage accounted for

—----------

Section 4 - System Features:

- ☐ Organization and Completeness
    - ☐ Is the expected behavior documented for all anticipated error conditions?
    - ☐ Does the specification include all of the known customer or system needs?
    - ☐ Is the implementation priority of each requirement included?
    - ☐ Do the requirements provide an adequate basis for design?
    - ☐ Are all requirements written at a consistent and appropriate level of detail?
- ☐ Correctness
    - ☐ Are any specified error messages unique and meaningful?
    - ☐ Can all of the requirements be implemented within known constraints?
- ☐ Quality Attributes
    - ☐ Is the resource allocation of the requirements defined
- ☐ Traceability
    - ☐ Is each software functional requirement traceable to a higher-level requirement?

—----------

Section 5 - Other Nonfunctional Requirements:

- ☐ Organization and Completeness
    - ☐ Are all qualities needed properly specified?
- ☐ Quality Attributes
    - ☐ Are all performance objectives properly specified?
    - ☐ Are all security and safety considerations properly specified?
    - ☐ Are other pertinent quality attribute goals explicitly documented and quantified, with the acceptable tradeoffs specified?

—----------

Section 6 - Other Requirements:

- ☐ Organization and Completeness
    - ☐ Are all TBDs given a direct match (no using the same identity for two different specifications even if they are similar)
- ☐ Correctness
    - ☐ Is there overlap between use cases?

- [ ] Is there overlap between TBDs?
- [ ] Is each use case written in clear, concise, unambiguous language?
- [ ] Is each use case in scope for the project?
- [ ] Is each use case free from content and grammatical errors?
- [ ] Is any necessary information missing from a use case?

- [ ] Quality Attributes (Modularity, Portability, Efficiency, etc. [TERRIFIC UMP])
  - [ ] Are TBDs sufficiently reusable?
  - [ ] Are the TBDs sufficiently interoperable?
  - [ ] Is it difficult to maintain the list of TBDs?
  - [ ] Is it easy to use TBDs within two sections without redefining them (refer to the same thing in Section 4 and Section 5, for example)?
  - [ ] Are TBDs reused with the same name, or reused with new names with a reference to the original (for example, if TBD is introduced in Section 5 and is called 5.1, then would you refer to it as 5.1 in Section 6, or would you refer to it as 6.1 and then define 6.1 as a copy of 5.1)?
- [ ] Traceability
  - [ ] Is each TBD uniquely and correctly identified?
- [ ] Special Issues
  - [ ] Are all use cases genuine use cases?