Design Rubric

Project’s Team: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Project’s Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Presented by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Notes Taken by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Other REVIEW Team Members: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Grading Levels:* Beginning** (Needs work), **Developing** (Satisfactory), **Accomplished** (Good), **Exemplary** (Excellent)

| **Attributes/Quality** | **Grading (B,D,A,E)** | **Notes and Rationale** |
| --- | --- | --- |
| **Problem Statement**   * **Clarity**: Objective of the project is readily identifiable, clear, focused, and can be stated in a precise and concise manner. * **Scope** of the problem is well-defined and justified. * **Has the Problem Statement improved since the Proposal stage, reflecting given feedback** |  |  |
| **Design – These Criteria must be met by each of the following design components**   * Feasible: Solution is practical considering various constraints (e.g., time, cost, complexity, resources) * Informed: Solution has evidence of merit (e.g., literature, rapid prototyping/testing) and addresses the problem. * Creative: Solution is elegant and appropriately engineered. Considers multiple factors (e.g., technological, economic, social, environmental, human). | | |
| **Design – System Architecture**   * Deployment of Function to Nodes relates to the Problem and the Technical Requirements * Organization visibly conveys structure of architecture * Labels completely define the components of the architecture |  |  |
| D**esign – Communication Protocols**   * Completeness – All system-wide behaviours are described through a complete though efficient combination of messages (lists, sequence diagrams) and message contents (formats) * Robust – Protocols consider error scenarios \*\* |  |  |
| **Database**   * Complete (including data types) schema of 2 connected tables, consistent with Communication Protocols. |  |  |
| **Design – Hardware**   * Individual design for each part – part description, key parameters including power, sample schematic; microcontroller interface, and the software interface Schematics * Testing – Procedures planned. |  |  |
| **Design – Software \*\*\***   * Algorithms for any complex operations are explained (e.g. image processing) * Communication Protocols evident in design (e.g. Messages correspond with methods or functions) * Testing is embedded in the solution (through monitoring, logging, demos) |  |  |
| **Use of Engineering Tools**   * Diagrams (UML, schematics) are informed, formal, consistent with standards and work together to communicate a coherent system-wide design |  |  |
| **Written Communication**   * Understandable in language (grammar/spelling) * Understandable in composition: Clear, concise, and organized; Appropriate technical and formal tone and style * Logical: Good reasoning, including use of evidence. Communication is cohesive and coherent. * Convincing: Persuasive and engaging. Accessible, yet technically sound. |  |  |

*\*\** ***Errors in Communication Protocols – You may include dropped UDP packets, but you MUST consider errors above and beyond that are particular to your system – What if a message arrives out of order? What if a message arrives that does not make sense in the current state of the receiver? During your integration phase, it is much much, much, much more likely that the error is in your code than it is due to a lost UDP packet.***

\*\*\* Code may be either FUNCTIONAL (C code) or OBJECT-ORIENTED (Python/Java). For functional code, use call diagrams and flowcharts; for object-oriented code, use class diagrams, object diagrams, sequence diagrams