# Tab 6 – System and Application Architecture

1. GENERAL OVERVIEW

Respondent to provide a description of the proposed system and application architecture for the proposed application.

1. SYSTEM AND APPLICATION ARCHITECTURE QUESTIONS

**Table 6-01: System and Application Architecture Questions**

|  |  |
| --- | --- |
| 1. What is the source language(s) of the product? |  |
| 1. How many environments are available with your proposed solution at no additional cost (e.g., test, training, production)? |  |
| 1. Describe how often major and minor software updates are provided, as well as the level of LFCHD resources required for a major update and the level of resources required for a minor update. |  |
| 1. Please describe the major/minor upgrade process that is required if the solution requires a client-based installation. |  |
| 1. List all browsers that are certified for use with the application and describe any required browser add-ons, function enablement, etc. |  |
| 1. The underlying architecture of the application design is important to LFCHD. Please describe your system architecture model and explain the capabilities and features of this model that led to your use of it in developing this system. |  |
| 1. Please describe how data privacy and security compliance is supported within your proposed software solution. Is the system HIPAA compliant? |  |
| 1. Describe your approach to ensure scalability of the product. This includes transaction growth, upgrades, and replacements of components of the architecture, technology, and application. |  |
| 1. List all hardware/operating system/database platforms upon which the product is supported. Provide specifications in terms of processors, processor speed, memory requirements, and other sizing and capacity factors to assist LFCHD in budgeting for and acquiring hardware. List which industry standard benchmarks or guidelines measures are used to establish this recommendation. |  |
| 1. Describe the design philosophy of your application. Include in your response the degree to which there is a common design philosophy across all modules, common programming languages and tools, and the extent of shared software code across all applications. |  |