|  |  |  |
| --- | --- | --- |
| **For Assessment of:<Project Name>** | | |
| Reviewer Name | Venugopal K | |
| Date Reviewed |  | |
| Project Technical Architect (TA) Name | <Enter Your Name> | |
| System Design Documentation (List all documents that are in focus for this review or provide link to source documentation folder/website) | Document Name: | |
| Version | Revision Date |
| Instructions | **Project TA:** For all questions marked as ‘N/A’ please provide an explanation either under the question or as a general note within this review document.  **Reviewer:** For all questions you mark as ‘N’ please provide an explanation either under the question or as a general note or provide a marked up version of the Project System Design documentation with your redlines, comments and/or questions. | |
| General Project TA Notes |  | |
| General Reviewer Notes |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Project TA (Y/N,N/A)** | **Reviewer (Y/N,N/A)** |
| **1.General Architecture Consideration** |  |  |
| Is the software architecture as simple as possible (but no simpler)? |  |  |
| Is the architecture complete? |  |  |
| Is the proposed solution realizable? |  |  |
| Are all relevant architectural views documented? |  |  |
| Are cross-cutting issues clearly and generally Identified? |  |  |
| Are all stakeholders identified? |  |  |
| Is all building vs. buy decisions included and justified? |  |  |
| Are the major system goals clearly stated? |  |  |
| Does architecture take into account current system or environment constraints (that cannot be avoided)? |  |  |
| If possible, are past, successful designs re-used? |  |  |
| Have all appropriate artifacts been generated? |  |  |
| Have known design risks been identified, analyzed, planned for and mitigated? |  |  |
| Does the design leverage best practices such as design patterns'? |  |  |
| Is the relationship to the software requirements clearly explained and motivated? |  |  |
| Is the flexibility of the architecture demonstrated? |  |  |
|  |  |  |
| **2.High Level Design Consideration** |  |  |
| Does the design have conceptual integrity? |  |  |
| Can the design be implemented within our technology and environment constraints? |  |  |
| Does the design used standard techniques and avoid exotic, hard-to-understand elements? |  |  |
| Is the design extensible? (i.e., will it support future changes or expected changes)? |  |  |
| Does the design use existing, shared or shareable, components? |  |  |
| Is the use of shared resources indicated? |  |  |
|  |  |  |
| **3. Technical Architecture** — describe the technical direction, overall technical solution in terms of class diagrams, objects and components |  |  |
| Is the architectural design approach specified? |  |  |
| Does the architectural design approach include the methodology to be used in building the system? |  |  |
| Does the architectural design approach include the techniques to be used in building the system? |  |  |
| Does the System Architecture Description define the design approach include the rationale for the methodology and the techniques used in determining that rationale? |  |  |
| Does the architectural design approach include the decisions about the system’s behavioral design and other decisions affecting the selection and design of system components? |  |  |
| Is the architecture design depicted using diagrams (i.e. Class Diagrams, Object Mapping, Component Diagram, Sequence Diagram) |  |  |
| Does the architecture design represent the system component(s), hardware, networks, and any other pertinent major system components (e.g., databases, operating systems) that support the complete system? |  |  |
| Is the overall system architecture, including all system components, completely described? |  |  |
| Application is partitioned into logical layers. |  |  |
| Layers represent a logical grouping of components |  |  |
| The tradeoffs of abstraction and loose coupling are well understood for your design |  |  |
| Every component has a single responsibility. |  |  |
| Components are grouped logically into layers |  |  |
| Are the component descriptions sufficiently precise? |  |  |
| Are the relationships between the components explicitly documented? |  |  |
| Components within each layer are cohesive. |  |  |
| Abstraction is used to design loose coupling between layers |  |  |
|  |  |  |
| **4.Structure** |  |  |
| Your design is based on a common application structure such as Client/Server or N-Tie |  |  |
| Deployment environment security requirements are well understood. For example, many security policies require physical separation of presentation logic from business logic across different sub-nets. |  |  |
| Scalability and reliability requirements are well understood for your application. |  |  |
| Deployment scenarios are well understood for your application. |  |  |
|  |  |  |
| 5. **Exception Management** |  |  |
| The design outlines a standardized approach to structured exception handling across the application. |  |  |
| Application exception handling minimizes the information disclosure in case of an exception. |  |  |
| The design identifies generic error messages that are returned to the client |  |  |
| Application errors are logged to the error log. |  |  |
| Is there a strategy for handling abnormal termination? |  |  |
|  |  |  |
| **6. Auditing and Logging** |  |  |
| The design identifies the level of auditing and logging necessary for the application and identifies the key parameters to be logged and audited. |  |  |
| The design considers how to flow caller identity across multiple tiers (at the operating system or application level) for auditing |  |  |
| The design identifies the storage, security, and analysis of the application log files. |  |  |
|  |  |  |
| **7.Validation** |  |  |
| Trust boundaries are identified, and all the inputs are validated when they cross the trust boundary. |  |  |
| A centralized validation approach is used |  |  |
| Client-side validation is used for user experience and server-side validation is used for security. |  |  |
|  |  |  |
| **8.Workflow** |  |  |
| Workflow management requirements are well understood. |  |  |
| Service interfaces are used to interact with external workflow providers |  |  |
|  |  |  |
|  |  |  |