**Iteration 1: Establish an Overall System Structure**

**Step 1: Review Inputs**

|  |  |
| --- | --- |
| Category | Details |
| Design purpose | This is a Greenfield system from a mature domain. The purpose is to produce a sufficiently detailed design to support the construction of the system. |
| Primary functional requirements | From the requirements given, the primary ones were determined to be  UC-1: It directly supports the core business  UC-2: It directly supports the core business  UC-6: It directly supports the core business  UC-11: Due to the technical associated challenges  All of these requirements are considered drivers |
| Quality attribute scenarios | The most important quality attribute scenarios have been prioritized as follows:   |  |  |  | | --- | --- | --- | | Requirement ID | Importance | Difficulty of Implementation | | QA-1 | High | Low | | QA-2 | High | High | | QA-5 | High | High | | QA-6 | Medium | High | | QA-7 | Medium | High |   QA-1, QA-2, QA-5, QA-6 and QA-7 have been selected as drivers |
| Constraints | CON1: The system must be accessible through all major web browsers  CON2: The system must support low-bandwidth connections  CON8: Administrative constraints  All of these constraints are considered drivers |

**Step 2: Establish Iteration Goals by Selecting Drivers**

This is the first iteration, so the goal is establish an overall system architecture, more specifically to defining the structure of the system. As previously mentioned, the architect must be mindful of the following:

* QA-1 Privacy
* QA-2 Availability
* QA-5 Security
* QA-6 Interoperability
* QA-7 Extensibility
* CON1: The system must be accessible through all major web browsers
* CON2: The system must support low-bandwidth connections
* CON8: Administrative constraints

**Step 3: Choose One or More Element of the System to Refine**

Because this is a greenfield development, we choose the entire system to refine.

**Step 4: Choose One or More Design Concepts that Satisfy the Selected Drivers**

|  |  |
| --- | --- |
| Design Decision and Location | Rationale |
| Structure the client using the Rich Internet Applications reference architecture. | * Supports development of rich applications that do not have to be installed on a client’s PC. * The software can be run on a device with any standards-compliant browser. * Supports the rich user interface necessary for this application, without limiting compatibility to certain operating systems. * Compliant with CON-1 * RIA doesn’t fulfill CON-2 well as standard web applications, but the application can be optimized to perform sufficiently well on low-bandwidth connections.   Discarded Alternatives:   |  |  | | --- | --- | | Alternative | Reason for Discarding | | Web Application | * Does not support rich user interfaces necessary for a complex feature set * Does not support advanced client-side business logic * Not designed for high-bandwidth connection, making it inappropriate for file uploads | | Mobile Applications | * Not compliant with CON-1 | | Rich Client Application | * Not compliant with CON-1 | |
| Structure server-side architecture using the Service Application reference architecture. | * Does not have a user interface * Provides services that are consumed by other applications (client-side) * Low coupling of client-side and server-side software makes the system highly extensible, evolvable and interoperable with external systems, making this option compliant with QA-6 and QA-7.   Discarded Alternatives: No other alternatives were considered. No reference architecture could plausibly satisfy requirements |

**Step 5: Instantiate Architectural Elements, Allocate Responsibilities and Define Interfaces**

|  |  |
| --- | --- |
| Design Decision and Location | Rationale |
| Remove local data storage in the client application. Data will be served server-side | * Network connections are reliable, so there is no need to store data locally * Storage of large amounts of persistent data on the RIA reference model is impossible to reliably implement (storage of small amounts of data may be permitted through cookies) * Business logic is primarily server-based. Persisting data server-side allows for greater performance and configurability * Removes need to ensure data is consistent across various clients (eg, do students, administrators and teachers see the same grades?) * Directly supports UC-11 |
| Primary business logic will be server-side | * Supports QA-5 by ensuring high degree of security (eg, students can’t tamper with graded through modifying client-side logic). * Supports QA-7 by making the system highly extensible; client support will not be necessary to add new functionality. * Some business logic can remain client-side, particularly if it is not security-sensitive, and does not concern any other actors. |

**Step 6: Sketch Views and Record Design Decisions**



The RIA architecture was modified to remove the plug-in execution container. This system will not be using plug-ins. Rather, it will be using HTML5 and JavaScript to maintain compatibility with as many browsers and systems as possible.

|  |  |
| --- | --- |
| Element | Responsibility |
| Rich UI Module | Renders the user interface and receives user input |
| Business Processing Module (Client Side) | Implement business operations that can be performed locally, or exposes business functionality from the server side |
| Communication Module (Client Side) | Consume the services provided by the application running on the server side |
| Service Interface (Server Side) | Exposes services that are consumed by the clients |
| Business Modules (Server Side) | Implement business operations |
| Business Logic (Server Side) | Contain modules that preform business logic operations that require processing on the server side |
| DB Access Module (Server Side) | Responsible for the persistence of business entities in the database. |
| Cross Cutting (Server Side) | This layer has modules that cut across different layers. This includes security, I/O and logging. |

**Step 7: Perform Analysis of Current Design an Review Iteration Goal and Achievement of Design Purpose**

|  |  |  |  |
| --- | --- | --- | --- |
| Not Addressed | Partially Addressed | Completely Addressed | Design Decisions Made During the Iteration |
| UC-1 |  |  | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
|  | UC-2 |  | Selected reference architecture established the modules that can support large file uploads. |
| UC-6 |  |  | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
|  | UC-11 |  | The Service Application reference architecture will enable maintainers to maintain backups of the entire system |
| CON -8 |  |  | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
| QA-1 |  |  | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
| QA-2 |  |  | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
|  | QA-5 |  | Selected reference architecture established the modules that will support this functionality. |
|  | QA-6 |  | Selected reference architecture established the modules that will support this functionality. |
|  | QA-7 |  | Selected reference architecture established the modules that will support this functionality. |
|  |  | CON-1 | Use of standard web technologies allows the service to be accessible through all major web browsers |
| CON-2 |  |  | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |