**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  R1: It directly supports the core business  R4: It directly supports the core business  R11: It directly supports the core business  R14: It directly supports the core business  R15: It directly supports the core business  R16: It directly supports the core business  R48: It directly supports the core business  R53: It directly supports the core business  R54: It directly supports the core business  R53: It directly supports the core business  R88: Due to the technical associated challenges  R89: Due to the technical associated challenges  R97: It directly supports the core business  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 | | R27 | High | Low | | R30 | High | High | | R44 | High | High | | R96 | Medium | High |   R27, R30, R92 and R44 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  CON3: System must be able to handle at least 500 simultaneous users at a time  Both 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. The selected drivers are outlined in the table above.

**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 R92 and R96.   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 R88 (allow maintainers to create back ups of entire system) |
| Primary business logic will be server-side | * Supports R44 by ensuring high degree of security (eg, students can’t tamper with graded through modifying client-side logic). * Supports R92 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 companitiblity 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 | Requirement Description | Design Decisions Made During the Iteration |
|  | R1 |  | The system shall provide static course information | Selected reference architecture established the modules that will support this functionality. |
|  | R4 |  | The system shall provide dynamic course information | Selected reference architecture established the modules that will support this functionality. |
| R11 |  |  | The system shall enable students to subscribe/unsubscribe to courses | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
| R14 |  |  | The system shall be able to let students submit textual content | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
|  | R15 |  | The system shall be able to let students upload files | Selected reference architecture established the modules that can support large file uploads. |
| R16 |  |  | The System shall allow sending messages to individuals, teams or all course participants at once | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
| R48 |  |  | The system shall allow lecturers to create courses | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
| R53 |  |  | The system shall allow lecturers to upload course material for lectures | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
| R54 |  |  | The system shall enable lecturers to manage grades (insert, update, calculate final grade) | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
|  | R88 |  | The system shall allow maintainers to create back-ups of the entire system | The Service Application reference architecture will enable maintainers to maintain backups of the entire system |
|  | R89 |  | The system shall allow maintainers to restore partial and complete back-ups of a specific date | The Service Application reference architecture will enable maintainers to restore backups of the system |
| R97 |  |  | The system shall allow only the administration to manage courses | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
| R27 |  |  | The system shall protect user’s privacy | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
| R30 |  |  | The system shall have high availability | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |
|  | R44 |  | The system will be secure | Selected reference architecture established the modules that will support this functionality. |
|  | R96 |  | The system shall be interoperable with secondary university systems | Selected reference architecture established the modules that will support this functionality. |
|  |  | CON-1 | The system must be accessible through all major web browsers | Use of standard web technologies allows the service to be accessible through all major web browsers |
| CON-2 |  |  | The system must support low-bandwidth connections | No relevant decisions made, as it is necessary to identify the elements that participate in the use case associated with the scenario. |