Iteration 1

ADD Step 1: Review Inputs

|  |  |
| --- | --- |
| Category | Details |
| Design Purpose | This is a greenfield system for a mature domain. The purpose is to produce a sufficiently detailed design to support the construction of the system. |
| Primary functional requirements | From the use cases presented, the primary ones were determined to be: UC-2: Because it directly supports the core business  UC-3: Because it directly supports the core business  UC-4: Because it directly supports the core business  UC-5: Because it directly supports the core business |
| Quality attribute scenarios | The scenarios were previously described, they have now been prioritized as follows:   |  |  |  | | --- | --- | --- | | Scenario ID | Importance to the Customer | Difficulty of Implementation According to the Architect | | QA-1 | Medium | Low | | QA-2 | High | High | | QA-3 | Medium | Medium | | QA-4 | High | Low | | QA-5 | High | High |   From the list, only QA-2, QA-4, and QA-5 are selected as drivers. |
| Constraints | All of the constraints discussed previously are included as drivers. |
| Architectural constraints | All of the architectural constraints discussed previously are included as drivers |

Step 2: Establish Iteration Goal by Selecting Drivers

EXAMPLE:

* QA-1: Security
* QA-2: Availability
* QA-4: User Friendliness
* CON-4: Constrained to Java application compatibility
* CRN-2: Leverage team’s knowledge on Java and Swing

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

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

|  |  |
| --- | --- |
| Design Decisions and Location | Rationale |
| Logically structure the client part of the system using the Rich Client Application reference architecture | “Rich client applications are installed and run on a user’s machine. Because the application runs on the user’s machine, its user interface can provide a high-performance, interactive, and rich user experience.” (Cervantes, 2016)  This decision allows us to leverage the familiarity with the Java technologies which addressed CON-4 and CRN-2. Since we are not using web technologies and our system is not accessible from a web browser using Java technologies is an effective solution.  Discarded alternatives:   |  |  | | --- | --- | | Alternative | Reason for Discarding | | Mobile Applications | This type of reference architecture is more suited for handheld devices. We want our system to be accessible from student laptop computers and the desktop computers located on campus. | | Web Applications | This reference architecture is discarded due to unfamiliarity with designing and developing secure, full stack web applications that provide rich user interface experience. | | Rich Internet applications | Just like web application reference architecture, this alternative is also discarded due to unfamiliarity with web technologies and the rich development capabilities provided by the Java environment. | |
| Logically structure the server part of the system using the Service Application reference architecture | “Service applications do not provide a user interface but rather expose services that are consumed by other applications” (Cervantes, 2016) Since this part of the system does not need to be interactive, we are not worried about the presentation layer. Loose coupling that comes with using Service Application reference architecture also help us achieve high availability (QA-2) as system maintenance can be done during downtimes without having a negative impact on the client side of our system. |
| Physically structure the application using the three-tier deployment pattern | A three tier deployment is appropriate since the system requires the use of a database, a middle layer to establish the business logic and a client layer (e.g. student’s laptop). Other n-tier patterns are discarded because extra servers are not required (when n>4) and a 2-tier architecture does not include a database layer. |
| Build the user interface of the client application using JavaFX | The developer team is already familiar with Java technologies (CRN-2) and a user friendly (QA-4) interface can easily be created with this decision. |
| Deploy the application using Spring | Although it can quite complex, Spring provides great tool support, easy integration with other frameworks and security (QA-1) |

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

|  |  |
| --- | --- |
| Design Decision and Location | Rationale |
| Remove local data sources in the rich client application |  |
| Create a module dedicated to accessing the time servers in the data later of the Server Application reference architecture |  |

Step 6: Sketch Views and Record Design Decisions

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

|  |  |  |  |
| --- | --- | --- | --- |
| Not Addressed | Partially Addressed | Completely Addressed | Design Decisions Made During the Iteration |
|  |  | CRN-2 |  |
|  |  | CON-4 |  |
|  |  | QA-1 |  |
|  |  | QA-2 |  |
|  |  | QA-4 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |