Iteration 2

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

|  |  |
| --- | --- |
| Design Decisions and Location | Rationale and Assumptions |
| Create a Domain Model for the application | In order to identify the major entities of the system and their relationship with each other in the domain, a domain model needs to be created. No alternatives exist for the domain model as it eventually needs to be discussed. |
| Identify Domain Objects that map to functional requirements | A domain object is a building block of the application as it encapsulates the functional elements. |
| Decompose Domain Objects into general and specialized components | Functionality is represented by domain objects but some small-scaled extra elements located within the layers might be needed. Each module is associated with a layer. |
| Use Spring framework | Spring is selected as the development team is already familiar with it. It is a popular Java framework that is light and it has a good tool support. |

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

|  |  |
| --- | --- |
| Design Decisions and Location | Rationale |
| Create only an initial domain model | In order to accelerate the design process only an initial domain model is created. |
| Map the system use cases to domain objects | The system’s use cases need to be analyzed to identify the domain objects. For each use case, domain objects are identified to achieve CRN-3. |
| Decompose the domain objects across the layers to identify layer-specific modules with an explicit interface | The architect identifies the modules based on the primary use cases. It is recommended that other team members contribute to this process by identifying the rest of the modules. CRN-3 is addressed again by dividing the work between the members in the team. As modules are being identified, a new architectural concern is realized:  CRN-4: Most of the module should be unit tested. |
| Connect components associated with modules using Spring | With Spring the modules can be unit tested and different aspects can be supported. |
| Associate frameworks with a module in the data layer | Spring is associated with the modules in the server side. |