Department of Veterans Affairs

**Automated Surgical Risk Calculator (ASRC)**

Technical Manual



**May 2015**

Version 1.6

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 11/25/2014 | 1.0 | Document Shell | D. Tombs |
| 11/25/2014 | 1.1 | Technical Edit | S. Vetzel |
| 12/1/2014 | 1.2 | Partial Completion | D. Tombs |
| 12/17/2014 | 1.3 | Technical Edit | S. Vetzel |
| 4/23/2015 | 1.4 | Define Administrative Pages and the Software Architecture section | D. Tombs |
| 5/13/2015 | 1.5 | Initial update for VistA Patch Detailed Design | J. Swesky |
| 5/18/2015 | 1.6 | Technical Edit | S. Ambrose |

Table of Contents

[1. Introduction 1](#_Toc419792218)

[1.1. Identification 1](#_Toc419792219)

[1.2. Scope 1](#_Toc419792220)

[1.3. User Characteristics 1](#_Toc419792221)

[1.4. Definitions, Acronyms, and Abbreviations 1](#_Toc419792222)

[1.5. References 1](#_Toc419792223)

[2. Background 3](#_Toc419792224)

[2.1. Overview of the System 3](#_Toc419792225)

[2.2. Overview of the Business Process 3](#_Toc419792226)

[2.3. Business Benefits 3](#_Toc419792227)

[2.4. Assumptions and Constraints 4](#_Toc419792228)

[2.4.1. Design Assumptions 4](#_Toc419792229)

[2.4.2. Design Constraints 4](#_Toc419792230)

[2.4.3. Design Trade-offs 4](#_Toc419792231)

[2.5. Overview of the Significant Requirements 5](#_Toc419792232)

[2.5.1. Overview of Significant Functional Requirements 5](#_Toc419792233)

[2.5.2. Overview of Functional Workload / Performance Requirements 5](#_Toc419792234)

[2.5.3. Overview of Operational Requirements 5](#_Toc419792235)

[2.5.4. Overview of the Technical Requirements 5](#_Toc419792236)

[2.5.5. Overview of the Security or Privacy Requirements 6](#_Toc419792237)

[2.5.6. Overview of System Criticality and High Availability Requirements 6](#_Toc419792238)

[3. Conceptual Design 7](#_Toc419792239)

[3.1. Conceptual Application Design 7](#_Toc419792240)

[3.1.1. Application Context 7](#_Toc419792241)

[3.1.2. High-Level Application Design 8](#_Toc419792242)

[3.2. Conceptual Data Design 9](#_Toc419792243)

[3.2.1. Project Conceptual Data Model 9](#_Toc419792244)

[3.2.2. User Interface Data Mapping 10](#_Toc419792245)

[3.2.2.1. Calculation Pages 10](#_Toc419792246)

[3.2.2.1.1. New Calculation Page 11](#_Toc419792247)

[3.2.2.1.2. Enter Risk Variables Page 12](#_Toc419792248)

[3.2.2.1.3. Calculation Results Page 13](#_Toc419792249)

[3.2.2.2. Administrative Pages 14](#_Toc419792250)

[3.2.2.2.1. Administration Home 15](#_Toc419792251)

[3.2.2.2.2. Edit Variable Pages 16](#_Toc419792252)

[3.2.2.2.3. Edit Rule Page 17](#_Toc419792253)

[3.2.2.2.4. Edit Model Page 17](#_Toc419792254)

[3.2.2.3. Administrative Reports 18](#_Toc419792255)

[3.2.2.4. Unmapped Data Element 18](#_Toc419792256)

[4. System Architecture 19](#_Toc419792257)

[4.1. Hardware Architecture 19](#_Toc419792258)

[4.2. Software Architecture 19](#_Toc419792259)

[4.2.1. The Server-Side Components 21](#_Toc419792260)

[4.2.2. The Client-Side Components 21](#_Toc419792261)

[4.3. Service Oriented Architecture / ESS 22](#_Toc419792262)

[4.4. Enterprise Architecture 22](#_Toc419792263)

[5. Data Design 24](#_Toc419792264)

[5.1. DBMS Files 24](#_Toc419792265)

[5.2. Non-DBMS Files 24](#_Toc419792266)

[6. Detailed Design 26](#_Toc419792267)

[6.1. Hardware Detailed Design 26](#_Toc419792268)

[6.2. Software Detailed Design 26](#_Toc419792269)

[6.2.1. Database Repository 26](#_Toc419792270)

[6.2.2. Java Web Application Detailed Design 27](#_Toc419792271)

[6.2.3. VistA Patch Detailed Design 27](#_Toc419792272)

[6.2.3.1. Routines (Entry Points) 27](#_Toc419792273)

[6.2.3.2. Templates 30](#_Toc419792274)

[6.2.3.3. Data Entries Affected by the Design 31](#_Toc419792275)

[The following is a complete FileMan listing of the new file: 32](#_Toc419792276)

[6.2.3.4. Unique Record(s) 35](#_Toc419792277)

[6.2.3.5. File or Global Size Changes 36](#_Toc419792278)

[6.2.3.6. Mail Groups 36](#_Toc419792279)

[6.2.3.7. Security Keys 37](#_Toc419792280)

[6.2.3.8. Options 39](#_Toc419792281)

[6.2.3.9. Protocols 41](#_Toc419792282)

[6.2.3.10. Remote Procedure Call (RPC) 43](#_Toc419792283)

[6.3. Service Oriented Architecture / ESS Detailed Design 44](#_Toc419792284)

[6.3.1. Service Description for <Consumed Service Name> 45](#_Toc419792285)

[6.3.2. Service Design for <Provided Service Name> 45](#_Toc419792286)

[7. External System Interface Design 46](#_Toc419792287)

[7.1. Interface Architecture 46](#_Toc419792288)

[7.2. Interface Detailed Design 46](#_Toc419792289)

[8. Security and Privacy 49](#_Toc419792290)

[8.1. Security 49](#_Toc419792291)

[8.2. Privacy 49](#_Toc419792292)

[A. Additional Information 50](#_Toc419792293)

[A.1. RTM 50](#_Toc419792294)

[A.2. Packaging and Installation 50](#_Toc419792295)

[A.3. Design Metrics 50](#_Toc419792296)

[A.4. Acronym List and Glossary 50](#_Toc419792297)

List of Figures

[Figure 1 - Application Context Diagram 7](#_Toc419792298)

[Figure 2 - High-Level Application Design 8](#_Toc419792299)

[Figure 3 - Conceptual Data Model 9](#_Toc419792300)

[Figure 4 - Application Clinical Workflow 11](#_Toc419792301)

[Figure 5 - Notional New Calculation Page 12](#_Toc419792302)

[Figure 6 - Notional Enter Risk Variables Page 13](#_Toc419792303)

[Figure 7 - Notional Administration Home, Partial 15](#_Toc419792304)

[Figure 8 - Notional Edit Discrete Numerical Variable 16](#_Toc419792305)

[Figure 9 - Notional Edit Rule Page 17](#_Toc419792306)

[Figure 10 - Notional Edit Model Page 18](#_Toc419792307)

[Figure 11 - ASRC Hardware Components 19](#_Toc419792308)

[Figure 12 - Software Architecture 20](#_Toc419792309)

List of Tables

[Table 1 - Significant Functional Requirements 5](#_Toc419792310)

[Table 2 - Technical Requirements 5](#_Toc419792311)

[Table 3 - Security or Privacy Requirements 6](#_Toc419792312)

[Table 4 - External System Interfaces 7](#_Toc419792313)

[Table 5 - Externally Shared Data Stores 8](#_Toc419792314)

[Table 6 - Objects in the High Level Application Design 9](#_Toc419792315)

[Table 7 - Routines (Instructions) 28](#_Toc419792316)

[Table 8 - Routines (Grouping) 30](#_Toc419792317)

[Table 9 - Templates (Instructions) 31](#_Toc419792318)

[Table 10 - Templates 31](#_Toc419792319)

[Table 11 - Data Entries Affected by the Design 32](#_Toc419792320)

[Table 12 - Unique Record ID 35](#_Toc419792321)

[Table 13 - File or Global Size Changes 36](#_Toc419792322)

[Table 14 - Mail Groups (Instructions) 36](#_Toc419792323)

[Table 15 - Mail Groups 37](#_Toc419792324)

[Table 16 - Security Keys (Instructions) 38](#_Toc419792325)

[Table 17 - Security Keys 38](#_Toc419792326)

[Table 18 - Options (Instructions) 40](#_Toc419792327)

[Table 19 - Options 40](#_Toc419792328)

[Table 20 - Protocols (Instructions) 41](#_Toc419792329)

[Table 21 - Protocols 42](#_Toc419792330)

[Table 22 - RPCs (Instructions) 43](#_Toc419792331)

[Table 23 - RPCs 44](#_Toc419792332)

[Table 24 - Glossary 50](#_Toc419792333)

# Introduction

This Technical Manual details the Automated Surgical Risk Calculator (ASRC) Tool technical solution as built. It documents the solution from a physical, logical, business workflow, and software architectural perspective depicting the various models and layers of the solution and methodologies used. It describes how the solution works in parts as well as a whole.

Please see the project README for background on the application.

## Identification

The Tool includes both a Java Web Application and a supporting VistA patch for VistA integration. This document is meant to accompany version 0.7 of both components.

## Scope

This Manual documents the technical solution from the following perspectives:

* Physical: the devices, possibly virtual, on which components run
* Logical: the components of the system
* Business Workflow: how the components fit in to the business workflow
* Software Architectural: the software frameworks and design of the components

Network design is out of the scope of this manual and must be designed externally to the Automated Surgical Risk Calculator Tool.

## User Characteristics

There are two main groups of users of the tool: clinical users and administrative users.

Clinical users use the Tool to perform risk calculations. They are not necessarily familiar with databases, configuration of software systems, or software components.

Administrative users modify the Tool configuration primarily to update the risk models year by year. They are familiar with SQL databases, configuration of software systems, and the basics of software components. They are not, however, familiar with the detailed design and implementation of software systems.

## Definitions, Acronyms, and Abbreviations

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to be properly interpreted. This information may be provided by reference to one or more appendixes or by reference to other documents.

## References

This subsection should:

1. Provide a complete list of all documents referenced elsewhere.
2. Identify each document by title, report number - if applicable - date, and publishing organization.
3. Specify the sources from which the references can be obtained.

This information may be provided by reference to an appendix or to another document.

# Background

## Overview of the System

The Automated Surgical Risk Calculator Tool (“the Tool”) can be used at the time the patient is considered for surgical referral by a primary care provider and at the time a surgeon is requesting a surgery. This Tool will support clinical decision-making regarding perioperative risk (includes preoperative, intraoperative, and postoperative). Providers will verify patient-specific data that is automatically pulled from available data sources, enter remaining fields, and be provided with a real-time individual risk calculation of perioperative surgical mortality based on historic Veterans Affairs Surgical Quality Improvement Program (VASQIP) data and current VASQIP risk-adjusted models that are specialty-specific. The data entered and the calculated results will be available for viewing in the Computerized Patient Record System (CPRS) as a progress note. The data will also be transferred and stored as discrete fields in Veterans Health Systems and Technology Architecture (VistA) and a Structured Query Language (SQL) database for use by the National Surgery Office (NSO).

## Overview of the Business Process

Provide an overview of the business processes that this application will support. This subsection puts the System into perspective with other related Systems or Projects. It is suggested that this information be illustrated in a graphical format. The business processes may be provided in any number of graphical formats including Data Flow Diagrams, Unified Modeling Language (UML), or Business Process Execution Language (BPEL) as desired.

This section should include parties external to OIT (be sure to provide a reference or Business Process ID). Provide reference (Business Process ID).

An example is provided in Figure 1 below. In lieu of a diagram or the table that follows, the URL of a specific diagram (or model) may be provided in a publicly available location, as long as that the model provides the information in Table 3 (noted below).

## Business Benefits

There is an existing Risk Calculator tool provided by the NSO, but it is not widely used due to the fact that it is not accessible from within Computerized Patient Record System (CPRS) and that its current location is not well known to most surgical providers.

The Automated Surgical Risk Calculator Tool provides the following benefits:

* It integrates the Tool into the Electronic Health Record (EHR) environment for easy access by VA providers.
* It eliminates reentry of information already available within the EHR.
* When the risk calculation is performed, it saves the data entered and calculation results to the EHR for others to view or to be used for quality assessment.
* It permits calculation of risks other than 30-day mortality, to include longer term mortality related to frailty or associated co-morbidities.
* It permits updating the statistical risk models by administrative staff without software development effort.

## Assumptions and Constraints

This section describes the assumptions and constraints that impacted the design of the system.

### Design Assumptions

The development team made the following assumptions which influenced the design of the Tool:

* All data that the Tool must retrieve from VistA can be retrieved via new or existing Remote Procedure Calls (RPCs).
* A Clinical Context Object Workgroup (CCOW) implementation will be available to support operation of the Tool.

### Design Constraints

The following constraints (e.g., schedule, cost, and technical) impacted the design of the Tool:

* The Tool is a prototype that must be completed over a 12-month period. Further development and testing required for VA national release may be performed as a separate effort.
* There exists no widely used and easily augmented method of exposing VistA data via Web Services. (MDWS is widely used but cannot easily be augmented.)
* The Tool development team is more familiar with Java development than .NET development and therefore prefers Java-based solutions.
* The Tool must be testable in the VA’s Future Technology Lab (FTL).

### Design Trade-offs

Discuss the trade-offs involved with the design chosen and the reasons for your choices.

Example 1: an increase in security controls will likely entail a decrease in ease-of-use

Example 2: an increase in the flexibility of a system will entail a decrease in the simplicity of that system

For this reason, the designer must decide to put a higher value on some attributes over others. Some areas to consider include:

* Flexibility
* Interoperability
* Performance
* Reliability and robustness
* Usability (including 508 compliance)

## Overview of the Significant Requirements

### Overview of Significant Functional Requirements

The following functional requirements impacted the software design.

Table 1 - Significant Functional Requirements

| ID | Requirement |
| --- | --- |
| ASRC-16 | Launch the Tool from CPRS, sharing user and patient context |
| ASRC-152 | Search for Procedure from thousands of possible Procedures |
| ASRC-161 | Populate calculation variables from patient’s EHR in VistA |
| ASRC-91 | Allow manual entry and VistA override for all calculation variables |
| ASRC-139 | Derive calculation variables from other variables |
| ASRC-100 | Allow updating risk models without development effort |
| ASRC-156 | Allow provider to sign the calculation and save results to VistA and a national SQL database |
| ASRC-153 | Modify VistA Request for Surgery Workflow |
| ASRC-103 | Generate 3 reports in the Tool |

### Overview of Functional Workload / Performance Requirements

No functional workflow requirements were identified as part of this prototype.

### Overview of Operational Requirements

No operational requirements were identified as part of this prototype.

### Overview of the Technical Requirements

The following technical requirements impacted the software design.

Table 2 - Technical Requirements

| ID | Requirement |
| --- | --- |
| ASRC-161 | Populate calculation variables from patient’s EHR in VistA |
| ASRC-156 | Allow provider to sign the calculation and save results to VistA and a national SQL database |

### Overview of the Security or Privacy Requirements

The following significant security and privacy requirements impacted the software design.

Table 3 - Security or Privacy Requirements

| ID | Requirement |
| --- | --- |
| ASRC-100 | Allow updating risk models without development effort |
| ASRC-49 | Provider signs risk calculation via electronic signature code |

As the Tool was developed as a Prototype that will only handle test patient data, no PII/PHI requirements were identified.

### Overview of System Criticality and High Availability Requirements

No System Criticality or High Availability requirements were identified as part of this prototype.

# Conceptual Design

This section depicts the broad design of the solution, particularly in the context of user and external system interfaces. Throughout this section, VistA is regarded as an external system though ASRC includes a VistA integration patch. Section 6.2.3 documents the VistA patch design.

## Conceptual Application Design

### Application Context



Figure 1 - Application Context Diagram

Table 4 - External System Interfaces

| Name | Input Messages | Output Messages | Owner |
| --- | --- | --- | --- |
| Site VistAs | Retrieve Patient Data for calculation inputs | Store results from each calculation | Each VA Site |

Table 5 - Externally Shared Data Stores

| Name | Data Stored | Owner | Access |
| --- | --- | --- | --- |
| National ASRC Results Database | Calculation inputs and results from each calculation. Configuration data (e.g., risk model definitions) is also stored in this database for simplicity. | This System | Create, Read, Update, and Delete |

### High-Level Application Design

The below High-Level Application Design Diagram expands the application to show its major components. Since the design is conceptual, it does not identify specific technologies or software libraries with the exception of VistALink. VistALink is identified because it is the only VistA integration technology that satisfies the Design Constraints.



Figure 2 - High-Level Application Design

Table 6 - Objects in the High Level Application Design

| Name | Description | External Interfaces |
| --- | --- | --- |
| Domain Model | An Object-Oriented Model of the Risk Calculation Domain, including calculation input variables, the models themselves, and the calculation results. | None |
| VistALink | VistA-interfacing technology that allows making Remote Procedure Calls (RPCs) to VistA from Java. | Site VistAs |
| Web Application Frontend | Presents a Web user interface to the application. | The application’s users |
| Persistence Layer | Persists Domain Model objects to the relational database. | The National ASRC Results Database |

## Conceptual Data Design

### Project Conceptual Data Model

The following diagram is a conceptual data model showing the high-level data entities and their relationships. It is intended to depict the application’s data in a generic fashion, not as a table structure or an object class hierarchy.



Figure 3 - Conceptual Data Model

As shown, the data divides naturally into two subject areas: the model definition and the calculation results.

The primary entity within the Model Definition subject area is the risk model. A risk model has multiple input variables, each multiplied by a model-specific coefficient to calculate the final risk result. Multiple models may use the same variable but with different coefficients. Each surgical specialty has one or more risk models.

The primary entity within the Calculation Results subject area is the Risk Calculation. A Calculation represents a single risk calculation performed by a user. It records the patient, input variables, results, whether the results were signed or not, and other attributes associated with the calculation.

### User Interface Data Mapping

Users retrieve and modify application data through three user interfaces: the calculation pages, the administrative pages, and the administrative reports.

Note that all screenshots included in this manual are for illustrative purposes and may include contents different from what is show in the actual application.

#### Calculation Pages

Clinical users (see User Characteristics above) retrieve and modify application data only through the Calculation Pages. These pages use (but do not modify) data in the Model Definition subject area and store data in the Calculation Results subject area.

These pages provide the following workflow:



Figure 4 - Application Clinical Workflow

Each individual page of the Clinical Pages is summarized in the following sub-sections.

##### New Calculation Page

The New Calculation page begins a new risk calculation with having the user select the Surgical Specialty for the surgery to be performed.

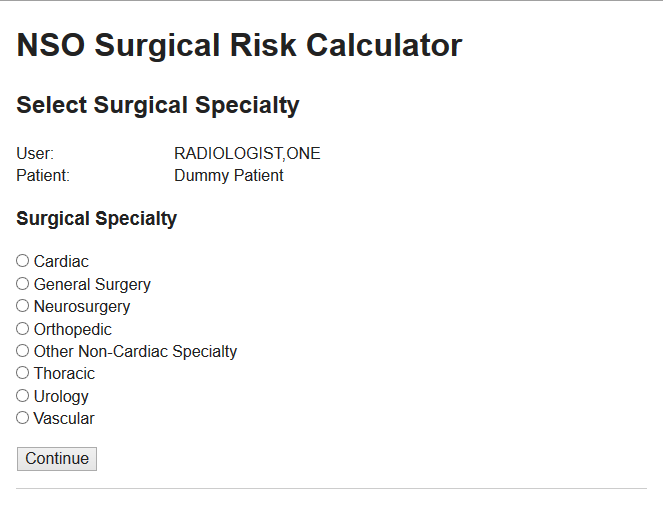


Figure 5 - Notional New Calculation Page

Note that this page is just one step of the calculation workflow and no data is saved to the database yet when the user completes the page.

##### Enter Risk Variables Page

The Enter Variables page allows the user to manually enter values for the risk calculation’s input variables, including overriding values that the application automatically retrieved from VistA/CPRS.

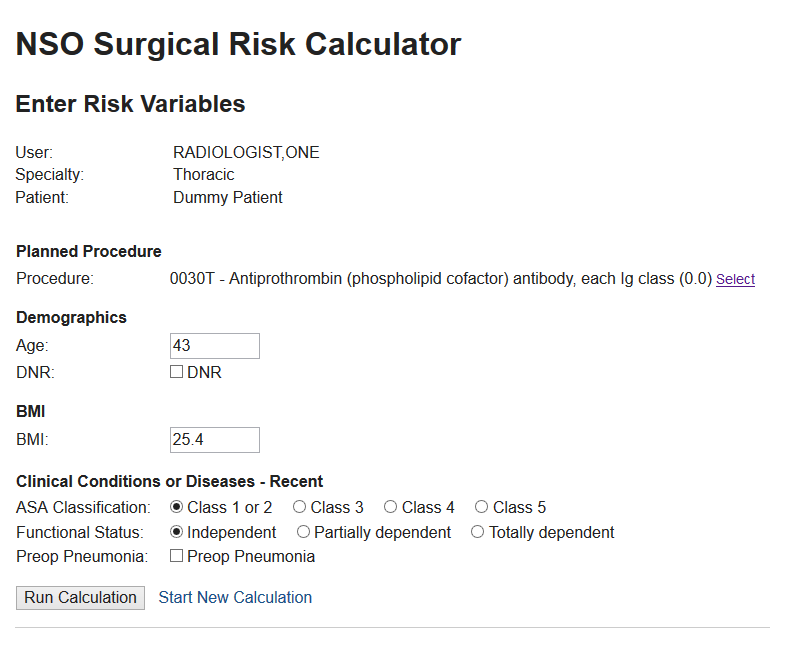


Figure 6 - Notional Enter Risk Variables Page

When the user clicks *Run Calculation*, the application performs the calculation and saves selected calculation data to the database for reporting purposes. The calculation results themselves are not yet saved.

##### Calculation Results Page

The Calculation Results page displays the calculated risk results as well as a read-only table of calculation inputs. It allows the user to sign the calculation.

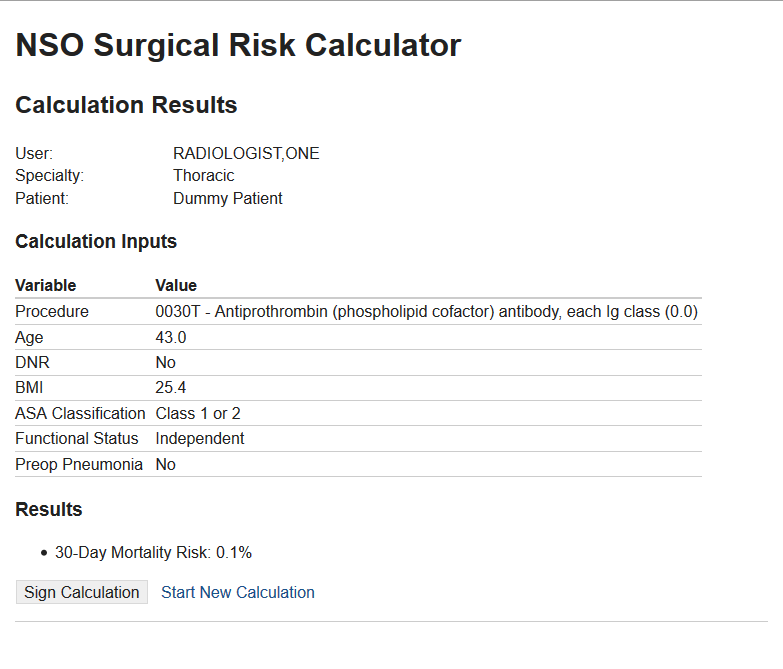


Figure 7 - Notional Calculation Results Page

Upon signature, the application saves all input values and output results to the database. It also stores a textual copy of the results page in VistA TIU for display on the CPRS Notes Tab and stores selected calculation data, including the results, in VistA Surgery to support the VistA workflow changes described elsewhere in this document.

#### Administrative Pages

Administrative users (see User Characteristics above) retrieve and modify data through the Administrative Pages. The primary purpose of these pages is to update the risk models used in calculations. These pages thus update data in the Model Definition subject area.

The ASRC User Guide covers the actual Model Administration workflow. This manual includes a representative sample of the Administrative Pages to explain the data mapping.

##### Administration Home

The Administration Home page gives the user a summary view of the current model configuration and contains links to other administrative pages.

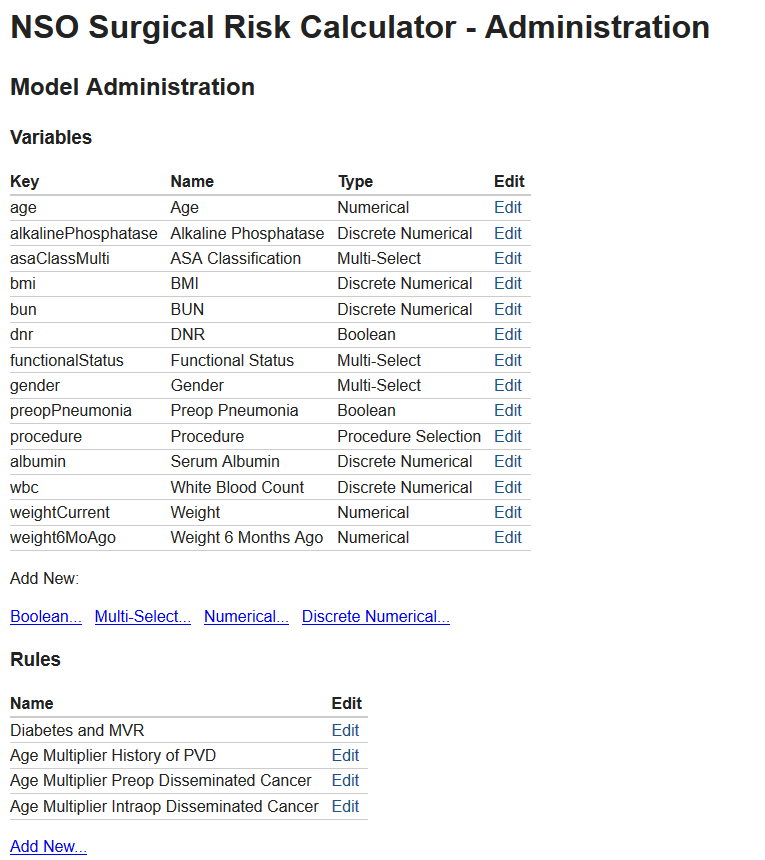


Figure 7 - Notional Administration Home, Partial

The above notional screen design for the Administration Home depicts the summary information and add/edit links for Variables and Rules. The implemented screen will also include Models and Specialties. The application only reads the Model Definition data on this page. The user may navigate to the Add and Edit pages to modify the data. Below sections depict a representative sample of the Edit pages. The Add pages are very similar, but add new entities as opposed to editing existing ones.

##### Edit Variable Pages

The Edit Variable pages allow users to edit the attributes of the variables. Each variable type (i.e. Boolean, Numerical, Multi-Select, etc.) has a slightly different screen design since each type has a different set of attributes. See the ASRC User Guide for an explanation of the different variable types.

A notional Edit Discrete Numerical Variable page is included below as a representative example:

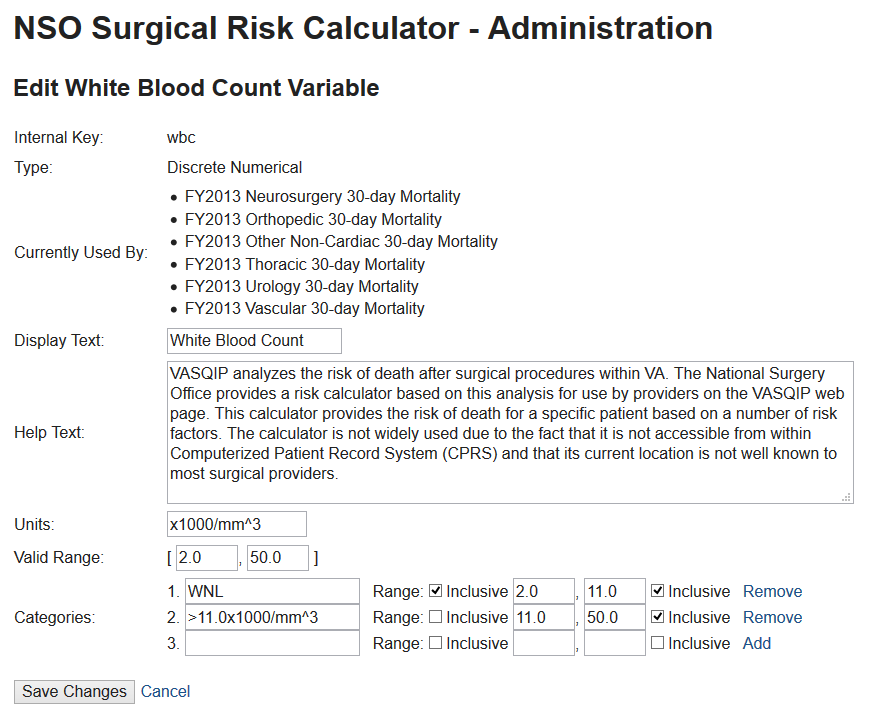


Figure 8 - Notional Edit Discrete Numerical Variable

As shown, the page includes some read-only reference information (such as the variable type and the models that currently use it) and other writable attributes. All writable attributes are of the Variable entity (see Section 3.2.1); some read-only data are of other Model Definition entities.

##### Edit Rule Page

The Edit Rule page allows the user to edit Rule entities.

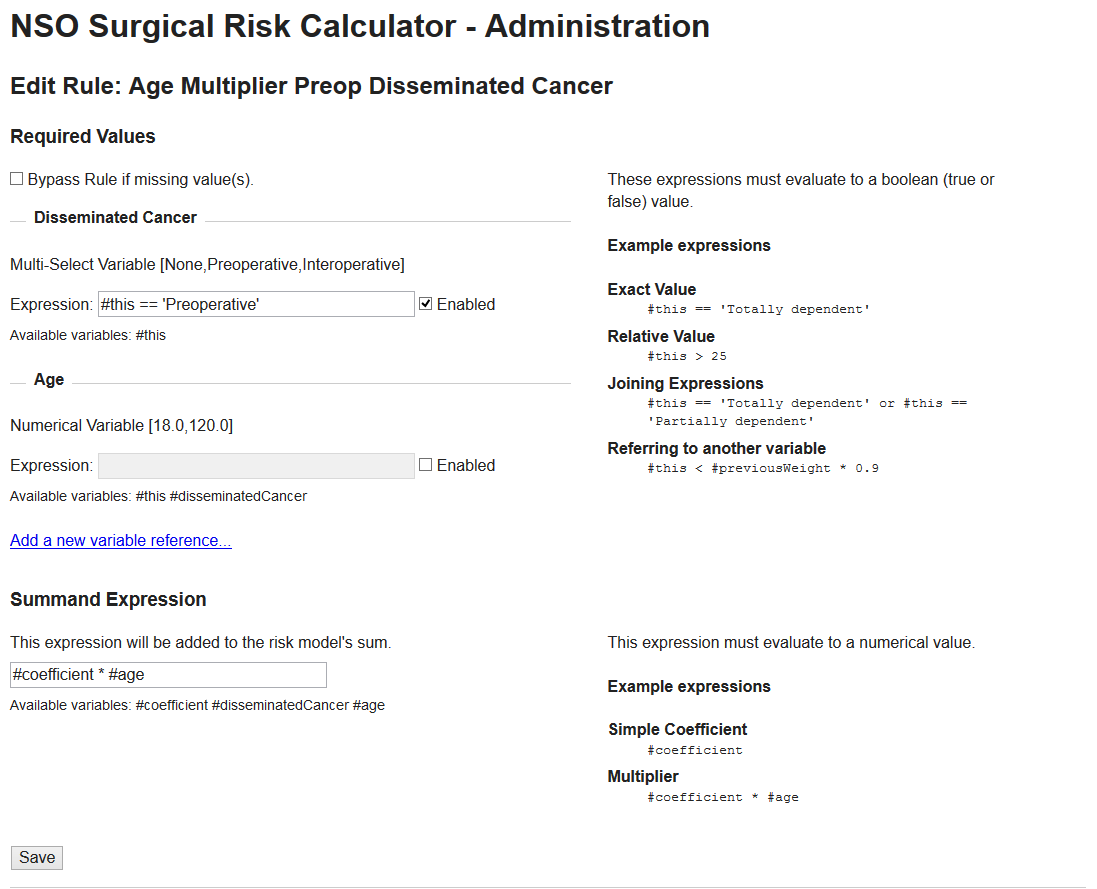


Figure 9 - Notional Edit Rule Page

##### Edit Model Page

The Edit Model page allows the user to edit Model entities.

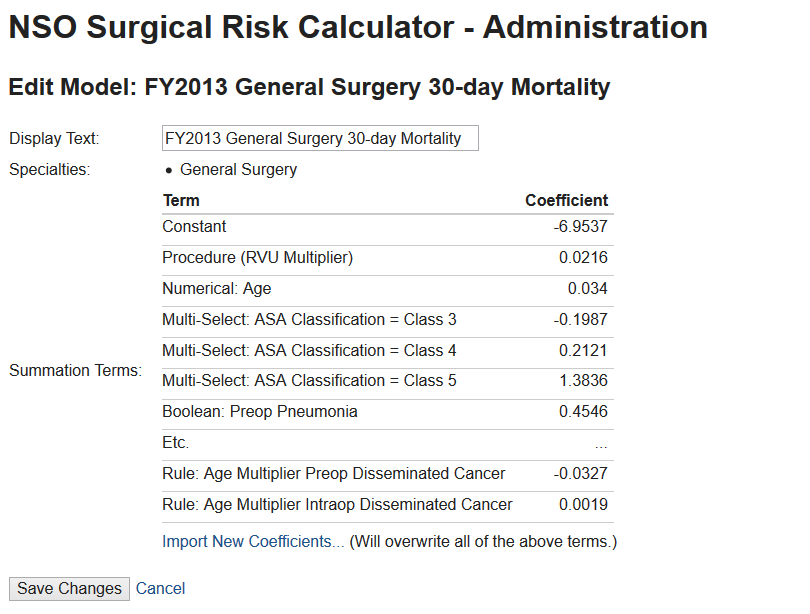


Figure 10 - Notional Edit Model Page

As shown, the page presents mostly read-only summary information for the current model. The user will bulk upload a new set of terms in tabular format via the shown *Import New Coefficients...* link.

#### Administrative Reports

The Administrative Reports are to be designed but will include, at minimum:

* A Utilization Report, including the number of users of the tool, whether each calculation was signed or not, and the time it took the user to complete the calculation.
* A Summary Report, including individual calculation outcomes grouped by CPT Code, Surgical Specialty, Facility, and User Type.

#### Unmapped Data Element

All persistent application data is mapped to one or more of the above pages and/or reports. There is no unmapped data.

# System Architecture

This section describes the system architecture for the project.

## Hardware Architecture

As a prototype project, the hardware architecture is loosely defined to include two servers: the VistA server and the Java Application Server. Any number of client workstations may access these servers. The following diagram depicts the interconnections between the hardware components. The direction of the arrows indicates which component initiates communication.



Figure 11 - ASRC Hardware Components

In a production deployment of the system, the hardware architecture may be adjusted to support the desired system capacity. For example, the RDBMS may be separated from the Web Application.

## Software Architecture

This section describes the overall software architecture of the system. Previous sections described the interface to external components, the SQL Database and Site VistAs, so this section concentrates on the architecture of the Web Application. For the purposes of this section, the ASRC VistA patch is considered part of VistA and is not described.

The below diagram illustrates the software architecture of the Web Application.



Figure 12 - Software Architecture

As with any web application, parts of the application run on the server and parts run on the client. The server-side components run exclusively in a Java application server, while the client-side components are a mixture of HTML5, CSS3, and JavaScript. The below sections detail these components.

### The Server-Side Components

The server-side components run exclusively in a Java application server. The prototype application runs in the GlassFish Server, Open Source Edition, provided by Oracle, but does not rely on any GlassFish-specific components and therefore is portable to other application servers.

The Java application running on the server contains four layers as shown above: the Presentation Layer, the Service Layer, the Domain Model, and the Data Layer. A summary of these layers follows, but the detailed design documentation resides in the Java source code (mostly in package Javadocs) to avoid becoming outdated.

The Domain Model is the core of the application and represents the application entities (calculations, variables, terms, etc.) as Java objects, incorporating both behavior and data. For more detail, see the Javadoc on the package gov.va.med.srcalc.domain[[1]](#footnote-2).

The Service Layer serves as a high-level, business-operation-oriented interface from the outside world to the Domain Model. For more detail, see the Javadoc on the package gov.va.med.srcalc.service[[2]](#footnote-3).

The Presentation Layer presents the application data to users (whether humans or machines) and, as appropriate, allows those users to modify the data. For more detail, see the Javadoc on the package gov.va.med.srcalc.web[[3]](#footnote-4).

The Data Layer maps between objects in the Domain Model and data stores. The application, as shown above, has two data stores: the SQL database and the site VistAs. For more detail former, see the Javadoc on the packages gov.va.med.srcalc.db[[4]](#footnote-5) and gov.va.med.srcalc.vista[[5]](#footnote-6).

### The Client-Side Components

The client-side components run exclusively in a user’s web browser. Unlike the Java Application Server, many different users—and therefore many different browser instances—may run the client-side code concurrently.

As with many web applications, server provides content to the browser primarily via HTML. This HTML simply defines page content, not the appearance. The HTML pages link to Cascading Style Sheets (CSS) in order to define the appearance. For example, the HTML may contain a table of values, but the CSS determines the size of the table, what borders will be shown between cells, etc.

Some, but not all, pages also include dynamic content such as dialog boxes. The HTML pages either include inline JavaScript or link to JavaScript files to implement this dynamic content. The application performs all Document Object Model (DOM) manipulation via the ubiquitous jQuery library.

Although many browsers can run the application, the target browser for the prototype is Internet Explorer 9. The prototype also supports Internet Explorer 8, albeit with reduced functionality, to enable testing in the VA Future Technology Lab.

## Service Oriented Architecture / ESS

ASRC does not provide or consume any services.

## Enterprise Architecture

As a prototype, the tool is not required to adhere to the VA Technical Reference Model (TRM). The development team, however, has attempted to adhere to it. The below table captures the various technologies used and their status in the TRM.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technology** | **Version** | **TRM Status** | **TRM Version** | **Comment** |
| DataTables | 1.10 | No | v15.1 |  |
| EqualsVerifier | 1.5 | No | v14.10 | Only used for automated tests. |
| Glassfish | 3.1.2.2 | Deprecated | v15.1 |  |
| Gradle | 1.12 | Approved | v15.1 |  |
| Hibernate | 4.2 | No | v15.1 | v4.3 is Approved |
| HSQLDB | 2.3.2 | No | v14.10 | Only used for automated tests. |
| Jackson (JSON) | 2.3 | Approved | v15.1 |  |
| Java EE | 7 | Approved | v15.1 |  |
| Java SE | 7 | Approved | v15.1 |  |
| Joda Time | 2.3 | Approved | v15.1 |  |
| jQuery | 1.11 | Approved w/ Constraints | v15.1 |  |
| jQuery UI | 1.11 | No | v15.1 |  |
| Junit | 4.12 | Approved | v15.3 |  |
| Log4j | 1.2.17 | Deprecated | v15.1 | Only used for automated tests. |
| Mockito | 1.9.5 | Approved | v15.1 |  |
| MySQL Database | 5.6 | Approved w/ Constraints | v15.1 |  |
| Selctivizr | 1.0.2 | No | V15.3 | Only for IE8 support. May be dropped for VA release. |
| SLF4J | 1.7 | Approved w/ Constraints | v15.1 |  |
| Spring Framework | 4.0.7 | Approved w/ Constraints | v15.1 | Will be deprecated soon. |
| Spring Security | 3.2.5 | No | v15.1 | Only version 3.1 is approved. We use version 3.2 for Spring 4 compatibility. |

# Data Design

This section outlines the design of the database management system (DBMS) and non-DBMS files associated with the system. For networks, detail the distribution of data and identify any changes to the logical data model that may occur due to software or hardware requirements.

Note: Provide a data dictionary appendix showing data element name, type, length, source, validation rules, maintenance, data stores, outputs, aliases, and description.

## DBMS Files

If a database will be used list and describe the logical requirements that exist for data formats, storage capabilities, data retention, data integrity, etc.

Describe how the database will be designed, including the following information, as appropriate:

* Logical model; provide normalized table layouts, entity relationship diagrams, and other logical design information
* DBMS schemas, subschemas, records, sets, tables, storage page sizes
* Access methods (such as indexed, via set, sequential, random access, sorted pointer array)
* Estimate the database file size or volume of data within the file, data pages, including overhead resulting from access methods and free space
* Definition of the update frequency of the database tables, views, files, areas, records, and sets
* Estimates on the number of transactions that the database may have to process.

For a description of Entity Relationship diagramming please refer to the whitepaper <<http://www-106.ibm.com/developerworks/rational/library/content/03July/2500/2785/2785_uml.pdf>>

## Non-DBMS Files

* Describe all non-DBMS files including narratives on the usage of each file.
* Identify if the file is used for input, output, or both; identify temporary files, which modules read and write the file, and similar.
* Identify record structures, record keys, indices, and reference data elements within the records.
* Define record length and blocking factors.
* Define the file access method such as: index sequential, virtual sequential, random access.
* Estimate the file size or volume of data within the file.
* Define the update frequency of the file if appropriate. Provide the estimated number of transactions per unit time and the statistical mean, mode, and distribution of those transactions.

# Detailed Design

This section describes the proposed design in detail. Provide the necessary information for the development team to integrate the hardware components and write the software code, so that the hardware and software components will provide a functional product. This is the detailed design, based upon the conceptual design (high level) that was described in the document up to this point. Most sections prior to this are needed for Milestone 1, on a best effort basis as the design is visualized and refined. This section is needed for Milestone 2, this is where the design in described in the conceptual sections is refined and an in depth detailed design is documented.

Note: Every design item should map back to a requirement.

## Hardware Detailed Design

The information requested in this section maybe provided by Engineering and/or the Developers. The information provided here is mainly for use by Engineering and Operations.

In this section, provide enough information for the developers to build and/or procure the system’s hardware. The level of detail requested should be treated as a general guideline and can be omitted if it needs to be filled in by Engineering and Operations.

Note: If this section becomes too lengthy, consider incorporating it as an appendix or reference it in a separate document. Add additional diagrams, if necessary, to describe each component and its functions.

Include the following information (as applicable):

* How much compute capacity? (MFLOPS, TPMs etc.)
* System Memory
* Local and Shared storage
* Network requirements (Bandwidth, Latency etc.)
* Public or Private cloud

## Software Detailed Design

This section provides conceptual and final detailed information associated with the design of the software being delivered. This should be an extension of the corresponding section from Section 3.1, but should contain additional detail as the project progresses.

### Database Repository

The Database Repository section in the RSD can be referenced in this section.

If a logical database design is a part of the system, it should be listed here. Logical database design should specify the logical requirements for any information that is to be placed into a database. This may include:

* Types of information used by various functions
* Frequency of use
* Accessing capabilities
* Data entities and their relationships
* Integrity constraints
* Data retention requirements.

Recommendation: Create a block diagram showing the databases and where the data resides.

### Java Web Application Detailed Design

The design element tables are provided for your convenience. Copy each table as many times as necessary to address multiple items within each section. Add rows and headings to the tables to provide any additional required information to define the item or to specify the modifications to the item. Numbering of the design element tables to align them underneath the applicable requirement or sub-requirement is recommended, but is left to the author’s discretion. For that reason, they are not numbered in this template.

Insert tables for:

* Frameworks
* Other Libraries
* Packages
* Classes
* Methods
* Integration Tests
* Unit Tests

### VistA Patch Detailed Design

The design element tables are provided for your convenience. Copy each table as many times as necessary to address multiple items within each section. Add rows and headings to the tables to provide any additional required information to define the item or to specify the modifications to the item. Numbering of the design element tables to align them underneath the applicable requirement or sub-requirement is recommended, but is left to the author’s discretion. For that reason, they are not numbered in this template.

#### Routines (Entry Points)

This section is an illustration that is VistA specific. The authors are free to organize this information by technology, different templates, or optional sections depending on the task at hand.

Complete the table for each routine affected by the functionality being designed.

Table 7 - Routines (Instructions)

| Routines | Instructions |
| --- | --- |
| **Routine Name** | SRASRC |
| **Enhancement Category** | New |
| **RTM** | N/A |
| **Related Options** | SR ASRC PATIENT (RPC), SR ASRC USER (RPC) |
| **Related Routines** | $$GET1^DIQ (Called by), DEM^VADPT (Called by) |
| **Data Dictionary (DD) References** | File 200 |
| **Related Protocols** | N/A |
| **Related Integration Control Registrations (ICRs)** | List proposed new ICRs and subscribed ICRs. Also, list any obscure Supported ICRs. |
| **Data Passing** | This routine is used as an RPC to pass data to a web application via the RPC Broker and VistALink. |
| **Input Attribute Name and Definition** | The patient "DFN" will be passed into the PAT tag (for SR ASRC PATIENT) as an input parameter. |
| **Output Attribute Name and Definition** | For both RPCs, there is a "RETURN" variable that will contain the single string of data being passed to the web application. |
| **Current Logic** | N/A |
| **Modified Logic (Changes are in bold)** | N/A |

|  |  |
| --- | --- |
| Routines | Instructions |
| **Routine Name** | SRASRC2 |
| **Enhancement Category** | New |
| **RTM** | N/A |
| **Related Options** | SR ASRC PROGRESS NOTE (RPC) |
| **Related Routines** | This routine calls the following APIs: DUZ^XUP, MAKE^TIUSRVP, SIGN^TIUSRVP2, $$DECRYP^XUSRB1, HASH^XUSHSHP, $$GET1^DIQ |
| **Data Dictionary (DD) References** | Files 2 and 8925.1 |
| **Related Protocols** | N/A |
| **Related Integration Control Registrations (ICRs)** | Supported ICRs: 2240, 3535, and 4409. |
| **Data Passing** | This routine is used as an RPC that receives data from a web application via the RPC Broker and VistALink and stores it into VistA. The RPC then returns status indication back to the web application. |
| **Input Attribute Name and Definition** | DUZ, SRESIG, DFN, and SRTIUX are all input parameters for the RPC. |
| **Output Attribute Name and Definition** | SRSTAT, which is the status indicator returned to the calling web application. |
| **Current Logic** | N/A |
| **Modified Logic (Changes are in bold)** | N/A |

|  |  |
| --- | --- |
| Routines | Instructions |
| **Routine Name** | SRASRC3 |
| **Enhancement Category** | New |
| **RTM** | N/A |
| **Related Options** | *SR ASRC RISK SAVE (RPC)* |
| **Related Routines** | This routine calls the following: ^%DT and FILE^DICN |
| **Data Dictionary (DD) References** | Files 2, 81, 200, and 136.1 |
| **Related Protocols** | N/A |
| **Related Integration Control Registrations (ICRs)** | N/A |
| **Data Passing** | This routine is used as an RPC that receives data from a web application via the RPC Broker and VistALink and stores it into VistA. The RPC then returns status indication back to the web application. |
| **Input Attribute Name and Definition** | *DFN, SRCPT, SRDTTM, and SRTIUX are all input parameters for the RPC.* |
| **Output Attribute Name and Definition** | *SRSTAT, which is the status indicator returned to the calling web application.* |
| **Current Logic** | N/A |
| **Modified Logic (Changes are in bold)** | N/A |

Table 8 - Routines (Grouping)

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  |  |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** |  | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **Data Passing** | Input | Output Reference | Both | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | |

| Current Logic |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
|  |

#### Templates

Complete Table 16 for each template affected by the functionality being designed. A short description of what change will be made to the templates should be included in this section.

Note: If preferred, copy and paste this section directly from VA FileMan DDs instead of using the tables.

Table 9 - Templates (Instructions)

| Templates | Instructions |
| --- | --- |
| **Template Name** | Identify the template affected by the functionality being designed |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **RSD Traceability** | List the Requirement Specification Document (RSD) item number within the SDD (i.e., If the RSD has a requirement of 3.3.1, add Support for a new API, then this column should list RSD Requirement 3.3.1) |
| **Template Type** | Indicate the type of template identified (Sort, Input, or Print). |
| **Related Options** | List options that directly call or are called by the template. |
| **Related Routines** | List routines that directly call or are called by the template. |
| **Data Dictionary (DD) References** | List files/fields that reference the template(s) through input transforms, and cross reference logic. |
| **Global References** | List the ICRs for global references that are outside your namespace. |

Table 10 - Templates

| Templates | Description | | | |
| --- | --- | --- | --- | --- |
| **Template Name** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RSD** |  | | | |
| **Template Type** | Sort | Input | Print | Other |
| **Related Options** |  | | | |

| **Related Routines** | **Routines “Called By”** | **Routines “Called”** |
| --- | --- | --- |
|  |  |  |

| Routines | Description |
| --- | --- |
| **Data Dictionary (DD) References** |  |
| **Global References** |  |

#### Data Entries Affected by the Design

Provide the following data for each field to be created, modified, or deleted or provide a “Before and After: Data Entries Affected by the Design.”

Identify the entries affected by the design. If a blanket change will be made to each entry affected, that change should be defined in this table.

Only changes that are unique to each record should be defined in the Unique Record(s) section (Section 6.2.2.3.5). Redundant information should not be entered into each chart in the Unique Record(s) section.

Table 11 - Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| PATIENT  136.01,.01 | NA | Patient ID from the Patient File (#2) |
| DATE/TIMESTAMP  136.01,1 | NA | Date and Time that the surgical risk calculation was run. |
| CPT CODE  136.01,2 | NA | CPT Code from the CPT File (#81) |
| AUTHOR  136.01,3 | NA | User from the New Person file (#200) who is running the surgical risk calculation. |
| RISK CALCULATION MODEL  136.01,4 | NA | This multiple field contains all surgical risk model names and calculated probabilities for a given surgical specialty. |
| RISK MODEL  136.014,.01 | NA | A free text field containing the surgical risk model name |
| RISK PROBABILITY  136.014,1 | NA | The risk probability (percentage) for a given surgical risk model |

#### The following is a complete FileMan listing of the new file:

STANDARD DATA DICTIONARY #136.1 -- SURGICAL RISK CALCULATIONS FILE

MAY 14,2015@12:40:12 PAGE 1

STORED IN ^SRO(136.1, (15 ENTRIES) SITE: VEHU MASTER UCI: GOLD,ROU

DATA NAME GLOBAL DATA

ELEMENT TITLE LOCATION TYPE

-------------------------------------------------------------------------------

DD ACCESS: @

RD ACCESS: @

WR ACCESS: @

DEL ACCESS: @

LAYGO ACCESS: @

AUDIT ACCESS: @

(NOTE: Kernel's File Access Security has been installed in this UCI.)

IDENTIFIED BY: DATE/TIMESTAMP (#1)

CROSS

REFERENCED BY: PATIENT(B)

INDEXED BY: PATIENT & DATE/TIMESTAMP (C)

CREATED ON: APR 21,2015 by PROGRAMMER,ONE

136.1,.01 PATIENT 0;1 POINTER TO PATIENT FILE (#2)

(Required)

LAST EDITED: MAY 13, 2015

HELP-PROMPT: NAME MUST BE 3-30 CHARACTERS, NOT NUMERIC OR

STARTING WITH PUNCTUATION

DESCRIPTION:

Patient ID from the Patient File (#2)

CROSS-REFERENCE: 136.1^B

1)= S ^SRO(136.1,"B",$E(X,1,30),DA)=""

2)= K ^SRO(136.1,"B",$E(X,1,30),DA)

RECORD INDEXES: C (#816)

136.1,1 DATE/TIMESTAMP 0;2 DATE

INPUT TRANSFORM: S %DT="E" D ^%DT S X=Y K:X<1 X

LAST EDITED: MAY 13, 2015

HELP-PROMPT: (No range limit on date)

DESCRIPTION: Date and Time that the surgical risk

calculation was run.

RECORD INDEXES: C (#816)

136.1,2 CPT CODE 0;3 POINTER TO CPT FILE (#81)

LAST EDITED: MAY 12, 2015

HELP-PROMPT: Enter a valid CPT Code

DESCRIPTION:

CPT Code from the CPT File (#81)

136.1,3 AUTHOR 0;4 POINTER TO NEW PERSON FILE (#200)

LAST EDITED: MAY 12, 2015

HELP-PROMPT: Enter name of the user running risk calcuation

DESCRIPTION: User from the New Person file (#200) who is

running the surgical risk calculation.

136.1,4 RISK CALCULATION MODEL 1;0 Multiple #136.14

(Add New Entry without Asking)

LAST EDITED: APR 23, 2015

DESCRIPTION: This multiple field contains all surgical risk

model names and calculated probabilities for a

given surgical specialty.

136.14,.01 RISK MODEL 0;1 FREE TEXT

INPUT TRANSFORM: K:$L(X)>99!($L(X)<1) X

LAST EDITED: MAY 12, 2015

HELP-PROMPT: Answer must be 1-99 characters in length.

DESCRIPTION: A free text field containing the surgical

risk model name

CROSS-REFERENCE: 136.14^B

1)= S ^SRO(136.1,DA(1),1,"B",$E(X,1,30),DA)=""

2)= K ^SRO(136.1,DA(1),1,"B",$E(X,1,30),DA)

136.14,1 RISK PROBABILITY 0;2 NUMBER

INPUT TRANSFORM: K:+X'=X!(X>100)!(X<0)!(X?.E1"."7.N) X

LAST EDITED: MAY 12, 2015

HELP-PROMPT: Type a number between 0 and 100, 1 decimal

digit.

DESCRIPTION: The risk probability (percentage) for a given

surgical risk model

FILES POINTED TO FIELDS

CPT (#81) CPT CODE (#2)

NEW PERSON (#200) AUTHOR (#3)

PATIENT (#2) PATIENT (#.01)

File #136.1

Record Indexes:

C (#816) RECORD REGULAR IR LOOKUP & SORTING

Short Descr: INDEX BY PAT & DT/TM

Set Logic: S ^SRO(136.1,"C",X(1),X(2),DA)=""

Kill Logic: K ^SRO(136.1,"C",X(1),X(2),DA)

Whole Kill: K ^SRO(136.1,"C")

X(1): PATIENT (136.1,.01) (Subscr 1) (forwards)

X(2): DATE/TIMESTAMP (136.1,1) (Subscr 2) (forwards)

INPUT TEMPLATE(S):

PRINT TEMPLATE(S):

SORT TEMPLATE(S):

FORM(S)/BLOCK(S):

#### Unique Record(s)

List the unique record ID(s) that will be affected by the changes implemented by the design. This is commonly done in the .01 field. The values defined in the Current Value and New Value columns should be the exact value of the data. For each unique record ID, copy this table and provide the information.

Table 12 - Unique Record ID

| Field Name(s) | Current Value | New Value |
| --- | --- | --- |
|  |  |  |

#### File or Global Size Changes

Indicate the change to the size of the file or global as a result of the design implemented with this description. Global size changes tie back to the business requirements and RSD. Growth or reduction in the size of the global should be indicated in this section. If the file is static across all VistA systems, a blanket statement of how the change will affect the size of the global will suffice.

For example, “The National Procedure file is a new file and will require 8.7K of disk space to install.”

If a file is dynamic and its size may vary from VistA system to VistA system, the description should indicate the change in the file per record and the number of records that the site may anticipate. For example, if a field is being added to the patient file that will result in an increase of 7K per patient, the site can estimate the global growth based on the number of entries in that file.

Note: If the Capacity Planning analysis is available, then enter it here. If not, then use the Project Team projection.

Table 13 - File or Global Size Changes

| File/Global Name(s) | Estimated Increase | Estimated Decrease |
| --- | --- | --- |
|  |  |  |

#### Mail Groups

Complete the table for each of the mail groups affected by the functionality being designed. A short description of what changes will be made to the affected mail groups should be included in this section.

Note: If preferred, this can be captured directly from VA FileMan DDs after the fact.

Table 14 - Mail Groups (Instructions)

| Mail Groups | Instructions |
| --- | --- |
| **Mail Group Name** | List the name of the mail group being modified. The mail group name may include a domain name. |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **Related Options** | List options that directly reference the file. |
| **Related Routines** | List routines that reference the mail group. |
| **Data Dictionary (DDs) References** | List files that reference the mail group through input transforms, cross-reference logic, etc. |
| **Related Protocols** | List protocols that directly reference the mail group. |
| **Mail Group Description** | Describe the purpose for the mail group. |
| **Self-Enrollment Allowed** | Check the appropriate box either Yes or No. |
| **Type** | Check the appropriate box either Public or Private. |

Table 15 - Mail Groups

| Mail Groups | Activities | | | |
| --- | --- | --- | --- | --- |
| **Mail Group Name** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  |  |

| Mail Groups | Instructions | |
| --- | --- | --- |
| **Data Dictionary (DD) References** |  | |
| **Related Protocols** |  | |
| **Mail Group Description** |  | |
| **Self-Enrollment Allowed** | Yes | No |
| **Type** | Public | Private |

#### Security Keys

This section lists the specific security keys affected by the functionality being designed. A short description of the changes that will be made to the security keys affected should be included in this section.

Note: If preferred, this can be captured directly from VA FileMan DDs after the fact.

Table 16 - Security Keys (Instructions)

| Security Keys | Instructions |
| --- | --- |
| **Security Key Name** | List the specific name of the security key being modified. |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **Related Options** | List options that directly reference the security key. |
| **Related Routines** | List routines that reference the security key. |
| **Data Passing** | Check the appropriate box. Enter a short description of an event that would trigger the new/changed routine, for example, a note that the change to the security key will be referenced through user menu driven options, routines, etc. This section refers specifically to the change implemented with the design. |
| **Security Key Description** | List a brief description of the security key. |
| **Subordinate Keys** | List any subordinate keys. |
| **Mutually Exclusive Keys** | Enter the name of a key that may not be held jointly with this one. |
| **Granting Condition Logic** | Define the logic for the Granting Condition of the Security Key affected by the functionality being designed. |
| **Current Logic** | If the security key currently has a granting condition, define the current logic for that granting condition. If the security key did not exist before, indicate that there is currently no security key. |
| **Modified Logic  (Changes are in bold)** | Define the granting condition that the design will implement. If the security key is new to the field, define the logic here. |
| **Hierarchical Precedence** | Define which key is used if one key will take precedence over another key. |

Table 17 - Security Keys

| Security Keys | Activities | | | |
| --- | --- | --- | --- | --- |
| **Security Key Name** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  |  |

| Security Keys | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Security Key Description** |  | | | | |
| **Subordinate Keys** |  | | | | |
| **Mutually Exclusive Keys** |  | | | | |
| **Granting Condition Logic** |  | | | | |

| Current Logic |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
|  |

| Security Keys | Activities |
| --- | --- |
| **Hierarchical Precedence** |  |

#### Options

Complete the table for each of the options affected by the functionality being designed. A short description of the changes that will be made to the options affected should be included. Changes to the OPTION file (#19) are to be included, not the functionality of the option invoked.

Note: If preferred, this can be captured directly from VA FileMan DD after the fact.

Table 18 - Options (Instructions)

| Options | Instructions |
| --- | --- |
| **Option Name  (MENU TEXT field)** | SR ASRC |
| **Enhancement Category** | New |
| **Associated Menu Options that will invoke this reference** | N/A |
| **Data Passing** | The purpose of this menu option is for RPC Context setting in a VistA web application that uses VistALink to communicate with VistA. |
| **Menu Text Description** | ASRC RPC MENU |
| **Option Type** | Broker (Client/Server) |
| **Option Definition** | The following options (RPCs) will be contained within this option: GMV EXTRACT REC, GMV LATEST VM, SR ASRC USER, SR ASRC PATIENT, SR ASRC PROGRESS NOTE, SR ASRC RISK SAVE |
| **Current Entry Action Logic** | N/A |
| **Modified Entry Action Logic (Changes are in bold)** | N/A |
| **Current Exit Action Logic** | N/A |
| **Modified Exit Action Logic  (Changes are in bold)** | N/A |

Table 19 - Options

| Options | Activities | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Option Name** |  | | | | | | | | | | |
| **Enhancement Category** | New | Modify | | | | Delete | | | No Change | | |
| **Associated Menu Options that will invoke this reference** |  | | | | | | | | | | |
| **Data Passing** | Input | | Output | | Both | | | Global Reference | | | Local Reference |
| **Menu Text Description** |  | | | | | | | | | | |
| **Option Type** | Edit | | | Print | | | Menu | | | Inquire | |
| Action | | | Run Routine | | | Other | | |  | |
| **Associated Routine** |  | | | | | | | | | | |
| **Option Definition** |  | | | | | | | | | | |

| Current Entry Action Logic |
| --- |
|  |

| Modified Entry Action Logic (Changes are in bold) |
| --- |
|  |

| Current Exit Action Logic |
| --- |
|  |

| Modified Exit Action Logic (Changes are in bold) |
| --- |
|  |

#### Protocols

Complete the table for each of the protocols affected by the functionality being designed. A short description of the changes that will be made to the protocols affected should be included in this section. Changes to the PROTOCOL file (#101) are to be included, not the functionality of the protocol invoked.

Note: If preferred, this can be captured directly from VA FileMan DDs after the fact.

Table 20 - Protocols (Instructions)

| Protocols | Instructions |
| --- | --- |
| **Protocol Name** | List the name of the protocol affected. |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **Associated Protocols** | List the ancestors of the protocol being designed, i.e., those protocols that contain the respective protocol as an item. |
| **Data Passing** | Check the appropriate box. An event that would trigger the new/changed protocol should be included in this section. An example would be a note that the change to the protocol will be referenced through the VA event driver, List Manager, user selection of a protocol from the VA Kernel Menu Management system. This section refers specifically to the change implemented with the design. |
| **Item Text Description** | Enter the protocol's text as it appears to the user on the menu or sub-header. |
| **Protocol Type** | Define the type of protocol to be executed |
| **Associated Routine** | List any associated routines affected by the protocol being designed. |
| **Current Entry Action Logic** | Define the current logic for the entry action of the protocol affected by the functionality being designed. If the entry action did not exist before, indicate that there currently is no entry action. |
| **Modified Entry Action Logic  (Changes are in bold)** | Define the entry action that the design will implement. If the entry action is new to the field, define the logic here. |
| **Current Exit Action Logic** | Define the current logic for the exit action of the protocol affected by the functionality being designed. If the exit action did not exist before, indicate that there currently is no exit action. |
| **Modified Exit Action Logic  (Changes are in bold)** | Define the exit action that the design will implement. If the exit action is new to the field, define the logic here. |

Table 21 - Protocols

| Protocols | Activities | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** |  | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **Associated Protocols** |  | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | N/A | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** |  | | | | | | | | | | |

| Current Entry Action Logic |
| --- |
|  |

| Modified Entry Action Logic (Changes are in bold) |
| --- |
|  |

| Current Exit Action Logic |
| --- |
|  |

| Modified Exit Action Logic (Changes are in bold) |
| --- |
|  |

#### Remote Procedure Call (RPC)

Complete the table for each RPC affected by the functionality being designed.

Note: If preferred, this can be captured directly from VA FileMan DDs after the fact.

Table 22 - RPCs (Instructions)

| RPCs | Instructions |
| --- | --- |
| **Name** | SR ASRC PATIENT |
| **TAG^RTN** | PAT^SRASRC |
| **Input Parameters** | DFN |
| **Results Array** | RETURN($J) = "Patient Name from Patient file (#2)" ^ "Patient's Age" ^ "Patient's Gender (M or F)" |
| **Description** | Based on the DFN sent as the input parameter, this RPC returns the name of the patient, along with age and gender. |

|  |  |
| --- | --- |
| RPCs | Instructions |
| **Name** | SR ASRC PROGRESS NOTE |
| **TAG^RTN** | ENTER^SRASRC2 |
| **Input Parameters** | DFN, DUZ, SRESIG, SRTIUX |
| **Results Array** | SRSTAT = "1 for successful or 0 for unsuccessful" ^ "Success message or error message" |
| **Description** | This RPC has two functions. First, it checks to see if a valid electronic signature code was entered by the user. If the signature was successful, it will attempt to create a new Progress Notes record for the Surgical Risk Calculations data. |

|  |  |
| --- | --- |
| RPCs | Instructions |
| **Name** | SR ASRC RISK SAVE |
| **TAG^RTN** | ENTER^SRASRC3 |
| **Input Parameters** | DFN, SRCPT, SRDTTM, SRTIUX |
| **Results Array** | SRSTAT = "1 for successful or 0 for unsuccessful" ^ "Success message or error message" |
| **Description** | This RPC attempts to store Surgical Risk Calculation data for a given patient into the Surgical Risk Calculations file (#136.1) in VistA. |

|  |  |
| --- | --- |
| RPCs | Instructions |
| **Name** | SR ASRC USER |
| **TAG^RTN** | USER^SRASRC |
| **Input Parameters** | N/A |
| **Results Array** | *RETURN($J) = "User name from New Person file (#200)"* |
| **Description** | Based on the DUZ of the current VistALink session, this RPC returns the name of the user. |

Table 23 - RPCs

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** |  | | |
| **TAG^RTN** |  | | |
| **Input Parameters** |  | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** |  | | |

## Service Oriented Architecture / ESS Detailed Design

This section provides details of provided and consumed services as follows:

* Consumed Services: Provide link to Service Description Document for each consumed service.
* Provided Services: Give service design for each provided service.

The information you provide here will be used to upload to the ESS Registry and Repository. At some point in the near future, we do not expect these SOA artifacts such as SLA, Service Description, etc. to be static documents. They will be dynamically generated from the ESS Registry and Repository tool in the form of reports. Any application and service integration design is also documented here.

A list of currently available Enterprise Shared Services is available here: <insert link to ESS list>

### Service Description for <Consumed Service Name>

Provide link to Service Description document for the consumed service. This section will repeat for each consumed service. The Service Description includes Service Interface and Service Level Definition (SLD) to address anticipated capacity requirements.

### Service Design for <Provided Service Name>

This section should describe the detailed service design for each ESS and SOA service needed to obtain an intended result. The Service Design includes Service Interface and Service Level Definition (SLD) to address anticipated capacity requirements.

This section will repeat for each **provided** service.

# External System Interface Design

This section details interfaces external to system, that are NOT services (ESS/SOA). Typically, these may include, RPCs, Flat Data Files etc.

External systems are systems that are not within the scope of the system under development, regardless of whether the other systems are managed by the vendor or its client.

In this section, describe the interface(s) between the system under development (i.e., the system that is the subject of this SDD) and external systems and/or subsystem(s).

It is best to illustrate these sections with annotated diagrams to clearly identify the various elements of the interfaces.

## Interface Architecture

Describe the interface(s) between the system being designed and other systems. Include the interface architecture(s) being implemented, such as wide area networks, gateways, etc. Provide diagrams showing the communications path(s) between this system and other systems.

## Interface Detailed Design

Provide sufficient detail about the interface requirements for the development team to format, transmit, and/or receive data across the interface.

Include the following information (as appropriate):

* Data format requirements; if data must be reformatted before it is transmitted or after incoming data is received. Describe the tools and/or methods for the reformat process.
* Specifications for hand-shaking protocols between systems; content and format of hand-shake messages, timing for exchanging these messages, and errors handling.
* Format(s) for reports exchanged between the systems.
* Graphical representation of the connectivity between systems, showing the direction of data flow.
* Query and response descriptions.
* Describe the individual data elements that the interfacing entity(s) will provide, store, send, access, and receive, such as:
* Names/identifiers
  + Data Element Name
  + Data Format/Length
  + Data Type
  + Definition
  + Non-Technical Name
  + Non-Technical Synonyms
  + Specifications
  + Synonyms
* Range or enumeration of possible values (e.g., 0-99)
* Accuracy and precision (number of significant digits)
* Priority, timing, frequency, sequencing, and other constraints
* Security and privacy constraints
* Sources (setting/sending entities) and recipients (using/receiving entities).

Describe the data element assemblies (records, messages, files etc.) that the interfacing entity(s) will provide, store, and send, such as:

* Names/identifiers
  + Technical Name, e.g., data structure name
  + Non-technical Names, e.g. synonyms
* Data elements
* Medium/structure of data elements/assemblies
* Visual characteristics (e.g. layouts, fonts, icons etc.)
* Relationships among assemblies
* Security and privacy constraints
* Sources and recipients.

Describe the communication methods that the interfacing entity(s) will use for the interface, such as:

* Communication links, bands, frequencies, and media
* Message formatting
* Flow control (e.g. sequence numbering)
* Data transfer rate
* Routing
* Transmission services
* Safety
* Security and privacy considerations.

Describe characteristics of the protocols that the interfacing entity(s) will use for the interface, such as:

* Priority/layer of the protocol
* Packeting
* Legality checks, error control
* Recovery procedures
* Synchronization
* Status, identification, and other reporting features.

Where appropriate describe other characteristics, such as physical compatibility of the interfacing entity(s) (dimensions, tolerances, loads, voltages, plug compatibility, etc.)

# Security and Privacy

## Security

Describe specific security mechanisms at the application level, as guided by NIST 800-53 revision 3 (or most current version). Also, summarize the security mechanisms to be provided by the VA GSSs. Reference the Security Risk Assessment.

The following information will be provided to address security controls:

A high-level description of the security controls, grouped according to the 18 control families identified in NIST 800-53 revision 3 (or most current version). A description of all 18 control families must be addressed; if a control family is not applicable, then state that control family does not apply and explain why it does not apply.

A description of the specific security controls that will be provided by existing VA infrastructure or VA GSSs.

Describe the planned use by the application of the infrastructure’s centralized security mechanisms and VA GSSs (in particular, the identification and authentication, access control, and audit mechanisms), and infrastructure mechanisms, (e.g., Directory Services) to store user account information. Sufficient detail should be provided to show the feasibility of the integration and/or inter operation of application security mechanisms with infrastructure security mechanisms.

## Privacy

Identify privacy design considerations. Describe specific privacy mechanisms at the application. Describe how the application’s privacy requirements will be met. Reference the System Security Plan (SSP) and Privacy Impact Assessment (PIA).

1. Additional Information

Attach any addition information that supplements the design specification.

* 1. RTM

Include an RTM that traces modules and data structures to the software requirements. A reference to the location of the RTM is also acceptable.

* 1. Packaging and Installation

Outline any special considerations for software packaging and installation.

* 1. Design Metrics

Describe all metrics to be used during the design activity.

* 1. Acronym List and Glossary

Identify and define all acronyms and terms that establish meaning within the context of the plan.

Table 24 - Glossary

| Term | Meaning |
| --- | --- |
|  |  |
|  |  |
|  |  |

See TOGAF® 9.1, Part III: ADM Guidelines & Techniques, Gap Analysis on TOGAF website at <http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap27.html>

1. Available online at: <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/domain/package-summary.html> [↑](#footnote-ref-2)
2. Available online at: <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/service/package-summary.html> [↑](#footnote-ref-3)
3. Available online at: <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/web/package-summary.html> [↑](#footnote-ref-4)
4. Available online at: <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/db/package-summary.html> [↑](#footnote-ref-5)
5. Available online at: <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/vista/package-summary.html> [↑](#footnote-ref-6)