Department of Veterans Affairs

**Automated Surgical Risk Calculator (ASRC)**

Technical Manual



**September 2015**

Version 0.08

Revision History

| Date | Version | Description | Author | Reviewer | Issue Date |
| --- | --- | --- | --- | --- | --- |
| 11/25/2014 | 0.01 | Document Shell | D. Tombs | S. Vetzel | 11/25/2014 |
| 12/01/2014 | 0.02 | Partial Completion | D. Tombs | S. Vetzel | 12/17/2014 |
| 04/23/2015 | 0.03 | Define Administrative Pages and the Software Architecture section | D. Tombs |  |  |
| 05/13/2015 | 0.04 | Initial update for VistA Patch Detailed Design | J. Swesky | S. Ambrose | 05/18/2015 |
| 06/12/2015 | 0.05 | Update VistA section for Sprint 8 | J. Swesky | S. Ambrose | 06/12/2015 |
| 07/09/2015 | 0.06 | Update Enterprise Architecture  Update VistA Routines section for Sprint 9 | D. Tombs  J. Swesky | S. Ambrose | 07/14/2015 |
| 08/05/2015 | 0.07 | Update VistA sections for Sprint 10 | D. Tombs  J. Swesky | S. Ambrose | 08/11/2015 |
| 09/02/2015 | 0.08 | Update VistA sections for Sprint 11. Populate all remaining sections. | J. Swesky  D. Tombs  J. McLane | S. Ambrose | 09/07/2015 |

Table of Contents

[1 Introduction 1](#_Toc429416233)

[1.1 Identification 1](#_Toc429416234)

[1.2 Scope 1](#_Toc429416235)

[1.3 User Characteristics 1](#_Toc429416236)

[1.4 References 1](#_Toc429416237)

[2 Background 2](#_Toc429416238)

[2.1 Overview of the System 2](#_Toc429416239)

[2.2 Overview of the Business Process 2](#_Toc429416240)

[2.3 Business Benefits 2](#_Toc429416241)

[2.4 Assumptions and Constraints 2](#_Toc429416242)

[2.4.1 Design Assumptions 2](#_Toc429416243)

[2.4.2 Design Constraints 3](#_Toc429416244)

[2.4.3 Design Trade-offs 3](#_Toc429416245)

[2.5 Overview of the Significant Requirements 3](#_Toc429416246)

[2.5.1 Overview of Significant Functional Requirements 3](#_Toc429416247)

[2.5.2 Overview of Functional Workload / Performance Requirements 3](#_Toc429416248)

[2.5.3 Overview of Operational Requirements 4](#_Toc429416249)

[2.5.4 Overview of the Technical Requirements 4](#_Toc429416250)

[2.5.5 Overview of the Security or Privacy Requirements 4](#_Toc429416251)

[2.5.6 Overview of System Criticality and High Availability Requirements 4](#_Toc429416252)

[3 Conceptual Design 4](#_Toc429416253)

[3.1 Conceptual Application Design 5](#_Toc429416254)

[3.1.1 Application Context 5](#_Toc429416255)

[3.1.2 High-Level Application Design 5](#_Toc429416256)

[3.2 Conceptual Data Design 7](#_Toc429416257)

[3.2.1 Project Conceptual Data Model 7](#_Toc429416258)

[3.2.2 UI Data Mapping 7](#_Toc429416259)

[3.2.2.1 Calculation Pages 8](#_Toc429416260)

[3.2.2.1.1 New Calculation Page 9](#_Toc429416261)

[3.2.2.1.2 Enter Risk Variables Page 10](#_Toc429416262)

[3.2.2.1.3 Calculation Results Page 11](#_Toc429416263)

[3.2.2.2 Administrative Pages 12](#_Toc429416264)

[3.2.2.2.1 Administration Home 12](#_Toc429416265)

[3.2.2.2.2 Edit Variable Pages 13](#_Toc429416266)

[3.2.2.2.3 Edit Rule Page 14](#_Toc429416267)

[3.2.2.2.4 Edit Model Page 15](#_Toc429416268)

[3.2.2.3 Administrative Reports 16](#_Toc429416269)

[3.2.2.4 Unmapped Data Element 16](#_Toc429416270)

[4 System Architecture 16](#_Toc429416271)

[4.1 Hardware Architecture 17](#_Toc429416272)

[4.2 Software Architecture 17](#_Toc429416273)

[4.2.1 The Server-Side Components 18](#_Toc429416274)

[4.2.2 The Client-Side Components 19](#_Toc429416275)

[4.3 Service Oriented Architecture (SOA) / Electronic Safety and Security (ESS) 20](#_Toc429416276)

[4.4 Enterprise Architecture 20](#_Toc429416277)

[5 Data Design 21](#_Toc429416278)

[5.1 Database Managements System (DBMS) Files 22](#_Toc429416279)

[5.2 Non-DBMS Files 22](#_Toc429416280)

[6 Detailed Design 23](#_Toc429416281)

[6.1 Software Detailed Design 23](#_Toc429416282)

[6.1.1 Java Web Application Detailed Design 23](#_Toc429416283)

[6.1.1.1 Third Party Frameworks 23](#_Toc429416284)

[6.1.1.2 Design Patterns 23](#_Toc429416285)

[6.1.1.3 Code Conventions 24](#_Toc429416286)

[6.1.2 VistA Patch Detailed Design 24](#_Toc429416287)

[6.1.2.1 Routines (Entry Points) 24](#_Toc429416288)

[6.1.2.2 Templates 42](#_Toc429416289)

[6.1.2.3 Data Entries Affected by the Design 42](#_Toc429416290)

[6.1.2.4 Unique Record(s) 46](#_Toc429416291)

[6.1.2.5 File or Global Size Changes 46](#_Toc429416292)

[6.1.2.6 Mail Groups 47](#_Toc429416293)

[6.1.2.7 Security Keys 47](#_Toc429416294)

[6.1.2.8 Options 48](#_Toc429416295)

[6.1.2.9 Protocols 49](#_Toc429416296)

[6.1.2.10 RPC 50](#_Toc429416297)

[6.2 SOA / ESS Detailed Design 52](#_Toc429416298)

[7 External System Interface Design 52](#_Toc429416299)

[7.1 Interface Architecture 52](#_Toc429416300)

[7.2 Interface Detailed Design 53](#_Toc429416301)

[8 Security and Privacy 54](#_Toc429416302)

[8.1 Security 54](#_Toc429416303)

[8.1.1 User Authentication 54](#_Toc429416304)

[8.1.1.1 Clinical User Authentication 55](#_Toc429416305)

[8.1.1.2 Administrative User Authentication 55](#_Toc429416306)

[8.2 Privacy 56](#_Toc429416307)

[A. Additional Information 57](#_Toc429416308)

[A.1 RTM 57](#_Toc429416309)

[A.2 Packaging and Installation 57](#_Toc429416310)

[A.3 Design Metrics 57](#_Toc429416311)

[A.4 Acronym List and Glossary 57](#_Toc429416312)

List of Figures

[Figure 1 - Application Context Diagram 5](#_Toc429416313)

[Figure 2 - High-Level Application Design 6](#_Toc429416314)

[Figure 3 - Conceptual Data Model 7](#_Toc429416315)

[Figure 4 - Application Clinical Workflow 9](#_Toc429416316)

[Figure 5 - Notional New Calculation Page 10](#_Toc429416317)

[Figure 6 - Notional Enter Risk Variables Page 11](#_Toc429416318)

[Figure 7 - Notional Calculation Results Page 12](#_Toc429416319)

[Figure 8 - Notional Administration Home, Partial 13](#_Toc429416320)

[Figure 9 - Notional Edit Discrete Numerical Variable 14](#_Toc429416321)

[Figure 10 - Notional Edit Rule Page 15](#_Toc429416322)

[Figure 11 - Notional Edit Model Page 16](#_Toc429416323)

[Figure 12 - ASRC Hardware Components 17](#_Toc429416324)

[Figure 13 - Software Architecture 18](#_Toc429416325)

[Figure 14 - Simplified Risk Model Relations 21](#_Toc429416326)

[Figure 15 - Signed Calculation Relations 22](#_Toc429416327)

[Figure 16 - Completed Calculation Schema 22](#_Toc429416328)

[Figure 17 - VistALink Communication 53](#_Toc429416329)

[Figure 18 - Security Design 54](#_Toc429416330)

List of Tables

[Table 1 - Significant Functional Requirements 3](#_Toc429416331)

[Table 2 - Technical Requirements 4](#_Toc429416332)

[Table 3 - Security or Privacy Requirements 4](#_Toc429416333)

[Table 4 - External System Interfaces 5](#_Toc429416334)

[Table 5 - Externally Shared Data Stores 5](#_Toc429416335)

[Table 6 - Objects in the High Level Application Design 6](#_Toc429416336)

[Table 7 - TRM Status 20](#_Toc429416337)

[Table 8 - Routines (Grouping) 24](#_Toc429416338)

[Table 9 - Templates 42](#_Toc429416339)

[Table 10 - Data Entries Affected by the Design 42](#_Toc429416340)

[Table 11 - Unique Record ID 46](#_Toc429416341)

[Table 12 - File or Global Size Changes 46](#_Toc429416342)

[Table 13 - Mail Groups 47](#_Toc429416343)

[Table 14 - Security Keys 47](#_Toc429416344)

[Table 15 - Options 48](#_Toc429416345)

[Table 16 - Protocols 49](#_Toc429416346)

[Table 17 - RPCs 50](#_Toc429416347)

[Table 18 - User Group Authentication and Authorization 55](#_Toc429416348)

[Table 19 - Glossary 57](#_Toc429416349)

# 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 the Veterans Health Information Systems and Technology Architecture (VistA) patch for VistA integration. This document is meant to accompany version 0.11 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 ASRC 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 Structured Query Language (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.

## References

* The ASRC User Guide[[1]](#footnote-2).
* The ASRC README[[2]](#footnote-3).

# Background

## Overview of the System

The ASRC 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 VistA SQL database for use by the National Surgery Office (NSO).

## Overview of the Business Process

## 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 CPRS and that its current location is not well known to most surgical providers.

The ASRC Tool provides the following benefits:

* It integrates the Tool into the Electronic Health Record (EHR) environment for easy access by Department of Veterans Affairs (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. (Medical Domain 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

The development team made the following trade-offs when designing the Tool:

* The team favored flexibility of Risk Model configuration over simplicity of the configuration. (See Section 3.2.2.2.)
* The team favored maintainability of the Tool over performance.
* The team favored Tool functionality over performance.
* The team favored Tool reliability (producing correct results) over robustness (gracefully handling error conditions).
* The team favored Tool functionality over robustness.

## 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 only handles test patient data, no Personally Identifiable Information (PII)/Protected Health Information (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 6.1.2 documents the VistA patch design. This section therefore focuses on the Java Web Application (“the application”).

## 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 in Section 2.4.2.



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 RPCs to VistA from Java. | Site VistAs |
| Web Application Frontend | Presents a Web User Interface (UI) 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.

### UI Data Mapping

Users retrieve and modify application data through three UIs: 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 Section 1.3 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.

Notational New Calculation Page


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.

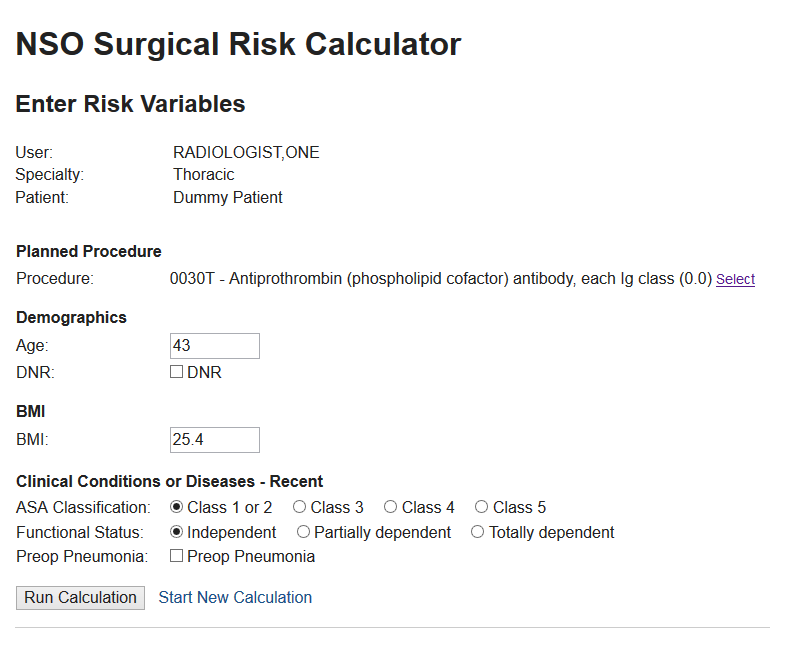


Figure 6 - Notional Enter Risk Variables Page

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.

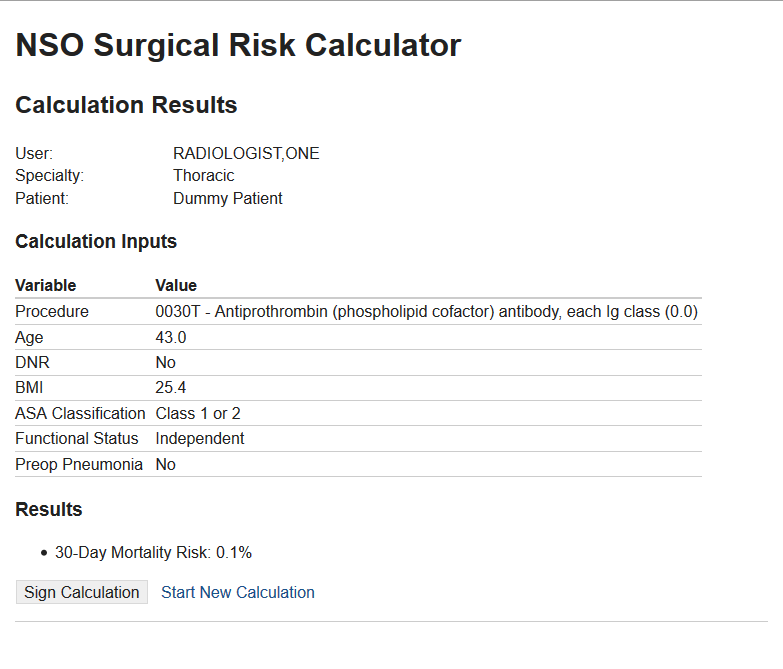


Figure 7 - Notional Calculation Results Page

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 Text Integration Utility (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 Section 1.3 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.

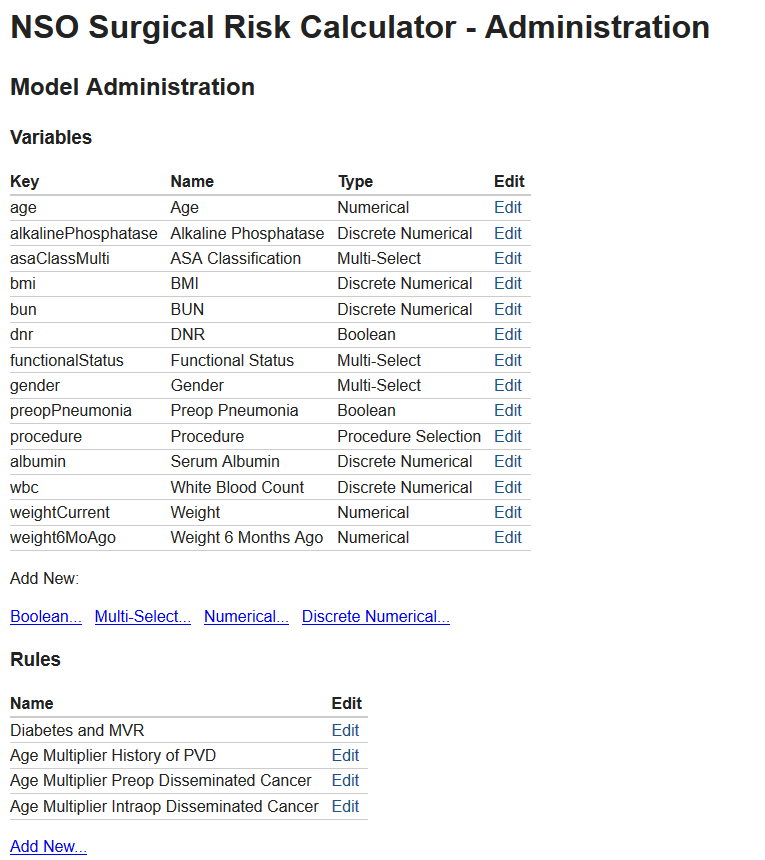


Figure 8 - Notional Administration Home, Partial

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:

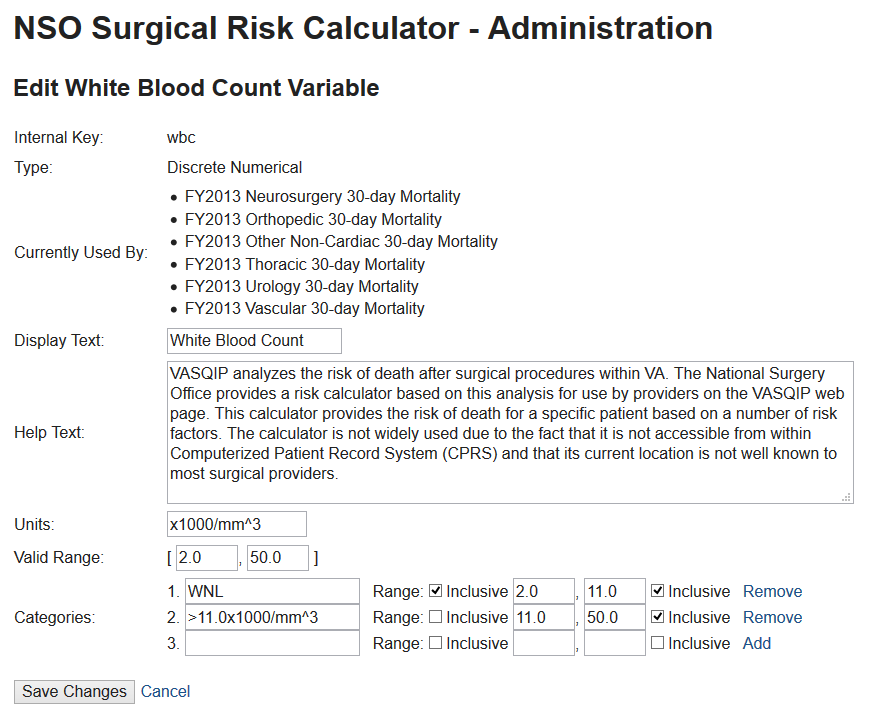


Figure 9 - Notional Edit Discrete Numerical Variable

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.

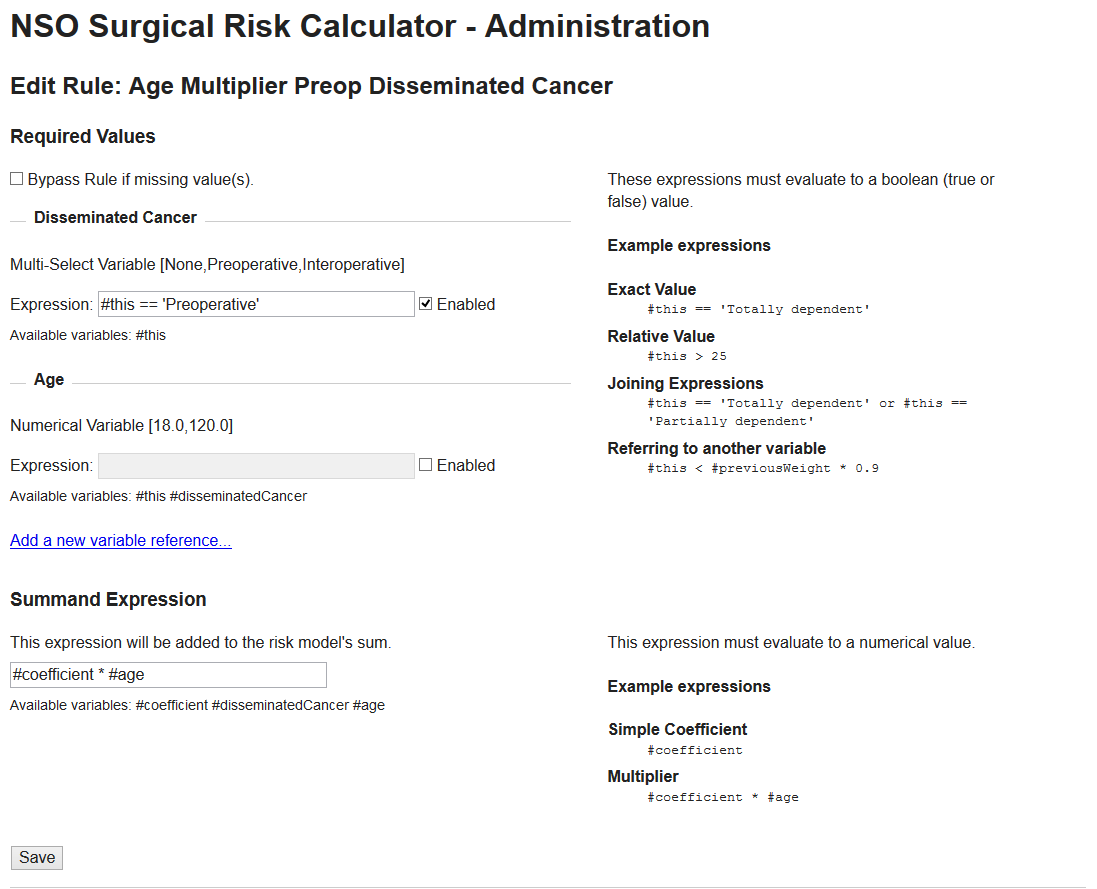


Figure 10 - Notional Edit Rule Page

##### Edit Model Page

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

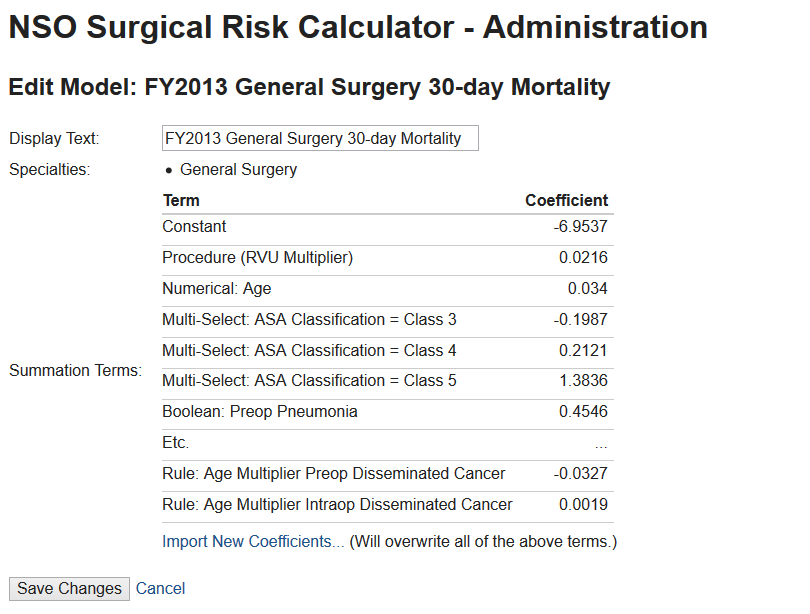


Figure 11 - Notional Edit Model Page

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 Current Procedural Terminology (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 12 - 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 Relational Database Managements System (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 diagram below illustrates the software architecture of the Web Application.



Figure 13 - 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 HyperText Markup Language (HTML) 5, Cascading Style Sheets (CSS) 3, 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[[3]](#footnote-4).

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[[4]](#footnote-5).

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[[5]](#footnote-6).

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[[6]](#footnote-7) and gov.va.med.srcalc.vista[[7]](#footnote-8).

### 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 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 include either the 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 FTL.

## Service Oriented Architecture (SOA) / Electronic Safety and Security (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.

Table 7 - TRM Status

| Technology | Version | TRM Status | TRM Version | Comment |
| --- | --- | --- | --- | --- |
| Apache Commons Comma-Separated Value (CSV) | 1.1 | Not Listed | V15.6 | Could not find any CSV library on the TRM. |
| Data Tables | 1.10 | Not Listed | v15.1 |  |
| Equals Verifier | 1.5.1 | Not Listed | v14.10 | Only used for automated tests. |
| Glassfish | 3.1.2.2 | Deprecated | v15.1 |  |
| Gradle | 1.12 | Approved | v15.1 |  |
| Hibernate | 4.2 | Not Listed | v15.1 | v4.3 is Approved |
| HSQLDB | 2.3.2 | Not Listed | v14.10 | Only used for automated tests. |
| Jackson (JSON) | 2.3 | Approved | v15.1 |  |
| Jadira Userype Core | 4.0.0.GA | Not Listed | v15.7 |  |
| Java Platform Enterprise Edition (Java EE) | 7 | Approved | v15.1 |  |
| Java Platform Standard Edition (Java SE) | 7 | Approved | v15.1 |  |
| Joda Time | 2.7 | Approved | v15.1 |  |
| Joda Time Java Server Pages (JSP) Tags | 1.1.1 | Not Listed | v15.7 |  |
| jQuery | 1.11 | Approved w/ Constraints | v15.1 |  |
| jQuery UI | 1.11 | Not Listed | 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 | Not Listed | 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 | Not Listed | v15.1 | Only version 3.1 is approved. We use version 3.2 for Spring 4 compatibility. |

# Data Design

The application is dynamic in the data that is presented to the user. As such, nearly all application data is stored in the remote database. This allows the tool to be completely configurable based on what the administrative users deem accurate.

Although the resultant schema is more detailed and exact than Figure 14, the figure does show the general structure of class relations for risk models. It should be noted that there are multiple Variable types as well as multiple RiskModelTerms. The relations shown in the figure are translated to the database schema via Java persistence annotations and Hibernate’s mapping. Figure 15 is much the same, except that it refers to previously completed calculations. The calculations are stored whether a user signed them.Simplified Risk Model Relations


Figure 14 - Simplified Risk Model Relations

Signed Calculation Relations


Figure 15 - Signed Calculation Relations

## Database Managements System (DBMS) Files

The type of database being used is MySQL. Since the application uses Hibernate as an Object Relational Mapping (ORM), the code to create the database and its tables is produced by Hibernate. The schema for completed calculations is shown in Figure 16. One way to generate visual schemas is by using the reverse engineering feature of MySQL Workbench.Completed Calculation Schema


Figure 16 - Completed Calculation Schema

## Non-DBMS Files

| File Name | Type | Description |
| --- | --- | --- |
| messages.properties | Input | Contains message strings for external display. The messages are used for UI display to communicate messages to the user. |
| insert\_dummy\_models.sql | Input | This file is used for testing. All JUnit tests will use this file for test data instead of using the actual MySQL database. This file must be updated when the database’s schema is update to ensure all Junit tests run properly. |

# Detailed Design

## Software Detailed Design

### Java Web Application Detailed Design

#### Third Party Frameworks

For a comprehensive list of frameworks and third-party software, see Table 7. For any java package, class, or method documentation refer to the generated Javadocs on Github[[8]](#footnote-9). Regarding the java packages, classes should be placed into the proper packages in order to improve the overall design of the application. The application uses the Hexagonal Design Pattern in order to separate the domain of the application from the adapters that it interacts with, such as the web view. See Figure 2. As such, classes that are related to the web portion of the application belong in the web package and classes related to the database belong in the database package.

Selectivizr is a Javascript utility that allows Internet Explorers 6-8 to emulate CSS3. As this application’s target browser is IE 8, Selectivizr is necessary in several places to ensure that compatibility is maintained with the look and feel of the application. If compatibility is desired, Selectivizr should be used in future development.

The primary interface to VistA is the VistALink software package.

VistALink is a core library used by the application as the application communicates with VistA exclusively via VistALink. See Section 7 for more information.

#### Design Patterns

The automated testing used in the application is the Junit test package. Although the Junit tests are labeled as in another package, they do not need to import classes that have the same path except for the “main” or “test” path. For instance, CalculationTest.java is in the same package as Calculation.java and as such does not need to import Calculation.java.

Data Access Objects (DAO) are primarily used to provide an interface to the database. In the application there are currently many DAOs used to access the database for a variety of reasons. Some examples are VariableDao, SpecialtyDao, and ResultsDao. Each DAO has a specific purpose and should be kept to only providing an access point for the type of queries stated in the documentation for that class.

The use of Spring Controllers helps to facilitate RESTful patterns in the web application. Although each controller may contain multiple GET and POST methods, each method will by definition relate to a single GET or POST call if annotated properly. The POST/redirect/GET pattern is observed in the application in order to prevent multiple problems, including duplicate form submission.

#### Code Conventions

Google’s Guava provides many libraries that help to enhance the core functionality of the Java language. This includes functionality for collections, input and output, validation, and more. Two of the most commonly used functionalities from Guava are immutable collections and optional objects. Immutable collections are used extensively throughout the application as a means to provide the programmer with a collection that is certain not to change. For instance, when variable groups are returned from the database, they are gathered from the database first and then put into an ImmutableCollection to be returned so that the programmer cannot add or remove items from the collection. Optional variables can also be extremely useful in cleaning up code that previously used null values when a variable is not initialized. Each instance of AbstractVariable has a fHelpText field that may or may not have information available for that field. Normally the help text would be left as a null if there was no information available, but by using a com.google.common.base.Optional<String> instance instead of a String instance the fHelpText field may never be null.

Object-oriented design patterns are prevalent throughout the application’s source code. Ideas such as encapsulation and polymorphism help to improve the cohesion of code as well as making it easier to modify in future development work.

In the spirit of encapsulation, there are many references in the code comments to “DRY” or “DRYing” out the code (Don’t Repeat Yourself). Not only does this cut down on the size of files, but also it helps with readability and understanding in the code. Whenever possible, avoid code duplication and elongated methods.

Inheritance is also utilized heavily in the application. The most prominent example of inheritance in the application would be the different types of variables that are displayed for the user during a risk calculation. Each variable inherits from a common base class, and that allows for each derived class to share common behavior, while in turn deriving behavior specific to that class.

Polymorphism is another object-oriented design pattern that allows for flexibility in several spots in the application’s source code. Several Visitor patterns take advantage of polymorphism to decide what behavior should occur at runtime. This removes duplicated code that would normally take place in branching statements and cuts down on code complexity. Since the application deals with many different derived classes where the derived class decides the behavior rather than the calling code, polymorphism should be applied as a design pattern.

### VistA Patch Detailed Design

#### Routines (Entry Points)

Table 8 - Routines (Grouping)

| Routines | Activities |
| --- | --- |
| Routine Name | SRASRC |
| Enhancement Category | New |
| Requirements Traceability Matrix (RTM) | N/A |
| Related Options | SR ASRC PATIENT (RPC) |
| Related Routines | EN^DIQ1 (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) | SRASRC ;SBX/JAS - RETURN VISTA PATIENT/USER DATA ; 11/17/14  ;;3.0;Surgery;\*\*\*\*;24 Jun 93;Build 2  PAT(RETURN,DFN) ;  K RETURN  D DEM^VADPT  S RETURN($J)=VADM(1)\_"^"\_VADM(4)\_"^"\_$P(VADM(5),"^")  K VADM  Q  CLASS(RETURN) ;  K RETURN,SRUC,SRCNT,^UTILITY("DIQ1",$J)  Q:$G(DUZ)=""!('$D(^VA(200,DUZ)))!('$D(^VA(200,DUZ,"USC1")))  S SRUC=""  F S SRUC=$O(^VA(200,DUZ,"USC1",SRUC)) Q:SRUC="" I $D(^VA(200,DUZ,"USC1",SRUC,0)) D  . S DIC=200,DR="8932.1",DA=DUZ  . S DR(200.05)=".01;3",DA(200.05)=SRUC  . D EN^DIQ1  Q:'$D(^UTILITY("DIQ1",$J,200.05))  S SRUC="",SRCNT=0  F S SRUC=$O(^UTILITY("DIQ1",$J,200.05,SRUC)) Q:SRUC="" I $D(^UTILITY(  "DIQ1",$J,200.05,SRUC,.01)) D  . Q:^UTILITY("DIQ1",$J,200.05,SRUC,3)'=""  . S SRCNT=SRCNT+1  . S RETURN(SRCNT)=^UTILITY("DIQ1",$J,200.05,SRUC,.01)  Q |

| Routines | Activities |
| --- | --- |
| Routine Name | SRASRC2 |
| Enhancement Category | New |
| RTM | N/A |
| Related Options | SR ASRC PROGRESS NOTE (RPC) |
| Related Routines | This routine calls the following Application Programming Interfaces (APIs): DUZ^XUP, MAKE^TIUSRVP, SIGN^TIUSRVP2, $$DECRYP^XUSRB1, HASH^XUSHSHP, $$GET1^DIQ |
| DD References | Files 2 and 8925.1 |
| Related Protocols | N/A |
| Related 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) | SRASRC2 ;SBX/JAS - RPC TO STORE ASRC PROGRESS NOTE ; 03/27/15  ;;3.0;Surgery;\*\*\*\*;24 Jun 93;Build 2  ;  ; Reference to API DUZ^XUP supported by IA# 4409  ; Reference to API MAKE^TIUSRVP supported by IA# 3535  ; Reference to API $$ENCRYP^XUSRB1 supported by IA# 2240  ; Reference to API SIGN^TIUSRVP2 supported by IA# 3535  ;  ENTER(SRSTAT,DUZ,SRESIG,DFN,SRTIUX) ;  ;  ; SRSTAT = (by ref) TIU DOCUMENT # (PTR to 8925)  ; = 0^Explanatory message if not a success  ; DUZ = New Person (#200)  ; SRESIG = Electronic Signature Code  ; DFN = Patient (#2)  ; SRTIUX = (by ref) array containing field data and document body  ;  N SRVALID,SRTITLE,SRTITDA  D VALIDATE  ;  I '+SRVALID S SRSTAT="0^Incorrect Electronic Signature Code Entered." Q  I '$D(DUZ(0))!('$D(DUZ(2))) D DUZ^XUP(DUZ)  ; Visit information not relevant to risk calculation  S (SRVDT,SRVLOC,SRVSIT,SRVSTR,SRSUPP,SRNOA)=""  ; Progress note title associated with surgical risk calculation  S SRTITLE="SURGERY NOTES"  S SRTITDA=$O(^TIU(8925.1,"B",SRTITLE,""))  I SRTITDA="" S SRSTAT="0^Invalid Progress Note Title." D QUIT Q  S SRTIUX(1202)=DUZ  ;  D MAKE^TIUSRVP(.SRDOC,DFN,SRTITDA,SRVDT,SRVLOC,SRVSIT,.SRTIUX,SRVSTR,SRSUPP,SRNOA)  ;  ; DFN = Patient (#2)  ; SRTITDA = TIU Document Definition (#8925.1)  ; SRVDT = Date(/Time) of Visit  ; SRVLOC = Visit Location (HOSPITAL LOCATION)  ; SRVSIT = Visit file ien (#9000010)  ; SRTIUX = (by ref) array containing field data and document body  ; SRVSTR = Visit string (i.e., VLOC;VDT;VTYPE)  ; SRSUPP = Indicates whether or not to suppress execution of COMMIT AC  TION  ; SRNOA = if 1=Do Not Set ASAVE cross-reference  ;  K SRTIUX  I '+SRDOC S SRSTAT=SRDOC D QUIT Q  S SRTIUDA=SRDOC  S SRTIUX=SRESIG  ;  D SIGN^TIUSRVP2(.SRSTAT,SRTIUDA,SRTIUX)  ;  ; SRSTAT = Result of Signature attempt on Note  ; 1 for successful signature, 0 for failed attempt  ; SRTIUDA = TIU DOCUMENT # (PTR to 8925)  ; SRTIUX = Encrypted e-signature  ;  I SRSTAT'=0 S SRSTAT="0^Progress note could not be signed at this time.  " D QUIT Q  S SRSTAT="1^Progress note was created and signed successfully."  D QUIT  Q  ;  VALIDATE ; Validate Electronic Sign code  ;  ; SRESIG = Contains the signature to be validated  ; DUZ = New Person (#200) record to validate e-sig against  ; SRVALID = Contains a 1 if a valid e-sig (0 if not valid),  ; user's sig block print name, and signature block title  ;  S SRVALID=0 N SRDESIG  I SRESIG="" Q  ; Decrypt esig and pass to Hashing routine for file 200 validation  S SRDESIG=$$DECRYP^XUSRB1(SRESIG)  S X=$G(SRDESIG)  D HASH^XUSHSHP  I X'="",(X=$$GET1^DIQ(200,DUZ,20.4,"I")) D  .S SRVALID=1  .S SRVALID=SRVALID\_U\_$$GET1^DIQ(200,DUZ,20.2)\_U\_$$GET1^DIQ(200,DUZ,20.3)\_U\_"EOL999"  .Q  K X,SRDESIG  Q  QUIT  K SRDOC,SRVALID,SRESIG,SRTIUDA,SRTITLE,SRTITDA  K SRVDT,SRVLOC,SRVSIT,SRVSTR,SRSUPP,SRNOA  Q |

| Routines | Activities |
| --- | --- |
| 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 |
| DD References | Files 2, 81, 200, and 136.1 |
| Related Protocols | N/A |
| Related 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) | SRASRC3 ;SBX/JAS - RPC TO STORE ASRC RESULTS IN VISTA ; 04/30/15  ;;3.0;Surgery;\*\*\*\*;24 Jun 93;Build 2  ;  ENTER(SRSTAT,DFN,SRCPT,SRDTTM,SRTIUX) ;  ;  ; \*\* RPC input/output parameter definitions \*\*  ;  ; Input Parameters:  ; DFN = Patient ien (file #2 pointer)  ; SRCPT = CPT code (file #81)  ; SRDTTM = Date/Timestamp in Mumps format  ; SRTIUX = Array containing - Model Name ^ Risk Prob.  ; SRTIUX(1) = "FY2013 Cardiac 30-Day^56.0"  ; SRTIUX(2) = "FY2013 Cardiac 90-Day^10.0"  ; DUZ = User ien (file #200 pointer) - passed via context sharing  ;  ; Output Parameter:  ; SRSTAT = 1^Success message, if record was successfully created  ; = 0^Explanatory message, if not a successful  ;  ;  ; \*\* Data validation section for input parameters \*\*  ;  K SRSTAT,SRDA,SRMOD,SRPROB  S SRSTAT=1  ; User check  I $G(DUZ)="" S SRSTAT="0^User DUZ is not defined." Q  I '$D(^VA(200,DUZ,0)) S SRSTAT="0^User DUZ is not valid." Q  ; Patient check  I $G(DFN)="" S SRSTAT="0^Patient DFN is not defined." Q  I '$D(^DPT(DFN,0)) S SRSTAT="0^Patient DFN is not valid." Q  ; CPT check  ;I $G(SRCPT)="" S SRSTAT="0^CPT Code is not defined." Q  ;I '$D(^ICPT("B",SRCPT)) S SRSTAT="0^CPT Code is not valid." Q  I $G(SRCPT)="" S SRCPT=""  I SRCPT'="",'$D(^ICPT("B",SRCPT)) S SRSTAT="0^CPT Code is not valid." Q  I SRCPT'="",$D(^ICPT("B",SRCPT)) S SRCPT=$O(^ICPT("B",SRCPT,""))  ; Date/Time check  I $G(SRDTTM)="" S SRSTAT="0^Date/Time is not defined." Q  S X=SRDTTM,%DT="NR" D ^%DT  I Y="-1" S SRSTAT="0^Date/Time format is not valid." Q  S SRDTTM=Y  ; Risk data check  I '$D(SRTIUX) S SRSTAT="0^Model and Risk data is not defined." Q  S SRDA="" F S SRDA=$O(SRTIUX(SRDA)) Q:SRDA=""!('SRSTAT) I $D(SRTIUX(SRDA)) D  . S SRMOD=$P(SRTIUX(SRDA),"^",1),SRPROB=$P(SRTIUX(SRDA),"^",2)  . I $L(SRMOD)>99 S SRSTAT="0^Model name is greater than 99 characters."  Q  . I SRPROB'?0N.2N1"."1N S SRSTAT="0^Risk probability is not a valid per  centage." Q  Q:'SRSTAT  ;  ; \*\* Create new risk calculation record entry \*\*  ;  K DO  S DIC="^SRO(136.1,"  S DIC(0)="",DA=136.1  S X=DFN  S DIC("DR")="1////^S X=SRDTTM;2////^S X=SRCPT;3////^S X=DUZ"  D FILE^DICN  I Y=-1 S SRSTAT="0^Record creation was unsuccessful." Q  ;  D MULT  S SRSTAT="1^Record created successfully."  Q  ;  ; \*\* Store risk model data in multiple field \*\*  ;  MULT ;  K DO,SRDA,SRMOD,SRPROB  S DA(1)=+Y  S DIC=DIC\_DA(1)\_",1,"  S DIC(0)="L"  S SRDA="" F S SRDA=$O(SRTIUX(SRDA)) Q:SRDA="" I $D(SRTIUX(SRDA)) D  . K DIC("DR")  . S SRMOD=$P(SRTIUX(SRDA),"^",1),SRPROB=$P(SRTIUX(SRDA),"^",2)  . S DIC("DR")=".01////^S X=SRMOD;1////^S X=SRPROB"  . D FILE^DICN  Q |

| Routines | Activities |
| --- | --- |
| Routine Name | SRASRC4 |
| Enhancement Category | New |
| RTM | N/A |
| Related Options | SR ASRC LAB RESULTS (RPC) |
| Related Routines | This routine calls the following: RESULTS^LRPXAPI, $$TESTNM^LRPXAPIU, H^%DTC |
| DD References | N/A |
| Related Protocols | N/A |
| Related ICRs | N/A |
| Data Passing | This routine is used as an RPC to pass data to a web application via the RPC Broker and VistALink. |
| Input Attribute Name and Definition | DFN and SRLABNS are input parameters for the RPC. |
| Output Attribute Name and Definition | SRRET, which is the data field returned to the calling web application. |
| Current Logic | N/A |
| Modified Logic (Changes are in bold) | SRASRC4 ;SBX/JAS - RETURN MOST RECENT VISTA LAB RECORD ; 05/21/15  ;;3.0;Surgery;\*\*\*\*;24 Jun 93;Build 2  ;  ; Reference to API RESULTS^LRPXAPI supported by IA# 4245  ; Reference to API $$TESTNM^LRPXAPIU supported by IA# 4246  ;  ENTER(SRRET,DFN,SRLABNS) ;  ;  ; SRRET = A match is found - Lab Name ^ Lab Result ^ Lab Date ^ Lab U  nits of Measure  ; = No matches found - "" is returned  ; DFN = Patient file (#2) ien  ; SRLABNS = An array containing the possible Lab strings to check for  ;  K SRRET S SRRET=""  I DFN=""!('$D(SRLABNS)) Q  D RESULTS^LRPXAPI(.SRLABR,DFN,"",250)  I '$D(SRLABR) Q  S SRLDA="",SRFND=0  F S SRLDA=$ZP(SRLABR(SRLDA)) Q:SRLDA=""!(SRFND) I $D(SRLABR(SRLDA)) D  . S SRLREC=SRLABR(SRLDA)  . S SRTEST=$$TESTNM^LRPXAPIU($P(SRLREC,"^",2))  . S SRNSDA=""  . F S SRNSDA=$O(SRLABNS(SRNSDA)) Q:SRNSDA="" I $D(SRLABNS(SRNSDA)) D  . . I SRLABNS(SRNSDA)=SRTEST S SRFND=1 D Q  . . . S X=$P(SRLREC,"^") D H^%DTC  . . . S SRDATE=$ZDATE(%H)\_"@"\_$ZTIME(%T)  . . . S SRRET=SRTEST\_"^"\_$P(SRLREC,"^",4)\_"^"\_SRDATE\_"^"\_$P($P(SRLREC,"  ^",8),"!",7)  K SRLABR,SRLDA,SRFND,SRLREC,SRTEST,SRNSDA,SRDATE  Q |

| Routines | Activities |
| --- | --- |
| Routine Name | SRASRCCK |
| Enhancement Category | New |
| RTM | N/A |
| Related Options | Make Operation Requests  Delete or Update Operation Requests |
| Related Routines | This routine calls the following: $$GET1^DIQ, %DTC |
| DD References | 81 AND 136.1 |
| Related Protocols | N/A |
| Related ICRs | N/A |
| Data Passing | N/A |
| Input Attribute Name and Definition | N/A |
| Output Attribute Name and Definition | N/A |
| Current Logic | N/A |
| Modified Logic (Changes are in bold) | SRASRCCK ;SBX/JAS - ASRC CHECK ; 07/02/15  ;;3.0;Surgery;\*\*\*\*;24 Jun 93;Build 2  ;  Q:$D(^SRO(137,SRASRCC))  K SRASRCD,SRASRCX,SRASRCDT,SRASRCR,SRASRCF,SRASRCRM,SRLINE  S SRLINE="" F I=1:1:80 S SRLINE=SRLINE\_"="  S SRASRCX=0,SRASRCD=$$GET1^DIQ(81,SRASRCC,.01)  I '$D(^SRO(136.1,"C",DFN,SRASRCC)) S SRASRCX=1  S SRASRCDT="",SRASRCF=0  F S SRASRCDT=$ZP(^SRO(136.1,"C",DFN,SRASRCC,SRASRCDT)) Q:SRASRCDT=""!(SRASRCX)!(SRASRCF) I $D(^SRO(136.1,"C",DFN,SRASRCC,SRASRCDT)) D  . S X1=DT,X2=SRASRCDT D ^%DTC I X>60 S SRASRCX=1 Q  . S SRASRCR=""  . F S SRASRCR=$ZP(^SRO(136.1,"C",DFN,SRASRCC,SRASRCDT,SRASRCR)) Q:SRASRCR=""!(SRASRCF) I $D(^SRO(136.1,SRASRCR,1,0)) D  . . S SRASRCF=1  . . W !!!,SRLINE,!!,?12,"PATIENT'S MORTALITY RISK ASSOCIATED WITH CPT "\_SRASRCD  . . W !,?12,"--------------------------------------------------"  . . S SRASRCRM=""  . . F S SRASRCRM=$O(^SRO(136.1,SRASRCR,1,SRASRCRM)) Q:SRASRCRM="" I $D(^SRO(136.1,SRASRCR,1,SRASRCRM,0)) D  . . . W !,?5,$E($P(^SRO(136.1,SRASRCR,1,SRASRCRM,0),"^"),1,50)  . . . W ?65,$P(^SRO(136.1,SRASRCR,1,SRASRCRM,0),"^",2)\_" %"  . . W !!,SRLINE,!!  I SRASRCX D  . W !!!,SRLINE,!!,?3,"This request cannot be entered because the procedure "\_SRASRCD\_" you have"  . W !,?3,"selected requires an Automated Surgical Risk Calculator (ASRC) record"  . W !,?3,"result in the past 60 days. The requesting provider must complete and"  . W !,?3,"sign an ASRC mortality risk record for this patient and CPT code using"  . W !,?3,"the CPRS Tools menu - Automated Surgical Risk Calculator.",!!,SRLINE  . W !!,"Press RETURN once the calculation has been completed " R X:DTIME  . W !!!  . I $D(^SRO(136.1,"C",DFN,SRASRCC)) D  . . S SRASRCDT=$O(^SRO(136.1,"C",DFN,SRASRCC,"")) I SRASRCDT[DT D  . . . S SRASRCR=$O(^SRO(136.1,"C",DFN,SRASRCC,SRASRCDT,"")) I $D(^SRO(136.1,SRASRCR)) D  . . . . W !!!,SRLINE,!!,?12,"PATIENT'S MORTALITY RISK ASSOCIATED WITH C  PT "\_SRASRCD  . . . . W !,?12,"--------------------------------------------------"  . . . . S SRASRCRM=""  . . . . F S SRASRCRM=$O(^SRO(136.1,SRASRCR,1,SRASRCRM)) Q:SRASRCRM="" I $D(^SRO(136.1,SRASRCR,1,SRASRCRM,0)) D  . . . . . W !,?5,$E($P(^SRO(136.1,SRASRCR,1,SRASRCRM,0),"^"),1,50)  . . . . . W ?65,$P(^SRO(136.1,SRASRCR,1,SRASRCRM,0),"^",2)\_" %"  . . . . W !!,SRLINE,!!  Q |

| Routines | Activities |
| --- | --- |
| Routine Name | SRASRC5 |
| Enhancement Category | New |
| RTM | N/A |
| Related Options | SR ASRC HEALTH FACTORS (RPC) |
| Related Routines | This routine calls the following: LIST^ORQQVS, H^%DTC, $$GET1^DIQ |
| DD References | N/A |
| Related Protocols | N/A |
| Related ICRs | N/A |
| Data Passing | This routine is used as an RPC to pass data to a web application via the RPC Broker and VistALink. |
| Input Attribute Name and Definition | DFN is input parameter for the RPC. |
| Output Attribute Name and Definition | SRRET, which is the data field returned to the calling web application. |
| Current Logic | N/A |
| Modified Logic (Changes are in bold) | SRASRC5 ;SBX/JAS - RETURN HEALTH FACTORS FOR PATIENT ; 07/15/15  ;;3.0;Surgery;\*\*\*\*;24 Jun 93;Build 2  ;  ; Reference to API LIST^ORQQVS supported by IA# 1690  ; Reference to ^AUPNVHF supported by IA# 3084  ;  ENTER(SRRET,DFN) ;  ;  ; SRRET = If records are found - Date of Visit ^ Health Factor name  ; = No records found - "" is returned  ; DFN = Patient file (#2) ien  ;  K SRRET,STDT,SRORY,SRVISDT,SRVISIT,SRACNT,SRHFDA,SRHF S SRRET=""  I DFN="" Q  S %H=$H-365 D YMD^%DTC S STDT=X  D LIST^ORQQVS(.SRORY,DFN,STDT,DT,"")  Q:'$D(SRORY)  ;  S SRCNT=0,SRACNT=0  F S SRCNT=$O(SRORY(SRCNT)) Q:SRCNT="" I $D(SRORY(SRCNT)) D  . S X=$P(SRORY(SRCNT),"^",2) D H^%DTC  . S SRVISDT=$ZDATE(%H)  . S SRVISIT=$P(SRORY(SRCNT),"^")  . I $D(^AUPNVHF("AD",SRVISIT)) D  . . S SRHFDA="" F S SRHFDA=$O(^AUPNVHF("AD",SRVISIT,SRHFDA)) Q:SRHFDA="" I $D(^AUPNVHF(SRHFDA,0)) D  . . . S SRHF=$$GET1^DIQ(9000010.23,SRHFDA,.01,"E")  . . . S SRACNT=SRACNT+1  . . . S SRRET(SRACNT)=SRVISDT\_"^"\_SRHF  Q |

| Routines | Activities |
| --- | --- |
| Routine Name | SRASRC6 |
| Enhancement Category | New |
| RTM | N/A |
| Related Options | SR ASRC ADL NOTES (RPC), SR ASRC DNR NOTES (RPC) |
| Related Routines | This routine calls the following: SELECTED^VSIT, NOTES^TIUSRVLV, TGET^TIUSRVR1 |
| DD References | N/A |
| Related Protocols | N/A |
| Related ICRs | N/A |
| 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 | DFN and SRSTR are input parameters for the RPC. |
| Output Attribute Name and Definition | SRRET, which is the data field returned to the calling web application. |
| Current Logic | N/A |
| Modified Logic (Changes are in bold) | SRASRC6 ;SBX/JAS - RETURN PROGRESS NOTES FOR PATIENT ; 07/24/15  ;;3.0;Surgery;\*\*\*\*;24 Jun 93;Build 2  ;  ; Reference to API SELECTED^VSIT supported by IA# 1905  ; Reference to API NOTES^TIUSRVLV supported by IA# 2812  ; Reference to API TGET^TIUSRVR1 supported by IA# 2944  ;  ENTER1(SRRET,DFN,SRSTR) ;  ;  ; SRRET = If records are found - Full Progress Note is returned in XML format  ; = No records found - "" is returned  ; DFN = Patient file (#2) ien  ; SRSTR = Enterprise Title  ;  K SRRET  Q:DFN=""!(SRSTR="")  S SRTYPE=1  D ENTER(.SRRET,DFN,SRSTR,SRTYPE)  Q  ;  ENTER2(SRRET,DFN,SRSTR) ;  ;  ; SRRET = If records are found - Full Progress Note is returned in XML format  ; = No records found - "" is returned  ; DFN = Patient file (#2) ien  ; SRSTR = Search string for Enterprise Title  ;  K SRRET  Q:DFN=""!(SRSTR="")  S SRTYPE=2  D ENTER(.SRRET,DFN,SRSTR,SRTYPE)  Q  ;  ENTER(SRRET,DFN,SRSTR,SRTYPE) ;  K SRVST,SRNOTE,^TMP("VSIT",$J),SRCNT,SR2CNT,SRASRC,SRESB,SRTCNT S (SRCNT,SR2CNT)=0  ; TEST EXAMPLE: D SELECTED^VSIT(237,"","") RESULTS IN: ^TMP("VSIT",$J)  S SR2CNT=SR2CNT+1,SRRET(SR2CNT)="<notes>"  D SELECTED^VSIT(DFN,"","")  I '$D(^TMP("VSIT",$J)) D XQUIT Q  S SRVST=""  F S SRVST=$O(^TMP("VSIT",$J,SRVST)) Q:SRVST="" D  . K ^TMP("TIULIST",$J),SRNOTE,SRREVDT,SRREVDT2  . ; TEST EXAMPLE: D NOTES^TIUSRVLV(.SRNOTE,11312) RESULTS IN: ^TMP("TIULIST",$J)  . D NOTES^TIUSRVLV(.SRNOTE,SRVST)  . I $D(^TMP("TIULIST",$J)) D  . . S SRREVDT=""  . . F S SRREVDT=$O(^TMP("TIULIST",$J,SRREVDT)) Q:SRREVDT="" D  . . . S SRREVDT2="" F S SRREVDT2=$O(^TMP("TIULIST",$J,SRREVDT,SRREVDT2  )) Q:SRREVDT2="" I $D(^TMP("TIULIST",$J,SRREVDT,SRREVDT2)) D  . . . . S SRCNT=SRCNT+1  . . . . S SRASRC(SRREVDT,SRCNT)=$P(^TMP("TIULIST",$J,SRREVDT,SRREVDT2),"^")  K ^TMP("VSIT",$J),^TMP("TIULIST",$J)  I $D(SRASRC) D  . K ^TMP("TIUVIEW",$J),SRNOTED  . S SRREVDT=""  . F S SRREVDT=$O(SRASRC(SRREVDT)) Q:SRREVDT="" D  . . S SRCNT="" F S SRCNT=$O(SRASRC(SRREVDT,SRCNT)) Q:SRCNT="" D  . . . S SRTIU=SRASRC(SRREVDT,SRCNT)  . . . Q:SRTIU=""  . . . ; TEST EXAMPLE: D TGET^TIUSRVR1(.SRNOTED,11210) RESULTS IN: ^TMP("TIUVIEW",$J)  . . . D TGET^TIUSRVR1(.SRNOTED,SRTIU)  . . . I $D(^TMP("TIUVIEW",$J)) D  . . . . I SRTYPE=1 D  . . . . . K SRTBOD,SRLOC,SRSIGNDT,SRTIULN,SRETIT,SRBCNT S (SRESB,SRQUIT,SRBCNT)=0  . . . . . S SRTCNT="" F S SRTCNT=$O(^TMP("TIUVIEW",$J,SRTCNT)) Q:SRTCNT=""!(SRQUIT) D  . . . . . . S SRTIULN=^TMP("TIUVIEW",$J,SRTCNT)  . . . . . . I SRTIULN["STANDARD TITLE:" D  . . . . . . . S SRETIT=$P($P(SRTIULN,": ",2)," ",1)  . . . . . . . I SRETIT'=SRSTR S SRQUIT=1  . . . . . . Q:SRQUIT  . . . . . . I SRTIULN["/es/" S SRESB=1  . . . . . . I SRTIULN["Signed:" S SRSIGNDT=$P(SRTIULN,": ",2)  . . . . . . I SRTIULN["LOCAL TITLE:" S SRLOC=$P($P(SRTIULN,": ",2)," ",1)  . . . . . . I 'SRESB&(SRTIULN'["TITLE:")&(SRTIULN'["DATE OF NOTE:")&(SRTIULN'["AUTHOR:")&(SRTIULN'["URGENCY") D  . . . . . . . I SRTIULN'=""&(SRTIULN'=" ") S SRBCNT=SRBCNT+1,SRTBOD(SRBCNT)=SRTIULN  . . . . . I $D(SRTBOD)&($G(SRLOC)'="")&($G(SRSIGNDT)'="") D GENXML  . . . . I SRTYPE=2 D  . . . . . K SRTBOD,SRLOC,SRSIGNDT,SRTIULN,SRETIT,SRBCNT,SRFND S (SRFND,SRESB,SRQUIT,SRBCNT)=0  . . . . . S SRTCNT="" F S SRTCNT=$O(^TMP("TIUVIEW",$J,SRTCNT)) Q:SRTCNT=""!(SRQUIT) D  . . . . . . S SRTIULN=^TMP("TIUVIEW",$J,SRTCNT)  . . . . . . I SRTIULN["TITLE:" D  . . . . . . . S SRETIT=$P($P(SRTIULN,": ",2)," ",1)  . . . . . . . I SRETIT[SRSTR S SRFND=1  . . . . . . I SRTIULN["AUTHOR"&('SRFND) S SRQUIT=1 Q  . . . . . . I SRTIULN["/es/" S SRESB=1  . . . . . . I SRTIULN["Signed:" S SRSIGNDT=$P(SRTIULN,": ",2)  . . . . . . I SRTIULN["LOCAL TITLE:" S SRLOC=$P($P(SRTIULN,": ",2)," ",1)  . . . . . . I 'SRESB&(SRTIULN'["TITLE:")&(SRTIULN'["DATE OF NOTE:")&(SR  TIULN'["AUTHOR:")&(SRTIULN'["URGENCY") D  . . . . . . . I SRTIULN'=""&(SRTIULN'=" ") S SRBCNT=SRBCNT+1,SRTBOD(SRB  CNT)=SRTIULN  . . . . . I $D(SRTBOD)&($G(SRLOC)'="")&($G(SRSIGNDT)'="")&(SRFND) D GENXML  . . . . . Q  D XQUIT  K SRASRC,SR2CNT,SRTBOD,^TMP("TIUVIEW",$J)  Q  GENXML ;  S SR2CNT=SR2CNT+1  S SRRET(SR2CNT)="<note localTitle='"\_SRLOC\_"' signDate='"\_SRSIGNDT\_"'>"  S SR2CNT=SR2CNT+1,SRRET(SR2CNT)="<body>"  ;S SR2CNT=SR2CNT+1,SRRET(SR2CNT)="<![CDATA["  S SRBCNT="" F S SRBCNT=$O(SRTBOD(SRBCNT)) Q:SRBCNT="" D  . S SR2CNT=SR2CNT+1  . ;S SRRET(SR2CNT)=SRTBOD(SRBCNT)  . S SRRET(SR2CNT)="<![CDATA["\_SRTBOD(SRBCNT)\_"]]>"  ;S SR2CNT=SR2CNT+1,SRRET(SR2CNT)="]]>"  S SR2CNT=SR2CNT+1,SRRET(SR2CNT)="</body>"  S SR2CNT=SR2CNT+1,SRRET(SR2CNT)="</note>"  Q  XQUIT ;  S SR2CNT=SR2CNT+1,SRRET(SR2CNT)="</notes>"  Q |

| Routines | Activities |
| --- | --- |
| Routine Name | SRASRC7 |
| Enhancement Category | New |
| RTM | N/A |
| Related Options | SR ASRC ACTIVE MEDS (RPC) |
| Related Routines | This routine calls the following: OCL^PSOORRL |
| DD References | N/A |
| Related Protocols | N/A |
| Related ICRs | N/A |
| Data Passing | N/A |
| Input Attribute Name and Definition | N/A |
| Output Attribute Name and Definition | N/A |
| Current Logic | N/A |
| Modified Logic (Changes are in bold) | SRASRC7 ;SBX/JAS - RETURNS A PATIENT'S ACTIVE MEDICATION LIST ; 08/13/15  ;;3.0;Surgery;\*\*\*\*;24 Jun 93;Build 2  ;  ; Reference to API OCL^PSOORRL supported by IA# 2400  ;  ENTER(SRRET,DFN) ;  ;  ; SRRET = If records are found - Med ID ^ Med Name  ; = No records found - "" (null) value is returned in single array record  ; DFN = Patient file (#2) ien  ;  K ^TMP("PS",$J),SRRET,SRACNT,SRCNT,SRPS0 S SRRET="",SRACNT=0  I DFN="" Q  D OCL^PSOORRL(DFN,"","")  I '$D(^TMP("PS",$J)) S SRACNT=SRACNT+1,SRRET(SRACNT)="" Q  ;  S SRCNT=0  F S SRCNT=$O(^TMP("PS",$J,SRCNT)) Q:SRCNT="" I $D(^TMP("PS",$J,SRCNT,0)) D  . S SRPS0=^TMP("PS",$J,SRCNT,0)  . Q:$P(SRPS0,"^",9)'="ACTIVE"  . S SRACNT=SRACNT+1  . S SRRET(SRACNT)=$P(SRPS0,"^",1)\_"^"\_$P(SRPS0,"^",2)  I '$D(SRRET) S SRACNT=SRACNT+1,SRRET(SRACNT)=""  Q |

| Routines | Activities |
| --- | --- |
| Routine Name | SRSRQST1 |
| Enhancement Category | Modify |
| RTM | N/A |
| Related Options | Make Operation Requests |
| Related Routines | This routine calls the following: ^DIC and ^DIE |
| DD References | Files 81, 130 and 136.1 |
| Related Protocols | N/A |
| Related ICRs | N/A |
| Data Passing | N/A |
| Input Attribute Name and Definition | N/A |
| Output Attribute Name and Definition | N/A |
| Current Logic | K DR S DR="" I '$D(SREQ(27)) S DR="27T;" |
| Modified Logic (Changes are in bold) | ; JAS - 06/04/15 - ASRC PROJECT  ;K DR S DR="" I '$D(SREQ(27)) S DR="27T;"  K DR S DR=""  I '$D(SREQ(27)) D I SRASRCQ G SS  . K SRASRCQ,SRASRCC  . S SRASRCQ=0  . K DIC S DIC=81,DIC(0)="QEAMZ",DIC("A")="Principal Procedure Code (CPT): "  . S DIC("S")="I $$ACTIV^SROCPT(SRTN,+Y)"  . D ^DIC K DIC I Y<0,X="^" S SRASRCQ=1 Q  . I Y<0 S Y=""  . S SRASRCC=$P(Y,"^")  . ; DO ASRC CHECK  . I SRASRCC'="" D ^SRASRCCK  . S DR="27T///^S X=SRASRCC;"  . Q  ; End ASRC |

| Routines | Activities |
| --- | --- |
| Routine Name | SRCUSS1 |
| Enhancement Category | Modify |
| RTM | N/A |
| Related Options | Make Operation Requests  Delete or Update Operation Requests |
| Related Routines | This routine calls the following: ^DIC and ^DIE |
| DD References | Files 81, 130 and 136.1 |
| Related Protocols | N/A |
| Related ICRs | N/A |
| Data Passing | N/A |
| Input Attribute Name and Definition | N/A |
| Output Attribute Name and Definition | N/A |
| Current Logic | I 'Q("ED") D SET,^DIE S SRCUSS("OUT")=1 D RET Q |
| Modified Logic (Changes are in bold) | ; JAS - 07/01/15 - ASRC PROJECT  ;I 'Q("ED") D SET,^DIE S SRCUSS("OUT")=1 D RET Q  I 'Q("ED") D SET,^DIE S SRCUSS("OUT")=1 D CCHK,RET Q  CCHK ;  ; JAS - 07/01/15 - ASRC PROJECT  I $P(DR,";")="27T~" S SRASRCC=X D ^SRASRCCK  Q |

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

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

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| DD References |  | | | | |
| Related Protocols |  | | | | |
| Related 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

Table 9 - Templates

| Templates | Description | | | |
| --- | --- | --- | --- | --- |
| Template Name |  | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| Requirements Specification Document (RSD) |  | | | |
| Template Type | Sort | Input | Print | Other |
| Related Options |  | | | |

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

| Routines | Description |
| --- | --- |
| DD References |  |
| Global References |  |

#### Data Entries Affected by the Design

Table 10 - 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 calculation

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)

Table 11 - Unique Record ID

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

#### File or Global Size Changes

Table 12 - File or Global Size Changes

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

#### Mail Groups

Table 13 - 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 | |
| --- | --- | --- |
| DD References |  | |
| Related Protocols |  | |
| Mail Group Description |  | |
| Self-Enrollment Allowed | Yes | No |
| Type | Public | Private |

#### Security Keys

Table 14 - 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

Table 15 - Options

| 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 PATIENT, SR ASRC PROGRESS NOTE, SR ASRC RISK SAVE, SR ASRC LAB RESULTS, SR ASRC HEALTH FACTORS, SR ASRC PERSON CLASSES, SR ASRC ADL NOTES, ORQQPS LIST, SR ASRC DNR NOTES, SR ASRC ACTIVE MEDS |
| 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 |

| 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

Table 16 - 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) |
| --- |
|  |

#### RPC

Table 17 - RPCs

| RPCs | Activities |
| --- | --- |
| **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 | Activities |
| --- | --- |
| 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 | Activities |
| --- | --- |
| 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 | Activities |
| --- | --- |
| Name | SR ASRC LAB RESULTS |
| TAG^RTN | ENTER^SRASRC4 |
| Input Parameters | DFN, SRLABNS |
| Results Array | SRRET = Lab Name ^ Lab Result ^ Lab Date ^ Lab Units of Measure |
| Description | Based on the DFN and Lab strings passed in, this RPC will return the most recent lab match, or a null value if no matches are found. |

| RPCs | Activities |
| --- | --- |
| Name | SR ASRC PERSON CLASSES |
| TAG^RTN | CLASS^SRASRC |
| Input Parameters | N/A |
| Results Array | RETURN array = "Person Class(es) from New Person file (#200)" |
| Description | Based on the DUZ of the current VistALink session, this RPC returns the Person Class(es) for the user. |

| RPCs | Activities |
| --- | --- |
| Name | SR ASRC HEALTH FACTORS |
| TAG^RTN | ENTER^SRASRC5 |
| Input Parameters | DFN |
| Results Array | SRRET = Date of Visit ^ Health Factor |
| Description | Based on the DFN passed in, this RPC will return all health factors for this patient within the past year (365 days from current date,) or a null value if no matches are found. |

| RPCs | Activities |
| --- | --- |
| Name | SR ASRC ADL NOTES |
| TAG^RTN | ENTER1^SRASRC6 |
| Input Parameters | DFN, SRSTR |
| Results Array | SRRET array = Patient progress notes information in XML format |
| Description | Based on the DFN and Enterprise title passed in, this RPC will return matching progress notes for the patient in XML format. |

| RPCs | Activities |
| --- | --- |
| Name | SR ASRC DNR NOTES |
| TAG^RTN | ENTER2^SRASRC6 |
| Input Parameters | DFN, SRSTR |
| Results Array | SRRET array = Patient progress notes information in XML format |
| Description | Based on the DFN and search string passed in, this RPC will return progress notes for the patient in XML format that have either an Enterprise title or local note title that matches the passed search string. |

| RPCs | Activities |
| --- | --- |
| Name | SR ASRC ACTIVE MEDS |
| TAG^RTN | ENTER^SRASRC7 |
| Input Parameters | DFN |
| Results Array | SRRET array = Active medications for patient |
| Description | Based on the DFN, this RPC will return all active medications for the patient. |

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

## SOA / ESS Detailed Design

The system does not provide or consume any services.

# External System Interface Design

As shown in Figure 1, the only external system with which the application communicates is VistA. This section therefore describes the communication interface between the application and VistA.[[9]](#footnote-10)

## Interface Architecture

Figure 2 shows VistALink as a component of the application’s “hexagonal architecture”. The VistA interfacing code is all isolated from the domain model, with no dependency from the domain model to the interfacing code. The below diagram shows how the application, VistALink, and VistA actually communicate during system operation.



Figure 17 - VistALink Communication

As shown, the ASRC Web Application and VistALink both run as separate applications within the Application Server. VistALink is a Java EE Connector Architecture (JCA) Resource Adapter developed by the VA and used—unmodified—by the ASRC application. The container manages the ASRC application, the VistALink Resource Adapter, and the VistALink connection pools (one for each VistA instance). The ASRC Application obtains handles to the connection pools via the container-supplied Java Naming and Directory Interface (JNDI) and then actually communicates with VistA via the VistALink API.

## Interface Detailed Design

For detailed design documentation of the VistA interface, see the package-level documentation for gov.va.med.srcalc.vista[[10]](#footnote-11) and gov.va.med.srcalc.vista.vistalink[[11]](#footnote-12).

# Security and Privacy

## Security

As a prototype, the development team has not followed any National Institute of Standards and Technology (NIST) standards in designing application security. This section simply details the application’s security design.

Security in the Java Web Application is provided by the Spring Security Framework[[12]](#footnote-13). All authentication and authorization are performed on HyperText Transfer Protocol (HTTP) requests. (No method-level security is applied.)

The application integrates Spring Security via the provided Servlet Filter as shown below. The filter is declared in web.xml and configured in the Spring Application Context Extensible Markup Language (XML) file, applicationContext.xml.



Figure 18 - Security Design

### User Authentication

As described in Section 1.3, there are two distinct groups of users: clinical users and administrative users. These user groups access distinct parts of the application and authenticate using different methods, as shown in the below table.

Table 18 - User Group Authentication and Authorization

| User Group | User Store | Authentication Method | Authorized to Access |
| --- | --- | --- | --- |
| Clinical Users | VistA (NEW PERSON file) | VistA Access/Verify Codes or CCOW | Risk Calculations |
| Administrative Users | Static application configuration | Username and Password | Risk Model Administration and Administrative Reports |

The following subsections describe authentication for each user group.

#### Clinical User Authentication

The following facts influenced the design of Clinical User authentication:

* Since Clinical Users launch the Tool from CPRS, they already have an active VistA logon.
* During Risk Calculations, the Tool must read and write VistA patient data.
* The requirement to share user and patient context with CPRS (see Section 2.5.1).

Since Clinical Users are already logged in to VistA (via CPRS) when launching the tool and must have authorization to read and write patient data in VistA, VistA is the natural user store for Clinical Users.

Due to the requirement to share user and patient context with CPRS, the Tool supports CCOW-based Single Sign-On (SSO). The Tool also supports standard access/verify code authentication as a fallback. For more information on Clinical User authentication, see the Spring Application Context XML files, which contain detailed comments describing the security configuration.

#### Administrative User Authentication

The following facts influenced the design of Administrative User authentication:

* Administrators may not have access to CPRS in any VistA division.
* Tool administration is not tied to any particular VistA division: the configuration is global across the tool.
* Tool administration does not require loading of any data from VistA.
* VA shared authentication services (e.g., Identity and Access Management) were not available for use by this prototype.
* VA shared authentication services will likely be available for a VA implementation of this prototype.

Since Administrative Users may not have access to CPRS and VistA access is not necessary for administration, VistA is not an applicable user store for Administrative Users. Since VA shared authentication services were not available for use by this prototype, there was no external user store available.

Since a VA implementation of this prototype will likely use VA shared authentication services, the prototype simply uses a static, in-memory user store. This user store hardcodes user accounts in the Spring Application Context XML configuration and is appropriate only for development and testing. It should be replaced for any kind of production use.

For more information on Administrative User authentication, see the Spring Application Context XML files, which contain detailed comments describing the security configuration.

## Privacy

No PII/PHI requirements were identified for the purposes of this prototype Tool. (See Section 2.5.5.)

1. Additional Information

A.1 RTM

A.2 Packaging and Installation

A.3 Design Metrics

A.4 Acronym List and Glossary

Table 19 - Glossary

| Term | Meaning |
| --- | --- |
| API | Application Programming Interface |
| ASRC | Automated Surgical Risk Calculator |
| CCOW | Clinical Context Object Workgroup |
| CPRS | Computerized Patient Record System |
| CPT | Current Procedural Terminology |
| CSS | Cascading Style Sheets |
| CSV | Comma-Separated Value |
| DAO | Data Access Objects |
| DBMS | Database Managements System |
| DD | Data Dictionary |
| DOM | Document Object Model |
| EHR | Electronic Health Record |
| ESS | Electronic Safety and Security |
| FTL | Future Technology Lab |
| HTML | HyperText Markup Language |
| HTTP | HyperText Transfer Protocol |
| ICR | Integration Control Registration |
| Java EE | Java Platform Enterprise Edition |
| Java SE | Java Platform Standard Edition |
| JCA | Java EE Connector Architecture |
| JNDI | Java Naming and Directory Interface |
| JSON | Jackson |
| JSP | Java Server Pages |
| MDWS | Medical Domain Web Services |
| NIST | National Institute of Standards and Technology |
| NSO | National Surgery Office |
| ORM | Object Relational Mapping |
| PHI | Protected Health Information |
| PII | Personally Identifiable Information |
| RDBMS | Relational Database Managements System |
| RPC | Remote Procedure Call |
| RSD | Requirements Specification Document |
| RTM | Requirements Traceability Matrix |
| SOA | Service Oriented Architecture |
| SQL | Structured Query Language |
| SSO | Single Sign-On |
| TIU | Text Integration Utility |
| TRM | Technical Reference Model |
| UI | User Interface |
| VA | Department of Veterans Affairs |
| VASQIP | Veterans Affairs Surgical Quality Improvement Program |
| VistA | Veterans Health Information Systems and Technology Architecture |
| XML | Extensible Markup Language |

1. <https://github.com/VHAINNOVATIONS/ASRCM/blob/master/docs/ASRC_User_Guide.docx> [↑](#footnote-ref-2)
2. <https://github.com/VHAINNOVATIONS/ASRCM/blob/master/README.md> [↑](#footnote-ref-3)
3. Available online at: <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/domain/package-summary.html> [↑](#footnote-ref-4)
4. Available online at: <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/service/package-summary.html> [↑](#footnote-ref-5)
5. Available online at: <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/web/package-summary.html> [↑](#footnote-ref-6)
6. Available online at: <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/db/package-summary.html> [↑](#footnote-ref-7)
7. Available online at: <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/vista/package-summary.html> [↑](#footnote-ref-8)
8. Available online at: <http://vhainnovations.github.io/ASRCM/srcalc/javadoc> [↑](#footnote-ref-9)
9. As described in Section 3, the Tool does include an accompanying VistA patch, but since the Tool communicates with VistA using pre-existing methods (not created or enhanced in the patch), VistA is considered an external system for the purposes of this section. [↑](#footnote-ref-10)
10. <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/vista/package-summary.html> [↑](#footnote-ref-11)
11. <http://vhainnovations.github.io/ASRCM/srcalc/javadoc/gov/va/med/srcalc/vista/vistalink/package-summary.html> [↑](#footnote-ref-12)
12. <http://projects.spring.io/spring-security/> [↑](#footnote-ref-13)