Automated Surgical Risk Calculator (ASRC)

Technical Manual



Department of Veterans Affairs

November 2014

Version 1.1

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 11/25/2014 | 1.0 | Document Shell | D. Tombs |
| 11/25/2014 | 1.1 | Technical Edit | S. Vetzel |

Table of Contents

[1. Introduction 1](#_Toc404679687)

[1.1. Identification 1](#_Toc404679688)

[1.2. Scope 1](#_Toc404679689)

[1.3. User Characteristics 1](#_Toc404679690)

[1.4. Definitions, Acronyms, and Abbreviations 2](#_Toc404679691)

[1.5. References 2](#_Toc404679692)

[2. Background 3](#_Toc404679693)

[2.1. Overview of the System 3](#_Toc404679694)

[2.2. Overview of the Business Process 3](#_Toc404679695)

[2.3. Business Benefits 3](#_Toc404679696)

[2.4. Assumptions and Constraints 3](#_Toc404679697)

[2.4.1. Design Assumptions 3](#_Toc404679698)

[2.4.2. Design Constraints 3](#_Toc404679699)

[2.4.3. Design Trade-offs 4](#_Toc404679700)

[2.5. Overview of the Significant Requirements 4](#_Toc404679701)

[2.5.1. Overview of Significant Functional Requirements 4](#_Toc404679702)

[2.5.2. Overview of Functional Workload / Performance Requirements 5](#_Toc404679703)

[2.5.3. Overview of Operational Requirements 5](#_Toc404679704)

[2.5.4. Overview of the Technical Requirements 6](#_Toc404679705)

[2.5.5. Overview of the Security or Privacy Requirements 6](#_Toc404679706)

[2.5.6. Overview of System Criticality and High Availability Requirements 6](#_Toc404679707)

[3. Conceptual Design 8](#_Toc404679708)

[3.1. Conceptual Application Design 8](#_Toc404679709)

[3.1.1. Application Context 8](#_Toc404679710)

[3.1.2. High-Level Application Design 9](#_Toc404679711)

[3.2. Conceptual Data Design 11](#_Toc404679712)

[3.2.1. Project Conceptual Data Model 11](#_Toc404679713)

[3.2.2. Database Information 12](#_Toc404679714)

[3.2.3. User Interface Data Mapping 12](#_Toc404679715)

[3.2.3.1. Application Screen Interface 12](#_Toc404679716)

[3.2.3.1.1. <Insert name of screen> 12](#_Toc404679717)

[3.2.3.2. Application Report Interface 13](#_Toc404679718)

[3.2.3.3. Unmapped Data Element 13](#_Toc404679719)

[4. System Architecture 14](#_Toc404679720)

[4.1. Hardware Architecture 14](#_Toc404679721)

[4.2. Software Architecture 14](#_Toc404679722)

[4.3. Service Oriented Architecture / ESS 14](#_Toc404679723)

[4.4. Enterprise Architecture 15](#_Toc404679724)

[5. Data Design 16](#_Toc404679725)

[5.1. DBMS Files 16](#_Toc404679726)

[5.2. Non-DBMS Files 16](#_Toc404679727)

[5.3. Data View 17](#_Toc404679728)

[6. Detailed Design 18](#_Toc404679729)

[6.1. Hardware Detailed Design 18](#_Toc404679730)

[6.2. Software Detailed Design 18](#_Toc404679731)

[6.2.1. Database Repository 18](#_Toc404679732)

[6.2.2. Java Web Application Detailed Design 19](#_Toc404679733)

[6.2.3. VistA Patch Detailed Design 19](#_Toc404679734)

[6.2.3.1. Routines (Entry Points) 19](#_Toc404679735)

[6.2.3.2. Templates 21](#_Toc404679736)

[6.2.3.3. Data Entries Affected by the Design 22](#_Toc404679737)

[6.2.3.4. Unique Record(s) 23](#_Toc404679738)

[6.2.3.5. File or Global Size Changes 23](#_Toc404679739)

[6.2.3.6. Mail Groups 23](#_Toc404679740)

[6.2.3.7. Security Keys 24](#_Toc404679741)

[6.2.3.8. Options 26](#_Toc404679742)

[6.2.3.9. Protocols 28](#_Toc404679743)

[6.2.3.10. Remote Procedure Call (RPC) 30](#_Toc404679744)

[6.3. Service Oriented Architecture / ESS Detailed Design 30](#_Toc404679745)

[6.3.1. Service Description for <Consumed Service Name> 31](#_Toc404679746)

[6.3.2. Service Design for <Provided Service Name> 31](#_Toc404679747)

[7. External System Interface Design 32](#_Toc404679748)

[7.1. Interface Architecture 32](#_Toc404679749)

[7.2. Interface Detailed Design 32](#_Toc404679750)

[8. Security and Privacy 35](#_Toc404679751)

[8.1. Security 35](#_Toc404679752)

[8.2. Privacy 35](#_Toc404679753)

[A. Additional Information 36](#_Toc404679754)

[A.1. RTM 36](#_Toc404679755)

[A.2. Packaging and Installation 36](#_Toc404679756)

[A.3. Design Metrics 36](#_Toc404679757)

[A.4. Acronym List and Glossary 36](#_Toc404679758)

# Introduction

This Technical Manual details the Automated Surgical Risk Calculator 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 0.1 of both components.

## Scope

This subsection should provide a high level description of the deliverable.

The scope establishes the boundaries of the Technical Manual and should describe both features within and outside of the project scope. Therefore, any requirements not included in the design should be listed below as exclusions. Alternatively, any requirements included in the design should be listed below as inclusions.

Table : Scope Inclusions

| Includes |
| --- |
|  |
|  |
|  |

Table 2: Scope Exclusion

| Excludes |
| --- |
|  |
|  |
|  |

## User Characteristics

Describe the intended user base of the proposed system.

Describe the attributes of the user community (and their proficiency with software systems) and the technical community (and their familiarity with support and maintenance).

## 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

Provide a brief overview of the system in business terms, i.e.

* Explain what the system does?
* Why it is being developed or modified?
* The benefits of developing the system.

This section should identify:

* The major users of the system,
* The participants in its operation, and
* The functions that they perform.

## 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

Describe the business benefits of developing this system in brief (4 – 5 sentences). A link to the Business Requirements Document (BRD) may be included to maintain brevity of the document and provide additional information as needed.

## Assumptions and Constraints

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

### Design Assumptions

Identify any specific assumptions that were made which influenced the design of this system.

### Design Constraints

Describe any unusual conditions or constraints which limited the range of design choices that were available, or impacted the design choice that was made.

E.g. schedules, costs, technical constraints such as the organization’s commitment to a specific development platform or programming language.

### 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

The material in this section is not to replace either the existing functional or technical requirements documents, nor serve as the basis for the Requirements Traceability Matrix, but only to inform non-project personnel reading this document of the basis for the design.

### Overview of Significant Functional Requirements

Provide an overview of the pivotal (i.e. that force design decisions) functional requirements for the system. [Cutting and pasting large chunks of text from the BRD is **not** appropriate, links to BRD, RSD and RTM are encouraged as it maintains the brevity of the document and minimizes duplication.]

The goal is to identify the major functions to be performed and the few major requirements that drive the design that is described in the sections below. The goal is not to include the full set of requirements in this document or to replace the functional requirements documents.

The emphasis should be on identifying the impact that these requirements have on the design.

These may either be a synopsis of the major requirements or specific selected requirements. Please see reference [#] in the document reference list provided above.

| ID | Requirement |
| --- | --- |
| <Requirement  Number from  Functional  Requirement  Document> | <Requirement text> |

*Table 2: Functional Requirements*

### Overview of Functional Workload / Performance Requirements

Describe the amount of work to be performed in business terms. The description should be independent of any technical design decisions (e.g. describe number of air passengers arriving rather than the number of transactions) and any business performance goals (e.g. do not describe that each passenger shall be screened at once before the aircraft takes off and once after he lands).

| ID | Requirement |
| --- | --- |
| <Requirement  Number from  Functional  Requirement  Document> | <Requirement text> |

*Table 3: Workload and Performance Requirements*

### Overview of Operational Requirements

| ID | Requirement |
| --- | --- |
| <Requirement  Number from  Functional  Requirement  Document> | <Requirement text> |

*Table 4: Operational Requirements*

### Overview of the Technical Requirements

To the extent that they are known, provide the pivotal (i.e. that force design decisions) technical requirements that drive the conceptual design in the table below.

| ID | Requirement |
| --- | --- |
| <Requirement  Number from  Functional  Requirement  Document> | <Requirement text> |

*Table 5: Technical Requirements*

### Overview of the Security or Privacy Requirements

To the extent that they are known, provide any special security requirements that are unique to this system. State whether it is expected that this system will require a new A&A or will it be covered by another system’s A&A, or an update to A&A is required. If this system will be included in another system’s A&A, identify that system. Details about the FISMA classification requirements should be documented as appropriate.

| ID | Requirement |
| --- | --- |
| <Requirement  Number from  Functional  Requirement  Document> | <Requirement text> |

*Table 6: Security Requirements*

### Overview of System Criticality and High Availability Requirements

Describe the mission criticality of the system and the degree to which continuous operation (i.e.,

99.999% availability) is required or not required. Describe the extent to which geographically distributed, high availability designs are required. If the system is not mission critical, and high availability is not required, provide the following information:

* Availability requirement
* Allowable downtime
* Recovery Time Objectives(RTO) (for disaster recovery) Requirement
* Recovery Point Objectives(RPO) (for disaster recovery) Requirement

Describe the approach that will be taken to provide the required level of availability and disaster recovery.

# Conceptual Design

This section of the SDD provides details about the following topics:

* Conceptual Application Design
* Conceptual Data Design
* Conceptual Infrastructure Design

## Conceptual Application Design

This section provides the conceptual design of the application that is being produced by this project. [There should be a “To-Be” and a “This-Increment” view of the design, in addition to an “As-Is” view if an existing system. The “To-Be” view should include the future application context, and application high level design. The “This-Increment” view should have application context and high level design.]

### Application Context

The following figure represents the context in which the application will exist.

Please provide a diagram showing the context within which the application exists. The diagram should include:

* One object for the system that is the subject of this design,
* One object for each system or external service with which this system interfaces,
* One object for each Program Office system or subsystem with which this system interacts, and
* One for each data store that this system shares with other systems.

Sample Application Context Diagram

Figure : Sample Application Context Diagram

Table 5 describes the information in the Application Context Diagram in four sections. Note that the system for which this design applies is represented by a single object (typically in the center of the diagram). Therefore, it is not referred to in Table 5 below.

Table 5 (Grouping): Application Context Description

External System Interfaces

| ID | Name | Related Object | Input Messages | Output Messages | External Party |
| --- | --- | --- | --- | --- | --- |
| < ID from diagram> | <Interface name from the object rows above> | <Object from the list above that is the source of this interface> | <For each input message, enter a business description of the data being input> | <For each output message, enter a business description of the data being output> | <Name of external party> |

Externally Shared Data Stores

| ID | Name | Data Stored | Owner | Access |
| --- | --- | --- | --- | --- |
| < ID from diagram> | <Name of the data store> | <Description of the data being stored> | <This System / Name of OIT or external organization> | <Enter the Create, Read, Update, or Delete (CRUD) operations that this system does on this data store> |

### High-Level Application Design

The High-Level Application Design identifies the major components of the application and the relationships of the major application components to each other and to the surrounding applications. The major components of the application are at the subsystem or top-level service area. Many different graphical formats are acceptable for the High-Level Application Design Diagram. Lower-level services will be defined and documented in the Logical Application Design section.

Figure 3 illustrates a High-Level Application Design in the form of a dataflow diagram. This diagram differs from the diagram in Figure 2 in that the single object representing this system in Figure 2 is decomposed into its major components. Use Table6 to describe the objects in Figure 3.

Note: If an extension to a legacy system is being developed without use of services, all references to “Service” should be changed to “Subsystem.”

A Collaboration Diagram, or in the case of Services, a Service Capability Diagram may be included instead or an Application Diagram if it illustrates the subject better.

Sample High-Level Application Design

Figure : Sample High-Level Application Design

Table 6: Objects in the High Level Application Design

Objects / Components to be Built or Modified

| ID | Name | Description | Service or Legacy Code | External Interface Name | External Interface ID | Internal Interface Name | Internal Interface ID | SDP Sections 1&2 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| < ID from diagram> | <Name of high level service or internal subsystems> | <Business level discussion of the function or role of the service or subsystem> | <Service / modification to legacy system> | <Name of each of the external interfaces to this object> | <ID of each of the external interfaces to this object> | <Name of each of the internal interfaces to this object> | <ID of each of the internal interfaces to this object> | [Approved / Submitted / Being Developed] |

Internal Data Stores

| ID | Name | Data Stored | Steward | Access |
| --- | --- | --- | --- | --- |
| < ID from diagram> | <Name of the data store> | <Description of the data being stored> | <Name of the system/subsystem /service that is the steward for the data> | <Which CRUD operations does this system do on this data store> |

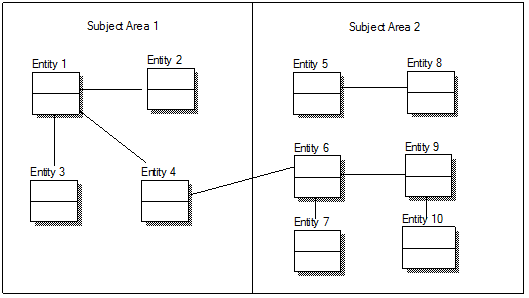
## Conceptual Data Design

### Project Conceptual Data Model

A project conceptual data model (CDM) is a high-level representation of the data entities and their relationships. It does not normally include the data elements that comprise each entity. It is a first step toward developing the more detailed logical data model (LDM) that will be provided during the Logical Data Design.

Figure 4 illustrates a sample of a project CDM.

Figure : Sample Project Conceptual Data Mode

****

### Database Information

Use Table 9 to identify all the databases that will be created, replaced, interfaced with, or whose structure will be modified (i.e., add or delete tables or add or delete columns to a table) as part of this effort.

Table 9: Database Inventory

| Database Name | Description | Type | Steward |
| --- | --- | --- | --- |
| <Name> | <Description> | <Create/Replace/Interface /Modify> | <Application/Organization that is the steward> |

### User Interface Data Mapping

This section describes and defines the format and information that will be available for users of the product to be able to enter data into the database or to retrieve information from the database, if applicable.

#### Application Screen Interface

Create a new subsection for each screen of the Graphical User Interface (GUI) that users will have access to, in order to enter or update information in the database.)

##### <Insert name of screen>

Figure 5: <screen name> Screen represents the screen that <describes what the screen accomplishes>; Table 10 describes it. Paste a screenshot below and complete the table to describe the screen.

Figure : *<screen name>* Screen

#### Application Report Interface

This section describes and defines the reports that will be available in the user interface, if applicable.

#### Unmapped Data Element

In this section describe any database element that was not mapped to a screen and the reason the data element(s) was not mapped. This section may be skipped if there is no User Interface involved in the project, such a building a service offering etc.

# System Architecture

This section describes the system and/or subsystem(s) architecture for the project. Discuss the general architectural decisions that have been approved. Include diagrams where appropriate.

## Hardware Architecture

Describe the system hardware architecture and indicate whether the processing system is distributed or centralized. List and describe the hardware modules with diagrams showing the connectivity between the modules. If possible, identify the type, number, and location of servers, workstations, processors, backup systems, and output devices. There should also be information presented in this section, relating to the capacity planning of the system.

## Software Architecture

Describe the overall system software and organization. List and describe the software modules (i.e., including functions, subroutines, or classes), programming languages, and development tools.

Describe all software required to support the system, and specify the physical location of all software systems. Identify database platforms, compilers, utilities, operating systems, and communications software.

Provide diagrams that illustrate the segmentation levels down to the lowest level. Include names and reference numbers for all features on the diagrams. Include a narrative that expands on and enhances the understanding of the functional breakdown.

Note: Diagrams should map to the Requirements Specification Document’s data flow diagrams.

## Service Oriented Architecture / ESS

This subsection of the SDD should put the product into perspective with other related products. This is achieved in the high level design.

* If the product is independent and totally self-contained, it should be so stated here.
* If the SDD defines a product that is a component of a larger system, as occurs frequently, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software. It is highly recommended that the SDD and other related artifacts of the larger system are included by reference, with links and not duplicate huge chunks of it here, which could potentially get out of sync. Integration projects depend on all parties understanding the same things about their relationships, and such information should be in one document and referenced by link as needed.

A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.

Services Provided: Those shared services that will be provided as part of this application (if the project is a combined solution and service development project). The Data Exchanges should then be included as part of whatever service is providing them. This may also be described as an attribute of the components listed in the high level application design when appropriate.

Service Required/Consumed: This would be the services this solution/application depends on. Again, data exchanges should be included as part of the service descriptions. This should also be adequately described in the conceptual and integration sections as appropriate.

Describe the Service Oriented Architecture of the system. Describe the Enterprise Shared Services consumed or provided by the system.

Provide a diagram depicting the Enterprise Shared Services between the system and subsystem modules.

If the system currently being built is in-flight or in-transition, then depict the as-is, interim and target states of the system with diagrams, and identify the Enterprise Shared Services consumed or provided. This will be part of the conceptual solution design.

If the solution proposed is a duplication of an existing service, or a stand-alone silo solution, then appropriate justification needs to be provided.

## Enterprise Architecture

Describe the Enterprise Architecture of the system.

Show adherence to the VA Technical Reference Model (TRM)/ Standards Profile (SP). New system development and selection must adhere to approved standards and rules, unless it proves to be more cost-effective over the life of the application to deviate from the standards. The standards, strategies, and guidelines establish the fundamental technologies enabling the VA to meet many of its business and information system goals. By using these standards, the VA can promote interoperability, portability and adaptability within systems, promote quality assurance, place the VA in a position to utilize current technology, and provide a framework for IT application and infrastructure development. The current TRM/SP is located VA Enterprise Architecture (EA) v2.1 at <http://trm.oit.va.gov/>.

# 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.

## 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.

## Data View

A "Data View" should be included in the Architectural Representation whenever persistent data objects are included in the system (they are typically present in most software systems). The data view describes the logical data model of the system and includes an Entity Relationship Diagram (ERD). 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>>

# 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 14: Routines (Instructions)

| Routines | Instructions |
| --- | --- |
| **Routine Name** | List the routine affected by the functionality being designed. |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **RTM** | List the 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 in this column list RSD Requirement 3.3.1) |
| **Related Options** | List options that directly call or are called by the routine. |
| **Related Routines** | List routines that directly call or are called by the routine. |
| **Data Dictionary (DD) References** | List files that reference the routine through input transforms, cross reference logic, etc. |
| **Related Protocols** | List protocols that reference or are referenced by the routine. |
| **Related Integration Control Registrations (ICRs)** | List proposed new ICRs and subscribed ICRs. Also, list any obscure Supported ICRs. |
| **Data Passing** | Check the appropriate box. Also a short description of what invokes the new/changed routine should be included in this section. An example of such a description would be a note that the new/changed routine will be invoked as part of a function call or it would be invoked through user menu-driven options, system protocols, HL7 Logical Links, etc. This section refers specifically to the change implemented with the design. |
| **Input Attribute Name and Definition** | List the Input Attributes passed into the new or changed routine logic. Each attribute should be defined. |
| **Output Attribute Name and Definition** | List the Output Attributes returned from the new or changed routine logic. Each attribute should be defined. |
| **Current Logic** | Define the current logic in the routine that the design will modify. If this is new code, enter “N/A”. |
| **Modified Logic (Changes are in bold)** | Define the logic in the routine that the design will implement. |

Table 15 (Grouping): Routines

| 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 16: 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 17: 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 20: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
|  |  |  |

#### 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 21: 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 22: 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 23: 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 24: 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 25: 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 26: 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 27: Options (Instructions)

| Options | Instructions |
| --- | --- |
| **Option Name  (MENU TEXT field)** | Enter the name of the option affected. |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **Associated Menu Options that will invoke this reference** | List the menu type options on which the respective option is or will be contained. |
| **Data Passing** | Check the appropriate box. Also a short description of what invokes the new/changed routine should be included in this section. An example of such a description would be a note that the change to the option will be referenced through VA Mailman server messages, user selection of the option from the VA Kernel Menu Management system, etc. This section refers specifically to the change implemented with the design. |
| **Menu Text Description** | Enter the name of the option as it will be displayed to the user within the menu system. |
| **Option Type** | Specify the type of option |
| **Option Definition** | Provide all the information necessary to fully define the option. Include options that are included in the menu, if applicable. |
| **Current Entry Action Logic** | Define the current logic for the entry action of the option 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 option 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. |

Table28: 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.

Table29: 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 30: 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 31: RPCs (Instructions)

| RPCs | Instructions |
| --- | --- |
| **Name** | List the specific name of the RPC affected. |
| **TAG^RTN** | List the tag (label) and routine. |
| **Input Parameters** | This field is used to identify an input parameter for the API. |
| **Results Array** | This field tells the RPC Broker how to process the resulting data from the call. |
| **Description** | Provide a brief description of the RPC affected. |

Table 32: 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 59: 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>