Perceptive Reach:

Integrated Reach Database System

(IRDS)

System Design Document



Department of Veterans Affairs

October 2014

Version 0.2

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 10/07/2014 | 0.0 | First Draft | Paul Bradley |
| 10/24/2014 | 0.1 | Added Database Content | Bill Balshem |
| 10/27/2014 | 0.2 | Added Analytics Content | Kristopher Hoover |
| 10/28/2014 | 0.3 | Added Dashboard Content | Robert Snelling |
|  |  |  |  |

Artifact Rationale

The System Design Document (SDD) is a dual-use document that provides the conceptual design as well as the as-built design. This document will be updated as the product is built, to reflect the as-built product. Per the Project Management Accountability System (PMAS) Guide, the SDD as a conceptual design is required prior to the Milestone 1 Review. (Sections 1, 2, 3, 4, 5, 7, 9 need to be populated, as applicable.) The as-built design for each delivery must be incorporated prior to the Milestone 2 Review. (The entire document needs to be populated or updated, as applicable.)

Table of Contents

[1. Introduction 1](#_Toc398548352)

[1.1. Purpose of the SDD 1](#_Toc398548353)

[1.2. Identification 1](#_Toc398548354)

[1.3. Scope 1](#_Toc398548355)

[1.4. Constraining Policies, Directives and Procedures 2](#_Toc398548356)

[1.5. User Characteristics 2](#_Toc398548357)

[1.6. Relationship to Other Documents and Plans 2](#_Toc398548358)

[1.7. Definitions, Acronyms, and Abbreviations 2](#_Toc398548359)

[1.8. References 2](#_Toc398548360)

[2. Background 3](#_Toc398548361)

[2.1. Overview of the System 3](#_Toc398548362)

[2.2. Overview of the Business Process 3](#_Toc398548363)

[2.3. Business Benefits 4](#_Toc398548364)

[2.4. Assumptions and Constraints 4](#_Toc398548365)

[2.4.1. Design Assumptions 4](#_Toc398548366)

[2.4.2. Design Constraints 5](#_Toc398548367)

[2.4.3. Design Trade-offs 5](#_Toc398548368)

[2.5. Overview of the Significant Requirements 5](#_Toc398548369)

[2.5.1. Overview of Significant Functional Requirements 5](#_Toc398548370)

[2.5.2. Overview of Functional Workload / Performance Requirements 6](#_Toc398548371)

[2.5.3. Overview of Operational Requirements 6](#_Toc398548372)

[2.5.4. Overview of the Technical Requirements 7](#_Toc398548373)

[2.5.5. Overview of the Security or Privacy Requirements 7](#_Toc398548374)

[2.5.6. Overview of System Criticality and High Availability Requirements 7](#_Toc398548375)

[2.5.7. Single Sign-on Requirement 8](#_Toc398548376)

[2.5.8. Requirement for Use of Enterprise Portals 8](#_Toc398548377)

[2.5.9. Special Device Requirements 8](#_Toc398548378)

[2.6. Legacy System Retirement 8](#_Toc398548379)

[3. Conceptual Design 9](#_Toc398548380)

[3.1. Conceptual Application Design 9](#_Toc398548381)

[3.1.1. Application Context 9](#_Toc398548382)

[3.1.2. High-Level Application Design 11](#_Toc398548383)

[3.1.3. Application Locations 12](#_Toc398548384)

[3.2. Conceptual Data Design 13](#_Toc398548385)

[3.2.1. Project Conceptual Data Model 13](#_Toc398548386)

[3.2.2. Database Information 13](#_Toc398548387)

[3.2.3. User Interface Data Mapping 13](#_Toc398548388)

[3.2.3.1. Application Screen Interface 14](#_Toc398548389)

[3.2.3.1.1. <Insert name of screen> 14](#_Toc398548390)

[3.2.3.2. Application Report Interface 14](#_Toc398548391)

[3.2.3.2.1. <Insert name of report> 14](#_Toc398548392)

[3.2.3.3. Unmapped Data Element 15](#_Toc398548393)

[3.3. Conceptual Infrastructure Design 15](#_Toc398548394)

[3.3.1. System Criticality and High Availability 15](#_Toc398548395)

[3.3.2. Special Technology 16](#_Toc398548396)

[3.3.3. Technology Locations 16](#_Toc398548397)

[3.3.4. Conceptual Infrastructure Diagram 17](#_Toc398548398)

[3.3.4.1. Location of Environments and External Interfaces 17](#_Toc398548399)

[3.3.4.2. Conceptual Production String Diagram 17](#_Toc398548400)

[4. System Architecture 19](#_Toc398548401)

[4.1. Hardware Architecture 19](#_Toc398548402)

[4.2. Software Architecture 19](#_Toc398548403)

[4.3. Network Architecture 19](#_Toc398548404)

[4.4. Service Oriented Architecture / ESS 19](#_Toc398548405)

[4.5. Enterprise Architecture 20](#_Toc398548406)

[5. Data Design 21](#_Toc398548407)

[5.1. DBMS Files 21](#_Toc398548408)

[5.2. Non-DBMS Files 21](#_Toc398548409)

[5.3. Data View 22](#_Toc398548410)

[6. Detailed Design 23](#_Toc398548411)

[6.1. Hardware Detailed Design 23](#_Toc398548412)

[6.2. Software Detailed Design 23](#_Toc398548413)

[6.2.1. Conceptual Design 23](#_Toc398548414)

[6.2.1.1. Product Perspective 23](#_Toc398548415)

[6.2.1.1.1. User Interfaces 24](#_Toc398548416)

[6.2.1.1.2. Hardware Interfaces 24](#_Toc398548417)

[6.2.1.1.3. Software Interfaces 24](#_Toc398548418)

[6.2.1.1.4. Communications Interfaces 24](#_Toc398548419)

[6.2.1.1.5. Memory Constraints 24](#_Toc398548420)

[6.2.1.1.6. Special Operations 24](#_Toc398548421)

[6.2.1.2. Product Features 25](#_Toc398548422)

[6.2.1.3. User Characteristics 25](#_Toc398548423)

[6.2.1.4. Dependencies and Constraints 25](#_Toc398548424)

[6.2.2. Specific Requirements 26](#_Toc398548425)

[6.2.2.1. Database Repository 26](#_Toc398548426)

[6.2.2.2. System Features 26](#_Toc398548427)

[6.2.2.3. Design Element Tables 27](#_Toc398548428)

[6.2.2.3.1. Routines (Entry Points) 27](#_Toc398548429)

[6.2.2.3.2. Templates 29](#_Toc398548430)

[6.2.2.3.3. Bulletins 30](#_Toc398548431)

[6.2.2.3.4. Data Entries Affected by the Design 31](#_Toc398548432)

[6.2.2.3.5. Unique Record(s) 31](#_Toc398548433)

[6.2.2.3.6. File or Global Size Changes 31](#_Toc398548434)

[6.2.2.3.7. Mail Groups 32](#_Toc398548435)

[6.2.2.3.8. Security Keys 33](#_Toc398548436)

[6.2.2.3.9. Options 34](#_Toc398548437)

[6.2.2.3.10. Protocols 36](#_Toc398548438)

[6.2.2.3.11. Remote Procedure Call (RPC) 38](#_Toc398548439)

[6.2.2.3.12. Constants Defined in Interface 38](#_Toc398548440)

[6.2.2.3.13. Variables Defined in Interface 38](#_Toc398548441)

[6.2.2.3.14. Types Defined in Interface 38](#_Toc398548442)

[6.2.2.3.15. GUI 39](#_Toc398548443)

[6.2.2.3.16. GUI Classes 39](#_Toc398548444)

[6.2.2.3.17. Current Form 39](#_Toc398548445)

[6.2.2.3.18. Modified Form 39](#_Toc398548446)

[6.2.2.3.19. Components on Form 39](#_Toc398548447)

[6.2.2.3.20. Events 40](#_Toc398548448)

[6.2.2.3.21. Methods 40](#_Toc398548449)

[6.2.2.3.22. Special References 40](#_Toc398548450)

[6.2.2.3.23. Class Events 40](#_Toc398548451)

[6.2.2.3.24. Class Methods 40](#_Toc398548452)

[6.2.2.3.25. Class Properties 40](#_Toc398548453)

[6.2.2.3.26. Uses Clause 40](#_Toc398548454)

[6.2.2.3.27. Forms 40](#_Toc398548455)

[6.2.2.3.28. Functions 41](#_Toc398548456)

[6.2.2.3.29. Dialog 43](#_Toc398548457)

[6.2.2.3.30. Help Frame 44](#_Toc398548458)

[6.2.2.3.31. HL7 Application Parameter 45](#_Toc398548459)

[6.2.2.3.32. HL7 Logical Link 45](#_Toc398548460)

[6.2.2.3.33. COTS Interface 46](#_Toc398548461)

[6.3. Network Detailed Design 47](#_Toc398548462)

[6.4. Service Oriented Architecture / ESS Detailed Design 47](#_Toc398548463)

[6.4.1. Service Description for <Consumed Service Name> 47](#_Toc398548464)

[6.4.2. Service Design for <Provided Service Name> 47](#_Toc398548465)

[6.4.2.1. Introduction 47](#_Toc398548466)

[6.4.2.1.1. Purpose and Scope of Service 47](#_Toc398548467)

[6.4.2.1.2. Links to Other Documents 48](#_Toc398548468)

[6.4.2.2. Service Details 48](#_Toc398548469)

[6.4.2.2.1. Service Identification 48](#_Toc398548470)

[6.4.2.2.2. Service Versions 49](#_Toc398548471)

[6.4.2.2.3. Summary of Design and Platform Details 49](#_Toc398548472)

[6.4.2.2.3.1. SOA Pattern(s) Implemented 49](#_Toc398548473)

[6.4.2.2.3.2. COTS Platform vendor names and versions for hosting platform 49](#_Toc398548474)

[6.4.2.3. Dependencies 49](#_Toc398548475)

[6.4.2.4. Service Design Details 50](#_Toc398548476)

[6.4.2.4.1. Interface Technical Specs 50](#_Toc398548477)

[6.4.2.4.1.1. Service Invocation Type 50](#_Toc398548478)

[6.4.2.4.1.2. Service Interface Type 50](#_Toc398548479)

[6.4.2.4.1.3. Service Name 50](#_Toc398548480)

[6.4.2.4.1.4. Interface 50](#_Toc398548481)

[6.4.2.4.1.5. End Points 50](#_Toc398548482)

[6.4.2.4.1.6. Operations or Methods 50](#_Toc398548483)

[6.4.2.4.1.7. Message Schemas 51](#_Toc398548484)

[6.4.2.4.2. Information Model 51](#_Toc398548485)

[6.4.2.4.2.1. Class Diagram and Description of Entities Involved 51](#_Toc398548486)

[6.4.2.4.2.2. Mappings from ELDM to Standards Based Schemas 51](#_Toc398548487)

[6.4.2.4.3. Behavior Model (AKA Use Case Realization) 51](#_Toc398548488)

[6.4.2.4.3.1. Use Cases (Use Case Model) 51](#_Toc398548489)

[6.4.2.4.3.2. Interaction Diagrams 51](#_Toc398548490)

[6.4.2.5. Gap Analysis 51](#_Toc398548491)

[6.4.2.5.1. Variances from Enterprise Target Architecture 52](#_Toc398548492)

[6.4.2.5.2. Variances from SLDs 52](#_Toc398548493)

[6.4.2.5.3. Variances from Standards and Policies 52](#_Toc398548494)

[6.4.2.5.4. Justification for Exceptions and Mitigation 52](#_Toc398548495)

[7. External System Interface Design 53](#_Toc398548496)

[7.1. Interface Architecture 53](#_Toc398548497)

[7.2. Interface Detailed Design 53](#_Toc398548498)

[8. Human-Machine Interface 55](#_Toc398548499)

[8.1. Interface Design Rules 55](#_Toc398548500)

[8.2. Inputs 55](#_Toc398548501)

[8.3. Outputs 55](#_Toc398548502)

[8.4. Navigation Hierarchy 55](#_Toc398548503)

[8.4.1. Screen [x.1] 55](#_Toc398548504)

[8.4.2. Screen [x.2] 55](#_Toc398548505)

[8.4.3. Screen [x.3] 55](#_Toc398548506)

[9. Security and Privacy 56](#_Toc398548507)

[9.1. Security 56](#_Toc398548508)

[9.2. Privacy 56](#_Toc398548509)

[Attachment A – Approval Signatures 57](#_Toc398548510)

[A. Additional Information 58](#_Toc398548511)

[A.1. RTM 58](#_Toc398548512)

[A.2. Packaging and Installation 58](#_Toc398548513)

[A.3. Design Metrics 58](#_Toc398548514)

[A.4. Acronym List and Glossary 58](#_Toc398548515)

[A.5. Required Technical Documents 58](#_Toc398548516)

[A.6. Attach Documents 58](#_Toc398548517)

# Introduction

VA is seeking to expand suicide prevention to include upstream approaches, designed to reduce initiation or escalation of a risk factor. Upstream suicide interventions target individuals or groups who exhibit biological, psychological, or social risk factors that are more prominent among high-risk groups than among the larger population. Understanding the unique needs of our nation’s Veterans and the military culture as it relates to stigma and mental health is important for early intervention. The goal of the Integrated Reach Database System (IRDS) innovation is to promote the general health of the Veteran population and effectively intervene in issues before they escalate in crisis.

The IRDS solution innovates the current process of risk data collection, analysis, and use in effective intervention strategy. The solution will harness the power of large and diverse data stores to aggregate, analyze and identify risk onset as well as reveal previously unidentified at-risk individuals and populations as a holistic and integrated approach.

The IRDS innovation will serve to bolster the three major components of VHA’s Strategic Plan for Suicide Prevention: surveillance, risk and protective factors, and prevention interventions. The IRDS innovation will target antecedent events specific to Veteran populations prior to the onset of risk to mitigate the development of risk.

## Purpose of the SDD

Describe the purpose of the document and its intended audience.

For example:

“The purpose of this document is to describe in sufficient detail how the proposed system will be constructed. The SDD translates the requirement specifications into a document from which the developers can create the actual system. It identifies the top-level system architecture, and identifies hardware, software, communication, and interface components.”

## Identification

Identify the system and software which apply to the SDD, including: identification number(s), title(s), abbreviation(s), version number(s), and release number(s). Identify all standards (e.g., American National Standards Institute [ANSI], International Organization for Standardization [ISO], Institute of Electrical and Electronics Engineers [IEEE], etc.).

## Scope

The IRDS Interface Design Specification document describes the relationship between IRDS and each of the external systems connected to it in terms of data items, protocols, and timing of events.

This Interface Design Specification will describe what data will be transferred between the IRDS and its input sources and output destinations such as:

* Input: Suicide Data Repository (SDR)
* Input: Veterans Benefits Administration XYZ
* Input: VISN XYZ VistA
* Output: VA Suicide Prevention Coordinators
* Output: Rutgers UHBC Outreach and Intervention Coordinators & Clinicians
* Output VA Suicide Prevention stakeholders

This document should be read in conjunction with the IRSD System Design Document.

## 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).

## Relationship to Other Documents and Plans

The following IRDS documents may be referenced in tandem with the information recorded here:

* Project Management Plan (PMP)
* IRDS Interface Design Specification

## Project Management Plan (PMP)Definitions, Acronyms, and Abbreviations

| Acronym | Term |
| --- | --- |
| BIRLS | Beneficiary Identification Records Locator System |
| CD | Compact Disk |
| CDC | Center for Disease Control |
| DoD | Department of Defense |
| ETL | Extract, Transform, Load |
| GB | Gigabyte |
| ICD | International Classification of Diseases |
| IM/IT | Information Management/Information Technology |
| IRDS | Integrated Reach Database System |
| IT | Information and Technology |
| NDI | National Death Index |
| OIT | Office of Information and Technology |
| OMHS | Office of Mental Health Services |
| SAS | Statistical Analysis System |
| SDCD | State Death Certificate Data |
| SDR | Suicide Data Repository |
| SFTP | Secure File Transfer Protocol |
| SMITREC | Serious Mental Illness Treatment Resource and Evaluation Center |
| SPAN | Suicide Prevention Applications Network |
| SQL | Structured Query Language |
| SSIS | SQL Server Integration Services |
| SSN | Social Security Number |
| TB | Terabyte |
| UI | User Interface |
| VA | Department of Veterans Affairs |
| VCL | Veterans Crisis Line |
| VHA | Veterans Health Administration |
| VSSC | VHA Support Service Center |

# Background

## Overview of the System

The Perceptive Reach development and field pilot proposes to combine technology, outreach and clinical support to realize a clinically based data-driven early intervention and treatment solution aimed at suicide prevention. IRDS is a solution for analyzing multiple and integrated data sets with cutting-edge data analytic techniques and visualizations to identify at-risk individuals and populations and provide proactive and secure notifications of these results to Veteran support services. The Perceptive Reach project proposes to expand the capabilities of the Suicide Data Repository to include new interfaces to clinical data sources, integrated data analytics capabilities, a surveillance dashboard, and secure messaging.



Figure 1

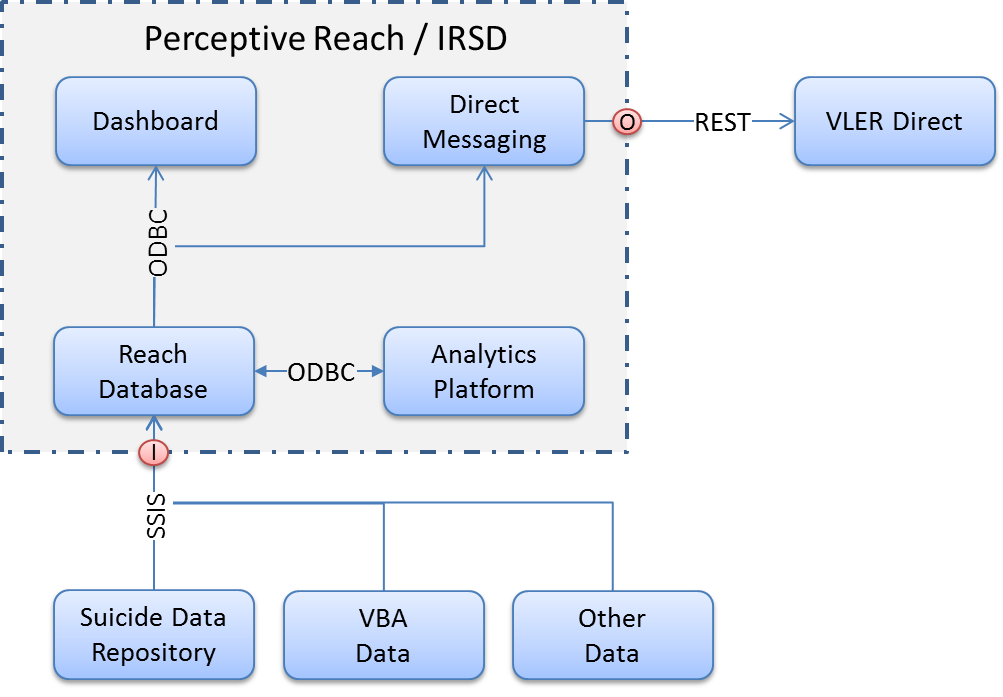


Figure 2

## 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).



Figure 1: Sample Business Processes Diagram

Please provide numeric identifiers for all business processes in Figure 1 and enter them in Table 3.

Table 3: Business Process

| Business Process ID | Business Process Name | Type | Owner | Description |
| --- | --- | --- | --- | --- |
| < Enter the Business Process ID number> | <Enter Business Process name> | <Select one: Existing or Modernized> | <Enter organization performing the process> | <Enter business language description of the business processes> |

## 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.

### Single Sign-on Requirement

Document any use of any mandated single sign-on solution (i.e., Kerberos, and Active Directory). If there are any project requirements which would limit or inhibit use of the mandated single sign-on solution, provide those requirements here.

### Requirement for Use of Enterprise Portals

All new user interfaces are required to use the Enterprise Portals. If any user interfaces are not planned to be through the approved portals, please identify those requirements for the user interface that will not allow use of Enterprise Portals.

### Special Device Requirements

State the requirement for any special devices (e.g., Diagnostic equipment, X-Ray units, Mobile units etc.) to be part of this system.

## Legacy System Retirement

This section is only applicable to projects that are 1A and 2A type projects that are replacing a legacy system. Other categories of projects can state that this section is not applicable to their category of project.

In Table 4, identify each of the applications or application components that will be retired or will have their workloads significantly reduced as a result of the design and deployment of this system.

Table 4: Proposed Legacy Retirements

| Legacy System or  Legacy System Component | System Retired or  Workload Reduced | Quantify the Workload Reduction |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# 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.



Figure 2: 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

Object

| ID | Name | Description | Interface Name | Interface System |
| --- | --- | --- | --- | --- |
| < ID from diagram> | Improved Risk Model | Risk model to predict risk of suicide for veterans | <Name of each of the Interfaces to this object> | <Systems with which this system interfaces> |

The main goal of the modeling effort is to continue to utilize and improve the existing suicide completion risk model developed by Dr. Bossarte and his analytics team so that (1) near-time data on veterans can be used to predict the risk of suicide completion for a specific veteran and (2) the VA Suicide Prevention Coordinators and outreach staff are notified of veterans with elevated risk levels for suicide and can take appropriate preventative actions. The current risk model uses a logistic regression framework and approximately 380 inputs from VHA and NDI data. Using this model as a starting point, potential improvements will be considered and implemented if they are shown to improve the out-of-sample predictive power of the existing model. Potential improvements may include but will not be limited to: new data sources, alternative input variable definitions, testing of variable interactions, and alternative model structure.

The risk model uses a logistic regression model with approximately 380 input variables. This model is currently being reviewed to understand variable data sources and definitions and model development and testing processes. Once the review is completed, new data sources can be utilized and variables constructed and tested to determine if they can be used as enhancements to the current risk model. Also, alternative methods to logistic regression may be tested to understand if they improve the predictive power of the current risk model.

Interfaces External to OIT

| 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> |

Interfaces Internal to OIT

| 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> | VA Suicide Data Repository (SDR) | Periodoically imports data from 4 VA data sources:   1. National Death Index (NDI) Mortality Dearch Results 2. State Death Certificate Data 3. Suicide Prevention Applications Network (SPAN) 4. Veterans Crisi Line | VA Office of Public Help | Read access |

### 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 3: 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> |

### Application Locations

Use Table 7 to specify the locations at which the application components will be hosted.

Consideration should be given to adopt cloud technologies as potential solutions. Leveraging cloud technologies is part of a larger effort by the Office of Management and Budget (OMB) to reform Federal IT Management. Considerations such as regional deployments etc. should be documented in this section.

Table 7: Application Locations

| Application Component | Description | Location at Which Component is Run | Type |
| --- | --- | --- | --- |
| <Component name> | <Description> | <Facility name> | <Presentation Logic/Business Logic/Data Logic/Interface Code> |

Table 8: Application Users

| Application Component | Location | User |
| --- | --- | --- |
| <Component name> | <Facility name> | <Role> |

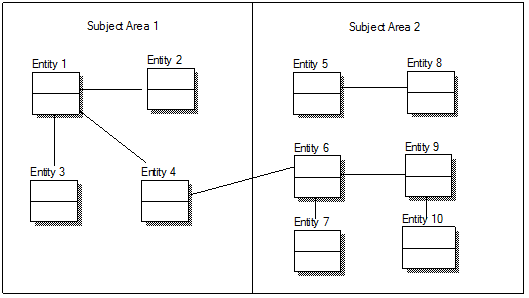
## Conceptual Data Design

### 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 4: 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 |
| --- | --- | --- | --- |
| <Reach | SQL Server database(s) that will   1. Import data from SDR and other internal/external sources 2. Store Analytics output to be used by IRDS dashboard ad messaging | Create | ? |
| SDR | See section 3.1.1, Externally Shared Data Stores table | Interface |  |

### 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.

TBD – The PwC Analytics team is currently working with the VA to determine:

* Which VA sources will be imported into the Reach database
* Which, if any external sources will be imported into the Reach database
* What data elements from SDR will be imported into the Reach database

#### 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 5: *<screen name>* Screen

Table 10: *<screen name>* Screen Description

| Graphical User Interface (GUI) Field | Table (Database Table that field connects to) | Field (Field in Table that the GUI field connects to) | Comments |
| --- | --- | --- | --- |
| <Name> | <xxx> | <PATIENT\_ NAME> | <Add any comments or descriptive information that would be relevant to the tester> |
| <SSN> | <xxx> | <SSN> |  |
| Date of Birth (Age) | yyyy | DATE\_OF\_BIRTH DATE\_OF\_DEATH (if deceased) |  |

#### Application Report Interface

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

##### <Insert name of report>

<Create a new subsection for each report> Figure 6 represent <name> screen and Table 16 describes it…

Figure 6 represents the <report name>; Table 11 describes it. Paste a screenshot of the report below and complete the table to describe the report.

Figure 6: *< Report name>* Report

Table 11: *<Report name>* Description

| Report Column | Data Source *<TableName. FieldName>* |
| --- | --- |
| Patient | <xxx.PATIENT\_NAME> |
| SSN | <xxx.SSN> |
| DoB | <yyyy.DATE\_OF\_BIRTH> |

#### 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.

## Conceptual Infrastructure Design

The Conceptual Infrastructure Design should describe any unique technology that will be used, which are either part of this system, or will attach to this system.

All information should be provided to the extent that it is known. Because the system is at a preliminary design stage, it is expected that the information provided may need to be changed during later design stages or increments.

The Conceptual Infrastructure Design is a high-level overview of the infrastructure that will be used to support the application. Primary emphasis is on the environments that will be required and the locations at which they will be installed. The Conceptual Infrastructure Design becomes more detailed at later stages as more information is collected regarding the system, and the infrastructure requirements (i.e., capacity requirements) are better known.

### System Criticality and High Availability

Describe the approach that will be taken to meeting the system criticality and high availability requirements identified in Section 2.5.6, including the extent to which geographically distributed, high availability designs are planned. Describe the approach that is taken towards high availability as well as any workload distribution scheme that is planned to support the high availability implementation (e.g., restricting updates to a single node).

If the system is not mission critical and high availability is not required, then describe the approach that will be taken to provide the requisite level of availability and disaster recovery.

### Special Technology

If any special technology was identified in Section 2.5.9 as part of this system, describe the device and the type of location at which it will be installed. This information may be provided using Table 12.

Table 12: Special Technology Requirements

| Special Technology | Description | Notional Location | TRM Status |
| --- | --- | --- | --- |
| <Name> | <Business language description> | <At what type of location will this technology be deployed?> | <Is this technology in the TRM?  (Yes / No)> |

### Technology Locations

This section describes the various technology components that will be used. If known, provide the name of the datacenter at which the technology will be installed. If not, specify as Site A, Site B etc. Provide this information in Table 13.

Table 13: Technology Location Details

| Technology Component  Production 1 | Location | Usage |
| --- | --- | --- |
| Workstations |  |  |
| Special Hardware |  |  |
| Interface Processors |  |  |
| Legacy Mainframe |  |  |
| Legacy Application Server |  |  |
| Legacy Databases |  |  |
| Other |  |  |

| Technology Component  Production 2 | Location | Usage |
| --- | --- | --- |
| <copy from Prod 1 set, or enter new ones as appropriate> |  |  |

| Technology Component  Certification | Location | Usage |
| --- | --- | --- |
|  |  |  |

| Technology Component  Education | Location | Usage |
| --- | --- | --- |
|  |  |  |

| Technology Component  Test | Location | Usage |
| --- | --- | --- |
|  |  |  |

| Technology Component  Development | Location | Usage |
| --- | --- | --- |
|  |  |  |

### Conceptual Infrastructure Diagram

#### Location of Environments and External Interfaces

Create a diagram to show the environments that will be supported. As illustrated in Figure 7, the diagram should show the following:

* Local networks to which they will be attached (Production, Test, or Development)
* Locations at which they will be installed
* External connections (each external interface should be shown in terms of where it enters the network).



Figure 7: Sample Conceptual Networks and Environments

#### Conceptual Production String Diagram

Create a diagram to show the configuration of a single production string to the extent that it is known. It is likely that this diagram will be highly notional and may show such items as enterprise service bus, application servers, and database servers.

Additional components, such as the mainframe, other Web servers, or other major components should be included if they are expected to be required.

Figure 8: Conceptual Production String Diagram

## 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.

The development process and architecture is characteristic of the following key components:

* First, the infrastructure follows a principle of automation. This automation includes compile/build/packaging; infrastructure deployments onto local developer workstations; and infrastructure deployments in test and production.
* Second, this system architecture favors a lightweight infrastructure rather than a large unmanageable infrastructure. This manifests itself in the use of small autonomous components, rather than a traditional application server.

The system developed under the Perceptive Reach (IRDS) project will be designed to run on a cloud-based environment.

## 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.

The program will be setup within the VA’s EO cloud environment.

Initial Requirments (Cloud Based – Prototype Server):

* MS Windows Server 2012 64-bit
* Intel Xeon CPU E5-2670, 2.6GHZ
* 16GB RAM
* 60GB HDD
* MS SQL Server 2012 Enterprise Edition
* Users: 16 (8 simultaneous logins, minimum)

Planned Requirements (Cloud Based – Development/Test Server):

* MS Windows Server 2012 64-bit
* Intel Xeon E5-2600 Family (2670 or 2690), 2.6GHZ or better
* 32GB RAM
* 500GB HDD
* MS SQL Server 2012 Enterprise Edition
* Users: 16 (8 simultaneous logins, minimum)

Future Requirements:

* TBD

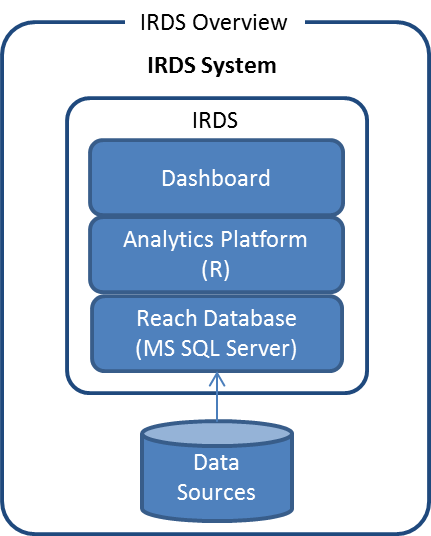
For further details on the Hardware Architecture, please refer to section 5.1 Hardware Detailed Design.

## 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.

The IRDS innovation will develop and demonstrate a new SQL database that aggregates both VA and non-VA data sources to be used to facilitate identification of at-risk individuals and populations, an integrated data analytics solution that includes open source data analysis and visualization tools, and an open standard based secure messaging solution to inform authorized individuals of analysis results.



**Reach Database:**

* Platform: SQL Server 2012
* Data Importing and ETL Tools: SQL Server Integration Services, FMQL or RPC (for interfacing directly with VISTA)

**Analytics Platform:**

The enhanced risk model will be coded in the open source statistical language R. Once an enhanced model is finalized, the model coefficients will either stay in R or be transitioned into SQL code. If the model remains in R, this may make running the model daily potentially a more difficult process due to lengthy data processing times. The alternative to this possible issue might be to store the data processing and model coefficients in SQL. If this approach is used, R would still be used at a specified frequency to refresh the parameter estimates of the risk model.

* R 3.1.2 (a language and environment for statistical computing and graphics)
* RStudio (a powerful and productive user interface for R)

**Dashboard:**

The role of the dashboard is to provide visualization tools that display the current status of metrics and key performance indicators (KPIs) for IRDS. The dashboard will consolidate and arrange preselected metrics and analytic results on within a browser based web application. Node.js will provide the base architecture for the web server providing the primary services of the dashboard through HTML, Javascript, and CSS to drive the behavior of the application.

* Node.js (JavaScript runtime engine)
* Express.js (Node.js module used to handle routing of HTTP calls)
* HTML, JavaScript, and CSS
* Angular.js (JavaScript framework for extending HTML)
* Bootstrap (powerful front-end framework for faster and easier web development)

For further details on the Software Architecture, please refer to section 5.2 Software Detailed Design.

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.

## Network Architecture

Describe communications within the system, such as local area networks (LANs) and buses. Include the communications architecture(s) being implemented, such as X.25 and token ring.

Provide a diagram depicting the communications path(s) between the system and subsystem modules.

## 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.

TBD:

At this time assumptions are:

1. SDR Data will be pulled directly from the SDR server/databases via a SQL connection and querying of the data
2. VBA data will pulled from SQL tables located at Corporate Data Warehouse (CDW) via a SQL connection and querying of the data
3. VISTA data will be accesed and imported through either currently exiting and/or custom RPC calls, or FMQL.
4. It is possible that some data sources may be interfaced by other means such Web Service calls or file formats such as FLAT files.

## 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.

The strength and robustness of a predictive model is dependent on the data inputs that are used to develop the model. Thus, identifying a robust and reliable list of potential data inputs is essential. By reviewing academic literature and holding discussions with clinicians and other subject matter specialists on both risk and protective factors for suicide ideation and completion, a number of variable categories have arisen as critical inputs into the risk model: demographics, clinical, and socioeconomic. It is expected that the list of specific data inputs will increase as academic literature review and discussions with subject matter specialists continue.

After identifying the data points potentially useful for model development, data sources will need to be identified that capture the information on relevant variables. The current risk model uses data from VHA and NDI. It is expected that some of the additional data inputs will be derived from the same VHA and NDI data sources; however, additional data inputs will be considered from new data sources. One of the data sources that may be useful for model development is VBA. From the initial discussions with subject matter specialists on the VBA data, there are multiple datasets at VBA that contain information on veterans including: military service, financials, medical, and demographics.

Model input data will be housed within a SQL Server database behind a VA firewall. To be able to access the data, R statistical software will connect directly to the database to access the data for analysis.

The ultimate output from the enhanced risk model will be a veteran-specific risk score based on the predicted likelihood of suicide risk. The risk score may be a numeric score bound between two limits, an unbounded numeric score, an ordinal categorical score (for example: High, Medium, Low), or a unordered categorical variable (for example: PTSD, Divorce, Family Death, etc). The final decision on risk score methodology will be made after the risk model is finalized and dashboard/notifications end users are consulted.

The outputs from the risk model will be stored in the SQL Server instance that also houses the raw data inputs into the model. Since the exact output of the statistical model is not yet defined, there is no definition as to the exact storage protocol of model outputs, only that they will be stored in the SQL Server. The logic behind storing the model outputs in the SQL Server is to allow the dashboard and messaging applications to pull these data from a single source rather than multiple sources.

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.

TBD – Requirements are currently being gathered to determine what data will be imported into the IRDS system and how the data model will be designed to store the data. It has been decided that the are 2 components to the data

1. Tables to store source data that will be imported into the Reach database
2. Tables to store anlysis output from R programs which have code to encapsulate a risk model and a surveillance model

Each of the data components above will be conainted in their own schema. The source data tables will be stored in the default .dbo schema, and the analysis output tables will be stored in the ‘Analytics’ schema. An additional schema may be added for system tables and temp tables accessed by processes such as SSIS packages.

## 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.

TBD –At this point it is assumed that some VHA data will be imported from the VISTA system, which uses A MUMPS data store and will be accessed through either RPC calls and/or FMQL.

## 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>>

TBD – Requirements are currently being gathered for which data sources and data elements from those sources will be imported into the reach database and what the relationship will be between those elements when they are stored in the database.

# 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 the Requirements Specification Document. These should be captured in the Requirement Traceability Matrix (RTM).

TBD – Requirements are currently being gathered.



## 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

TBD – Requirements are currently being gathered

## 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.

TBD – Requirements are currently being gathered

### Conceptual Design

This section introduces the conceptual information that establishes the basis for how the software will be built.

#### Product Perspective

This subsection of the SDD should put the product into perspective with other related products. If the product is independent and completely self-contained, it should be stated here. If the SDD defines a product that is a component of a larger system, 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.

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

Sections of the Requirements Specification Document (RSD) can be referenced in the subsections, if applicable.

##### User Interfaces

This subsection should specify the logical characteristics of each interface between the software product and its users. This includes those configuration characteristics necessary to accomplish the software requirements (e.g., screens, roll and scroll, GUI interface).

Recommendation: Create a block diagram showing the user interfaces.

##### Hardware Interfaces

This subsection should specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics (for example, hardware platform or mainframe versus personal computer). It also covers matters such as what devices the system will support, how they will be supported, and protocols. Examples include scanners, pen driven devices, and radio frequency devices.

Recommendation: Create a block diagram showing the hardware interfaces.

##### Software Interfaces

This subsection should specify the use of other required software products (e.g., VA Kernel, VA FileMan, Windows NT); and interfaces with other applications or other systems such as commercial off-the-shelf (COTS) or national databases. Specify the application interfaces (e.g., the linkage between an accounts receivable system and a general ledger system and a COTS software package that will be interfaced using an existing interface). This section should provide the following information for each required software product:

* Name
* Version number
* Discussion of the purpose of the interfacing software as related to this software product
* Definition of the interface in terms of message content and format (e.g., Health Level Seven [HL7], electronic data interchange).

##### Communications Interfaces

This subsection should specify the various interfaces to communications such as local network protocols, e-mail, Transmission Control Protocol (TCP), modems.

Recommendation: Create a block diagram showing the communications interfaces.

##### Memory Constraints

This subsection should specify any applicable characteristics and limits on memory or partition size.

##### Special Operations

This subsection should specify the special operations required by the user such as backup, recovery, and archiving operations.

This section should also include any operations for external devices or COTS systems.

#### Product Features

This subsection should provide a summary of the major features of the software.

For example, an SDD for an accounting program might use this section to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those features requires.

Note: For clarity, remember these items when creating this section of the SDD:

* The features should be organized in a way that makes the list of features understandable to the customer or to anyone else reading the document for the first time.
* Textual or graphical methods can be used to show the different features and their relationships.
* Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.

#### User Characteristics

This subsection should describe the general characteristics of the intended users of the product, including experience and technical expertise. It should not be used to state specific requirements but rather should provide the reasons why certain specific requirements are specified in the RSD.

#### Dependencies and Constraints

This subsection should provide a description of any other items that will limit the developer’s options. The following list includes items that limit the developer’s options.

* Regulatory policies
* Hardware limitations (for example, signal timing requirements)
* Interfaces to other applications
* Parallel operation
* Audit functions
* Control functions
* Higher-order language requirements
* Reliability requirements
* Criticality of the application
* Safety and security considerations
* Usability (including 508 compliance)

This section of the SDD should contain all the software design to a level of detail sufficient to enable programmers to develop a system that satisfies the requirements defined in the RSD. It should be detailed so as to make it easy for technical staff to find the methods to complete the designed function.

These requirements should, at minimum, include the following items:

* An indication of the associated requirement(s) in the RSD which is being designed
* A description of the functionality being designed
* The design entities (and their attributes) affected
* The algorithm executed (where appropriate) to implement the functionality.

Because the Dependencies and Constraints section is often the largest and most important part of the SDD, the following principles apply:

* Specific design should be cross-referenced to earlier, related documents (e.g., the RSD).
* All design should be uniquely identifiable.
* Items in this section should be identified from a technical level rather than an end user level. (i.e., an option name should be identified rather than the menu text for that option).

### Specific Requirements

#### 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.

TBD – see section 5.1

#### System Features

Describe the system features, functional requirements, sub-requirements, etc. which can be organized in an outline format that matches the RSD. Specific formatting and organization of the paragraphs (i.e., section numbering) is left to the discretion of the author and is dependent on the level of detail essential to fully describe the design. Some designs may only require two levels; others may require multiple levels. The information necessary to define the items or to specify modifications to the items affected by the functionality being designed should be provided in the appropriate design element tables. Where feasible, instead of duplicating the RSD, it can be referenced via a link, to avoid unnecessary duplication. The key goal is to provide traceability to requirements.

#### Design Element Tables

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** |  |

##### Bulletins

If the project develops or affects bulletins, then complete this section; if not then state that the section is not applicable and delete the tables and content of the section. Complete the table for each bulletin affected by the functionality being designed. A short description of what change will be made to the bulletins 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 18: Bulletins (Instructions)

|  |  |
| --- | --- |
| Bulletins | Instructions |
| **Bulletin Name** | List the specific bulletin 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 send the bulletin. |
| **Related Routines** | List routines that directly send the bulletin. |
| **Mail Subject** | List the subject of the mail message, i.e., which bulletin this affects. |
| **Mail Group** | List the mail group (recipients) of the mail message. |
| **Parameters** | List necessary parameters. |
| **Data Dictionary (DD) References** | List files/fields that reference the bulletin(s) through input transforms, cross reference logic, etc. should be listed under Data Dictionary (DD) References. |

Table 19: Bulletins

| Bulletins | Description | | | |
| --- | --- | --- | --- | --- |
| **Bulletin Name** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |

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

| Routines | Description |
| --- | --- |
| **Mail Subject** |  |
| **Mail Group** |  |
| **Parameters** |  |
| **Data Dictionary (DD) 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** |  | | |

##### Constants Defined in Interface

Provide the name and description.

Table 33: Constants Defined in Interface

| Name | Description |
| --- | --- |
|  |  |

##### Variables Defined in Interface

Provide the name, type, and description.

Table 34: Variables Defined in Interface

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Types Defined in Interface

Provide the name, type, and description.

Table 35: Types Defined in Interface

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### GUI

List the GUI affected by the functionality being designed and include a short description of the changes made to the affected GUI. The headers in the following tables have names for the information outlined. There are a number of items in this section that would generally be global information and visible to all other aspects.

Table 36: GUI

| Unit Name | Description |
| --- | --- |
|  |  |

##### GUI Classes

Table 37: GUI Classes (Instructions)

| GUI Classes | Instructions |
| --- | --- |
| **Class Name** | List the name of the class affected. The headers in the following tables have names for the information outlined. Note that only the new properties and methods for a class are listed below. All ancestor properties and methods are still available and unchanged. |
| **Derived From Class** | List the class that this is derived from, its parent and any interfaces listed as part of this class. |
| **Purpose** | Describe the functionality that users can access from this class and related form, if any. |

Table 38: GUI Classes

| GUI Classes | Instructions |
| --- | --- |
| **Class Name** |  |
| **Derived From Class** |  |
| **Purpose** |  |

##### Current Form

Provide a screen capture or graphical representation of the current layout.

##### Modified Form

Provide a screen capture or graphical representation of the layout that the design will implement.

##### Components on Form

Table 39: Components on Form

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
|  |  |  |

##### Special References

Include references that are not listed elsewhere.

| Special Reference Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Class Events

Table 42: Class Events

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Class Methods

Table 43: Class Methods

| Name | Procedure/Function | Description |
| --- | --- | --- |
|  |  |  |

##### Class Properties

Table 44: Class Properties

| Class Properties Name | Type | Visibility | Description |
| --- | --- | --- | --- |
|  |  |  |  |

##### Uses Clause

Use this section to provide a uses clause that lists the other units (code or form units) that this unit will use. This may be documented in the form of a Unified Modeling Language (UML) drawing.

##### Forms

This section lists the forms that will be affected or created by the functionality being designed. A short description of the change that will be made to the forms should be included.

Table 45: Forms (Instructions)

| Forms | Instructions |
| --- | --- |
| **Form Name** | List the name of the form affected by the functionality being designed. |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **Form Functionality** | Describe the form’s functionality and refer to the usage of the form. An example of such a description is “This form is used to enter patient demographic data.” |
| **Current Form Layout** | Define the current form layout that the design will modify. If this is a new form, enter “N/A”. |
| **Modified Form Layout (Changes are in bold)** | Define the form layout that the design will implement. |

Table 46: Forms

| Forms | Description | | | |
| --- | --- | --- | --- | --- |
| **Form Name** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Form Functionality** |  | | | |

| Current Form Layout |
| --- |
|  |

| Modified Form Layout (Changes are in bold) |
| --- |
|  |

##### Functions

The functions affected by the capabilities being designed should be listed in this section. A short description of what change will be made to the functions and/or new functions should be included.

Table 47: Forms (Instructions)

| Functions | Instructions |
| --- | --- |
| **Function Name** | List the specific function affected by the capability being designed. |
| **Short Description** | List a short description of the change that will be made to the functions and/or new functions. |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **Related Options** | List the options that directly call or are called by the function. |
| **Related Routines** | List the routines that directly call or are called by the function. |
| **Data Dictionary (DD) References** | List the files that reference the function through input transforms, cross reference logic, etc. |
| **Related Protocols** | List the protocols that reference or are referenced by the function. |
| **Related Integration Control Registrations (ICRs)** | List proposed new ICRs and subscribed ICRs. Also, list any obscure Supported ICRs. |
| **Data Passing** | Check the appropriate box. An event that would trigger the new/changed function should be included in this section. An example of such a description would be a note that the new/changed function will be invoked as part of a function call or it would be invoked through 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 function logic. Each attribute should be defined. |
| **Output Attribute Name and Definition** | List the output attributes returned from the new or changed function logic. Each attribute should be defined. |
| **Current Logic** | Define the current logic in the function that the design will modify. If this is new code, enter “N/A”. |
| **Modified Logic (Changes are in bold)** | Define the logic in the function that the design will implement. |

Table 48: Forms

| Function Name | Activities | | | |
| --- | --- | --- | --- | --- |
| **Short Description** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** |  | | | |

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

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

| Current Logic |
| --- |
|  |

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

##### Dialog

In this section list the changes to the DIALOG file (#.84).

Table 49: Dialog (Instructions)

| Dialog | Instructions |
| --- | --- |
| **Dialog Message (Description)** | List the specific message affected or needed by the changes being designed. |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **Dialog Message (Description) Condition** | Describe the dialog message (description) functionality. An example of such a description would be the condition that would trigger the output of the message (dialog). This section refers to the condition generating the message (dialog). |
| **Current Dialog Message (Description)** | Define the current dialog message (description) that the design will modify. If this is a new dialog message (description) enter N/A. |
| **Modified Dialog Message (Description)  (Changes are in bold)** | Define the dialog message (description) that the design will implement. |

Table 50: Dialog

| Dialog | Instructions | | | |
| --- | --- | --- | --- | --- |
| **Dialog Message (Description)** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Dialog Message (Description) Condition** |  | | | |
| **Current Dialog Message (Description)** |  | | | |
| **Modified Dialog Message (Description)  (Changes are in bold)** |  | | | |

##### Help Frame

A short description of what change will be made to the Help Frame text and/or new text should be included in this section. Help frames may be associated with options or with data dictionary fields to provide on-line instruction.

Table 51: Help Frame (Instructions)

| Help Frame | Instructions |
| --- | --- |
| **Help Frame Text** | List the text affected or needed by the changes being designed. |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **Help Frame Text Calling Mechanism** | Provide a short description of the mechanism used to call the Help Frame text in this section. An example of a mechanism would be the name of the routine or an explanation of how the Help Frame is called. An example of a calling mechanism would be the Standard VA FileMan API and the keystroke(s) that would trigger the output of the text. |
| **Current Help Frame Text** | List the current Help Frame Text that the design will modify. If new text enter N/A. |
| **Modified Help Frame Text (Changes are in bold)** | List the Help Frame Text that the design will modify. |

Table 52: Help Frame

| Help Frame | Description | | | |
| --- | --- | --- | --- | --- |
| **Help Frame Text** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Help Frame Text Calling Mechanism** |  | | | |

| Current Help Frame Text |
| --- |
|  |

| Modified Help Frame Text (Changes are in bold) |
| --- |
|  |

##### HL7 Application Parameter

Table 53: HL7 Application Parameter (Instructions)

| HL7 Application Parameter | Instructions |
| --- | --- |
| **HL7 Application Parameter Name** | List the HL7 Application Parameter affected or needed by the changes being designed. |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **Application Status** | Check the appropriate box in the applicable column for Current and Modified |
| **Facility Name** | List the current and modified value in the appropriate column. |
| **Country Code** | List the current and modified value in the appropriate column. |
| **HL7 Field Separator** | List the current and modified value in the appropriate column. |
| **HL7 Encoding Characters** | List the current and modified value in the appropriate column. |
| **Mail Group** | List the current and modified value in the appropriate column. |

Table 54: HL7 Application Parameter

| HL7 Application Parameter Name | Description |
| --- | --- |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Enhancement Category** | New | Modify | Delete | No Change |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Application Status** | Active | Inactive | Active | Inactive |

| Enhancement Category | Current | Modified |
| --- | --- | --- |
| **Facility Name** |  |  |
| **Country Code** |  |  |
| **HL7 Field Separator** |  |  |
| **HL7 Encoding Characters** |  |  |
| **Mail Group** |  |  |

##### HL7 Logical Link

Table 55: HL7 Logical Link (Instructions)

| HL7 Logical Link | Instructions |
| --- | --- |
| **HL7 Logical Link Parameter (LLP) Name** | List the specific HL7 Logical Link affected or needed by the changes being designed. |
| **Enhancement Category** | Check the appropriate box: New, Modify, Delete, or No Change. |
| **Node** | List the current and modified value in the appropriate column. |
| **Institution** | List the current and modified value in the appropriate column. |
| **Domain** | List the current and modified value in the appropriate column. |
| **Autostart** | List the current and modified value in the appropriate column. |
| **Queue Size** | List the current and modified value in the appropriate column. |
| **LLP Type** | List the current and modified value in the appropriate column. |

Table 56: HL7 Logical Link

| HL7 Logical Link | Description |
| --- | --- |
| **HL7 Logical Link Parameter Name** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Enhancement Category** | **New** | **Modify** | **Delete** | **No Change** |

| Enhancement Category | Current | Modified |
| --- | --- | --- |
| **Node** |  |  |
| **Institution** |  |  |
| **Domain** |  |  |
| **Autostart** |  |  |
| **Queue Size** |  |  |
| **LLP Type** |  |  |

##### COTS Interface

The specific communication method(s) and Application Interface(s) that will be created or modified for the COTS system being interfaced should be described in this section. A short description of the existing tools that will be used and any new tools that will be developed should also be included.

Table 57: COTS Interface (Instructions)

| COTS Interface | Instructions |
| --- | --- |
| **Communication Method** | List the specific communication method created or modified for the functionality being designed. |
| **Application Interface** | List the specific application interface created or modified for the functionality being designed. |

Table 58: COTS Interface

| COTS Interface | Description |
| --- | --- |
| Communication Method |  |
| Application Interface |  |

## Network Detailed Design

Provide enough detailed information about the communication requirements to build and/or procure the communication components for the system. This section should provide sufficient detail to support the procurement of hardware for the system installation. Include the following information in the form of detailed designs (as appropriate):

* Details of servers and clients to be included on each area network
* Specifications for bus timing requirements and bus control
* Format(s) for data being exchanged between components
* Diagrams showing connectivity between components, data flow (if applicable), and distances between components
* LAN topology.

## 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.

#### Introduction

##### Purpose and Scope of Service

This service was described at a high level in the charter document. Please refer to it here via a link.

##### Links to Other Documents

Provide links to other documents created for this service so far in the SOA lifecycle. At a minimum, provide links to:

* Service Charter
* Service Roadmap
* Service Description

#### Service Details

##### Service Identification

This section will be written as a table to provide a quick reference to the service's what, where, why and how - cheat sheet.

| Service Attribute | Value |
| --- | --- |
| Name and Alias (if any) | Name of the service and other names for the service, which might be used by someone searching for this service. Please follow ESS naming standards. |
| Overview | Brief textual overview of the service. |
| Version | Version number of the service being described here |
| Latest Status | This field shows the latest status for the above referenced version of this service! The status of a service shows the progress of the service from initiation through development, deployment, and eventual retirement. The status also has a status date associated with the status - and we will be using the latest one here in this document. Valid values include: Inception, Design, Provisioning, Certification / Testing, Operation, Deprecated, Retired, Rejected - Owner has decided not to develop the service. |
| Service Type | Used to define applicable architecture patterns. Examples (from Open Group):  • Interaction  • Process  • Information  • Partner  • Business Application  • Access  • Service Connectivity |
| Architecture Layer | Referred to as class in VA Service template. Used to define applicable architecture patterns and relationships to governing bodies. Examples:  • Solution  • Process  • Information  • Utility  • Underlying |
| Business Domain | Business Vertical or Business Division where this service belongs. |
| Service Domain | The service or technical domain that the service belongs to. Can be used to establish the namespace. |
| Business Organization and Owner | Person who approves this service & any changes. Include email. |
| Technical Organization and Owner | Person responsible for provisioning (specifying, acquiring certifying) this service. Include email. |
| Development Organization and Owner | Person who is responsible for the development processes and activities for this service. Include email. |
| Support Organization and Owner | Person who is responsible for the support of this service while in production. Include email. |
| Target Consumer Organization(s) and Owner(s) | Organizations and/or developers roles that service is intended for. |

##### Service Versions

|  |  |  |
| --- | --- | --- |
| Version Numbers | Current Status of Version | A Brief Description of the change implemented in that version |
| This version | Being Designed |  |
| Example: version 2 | Example: In production. Will be retired with this release. | Example: This release added the ability to look up a person by address.  Provide a link to each version of the service. |
| Example: version 1 | Example: Retired. | Example: This release provided the base minimum functionality to look up a person by name.  Provide a link to each version of the service. |

##### Summary of Design and Platform Details

###### SOA Pattern(s) Implemented

Name of the SOA pattern implemented – for instance, this may be a Pub/Sub model. Just a name and reference to the document or book with the pattern is sufficient for popular patterns or VA's own patterns. If you are using some esoteric pattern, more details will help.

###### COTS Platform vendor names and versions for hosting platform

Example, TIBCO.

#### Dependencies

The Dependency Model identifies other services, systems, databases, etc. that [Service Name] is dependent upon or interacts with to perform its function.

This section should clearly identify all sources and external systems that are accessed by this service to fulfill the service consumers’ request. This section should include diagrams to show as much detail as necessary to inform the developer. Provide a context diagram for the service.

Note: Here our primary audience includes the providers of the service. So this document in general will emphasize system components and sub-systems as much as external interactions.

#### Service Design Details

The next sub-section on Interface Technical Specs **could be** just a copy from the corresponding sub-section in Interface section in the Service Description Document. Here, you could provide more detail necessary for building this service but **the interface spec needs to be consistent between this document and the Service Description Document**. This section contains all information necessary to fully describe an interface published by this service...

##### Interface Technical Specs

The technical specification allows developers of service consumers to locate and discover the service for run time consumption.

###### Service Invocation Type

Such as: SOAP over HTTP, REST.

###### Service Interface Type

Such as: WSDL via Web Service 2.0

###### Service Name

Technical Service Name. Comply with ESS naming standards.

###### Interface

Link to WSDL or other interface document.

###### End Points

Provide if known! Calls that can be made into the service. Can be referenced to the WSDL or can be in a separate table.

###### Operations or Methods

In the table below, the technical names of the operations, inputs and outputs are used. Inputs and outputs, if parameters, must have a data type.

Non-primitive data types must be defined in the Service Information Model section.

This table could be generated automatically from the WSDL content or its equivalent.

Style can take any of these values: Parameters or Document; and One-way or Request-response or Solicit-response or Notification.

Use a separate column for the operation purpose if you wish.

You might use abbreviations in the Faults column and explain the abbreviations used below the table. For example, NF = Not Found, MI = Missing Input.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation Name | Inputs | Outputs | Transactional Qualities if relevant (Updating?, Atomic?, Can participate in transaction?) | Pre and Post Conditions | Exception (s) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Provide a link to the Service Information model so that the consumer of your system knows the schema for the input and output parameters.

###### Message Schemas

Provide definitions or links to definitions of the message(s) related to the service operations. These may be dependent on the implementation style and protocol binding of the interface.

##### Information Model

Even though this section looks similar to the corresponding section 3.2 in Service Description, remember that the primary objective here is to facilitate construction and to gain approvals from governing bodies. So you will provide more of a “white box” view of the design here to help your developers code the service.

###### Class Diagram and Description of Entities Involved

Map out all entities involved in the service: input, output, exceptions, entities manipulated in persistent media/DBs, intermediate entities created in memory etc.

###### Mappings from ELDM to Standards Based Schemas

Provide mappings from your native schema to any standards based schemas your service will use to communicate outside. For instance, if you are using HL7 based messages then you will show how data is converted from your native schema to HL7.

##### Behavior Model (AKA Use Case Realization)

The Behavior Model defines the actions and processes supported by the service. Actions and methods represented in the use cases and sequence diagrams shown below are further defined by the operation contracts and the message payloads.

###### Use Cases (Use Case Model)

How does this service fit into the larger use case model of the consumer? You may need multiple models for multiple consumers. Focus is **not** on the internal workings of the new service instead of the calls made from external consumers. Just a summary or the Use Case Diagram may be sufficient. List the alternative and exception flows. Reference the detailed design documents via a URL.

###### Interaction Diagrams

Cut and paste screen shot from RSA or similar tool or provide link to the model. Provide description to help developers build your service. The interaction diagrams should depict external interactions and internal sequences of calls between internal components. The sequence diagram should cut through all layers to show the main, alternate and exception flows.

#### Gap Analysis

Provide a Gap Analysis (Reference) to demonstrate compliance of this service with various standards, policies, guidelines and laws. The Gap Analysis may take the form of a matrix as shown in the sample below. This will help the governance boards expedite your request.

| Design Elements🡪  Policies / SLD elements etc↓ | Design  Element A | Design  Element B | Design  Element C | Comment for non-conformance |
| --- | --- | --- | --- | --- |
| Policy X | Match |  |  |  |
| Policy Y |  | Partial |  |  |
| Policy Z |  |  |  | Commercial encryption server in prod will have to address this policy. |
| Policy A |  |  |  | Compliance with this policy not required until next year. |
| New / Additional Features |  |  | New element minimizes manual intervention |  |

##### Variances from Enterprise Target Architecture

This list of “variances” will become a submission to the ESS dispensation process.

##### Variances from SLDs

This list of “variances” will become a submission to the ESS dispensation process.

##### Variances from Standards and Policies

This list of “variances” will become a submission to the ESS dispensation process.

##### Justification for Exceptions and Mitigation

This section will list out any non-functional and functional requirements that are not being met. The non-conformance may be in violation of elements of SLDs, enterprise architecture (TRM Technology Reference Model), privacy policies or guidelines. For each exception provide:

1. Reasons for non-conformance (cost, time, technology, etc.)
2. Mitigating actions taken to reduce the impact of non-conformance
3. Plan (roadmap) to come back into conformance

This list can grow depending on what the Review bodies may ask for.

# 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.)

# Human-Machine Interface

Describe the human-machine interface (i.e., GUI) relative to the user. Additional information may be added if the suggested headings are inadequate.

## Interface Design Rules

Identify conventions and standards for designing the GUI.

## Inputs

Identify the input media used by the user (i.e., operator) for providing information to the system, such as data entry screens, optical character readers, bar scanners, etc.

Identify the messages associated with operator inputs, including the following:

* Form(s) if the input data is keyed or scanned for data entry
* Access restrictions
* Security considerations.

## Outputs

Describe the system output design relative to the user. System outputs include reports, data display screens, query results, etc.

Identify the following, if appropriate:

* Access restrictions or security considerations
* Description of the purpose of the output
* Report requirements, including frequency of periodic reports
* Screen contents. (Provide a graphic representation of each layout. Define all data elements associated with the layout).

## Navigation Hierarchy

Provide a diagram of the navigation hierarchy that shows how a user moves through the GUI.

### Screen [x.1]

Provide the layout of all input data screens or GUIs. Provide a graphic representation of each GUI, for example, a low-resolution screenshot. Define all data elements associated with each screen or GUI, or reference the data dictionary. Label each data input screen and/or GUI.

### Screen [x.2]

Provide a graphic representation of each GUI, for example, a low-resolution screenshot. Define all data elements associated with each screen or GUI, or reference the data dictionary.

### Screen [x.3]

Provide a graphic representation of each GUI, for example, a low-resolution screenshot. Define all data elements associated with each screen or GUI, or reference the data dictionary.

# 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).

Attachment A – Approval Signatures

This section is used to document the approval of the System Design Document. The review should be conducted face to face where signatures can be obtained ‘live’ during the review. If unable to conduct a face-to-face meeting then it should be held via LiveMeeting and concurrence captured during the meeting. The Scribe should add /es/name by each position cited. Example provided below.

The Chair of the governing Integrated Project Team (IPT), Business Sponsor, IT Program Manager, and Project Manager are required to sign.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

< Integrated Project Team (IPT) Chair >

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

< Business Sponsor >

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

< IT Program Manager >

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

< Project Manager >

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 |
| --- | --- |
|  |  |
|  |  |
|  |  |

* 1. Required Technical Documents

The following documents must be submitted for review to support proper approval:

* Conformance Validation Statement (CVS) - Section 508
* For additional information regarding how to obtain proper approval for this project, refer to the following documents:
* IT Infrastructure Standards
* Systems Engineering and Design Review (SEDR) process
* Enterprise Architecture Web page
* One-VA TRM
  1. Attach Documents

Once the SDD is approved, submit the AERB Design Compliance Decision Certificate as an attachment to the completed and approved SDD.

Template Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| September 2014 | 2.7 | Adds Enterprise Shared Services terms and requires AERB Compliance Certificate attachment. | Process Management |
| August 2014 | 2.6 | Signature block update authorized by AERB CR\_018934 | Process Management |
| March 2014 | 2.5 | Section 508 repairs to new version approved by AERB Chair approved | Process Management |
| August 2013 | 2.3 | Replaced the Service Architecture sub-section with new sub-sections for consumed and provided services. Also applied miscellaneous feedback from VA team. | ASD Enterprise Shared Services (ESS) Work Group |
| June 2013 | 1.3 | Upgraded to MS Office 2007-2010 format | Process Management |
| June 2013 | 1.2 | Address inconsistencies in Section 3, Conceptual Design, Correct headings | Process Management |
| March 2013 | 1.1 | Formatted to documentation standards and edited for Section 508 conformance | Process Management |
| January 2013 | 1.0 | Initial Document | PMAS Business Office |

Place latest revisions at top of table.

The Template Revision History pertains only to the format of the template. It does not apply to the content of the document or any changes or updates to the content of the document after distribution.

The Template Revision History can be removed at the discretion of the author of the document.

Remove blank rows.

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>