Computerized Patient Record System

Enhancements Phase I

System Design Document



**September 2016**

**Version 1.0**

Department of Veterans Affairs

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 9/29/2016 | 1.1 | Added changes for NSR 20100911 | PII |
| 8/24/2016 | 1.0 | Initial Version | PII |

Place latest revisions at top of table.

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

Remove blank rows.

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.

When to Complete Each Section of the SDD

| Section | Completed On or Before PMAS Phase | Rationale |
| --- | --- | --- |
| 1 – Introduction | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 2 - Background | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 3 – Conceptual Design | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 4 – System Architecture | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 5 – Data Design | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 6 – Detailed Design | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 7 – External System Interface Design | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 8 – Human Machine Interfaces | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| Attachments | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |

A product’s system design should be defined conceptually prior to the allocation of personnel and resources that occur at project initiation. This gives the enterprise an opportunity to evaluate IT investments before project teams are stood up and funding is allocated. Sections 1- 4 which discuss the high level design should be completed prior to MS 0. All sections should be completed and updated before MS 1. Projects will need to address all SDD approval constraints prior to the MS 2 review. In addition, the SDD should reflect the as-built product going into the MS 2 review.

| Activity | New Capability (1) | Feature Enhancement (2) |
| --- | --- | --- |
| **Field Deployment (A)** | Yes | Yes |
| **Cloud/Web Deployment (B)** | No | No |
| **Mobile Application (C)** | No | No |

Table of Contents

[1. Introduction 1](#_Toc420996797)

[1.1. Scope 1](#_Toc420996798)

[1.2. User Profiles 1](#_Toc420996799)

[2. Background 1](#_Toc420996800)

[2.1. Overview of the System 1](#_Toc420996801)

[2.2. Overview of the Business Process 1](#_Toc420996802)

[2.3. Overview of the Significant Requirements 1](#_Toc420996803)

[3. Conceptual Design 2](#_Toc420996804)

[3.1. Conceptual Application Design 2](#_Toc420996805)

[3.1.1. Application Context 2](#_Toc420996806)

[3.1.2. High-Level Application Design 4](#_Toc420996807)

[3.1.3. Application Locations 7](#_Toc420996808)

[3.2. Conceptual Data Design 8](#_Toc420996809)

[3.2.1. Project Conceptual Data Model 8](#_Toc420996810)

[3.2.2. Database Information 8](#_Toc420996811)

[3.2.3. User Interface Data Mapping 8](#_Toc420996812)

[3.2.3.1. Application Screen Interface 9](#_Toc420996813)

[3.2.3.1.1. *<Insert name of screen>* 9](#_Toc420996814)

[3.2.3.2. Application Report Interface 9](#_Toc420996815)

[3.2.3.2.1. *<Insert name of report>* 9](#_Toc420996816)

[3.2.3.3. Unmapped Data Element 10](#_Toc420996817)

[3.3. Conceptual Infrastructure Design 10](#_Toc420996818)

[3.3.1. System Criticality and High Availability 10](#_Toc420996819)

[3.3.2. Special Technology 11](#_Toc420996820)

[3.3.3. Technology Locations 11](#_Toc420996821)

[3.3.4. Conceptual Infrastructure Diagram 12](#_Toc420996822)

[3.3.4.1. Location of Environments and External Interfaces 12](#_Toc420996823)

[3.3.4.2. Conceptual Production String Diagram 14](#_Toc420996824)

[4. System Architecture 14](#_Toc420996825)

[4.1. Hardware Architecture 14](#_Toc420996826)

[4.2. Software Architecture 14](#_Toc420996827)

[4.3. Network Architecture 14](#_Toc420996828)

[4.4. Service Oriented Architecture / ESS 14](#_Toc420996829)

[4.5. Enterprise Architecture 15](#_Toc420996830)

[5. Data Design 16](#_Toc420996831)

[5.1. DBMS Files 16](#_Toc420996832)

[5.2. Non-DBMS Files 16](#_Toc420996833)

[5.3. Data View 17](#_Toc420996834)

[6. Detailed Design 17](#_Toc420996835)

[6.1. Hardware Detailed Design 17](#_Toc420996836)

[6.2. Software Detailed Design 17](#_Toc420996837)

[6.2.1. Conceptual Design 17](#_Toc420996838)

[6.2.1.1. Product Perspective 18](#_Toc420996839)

[6.2.1.1.1. User Interfaces 18](#_Toc420996840)

[6.2.1.1.2. Hardware Interfaces 18](#_Toc420996841)

[6.2.1.1.3. Software Interfaces 18](#_Toc420996842)

[6.2.1.1.4. Communications Interfaces 18](#_Toc420996843)

[6.2.1.1.5. Memory Constraints 19](#_Toc420996844)

[6.2.1.1.6. Special Operations 19](#_Toc420996845)

[6.2.1.2. Product Features 19](#_Toc420996846)

[6.2.1.3. User Characteristics 19](#_Toc420996847)

[6.2.1.4. Dependencies and Constraints 19](#_Toc420996848)

[6.2.2. Specific Requirements 20](#_Toc420996849)

[6.2.2.1. Database Repository 20](#_Toc420996850)

[6.2.2.2. System Features 21](#_Toc420996851)

[6.2.2.3. Design Element Tables 21](#_Toc420996852)

[6.2.2.3.1. Routines (Entry Points) 21](#_Toc420996853)

[6.2.2.3.2. Templates 23](#_Toc420996854)

[6.2.2.3.3. Bulletins 24](#_Toc420996855)

[6.2.2.3.4. Data Entries Affected by the Design 25](#_Toc420996856)

[6.2.2.3.5. Unique Record(s) 25](#_Toc420996857)

[6.2.2.3.6. File or Global Size Changes 25](#_Toc420996858)

[6.2.2.3.7. Mail Groups 26](#_Toc420996859)

[6.2.2.3.8. Security Keys 27](#_Toc420996860)

[6.2.2.3.9. Options 29](#_Toc420996861)

[6.2.2.3.10. Protocols 30](#_Toc420996862)

[6.2.2.3.11. Remote Procedure Call (RPC) 32](#_Toc420996863)

[6.2.2.3.12. Constants Defined in Interface 32](#_Toc420996864)

[6.2.2.3.13. Variables Defined in Interface 33](#_Toc420996865)

[6.2.2.3.14. Types Defined in Interface 33](#_Toc420996866)

[6.2.2.3.15. GUI 33](#_Toc420996867)

[6.2.2.3.16. GUI Classes 33](#_Toc420996868)

[6.2.2.3.17. Current Form 34](#_Toc420996869)

[6.2.2.3.18. Modified Form 34](#_Toc420996870)

[6.2.2.3.19. Components on Form 34](#_Toc420996871)

[6.2.2.3.20. Events 34](#_Toc420996872)

[6.2.2.3.21. Methods 34](#_Toc420996873)

[6.2.2.3.22. Special References 34](#_Toc420996874)

[6.2.2.3.23. Class Events 34](#_Toc420996875)

[6.2.2.3.24. Class Methods 34](#_Toc420996876)

[6.2.2.3.25. Class Properties 35](#_Toc420996877)

[6.2.2.3.26. Uses Clause 35](#_Toc420996878)

[6.2.2.3.27. Forms 35](#_Toc420996879)

[6.2.2.3.28. Functions 36](#_Toc420996880)

[6.2.2.3.29. Dialog 37](#_Toc420996881)

[6.2.2.3.30. Help Frame 38](#_Toc420996882)

[6.2.2.3.31. HL7 Application Parameter 39](#_Toc420996883)

[6.2.2.3.32. HL7 Logical Link 40](#_Toc420996884)

[6.2.2.3.33. COTS Interface 40](#_Toc420996885)

[6.3. Network Detailed Design 41](#_Toc420996886)

[6.4. Security and Privacy 41](#_Toc420996887)

[6.4.1. Security 41](#_Toc420996888)

[6.4.2. Privacy 42](#_Toc420996889)

[6.5. Service Oriented Architecture / ESS Detailed Design 43](#_Toc420996890)

[6.5.1. Service Description for <Consumed Service Name> 43](#_Toc420996891)

[6.5.2. Service Design for <Provided Service Name> 43](#_Toc420996892)

[6.5.2.1. Introduction 43](#_Toc420996893)

[6.5.2.1.1. Purpose and Scope of Service 43](#_Toc420996894)

[6.5.2.1.2. Links to Other Documents 43](#_Toc420996895)

[6.5.2.2. Service Details 44](#_Toc420996896)

[6.5.2.2.1. Service Identification 44](#_Toc420996897)

[6.5.2.2.2. Service Versions 45](#_Toc420996898)

[6.5.2.2.3. Summary of Design and Platform Details 45](#_Toc420996899)

[6.5.2.2.3.1. SOA Pattern(s) Implemented 45](#_Toc420996900)

[6.5.2.2.3.2. COTS Platform vendor names and versions for hosting platform 45](#_Toc420996901)

[6.5.2.3. Dependencies 45](#_Toc420996902)

[6.5.2.4. Service Design Details 45](#_Toc420996903)

[6.5.2.4.1. Interface Technical Specs 45](#_Toc420996904)

[6.5.2.4.1.1. Service Invocation Type 46](#_Toc420996905)

[6.5.2.4.1.2. Service Interface Type 46](#_Toc420996906)

[6.5.2.4.1.3. Service Name 46](#_Toc420996907)

[6.5.2.4.1.4. Interface 46](#_Toc420996908)

[6.5.2.4.1.5. End Points 46](#_Toc420996909)

[6.5.2.4.1.6. Operations or Methods 46](#_Toc420996910)

[6.5.2.4.1.7. Message Schemas 46](#_Toc420996911)

[6.5.2.4.2. Information Model 47](#_Toc420996912)

[6.5.2.4.2.1. Class Diagram and Description of Entities Involved 47](#_Toc420996913)

[6.5.2.4.2.2. Mappings from ELDM to Standards Based Schemas 47](#_Toc420996914)

[6.5.2.4.3. Behavior Model (AKA Use Case Realization) 47](#_Toc420996915)

[6.5.2.4.3.1. Use Cases (Use Case Model) 47](#_Toc420996916)

[6.5.2.4.3.2. Interaction Diagrams 47](#_Toc420996917)

[6.5.2.5. Gap Analysis 47](#_Toc420996918)

[6.5.2.5.1. Variances from Enterprise Target Architecture 48](#_Toc420996919)

[6.5.2.5.2. Variances from SLDs 48](#_Toc420996920)

[6.5.2.5.3. Variances from Standards and Policies 48](#_Toc420996921)

[6.5.2.5.4. Justification for Exceptions and Mitigation 48](#_Toc420996922)

[7. External System Interface Design 48](#_Toc420996923)

[7.1. Interface Architecture 48](#_Toc420996924)

[7.2. Interface Detailed Design 49](#_Toc420996925)

[8. Human-Machine Interface 50](#_Toc420996926)

[8.1. Interface Design Rules 50](#_Toc420996927)

[8.2. Inputs 50](#_Toc420996928)

[8.3. Outputs 50](#_Toc420996929)

[8.4. Navigation Hierarchy 51](#_Toc420996930)

[8.4.1. Screen [x.1] 51](#_Toc420996931)

[8.4.2. Screen [x.2] 51](#_Toc420996932)

[8.4.3. Screen [x.3] 51](#_Toc420996933)

[9. Attachment A – Approval Signatures 51](#_Toc420996934)

[A. Additional Information 52](#_Toc420996935)

[A.1. Identification of Technology and Standards 52](#_Toc420996936)

[A.2. Constraining Policies, Directives and Procedures 52](#_Toc420996937)

[A.3. Requirements Traceability Matrix 52](#_Toc420996938)

[A.4. Packaging and Installation 52](#_Toc420996939)

[A.5. Design Metrics 52](#_Toc420996940)

# Introduction

Identify and provide a brief introduction of the system that is the subject of this SDD.

## Scope

The CPRS EP1 BRD can be found at:

CPRS EP1 Business Requirements Document

## User Profiles

There are several types of health professionals who use Computerized Patient Record System (CPRS). Primarily, the intended users are clinicians such as physicians, nurses and pharmacists who are dealing with patient care. In addition, there are laboratorians, financial staff and others who use CPRS to find patient-related information. These users are trained on how to use and customize it to fit their needs per their required needs. There are a large number of documents and presentations on many of the functions of CPRS, including online training. To accommodate the varying proficiency levels vary among those using CPRS, support is provided via multiple channels, such as context-sensitive help, various support teams, web sites and training on specific features of the software.

The Information Technology (IT) staff who support CPRS are typically supporting multiple systems and are familiar with support and maintenance of Veterans Health Information Systems and Technology Architecture (VistA) systems. Documentation is provided with each release of CPRS, covering proper installation techniques, and support is provided by the development team during deployment. In addition, there are staff available to assist the IT representatives should problems arise.

# Background

## Overview of the System

CPRS provides an integrated patient record system for clinicians, managers, Quality Assurance (QA) staff, and researchers. The primary goal of CPRS is to provide a fast and easy-to-use application that makes available to providers the information needed in the clinical workflow process. The CPRS user interface is integrated with VistA to facilitate reviewing, documenting and preserving of coordinated care information and improved accessibility of online clinical information and results.

## Overview of the Business Process

CPRS EP1 supports the following business processes:

* Medication order entry in both CPRS and Inpatient Medications
* Review of patient’s electronic charts
* Improved patient safety related to patient selection and display areas

The Requirements Specification Document (RSD) is contained in Rational Dynamic Object Oriented Requirements System (DOORS) Next Generation (NG) and can be obtained from the following link:

DOORS NG - EP1 - Requirements

## Overview of the Significant Requirements

The CPRS EP1 Business Requirements Document (BRD) can be found at the following link:

CPRS EP1 Business Requirements Document

The RSD is contained in Rational DOORS NGand can be obtained from the following link:

DOORS NG - EP1 - Requirements

1. GENDER/SEX FIELD IN VETERANS HEALTH INFORMATION SYSTEMS AND TECHNOLOGY ARCHITECTURE (VistA)/COMPUTER PATIENT RECORD SYSTEM (CPRS) - REQUEST #20130305

2. ALLOW TRANSFER BETWEEN INPATIENT, OUTPATIENT, AND Non-VA MEDS- REQUEST #20080617

3. CLINIC ORDERS – INPATIENT MEDICATIONS EDITS AND BACKDOOR ENTRY- REQUEST #20120502

4. ADDITION OF A FUNCTION TO ENTER STOP AND START DATES FOR MEDICATIONS IN CPRS- REQUEST #20090602

5. Medication Reconciliation Worksheet Non-VA MEDICATION DOSE ENHANCEMENT – REQUEST #20100911

6. CPRS OPTION TO PROHIBIT MEDICATION RENEWAL- REQUEST 20110809

7. Non-VA MEDICATION VIEW INCONSISTENCY- REQUEST #20110315

8. PHARMACY VISTA ORDERS TO DISPLAY DATES IN MM/DD/YYYY FORMAT, AND ADD A HARD STOP FOR DATE ENTRY- REQUEST #20111103

9. ADDRESS NOTIFICATION SOFTWARE PROBLEM SEEN WHEN AN ORDERING PROVIDER CHANGES ROLE- REQUEST #20130504

10. PATIENT SELECTION – SIMILAR PATIENTS BOX –REQUEST #20131005

11. CPRS: ALLERGY VIEW/PREVENT ORDERING MEDS THAT CAUSE ANAPHYLAXIS- REQUEST #20131101

# 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

Provide details of the ‘As-is” view of the existing system along with the design of “the current increment” and the “To-be.”. The “To-Be” view should include the future application context, and application high level design. The “current increment” view should have application context and high level design of this specific increment that this SDD addresses.

### Application Context

.

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

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



Sample Application Context Diagram

Sample Application Context Diagram

Figure : Sample Application Context Diagram

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

Table 5 (Grouping): Application Context Description

Object

| ID | Name | Description | Interface Name | Interface System |
| --- | --- | --- | --- | --- |
| < ID from diagram> | <Enter name of external system, organization, or agency> | <High level discussion of the purpose of the information interchange> | <Name of each of the Interfaces to this object> | <Systems with which this system interfaces> |

Interfaces External to OI&T

| 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 OI&T

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

Externally Shared Data Stores

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

### High-Level Application Design

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

**Error! Reference source not found.** 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

Table 6 to describe the objects in **Error! Reference source not found.**.

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

Sample High-Level Application Design

Figure : Sample High-Level Application Design

Table 6: Objects in the High Level Application Design

Objects / Components to be Built or Modified

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

Internal Data Stores

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

### 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> |

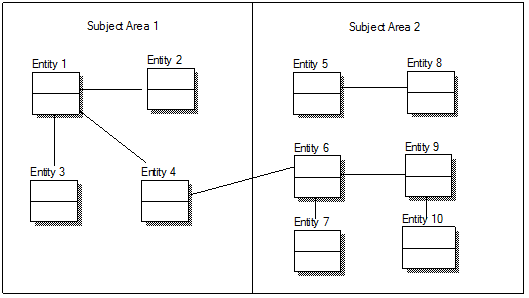
## Conceptual Data Design

### 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 3 illustrates a sample of a project CDM.

Figure : Sample Project Conceptual Data Mode

****

### Database Information

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

Table 9: Database Inventory

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

### User Interface Data Mapping

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

#### Application Screen Interface

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

##### *<Insert name of screen>*

Figure 4: <screen name> Screen represents the screen that <describes what the screen accomplishes>; Table 10 describes it. Paste a screenshot below and complete the table to describe the screen.

Figure : *<screen name>* Screen

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

Figure : *< Report name>* Report

Table 11: *<Report name>* Description

| Report Column | Data Source *<Table Name. 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.

. 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>  n/a. no special technology for CPRS EP1 | <Business language description> | <At what type of location will this technology be deployed?> | <Is this technology in the Technical Reference Model (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 6, 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).



Sample Conceptual Networks and Environments

Sample Conceptual Networks and Environments

Figure : Sample Conceptual Networks and Environments

#### Conceptual Production String Diagram

Create a diagram to show the configuration of a single production string.

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

Figure : 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.

## Hardware Architecture

CPRS is a legacy application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. As such, it uses the existing VistA hardware architecture.

Some VistA instances are local to the VA Medical Centers (VAMC) that use it. Others are installed in Regional Data Processing Centers (RDPC). Most of these instances have hot backup sites located in another, geographically separated, location.

The primary architecture at this time is a cluster of Linux servers that act as the application server with the client workstations connecting to this cluster. This Linux cluster is connected to a cluster of VMS servers that act as the database server, where the VistA database resides.

The client workstations may be local to the VAMC or they may be remote at Community Based Outpatient Clinics (CBOC), other VAMCs (in the case of integrated sites) or may even be one-off remote workstations that connect via Citrix Access Group (CAG) or Virtual Private Network (VPN).

While personnel outside of Product Development are responsible for determining the best configurations and ensuring adequate hardware and network connectivity, the CPRS EP1 project considers additional space and potential performance impacts. The ultimate goal is to add no more than 5% to disk space or Central Processing Unit (CPU) requirements.

During the field testing phase, any additional files created are monitored to ensure this is not exceeded. Local and regional IT staff monitor the systems and notify CPRS development if the resource demands exceeds expectations based on the development environment.



## Software Architecture

CPRS is a legacy application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. CPRS EP1 will not change that underlying software architecture.

CPRS EP1 will use Delphi and Massachusetts general hospital Utility Multi-Programming System (MUMPS) as the primary programming languages.

CPRS EP1 is currently using Rational Change and Configuration Management (CCM) as the source code control system.

CPRS EP1 requires no additional software for support, outside the existing VistA infrastructure.

## Network Architecture

CPRS is a legacy application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. The executable portion of CPRS EP1 will continue communicate using the existing network architecture that supports the legacy VistA systems.

CPRS EP1 uses remote procedure calls (RPCs) over the local, or wide, area network to communicate between the client and the VistA instance. This communication uses Kernel’s broker package.

Reference the hardware architecture for a high-level overview of the communication pathways.

## Service Oriented Architecture / ESS

CPRS is a legacy application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. CPRS EP1 will enhance the existing CPRS system.

Refer to the VistA Monograph for a full explanation of the larger system of which CPRS EP1 is a part.

Note: The CPRS architecture does not supply new services nor does it consume existing services.

## Enterprise Architecture

CPRS EP1 is using Delphi XE8 for the Delphi development.

The server-side code is written using Cache/MUMPS, which is approved under the TRM.

# Data Design

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

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

## DBMS Files

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

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

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

## Non-DBMS Files

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

## Data View

A "Data View" should be included in the Architectural Representation whenever persistent data objects are included in the system (they are typically present in most software systems). The data view describes the logical data model of the system and includes an Entity Relationship Diagram (ERD). For a description of Entity Relationship diagramming please refer to the whitepaper <http://URL

# 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.

Note: Every design item should map back to the Requirements Specification Document. These should be captured in the Requirement Traceability Matrix (RTM).

## Hardware Detailed Design

N/A. CPRS EP1 will use the existing hardware infrastructure and will be designed to require no significant increase in data storage capacity nor CPU resources.

## 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.

### 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 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

CPRS EP1 will enhance the existing CPRS system. No modifications to the existing hardware interfaces are planned.

Currently, CPRS, using the Kernel system, will support any device that Kernel supports. CPRS EP1 will not control this support.

##### Software Interfaces

CPRS communicates with all the VistA clinical applications and several of the financial applications. In addition, CPRS uses FileMan and Kernel. CPRS is written using the current nationally released versions of each of these packages and plans to continue to use and support the currently released versions.

##### Communication Interfaces

No modifications will be made to the existing communication interfaces, which are under the control of Engineering.

##### Memory Constraints

N/A

##### Special Operations

N/A

#### Product Features

#### User Characteristics

CPRS is used primarily by clinicians such as physicians, nurses, physician assistants, nurse practitioners, pharmacists and other ancillary clinical users.

#### Dependencies and Constraints

CPRS EP1 must utilize the existing hardware and network infrastructure. Therefore, the increase in network, memory, CPU and data storage should not be significant. Some remote installations, such as CBOCs are particularly susceptible.

CPRS EP1 must be 508 user accessibility compliant.

The design must take into account that CPRS is a mission critical application.

### Specific Requirements

#### Database Repository

Data for CPRS EP1 will be stored in the existing VistA database.

#### System Features

Please refer to the RSD:

The RSD is contained in DOORS NG and can be obtained from the following link:

DOORS NG - EP1 - 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. |

NSR: 20100911 – Medication Reconciliation Worksheet Non-VA Medication

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSPST2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 555637, 386545, 665275 | | | |
| **Related Options** | Medication worksheet in CPRS | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Only called from CPRS MWS Health Summary option | Standard FileMan DBS calls  GMTSUP  ORCSAVE2  ORQ12  ORWPS  ORX8  PSNAPIS  PSO52API  PSO59  PSS50  VADPT  VASITE  XLFDT |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | None | | | | |
| **Related Protocols** | None | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | |
| **Data Passing** | Input | Output Reference | Both | Global Reference | Local |
| **Input Attribute Name and Definition** | Name: DFN  Definition: Internal Entry number of the patient in file #2 | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | |

| Current Logic |
| --- |
|  |

| Modified Logic (Changes are in bold) | | | | |
| --- | --- | --- | --- | --- |
| ---------------------------------------  GMTSPST2.INT.1  **+108 ;I $P(LISTNODE,"^",1)["N;" D**  **+109 ;. S:$P(LISTNODE,"^",4)="ACTIVE" NVA($P(LISTNODE,"^",2),+LISTNODE)=**  **LISTNODE**  **+110 ;. S RETURN=1**  ...................  |"CPRS31"|GMTSPST2.INT.1  +108 I $P(LISTNODE,"^",1)["N;" D  +109 . S:$P(LISTNODE,"^",4)="ACTIVE" NVA($P(LISTNODE,"^",2),+LISTNODE)=L  ISTNODE    +110 . S RETURN=1  ---------------------------------------  GMTSPST2.INT.1  **+122 . I $P(RXS(RXSIEN),";")["P"!($P(RXS(RXSIEN),";")["N") D GETPEND(RXS**  **IEN) S PSOQPEND=1 Q ;->**  ...................  |"CPRS31"|GMTSPST2.INT.1  +122 . I $P(RXS(RXSIEN),";")["P" D GETPEND(RXSIEN) S PSOQPEND=1 Q ;->  ---------------------------------------  GMTSPST2.INT.1  **+157 N PSOQPDN,PSOQDIND,PSOQ100,PSOQSCT,GMTSPST2,A**  ...................  |"CPRS31"|GMTSPST2.INT.1  +157 N PSOQPDN,PSOQDIND,PSOQ100,PSOQSCT,GMTSPST2  GMTSPST2.INT.1  **+165 S A=$P(RXS(RXSIEN),";")**  **+166 I A["P" F PSOQSCT=2:1:$O(GMTSPST2(":"),-1) S RXS("D","\*\*PENDING\*\* "**  **\_PSOQPDN,PSOQSCT)=GMTSPST2(PSOQSCT)**  **+167 I A["N" F PSOQSCT=2:1:$O(GMTSPST2(":"),-1) S RXS("D","\*\*NON-VA\*\* "\_**  **PSOQPDN,PSOQSCT)=GMTSPST2(PSOQSCT)**    ...................  |"CPRS31"|GMTSPST2.INT.1  +165 F PSOQSCT=2:1:$O(GMTSPST2(":"),-1) S RXS("D","\*\*PENDING\*\* "\_PSOQPDN  ,PSOQSCT)=GMTSPST2(PSOQSCT) | | | | |
| 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** |  |
| **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** | Select the appropriate category: 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.

## Security and Privacy

### Security

This project will continue to use the existing VistA / CPRS security model.

### Privacy

This project will continue to use the existing VistA / CPRS privacy model.

## Service Oriented Architecture / ESS Detailed Design

CPRS is a legacy GUI application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. CPRS EP1 is enhancing the existing CPRS system.

Refer to the VistA Monograph for a full explanation of the larger system that CPRS EP1 is a part of.

Of note: CPRS’s architecture does not supply new enterprise services or consume enterprise services

### 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)

Describe how this service fits 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

CPRS follows VA standards, Standards and Conventions (SAC) guidelines, as well as guidelines from Human Factors/Patient Safety.

## Inputs

Mouse and keyboard—no special or unusual input device is required.

## Outputs

Please refer to section 6.2 for details regarding output changes for each of the requests included in CPRS EP1.

## 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.

# 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 Business Sponsor and Project Manager are required to sign.

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

Signed: Date:

< Business Sponsor >

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

Signed: Date:

< Project Manager >

1. Additional Information

Attach any addition information that supplements the design specification.

* 1. Identification of Technology and Standards

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.).

* 1. Constraining Policies, Directives and Procedures

Identify any constraints or requirements placed on this document by policies, directives, or procedures.

* 1. Requirements Traceability Matrix

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.

Template Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| June 2015 | 2.10 | Changed Heading 1 default setting to eliminate page break before | Process Management |
| May 2015 | 2.9 | Edited for Section 508 conformance and remediated with Common Look Office tool | Process Management |
| February 2015 | 2.8 | Incorporates revisions from PMAS Reform Lockdown; namely removing requirements for information that can be obtained from other PMAS authoritative sources. | Andrew Slawter, Office of Technology Strategies |
| 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>