

System Design Description Instructions

Version 1.2 • 14 JAN 2008



Version History

This and other Framework Extension tools are available on the Framework Web site.

Release Date	Description
14-Jan-2008	Version 1.2 released. Modified "Using this Template" section of the Template and italicized all section instructions to align with the Framework and Change Request (CR) #34. CR #34 was recommended by the Framework Change Advisory Board (CAB) and approved by DIR.
13-Mar-2007	Version 1.1 released. Made minor modifications to indicate Framework Extension.
30-Jun-2006	Version 1.0 Instructions and Template Released.

Contents

Introduction	1
Use of the System Design Description.....	1
Section 1. Introduction	2
1.1 Purpose.....	2
1.2 Scope.....	2
Section 2. System Architecture.....	2
2.1 Architectural Design Approach	2
2.2 Architecture Design.....	3
Section 3. Data Dictionary.....	4
Section 4. System Domain Design	5
4.1 System Domain Chart.....	6
4.2 System Domains.....	7
Section 5. Data Design	9
5.1 Persistent/Static Data	9
5.2 Transient/Dynamic Data	10
5.3 External Interface Data	10
5.4 Transformation of Data	10
Section 6. User Interface Design	11
6.1 User Interface Design Overview	11
6.2 User Interface Navigation Hierarchy.....	11
6.3 User Function Categories (or Use Cases).....	12
Section 7. Other Interfaces	14
Section 8. Other Design Features.....	15
Section 9. Requirements Traceability Matrix	15
Section 10. References.....	15
Section 11. Glossary.....	16
Section 12. Revision History	16
Section 13. Appendices	16
Appendix A. Sample Data Dictionary Table	17
Appendix B. Sample Screen/Other User Interface Fields Table.....	18

Introduction

The System Design Description (SyDD) Template and Template Instructions are included within the System Development Life Cycle (SDLS) Extension of the Texas Project Delivery Framework (Framework) to establish a consistent method for documenting a system design. A system design is the design fulfillment of the requirements stated in the System Requirements Specification (SyRS) and the basis for the implementation of the system to be built. The SyDD documents the architecture and design of the system. The purpose of the SyDD is to communicate in sufficient detail how the system is to be constructed.

Providing documentation of the system design can reduce project risk by reducing uncertainty in the implementation of the system. Documentation of the system design contributes to the success of information technology systems by establishing and communicating how the properties of the system requirements will be transitioned into a design. Expectations for all aspects of the system's features and performance can be contrasted with the design in order to identify and resolve potential design flaws. Identification and resolution of design flaws and problems positively impact the quality and customer satisfaction of the implemented system. In addition, flaws and problems corrected early in systems development have less of an impact on a project's schedule and budget.

Use of the System Design Description

Within the Framework, System Development Life Cycle (SDLC) tools are included as an extensible Framework toolset. Use of this toolset is intended to be tailored or customized to meet project requirements and minimize project risk. Project requirements may be met and risk minimized by producing a System Design Description (SyDD) or a Software Design Description (SDD) as the sole design description, or by producing a SyDD in conjunction with a SDD. If the SyDD is produced in conjunction with a SDD, it may be appropriate to state "not applicable" in a document section that will be addressed in the SDD. If "not applicable" is used, a justification must be included in the document section. The justification must convey the basis for why the document section content does not apply to the project at all or at that point during project delivery.

The SyDD is completed, reviewed, and approved in the Project Planning Review Gate. The SyDD documents and communicates sufficient details regarding the architecture and design of the system to enable the technical community to produce specifications and construct the system. Technical resources that may not be familiar with the project will use the SyDD to build the system or components of the system.

The format of the SyDD Template serves as a basis for creating an actual project document. Customize the SyDD Template, as directed within the SyDD Template Instructions, to contain the sections necessary to comprehensively document the system design.

The SyDD should be developed in coordination with and be accessible by appropriate project team and stakeholder entities. In addition, all information in the SyDD should be consistent with the Project Plan and the related project documents. All documented system requirements, including interfaces, should be addressed by the design.

Approval of the SyDD constitutes agreement that the system design documented within satisfies the approved and baselined system requirements. Once approved, changes can be made to the design in the SyDD only through the change management process.

Note: Examples included in the System Design Description Template Instructions have no design relationship to each other and are intended for illustration purposes only.

The System Design Description should contain descriptive labels for and references to every figure, table, and diagram included within the document.

Section 1. Introduction

Provide high-level introductory information about the System Design Description (SyDD) in the following subsections. This section should stand alone as an executive summary.

1.1 Purpose

Describe the purpose of the SyDD and its intended audience.

1.2 Scope

Describe the scope of the system to be produced. Within the description:

- Provide an overview of the architecture and design of the system
- Identify the system components and include a short description of each
- Explain what each component will and will not do. Provide an overview of each component and describe the relevant benefits, objectives, and goals.

This description should be consistent with similar statements in preceding project documents.

Section 2. System Architecture

2.1 Architectural Design Approach

Describe the architectural design approach. Within the approach include:

- The methodology and techniques used in architecting the system

- The rationale for the methodology and techniques used, such as alignment with principles, practices, and standards
- Decisions about the system's behavioral design and other decisions affecting the selection and design of system components

2.2 Architecture Design

Provide and describe a figure that depicts the overall system architecture, including the system component(s) in software, hardware, networks, and any other pertinent major system components (e.g., databases, operating systems) that support the complete system. This depiction will typically require a diagram showing the major hardware components (drawn as titled boxes) and the software that resides on them (as text within the boxes), the major databases (drawn as named cylinders), and any interfaces between these components (drawn as named lines with arrowheads to depict the direction of the interface). An example of an architecture diagram is provided below.

Information in this section must be consistent with existing system architecture documentation for the project.

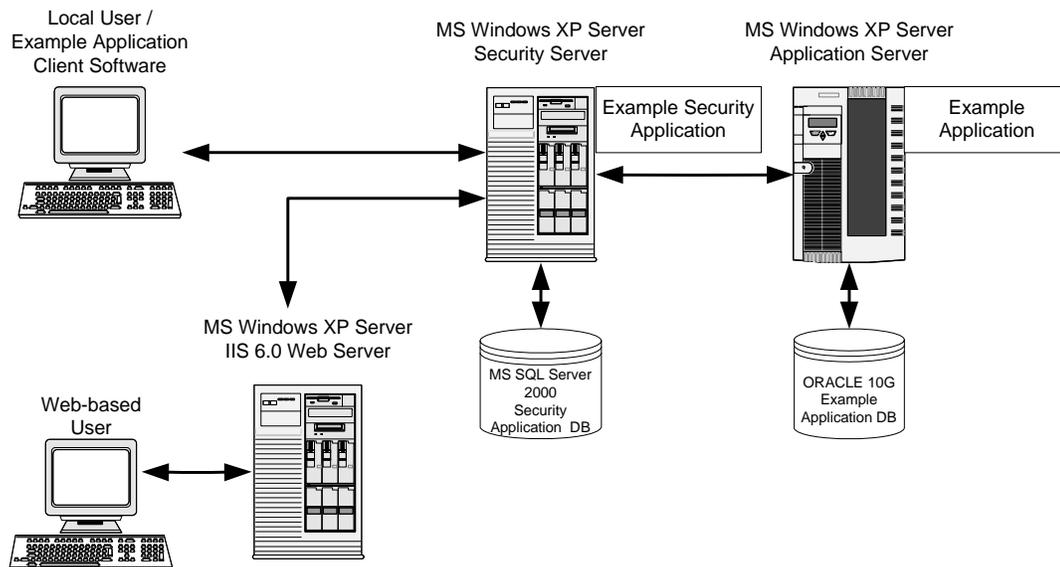


Figure 1. Example of an Architecture Diagram

Section 3. Data Dictionary

Provide a reference to the location of or provide the actual Data Dictionary Table that contains a description of each element in the system. The table should include the entity name and the following details about the data element:

- name
- definition
- data type (e.g., text, character, integer)
- storage format
- scale
- bounds
- display format
- mandatory entry or fill information (e.g., element is required, every character of the element is required)
- default value
- list of functions or other architectural features that can create and modify its values
- list of functions or other architectural features that read its values
- constraints on the data (e.g., some data is protected by Family Educational and Rights and Privacy Act (FERPA))

A sample Data Dictionary Table is provided as an additional tool in the Appendix.

An example of a Data Dictionary Table entry for a customer address table is provided below.

Note: Maintaining the Data Dictionary Table as a separate document and performing appropriate updates in a controlled fashion—rather than including it within the SyDD and requiring that the SyDD be revised each time the Data Dictionary Table is modified—is more efficient.

Entity Name	Element Name	Definition	Type	Storage Format	Scale	Bounds	Display Format	Mandatory Entry/Fill	Default Value	Modified by	Read by	Constraints
Customer Address	Address	Customer Address	Text	Any	n/a	n/a		required	n/a	Purchase, Lease	Purchase, Lease	n/a
Customer Address	City	Customer City	Char	Any	n/a	n/a	Any	required	n/a	Purchase, Lease	Purchase, Lease	n/a
Customer Address	State	Customer State	Char2	Caps Alpha	AZ-WY	n/a	Caps Alpha	required	n/a	Purchase, Lease	Purchase, Lease	Must be valid US State abbreviation
Customer Address	Zip	Customer Zip	Int	5 digit		00000-99999	nnnnn	5 digits required	n/a	Purchase, Lease, Report	Purchase, Lease, Report	n/a

Table 1. Example of a Data Dictionary Table Entry

Section 4. System Domain Design

Document the system domain design in the following subsections. The system design is represented as a set of domains or views. A domain or view is a representation or description of the entire system for a single technical category, grouping, or perspective. The domains or views form the building blocks of the technology blueprint of the system. These technical categories provide perspective and structure in the process of representing the design. Although the domains, when combined, form a representation of the whole system, they are largely independent of one another. Domains can include, but are not limited to Business Function Operations, Reporting, Operating System, Network, Hardware, User, Service, Application, Execution, Deployment, and System Interfaces.

4.1 System Domain Chart

Provide a figure depicting the set of system domains showing major components and their relationships. A domain may contain more than one component. An example of a domain chart is provided below.

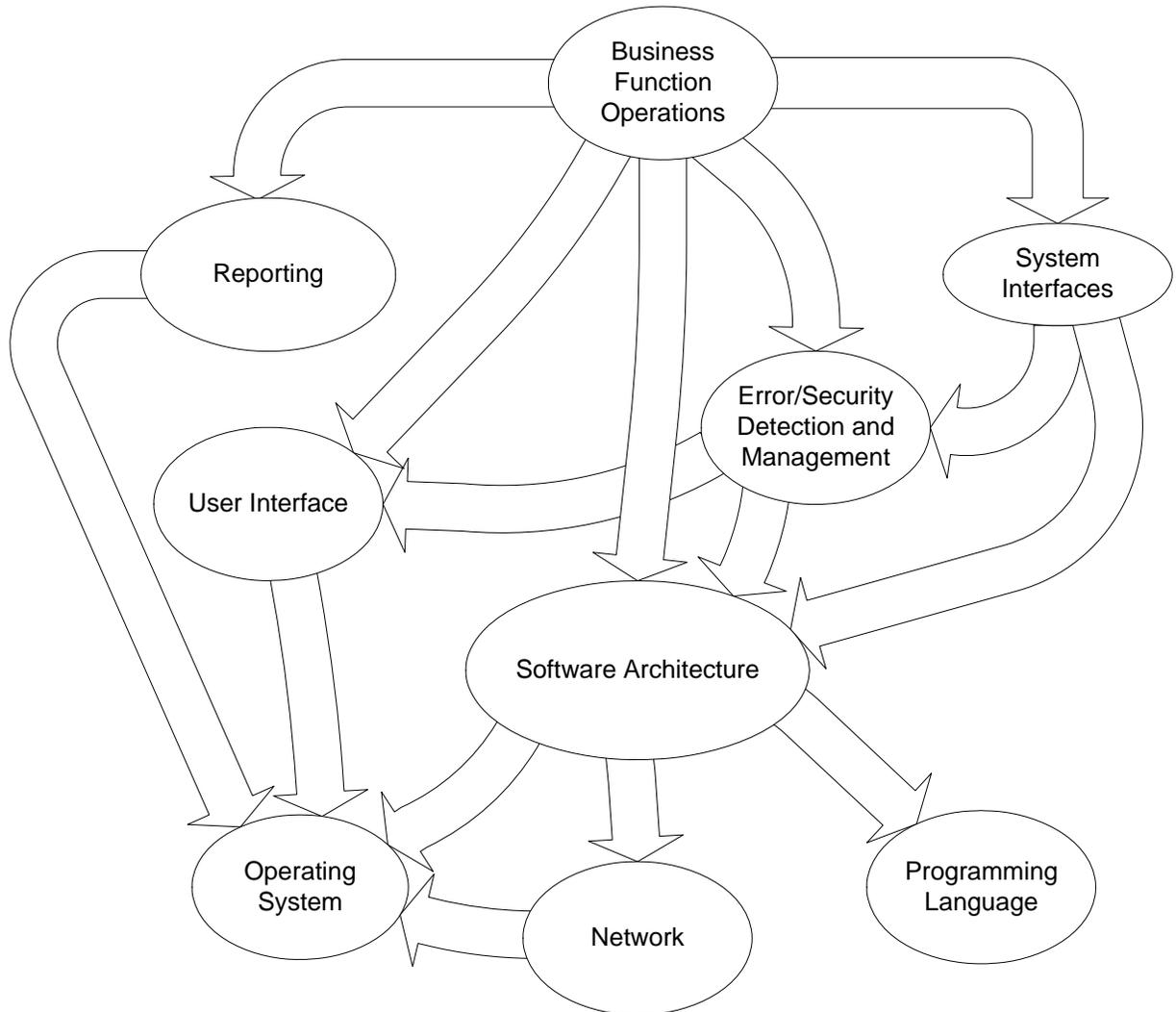


Figure 2. Example of a Domain Chart

4.2 System Domains

Customize this section to contain the subsections necessary to comprehensively document the domains, components, functions, and tasks of the system design. Each subsection should be labeled appropriately and titled for a specific domain, component, or task. The logical structure of the hierarchy is:

Domain X, where X is a specific domain name

Component Y, where Y is a specific component name

Task Z, where Z is a specific task name

Describe each domain within the design. Depict and describe the hierarchy of domains, components, functions, and tasks. These domains may include hardware, application, user, service, and other domains. Any number of domains, components, and tasks may exist.

Subsection templates for documenting Domain X are provided below.

4.2.X Domain X

Provide a domain hierarchy chart and a high-level description of Domain X and the family of components that make up Domain X. In the domain hierarchy chart depict the hierarchy of the component relationships within Domain X. An example of a hierarchy chart for a domain is provided below.

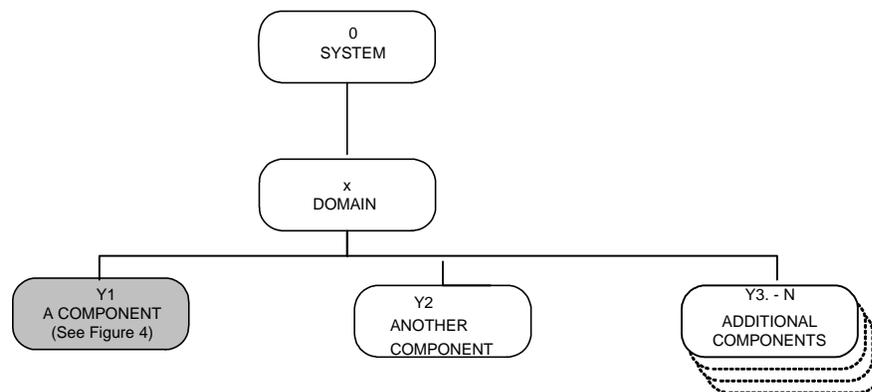


Figure 3. Example of a Hierarchy Chart of the Components within Domain X

4.2.x.y Component Y1 of Domain X

Provide a hierarchical depiction and high-level description of Component Y1 of Domain X. Within the description, include a functional decomposition of the component into its lower-level functions, following principles of top-down design. An example of a hierarchy chart for a component is provided below.

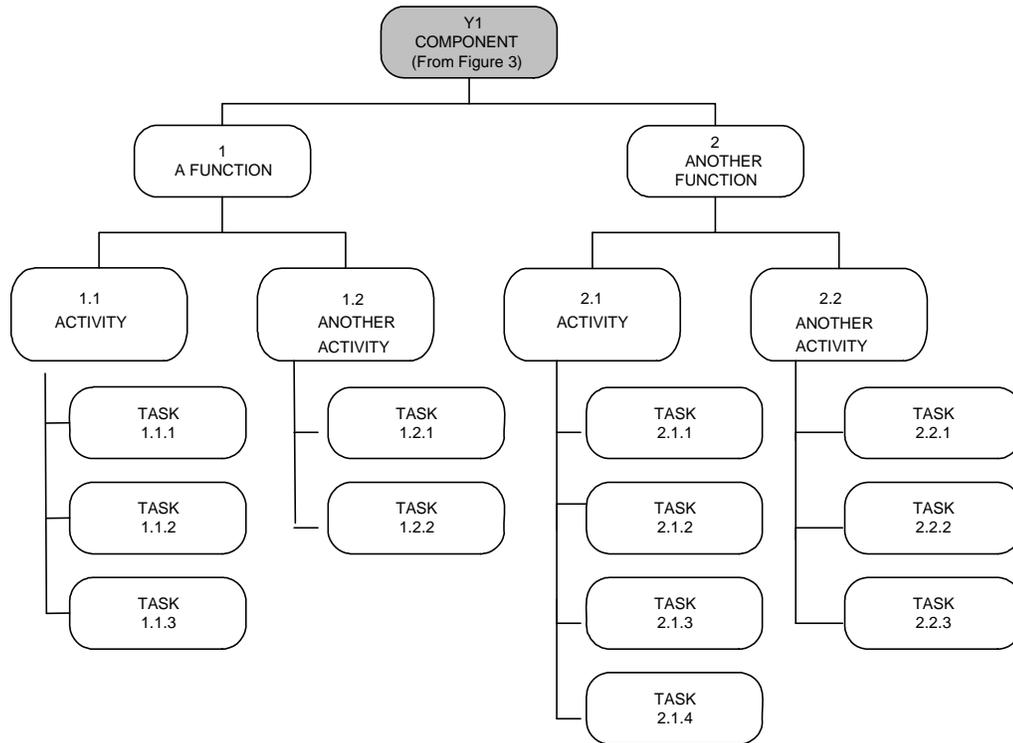


Figure 4. Example of a Hierarchy Chart for a Set of Functions for Component Y1

4.2.x.y.z Task Z of Component Y1 of Domain X

Provide a high-level description of Task Z of Component Y1 of Domain X. If appropriate for the design, provide diagrams and narratives that specify the flow of data and control at a component- or task-level. An example of a data/control flow diagram for a component is provided below.

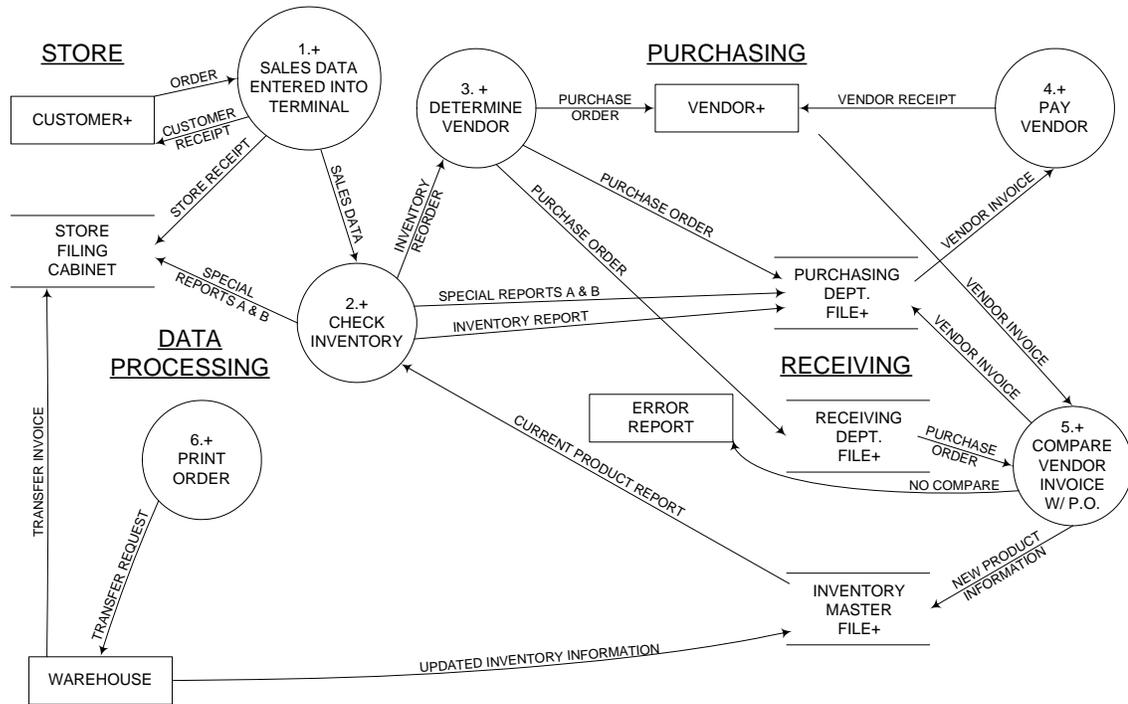


Figure 5. Example of a Data/Control Flow Diagram

Section 5. Data Design

Customize the following subsections to describe the data contained in databases and other data structures shared between design elements of the system design, include persistent/static data, transient/dynamic data, external interface data, and transformation of data. Label and title each subsection appropriately.

5.1 Persistent/Static Data

Persistent/static data is data that is stored by a system at the end of execution, and retrieved later for additional processing.

5.1.X Persistent/Static Data Store X

Describe and provide an illustration of the logical data model or entity relationship diagram(s) for the Persistent/Static Data Store X. Include the purpose and general configuration of the data store. An example of a database logical mode is included below.

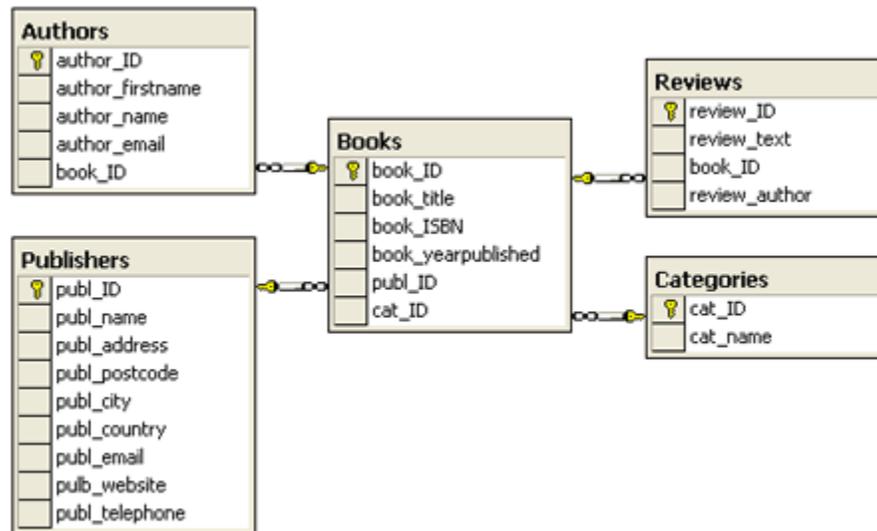


Figure 6. Example of a Persistent/Static Data Store Logical Model

5.2 Transient/Dynamic Data

Transient/dynamic data is data used by the system that does not persist after execution is completed. Provide a description of the system's transient/dynamic data design and its general configuration. Include the purpose for each of the transient/dynamic data design elements.

5.3 External Interface Data

Describe and, if appropriate, provide diagrams of the external interfaces' data design. Include the purpose and general configuration of the data design elements.

5.4 Transformation of Data

Systems often require the transformation of data formats or structures. This is especially true when systems exchange information with other systems. If the system performs explicit data transforms or contains distinct data transform components, describe the system's data transformation design. Include the general configuration and purpose for each of the data transform design elements, and the transformation mapping rules.

Section 6. User Interface Design

6.1 User Interface Design Overview

Provide a high-level description of the user interface for this system. Describe any systems requirements (e.g., performance and usability) associated with all of the user interfaces.

6.2 User Interface Navigation Hierarchy

Provide and describe a diagram of the navigation hierarchy that illustrates how a user moves through the user interface. An example of a navigation hierarchy diagram that illustrates how a user moves through the pages of a user interface is provided below.

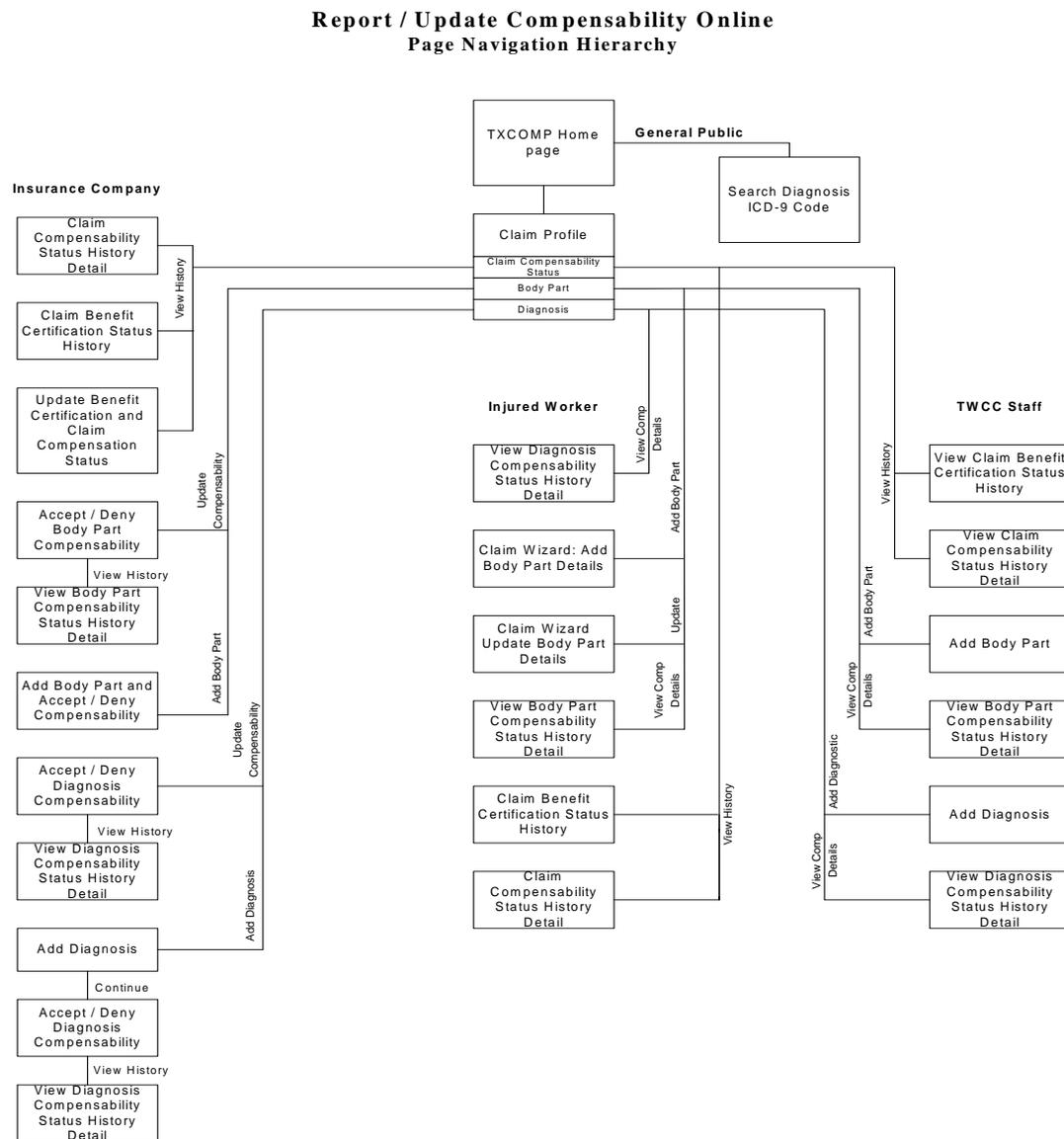


Figure 7. Example of a User Interface Navigation Hierarchy Diagram

6.3 User Function Categories (or Use Cases)

Customize the following subsections to accurately and comprehensively document each category of user function (e.g., transactions, reports, administration) or use case that requires an interface. Typically, a major category of user function correlates to a particular use case in the SRS, and each use case described in the SRS will require one or more screens for those use cases depicting a user.

Document each category of user function or use case individually in a corresponding subsection. Label each subsection appropriately and title each subsection descriptively to indicate the function or use case being documented.

6.3.X Function (or Use Case) X

Provide a description of the function supporting this category of user interfaces. Where applicable, this description may be derived from its source use case.

6.3.x.y Function (or Use Case) X Screen/Report Format/Other User Interface XX

Provide a description, and if appropriate, an image or mockup of each screen, report, or other user interface within this function or use case. For reports that use standard reporting tools (e.g., Crystal Reports) or standard data exchange languages (e.g., XML), describe the form and formatting for Report XX that uses these technologies.

Examples of an image or mockup of a report and screen are provided below.

High School Graduates' Longitudinal Analysis — Statewide

Metrics	Year 2000-2001	1999-2000	1998-1999
Graduates Minimum High School Program	97,827	121,232	112,698
Minimum High School Program (%)	45.43%	56.94%	55.41%
Graduates Recommended and Advanced High School Program	99,454	76,358	56,398
Recommended and Advanced High School Program (%)	46.19%	35.86%	27.73%
Graduates Distinguished Achievement and Advanced Honors Program	10,661	8,463	27,522
Distinguished Achievement and Advanced Honors (%)	4.95%	3.97%	13.53%
Graduates Individual Education Plan	7,374	6,872	6,775
Individual Education Plan (%)	3.42%	3.23%	3.33%
Total Number of Graduates	215,316	212,925	203,393

Table 2. Example of a Report Format

General Information

County / District Number: 123456 e.g. 019231

Fiscal Agent Name: Test Grantee 1 e.g. Dallas ISD

Federal Project ID: 123456789 Year 1 NOGA ID

Combined Schools: Test Grantee Combined Schools

LPA Grantee ID: 1234567890

Cycle: 1

Type: School District

Project Description: Test Grantee Project Description

Active:

Site Information (Mailing Address)

Address Line 1: 123 Main Street

Address Line 2: Line 2

City: Austin

State: Texas

Zip: 78701 2342

Phone: 512 555 1234 Ext: 222

Fax: 512 444 5678

Web Site: http://www.TestGrantee.org

Figure 8. Example Screen Mockup

6.3.x.y.1 Function (or Use Case) X Screen/Other User Interface XX Fields

Provide a table that includes the following information for each field on each screen or other user interface within this function or use case:

- field name
- field label
- data source (e.g., data dictionary source or user entry)
- data type (e.g., text, character, integer)
- storage format
- scale
- bounds
- display format

- mandatory entry or fill information (e.g., element is required, every character of the element is required)
- default value
- constraints or special restrictions on the data (e.g., non-display asterisks for passwords)

In addition, if the data is selected from a pick list, then include the list of possible values or their description. If the content of a field is derived from client side calculations using other fields or values, then specify the algorithm for the calculation in a descriptive footnote to the table. If the content of a field is derived from server side calculations or lookups, then specify the source of that calculation (e.g., the class or stored procedure where the calculation occurs).

Also, specify the error messages to be displayed when the input does not meet requirements (e.g., type, format, scale, bounds, or constraints) for the field.

A sample Screen/Other User Interface Fields Table is provided as an additional tool in the Appendix.

An example of a Screen Fields Table is included below.

Field Name	Label	Source	Type	Storage Format	Scale	Bounds	Display Format	Mandatory Entry/Fill	Default Value	Constraints
Dist	County/ District Number	PID database	Num	6 digit	100000	1000001 - 999999	As stored	Yes	n/a	Must match PID lookup
FIS	Fiscal Agent Name	PID database	Text	80 char	n/a	n/a	As stored	Yes	n/a	Must match PID lookup

Table 3. Example of a Screen Fields Table

Section 7. Other Interfaces

Customize the following subsections to accurately and comprehensively document the design of any additional interfaces not described in the previous sections, including specific application-to-application interfaces, database-to-database interfaces, or other interfaces. In addition, identify the technology that will be used to enable the interaction. Each subsection should be labeled appropriately and titled descriptively to indicate the interface being documented.

7.X Interface X

Describe the interface design including the technology (e.g., XML), the protocol (e.g., TCP), any specific message formats, error conditions, handshakes, initiation and closure, and other features that define the design of the interface.

Section 8. Other Design Features

Describe any design features that are not captured in the previous sections.

Section 9. Requirements Traceability Matrix

In this section, provide reference to the location of the Requirements Traceability Matrix (RTM) that indicates traceability from the system requirements documented in the System Requirements Specification (SyRS) to the design elements documented in the System Design Description (SyDD).

The RTM is initiated in the SyRS and is updated appropriately during the life of the project to indicate traceability to the design elements documented in the SyDD, the software requirements documented in the Software Requirements Specification (SRS), and the design elements documented in the Software Design Description (SDD). The completed RTM assures that every requirement has been addressed in the design and that every design element addresses a requirement. The RTM also provides the necessary traceability for integration, acceptance, regression, and performance testing.

The Requirements Traceability Matrix in the SyDD should:

- Contain the columns used to illustrate traceability of the system requirements to the system design elements, software requirements, and software design elements
- Contain the columns necessary to illustrate traceability for integration, acceptance, regression, and performance testing
- Be populated with all requirements documented in the SyRS
- Be populated with all design elements documented in the SyDD
- Indicate traceability from the system requirements documented in the SyRS to the design elements documented in the SyDD
- Indicate the source or origin of each requirement

A sample RTM template is provided as an additional tool in the Appendix of the SyRS Template Instructions.

Section 10. References

Provide a list of all documents and other sources of information referenced in the System Design Description and utilized in developing the System Design Description. Include for each the document number, title, date, and author.

Section 11. Glossary

Define all terms and acronyms required to interpret the System Design Description properly.

Section 12. Revision History

Identify changes to the System Design Description.

Section 13. Appendices

Include any relevant appendices.

Appendix A. Sample Data Dictionary Table

Entity Name	Element Name	Definition	Type	Storage Format	Scale	Bounds	Display Format	Mandatory Entry/Fill	Default Value	Modified By	Read By	Constraints

Appendix B. Sample Screen/Other User Interface Fields Table

Screen/Other User Interface Name: _____

Field Name	Label	Source	Type	Storage Format	Scale	Bounds	Display Format	Mandatory Entry/Fill	Default/Value	Constraints