<Date>

**Version <X.x>**

**NOTES TO THE AUTHOR/TEMPLATE INSTRUCTIONS**

This template includes instructions and boilerplate text for this document. The author should note that:

* Each section provides instructions or describes the intent, assumptions, and context for content included in that section. Instructional text appears in *blue italicized text* throughout this template.
* Replace instructional text in each section with project specific information.
* Use or modify boilerplate examples of wording and formats for text and tables as appropriate.
* Search and replace all text enclosed in angle brackets - < > - with project specific information (e.g., <Project Name> or <Project Acronym>.

Follow these steps when using this template:

1. All documents must be compliant with Section 508 requirements. Refer to [FNS 504-508 Compliance Reference Library](https://fncspro.usda.net/offices/oit/Documents/Forms/AllItems.aspx?RootFolder=%2Foffices%2Foit%2FDocuments%2F504%2D508%20Compliance%20%2D%20Accessibility&View=%7BB47848DF%2D6059%2D4D43%2DAB0C%2D8ECEDC92AD4C%7D) or [Section508.gov](https://section508.gov/) for more information.
2. Modify any boilerplate text, as appropriate, for your specific project.
3. Use Styles for new sections such as Heading 1, Heading 2 and Std Para.
4. Place Table captions and descriptions *above* the table and centered. All tables must have an associated tag providing appropriate alternative text for Section 508 compliance.
5. Place Figure captions and descriptions *below* the figure and centered. All figures must have an associated tag providing appropriate alternative text for Section 508 compliance.
6. Update the Table of Contents and any List of Tables or List of Figures by right-clicking it and selecting Update field / Update entire table.
7. Delete this “Notes to the Author/Template Instructions” page and all instructional text to the author before finalizing the draft of the document.

**SDLC TEMPLATE REVISION HISTORY**

| VERSION | DATE | CHANGE DESCRIPTION |
| --- | --- | --- |
| 1.0 | 03/21/13 | Created the document |
| 1.1 | 09/09/13 | Re-formatted the document. |
| 1.2 | 03/23/17 | Re-formatted the document. |
| 1.3 | 04/06/17 | Updated the document to reflect changes |
| 1.4 | 11/02/20 | Updated to reflect new FNS Agile SDLC processes and comply with Section 508 standards. |

**SDLC TEMPLATE CONTACT INFORMATION**

| RESPONSIBILILTY | CONTACT PERSON | EMAIL ADDRESS |
| --- | --- | --- |
| Portfolio Management Division Director, Chief Portfolio Officer | Joe Shaw | [Joseph.Shaw@usda.gov](mailto:Joseph.Shaw@usda.gov) |
| IT Governance Manager | Kevin Russ | [Kevin.Russ@usda.gov](mailto:Kevin.Russ@usda.gov) |
| SDLC Lead | Max Mounger | [Max.Mounger@usda.gov](mailto:Max.Mounger@usda.gov) |

*Complete the below table to track the version history of this document.*

**DOCUMENT REVISION HISTORY**

| **VERSION** | **DATE** | **AUTHOR** | **CHANGE DESCRIPTION** |
| --- | --- | --- | --- |
| x.x |  |  |  |
| x.x |  |  |  |
| x.x |  |  |  |

*List the relevant area of responsibility, contact person and email address for this document.*

**CONTACT INFORMATION**

| **RESPONSIBILILTY** | **CONTACT PERSON** | **EMAIL ADDRESS** |
| --- | --- | --- |
| Area of Responsibility |  |  |
| Area of Responsibility |  |  |
| Area of Responsibility |  |  |
| Area of Responsibility |  |  |
| Area of Responsibility |  |  |

Table of Contents

[1. Introduction 1](#_Toc55236398)

[1.1 Scope 1](#_Toc55236399)

[1.2 Points of Contact 1](#_Toc55236400)

[1.3 Document References 1](#_Toc55236401)

[2. System Overview 2](#_Toc55236402)

[2.1 System Description 2](#_Toc55236403)

[2.2 Design Constraints 3](#_Toc55236404)

[2.3 Future Contingencies 3](#_Toc55236405)

[3. System Architecture Detailed Design 4](#_Toc55236406)

[3.1 Logical View 4](#_Toc55236407)

[3.2 Hardware 5](#_Toc55236408)

[3.3 Software 6](#_Toc55236409)

[3.3.1 Module Description 7](#_Toc55236410)

[3.3.2 Module Constraints 7](#_Toc55236411)

[3.3.3 Function(s)/Services 8](#_Toc55236412)

[3.3.4 Data Model 8](#_Toc55236413)

[3.4 File and Database Architecture 8](#_Toc55236414)

[3.5 Security Architecture 12](#_Toc55236415)

[4. User Interface 12](#_Toc55236416)

[5. External Interfaces 14](#_Toc55236417)

[Attachment A – Additional reference / information 16](#_Toc55236418)

**LIST OF TABLES**

[Table 1 - Points of Contact 1](#_Toc54076312)

[Table 2 - Release Documentation 2](#_Toc54076313)

[Table 3 - Design Contingencies 3](#_Toc54076314)

[Table 4 - Hardware Architecture 5](#_Toc54076315)

[Table 5 – Software Architecture 7](#_Toc54076316)

[Table 6 - Data Dictionary 11](#_Toc54076317)

**LIST OF FIGURES**

[Figure 1 – High-Level System Architecture Diagram 3](#_Toc54076318)

[Figure 2 – Logical View 5](#_Toc54076319)

[Figure 3 – Physical Architecture 6](#_Toc54076320)

[Figure 4 - Software Architecture 7](#_Toc54076321)

[Figure 5 - Sample Data Model Diagram 8](#_Toc54076322)

[Figure 6 – Example Normalized Table Layout 9](#_Toc54076323)

[Figure 7 - Sample Entity Relationship Diagram 10](#_Toc54076324)

[Figure 8 - Sample Module Screen 13](#_Toc54076325)

[Figure 9 - Sample Report 14](#_Toc54076326)

[Figure 10 – External Interfaces Architecture 15](#_Toc54076327)

*List the acronym reference and definition or description for each acronym contained in this document.*

**ACRONYM LIST**

| **REFERENCE** | **DEFINITION** |
| --- | --- |
| COTS | Commercial Off-The-Shelf |
| DBMS | Database Management System |
| FNS | Food and Nutrition Service |
| GUI | Graphical User Interface |
| HLD | High Level Design |
| IT | Information Technology |
| ITG | Information Technology Governance |
| OIT | Office of Information Technology |
| PTM | Product Traceability Matrix |
| SDD | System Design Document |
| SME | Subject Matter Expert |
| SSO | Single Sign-On |
| UML | Unified Modeling Language |
| USDA | United States Department of Agriculture |

# Introduction

*Insert statement about the purpose of this System Design Document (SDD) and what it intends to do or paste information from existing document into this document. Since the SDD is a living document in agile, complete it at the end of the initial development cycle. Update the SDD with each major release or review it at a minimum annually to address design updates, as appropriate.*

Also describe the purpose of the system and its benefits to the Food and Nutrition Service (FNS). Below is sample language.

The System Design Document within agile development is a living document that spans the lifecycle of development activities. This document describes the system overview; system and subsystem architecture; detailed design for user interfaces, hardware, software, file and database architecture, security, and internal communications; and external interfaces.

## Scope

Provide a description of the intended scope of the system, how it will accomplish its purpose.

## Points of Contact

Identify the key points of contact involved in the system development project, including project sponsors, user organizations, development organizations, support organizations, and certifying organizations. For each individual, provide the name, title, telephone number and e-mail address as shown in the below Points of Contact table.

The table below lists the primary points of contacts for the <Project Name> application team.

Table - Points of Contact

| **Name** | **Title** | **Contact Phone Number** | **Contact Email** |
| --- | --- | --- | --- |
|  | Project Sponsor |  |  |
|  | Product/Business Owner |  |  |
|  | OIT Project Manager |  |  |
|  | SME |  |  |
|  | Business Analyst |  |  |
|  | Architect |  |  |
|  | Developer |  |  |
|  | Tester |  |  |
|  | End User |  |  |

## Document References

List any other documents that support this SDD. Include any referenced project documents, documents that preceded this SDD, or documents that provided information for its development. Add rows to the table as necessary. This information may be included in separate attachments, if necessary.

The following table lists the important and referenced documentation for this SDD.

Table - Release Documentation

| **Document Title** | **Version No.** | **Last Update** | **Comments** |
| --- | --- | --- | --- |
| <insert document title> |  |  |  |
| <insert document title> |  |  |  |
| <insert document title> |  |  |  |

# System Overview

This section provides system overview information to include system description, design constraints, and future contingencies.

## System Description

Describe the system in narrative form using non-technical terms. Provide a high-level design (HLD) explaining the architecture that would be used for developing the system. The architecture diagram provides an overview of the entire system, identifying the main components of the new system and their interfaces. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable. Refer to the Product Traceability Matrix (PTM) or Product Backlog to identify the allocation of the functional requirements into this SDD.

Provided below is a simplified example of a high-level architecture diagram.

Replace picture by right-clicking on picture and selecting Change Picture.

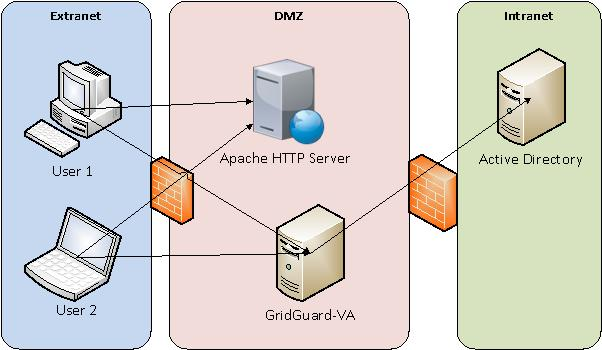


Figure – High-Level System Architecture Diagram

## Design Constraints

Describe any constraints in the system design (reference any trade-off analyses conducted such as resource use versus productivity, or conflicts with other systems) and include any assumptions made by the project team in developing the system design.

## Future Contingencies

Describe any contingencies that might arise in the design of the system that may change the development direction. Possibilities include lack of interface agreements with outside agencies or unstable architectures known at document completion. Address any possible workarounds or alternative plans.

The following table describes the design contingencies and possible workarounds or alternative plans for the system.

Table - Design Contingencies

| **No.** | **Design Contingencies** | **Workarounds/Alternative Plans** |
| --- | --- | --- |
| 1 | Name of the design contingency | Briefly describe workaround or alternative plans. |
| 1 |  |  |
| 2 |  |  |

# System Architecture Detailed Design

This section describes the system and/or subsystem(s) architecture for the project. It provides the information needed for a system development team to actually build and integrate the hardware components and code, integrate the software modules, and interconnect the hardware and software segments into a functional product. Additionally, this section addresses the detailed procedures for combining separate commercial off-the-shelf (COTS) packages into a single system, if applicable. Map every detailed requirement/user story back to the requirements document/Product Backlog, and present the mapping in a traceability matrix format. Include an appendix to this SDD. Minimize references to external entities if possible. Do not include sensitive information such as IP addresses in diagrams or tables.

This section describes the system and/or subsystem architecture and detailed design for the project to include system logical view. It also describes separately the architecture for hardware, software, file and database architecture, internal communications, and security.

## Logical View

Provide any related logical views or provide a reference to where they are located. The logical view provides a basis for understanding the structure and organization of the system design. The logical view is concerned with the functionality that the system provides to end-users. This view shows the components (objects) of the system as well as their interaction/relationship. You can use Unified Modeling Language (UML) diagrams to represent the logical view, and include class diagrams and state diagrams, as appropriate. Below is an example of a logical model.

Replace picture by right-clicking on picture and selecting Change Picture.

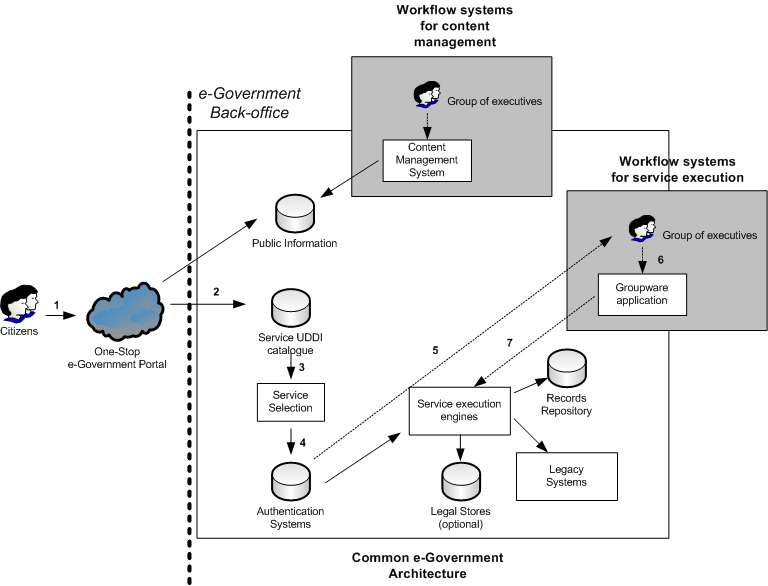


Figure – Logical View

## Hardware

In this section, describe the overall system hardware. Include a list of hardware components with a brief description of each item. Include a physical architecture diagram showing the connectivity between the components. Provided below is a simple example.

A hardware component is the lowest level of design granularity in the system. Depending on the design requirements, there may be one or more components per system. This section should provide enough detailed information about individual component requirements to correctly build and/or procure all the hardware for the system (or integrate COTS items).

If there are many components or if the component documentation is extensive, place it in an appendix or reference a separate document. Add additional diagrams and information, if necessary, to describe each component and its functions adequately. Follow industry standard component specification practices. For COTS procurements, include specific vendor item names if appropriate.

Create any subsections necessary to organize and present your information.

The following table details the overall system hardware with a description.

Table - Hardware Architecture

| **No.** | **Hardware** | **Description** |
| --- | --- | --- |
| 1 | Name of the hardware | Description of hardware |
| 1 |  |  |
| 2 |  |  |

The following diagram shows the physical architecture and connectivity between components.

Replace picture by right-clicking on picture and selecting Change Picture.

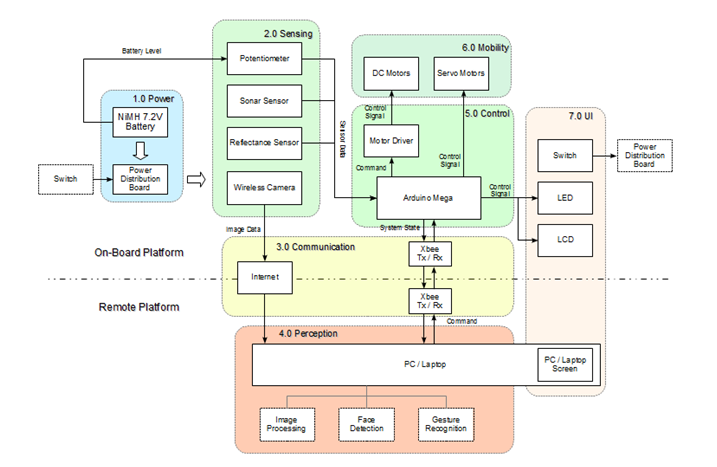


Figure – Physical Architecture

## Software

Describe the overall system software. Include a list of software modules (this could include functions, subroutines, or classes), computer languages, and programming computer-aided software engineering tools (with a brief description of the function of each item).

A software module is the lowest level of design granularity in the system. Depending on the software development approach, there may be one or more modules per system. This section should provide enough detailed information about logic and data necessary to completely write source code for all modules in the system (and/or integrate COTS software programs).

If there are many modules or if the module documentation is extensive, place it in an appendix or reference a separate document. Add additional diagrams and information, if necessary, to describe each module, its functionality, and its hierarchy. Follow industry standard module specification practices. Include the following information in the detailed module designs:

* *A description of each module, its function(s), the conditions under which it is used (called or scheduled for execution), its overall processing, logic, interfaces to other modules, interfaces to external systems, security requirements, etc.*
* *For COTS packages, specify any call routines or bridging programs to integrate the package with the system and/or other COTS packages (for example, Dynamic Link Libraries)*
* *Data elements, record structures, and file structures associated with module input and output*
* *Graphical representation of the module processing, logic, flow of control, and algorithms, using an accepted diagramming approach (for example, structure charts, action diagrams, flowcharts, etc.)*

Use the below subsections to organize the information, if appropriate. Delete if not applicable.

Note: The diagrams should map to any data flow diagrams or other diagrams, providing the physical process and data flow.

Table – Software Architecture

| **No.** | **Software** | **Description** |
| --- | --- | --- |
| 1 | Name of software | Description of software |
| 1 |  |  |
| 2 |  |  |

The following diagram shows the connectivity between components.

Replace picture by right-clicking on picture and selecting Change Picture.

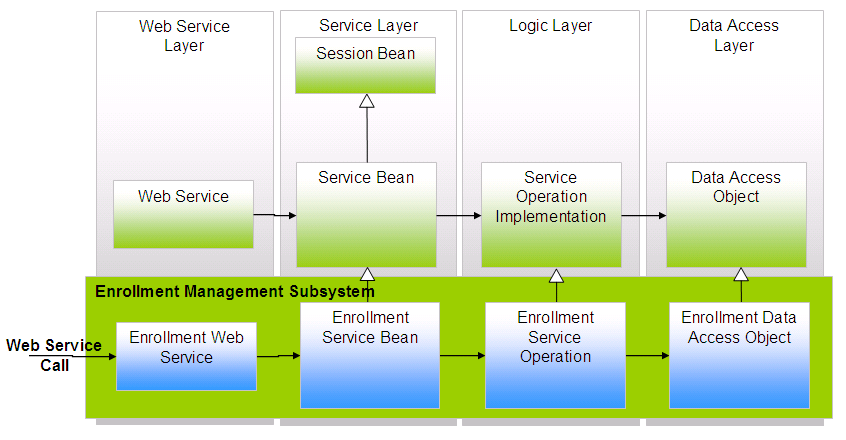


Figure - Software Architecture

### Module Description

Provide a detailed description of each software module.

### Module Constraints

Identify any relevant assumptions, limitations, or constraints for this component. This should include constraints on timing, storage, or component state, and might include rules for interacting with this component.

### Function(s)/Services

List functions and services related to this module.

### Data Model

Provide specific data model diagram for the module. For example:

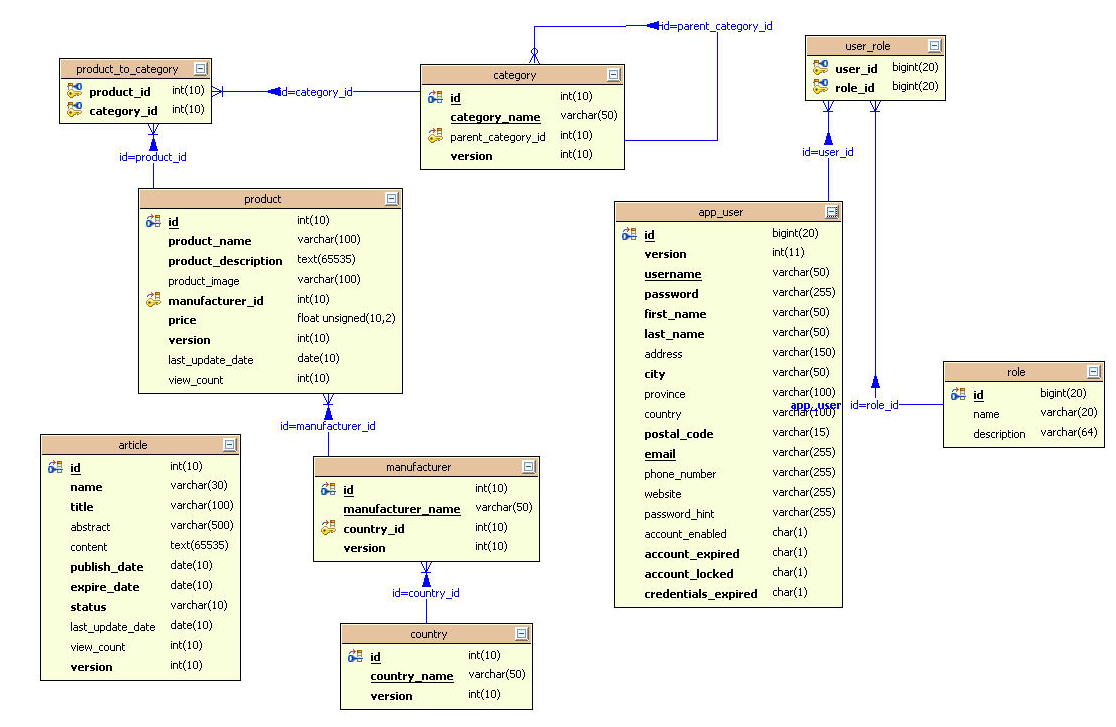


Figure - Sample Data Model Diagram

## File and Database Architecture

Describe the final design of all Database Management System (DBMS) files and the non-DBMS files associated with the system under development.

Provide a comprehensive data dictionary showing data element name, type, length, source, validation rules, maintenance (i.e., create, read, update, and delete capability), data stores, outputs, aliases, and description. This can be included as an attachment or appendix.

Include the following information as appropriate for non-database management system files (refer to the data dictionary):

* *Refined logical model; provide normalized table layouts, entity relationship diagrams, and other logical design information*
* *A physical description of the DBMS schemas, sub-schemas, records, sets, tables, and storage page sizes*
* *Access methods (i.e., indexed, via set, sequential, random access, or sorted pointer array)*
* *Estimate of the DBMS file size or volume of data within the file and data pages, including overhead resulting from access methods and free space*
* *Definition of the update frequency of the database tables, views, files, areas, records, sets, and data pages; estimate the number of transactions if the database is an online transaction-based system*

Create subsections as necessary to organize your information.

Provided below is an example of normalized table layouts, if applicable.

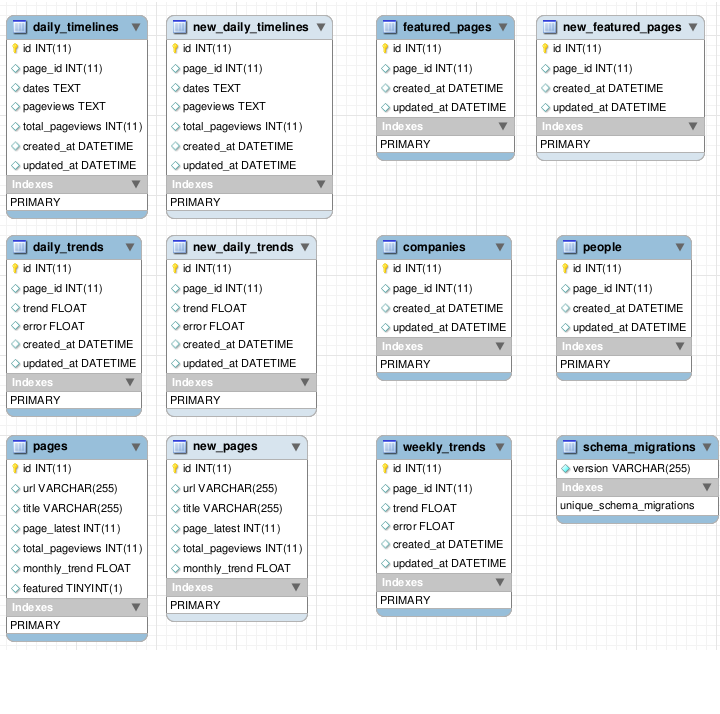


Figure – Example Normalized Table Layout

Below is an example of an Entity Relationship Diagram, if applicable.

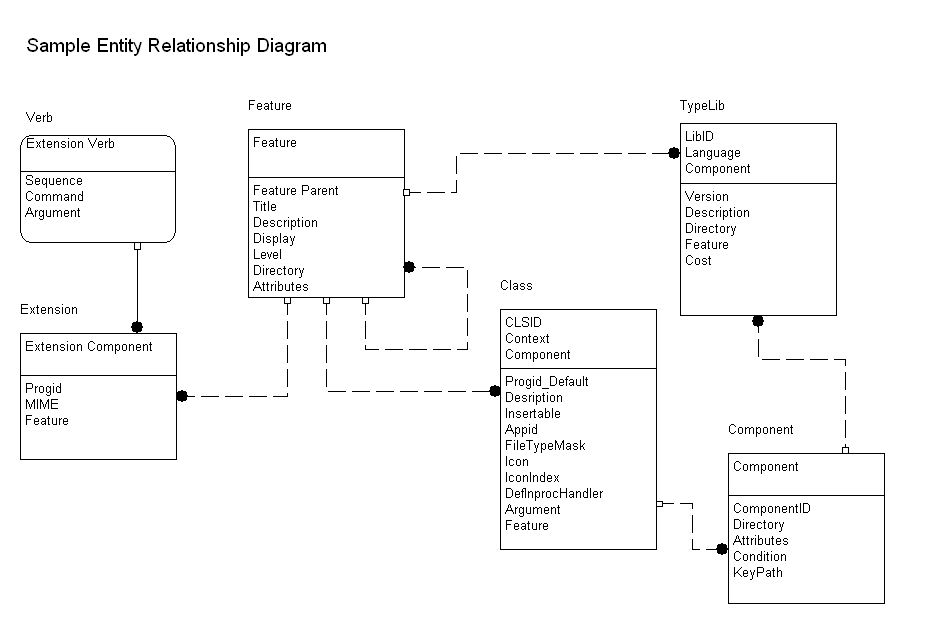


Figure - Sample Entity Relationship Diagram

A note on null values: Null is the absence of a recorded value for a field. A null value differs from a value of zero in that zero may represent the measure of an attribute, while a null value indicates that no measurement has been taken.

Below is the data dictionary for <Project Name>.

Table - Data Dictionary

| **Element or Value Display Name** | **Description** | **Data Type** | **Character Length** | **Acceptable Values** | **Required** | **Accepts Null Value?** |
| --- | --- | --- | --- | --- | --- | --- |
| What is the name used in your data file? | Write a brief definition, stated in the singular that could stand alone from other element definitions. | For example, indicate varchar, integer, date, etc. | For example, the maximum length for Excel is 255, so indicate 255 or less. | List all acceptable values, separated by pipes ( | ). This may be a field name or a range of values. | Enter y/n to indicate whether this field is required. | This is required to run calculations on your data. Indicate y/n if null value is allowable. |
| Element 1 |  |  |  |  |  |  |
| Element 2 |  |  |  |  |  |  |

## Security Architecture

Describe the application/system's user authentication control mechanisms (firewalls, single sign on (SSO), encryption, role-based, biometrics). Discuss the controls in place to authorize or restrict the activities of users and personnel within the application/system. If the general public accesses the application/system, discuss the additional security controls used to protect the application/system's integrity.

Provide a graphical representation with detailed information for each of the individual security hardware components. Specify the design for the below items as required.

* *Authentication*
* *Authorization*
* *Logging and Auditing*
* *Encryption*
* *Network ports usage*
* *Intrusion Detection and Prevention*

The design should be based on the designated system security level and provide adequate protection against threats and vulnerabilities.

# User Interface

In this section, provide the detailed design of the system and subsystem inputs and outputs relative to the user. Add any additional information to this section and organize it according to whatever structure best presents the user input and output designs. Depending on the particular nature of the project, it may be appropriate to repeat these sections at both the subsystem and design module levels.

For a developed application, this section can include subsections for User Interface Overview, User Interface Screens, and Reports. Add subsections as appropriate to document design information. Include screen captures of completed screens.

Below is a graphical representation of a module:

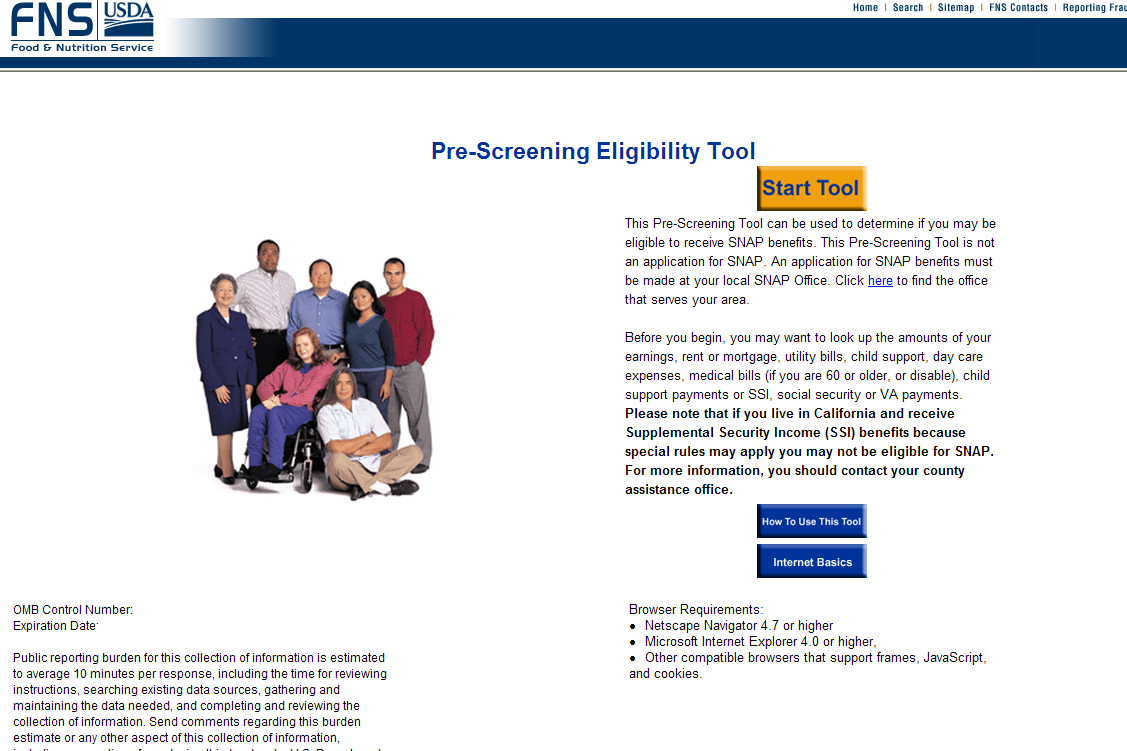


Figure - Sample Module Screen

If you are grouping by Inputs and Outputs:

For Inputs - Provide the layout of all input data screens or graphical user interfaces (GUIs) (for example, windows). Provide a graphic representation of each interface. Define all data elements associated with each screen or GUI, or reference the data dictionary. Discuss the miscellaneous messages associated with user inputs, including the following:

* *Description of any access restrictions or security considerations*
* *Each transaction name, code, and definition, if the system is a transaction-based processing system*

For Outputs - Describe the system output design relative to the user. Show a mapping to the high-level data flows described earlier, if applicable. System outputs include reports, data display screens and GUIs, query results, etc. Provide the following, if appropriate:

* *Description of report and screen contents (provide a graphic representation of each layout and define all data elements associated with the layout or reference the data dictionary)*
* *Description of the purpose of the output, including identification of the primary users*
* *Description of any access restrictions or security considerations*

Below is an example for a report.

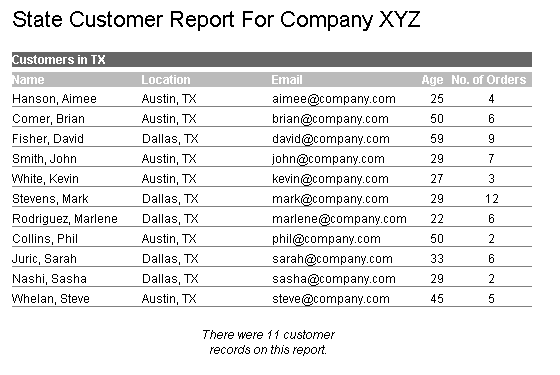


Figure - Sample Report

# External Interfaces

External systems are any systems that are not within the scope of the system under development, regardless whether USDA, FNS, or another agency manages the other systems. In this section, describe the electronic interface(s) between this system and each of the other systems and/or subsystem(s), emphasizing the point of view of the new system. Include the planned interface architecture(s).

To deliver the interface design section in an agile approach, provide a high-level design and pictorial representation at the beginning of iteration 0. At each iteration progression, document the subsystem or interface design and update any diagrams.

For each system that provides information exchange with the system under development, there is a requirement for rules governing the interface. Provide enough detailed information about the interface requirements to correctly format, transmit, and/or receive data across the interface. Include the following information in the detailed design for each interface (as appropriate):

* *The data format requirements; if there is a need to reformat data before they are transmitted or after incoming data is received, tools and/or methods for the reformat process should be defined*
* *Specifications for hand-shaking protocols between the two systems; include the content and format of the information to be included in the hand-shake messages, the timing for exchanging these messages, and the steps to be taken when errors are identified*
* *Format(s) for error reports exchanged between the systems; should address the disposition of error reports; for example, retained in a file, sent to a printer, flag/alarm sent to the user, etc.*

Replace picture by right-clicking on picture and selecting Change Picture.

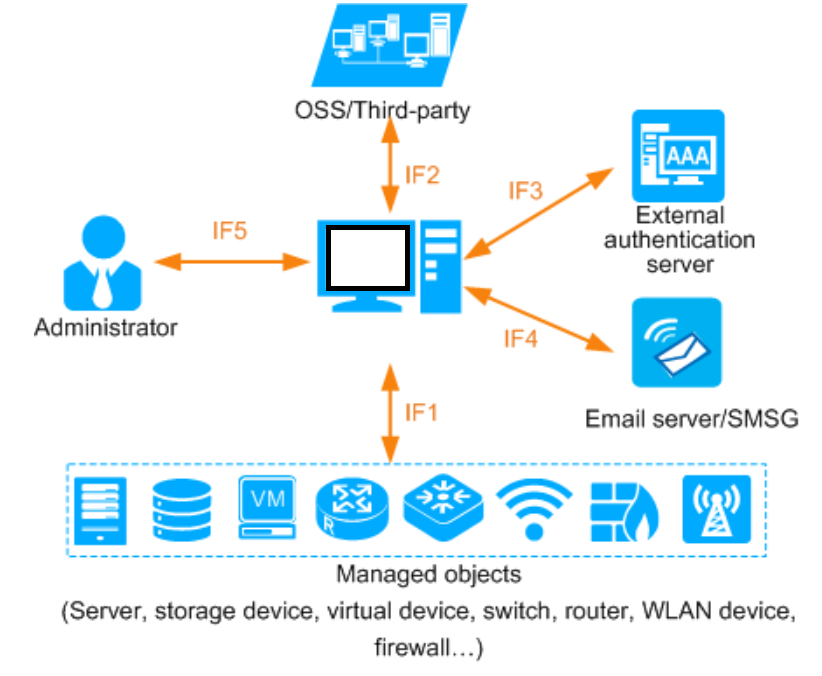


Figure – External Interfaces Architecture

Attachment A – Additional reference / information

<Insert Link to Object if applicable>