USDA UNITED STATES DEPARTMENT OF AGRICULTURE

System Requirements Specification (SRS)

of the

(Subsystem and Version #)

(Delivery Date)
(Document Revision Number)

Contract (No.)

Task (No.)

GSA Contract (No.)

Prepared for:

The United States Department of Agriculture
Food & Nutrition Service (FNS)/
Information Technology Division (ITD)
3101 Park Center Drive
Alexandria, VA 22302

TABLE OF CONTENTS

<System> System Requirements Specification Revisions

Revision Number	Date	Description

1.0 INTRODUCTION

The System Requirements Specification (SRS) is a formal statement of the application functional and operational requirements. It serves as a contract between the developer and the customer for whom the system is being developed. The developers agree to provide the capabilities specified. The client agrees to find the product satisfactory if it provides the capabilities specified in the SRS.

Sections 1 through 9 describe the contents of the SRS in accordance with the software development life cycle (SDLC). Section 10 is provided for informational purposes only and describes the general requirements for generating a Requirements Traceability Matrix (RTM) in tandem with the SRS. It provides general guidelines for writing requirements. Section 11 provides information on optional Attachments and Appendices.

A brief description of SRS functions, characteristics, and requirements structure includes the following:

- The SRS provides the following functions:
  - Designing and developing the application system
  - Evaluating the product in all subsequent phases of the lifecycle
  - Determining the success of the project
- The SRS has the following characteristics:
  - Demonstrates that the application provides value to FNS in terms of the business objectives and business processes.
  - Contains a complete set of requirements for the application.
  - Is solution independent. The SRS is a statement of what the application is to do—not of how it works. The SRS does not commit the developers to a design. For that reason, any reference to the use of a specific technology is inappropriate in an SRS, unless the technology is listed as a system constraint (see Section 2.2).
- The SRS provides the following requirements, where a requirement is defined as a condition the application must meet for the customer to find the application satisfactory. A requirement has the following characteristics:
  - Provides a benefit to the organization. That benefit is directly traceable to the business objectives and business processes of the FNS.
  - Describes the capabilities the application must provide in business terms.
  - Does not describe how the application provides that capability.
  - Does not describe such design considerations as computer hardware, operating system, and database design.
  - Is stated in unambiguous words. Its meaning is clear and unmistakable.
  - Is stated in the positive mode, using the word "shall" (e.g., The system shall. . .). Only one instance of "shall" is permitted in a paragraph.

– Is verifiable.

1.1 Purpose

In this section, provide the purpose this application is intended to serve. Describe the business objectives and business processes from the Concept of Operations (ConOps) document and the cost-benefit analysis (CBA) that this application supports.

1.2 Scope

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

1.3 Policy

Identify FNS policy decisions that affect the conduct of CM on the project.

1.4 System Description

In this section, provide an overview of the physical system.

Describe the system architecture, operating system, and application languages.

Give an estimate of the size and complexity of the system in terms of number of user types, number of locations, interfaces, number of major processes, data capacity in business terms, numbers of major processes, etc.

Summarize the conditions that created the need for the new system (or capability).

Include any relevant background, such as the number of sites that are using the system.

Identify other legacy or new systems with which this system interfaces.

1.5 Points of Contact

List the names, titles, and roles of the major participants in the project. At a minimum, list the following:

- FNS IT Project Manager
- Development project leader
- User contacts
- FNS employee whose signature constitutes approval authority for the SRS

1.6 Document References

List the documents that are sources or references for this SRS. Include meeting summaries, white paper analyses, and SDLC deliverables, as well as any other documents that preceded this SRS and provided information for the development of it. Also reference any documents that provided information on the relevant FNS version or business plan.

1.7 Glossary

Include a list of terms, abbreviations, and acronyms used in this document. If the list is longer than a page, it can be included as an Appendix or Attachment to the SRS. If an attachment or appendix is used, include a reference to it here.

2.0 ASSUMPTIONS AND CONSTRAINTS

2.1 Assumptions

State the assumptions associated with development of the system, where assumptions are defined as future situations, beyond the control of the project, whose outcomes influence the success of a project. The following are examples of assumptions:

- Availability of a hardware/software platform
- Pending legislation
- Court decisions that have not been rendered
- Future trends in immigration and naturalization
- Developments in technology

2.2 Constraints

State the constraints associated with the development of the system, where constraints are defined as conditions outside the control of the project that limit the design alternatives.

Constraints can be broadly categorized as technical and non-technical. The following are examples of both types of constraints:

- Government regulations
- Technical standards imposed on the solution (for example, the use of a specific Database Management System)
- Strategic decisions

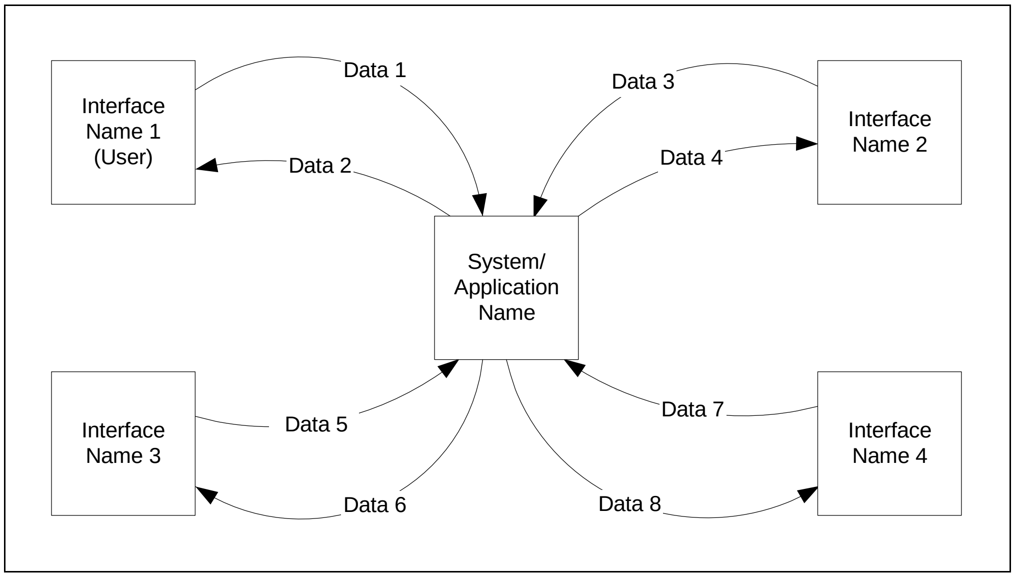
Distinguish constraints from preferences, as follows:

- Constraints exist because of real business conditions. For example, a delivery date is a constraint only if there are real business consequences that will happen as a result of not meeting the date. If failing to have the subject application operational by the specified date places the FNS in legal default, the date is a constraint.
- Preferences are arbitrary. For example, a date chosen arbitrarily is a preference. Preferences, if included in the SRS, should be noted as such.

3.0 CONTEXT

Provide a context diagram of the system, with explanations as applicable. The context of a system being developed refers to the connections and relationships between the system and the outside world. Context Diagrams are often used to illustrate these connections and relationships. Exhibit 1-1 illustrates a generic context diagram. If users interact with this system, user interfaces must be shown explicitly in the context diagram.

Exhibit 1-1. Generic Context Diagram



4.0 FUNCTIONAL REQUIREMENTS

The functional requirements describe the core functionality of the application. This section includes the data and functional process requirements. Generally, the ConOps is the source for specified or derived requirements.

4.1 Functional Process Requirements

Process requirements describe what the application must do. Process requirements relate the entities and attributes from the data requirements to the users' needs.

State the functional process requirements in a manner that enables the reader to see broad concepts decomposed into layers of increasing detail.

Exhibit 1-2 is a sample of process requirements levels stated in such a manner. It is presented only as an example of requirements headings decomposed from generalities into levels of increasing detail.

Exhibit 1-2: Sample Process Requirements

Section ID	Requirement Description
4.1	Functional Process Requirements (this section)
4.2	Manage Naturalization Process
4.2.1	Record Request for Naturalization
4.2.2	Determine Eligibility for Naturalization
4.2.3	Determine Suitability for Naturalization
4.2.3.1	Determine Proficiency with the English Language
4.2.3.2	Determine Knowledge of U.S. History, Law, and Customs
4.2.3	Administer Citizenship Oath
4.2.4	Confer Citizenship
4.3	Manage Schedules of FNS Personnel in Performing Naturalization Activities

4.1.1 Requirement Statement

The requirements must be phrased with the definitive word “shall” and have a unique number for reference purposes. Each requirement and its identifier must be in a separated paragraph, i.e., one “shall” per paragraph. See also Section 10 of this document. The on-line RequisitePro FNS Template contains the standard formatting for requirements numbers, including prefixes.

Exhibit 1-3 is an example of requirements stated in such a manner.

Exhibit 1-3: Sample Process Requirements

Section/ Requirement ID	Requirement Definition
4.5.2	Provide Project Planning Capability
SR4.5.2.0*	The system shall provide a project planning capability
SR4.5.2.0.1	The system shall allow authorized users to enter project plans and timelines
SR4.5.2.0.2	The system shall allow authorized users to review, change, or update project plans and timelines
* For this example, “F” is used to designate a Functional requirement. Other designations could be “DF” for Design Feature or “STRQ” for Stakeholder Request.	

4.1.2 Requirement Cross-Identification

4.1.2.1 Identification In Text

The unique identification of requirements is an essential attribute of the requirement itself. Requirements that include lists can be handled in one of two ways. Each item in a list can have its own “shall” statement or be numbered (a), (b), (c), etc., or (i), (ii), (iii), etc. In this way, a requirement can be identified by its paragraph requirement number coupled with letter (a) or number (i).

Caution: Bullets or other repetitive symbols are not permissible because they would not be unique identifiers.

Note: The parentheses shown around the sublevel identifiers are shown for emphasis purposes only and are not part of the standard.

4.1.2.2 Identification In Table

Sometimes requirements such as performance numbers are listed in tables. In such a case, each cell in a table is a requirement. Therefore, the first column of the table shown contains a unique identifier. Additionally, under the labels of the columns must be the letters (a), (b), (c), etc., or (i), (ii), (iii), etc., for the same reasons as discussed previously. Exhibit 1-4 illustrates the generic format of requirements in tables. Use as many columns as needed.

Exhibit 1-4: Generic Format of Requirements in Tables

Requirement Number	<Label 1> (a)	<Label 2> (b)	<Label 3> (c)	<Label 4> (d)	<Label 5> (e)
#	Requirement	Requirement	Requirement	Requirement	Requirement

Note: Include the letters (a), (b), (c), etc., or (i), (ii), (iii), etc.

4.1.3 Process Requirements

Process requirements may be supplemented with data flow diagrams, text, or any technique that provides the following information about the processes performed by the application:

- Context
- Detailed view of the processes, including:
  - Data (attributes) input to and output (including reports) from processes
  - High-level logic used inside the processes to manipulate data (do not state "how")
  - Accesses to stored data

Illustrate the high-level functions of the system in a “level 0” diagram. Exhibit D1-5 depicts an example of a level-0 diagram.

Caution: The major functions must match those identified in the Level-0 diagram.

4.2 Function “N”

Provide the requirements for each function of the system, adding subsections (e.g., 4.2, 4.3, etc.) as needed and as described in Section 4.

5.0 INTERFACE REQUIREMENTS

Interface requirements describe interaction of the system with users, hardware, software, and communications.

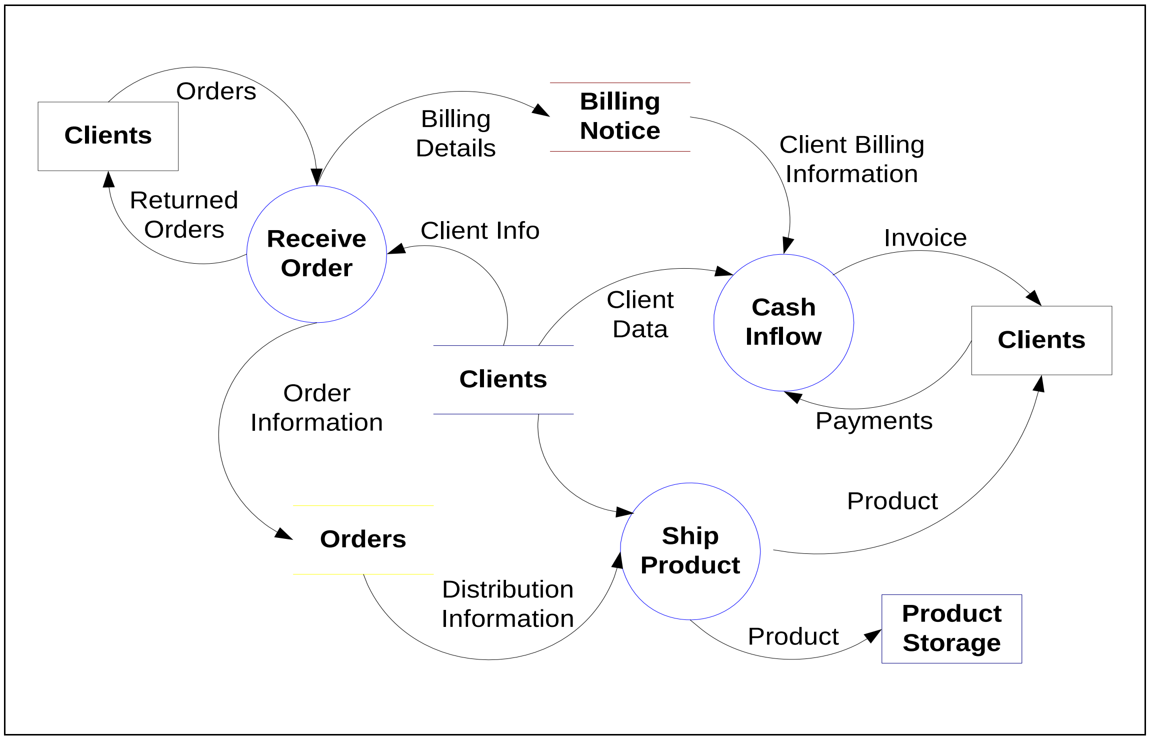
5.1 User Interfaces

Describes the user interfaces that are to be implemented by the system.

5.2 Hardware Interfaces

Define any hardware interfaces that are to be supported by the system, including logical structure, physical addresses, and expected behavior.

Exhibit 1-5. Sample Level-0 Data Flow Diagram



5.3 Software Interfaces

Name the applications with which the subject application must interface. State the following for each such application:

- Name of application
- Owner of application (if external to the FNS)
- Details of interface (only if determined by the other application)

5.4 Communications Interfaces

Describe any communications interfaces to other systems or devices, such as local area networks.

6.0 DATA REQUIREMENTS

Describe the data requirements by providing data entities, their decomposition, and their definitions. The data requirements describe the business data needed by the application system. Data requirements do not describe the physical database and they are not at the level of identifying field names. For this purpose, a data dictionary should be provided, showing data entities, their attributes, and description. Refer to Section 4.1 for guidelines on formatting requirements in tables.

7.0 OPERATIONAL REQUIREMENTS

Provide the operational requirements in this section. Operational requirements describe how the system will run and communicate with operations personnel.

Do not state how these requirements will be satisfied. For example, in the Reliability section, answer the question, “How reliable must the system be?” Do not state what steps will be taken to provide reliability. The rules for stating requirements, outlined in Section 4.1, also apply to these requirements.

Distinguish preferences from requirements. Requirements are based on business needs. Preferences are not. If, for example, the user expresses a desire for sub-second response but does not have a business-related reason for needing it, that desire is a preference.

Other applicable requirements on system attributes may be added to the list of subsections below. If there is a ConOps for the system or application, all subsections listed in Section 6 of the ConOps document must be addressed in Section 7 of the SRS.

7.1 Security

The Security Section describes the need to control access to the data. This includes controlling who may view and alter application data. Use the following criteria:

- State the consequences of the following breaches of security in the subject application:
  - Erasure or contamination of application data
  - Disclosure of Government secrets
  - Disclosure of privileged information about individuals
- State the type(s) of security required. Include the need for the following as appropriate:
  - State if there is a need to control access to the facility housing the application.
  - State the need to control access by class of users. For example, “No user may access any part of this application who does not have at least a (specified) clearance.”
  - State the need to control access by data attribute. State, for example, if one group of users may view an attribute but may not update it while another type of user may update or view it.
  - State the need to control access based on system function. State, for example, if there is a need to grant access to certain system functions to one type of users, but not to others. For example, "The system shall make Function N available to the System Administrator only."
  - State if there is a need for accreditation of the security measures adopted for this application. For example, C2 protection must be certified by an independent authorized organization.

7.2 Audit Trail

List the activities that will be recorded in the application’s audit trail. For each activity, list the data to be recorded.

7.3 Data Currency

Data currency is a measure of how recent data are. This section answers the question, "When the application responds to a request for data, how current must the data be?" Answer that question for each type of data request.

7.4 Reliability

Reliability is the probability that the system will be able to process all work correctly and completely without being aborted. Reliability is evaluated as follows:

- State the following in this section:
  - What damage can result from failure of this system?
    - Loss of human life
    - Complete or partial loss of the ability to perform a mission-critical function
    - Loss of revenue
    - Loss of employee productivity
  - What is the minimum acceptable level of reliability?
- State required reliability in any of the following ways:
  - Mean Time Between Failure is the number of time units the system is operable before the first failure occurs.
  - Mean Time To Failure is computed as the number of time units before the system is operable divided by the number of failures during the time period.
  - Mean Time To Repair is computed as the number of time units required to perform system repair divided by the number of repairs during the time period.

7.5 Recoverability

Recoverability is the ability to restore function and data in the event of a failure.

Answer the following questions in this section:

- In the event the application is unavailable to users (down) because of a system failure, how soon after the failure is detected must function be restored?
- In the event the database is corrupted, to what level of currency must it be restored? For example "The database must be capable of being restored to its condition of no more than 1 hour before the corruption occurred."
- If the processing site (hardware, data, and onsite backup) is destroyed, how soon must the application be able to be restored?

7.6 System Availability

System availability is the time when the application must be available for use. Required system availability is used in determining when maintenance may be performed.

In this section, state the hours (including time zone) during which the application is to be available to users. For example, "The application must be available to users Monday through Friday between the hours of 6:30 a.m. and 5:30 p.m. EST." If the application must be available to users in more than one time zone, state the earliest start time and the latest stop time.

Include the times when usage is expected to be at its peak. These are times when system unavailability is least acceptable.

7.7 Fault Tolerance

Fault tolerance is the ability to remain partially operational during a failure. Describe the following in this section:

- Which functions do not need to be available at all times?
- If a component fails, what (if any) functions must the application continue to provide?
- What level of performance degradation is acceptable?

For most applications, there are no fault tolerance requirements. When a portion of the application is unavailable, there is no need to be able to use the remainder of the application.

7.8 General Performance

Describe the requirements for the following:

- Response time for queries and updates
- Throughput
- Expected rate of user activity (for example, number of transactions per hour, day, or month)

Specific performance requirements related to a specific functional requirement should be listed with that functional requirement (see Section 4.1 for guidance).

7.9 Capacity

List the required capacities and expected volumes of data in business terms. For example, state the number of cases about which the application will have to store data. For example, "The system shall be able to process a projected volume of 600 applications for naturalization per month." State capacities in terms of the business. Do not state capacities in terms of system memory requirements or disk space.

7.10 Data Retention

Describe the length of time the data must be retained. For example, "Information about an application for naturalization shall be retained in immediately accessible form for 3 years after receipt of the application."

8.0 USER CLASSES AND MODES OF OPERATION

8.1 Classes/Categories of Users

Identify and describe the major classes/categories of users that will interact with the system (or capability).

8.2 User Classes Mapped to Functional Features

In this section provide an explanation of what functions each user organization can access or use. Define any variations in the user work process that correspond to the use of the system by the different classes of users.

8.3 Operational Scenarios

Develop sample usage scenarios for each major user class that show what inputs will initiate the system functions, system interactions, and what outputs are expected to be generated by the system. The scenarios should be comprehensive, to the extent that all user types and all major functions are covered.

9.0 GUIDELINES FOR REQUIREMENTS DEVELOPMENT

The following subsections are not components of the SRS. They are given here as guidelines for developing requirements.

9.1 Criteria for Good Requirement

Good requirements are necessary for the development of systems that satisfy user expectations. Good requirements support clear design, development, and test procedures. A good requirement has several important characteristics, as described herein. These criteria for good quality requirements are based on the IEEE Std 830, *Recommended Practice for Software Requirements Specifications*, dated November 1998, and are described herein.

9.2 Unambiguous

It has only one interpretation. It avoids technical jargon. If specific technological terms are required for clarity, they must be defined in the requirement.

9.3 Complete

It contains all information needed to write software for the requirement that is acceptable to the customer. It does not contain the phrase “to be determined” (TBD) unless it is accompanied by a description of the conditions required for its elimination, as well as a specification of who is responsible for its elimination.

9.4 Correct

It adequately reflects the desired software function and performance.

9.5 Consistent

It does not conflict with other requirements within the same system or related systems. It agrees with requirements defined in higher-level system documentation.

9.6 Verifiable

It can be tested and confirmed.

9.7 Appropriate

It adds value to an organization by improving its process. It is within the scope of the current statement of work.

9.8 Implementation Independent

It can be satisfied by more than one design and implementation. It does not mention specific hardware or software.

9.9 Achievable

It can be implemented within project time and budget constraints using available technology.

9.10 Written in a Consistent Format

It begins with “The system shall….” Multiple sentences are phrased, “The system shall also….” (It is preferable that each requirement be stated in its own paragraph, i.e., one “shall” statement per paragraph.)

9.11 Concise

It describes a single need. It is as short as possible without adversely impacting understandability.

9.12 Ranked for Importance

It has an identifier to distinguish it in a document and for reference purposes.

10.0 REQUIREMENTS TRACEABILITY MATRIX

When preparing an SRS, during the Requirements Definition Phase, also, create an initial RTM. Provide the initial RTM as a standalone document. Requirements and their unique identifiers appearing in the body of the SRS must appear verbatim in the RTM. The other significant column of the RTM at this stage of its development is the source of the requirement, which for the most part would come from the associated ConOps. Additionally, every requirement entry in the RTM must be identified with its source.

11.0 ATTACHMENTS OR APPENDICES

Note: Attachments are listed as shown in the SRS outline at the end of this document. They are numbered in sequence here as part of the descriptive text.

11.1 Attachment/Appendix A—Glossary

If a glossary has not been provided as Section 1.6, in this attachment (or appendix), provide business terms peculiar to the system as well as the acronyms and abbreviations used in the document.

11.2 Attachment/Appendix B—Other

Provide any other attachments (or appendices) as needed; e.g., Data Dictionary.

Note: An initial RTM shall be provided in the SRS. The final RTM will be provided as a standalone document. See Section 11.