Software Requirements Specification

for

Enterprise System for Grants, Contracts and Loans Management

Version 1.0 **Draft (Not Yet Reviewed by User Group)**

Office of Financial Management

Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Table of Contents ii
Revision History ii
1. Introduction 1
1.1 Purpose 1
1.2 Document Conventions 1
1.3 Intended Audience and Reading Suggestions 1
1.4 Project Scope 1
1.5 References 1
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Features 2
2.3 User Classes and Characteristics 2
2.4 Operating Environment 2
2.5 Design and Implementation Constraints 3
2.6 User Documentation 4
2.7 Assumptions and Dependencies 4
3. System Features 4
3.1 System Feature 1 4
3.2 System Feature 2 (and so on) 5
4. External Interface Requirements 5
4.1 User Interfaces 5
4.2 Hardware Interfaces 6
4.3 Software Interfaces 6
4.4 Communications Interfaces 6
5. Other Nonfunctional Requirements 7
5.1 Performance Requirements 7
5.2 Safety Requirements 7
5.3 Security Requirements 7
5.4 Software Quality Attributes 8
6. Other Requirements 9
Appendix A: Glossary 9
Appendix B: Analysis Models 10
Appendix C: Issues List 10

Revision History

Name	Date	Reason For Changes	Version

1. Introduction (Note: when acronym is chosen, replace GCLM with the adopted acronym)

1.1 Purpose

This Software Requirements Specification (SRS) documents the set of features and functions for the Enterprise System for Grants, Contracts and Loans Management.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

This document is intended for OFM system development and support staff, including developers, project managers, product managers, testers, trainers, documentation writers, infrastructure staff, database architects, architecture support staff and management.

It is intended for customers in the User Group who are assisting in the development, review, validation and prioritization of requirements.

It is intended for persons interested in the functions and capabilities of GCLM.

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Project Scope

Refer to the Charter for the Enterprise System for Grants, Contracts and Loan Management.

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

GCLM will replace the Washington State Department of Ecology's (ECY) aged Contracts & Grants Management System and serve as a mission-critical system for the Department of Community, Trade and Economic Development (CTED). The Office of Financial Management (OFM) will lead the effort, joined by the ECY and CTED. It will position itself to support the vision coming out of OFM's Roadmap project.

For Grants Management, GCLM will support the process of applying for grants, evaluating and awarding grants, daily grants project management, payments, closures and reporting/queries. For Contracts Management, it will support the process of documenting and establishing contracts, daily contracts management, payments, closures and reporting/queries. For Loans Management, it will support only the accounts payable process for loans.

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Features

<Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or a class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.>

2.4 Operating Environment

OE-1: The system must be compatible with OFM standard Intel based hardware with Microsoft Windows 2003 and IIS 6.0.

OE-2: The system must utilize OFM standard Microsoft SQL 2000/2005 for all database functionality.

OE-3: The system must have a browser based thin client user interface for all system users. The system should not require any system vendor supplied software to be loaded onto a user's workstation prior to use.

OE-4: For an OFM developed and/or maintained system, the system should utilize the standard reporting, ad-hoc reporting, and data query features delivered by the Enterprise Reporting group for ad-hoc reporting requirements not provided by the system.

OE-5: The proposed solution must be scalable, with simplicity of scaling options for all aspects of hardware, software, site management services, connectivity, and the number of concurrent users.

OE-6: The system must allow access from standard pc hardware across the statewide intergovernmental network (IGN) and through the DIS Fortress server.

OE-7: The client portion of the system must run on a Windows based pc with Internet Explorer 6.0.

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

CO-1: The system design, code and maintenance documentation shall conform to the *OFM Application Technology Architecture – Application Standards - .NET Application Standards*. (http://ofm004/ata/standards/standards.htm)

CO-2: The system must utilize OFM standard Microsoft SQL 2000/2005 for all database functionality.

CO-3: For an OFM developed and/or maintained system the system screen displays shall utilize the *OFM WebForms Application Framework*. **Placeholder – document is still in draft form** (\\ofmsws18\Projects\Webforms_framework\ProjectDocuments\Requirements\SRS for Web Forms Framework.doc)

CO-4: For an OFM developed and/or maintained system the OFM standard development language is Microsoft C#. To allow for additional development or modification in the future, the system application must be in this language and supported by Microsoft Visual Studio.

CO-5: All external interfaces will be based on real-time messaging with guaranteed delivery or via file import/export.

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 System Documentation

SD-1: There must be clear and comprehensive documentation of the solution to include: installation documentation, system documentation including component design and ddata design and vendor support for system problems and issues.

SD-2: The system shall provide comprehensive operational documentation including, but not limited to, online help and user guide. User documentation should clearly describe the procedures that will maintain the operational quality of the system.

SD-3: There must be clear and comprehensive installation documentation that allows OFM to determine the impact of the installation.

SD-4: There must be clear and comprehensive maintenance and support documentation that allows OFM to determine the impact of implementation AND ongoing maintenance and support.

SD-5: There must be clear and comprehensive system training documentation.

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

3.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

3.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings,

such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

3.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

2.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:
REQ-2:

3.2 System Feature 2 (and so on)

4. External Interface Requirements

4.1 User Interfaces

UI-1: For an OFM developed and/or maintained system the system screen displays shall utilize the *OFM WebForms Application Framework*. (\\ofmsws18\Projects\Webforms_framework\ProjectDocuments\Requirements\SRS for Web Forms Framework.doc)

UI-2: The system shall provide context sensitive help.

UI-3: The system must allow a user to login to the system using standard OFM authentication methods.

UI-4: The system must provide the user with a clear method of exiting the system (e.g. a "logout" button).

UI-5: The system must be built to facilitate accessibility for individuals with disabilities based on OFM standards (See Appendix ???). This requirement includes web-based applications, software systems, and electronically published documents. These technologies should provide the same functionality to individuals with disabilities as it provides to individuals without disabilities. The accessibility standards are based on section 508 of the Rehabilitation Act of 1973, W3C XHTML 1.0, CSS, and WCAG 1.0 standards.

http://www.access-board.gov/sec508/standards.htm

http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/
http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/full-checklist
http://isb.wa.gov/tools /webguide/accessibility.aspx

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

4.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

4.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

4.4 Communications Interfaces

CI-1: The system must be capable of sending an e-mail message to the users involved in the workflow, notifying them of any approval and/or payment status changes.

CI-2: The system must be capable of assigning and sending a new password to a user upon a user's request.

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

5. Other Nonfunctional Requirements

5.1 Performance Requirements

PE-1: The system shall be available xx.x% of the time for use from 7:00 a.m. to 7:00 p.m. seven days a week.

PE-2: All web pages generated by the system shall be fully downloadable in no more than xx seconds over a 50KBps modem connection.

PE-3: No system function shall time out.

PE-4: Responses to queries shall take no longer than xx seconds to load onto the screen after the user submits the query.

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

SE-1: The system must protect data from wrongful access. This includes protection of data throughout its entire lifecycle including when at rest, when transmitted across networks, and when being processed. Data exchanged between client software and host software must be managed in a secure way by the GCLM application. Confidential data should never be in clear text.

SE-2: User security must include the ability to control access to confidential data based on users' authorization rights.

SE-3: Include traceability information: who did what, when, and using what computer.

SE-5: Clearly warn users against putting confidential information into the system (OFM to draft warning).

SE-6: Include and enforce user permissions and restrictions using a role-based approach.

SE-7: Provide the ability to set up the following roles in the system:
- Need to insert roles here

SE-8: The system must provide application level user authentication and authorization tools. These tools will be used to limit access to authorized users only. Active Directory?

SE-9: The system should be password protected. The system must be able to enforce the State's strong password guidelines as well as State password expiration. Password expiration time span must be configurable and also provide for a system maximum default.

SE-10: The system should provide workflow/routing such that rules can be established. Based on those rules the workflow engine would determine the next step for routing. The routing needs to be flexible enough to be overridden while in process to allow for user-initiated exceptions.

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

Availability-1: The system shall be available xx.x% of the time for use from 7:00 a.m. to 7:00 p.m. seven days a week.

Availability-2: There will be a maintenance window on the last Thursday of each month from 5:00 a.m. to 7:00 a.m. the following morning for server security patch installation.

Availability-3: The system will be accessible via the state intranet or the internet through the Fortress.

Conversion-1: The data structures of the solution must allow and provide information on conversion of existing data. Specific requirements of conversion have not been determined.

Flexibility-1: In order to meet the challenge of changing business rules, wherever possible rules that are likely to change with any frequency should be externalized so that changes can be made without recompiling and redeploying the system software.

Interoperability-1: The system must be able to import??

Interoperability-2: The system must be able to produce a batch transaction file for submission to one or more external accounting systems.

Maintainability-1: An OFM developed/maintained solution shall meet OFM architecture principles and current OFM application development standards such as: .NET, C#, SQL Server, Active Directory Authentication, Crystal Enterprise reporting. All code will be commented using xml comment tags.

Maintainability-2: A purchased off-the-shelf product will be evaluated based partly on its architecture and possible impacts that architecture will have on OFM's ability to support and maintain the new system.

Maintainability-3: Either a purchased or OFM developed system shall provide the following:

- Central administration of data.
- A layered architecture with clear logical boundaries.
- Message-based and loosely coupled interfaces.
- Event-driven transactions
- Cohesive components that support a small set of functions for ease of testing.
- Central administration of business rules.

Portability-1: This application must be able to be deployed in a Windows 2000/2003 server farm running IIS 5.0/6.0 using SQL Server 2000/2005 as its backend database server.

Reliability-1: The system must be highly reliable since we are dealing with payments.

Reliability-2: There must be safeguards such that if a batch of transactions does not complete, there must be a method for resubmission of the transaction.

Reliability-3: The system should run in a web farm of redundant servers so that capacity can be added if the system becomes overtaxed.

Reusability-1: An OFM developed solution shall be designed with reusability in mind. During the design process, common system features will be created in such a way that they can be reused by other systems. Some possible examples of this are common authentication and workflow/routing.

Robustness-1: The system must provide meaningful error messages to users when faced with invalid user input.

Robustness-2: The system must fail gracefully if connections to the backend databases are terminated.

Robustness-3: The data access should be transactional so that when errors occur a rollback of partially completed transactions is possible.

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>