Conflict of Interest System

MTM Program Product Software Requirements Specification

Version 0.1
2/18/2022

Template Version 3.5

Standard Version Number: 3.5
Standard Version Date: March 10, 2018

Version History

Version	Date	Authors	Comment
0.1	2/18/2022	Nathan Blakenship, Matthew Gallagher, Tucker Kane, Brandon Mitchell	Students of ESOF 328, Spring 2022

Template Version History

Version	Date	Authors	Comment
3.0	7/21/2012	Frank Ackerman	Initiating standards versions
3.1	8/2/2012	Frank Ackerman	Some non-functional requirements definitions .Added Adaptability, Enhanceability, and Portability
3.2	1/17/2013	Frank Ackerman	Added usability comment
3.3	3/6/2013	Frank Ackerman	Added a bit more explanatory text and final section 8.
3.5	3/10/2018	Celia Schahczenski	Changed format of dates, rearranged, renamed items, removed Illustrative Use Cases, increased some explanations, added appendices including data and report sections.

Montana Tech Software Engineering Students:

These Montana Tech Method software engineering standards encapsulate Dr. Ackerman's decades of experience in the software industry, the IEEE software engineering standards, and many suggestions from various texts. They have gone through many revisions and additions over the last several years. They are part of your software engineering studies so that (1) you may have the experience of developing software to a standard (which you may find you need to do if you take a job that requires high reliability software), and so that (2) you will have the experience of developing high quality software. You are also invited to participate in the continuing evolution of these standards by studying them critically and making suggestions for their improvement and correction.

TABLE CONTENTS

Questions/comments:
Q: Who exactly are the people who need to sign the COI form?

A: ??? Good question!

Q: Does Montana Techs Website act as the sole security interface? If so is there a better name for the authentication system

A: ??? When I go into the COI portion I see the label "Information Technology – Single Sign-On.

1 Introduction

This section provides an overview of the *Conflict of Interest (COI) System*, the purpose of this document, and definitions, acronyms and references related to conflicts of interest.

1.1   Software Purpose and Scope

The purpose of the *Conflict of Interest System* (*COI) System* is to help the Research Office handle conflicts of interest for *Montana Technological University*. For *Montana Tech* affiliated members who fall under the COI policy, the *COI System* is an interactive user interface, storage, and reporting system that will reduce user and staff input to less than 40 hours per year, guarantee a 95% or better completion rate for all faculty and staff, and generate annual reports. Unlike the previous systems used by the research office, this system accurately collects statistics, chooses correct recipients, saves time, educates users on the COI policy, and generates reports.

1.2   Document Purpose and Contents

The purpose of this Software Requirements Specification (SRS) is to give readers an understanding of Montana Tech's goals and needs for a *COI System*. It provides a guide for future developers on the desired features, functionality and behaviors of the *COI System.* This document can be used to design tests to ensure an implementation behaves as intended.

Customers sometimes find sample interfaces easier to understand than documents such as this SRS. Sample user interfaces demonstrate one way that the software could appear. This document goes further to tell precisely what functionality is needed.

This document does not attempt to tell how this software should be implemented except in those cases where the customers want the application to be developed in a particular way. Deciding exactly what a system should do, before deciding how it will do it, reduces development time considerably.

This SRS was developed by students in Software Requirement and Specification (ESOF 328), in the Spring 2022 at *Montana Technological University*. It has been developed in part by faculty members, administrative personal and students. Thanks goes to Angela Lueking, Jill Yoder, Muhammad Abdul Basit UR Rahim, Trevor Osborne, and Ryan Hessler. The main audience of this document are the clients as well as the developers that will use this document to implement the system.

1.3   Definitions, Acronyms, Abbreviations and References

This section defines technical terms used in this document, as well as the expansions of acronyms and abbreviations, and important references.

1.3.1   Definitions

According to the Conflict of Interest and Financial Disclosure policy, a conflict of interest occurs " … when there is a potential divergence between the employee's private interests and professional obligations to Montana Tech, such that an independent observer might reasonably question whether the employee's professional actions or decisions could be influenced by considerations of personal gain (financial or otherwise)."

Status of COI forms - not started, in progress, completed <mark>– This will probably get moved elsewhere in the document.</mark>

1.3.2   Acronyms and Abbreviations

COI	Conflict of Interest
SRS	Software Requirements Specification

1.3.3   References

Conflict of Interest and Financial Disclosure Policy (1998), *Montana Technological University,*
https://www.mtech.edu/research/files/conflict-interest-financial-disclosure.pdf

Conflict of Interest Disclosure Statement and Certification Portal, *Montana Technological University,*
https://mtlbsso.mtech.edu/idp/profile/cas/login?execution=e1s1

Ellucian Banner Human Resources, <mark>----Is this the name we should be using?</mark>
https://www.ellucian.com/solutions/ellucian-banner-human-resources

2 General Factors

A high-level overview of what the *COI System* will do, its running environment, who will use it, its dependencies, and assumptions made about it are included in this section.

2.1 Product Perspective

This system is meant to serve as a replacement to the current COI process. In addition, this project will be dependent on the *Ellucian Banner Human Resources* system.

2.2 Product Functions

The *COI System* will:
- Educate: The system will educate users on what is a conflict of interest, the importance of disclosing conflicts, and the penalties for not properly disclosing information on the form. <mark>--- I didn't hear anything about penalties, should we include?</mark>
- Sign: The system will enable faculty, staff and students to complete, sign, and submit COI forms online, approving the document with their digital signature.
- Manage forms: The system will aid *Research Office* personnel to easily see the status of all COI forms.
- Manage conflicts: The system will aid *Research Office* personnel to easily see who has potential conflicts, the status of those conflicts and to record how those conflicts will be managed.
- Report: The system will generate yearly reports on the data gathered from the signed COI forms to submit to *Montana Technological University's Board of Regents.*

Error! Reference source not found.shows the context in which the *COI System* will operate.
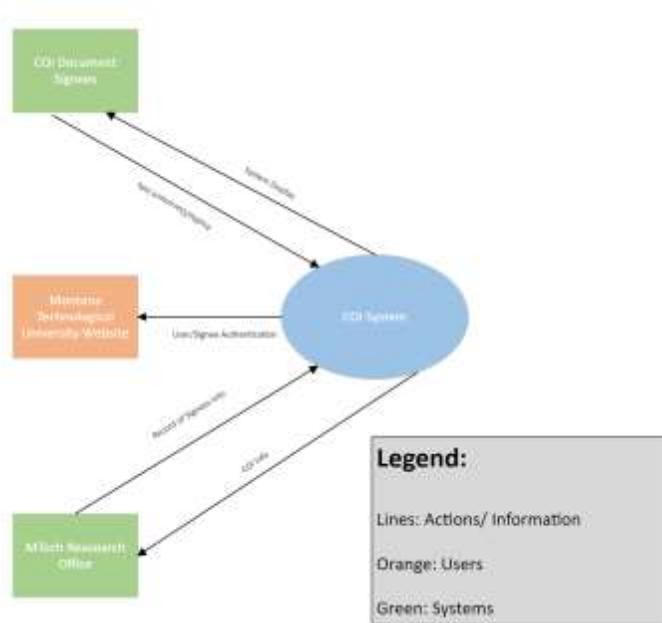
Figure 1 Context Diagram

2.3   Environmental Conditions

The Montana Board of Regents requires the submission of annual conflict of interest reports from each of its universities, including *Montana Technological University*. Banner holds the information necessary to uniquely identify employees, staff and students. Microsoft Outlook is the most likely tool to be used to distribute COI forms to faculty, since each member of faculty, and every student, has their own unique email address. **Error! Reference source not found.**
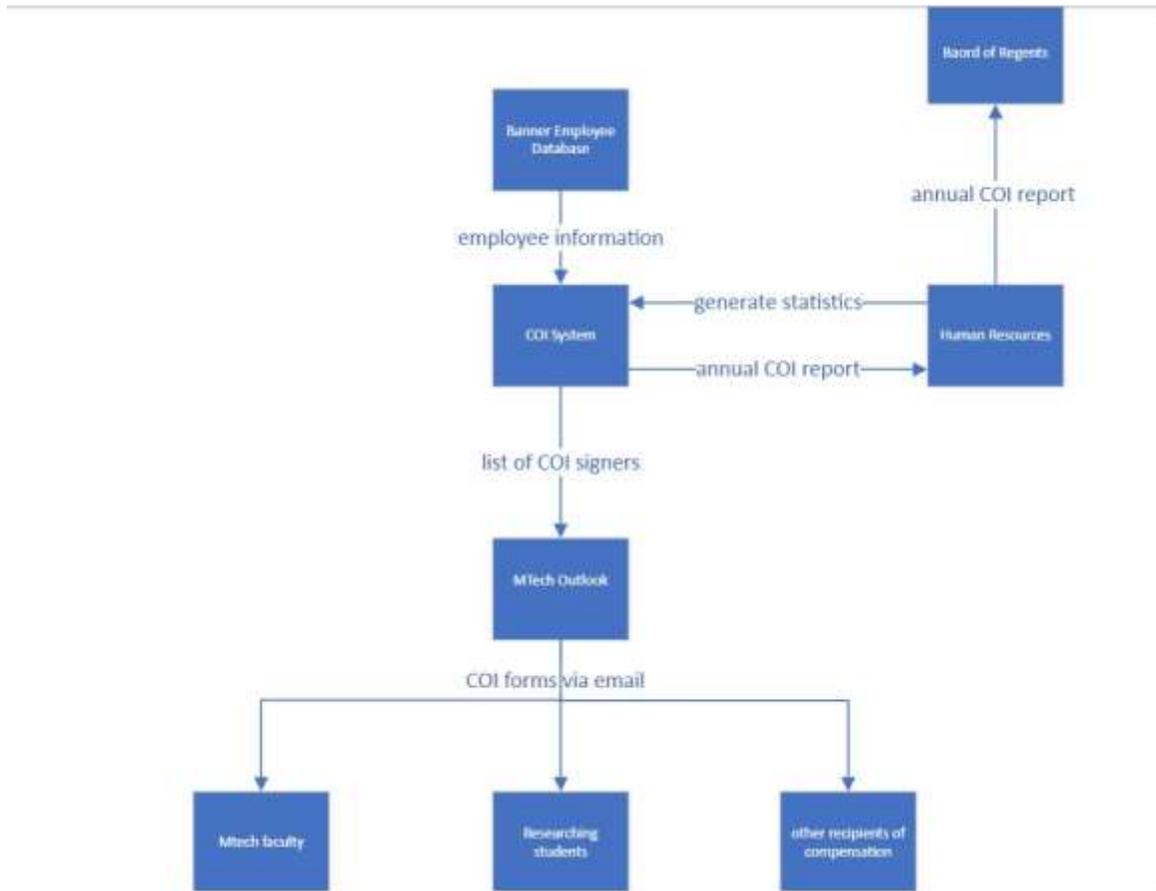
Figure 2 Ecosystem Map

2.4   User Characteristic

The primary users of this system will be full or part time faculty, staff and selected students of *Montana Technological University* and *Research Office* personnel. Faculty, staff and student employees will use the system to complete, sign, and submit COI forms. *Research Department* personnel will use the system to see the status of forms, keep track of conflicts, and generate yearly reports for the *Board of Regents*.

2.5   Dependencies

The *COI System* is dependent on the *Central Authentication System* of Montana Technological University --- Is part of the COI System a "portal" (see the email) and does that portal need to use CAS?

The system will be dependent on *Ellucian Banner Human Resources* for the legal names of employees. --- Does it also get student information from Banner.

2.6   Assumptions

Some assumptions for this system are:
- Signers are "anyone who interacts with outside sources through Montana Tech". This includes anyone on payroll through the school, as well as some students in research/clubs.
- All actual conflicts of interest will be managed by the *Research Office*. This product will not manage any conflicts, only sort them and generate some statistics.
- This system will be available on any computer with a modern web browser and Internet connectivity.

3   Use Cases

[Use cases describe possible interactions between an actor and a system that results in an outcome that provides value to the actor. Develop these use cases with the client. This section may begin with a use case diagram, an analysis model that identifies the actors who can interact with a system, along with the various use cases with which each actor might be involved.

If no use cases exist for this product, this section should read "Use Cases were not developed for this specification".]

3.1   Actor

[This subsection lists the various actors that will interact with the proposed system, along with the interactions that these actors may perform. An actor is a person, or other entity external to the software system, who may interacts with the proposed system to accomplish tasks. Actors may represent roles, identified from the customer community that will use the product.

The following is a template for documenting user classes and the associated use cases.]

Primary Actor	Use Cases
Actor 1	1. *Use case 1* 2. *Use case 2* *...*
Actor 2	3. *Use case 1* 4. *Use case 2* *...*
...	*...*

3.2   Use Cases

[This subsection contains use cases of the proposed system. Use cases can be "casual" or "fully dressed". The template given is for a fully dressed use case. Omit portions of this template for casual use cases.]

3.2.1   [Use Case Name]

Created By:	*Name 1*	*Last Updated By:*	*Name 2*
Date Created:	*Month dd, yyyy*	*Date Last Updated:*	*Month dd, yyyy*
Actors:	*User class name*		
Description:	*Description of use case*		
Preconditions:	*1. Precondition 1 or "none"* *2. ...*		
Postconditions:	*1. Postcondition 1 or "none"* *2. ...*		
Normal Flow:	*1.0 Description phrase* *1. Step 1.* *2. Step 2.* *3. ...*		
Alternative Flows:	*1.1 Description phrase for alternative flow (branch after step n)* *1. Step 1.* *2. Step 2.* *...* *n. Return to Step m.* *1.2 Description phrase for 2nd alternative flow (branch after step l)* *1. ...*		
Exceptions:	*1.0.E.1 Description phrase for exception (at step k)* *1. Step 1.* *2. Step 2.* *...* *1.0.E.2 Description phrase for 2nd exception (at step j)* *1. ...*		
Includes/Extends:	*Name of included use cases, name of the use cases that this use case extends, or "none"*		
Priority:	*Low, medium or critical*		
Frequency of Use:	*How often it is expected that this use case will be performed.*		
Business Rules:	*Business rules can be listed, or another document can be referenced, or "none"*		

Special Requirements:	*1. Special requirements 1 concerning this use case.* *2. ...* *or "none"*
Assumptions:	*1. Assumption 1 concerning this use case.* *2. ...* *or "none"*
Notes and Issues:	*1. Note 1 concerning this use case.* *2. ...* *or "none"* *This can serve as a placeholder for extraneous information.*

3.2.2   [Use Case Name 2]

...

4   Specific Requirements

The following section contains all of the requirements for the COI System. The details within this section are defined as individual, specific requirements. Each requirement is tagged with a priority to indicate its importance. In order from least importance to most importance, the possible priority levels are: low, medium, high, and critical. Each requirement is clearly identified for tracking.

4.1   Functional Requirements

[This subsection should specify how the software product will react to every possible input situation. It describes all the actions that must take place in the software in response to every input. Pertinent changes in the environment are considered to be inputs.

Care must be taken to avoid dropping into design details. In the user cannot directly experience the effect of a requirement it probably crossed the line into design.

Functional requirements should be logically grouped. Each group should have a short, unique (within the SRS) abbreviation and a number. The word processing section number will probably change as the SRS is developed.

For each identified requirement an optional rationale for that requirement may be given.

Most modern software should provide at least a modicum of user help. For very complex applications in situ help may be supplemented by a user's manual (or manual page) but for many simple applications comprehensive in situ help is sufficient.]

4.2   Quality Attributes

[This subsection specifies criteria used to judge the operation of a system, rather than specific behaviors of the system. Specify the specific behavior of the system in the functional requirements.]

4.2.1   Availability

4.2.2   Human Factors

[Not everyone has the same inherent mental and physical capabilities vis-à-vis a given computer application. For example if sound is part of the application, will other clues be given that will enable a hard of hearing user to use the proposed application as well as person with normal hearing; similarly for color blindness. Define these factors, if necessary, with validation criteria.]

4.2.3   Usability

4.2.4   Performance

4.2.5   Security

4.2.6   Reliability

[Reliability is specified as mean-time-to failure of an operational item. An operational profile must be specified.]

4.2.7   Maintainability

4.2.8   Enhanceability/Extendibility

[If the future it might be necessary to change the Functional requirements in specified ways, what is the maximum estimated effort required to make such changes and what is the rationale for this estimate?]

4.2.9   Portability

[If in the future it might be necessary to change the above Development or Delivery Environments (DV or DL) to other specified environments, what is the maximum estimated effort required to implement such changes and what is the rationale for this estimate]

4.2.10 V&V Activities

4.2.11 Adaptability

[If it is specified that in the future it might be necessary to change any of the above Non-Functional requirements, what is the maximum estimated effort required to implement such changes and what is the rationale for this estimate.]

4.3   Non-Functional Requirements Which Are Not Quality Attributes

[This subsection specifies non-functional criteria such as platform, deployment, interface, design and document requirements. If there is not a document describing project requirements, those requirements (cost, schedule, etc.) can be placed here.]

4.3.1   External Interface Requirements

4.3.1.1   Hardware

4.3.1.2   Software

4.3.1.3   Communications

4.3.2   Development Environment

4.3.3   Delivery Environment

4.3.3.1   Site

[This subsection should specify any requirements for installation or operation of the software that might change the pre-existing configuration of the user site.]

4.3.3.2   Operations

[This subsection should specify normal and special operations required by the user to include:
- *Various modes of operation within the user organization*
- *Periods of interactive operations and unattended operations*
- *Data processing support functions*
- *Backup and recovery operation.]*

4.3.4   Design Constraints

[Sometimes a client will require certain design constraints, for example the use of a certain system configuration or the use of particular algorithm. Such constraints are described in this subsection.]

4.3.5   Database

[This optional subsection specifies requirements for any database to be developed as part of the product. The information in this section may include:

- *Types of information to be stored*
- *Table attributes (queried, supporting, updated)*
- *Frequency of access*
- *Accessing capabilities and requirements*
- *Data elements and file descriptors*
- *Retention requirements for data.]*

Take care to avoid design details. Unless so requested by the client, this section should only contain as much information about saved data as is necessary to fully document any of the requirements given above.]

4.3.6   Deliverable Items, Dates and Conditions

4.3.7   Cost

4.3.8   Standards

5   Future Enhancements

[This section should describe any future enhancements that are contemplated at the time this SRS completed. If there is no known possibility that this product will be enhanced in the future this section should read : It is not expected that there will be any future enhancements to this product.*]*

Appendices

[In some cases, it is helpful to move items out of the main portion of the Software Requirements and Specification Document. These items can appear here. Alternatively, move these items into the main part of the document.]

Appendix A: Definitions, Acronyms, and Abbreviations

[This appendix should provide the definitions of all terms, acronyms, and abbreviations required to fully understand your SRS.]

Definitions

software failure	a failure will be attributed to this software product whenever one of the delivered work products does not meet the requirements specified in this SRS, or does not meet ordinary and reasonable customer/user expectations.

Acronyms and Abbreviations

DB	Database
HW	Hardware
SDD	Software Design Description
SRS	Software Requirements Specification
SW	Software

Appendix B: Analysis Models

[Optionally, include any pertinent analysis models, such as activity diagrams, state-transition diagrams, entity-relationship diagrams, or a formal specification.]

Appendix C: Data Dictionary

[The data dictionary defines the composition of data structures and the meaning, data type, length, format, and allowed values for the data elements that make up those structures. In many cases, storing the data dictionary as a separate artifact, rather than embedding it in an SRS is beneficial. This also increases its reusability potential in other projects.

List data items alphabetically. Make each name a bookmark so each time the name occurs in this SRS it can be link to this entry via a hyperlink. Choose names with care. The expectation is that these names will persist in the design and implementation.]

Data Element	Description	Composition or Data type	Length	Values
Name of data item being defined	*Textual description of the business meaning of the data element*	*For primitive data elements: data type (integer, floating point, alphabetic, date, etc.) and, as appropriate, format (e.g. date as MM/DD/YYYY).* *For data structures show the components that comprise the structure. ,*	*Maximum number of characters for primitives; blank for structures*	*List of allowed values, default, rules governing legal values, and any other description of the data values*
...	...	..	...	...

Appendix D: Report Specification

[This optional appendix contains descriptions of reports that the system needs to generate. Many applications involve generating reports from one or more databases, files or other information sources. Exploring the content and format of the reports needed is an important aspect of requirements develop. Describe the contents and layouts of each report, including changes being made in an existing version of the report. Indicate the conditions that will trigger generating the report (e.g., manual or automatic) the timing of report generation, and the disposition of the report, such as to whom it is sent or where it is stored.]

Use the following template to document business rules.

Report ID:	
Report Title:	
Report Purpose:	
Data Sources:	
Frequency and Disposition:	
Latency:	
Visual Layout:	

Header and Footer:	
Report Body:	
End-of-Report Indicator:	
Interactivity:	
Security Access Restrictions:	

If appropriate, provide a mock-up or a sample of the report, or an illustration of a similar existing report, showing the desired layout.]

Appendix E: Business Rules

Several business rules we have identified are:
- Faculty must use their legal name as defined in *Banner* when signing forms.
- A form is considered complete when both the faculty member and an administrator in the Office of Sponsored Programs have signed.
- Students are considered full-time if they work twenty or more hours a week.
- According to the Conflict of Interest and Financial Disclosure policy:

   Compliance with this policy is required by all full-time and part-time Montana Tech employees, including students who receive compensation from Montana Tech and students or others who design, conduct, or report research, educational, or public service activities of Montana Tech.

ID	Rule Definition	Type of Rule	Static or Dynamic	Source
BR-1	Every Conflict form must be signed by the Vice Chancellor of Research and the person filling out the form	Fact	Static	COI Policy, 1998
...	...	..	...	...

Appendix F: Sample User Interface

 [If a sample user interface exists, place it here. Make it clear that this user interface is only an example. If something is required in the user interface, state that earlier in this document.]

Appendix G: Issues

[This optional appendix is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending.]