Family Thread

Software Requirements Specification (SRS) Template v1.0

[Instructions Specific to the course SRS including formatting and any notable notation.]

This document is an annotated outline intended for specifying software requirements and is adapted from IEEE 29148-2018.

[Version #1.7]
Prepared By: [Team B]
Prepared For [CSC380 at SUNY Oswego]
[05/01/2024]

Table of Contents

Change Log

Name	Date	Reason For Changes	Version
Saurav Lamichhane	02/01/24	Worked on - Cover Page - Introduction - Scope - Grammatical Review	1.0
Crislenny Uceta	02/01/24	1. Added Moore’s statement to 1.1 2. Added placeholders for product perspectives 3. Worked on Scope 4. Added “To be determined” to sessions 1.3.1.1 - 1.3.1.8 since we don’t know yet 5. Worked on 1.3.2, 1.3.3, and 1.3.4	1.0
Crislenny Uceta	02/02/24	1. Updated 1.1	1.0

Saurav Lamichhane	02/02/24	Removed Fluff Remove Introduction (Redundant) Organized Page, made fonts + font size consistent Added on to the questions on 1.3.3	1.0
Ian Venton	02/02/24	Updated: 1.3.1.1 1.3.1.2	1.0
Phone Pyae Sone Phyo	02/02/24	Updated/Modification: 1.2 Modification to scope (included/excluded)	1.1
Phone Pyae Sone Phyo	02/09/24	1.4 Definitions 1.3.2 Product Functions	1.1
Crislenny Uceta	02/09/24	1.3.1.1 System Interfaces 1.3.1.2 User Interfaces	1.1
Saurav Lamichhane	02/09/24	1.3.1.2: Organized the Diagrams and added Google Drive links Updated Cover Page (version # and date)	1.1
Crislenny Uceta	02/16/24	Added Class UML Diagram	1.2
Saurav Lamichhane	02/19/24	Updated - UpdateLog (added Cris for 2/16) - Version #	1.3
Crislenny Uceta	02/23/24	1.3.1.2 Added View Trees, Edit Tree, Edit/Add Person activity diagrams. And placed Use Case diagram and Class diagram in the section	1.4
Saurav Lamichhane	02/23/24	1.3.1.2: Added Activity Diagrams - CreateTrees - Login	1.4

		- MyAccount - ShareTree - ShareTreePermissions	
Saurav Lamichhane	03/01/24	3.4: Interface Requirement - Added what the Family Thread can import and export 3.5: Database Requirements - Added what's stored in the DB - Added DB to abbreviations list 1.3.1.3: Hardware Interfaces Added Gnatt Chart	1.5
Saurav Lamichhane	3/08/24	Updated cover page - Added UI Mockups	1.6
Saurav Lamichhane	5/1/24	Final changes - 1.3.1.6 Memory Constraints - 1.3.1.7 Site Adaptation - 1.3.1.8 Interfaces with Services - 3.2 Performance Requirements - 3.6 Design Constraints	1.7

1. Introduction

[This section of the document is intended to give a wide overview of the project, not identify specific solutions. If a section does not apply, ***do not remove it from the document,*** *instead, write a brief note on why the section does not apply to the SUD.]*

1.1 Purpose

For professional genealogists or genealogy enthusiasts who might be interested in the visual representation of their lineage. The Family Thread is a system software that family members can use to create a family tree. Unlike Family Echo or Ancestry.com, our product allows users to add custom portraits, and work with other collaborators to create a more accurate representation.

1.2 Scope

[Identify the scope of the SRS document, what will be covered, what will be excluded? Additionally, identify the general scope of the system you are building. Keep these brief, use bullet points to help organize the information.]

Included	*Excluded*
● Portrait ○ Needs a certain level of permission to see the portraits ● Description ○ Name ○ Date Of Birth / Death ○ Place of Birth / Death ○ Biological Sex ○ Gender Transition / Name Change ○ God family ● Hold a large amount of data for families ○ Ability to zoom into a nuclear family ● Infer relationships ● Recognize duplicate data ● Collaborations ○ View Only ○ Editors ○ Owner ● User Sign-in (Google) ● Legend ○ Describe how relationships are drawn ○ Dotted - Not blood-related Drawn - Blood-related/Married ● Big Breaks in Family Trees ○ It’s going to span for a long period of time, so no tall information is available ● Tabs for “account”, “family trees”, “collaborators” ● Saving Mechanism ○ HTML	● Fancy illustrations that complicates the actual family tree structure ● Printing Mechanism ○ No export format(pdf/csv) other than html ●

1.3 Product Overview

1.3.1 Product Perspective

[Identify the product's relationships with other products already in existence or currently in development. If this product is part of a larger system, identify that here and notate how the system will interact with external actors.}

1.3.1.1 System Interfaces

[List the system interfaces and identify the functionality of that system. How will it be used in conjunction with the system you are developing?]
TBD.

1.3.1.2 User Interfaces

[List the characteristics of each interface between the product and its users. Include the modes of operation in the user organization, data processing support functions, and any backup or data recovery operations present in the system.]

You can access the images below at higher quality on Google Drive

1.3.1.3 Hardware Interfaces

[Specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics. It also covers such matters as what devices are to be supported, how they are to be supported, and any relevant protocols.]

Personal Computer: It is designed on a web browser so it would be accessible for computers using the internet.
Cellphones: Editing the trees will not be optimized for phones. Viewing trees will be supported.

1.3.1.4 Software Interfaces

[List the name, mnemonic, specification number, version number, and source of each software product, be sure to include the purpose of each software product you are integrating and how it will be incorporated into the SUD.]

None

1.3.1.5 Communication Interfaces

[Identify the various networking protocols and other such communication interfaces. Unique and non-standard protocols should be documented here, standard protocols should be referenced and cited here and in the reference document section]

We have an email service that allows us to connect to the wireless network and send emails to the user.

This service uses Spring Mail for Springboot to manage all network connections.

1.3.1.6 Memory Constraints

[List any limitations on primary and secondary memory, as well as any applicable characteristics.]

For MongoDB, there are several constraints:
Documents cannot be larger than 16 mb.
The total size of all documents cannot be over 64 terabytes. For our project, we do not need to worry about this, but if it were to release to the public, we would need to implement an alternative.

1.3.1.7 Site Adaptation

[Identify any situations, initializations, or protocols that are specific to a give site, mission, or operational model. If you are tailoring a product to meet specific client needs that differ from normal operation, include it in this section.]

None

1.3.1.8 Interfaces with Services

[List any interaction with any SAAS (Software as a Service) or cloud based services.]

None.

1.3.2 Product Functions

[This section of the document is intended to describe all of the major functions the product will be able to perform. Functions should be organized in such a way that the list of functions is understandable to anyone who is reading through the document for the first time.]

- Users will be able to CRUD with other users to build up the family tree.
- Users(owners) will be able to share the family tree with anyone associated with a gmail account
- Users(owners) are able to make the family tree to be either public or private.
- Using a mouse/trackpad to move around in the family tree visual
    - Zooming in and out
- Creating and editing family trees
    - Clicking on an individual's nodes to display the details for that particular person.

1.3.3 User characteristics

[Describe the characteristics of the product's intended group of users. Include things such as technical expertise, and any information that may impact usability or accessibility to the product.]

The main target audience are genealogists or enthusiasts who share an interest in documenting family trees.

The intended user must have some proficiency in computer systems.

- Some proficiency indicates the capability to use web applications and the ability to learn how to use new tools.

1.3.4 Limitations

[Identify any limitations that will impact the development. Include things such as hardware limitations, safety and security considerations, quality requirements, regulatory requirements and/or policies, etc.]

- Storage
- Costs (Family Tree Library Access)

1.4 Definitions

[Include any terms and definitions needed to understand the SRS or the System Under Development (SUD), terms placed here should also be placed in the Appendix. Terms should use the full name and the general definition of that term, any abbreviations that will be used in the document and the source should be placed in Appendix 5.2]

Term	Definition
Private Tree	A family tree in which the tree is accessible only to the users with owner, editor, and viewer role.
Public Tree	A family tree in which the tree is viewable by anyone with a link. (Note: A public tree is bound to be shown on the home page of website)
System Under Development	The system which is actively being developed.

2. References

[Include citations to external sources and resources in this section. References to other internal documents can be placed here but should also be referenced in the appendix.
Example: ISO/IEC/IEEE 29148.2018, Systems and software engineering
Life cycle processes — Requirements engineering]

3. Requirements

[This section should contain all the software requirements at a level of detail sufficient enough to enable designers to design a system, and for testers to test that system, in a way that satisfies the requirements. Each requirement should be perceivable by users, operators, or other external systems. At minimum, the description should include the inputs and outputs of the system, and all functions performed by the system in response to an input or in support of an output. Specific requirements should include the following characteristics:

- *Correct*
- *Unambiguous*
- *Complete*
- *Consistent*
- *Verifiable*
- *Modifiable*
- *Traceable*
- *Ranked for importance and/or stability*
- *uniquely identifiable (usually by numbering)*
- *organized in a way that allows for maximum readability*

The purpose of the requirement is not to dictate design, but rather to guide designers to make the safest, most correct version of the system possible. Do not attempt to build solutions to your written requirements]

3.1 Functions

[Define the fundamental actions that the system must stake in order to accept inputs and generate outputs. It may make sense to organize or partition the functional requirements into sub-functions or sub-processes, do not expect development to mimic this organization. Functional requirements are typically identified using "shall" statements, and some functions that should be included in this section are:

- *Validity checks on the inputs*
  - *"the system shall check the validity of the inputs it receives."*
- *Exact sequence of operations*

    - *"The system shall receive the input, then it shall process the input, and then it shall generate the output."*
- *Responses to abnormal situations including:*
    - *Overflow*
        - *"If the system experiences data overflow, it shall temporarily stop accepting new inputs."*
    - *Communication facilities*
    - *Error Handling and Recovery*
- *Effect of Parameters*
- *Relationships of outputs to inputs*
    - *I/O sequences*
    - *Any formulas used for I/O conversion*

3.2 Performance Requirements

[In measurable terms, specify the numerical requirements of the system. Include static performance requirements such as the number of terminals, simultaneous users, etc. As well as dynamic performance requirements such as the number of tasks able to be completed in a set period of time.]

Number of Simultaneous Users:
Optimal: Dependent on Moxie Capability.
In theory, it can handle as much as Moxie can hold.
Reality: We do not expect more than 20 people to use the website at once.
Additionally, not a lot of people will be visiting the same tree simultaneously.

Number Of Tasks: [Tree Creation, Node Creation]
Optimal: As much as possible
Reality: 1 task per second
We limit the user's ability to do multiple tasks because the family tree nodes are dependent on each other. They could open up multiple trees and edit them, but no people are not going to be able to edit more than a few nodes per second.

3.3 Usability Requirements

[Define usability and quality requirements that are measurable in effectiveness, efficiency, satisfaction, and in avoidance of harm that could arise from specific use cases.]

3.4 Interface Requirements

[List all inputs and outputs from the system. It should mirror but not repeat the information found in sections 4.2- 4.6. For each defined interface, be sure to include:

- *the name of the item*
- *description of the purpose of the interface*
- *source of input OR output destination*
- *range, accuracy and/or tolerance*
- *units of measurement*
- *timing*
- *I/O relationships*
- *data formats*
- *command formats*

any information included within the I/O.]]

Import:	Export:
- JSON File that can be turned into a family tree	- Tree as a JSON File

3.5 Logical Database Requirements

[Identify the logical requirements for information that will be place in a database. These requirements should consider and include:

- *Types of information that will be used.*
- *Frequency of use*
- *accessibility*
- *integrity constraints*
- *security*
- *data retention*
- *data entities and their relationships.]*

Database Requirements:

1. User Class
    a. Permission Levels
    b. Family Trees
    c. Shared Trees
    d. Oauth Tokens
2. Family Tree
    a. All the people included in the tree

3.6 Design Constraints

[List any constraints on the system. These constraints should be from external sources such as regulatory standards, legal, or project limitations.]

Balkan Family Tree requires you to pay in order to actively store the Family Tree, so we've stored all of our Family Tree in our own database. This means that sharing permission was limited as we wouldn't be able to allow multiple users to edit the tree at the same time. They could access and edit it one by one, but they would have to refresh the page manually after every edit.

3.7 Software System Attributes

[For each of the attributes of the software system (Reliability, Availability, Security, Maintainability, Portability, etc.), list the factors that will establish functionality or stability. For example, when establishing requirements for the Security attribute, you may include one that restricts communication between two one area of the program and another.]

3.8 Supporting Information

[Add any additional information needed to understand the SRS, include things like background information, problem descriptions, packaging instructions for code, sample input/output formats, etc.]

4. Verification

[List all inputs and outputs from the system. It should mirror but not repeat the information found in sections 3.1- 3.8. For each defined interface, be sure to include:

- *the name of the item*
- *description of the purpose of the interface*
- *source of input OR output destination*
- *range, accuracy and/or tolerance*
- *units of measurement*
- *timing*
- *I/O relationships*
- *data formats*
- *command formats*
- *any information included within the I/O.]*

4.1 Functions

[See sections 4.0, 3.1 for specific directions about what outputs should be included here.]

4.2 Performance Requirements

[See sections 4.0, 3.2 for specific directions about what outputs should be included here.]

4.3 Usability Requirements

[See sections 4.0, 3.3 for specific directions about what outputs should be included here.]

4.4 Interface Requirements

[See sections 4.0, 3.4 for specific directions about what outputs should be included here.]

4.5 Logical Database Requirements

[See sections 4.0, 3.5 for specific directions about what outputs should be included here.]

4.6 Design Constraints

[See sections 4.0, 3.6 for specific directions about what outputs should be included here.]

4.7 Software System Attributes

[See sections 4.0, 3.7 for specific directions about what outputs should be included here.]

4.8 Supporting Information

[See sections 4.0, 3.8 for specific directions about what outputs should be included here.]

5. Appendix A – Tailoring Policies

5.1 Assumptions and dependencies

[Identify any and all factors that may impact the implementation and execution of the requirements written below. These factors do not add a constraint but may impact development if they are changed. Example: a major update to an operating system(OS) on which the SUD is intended to run impacts the implementation of one of the core features. The version of the OS that the system had intended to run on should be listed in this section.]

5.2 Acronyms and Abbreviations

Term	Definition	Abbreviation	Source (if applicable)
Example: context of use	*users, tasks, equipment (hardware, software and*	*COU*	*[SOURCE: ISO/IEC 25000:2014, 4.2]*

	materials), and the physical and social environments in which a product is used		
System Under Development	The system which is actively being developed.	SUD	
To Be Determined	The work has not been resolved yet and is being thought about.	TBD	
Database	Collection of organized data	DB	

5.3 Tailoring Policies

Tailoring is not a requirement to bring the document into compliance with the standards set by IEEE 29148-2018. Tailoring should only occur when conformance to the standard is not possible or practical. The act of tailoring is the modification and/or removal of one of the content sections outlined in this document, adding additional information items for organization is not considered tailoring. Tailoring should only occur when factors or circumstances:

- surround an organization that is using the document
- influence a project using this document to meet an agreement
- reflect the needs of an organization.

When tailoring the document, the following activities shall be implemented:

- Identify and document the circumstances that may influence tailoring.
    - novelty, size and complexity
    - stability of operating environments
    - variety in operating environments
    - starting date and duration
    - emerging technology
    - availability of services of enabling systems
    - other standards with which the document needs to conform.
- Identify and get input from all parties impacted by the tailoring process.
    - Such as stakeholders, contributors, and other interested parties
- Delete the information contents that require tailoring.

6. Appendix B – Copyright

This document is based on a template meeting the ISO/IEC/IEEE 29148-2018 standard, available at https://www.iso.org/standard/72089.html. Template authors are:

Dr. rer. nat. Bastian Tenbergen,
Associate Professor of Software Engineering
bastian@tenbergen.org

Mikayla Conner-Spagnola, MA
Independent Consultant
mconner@oswego.edu

Department of Computer Science
State University of New York at Oswego
Oswego, NY 13126, United States