Software Requirements Specification

Version 1.0

<<Annotated Version>>

April 15, 2018

Web Publishing System

Joan Teamleader

Paul Adams

Bobbie Baker

Charles Charlie

Submitted in partial fulfillment

Of the requirements of

CS 310 Software Engineering

<<Any comments inside double brackets such as these are *not* part of this SRS but are comments upon this SRS example to help the reader understand the point being made.

Refer to the SRS Template for details on the purpose and rules for each section of this document.

This work is based upon the submissions of the Spring 2004 CS 310. The students who submitted these team projects were Thomas Clay, Dustin Denney, Erjon Dervishaj, Tiffanie Dew, Blake Guice, Jonathan Medders, Marla Medders, Tammie Odom, Amro Shorbatli, Joseph Smith, Jay Snellen, Chase Tinney, and Stefanie Watts. >>

**Table of Contents**

Table of Contents i

List of Figures ii

1.0. Introduction 1

1.1. Purpose 1

1.2. Scope of Project 1

1.3. Glossary 2

1.4. References 2

1.5. Overview of Document 2

2.0. Overall Description 4

2.1 System Environment 4

2.2 Functional Requirements Specification 5

2.2.1 Reader Use Case 5

Use case: Search Article 5

2.2.2 Author Use Case 6

Use case: Submit Article 6

2.2.3 Reviewer Use Case 7

Use case: Submit Review 7

2.2.4 Editor Use Cases 8

Use case: Update Author 8

Use case: Update Reviewer 9

Use case: Update Article 9

Use case: Receive Article 10

Use case: Assign Reviewer 11

Use case: Receive Review 11

Use case: Check Status 12

Use case: Send Response 12

Use case: Send Copyright 13

Use case: Remove Article 14

Use case: Publish Article 14

2.3 User Characteristics 15

2.4 Non-Functional Requirements 15

3.0. Requirements Specification 17

3.1 External Interface Requirements 17

3.2 Functional Requirements 17

3.2.1 Search Article 17

3.2.2 Communicate 18

3.2.3 Add Author 18

3.2.4 Add Reviewer 19

3.2.5 Update Person 19

3.2.6 Update Article Status 20

3.2.7 Enter Communication 20

3.2.8 Assign Reviewer 21

3.2.9 Check Status 21

3.2.10 Send Communication 22

3.2.11 Publish Article 22

3.2.12 Remove Article 23

3.3 Detailed Non-Functional Requirements 23

3.3.1 Logical Structure of the Data 23

3.3.2 Security 25

Index 26

**List of Figures**

Figure 1 - System Environment 4

Figure 2 - Article Submission Process 6

Figure 3 - Editor Use Cases 8

Figure 4 - Logical Structure of the Article Manager Data 23

**1.0. Introduction**

***1.1. Purpose***

The purpose of this document is to present a detailed description of the Web Publishing System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

***1.2. Scope of Project***

This software system will be a Web Publishing System for a local editor of a regional historical society. This system will be designed to maximize the editor’s productivity by providing tools to assist in automating the article review and publishing process, which would otherwise have to be performed manually. By maximizing the editor’s work efficiency and production the system will meet the editor’s needs while remaining easy to understand and use.

More specifically, this system is designed to allow an editor to manage and communicate with a group of reviewers and authors to publish articles to a public website. The software will facilitate communication between authors, reviewers, and the editor via E-Mail. Preformatted reply forms are used in every stage of the articles’ progress through the system to provide a uniform review process; the location of these forms is configurable via the application’s maintenance options. The system also contains a relational database containing a list of Authors, Reviewers, and Articles.

***1.3. Glossary***

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Active Article | The document that is tracked by the system; it is a narrative that is planned to be posted to the public website. |
| Author | Person submitting an article to be reviewed. In case of multiple authors, this term refers to the *principal author*, with whom all communication is made. |
| Database | Collection of all the information monitored by this system. |
| Editor | Person who receives articles, sends articles for review, and makes final judgments for publications. |
| Field | A cell within a form. |
| Historical Society Database | The existing membership database (also HS database). |
| Member | A member of the Historical Society listed in the HS database. |
| Reader | Anyone visiting the site to read articles. |
| Review | A written recommendation about the appropriateness of an article for publication; may include suggestions for improvement. |
| Reviewer | A person that examines an article and has the ability to recommend approval of the article for publication or to request that changes be made in the article. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| User | Reviewer or Author. |

***1.4. References***

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

***1.5. Overview of Document***

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

**2.0. Overall Description**

***2.1 System Environment***

**Figure - System Environment**

The Web Publishing System has four active actors and one cooperating system.

The Author, Reader, or Reviewer accesses the Online Journal through the Internet. Any Author or Reviewer communication with the system is through email. The Editor accesses the entire system directly. There is a link to the (existing) Historical Society.

<< The division of the Web Publishing System into two component parts, the Online Journal and the Article Manager, is an example of using domain classes to make an explanation clearer. >>

***2.2 Functional Requirements Specification***

This section outlines the use cases for each of the active readers separately. The reader, the author and the reviewer have only one use case apiece while the editor is main actor in this system.

2.2.1 Reader Use Case

Use case:  **Search Article**

**Diagram:**

**Brief Description**

The Reader accesses the Online Journal Website, searches for an article and downloads it to his/her machine.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Reader has already accessed the Online Journal Website.

1. The Reader chooses to search by author name, category, or keyword.
2. The system displays the choices to the Reader.
3. The Reader selects the article desired.
4. The system presents the abstract of the article to the reader.
5. The Reader chooses to download the article.
6. The system provides the requested article.

**Xref:** Section 3.2.1, Search Article

**Figure - Article Submission Process**

The *Article Submission Process* state-transition diagram summarizes the use cases listed below. An Author submits an article for consideration. The Editor enters it into the system and assigns it to and sends it to at least three reviewers. The Reviewers return their comments, which are used by the Editor to make a decision on the article. Either the article is accepted as written, declined, or the Author is asked to make some changes based on the reviews. If it is accepted, possibly after a revision , the Editor sends a copyright form to the Author. When that form is returned, the article is published to the Online Journal. Not shown in the above is the removal of a declined article from the system.

2.2.2 Author Use Case

In case of multiple authors, this term refers to the *principal author*, with whom all communication is made.

Use case: Submit Article

**Diagram:**

**Brief Description**

The author either submits an original article or resubmits an edited article.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Author has already connected to the Online Journal Website.

* The Author chooses the *Email Editor* button.
* The System uses the *sendto* HTML tag to bring up the user’s email system.
* The Author fills in the Subject line and attaches the files as directed and emails them.
* The System generates and sends an email acknowledgement.

**Xref:** Section 3.2.2, Communicate

2.2.3 Reviewer Use Case

Use case: Submit Review

**Diagram:**

**Brief Description**

The reviewer submits a review of an article.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Reviewer has already connected to the Online Journal Website.

* The Reviewer chooses the *Email Editor* button.
* The System uses the *sendto* HTML tag to bring up the user’s email system.
* The Reviewer fills in the Subject line and attaches the file as directed and emails it.
* The System generates and sends an email acknowledgement.

**Xref:** Section 3.2.2, Communicate

2.2.4 Editor Use Cases

The Editor has the following sets of use cases:

**Figure - Editor Use Cases**

**Update Information use cases**

Use case: Update Author

**Diagram:**

**Brief Description**

The Editor enters a new Author or updates information about a current Author.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the main page of the Article Manager.

* The Editor selects to *Add/Update Author*.
* The system presents a choice of adding or updating.
* The Editor chooses to add or to update.
* If the Editor is updating an Author, the system presents a list of authors to choose from and presents a grid filling in with the information; else the system presents a blank grid.
* The Editor fills in the information and submits the form.
* The system verifies the information and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.3, Add Author; Section 3.2.5 Update Person

Use case: Update Reviewer

**Diagram:**

**Brief Description**

The Editor enters a new Reviewer or updates information about a current Reviewer.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the main page of the Article Manager.

* The Editor selects to *Add/Update Reviewer*.
* The system presents a choice of adding or updating.
* The Editor chooses to add or to update.
* The system links to the Historical Society Database.
* If the Editor is updating a Reviewer, the system and presents a grid with the information about the Reviewer; else the system presents list of members for the editor to select a Reviewer and presents a grid for the person selected.
* The Editor fills in the information and submits the form.
* The system verifies the information and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.4, Add Reviewer; Section 3.2.5, Update Person

Use case: Update Article

**Diagram:**

**Brief Description**

The Editor enters information about an existing article.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the main page of the Article Manager.

* The Editor selects to *Update Article*.
* The system presents s list of active articles.
* The system presents the information about the chosen article.
* The Editor updates and submits the form.
* The system verifies the information and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.6, Update Article Status

**Handle Article use cases**

Use case: Receive Article

**Diagram:**

**Brief Description**

The Editor enters a new or revised article into the system.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the main page of the Article Manager and has a file containing the article available.

* The Editor selects to *Receive Article*.
* The system presents a choice of entering a new article or updating an existing article.
* The Editor chooses to add or to update.
* If the Editor is updating an article, the system presents a list of articles to choose from and presents a grid for filling with the information; else the system presents a blank grid.
* The Editor fills in the information and submits the form.
* The system verifies the information and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.7, Enter Communication

Use case: Assign Reviewer

This use case extends the *Update Article* use case.

**Diagram:**

**Brief Description**

The Editor assigns one or more reviewers to an article.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

* The Editor selects to *Assign Reviewer*.
* The system presents a list of Reviewers with their status (see data description is section 3.3 below).
* The Editor selects a Reviewer.
* The system verifies that the person is still an active member using the Historical Society Database.
* The Editor repeats steps 3 and 4 until sufficient reviewers are assigned.
* The system emails the Reviewers, attaching the article and requesting that they do the review.
* The system returns the Editor to the *Update Article* use case.

**Xref:** Section 3.2.8, Assign Reviewer

Use case: Receive Review

This use case extends the *Update Article* use case.

**Diagram:**

**Brief Description**

The Editor enters a review into the system.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

* The Editor selects to *Receive Review*.
* The system presents a grid for filling with the information.
* The Editor fills in the information and submits the form.
* The system verifies the information and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.7, Enter Communication

**Check Status use case:**

Use case: Check Status

**Diagram:**

**Brief Description**

The Editor checks the status of all active articles.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the main page of the Article Manager.

* The Editor selects to *Check Status*.
* The system returns a scrollable list of all active articles with their status (see data description in section 3.3 below).
* The system returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.9, Check Status

**Send Recommendation use cases:**

Use case: Send Response

This use case extends the *Update Article* use case.

**Diagram:**

**Brief Description**

The Editor sends a response to an Author.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

* The Editor selects to *Send Response*.
* The system calls the email system and puts the Author’s email address in the Recipient line and the name of the article on the subject line.
* The Editor fills out the email text and sends the message.
* The system returns the Editor to the Article Manager main page.

**Xref:** Section 3.210, Send Communication

Use case: Send Copyright

This use case extends the *Update Article* use case.

**Diagram:**

**Brief Description**

The Editor sends a copyright form to an Author.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

* The Editor selects to *Send Copyright*.
* The system calls the email system and puts the Author’s email address in the Recipient line, the name of the article on the subject line, and attaches the copyright form.
* The Editor fills out the email text and sends the message.
* The system returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.10, Send Communication

Use case: Remove Article

This use case extends the *Update Article* use case.

**Diagram:**

**Brief Description**

The Editor removes an article from the active category.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

* The Editor selects to remove an article from the active database.
* The system provides a list of articles with the status of each.
* The Editor selects an article for removal.
* The system removes the article from the active article database and returns the Editor to the Article Manager main page.

**Xref:** Section 3.2.12, Remove Article

**Publish Article use case:**

Use case: Publish Article

This use case extends the *Update Article* use case.

**Diagram:**

**Brief Description**

The Editor transfers an accepted article to the Online Journal.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the article using the *Update Article* use case.

* The Editor selects to *Publish Article*.
* The system transfers the article to the Online Journal and updates the search information there.
* The system removes the article from the active article database and returns the Editor to the Article Manager home page.

**Xref:** Section 3.2.11, Publish Article

<< Since three of the actors only have one use case each, the summary diagram only involves the Editor. Adapt the rules to the needs of the document rather than adapt the document to fit the rules. >>

***2.3 User Characteristics***

The Reader is expected to be Internet literate and be able to use a search engine. The main screen of the Online Journal Website will have the search function and a link to “Author/Reviewer Information.”

The Author and Reviewer are expected to be Internet literate and to be able to use email with attachments.

The Editor is expected to be Windows literate and to be able to use button, pull-down menus, and similar tools.

The detailed look of these pages is discussed in section 3.2 below.

***2.4 Non-Functional Requirements***

The Online Journal will be on a server with high speed Internet capability. The physical machine to be used will be determined by the Historical Society. The software developed here assumes the use of a tool such as Tomcat for connection between the Web pages and the database. The speed of the Reader’s connection will depend on the hardware used rather than characteristics of this system.

The Article Manager will run on the editor’s PC and will contain an Access database. Access is already installed on this computer and is a Windows operating system.

**3.0. Requirements Specification**

***3.1 External Interface Requirements***

***Requesting a link with the database to install the reservation***

***The rule fields define the time, hour, and day fields and give the manager available trips in time and place and his role to check on company reservation***

***3.2 Functional Requirements***

The Logical Structure of the Data is contained in Section 3.3.1.

3.2.1 yourroad

|  |  |
| --- | --- |
| **Use Case Name** | yourroad |
| **XRef** | Section 2.2.1, yourroad  SDD, Section 7.1 |
| **the user** | Make reservations with an app |
| **Basic Path** | The user opens the app, the reservation interface appears, chooses the place, time, and flight they want, and presses OK where you go to the database and install the reservation |
| **Alternative Paths** | * In step 2, Go to the company and book with it |
|  |  |
| **Exception Paths** | cancellation of reservation. |
| **Other** | The reservation list is created |

3.2.2 Add reservation

|  |  |
| --- | --- |
| **Use Case Name** | Add reservation |
|  |  |
| **user** | The user selects to add a new reservation to the database. |
| **Basic Path** | * The user opens the app, the reservation interface appears, chooses the place, time, and flight they want, and presses OK where you go to the database and install the reservation. |
| **Alternative Paths** | Go to the company and book with it |
| **Exception Paths** | cancellation of reservation. |
|  |  |

3.2.3 Update reservation Status

|  |  |
| --- | --- |
| **Use Case Name** | Update reservation Status |
|  |  |
| **user** | The user selects to update the status of an reservation in the database. |
|  |  |
| **Basic Path** | * The user chooses the reservation they want to modify * The rule displays and updates the reservation. |
| **Alternative Paths** | Go to the company and amend the reservation |
| **Postcondition** | The database has been updated. |
| **Exception Paths** | If the reservation is not in the database, it will be canceled and re-booked |
| **Other** | This use case can be used to amend a reservation or to re-reserve |

3.2.4 Remove reservation

|  |  |
| --- | --- |
| **Use Case Name** | Remove reservation |
| **XRef** | Section 2.2.4, Remove reservation  SDD, Section 7.12 |
| **User** | The user selects to remove reservation from the database. |
|  |  |
| **Basic Path** | * The system provides a list of all reservations * The user chooses to reserve * The system displays the reservation information and asks the manager to confirm the deletion * The manager confirms the deletion. |
| **Alternative Paths** | Go to the company and confirm the deletion |
| **Postcondition** | The reservation is removed from the database. |
| **Exception Paths** | The user may abandon the reservation |
|  |  |

***3.3 Detailed Non-Functional Requirements***

3.3.1 Logical Structure of the Data

The logical structure of the data to be stored in the internal Article Manager database is given below.

**Figure - Logical Structure of the Article Manager Data**

The data descriptions of each of these data entities is as follows:

**Author Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of principle author |  |
| Email Address | Text | Internet address |  |
| Article | Pointer | Article entity | May be several |

**Reviewer Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of principle author |  |
| ID | Integer | ID number of Historical Society member | Used as key in Historical Society Database |
| Email Address | Text | Internet address |  |
| Article | Pointer | Article entity of | May be several |
| Num Review | Integer | Review entity | Number of not returned reviews |
| History | Text | Comments on past performance |  |
| Specialty | Category | Area of expertise | May be several |

**Review Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Article | Pointer | Article entity |  |
| Reviewer | Pointer | Reviewer entity | Single reviewer |
| Date Sent | Date | Date sent to reviewer |  |
| Returned | Date | Date returned; null if not returned |  |
| Contents | Text | Text of review |  |

**Article Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Article |  |
| Author | Pointer | Author entity | Name of principle author |
| Other Authors | Text | Other authors is any; else null | Not a pointer to an Author entity |
| Reviewer | Pointer | Reviewer entity | Will be several |
| Review | Pointer | Review entity | Set up when reviewer is set up |
| Contents | Text | Body of article | Contains Abstract as first paragraph. |
| Category | Text | Area of content | May be several |
| Accepted | Boolean | Article has been accepted for publication | Needs Copyright form returned |
| Copyright | Boolean | Copyright form has been returned | Not relevant unless Accepted is True. |
| Published | Boolean | Sent to Online Journal | Not relevant unless Accepted is True. Article is no longer active and does not appear in status checks. |

The Logical Structure of the data to be stored in the Online Journal database on the server is as follows:

**Published Article Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Article |  |
| Author | Text | Name of one Author | May be several |
| Abstract | Text | Abstract of article | Used for keyword search |
| Content | Text | Body of article |  |
| Category | Text | Area of content | May be several |

3.3.2 Security

The server on which the Online Journal resides will have its own security to prevent unauthorized *write*/*delete* access. There is no restriction on *read* access. The use of email by an Author or Reviewer is on the client systems and thus is external to the system.

The PC on which the Article Manager resides will have its own security. Only the Editor will have physical access to the machine and the program on it. There is no special protection built into this system other than to provide the editor with *write* access to the Online Journal to publish an article.

**Index**

Abstract, 6, 17, 27

add, 9, 11, 19, 20, 21

Add, 8, 9, 19

Article, 1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28

Article Manager, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 19, 20, 21, 22, 23, 24, 25, 28

Author, 1, 4, 5, 6, 7, 8, 9, 13, 14, 16, 17, 19, 20, 22, 23, 25, 26, 27

Category, 5, 14, 17, 18, 20, 21, 23, 26, 27

Database, 2, 9, 11, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 25, 26, 27

Editor, 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 28

Field, 17, 19, 20

Form, 1, 6, 9, 10, 11, 12, 14, 19, 20, 21, 23, 24, 27

Grid, 9, 11, 12, 19, 20, 21

Historical Society, 1, 5, 9, 11, 16, 17, 19, 20, 26

Online Journal, 4, 5, 6, 7, 15, 16, 17, 18, 24, 27, 28

Reader, 4, 5, 6, 16, 17, 18

Review, 1, 7, 11, 12, 18, 21, 23, 26, 27

Reviewer, 1, 4, 5, 6, 7, 9, 11, 16, 17, 19, 20, 21, 22, 23, 26, 27

Security, 27, 28

Status, 11, 12, 13, 14, 17, 21, 22, 23, 27

update, 9, 11, 20, 21

Update, 8, 9, 10, 11, 12, 13, 14, 15, 17, 19, 20, 21, 22

User, 7, 16, 18

Web Publishing System, 1, 4, 5, 17