Software Requirements Specification

Version 1.0

<<Annotated Version>>

April 15, 2018

Red Crescent Human Resources Office management system

Aya Arnous

Esraa Balasem

Rouaa Awarek

Aya Sharbek

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](#_Toc77487619)

[List of Figures ii](#_Toc77487620)

[1.0. Introduction 1](#_Toc77487621)

[1.1. Purpose 1](#_Toc77487622)

[1.2. Scope of Project 1](#_Toc77487623)

[1.3. Glossary 2](#_Toc77487624)

[1.4. References 2](#_Toc77487625)

[1.5. Overview of Document 2](#_Toc77487626)

[2.0. Overall Description 4](#_Toc77487627)

[2.1 System Environment 4](#_Toc77487628)

2.2 Functional requirment spesefication………………………………………………………………..4

2.2.1 Volunteer use case…………………………………………………………………………….4

2.2.1 Submit usecase…………………………………………………………………………………4

2.2.2 Team usecase………………………………………………………………5

2.2.2 usecase: filter requistsdiagram……………………………………………5

2.2.3 Project maneger usecase…………………………………………………..6

2.2.3 Determination of compensation usecase……………………………………6

2.2.4 Project coordinator usecas………………………………………………7

2.2.4 issusing as acing usecase………………………………………………..7

2.3 user characteristics----------------------------------------------------------------------- 10

2.4 non-functional requirments------------------------------------------------------- 10

# 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

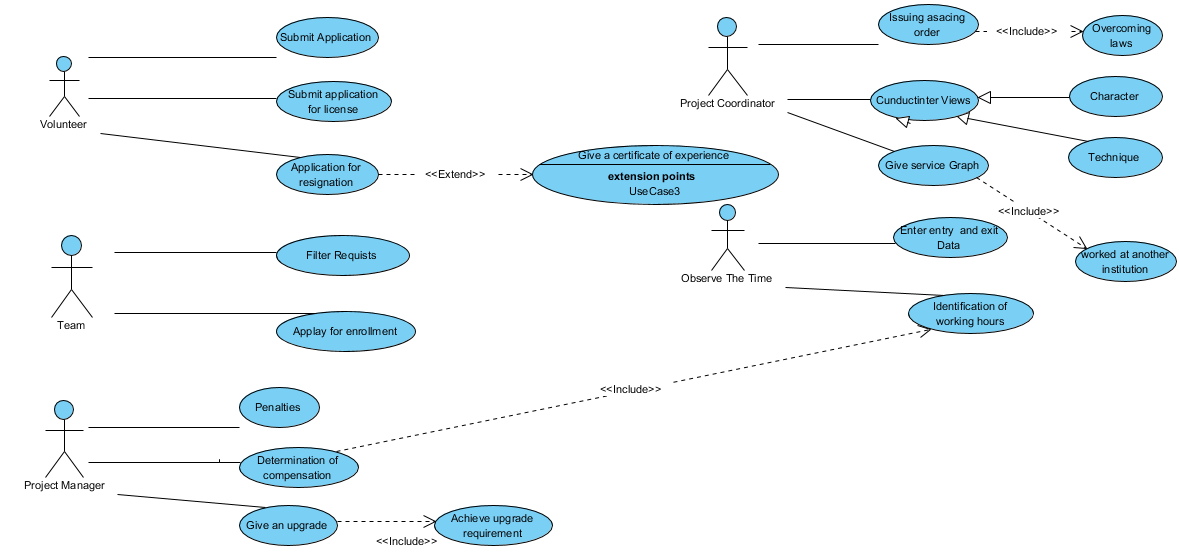


Figure 1 - System Environment

The goal of this file is to describe the system to be developed, and to present the tasks that this system should provide to meet all customer needs, as this file contains the expected duration of project completion and final delivery

## 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 submit Use Case

#### Use case: submit

**Diagram:**

**Brief Description**

voluteer

Submit Aplication

The applications were submitted by filling out the (volunteer requests) forms

**Initial Step-By-Step Description**

1-Applications submitted are filtered according to certain criteria 2

2. Accepted persons are invited to be interviewed in the Human Resources Department

3. The volunteer will undergo a three-month trial period

4 Training sessions for volunteers are held in some departments

**Xref:** Section 3.2.1,

### 2.2.2 Team Use Case

#### Use case: Filter requists

team

Filter requists

**Diagram:**

**Brief Description**

The team filter the reuists to choose the best

**Initial Step-By-Step Description**

1\_ Volunteer requests are submitted at the Human Resources Office

2\_These requests are stored in the archive

3\_ The human resources department team filters these requests according to specific criteria set by the administration

**Xref:** Section 3.2.2,

### 2.2.3 Deretmination of compenation Use Case

#### Use case: Determination of compenation

**Diagram:**

Project maneger

Determination o

Of

compenation

**Brief Description**

The project maneger determain compenation

**Initial Step-By-Step Description**

1\_Volunteer entry and exit information is recorded

2\_ The number of hours worked by the volunteer is calculated

3\_ A test is made if this volunteer is subject to a full shift system or fixed hours

4\_ The appropriate compensation is determined for the volunteer

5\_ The compensation raises the volunteer's account

**Xref:** Section 3.2.2

### 2.2.4 issusing asacing order Use Cases

#### Use case: issusing asacing order Author

**Diagram:**

Project coordinator

issusing asacing order

**Brief Description**

Alerts are added to the volunteers' files, which violate working conditions or whose work breaks exceeded three months.

**Xref:** Section 3.2.3

#### Use case: Update Article

**Diagram:**

Observe the time

Indentifecation of working houres

**Brief Description**

The observer enters information about the working houres.

**Initial Step-By-Step Description**

1. The working hours of the volunteers are determined as rotating periods.

2. The value of one compensation is determined for all volunteers, in the event that they commit to rotating in a complete paralysis, so the volunteer’s work hours are referred to to determine the appropriate compensation for the periods in which he is delegated.

**Xref:** Section 3.2.6,

## 2.3 User Characteristics

Volunteer: The volunteer must have knowledge of the nature of the work, and the volunteer must be 18 years or older

the team: The employees should be responsible for receiving volunteer requests in order to accomplish the required services and be in good dealing with volunteer applicants

project manager: It manages the project in order to accomplish the required services in order to integrate its responsibilities with the resource team

project coordinator: He must be fully knowledgeable to supervise the vacancy department for interviews

observe the time: He must be fully aware of the staff entry and exit times

## 2.4 Non-Functional Requirements

1- The system works on a local network.

2- Entering and leaving using the handprint.

3- Back up volunteer data.

4- The system should be easy to use and flexible.

5- The system should be highly secure.

3.0. Requirements Specification

## 3.1 External Interface Requirements

## 3.2 Functional Requirements

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

### 3.2.1

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **XRef** |  |
| **Trigger** |  |
| **Precondition** |  |
| **Basic Path** |  |
| **Alternative Paths** | . |
| **Postcondition** |  |
| **Exception Paths** |  |
| **Other** |  |

### 3.2.2

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **XRef** |  |
| **Trigger** |  |
| **Precondition** |  |
| **Basic Path** | . |
| **Alternative Paths** |  |
| **Postcondition** |  |
| **Exception Paths** |  |
| **Other** |  |

### 3.2.3

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **XRef** |  |
| **Trigger** |  |
| **Precondition** |  |
| **Basic Path** | . |
| **Alternative Paths** |  |
| **Postcondition** |  |
| **Exception Paths** |  |
| **Other** | . |

### 3.2.4

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **XRef** |  |
| **Trigger** |  |
| **Precondition** |  |
| **Basic Path** |  |
| **Alternative Paths** | , |
| **Postcondition** |  |
| **Exception Paths** |  |
| **Other** |  |

### 3.2.5

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **XRef** |  |
| **Trigger** | . |
| **Precondition** |  |
| **Basic Path** | . |
| **Alternative Paths** |  |
| **Postcondition** |  |
| **Exception Paths** |  |
| **Other** | . |

## 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.

s

Figure 4 - Logical Structure of the Article Manager Data

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 3.3.2 Security

The server on which the Online site 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.