Software Requirements Specification

Version 1.0

<<Annotated Version>>

June 22, 2020

The Syrian Arab Red Crescent

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

[3.0. Requirements Specification 17](#_Toc77487650)

[3.1 External Interface Requirements 17](#_Toc77487651)

[3.2 Functional Requirements 17](#_Toc77487652)

[3.2.1 Submit](#_Toc77487653)Request

[3.2.2 FilterRequest](#_Toc77487654)

[3.2.3 Determine](#_Toc77487655)TheCompensationOfTheVolunteer

[3.2.4 Issuing](#_Toc77487656)Warning

[3.2.5 DetermineWorkingHours](#_Toc77487657)

[3.3 Detailed Non-Functional Requirements 23](#_Toc77487665)

[3.3.1 Logical Structure of the Data 23](#_Toc77487666)

[3.3.2 Security 25](#_Toc77487667)

[Index 26](#_Toc77487668)

# 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

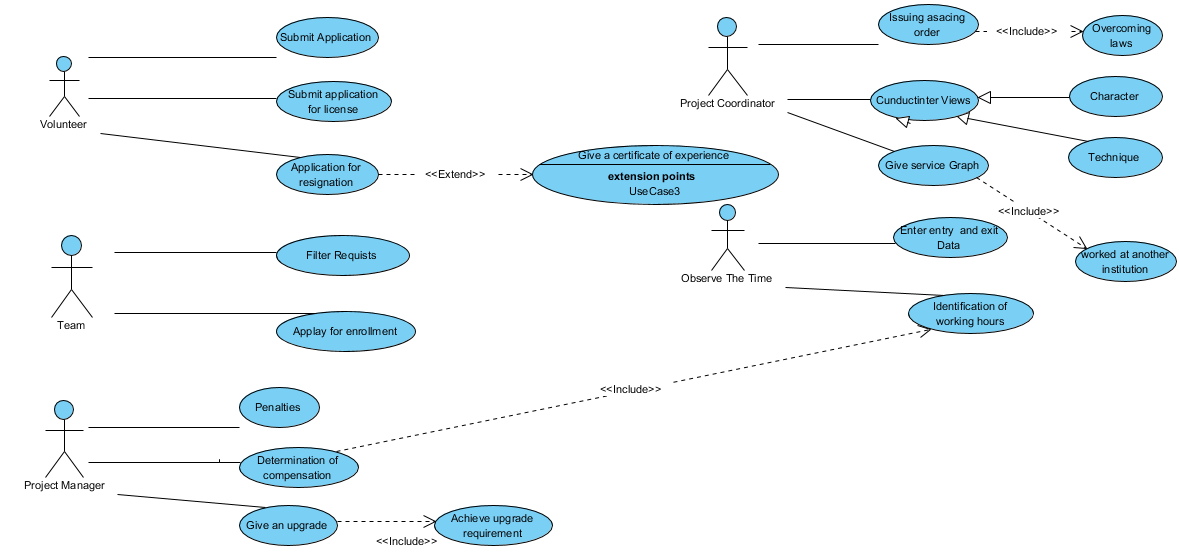


Figure 1 - System Environment

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

The only link to an external system is the link to the Syrian Arab Red Crescent Database to verify the membership of a Volunteer . The Editor believes that a society member is much more likely to be an effective Volunteer and has imposed a membership requirement for a Volunteer. The **SARC** Database fields of interest to the organization are member’s name, membership (ID) number, and email address (an optional field for the SARC Database).

The *Assign* Volunteer *r* use case sends the Volunteer ID to the SARC Database and a Boolean is returned denoting membership status. The *Update* Volunteer use case requests a list of member names, membership numbers and (optional) email addresses when adding a new Volunteer. It returns a Boolean for membership status when updating a Volunteer.

## 3.2 Functional Requirements

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

### 3.2.1 Submit Request:

|  |  |
| --- | --- |
| **Use Case Name** | Submit a Request |
| **XRef** | Section 2.2.1, Search Article  SDD, Section 7.1 |
| **Trigger** | The volunteer submit a volunteer request with the assistance of the human resources team. |
| **Precondition** | The presence of advertisements on the organization's website about the opening of volunteering. |
| **Basic Path** | 1. Put advertisements on the organization's website about opening the door to volunteering. 2. The volunteer goes to the Human Resources Office. 3. Volunteer requests are filled out. 4. Storage of requests for volunteering to carry out the next stage of filtering and choosing the accepted ones.   The system displays the Abstract for the article. |
| **Postcondition** | The request process is successful and recorded. |
| **Exception Paths** | The application period has expired. |
| **Other** | - |

### 3.2.2 Filter Requests:

|  |  |
| --- | --- |
| **Use Case Name** | Filter Requests |
| **XRef** | Section 2.2.2, Filter Requests |
| **Trigger** | The Human Resources Department filters requests submitted by volunteers. |
| **Precondition** | The success of the filtering process and communication with the accepted ones whose mark exceeded 60 and above. |
| **Basic Path** | 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 the administration.   An contact with accepted persons whose marks have 60 or more is completed so that the stage of submitting requests and under 60 is placed as a precaution. |
| **Postcondition** | The success of the filtering process and communication with the accepted ones whose mark exceeded 30 and above |
| **Exception Paths** | Store requests of people whose marks are below 60 and reserve them. |
| **Other** | - |

### 3.2.3 Determine the compensation of the volunteer:

|  |  |
| --- | --- |
| **Use Case Name** | Determine the compensation of the volunteer |
| **XRef** | Section 2.2.3, Determine the compensation of the volunteer |
| **Trigger** | The project manager grants specific compensation to this volunteer. |
| **Precondition** | Ensure full rotation of the volunteer or specific hours. |
| **Basic Path** | 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 the volunteer is subject to a full shift system or fixed hours.  4. The appropriate compensation is determined for the volunteer  The compensation raises the volunteer's account. |
| **Postcondition** | Raise compensation for the volunteer. |
| **Exception Paths** | Failure to pay compensation to the volunteer due to his interruption in work or failure to shift the number of specified hours. |
| **Other** | . - |

### 3.2.4 Issuing Warnings:

|  |  |
| --- | --- |
| **Use Case Name** | Issuing Warnings |
| **XRef** | Section 2.2.4, Issuing Warnings |
| **Trigger** | The project coordinator submits the warning request and puts it into. |
| **Precondition** | Repeat the unseen more than once. |
| **Basic Path** | 1. Alerts are added to the volunteer's files, in violation of working conditions, or those whose breaks exceeded three month.   Sanctions are issued by the project manager against the volunteer if he finds repeated warnings issued against him |
| **Alternative Paths** | The Volunteer does not receive the warnings. |
| **Postcondition** | The volunteer will receive a warnings. |
| **Exception Paths** | Salary reduction of 50%. |
| **Other** | - |

### 3.2.5 Determine working hours:

|  |  |
| --- | --- |
| **Use Case Name** | Determine working hours |
| **XRef** | Section 2.2.5, Determine working hours |
| **Trigger** | .Fingerprint device. |
| **Precondition** | Log in to the employee. |
| **Basic Path** | 1. Working hours for volunteers are determined as shift periods.   The value of one compensation is determined for all volunteers, in the event that they adhere to rotation completely, as it refers of the working hours to determine the appropriate compensation for the periods in which he rotated. |
| **Postcondition** | The employee's commitment to work. |
| **Exception Paths** | The employee's commitment to work. |
| **Other** | . Not Logged in. |

## 3.3 Detailed Non-Functional Requirements

### 3.3.1 Logical Structure of the Data

The logical structure of the database is given below.

Review

team

Request

Volunteer

Provid

sent to

join

s

has

Figure 4 - Logical Structure of the Article Manager Data

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

**Project Manager**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Manager | - |
| Email Address | Text | Internet address | - |
| Permit | String | All Permit | - |

**Volunteer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Volunteer | - |
| ID | Integer | Identification Card within the Organization | - |
| Email Address | Text | Internet address | - |
| Blood Type | String | Positive or negative | - |
| Personal Picture | Pixel | Colored | - |

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Number of team members | Pointer | Working Team | - |
| Name | Pointer | Name Team | Group of Member |
| Department | String | Human Resources Team | Responsible for filtering requests |

**Project Coordinator**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Project Coordinator | - |
| Email Address | Text | Internet address | - |
| File the Volunteer | Documents | Volunteer Information | Warnings are added in it if the Volunteer Violates the conditions |

The Logical Structure of the data to be stored is as follows:

**Published Volunteer Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Volunteer | - |
| ID | Integer | Identification Card within the Organization | - |
| Email Address | Text | Internet Address | - |
| Blood Type | String | Positive or negative | - |
| Personal Picture | Pixel | Colored | - |

### 3.3.2 Security

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

# 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