Software Requirements Specification

*This document outline is based on the IEEE Standard 830-1993 for Software Requirements Specifications.*

*This document was created in part by Steve Mattingly (smattingly@computer.org).*

*This document should specify what functions are to be performed on what data to produce what results at what location for whom.*

*A properly written SRS limits the range of valid designs, but does not specify any particular design.*

*A good SRS is*

* *Correct (accurately captures the “real” requirements)*
* *Unambiguous (all statements have exactly one interpretation)*
* *Complete (where TBDs are absolutely necessary, document why the information is unknown, who is responsible for resolution, and the deadline)*
* *Consistent*
* *Ranked for importance and/or stability*
* *Verifiable (avoid soft descriptions like “works well”, “is user friendly”; use concrete terms specify measurable quantities)*
* *Modifiable (evolve the SRS only via a formal change process, preserving a complete audit trail of changes)*
* *Traceable (cross reference with source documents and spawned documents)*

*The paragraphs written in the “Comment” style are for the benefit of the person writing the document and should be removed before the document is finalized.*

**Version: Draft**

**September 11, 1998, Changes: August 2013**

**Revision Chart**

*This chart contains a history of this document’s revisions. The entries below are provided solely for purposes of illustration. Entries should be deleted until the revision they refer to has actually been created.*

*The document itself should be stored in revision control, and a brief description of each version should be entered in the revision control system. That brief description can be repeated in this section. Revisions do not need to be described elsewhere in the document except inasmuch as they explain the development plan itself.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| Draft | TBD | Initial draft created for distribution and review comments | TBD |
| Preliminary | TBD | Second draft incorporating initial review comments, distributed for final review | TBD |
| Final | TBD | First complete draft, which is placed under change control | TBD |
| Revision 1 | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| Revision 2 | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| etc. | TBD | TBD | TBD |

**Contents**

**1.** **Introduction**

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

**2.** **Overall Description**

2.1 Product Perspective

*2.1.1* *System Interfaces*

*2.1.2* *User Interfaces*

*2.1.3* *Hardware Interfaces*

*2.1.4* *Software Interfaces*

*2.1.5* *Communications Interfaces*

*2.1.6* *Memory Constraints*

*2.1.7* *Site Adaptation Requirements*

2.2 User Characteristics

2.3 Constraints

2.4 Assumptions and Dependencies

**3.** **Specific Requirements**

3.1 External Interface Requirements

*3.1.1* *User Interfaces*

*3.1.2* *Hardware Interfaces*

*3.1.3* *Software Interfaces*

*3.1.4* *Communications Interfaces*

3.2 Software Product Features

*3.2.1* *Feature 1*

3.3 Performance Requirements

3.4 Software System Attributes

*3.4.1* *Reliability*

*3.4.2* *Availability*

*3.4.3* *Security*

*3.4.4* *Maintainability*

*3.4.5* *Portability*

3.5 Logical Database Requirements

3.6 Other Requirements

**4.** **Appendices**

# Introduction

## Purpose

*Στα πλαίσια του μαθήματος Τεχνολογία Λογισμικού μας ανατέθηκε να αναπτύξουμε ένα σύστημα αναφοράς ηλεκτρονικών εγκλήματων και της προώθησης τους στις κατάλληλες υπηρεσίες και αρχές. Το σύστημα θα πρέπει να παρέχει την δυνατότητα στους χρήστες1 να μπορούν να επικοινωνούν μέσω της ιστοσελίδας που θα αναπτυχθεί, με τους αρμόδιους για την αναφορά ενός περιστατικού. Ο σκοπός του παρών εγγράφου είναι η εύρεση και η ορθή καταγραφή των απαιτήσεων του πιο πάνω συστήματος. Το έγγραφο αυτό απευθύνεται στον πελάτη (CNTI2) καθώς και στο development team3.*

## Scope

*Identify the software product(s) to be produced by name. Explain what the products will and will not do. Describe how the software will be used, and identify relevant benefits,* *objectives, and goals.*

*Οι χρήστες του συστήματος όπως προαναφέρθηκε θα μπορούν να αναφέρουν ένα ηλεκτρονικό έγκλημα ή να αιτηθούν κάποιου είδους βοήθεια.*

*Το σύστημα θα περιλαμβάνει 2 κύριες λειτουργίες:*

1. ***Hotline***

*Ο σκοπός αυτού του υποσυστήματος είναι η αναφορά σοβαρών περιστατικών και η παραπομπή τους στο τμήμα ηλεκτρονικού εγλήματος της αστυνομίας Κύπρου. Τα περιστατικά ανήκουν στις εξής κατηγορίες: Child Sexual abuse Material, Online Grooming, Black mail. Οι χρήστες θα μπορούν μέσω τις ιστοσελίδας να καταγγέλουν ένα περιστατικό είτε ανώνυμα είτε δίνοντας στοιχεία επικοινωνίας. Ο κύριος λόγος ύπαρξης του Hotline, είναι η παροχή ανωνυμίας κατά την αναφορά της υπόθεσης σε αντίθεση με την απευθείας αναφορά τους στην αστυνομία όπου θα χρειαστεί κατάθεση στοιχείων. Με αυτόν τον τρόπο το σύστημα παροτρείνει τους χρήστες να κάνουν την καταγγελία χωρίς κανένα δισταγμό. Στόχος του Hotline, είναι η έγκαιρη αναφορά της υπόθεσης στην αστυνομία.*

*Παράλληλα θα γίνεται καταγραφή , η κατηγορία και ημερομηνία*

1. ***Helpline***

*Παροχή βοήθειας σε θέματα: Cyberbullying, Change settings, Commercial risks/threats, Excessive use, Grooming, Hacking, Identity Thefts, Other potentially harmful content, Online reputation, Phishing, Privacy, Sexting, Sexual harassment, Spam, Technical settings, Unsolicited contact from strangers.*

*Hotline Για σοβαρής μορφής περιστατικά*

*Child Sexual abuse Material(Online Grooming)*

*Black mail*

*Helpline*

*Για πιο ήπιας μορφής περιστατικά*

*Cyber-bulling*

*Online hate*

## Definitions, Acronyms, and Abbreviations

*Define all terms, acronyms, and abbreviations used in this document.*

***Terms:***

*1.Χρήστης - Το άτομο το οποίο θέλει να αναφέρει ένα περιστατικό ηλεκτρονικού εγκλήματος*

***Acronyms:***

*2. CNTI :Cyprus Neuroscience & Technology Institute*

*Email : Electronic mail*

***Abbreviations:***

*Development Team:*

## References

*List all the documents and other materials referenced in this document. This section is like the bibliography in a published book.*

*[1] Hotlines: Responding to reports of illegal online content, October 2013*

[2] <http://www.cyberethics.info/helpline-en.html>

# Overall Description

*In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements.*

*No text is necessary between the heading above and the heading below unless otherwise desired.*

## Product Perspective

*This section should place the product in perspective with other related products. If the product is independent and self-contained, state that here. Otherwise, identify interfaces between the product and related systems.*

### System Interfaces

*List each system interface and identify the related functionality of the product.*

### User Interfaces

*Specify the logical characteristics of each interface between the software product and its users (e.g., required screen formats, report layouts, menu structures, or function keys).*

*Specify all the aspects of optimizing the interface with the person who must use the system (e.g., required functionality to provide long or short error messages). This could be a list of do’s and don’ts describing how the system will appear to the user.*

### Hardware Interfaces

*Specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols.*

### Software Interfaces

*Specify the use of other required software products (e.g., a DBMS or operating system), and interfaces with other application systems. For each required software product, provide the following:*

* *Name*
* *Mnemonic*
* *Specification Number*
* *Version Number*
* *Source*

*For each interface, discuss the purpose of the interfacing software, and define the interface in terms of message format and content. For well-documented interfaces, simply provide a reference to the documentation.*

### Communications Interfaces

*Specify any interfaces to communications such as local area networks, etc.*

### Memory Constraints

*Specify any applicable characteristics and limits on RAM, disk space, etc.*

### Site Adaptation Requirements

*Define requirements for any data or initialization sequences that are specific to a given site, mission, or operational mode. Specify features that should be modified to adapt the software to a particular installation.*

## User Characteristics

*Describe the general characteristics of the intended users, including*

* *educational level*
* *experience*
* *technical expertise*

## Constraints

*Describe any other items that will constrain the design options, including*

* *regulatory policies*
* *hardware limitations*
* *interfaces to other applications*
* *parallel operation*
* *audit functions*
* *control functions*
* *higher-order language requirements*
* *signal handshake protocols*
* *reliability requirements*
* *criticality of the application*
* *safety and security considerations*

## Assumptions and Dependencies

*List factors that affect the requirements. These factors are not design constraints, but areas where future changes might drive change in the requirements.*

# Specific Requirements

*This section should describe all software requirements at a sufficient level of detail for designers to design a system satisfying the requirements and testers to verity that the system satisfies requirements.*

*Every stated requirement should be externally perceivable by users, operators or other external systems.*

*At a minimum, these requirements should describe every input into the software, every output from the software, and every function performed by the software in response to an input or in support of an output.*

*All requirements should be uniquely identifiable (e.g., by number).*

*No text is necessary between the heading above and the heading below unless otherwise desired.*

## Software Product Features

### Feature 1

*Repeat subsections at this level and below for each feature. Each feature may also be described through a user story that will be broken down to functional requirements. You can find some examples in the Wikipedia article (https://en.wikipedia.org/wiki/User\_story).*

#### Purpose

#### Associated Functional Requirements

##### Functional Requirement 1

*Repeat subsections at this level and below for each associated functional requirement.*

*Each functional requirement may be described in natural language, pseudocode, or in four subsections as follows. Functional requirements include:*

* *validity checks on inputs*
* *exact sequencing of operations*
* *responses to abnormal situations, including error handling and recovery*
* *effects of parameters*
* *relationships of inputs to outputs, including input/output sequences and formulas for input to output conversion*

###### Introduction

###### Inputs

###### Processing

###### Outputs

## Performance Requirements

*Specify static and dynamic numerical requirements placed on the software or on human interaction with the software.*

*Static numerical requirements may include the number of terminals to be supported, the number of simultaneous users to be supported, and the amount and type of information to be handled.*

*Dynamic numerical requirements may include the number of transactions and tasks and the amount of data to be processed within certain time period for both normal and peak workload conditions.*

*All of these requirements should be stated in measurable form.*

## Software System Attributes

*The following items provide a partial list of system attributes that can serve as requirements that should be objectively verified.*

### Reliability

*Specify the factors needed to establish the software’s required reliability.*

### Availability

*Specify the factors needed to guarantee a defined level of availability.*

### Security

*Specify the factors that will protect the software from accidental or malicious access, misuse, or modification. These factors may include:*

* *cryptography*
* *activity logging*
* *restrictions on intermodule communications*
* *data integrity checks*

### Maintainability

*Specify attributes of the software that relate to ease of maintenance. These requirements may relate to modularity, complexity, or interface design. Requirements should not be placed here simply because they are thought to be good design practices.*

### Portability

*Specify attributes of the software that relate to the ease of porting the software to other host machines and/or operating systems.*

## Logical Database Requirements

*Specify the requirements for any information that is to be placed into a database, including*

* *types of information used by various functions*
* *frequency of use*
* *accessing capabilities*
* *data entities and their relationships*
* *integrity constraints*
* *data retention requirements*

*Please include also a database diagram (E-R diagram) if available.*

## Other Requirements

# Appendices

*Include supporting detail that would be too distracting to include in the main body of the document.*