  
   
Software Requirements   
 Specification (SRS)

Guardian Information Security Management Application

Version 1.0

Prepared by: Jordan Carter

Group 9

<Group Name>

|  |  |  |
| --- | --- | --- |
| * Jordan Carter |  | <e-mail> |
| * Dominique Frazier |  | <e-mail> |
| * Andrew Williams |  | <e-mail> |
| * Dylan Ellington |  | <e-mail> |
| * Chase Moore * Lucas D’Andrea |  | <e-mail> |

|  |  |
| --- | --- |
| Instructor: | Dr. Kamran Sartipi |
| Course: | CSCI 3030 (Software Engineering I) |
| Date: | <place the date of submission here> |

Contents

Revisions iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3 Specific Requirements 5

3.1 External Interface Requirements 5

3.2 Functional Requirements 6

3.3 Behaviour Requirements 6

4 Other Non-functional Requirements 7

4.1 Performance Requirements 7

4.2 Safety and Security Requirements 7

4.3 Software Quality Attributes 7

5 Other Requirements 8

Appendix A – Data Dictionary 9

Appendix B - Group Log 10

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| Initial Draft  1.0 | Jordan Roger Carter | Initial SRS based on known data at the beginning of the project. | 00/00/00 |

# *<In this template you will find text bounded by the “<>” symbols. This text appears in italics and is intended to guide you through the template and provide explanations regarding the different sections in this document. There are two types of comments in this document. These comments that are in black are intended specifically for that course. These comments that are in blue are more general and apply to any SRS. Please, make sure to delete all of the comments before submitting the document.*

# *The explanations provided below, do not cover all of the material, but merely, the general nature of the information you would usually find in SRS documents. It is based on the IEEE requirements and was adapted specifically for the needs of Software Engineering courses. Most of the sections in this template are required sections, i.e. you must include them in your version of the document. Failure to do so will result in marks deductions. Optional sections will be explicitly marked as optional. If you have any questions regarding this document please refer to the MiniThermostat SRS example on the course web-site.>*

# Introduction

*<TO DO: Please provide a brief introduction to your project and a brief overview of what the reader will find in this section.>*

## Document Purpose

This SRS (Revision 1.0) is for the Guardian Information Security Management Application version 1.0 to specify the Software Requirements of the application in its entirety including the Asset management, Risk Management, and Report Generation modules of the Guardian Information Security Management Application.

## Product Scope

The Guardian Information Security Management Application will consist of Asset Management, Risk Management and Report Generation. The Asset management module will allow organizations to input assets into the system for storage along with associated information necessary for information security management purposes. The associated information will include individual asset identification information along with group identification information. The groups of assets can then be encompassed with the appropriate risk management policy and procedure to decrease any tedious operations on multiple assets. Grouping can then be given a hierarchal structure to allow multiple policies and procedures adjustments to occur in overlapping structures.

The Risk Management will allow companies to create standardized adjustments to asset risk ratings based on risk management procedures and policies stored in the application and apply them to groups or individual assets. The uniform classification of assets within the application will allow companies to remove any interdepartmental ambiguity and redundancy within the information security management process. The application will set all new assets up with the appropriate mapping of security requirements based on the asset identification information. The universal asset list will ensure standardized format for asset tracking and reports of asset coverage by information security policy can be generated by the application to share with upper management and the board. The report types can be selected to provide information pertinent to the distribution group.

## Intended Audience and Document Overview

This document is intended to provide a reference for the application client and developers.

<INSERT A DOCUMENT OVERVIEW HERE PRIOR TO SUBMISSION>

<JORDAN CARTER>

## Definitions, Acronyms and Abbreviations

GISMA – Guardian Information Security Management Application

IT – Information Technology

IS – Information Security

CIA – Confidentiality, Integrity, and Availability

## Document Conventions

<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1” margins found in this template. For Section and Subsection titles please follow the template.

TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

<JORDAN CARTER>

## References and Acknowledgments

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.

TO DO: Use the standard IEEE citation guide for this section. An example citation guide is posted for you on the website.>

<INSERT APIs USED TO REFERENCE HERE – JC>

WATERLOCK

SAILS

# Overall Description

## Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible.

TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the environment and in what context it is being used.>

Many information security management teams are currently using a combination of the Microsoft Office suite to accomplish task, but without utilizing a central system this leads to non-standardized methods of tracking and representing data. This approach does not offer the benefit of a centralized system which can create uniformity and automate redundancies present in the current information security management practices.

## Product Functionality

**(Optional)** Provide a Data Flow Diagram of the system to show how these functions relate to each other.>

* Individual User log on
* Asset Creation/Deletion/Modification
* Risk Policy and Procedure Creation/Deletion/Modification
* Group Creation
* Risk Policy and Procedure mapping
* Report Generation

## Users and Characteristics

<Identify the various users that you anticipate will use this product. Users may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience.

TO DO:

1. Describe the pertinent characteristics of each user. Certain requirements may pertain only to certain users.

3. Distinguish the most important users for this product from those who are less important to satisfy.>

Information Security Management – These are the administrators of information security policy. They have control over defining security requirements and risk reduction procedures as well as controlling what groups and assets the policy/procedure will be applied to within the application.

Other IT management – depending on organizational structure the overall procurement and maintenance of assets within the organization. Sometimes it may be convenient to give asset management options to individuals without providing the risk management editing rights.

IT/IS Personnel – Members of the information security or information technology teams that would find the ability to view the application data without the ability to edit.

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface

TO DO: As stated above, in at least one paragraph, describe the environment your system will have to operate in. Make sure to include the minimum platform requirements for your system. >

Environment – windows 10, 1Ghz or faster processor

<Dylan Ellington>

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).

TO DO: In this section you need to consider all of the information you gathered so far, analyze it and correctly identify at least 5 constraints.>

<Dylan Ellington>

## User Documentation

Online Training – There will be online training tutorials provided on our website as well as a quick help guide.

User Manual - The User manual can be found on our website at <insert website> in a printable format.

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

TO DO: Provide a short list of some major assumptions that might significantly affect your design. For example, you can assume that your client will have 1, 2 or at most 50 Automated Banking Machines. Every number has a significant effect on the design of your system. >

<Andrew Williams>

<LIST APIs>

<Frameworks>

<If the server is updated or changed the software may not work properly>

# Specific Requirements

## External Interface Requirements

### User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., Cancel) that will appear on every screen, error message display standards, and so on. Define the software components for which a user interface is needed.

TO DO: The least you can do for this section is to describe in words the different User Interfaces and the different screens that will be available to the user. Those who will be able to provide optional Graphical User Interface screenshots, will be rewarded by extra marks.>

**Asset Management Interface**

1. Add individual asset – this will allow the user to add individual organization assets. The User will provide an asset ID, Description, Confidentiality rating, and Integrity rating, and the software will generate a record of the asset for the asset management view.
2. Import Asset list – This will allow the user to import a .csv file of assets into the application by using the standard format provided in the user manual.
3. Edit/Delete Asset – This will allow the user to edit information provided for a previously existing asset or delete the record entirely
4. Create Asset group – This will allow the user to create an asset group by providing a title and the associated assets to be grouped
5. Edit/Delete Asset group – This will allow the user to edit the information of a previously existing asset group or delete the record entirely

**Risk Management Interface**

1. Add Risk Mitigation Strategy – This will allow the user to create a risk mitigation strategy. The user will provide information about the strategy, including a description, CIA adjustments, and applied groups.
2. Edit/Delete Risk Mitigation Strategy – This will allow the user to edit or delete a previously existing strategy

**Report Generation Interface**

1. Option Selection – This will allow the user to select the desired options to be generated in the report.
2. Generate report – This will generate a text document based on the options selected and export it to the system.

### Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.

TO DO: Please provide a short description of the different hardware interfaces. If you will be using some special libraries to communicate with your software mention them here. In case you have more than one hardware interface divide this section into subsections.>

<Andrew Williams>

<insert the windows library to communicate through the mouse and keyboard.>

### Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>

<Dylan Ellington>

<insert information on database MySQL , NodeJS web api, sails vs 1.0 beta, WPF,>

### Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >

<Chase Moore>

Restful API over HTTP

## Functional Requirements

*< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions with specific explanations regarding every function.*

*TO DO: Brake the functional requirements to several functional areas and divide this section into subsections accordingly. Provide a detailed list of all product operations related to these functional areas.*

*<USE THIS FOLLOWING TEXT FROM SECTION 2.2 TO WRITE A FULL DESCRIPTION OF THE FUNCTIONAL REQUIREMENTS*

* Individual User log on
* Asset Creation/Deletion/Modification
* Risk Policy and Procedure Creation/Deletion/Modification
* Group Creation
* Risk Policy and Procedure mapping
* Report Generation

*>*

## Behaviour Requirements

### Use Case View

<A use case defines a goal-oriented set of interactions between external actors and the system under consideration. Since sometimes we will not be able to specify completely the behaviour of the system by just State Diagrams, we use use-cases to complete what we have already started in section 3.3.1.

TO DO: Provide a use case diagram which will encapsulate the entire system and all possible actors. Do not include detailed use case descriptions (these will be needed when you will be working on the Test Plan), but make sure to include a short description of what every use-case is, who are the actors in your diagram. For more information please refer to your UML guide and the MiniThermostat SRS example file.>

<Dominque Frazier insert here>

# Non-functional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide at least 5 different performance requirements based on the information you collected from the client. For example you can say “1. Any transaction will not take more than 10 seconds, etc…>

<Dominque Frazier>

## Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

* Provide at least 3 different safety requirements based on your interview with the client or, on your related research about the project, and again you need to be creative here.
* Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>

<Andrew Williams>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, & etc… Do not forget to include such attributes as the design for change. Please note that you need to include at least 2 quality attributes, but it is the mere minimum and it will not receive the full marks.>

<Chase Moore>

# Other Requirements

<This section is **Optional.** Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A – Data Dictionary

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>