***Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

**Software Requirements**

**Specification**

**for**

**<Project>**

**Version 1.0 approved**

**Prepared by <author>**

**<organization>**

**<date created>**

***Software Requirements Specification for <Project> Page ii***

Table of Contents

[1. Introduction 4](#_Toc505982074)

[1.1. Purpose 4](#_Toc505982075)

[1.2. Document Conventions 4](#_Toc505982076)

[1.3. Intended Audience and Reading Suggestions 4](#_Toc505982077)

[1.4. Project Scope 4](#_Toc505982078)

[1.5. References 5](#_Toc505982079)

[2. Overall Description 6](#_Toc505982080)

[2.1. Product Perspective 6](#_Toc505982081)

[2.2. Product Features 6](#_Toc505982082)

[2.3. User Classes and Characteristics 6](#_Toc505982083)

[2.4. Operating Environment 6](#_Toc505982084)

[2.5. Design and Implementation Constraints 6](#_Toc505982085)

[2.6. User Documentation 6](#_Toc505982086)

[2.7. Assumptions and Dependencies 6](#_Toc505982087)

[3. System Features 8](#_Toc505982088)

[3.1. Login function 8](#_Toc505982089)

[3.1.1. Description and Priority 8](#_Toc505982090)

[3.1.2. Stimulus/Response Sequences 8](#_Toc505982091)

[3.1.3. Functional Requirements 8](#_Toc505982092)

[3.2. Staff Control Panel 8](#_Toc505982093)

[3.2.1. Description and Priority 8](#_Toc505982094)

[3.2.2. Stimulus/Response Sequences 8](#_Toc505982095)

[3.2.3. Functional Requirements 8](#_Toc505982096)

[3.3. Admin Control Panel 8](#_Toc505982097)

[3.3.1. Description and Priority 8](#_Toc505982098)

[3.3.2. Stimulus/Response Sequences 8](#_Toc505982099)

[3.3.3. Functional Requirements 8](#_Toc505982100)

[4. External Interface Requirements 9](#_Toc505982101)

[4.1. User Interfaces 9](#_Toc505982102)

[4.2. Hardware Interfaces 9](#_Toc505982103)

[4.3. Software Interfaces 9](#_Toc505982104)

[4.4. Communications Interfaces 9](#_Toc505982105)

[5. Other Nonfunctional Requirements 9](#_Toc505982106)

[5.1. Performance Requirements 9](#_Toc505982107)

[5.2. Safety Requirements 9](#_Toc505982108)

[5.3. Security Requirements 10](#_Toc505982109)

[5.4. Software Quality Attributes 10](#_Toc505982110)

[6. Other Requirements 10](#_Toc505982111)

[Appendix A: Glossary 10](#_Toc505982112)

[Appendix B: Analysis Models 10](#_Toc505982113)

[Appendix C: Issues List 10](#_Toc505982114)

# Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the functionalities of the Warehouse Management Tool system. This document will cover each of the system’s intended features, as well as offer a preliminary glimpse of the software application’s User Interface (UI). The document will also cover hardware, software, and various other technical dependencies.

## Document Conventions

This document is unique. Its scope is to describe the requirements of the Warehouse Management Tool system and the interfaces for it. This document has been created before the actual program is constructed, hence is it important to note that future changes to the program should be included in this requirement specification document, in order to maintain its usefulness. Since the program has not been implemented yet, this document can be used as a manual for development of the system.

## Intended Audience and Reading Suggestions

This document may be used by the staff or clients of the Warehouse Management Tool system. They can easily understand the features and the operations of this system. They can also be informed of all the possibilities the program provides. Users can get a general image of the program and use it with better ease and understanding. This document could be studies by the programmers, so they have the chance to identify and alter possible problems that needs improvement. Also, they can fully understand the nature of the program and implement new features to it. It is recommended to study the document carefully in order to fully understand the program and features as well as identifying errors. Basically, this document helps users and testers to understand how the Warehouse Management Tool system is supposed to work and also to identify bugs and errors.

## Project Scope

Project will be developed under the scenario of Warehouse Management Tool System, WM tool system in short. Web portal for this domain is currently not specified. An effective Warehouse Management Tool is essential for streamlining process, long range planning and optimization. It gives a better understanding of day-to-day operation with its summary information of how a warehouse is performing. The managers could use this data for making critical management decisions.

It is critical for the warehouse management to have an effective system in place due to various reasons such as security, accuracy, availability and so on. We are requested to use a test-driven approach to develop the warehouse management tool which needs to meet the following requirements.

List of features include:

* Ability to add/update records on the incoming stock and outgoing stock easily.
* Track the stock in accordance to their category and sub-category.
* Ability to search and display available stock for a particular category and/or sub-category.
* Ability to search and display stock in accordance to price range and quantity in ascending and descending order.
* Daily, weekly or monthly summary report of total incoming and outgoing stock details.
* Provide login authentication process.
* Login security feature with option of encrypting the data stored.
* If there are three unsuccessful login attempts, the record will be marked as “locked” and one is not allowed to login to the system.
* The system shall encrypt and decrypt the password.
* Provide stock item alerts when it drops below a threshold set previously.

## References

Referenced to CSCI222\_Assignment1 words document for the features included.

# 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. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

## Product Features

*<Summarize the major features the product contains or the significant functions that it performs or*

*lets the user perform. Details will be provided in Section 3, so only a high level summary is*

*needed here. Organize the functions to make them understandable to any reader of the SRS. A*

*picture of the major groups of related requirements and how they relate, such as a top level data*

*flow diagram or a class diagram, is often effective.>*

## User Classes and Characteristics

*<Identify the various user classes that you anticipate will use this product. User classes may be*

*differentiated based on frequency of use, subset of product functions used, technical expertise,*

*security or privilege levels, educational level, or experience. Describe the pertinent characteristics*

*of each user class. Certain requirements may pertain only to certain user classes. Distinguish the*

*favored user classes from those who are less important to satisfy.>*

## 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.>*

## Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might*

*include: corporate or regulatory policies; 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).>*

## User Documentation

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that*

*will be delivered along with the software. Identify any known user documentation delivery formats*

*or standards.>*

***Software Requirements Specification for <Project> Page 3***

## 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, unless they are already documented elsewhere (for example, in the vision and*

*scope document or the project plan).>*

# System Features

## Login function

### Description and Priority

Allows the user to log in then proceeds to authenticate the user before directing them to the appropriate control panel based on their role. High priority.

### Stimulus/Response Sequences

* Login
  + Wrong username/ password
  + Account locked out
* Role Validation
  + Staff
  + Admin

### Functional Requirements

* REQ-1: Login function must be able to authenticate user based on entered username and password.
* REQ-2: Login function must be able to encrypt password when authenticating user.
* REQ-3: Login function must be able to read the user’s roles and restrict access to certain functions.
* REQ-4: Users with admin roles may not be locked out of their accounts.

## Staff Control Panel

### Description and Priority

Used primarily by staff to add and remove stocks from database. They can also display the summary report of a period of time. High priority.

### Stimulus/Response Sequences

* Search Product
* Edit Product
* Add Product
* Remove Product
* Summary Report
* Retrieve Product

### Functional Requirements

* REQ-1: Must be able to encrypt and decrypt the database file when manipulating fields.

## Admin Control Panel

### Description and Priority

Used by staff with admin roles to manage accounts. Includes the ability to Medium Priority.

### Stimulus/Response Sequences

* Create account
* Reset Account
* Delete Account
* View Account

### Functional Requirements

* REQ-1: Must only be accessible by users with admin roles

# External Interface Requirements

## User Interfaces

<TODO>

## 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, and*

*communication protocols to be used.>*

## Software Interfaces

*<Describe the connections between this product and other specific software components (name*

*and version), including databases, operating systems, 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.*

*Refer to documents that describe detailed application programming interface protocols. 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.>*

## 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.>*

***Software Requirements Specification for <Project> Page 5***

# Other Nonfunctional 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.>*

## Safety 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.>*

## Security Requirements

*<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. Refer to any external policies or regulations containing security issues that affect the*

*product. Define any security or privacy certifications that must be satisfied.>*

## 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.>*

# Other Requirements

*<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: Glossary

*<Define all the terms necessary to properly interpret the SRS, including acronyms and*

*abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire*

*organization, and just include terms specific to a single project in each SRS.>*

***Software Requirements Specification for <Project> Page 6***

# Appendix B: Analysis Models

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams,*

*state-transition diagrams, or entity-relationship diagrams*.>

# Appendix C: Issues List

*< This is a dynamic list of the open requirements issues that remain to be resolved, including*

*TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>*