**Software Requirements**

**Specification**

**for**

**Data Corruption Back-up & Recovery**

**Version 1.0 approved**

**Prepared by Data Corruption Back-up & Recovery Team**

**Asia Pacific College**

**November 23, 2016**

***Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table of Contents ..........................................................................................................................** | | | **ii** |
| **Revision History ............................................................................................................................** | | | **ii** |
| **1.** | **Introduction ..............................................................................................................................** | | **1** |
|  | 1.1 | Purpose ............................................................................................................................................ | 1 |
|  | 1.2 | Document Conventions .................................................................................................................... | 1 |
|  | 1.3 | Intended Audience and Reading Suggestions .................................................................................. | 1 |
|  | 1.4 | Product Scope .................................................................................................................................. | 1 |
|  | 1.5 | References ........................................................................................................................................ | 1 |
| **2.** | **Overall Description ..................................................................................................................** | | **2** |
|  | 2.1 | Product Perspective ......................................................................................................................... | 2 |
|  | 2.2 | Product Functions ............................................................................................................................ | 2 |
|  | 2.3 | User Classes and Characteristics ..................................................................................................... | 2 |
|  | 2.4 | Operating Environment .................................................................................................................... | 2 |
|  | 2.5 | Design and Implementation Constraints .......................................................................................... | 2 |
|  | 2.6 | User Documentation ........................................................................................................................ | 2 |
|  | 2.7 | Assumptions and Dependencies ...................................................................................................... | 3 |
| **3.** | **External Interface Requirements ...........................................................................................** | | **3** |
|  | 3.1 | User Interfaces ................................................................................................................................. | 3 |
|  | 3.2 | Hardware Interfaces ......................................................................................................................... | 3 |
|  | 3.3 | Software Interfaces .......................................................................................................................... | 3 |
|  | 3.4 | Communications Interfaces ............................................................................................................. | 3 |
| **4.** | **System Features ........................................................................................................................** | | **4** |
|  | 4.1 | System Feature 1 .............................................................................................................................. | 4 |
|  | 4.2 | System Feature 2 (and so on) ........................................................................................................... | 4 |
| **5.** | **Other Nonfunctional Requirements .......................................................................................** | | **4** |
|  | 5.1 | Performance Requirements .............................................................................................................. | 4 |
|  | 5.2 | Safety Requirements ........................................................................................................................ | 5 |
|  | 5.3 | Security Requirements ..................................................................................................................... | 5 |
|  | 5.4 | Software Quality Attributes ............................................................................................................. | 5 |
|  | 5.5 | Business Rules ................................................................................................................................. | 5 |
| **6.** | **Other Requirements ................................................................................................................** | | **5** |
| **Appendix A: Glossary....................................................................................................................** | | | **5** |
| **Appendix B: Analysis Models .......................................................................................................** | | | **5** |
| **Appendix C: To Be Determined List ............................................................................................** | | | **6** |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
| Maria Letty Laureta | 11/23/16 | Created Document | 1.0 |
| Kent Michael Miculob | 12/01/16 | Document Revision | 1.1 |

**1. Introduction**

**1.1 Purpose**

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>*

The purpose of the Software Requirement System is to clearly define the system under development, Namely Hydra Data Protection Tool (HDPT). The HDPT is a Tool embedded in a flash drive, which the main purpose is to protect the data that is currently working on. And to mitigate the effects of data corruption to the end-users.

**1.2 Document Conventions**

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

*Upon reading this paper, technical terms might be encountered, the terms will be defined as the following:*

*Checksum: act of comparing the hash value of the file, to check for its integrity.*

*Hash Value: is a one-way encryption for the main purpose of maintaining the file integrity*

*Data Corruption: a state where your file gets unusable due to certain causes.*

*Flash drive: external storage device*

*End-Users: The Users of the final product.*

**1.3 Intended Audience and Reading Suggestions**

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

The intended audience of this document are the end-users which includes students and office workers. Other intended audience are the HDPT Project Manager, HDPT Project developer, HDPT Adviser.

**1.4 Product Scope**

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>*

*The product will be used to protect the files of the end-users from events of data corruption, and aims to be more efficient and helpful than the existence similar tools. The product’s protection will be mainly focused on the ones existing in the Back-up folder located at the flash drive.*

**1.5 References**

## 

This document follows documents for the product in projects2.apc.edu.ph/wiki and GitHub. The following are some of the references for this document:

* Change Management Plan
* Project Vision and Scope
* Statement of Work
* Quality Plan

All documents listed above were documented by the project team together.

**2. Overall Description**

**2.1 Product Perspective**

*The product is a tool that aims to help the end-users maintain a back-up data of their files, which some users find it a bit hassle to maintain back-up each and every time, and most often they do it casually. The product wants to maintain a back-up of their file without being a hassle for the end-users’ 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.>*

**2.2 Product Functions**

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) 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 object class diagram, is often effective.>*

The product itself has three major function which are:

* Real-time Back-up – Where the file is automatically backed up in real-time.
* Synchronization – Where the user is working on the file, the last activity of the user will automatically save on file itself
* Mobility – The user can bring the tool anywhere does it go.
* Corruption scanning – which detect if there is an existing corrupted data in the back-up folder.
* Corruption Scanning – The system will attempt to fix the corrupted data.

**2.3 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 most important user classes for this product from those who are less important to satisfy.>*

*The HDPT has two main users namely the normal users, and the privilege users. The normal users are allowed to scan and fix the identified corrupted data, they could also back-up their file anytime they want to. Privilege users are the only ones allowed to retrieve back-up data.*

**2.4 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.>*

*The system will operate in a windows based platform with a NTFS file architecture, the system will be stored in a flash drive, and the system will also use java platform and AutoIt Scripting tool.*

**2.5 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).>*

*Design and Implementation constrains were identified as the following:*

* The memory of the flash drive should at least less than 90 percent to attain the systems optimal performance
* The system will be using Java language.
* It is also assumed that the end-users will took the responsibility for protecting the hardware in cases where the flash drive itself became corrupted.

**2.6 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.>*

*This SRS document will be delivered along with the User Manual Document, which contain the steps on how to use the software, help and tutorial would also be included there.*

**2.7 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).>*

There are assumptions made during the creation of the project and it is necessary since it is out of the scope of the project but could probably have an impact to the system.

* It is assumed that the computers are properly equipped with anti-virus such that the storage device wouldn’t be affected by any malicious program.
* The user is responsible enough to protect the hardware, where the software is stored.

**3. External Interface Requirements**

**3.1 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., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

In our product system, the UI is where the user and system can interact with. The system has buttons where these buttons are made up of functionalities of scanning, memory checking, retrieve the back-up files, viewing logs, and downloading logs. The UI also shows the files are inside of the tool.

**3.2 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.>*

The hardware interface that can be involve in our project is the Flash Drive where the Flash Drive is a tool that we can embed our system in it.

**3.3 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.>*

In software interface, the team used Java Language to build the system also we used the Netbeans as a compiler where we can run the system prototype. The system requires

**4. System Features**

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

*The product has 3 main features that work together to protect the user’s data, namely Real-time back-up, Scan for corrupted data, Repair corrupted data. Some minor feature like being Handy and Mobile, would encourage the user to back-up, since it is easy to use and doesn’t distract the user from what his doing.*

**4.1 Real-Time Back-up**

*<Don’t really say “System Feature 1.” State the feature name in just a few words.>*

1. Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

*The Real-time back-up feature is highly prioritize, since the best way to protect each and everyone’s documents to data corruption, is through data redundancy, which makes this feature more important. It would also encourage people to back-up their file since the method is much easier to use from the end-users perspective.*

1. Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

*Real-time back-up: After the user execute a file, the system would duplicate the file, and would incrementally save during 2 seconds.*

*.*

1. Functional Requirements

*Conditions must be met, before the user can use the necessary functions, such as, the memory space should at least less than 90 percent full. So that optimum performance and real-time back-up is established*

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1: Storage Space is less than 90 percent

REQ-2: The user’s environment must be Windows OS with NTFS file structure

REQ-3: The Flash must also have a NTFS file structure.

## 4.2 Scan for Data corruption

4.2.1 Description and Priority

*This feature, would conduct a checksum on the file’s hash value, which would detect if data corruption has occurred.*

4.2.2 Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

*The user must execute the “Scan for Corruption” button before the system conducts checksum on the file’s hash value.*

*.*

4.2.3 Functional Requirements

REQ-1: Storage Space is less than 90 percent

REQ-2: The user’s environment must be Windows OS with NTFS file structure

REQ-3: The Flash must also have a NTFS file structure.

## 4.3 Scan for Data corruption

4.3.1 Description and Priority

*This feature, would conduct a checksum on the file’s hash value, which would detect if data corruption has occurred.*

4.3.2 Stimulus/Response Sequences

The user must have identified a corrupted data, and must execute the repair function.

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

*The user must execute the “Scan for Corruption” button before the system conducts checksum on the file’s hash value.*

*.*

4.3.3 Functional Requirements

REQ-1: Storage Space is less than 90 percent

REQ-2: The user’s environment must be Windows OS with NTFS file structure

REQ-3: The Flash must also have a NTFS file structure.

REQ-4: The Scan for Corruption must be issued before this can be executed.

REQ-5: There is a detected file corruption.

**5. Other Nonfunctional Requirements**

**5.1 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.>*

*The HDPT product requires that the storage device (flash drive) to be less than 90% full, since the system also occupied space in the flash drive. And real-time back-up cannot be issued if the flash drive is almost full.*

**5.2 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.>*

*Due to physical causes, the corruption might occur to the flash drive itself. The user must keep it safe as always to avoid this kind of scenario to happen.*

**5.3 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.>*

*upon retrieving the data, the user needs to acquire the password to authenticate whether he has an admin privilege or not.*

**5.4 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.>*

*The software helps the users to protect their data with ease, since some tools are too complicated for them to use. And some find back-up data as a chore that is often done once a week.*

**5.5 Business Rules**

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

**6. 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.>*

**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: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*