**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 |
| Maria Letty Laureta | 12/03/16 | Continue Document | 1.2 |

**1. Introduction**

**1.1 Purpose**

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**

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**

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**

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. **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.

**2.2 Product Functions**

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 Fix – The system will attempt to fix the corrupted data.

**2.3 User Classes and Characteristics**

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**

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**

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**

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**

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**

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**

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**

In software interface, the team used Java Language to build the system also we used the Netbeans IDE as a compiler where we can run the system prototype. Also, we used the AutoIt, where we can do the scripting.

**4. System Features**

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**

1. Description and Priority

The Real-time back-up feature is highly prioritized, 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

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

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

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 Fix 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

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

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.

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**

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**

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**

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**

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**

The following are the rules of each involved including the system itself.

* The user’s file is automatically back-up in Flash Drive
* The user can view and download its logs
* The user may retrieve the files from the back-up data.
* The system will scan for the corrupted text file

**6. Other Requirements**

There are no more required for the system. If there are bugs, it must check to the admin to fix the bugs of the system. Always keep the Flash Drive secure in order to keep data safe.

**Appendix A: Glossary**

Hash Value - is a numeric value of a fixed length that uniquely identifies data.

Real Time Back-Up - refers to backup of computer data by automatically saving a copy of every change made to that data, essentially capturing every version of the data that the user saves.

Java Language - a general-purpose, concurrent, strongly typed, class-based object-orientedlanguage*.*  
Netbeans IDE - a free and open source integrated development environment for application development on Windows, Mac, Linux, and Solaris operating systems.

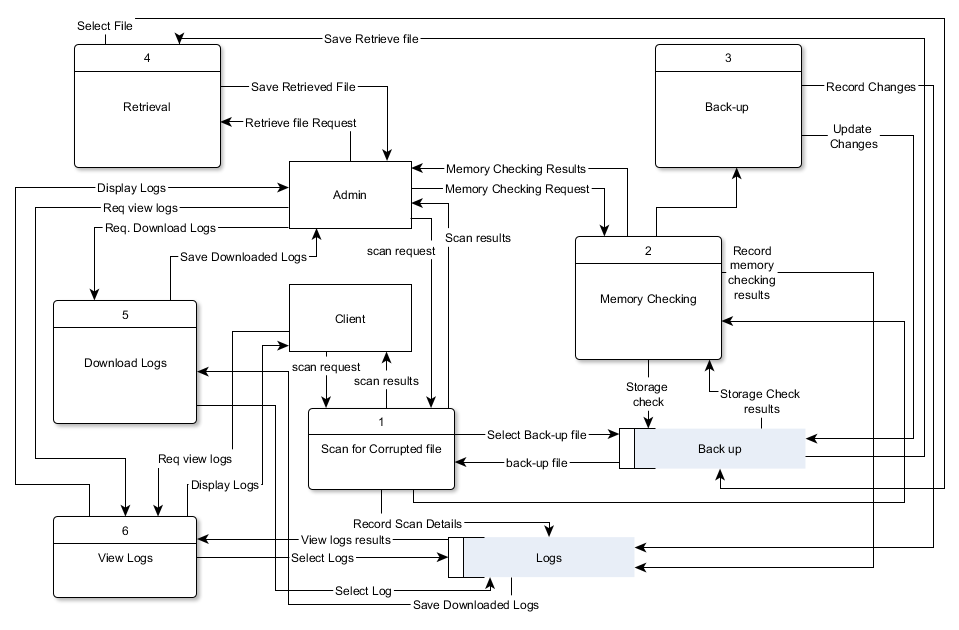
AutoIt - is a freeware BASIC-like scripting language designed for automating the Windows GUI and general scripting.

**Appendix B: Analysis Models**

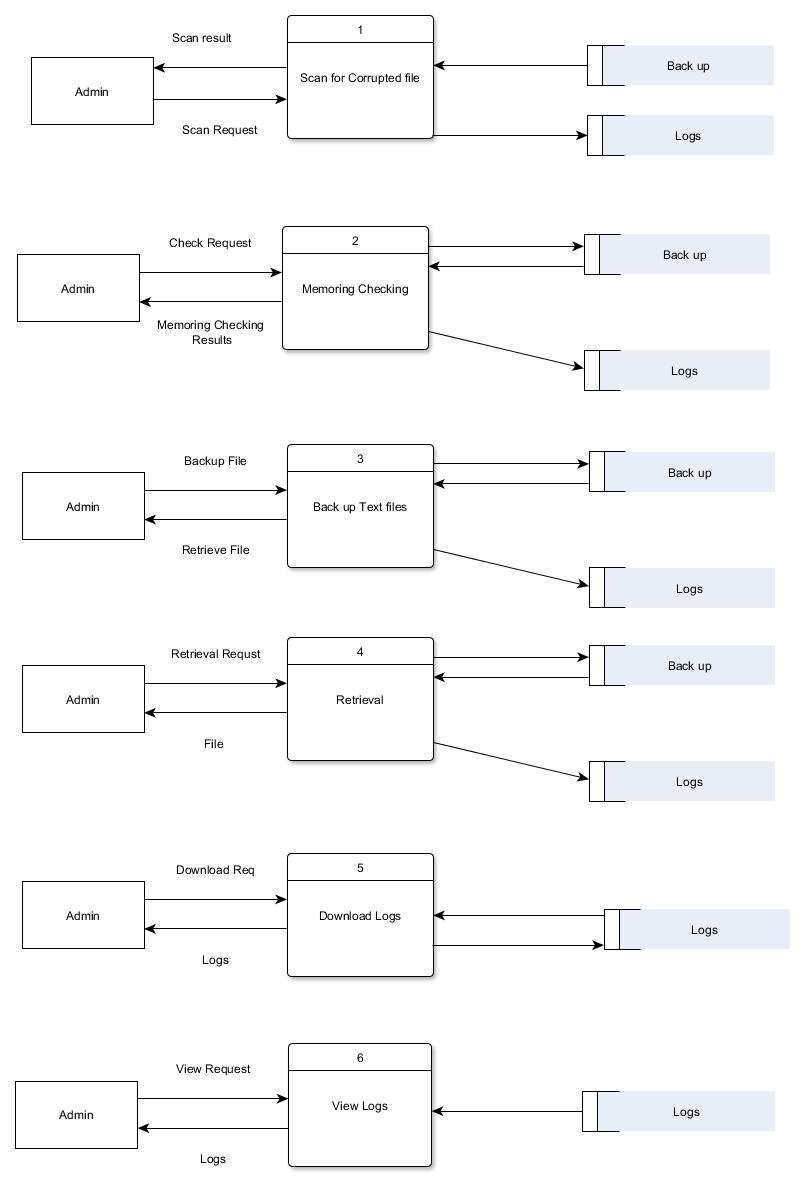
<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagram

Data Flow Diagram

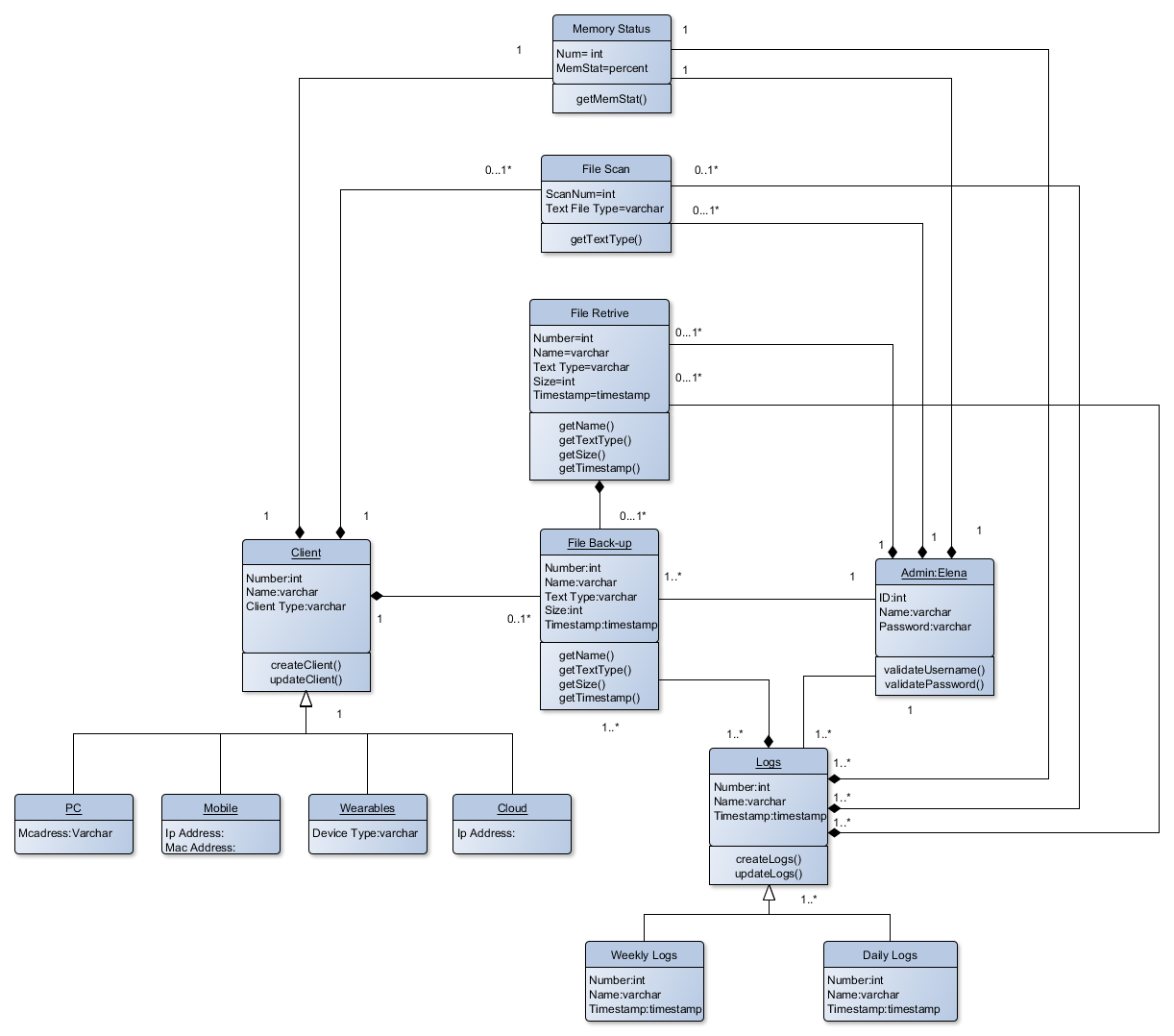
Level 0



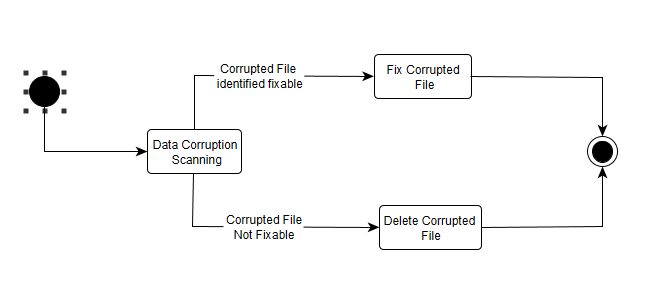
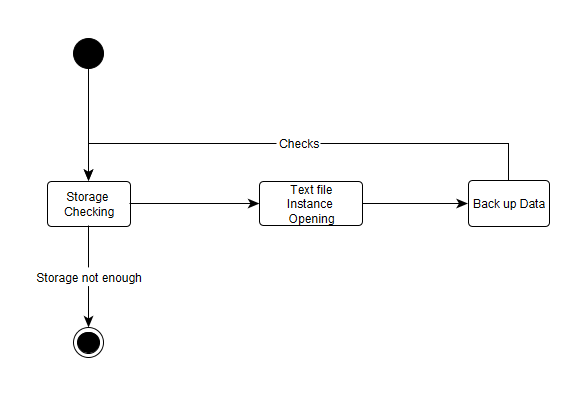
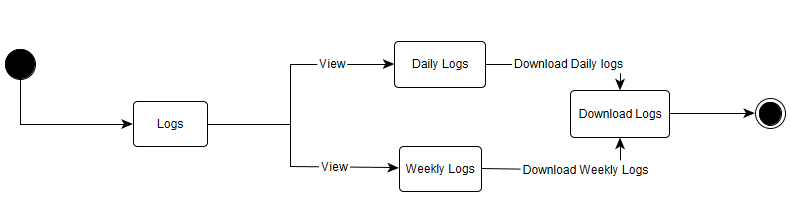
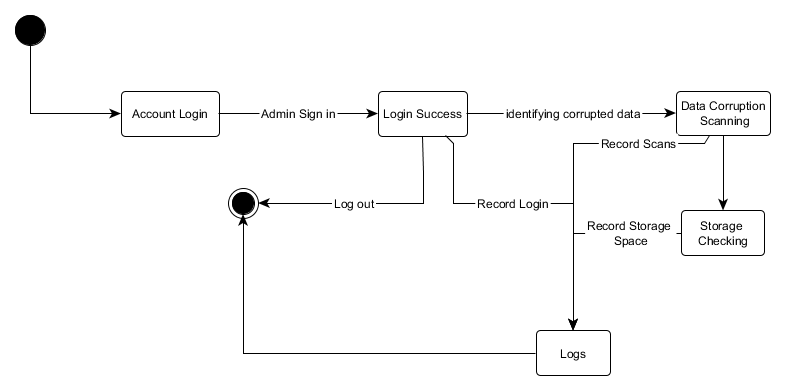
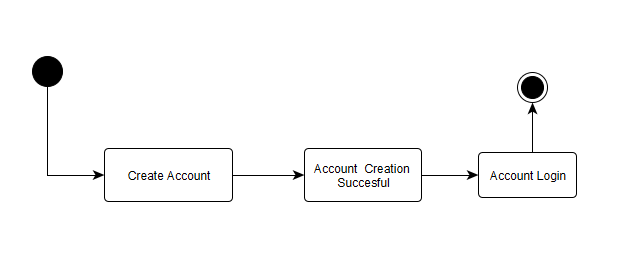
DFD Fragments



Class Diagram



State Diagram



Entity- Relationship Diagram

