Software Requirements

Specification

For

ExpirePDF



Version 1.0 Approved

July 9, 2010

**Prepared by:**

Ryan Ross, Henry Blum,

Bjorn Carandang,

and Rashid Al-Naemi

Drexel University

Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Comments** |
| 1.0 | 2010-7-15 | Initial version |

Table of contents

[1 Introduction 4](#_Toc267335172)

[1.1 Purpose 4](#_Toc267335173)

[1.2 Intended Audience 4](#_Toc267335174)

[1.3 Project Scope 4](#_Toc267335175)

[2 Overall Description 5](#_Toc267335176)

[2.1 Product Perspective 5](#_Toc267335177)

[2.1.1 System Interfaces 5](#_Toc267335178)

[2.1.2 User Interfaces 5](#_Toc267335179)

[2.1.3 Hardware Interfaces 5](#_Toc267335180)

[2.1.4 Software Interfaces 6](#_Toc267335181)

[2.1.5 Communications Interfaces 6](#_Toc267335182)

[2.2 Product Functions 6](#_Toc267335183)

[2.3 User Characteristics 6](#_Toc267335184)

[2.4 Design Constraints 6](#_Toc267335185)

[3 Functional Requirements 7](#_Toc267335186)

[3.1 Functional Requirements 7](#_Toc267335187)

[3.1.1 Creating an encrypted PDF 7](#_Toc267335188)

[3.1.2 Viewing an encrypted PDF 8](#_Toc267335189)

[4 Non-functional Requirements 9](#_Toc267335190)

[4.1 Installation Requirements 9](#_Toc267335191)

[4.2 Memory Requirements 9](#_Toc267335192)

[4.3 Help Requirements 9](#_Toc267335193)

[4.4 Portability Requirements 9](#_Toc267335194)

[5 User Interface 10](#_Toc267335195)

[6 System Evolution 11](#_Toc267335196)

[7 Glossary 12](#_Toc267335197)

[8 Index 13](#_Toc267335198)

# Introduction

## Purpose

This document specifies all the software requirements for the ExpirePDF, digital rights management system, version 1.0. These requirements directly relate to the functionalities, performance, constraints, attributes, and interfaces of the system.

## Intended Audience

The intended audience of this document exclusively includes the developers, the testers, and the end-users of the ExpirePDF system.

## Project Scope

ExpirePDF is a tool to control the access to a digital document by an end user. The concept for this project is to be able to lend a document with one person, and prevent them from sharing that document with others, or from keeping it forever. There are two primary goals of the ExpirePDF system. The first goal is to allow a publisher to set an expiration date for a PDF, after which the document should no longer be readable. The second goal is to provide publishers with a system for encrypting a PDF, such that its expiration date will not be tampered with. In order for these goals to be deployed effectively, a subsequent goal is to provide a system for the end user to interact with these encrypted PDFs.

In order to meet a contracted development schedule, this project only aims to create a DRM system capable of preventing the sharing of documents by end users without significant technical knowledge of computer software systems.

# Overall Description

## Product Perspective

Publishers need a way of maintaining and enforcing their copyright when dealing with digital content, which ExpirePDF aims to accomplish. As described in section 1.3 of this document, ExpirePDF will be designed to limit distribution of digital documents, specifically those in the PDF format. At present, there are no free and open solutions for PDF DRM. We accomplish this with our own system to reliably protect an independent party’s distributable documents, no matter where the document is.

### System Interfaces

The ExpirePDF software integrates an existing encryption system to provide its cryptography functions.

### User Interfaces

The two ExpirePDF clients have simple interfaces that value ease of use. Individual PDFs are handled by both clients in ways that make sense in the scheme of property protection. The ExpirePDF Publisher and the Viewer facilitate quick access to all primary functions. However, the number of functions presented in both the publisher and viewer is kept as low as possible while still providing the primary use for each program.

The interface for the publisher is structured to allow for quick management of protected documents, and the creation of new protected documents. The viewer interface provides quick access to a file selector and information on the current document.

### Hardware Interfaces

The ExpirePDF clients run on any computer hardware meeting the following criteria:

* Capable of running a Microsoft Windows® operating system, version XP or newer.
* Includes common computing components, such as monitor, a keyboard and a pointing device.
* Includes writable volatile storage.

### Software Interfaces

The ExpirePDF software integrates some external software to provide functionality.

**Distribution**: The ExpirePDF software exists as a standalone software package with appropriate functionality (the Viewer and Publisher programs packaged together). A document end-user often has no need for the Publisher package, so the ExpirePDF Viewer also exists as a standalone application.

**Client**: The ExpirePDF software (both clients) interfaces with the user’s operating system libraries to accomplish its goals. Particularly, the programs interface with .NET Framework 3.5 for graphical user interface components.

### Communications Interfaces

No direct communication will exist between any main server and either client (Viewer or Publisher). The only need for communications are as a possible means to transfer documents to the end user from online sources, or to aquire the ExpirePDF software.

## Product Functions

The ExpirePDF system will provide the following functions:

1. Creation of DRM-controlled PDFs for redistribution by a Publisher
2. Opening and reading of DRM-controlled PDFs by a Viewer
3. Control of PDFs on the Viewer side by the Viewer client (“Expiry”)

## User Characteristics

The users of ExpirePDF need be familiar with only basic file operations (opening files) in order to use File Open/Selection operations prompted by the Viewer or Publisher clients.

## Design Constraints

ExpirePDF is used for limited distribution of documents via electronic means, using encryption schemes to accomplish this. However, because of the decentralized nature of ExpirePDF’s DRM system, it will not be absolutely secure considering it may fall under extended attack by a more-than-casual user or dedicated hacker. Because of this, although we may attempt to secure the system from outside attack, despite our best efforts an experienced party will be able to undermine the security of our application.

# Functional Requirements

**Each requirement has a priority, which have the following meanings:**

1. **Priority 1:** These requirements must be fully satisfied and verified in order for the system to be released.
2. **Priority 2:** These requirements are expected to be verified, but their non-fulfillment will not affect the core capabilities of the system.
3. **Priority 3:** These requirements are not expected to be verified, and their non-fulfillment will not affect the core capabilities of the system. However they are still highly desired requirements.
4. **Priority 4:** These requirements are the first steps for the evolution of this software system, but are not expected.

## Functional Requirements

### Creating an encrypted PDF

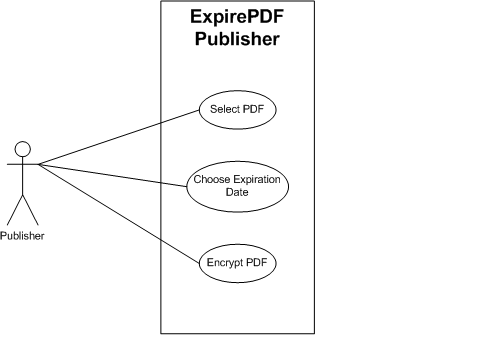


Figure 1 Publisher Use Case Diagram

**1100** The Publisher encrypts an existing PDF file such that it is not readable by PDF reader software. **(Priority 1)**

**1200** The Publisher provides options to control the length of time during which the PDF can be decrypted **(Priority 1)**

**1300** The Publisher provides a file selector to choose the PDF file to encrypt **(Priority 1)**

### Viewing an encrypted PDF

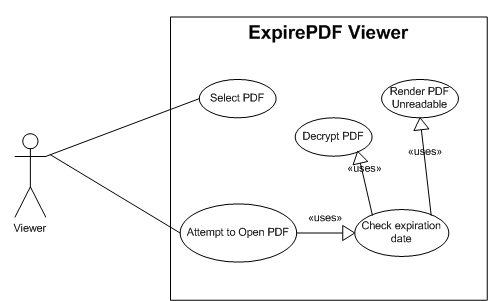


Figure 2 Viewer Use Case Diagram

**2100** The Viewer reads the Publisher-encrypted PDF, and is capable of decrypting it. **(Priority 1)**

**2200** The Viewer determines whether a Publisher-encrypted PDF has a limit on the time during which the PDF can be decrypted. **(Priority 1)**

**2300** The Viewer only decrypts a Publisher-encrypted PDF during the time period for which it was configured. **(Priority 1)**

**2400** The Viewer does not provide the user the unencrypted PDF. **(Priority 1)**

**2500** The Viewer uses the local system’s PDF reading application to display a decrypted document. **(Priority 1)**

**2600** The Viewer renders a Publisher-encrypted PDF unreadable after the document’s expiration date arrives. **(Priority 1)**

# Non-functional Requirements

## Installation Requirements

**3110** The ExpirePDF Publisher and Viewer are installed by the end user through a standard Windows installer. **(Priority 3)**

## Memory Requirements

**3210** The Publisher requires less than 512MB of RAM for encryption operations. **(Priority 1)**

**3220** The Viewer requires less than 512MB of RAM for decryption operations. **(Priority 1)**

## Help Requirements

**3310** The Publisher contains user assistance provided through the Windows help system, to aid users with the use of the software. **(Priority 2)**

**3320** The Viewer contains user assistance provided through the Windows help system, to aid users with the use of the software. **(Priority 2)**

## Portability Requirements

**3410** The ExpirePDF software solution is able to run on Linux distributions, through the use of the Mono implementation of .NET. **(Priority 4)**

# User Interface

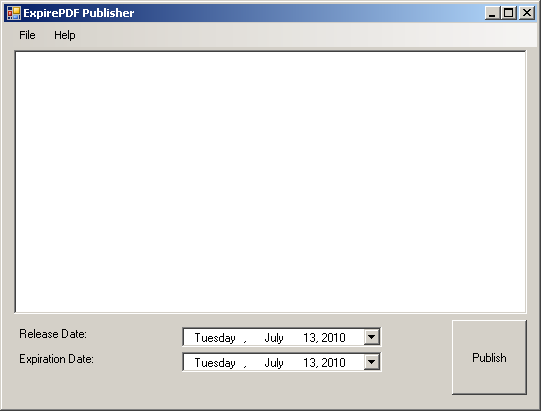


Figure 3 ExpirePDF Publisher User Interface

Figure 3 is the main view for the ExpirePDF Publisher system. It contains fields for selecting any number of PDFs for encryption and future distribution by the Publisher party, and various dialogs for customizing the distribution mode.

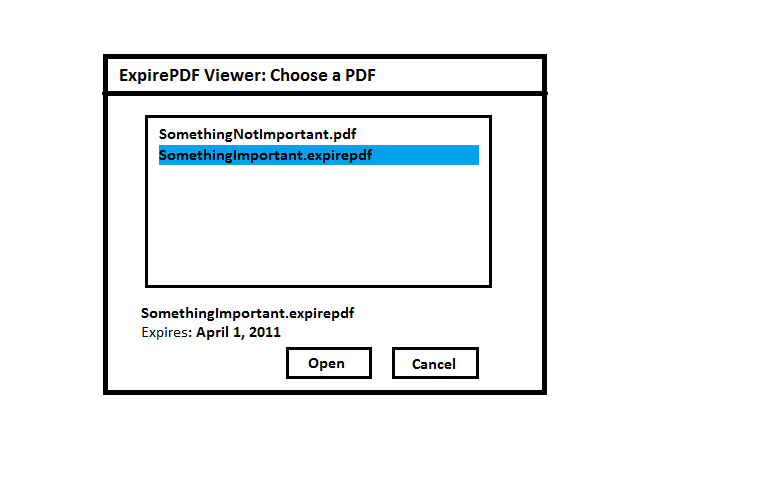


Figure 4 ExpirePDF Viewer User Interface

Figure 2 is the Viewer’s file selection dialog for the ExpirePDF Viewer system. In case the user does not use the OS-integrated file opening dialog, the ExpirePDF Viewer can open any number of files for encryption for viewing by the end-user.

# System Evolution

The future versions of ExpirePDF will focus on three paths of improvement and growth. The first is an expansion of supported operating systems. Initially created for Windows, the ExpirePDF system uses the .NET Framework 3.5 to include the common components on the system. For Linux and Mac OS environments, an open source .NET Framework implementation named Mono already exists. The first step to bringing ExpirePDF to those environments will be determining whether executing it with Mono is capable of supporting all of our priority 1 requirements. If that approach is successful, then not only will the first path of evolution be simplified, future maintainability will be improved, as there would only be a single codebase. If it is determined that Mono is insufficient, the following step for this path will be to investigate other programming languages to which ExpirePDF might be ported.

The second path of evolution is adding support for a larger number of digital document formats. The choice of PDF for the initially supported format was based on its ubiquity. However, there is a wide variety of formats which are shared, especially when each format can have different, incompatible versions. In addition to adding the common Microsoft document formats, ExpirePDF will also target the Open Document Format as well.

Finally, the third direction of improvement is integrating ExpirePDF more tightly with the reading applications and the operating system as well. The idea is that the less additional interaction ExpirePDF requires, the more enjoyable the user experience will become. The primary targets are to have ExpirePDF documents behave just like an unencrypted file, in the document reading application and in the file browser, when the file is still valid.

# Glossary

**Copyright**: the set of exclusive rights granted to the author or creator of an original work, including the right to copy, distribute and adapt the work.

**Cryptography**: the science of analyzing and deciphering codes and ciphers and cryptographs.

**Decryption**: (e.g. “software for encryption” can typically also perform decryption), to make the encrypted information readable again (i.e. to make it unencrypted).

**Digital** **Rights** **Management**: generic term for access control technologies that can be used by hardware manufacturers, publishers, copyright holders and individuals to impose limitations on the usage of digital content and devices.

**DRM**: see Digital Rights Management

**Encryption**: the process of transforming information (referred to as plaintext) using an algorithm (called a cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key.

**OpenPGP**: An open-source implementation of PGP.

**PDF**: stands for Portable Document Format, a commonly used format for digital documents.

**PGP**: stands for “Pretty Good Privacy” and is the name of a data encryption/decryption suite.

**Publisher**: In the ExpirePDF scheme, a party who wishes to share information, possibly copyrighted, in a secure manner with other parties, usually Viewers.

**Viewer**: In the ExpirePDF scheme, a party who views information created by a Publisher in a secure and controlled manner.

# Index

**ExpirePDF Publisher**

Requirements, p7

User interface, p10

**ExpirePDF Viewer**

Requirements, p8

User interface, p11