Software Requirements Specification

for

QTC Exam File Manager

Prepared by Dean Baquir, Norman Avery, Josh Mermelstein, Daniel Ibanez, Noe Lopez, Abdullah Alwabel, Jeffery Lum, Jose Mierzejewski, Sameen Khan, Wilson Tobar

Sponsored by QTC

12/04/21

Table of Contents

[1. Introduction 4](#_Toc89538335)

[1.1 Purpose 4](#_Toc89538336)

[1.2 Intended Audience and Reading Suggestions 4](#_Toc89538337)

[1.3 Product Scope 4](#_Toc89538338)

[1.4 Definitions, Acronyms, and Abbreviations 5](#_Toc89538339)

[1.5 References 5](#_Toc89538340)

[2. Overall Description 5](#_Toc89538341)

[2.1 System Analysis 5](#_Toc89538342)

[2.2 Product Perspective 6](#_Toc89538343)

[2.3 Product Functions 6](#_Toc89538344)

[2.4 User Classes and Characteristics 6](#_Toc89538345)

[2.5 Operating Environment 7](#_Toc89538346)

[2.6 Design and Implementation Constraints 7](#_Toc89538347)

[2.7 User Documentation 7](#_Toc89538348)

[2.8 Assumptions and Dependencies 7](#_Toc89538349)

[2.9 Apportioning of Requirements 8](#_Toc89538350)

[3. External Interface Requirements 8](#_Toc89538351)

[3.1 User Interfaces 8](#_Toc89538352)

[3.2 Hardware Interfaces 8](#_Toc89538353)

[3.3 Software Interfaces 8](#_Toc89538354)

[3.4 Communications Interfaces 9](#_Toc89538355)

[4. Requirements Specification 9](#_Toc89538356)

[4.1 Functional Requirements 10](#_Toc89538357)

[4.2 External Interface Requirements 10](#_Toc89538358)

[4.3 Logical Database Requirements 11](#_Toc89538359)

[4.4 Design Constraints 11](#_Toc89538360)

[5. Other Nonfunctional Requirements 11](#_Toc89538361)

[5.1 Performance Requirements 11](#_Toc89538362)

[5.2 Safety Requirements 12](#_Toc89538363)

[5.3 Security Requirements 12](#_Toc89538364)

[5.4 Software Quality Attributes 12](#_Toc89538365)

[5.5 Business Rules 12](#_Toc89538366)

[6. Legal and Ethical Considerations 12](#_Toc89538367)

[Appendix A: Glossary 13](#_Toc89538368)

[Appendix B: Analysis Models 13](#_Toc89538369)

[Appendix C: To Be Determined List 13](#_Toc89538370)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| Norman Avery | 11/04/2021 | Initial Version | 1 |
| Norman Avery | 12/04/2021 | Updated |  |
|  |  |  |  |
|  |  |  |  |

<Add rows as necessary when the document is revised. This document should be consistently updated and maintained throughout your project. If ANY requirements are changed, added, removed, etc., immediately revise your document.>

# 1. Introduction

Exam File Manager (EFM) is a file managing web application used for browsing and modifying documents. The application is being redesigned to allow for extensibility. This is done by using plugins. The application will authenticate users and place them in their respective line of business. The line of business will determine the functionality available to the user. The application will allow users to download, scan, repair, delete, index, and view documents. The supported file types are .pdf, .doc, .docx.

##  1.1 Purpose

This document is intended to explain the functionality of the Exam File Manager and elaborate on QTC’s requirements for how the software will work. The Exam File Manager is currently under development as a version 1.0 software tool. This document contains all the requirements of the software and a description of how the application works. A technical glossary is provided at the end of the document to improve user readability and understanding.

## 1.2 Intended Audience and Reading Suggestions

Developers should review this entire document to gain a better understanding of the features of Exam File Manager. Users from different lines of business should read the requirements and description section. Furthermore, users may find the interface section useful and should refer to Section 6 for legal and ethical issues arising from the implementation of this application. Users will find the description and interfaces sections useful. Testers should read the requirements section and the description section. Section 2 of the document provides a detailed section on what the application is and what it should be able to do. Section 3 provides information about the external interface requirements of the Exam File Manager. Section 4 provides detailed requirements specifications for Exam File Manager. The requirements should include a description of each input to the system, each output, and all the functions that the system performs in response to the input or output. Section 5 provides specific details about performance requirements, safety requirements, security requirements, software quality attributes and business rules. Finally, Section 6 provides information about the legal and ethical implications of the software.

## 1.3 Product Scope

The Exam File Manager is a web application used as file repository to view on standard HTML5 compatible browsers such as Mozilla Firefox, Safari, Internet Explorer, Microsoft Edge, Opera, and Google Chrome. The Exam File Manager is currently intended for private use by QTC and its associates. This application provides users from different lines of business access to specific features depending on their respective line of business. Such features include the ability to scan new medical exam documents, as well as the ability to view medical exam documents. Other users will have access to different features according to their respective line of business. All files will be stored in the Exam File Manager. If a user belonging to a line of business deletes a file, it will be deleted from their view. The deleted file will still exist in the Exam File Manager - however it will not be accessible to the user belonging to the respective line of business.

## 1.4 Definitions, Acronyms, and Abbreviations

Definitions are provided in Appendix A: Glossary.

## 1.5 References

https://api.nuget.org/v3/index.json

https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-5.0

https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-5.0&tabs=visual-studio

https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/views/using-the-tagbuilder-class-to-build-html-helpers-cs

https://docs.microsoft.com/en-us/aspnet/web-api/overview/security/preventing-cross-site-request-forgery-csrf-attacks

https://docs.microsoft.com/en-us/aspnet/web-pages/overview/getting-started/introducing-razor-syntax-c

https://docs.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.razor.language.razorengine?view=aspnetcore-5.0

https://docs.microsoft.com/en-us/dotnet/core/tutorials/creating-app-with-plugin-support

https://docs.microsoft.com/en-us/dotnet/framework/mef/

https://docs.microsoft.com/en-us/learn/modules/microservices-aspnet-core/?WT.mc\_id=dotnet-35129-website

https://docs.microsoft.com/en-us/learn/paths/create-microservices-with-dotnet/

https://dotnet.microsoft.com/apps/aspnet/microservices

https://dotnet.microsoft.com/learn/aspnet/microservice-tutorial/intro

https://stackoverflow.com/questions/3058/what-is-inversion-of-control

https://stackoverflow.com/questions/38138100/addtransient-addscoped-and-addsingleton-services-differences

https://www.ada.gov/

https://www.c-sharpcorner.com/article/file-upload-and-download-using-asp-net-mvc-5-for-beginners/

https://www.c-sharpcorner.com/UploadFile/85ed7a/dependency-injection-in-C-Sharp/

https://www.freecodecamp.org/news/a-quick-intro-to-dependency-injection-what-it-is-and-when-to-use-it-7578c84fa88f/

https://www.tutorialspoint.com/asp.net\_mvc/asp.net\_mvc\_filters.htm

https://www.tutorialsteacher.com/ioc/inversion-of-control

https://www.youtube.com/watch?v=eoWGiQla1kk

https://www.youtube.com/watch?v=jhtbhSpV5YA

https://www.youtube.com/watch?v=NYr0IgK2yAk

https://youtu.be/BfEjDD8mWYg

https://youtu.be/eiABHMxjIXw

https://youtu.be/Pi46L7UYP8I

# 2. Overall Description

EFM new design will be designed for extensibility. This feature will enable the ability to dynamically configure the application, customized for a specific line of business. The method of configuration employs the use of a developer plugin (assemblies) that will configure the application at run time. To make it possible for the web application to interact with external systems, additional services will be configured by the plugin, and injected into the application via dependency injection.

## 2.1 System Analysis

The current Exam File Manager application needs to be redesigned for extensibility. Currently, the EFM application uses excessive if-then logic to distinguish which line of businesses that the application’s business logic executes against. This web application will be designed to solve these issues with the following goals:

1. The new approach will increase scalability, as Lines of Business are installed to the web application by assembly plugins, being authenticated based on user roles and plugin configuration. The underlying web application will be generic in nature, but configurable and extensible using assembly plugins.
2. The next step is to install Lines of Business by implementing as assembly plugin that contains all necessary information that is related to those lines of business. The plugins will be added as collection to a service object.
3. The service object shall be injected into a web controller via dependency injection to perform all necessary functions related to EFM. The features (Index, View, Delete, etc.) are implemented as plugins.
4. We must make sure that EFM can configure features to a specific line of business and render to a dynamic Razor page.
5. Authentication will be taken care of by the QTC’s authentication module, which gives the roles, security levels and Lines of Business

Some of the possible hurdles could be extending QTC’s authentication model into the application. Currently not all details of QTC’s authentication module are known with reference to structure of the module. The solution will be to obtain documentation for QTC’s authentication module to better understand its technology as a solution.

## 2.2 Product Perspective

EFM will be interacting with different type of systems that includes File Repositories, a Database, Assembly Plugin, local file system and QTC’s authentication module. The database shall interact with EFM through SQL procedures. The local file system shall be used to import files into EFM. The Assembly plugin contains all necessary information related to a line of business. The file repositories shall be cloud-based, by a service of QTC’s choice.

EFM is dependent of QTC’s authentication module. This authentication module is responsible for all the roles, security levels, and Line(s) of Business that will enable the plugin configuration.




## 2.3 Product Functions

EFM is a file managing application that is used for browsing and modifying documents including medical records and Diagnostics. EFM will authenticate users and associate them with a Line of Business.

EFM will possess the following common features:

* Confirm - This feature confirms that the indexing of a medical record is complete.
* Delete – This feature deletes a file by renaming it and hiding from view
* Import - This feature allows a user to import a file into EFM
* Index - This feature will display a window, prompt input from user, and set the pages belonging to a specialty. For example: pages 1-10 belong to one specialty, and 11-13 would belong to another.
* Repair- This feature will attempt to repair a file if it is corrupt
* Scan - This feature allows documents to be scanned by physical hardware scanner attached to the user’s computer, saving the file into EFM
* Search- This feature allows a user to return a view displaying specific files respective to two parameters: an account number and a date range.
* View – This feature displays in a web page the file associated with a medical record

## 2.4 User Classes and Characteristics

The application will have many different classes of users. Users will be separated into two categories: VA access and non-VA access. Depending on the category that a user falls into, the user will have access to different types of documents. The application will be frequently used since document management is an important aspect of QTC’s business. The application is expected to be used by users with an understanding of simple file operations such as downloading, scanning, deleting, and viewing.

## 2.5 Operating Environment

The application will be accessed by a web browser. The application can be accessed by any computer with a web browser. The application can also be accessed by most desktop operating systems such as Windows, MacOS, and Linux.

## 2.6 Design and Implementation Constraints

The application was created using Microsoft’s .NET 5 web framework. The programming language used is C#. The application also makes use of “.dll” files that must be compiled independent of the application’s runtime environment, by implementing a provided interface. Since downloading and scanning files are the application’s major functions, the user’s computers should be able to access local files.

## 2.7 User Documentation

User documentation is still pending creation.

## 2.8 Assumptions and Dependencies

It is assumed that the application software will be installed and executed inside a Windows Server 2016 operating system. If the operating system is not available, the software requirements should be modified accordingly to reflect the change.

The application software will use a third-party authentication module to provide the defined permissions set for each user. This application depends on this information to define which functionalities are permitted to be used by the user. If the third-party authentication module is not available, the software requirements should be modified accordingly.

The application software will use third-party software for natural language processing. This application depends on this software to enable the repair functionality to attempt recovery of corrupted files. If the third-party software for natural language processing is not available, the software requirements should be modified accordingly.

The application software will use third-party TWAIN drivers installed in the local machine where this application will be used. The drivers correspond to the scanning device that is installed in the local machine and is used to scan documents being imported by the application. If the third-party TWAIN drivers for the scanning device are not available, the software requirements should be modified accordingly.

## 2.9 Apportioning of Requirements

If the web application release is delayed, a list of requirements to be used for the next version of the web application will be provided. These requirements will be developed in a future release of the web application.

# 3. External Interface Requirements

## 3.1 User Interfaces

The user interface will consist of a tab-based file management system with multiple buttons.
Current version is still being worked on and additional buttons will be included in the final version of this document.



The interface will include buttons that are associated with the following common features:

* Append files
* Scan files
* View files
* Delete files
* Textboxes to enable users to search for specific files
* Search button to validate input and return related files

Design standards will be compliant to the Americans with Disabilities Act (ADA)

## 3.2 Hardware Interfaces

This software does not interact with any hardware device, therefore, the software does not have a hardware interface requirement

## 3.3 Software Interfaces

User interfaces and overall functionality will be built within .NET Core version 5 framework.
The application will need to connect to an external file repository and database. The Database technology has not been finalized and is still in review. Application will be HTML5 compatible

## 3.4 Communications Interfaces

Exam File Manager is as a web application, and it will communicate to the IIS web server manager using the .NET Framework. The program will function on HTML5 compatible web browsers. The web application will communicate with a SQL server to retrieve and store information from the database. The web application will use Windows authentication and provide secure communication between networks via Hypertext Transfer Protocol Secure (HTTPS).

# 4. Requirements Specification

## 4.1 Functional Requirements

|  |  |
| --- | --- |
| Requirement No. | Description |
| 1.1.1 | The system shall implement an MVC ASP.NET web solution. |
| 1.1.2 | The system shall allow a pre-authorized user to select which of their lines of business to interact with. |
| 1.1.3 | The system shall display the files belonging to a user's selected line of business. |
| 1.1.4 | The system shall provide the following functionality based on the user's role and hide functionality that is not. |
| 1.1.5 | The system shall provide search functionality based on account number and/or a date range. |
| 1.1.6 | The system shall provide scan functionality to import new files. |
| 1.1.7 | The system shall provide functionality to delete existing files. |
| 1.1.8 | The system shall provide functionality to view files. |

## 4.2 External Interface Requirements

|  |  |
| --- | --- |
| Requirement No. | Description |
| 1.2.1 | The system shall receive from an auth module all roles and lines of business associated with a user |
| 1.2.2 | The system shall provide a user interface with buttons to trigger functionality |
| 1.2.3 | The system shall display buttons for functionality that a user has access to, and “grey out” or hide buttons that a user does not have access to. |
| 1.2.4 | The system shall interface with a database to retrieve files and information |

## 4.3 Logical Database Requirements

|  |  |
| --- | --- |
| Requirement No. | Requirement Description |
| 1.3.1 | The database shall store the file locations of all files |
| 1.3.2 | The database shall store roles associated with users  |
| 1.3.3 | The system will use a dictionary object to create service objects to connect to a file repository and database |
| 1.3.4 | The database shall have a table that displays the relation between a file and its category. |



## 4.4 Design Constraints

To display information, a SQL database must be connected to the program. The program will interact with the database to retrieve files and display them into the view. The application will interact with the database to display different information for different lines of businesses. Any device that supports an HTML5 web browser and has access to a stable internet connection can operate the application. It also needs all necessary requirements for windows authentication, as using this application will need verification and authorization to use the main functions.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

This application shall not crash and should load the necessary requirements (.dll) plugins within 60 seconds. It shall only load the necessary plugin needed for a specific user's line of business and shall not load other plugins. This web application must be capable of handle up to of 4,000 concurrent users without affecting application performance. The authentication module shall not negatively affect the performance of the web application.

By design, the “download” feature is slow, and therefore will need to use a “repair” function as a faster means. As the reliability of the download feature is unpredictable, it shall be used as a last resort when attempting to access a record. The performance of download depends on the size of the file and the amount of corruption that exists in the file.

All controllers shall be able to render a user page after authenticating the user and retrieving plugin information.

## 5.2 Safety Requirements

The user will be able to make database changes, upload, and delete files. It is necessary that this application only makes changes when the user is authorized to do so, and that all changes allowed are within the predefined constraints of the system.

The repair feature allows a user to attempt to access a corrupted file in the case that download is not successful.

Deleting a file renames it but does not physically delete it. This is to ensure that files can be recovered in the case of accidental deletion.

## 5.3 Security Requirements

Using authentication will allow this application to authorize users access to certain features. The web application also contains features to view and edit sensitive medical documents. The requirements as of now are not finalized but will need proper regulations and methods that meet required compliances that are standard to the lines of business’ respective industry.

QTC authentication uses Windows authentication. The application will use an authentication module in between the user and the controller as gated access and will remember who the user is, as well as all the users' lines of business. It will also know what role, and roles that user has within each line of business it pertains to.

## 5.4 Software Quality Attributes

### TESTABILITY

Source – System User

Stimulus – Analysis of compiled plugin .dll assembly

Artifact - compiled assembly plugin

Environment - Deployment

Response – Test environment consisting of replicated version of web application populated with sampled data

Response Measure – 100 Percent executable if statements are executed. 100 percent probability of failure if fault exists

### MODIFIABILITY

Source - Developer

Stimulus - Wishes to add functionality

Artifact – Compiled Assembly plugin (dll)

Environment - At runtime

Response – Compiled Assembly (dll) is copied to file location as specified by web application.

Response Measure – Should not affect functionality of any other features

## 5.5 Business Rules

Business rules typically apply to which lines of business a user is allowed to access, as well as which respective common features area available to the user. These rules are dictated by the authentication module implemented in the web application. Access to the application’s lines of business and features are controlled by roles that a user belongs to, as defined by the requirements of the respective business units.

# 6. Legal and Ethical Considerations

All legal decisions are made by QTC, as this is their application and database. To that extent, QTC legality is not finalized until other documentation is provided, however all transactions with files and users will be done by means of QTC’s considerations with respect to security legal requirements, including source code related to sensitive documents that this application will use.

Ethically, choices will be made in consideration of sensitive issues. It is important to understand that this application will deal with very sensitive information and that all possibilities of security breaches should be addressed at the appropriate times.

# Appendix A: Glossary

**AllPlugins\_Service Object -** A service object where all the plugins necessary for executing the EFM web program would be loaded as a collection during runtime. Stored by line of business.

**Assembly Plugin -** A plugin that is designed in an abstract manner for individual lines of businesses, roles, and other common but implicitly specified components and attributes

**Confirm -** A button that confirms the indexing of medical records

**Controller -** The medium between the Model and View in the MVC architecture that adds, modifies and display data as per the specifications in the Model and View

**Delete –** Renames a file to hide it from the user’s view instead of explicit deletion of a file

**Dependency Injection -** A design pattern that provides an ease of accessing objects outside of the class that it depends on without the manual labor of modification

**EFM -** Exam File Manager - the name of the web application.

**Entity Framework Core -** An object to database mapper for .NET which supports LINQ queries, updates, schema migrations etc. that works with many “flavors” of databases

**Extensibility/Scalability -** Indicates the motive of “dynamically expanding” the current organization (QTC) with several Lines of Business (a.k.a. LOB)

**File Repositories -** A system on a local machine or a server that assists in file storage and version control of those files

**Generic -** A design pattern to implement abstract classes and interfaces to define general objects in object-oriented programming to enable the creation of flexible objects which consist of slight variations in their respective attributes

**GUI -** Graphical User Interface

**Import -** Enabling the user to import files from their local file repository.

**Index -** Configure range and count of pages

**Landing page -** The first point of entry in EFM based off GUI and UI flow.

**LOB -** Line of Business

**Members -** Variables other than attributes that define functionalities

**Methods -** A concept in object-oriented programming to define functionalities and attributes

**Plugin -** A software that functions as a “sub-software” or a “mini software” that anchors/supports the main software by providing its own functionalities as services. For example, supporting a web browser with several plugins with smaller functionalities such as upload, import, and view.

**PluginLoaderHelper -** A class that will load the plugin and instantiate an object containing the respective business logic for each common feature.

**QTC -** Quality Timeless Customer Service - Name of the organization

**Queries -** A request for accessing data from a database table or a set of tables. Retrieves rows from a table, or tables in the database.

**Razor pages -** A web application programming model that simplifies ASP.NET MVC by adopting file-based routing approach

**Repair -** Enable the user to view files or prompt them to redownload it

**RetrieveFeatureNames -** Function belonging to the AllPlugins\_Service\_Object that retrieves a list of available features defined in the plugin

**Scan -** Involvement of hardware to digitize physical data

**Search -** Returns data to the user view as per the keyword inserted

**TWAIN -** Technology Without An Interesting Name that is used to pull images from a scanner or digital camera directly into a software that supports this method of technology

**View -** A button that allows the user to view the file associated to a specific component of the data in the view

# Appendix B: Analysis Models

N/A

# Appendix C: To Be Determined List

* Integrated File Repository technical solution
* Integrated Database technical solution
* Integrated Authentication technical solution