Software Requirements Specification

for

Digital Declaration of Conformity

Version 1.0

Prepared by John Palladino, Dom Rohan, and Sam Shadle

Franciscan University of Steubenville

September 29, 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Initial Draft	10/6/22	Initial Draft	0.5
Final Draft	10/11/22	Jon Crumpacker and Prof. Wessel Suggestions	1.0

1. Introduction

1.1 Purpose

This product will allow a church authority to manage the Imprimatur and Declaration of Conformity process digitally in a time-effective way, by managing the review process between publishers and reviewers in a centralized workflow environment. Once approved and given a digital Imprimatur or Declaration of Conformity, a work cannot be digitally altered without losing the Imprimatur or Declaration of Conformity.

1.2 Document Conventions

- The terms Imprimatur and Declaration of Conformity will be used interchangeably throughout this document to refer to an official Church stamp of approval on published media

1.3 Intended Audience and Reading Suggestions

This document is intended for internal use only, for the project team to track the stakeholders' requirements and ensure they are actively being fulfilled by the project's work activities.

1.4 Product Scope

Our project will provide a way to collect and store documents for review, automate the review process, allow human in-the-loop intervention, provide a user friendly interface, a secure digital stamp of approval, provide adequate cybersecurity to block hackers or embedded nefarious content, and archive the final approved digital version.

1.5 References

Project Plan Document
Current Conformity Review Process Document
Turnitin Feedback Studio Demo: https://www.turnitin.com/static/feedback-studio-demo/

2. Overall Description

2.1 Product Perspective

The Catholic Church's existing process for issuing an Imprimatur or Declaration of Conformity, which has existed for thousands of years, utilizes inefficient physical and paper methods and various, non-uniform technology. However, in modern times, the internet is more commonly used for accessing media, and the Church wants to increase its digital capabilities and throughput. As a result:

- The review process takes over 9 months per work, and
- There is no way to verify the integrity of a digital work once it is approved

2.2 Product Functions

System Features

- General User: UI for Publisher to facilitate document submissions for review
- Diocese User: UI for Church verification staff to review submitted documents
- Diocese review process (e.g., Workflow) partial automation by software tools
- Allow for manual inspection of the submitted article, log comments
- For successfully reviewed document (passed), the system will insert a Digital Imprimatur or Declaration of Conformity in the document (and automatically lock down the document), and its appearance in product
- Add function to store the document, and provide sender a copy
- Provide a way for end consumers to verify the Imprimatur or Declaration of Conformity and authenticity of document

System Products

- Document review findings report for Institute or Diocese staff,
- Log and report trail of reviewers and comments.

2.3 User Classes and Characteristics

- Content Publishers: to receive a faster review process
- Reviewers: to provide a simpler/easier review process
- End Consumers: a higher volume of more affordable, high quality products

2.4 Operating Environment

- The development environment will use a local database and server running on the Franciscan development test servers
- We will use a Windows server to run our application
- Our options for a production environment are
  - Microsoft Azure
  - Amazon Web Services
  - A dedicated server
- Database options:
  - Local: MySQL
  - Production: Microsoft azure(AWS)

2.5 Design and Implementation Constraints

This project must be completed in the context of a class. As a result, the word activities may need to be modified to fit the goals and schedule of the class. In addition, there is no possibility of a budget.

2.6 User Documentation

- Onboarding tutorial

- Help button/site which addresses the most common issues for each page

2.7 Assumptions and Dependencies

- Must work with the existing process of Imprimatur or Declaration of Conformity approval
- Church decision to manage by USCCB, or each Diocese
- Church existing tools
- Desktop or mobile device with secure Internet connection

3. External Interface Requirements

3.1 User Interfaces

Javascript (React with MUI for frontend, Node for backend)

3.2 Hardware Interfaces

For project development, everyone should be able to use existing hardware and networks; no purchases should be required for development (there is also the existing lab in JSerra).

3.3 Software Interfaces

- For the project code: git/github
- For podcasts: RSS feeds
- Javascript (React with MUI for frontend, Node for backend)
- MySQL for database
- For documenting the project progress and sprints: Jira

3.4 Communications Interfaces

Email will be used to notify the Publisher when the document is finished with the review process. The email will communicate whether the document was passed or denied. We will be using HTTPS.

4. System Features

4.1 Publisher: Submit a Document for Review

4.1.1 Description and Priority

To get the Catholic Church’s seal of approval for their books, a publisher needs to submit them for review first. A document is submitted by a publisher and then sent to a publishing house for review. This is the primary functionality of our system, so it is a high priority feature.

4.1.2 Stimulus/Response Sequences

SEQ-1: Screen prompts the user to upload or drag a file

SEQ-2: User selects a file

SEQ-3: System verifies the document is a valid type

SEQ-4: The system begins uploading the file

SEQ-5: The system shows upload progress,
and an option to cancel the upload

4.1.3 Functional Requirements

REQ-1: Begin document upload
REQ-2: Verify document is valid type
REQ-3: Drag and drop document for upload
REQ-4: See upload progress
REQ-5: Cancel upload

4.2 Reviewer: Download submitted documents

4.2.1 Description and Priority

Once a document has been sent for review, the reviewing house must be able to access it to do their jobs. The first method of document access we will implement will be downloading a document. This is also a high priority feature, as the system will not function without it.

4.2.2 Stimulus/Response Sequences

SEQ-1: Reviewer views incoming documents

SEQ-2: Reviewer selects the download option for a document

SEQ-3: Reviewer selects a format for download

SEQ-4: Download begins

4.2.3 Functional Requirements

REQ-1: See incoming documents for review
REQ-2: Download documents in uploaded format
REQ-3: Convert documents to a requested format

4.3 Reviewer: View Submitted Document on the Web

4.3.1 Description and Priority

Reviewers will want to see a document before they download it, and might also want a good interface to read through documents and mark them up. This is a medium high priority feature. On a scale of 1 to 9, it has an 8 for benefit, a 1 for penalty, a 6 for cost, and a 4 for risk.

4.3.2 Stimulus/Response Sequences

SEQ-1: Reviewer views incoming documents

SEQ-2: Reviewer selects the view option for a document

SEQ-3: System navigates to the document viewer

SEQ-4: Document viewer loads the document in question

SEQ-5: Document viewer is available for use

4.3.3 Functional Requirements

REQ-1: Display documents of different formats
REQ-2: Provide easy navigation between sections of the document
REQ-3: Save the reading location of the reviewer

4.4 Reviewer: Leave comments on the document

4.3.1 Description and Priority

The reviewer can directly annotate a section of a document with comments. This will make it easier for the publisher to edit a rejected document to conform to the teachings of the church, and cut down on the amount of time spent on searching for sections with feedback. An additional sub-feature to consider is time-keeping for reviewers, since they are paid hourly. This is a medium priority feature, with a 7 for benefit, a 2 for penalty, a 7 for cost, and a 3 for risk.

4.3.2 Stimulus/Response Sequences

SEQ-1: Reviewer opens a document in the viewer

SEQ-2: Reviewer chooses text to annotate

SEQ-3: Reviewer leaves a comment on the text

4.3.3 Functional Requirements

REQ-1: Annotate document with feedback
REQ-2: Sync feedback between publisher and reviewer
REQ-3: Capture and sequence comments in the system for laterm reporting

4.5 Reviewer: Provide a full report on a document

4.4.1 Description and Priority

In addition to using the built-in tools on the website, a reviewer can provide free-form comments to the publisher. These may take the form of some text or a document upload. This feature has a 4 for benefit, 0 for penalty, 5 for cost, and 4 for risk.

4.4.2 Stimulus/Response Sequences

SEQ-1: Reviewer opens a submitted document

SEQ-2: Reviewer selects an option to send feedback

SEQ-3: Reviewer selects a file to send

SEQ-4: The file is sent back to the publisher

4.4.3 Functional Requirements

REQ-1: Upload a report
REQ-2: Show the report to the publisher

4.6 Publisher: Contact reviewer(s)

4.5.1 Description and Priority

The publisher should be able to reach out to the individual person reviewing their submission, to resolve any issues and in case they have any questions. This feature has a benefit of 6, penalty of 1, cost of 2, and risk of 2.

4.5.2 Stimulus/Response Sequences

SEQ-1: Publisher selects a document

SEQ-2: Publisher selects an option to contact the reviewer

SEQ-3: Reviewer's contact information appears

4.5.3 Functional Requirements

REQ-1: Display reviewer(s)'s contact information

4.7 Reviewer: Track where the document is in the review process

4.6.1 Description and Priority

Both the publisher and the reviewer should be able to track a document's progress at all times. This is a medium priority feature. This feature has a benefit of 5, a penalty of 1, a cost of 1, and a risk of 1.

4.6.2 Stimulus/Response Sequences

SEQ-1: User selects an open document

SEQ-2: User views the status of the document

4.6.3 Functional Requirements

REQ-1: View list of all documents in progress

REQ-1A: For each document in progress, view next steps for publisher

REQ-1B: For each document in progress, view next steps for reviewer

REQ-2: View list of approved documents

REQ-3: View list of denied documents

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Web page should load in 4 seconds 95% of the time
- Server should be able to handle 100s of requests at the same
- Documents of a reasonable size should be uploaded in less than 10 seconds
- Server should have 99.9% uptime

5.2 Safety Requirements

Making sure documents are securely held in Database and are not tampered with during input, review and output stages. There will not be direct editing of the submitted document, rather comments can be made by the reviewer and sent back to the publisher.

Creating a safe database by not allowing any foreign malware or foreign access to the database. Users will be given a username and create a secure manual password to allow access to the database.

5.3 Security Requirements

- Passwords must be secure
- Passwords must be changed often
- Passwords must not be accessible to the client
- Users must not be able to access other users' information
- Users must not be able to perform as a different role
- Users must not be able to change any documents
- Passwords must only be stored in a hashed form
- DDOS prevention must be used
- Security will be used at the database level, at the backend level, and at the frontend level

5.4 Software Quality Attributes

- Compliance to relevant standards
- Reliable – for example up and running (99.98%)
- Available
- Maintainability, security patches
- Cloud based,
- Install-ability, resource utilisation

- Change-ability
- Recoverability
- Accurate
- In-line instructions & support
- UI aesthetics
- Mobile friendliness/responsiveness
- Secure

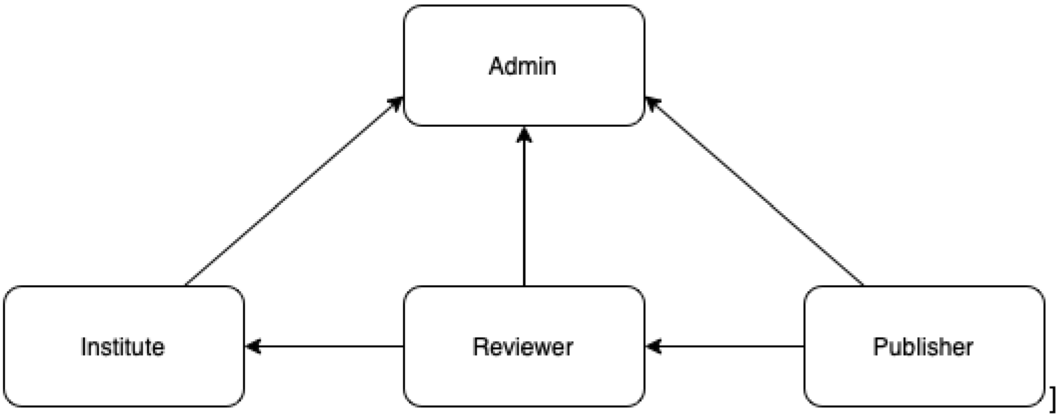
5.5 Business Rules

- Publisher will be the lowest level in the database. Functions include submitting the proposed document for review, checking on status of document throughout the review process, and reclaiming of document after process.
- Reviewer will be the second-lowest level in the database. Functions include access to documents that are in queue for review in the database, ability to comment on documents, and after successful review the document reviewer will forward the document to Diocese level.
- Diocese or Institute would be the second highest security level. Functions include everything a publisher can do, along with viewing a reviewer's comments, and issuing a final Imprimatur or Declaration of Conformity.
- Admin role refers to the highest level, which serves as a frontend for technical changes to the product, to be made only by product admins or those people responsible for implementing the product in their specific Diocese.

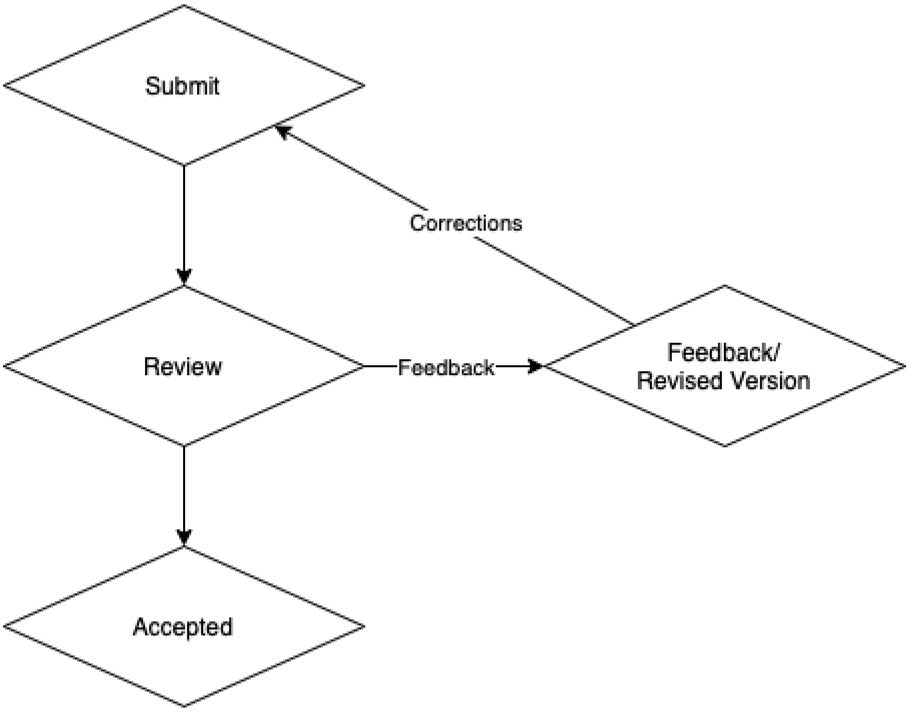
6. Other Requirements

Appendix A: Analysis Model

System Flow Diagram



System Process Diagram



Appendix B: To Be Determined List

- Appendix: Glossary