Software Design Description

Digital Declaration of Conformity

Senior Projects

Version 0.1 September 1st, 2022

FRANCISCAN UNIVERSITY OF STEUBENVILLE

By

Dominic Rohan

Sam Shadle

John Palladino

1 Introduction

1.1 Scope

Describe the architecture and development of the product in detail. 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.

Portray the layout of Client-server/servers, databases, client, encryption layer, document manager and authenticators.

1.2 Reference

- Project Plan Document
- Current Conformity Review Process Document
- Requirements Spec Document

1.3 Context

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)

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

1.4 Stakeholders

- Diocese of Baltimore
- Jon Crumpacker

1.5 Summary

Begin construction of the product in upcoming weeks using the examples below. Construction will be in line with the parameters of the Diocese of Baltimore by using various alleyways of various languages, code, and databases.

1.6 Glossary

Imprimatur: (Latin: “let it be printed”), in the Roman Catholic church, a permission, required by contemporary canon law and granted by a bishop, for the publication of any work on Scripture or, in general, any writing containing something of peculiar significance to religion, theology, or morality. [reference]

Publisher View: Lowest level of viewing and using the product with the ability to upload document and to view the current status of the submitted document throughout the imprimatur process

Reviewer View: Middle level of using the product with the ability to pull an uploaded document out of database and assign, “in progress” to the document. A Reviewer will be able to browse over the document and leave necessary comments, but unable to edit the document in any way.

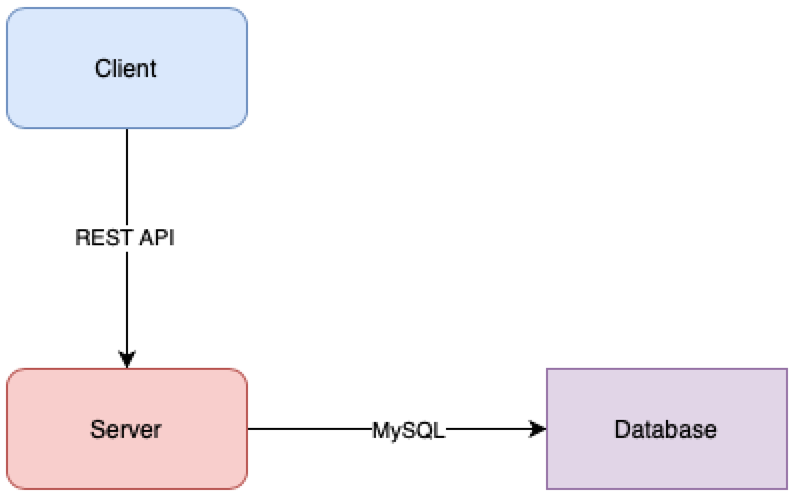
Verifier View: Verifier will be able to easily and reliably verify that this document has been granted a digital Imprimatur, regardless of the service used to write the document. If the document has been altered in any way, the content will not be served and the document will not be sent.

1.7 Change History

Version	Changes	Author
0.1	Added formatting	John Palladino
0.2	Added Quality Attributes	John Palladino

2 Views

2.1 Client-Server Architecture



Abstract

Our app uses a Client-Server architecture. The client gets requests from the server using a REST API, specified later in the document. The server uses MySQL to connect to a database and store all of our data.

Quality Attributes

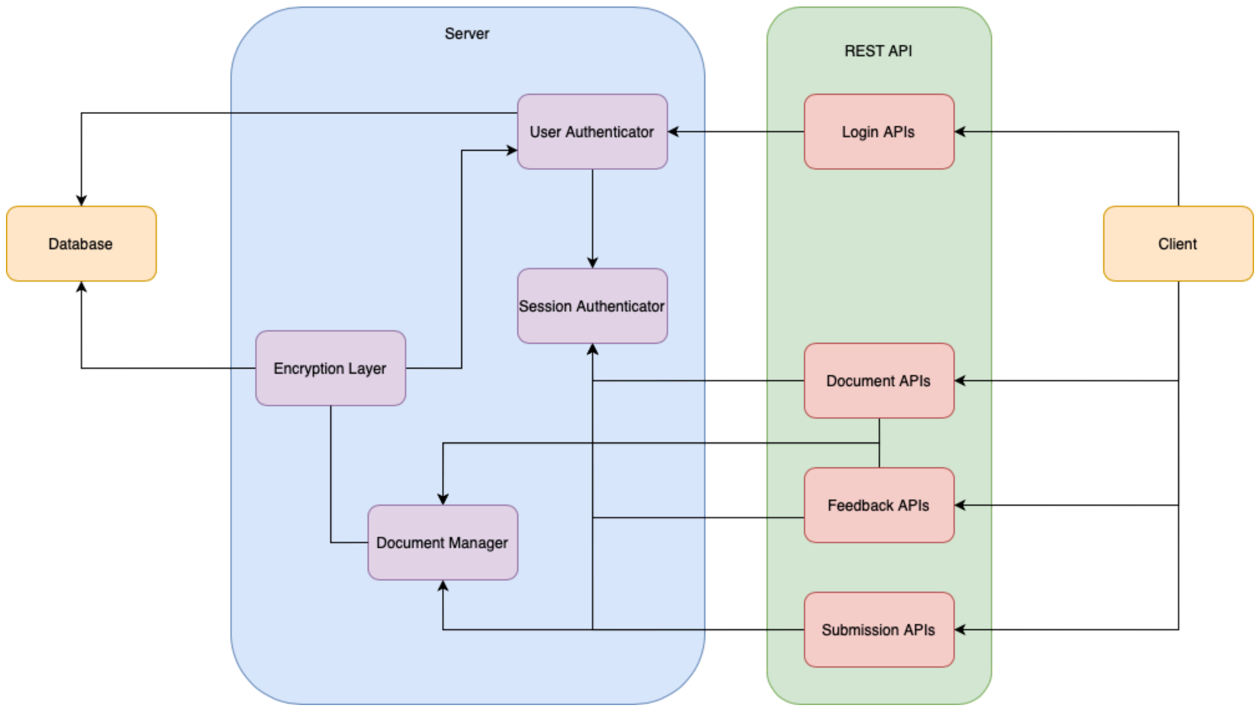
Our server will be reliable, available, and maintainable.

Entities

Our app has a client, server, and database portion

2.2 Server

2.2.1 Server Block Diagram



Entities:

Database
Client

Login API
Document API
Feedback API
Submission API

Encryption Layer
Document Manager
User Authenticator
Session Authenticator

2.2.1.1 Database

Abstract

The server will use Connector/Node.js to connect to the database

2.2.1.2 Client

Abstract
The server will use REST APIs with JSON to connect to the client

2.2.1.3 Encryption Layer

Abstract

The encryption layer provides an abstraction over the database connection layer. The encryption layer provides a common interface to securely store documents in the database. The details of the encryption are an implementation layer.

Subentities

- Database Connector
- Input API

Dependencies

- Database Communication layer
- Bcryptjs

Quality Attributes

- Accurate
- Change-ability
- Secure

2.2.1.4 Document Manager

Abstract

The Document Manager provides a high-level API to store documents in the database. The document manager provides CRUD functionality over the database, and can be used both for documents waiting for feedback and documents providing feedback. Although it is a lower level component, the Document Manger still relies on the session authenticator to ensure all queries are verified.

Dependencies

- Encryption Layer
- Session Authenticator

Quality Attributes

- Secure
- Available

2.2.1.5 Session Authenticator

Abstract

The Session Authenticator provides an in-memory cache for the user authenticator, and also manages login sessions. The session authenticator can create a new session, end a session, or verify a session.

Subentities

- HashMap from authentication tokens to authenticated user structure
- Authenticated user structure
    - Authentication token
    - User ID
    - Login permissions

Dependencies

- User Authenticator

Quality Attributes

- Secure
- Reliability
- Availability
- Recoverability
- Changeability

2.2.1.6 User Authenticator

Abstract

The User authenticator communicates with the database to ensure that all logins are valid. The user authenticator also keeps a record of every logged in session. The user authenticator abstracts the database implementation away.

Dependencies

- Connector/Node.js
- Database

Quality Attributes

- Secure
- Reliability

2.2.2 REST Endpoints

REST Endpoints

Users

Create a user

/user/create

Creates a user

Input

email: The user’s email

name: The user’s full name

password: The user’s password

permissions: What permissions the user has

Output

user_id: The users id

invalid_token: An invalid user token

Permissions

N/A

Error Codes

400 Bad request

The request is badly formatted
403 Forbidden
The users permissions were invalid

415 Unsupported Media Type
Any data except JSON was sent

User login

/user/login

Logs a user into the system

Input

user_id: The users id

invalid_token: The users invalid token

permissions: Permissions the user is requesting

Output

token: The users valid token

Permissions

N/A

Error Codes

400 Bad request
The request is badly formatted
403 Forbidden
The users permissions were invalid
415 Unsupported Media Type
Any data except JSON was sent

Invalidate login

/user/invalidate

Invalidates a login

Input

user_id: The users id
token: The users token

Output

N/A

Permissions

N/A

Error Codes

400 Bad request
The request is badly formatted
403 Forbidden
The users permissions were invalid

415 Unsupported Media Type
Any data except JSON was sent

Submission

Create submission

/sub/create

Creates a new submission from a publishing house

Input

token: The users login token
name: The name of the submission
description: The description of the token

Output

submission_id: The unique identifier for the submission

Permissions

Publisher permissions are needed to create a submission
CreateSubmission permissions are needed to create a submission

Error Codes

400 Bad Request
The sent JSON was invalid
403 Forbidden
The user's permissions weren't valid
413 Payload Too Large

The name or description were too long

415 Unsupported Media Type

Any data except JSON was sent

Read submission info

/sub/read

Given a submission, if the user has permission to access it, returns the metadata for that submission

Input

token: The users login token

submission_id: The submission to access

Output

submission: Submission metadata

Permissions

User has been assigned to the submission

Error Codes

400 Bad request

The JSON is badly formatted

403 Forbidden

The users permissions were invalid

404 Not found

The specified submission_id was not found

415 Unsupported Media Type

Any data except JSON was sent

List documents

/sub/read/docs

Lists the documents attached to a submission which are available to the logged in user.

Input

token: The users login token
submission_id: The submission to access

Output

document_ids: A list of the documents attached to this submission

Permissions

The user must be associated with the submission

Error Codes

400 Bad request
The request is badly formatted
403 Forbidden
The users permissions were invalid
404 Not found
The specified submission_id was not found
415 Unsupported Media Type
Any data except JSON was sent

List Feedback

/sub/read/feedback

Lists the feedback documents attached to a submission which are available to the logged in user.

Input

token: The users login token

submission_id: The submission to access

Output

feedback_ids: A list of the feedvack attached to this submission

Permissions

The user must be associated with the submission

Error Codes

400 Bad request

The request is badly formatted

403 Forbidden

The users permissions were invalid

404 Not found

The specified submission_id was not found

415 Unsupported Media Type

Any data except JSON was sent

Update Submission Metadata

/sub/update

Updates the metadata associated with the submission

Input

token: The users login token

submission_id: The identifier for the submission to update

delta: The changes to make to the submission name description

Output

N/A

Permissions

- User must be a publisher
- User must be associated with the submission
- User must have write permissions

Error Codes

- **400** Bad Request
  The sent request was invalid
- **403** Forbidden
  The user's permissions weren't valid
- **404** Not found
  The specified submission_id was not found
- **413** Payload Too Large
  The name or description were too long
- **415** Unsupported Media Type
  Any data except JSON was sent

Retract submission

/sub/delete

Retracts a submission

Input

- token: The users login token
- submission_id: The submission to retract

Output

N/A

Permissions

User must be a publisher

User must be associated with the submission User must have delete permissions

Error Codes

400 Bad request

The JSON is badly formatted

403 Forbidden

The users permissions were invalid

404 Not found

The specified submission_id was not found

415 Unsupported Media Type

Any data except JSON was sent

Documents

Upload Document

/doc/upload

Uploads a document to a submission

Input

token: The users login token
submission_id: The submission to attach the document to

document: The document being uploaded

Output

document_id: The created id for the uploaded document

Permissions

User must be a publisher

User must be associated with the submission

Error Codes

400 Bad request
The request is badly formatted

403 Forbidden
The users permissions were invalid

404 Not found
The specified submission_id was not found

415 Unsupported Media Type
Any data except JSON was sent

Read Document

/doc/read

Reads the metadata of a document

Input

token: The users login token

document_id: The document to download

Output

metadata: Document metadata

Permissions

User must be associated with the submission the document is attached to

Error Codes

400 Bad request
The request is badly formatted
403 Forbidden
The users permissions were invalid
404 Not found
The specified document_id was not found
415 Unsupported Media Type
Any data except JSON was sent

Download document

/doc/download

Downloads an uploaded document

Input

token: The users login token
document_id: The document to download

Output

The document

Permissions

User must be associated with the submission the document is attached to

Error Codes

400 Bad request
The request is badly formatted
403 Forbidden

The users permissions were invalid

404 Not found

The specified document_id was not found

415 Unsupported Media Type

Any data except JSON was sent

Retract Document

/doc/delete
Downloads an uploaded document

Input

token: The users login token

document_id: The document to delete

Output

N/A

Permissions

User must be associated with the submission the document is attached to User must be a publisher

User must have delete permissions

Error Codes

400 Bad request

The request is badly formatted

403 Forbidden

The users permissions were invalid

404 Not found

The specified document_id was not found

415 Unsupported Media Type

Any data except JSON was sent

Feedback

Upload Feedback

/feedback/upload

Uploads feedback to a submission

Input

token: The users login token

submission_id: The submission to attach the feedback to

feedback: The feedback being uploaded

Output

feedback_id: The created id for the uploaded feedback

Permissions

User must be a reviewer

User must be associated with the submission

Error Codes

400 Bad request

The request is badly formatted

403 Forbidden

The users permissions were invalid

404 Not found

The specified submission_id was not found

415 Unsupported Media Type

Any data except JSON was sent

Read Feedback

/doc/read

Reads the metadata of feedback

Input

token: The users login token
feedback_id: The feedback to download

Output

metadata: Feedback metadata

Permissions

User must be associated with the submission the feedback is

attached to

Error Codes

400 Bad request
The request is badly formatted
403 Forbidden
The users permissions were invalid
404 Not found
The specified document_id was not found
415 Unsupported Media Type
Any data except JSON was sent

Download Feedback

/feedback/download

Downloads uploaded feedback

Input

token: The users login token

document_id: The feedback to download

Output

The feedback

Permissions

User must be associated with the submission the document is attached to

Error Codes

400 Bad request
The request is badly formatted

403 Forbidden
The users permissions were invalid

404 Not found
The specified document_id was not found

415 Unsupported Media Type
Any data except JSON was sent

Retract Feedback

/feedback/delete

Downloads uploaded feedback

Input

token: The users login token

feedback_id: The feedback to delete

Output

N/A

Permissions

User must be associated with the submission the document is attached to User must be a reviewer

User must have delete permissions

Error Codes

400 Bad request
The request is badly formatted

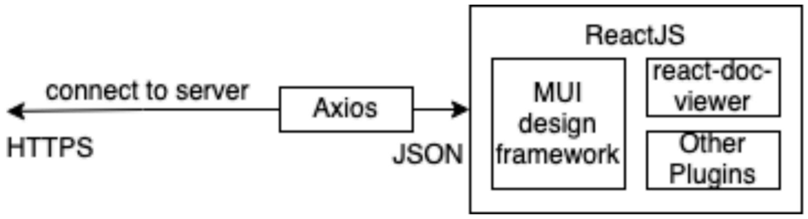
403 Forbidden
The users permissions were invalid

404 Not found
The specified document_id was not found

415 Unsupported Media Type
Any data except JSON was sent

2.3 Frontend

2.3.1 Frontend Overview



Abstract

The webapp's frontend will consist of ReactJS with a number of selected components, including the MUI design framework. React will use Axios to request data from the server, which will be returned by the server in JSON format before being rendered by the particular view in question.

Subentities

- ReactJS
- Any React components, including MUI, react-doc-viewer, and any others
- Axios

Dependencies

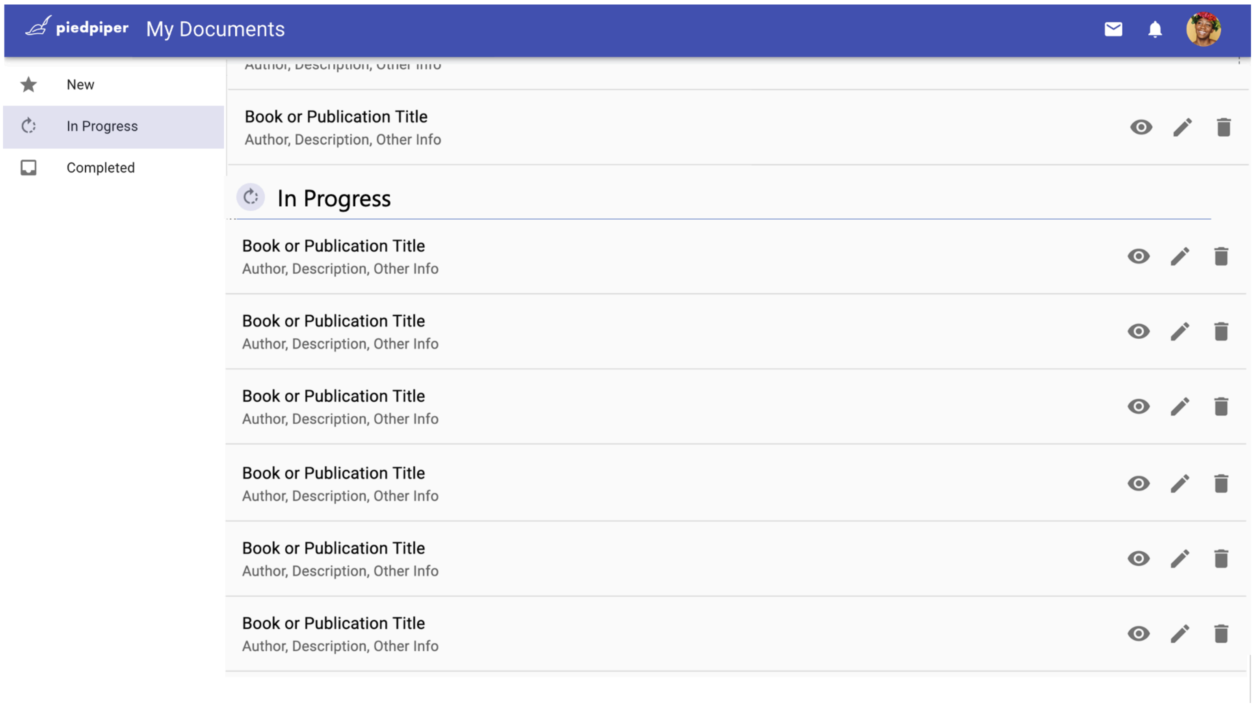
- Backend server, able to return data in JSON format

Quality Attributes

- Cloud based
- Install-ability, resource utilization

2.3.2 Frontend Design Mockups

2.3.2.1 “My Documents” Mockup


piedpiper
My Documents
New
In Progress
Completed
Book or Publication Title
Author, Description, Other Info
In Progress
Book or Publication Title
Author, Description, Other Info
Book or Publication Title
Author, Description, Other Info
Book or Publication Title
Author, Description, Other Info
Book or Publication Title
Author, Description, Other Info
Book or Publication Title
Author, Description, Other Info
Book or Publication Title
Author, Description, Other Info

Abstract

From the perspective of the reviewer, this is the main page a user will see. Documents are sorted and grouped by New, In Progress, and Completed. Each document contains a Title, Author, Description, and Other Info. Each document can be viewed, opened for comments/edits, or deleted.

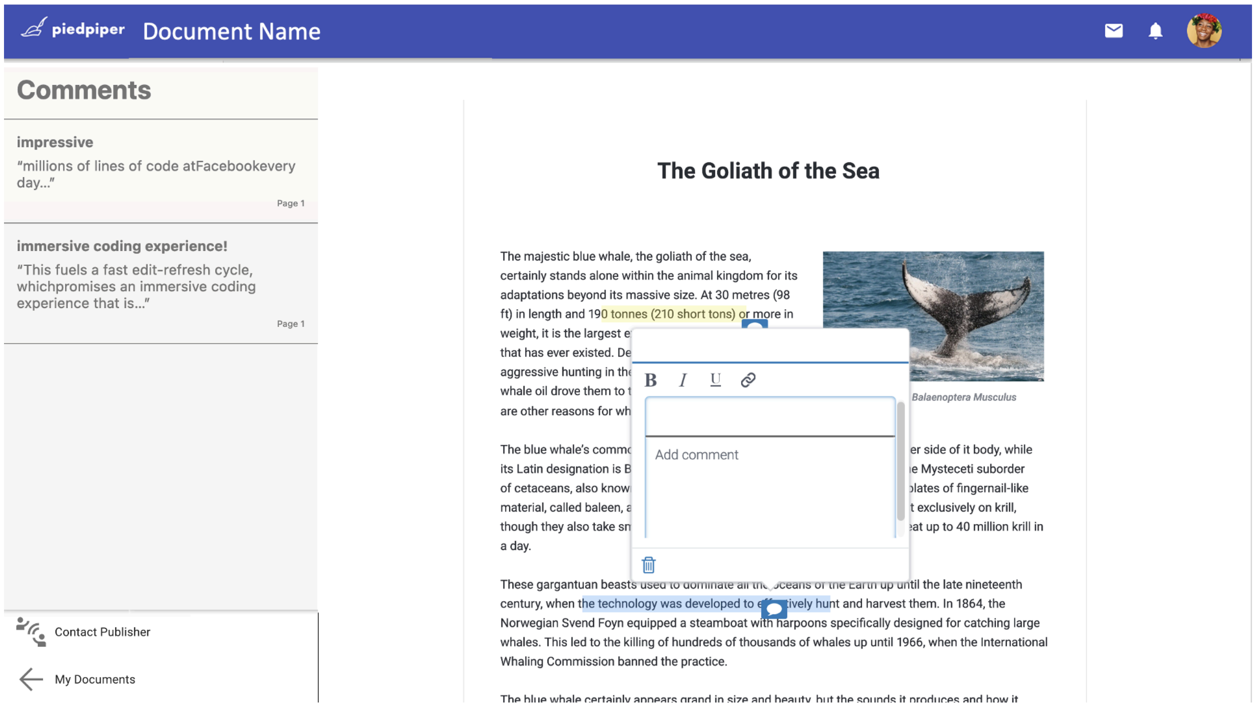
Dependencies

- MUI
- ReactJS

Quality Attributes

- UI aesthetics
- Mobile friendliness/responsiveness

2.3.2.2 “Document View” Mockup


piedpiper Document Name

Comments

impressive
"millions of lines of code atFacebookevery day..."
Page 1

immersive coding experience!
"This fuels a fast edit-refresh cycle, whichpromises an immersive coding experience that is..."
Page 1

Contact Publisher

My Documents

The Goliath of the Sea

The majestic blue whale, the goliath of the sea,
certainly stands alone within the animal kingdom for its
adaptations beyond its massive size. At 30 metres (98
ft) in length and 190 tonnes (210 short tons) or more in
weight, it is the largest e
that has ever existed. De
aggressive hunting in th
whale oil drove them to t
are other reasons for wh

Balaenoptera Musculus

B I U

Add comment

The blue whale's comm er side of it body, while
its Latin designation is B e Mysteceti suborder
of cetaceans, also know plates of fingernail-like
material, called baleen, a t exclusively on krill,
though they also take sn eat up to 40 million krill in
a day.

These gargantuan beasts used to dominate all the oceans of the Earth up until the late nineteenth
century, when the technology was developed to effectively hunt and harvest them. In 1864, the
Norwegian Svend Foyn equipped a steamboat with harpoons specifically designed for catching large
whales. This led to the killing of hundreds of thousands of whales up until 1966, when the International
Whaling Commission banned the practice.

The blue whale certainly appears grand in size and beauty, but the sounds it produces and how it

Abstract

From the perspective of the reviewer, this is a page a user will see when viewing a document for review. The reviewer can make a comment by highlighting some text; they can then leave a rich text comment. The reviewer can view all comments and navigate to a specific comment by clicking on it. There is also an option to contact the publisher and return to the previous page.

Dependencies

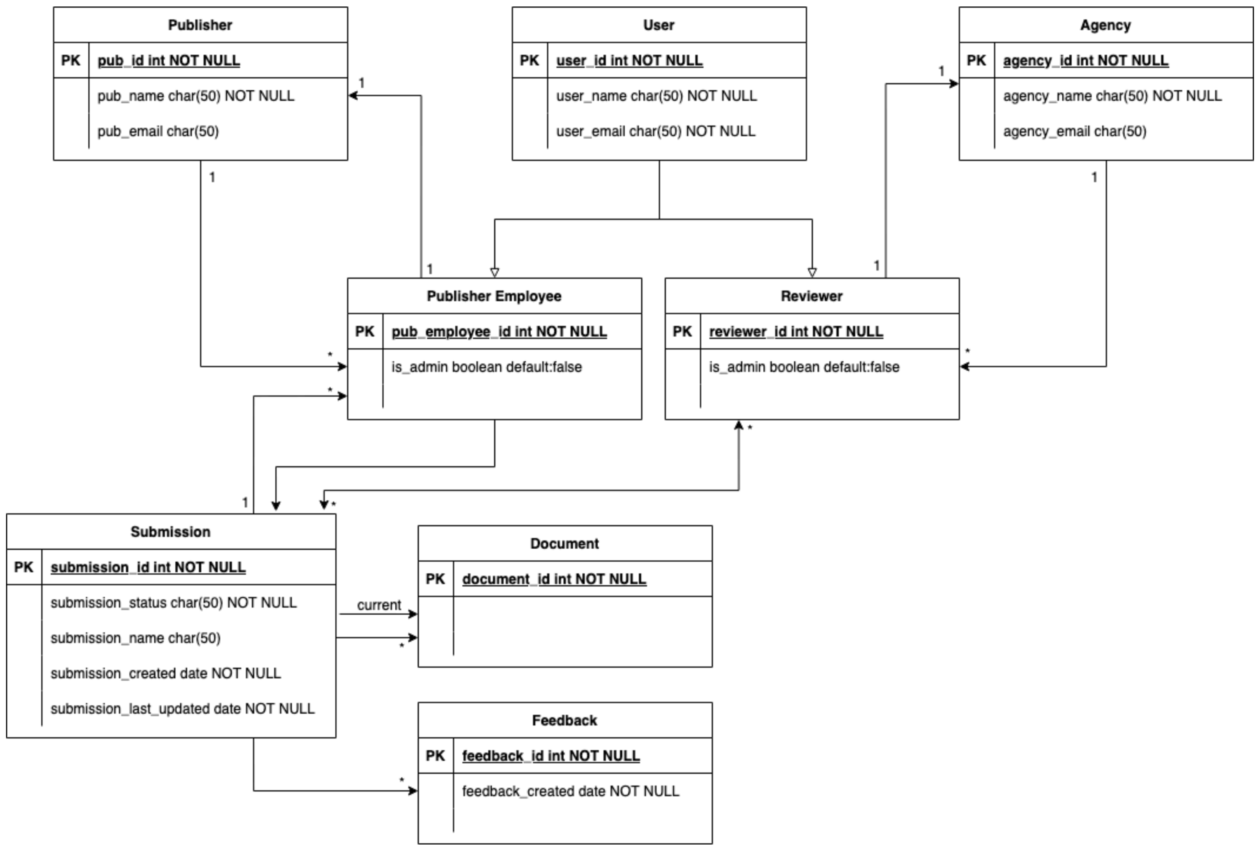
- react-doc-viewer
- MUI
- ReactJS

Quality Attributes

- UI aesthetics
- Mobile friendliness/responsiveness

2.4 Database

2.4.1 Entity Relationship Diagram


Publisher
PK
pub_id int NOT NULL
pub_name char(50) NOT NULL
pub_email char(50)
User
PK
user_id int NOT NULL
user_name char(50) NOT NULL
user_email char(50) NOT NULL
Agency
PK
agency_id int NOT NULL
agency_name char(50) NOT NULL
agency_email char(50)
Publisher Employee
PK
pub_employee_id int NOT NULL
is_admin boolean default:false
Reviewer
PK
reviewer_id int NOT NULL
is_admin boolean default:false
Submission
PK
submission_id int NOT NULL
submission_status char(50) NOT NULL
submission_name char(50)
submission_created date NOT NULL
submission_last_updated date NOT NULL
Document
PK
document_id int NOT NULL
Feedback
PK
feedback_id int NOT NULL
feedback_created date NOT NULL
current
1
*

Abstract

The database will be structured with three roles: Publisher, User, and Agency. Each of these has an Employee, and each agency has a Reviewer. Each also has a Document, Submission, and Feedback.

Dependencies

- MySQL Server and Database

Quality Attributes

- Compliance to relevant standards
- Reliable – for example up and running (99.98%)
- Available
- Maintainability, security patches
- Recoverability
- Accurate
- Secure