|  |  |
| --- | --- |
| ***Commfluence*** | ***Functional Specification*** |

# Detailed Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | CreateDocument.js | | |
| **Prepared by** | Jacob Hanson | **Date Prepared** | 10/17/2022 |
| **Effort Level** | Low | **Risk Level** | Low |
| **Executive Summary** | This functional specification describes the CreateDocument.js | | |

# APPROVERS/REVIEWERS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Functional Reviewer** | **Functional Review Date** | **Technical Reviewer** | **Technical Review Date** |
| 1.0 | Jacob Hanson | 4/25/2024 | NA | NA |
|  |  |  |  |  |

Note: Once this design is approved, any changes made must go through the Change Management process.

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Release Date** | **Author(s)** | **Summary of Changes** |
| 1.0 | 4/25/2024 |  | Jacob Hanson | Document Creation |
| 1.1 | 4/27/2024 |  | Jacob Hanson | Document Completed |
|  |  |  |  |  |

# Summary

The CreateDocument component is designed to facilitate the creation of new documents within a project and navigate to the newly created document. Leveraging React and React Router, this component integrates with Firebase Firestore for document management and utilizes authentication context for user identification.

# Solution Overview

Document Viewer

Solution Description

The Create Document component is a React-based solution tailored for creating new documents within a project environment. It enables users to initiate a new document creation process seamlessly, ensuring efficient project management. The core functionalities include:

* **Document Creation**: Automatically generates a new document upon component mounting.
* **Programmatic Navigation**: Utilizes React Router's use Navigate hook to redirect users to the newly created document after creation.
* **Firebase Integration**: Utilizes Firestore service functions for storing document data in the cloud database.
* **User Authentication**: Utilizes authentication context to identify the current user and associate document creation with their account.

# Screenshot:

(can add screenshot)

# Security Considerations

NA

# Functional Test Requirements

|  |  |
| --- | --- |
| **Test Step** | **Planned Results** |
| Open the application and navigate to the project page. | The project page should be displayed, showing options to create a new document. |
| Click on the "Create New Document" button. | The CreateDocument component should mount, initiating the document creation process. |
| Verify if a new document is automatically generated. | Upon mounting, a new document should be automatically created within the project environment. |
| Check if the user is redirected to the new document. | After document creation, the user should be redirected to the newly created document page using React Router's useNavigate hook. |
| Verify if the document data is stored in Firestore. | The document data, including its content and metadata, should be stored securely in Firestore for future reference and retrieval. |
| Ensure user authentication is enforced. | The component should utilize authentication context to ensure that only authenticated users can create documents, maintaining a secure environment. |
| Test document creation with various user accounts. | Users with different authentication credentials should be able to create documents successfully, with each document associated with the respective user. |

# Appendix A: Terminology

|  |
| --- |
| **Terminology** |
| Effort Level – level of effort includes both development and testing efforts and generally defined as follows   * Low (1-16 hours) * Medium (17-40 hours) * High (40+ hours)   Risk level – level of risk takes into account both technical risk as well as impact to business users and generally defined as follows   * Low (No risk to business and little or no technical issues) * Medium (Minimal impact to business and/or some technical risk) * High (Potentially large impact to business and/or considerable technical risk requiring off-hours transports such as structure or table changes)   Type of Development/Request – Development/request types are separated into three main areas as below   * + New (new development of technical object)   + Enhancement (change to existing technical objects due to new requirements)   + Defect (change to existing technical objects due to break-fix)   + Configuration-only (change does not require corresponding TS) |