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

# Detailed Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | Authentication.jsx | | |
| **Prepared by** | Jacob Hanson | **Date Prepared** | 4/22/24 |
| **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 | 4/26/2024 |  | Jacob Hanson | Document Completed |
|  |  |  |  |  |

# Summary

The Authentication component facilitates user authentication within the application, allowing users to sign in or create a new account. Leveraging Firebase authentication and Firestore integration, users can securely access the application's features. Key functionalities encompass:

# Solution Overview

User Authentication

• Sign-in Interface: Users are presented with a form to input their email and password for sign-in.

• Account Creation: Users have the option to create a new account by providing their email and password. • Toggle Functionality: Provides a toggle link to switch between sign-in and account creation modes.

Solution Description

• Authentication Component: The primary React component responsible for rendering the authentication interface and managing user interactions.

• Firebase Authentication: Utilizes Firebase authentication methods for sign-in and account creation.

• Firestore Integration: Upon successful account creation, user data is stored in Firestore for future reference.

• Routing: Redirects users to the projects page upon successful authentication.

# Screenshot:

(can add screenshot)

# Security Considerations

NA

# Functional Test Requirements

|  |  |
| --- | --- |
| **Test Step** | **Planned Results** |
| Open the application and navigate to the login page. | The login page should be displayed, showing the email and password input fields, and a button to submit the form. |
| Enter valid email and password credentials. | The email and password fields should accept input without any issues. |
| Click on the "Sign In" button. | If the credentials are correct, the user should be authenticated and redirected to the projects page. Otherwise, an error message should be displayed. |
| Click on the "Create an account" toggle link. | The page should switch to account creation mode, displaying input fields for email and password, along with a button to create the account. |
| Verify the creation of the account in Firestore. | Check if the user's data (UID and email) is correctly stored in Firestore after the account creation process. |
| Attempt to create an account with an invalid email. | If an invalid email format is entered, the system should display an error message indicating that the email is invalid. |
| Attempt to create an account with a weak password. | If a weak password is entered, the system should display an error message indicating that the password is not strong enough. |

# 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) |