**Technical Document**

1. Developed an LWC (Lightning Web Component) that iterates over a list of related sObjects and displays them within a customizable Lightning card. This component dynamically adapts to different record pages based on the provided configuration.
2. Enhanced the LWC metadata (meta.xml) file to allow the component to be added to record pages. Introduced configuration parameters, such as childObjectApiName, relationshipFieldName, and fieldSetName, enabling the component to be flexible across various Salesforce objects. These parameters can be set directly within the App Builder interface when adding the component to a page.
3. Built an Apex class, RelatedObjectRecordsController, that contains a method getRelatedRecords. This method retrieves the list of related sObject records by utilizing the childObjectApiName, relationshipFieldName, fieldSetName, and parentRecordId parameters. The class is designed to be reusable and extensible for different types of related object queries.
4. In the LWC JavaScript controller, the Apex method getRelatedRecords is invoked within the connectedCallback lifecycle method. The @api decorator is used to capture the recordId of the parent record, as well as the other configuration parameters set on the record page. These values are passed to the Apex method to fetch the relevant sObject records dynamically.
5. Upon receiving the response from the Apex method, the data is transformed in the processAllFieldsOfRecords JavaScript method. This method processes the retrieved records, mapping them into a format that displays field names and values, which can be customized further based on field set definitions.
6. Implemented caching strategies within the LWC component to improve performance, especially when dealing with a large volume of related records. The caching mechanism reduces server calls by storing previously retrieved records and only refreshing the data when necessary.
7. Added error handling and user-friendly notifications in the LWC component. If the Apex method encounters any issues during data retrieval, the component gracefully handles the error and displays an appropriate message to the user without affecting the overall user experience.
8. Provided detailed documentation and comments within the codebase to help understand
9. A test class is developed to validate the functionality of the Apex class.