

Sibling Account & Primary Contact Table

Version Control

Version	Date	Author(s)	Rationale
0.1	7/11/2020	Sai Wusirika	Implement VF Page skeleton with table Create Controller class and populate table with sibling accounts Started test class for controller Creating two new fields on Contact object
0.2	7/12/2020	Sai Wusirika	Added formatting changes to table, button formatting Added Wrapper subclass to controller Added and tested error handling functionality Finished test class for controller

Github Link to Repository: <https://github.com/swusirika/SiblingAccountTable>

Table of Contents

[Title Page](#)

[Version Control](#)

[Project Scope](#)

[Functional Specifications:](#)

[Functional Modifications](#)

[Technical Design](#)

[Org Specifications](#)

[Security](#)

[Future Release Goals](#)

Project Scope

- Requirements:
 - Must include a Visualforce page that includes a search field for an account Id and the corresponding table of sibling accounts (which have the same parent account).
 - Must also include the primary contact for each sibling account in the same row in the table
 - Primary contact information must be editable with a save feature
 - Rows in table must be selectable
 - Must have a button to change selected primary contacts' status to Contacted
 - Must have error handling for all the above features
 - Must include test classes for all features

Functional Specifications:

The following is a complete overview of different functionalities that are part of this implementation including search fields, buttons and table modifications.

- Account Search:
 - a. Account Search Field:
 - i. Accepts Account Name or Id to search for sibling accounts which have the same parent Account
 - b. Find Sibling Accounts Button:
 - i. Visualforce Command button to calls function in controller class to:
 1. Query using Account's name or ID user entered in search box to find account & that account's parent account Id
 2. Use Parent account Id to find all child accounts except for account used in search and their associated primary contacts
 3. Populate List of contact records to show in table
 4. Considers the following error cases:
 - a. If account with name, Id searched does not exist
 - b. If account searched does not have a parent account
 - c. If account searched does not have sibling accounts
- Sibling Accounts and Primary Contacts Table:
 - a. Table Columns:
 - i. Checkbox for each row to select each row
 - ii. Account Name with hyperlink to account detail page
 - iii. Contact information for primary contact all of which are editable
 - b. Mark as Contacted button:
 - i. Command button to call function in controller class to:
 1. Access the primary contact associated to Sibling account in each row

2. See if it that row is selected and change the Contact record's new custom field "Status" to "Contacted"
- c. Save button:
 - i. After the user has changed any of the primary contact fields. Clicking this button will call save function in controller to update primary contact fields on their respective records

Functional Modifications

The following modifications were made because of requirement gaps or to improve user experience. Reasons for each modification are provided below:

Functionality	Modification	Reason
Account Search	Search using Account Name or Account Id	To improve user experience Easier for users to use the Account name rather than the Id
Primary Contact Name	Split Contact Name field into two fields for First and Last Name (required)	Contact Name field cannot be modified since it is based on First & Last name Including both First & Last name fields is a simplest implementation to allow Contact name changes
Multiple Primary Contacts	Allows table to show multiple primary contacts associated to sibling accounts if they exist	Otherwise only the first primary contact will be shown in table
Metadata Changes	Added Status & Tax ID fields to Contact	Fields need to be shown in Table but are not standard fields on contact

Technical Design

The following is a complete technical design of the metadata changes, visualforce page & controller used to implement the solution.

- Metadata Changes: Two New Fields
 - Contact Object:
 - Status__c field [String]: Contacted, Not Contacted
 - Tax_Id__c field [String]
- Visualforce Page: SiblingAccountTable
 - PageBlock section for messages
 - Clicking on messages will automatically clear them
 - Contains two page blocks: one to search for an account and another to show to sibling account table
 - Account Search block:
 - Input Text for search key (either Account Name or Id used to search for account & sibling accounts)
 - Command button to run SiblingAccountTableController.getSiblingAccounts function and rerender the output panel for messages and the output panel for the table
 - Sibling Account & Contact Table:
 - Contains two output panels:
 - if contactWrapperList is populated
 - Populates pageBlockTable using contactWrapperList
 - Command button to Mark as Contacted: SiblingAccountTableController.markContacted function
 - Rerenders messages & table panel
 - Command button to Save: calls SiblingAccountTableController.save function
 - Rerenders messages & table panel
 - contactWrapperList is not populated: Show message that there are no Sibling accounts or related primary contact records associated to the searched account
 - Table Columns:
 - inputCheckbox to select a row (references contactWrapper's boolean selected field)
 - Account Name: output link to Account detail page
 - Contact First & Last Name: both editable
 - Contact Status: editable
 - Contact Mobile Phone Number: editable

- Contact Email: editable
 - Contact Tax Id: editable
- Controller: SiblingAccountTableController:
 - Variables:
 - siblingPrimaryContactList: private list of primary contacts
 - searchKey: public value that user enters to search for sibling accounts
 - contactWrapperList: public list of contactWrapper records each with their associated contact & Boolean selected value
 - updateStatusContacts: private list used to update primary contacts' status when markContacted function is invoked
 - Constructor: initiates new contactWrapperList & updateStatusContacts
 - getSiblingAccounts:
 - Clears contactWrapperList
 - Queries for account & checks if account or account's parent Id field is null (error handling for each)
 - Populates siblingPrimaryContactList from query for primary contacts
 - Populates contactWrapperList based on values in siblingPrimaryContactList
 - Save:
 - Updates siblingPrimaryContactList with error handling
 - markContacted:
 - Clears updateStatusContacts
 - Goes through contactWrapperList and sees which rows are selected in the table
 - Updates related Contact's Status to "Contacted" with error handling
 - Clear:
 - Clears the contactWrapperList
 - clearMessage:
 - Clears all messages on the visualforce page
 - Used to clear messages from messages output panel when user clicks on message
 - selectAll:
 - Used to select all rows at once (was unable to implement due to time but tabled to be included in a future release)

Org Specifications

- Dev Org: write code for controller, visualforce page, & import account records for test data
 - Have all dev users with proper permissions for Account & Contact objects
 - Create test classes for Controller, handler, and helper classes
 - Create temporary test data for devs to run unit tests on controller
- QA Org: using git branch, import code to org
 - Import account data and contact data using data Loader into org
 - Link Accounts & Contacts used in each test case to their respective JIRA
- Production Org:
 - Import code, test classes (after 75% code coverage has been achieved for all test classes) and import real data for testing using data Loader

Security

This section will cover permissions for objects, fields, visualforce pages & apex classes used in this implementation

- Need to identify which user profiles will be using the visualforce page
- Create Permission Set: Sibling Account Permissions
 - Add required user profiles to permission set
 - Give access to following in permission set:
 - Visualforce Page: SiblingAccountTable
 - Apex Class: SiblingAccountTableController permissions are automatically included when adding the visualforce page
 - Contact Object: Read/write for object & fields
 - FirstName, LastName, Name, Status, Mobile Phone Number, Email, Tax Id
 - Account Object: Read access to object & fields
 - Account Name, ParentId
- For External Partners:
 - Add External partner as user in Salesforce org with new Profile “External Partner”
 - Add “External Partner” profile to “Sibling Account Permissions” permission set
 - This will allow the external partner to access/use the Sibling Account Table visualforce page & make sure that partner does not have access to any other functionality in the org.

Future Release Goals

The following is a list of future modifications to improve user experience or add new functionality to the design

- Table:
 - Adding Select All functionality to Table so user can select full list of rows at once
 - Already created function in controller
 - Need to add javascript function to select/deselect all rows when inputCheckbox associated to selectAll is selected/deselected
 - Ordering all row by ascending or descending based on clicking column label
 - Will require one function in controller:
 - changeOrder: with parameters (field associated to column, descending or ascending)
 - Undo button: to undo last change made while clicking Save or Mark as Contacted buttons
 - Will require two sets of contactWrapper List (currentContactWrapper, lastContactWrapper) to keep previous version of records in table
 - Undo function in controller class to set currentContractWrapper to lastContactWrapper