Module 6 Lab B

Using the Xrm.Page Object model to implement complex validation logic

Scenario

On entering new records for bank account, the field values need to be validated, record saving to be cancelled and the user informed on the error. The rules are:

Savings accounts must have an initial new\_balance of 100 or more

Sortcode must prefix according to the country 044 for UK and 01 for USA

The accountnumer is only editable when on record creation, else it isn't.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | Create a new script file Lab6bStart.js, and enter the following functions.  OnFormLoad, and LockAccountNumber as shown below. | | 2 | Call the LockAccountNumber() from OnFormLoad | | 3 | In LockAccountNumber check if the form type is Create (value=1) using Xrm.Page.ui.getFormType()  If it is use  Xrm.Page .getControl(“new\_accountnumber”).setDisabled(false)  to enable the control on record creation  Otherwise setDisabled(true). | | 4 | Create 2 more functions in the same pattern, CheckLocationAgainstSortCode and CheckFundsForAccountType. | | 5 | In CheckLocationAgainstSortCode, get the values of the new\_sortcode and new\_banklocation, and check that  if the locaton.toUpperCase()==”UK” and sortcode.substr(0,3)==”044” then clearNotification  else  setNotification.  Repeat this for locaton.toUpperCase()==”USA” and sortcode.substr(0,2)==”01 | | 6 | In CheckFundsForAccountType select the value of the new\_accounttype optionset using Xrm.Page.getAttribute("new\_accounttype").getSelectedOption().value  and the value of the new\_balance using  Xrm.Page.getAttribute("new\_balance").getValue().  Use conditional logic for the accounttype being 100000000 (Savings) and the balance is less than 50 to display / hide a notification. | | 7 | Modify the QALab.OnFormLoad() function to register the OnChange Functions for the SortCode ,Location, Balance and AccountType fields as shown below. By doing this you will avoid having to register these individually for the OnChange event for each of these controls.  QALab.OnFormLoad = function () {  QALab.LockAccountNumber();  Xrm.Page.getAttribute("new\_sortcode").addOnChange(QALab.CheckLocationAgainstSortCode);  Xrm.Page.getAttribute("new\_banklocation").addOnChange(QALab.CheckLocationAgainstSortCode);  Xrm.Page.getAttribute("new\_balance").addOnChange(QALab.CheckFundsForAccountType);  Xrm.Page.getAttribute("new\_accounttype").addOnChange(QALab.CheckFundsForAccountType);  }; | | 8 | Upload the Javascript file as a webresource and associate the QALab.OnFormLoad function with the OnLoad event of the BankAccount Main form.      Note that we do not have to "wire up" any other event handlers as this is performed by the QALab.OnFormLoad function. | | 9 | Save and then publish your changes and test that the restrictions are implemented correctly by creating and editing a number of bank account records.  e.g. | | 10 | Remove the script association with the form as you did at the end of the previous lab. | |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |