**Internal Revenue Service**

**Data File Handler**

SOI

Contents

[Project Members 3](#_Toc1023373508)

[Project Summary 3](#_Toc1858125210)

[Requirements 5](#_Toc1221484335)

[System Specification 12](#_Toc1448121135)

[Flow chart 13](#_Toc1336551295)

[Standard Operating Procedure 14](#_Toc500449681)

# Project Members

David Tran (Project Lead/Mentor)

Gregorio L Doble (Project Lead/Mentor)

Mason B Doan (DCOS FO Student Volunteer/Programming Intern)

Rudy C Fuentes (DCOS FO Student Volunteer/Programming Intern)

Zachariah Zachariah (DCOS FO Student Volunteer/Programming Intern)

# Project Summary

This program will be used to verify or convert data files. The datafiles can be in text or xml format.

For verification, the program will 1) verify that a datafile conforms to a given record layout, and 2) verify that the information in the datafile is the same as results of direct queries from the source database. Verification involves reading a record layout and parsing a data file based on the information in the record layout. For a text data file (.dat or .txt) the record layout can be a txt file or an excel file. For an xml data file, the record layout will be an xsd file. The program will allow the users to choose numeric fields that they want to sum up for comparison with the database results. The users can also choose to extract a parsed-out data for one or more records. Verification fails if the record length in the data file does not match the length in the record layout. Verification will also fail if a numeric field gets populated with non-numeric characters. For verification against the database, the program will only provide the totals of the selected fields. The results can then be used to compare against the results from the database queries.

Conversion applies only for text data file. Text data files will be converted into an xml file. It involves a two-step process. It will first do a verification – reading a record layout and parsing the text datafile – then transforming the data into an xml format by wrapping it with the appropriate xml elements. As part of the conversion process, the program will allow the user to choose whether to create an xsd file that can be used to read the resulting xml data file.

Users should be able to choose between the two main operations, converting a file or verifying a file.

# Requirements

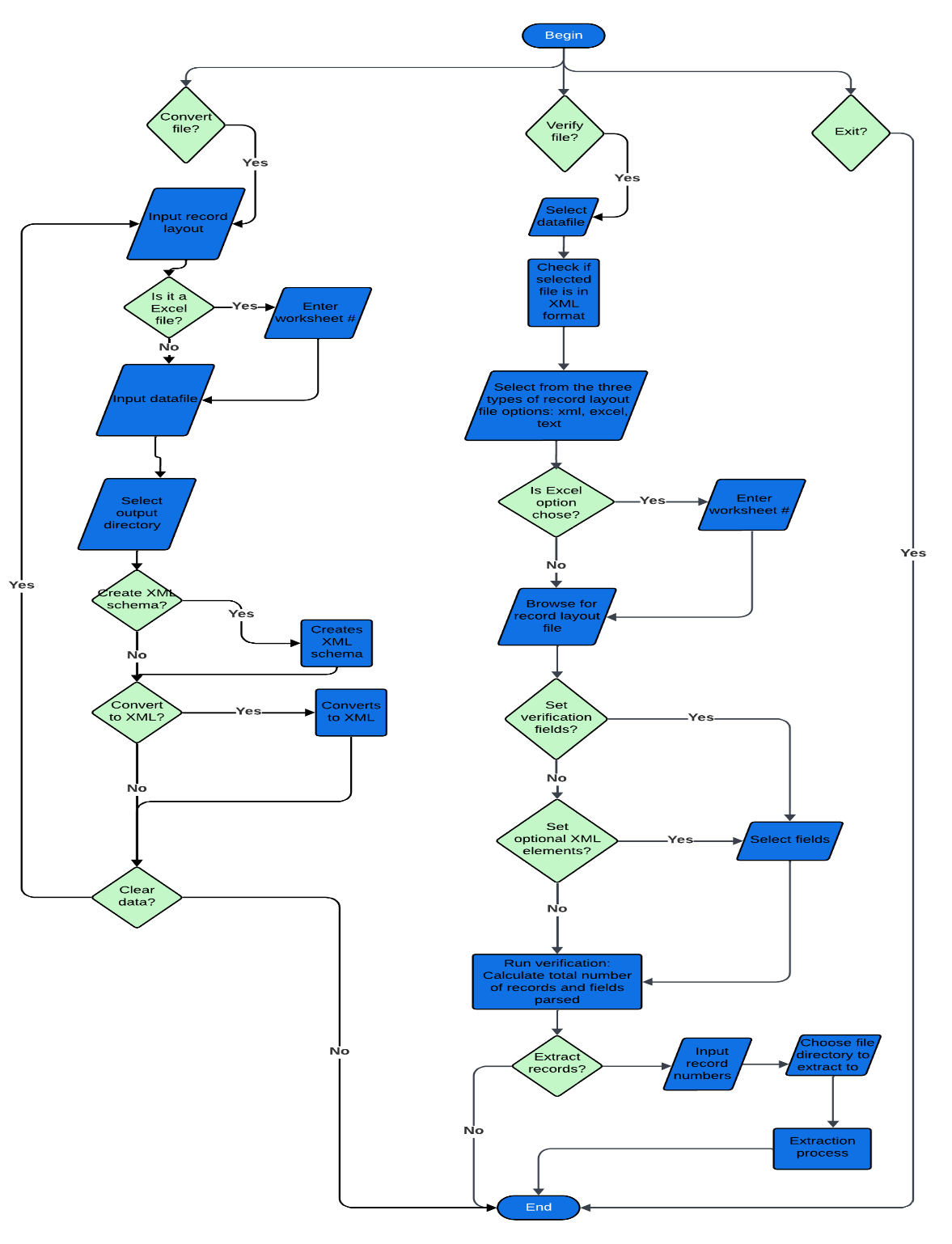
|  |
| --- |
| Program control flow |
| 1. Have to be able to choose between convert file and verify file and switch to the proper tool  |  |  | | --- | --- | | Use case | Test case | | **PCF001-1**: User can toggle between the two program functionalities of converting and verifying a file  **PCF001-2**: Once an option is selected, load new form data to perform operation related to what the user selected | **PCF-T001-1**: Able to determine if user wants to verify or convert file and get that input from the user  **PCF-T002-2**: Able to load in the selected tool for the user to interact with | |
| 1. Have to be able to exit the program  |  |  | | --- | --- | | Use case | Test case | | **PCF002**: User has the ability to exit out of the application/program resulting in exiting out of the program window and having all associated information erased | **PCF-T002-1**: Verify if all the information related to program input is erased and set to default  **PCF-T002-2**: Verify if the program window closes without any errors | |

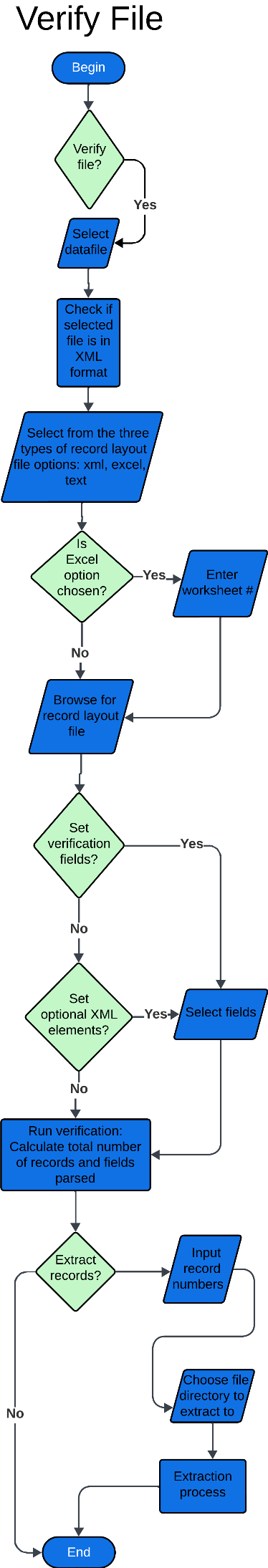
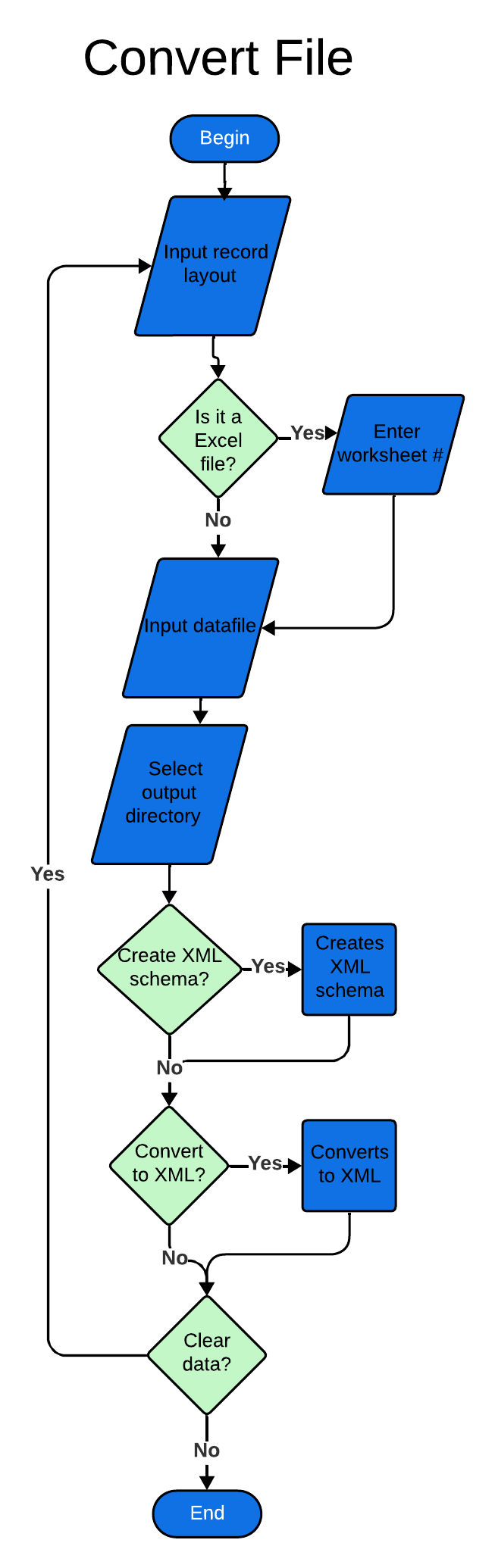
|  |
| --- |
| Converting a file |
| 1. Have to be able to read in an input record file from user that can be a .dat, .txt, or have no file extension  |  |  | | --- | --- | | Use case | Test case | | **C001**: Open a file browser to let the user select a file from their computer to serve as the input record layout | **CT001-1**: User is able to open a file browser and load in their file, this operation should complete with no error  **CT001-2**: verify that the file is of type .txt, .xsd, xls, xlsx, or have no file extension  CT001-3: veriry that if xml | |
| 1. Have to be able to specify worksheet number WITH record layout file with possible values range between 1 and 100 inclusive  |  |  | | --- | --- | | Use case | Test case | | **C002**: User can input a number between 1 to 100 (inclusive) for the specific Excel sheet number | **CT002**: Able to receive number from user and able to verify that the number is between 1 and 100 inclusive. | |
| 1. Have to be able to read in an input data file from user that can either be a .txt, .dat, or have no file extension  |  |  | | --- | --- | | Use case | Test case | | **C003**: Open a file browser to let the user select a file from their computer to serve as the input data file | **CT003-1**: User is able to open a file browser and load in their file and it should be the proper file type of either .txt, .dat,, .xml, or no file type  **CT003-2**: if XML data file is selected, .XSD must be the record layout file | |
| 1. Have to be able to get output directory from user so we know where to save the operation results  |  |  | | --- | --- | | Use case | Test case | | **C004**: User can open up a File explorer so the user can find the proper output directory | **CT004-1**: Verify that after the user clicks to choose directory, we get an output directory  **CT004-2**: Once we get a selected directory from user, verify that it is a valid directory that exists in the system | |
| 1. The operation can either be “create xml schema” or “convert to xml” 2. Have to be able to create xml schema and save to output directory as a file (xsd) 3. Have to be able to convert to xml and save to output directory as a file (xml) 4. Have to be able to create xml schema and convert to xml at the same time and create 2 files to the output directory (both xsd and xml)  |  |  | | --- | --- | | Use case | Test case | | **C005**: User can select one of these three operations to perform: “create xml schema”, “convert to xml”, and “create xml schema and convert to xml” | **CT005-1**: Verify that we get an option selected from the user to decide which operation to perform  **CT005-2**: The selected operation should perform and finish with no errors | |
| 1. Have to be able to clear out all of the user input  |  |  | | --- | --- | | Use case | Test case | | **C006**: User can clear out all of the data they have entered | **CT006**: Verify that all of the information the user entered is not present in any of the forms, textboxes, or that ask for user input | |

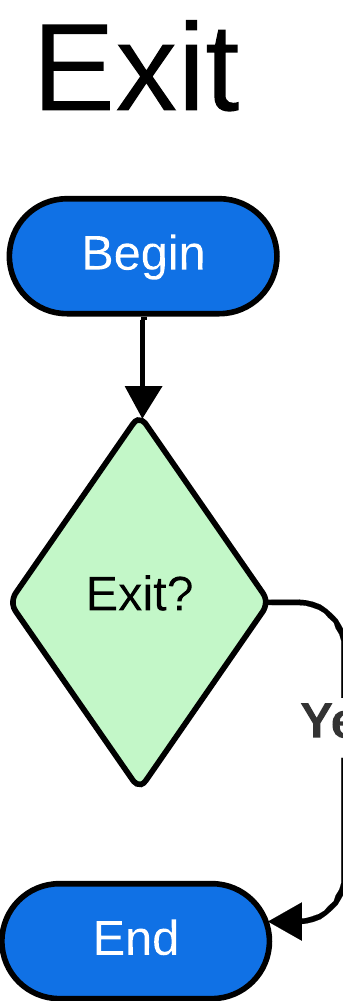
|  |
| --- |
| Verifying a file |
| 1. Have to be able to select a user specified datafile, it can be in .txt, .dat or .xml format  |  |  | | --- | --- | | Use case | Test case | | **V001**: Open a file browser to let the user select a file from their computer  **V002**: Allows user to display file path of selected file | **VT001-1**: Verify if the user can open a file browser  **VT001-2**: Verify if the user can select the three file types .txt, .dat or .xml  **VT001-3**: Verify if the user selected file is read in correctly with no errors  **VT001-4**: Verify if the file path is displayed | |
| 1. Have to be able to read in a datafile from user where the data is in xml format  |  |  | | --- | --- | | Use case | Test case | | **V002**: User can select datafiles types of .txt or .xml where the data is in xml structured format | **VT002-1**: Verify if the user selected datafile is of .txt or .xml extension  **VT002-2**:Verify if the datafile data is in valid xml format | |
| 1. User should specify record description file type from 3 options (XML Schema, Excel Record Layout (DERL), or Text Data Record Layout (DERL))  |  |  | | --- | --- | | Use case | Test case | | **V003-1**: Users can select the file type from 3 different options which are XML Schema, Excel Record Layout, or Text Data Record Layout to read in their record layout  **V003-2**: User can input a number between 1 to 100 (inclusive) for the specific Excel sheet number | **VT003-1**: Verify if the user can select their desired record layout file type option  **VT003-2**: Verify if the Excel Worksheet number that is entered can be read and is valid  **VT003-3:** Verify if the Excel Worksheet number is stored in the program | |
| 1. Have to be able to select a record layout file based on file type the user has selected  |  |  | | --- | --- | | Use case | Test case | | **V004-1**: User can select the record layout file of the type they have chosen as mentioned in V003  **V004-2**: Allows file path of user selected record layout file to be displayed | **VT004-1**: Verify if the user can open their file directory browser to select the file with no errors  **VT004-2**: Verify if the user can select the file  **VT004-3**: Verify that the file selected by the user matches the file type that they selected for the record description file type  **VT004-4**: Verify if the correct file path is displayed | |
| 1. Have to be able to load in record description and display that info to users  |  |  | | --- | --- | | Use case | Test case | | **V005-1**: User can load in their record description file to be displayed  **V005-2**: User should be able to sort the data of the record description file being displayed by position, selection, name, length and type (Integer, Character) | **VT005-1**: If file opened with no errors, then ensure you can display that info to users in a table format  **VT005-2**: Once table is displayed in the table format, the table should be sorted when user clicks on sort by position, selection, name, and type and should change the way the info is presented in the table | |
| 1. User should be able to set verification fields in table once the record description is loaded  |  |  | | --- | --- | | Use case | Test case | | **V006**: User can select fields from the record description table to set up for the verification process | **VT006-1**: Verify that the user can toggle set selection of the verification fields on and off  **VT006-2**: Verify that when toggled on and off, user can select fields for verification on the table when it is toggled on and NOT select fields for verification when it is toggled off | |
| 1. Set optional xml elements (to be used with xml verification), user should be able to set which fields are optional (meaning they may or may not be in the data)  |  |  | | --- | --- | | Use case | Test case | | **V007**: User can (only if they are doing xml verification) set the optional xml elements | **VT007-1**: Verify that the user can toggle set optional elements on and off  **VT007-2**: Verify that when toggled on and off, user can select fields so they are optional when it is toggled on and NOT select fields so they are optional when it is toggled off | |
| 1. User should be able to run verification once they have selected the verification fields and after verification is run, display that information to users  |  |  | | --- | --- | | Use case | Test case | | **V008**: User has the ability to run the verification process | **VT008-1**: Verification process should be started after button is clicked  **VT008-2**: Any results from the verification process should not crash the program and should be displayed to user  **VT008-3**: Verification process should finish | |
| 1. To troubleshoot any verification problems, user should be able to extract data to visually inspect data to see if there is any formatting errors  |  |  | | --- | --- | | **Use case** | **Test case** | | **V009**: Have an option to extract data where user can enter in the record number range to extract the data from | **VT009-1**: Verify that start of range and end of data range has been provided by the user  **VT009-2**: Should be able to extract data and display to user with no problem | |
| 1. After performing a successful verification operation, user should be able to save that information to output directory  |  |  | | --- | --- | | Use case | Test case | | **V010:** User can save their verification data after the verification has been performed | **VT010-1**: Should save after user selects option to save verification info  **VT010-2**: Should save file to desired output directory from user with no errors | |
| 1. Have to be able to clear out all selected options/input from user  |  |  | | --- | --- | | Use case | Test case | | **V011**: User can clear out all of the data they have entered | **VT011**: Able to clear out all of the user input data in the textboxes and selections |  1. Dynamic GUI updates  |  |  | | --- | --- | | Use case | Test case | | **UI001:** Implement real-time GUI updates to reflect the progress of the file operation(s) | **UIT001:** Progress bar accurately reflects the current status of file operations without freezing the UI |  1. Contextual help and tooltips  |  |  | | --- | --- | | Use case | Test case | | **UI002:** UI--Provide contextual help and tooltips for complex features to assist users. | **UIT002:** Hovering over a feature displays tooltip explaining its function |  1. Undo/redo actions for settings  |  |  | | --- | --- | | Use case | Test case | | **UI003:** Implement undo/redo functionality for configurable settings prior to file processing | **UIT003:** User can revert or reapply changes made to configurable settings prior to file processing | |

# System Specification

## Flow chart







## Program control flow

1. **Have to be able to choose between convert file and verify file and switch to the proper tool**

Provide a method to select either the convert file or verify file process.

If selection = “convert”

Perform “convert file process”.

Else if selection = “verify”

Perform “verify file process”.

End if

**2. Have to be able to exit the program**

If user select “exit”

           Open message box “Are you sure you want to exit?”

           If answer = “yes”

End all process.

Close all windows.

           Else

              Close message box.

              Return to current screen.

          End if

End if

## Convert file

1. **Have to be able to read in an input record file from user that can be a .dat, .txt, or have no file extension**

Prompt user to select a file from their computer

Display only in browser directory files with “.dat” or “.txt" extensions or those with no extensions

selectedFile = get file from the user

If selectedFile is null

display error message: “no file selected”

prompt user to select another file

Else

display selectedFile name in textbox

get file extension

If file extension is null or ".dat" or ".txt"

process selectedFile

Else

display error message "invalid file format..."

End if

End if

**2. Have to be able to specify worksheet number with record layout file with possible values ranging between 1 and 100 inclusive**

If file extension is .xlsx (Execl File)

prompt user to enter a Excel worksheet #

If sheet number >= 1 or sheet number <= 100

If worksheet# is a valid worksheet# in the file

store the number and process file

Else

display error message “Invalid Worksheet #”

prompt user to enter a new number

End if

Else

display error message “Enter # between 1 and 100 only”

prompt user to enter a new number

End if

End if

**3. Have to be able to read in an input data file from user that can either be a .txt, .dat, or have no file extension**

Prompt user to select a file from their computer

Display only in browser directory files with .dat or .txt extensions or those with no extensions

selectedFile = get file from the user

If selected file is null

display error message: “no file selected”

prompt user to select another file

Else

display selectedFile name in textbox

get file extension

If file extension is null or ".dat" or ".txt"

process selectedFile

Else

display error message "invalid file format..."

End if

End if

**4. Have to be able to get output directory from user so we know where to save the operation results**

Prompt user to select an output directory from their computer

Display output directory browser

selectedDirectory = get directory from the user

If selected directory is null

display error message: “no valid directory selected”

prompt user to select another directory

Else

display selectedDirectory name in textbox

process directory

End if

**5. The operation can either be “create xml schema” or “convert to xml”**

If create xml schema

create xml schema and save to output directory as a file (“.xsd”)

Else if convert to XML

convert to xml and save to output directory as a file (xml)

Else if create xml schema & convert to XML

create xml schema and convert to xml at the same time and create 2 files to the output directory (both xsd and xml)

End if

**6. Have to be able to clear out all of the user input**

If clear is selected

clear input fields

clear variables

clear any other data

End if

## Verify file

1. **Let user select a file via the file explorer**

If file is in .txt, .dat, or .xml format

accept the file

Else

prompt user to select a file that is in the proper format

endif

1. **Have to be able to read in a datafile from user where the data is in xml format**

If file is in xml format

accept file

check record layout file

If record layout file is in xsd

end select datafile

Else

prompt user: record file must be in xsd format

Endif

End if

1. **User should specify record description file type from 3 options (XML Schema, Excel Record Layout (DERL), or Text Data Record Layout (DERL))**

Let user select an option for the record description file type from 3 options

Op1: XML Schema

Op2: Excel Record Layout

Op3: Text Data Record Layout

If user does not select an option for the record description file type

prompt the user to select an option

Endif

1. **Have to be able to select a record layout file based on file type the user has selected**

Let user select a file

Check to see if file type matches the record description file type selected from user in previous step

If it matches, keep user's file selection

display record layout file path to user

Else

inform user to select the right file type

Endif

1. **Have to be able to load in record description and display that info to users**

Let user load in record file

Attempt to open file provided by user in previous step

If file opens with no errors

Display the info to user in a table

When table is open, let user sort how records appear in the table by position, selection, name, length, and type (Integer, character, float)

Else

notify user that there was an error opening file and to ensure previous steps were done correctly

Endif

1. **User should be able to set verification fields in table once the record description is loaded**

Let user select fields manually, select all numeric fields, or select all fields from the record description table to be verified in the verification process

Keep track of what fields have to be verified in the verification process

1. **Set optional xml elements (to be used with xml veriication), user should be able to set which fields are optional (meaning they may or may not be in the data)**

Let user select fields from the record description table to be verified in the verification process

Keep track of what fields have to be verified in the verification process in a data structure

1. **User should be able to run verification once they have selecgted the verification fields and after verification is run, display that information to users**

Let user run verification process

If verification process runs without errors

display results to user

Else

Show where in the record there was an issue so that the user can manually

extract data and see where in the datafile there was an error

Endif

1. **To troubleshoot any verification process problems, user should be able to extract data to visually inspect data to see if there is any formatting errors**

Let user enter in start and end for the data extraction range

Do data extraction

If data extraction is able to complete with no errors

Display info to user

Else

Ask user to verify that they entered in a correct start and end for data extraction range

Endif

1. **After performing a successful verification operation, user should be able to save that information to output directory**

Let user select a path to save output from the verification process

If the path provided by the user exists and is valid, continue to save file

If the file saves to output directory with no issues

Tell user the output was successfully saved

Else

Tell user that it could not be saved and to check previous steps for any errors and try again

Endif

Else

Ask user to select a valid path

Endif

1. **Have to be able to clear out all selected options/input from user**

Let user click a button to clear out all of the information entered into the program

Reset variables that represented user input from the form and clear out the view for the user to enter in new data

Let user click a button to clear out all of the information entered into the program

Reset variables that represented user input from the form and clear out the view for the user to enter in new data

Pseudocode:

Function clearData()

UserInput1 = null

UserInput 2 = null

ClearInputFields()

Function clearINputFields()

Input.field1.text = “”

InputField2.text = “”

# Standard Operating Procedure