Relocation Manager Documentation

(Relocation Manager Database)

**Student: Sebastian Vowels**

**Student Number: 472372854**

**Date: 26/03/21**

Unit: ICTPRG418

Teacher: Mark O’Reilly

Contents

[IPO Chart 3](#_Toc67649030)

[TOE Chart 4](#_Toc67649031)

[Stylised Data Flow 4](#_Toc67649032)

[Selected Pseudo Code 5](#_Toc67649033)

[Displaying and Saving Records 5](#_Toc67649034)

[DisplayRecord (CurrentRecord) 5](#_Toc67649035)

[SaveRecord (CurrentRecord) 5](#_Toc67649036)

[File data handling methods – read and write file 6](#_Toc67649037)

[ReadFile 6](#_Toc67649038)

[WriteFile 6](#_Toc67649039)

[Navigation Button Code 7](#_Toc67649040)

[First 7](#_Toc67649041)

[Last 7](#_Toc67649042)

[Previous 7](#_Toc67649043)

[Next 8](#_Toc67649044)

[Navigation Button Code 9](#_Toc67649045)

[New 9](#_Toc67649046)

[Save 9](#_Toc67649047)

[Exit 9](#_Toc67649048)

[Delete 9](#_Toc67649049)

[Find 10](#_Toc67649050)

[Structured Chart(s) 11](#_Toc67649051)

[Managing the GUI elements 11](#_Toc67649052)

[Managing the user Actions 11](#_Toc67649053)

[Class diagram 12](#_Toc67649054)

[Testing 13](#_Toc67649055)

[Test Plan 13](#_Toc67649056)

[Test Cases: 14](#_Toc67649057)

[Evidence of Debugging 18](#_Toc67649058)

[System generated program documentation 19](#_Toc67649059)

[User Manual for: Relocation Manager 20](#_Toc67649060)

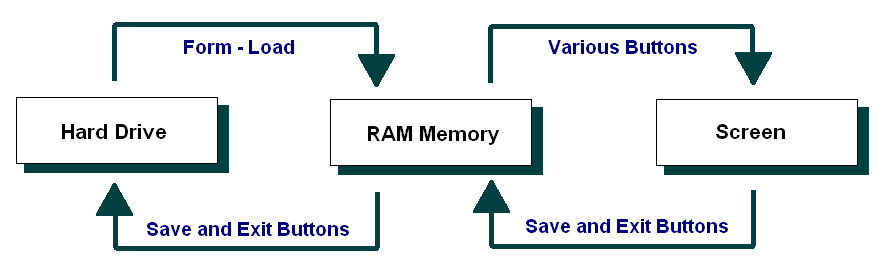
## IPO Chart

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Process** | **Output** | **Storage** |
| ‘*Records*’ (entries) from the data file:   * Contact Name * Contact Type * Contact Phone * Contact Web/Email * Contact Notes | * Open the file * Read the first entry * If not all entries have been loaded, load the next entry * Record the total number of entries read in. | * Show first record (entry) on screen | Data File  Temporary storage in RAM memory |
| * **New** button, then… * Contact Name * Contact Type * Contact Phone * Contact Web/Email * Contact Notes   Then,,,   * **Save** button | * Clear the screen ready for new record to be entered. * Allow user to enter information * Save the record (entry) back to RAM memory | New record (entry):   * Contact Name * Contact Type * Contact Phone * Contact Web/Email * Contact Notes | Updated records temporarily stored in RAM memory |
| * Search String and **Find** button | * Go through the record (entry) list until the Contact Name matches the entry in the Search text field. | * Record (entry) is shown on screen |  |
| * Record (entry) to be deleted presented on screen and then the **Delete** button. | * Remove entry from memory (replace array with new array minus current entry) * Set the number of entries in the list to current number – 1. | Display Blank Entry | Updated records temporarily stored in RAM memory |
| * **Save** button | * Save all the entries in RAM Memory into the specified output file. | Export of all records (entries). | All entries saved in a new copy of the data file. |
| * **Exit** button | * Save all data (as per Save button) and… * Close program | Export of all records (entries). | All entries saved in a new copy of the data file. |

## TOE Chart

|  |  |  |
| --- | --- | --- |
| **Task** | **Object** | **Event** |
| Load Contact Info Records | Form | Open |
| Clear screen ready for new entry | **New** button | Click |
| Save the current entry and then all entries to a clean copy of the data file | **Save** button | Click |
| Find the current entry | **Find** button | Click |
| Delete the current entry | **Delete** button | Click |
| Locate the first entry in the list and display it on screen | **First** (|<) button | Click |
| Locate the previous entry in the list and display it on screen | **Previous** (<)button | Click |
| Locate the next entry in the list and display it on screen | **Next** (>) button | Click |
| Locate the last entry in the list and display it on screen | **Last** (>|) button | Click |
| Save the current entry, then save all entries to a clean copy of the data file, and then close the program. | **Exit** button | Click |

## Stylised Data Flow



## Selected Pseudo Code

### Displaying and Saving Records

In line with our earlier data flow diagram and structured chart, we might start with a couple of the supporting methods that are called repeatedly.

In particular, let’s plan the method that Displays a Record, and the one that updates a record in RAM memory.

The pseudo code for these might be as follows:

### DisplayRecord (CurrentRecord)

Copy the Contact Name for the currentRecord to the Contact Name text field

Copy the Contact Type for the currentRecord to the Contact Type text field

Copy the Contact Phone for the currentRecord to the Contact Phone field

Copy the Contact Web/Email for the currentRecord to the Contact Web/Email field

Copy the Contact Notes for the currentRecord to the Contact Notes field

### SaveRecord (CurrentRecord)

Copy the Contact Name from the Contact Name text field to the currentRecord of the Contact Info Array

Copy the Contact Type from the Contact Type text field to the currentRecord of the Contact Info Array

Copy the Contact Phone from the Contact Phone text field to the currentRecord of the Contact Info Array

Copy the Contact Web/Email from the Contact Web/Email text field to the currentRecord of the Contact Info Array

Copy the Contact Notes from the Contact Notes text field to the currentRecord of the Contact Info Array

### File data handling methods – read and write file

Next let’s turn to the file handling methods.

The pseudo code for these might be as follows:

### ReadFile

If there is an error, then go to the ErrorHandler

Open the data file ready for reading

Set counter = 0

While there is more data

Read in a line of data

Put the entries in their respective arrays

Set counter = counter + 1

End While

Set the NumberOfEntries = counter

Close the data file

ErrorHandler:

Write the message: “There was a problem in loading the data in the data file”

### WriteFile

Open the data file for writing data

For all entries in the arrays (*For count = 1 to*

*NumberOfRecords*)

Write a line of data into the data file

End For

Close the data file

### Navigation Button Code

The navigation buttons - First, Last, Previous and Last - require code within the ActionPerformed method, calling other methods if/as required.

The pseudo code for these buttons might be as follows:

### First

Set the CurrentRecord = the first record in the array

Call: DisplayRecord(CurrentRecord)

### Last

Set the CurrentRecord = the last record entered into in the array

Call: DisplayRecord(CurrentRecord)

### Previous

Set the CurrentRecord = the CurrentRecord - 1 in the array

Call: DisplayRecord(CurrentRecord)

That said, we need to check if we have gone past the beginning of the array... We therefore could adjust the pseudo code to test for this:

If CurrentRecord > first (*0 for Java*), then:

Set the CurrentRecord = the CurrentRecord - 1 in the array

Call: DisplayRecord(CurrentRecord)

End If

### Next

Set the CurrentRecord = the CurrentRecord + 1 in the array

Call: DisplayRecord(CurrentRecord)

Again we need to check if we have gone past the last record (NumberOfRecords) in the array... We therefore could adjust the pseudo code to test for this:

If CurrentRecord < NumberOfRecords, then:

Set the CurrentRecord = the CurrentRecord + 1 in the array

Call: DisplayRecord(CurrentRecord)

End If

Note: The last record (last data entry) may be well short of the full size of the array. For instance there may be 67 computers in the arrays, but the arrays might hold up to 500 or 1000 entries.

### Navigation Button Code

The remaining buttons – New, Save, Exit, Delete and Find - require code within the actionPerformed method.

The pseudo code for these might be as follows:

### New

If the arrays are not full

Increment NumberOfEntries

Set the current record to the last record

Clear the array entries for the current record

Display the current record

End if

### Save

Save the current record

### Exit

Write the array data to the data file

Quit the program

### Delete

For all records from the current to the last record

Copy the next Contact Name to the current record

Copy the next Contact Phone to the current record

Copy the next Contact Type to the current record

Copy the next Contact Web/Email to the current record

Copy the next Contact Notes to the current record

End For

Decrement the NumberOfEntries

If the current entry is now past the new last entry

Set the current record back one

End if

Display the current entry

### Find

Set found = false

Set counter = o

While you have more records to check and the entry has  
not been found

If the current entry is equals to the search string

Set found = true

End If

Increment the counter

End While

If Found

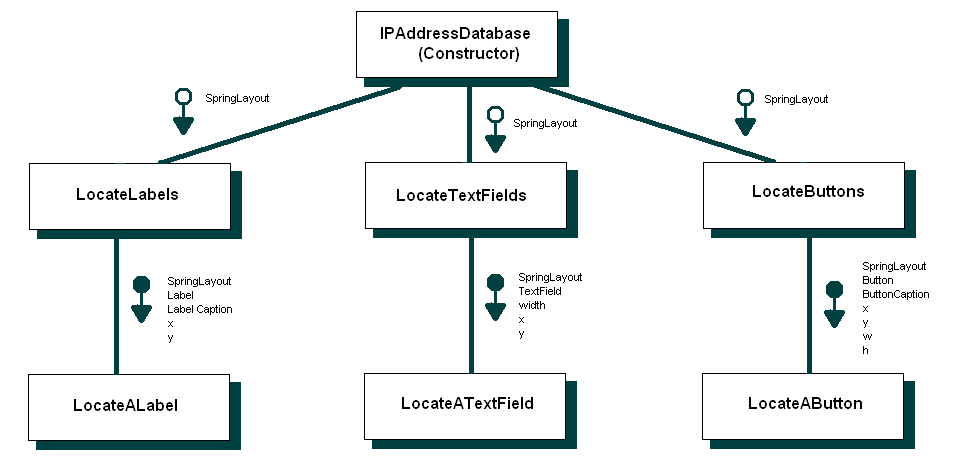
Set the current record to the one that was found

Display the current entry

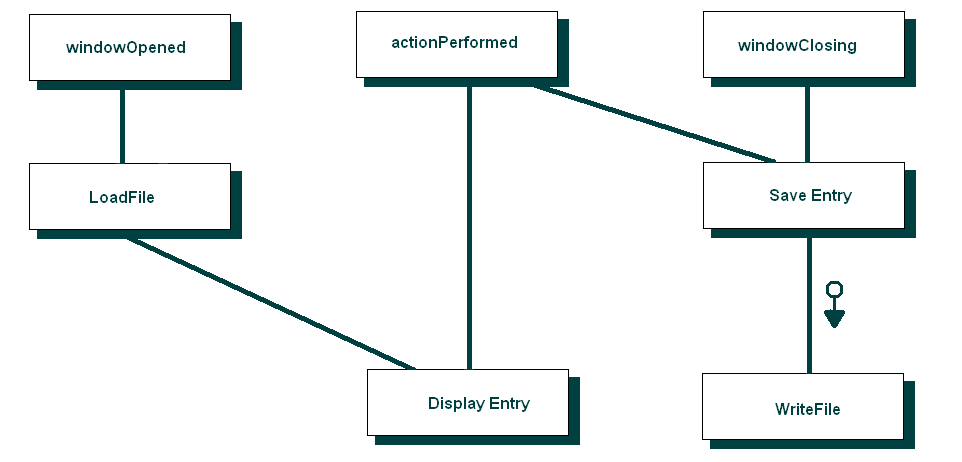
End if

## Structured Chart(s)

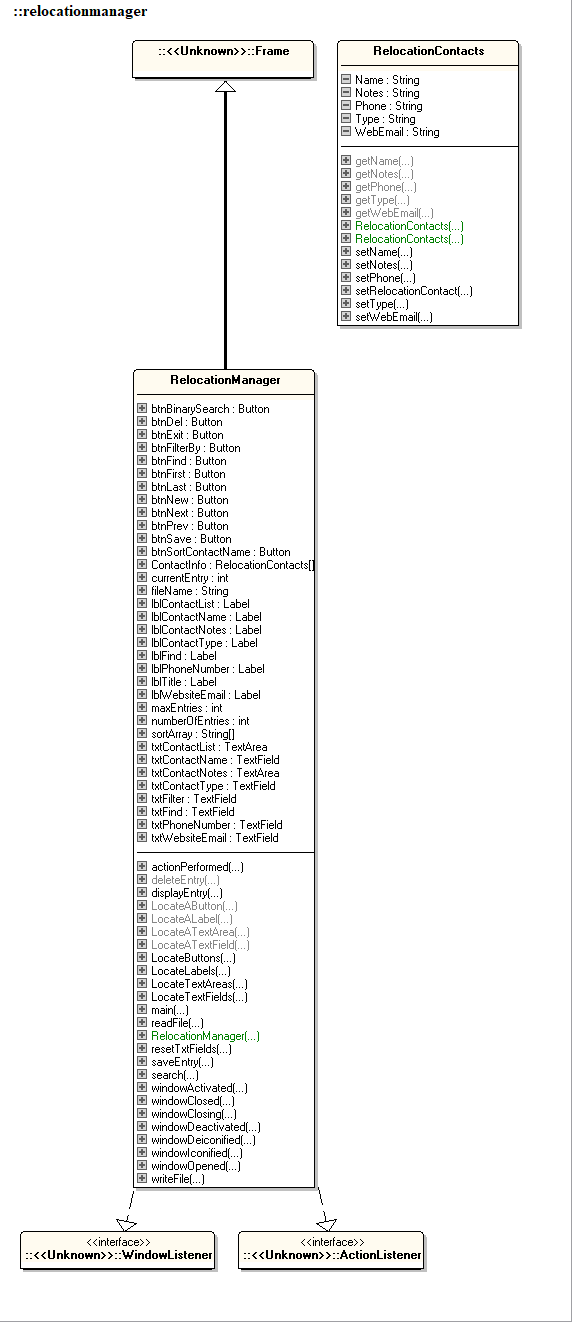
### Managing the GUI elements



### Managing the user Actions



## Class diagram

(Using **ESSModel.EXE *(supplied in Third Party Material)***):

## Testing

### Test Plan

The approach to testing that will be used is as follows:

1. Review the project requirements
2. Prepare a solution design
3. Develop the required application.
4. **Unit testing** – test each component (method / class) as it is being developed
5. **Integrated Testing** - Prepare a set of data (test cases) to check how the application will behave when appropriate data is read in and entered via the screen.
6. Prepare several sets of data (test cases) to check how the application will behave when data that will generate exceptions is read in from the data file.
7. Enter several sets of data (test cases) to check how the application will behave when data that will generate exceptions is entered via the screen.
8. **System Testing** – Deploy the application to a computer system / network / web host site similar to that of the client’s and retest selected test cases.
9. **Acceptance Testing** - Deploy the application to the client’s computer system / network / web host site and look to the client to retest using sample live data.
10. Archive the test plan, test cases and results to an appropriate backup location.

### Test Cases:

Sample data file 1 – Correct operation:

Data Input for correction operation:

Electricity Company;Utility Company;1122;www.electricitycompany�;Need to ring the Electricity Company to update;

The Phone Company;Utility Company;1234;www.the\_phone\_company...;Change of contact details can be updated via The Phone Company web portal.;

Our Internet Provider;Internet Company;3232;www.ourinternetprovider�;Change of contact details can be updated via The Internet Provider's web portal.;

Bob's Burgers;Burger Place;1800 Burgers;burger@burger.com;burgers ;

Stevens chip shop;Chip Shop;1800 Chips;stevens@chips.com;This is a chip shop;

Alfreds Alpacca Appliances;Alpacca Appliances;1800 Alpaccas;Alpaccas@appliances.com;This is a florist;

Frank;Frank;04Frank;frank.frank;He is frank;

Sample data file 2 – Testing for exceptions:

Nonoptimal working data (additional semi-colons (separators) throughout data):

Electricity Com; pany;Utility Company;1122;www.electricitycompany�;Need to ring the Electricity Company to update;

The Phone Company;Utility Company;1234;www.the\_phone\_company...;Change of contact ; details can be updated via The Phone Company web portal.;

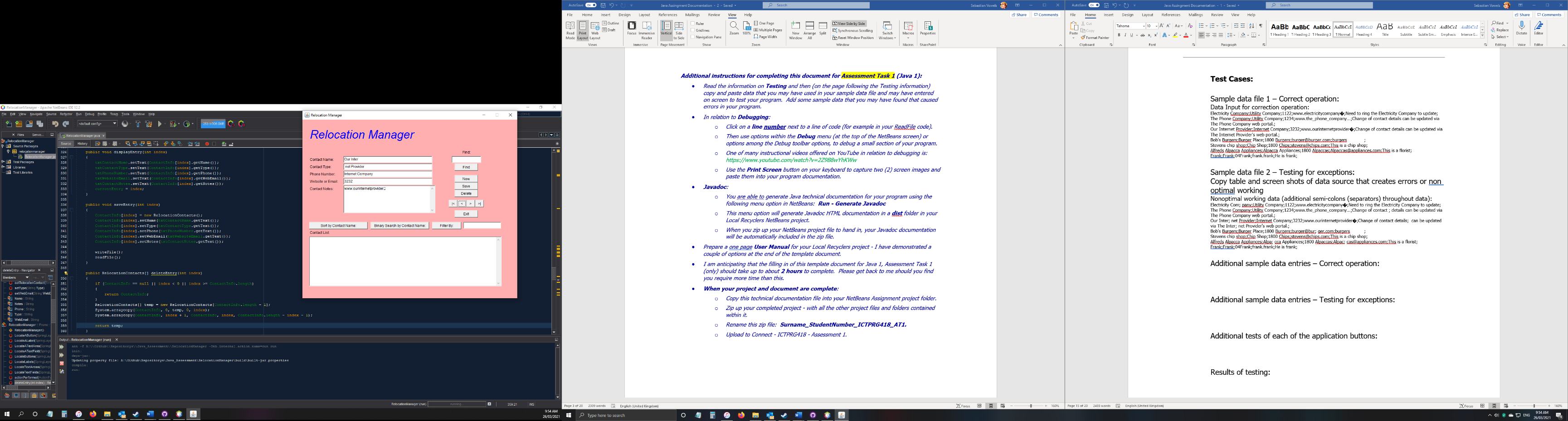
Our Inter; net Provider;Internet Company;3232;www.ourinternetprovider�;Change of contact details; can be updated via The Inter; net Provider's web portal.;

Bob's Burgers;Burger Place;1800 Burgers;burger@bur; ger.com;burgers ;

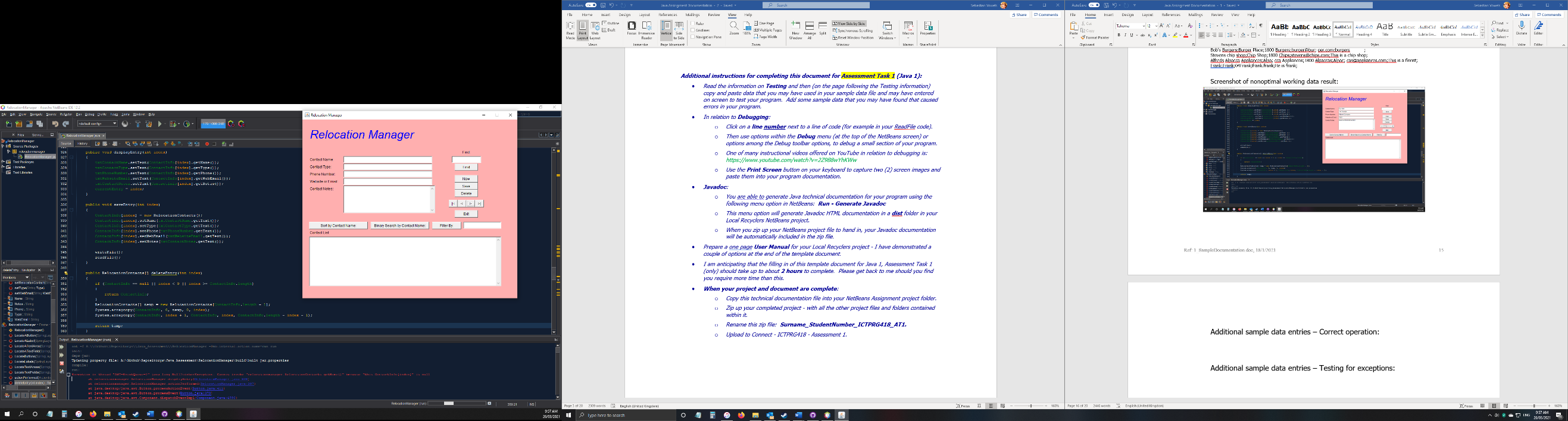
Stevens chip shop;Chip Shop;1800 Chips;stevens@chips.com;This is a chip shop;

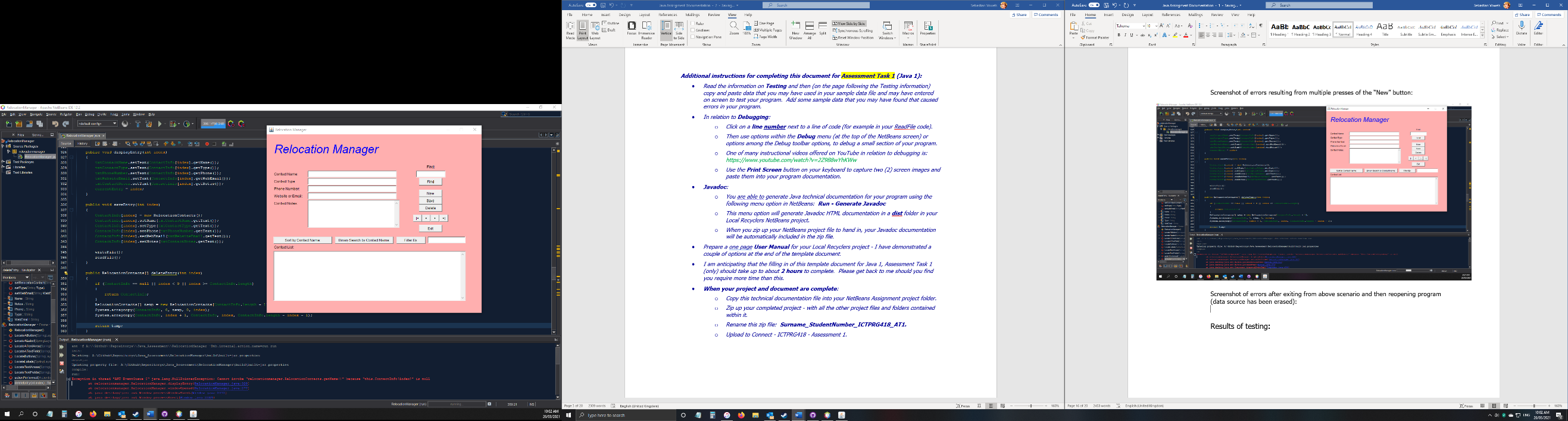
Alfreds Alpacca Appliances;Alpa; cca Appliances;1800 Alpaccas;Alpac; cas@appliances.com;This is a florist;

Frank;Frank;04Frank;frank.frank;He is frank;

Screenshot of nonoptimal working data result:

Screenshot of errors resulting from multiple presses of the “New” button:



Screenshot of errors after exiting from above scenario and then reopening program (data source has been erased):

Testing Blank Entries within data source

Data source:

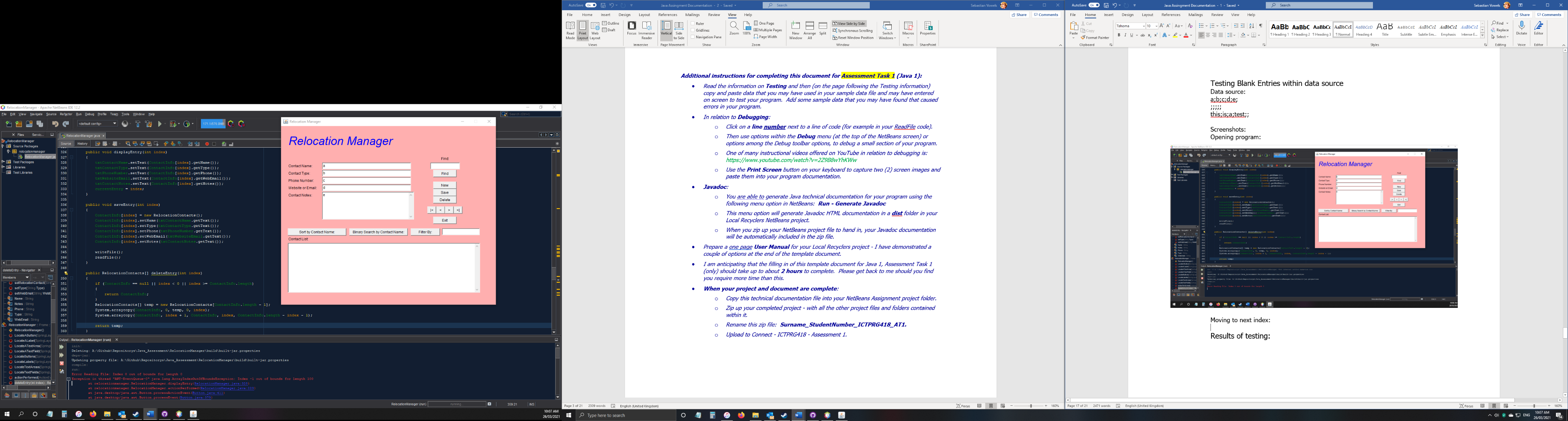
a;b;c;d;e;

;;;;;

this;is;a;test;;

Screenshots:

Opening program:

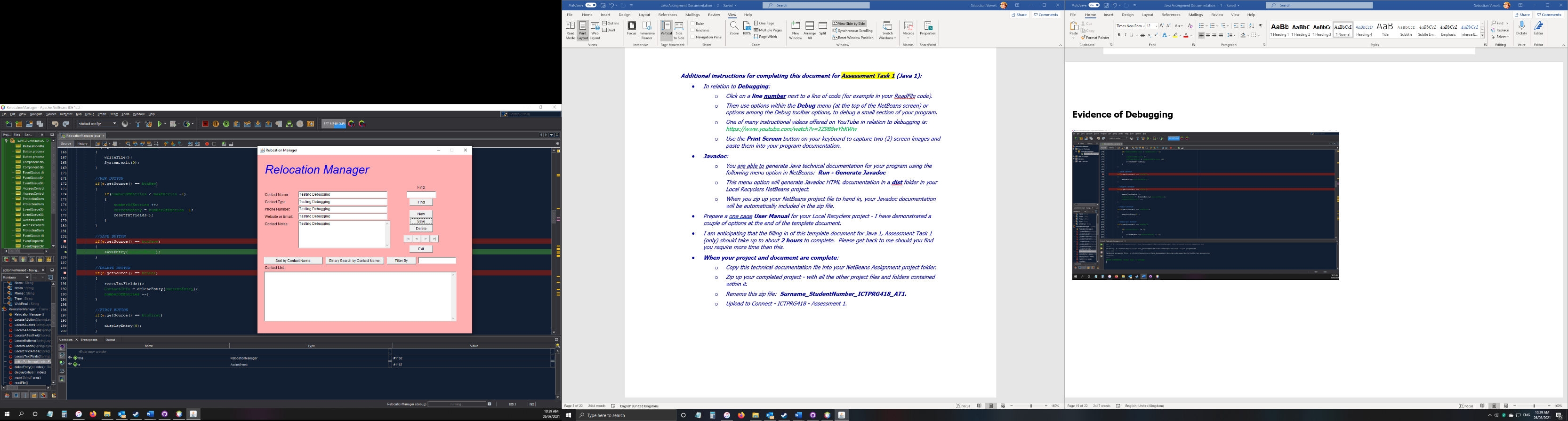
Moving to next index:

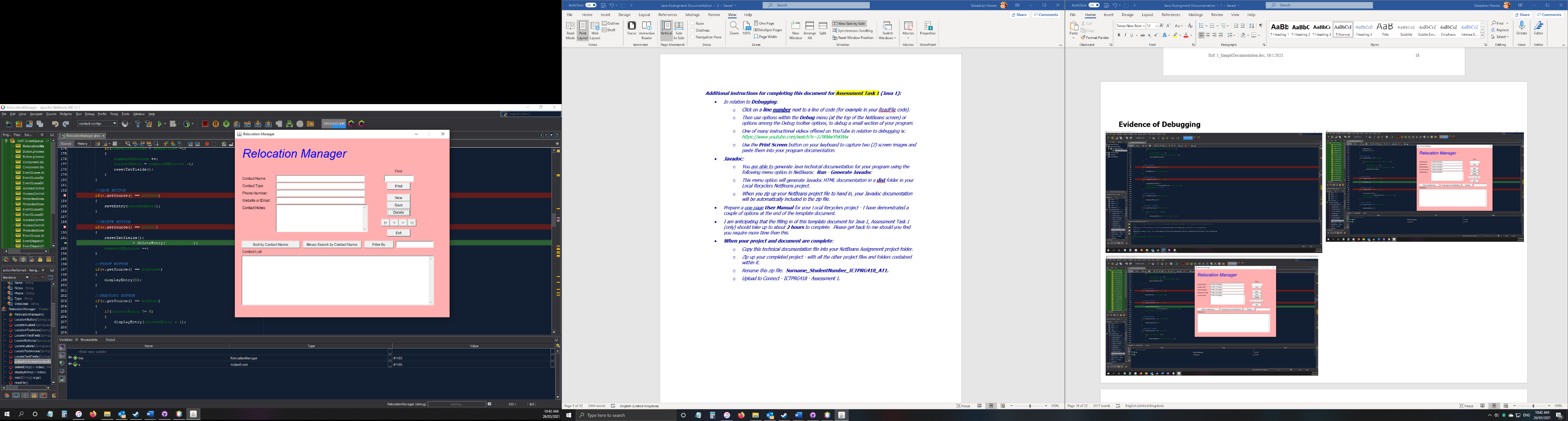
Results of testing:

The current build is stable and functional. The following errors have been found and documented:

* + The “New” button has been pressed multiple times
    - This increases the numberOfEntries variable by 1 each time
    - It also assigns the currentEntry variable to be one less than the numberOfEntries
    - This works for a single press (as designed) however if pressed again errors will occur as there is no data present in the array at the currentEntry index, causing an index out of bounds error.
  + Empty data set in data source
    - Entries will display if there is valid data.
    - When data is null, data will not display.
  + Dividers used inside data set
    - If a semi colon (which is the separator for the program) is used inside a contact info field, the program will be unable to recognise this and will divide at that point as instructed.
    - This causes separation of data when it isn’t intended.

## Evidence of Debugging





## System generated program documentation

Prepare automated program documentation using a facility provided within your IDE, or using a separate facility such as Javadoc or ESS Model.

Submit this automated documentation in a separate zip file.

* + This file has been included in the program folder.

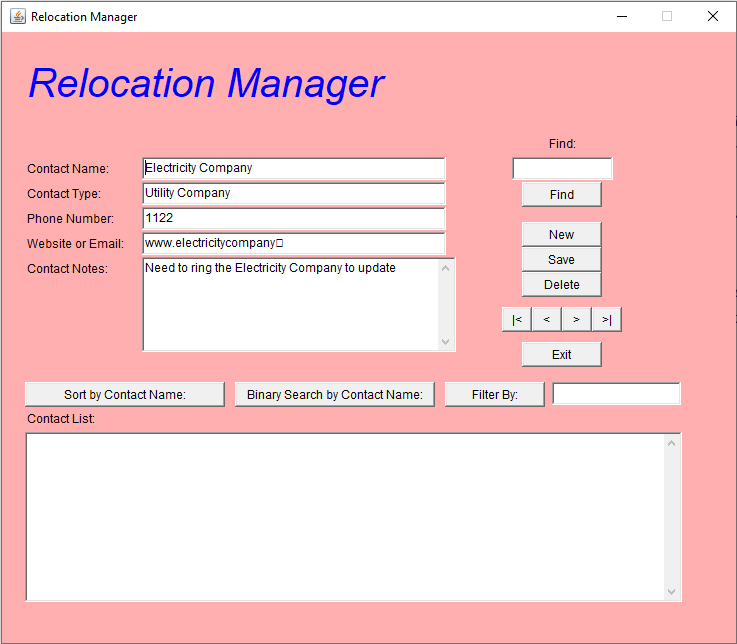
# User Manual for: Relocation Manager

Prepared for: *Sample Business/Organisation*

Type a search string in the Search text box sand click the **Find** button to locate an existing person’s entry.

Author: Sebastian Vowels

Date Prepared: *26/03/2021*



Click the **New** **Entry** button to add a new person’s entry.

Press the **Save** Button to save your entry.

Press the **Delete** Button to delete the current entry.

Use the **Navigation** buttons to go backwards and forwards

Always ensure you press the **Exit** button to save your work!

|  |  |
| --- | --- |
| How Do I?  **Add and Save a New Entry**   1. Click on the **New** button 2. Type in your new entry 3. Click on the **Save** button to save the entry | How Do I?  **Find and Update an Existing Entry**   1. Type the Contact Name in the text field 2. Click on the **Find** button 3. Check the entry presented is correct 4. Edit the entry as required 5. Click on the **Save** button to save the entry |