**H48W 35 Computing: Software Development  
Graded Unit 2**

**Project Stage: Development**

**Test Log**

**Student name: Daria Vekic**

**Student number: 586661**

**Date: March – May 2023**

Table of Contents

[Test Log 1](#_Toc134989555)

[Failed Tests 12](#_Toc134989556)

[Test 13 12](#_Toc134989557)

[Test 62 16](#_Toc134989558)

[Test 72 18](#_Toc134989559)

[Appendix 20](#_Toc134989560)

[Appendix 1 20](#_Toc134989561)

[Appendix 2 21](#_Toc134989562)

[Appendix 3 22](#_Toc134989563)

[Appendix 4 23](#_Toc134989564)

[Appendix 5 23](#_Toc134989565)

[Appendix 6 24](#_Toc134989566)

[Appendix 7 24](#_Toc134989567)

[Appendix 8 25](#_Toc134989568)

[Appendix 9 26](#_Toc134989569)

[Appendix 10 27](#_Toc134989570)

[Appendix 11 28](#_Toc134989571)

[Appendix 12 29](#_Toc134989572)

[Appendix 13 30](#_Toc134989573)

[Appendix 14 31](#_Toc134989574)

[Appendix 15 32](#_Toc134989575)

[Appendix 16 33](#_Toc134989576)

[Appendix 17 34](#_Toc134989577)

[Appendix 18 35](#_Toc134989578)

[Appendix 19 36](#_Toc134989579)

[Appendix 20 37](#_Toc134989580)

[Appendix 21 38](#_Toc134989581)

[Appendix 22 39](#_Toc134989582)

[Appendix 23 40](#_Toc134989583)

[Appendix 24 41](#_Toc134989584)

[Appendix 25 42](#_Toc134989585)

[Appendix 26 43](#_Toc134989586)

[Appendix 27 44](#_Toc134989587)

[Appendix 28 45](#_Toc134989588)

[Appendix 29 46](#_Toc134989589)

[Appendix 30 47](#_Toc134989590)

[Appendix 31 48](#_Toc134989591)

[Appendix 32 49](#_Toc134989592)

**Table of Figures**

[Figure 1 Solution Planning - Home interface design 12](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989509)

[Figure 2 JLabel used to hold ImageIcon 13](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989510)

[Figure 3 Get the ImageIcon using given file path 14](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989511)

[Figure 4 Class level variables 15](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989512)

[Figure 5 Setting Labels to hold the image 15](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989513)

[Figure 6 Retrieving image success 15](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989514)

[Figure 7 Possible correction of postcode formatting 16](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989515)

[Figure 8 Updated Person constructor to store properly formatted postcode 17](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989516)

[Figure 9 Test rerun success 17](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989517)

[Figure 10 Use of counter variable to control instantiation of Case 18](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989518)

[Figure 11 Code correction so counter is reset 19](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989519)

[Figure 12 C: drive before running system 20](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989520)

[Figure 13 C: drive after running system 20](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989521)

[Figure 14 Contents of directory 21](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989522)

[Figure 15 Text file containing hashed passwords 21](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989523)

[Figure 16 Username not found 22](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989524)

[Figure 17 Hash value before password reset 23](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989525)

[Figure 18 Hash value after reset 24](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989526)

[Figure 19 NullPointerException 24](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989527)

[Figure 20 Success message confirming Case added 25](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989528)

[Figure 21 Case details show the newly added Case 26](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989529)

[Figure 22 Blank fields are not allowed 27](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989530)

[Figure 23 Case created after trying to enter blank fields 28](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989531)

[Figure 24 Invalid Case Title error message 29](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989532)

[Figure 25 Incorrect separator used in DOB 30](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989533)

[Figure 26 Upper bounds of February in a leap year not accepted 31](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989534)

[Figure 27 Normal data for February if it is a leap year 32](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989535)

[Figure 28 Upper bounds of February if it is not a leap year 33](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989536)

[Figure 29 YYYY value must start with 1 or 2 34](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989537)

[Figure 30 Address line 1 only outlined red 35](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989538)

[Figure 31 Address line 2 only outlined red 36](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989539)

[Figure 32 Postcode using illegal separator 37](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989540)

[Figure 33 Non-UK postcode (Dublin) not acceptable 38](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989541)

[Figure 34 Belfast postcode accepted 39](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989542)

[Figure 35 Case created using Kirkwall postcode 40](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989543)

[Figure 36 Case created using Shetland postcode 41](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989544)

[Figure 37 Case created using Jersey postcode 42](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989545)

[Figure 38 Postcode case insensitive 43](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989546)

[Figure 39 Output showing postcode not formatted 43](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989547)

[Figure 40 Case created with postcode with no space 44](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989548)

[Figure 41 Output displays postcode in proper format 44](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989549)

[Figure 42 Phone number length too short 45](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989550)

[Figure 43 Phone number including a space accepted 46](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989551)

[Figure 44 Search results displayed in new window 47](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989552)

[Figure 45 Search results for invalid Case Reference Number 48](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989553)

[Figure 46 Failed test - Case should be created after re-entering normal data 49](file:///E:\Graded%20Unit\2.%20Development%20Phase\Testing\Test%20Log.docx#_Toc134989554)

# Test Log

Name: Daria Vekic  
Date: April / May 2023  
Software: CSC Room Booking System  
Version Number: 1

| **Test No.** | **Actual Result** | **Pass/Fail** | **Comments** |
| --- | --- | --- | --- |
|  | Correct directory created and contains “Users.txt”. | Pass | See Appendix 1. |
|  | “Users.txt” contains correct usernames and a hashed value of their passwords. | Pass | See Appendix 2. |
|  | Log in successful and Home GUI appears after pressing “Sign In”. | Pass |  |
|  | Error message is displayed indicating the username is not found. | Pass | See Appendix 3. |
|  | Error message is displayed indicating the password does not match. | Pass | See Appendix 4. |
|  | Pressing link results in a new window appearing ready to accept input. | Pass |  |
|  | Successfully gain access to Home GUI after resetting password and logging in with that new password. | Pass | See Appendix 5 for the password hash value before reset. See Appendix 6 for password hash value after reset. |
|  | Hash value only is displayed in text file. | Pass | As above. See Appendix 5 and Appendix 6. |
|  | Error message is displayed indicating the password doesn’t meet criteria. | Pass |  |
|  | Error message is displayed indicating the password doesn’t meet criteria. | Pass |  |
|  | Error message is displayed indicating the password doesn’t meet criteria. | Pass |  |
|  | Error message is displayed indicating the password doesn’t meet criteria. | Pass |  |
|  | **NullPointerException. System terminates as exception is not handled.** | **Fail** | **See Appendix 7 for error.** |
|  | New window is displayed ready for input when “Add New Case” button is pressed. | Pass |  |
|  | Home window remains visible when “Add New Case” window is closed. | Pass |  |
|  | Value of “Enquiry Type” is changed automatically according to value of “Case Type”. | Pass |  |
|  | Message confirming new Case has been added is displayed and the Case can be viewed in the console. | Pass | See Appendix 8 for success message and Appendix 9 for output. |
|  | Message is displayed to indicate blank fields are not allowed and all blank fields are outlined in red. | Pass | See Appendix 10. |
|  | Case is successfully created and added to system. | Pass | See Appendix 11. |
|  | Case is **not** created and error message is displayed indicating the Case Title is invalid. | Pass | See Appendix 12. |
|  | Case is **not** created and error message is displayed indicating the Case Title is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the Case Title is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the Case Title is invalid. | Pass |  |
|  | Case **is** created and added to system. Success message displayed indicating this. | Pass |  |
|  | Case **is** created and added to system. Success message displayed indicating this. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the Case Title is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the Case Title is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the Client first name is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the Client first name is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the DOB format is invalid. | Pass | See Appendix 13. |
|  | Case is **not** created and error message is displayed indicating the DOB format is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the DOB format is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the DOB format is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the DOB format is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the DOB format is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the DOB format is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the DOB format is invalid. | Pass |  |
|  | Case is **not** created and error message is displayed indicating the DOB must contain valid value. | Pass | See Appendix 14. |
|  | Case is successfully created. | Pass | See Appendix 15. |
|  | Error message is displayed informing user that DOB is incorrect. | Pass | See Appendix 16. |
|  | Case is successfully created. | Pass |  |
|  | Error message is displayed informing user that DOB is incorrect. | Pass |  |
|  | Error message is displayed informing user that DOB is incorrect. | Pass |  |
|  | Error message is displayed informing user that DOB is incorrect. | Pass |  |
|  | Error message is displayed informing user that DOB is incorrect. | Pass |  |
|  | Error message is displayed informing user that DOB is incorrect. | Pass |  |
|  | Error message is displayed informing user that DOB is incorrect. | Pass |  |
|  | Error message is displayed informing user that DOB is incorrect. | Pass |  |
|  | Error message is displayed informing user that DOB is incorrect. | Pass |  |
|  | Error message is displayed informing user that DOB is incorrect. | Pass |  |
|  | Error message is displayed informing user that DOB is incorrect. | Pass |  |
|  | Error message is displayed informing user that DOB is incorrect. | Pass | See Appendix 17. |
|  | Error message is displayed informing user that DOB is incorrect. | Pass |  |
|  | Error message is displayed informing user that field cannot be blank. Address line 1 only is outlined red. | Pass | See Appendix 18. |
|  | Error message is displayed informing user that field cannot be blank. Address line 2 only is outlined red. | Pass | See Appendix 19. |
|  | Error message is displayed informing user that postcode is invalid. Postcode field only is outlined red. | Pass | See Appendix 20. |
|  | Error message is displayed informing user that postcode is invalid. Postcode field only is outlined red. | Pass | See Appendix 21. |
|  | Case successfully created using Belfast postcode. | Pass | See Appendix 22. |
|  | Case successfully created using Kirkwall postcode. | Pass | See Appendix 23. |
|  | Case successfully created using Shetland postcode. | Pass | See Appendix 24. |
|  | Case successfully created using Jersey postcode. | Pass | See Appendix 25. |
|  | Case successfully created with case insensitive postcode. However, output does not display formatted postcode. | Fail | See Appendix 26. |
| **Note: Test number 62 rectified so output corrected from test 63 onwards. Test 62 is repeated at end of Test Log.** | | | |
|  | Case successfully created with postcode that does not include a space. Output displays properly formatted postcode. | Pass | See Appendix 27. |
|  | Error message is displayed informing user that phone number is invalid. Phone number field only is outlined in red. | Pass | See Appendix 28. |
|  | Error message is displayed informing user that phone number is invalid. Phone number field only is outlined in red. | Pass |  |
|  | Error message is displayed informing user that phone number is invalid. Phone number field only is outlined in red. | Pass |  |
|  | Error message is displayed informing user that phone number is invalid. Phone number field only is outlined in red. | Pass |  |
|  | Case successfully created. | Pass |  |
|  | The value displayed in “Case Type” dropdown list automatically changes to match selection in “Enquiry Type” dropdown list. | Pass |  |
|  | The search results for Case “003-PIN” are presented in a new window. | Pass | See Appendix 30. |
|  | Error message is displayed indicating the Case is not found. | Pass | See Appendix 31. |
|  | On re-entering normal data, the system does not allow for the Case to be created. | Fail | The system does not create a Case with data set 12 as expected. However, after re-entering data set 12 with a valid Case Title, the system still does not create the Case despite normal data being provided.  See Appendix 32. |

# Failed Tests

## Test 13

The purpose of this test was to verify that the system successfully retrieves the media resources required to ensure the graphical user interface reflects the design produced in Solution Planning. The SP report provides the following mockup to show what the Home window should look like:

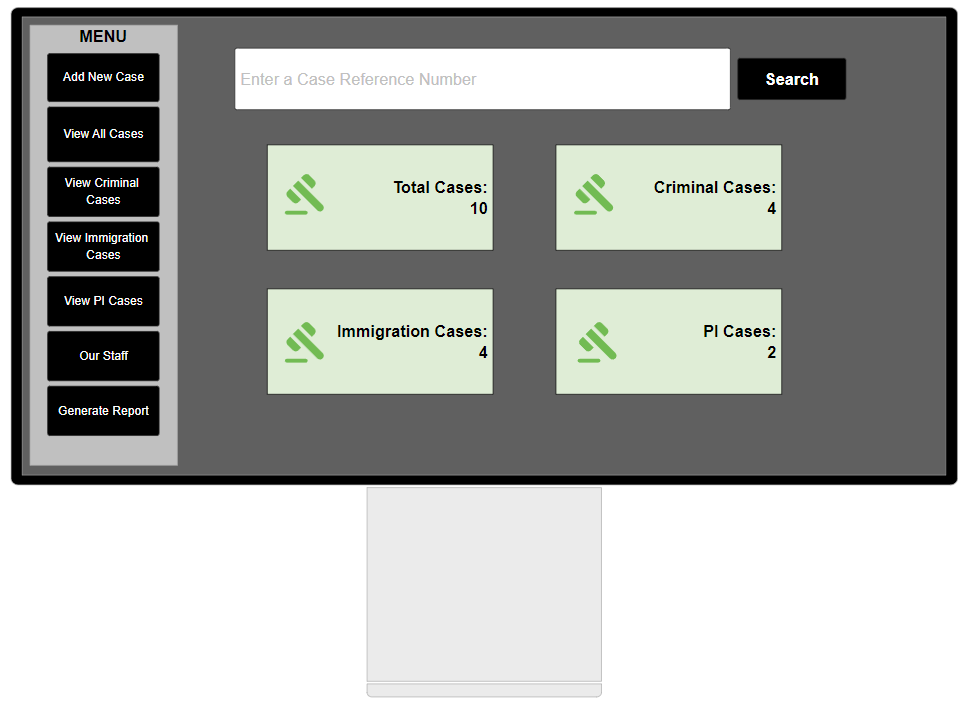


Figure Solution Planning - Home interface design

In class HomeInterface, the interface makes use of a JLabel and ImageIcon to display the required image. Each of the four labels are coded in this way; the example below shows the code for the “Total Cases” label as the code snippet becomes too unreadable to show all four examples:

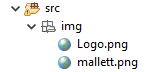
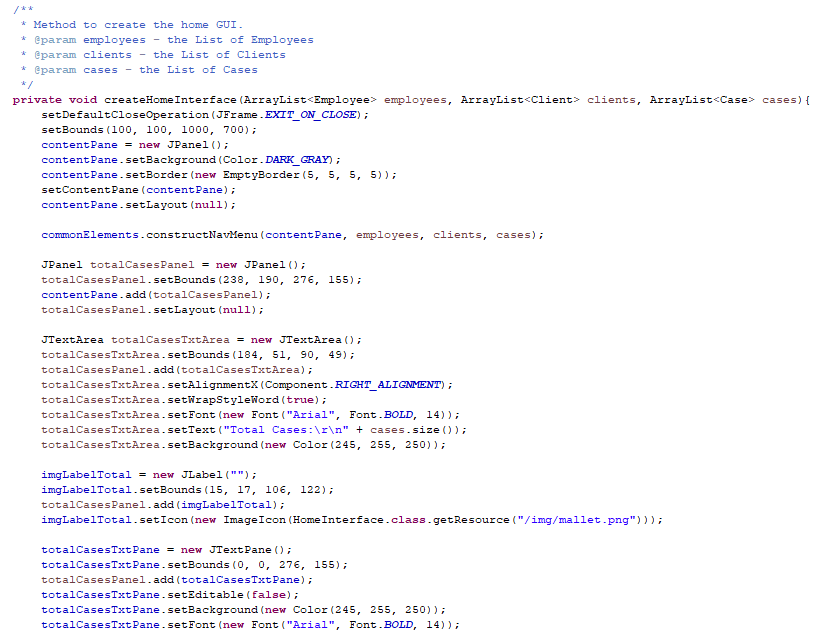


Figure JLabel used to hold ImageIcon

It is clear from Appendix 7 that the program does not successfully retrieve the images and results in a NullPointerException. It is not ideal for the user if the system terminates simply because an image cannot be retrieved. This failed test highlights the need for exception handling. The code snippet above shows the program uses one statement to create the label to contain the image, and one statement to include the image on that label.

However, a better way to implement this would be to make use of the method createImageIcon(String path, String description) offered in the tutorial “How to Use Icons”[[1]](#footnote-1). This method locates the given file (using path), and will return the ImageIcon for the given file; if it cannot locate the given file, it will return null:

/\*\* Returns an ImageIcon, or null if the path was invalid. \*/

protected ImageIcon createImageIcon(String path,

String description) {

java.net.URL imgURL = getClass().getResource(path);

if (imgURL != null) {

return new ImageIcon(imgURL, description);

} else {

System.err.println("Couldn't find file: " + path);

return null;

}

}

Oracle.com, 2019

So to implement this in the program, the method that constructs the Home interface can contain one line of code to return the image, rather than hard‑coding the file path multiple times (the same image is used four times):

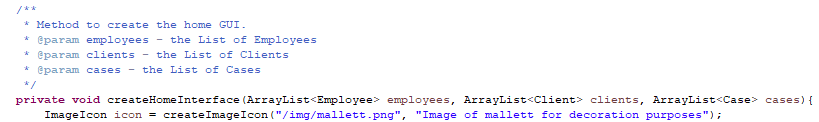


Figure Get the ImageIcon using given file path

Then, the four JLabels that contain the images can be moved to class level variables and can then simply be changed to all use the same image:

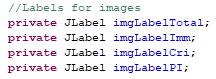


Figure Class level variables



Figure Setting Labels to hold the image

So now, when the system is run, the images are either successfully displayed or an error message is output to the console – either way, it does not prevent the system from terminating:

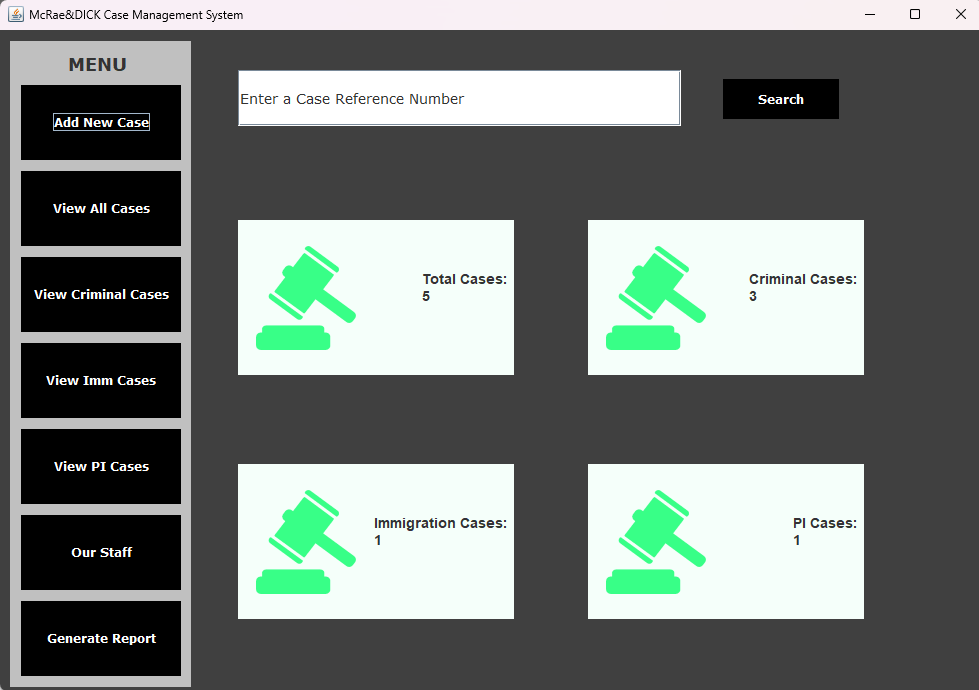


Figure Retrieving image success

## Test 62

From Appendix 26 it is clear that when the Case details are output to the console, the postcode is not being formatted correctly. The postcode is valid (it’s my postcode, so it’s definitely valid!), but it is being stored in the format that the user enters it in, which does not ensure any consistency in data storage. It would be possible to format this data in the method that is called when the “View All Cases” button is pressed – the implementation of this correction is shown below:

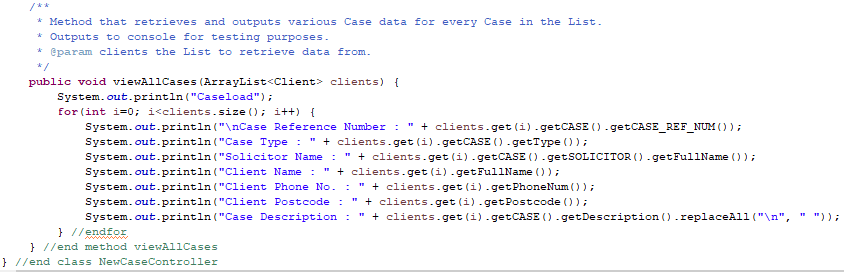
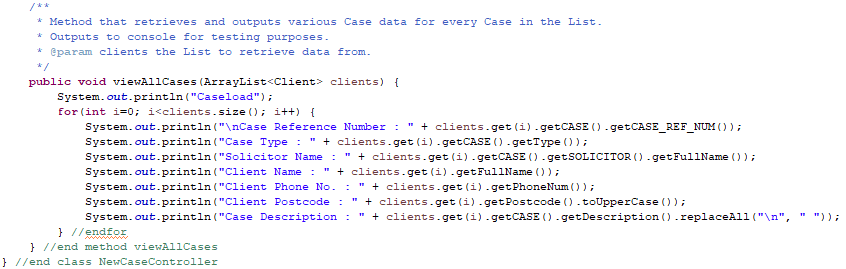


Figure Possible correction of postcode formatting

However, the change above only formats the output. This does not ensure that a properly formatted postcode is being **stored**; only displayed. So, it would be a better choice to format this data when the object is being constructed. To do this, the constructor of the Person class can be amended as follows:

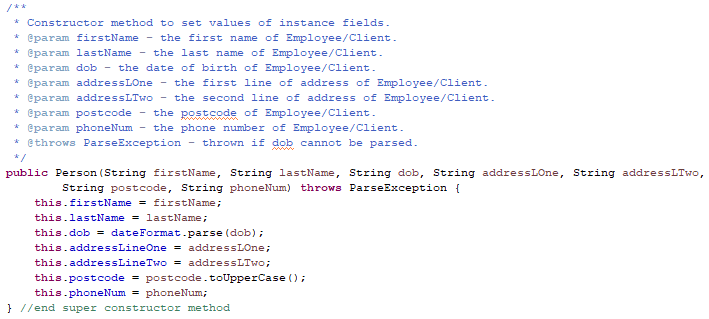
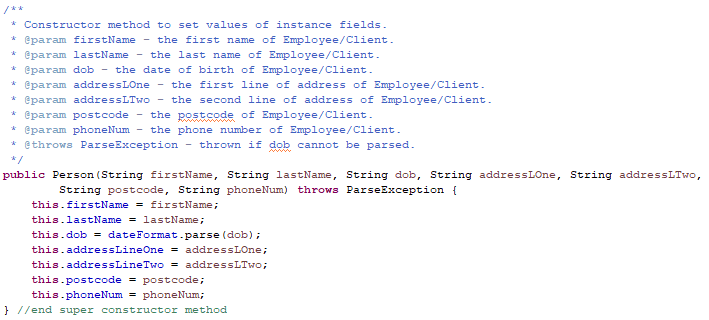


Figure Updated Person constructor to store properly formatted postcode

Now, when Case details are output, the postcode is displayed in the proper format even when lower case letters are used by the user:

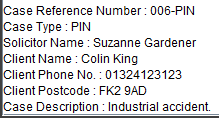
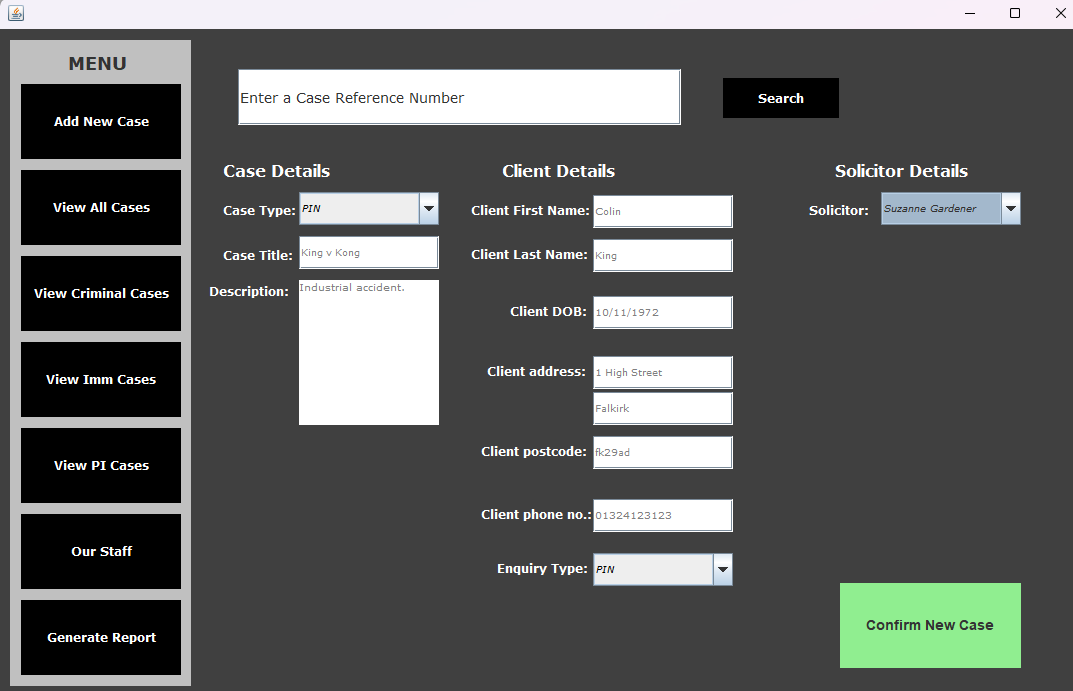


Figure Test rerun success

## Test 72

This test aimed to verify that the use of the invalidFields variable successfully controls the flow of the process of creating a new Case. This is a class level variable contained in class AddCaseInterface and is intended to be used to prevent a Case from being constructed if the user has entered invalid data. It is initialised to 0 and is used in the method shown below:



If invalidFields holds value of 0, the Case is ready to be constructed.

Figure Use of counter variable to control instantiation of Case

From the code snippet above, it is clear that this counter variable is not reset anywhere. So, if any of the data input is invalid, the invalidFields variable will hold a value that is greater than 0 and the user will see the error message when they try to confirm their input. After the user confirms their input, however, invalidFields still holds that value that is greater than 0. It needs to be reset to 0 for the user to then reattempt input using normal data:

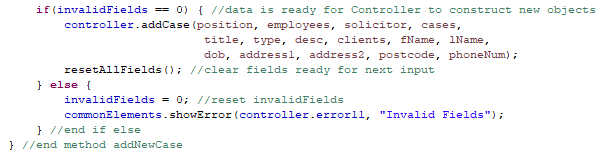


Figure Code correction so counter is reset

Now, if data set 12 is entered, the relevant error message is still being displayed, but when data set 12 is re‑entered using normal Case Title data, the Case is successfully created. This correction now changes the status of this test to a “Pass”.

# Appendix

## Appendix 1

Figure C: drive before running system

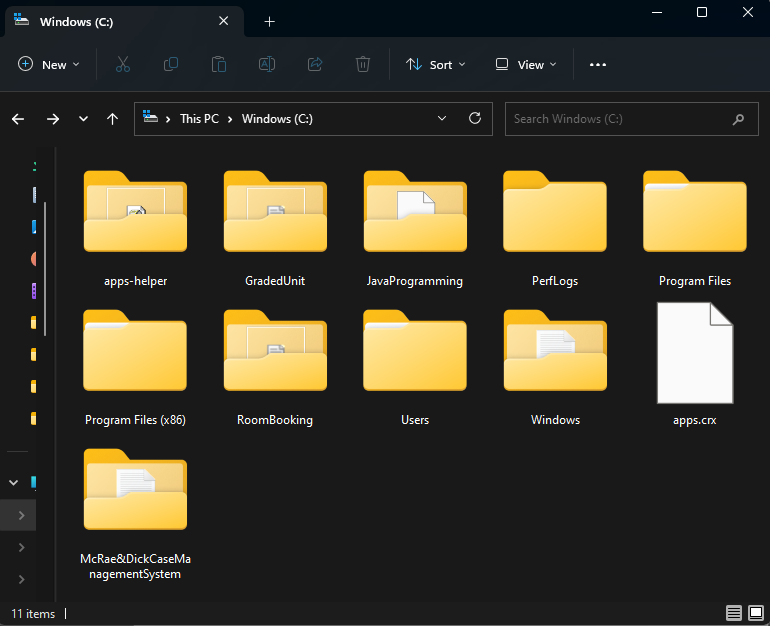
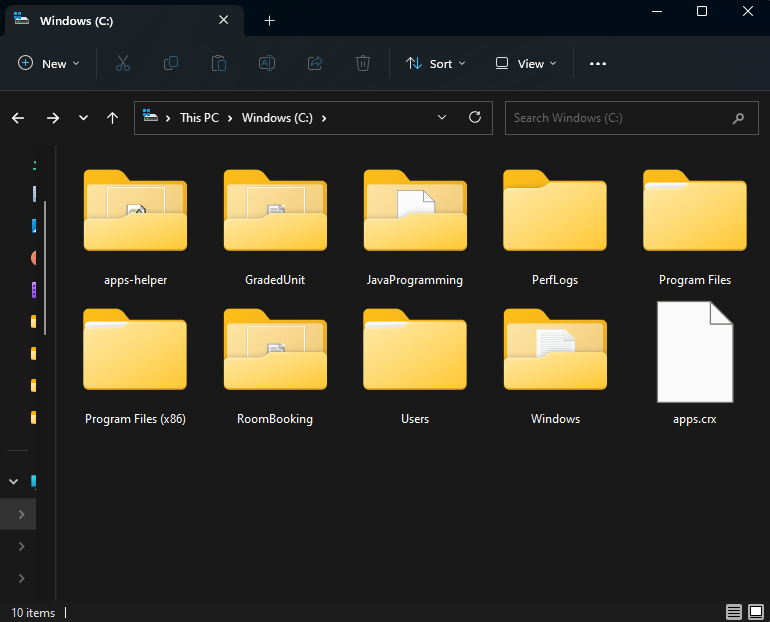


Figure C: drive after running system



Figure Contents of directory

## Appendix 2

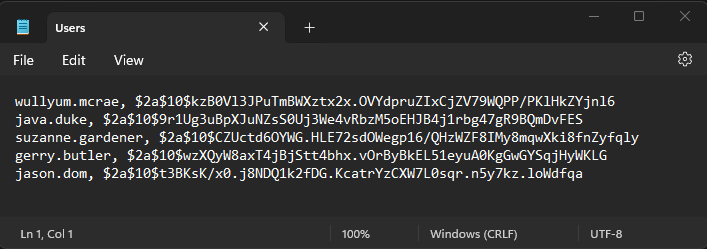


Figure Text file containing hashed passwords

## Appendix 3

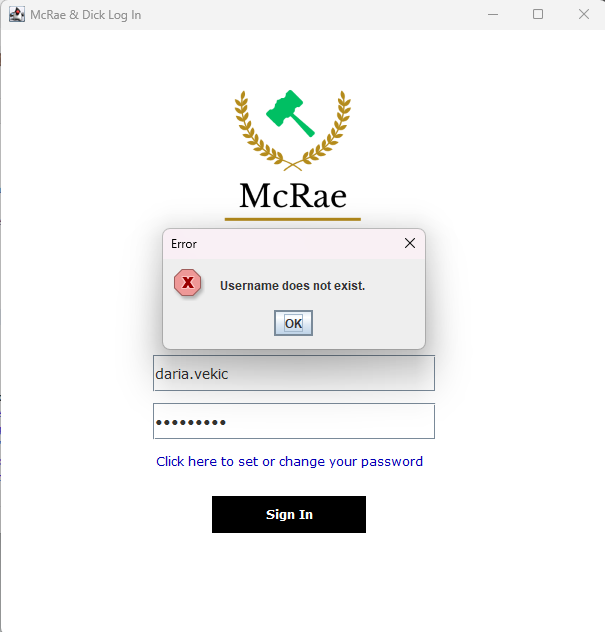


Figure Username not found

## Appendix 4

## Appendix 5

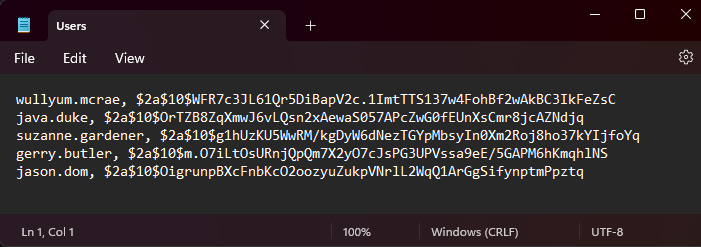


Figure Hash value before password reset

Original hash

## Appendix 6

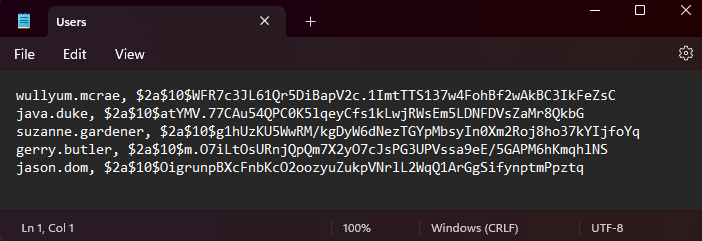


Figure Hash value after reset

New hash

## Appendix 7

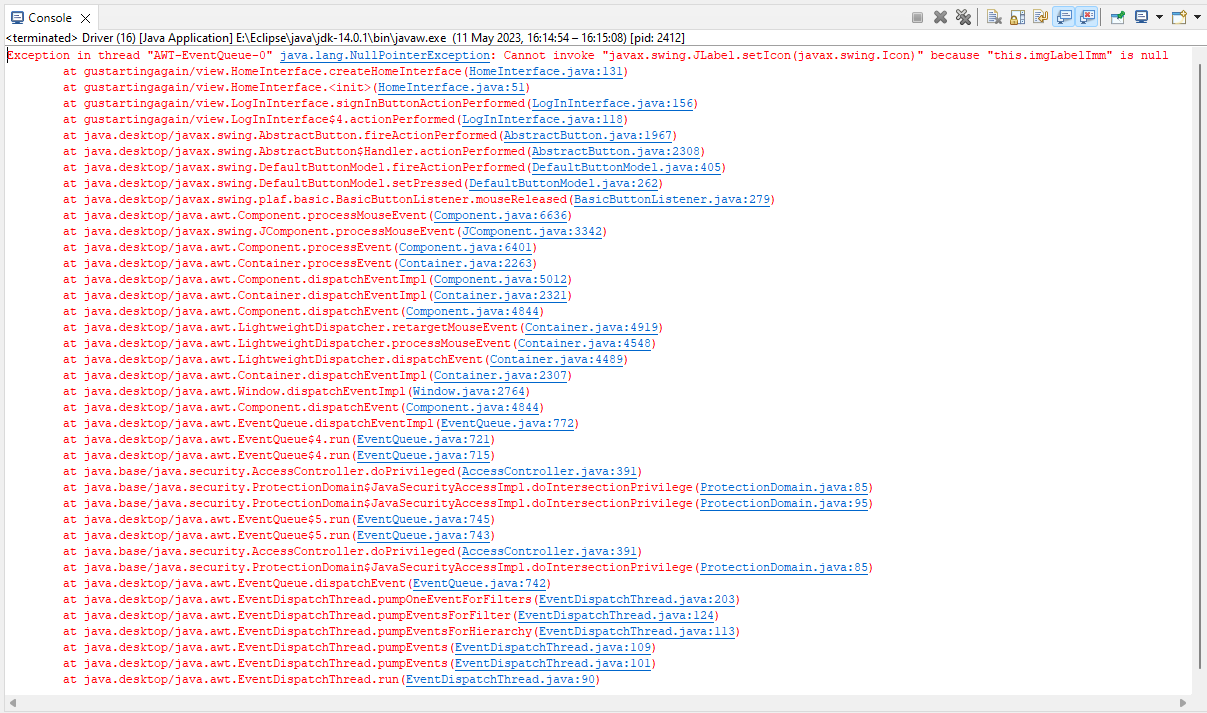


Figure NullPointerException

## Appendix 8

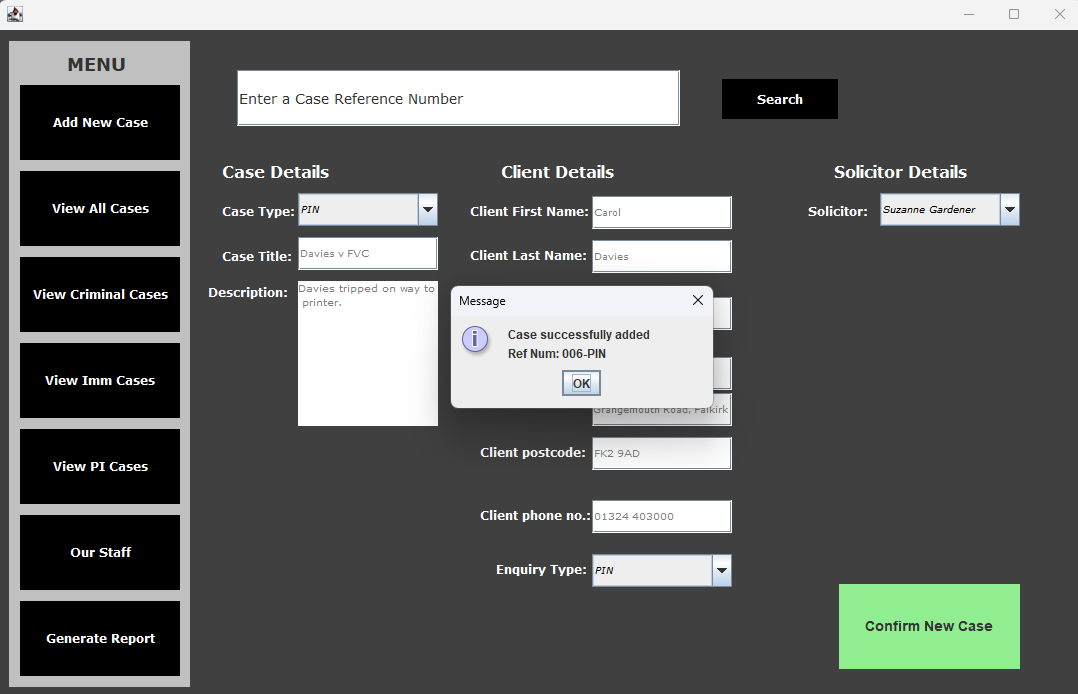


Figure Success message confirming Case added

## Appendix 9

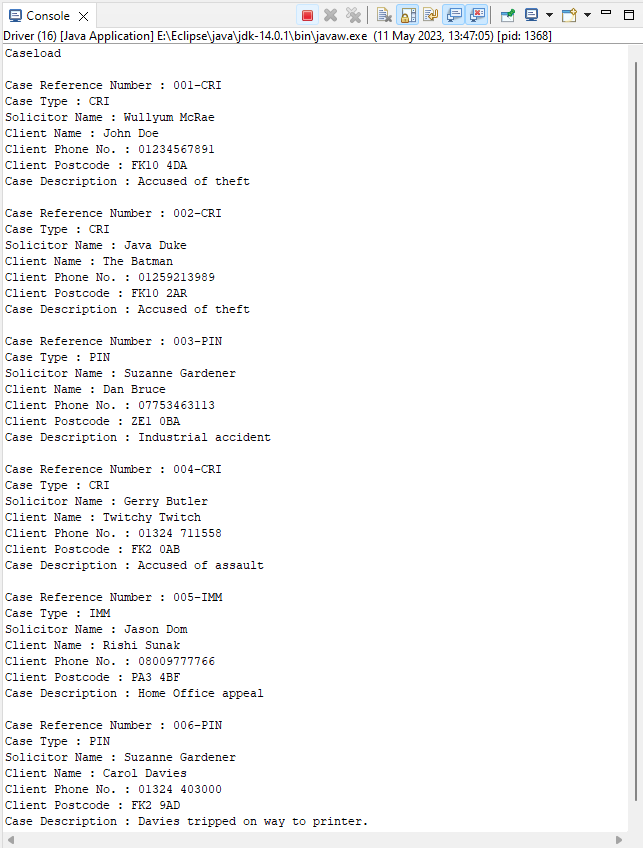


Figure Case details show the newly added Case

The new Case

## Appendix 10

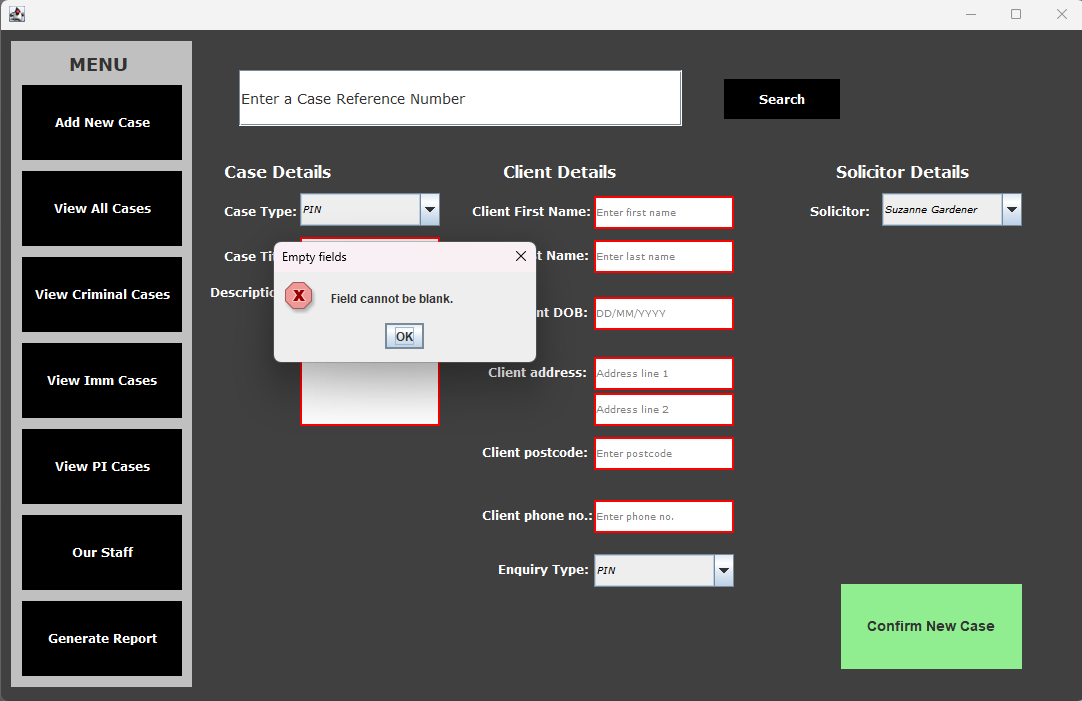
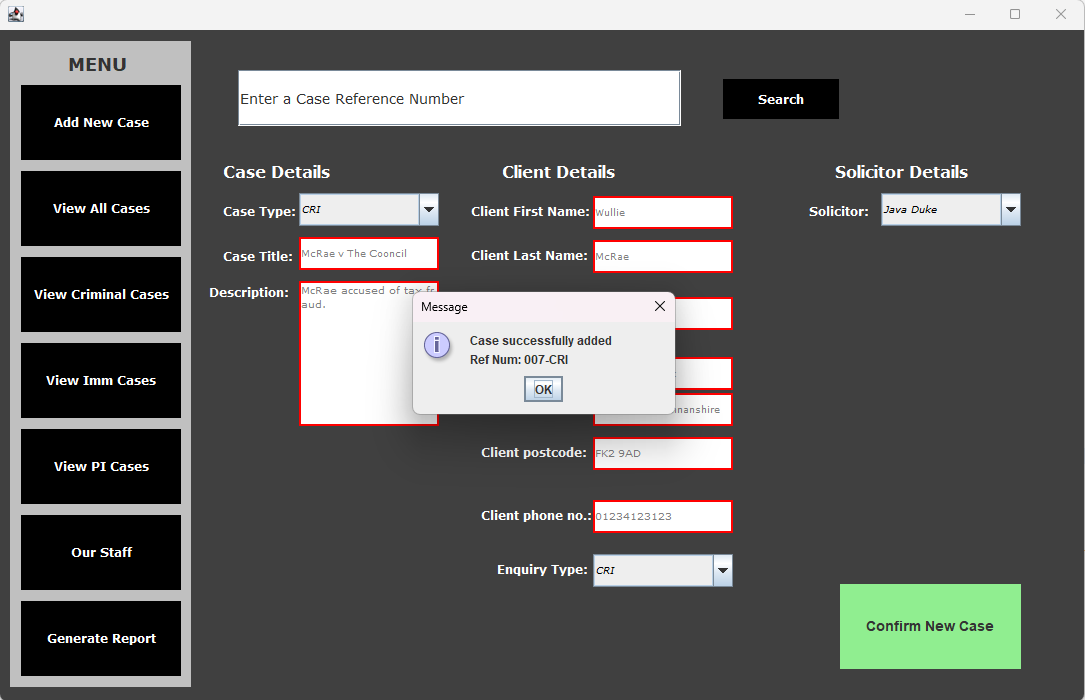


Figure Blank fields are not allowed

## Appendix 11

Figure Case created after trying to enter blank fields



## Appendix 12

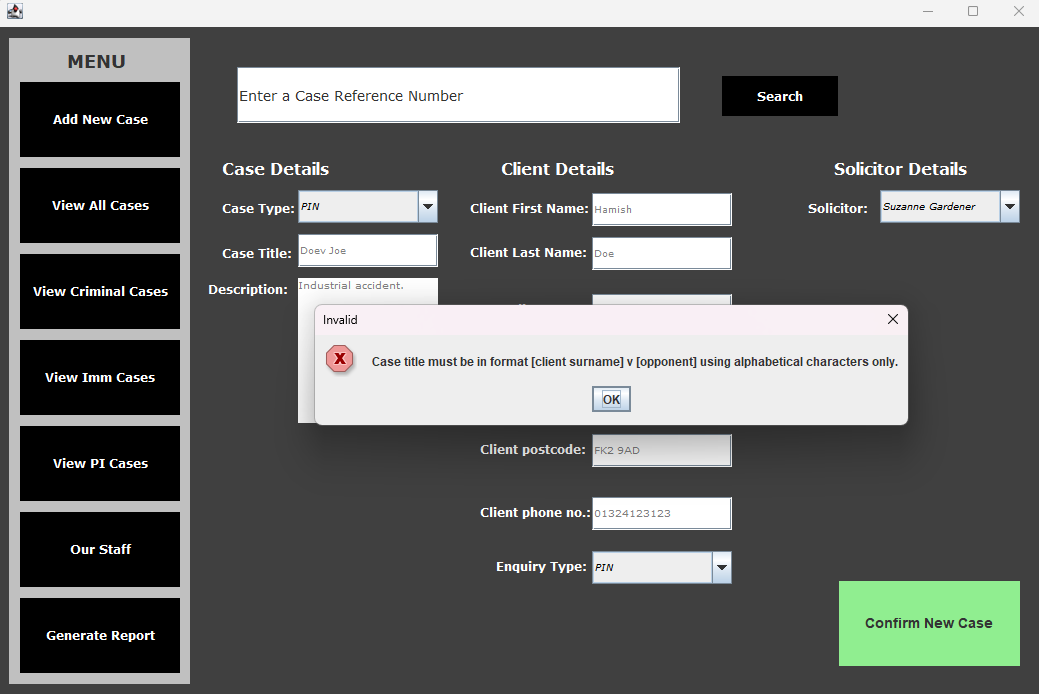


Figure Invalid Case Title error message

## Appendix 13

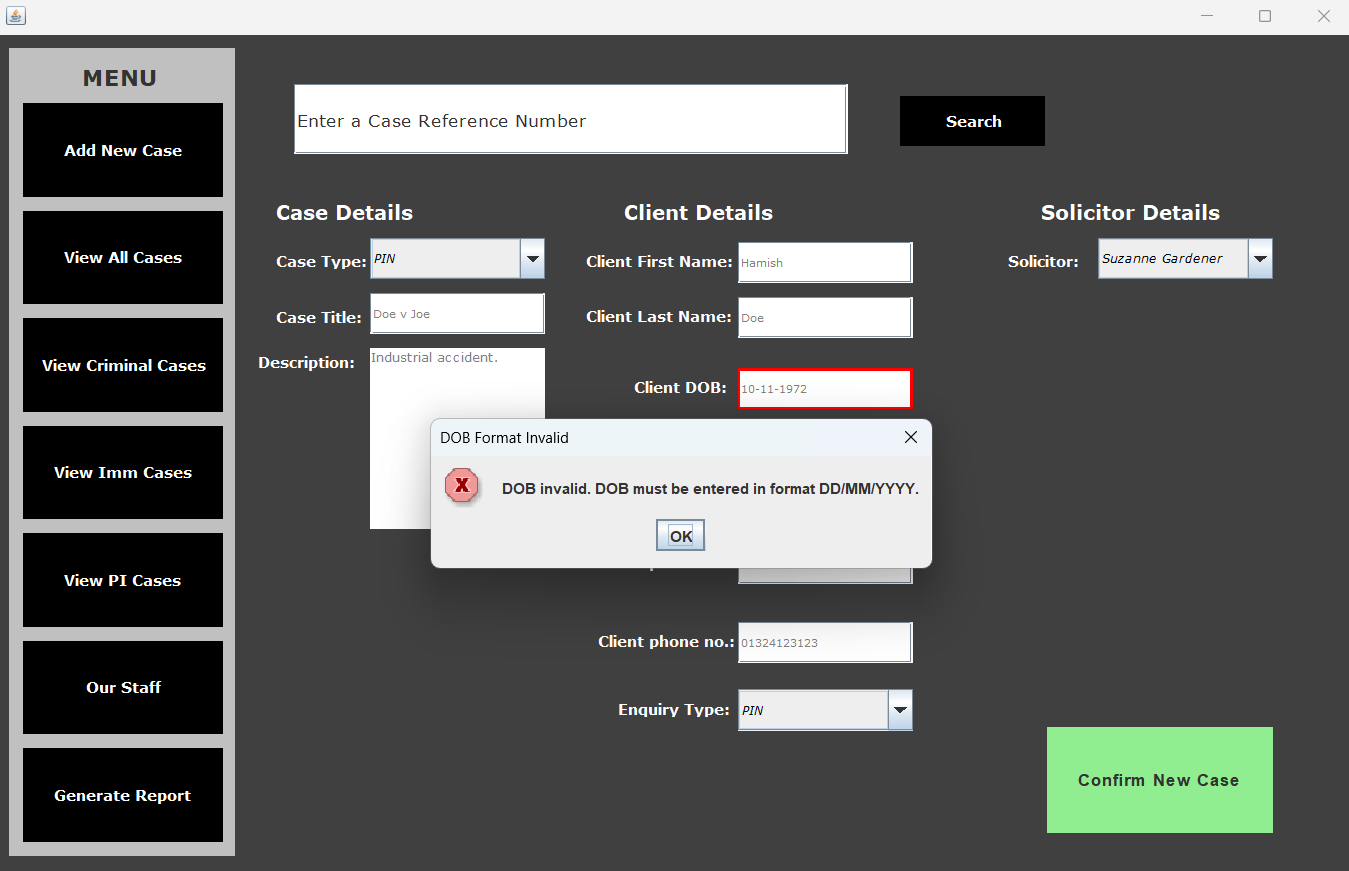


Figure Incorrect separator used in DOB

## Appendix 14

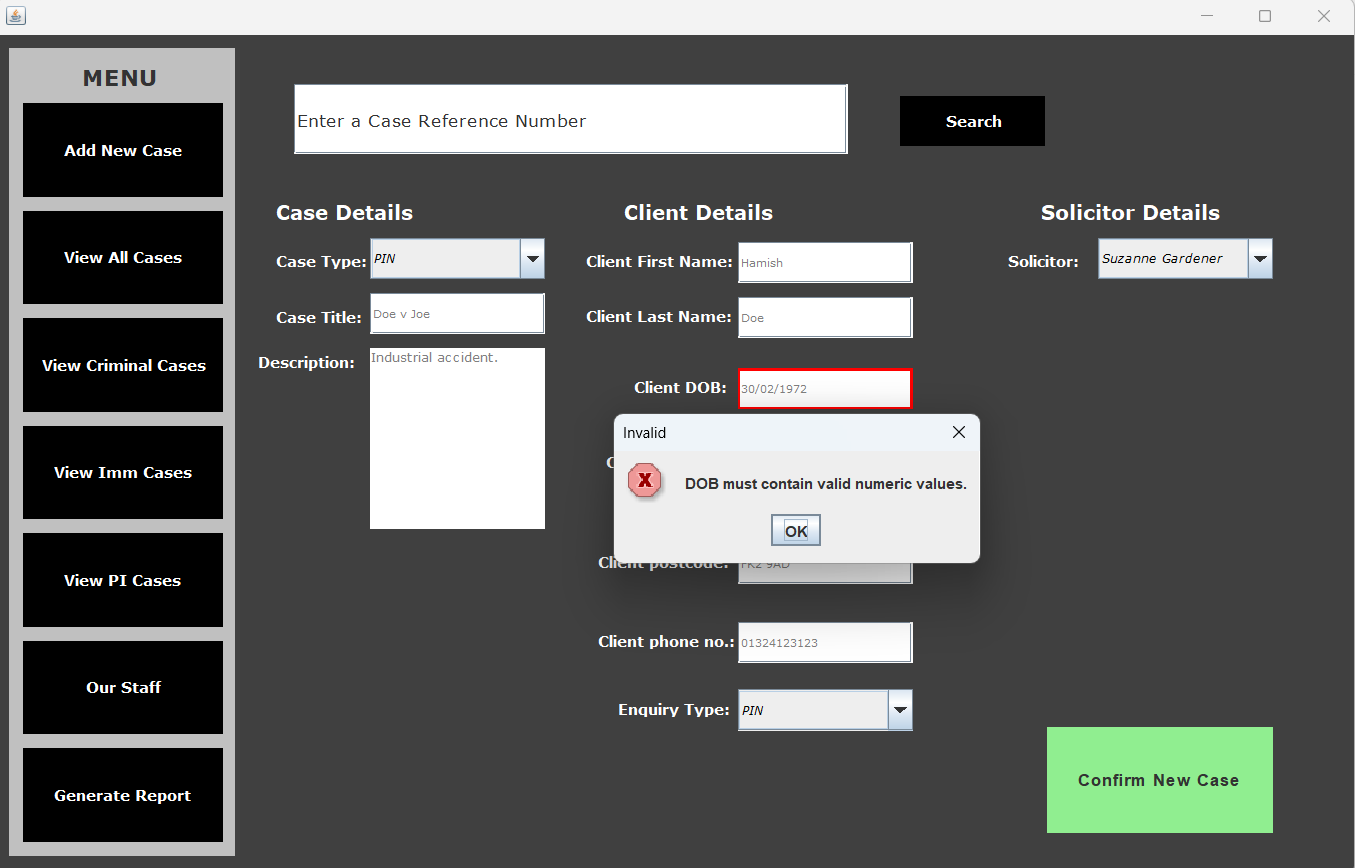


Figure Upper bounds of February in a leap year not accepted

## Appendix 15

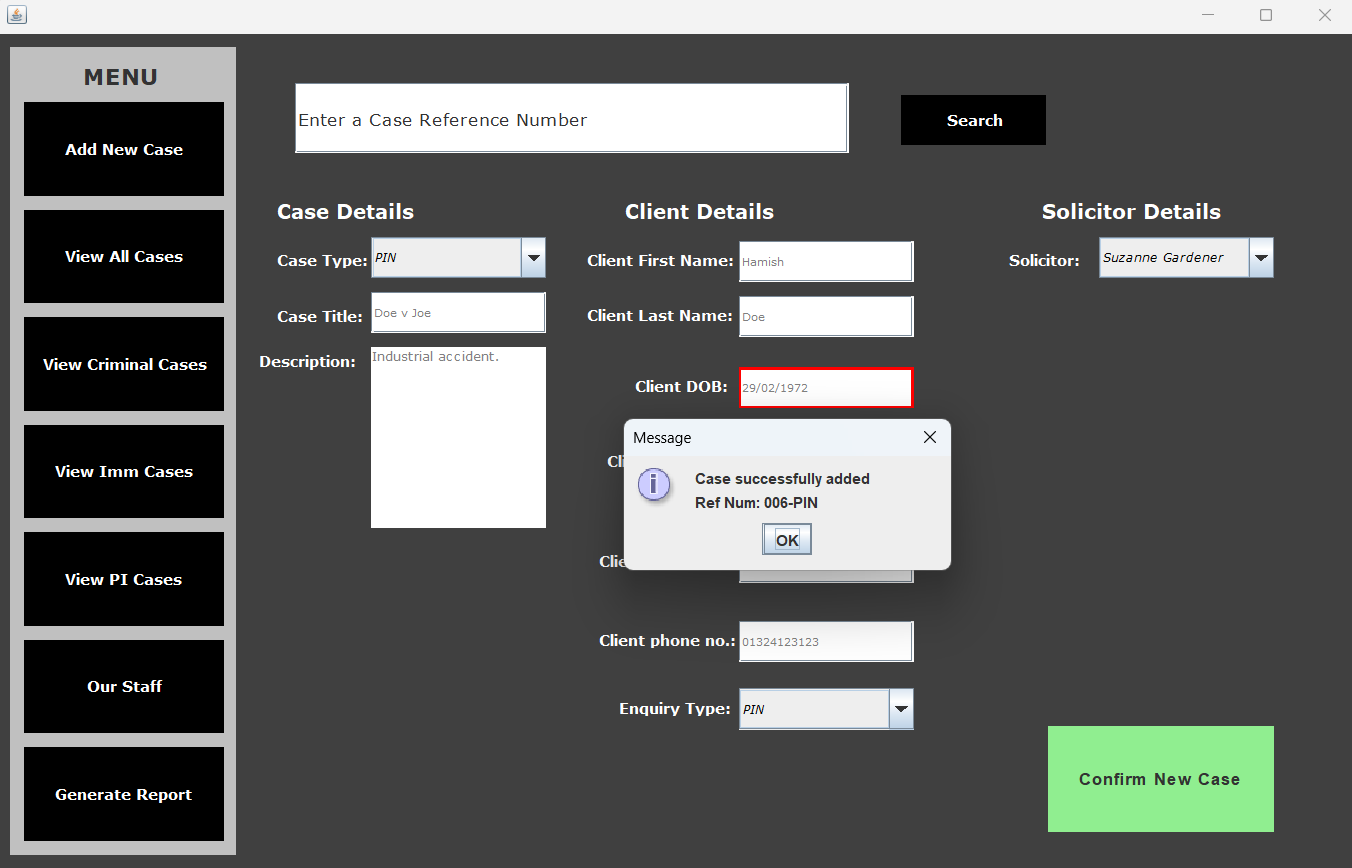


Figure Normal data for February if it is a leap year

## Appendix 16

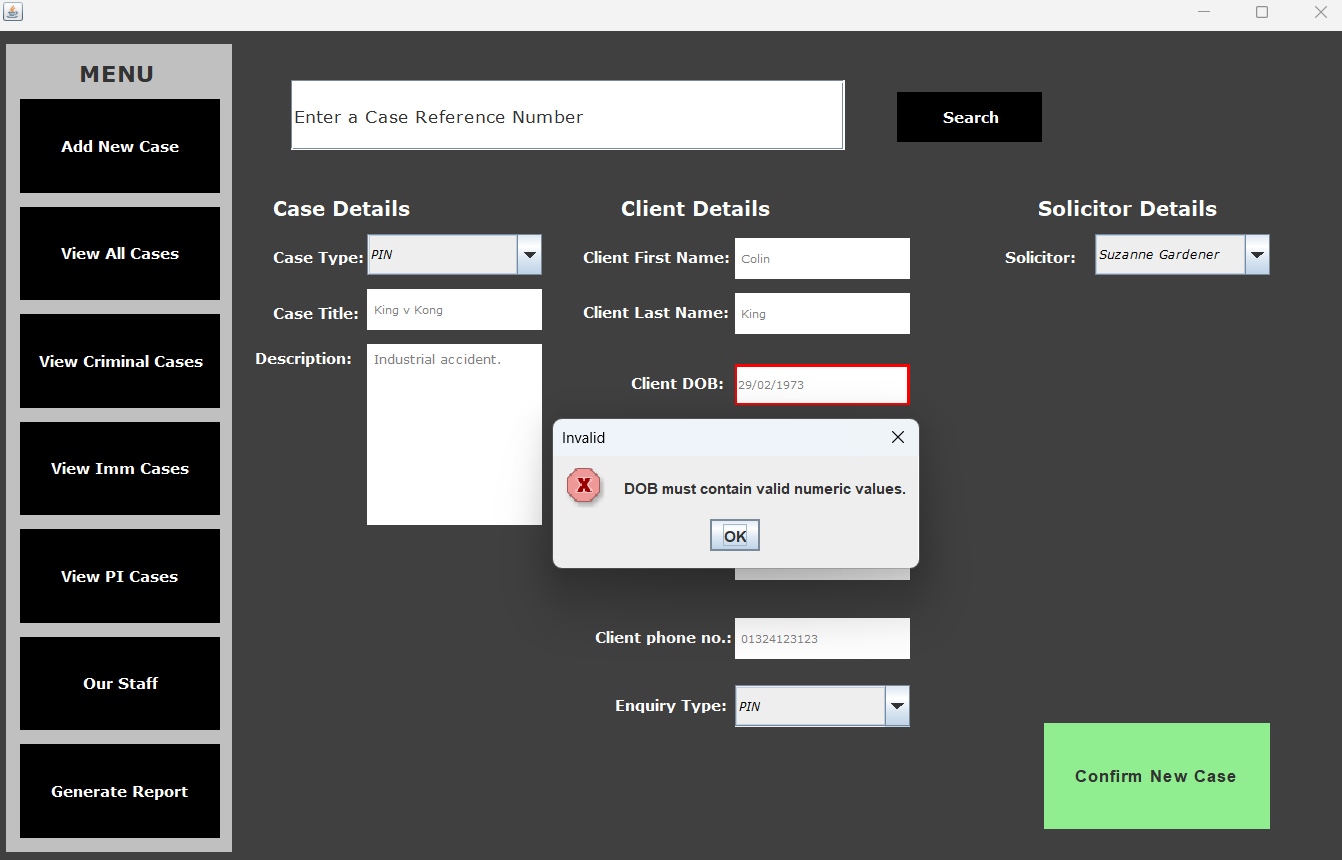


Figure Upper bounds of February if it is not a leap year

## Appendix 17

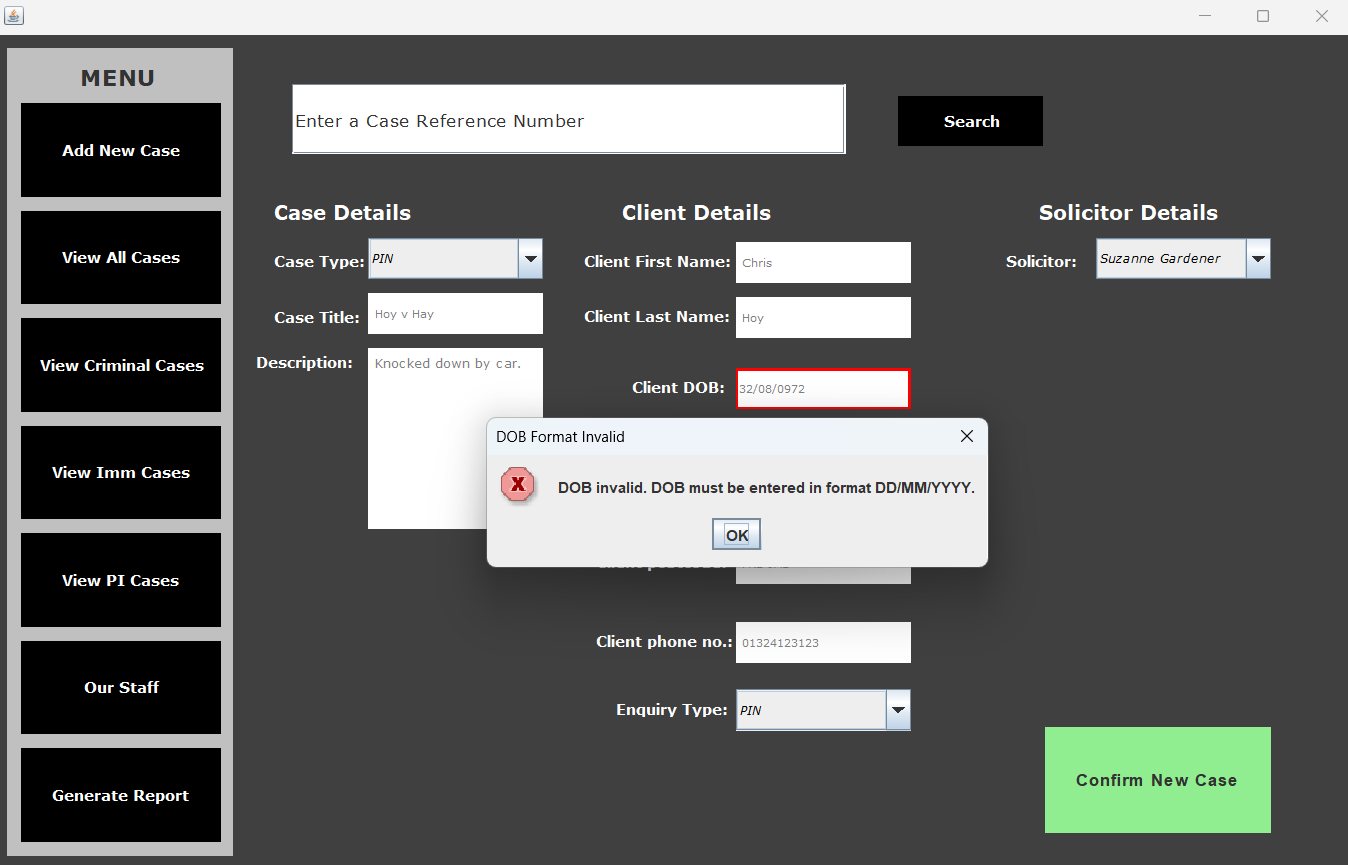


Figure YYYY value must start with 1 or 2

## Appendix 18

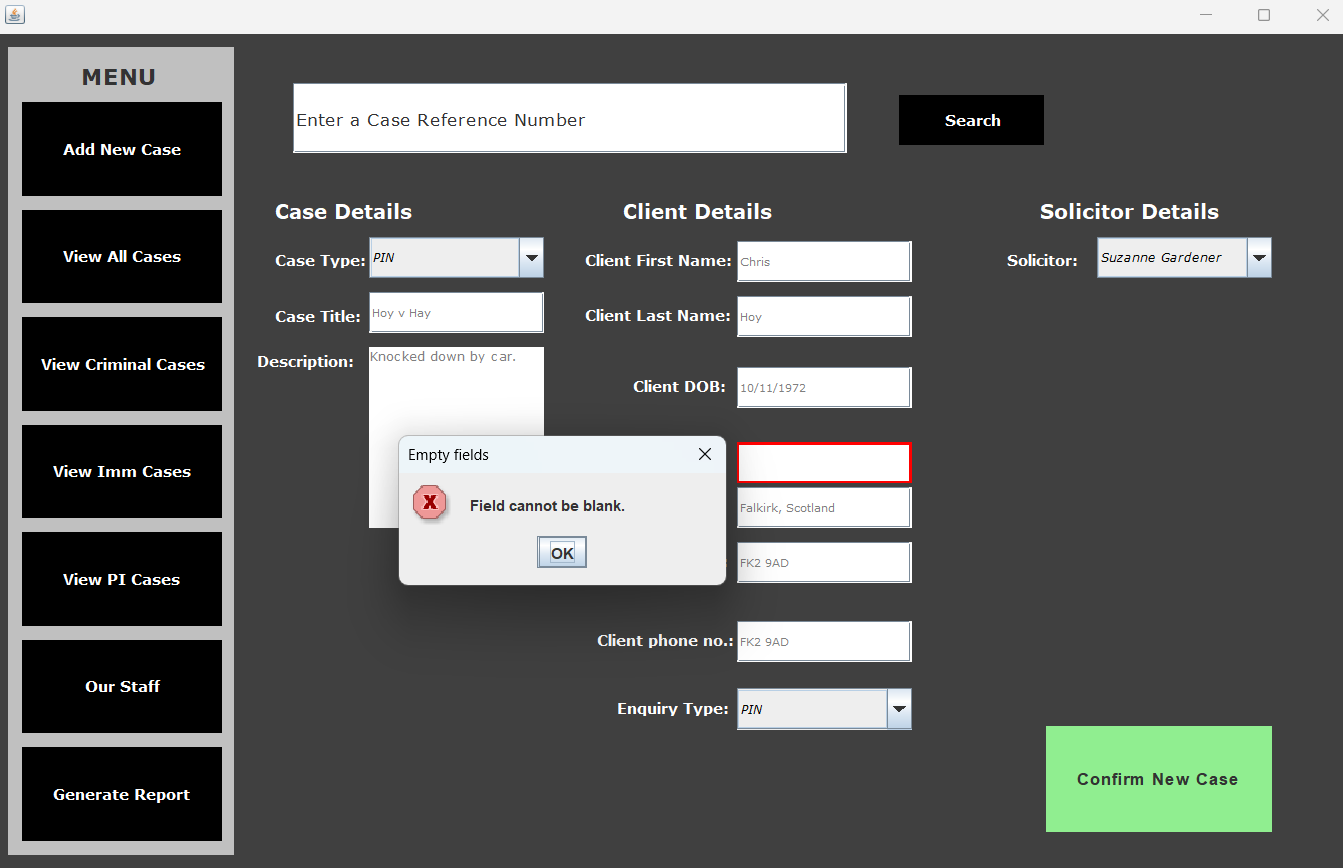


Figure Address line 1 only outlined red

## Appendix 19

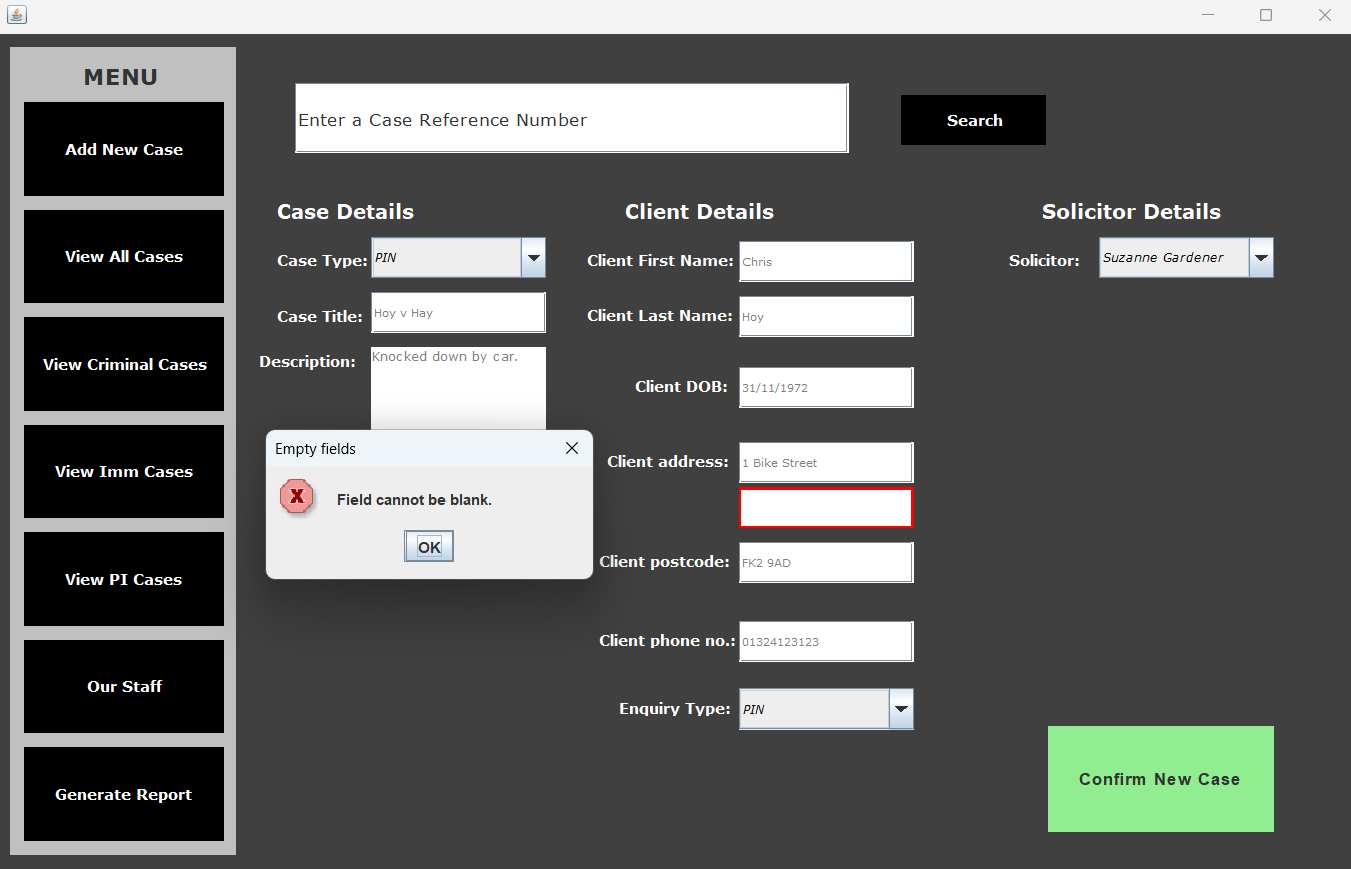


Figure Address line 2 only outlined red

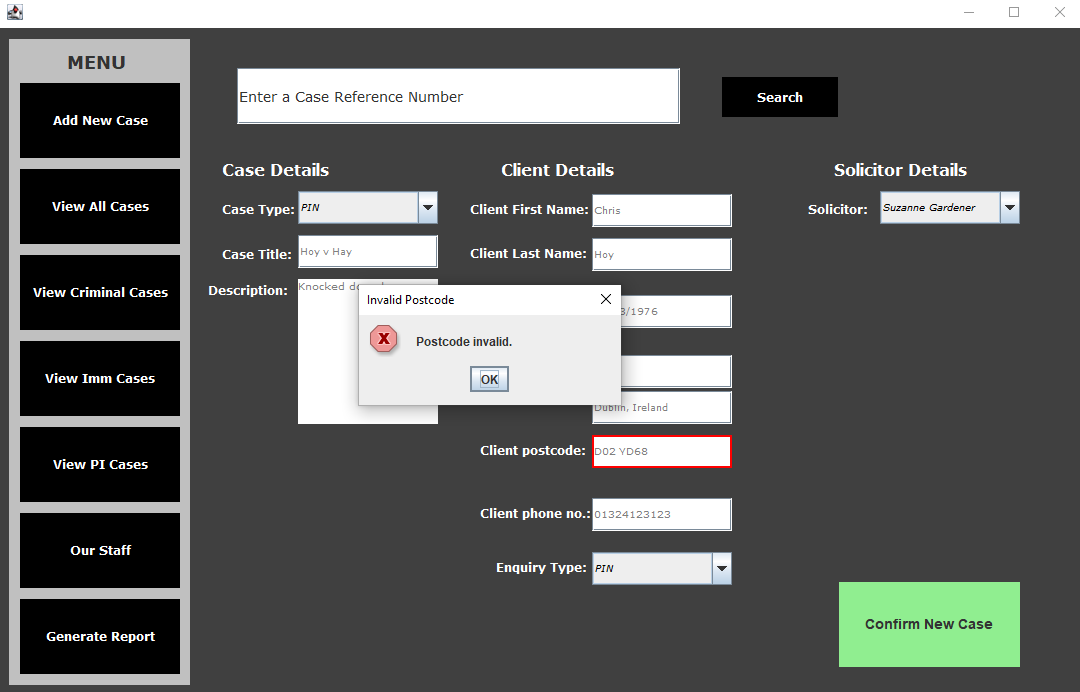
## Appendix 20



Figure Postcode using illegal separator

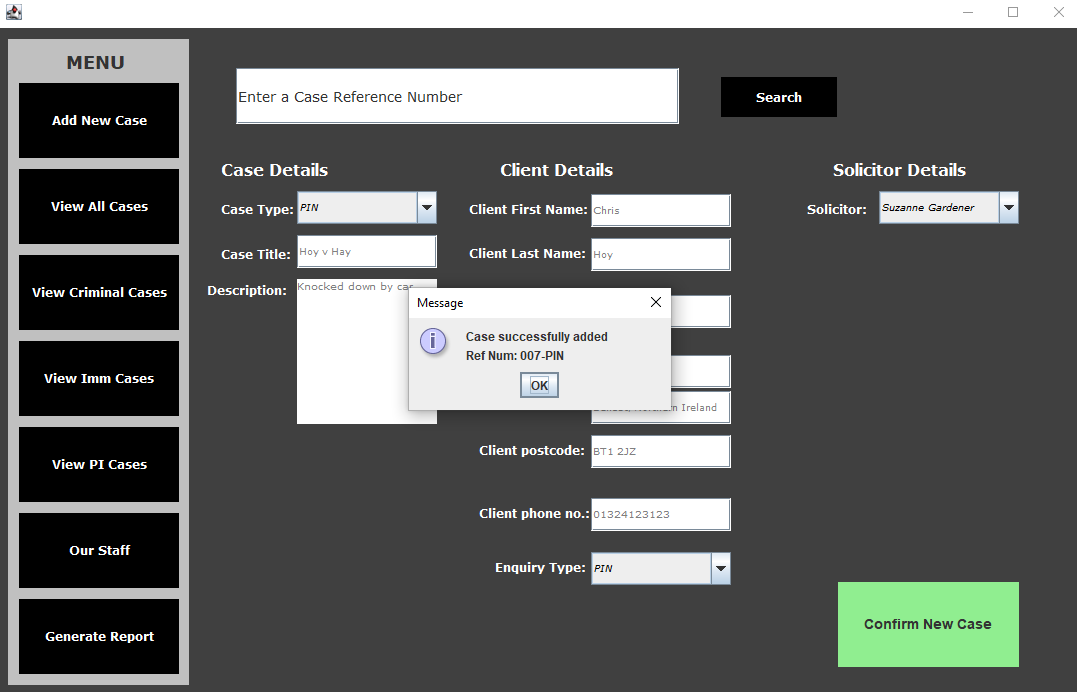
## Appendix 21

Figure Non-UK postcode (Dublin) not acceptable



## Appendix 22

Figure Belfast postcode accepted



## Appendix 23

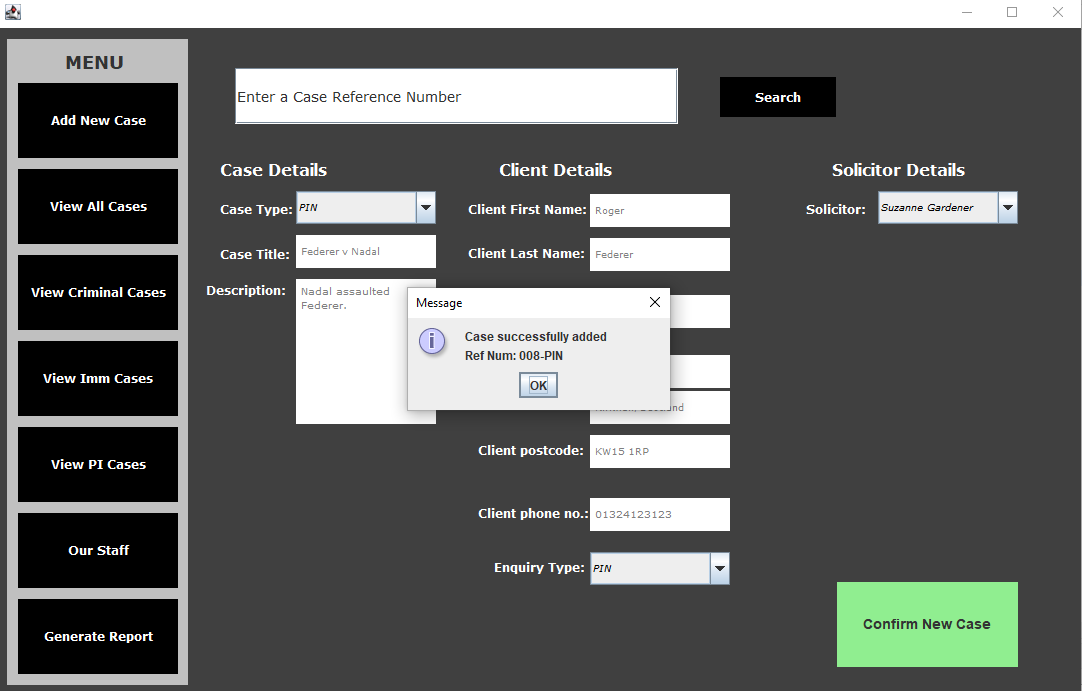


Figure Case created using Kirkwall postcode

## Appendix 24

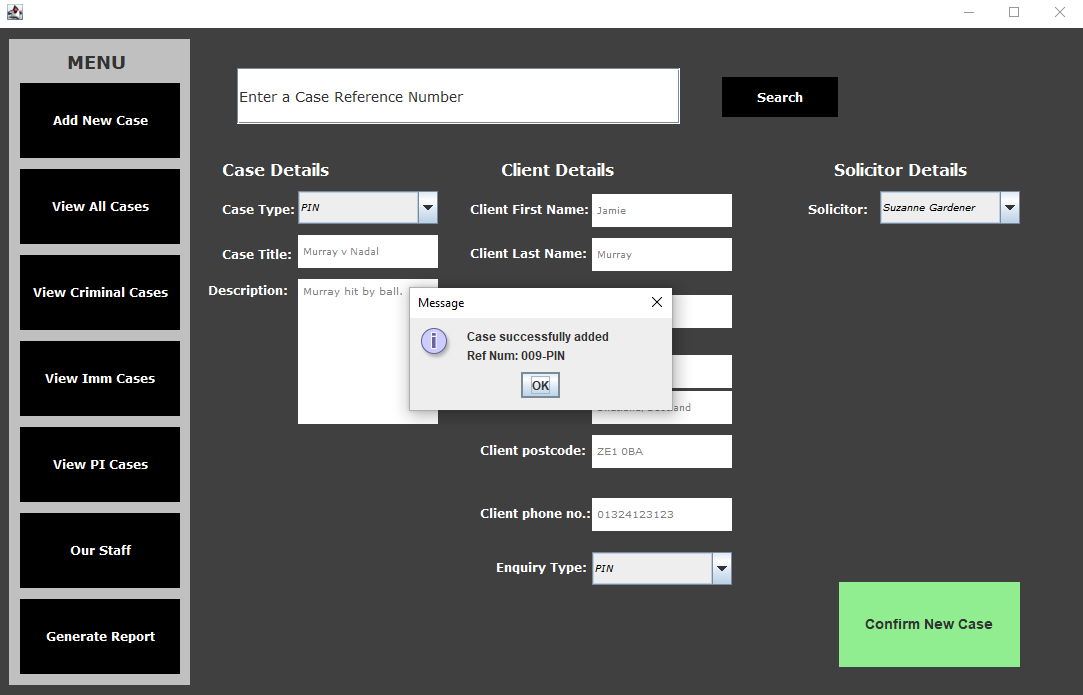


Figure Case created using Shetland postcode

## Appendix 25

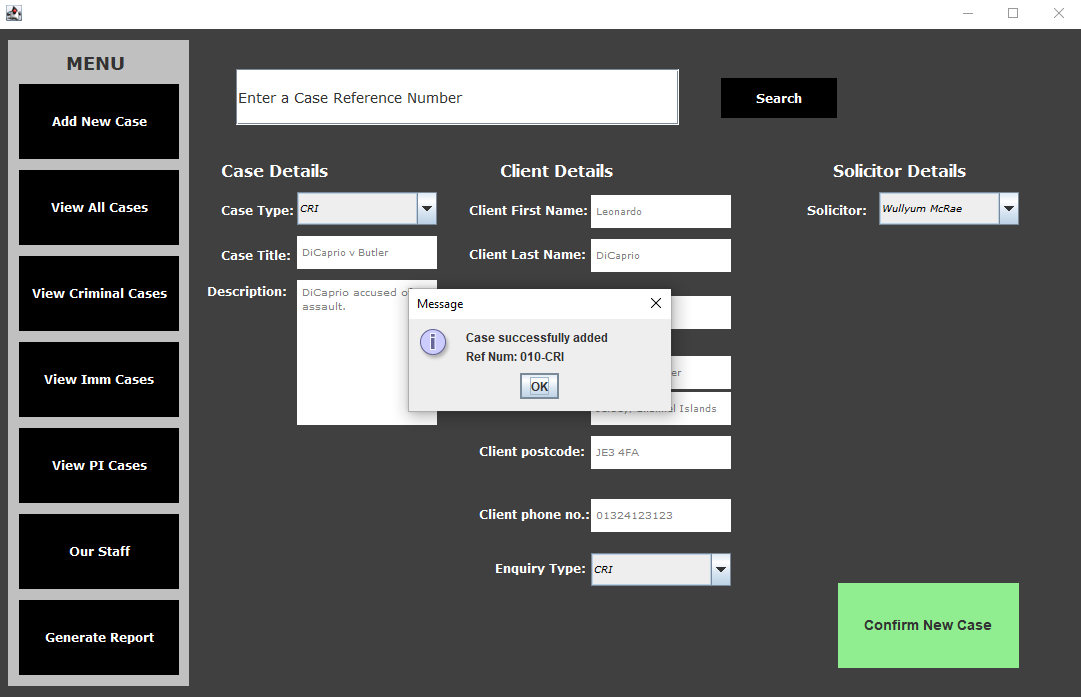


Figure Case created using Jersey postcode

## Appendix 26

Figure Postcode case insensitive

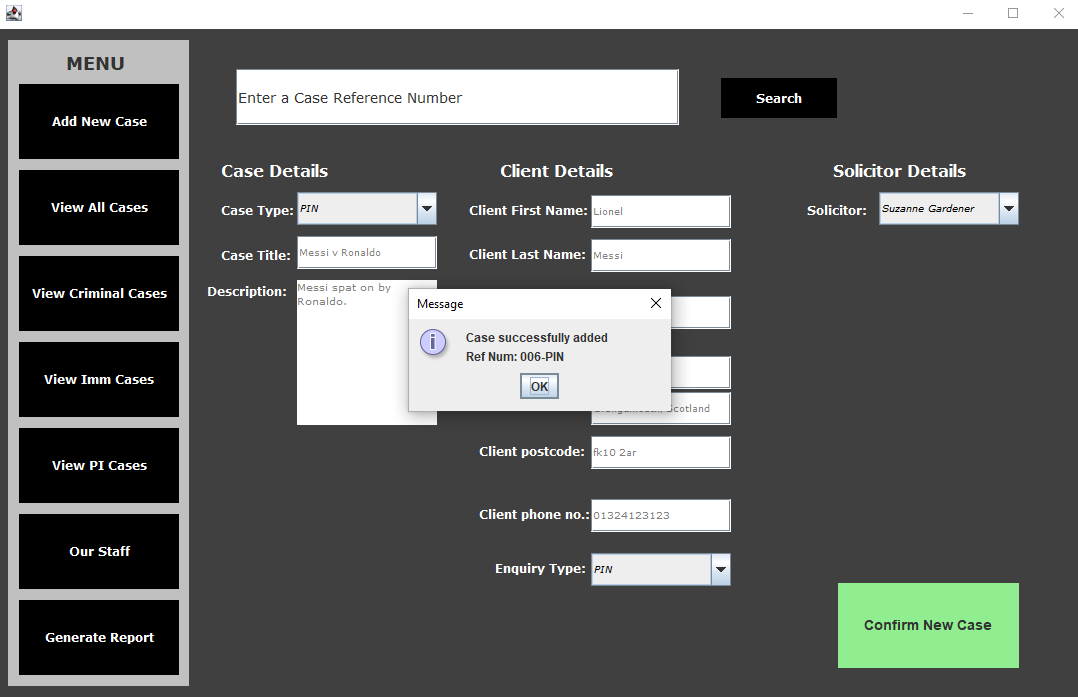
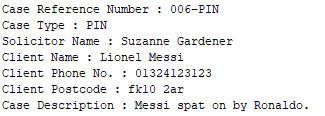


Figure Output showing postcode not formatted



## Appendix 27

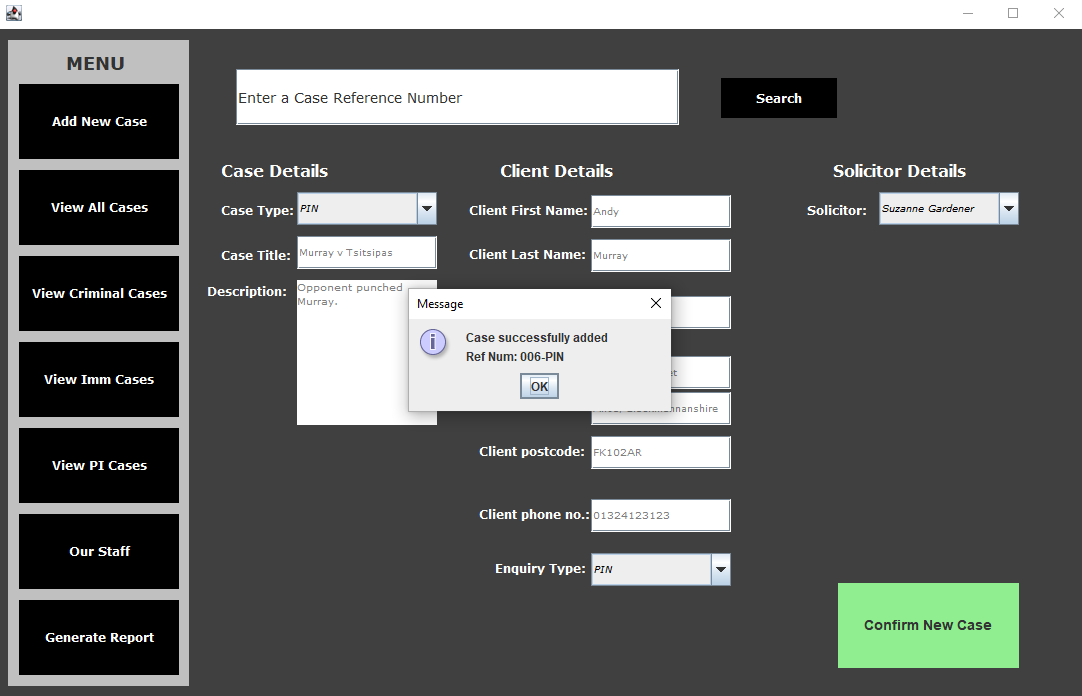


Figure Case created with postcode with no space

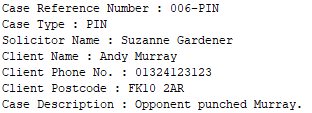


Figure Output displays postcode in proper format

## Appendix 28

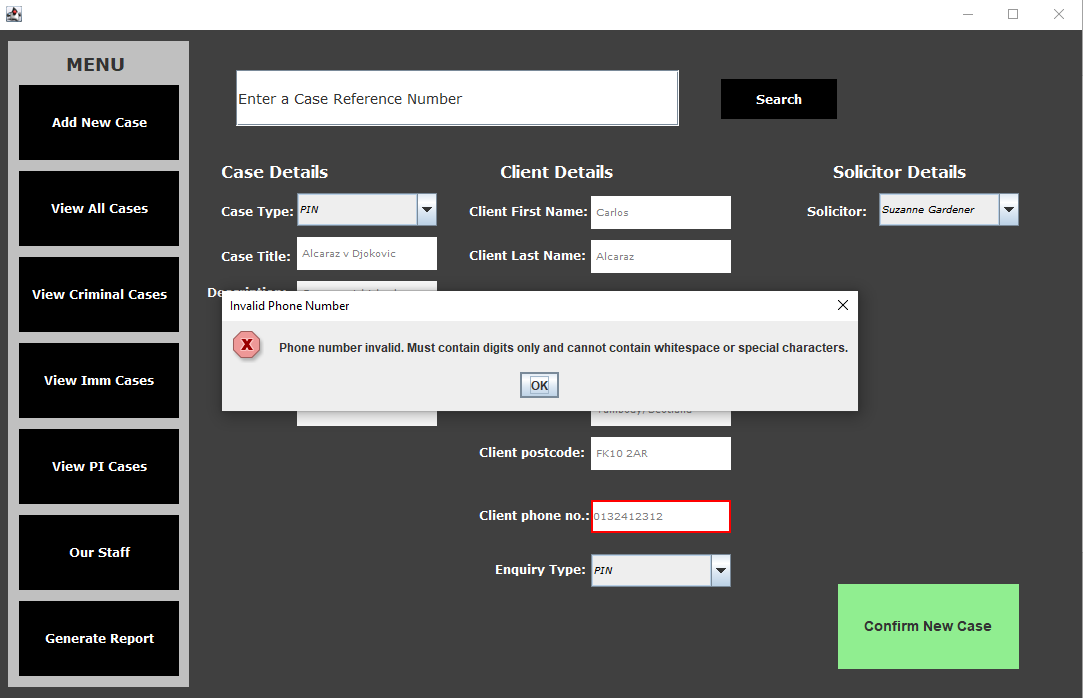


Figure Phone number length too short

## Appendix 29

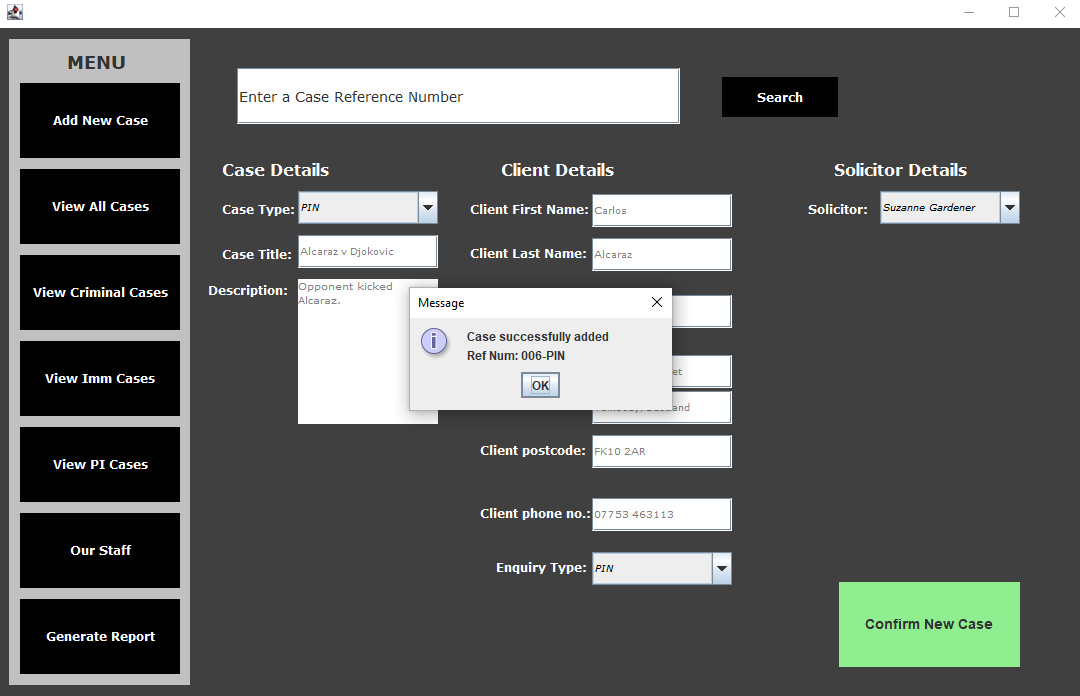


Figure Phone number including a space accepted

## Appendix 30

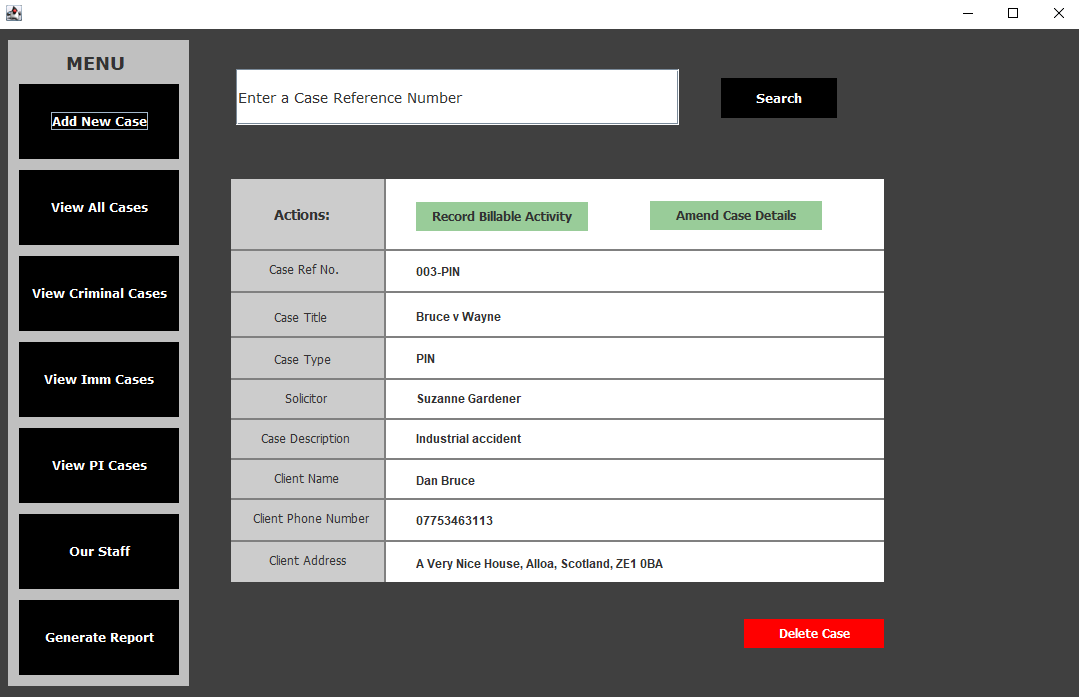


Figure Search results displayed in new window

## Appendix 31

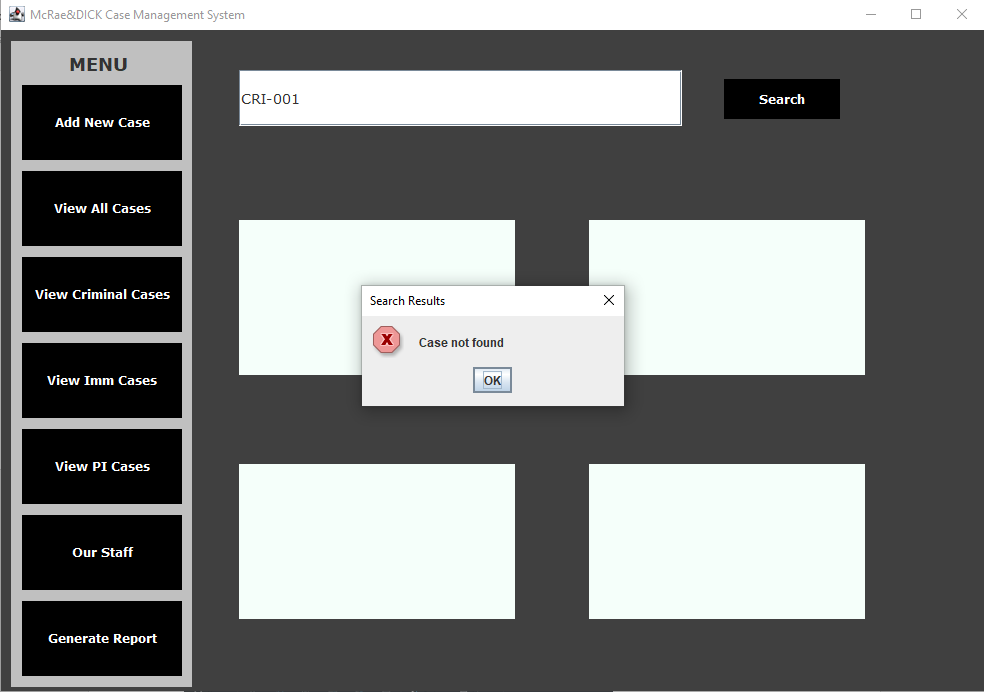


Figure Search results for invalid Case Reference Number

## Appendix 32

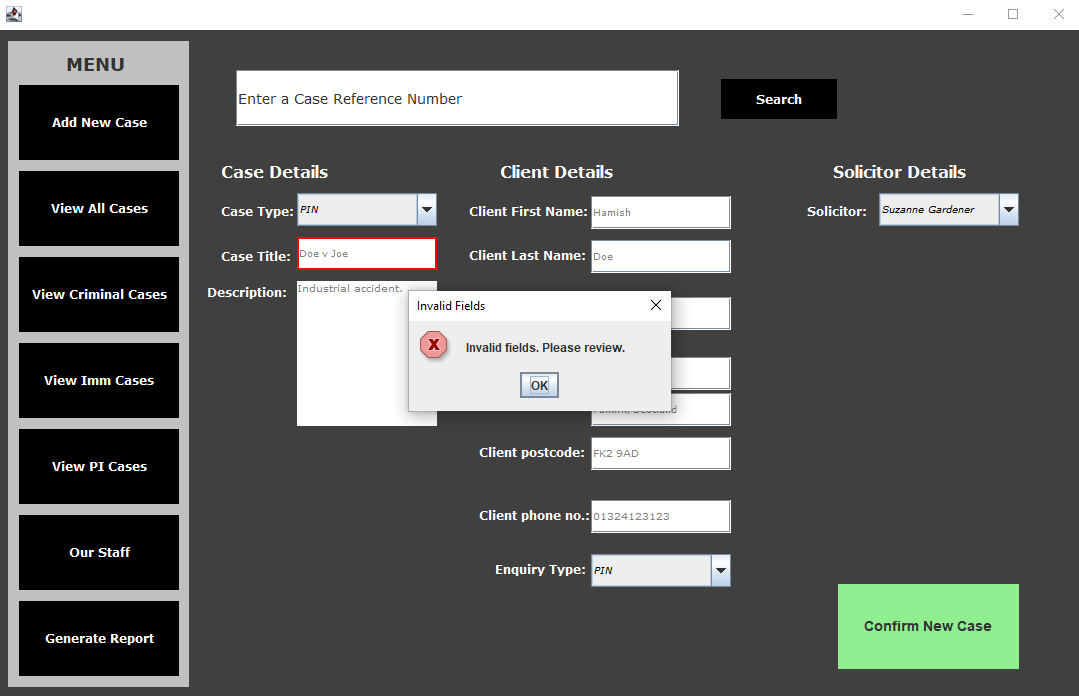


Figure Failed test - Case should be created after re-entering normal data

1. Found at: <https://docs.oracle.com/javase/tutorial/uiswing/components/icon.html> [↑](#footnote-ref-1)