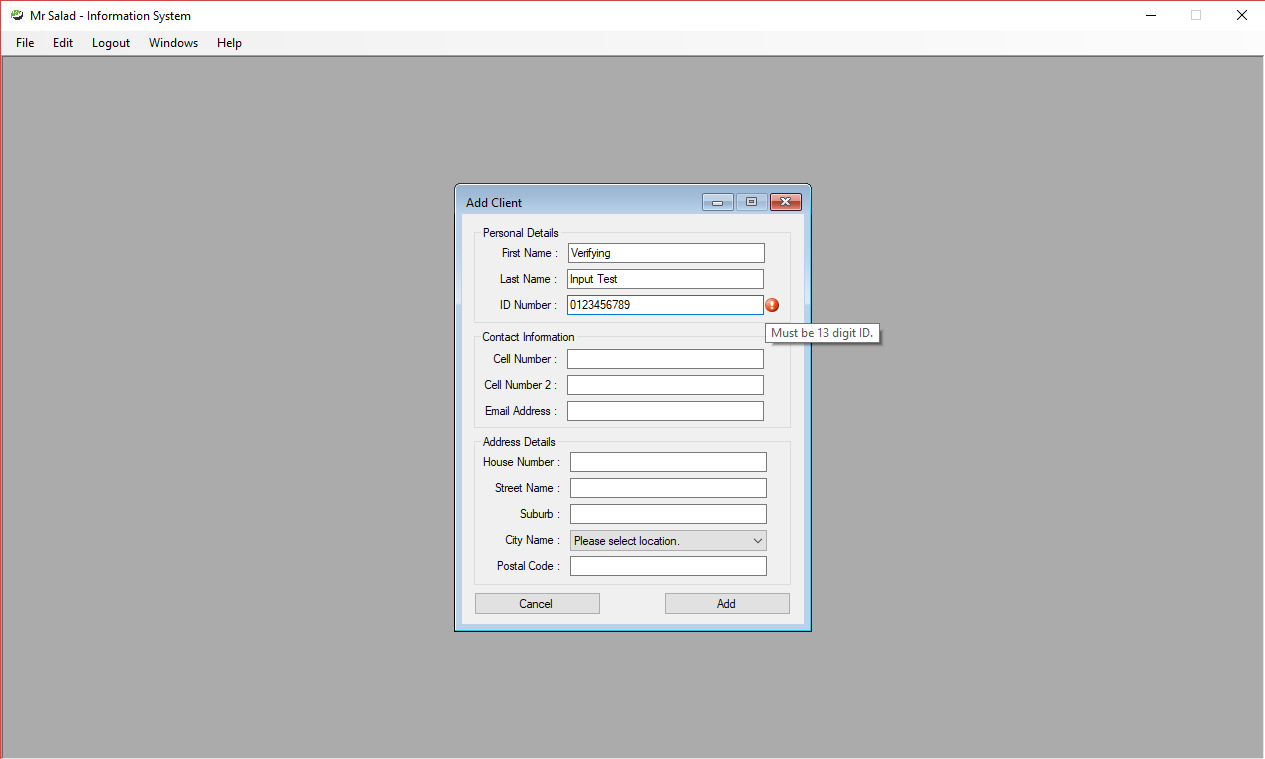
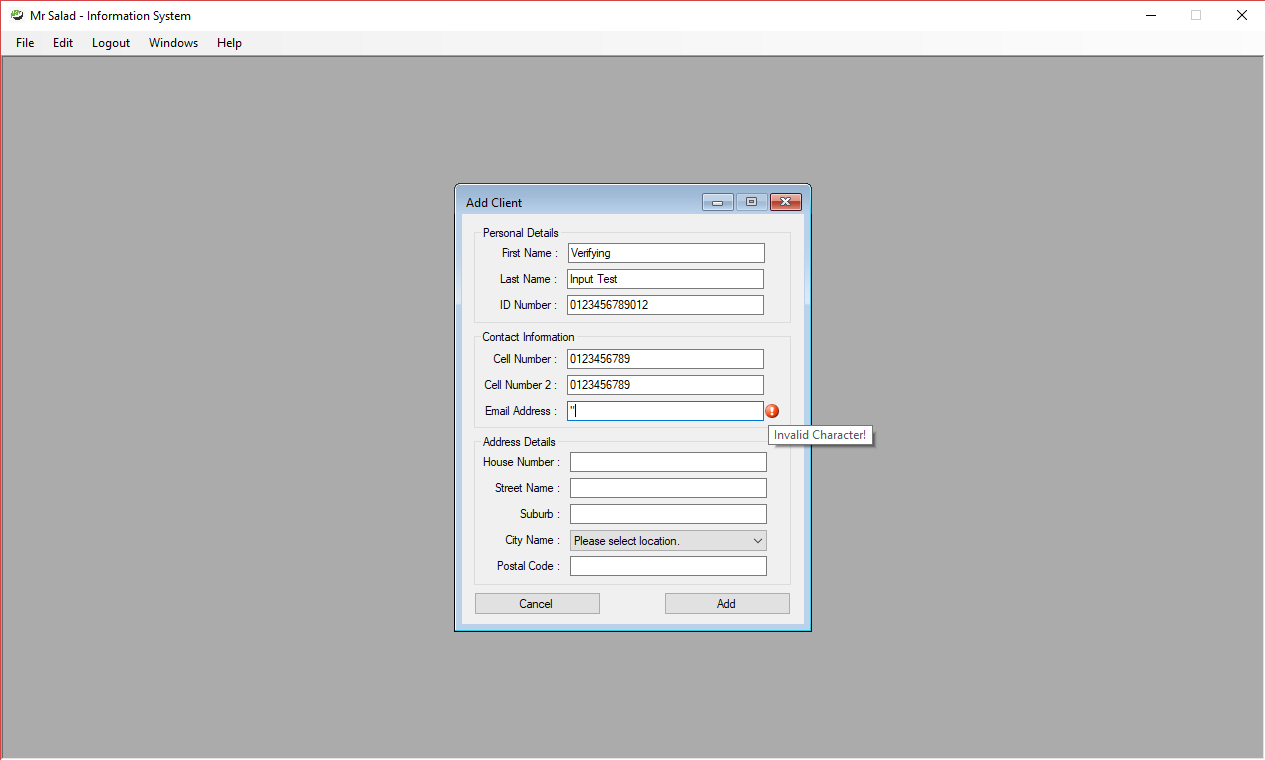
8. Example Code

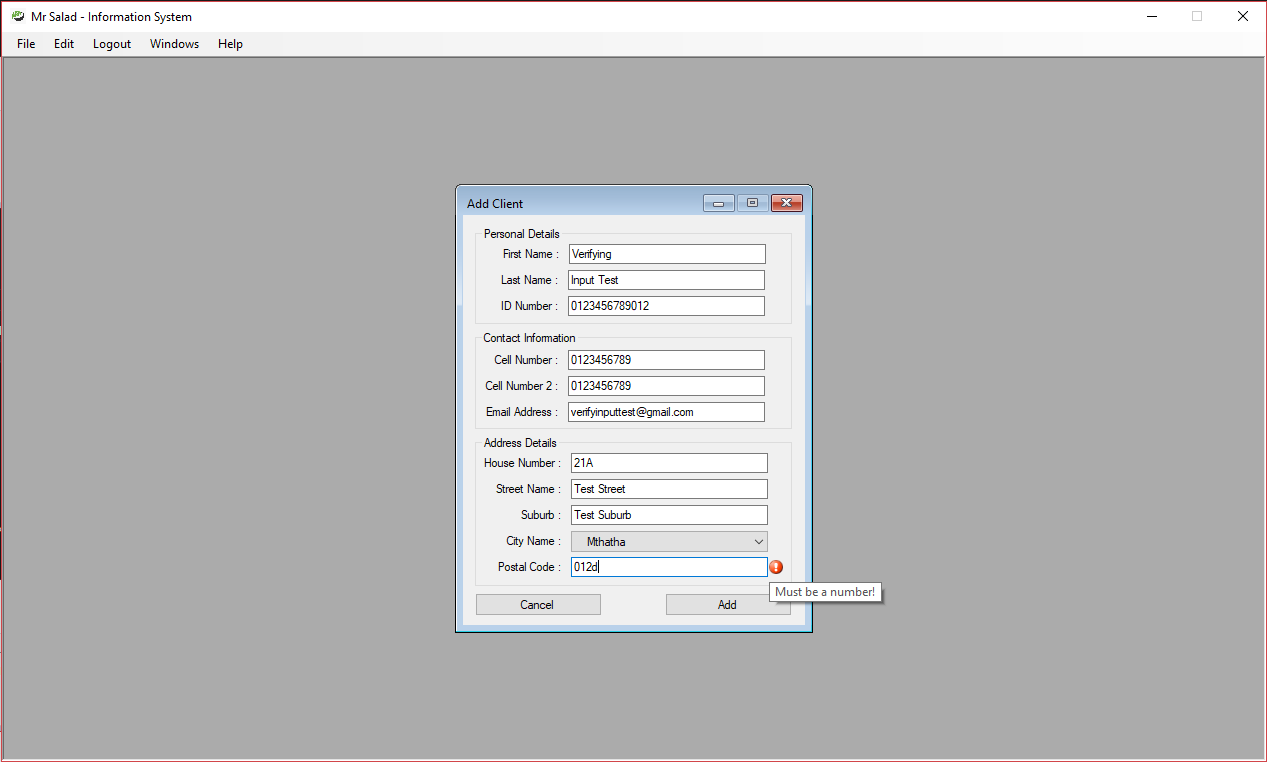
* 1. Verifying input



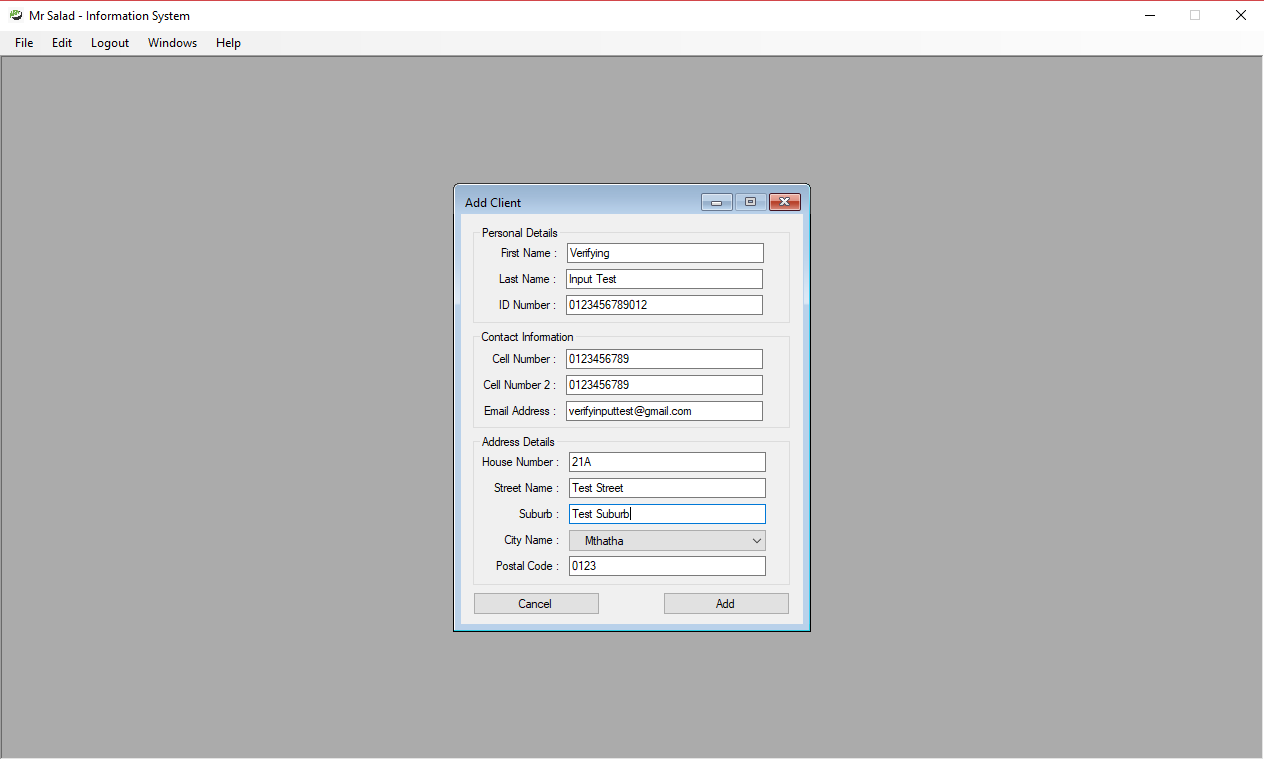
**Figure 1. Digit Length Verification**



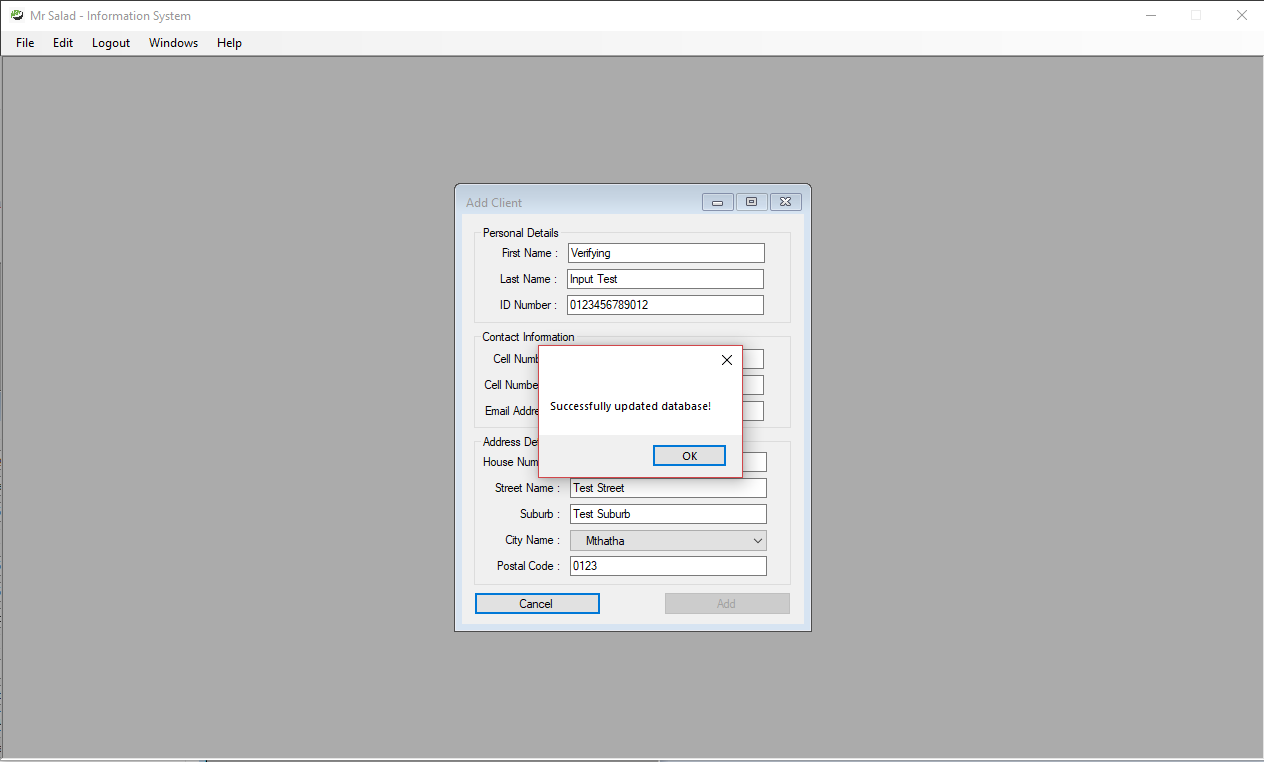
**Figure 2. Invalid Characters with regards to SQL injections**

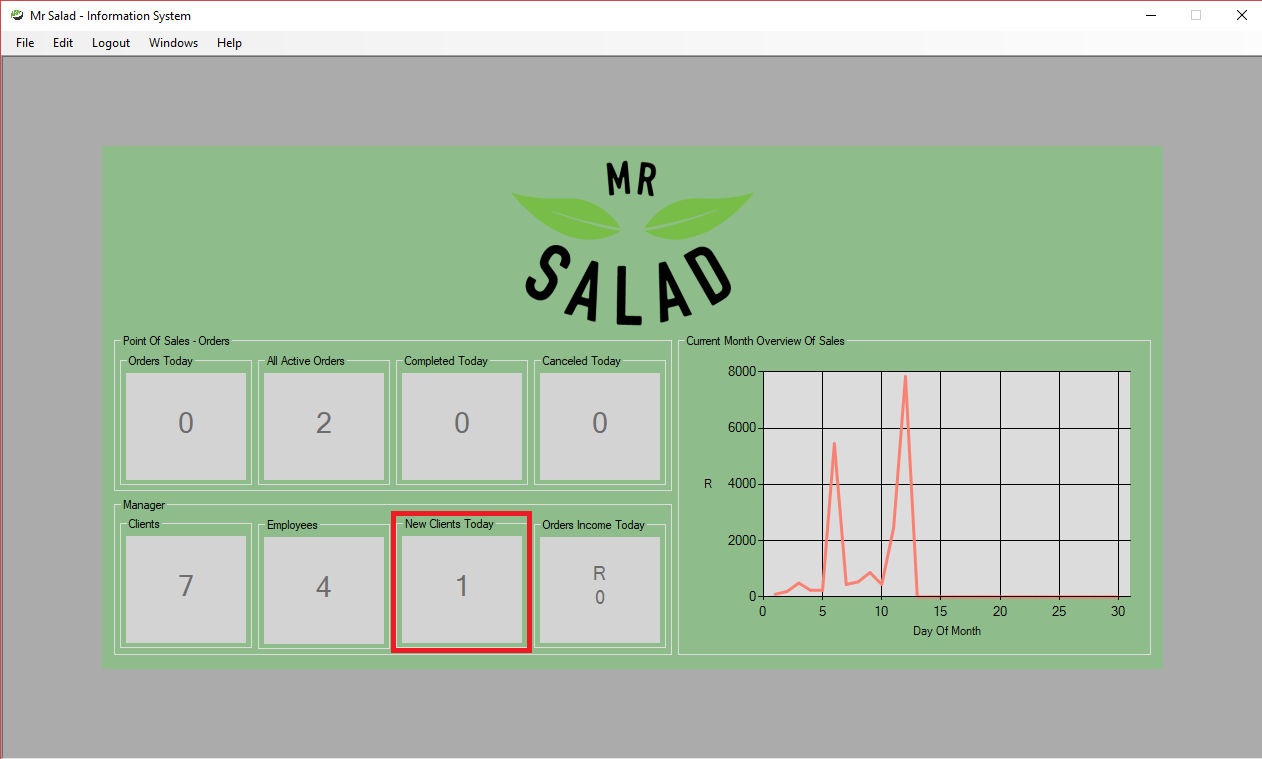


**Figure 3. Character Type Verification**

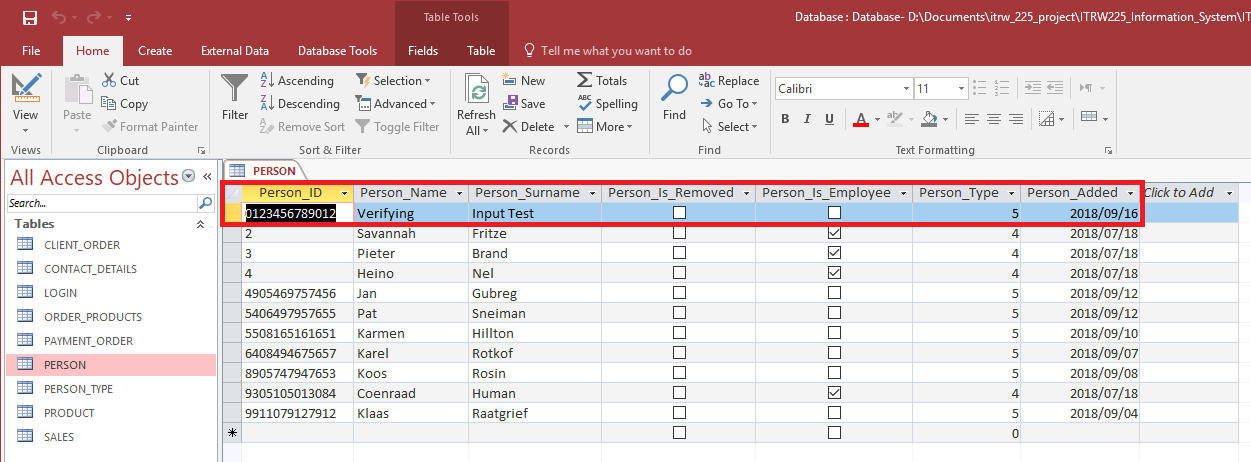


**Figure 4. All fields have valid input**

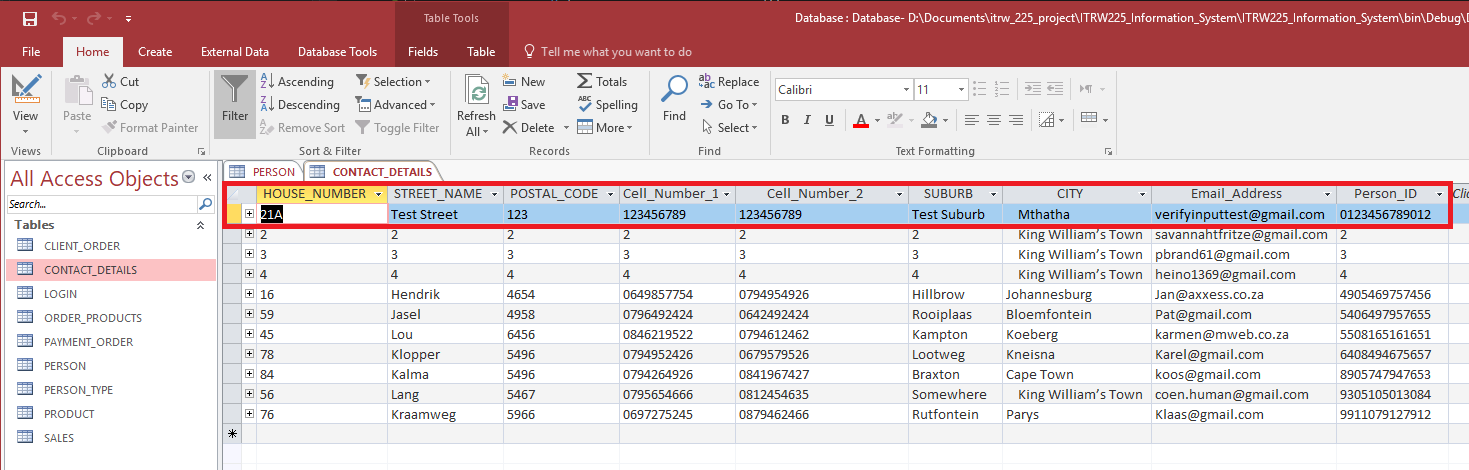
**Figure 5. Database updated successfully**



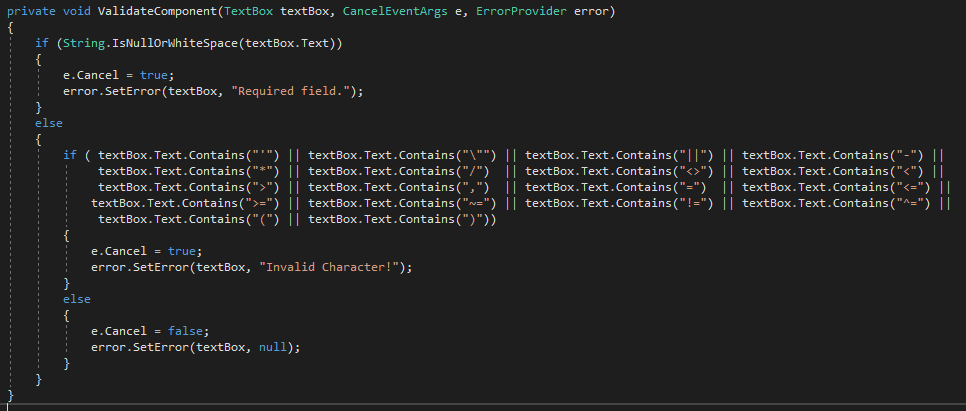
**Figure 6. Updated dashboard**



**Figure 7. Verifying updated input in database (PERSON table)**

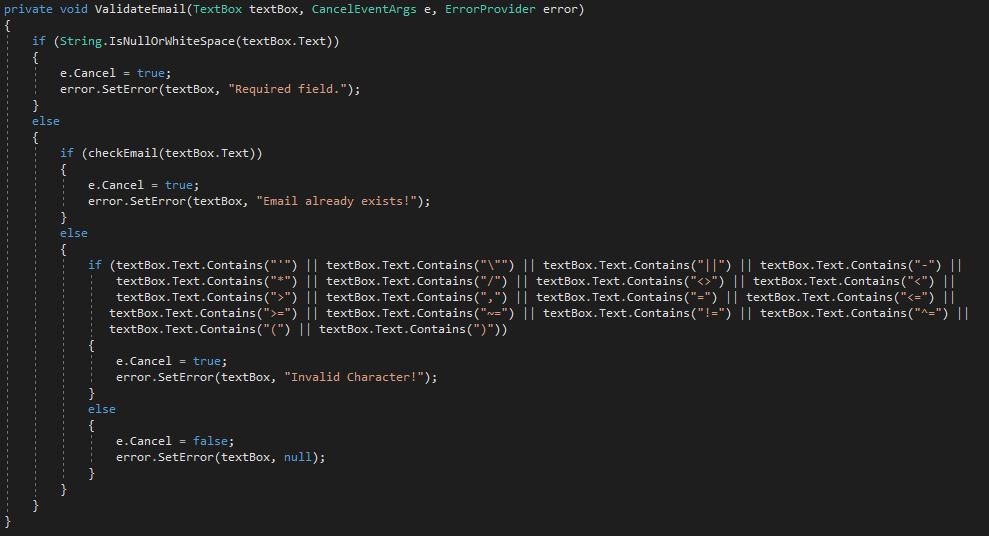


**Figure 8. Verifying updated input in database (CONTACT\_DETAILS table)**



**Figure 9. Code snippet of Validation on character entered**

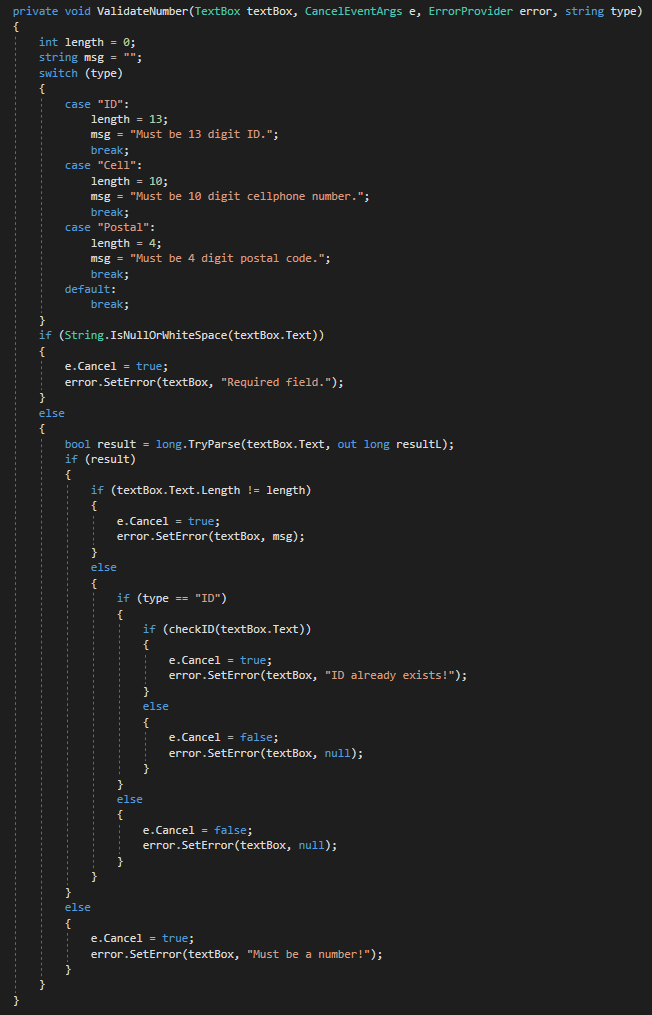
The method, ValidateComponent, used in Figure 9. is used to make sure when a special character is entered the program doesn’t crash but rather throws an error asking to input a valid character.



**Figure 10. Code Snippet of Validation on Email (Figure 2)**

The method, ValidateEmail, used in Figure 10. is used to make sure when a user inputs their email and it isn’t valid the program doesn’t crash but rather throws an error asking the user to re-enter email. The method also checks if the email is already in use and whether the user has entered an email seeing as it’s a required field.

The method, ValidateNumber, is a method that checks the length of the characters entered by the users to make sure the string is valid. Also makes sure there is an input as the field is a required field.

  
**Figure 11. Code Snippet of Validation on input length**