Why Software Fails

# Think of software failures that have impacted you.

Last year’s project doing Vehicle Shop database using Java and SQL building GUI (Graphic User Interface), I faced a software failure that can be highlighted the role of software quality and testing.

The issue happened because of oversight in handling user input for one of the fields, and that field was Bank Card Number of the customers, this field was very important because it is related to the invoice and to the purchased vehicles by the customers, if this field got error the whole App is facing troubles in all the GUI’s tabs.

A screenshot of a computer

Description automatically generated

# What was it? What Failed?

where I have failed to set a limit for the following:

1. The limit for how many MAX\_VALUE of integers needs to be accepted by the user.  
   MAX\_VALUE is representing the maximum positive number in decimal format to be accepted that can be represented in 32 bits and the number is 2147483647. This means that no number of type Integer that is greater than 2147483647 can exist in Java.
2. The second failure is that the user input was not limited to only numbers, rather was accepting any input, symbols, characters, decimal numbers, and this can make error especially the variable that been used is assigned as type of integer.

# Could testing have fixed it?

Test would have absolutely made change to the positive side, especially on user inputs testing would have highlighted the potential error that can be occurred and for restrictions on non-integers inputs. Various testing methodologies, such as unit testing, integration testing, and acceptance testing, could have been used to ensure that the software behaved as intended under different scenarios.

Additionally, implementing error handling messages blocks is the proper way also for preventing the Vehicle Database App from crashing, this approach is more user friendly and more software stability!

Finally, glad to say that this error was address in resent GitHub commits and this was handled by providing better user experience and better software stability by providing error messages blocks to prevent potential App crash.

1. If more than 16 numbers are entered by the user, then a block message will pop up to inform the user   
   A screenshot of a computer

   Description automatically generated

A text block pops up to notify the user that only 16 digits allowed.

1. If the user enters non-integer to the field a (only number) text will be displayed on the GUI to inform the user that only numbers is allowed in this field  
   A screenshot of a computer

   Description automatically generated

A text on GUI notifying the user that only numbers allowed in this field!

Please check Vehicles database on GitHub <https://github.com/EskandarAtrakchi/Vehicle-Shop-DB>

It shows the Vehicle Shop Database repo and the approach that been followed to address those two issues, and here is the testing plan that was conducted to make sure that the software is as much reliable as possible <https://github.com/EskandarAtrakchi/Vehicle-Shop-DB/raw/master/DB-Test%20plan.xlsx>

The error was mitigated in those two methods.

A screenshot of a computer

Description automatically generated

Which is allowing the field to consume only digits otherwise a message on GUI will pop up and notify the user.

A screenshot of a computer

Description automatically generated

Which allowing the field to take only 16 digits if 16 digits or more a message block will pop up to notify the user that only 16 digits is allowed.