PASSWORD VALIDATION

Abstract

Verify the password validation check with all the given requirements in the Acceptance criteria.

Prathibha Somarajan Nair

prathibhasnair32@gmail.com

Table of Contents

Technical Challenge

ACME website for allowing a user to update their account password. Creating a non-faulty login page with the Acceptance criteria provided. Write test cases for Password validation functionality, and well-known understanding of the user scenarios before writing, coding, and texting the same. Any loophole will risk the user's personal information as well as unauthorized access to sensitive information.

Actual Acceptance Criteria provided

- New and Old password cannot be the same
- Passwords must be at least 8 characters long.
- Passwords must contain at least one number and one special character.
- Passwords must be ASCII characters

Expected Acceptance Criteria

As a user, I want to create a new password for my web form in the initial setup screen. I want the password to meet a certain level of security requirements:

- Must have a minimum length of 8 characters
- Must contain at least one number
- Must contain at least one special character.
- Passwords must be ASCII characters.(Must contain at least one capital letter and must contain at least one lowercase letter)
- New and Old password cannot be the same.
- Must contains no white space.

When the user enters a password and the criteria is not met, the following will happen:

- Message will appear telling the user that the password entered does not meet the minimum requirements
- The message shall describe the minimum requirements
- The Submit button is disabled.

When the user enters a password and the criteria is met, the following will happen:

- The Submit button is enabled.
- The password is stored and used to login in with.

Test Plan Management E.g.

Steps	Action	Expected result	Attachments
5.	input password note: - Must have a minimum length of 8 characters - Must contain at least one capital letter - Must contain at least one lowercase letter - Must contain at least one number - Must contain at least one special character.	when the password does not meet the requirement, the password field will show Red colour on the font. once the password is satisfied with the requirement, the password field will show Blue	
6.	input confirm password note: the password should be the same as the inputed password	when the password is same as the input password, the confirm password will show Blue otherwise, the password field will show Red	

Fig.1

Exploratory Tests:

- Integration Settings
- Slack Settings
- Agile Platform-Jira Integration
- GitHub Integration

Refer the **PasswordValidationTestSheet.xlsx** Exploratory Testing Sheet **for elaborated test coverage.**

Unit tests

- Include JUnit Library in the Java Project
- Created JUnit Test case (or Test Suite)
- Run Test case (or Test Suite)
- JUnit 4 and 5(regex) used as it includes the annotation feature.

On executing the JUnit class, the console and JUnit result tab shows up,
The Console shows input and expected output after the run.
The JUnit result tab displays mainly the number of test cases run, number of errors and number of failures encountered i.e.,
Run: 17/17(meaning 17 testcase out of 17 testcase ran), Errors: 0 (no errors found in the test case executed), Failures: 0(no test cases failed)
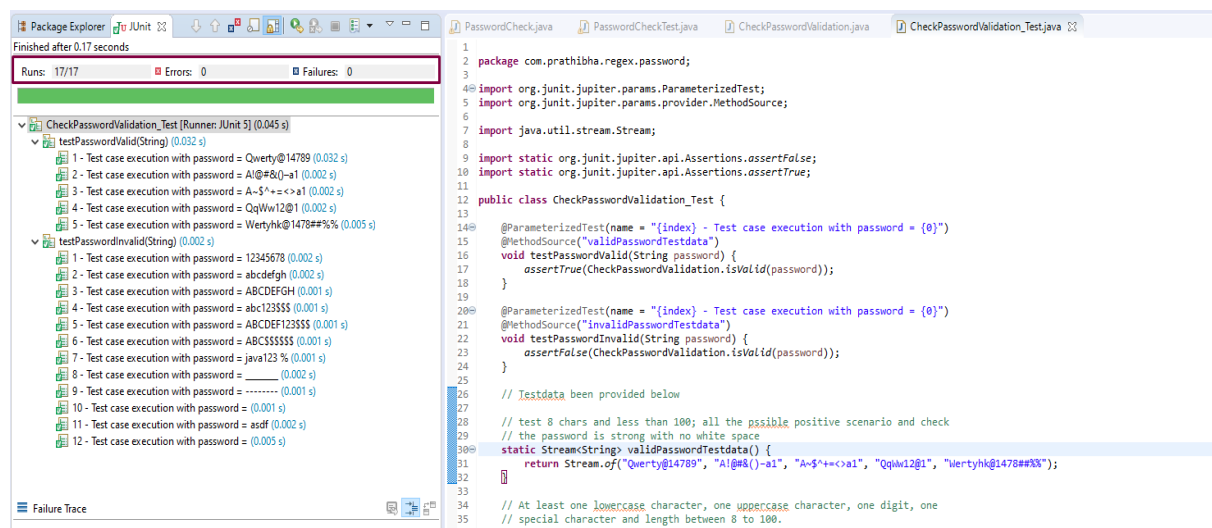

Fig.2

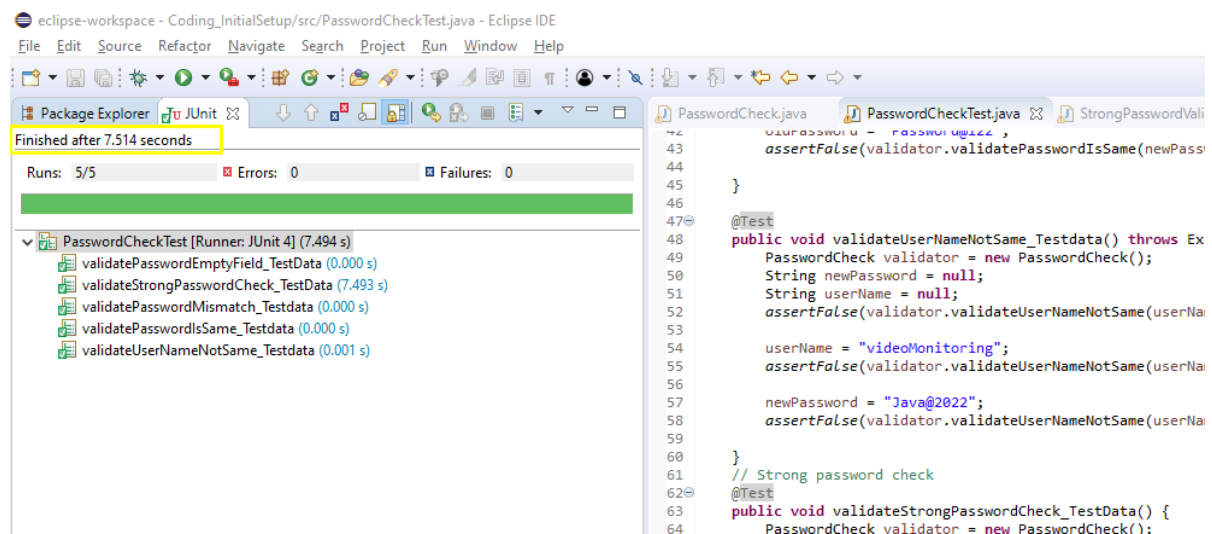The time taken to finish the execution of the tests.

Fig.3

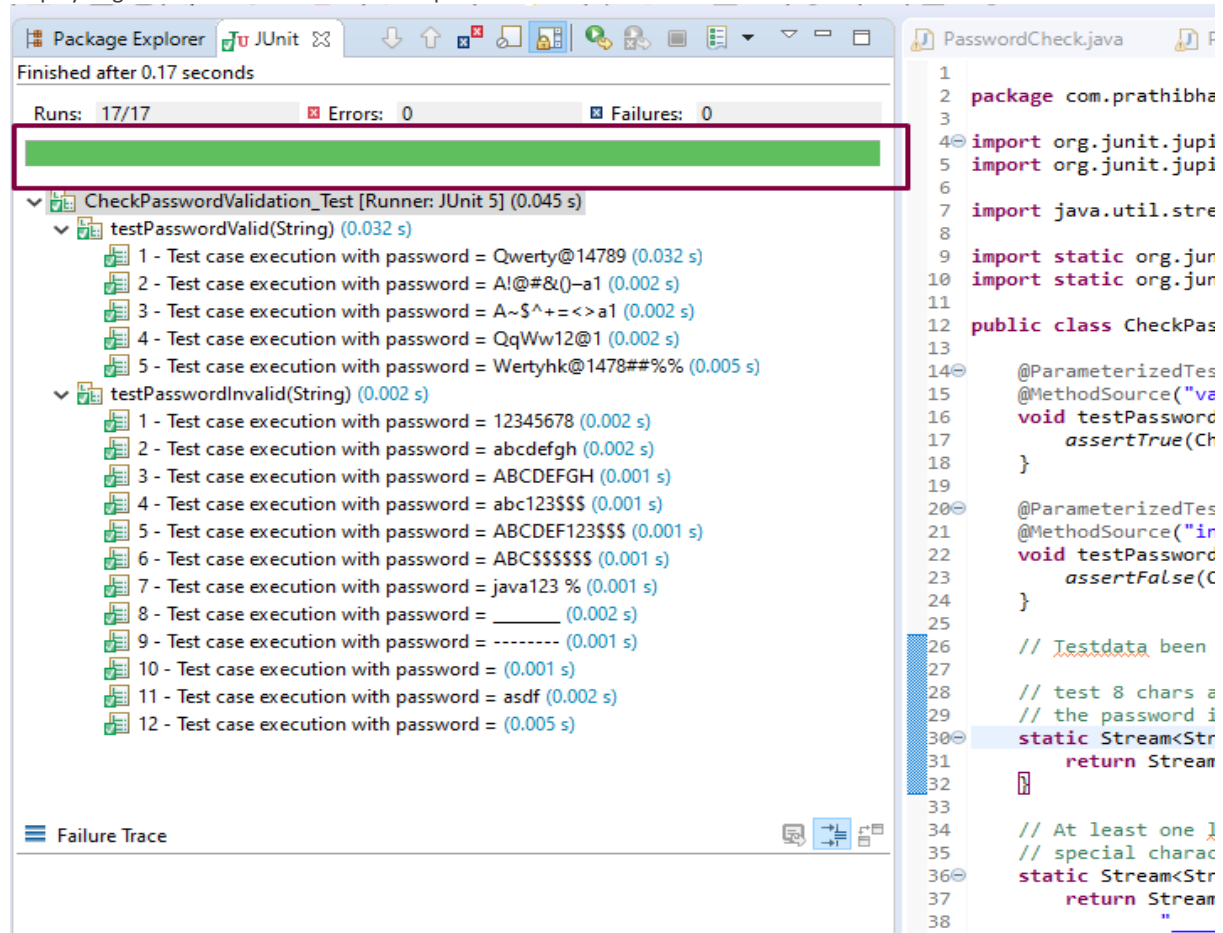Displays a green bar if all the test cases are passed.

Fig.4

Just above the timestamp on the JUnit tab, you see different icons: The first icon shows 'Next Failed Test', the second icon shows 'Previous Failed Test', and the third icon with a blue and red cross helps you to filter out only failed tests. The icon next to this is to filter only the test cases that were skipped during execution.
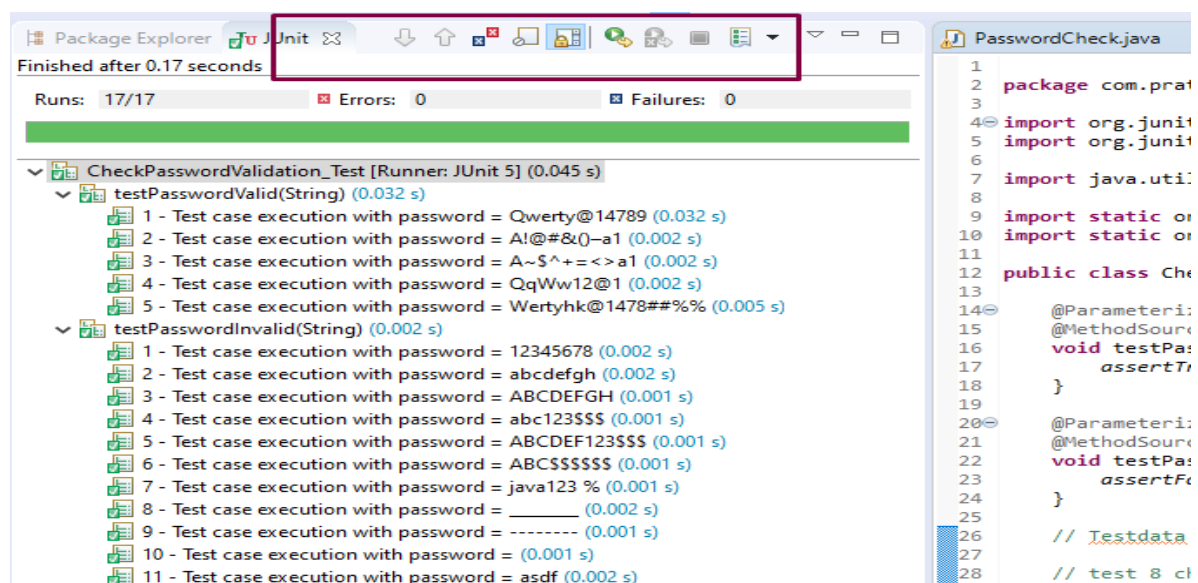

Fig.5

The Failures count now shows 1, with a red bar implying that the testcase has failed. Given below is a screenshot for your reference.
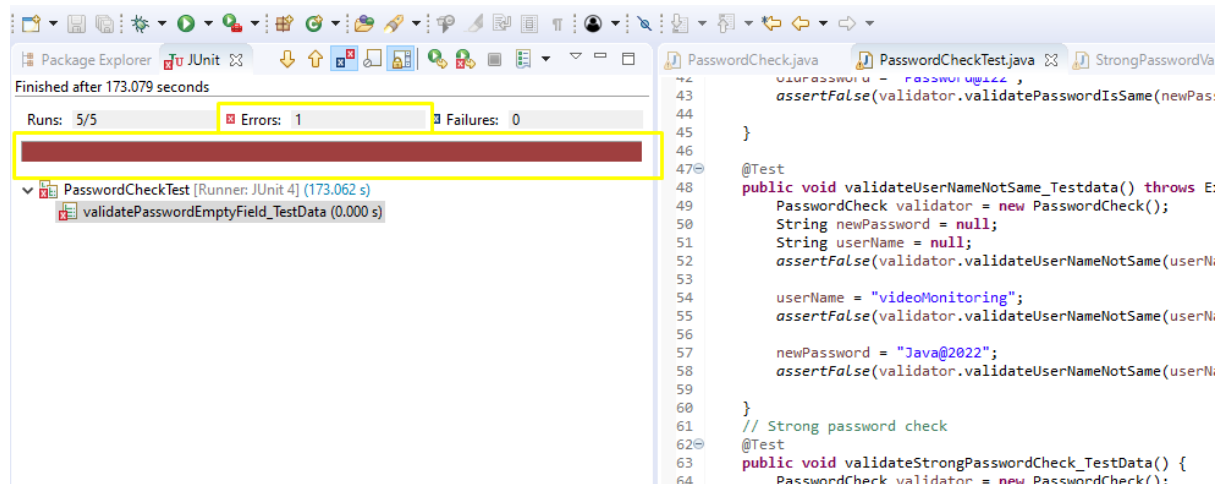

Fig.6

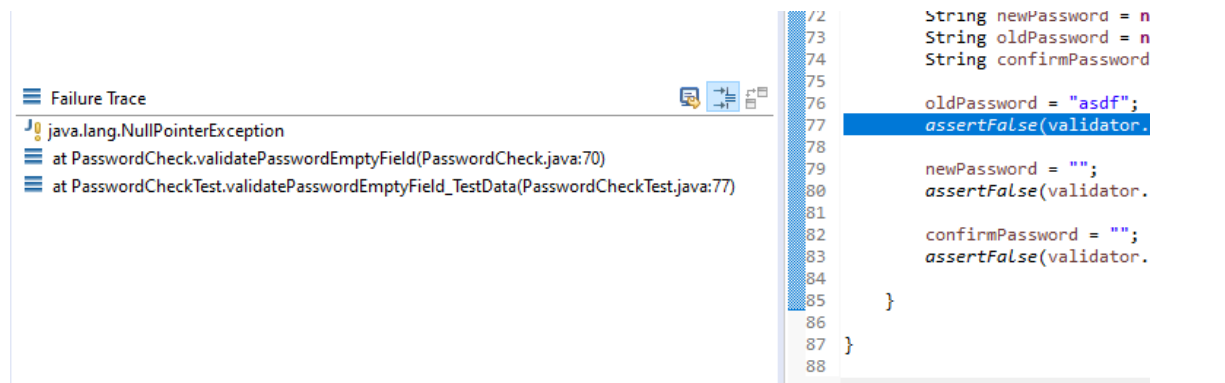'**Failure Trace**' at the bottom tab that shows the reason why the testcase failed.
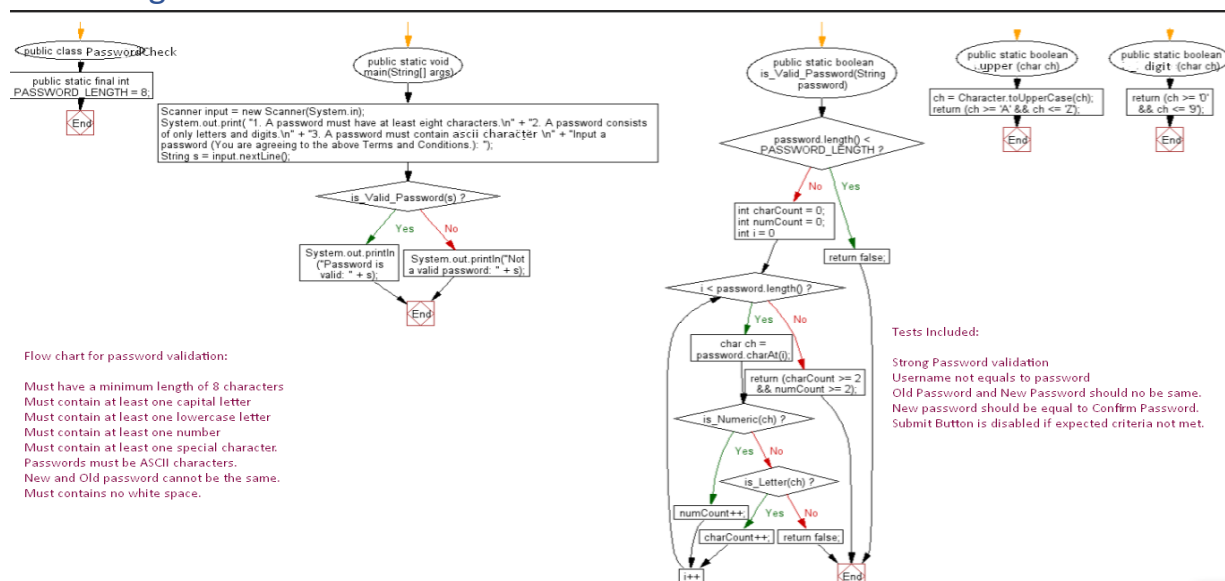

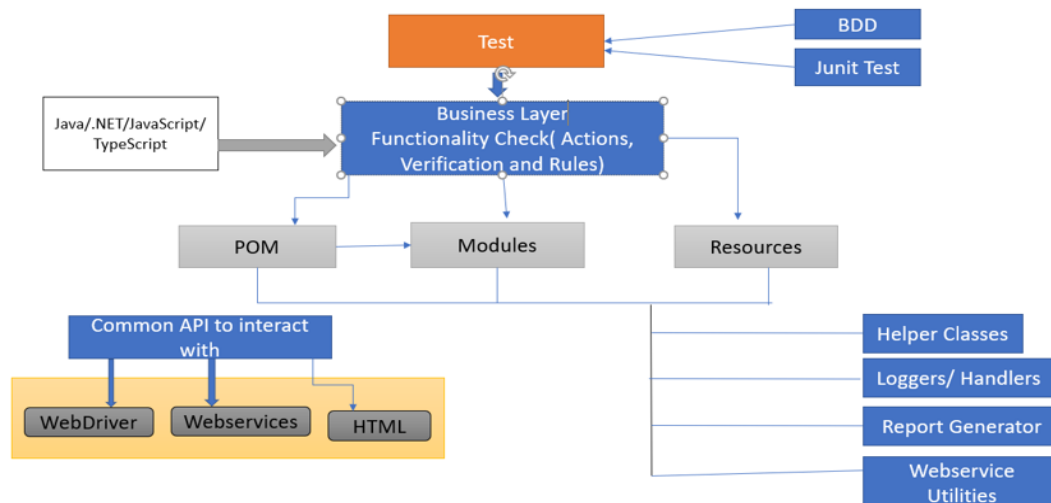**Fig.7**
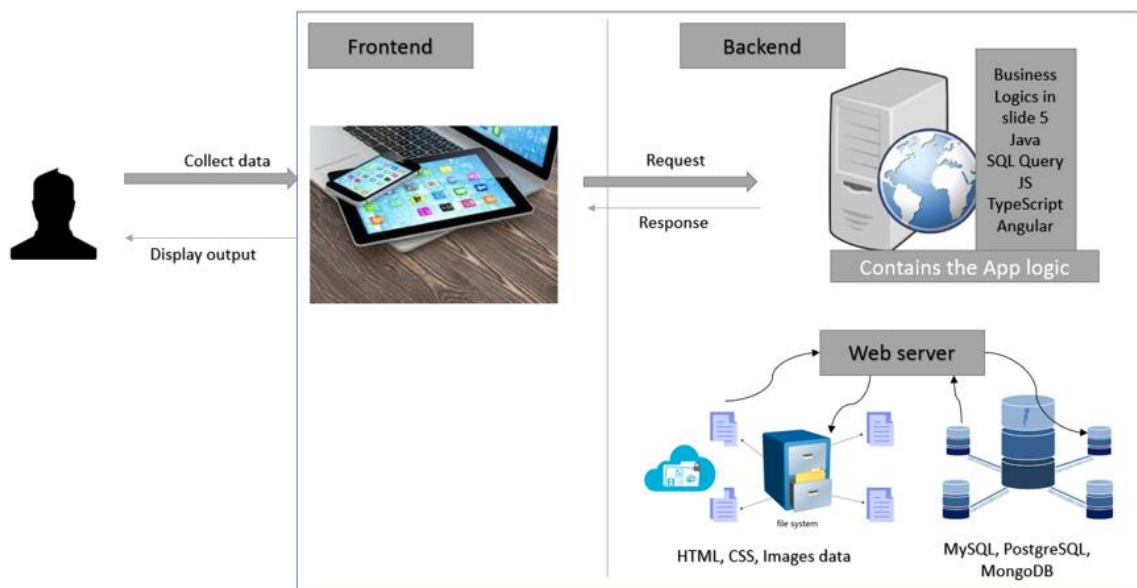
Block Diagram

Fig.8

Fig.9

Fig.10

Assumptions

Validation Testing:

Before you start testing the passwords, you should ask what specific rules of validation have been integrated by the developers on the project. Report bug if any of the criteria is not met.

Inevitability To fill the field

For example, check the fields of the "Password" and "Confirm password". Make sure that these fields must accept the same value.

Hiding the Password

Data in the Password field should be displayed on a screen in a form of dots or stars. At the right of the password entry field, a special bar or pop-up should be displayed. It is responsible for switching on/off the function of password displaying and its work should be also checked for validity.

Old Password

Password entry field can be present not only on the sign-up page but also on a web page where the user data can be edited. To keep a high level of security, it's also important that every new password of a client is different from the previous one and is unique within one account

Cross browser Testing

User should always validate the work of password entering on various browsers since Safari browser, Edge, Chrome, and Firefox, this will help the developer if he misses any special icons can easily overlap the nearby icon which is responsible for displaying/hiding a password.

Additional Requirements.

As Already mentioned, in the AC ;When the user enters a password and the criteria is not met, the following will happen:

- Message will appear telling the user that the password entered does not meet the minimum requirements
- The message shall describe the minimum requirements
- The Submit button is disabled.

The minimum requirements of 8-character Length and at least one special character, a digit, ASCII (an upper case, a lower case)and no white space, this makes a strong password.

When the user enters a password and the criteria is met, the following will happen:

- The Submit button is enabled.
- The password is stored and used to login in with.

All the test coverage been shown below refer **Fig.11**:

- Combine partial unrelated words together
- Combine word with number
- Substitute Word with number and symbol randomly
- Mix and match Word and number together randomly
- Mix meaningless Word, number, and symbol randomly, and at least 15 lengths

JUnit 5 Testing introduces new functionalities like dependency testing, grouping concept to make testing easier and more powerful and designed to cover all types of tests: unit, integration, functional, etc.
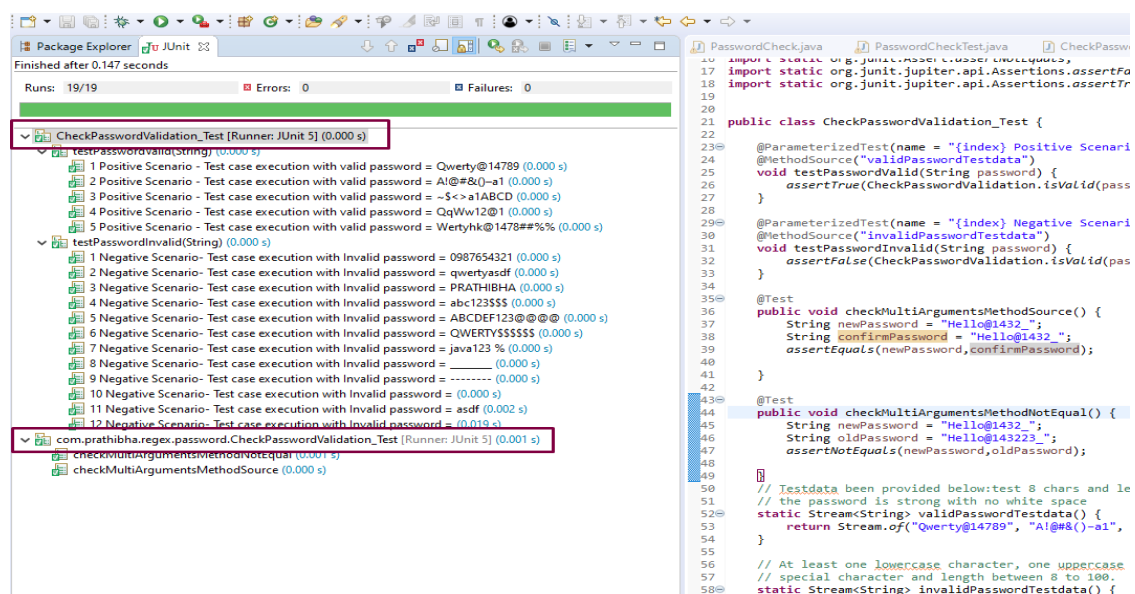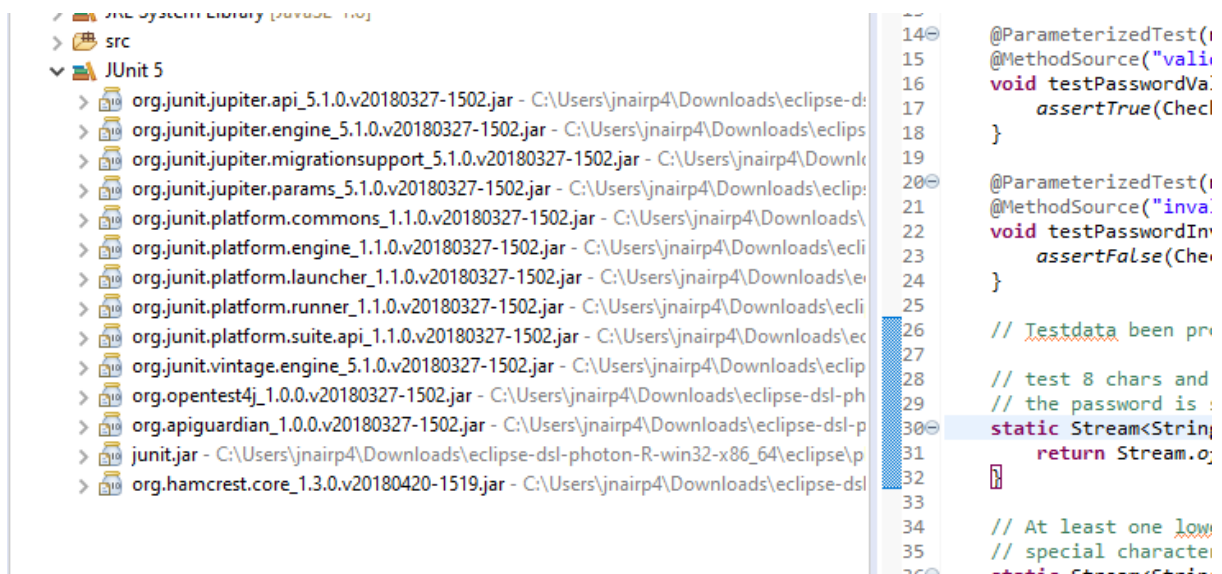

Fig. 11

Fig.12

Summary

Tests should be written before the coding. Good tests state how to best design the system for its intended use. They also prevent tendencies to over-build the software. When all the tests pass, the results can be captured. Whenever a customer reports a bug, first write the necessary unit test(s) to expose the bug(s) and fix them. This makes it almost impossible for the same bug to reappear or reproduce later while testing.

Run unit test as often as possible, ideally every time the code is changed. Run all your acceptance, integration, stress, and unit tests at least once per day based on builds.

References

https://junit.org/junit5/docs/5.0.1/api/org/junit/jupiter/api/Assertions.html

https://security.intuit.com/security-tips#use-best-practices-for-passwords,-usernames,-and-more

https://en.wikipedia.org/wiki/ASCII