Part 1 - Traceability Matrix:

Project	UpdatePwd	**Requirement Traceability Matrix**
Project Manager		
Contact	****7898**5	
Date	08/11/2024	

Requirement Id	**Feature**	**Requirement Description**	**TC ID**	**Test scenarios**	**Test Data**	**Status**	**Defect ID**	**Comments**
TR_001	Update account password	New and old password cannot be the same	TC_001	Verify update password doesn't happen when new and old password are same	**Valid**: Old: Test@1234 New: Test@12345 **Invalid:** Old: Test@1234 New: Test@1234			
TR_002		Password must be at least 8 characters long	TC_002	Verify system accepts new password only when it is at least 8 characters long	**Valid:** Test@123 (exact 8 characters) Test@12345 (more than 8 chars) **Invalid:** Test@1 (less than 8 chars)			
TR_003		Password must contain at least one number and one special character	TC_003	Verify system accepts new password only when it contains at least one number and one special characters	**Valid:** Testpas@1(at least one number and one special char) Test@12345! (more than one number and one special char) **Invalid:** Testpas1 (No special character) Testpas@(No number)			
TR_004		Password must be ASCII character	TC_004	Verify that the system only accepts ASCII characters in password	**Valid:** Password!123 (ASCII characters) **Invalid:** Pässwørd!123 (Non-ASCII characters)			

Test Scripts:

TC001: Verify that user is not able to update when new and old password are same

Steps:

1. On the password update page, enter the current password in the ‘Existing Password’ field.
2. Enter the same current password in the ‘New Password’ field.
3. Again, enter the same current password in the ‘New Password’ field.
4. Click on submit

Expected Result: Error message should be seen as the new password cannot be the same as old password.

TC002: Verify that error message is displayed when password is having lesser than 8 characters

Steps:

1. On the update password page, enter the current password in the ‘Existing Password’ field.
2. Enter a password less than 8 characters in the ‘New Password’ field.
3. Enter the same new password in ‘New Password’ field
4. Click on submit.

Expected Result: Error message must be seen notifying the user that password must be at least 8 characters long.

TC003: Verify that the error message is displayed when the password doesn’t contain at least one number and one special character.

Steps:

1. On the update password page, enter the current password in the ‘Existing Password’ field.
2. Enter password that does not contain any number or special character in the ‘New Password’ field.
3. Enter the same new password in ‘New Password’ field
4. Click on submit.

Expected Result: Error message must be seen notifying the user that password should contain at least one number and one special character.

TC004: Verify that password should contain only ASCII characters

Steps:

1. On the update password page, enter the current password in the ‘Existing Password’ field.
2. Enter password that contains non-ASCII character in ‘New Password’ field.
3. Enter the same new password in ‘New Password’ field
4. Click on submit.

Expected Result: Error message must be seen notifying the user that new password accepts only ASCII characters.

TC005: Verify user can update password with a valid new password

Steps:

1. On the password update page, enter the current password in the ‘Existing Password’ field.
2. Enter a valid new password which meets all requirements (at least 8 characters long, contains at least one number and one special character, and contains only ASCII characters) in the ‘New Password’ field.
3. Again, enter the above password in the ‘New Password’ field.
4. Click on submit

Expected Result: User should see successful message and password must get updated.

TC005: Verify password update doesn’t happen when both ‘New Password’ fields are not the same password.

Steps:

1. On the password update page, enter the current password in the ‘Existing Password’ field.
2. Enter a valid new password which meets all requirements (at least 8 characters long, contains at least one number and one special character, and contains only ASCII characters) in the ‘New Password’ field.
3. Again, enter any other valid password in the ‘New Password’ field.
4. Click on submit

Expected Result: Error message must be seen notifying the user that the new password does not match.

TC006: Verify cancel button

Steps:

1. On the password update page, enter the current password in the ‘Existing Password’ field.
2. Enter a valid new password which meets all requirements (at least 8 characters long, contains at least one number and one special character, and contains only ASCII characters) in the ‘New Password’ field.
3. Again, enter the above password in the ‘New Password’ field.
4. Click on cancel

Expected Result: Password must not get updated, and flow should get cancelled.

Questions:

1. **Did you make any assumptions, if so, what were they?**
   - User is on the update password page.
   - Error message will be displayed when old and new passwords are the same.
   - Error message will be displayed when the new password contains less than 8 characters.
   - Error message will be displayed when the new password doesn’t contain at least one number and one special character.
   - Error message will be displayed when the password contains non-ASCII characters.
   - Passwords in both ‘New Password’ fields should be the same.
   - Cancel button will abort the flow.

2. **Were any requirements missed? If so, which and how would you test them?**
   Requirement was missed related to the error messages which should come when a user enters invalid password.
   What should be the error message when the old and new password don't match.
   What should happen when both ‘New Passwords’ fields don't contain the same password.
   What should be the successful message when the password gets updated.
   What should the cancel button do?
   What should be the maximum length of the new password?
   For testing above, I will set up a meeting with the product owner or BA to discuss it and get the requirements.

3. **What tests would you carry forward into the regression suite? Why?**

   I would add below tests to regression suite:

   - Verify users can update their password when it is valid.

   - Verify passwords which are not 8 characters long are not accepted.

   - Verify a password which does not contain at least one number, and one special character is not accepted.

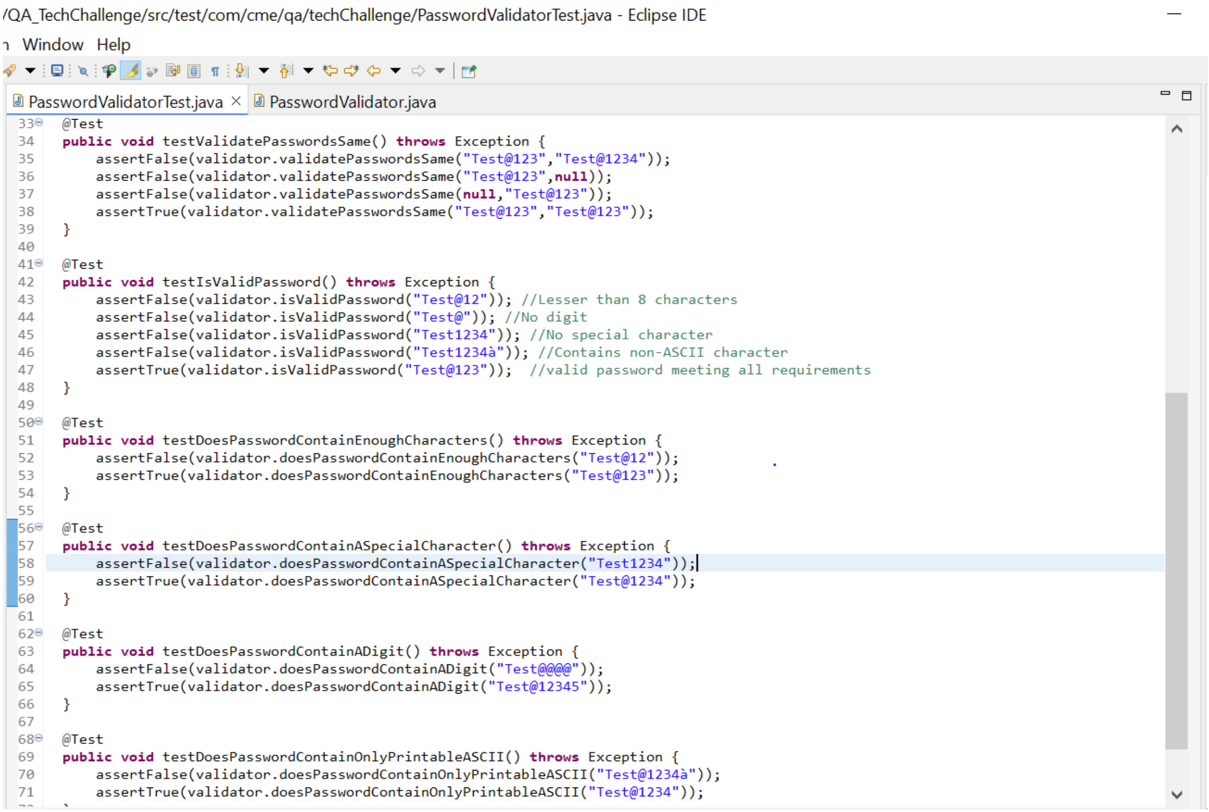
   - Verify password accepts only ASCII characters.

   I will carry above tests while regression as it’s the core functionality of password update feature and it should work as expected and it meets the specified requirements.

4. **What test would you automate? Why?**
   I would automate all regression tests to ensure the password update feature is functional and compliant, hoping it would be a stable feature which is unlikely to change frequently.

PART 2

Unit test added in PasswordValidatorTest.java. Full code is also uploaded.


/QA_TechChallenge/src/test/com/cme/qa/techChallenge/PasswordValidatorTest.java - Eclipse IDE
Window
Help
PasswordValidatorTest.java
PasswordValidator.java

```java
@Test
public void testValidatePasswordsSame() throws Exception {
    assertFalse(validator.validatePasswordsSame("Test@123","Test@1234"));
    assertFalse(validator.validatePasswordsSame("Test@123",null));
    assertFalse(validator.validatePasswordsSame(null,"Test@123"));
    assertTrue(validator.validatePasswordsSame("Test@123","Test@123"));
}

@Test
public void testIsValidPassword() throws Exception {
    assertFalse(validator.isValidPassword("Test@12")); //Lesser than 8 characters
    assertFalse(validator.isValidPassword("Test@")); //No digit
    assertFalse(validator.isValidPassword("Test1234")); //No special character
    assertFalse(validator.isValidPassword("Test1234à")); //Contains non-ASCII character
    assertTrue(validator.isValidPassword("Test@123"));  //valid password meeting all requirements
}

@Test
public void testDoesPasswordContainEnoughCharacters() throws Exception {
    assertFalse(validator.doesPasswordContainEnoughCharacters("Test@12"));
    assertTrue(validator.doesPasswordContainEnoughCharacters("Test@123"));
}

@Test
public void testDoesPasswordContainASpecialCharacter() throws Exception {
    assertFalse(validator.doesPasswordContainASpecialCharacter("Test1234"));
    assertTrue(validator.doesPasswordContainASpecialCharacter("Test@1234"));
}

@Test
public void testDoesPasswordContainADigit() throws Exception {
    assertFalse(validator.doesPasswordContainADigit("Test@@@@"));
    assertTrue(validator.doesPasswordContainADigit("Test@12345"));
}

@Test
public void testDoesPasswordContainOnlyPrintableASCII() throws Exception {
    assertFalse(validator.doesPasswordContainOnlyPrintableASCII("Test@1234à"));
    assertTrue(validator.doesPasswordContainOnlyPrintableASCII("Test@1234"));
```

—---

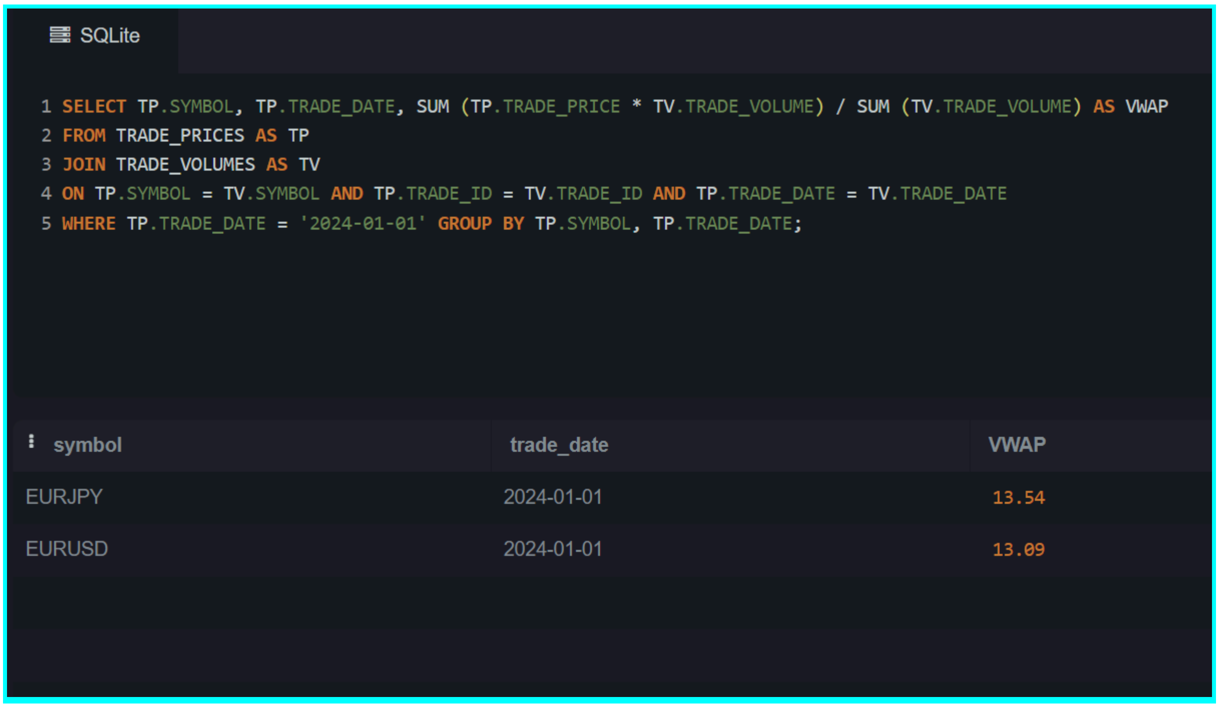
PART 3

SQL query to find the VWAP value for all symbols for the date "2024-01-01"

```
SELECT tp.symbol,tp.trade_date,SUM(tp.trade_price * tv.trade_volume) / SUM(tv.trade_volume) AS vwap
FROM trade_prices AS tp
JOIN
    trade_volumes AS tv
ON
    tp.symbol = tv.symbol
    AND tp.trade_id = tv.trade_id
    AND tp.trade_date = tv.trade_date
WHERE
    tp.trade_date = '2024-01-01'
GROUP BY
    tp.symbol, tp.trade_date;
```

Result:

Trade_date	**Symbol**	**Vwap**
01/01/2024	EURJPY	13.54
01/01/2024	EURUSD	13.09



```
SELECT TP.SYMBOL, TP.TRADE_DATE, SUM (TP.TRADE_PRICE * TV.TRADE_VOLUME) / SUM (TV.TRADE_VOLUME) AS VWAP
FROM TRADE_PRICES AS TP
JOIN TRADE_VOLUMES AS TV
ON TP.SYMBOL = TV.SYMBOL AND TP.TRADE_ID = TV.TRADE_ID AND TP.TRADE_DATE = TV.TRADE_DATE
WHERE TP.TRADE_DATE = '2024-01-01' GROUP BY TP.SYMBOL, TP.TRADE_DATE;
```

symbol	trade_date	VWAP
EURJPY	2024-01-01	13.54
EURUSD	2024-01-01	13.09