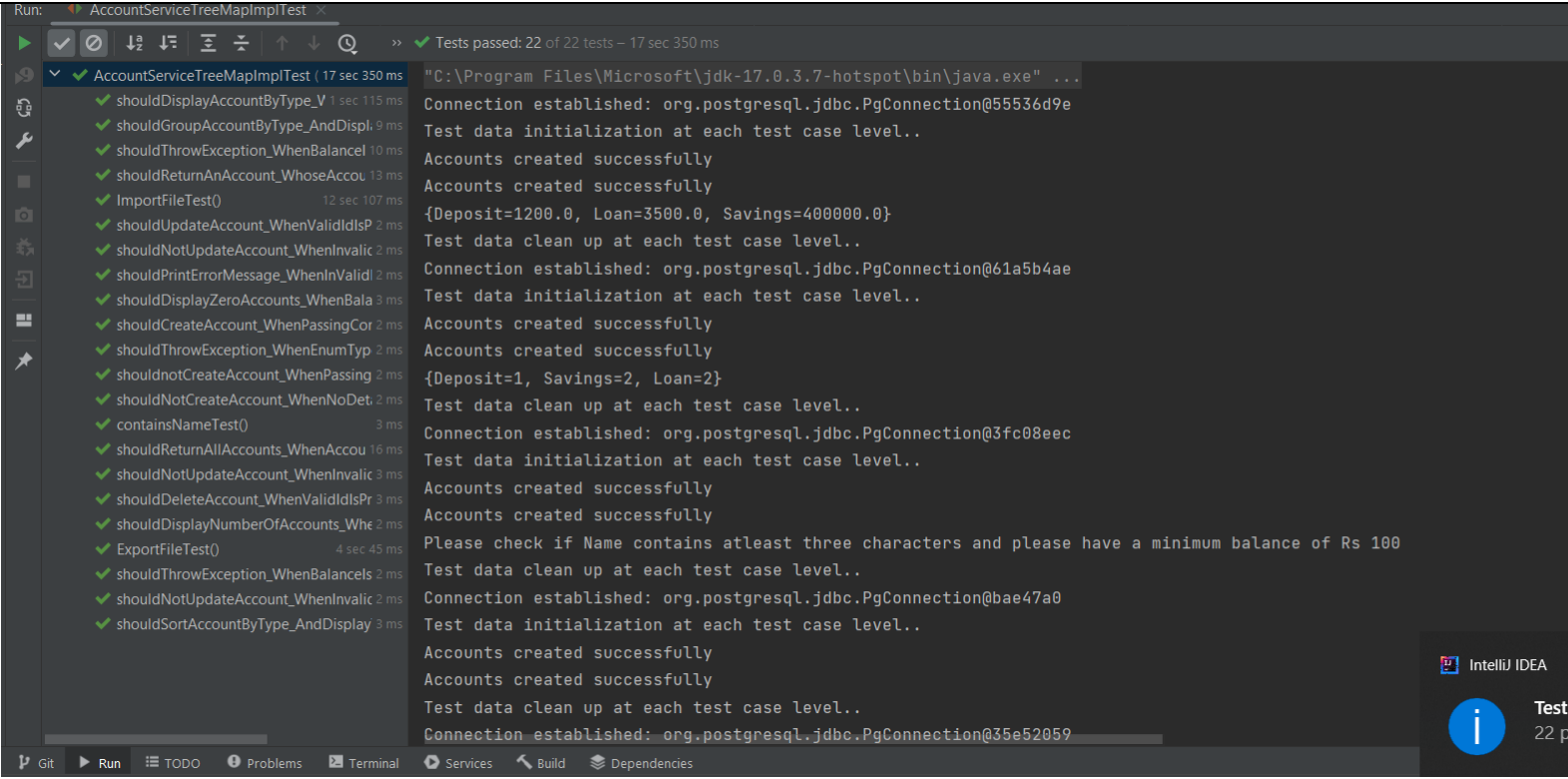


1. Test cases scenarios

# 1	Total numb er of test cases passe d	 <p>The screenshot displays the IntelliJ IDEA Run window for the test class <code>AccountServiceTreeMapImplTest</code>. The test suite consists of 22 tests, all of which passed successfully within a total execution time of 17 seconds and 350 milliseconds. The tests are listed on the left, each preceded by a green checkmark and followed by its duration. The right pane shows the output of these tests, which includes database connection logs, data initialization, successful account creation, and cleanup operations. The bottom status bar indicates the current tool is IntelliJ IDEA.</p> <p>Run: AccountServiceTreeMapImplTest</p> <p>Tests passed: 22 of 22 tests - 17 sec 350 ms</p> <p>AccountServiceTreeMapImplTest (17 sec 350 ms)</p> <ul style="list-style-type: none">✓ shouldDisplayAccountByType_V 1 sec 115 ms✓ shouldGroupAccountByType_AndDisplay 9 ms✓ shouldThrowException_WhenBalance 10 ms✓ shouldReturnAnAccount_WhoseAccou 13 ms✓ ImportFileTest() 12 sec 107 ms✓ shouldUpdateAccount_WhenValidIdIsP 2 ms✓ shouldNotUpdateAccount_WhenInvalid 2 ms✓ shouldPrintErrorMessage_WhenInvalid 2 ms✓ shouldDisplayZeroAccounts_WhenBala 3 ms✓ shouldCreateAccount_WhenPassingCor 2 ms✓ shouldThrowException_WhenEnumTyp 2 ms✓ shouldnotCreateAccount_WhenPassing 2 ms✓ shouldNotCreateAccount_WhenNoDet 2 ms✓ containsNameTest() 3 ms✓ shouldReturnAllAccounts_WhenAccou 16 ms✓ shouldNotUpdateAccount_WhenInvalid 3 ms✓ shouldDeleteAccount_WhenValidIdIsPr 3 ms✓ shouldDisplayNumberOfAccounts_Wh 2 ms✓ ExportFileTest() 4 sec 45 ms✓ shouldThrowException_WhenBalance 2 ms✓ shouldNotUpdateAccount_WhenInvalid 2 ms✓ shouldSortAccountByType_AndDisplay 3 ms <p>Output:</p> <pre>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" ... Connection established: org.postgresql.jdbc.PgConnection@55536d9e Test data initialization at each test case level.. Accounts created successfully Accounts created successfully {Deposit=1200.0, Loan=3500.0, Savings=400000.0} Test data clean up at each test case level.. Connection established: org.postgresql.jdbc.PgConnection@61a5b4ae Test data initialization at each test case level.. Accounts created successfully Accounts created successfully {Deposit=1, Savings=2, Loan=2} Test data clean up at each test case level.. Connection established: org.postgresql.jdbc.PgConnection@3fc08eec Test data initialization at each test case level.. Accounts created successfully Accounts created successfully Please check if Name contains atleast three characters and please have a minimum balance of Rs 100 Test data clean up at each test case level.. Connection established: org.postgresql.jdbc.PgConnection@bae47a0 Test data initialization at each test case level.. Accounts created successfully Accounts created successfully Test data clean up at each test case level.. Connection established: org.postgresql.jdbc.PgConnection@35e52059</pre>
--------	---	--

BANK OF BENGALURU: MOCKITO

Poornima Chandrasekaran

# 2	Before Each and After Each	<pre>Chandrasekaran @BeforeEach public void setup() throws AccountException, SQLException { System.out.println("Test data initialization at each test case level.."); // Initialize Test data Account account1 = new Account(id: 11, name: "Pankaj Kumar", type: "Loan", balance: 2000.0, active: true, LocalDate.of(year: 2022, month: 02, dayOfMonth: 27), accService.createAccount(account1); accounts.add(account1); Account account2 = new Account(id: 12, name: "Pradeep Kumar Pandey", type: "Deposit", balance: 1200.0, active: true, LocalDate.of(year: 2022, month: 05, dayO accService.createAccount(account2); accounts.add(account2); } Chandrasekaran @AfterEach public void cleanup() { System.out.println("Test data clean up at each test case level.."); }</pre>
# 3	Case #1	<pre>shouldCreateAccount_WhenPassingCorrectDetails @Test public void shouldCreateAccount_WhenPassingCorrectDetails() throws SQLException { Account account = new Account(id: 3, name: "Vihaan", type: "Savings", balance: 10000, active: true, LocalDate.now(Mockito.when(accountDao.createAccount(account)).thenReturn(t: true); try { assertEquals(expected: true, accService.createAccount(account)); } catch (com.lowes.bankingapp.exception.AccountException e) { throw new RuntimeException(e); } }</pre>

BANK OF BENGALURU: MOCKITO

Poornima Chandrasekaran

		<div style="background-color: #f0f0f0; padding: 5px;"> <p>✓ AccountServiceTreeMapImplTest (com.lowes.bank) 1 sec 222 ms</p> <ul style="list-style-type: none"> ✓ shouldCreateAccount_WhenPassingCorrectDe 1 sec 222 ms <pre>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin Connection established: org.postgresql.jdbc.PgConnec Test data initialization at each test case level.. Accounts created successfully Accounts created successfully Accounts created successfully Test data clean up at each test case level.. Process finished with exit code 0"</pre> </div>
#4	Case #2	<div style="background-color: #e6e6fa; padding: 5px;">shouldnotCreateAccount_WhenPassingInCorrectName</div> <div style="background-color: #f0f0f0; padding: 5px;"> <pre>@Test public void shouldnCreateAccount_WhenPassingInCorrectName() { Account account1 = new Account(id: 4, name: "Jo", type: "Savings", balance: 10000, active: true, LocalDate.no Throwable ex = assertThrows(Exception.class, () -> accService.createAccount(account1)); assertEquals(expected: "Account Creation Failed, Please Create Another Account", ex.getMessage()); }</pre> </div> <div style="background-color: #f0f0f0; padding: 5px;"> <p>✓ AccountServiceTreeMapImplTest (com.lowes.banking) 702 ms</p> <ul style="list-style-type: none"> ✓ shouldnotCreateAccount_WhenPassingInCorrectNi 702 ms <pre>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" ... Connection established: org.postgresql.jdbc.PgConnection@4d910fd6 Test data initialization at each test case level.. Accounts created successfully Accounts created successfully Please check if Name contains atleast three characters and please have a min: Test data clean up at each test case level.. Process finished with exit code 0"</pre> </div>

#5	Case 3	<div data-bbox="510 240 2032 272">shouldNotCreateAccount_WhenNoDetailsArePassed</div> <pre data-bbox="510 321 2032 573"> @Test public void shouldNotCreateAccount_WhenNoDetailsArePassed() { Account account = new Account(); Throwable ex = assertThrows(Exception.class, () -> accService.createAccount(account)); assertEquals("expected: 'Account Creation Failed, Please Create Another Account'", ex.getMessage()); } </pre> <div data-bbox="510 654 2032 1052"> <div> <div> ✓ AccountServiceTreeMapImplTest (com.lowes.bankinga 661 ms) </div> <div> ✓ shouldNotCreateAccount_WhenNoDetailsArePasse 661 ms </div> </div> <div> "C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\ Connection established: org.postgresql.jdbc.PgCon Test data initialization at each test case level. Accounts created successfully Accounts created successfully No Account details Test data clean up at each test case level.. Process finished with exit code 0 </div> </div>
#6	Case 4	<div data-bbox="510 1101 2032 1133">shouldThrowException_WhenBalanceIsLessThanTheRange</div>

		<pre> @Test public void shouldThrowException_WhenBalanceIsLessThanTheRange() { Account account1 = new Account(id: 4, name: "Anu", type: "Savings", balance: 10.0, active: true, LocalDate.now()); Throwable ex = assertThrows(Exception.class, () -> accService.createAccount(account1)); assertEquals(expected: "Account Creation Failed, Please Create Another Account", ex.getMessage()); } </pre> <div> <div> ✓ AccountServiceTreeMapImplTest (com.lowes.bankinga 777 ms ✓ shouldThrowException_WhenBalanceIsLessThanThi 777 ms </div> <div> "C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" ... Connection established: org.postgresql.jdbc.PgConnection@1649b0e6 Test data initialization at each test case level.. Accounts created successfully Accounts created successfully Please check if Name contains atleast three characters and please have a minimum length of 3 characters. Test data clean up at each test case level.. Process finished with exit code 0 </div> </div>
#7	Case 5	<pre> shouldThrowException_WhenBalanceIsNegative </pre> <pre> @Test public void shouldThrowException_WhenBalanceIsNegative() { Account account1 = new Account(id: 4, name: "Anu", type: "Savings", balance: -5.0, active: true, LocalDate.now()); Throwable ex = assertThrows(Exception.class, () -> accService.createAccount(account1)); assertEquals(expected: "Account Creation Failed, Please Create Another Account", ex.getMessage()); } </pre>

BANK OF BENGALURU: MOCKITO

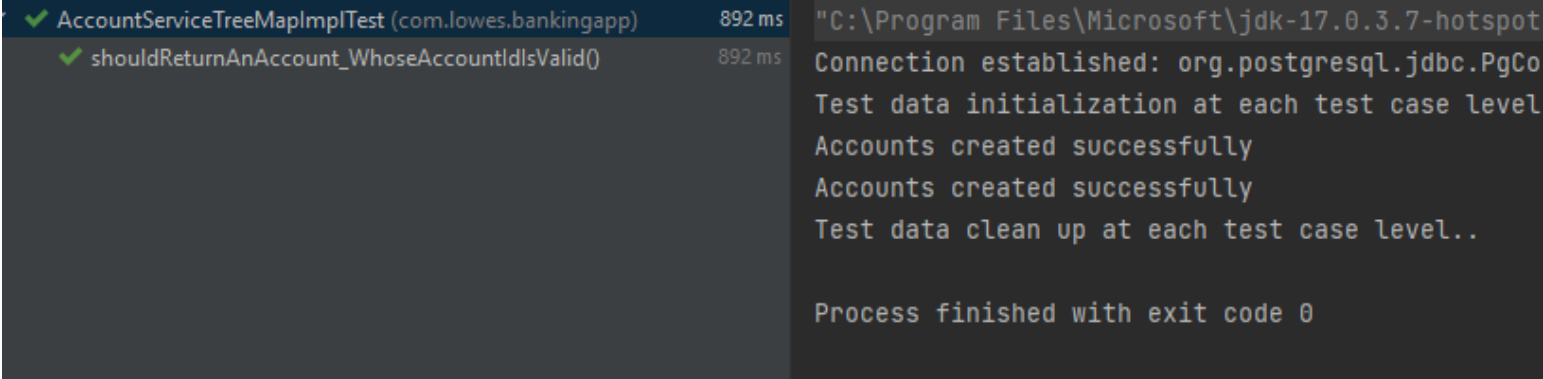
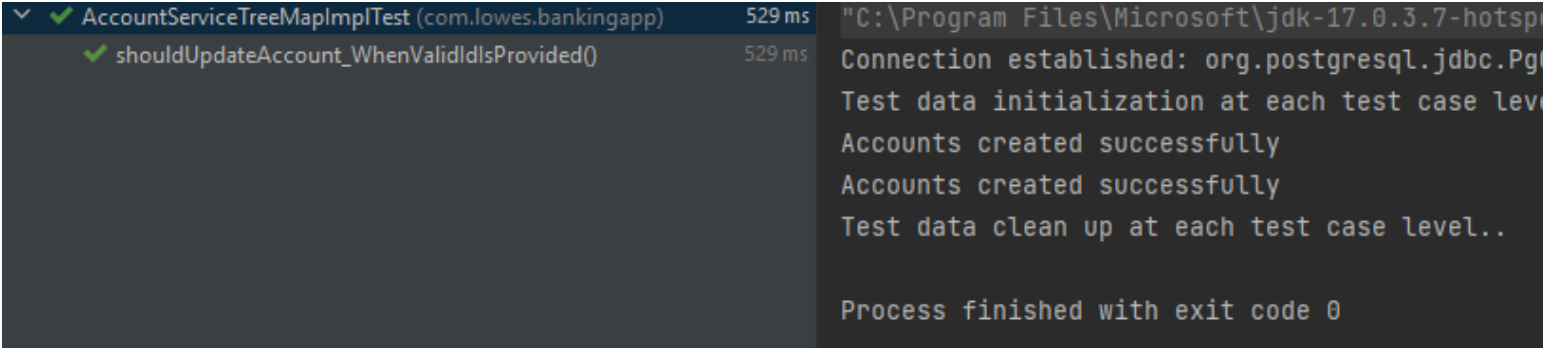
Poornima Chandrasekaran

		<div> <div> ✓ AccountServiceTreeMapImplTest (com.lowes.bank) 1 sec 349 ms ✓ shouldThrowException_WhenBalanceIsNegative 1 sec 349 ms </div> <div> "C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" ... Connection established: org.postgresql.jdbc.PgConnection@3315d2d7 Test data initialization at each test case level.. Accounts created successfully Accounts created successfully Please check if Name contains atleast three characters and please have a min. Test data clean up at each test case level.. Process finished with exit code 0 </div> </div>
#8	Case 6	<div>shouldThrowException_WhenEnumTypeIsValid</div> <div> <pre>@Test public void shouldThrowException_WhenEnumTypeIsValid() { Account account1 = new Account(id: 4, name: "Priya", type: "Personal", balance: 1000.0, active: true, LocalDate.now()); Throwable ex = assertThrows(Exception.class, () -> accService.createAccount(account1)); assertEquals(expected: "Invalid Type, Please Choose Deposit, Loan or Savings", ex.getMessage()); }</pre> </div> <div> <div> ✓ AccountServiceTreeMapImplTest (com.lowes.bankingapp) 1 sec 384 ms ✓ shouldThrowException_WhenEnumTypesInvalid() 1 sec 384 ms </div> <div> "C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" ... Connection established: org.postgresql.jdbc.PgCon Test data initialization at each test case level.. Accounts created successfully Accounts created successfully Test data clean up at each test case level.. Process finished with exit code 0 </div> </div>
#9	Case 7	<div>shouldReturnAllAccounts_WhenAccountIdIsNotMentioned</div>

		<pre> @Test public void shouldReturnAllAccounts_WhenAccountIdIsNotMentioned() throws SQLException { Mockito.when(accountDao.getAccounts()).thenReturn((List<Account>) accounts); Collection<Account> accounts = accService.getAccounts(); System.out.println(accounts); assertNotNull(accounts); } </pre>
		<pre> AccountServiceTreeMapImplTest (com.lowes.bankingapp) 1 sec 143 ms shouldReturnAllAccounts_WhenAccountIdIsNotMentioned 1 sec 143 ms </pre> <p> "C:\Program Files\Microsoft\jdk-17.0.3-hotspot\bin\java.exe" ... Connection established: org.postgresql.jdbc.PgConnection@4d910fd6 Test data initialization at each test case level.. Accounts created successfully Accounts created successfully [Account{id=11, name='Pankaj Kumar', type='Loan', balance=2000.0, active=true, created on=27-Feb-2022, updated on=2023-07-10}, Acc Test data clean up at each test case level.. Process finished with exit code 0 </p>
#1 0	Case 8	<pre> shouldReturnAnAccount_WhoseAccountIdIsValid @Test public void shouldReturnAnAccount_WhoseAccountIdIsValid() throws SQLException, AccountException { Account account = new Account(id: 3, name: "Vihaan", type: "Savings", balance: 10000, active: true, LocalDate.no Mockito.when(accountDao.getAccount(accid: 3)).thenReturn(account); assertEquals(expected: 3, accService.getAccount(id: 3).getId()); } </pre>

BANK OF BENGALURU: MOCKITO

Poornima Chandrasekaran

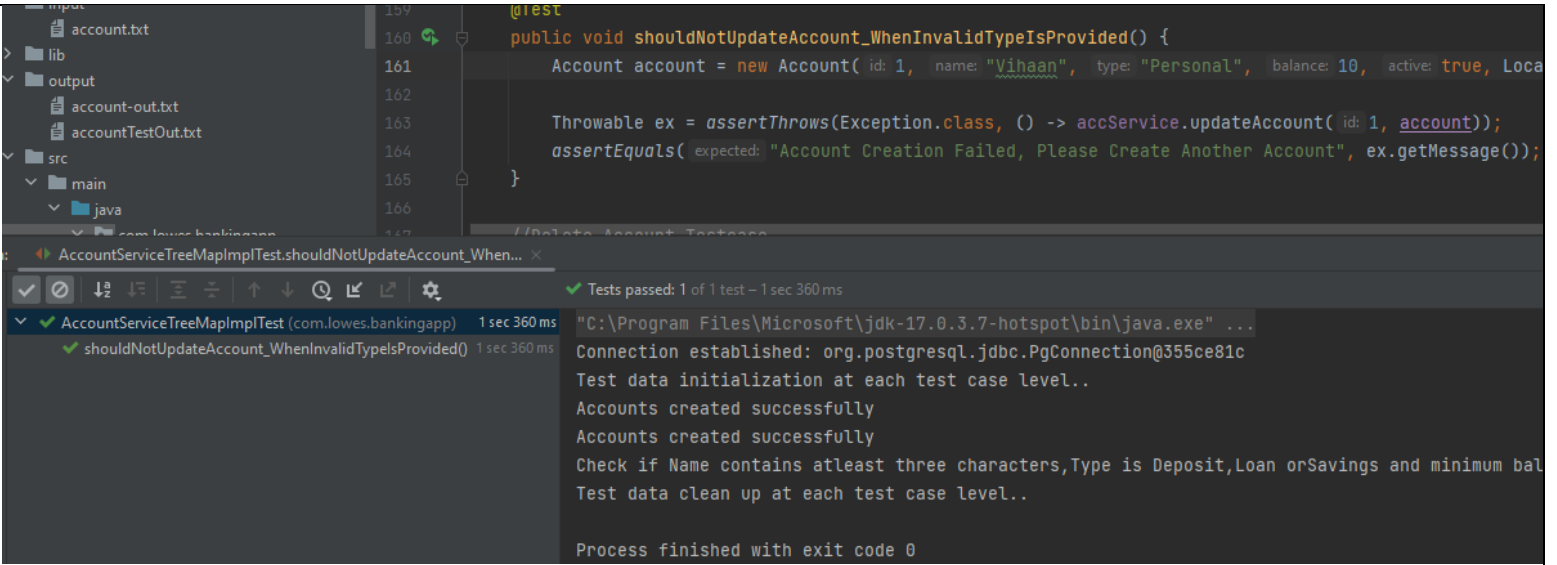
		
#1 1	Case 9	<p>shouldUpdateAccount_WhenValidIdIsProvided</p> <pre> @Test public void shouldUpdateAccount_WhenValidIdIsProvided() throws AccountException, SQLException { Account account = new Account(id: 12, name: "Vihaana", type: "Savings", balance: 10000, active: true, LocalDate.now()); Mockito.when(accountDao.updateAccount(id: 12, account)).thenReturn(true); assertEquals(expected: true, accService.updateAccount(id: 12, account)); } </pre> 
#1 2	Case 10	<p>shouldNotUpdateAccount_WhenInvalidIdIsProvided</p>

		<pre> @Test public void shouldNotUpdateAccount_WhenInvalidIdIsProvided() { Account account = new Account(id: 9, name: "Vihaan", type: "Savings", balance: 10000, active: true, LocalDate.now()); Throwable ex = assertThrows(Exception.class, () -> accService.updateAccount(id: 9, account)); assertEquals(expected: "Can't Update, no Account found", ex.getMessage()); } </pre>
		<div> <div> <div>✓ AccountServiceTreeMapImplTest (com.lowes.bankingapp)</div> <div>909 ms</div> </div> <div> <div>✓ shouldNotUpdateAccount_WhenInvalidIdsProvided()</div> <div>909 ms</div> </div> </div> <div> <p>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot"</p> <p>Connection established: org.postgresql.jdbc.PgCo</p> <p>Test data initialization at each test case level</p> <p>Accounts created successfully</p> <p>Accounts created successfully</p> <p>Test data clean up at each test case level..</p> <p>Process finished with exit code 0</p> </div>
#1 3	Case 11	shouldNotUpdateAccount_WhenInvalidName_BalanceIsProvided

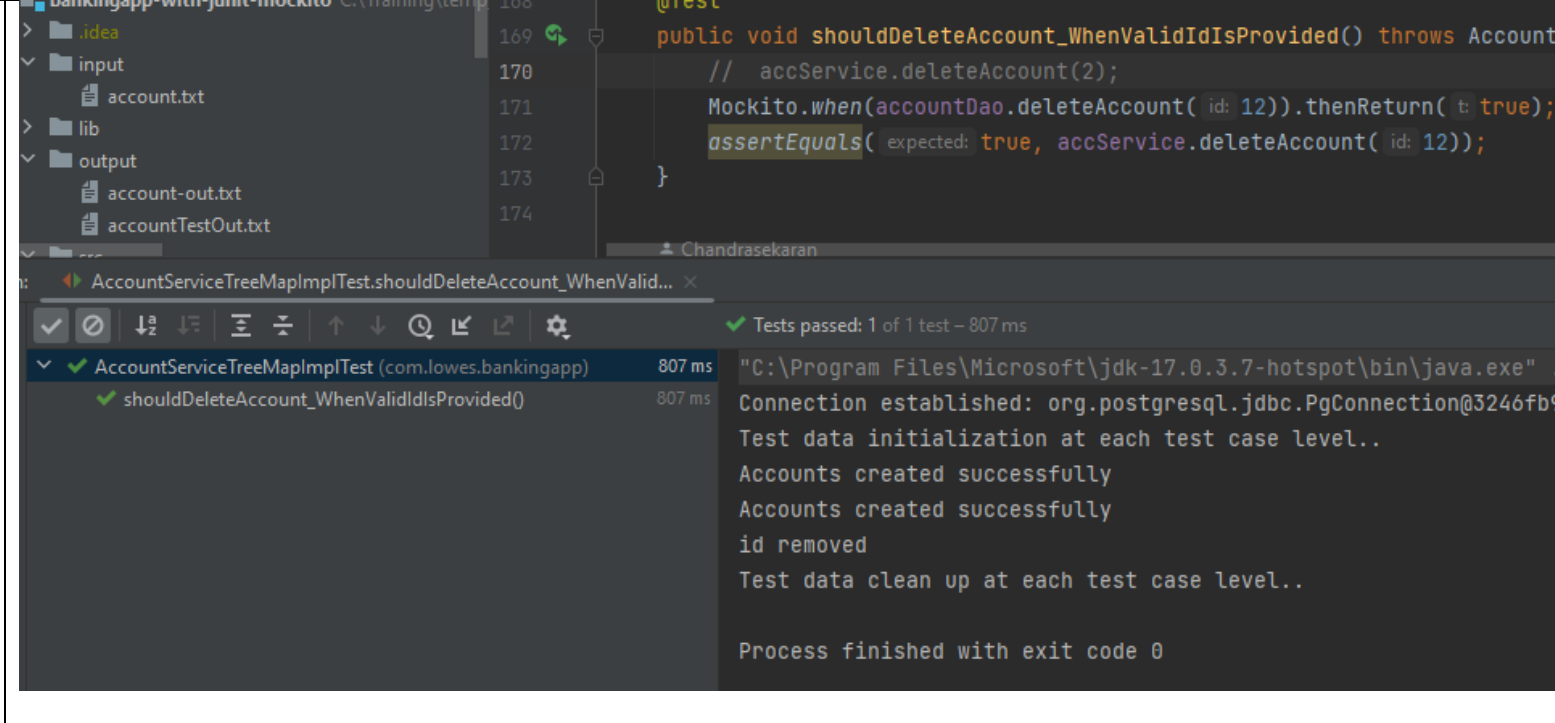
		<div><div><div><div>input</div><div>account.txt</div><div>lib</div><div>output</div><div>account-out.txt</div><div>accountTestOut.txt</div><div>src</div><div>main</div><div>java</div><div>com.lowes.bankingapp</div></div><div><div>152</div><div>153</div><div>154</div><div>155</div><div>156</div><div>157</div><div>158</div></div><div><div>Chandrasekaran</div><div>@Test</div><div>public void shouldNotUpdateAccount_WhenInvalidName_BalanceIsProvided() {</div><div>Account account = new Account(id: 1, name: "Vi", type: "Savings", balance: 10, active: true, Local</div><div>Throwable ex = assertThrows(Exception.class, () -> accService.updateAccount(id: 1, account));</div><div>assertEquals(expected: "Account Creation Failed, Please Create Another Account", ex.getMessage</div><div>}</div></div></div><div><div>AccountServiceTreeMapImplTest.shouldNotUpdateAccount_When...</div><div>Tests passed: 1 of 1 test – 653 ms</div><div>AccountServiceTreeMapImplTest (com.lowes.bankingapp) 653 ms</div><div>shouldNotUpdateAccount_WhenInvalidName_BalancelSProvie 653 ms</div><div>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" ...</div><div>Connection established: org.postgresql.jdbc.PgConnection@6a6afff2</div><div>Test data initialization at each test case level..</div><div>Accounts created successfully</div><div>Accounts created successfully</div><div>Check if Name contains atleast three characters,Type is Deposit,Loan orSavings and minimum</div><div>Test data clean up at each test case level..</div><div>Process finished with exit code 0</div></div></div>
#1 4	Case 12	shouldNotUpdateAccount_WhenInvalidTypeIsProvided

BANK OF BENGALURU: MOCKITO

Poornima Chandrasekaran

		 <pre> 159 @test 160 public void shouldNotUpdateAccount_WhenInvalidTypeIsProvided() { 161 Account account = new Account(id: 1, name: "Vihaan", type: "Personal", balance: 10, active: true, Loca 162 163 Throwable ex = assertThrows(Exception.class, () -> accService.updateAccount(id: 1, account)); 164 assertEquals(expected: "Account Creation Failed, Please Create Another Account", ex.getMessage()); 165 } 166 167 //Delete Account Testcase AccountServiceTreeMapImplTest.shouldNotUpdateAccount_When... Tests passed: 1 of 1 test - 1 sec 360 ms AccountServiceTreeMapImplTest (com.lowes.bankingapp) 1 sec 360 ms shouldNotUpdateAccount_WhenInvalidTypesProvided() 1 sec 360 ms "C:\Program Files\Microsoft\jdk-17.0.3-hotspot\bin\java.exe" ... Connection established: org.postgresql.jdbc.PgConnection@355ce81c Test data initialization at each test case level.. Accounts created successfully Accounts created successfully Check if Name contains atleast three characters,Type is Deposit,Loan orSavings and minimum bal Test data clean up at each test case level.. Process finished with exit code 0 </pre>
#1 5	Case 13	shouldDeleteAccount_WhenValidIdIsProvided

Poornima Chandrasekaran

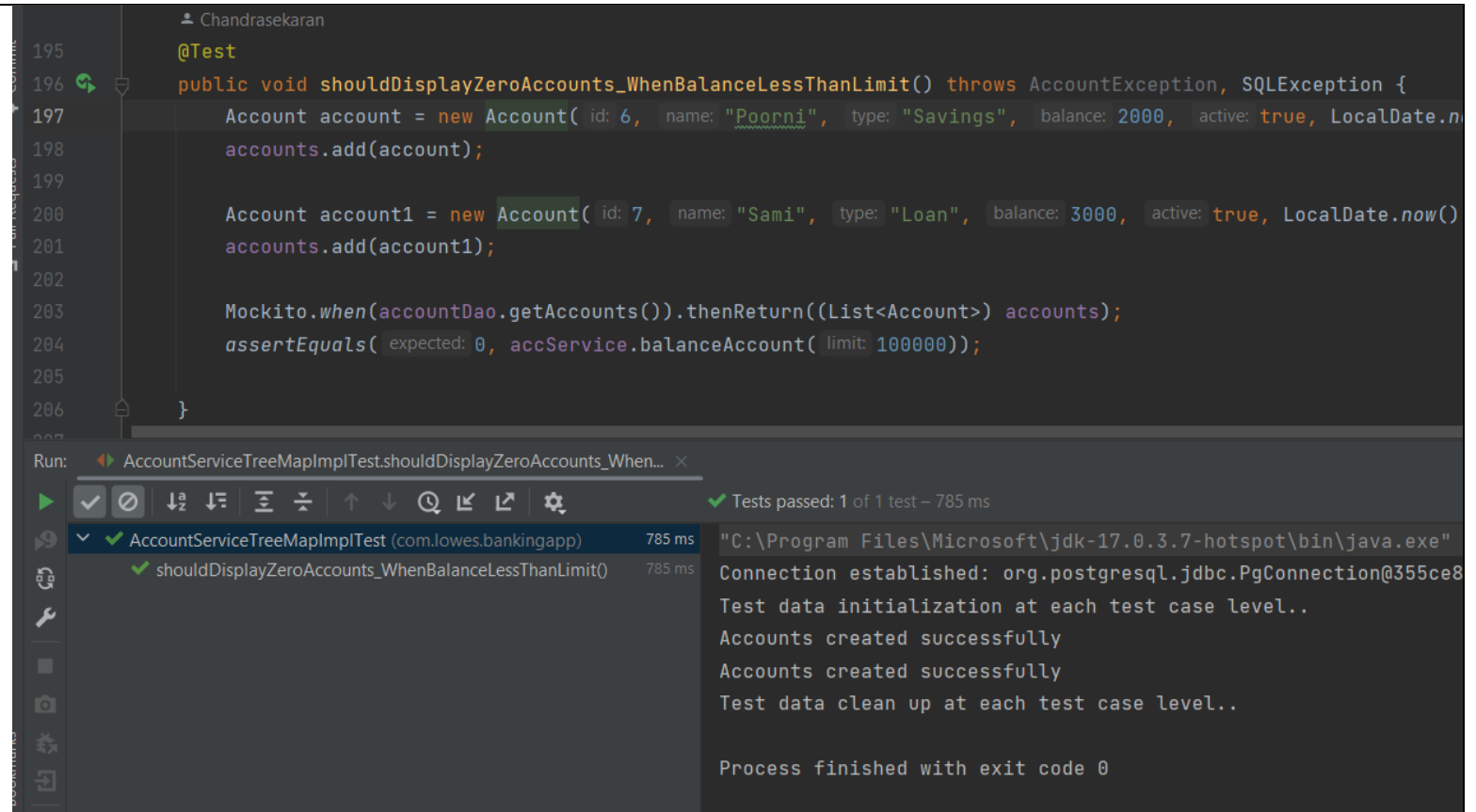
		
#16	Case 14	<u>shouldPrintErrorMessage_WhenInvalidIdIsProvided</u>

		<div><div><div>account.txt</div><div>lib</div><div>output</div><div>account-out.txt</div><div>accountTestOut.txt</div></div><div><div>176</div><div>177</div><div>178</div><div>179</div><div>180</div></div><div><div>@test</div><div>public void shouldPrintErrorMessage_WhenInvalidIdIsProvided() throws A</div><div>Throwable ex = assertThrows(Exception.class, () -> accService.dele</div><div>assertEquals(expected: "Can't Delete, No Account found", ex.getMessa</div><div>}</div></div></div> <div><div>AccountServiceTreeMapImplTest.shouldPrintErrorMessage_When...</div><div>Tests passed: 1 of 1 test – 1 sec 141 ms</div><div><div>AccountServiceTreeMapImplTest (com.lowes.bankingapp) 1 sec 141 ms</div><div>shouldPrintErrorMessage_WhenInvalidIdIsProvided() 1 sec 141 ms</div></div><div><div>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" .</div><div>Connection established: org.postgresql.jdbc.PgConnection@140c9f3</div><div>Test data initialization at each test case level..</div><div>Accounts created successfully</div><div>Accounts created successfully</div><div>Test data clean up at each test case level..</div><div>Process finished with exit code 0</div></div></div>
#1 7	Case 15	<div>shouldDisplayNumberOfAccounts_WhenBalanceGreaterThanLimit</div>

BANK OF BENGALURU: MOCKITO

Poornima Chandrasekaran

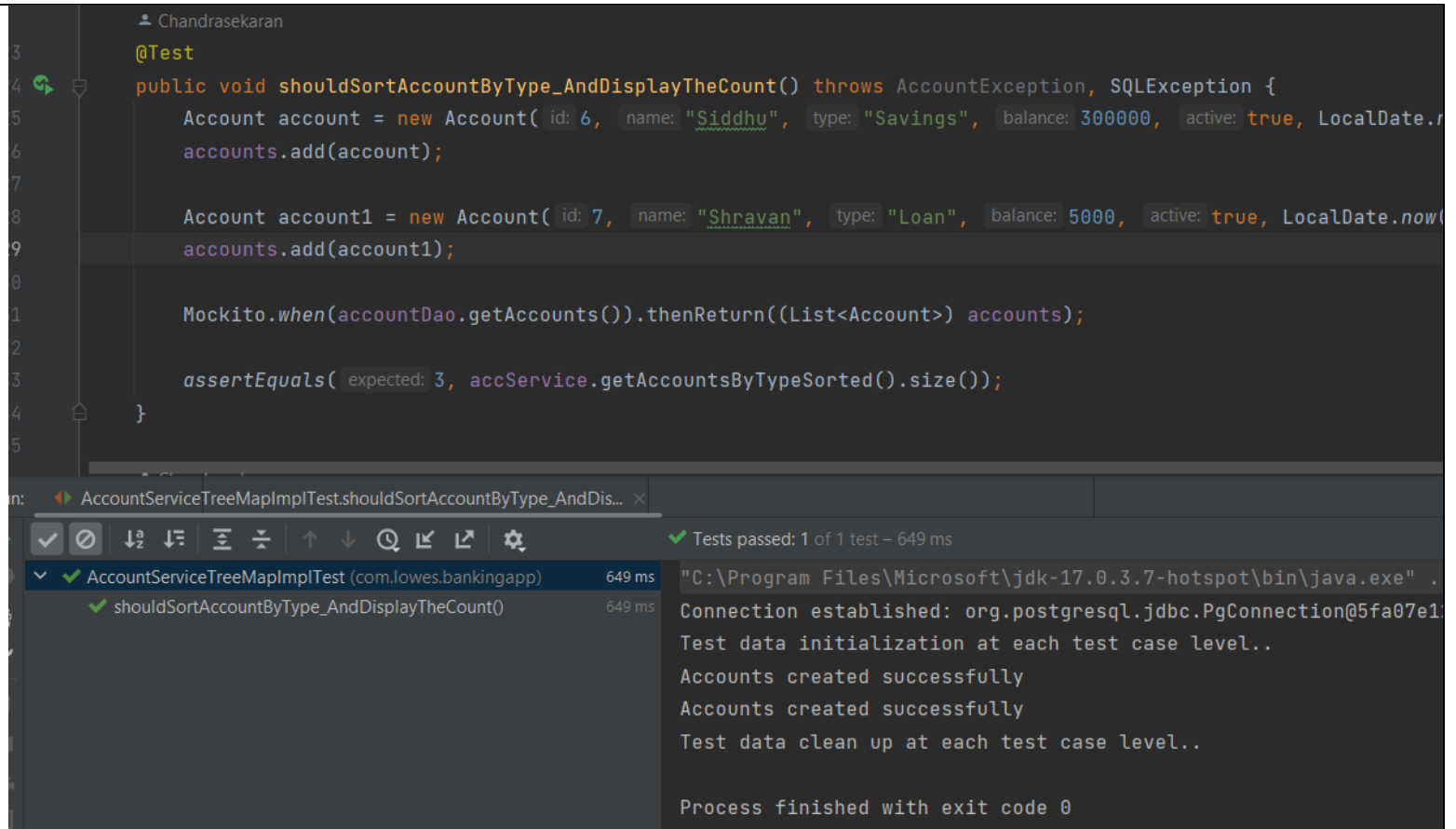
		<pre> 83 public void shouldDisplayNumberOfAccounts_WhenBalanceGreaterThanLimit() throws AccountException, SQLException 84 Account account = new Account(id: 6, name: "Poorni", type: "Savings", balance: 200000, active: true, LocalDate.now()); 85 accounts.add(account); 86 87 Account account1 = new Account(id: 7, name: "Sami", type: "Loan", balance: 300000, active: true, LocalDate.now()); 88 accounts.add(account1); 89 90 Mockito.when(accountDao.getAccounts()).thenReturn((List<Account>) accounts); 91 92 assertEquals(expected: 2, accService.balanceAccount(limit: 100000)); 93 } 94 </pre> <p>run: AccountServiceTreeMapImplTest.shouldDisplayNumberOfAccounts_... x</p> <p>Tests passed: 1 of 1 test – 671 ms</p> <p>AccountServiceTreeMapImplTest (com.lowes.bai 671 ms)</p> <p>shouldDisplayNumberOfAccounts_WhenBala 671 ms</p> <p>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" ...</p> <p>Connection established: org.postgresql.jdbc.PgConnection@4d910fd6</p> <p>Test data initialization at each test case level..</p> <p>Accounts created successfully</p> <p>Accounts created successfully</p> <p>Test data clean up at each test case level..</p> <p>Process finished with exit code 0</p>
#1 8	Case 16	shouldDisplayZeroAccounts_WhenBalanceLessThanLimit

		 <pre> 195 @Test 196 public void shouldDisplayZeroAccounts_WhenBalanceLessThanLimit() throws AccountException, SQLException { 197 Account account = new Account(id: 6, name: "Poorni", type: "Savings", balance: 2000, active: true, LocalDate.now()); 198 accounts.add(account); 199 200 Account account1 = new Account(id: 7, name: "Sami", type: "Loan", balance: 3000, active: true, LocalDate.now()); 201 accounts.add(account1); 202 203 Mockito.when(accountDao.getAccounts()).thenReturn((List<Account>) accounts); 204 assertEquals(expected: 0, accService.balanceAccount(limit: 100000)); 205 206 } </pre> <p>Run: AccountServiceTreeMapImplTest.shouldDisplayZeroAccounts_When... ×</p> <p>✓ Tests passed: 1 of 1 test – 785 ms</p> <ul style="list-style-type: none"> ✓ AccountServiceTreeMapImplTest (com.lowes.bankingapp) 785 ms <ul style="list-style-type: none"> ✓ shouldDisplayZeroAccounts_WhenBalanceLessThanLimit() 785 ms <p>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe"</p> <p>Connection established: org.postgresql.jdbc.PgConnection@355ce8</p> <p>Test data initialization at each test case level..</p> <p>Accounts created successfully</p> <p>Accounts created successfully</p> <p>Test data clean up at each test case level..</p> <p>Process finished with exit code 0</p>
#1 9	Case 17	shouldGroupAccountByType_AndDisplayTheCount

		<pre> @Test public void shouldGroupAccountByType_AndDisplayTheCount() throws AccountException, SQLException { Account account = new Account(id: 6, name: "Siddhu", type: "Savings", balance: 300000, active: true, LocalDate.now(), l accounts.add(account); Account account1 = new Account(id: 7, name: "Shravan", type: "Loan", balance: 5000, active: true, LocalDate.now(), l accounts.add(account1); Account account2 = new Account(id: 8, name: "Shashank", type: "Savings", balance: 400000, active: true, LocalDate.now(), l accounts.add(account2); Mockito.when(accountDao.getAccounts()).thenReturn((List<Account>) accounts); assertEquals(expected: 3, accService.getAccountsByType().size()); } </pre> <p>AccountServiceTreeMapImplTest.shouldGroupAccountByType_AndD...</p> <p>Tests passed: 1 of 1 test – 737 ms</p> <p>AccountServiceTreeMapImplTest (com.lowes.bankingapp) 737 ms</p> <p>shouldGroupAccountByType_AndDisplayTheCount() 737 ms</p> <p>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" ...</p> <p>Connection established: org.postgresql.jdbc.PgConnection@5fa07e12</p> <p>Test data initialization at each test case level..</p> <p>Accounts created successfully</p> <p>Accounts created successfully</p> <p>{Deposit=1, Savings=2, Loan=2}</p> <p>Test data clean up at each test case level..</p> <p>Process finished with exit code 0</p>
#2 0	Case 18	shouldSortAccountByType_AndDisplayTheCount

BANK OF BENGALURU: MOCKITO

Poornima Chandrasekaran

		 <pre> 3 Chandrasekaran 4 @Test 5 public void shouldSortAccountByType_AndDisplayTheCount() throws AccountException, SQLException { 6 Account account = new Account(id: 6, name: "Siddhu", type: "Savings", balance: 300000, active: true, LocalDate.now()); 7 accounts.add(account); 8 9 Account account1 = new Account(id: 7, name: "Shravan", type: "Loan", balance: 5000, active: true, LocalDate.now()); 10 accounts.add(account1); 11 12 Mockito.when(accountDao.getAccounts()).thenReturn((List<Account>) accounts); 13 14 assertEquals(expected: 3, accService.getAccountsByTypeSorted().size()); 15 } </pre> <p>Tests passed: 1 of 1 test – 649 ms</p> <p>AccountServiceTreeMapImplTest (com.lowes.bankingapp) 649 ms</p> <p>shouldSortAccountByType_AndDisplayTheCount() 649 ms</p> <p>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" .</p> <p>Connection established: org.postgresql.jdbc.PgConnection@5fa07e1</p> <p>Test data initialization at each test case level..</p> <p>Accounts created successfully</p> <p>Accounts created successfully</p> <p>Test data clean up at each test case level..</p> <p>Process finished with exit code 0</p>
#2 1	Case 19	shouldDisplayAccountByType_WhenAverageBalanceIsCalculated

		<pre> @Test public void shouldDisplayAccountByType_WhenAverageBalanceIsCalculated() throws AccountException, SQLException { Account account = new Account(id: 6, name: "Siddhu", type: "Savings", balance: 400000, active: true, LocalDate.now(), accounts.add(account); Account account1 = new Account(id: 7, name: "Shravan", type: "Loan", balance: 5000, active: true, LocalDate.now(), accounts.add(account1); Account account2 = new Account(id: 8, name: "Shashank", type: "Savings", balance: 400000, active: true, LocalDate.now(), accounts.add(account2); Mockito.when(accountDao.getAccounts()).thenReturn((List<Account>) accounts); assertEquals(expected: 3, accService.getAverageBalanceByType().size()); } </pre> <p>AccountServiceTreeMapImplTest.shouldDisplayAccountByType_WhenAverageBalanceIsCalculated 555 ms</p> <p>Tests passed: 1 of 1 test – 555 ms</p> <p>AccountServiceTreeMapImplTest (com.lowes.bankingapp) 555 ms</p> <p>shouldDisplayAccountByType_WhenAverageBalanceIsCalculated 555 ms</p> <p>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" ...</p> <p>Connection established: org.postgresql.jdbc.PgConnection@1649b0e6</p> <p>Test data initialization at each test case level..</p> <p>Accounts created successfully</p> <p>Accounts created successfully</p> <p>{Deposit=1200.0, Loan=3500.0, Savings=400000.0}</p> <p>Test data clean up at each test case level..</p> <p>Process finished with exit code 0</p>
#2 2	Case 20	containsNameTest

		<pre> @Test public void containsNameTest() throws SQLException { Account account = new Account(id: 6, name: "SiddhuKumar", type: "Savings", balance: 400000, active: true, LocalDate.now(), Lo accounts.add(account); Account account1 = new Account(id: 7, name: "Shravan", type: "Loan", balance: 5000, active: true, LocalDate.now(), Lo accounts.add(account1); Account account2 = new Account(id: 9, name: "Kumaran", type: "Loan", balance: 5000, active: true, LocalDate.now(), Lo accounts.add(account2); Mockito.when(accountDao.getAccounts()).thenReturn((List<Account>) accounts); assertEquals(expected: 4, accService.getAccountIdsContainsName("Kumar")); } </pre> <p>AccountServiceTreeMapImplTest.containsNameTest</p> <p>Tests passed: 1 of 1 test – 1 sec 215 ms</p> <p>AccountServiceTreeMapImplTest (com.lowes.bankingapp) 1 sec 215 ms</p> <p>containsNameTest() 1 sec 215 ms</p> <p>Connection established: org.postgresql.jdbc.PgConnection@4d910fd6</p> <p>Test data initialization at each test case level..</p> <p>Accounts created successfully</p> <p>Accounts created successfully</p> <p>Test data clean up at each test case level..</p> <p>Process finished with exit code 0</p>
#2 3	Case 21	ImportFileTest

		<pre> @Test public void ImportFileTest() throws IOException, SQLException { Account account = new Account(id: 16, name: "SiddhuKumar", type: "Savings", balance: 400000.00, active: true, LocalDate.now()); accounts.add(account); Account account1 = new Account(id: 17, name: "Shravan", type: "Loan", balance: 5000.00, active: true, LocalDate.now()); accounts.add(account1); //Import validation - 2 from BeforeEach block and 4 from the Test file Mockito.lenient().when(accountDao.createAccount(account)).thenReturn(true); retVal = accService.ImportData(filePath: "./input/account.txt"); assertEquals(expected: 8, retVal.size()); } </pre> <p>AccountServiceTreeMapImplTest.ImportFileTest</p> <p>Tests passed: 1 of 1 test – 12 sec 613 ms</p> <p>AccountServiceTreeMapImplTest (com.lowes.bankingapp) 12 sec 613 ms</p> <p>ImportFileTest() 12 sec 613 ms</p> <p>"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" ...</p> <p>Connection established: org.postgresql.jdbc.PgConnection@4d910fd6</p> <p>Test data initialization at each test case level..</p> <p>Accounts created successfully</p> <p>Accounts created successfully</p> <p>main Accounts data imported successfully.Test data clean up at each test</p> <p>Process finished with exit code 0</p>
#2 4	Case 22	ExportFileTest

```
public void ExportFileTest() throws IOException, SQLException {
    //Before Export- Checking for empty file
    String fileName = "./output/accountTestOut.txt";
    RandomAccessFile file = new RandomAccessFile(fileName, mode: "rw");
    BufferedReader br = new BufferedReader(new FileReader(fileName));
    file.setLength(0);
    assertEquals( expected: null, br.readLine());
    br.close();

    Mockito.when(accountDao.getAccounts()).thenReturn((List<Account>) accounts);
    //After export - checking for loaded file. Comparing the number of lines
    accService.ExportData(fileName, retVal);
    BufferedReader br1 = new BufferedReader(new FileReader(fileName));
    int lines = 0;
    while (br1.readLine() != null) lines ++;
    br1.close();
    assertEquals( expected: 2, lines);
}
```

AccountServiceTreeMapImplTest.ExportFileTest ×



✓ Tests passed: 1 of 1 test – 4 sec 588 ms

✓ AccountServiceTreeMapImplTest (com.lowes.bankingapp) 4 sec 588 ms

✓ ExportFileTest() 4 sec 588 ms

```
"C:\Program Files\Microsoft\jdk-17.0.3.7-hotspot\bin\java.exe" .
Connection established: org.postgresql.jdbc.PgConnection@1649b0e
Test data initialization at each test case level..
Accounts created successfully
Accounts created successfully
main Account data exported successfully.Test data clean up at ea
Process finished with exit code 0
```