Task 1: Create and Execute the test cases for the real world scenarios like banking system

Note:
It's essential to have a good understanding of the banking system, be organized, and document everything to ensure a thorough and effective manual functional test.

Banking System Test Cases

Here are some general steps you can follow:

1. **Understand the Banking System: First, you need to understand the banking system you are testing. This includes understanding the system's functionality, features, and how it works.**

2. **Identify the Test Cases: Once you understand the system, you need to identify the test cases that you will be testing. Test cases are essentially a set of steps that you will follow to verify the system's functionality.**

3. **Prepare the Test Data: You need to prepare test data that will be used in the test cases. This includes creating user accounts, generating test transactions, and any other relevant data.**

4. **Execute the Test Cases: With the test cases and data prepared, you can start executing the test cases. Ensure that you follow the steps in each test case carefully and record the results.**

5. **Report the Results: After executing the test cases, you need to report the results to the team. This includes noting any issues or defects found during testing.**

6. **Retest and Verify: Once the issues are resolved, retest the system to ensure that the issues have been fixed, and the system is functioning correctly.**

7. **Document the Testing Process: Document the testing process and create a report to help improve the testing process in the future.**

Simple banking system scenario for manual functional testing for freshers:

Scenario: User Login and Account Management

Objective: **To ensure that users can log in to the banking system and perform account management tasks such as checking account balance, transferring funds, and updating personal information.**

Test Cases:

1. **Verify that the login page is displayed correctly and the user can enter valid login credentials.**
2. **Verify that the system displays appropriate error messages for invalid login credentials.**
3. **Verify that the user can view account balance information.**
4. **Verify that the user can transfer funds to another account within the system.**
5. **Verify that the user can update personal information such as address and phone number.**
6. **Verify that the system displays appropriate error messages for invalid information entered.**
7. **Verify that the user can log out of the system.**

Test Data:

1. **Valid user login credentials (username and password).**
2. **Invalid user login credentials (wrong username or password).**
3. **Test accounts with varying account balances.**
4. **Test accounts to transfer funds to.**
5. **Test user personal information to update.**
6. **Invalid test data such as incorrect phone number format.**

Execution Steps:

1. **Open the login page of the banking system.**

2. **Enter valid user login credentials and click on the login button.**

3. **Verify that the account balance information is displayed.**
4. **Transfer funds to another account and verify that the balance is updated accordingly.**
5. **Update personal information and verify that the changes are reflected in the system.**
6. **Enter invalid information and verify that appropriate error messages are displayed.**
7. **Logout of the system.**
8. **Repeat the above steps with invalid login credentials.**
9. **Repeat the above steps with varying test data and scenarios.**

Reporting:

1. **Document the test results and report any defects found during testing.**
2. **Provide recommendations for improving the system's functionality and user experience.**

Other Additional Scenarios

Scenario 1: Create a New Account

Verify that a new account can be created with valid customer information
Verify that the account number generated is unique and follows the expected format
Verify that the initial balance is correctly set to zero

Scenario 2: Deposit Funds

Verify that funds can be deposited into an existing account with a valid account number and amount
Verify that the new balance is correctly updated after the deposit

Scenario 3: Withdraw Funds

Verify that funds can be withdrawn from an existing account with a valid account number and amount
Verify that the new balance is correctly updated after the withdrawal

Verify that an error is displayed if the withdrawal amount is greater than the available balance

Scenario 4: Transfer Funds

Verify that funds can be transferred between two existing accounts with valid account numbers and amounts
Verify that the new balances of both accounts are correctly updated after the transfer
Verify that an error is displayed if the transfer amount is greater than the available balance in the sender account

Extra Additional test cases cover different functionalities of the banking system and can help ensure the system is fully tested.

Scenario 5: Close Account

Verify that an existing account can be closed with a valid account number and reason for closure
Verify that the account is no longer accessible or available for further transactions

Scenario 6: Account Statement

Verify that an account statement can be generated for an existing account with a valid account number and date range
Verify that the statement shows all transactions within the specified date range and the current balance of the account

Scenario 7: Overdraft Protection

Verify that overdraft protection is enabled for an account with a valid account number and sufficient funds
Verify that the account balance does not go below zero even if a withdrawal is made that exceeds the available balance
Verify that an error is displayed if overdraft protection is not available or there are insufficient funds to cover the withdrawal

Scenario 8: Interest Calculation

Verify that interest is calculated correctly for an account with a valid account number and interest rate
Verify that the interest is added to the account balance at the appropriate interval (e.g. monthly or annually)

These additional test cases cover different functionalities of the banking system and can help ensure the system is fully tested.