Here are some test cases to thoroughly test the Create wallet functionality of Trust Wallet app:

**Scenario 1: Create new wallet through Secret phrase**

Positive Test Cases:

\* TC-01: Verify that a new wallet can be created without biometric:

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet"
* taps on “Secret phrase **Create**” 🡪 Skip
* User sets a strong password that meets the app's complexity requirements.
* Create 6-digit Passcode 🡪 tap “Deny” Biometric Login

Expected Result:

1. the user is directed to the Welcome abroad main wallet screen.
2. A new wallet is created successfully with unique name

\* TC-02: Verify Wallet creation with biometric:

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet"
* taps on “Secret phrase **Create**”” 🡪 Skip
* User sets a strong password that meets the app's complexity requirements.
* Create 6-digit Passcode 🡪 tap “Confirm” Biometric Login
* Authenticate trust wallet by touching “fingerprint sensor”

Expected Result:

1. the user is directed to the Welcome abroad main wallet screen.
2. A new wallet is created successfully with unique name

\* TC-03: Verify Wallet creation with manual Backup:

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet"
* taps on “Secret phrase **Create**” 🡪 Back up manually
* select radio buttons that meets the app's complexity requirements.
* Tap “Continue” 🡪 Continue 🡪 Confirm secret phrase 🡪 Continue
* User sets a strong password that meets the app's complexity requirements.
* Create 6-digit Passcode 🡪 tap “Confirm” Biometric Login
* Authenticate trust wallet by touching “fingerprint sensor”

Expected Result: A new wallet is created successfully, and the user is directed to the Welcome abroad main wallet screen.

\* TC-04: Verify Wallet creation with google drive Backup:

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet"
* taps on “Secret phrase **Create**” 🡪 Back up on google driver 🡪 Enter Backup name Ex:Test 🡪 Continue
* Choose google account 🡪 Continue 🡪 Enter password
* Tap “set encrypted password” 🡪 Confirm password 🡪Select policy radio buttons 🡪 tap Confirm
* select radio buttons that meets the app's complexity requirements.
* Tap “Continue” 🡪 Continue 🡪 Confirm secret phrase 🡪 Continue
* User sets a strong password that meets the app's complexity requirements.
* Create 6-digit Passcode 🡪 tap “Deny” Biometric Login

Expected Result: A new wallet is created successfully, and the user is directed to the Welcome abroad main wallet screen.

\* TC-05: Back navigation on Wallet creation Backup screen:

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet" 🡪 Tap back navigation arrow or device back button

Expected Result: The user will navigate to the home page.

\* TC-06: Back navigation on Wallet creation Passcode screen:

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet" 🡪 Skip
* Tap back navigation arrow or device back button

Expected Result: The user will not navigate to the previous page and the user needs to provide passcode to create wallet.

\* TC-07: Verify wallet creation with mismatched passcode

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet"
* taps on “Secret phrase **Create**” 🡪 Skip
* Create 6-digit Passcode 🡪
* On Confirm passcode enter invalid 6-digit Passcode

Expected Result: The user will navigate back to Create passcode screen to set passcode

\* TC-08: Verify that the Trust wallet user interface is user-friendly and intuitive.

* Input: App opened; Various screens and navigation menus

Test Steps:

* Navigate through the app's menus
* Perform common tasks

Expected Result: Interface is easy to understand and use.

Negative Test Cases:

\* TC-09: Cancel wallet creation from passcode screen

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet"
* taps on “Secret phrase **Create**” 🡪 Skip
* On a create password screen, User kill App and launch again

Expected Result:

1. The user will land to the same create password screen
2. Session maintenance once user kill and launch app again

\* TC-10: Network interruption while creating wallet:

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet"
* Turnoff network
* taps on “Secret phrase **Create**” 🡪 Skip
* User sets a strong password that meets the app's complexity requirements.
* Create 6-digit Passcode 🡪 tap “Deny” Biometric Login

Expected Result: A new wallet is created successfully, and the user is directed to the Welcome abroad main wallet screen.

**Scenario 2: Create new wallet through Swift**

\* TC-11: Verify that a new wallet can be created using Swift

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet"
* taps on “Swift **Create**” 🡪 taps on all checkboxes 🡪 continue
* Select radio buttons on “Quick quiz” screen 🡪 tap check answer 🡪 tap Got it, continue
* Enter wallet name Ex: Test 🡪 Done
* Tap continue on “create a passkey” screen 🡪 confirm fingerprint on “Processing screen”
* Create 6-digit Passcode 🡪 tap “Deny” Biometric Login 🡪 tap Get started

Expected Result:

1. Swift is ready screen should display
2. A new wallet is created successfully with unique name, and the user is directed to the home screen.

\* TC-12: Verify that a new wallet can be created using Swift, Once user cancel Swift is ready screen.

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet"
* taps on “Swift **Create**” 🡪 taps on all checkboxes 🡪 continue
* Select radio buttons on “Quick quiz” screen 🡪 tap check answer 🡪 tap Got it, continue
* Enter wallet name Ex: Test 🡪 Done
* Tap continue on “create a passkey” screen 🡪 confirm fingerprint on “Processing screen”
* Create 6-digit Passcode 🡪 tap “Deny” Biometric Login 🡪 tap Get started
* Cancel Swift is ready screen.

Expected Result:

1. A new wallet is created successfully with unique name, and the user is directed to the home screen.

\* TC-13: Verify that a new wallet name can be created once user don’t enter name as per rule.

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet"
* taps on “Swift **Create**” 🡪 taps on all checkboxes 🡪 continue
* Select radio buttons on “Quick quiz” screen 🡪 tap check answer 🡪 tap Got it, continue
* Enter wallet name Ex: Tes🡪

Expected Result:

1. The wallet name should not be created and Done button should be activated

\* TC-14: Verify that a new wallet can be created using Swift once user tap cancel on “create a passkey” screen

Pre-Condition: App is installed and launched for the first time.

Test Steps:

* User opens the app and click on Get started
* taps on "Create a new wallet"
* taps on “Swift **Create**” 🡪 taps on all checkboxes 🡪 continue
* Select radio buttons on “Quick quiz” screen 🡪 tap check answer 🡪 tap Got it, continue
* Enter wallet name Ex: Test 🡪 Done
* Tap cancel on “create a passkey” screen

Expected Result:

1. The user will navigate to the Set wallet name screen

\* TC-15: Verify data encryption

* Description: Verify that user data is encrypted at rest and in transit.
* Input: Sensitive user data (e.g., passcode)
* Steps: Inspect the network traffic and database storage to ensure that data is encrypted

Expected Result: User data is encrypted and protected from unauthorized access.

\* TC-16**:** Verifyprivate key encryption

* Description: Verify that the private key is encrypted and stored securely.
* Input: App opened; wallet created
* Steps: Attempt to access the private key directly from the device storage

Expected Result:  Private key is encrypted and cannot be accessed.