|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Steps** | **Expected Result** | **Status** |
| **TC01** | Verify console loading | Navigate to: <https://console.brahma.fi/> | Console should load without errors | Passed/Fails |
| **TC02** | Connect wallet | Click on 'Connect wallet and inject web3 provider' | Wallet should connect successfully | Passed/Fails |
| **TC03** | Bridge functionality | Click on the "Bridge" button | Bridging process should initiate successfully | Passed/Fails |
| **TC04** | Bridging with invalid asset | Select an unsupported token | Error message: Asset not supported | Passed/Fails |
| **TC05** | Bridging with large amount | Enter an excessively large amount | System should handle large amount gracefully | Passed/Fails |
| **TC06** | Verify relayer fee calculation | Observe displayed relayer fee | Relayer fee matches expected value | Passed/Fails |
| **TC07** | Insufficient funds | Enter amount greater than wallet balance | Error message: Insufficient funds | Passed/Fails |
| **TC08** | Incorrect recipient address | Enter invalid recipient address | Error message: Invalid recipient address | Passed/Fails |
| **TC09** | Handling network congestion | Initiate bridging during high congestion | System handles congestion gracefully | Passed/Fails |
| **TC10** | Bridging on Ethereum mainnet | Initiate bridging on Ethereum mainnet | Bridging works as expected | Passed/Fails |
| **TC11** | Bridging on testnet | Initiate bridging on a testnet (e.g., Ropsten) | Bridging functions correctly on testnet | Passed/Fails |
| **TC12** | Token approval | Click on "Approve" button | Token approved for bridging | Passed/Fails |
| **TC13** | Bridging without token approval | Attempt bridging without approval | Error message: Token needs approval | Passed/Fails |
| **TC14** | Concurrent bridging transactions | Initiate multiple transactions concurrently | System queues and processes fairly | Passed/Fails |
| **TC15** | Handling stuck transactions | Initiate a stuck transaction | System handles stuck transaction appropriately | Passed/Fails |
| **TC16** | Bridging with minimum amount | Enter the minimum amount (e.g., 0.0001 ETH) | Transaction processes successfully | Passed/Fails |
| **TC17** | Long recipient address | Enter recipient address with more than 42 characters | System handles long addresses without errors | Passed/Fails |
| **TC18** | Mandatory "Bridge to" field | Leave "Bridge to" field empty | Error message: Enter a valid address | Passed/Fails |
| **TC19** | Numeric input for "Time" | Enter non-numeric value in "Time" field | System rejects non-numeric input | Passed/Fails |
| **TC20** | Confirmation dialog | Confirm bridging action | Dialog prompts user to confirm transaction | Passed/Fails |

**Navigation** **Functionality**

**TC01 :** Verify console loading .

Steps:

1. Navigate to: <https://console.brahma.fi/>
2. Check how much time it takes to load the page
3. Check whether the page is loading without any error

Expected Result: Console should load without errors

**TC02 :** Connect wallet

Steps:

1. Click on connect wallet button.
2. After clicking we should see the different wallet options available.
3. After clicking on a wallet option we should be able to connect to the wallet.

Expected Result: Wallet should connect successfully

**Bridge Functionality:**

**TC03:** Verify that the “Bridge” button is functional.

Steps:

1. Navigate to the Brahma.fi Console.
2. Click on the “Bridge” button.

Expected Result: The bridging process should initiate successfully.

**Edge Cases:**

**TC04:** Test bridging with an invalid asset (e.g., an unsupported token).

Steps:

1. Navigate to the Console.
2. Select an unsupported asset (e.g., an imaginary token).
3. Click on the “Bridge” button.

Expected Result: An error message should be displayed, indicating that the asset is not supported.

**TC05:** Test bridging with a very large amount (e.g., beyond practical limits).

Steps:

1. Navigate to the Console.
2. Enter an excessively large amount for bridging.
3. Click on the “Bridge” button.

Expected Result: The system should handle the large amount gracefully (e.g., display a warning or prevent the transaction).

**Granularity and Details:**

**TC06:** Verify that the relayer fee (ETH) is correctly calculated.

Steps:

1. Navigate to the Console.
2. Enter valid parameters for bridging.
3. Observe the relayer fee displayed.

Expected Result: The relayer fee should match the expected value based on the transaction details.

**Transaction Failures:**

**TC07:** Verify behavior when bridging with insufficient funds.

Steps:

1. Navigate to the Console.
2. Select an asset.
3. Enter an amount greater than the wallet balance.
4. Click on the “Bridge” button.

Expected Result: An error message should be displayed, indicating insufficient funds.

**TC08:** Test bridging with an incorrect recipient address.

Steps:

1. Navigate to the Console.
2. Enter an invalid recipient address (e.g., a random string).
3. Click on the “Bridge” button.

Expected Result: An error message should appear, indicating an invalid recipient address.

**TC09:** Verify handling of network congestion during bridging.

Steps:

1. Navigate to the Console.
2. Initiate a bridging transaction during high network congestion.
3. Observe the response time.

Expected Result: The system should handle network congestion gracefully (e.g., display a loading indicator or retry mechanism).

**Network Environments:**

**TC10:** Verify bridging behavior on the Ethereum mainnet.

Steps:

1. Navigate to the Console.
2. Select the Ethereum mainnet.
3. Initiate a bridging transaction.

Expected Result: The bridging process should work as expected on the Ethereum mainnet.

**TC11:** Test bridging on a testnet (e.g., Ropsten or Rinkeby).

Steps:

1. Navigate to the Console.
2. Select a testnet (e.g., Ropsten).
3. Initiate a bridging transaction.

Expected Result: The bridging process should function correctly on the selected testnet.

**Token Approvals:**

**TC12:** Verify that the user can approve the token for bridging.

Steps:

1. Navigate to the Console.
2. Select an asset.
3. Click on the “Approve” button.
4. Confirm the transaction.

Expected Result: The token should be approved for bridging.

**TC13:** Test bridging without token approval.

Steps:

1. Navigate to the Console.
2. Select an asset.
3. Attempt to bridge without approving the token.

Expected Result: An error message should be displayed, indicating that the token needs approval.

**Pending Transactions:**

**TC14:** Verify behavior when initiating multiple bridging transactions concurrently.

Steps:

1. Navigate to the Console.
2. Initiate two or more bridging transactions simultaneously.
3. Observe how pending transactions are handled.

Expected Result: The system should queue and process transactions in a fair and efficient manner.

**TC15:** Test bridging with a transaction that gets stuck (e.g., due to network congestion or other issues).

Steps:

1. Navigate to the Console.
2. Initiate a bridging transaction.
3. Monitor the transaction status.

Expected Result: The system should handle stuck transactions appropriately (e.g., provide a way to cancel or retry).

**Edge Cases:**

**TC16:** Verify bridging with a minimum amount (e.g., 0.0001 ETH).

Steps:

1. Navigate to the Console.
2. Enter the minimum amount for bridging.
3. Click on the “Bridge” button.

Expected Result: The system should process the transaction successfully with the minimum amount.

**TC17:** Test bridging with an unusually long recipient address.

Steps:

1. Navigate to the Console.
2. Enter a recipient address with more than 42 characters.
3. Click on the “Bridge” button.

Expected Result: The system should handle long recipient addresses without errors.

**Validation Checks:**

**TC18:** Verify that the “Bridge to” field is mandatory.

Steps:

1. Navigate to the Console.
2. Leave the “Bridge to” field empty.
3. Click on the “Bridge” button.

Expected Result: An error message should prompt the user to enter a valid address.

**TC19:** Verify that the “Time” field accepts only numeric values.

Steps:

1. Navigate to the Console.
2. Enter a non-numeric value in the “Time” field.
3. Click on the “Bridge” button.

Expected Result: The system should reject non-numeric input.

**Confirmation Dialog:**

**TC20:** Verify that a confirmation dialog appears before initiating bridging.

Steps:

1. Navigate to the Console.
2. Enter valid parameters for bridging.
3. Click on the “Bridge” button.

Confirm the bridging action.

Expected Result: A confirmation dialog should display, asking the user to confirm the transaction.