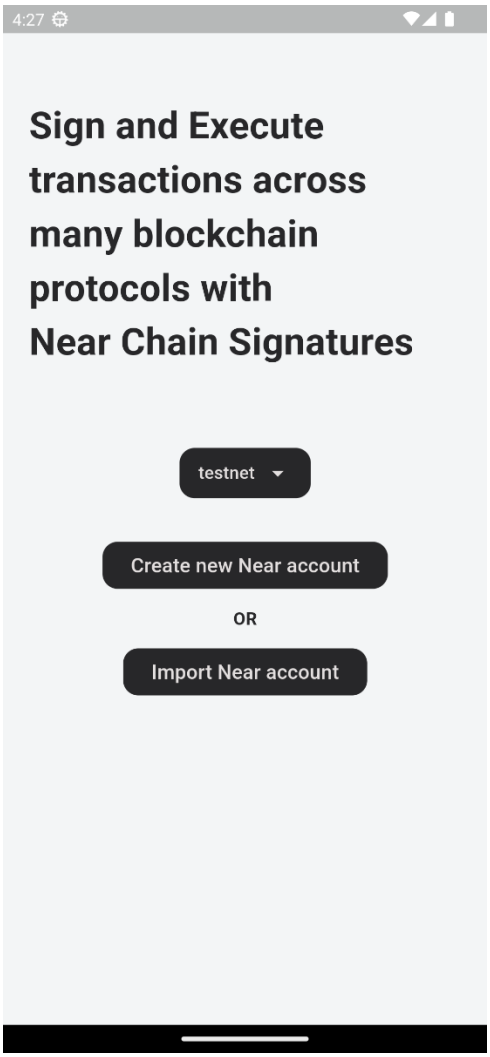
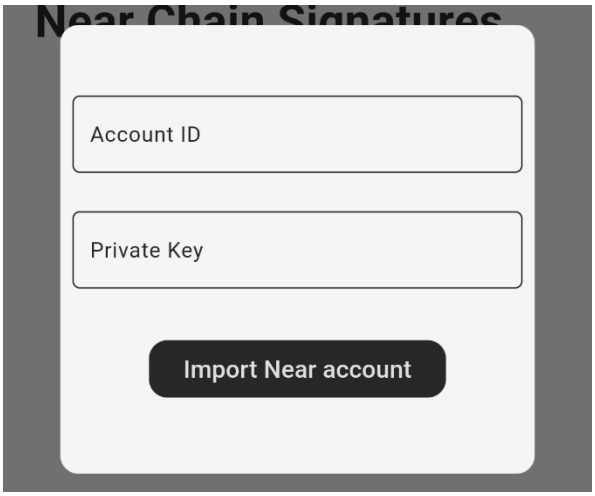
Near Chain Signatures usage example:

To begin working with Near Chain Signatures, you need to create or import a Near account.
Our application's initial screen looks like this:

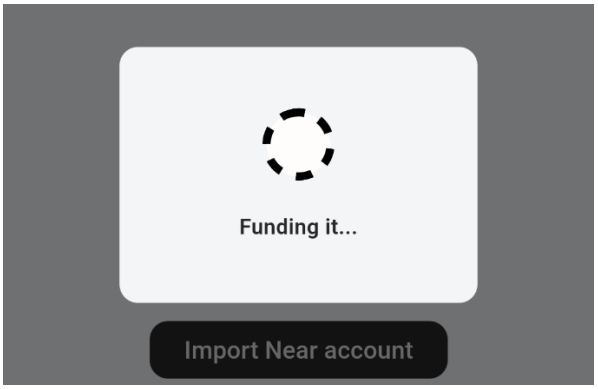


Here, you can choose the network (testnet/mainnet) to work with. Currently, only the testnet is available. The "Create new Near account" button allows you to create a new, funded Near account from scratch. Alternatively, you can import your existing Near account by tapping on "Import Near Account" and entering your account credentials. The private key must be in the format "ed25519: …".

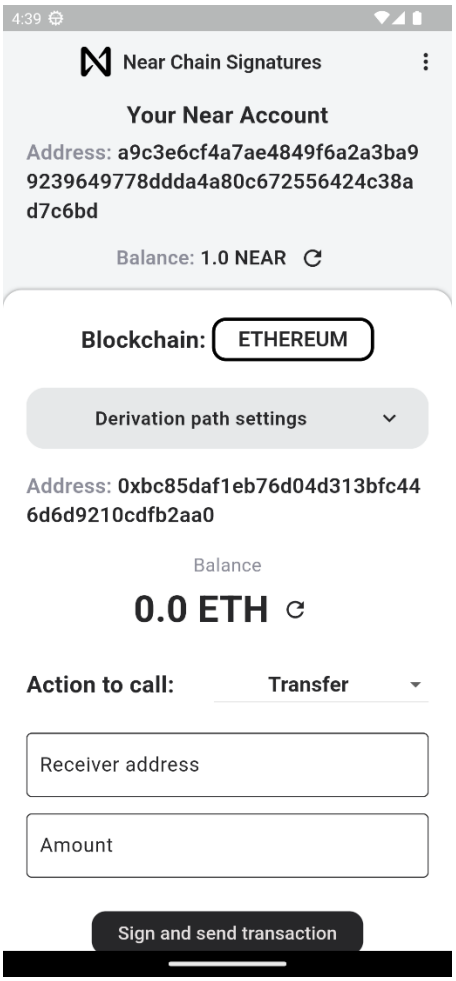
The Import Dialog looks like this:



The Near account creation process looks like this:



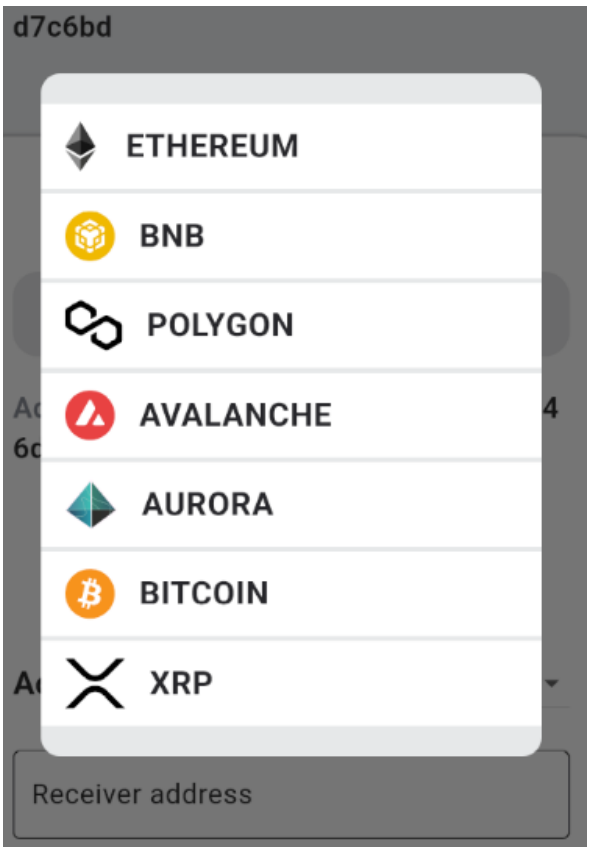
After creating or importing a Near account, you'll see the main action screen:



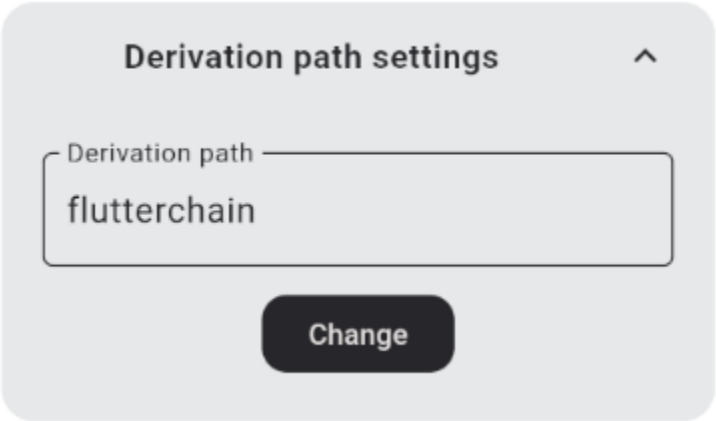
At the top of the page, you'll see your Near account ID and its balance. Be cautious with your balance, as calling Near chain transactions requires gas to work.

Below, you can see the blockchain information, including the address, balance, and actions that can be performed.
By tapping on the blockchain name, such as "Ethereum," here:

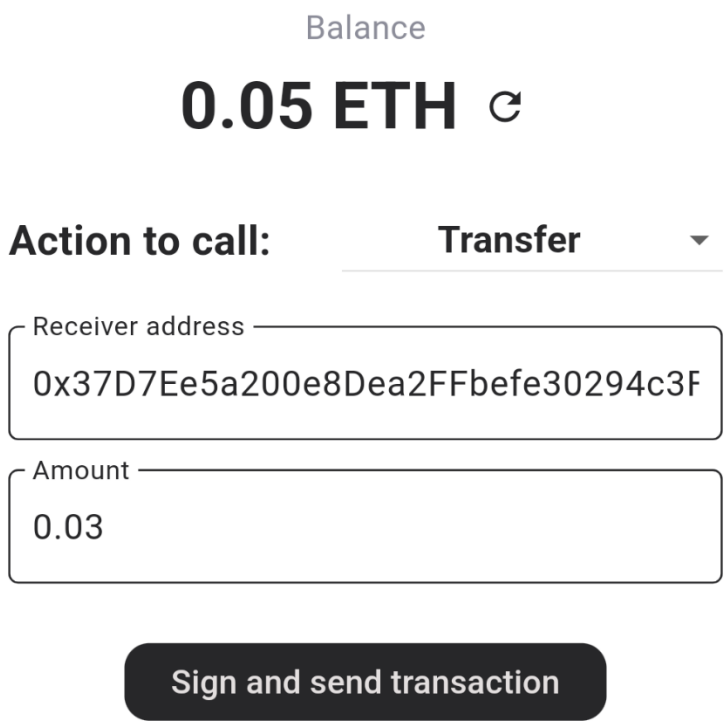
you can change the blockchain you want to work with:



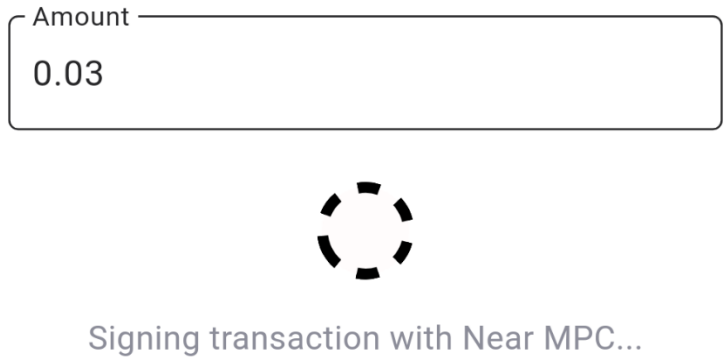
Additionally, you can generate another blockchain account by changing the "derivation path." By default, "flutterchain" is used, but you can specify your own.



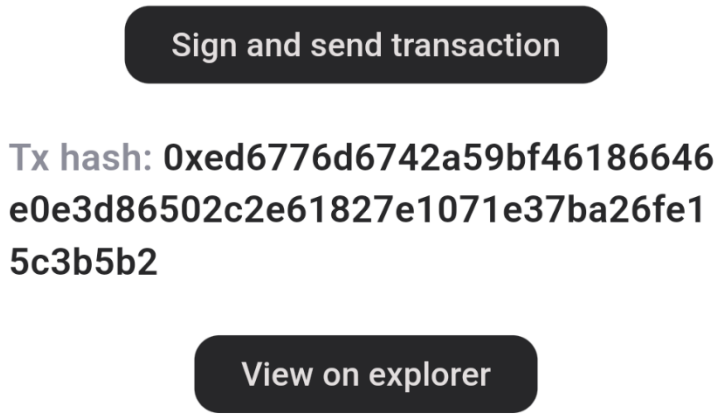
We can perform two types of actions with the blockchain: "Transfer" and "Smart contract call" (the latter is currently only available for EVM blockchains). After entering the receiver's address and the amount of currency, tap on "Sign and send transaction".



You'll be able to see the status of the executing operation below:



After successful execution, you'll see the transaction hash and a button to view your transaction result on a blockchain explorer in your browser.



To call a smart contract, we need to fill in the following fields:

Smart Contract Function field requires the function name of the smart contract and the type of arguments that it takes. If there are no arguments, then we leave it in the format "function1()".

The Smart Contract Args field requires arguments for this function. If there are no arguments, we leave it in the format "[]".