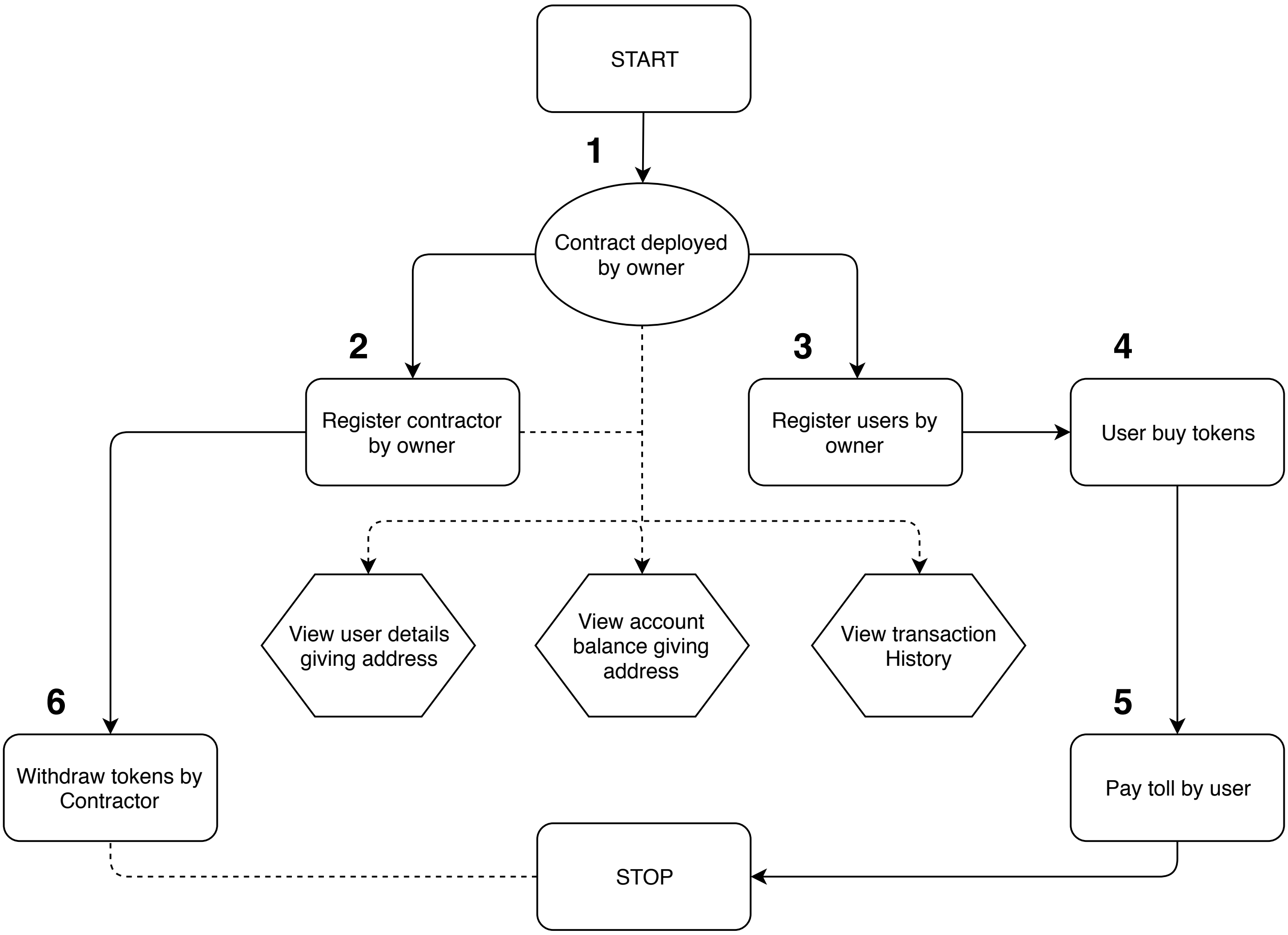
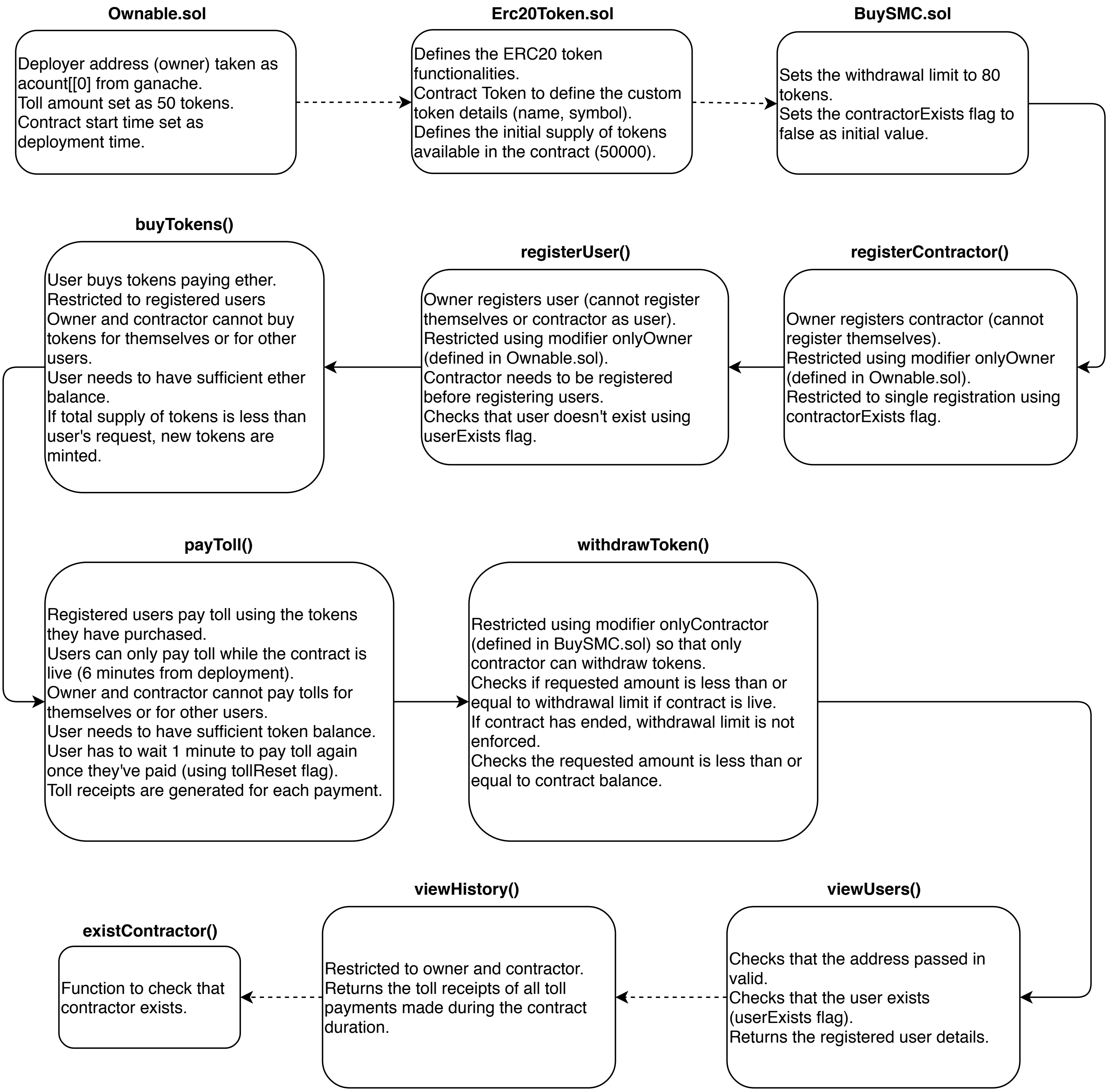

START
1
Contract deployed by owner
2
Register contractor by owner
3
Register users by owner
4
User buy tokens
View user details giving address
View account balance giving address
View transaction History
6
Withdraw tokens by Contractor
5
Pay toll by user
STOP

1. *Owner deploys the smart contract.*
2. *Owner should register a contractor to start the toll contract.*
3. *Owner needs to register valid users.*
4. *Registered users will be able to purchase tokens using ether.*
5. *User will be able to pay toll (50 tokens) using purchased tokens.*
6. *Contractor will be able to withdraw tokens from contract address once contract balance reaches 80 tokens. Once the contract expires(6 minutes), contractor can withdraw the remaining contract balance.*



**Ownable.sol**

Deployer address (owner) taken as acount[[0] from ganache.
Toll amount set as 50 tokens.
Contract start time set as deployment time.

**Erc20Token.sol**

Defines the ERC20 token functionalities.
Contract Token to define the custom token details (name, symbol).
Defines the initial supply of tokens available in the contract (50000).

**BuySMC.sol**

Sets the withdrawal limit to 80 tokens.
Sets the contractorExists flag to false as initial value.

**registerContractor()**

Owner registers contractor (cannot register themselves).
Restricted using modifier onlyOwner (defined in Ownable.sol).
Restricted to single registration using contractorExists flag.

**registerUser()**

Owner registers user (cannot register themselves or contractor as user).
Restricted using modifier onlyOwner (defined in Ownable.sol).
Contractor needs to be registered before registering users.
Checks that user doesn't exist using userExists flag.

**buyTokens()**

User buys tokens paying ether.
Restricted to registered users
Owner and contractor cannot buy tokens for themselves or for other users.
User needs to have sufficient ether balance.
If total supply of tokens is less than user's request, new tokens are minted.

**payToll()**

Registered users pay toll using the tokens they have purchased.
Users can only pay toll while the contract is live (6 minutes from deployment).
Owner and contractor cannot pay tolls for themselves or for other users.
User needs to have sufficient token balance.
User has to wait 1 minute to pay toll again once they've paid (using tollReset flag).
Toll receipts are generated for each payment.

**withdrawToken()**

Restricted using modifier onlyContractor (defined in BuySMC.sol) so that only contractor can withdraw tokens.
Checks if requested amount is less than or equal to withdrawal limit if contract is live.
If contract has ended, withdrawal limit is not enforced.
Checks the requested amount is less than or equal to contract balance.

**viewUsers()**

Checks that the address passed in valid.
Checks that the user exists (userExists flag).
Returns the registered user details.

**viewHistory()**

Restricted to owner and contractor.
Returns the toll receipts of all toll payments made during the contract duration.

**existContractor()**

Function to check that contractor exists.