**Project Overview**

QUOKKAOfficial is a smart contract developed to create a decentralized cryptocurrency that uses the Ethereum blockchain. It is an ERC-20 token that includes several features such as a transaction tax and a liquidity pool.

**Functional Requirements**

**Roles**

1. Owner - the person who deploys the contract and has the ability to change settings
2. User - any person who holds tokens of QUOKKAOfficial

**Features**

1. Transaction tax - a percentage of each transaction goes towards the tax fee, which is then distributed to the liquidity pool and holders of the token
2. Liquidity pool - a portion of the transaction tax is added to the liquidity pool, which helps maintain the stability of the token and increases its value
3. Excluded addresses - the ability to exclude certain addresses from the transaction tax and liquidity pool
4. Maximum transaction amount - a limit on the amount of tokens that can be transferred in a single transaction
5. Automatic liquidity pool and token burn - the contract automatically adds liquidity to the pool and burns tokens, which helps maintain the stability of the token and increases its value

**Use Case**

1. User purchases QUOKKAOfficial tokens using ETH on a cryptocurrency exchange.
2. User holds onto their tokens or trades them with other users on the exchange.
3. User transfers their tokens to another wallet, with a maximum transaction limit in place.
4. A portion of the transaction tax is distributed to the liquidity pool, increasing the value of the token.
5. A portion of the transaction tax is distributed to other token holders, incentivizing them to hold onto their tokens and maintain the stability of the token.
6. Automatic liquidity pool and token burn is activated, further increasing the value and stability of the token.

**Technical Requirements**

**Architecture Overview**

The QUOKKAOfficial contract is an ERC-20 token that utilizes the Ethereum blockchain. It includes several mappings and variables to keep track of the token supply, transaction tax, liquidity pool, excluded addresses, and more. The contract also uses several libraries, including SafeMath and Address, to prevent overflow and improve security. The contract utilizes the UniswapV2Router02 to provide liquidity to the pool and maintain the value of the token.

**Contract Information**

1. Assets - QUOKKAOfficial is an ERC-20 token that can be bought and sold on cryptocurrency exchanges.
2. Functions - the contract includes several functions for transferring tokens, adding and removing excluded addresses, and more.
3. Events - the contract includes several events that are emitted when certain actions are taken, including when the swap and liquidity feature is enabled and when liquidity is added to the pool.
4. Modifiers - the contract includes a modifier that locks the swap and liquidity feature during certain actions, to prevent potential security vulnerabilities.

**Use Cases**

1. Transfer tokens - User transfers QUOKKAOfficial tokens to another wallet using the **transfer** function.
2. Exclude address - Owner excludes a specific address from the transaction tax and liquidity pool using the **excludeFromReward** function.
3. Include address - Owner includes a specific address in the transaction tax and liquidity pool using the **includeInReward** function.
4. Change transaction tax - Owner changes the transaction tax fee using the **updateTaxFee** function.
5. Change liquidity fee - Owner changes the liquidity fee using the **updateLiquidityFee** function.
6. Enable swap and liquidity feature - Owner enables the swap and liquidity feature using the **enableSwapAndLiquify** function.
7. Swap tokens for ETH and add liquidity - The swap and liquidity feature is automatically triggered when the number of tokens sold reaches a certain threshold, and adds liquidity to the pool while burning tokens. This is all handled by the contract automatically.