README: Understanding DiscountLogic Smart Contract

This README provides a detailed explanation of the DiscountLogic smart contract and its associated contracts - LoyaltyToken and UnspentLoyaltyToken.

Overview

The provided Solidity code consists of three contracts:

1. LoyaltyToken.sol: ERC20 token representing a loyalty token named "DTK."
2. UnspentLoyaltyToken.sol: Contract for managing unspent loyalty tokens.
3. DiscountLogic.sol: Contract for handling discount logic and customer purchases using loyalty tokens.

**LoyaltyToken.sol**

This contract creates a custom ERC20 token named "DiscountToken" (DTK) and initializes the owner of the contract with an initial supply of 500 DTK tokens. It inherits from the OpenZeppelin ERC20 contract.

Functions and Properties:

* constructor: Initializes the token with the name "DiscountToken" and symbol "DTK". It also sets the contract deployer as the owner and mints 500 DTK tokens to the owner.

**UnspentLoyaltyToken.sol**

This contract manages the unspent loyalty tokens by allowing users to deposit their tokens. It holds a reference to the LoyaltyToken contract to perform token transfers.

Functions and Properties:

* constructor: Initializes the contract by setting the address of the LoyaltyToken contract.
* depositTokens: Allows a user to deposit tokens to this contract, updating the user's unspent token balance.

**DiscountLogic.sol**

Properties:

* ‘owner’, ‘discountToken’, unspentTokens’: Public variables to store the contract owner, LoyaltyToken contract reference, and UnspentLoyaltyToken contract reference, respectively.
* Customer struct: Defines a structure to store customer information such as name, loyalty token balance, and registration status.

Functions:

* constructor: Initializes the contract by setting the owner and references to the LoyaltyToken and UnspentLoyaltyToken contracts.

Actions:

* Sets the contract owner (owner = msg.sender).
* Initializes references to the LoyaltyToken and UnspentLoyaltyToken contracts.

registerCustomer

Purpose:

* Allows the contract owner to register a customer by storing their name and marking them as registered.

Actions:

* Verifies if the caller is the contract owner.
* Checks if the customer is not already registered.
* Registers the customer by storing their name and setting the registration status to true.

utilizeDiscount.

utilizeDiscount

Purpose:

* Enables a registered customer to utilize a discount by depositing tokens and applying the discount to the purchase amount.

Actions:

* Verifies if the requested discount is not more than 50%.
* Verifies if the customer is registered.
* Calculates the number of tokens required for the given discount and ensures the customer has enough tokens.
* Deposits the required tokens into the UnspentLoyaltyToken contract.
* Emits an event DiscountUtilized with the discount applied.

purchaseItem

Purpose:

* Allows a registered customer to purchase an item by applying the discount and transferring loyalty tokens based on the purchase amount.

Actions:

* Verifies if the customer is registered.
* Calls utilizeDiscount to apply the discount to the purchase amount.
* Calculates the final amount after the discount is applied.
* Transfers loyalty tokens from the owner to the customer based on the purchase amount.
* Emits a Purchase event.

The DiscountLogic contract serves as the core functionality for managing customers, their loyalty tokens, and the process of utilizing discounts while purchasing items. The functions are designed to handle registration, token utilization, and purchase transactions involving loyalty tokens.