pragma solidity ^0.5.0;

import "./IERC20.sol";

/\*\*

\* @dev Optional functions from the ERC20 standard.

\*/

contract ERC20Detailed is IERC20 {

string private \_name;

string private \_symbol;

uint8 private \_decimals;

/\*\*

\* @dev Sets the values for `name`, `symbol`, and `decimals`. All three of

\* these values are immutable: they can only be set once during

\* construction.

\*/

constructor (string memory name, string memory symbol, uint8 decimals) public {

\_name = name;

\_symbol = symbol;

\_decimals = decimals;

}

/\*\*

\* @dev Returns the name of the token.

\*/

function name() public view returns (string memory) {

return \_name;

}

/\*\*

\* @dev Returns the symbol of the token, usually a shorter version of the

\* name.

\*/

function symbol() public view returns (string memory) {

return \_symbol;

}

/\*\*

\* @dev Returns the number of decimals used to get its user representation.

\* For example, if `decimals` equals `2`, a balance of `505` tokens should

\* be displayed to a user as `5,05` (`505 / 10 \*\* 2`).

\*

\* Tokens usually opt for a value of 18, imitating the relationship between

\* Ether and Wei.

\*

\* NOTE: This information is only used for \_display\_ purposes: it in

\* no way affects any of the arithmetic of the contract, including

\* {IERC20-balanceOf} and {IERC20-transfer}.

\*/

function decimals() public view returns (uint8) {

return \_decimals;

}

}