```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract BankAccount {
    address public owner;
    mapping(address => uint256) public balances;

    constructor() {
        owner = msg.sender;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "Only the owner can call this function");
        _;
    }

    function deposit() public payable {
        require(msg.value > 0, "Please deposit a non-zero amount");
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 amount) public {
        require(amount > 0, "Withdrawal amount must be greater than 0");
        require(balances[msg.sender] >= amount, "Insufficient balance");

        balances[msg.sender] -= amount;
        (bool success, ) = payable(msg.sender).call{value: amount}("");
        require(success, "Withdrawal failed");
    }

    function getBalance() public view returns (uint256) {
        return balances[msg.sender];
    }
}
```