|  |
| --- |
| Assignment 4 |
| Dylan Morrissey20075666Secure Programming & Scripting |

Table of Contents

[Requirements 2](#_Toc512464123)

[Use Case Diagram 3](#_Toc512464124)

[Misuse Cases 4](#_Toc512464125)

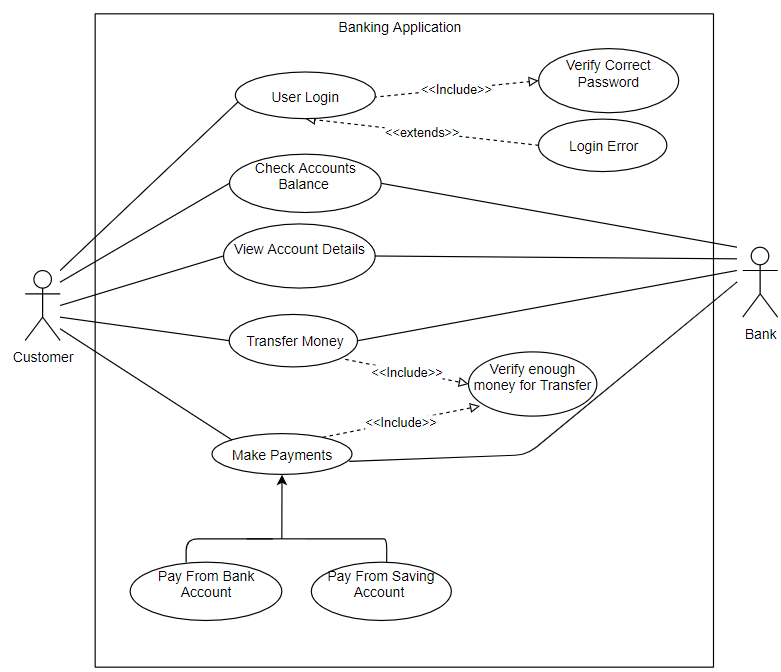
[Mitigation 5](#_Toc512464126)

# Requirements

The requirements for the Banking app are below:

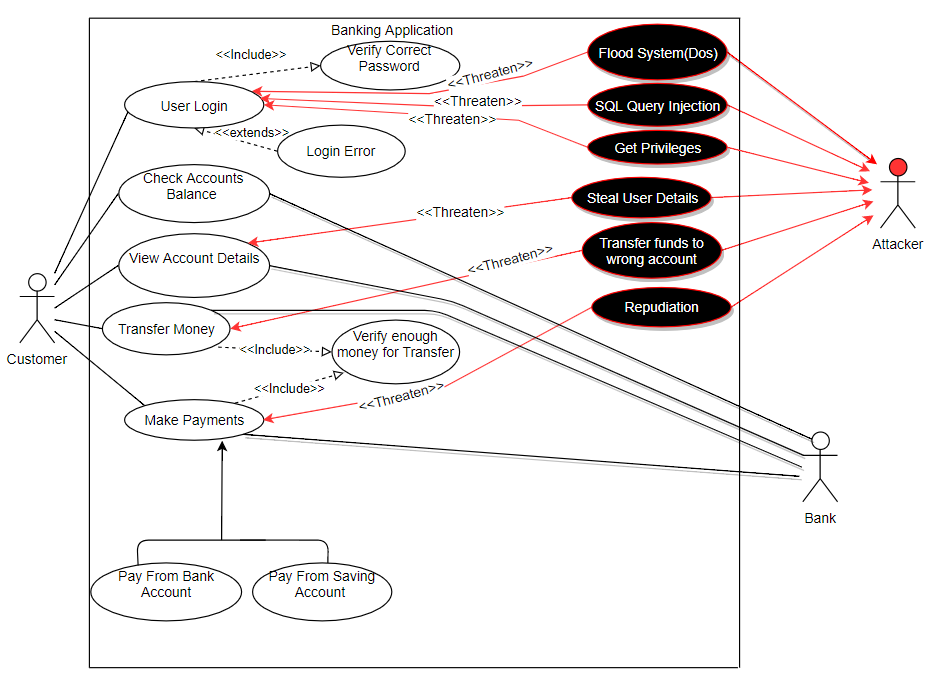
* The ability for the user to open the app and log into their account.
* Verification of the users log in.
* If password entered is correct login is verified
* If password is incorrect the system returns an error.
* Return users details If login details are true.
* Return a login error if details are incorrect.
* Ability for a logged in user to check their bank balance.
* The bank balance is confirmed with the bank and the correct balance for the user is correct.
* Allow the user to view their account details such as Iban and Ibic etc.
* The bank confirms these details for the user are correct.
* Allow the user to transfer money.
* When transferring money, the account must be checked to make sure it has sufficient funds to transfer.
* The Bank also follows up and confirms the transfer.
* The user can have the ability to make payments.
* These payments must also be confirmed to make sure the account has sufficient funds to make a payment.
* The payments are then followed up with the bank.
* The user when making a payment they can pay from their current account.
* Or the user when making a payment can pay from their savings account.

# Use Case Diagram



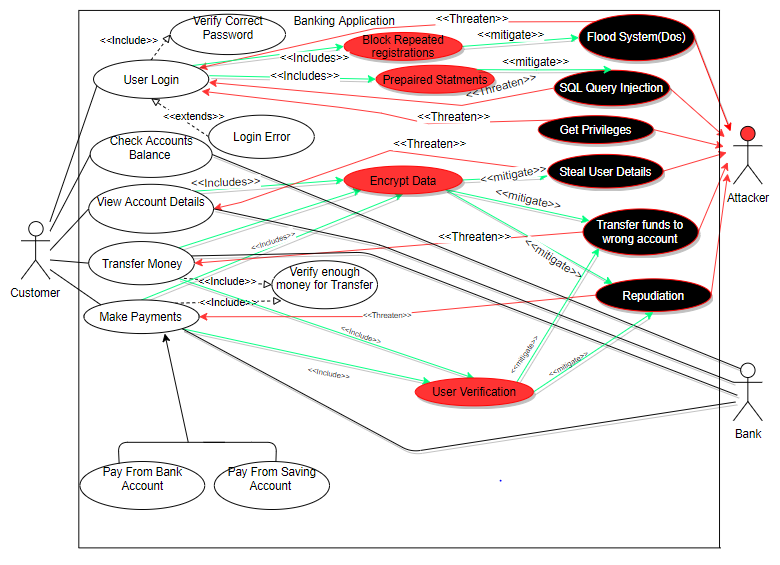
* Customer/user logs into the application.
* They can check their account details and these details are confirmed with the banks.
* The users can Check their bank balance, and this is confirmed with the Bank.
* The User Can Transfer money to another account.
* The User can make on going payments such as an electricity bill etc.
* They Can choose which account the payments come from.

# Misuse Cases



* Flood System(DoS) threatens the app by filling up the memory until it crashes causing a DOS attack.
* SQL query retrieve more data than they are permitted to or add unauthorised users.
* Get privileges to a higher level to where they can edit the app or cause serious damage.
* If the were to steal the user details this could be sold on the dark web or the attacker could pretend to be the user.
* An attacker could intercept a money transfer and change the destination of the transfer to another account not intended by the user.
* An attacker could make a payment and then claim to have not made that payment and this is repudiation.

Mitigation



* The App could block repeated registrations and thus reducing the risk of a DOS attack.
* A prepared statement could be made so only certain enters are permitted reducing the risk of a SQL injection attack occurring.
* The Encrypted data will make is harder for an attacker to decipher the data.
* User Verification for transactions could be required to show the user made the payment so they can’t deny making the payment.