HAYSTAQ

Opdracht Workshop Unit Test

Banking

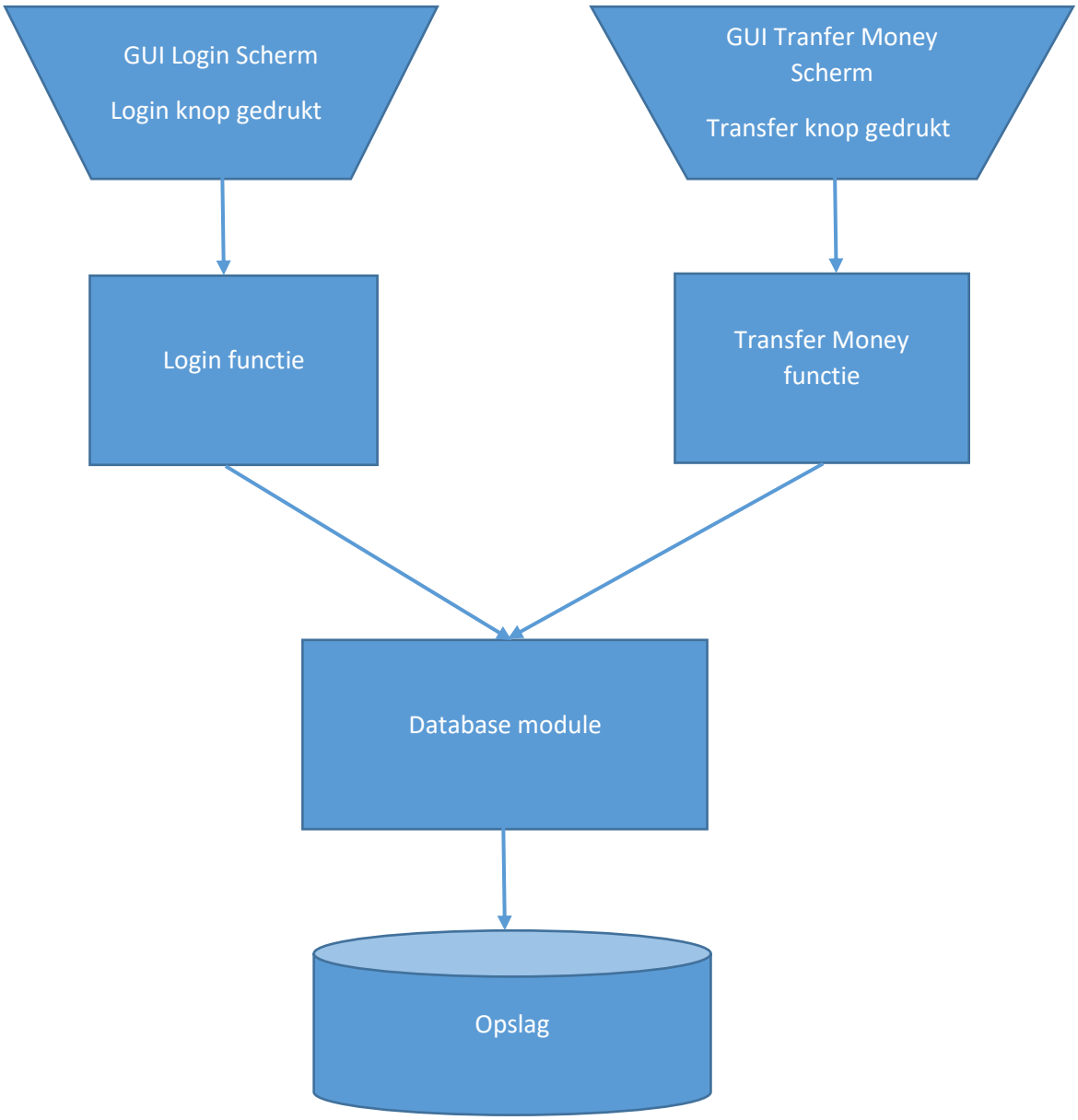
EXPERTS IN AUTOMATED SOFTWARE TESTING

Haystaq
Huize Muyserick
Maurick 1
5261 NA Vught

T 06 53 77 94 92
E info@haystaq.nl
W www.haystaq.nl

NL33 RABO 0315 5905 13
BTW NL.8571.45.769.B.01
KvK 67715478

Design


GUI Login Scherm
Login knop gedrukt
GUI Tranfer Money Scherm
Transfer knop gedrukt
Login functie
Transfer Money functie
Database module
Opslag

Voorbeeld Opdracht en Uitwerking

Pseudo Code

1. Example

Function: example
Input: value
Output: true/false

```
bool function example(value)
        if IsPositive(value) then
                return true
        else
                return false
        end if
end function
```

Unit Test Code example

1. example_ShouldReturnTrue_WhenValueIsPositive

Arrange
IsPositive should return true when value > 0 (mock)
Act
result = example(10)
Assert
result equals true

Opdracht 1 | Login

Pseudo Code

1. Login
Deze module (met 1 functie) wordt aangeroepen wanneer vanuit de GUI de login knop wordt gedrukt. Ook is er een controle dat de gebruiker niet meer dan drie keer heeft geprobeerd in te loggen.

Function: login
Input: username, password
Output: validation (true/false), errormessage (in case of false)

```
bool function login(username, password, out message)
        if (database.userexists(username)) then
                attemps = database.attempts(username)
                if (attempts greater than 3) then
                        message = “too many login attempts please wait”
                        return false
                end if
                if (database.validUsernamePassword(username, password)) then return true
                database.increaseAttempts(username)
        end if
        message = “invalid username or password”
        return false
end function
```

Hints

Database moet worden gestubd voor de Unit Test dus je moet er wel voor zorgen in je Arrange dat deze stukken de juiste data teruggeven.

Opdracht 2 | Transfer money

Pseudo Code

1. Transfer money
Deze module zal een gebruiker geld laten overschrijven van zijn/haar rekening naar een andere rekening. Het saldo van de rekening en de geldigheid van de ontvangende rekening zullen worden gecontroleerd.

Function: transferMoney
Input: account number, amount, receiving account
Output: validation (true/false), error message (in case of false)

```
bool function transferMoney(accountNumber, amount, receivingAccount, out message)
        balance = database.getBalance(accountNumber)
        valueToReturn = true
        if (balance >= amount) then
                if(!database.accountExists(receivingAccount) then
                        message = “receiving account is unknown”
                        valueToReturn = false
                end if

                if(database.accountExists(receivingAccount) then database.transferFunds(amount,
                        accountNumber,
                        receivingAccount)
        else
                message = “insufficient funds”
                valueToReturn = false
        end if
        return valueToReturn
end function
```

Opdracht 3 | Database

Pseudo Code

1. Database
Deze module wordt gebruikt door de Login en Transfer money modules. De module bevat allerhande functies die gebruikt worden voor het lezen en schrijven naar de database.

Function: userExists
Input: username
Output: validation (true/false)

```
bool function userExists(username)
        valueToReturn = false
        try
                connecttodatabase
                databaseretrieve all lines for username
                if numberoflines = 1 then valueToReturn = true
        exception
                throw exception
        return valueToReturn
end function
```

Function: attempts
Input: username
Output: attempts

```
int function attempts(username)
        try
                connecttodatabase
                databaseretrieve all lines for username
                get attempts from line
                return attempts
        exception
                throw exception
end function
```

Function: validUsernamePassword
Input: username, password
Output: validation (true/false)

```
bool function validUsernamePassword(username, password)
    try
        connecttodatabase
        databaseretrieve all lines for username
        get dbpassword from line
        return dbpassword==password
    exception
        throw exception
end function
```

Function: increaseAttempts
Input: username
Output: none

```
void function increaseAttempts(username)
    try
        connecttodatabase
        databaseretrieve all lines for username
        get attempts from line
        attempts = attempts + 1
        databasestore line with increased attempts and add attempttimestamp
    exception
        throw exception
end function
```

Function: cleanAttempts is run every minute
Input: none
Output: none

```
void function cleanAttempts()
        try
                connecttodatabase
                databaseretrieve all lines with > 3 attempts
                foreach line
                        get attempttimestamp
                        if attempttimestamp is older than 60 minutes then attempts = 0
                                databasestore line with attempts = 0
                        end if
                next
        exception
                throw exception
end function
```

Function: subtractFunds

Input: amount, senderAccount, receiverAccount
Output: none

```
void function subtractFunds (amount, senderAccount, receiverAccount)
        try
                connecttodatabase
                databaseretrieve all lines for senderAccount
                get balance from line
                newBalance = balance - amount
                databasestore line with newBalance
                databasestore line with -amount, senderAccount, receiverAccount
        exception
                throw exception
end function
```

Function: addFunds
Input: amount, receiverAccount , senderAccount
Output: none

```
void function addFunds(amount, receiverAccount, senderAccount)
        try
                connecttodatabase
                databaseretrieve all lines for receiverAccount
                get balance from line
                newBalance = balance + amount
                databasestore line with newBalance
                databasestore line with amount, receiverAccount, senderAccount
        exception
                throw exception
end function
```