**

**Implementation document**

M.G. den Hollander

Student number: 3803554

Fontys Hogescholen

ICT & Software Engineering

Version: 1.2

#### Version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** | **State** |
| 1 | 17-02-2023 | M.G. den Hollander | Created the document and added styling. | Concept |
| 1.1 | 12-04-2023 | M.G. den Hollander | Started working on documenting the implementation | Concept |
| 1.2 | 08-06-2023 | M.G. den Hollander | Finished the implementation document | Complete |

# Table of contents

[Table of contents 3](#_Toc137045127)

[1 Introduction 4](#_Toc137045128)

[2 Implementation 5](#_Toc137045129)

[UC-01 Checking unprocessed financial transactions 5](#_Toc137045130)

[UC-02 Printing documents 7](#_Toc137045131)

[UC-03 Data collection and processing 7](#_Toc137045132)

[UC-04 Converting and sending documents 12](#_Toc137045133)

# Introduction

Through the research conducted during the internship, it was determined that the best option for the company is to implement a different technique in order to view if it is possible to phase out the robot.

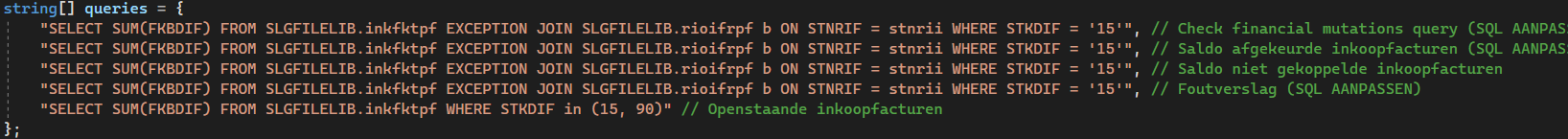
Once it was clear that an alternative to the robot had to be set up, requirements were made so that a prototype could be created. This implementation document contains all of the implementation steps that the alternative to the robot performs. This implementation will be linked to the requirements and use cases that the robot already performs, and the alternative will need to perform as well. All of these requirements and use cases can be found in the [analysis document](https://tentive-my.sharepoint.com/personal/marc_den_hollander_tentive_nl/Documents/Bureaublad/Portfolio%20Marc%20den%20Hollander/1.%20Analysis/Analysis%20document.docx).

It is hoped that this document will help the reader to better understand the implementation and will hopefully make it easier to make adjustments in the future when deemed necessary.

# Implementation

## UC-01 Checking unprocessed financial transactions

This use case involves checking if certain rules are empty on the AS400. Previously, the robot performed this check visually. However, an alternative approach simplifies the process by using a straightforward SQL statement that should return 0. The code for this approach is as follows:

* Since there are multiple use cases that utilize SQL statements, it was decided to create an array of these queries. By consolidating the SQL statements into an array, it provides a convenient and organized way to manage and execute the queries. This approach enhances code reusability and maintainability, allowing for efficient implementation and management of multiple use cases that involve SQL queries. Screenshot bijwerken
* To execute the first query, the following code is executed by opening a connection and retrieving number 1 from the array of queries. The result of this operation is stored in a recordset. This object allows for retrieving and manipulating data from a database.



* A timer is started. This is because the program needs to check every 15 minutes for a duration of 1 hour whether the financial transactions have been processed in the meantime. The count goes up for every attempt. After trying 4 times, the program moves on.





* If there are no more financial transactions, the application can proceed with the remaining steps of the process right away, as described in the following use cases.



* If there are still financial transactions after an hour, the application needs to initiate a script named “mutations.vbs” to notify the relevant department. Afterward, the process will be halted.

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

* The script that is initiated is fairly simple. The variables are defined at the beginning, followed by retrieving the current date, which is needed in the email header. The content of the text is defined based on the email body, which is a warning that the lines are not empty. Then, a new email is generated, all the necessary information is filled in, and finally, the email is sent.

Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving

* The result of this code looks like this on Outlook, marking the completion of this use case.

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

## UC-02 Printing documents

Deze use case houdt in dat er meerdere documenten geprint moeten worden op de AS400. Deze documenten worden geprint door programma’s die op de AS400 draaien, en kunnen worden aangeroepen als een methode. De code is als volgt:

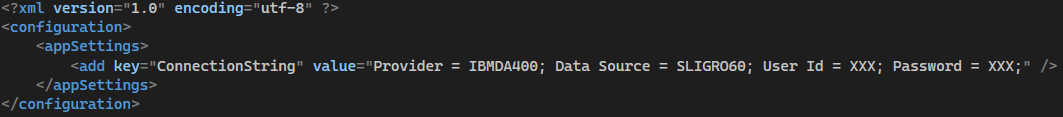
* De procedures worden aangemaakt als een string. Belangrijk om te noemen is dat er voor een document de datum van de volgende dag nodig is, waarvoor er een datetime variabele aangemaakt is en er 1 dag bij wordt opgeteld

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

## UC-03 Data collection and processing

To execute this use case, multiple scripts have been developed and executed by the C# console application. The application retrieves the required values from the AS400 using SQL statements, facilitated by an ADODB (ActiveX Data Objects for .NET) connection to the system. The code for this use case is as follows:

* An App.config file has been created to store the connection string required for connecting to the AS400. By organizing the connection string in a separate configuration file, sensitive information such as usernames and passwords can be safeguarded. For security purposes, the username and password are hidden from this screenshot. 
* The connection string from the previous step is retrieved, and the SQL statements are defined. These SQL statements are then added to the already existing array made during the first use case. Next to that, an array is created to store the results in a later step. Screenshot bijwerkenAfbeelding met tekst, schermopname, Lettertype

  Automatisch gegenereerde beschrijving
* In a "try-catch" method, the connection to the AS400 is initiated using the retrieved connection string, and a recordset is created for each query. This data is then stored as results in the array created in the previous step. Afbeelding met tekst, schermopname, software

  Automatisch gegenereerde beschrijving
* Lastly, a script is invoked, which is described later in this chapter. This script sends the retrieved results for further processing in an Excel document. Afbeelding met tekst, schermopname, Lettertype

  Automatisch gegenereerde beschrijving

The scripts required to perform the subsequent steps of this use case are written in Visual Basic (VBS) and Visual Basic for Applications (VBA). The first script, named "automation.vbs," written in VBS, performs the following tasks:

* All necessary variables are initialized at the beginning, and then the script checks for the values that are passed from the C# console application to this script. Afbeelding met tekst, schermopname, Lettertype

  Automatisch gegenereerde beschrijving
* An instance of Excel is launched, and the pre-existing Excel file named "dagaansluiting" is located. This file contains a macro written in VBA that will place the passed variables in the right positions. Further details about this macro will be described later in this chapter. Afbeelding met tekst, schermopname, Lettertype

  Automatisch gegenereerde beschrijving
* Finally, the script performs a check to ensure that Outlook is running. If not, it is initiated, and then the email script is executed, which sends the processed Excel document to the appropriate recipients.Afbeelding met tekst, Lettertype, schermopname, algebra

  Automatisch gegenereerde beschrijving

As mentioned earlier, an Excel document has been created to serve as a report for the finance department of Sligro. Here is an overview of its structure:

Afbeelding met schermopname, tekst, lijn, Perceel

Automatisch gegenereerde beschrijving

For the sake of confidentiality, the amounts in the screenshot have been set to 0 euros. To populate this report with the retrieved results, a macro has been implemented using VBS. This macro, named "DailyQuery," performs the following tasks:

* It initializes the necessary variables and retrieves the current date. To ensure clarity, the corresponding values for each column are clearly indicated. Afbeelding met tekst, schermopname, Lettertype

  Automatisch gegenereerde beschrijving
* The total amount is calculated by summing up the values, excluding the "Saldo openstaande inkoopfacturen”.

Afbeelding met tekst, Lettertype, wit, schermopname

Automatisch gegenereerde beschrijving

* The difference is calculated, which is an essential step for subsequent handling via email. Further details on this will be addressed in a later part of the script. Afbeelding met tekst, Lettertype, schermopname, lijn

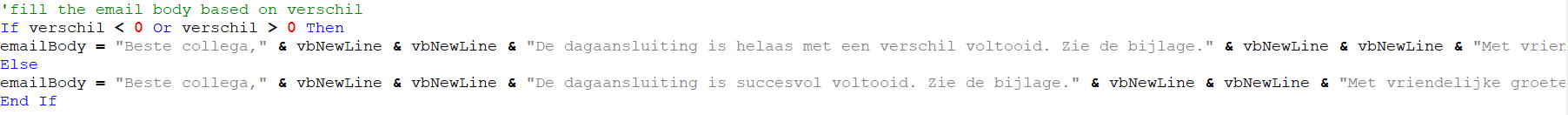
  Automatisch gegenereerde beschrijving
* This step involves placing the values in the right positions within the document. When executed the next day, new values will be inserted on the following line, along with the corresponding date. Afbeelding met tekst, Lettertype, nummer, schermopname

  Automatisch gegenereerde beschrijving
* In the final step of this macro, the difference value is stored in a .txt file. The reason behind this is that the value cannot be directly returned from the Excel macro for later handling via email. Hence, a simple approach was adopted by writing the value to a .txt file. This value is overwritten daily. Afbeelding met tekst, schermopname, Lettertype, lijn

  Automatisch gegenereerde beschrijving

The final script, named "mail.vbs," performs the action previously mentioned, which involves sending an email containing the complete “dagaansluiting” report. The script carries out the following tasks:

* All necessary variables are created at the beginning. Next, the difference calculated in the Excel macro is retrieved from the "value.txt" file. If there is an actual difference, it will affect the content of the email in a later step of this script. Additionally, the current date is retrieved, which needs to be mentioned by default.Afbeelding met tekst, schermopname, Lettertype

  Automatisch gegenereerde beschrijving
* As mentioned in the previous step, the email content is adjusted if there is a difference calculated. This notifies the recipients who then need to perform their own manual actions based on this difference.
* The final step of this script fills in the recipient, sets the email subject to the retrieved date, and attaches the Excel document to the email. After this, the email is sent, marking the completion of the use case. Afbeelding met tekst, Lettertype, schermopname

  Automatisch gegenereerde beschrijving
* The result of this script looks like thisAfbeelding met tekst, Lettertype, schermopname

  Automatisch gegenereerde beschrijving

## UC-04 Converting and sending documents