

TENTIVE



Sligro Food Group

Design 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 designs.	Concept
1.2	28-04-2023	M.G. den Hollander	Finished architecture of RPA solution.	Complete

Table of contents

Table of contents.....	3
1 Introduction	4
2 System Context Diagram.....	5
3 Container diagram.....	6
4 Architecture RPA solution.....	7



1 Introduction

During the graduation internship at Sligro, an important step in the development process was the creation of a design document. This document serves as a blueprint for the alternative to the robot and contains information about the designs and diagrams of all components of the project.

The ultimate goal of this document is to provide a detailed guide that will help in the implementation phase of the project. It is hoped that this document will help the reader to better understand the structure of the alternative to the robot, which will help in a later phase of implementation.

By providing a clear and structured overview of the design, the document aims to ensure that everyone involved in the project has a shared understanding of its objectives and requirements.



2 System Context Diagram

This chapter explains the system context of this project. The diagram below (figure 1) shows the entities and their interactions.

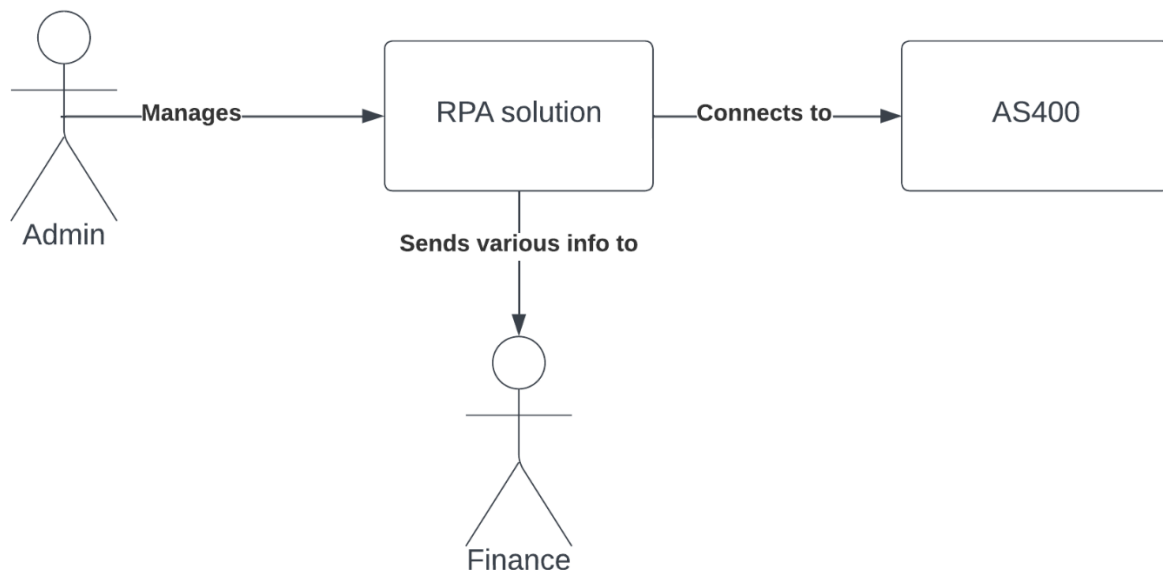


Figure 1 System Context Diagram

The global system used is the AS400. For more information about this system, please refer to the first research question in the research document to find out exactly what this system is. This AS400 system is used for the entire Sligro business operations, and for this project specifically, a process is being automated for the finance department of Sligro. In addition, it is managed by the IT department administrator. This means that both an "Admin" and "Finance" actor are present in this overview. This process is automated by an RPA tool used by the company, which involves a "robot" that goes through a certain process for the finance department called the "Dagaansluiting". This means that the RPA solution sends various information to the financial department like reports, files and emails. The alternative will have to do exactly the same thing using multiple scripts in the background. The chapter on the architecture of the RPA solution goes into more depth on this topic.

3 Container diagram

This chapter outlines the various containers of the project. The diagram below (Figure 2) explains how everything is divided into different containers.

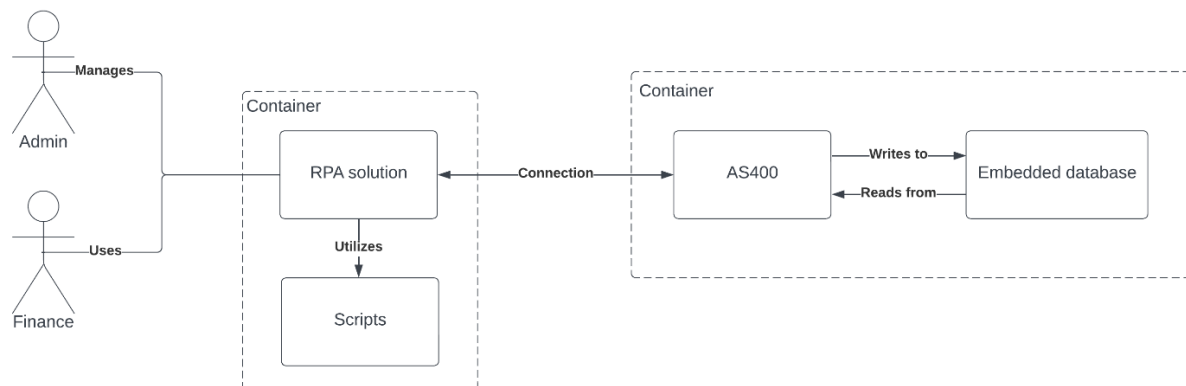


Figure 2 Container Diagram

Upon further exploration of the different containers, it can be observed that the RPA solution utilizes multiple scripts. More detailed explanations about these scripts are provided in the final chapter. These scripts are used to automate tasks within the company's process. The RPA solution itself does not rely on a database since there is no need for data storage. The data retrieved from the AS400 system is exclusively processed within Excel. Additionally, the AS400 system makes use of an embedded database to store all relevant information.

4 Architecture RPA solution

In this chapter, the architecture of the alternative to the current RPA will be shown. Figure 3 provides an overview of the different components along with further explanation down below.

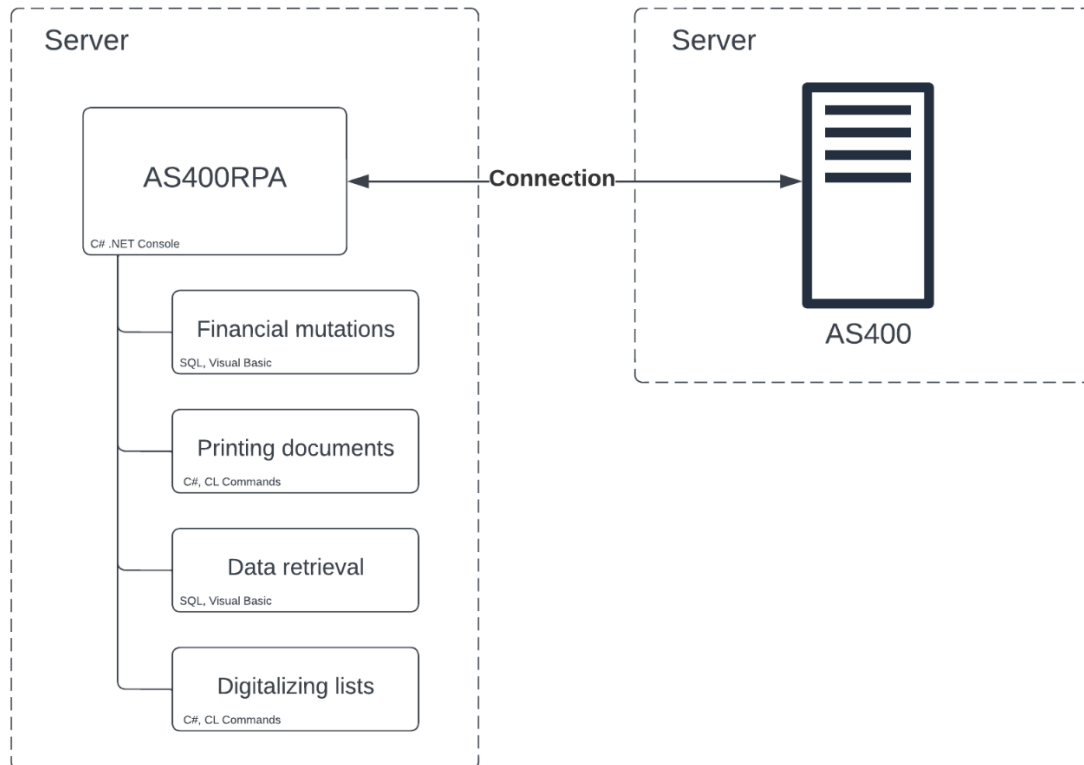


Figure 3 Architecture

The alternative consists of multiple scripts that will be executed by a global C# .NET Console (version 6) application called "AS400RPA." The first component is a check for any active financial transactions on the AS400. If there are any, an email needs to be sent, and every 15 minutes until 8 a.m., the system checks if these transactions have been processed. If they haven't been processed yet, the process cannot proceed. This component combines an SQL statement to check for specific empty records and Visual Basic for sending emails.

The second component involves printing documents on the AS400 for the next step in the process. This is done using C# and a Command Line Command, similar to invoking a method with parameters from the C# application to the AS400.

The third component focuses on retrieving data from spool files on the AS400 by using SQL statements and processing this data into an Excel document. Once the Excel document is populated, it is then sent via email. This component utilizes a combination of Visual Basic and Visual Basic for Applications.

The fourth and final component deals with digitizing lists created in the second component. This requires invoking an AS400 procedure, similar to what was done in step 2. In this procedure, the lists are converted to PDF files and sent to the appropriate location on a network drive.

