Case Piharati Asiakasprojektin automatisointi forms tarjouspyynnosta käyttäen Microsoft power automate sovellusrobotiikkaa

Tomi Palomäki

2023-01-31

Contents

1 Case Piharati Customer project automation from request for quotation forms using Microsoft power automate application robotics

1.1 Introduction

This project aims to automate the process of customer project management for Piharati by using Microsoft Power Automate and Forms. By submitting a form, customers will be able to provide their information which will be processed by the Power Automate flow and used to update an Excel template and change log.

1.2 Requirements

- Microsoft Power Automate
- Microsoft Forms
- Microsoft Excel

1.3 Flow Overview

1. Customer submits information through a Microsoft Forms form.
2. The form response is collected and stored in a variable.

3. A variable is created to store the information as a string value in the format specified in the form.
4. A search is conducted in the Excel template using the stored string value to find the corresponding customer information.
5. The Excel template is updated with the customer information.
6. The change log is updated with a new entry indicating that a quote request has been received.
7. A copy of the template is made and renamed according to the customer order.
8. The file path is initialized as a string variable.
9. The file names in the specified folder (template folder) are processed and renamed based on the customer order.

1.4 Detailed Explanation

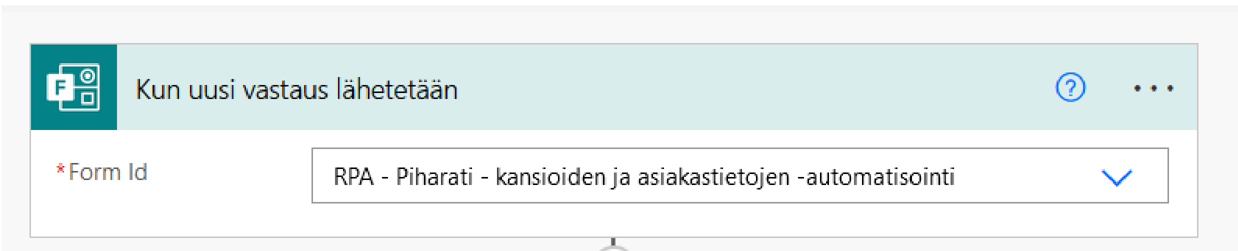

Figure 1: Forms link

With form link see Figure 1, the data from the filled form template is picked up for use in the program code.

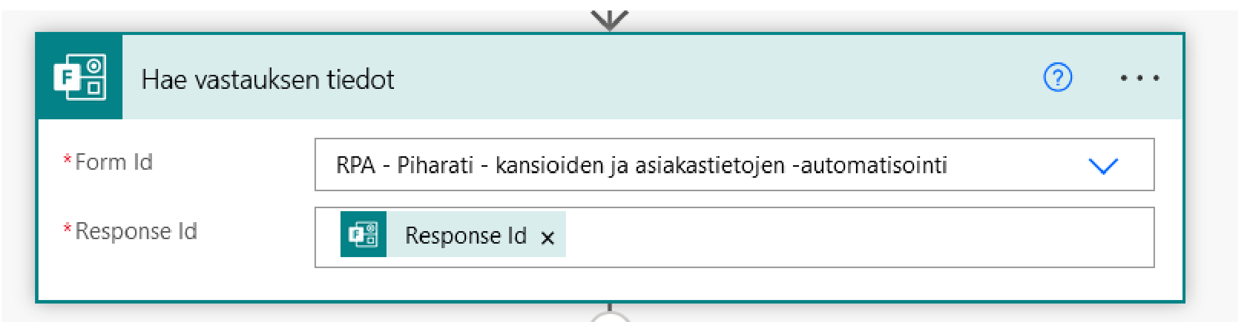

Figure 2: Retrierving data

Retrieving answers data for future reference (Figure 2).

Create a variable to store the form response data collected from the form, and create a string value based on the value field, which is stored in the variable(Figure 3).

Create a variable that will be used later to search for data in the excel template that corresponds to the string in the variable, based on the response collected from the form (Figure 4).

Update the new customer information to the template base as shown in the Figure 5. The complete variable names are not visible in the image, so the complete variable names are listed below in this code block.

key value= asiakkaan_nimi_puh_template

column2 = Asiakkaan nimi_puh

In this block, the key value variable is searched for the corresponding string in the Column2 column of the Excel file and updates the Asiakkaan_nimi_puh information in the column2 variable of this row's Column2 column.

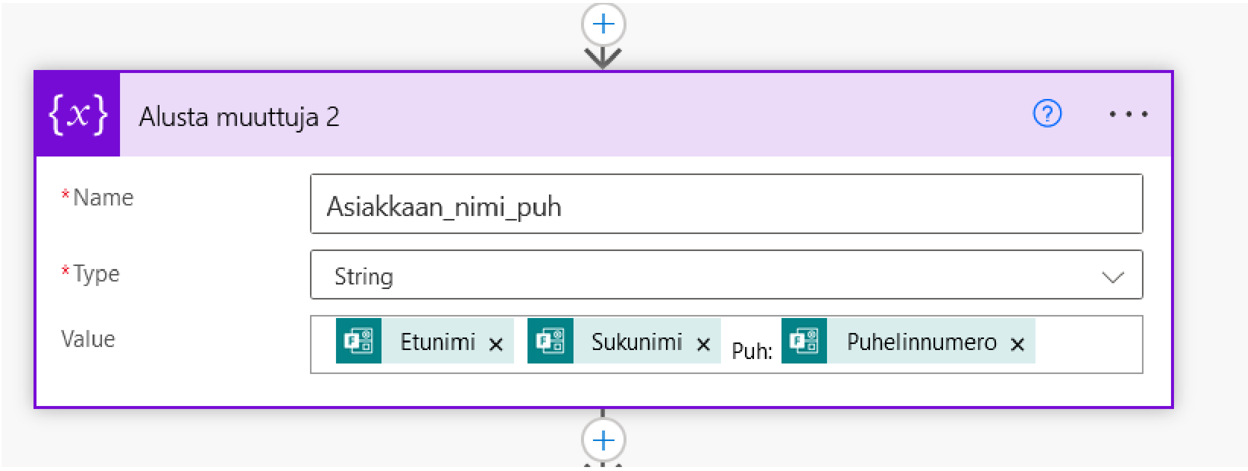

Figure 3: Creating variable

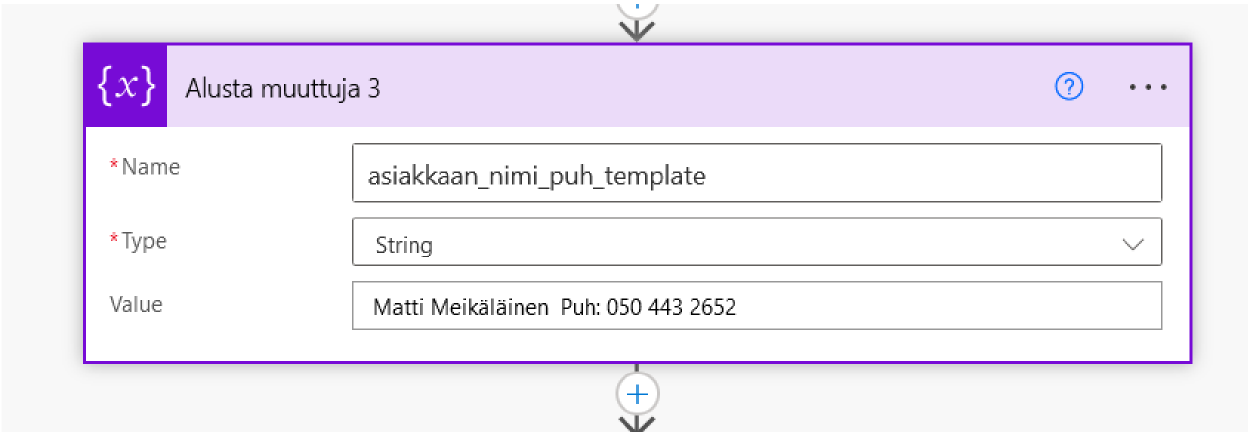

Figure 4: Creating template variable

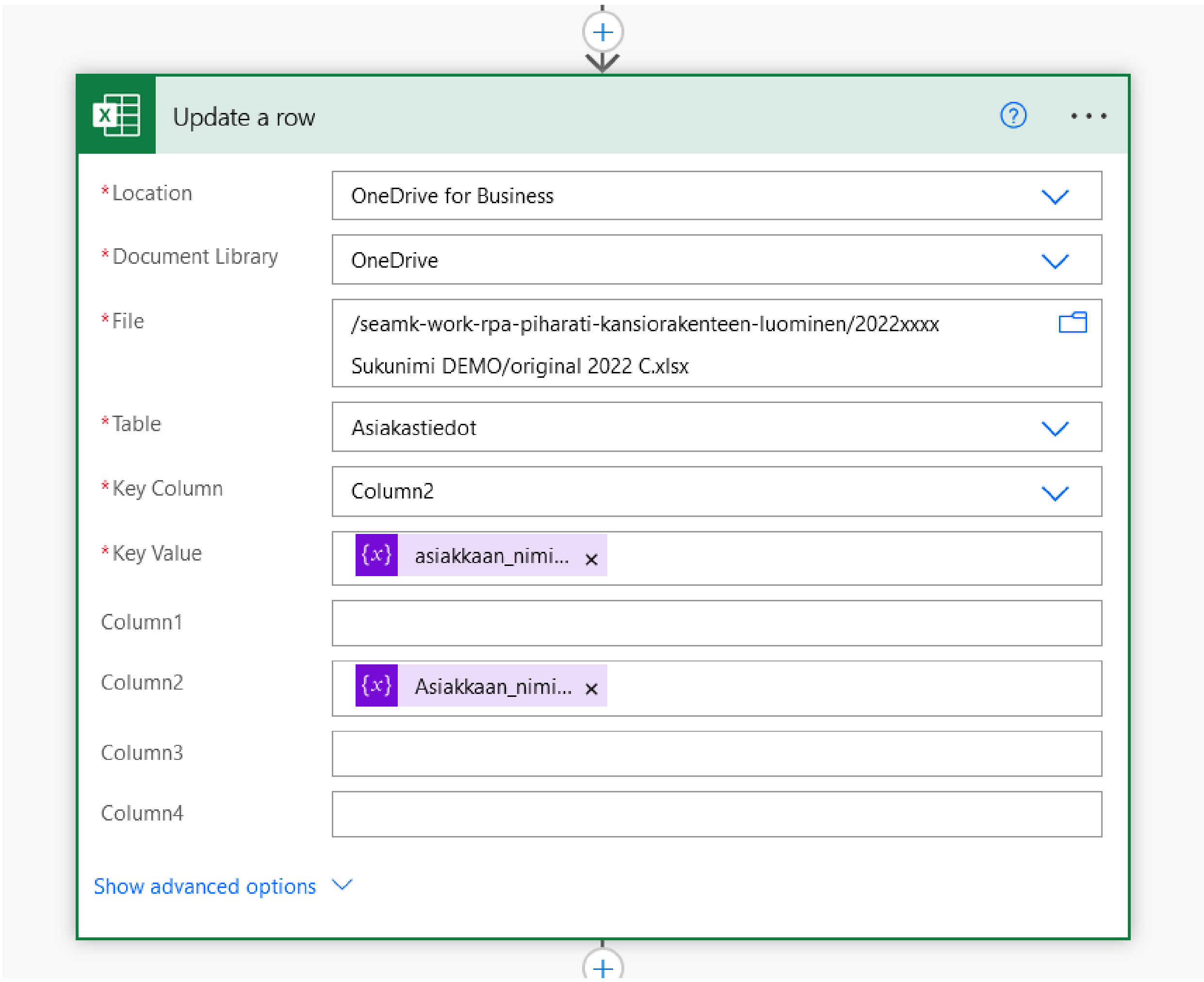

Figure 5: Update customer information

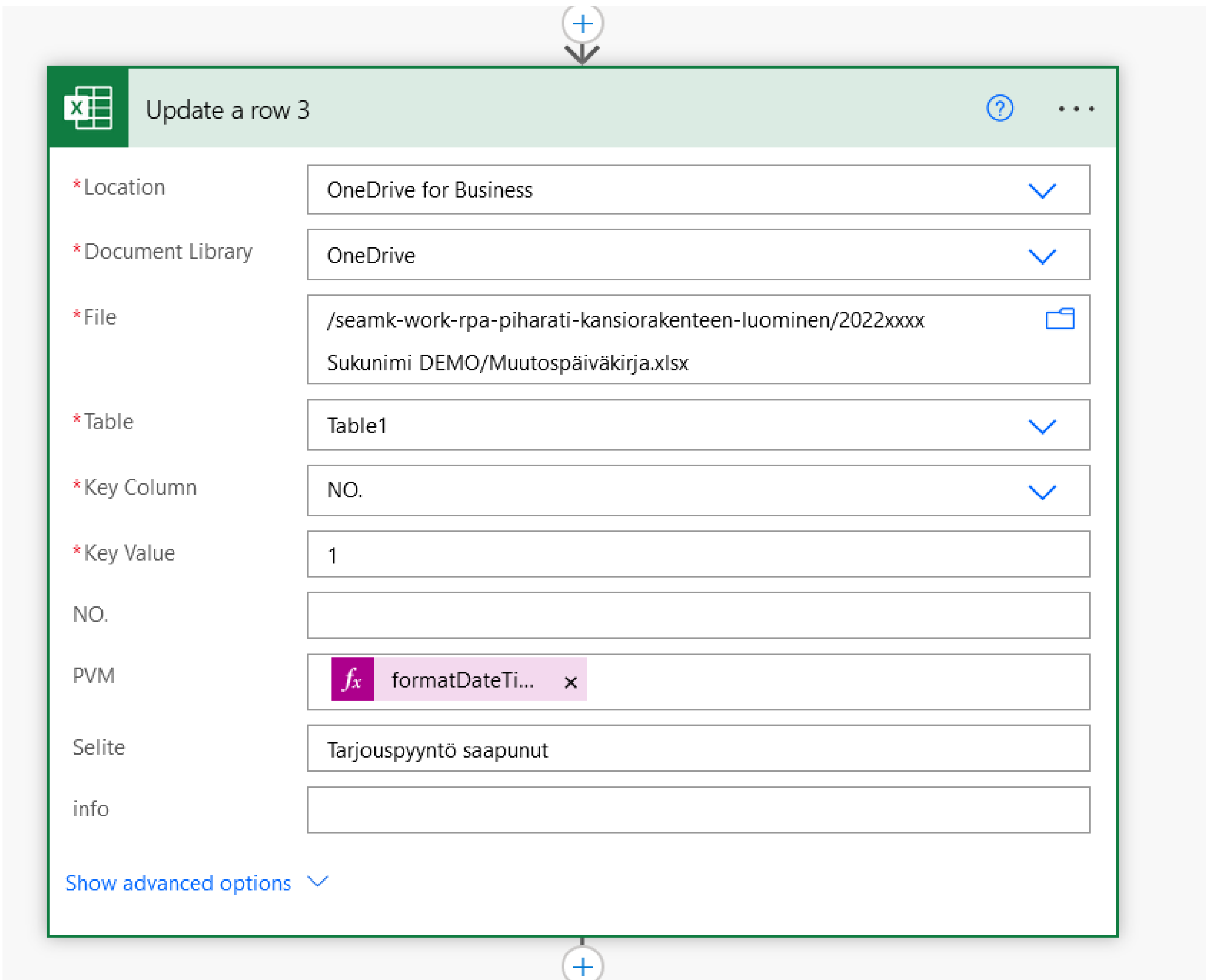

Figure 6: Change log

The update to the change log file is performed using the “update a row” program block. Here, the key-value field number 1 is searched in the NO. column of the Excel file and the text “Quotation request received” is updated in the Description column (Figure 6).

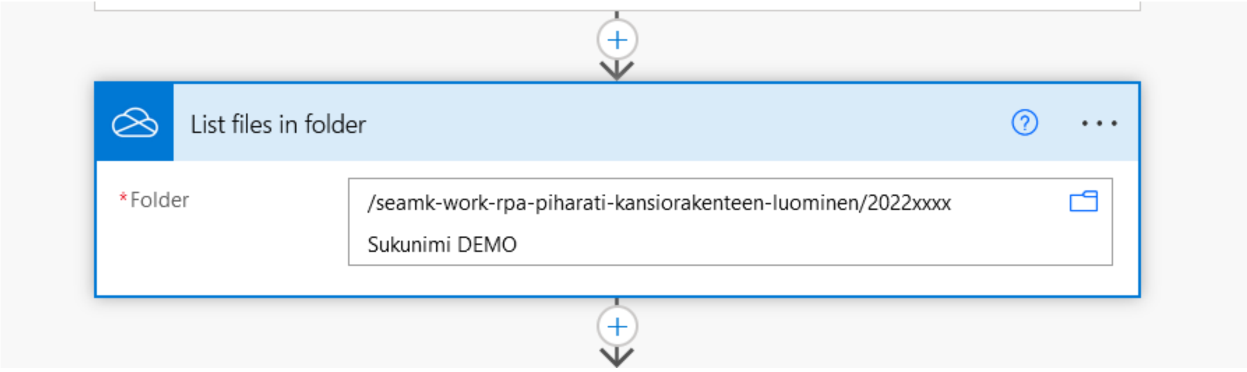

Figure 7: Pick names

The names of the files in the specified folder are picked up. In this case, the template folder. A copy is made from the template folder and the names are changed according to the customer order (Figure 7).

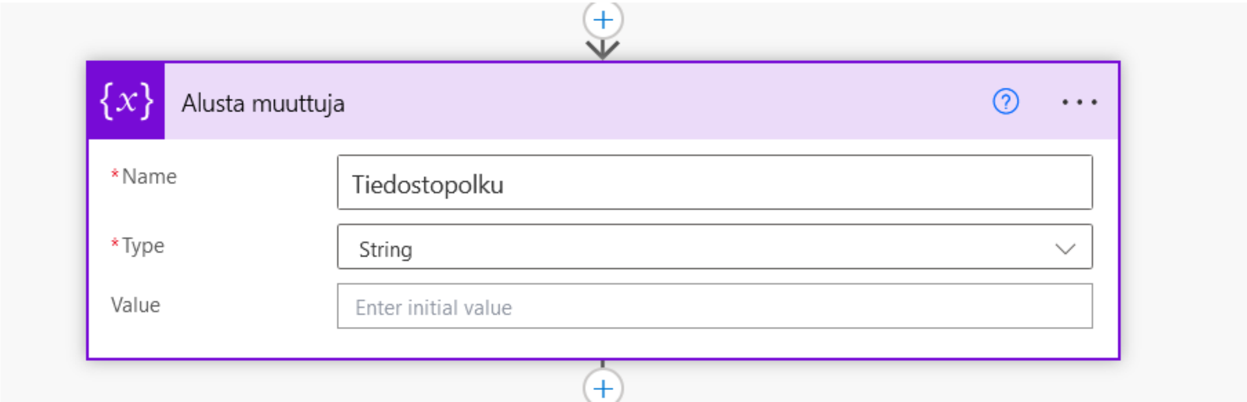

Figure 8: File path

Initialize a file path variable named “file_path” defined as a string (String) (Figure 8).

The “Apply to each 2” block is used to loop through all the file names that “Value” refers to. “Value” refers to the file names obtained in the previous “List files in folder” block. This block opens up the “Apply to each 2” block regarding the section where we determine if the “filename” is a folder. We loop through all the names and either create a folder or copy the file, depending on whether it is a file or a folder in the template folder (Figure 9).

Decision tree (Figure 10).

When it is a folder, a helper file is created to enable the creation of the folder, and this helper file is immediately deleted after the action is performed. The folder is created with the name specified by the forms customer information (Figure 11).

When it is a file, The file is copied to a folder named according to the customer information from the forms, and if the folder does not exist it will be created automatically (Figure 12).

Update back to the original template (Figure 13).

Search for:key value = asiakkaan nimi_puh

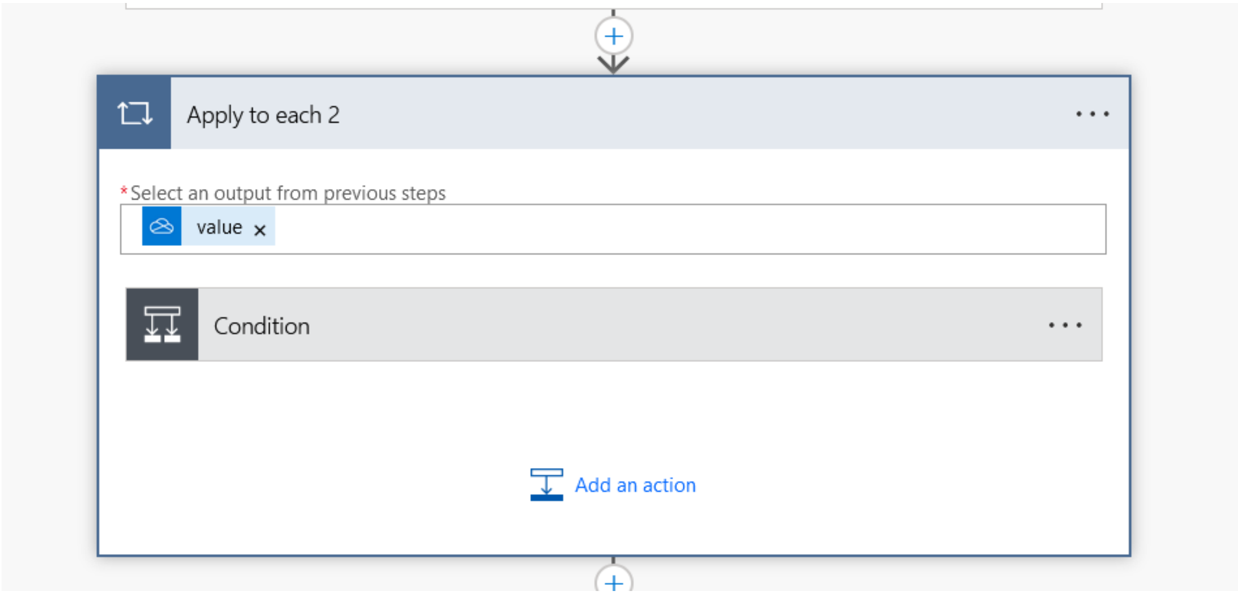

Figure 9: Apply to each

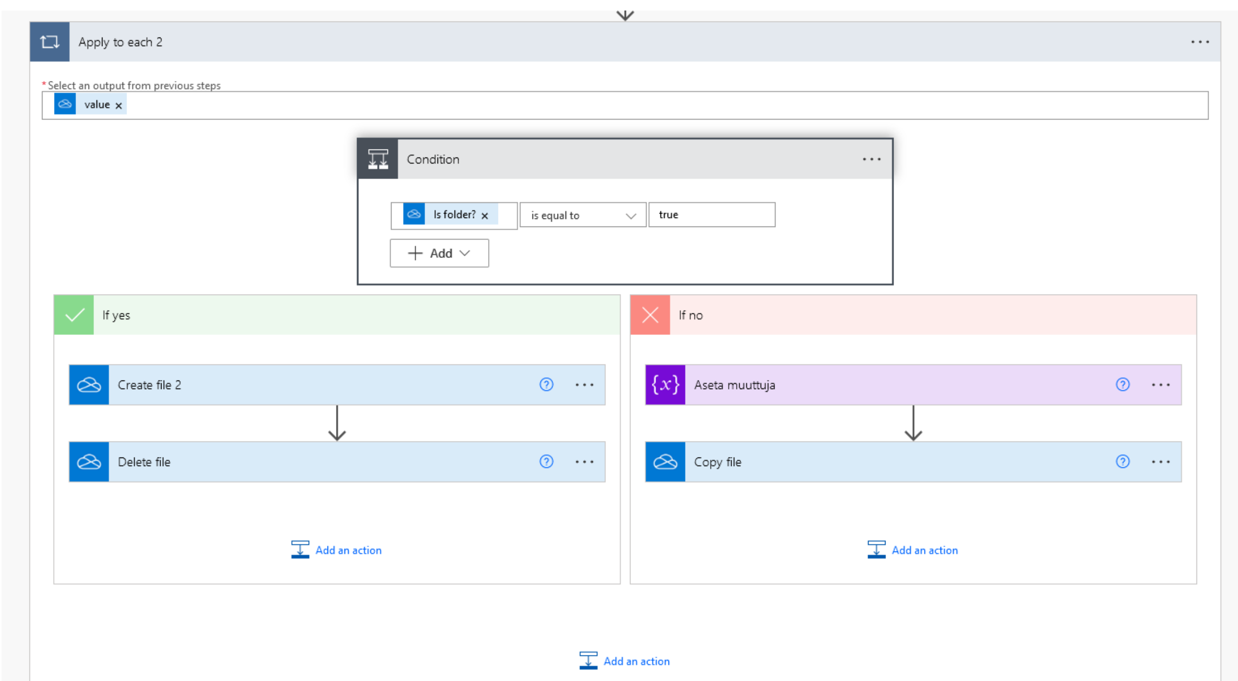

Figure 10: Decision tree (apply to each) opened

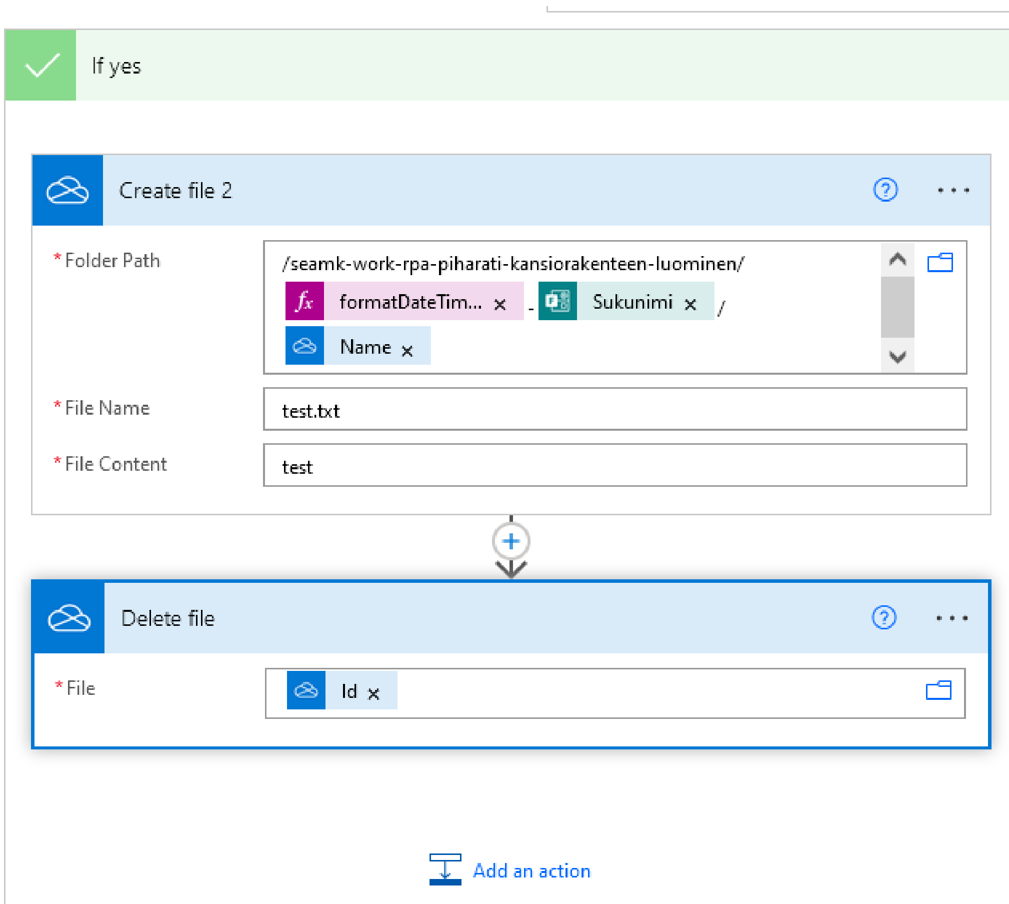

Figure 11: Helper file creation

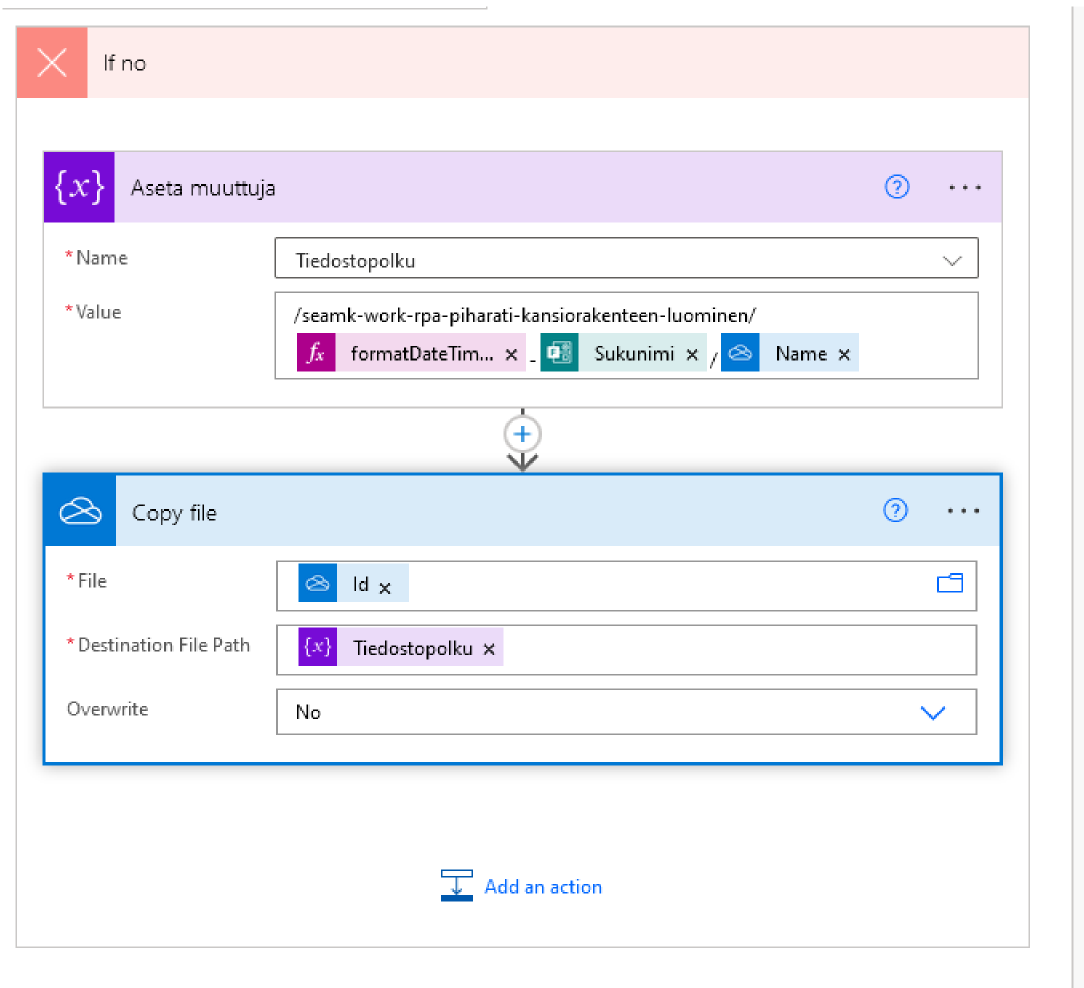

Figure 12: Copy files

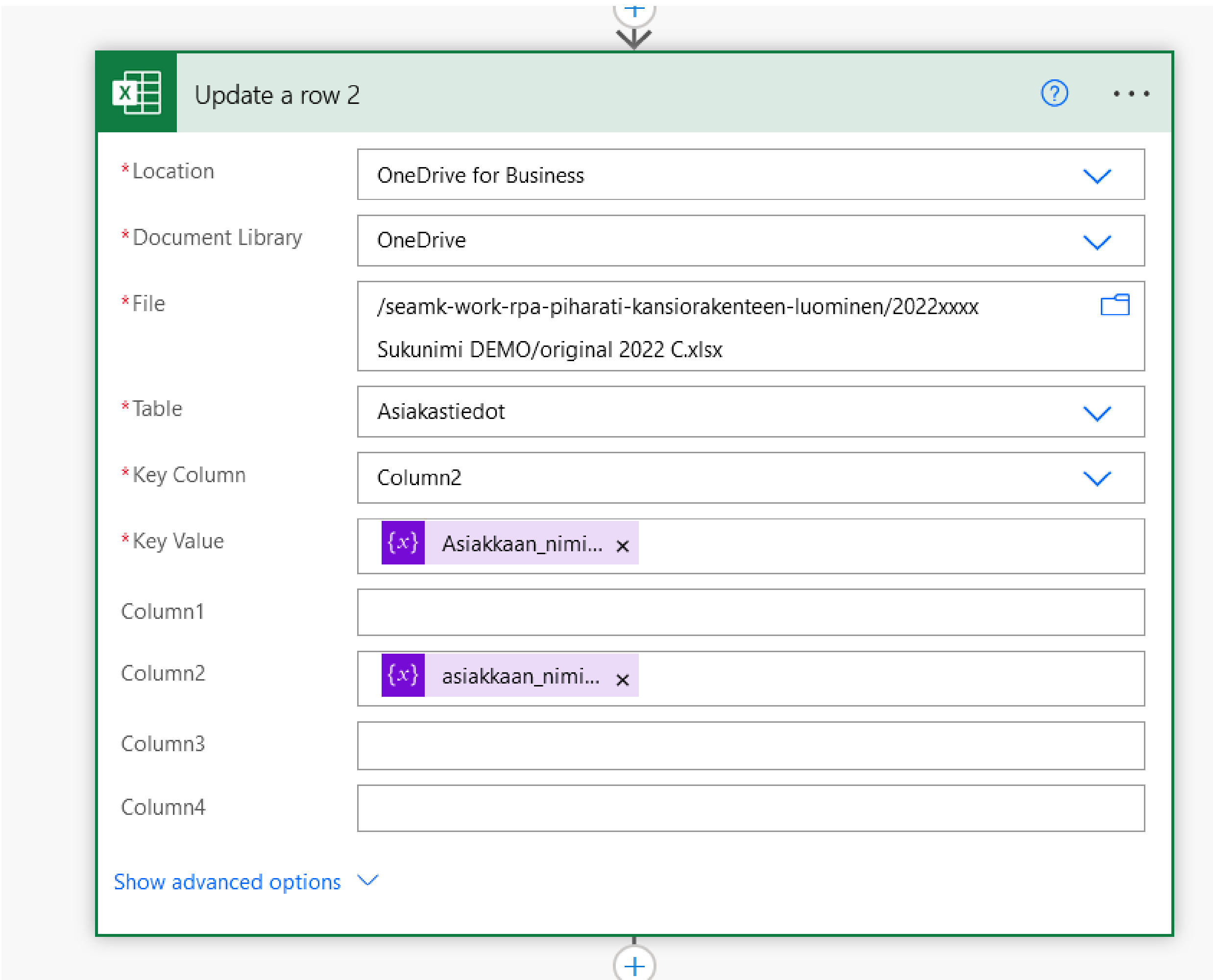

Figure 13: Updating row

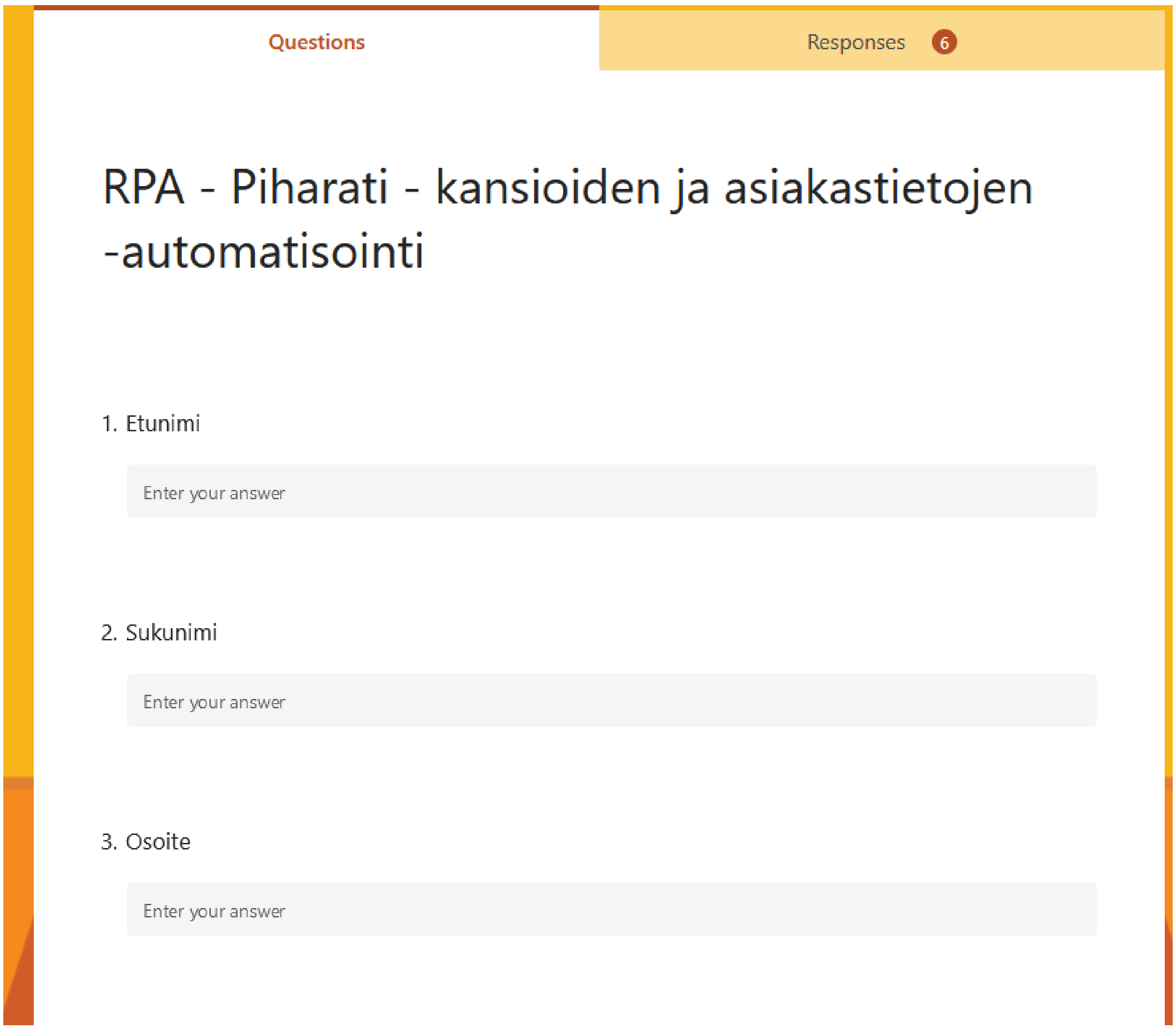

Figure 14: Top part of forms used

Fill in this: column2= asiakkaan_nimi_puh_template

The beginning section of the Forms template is shown in (Figure 14).

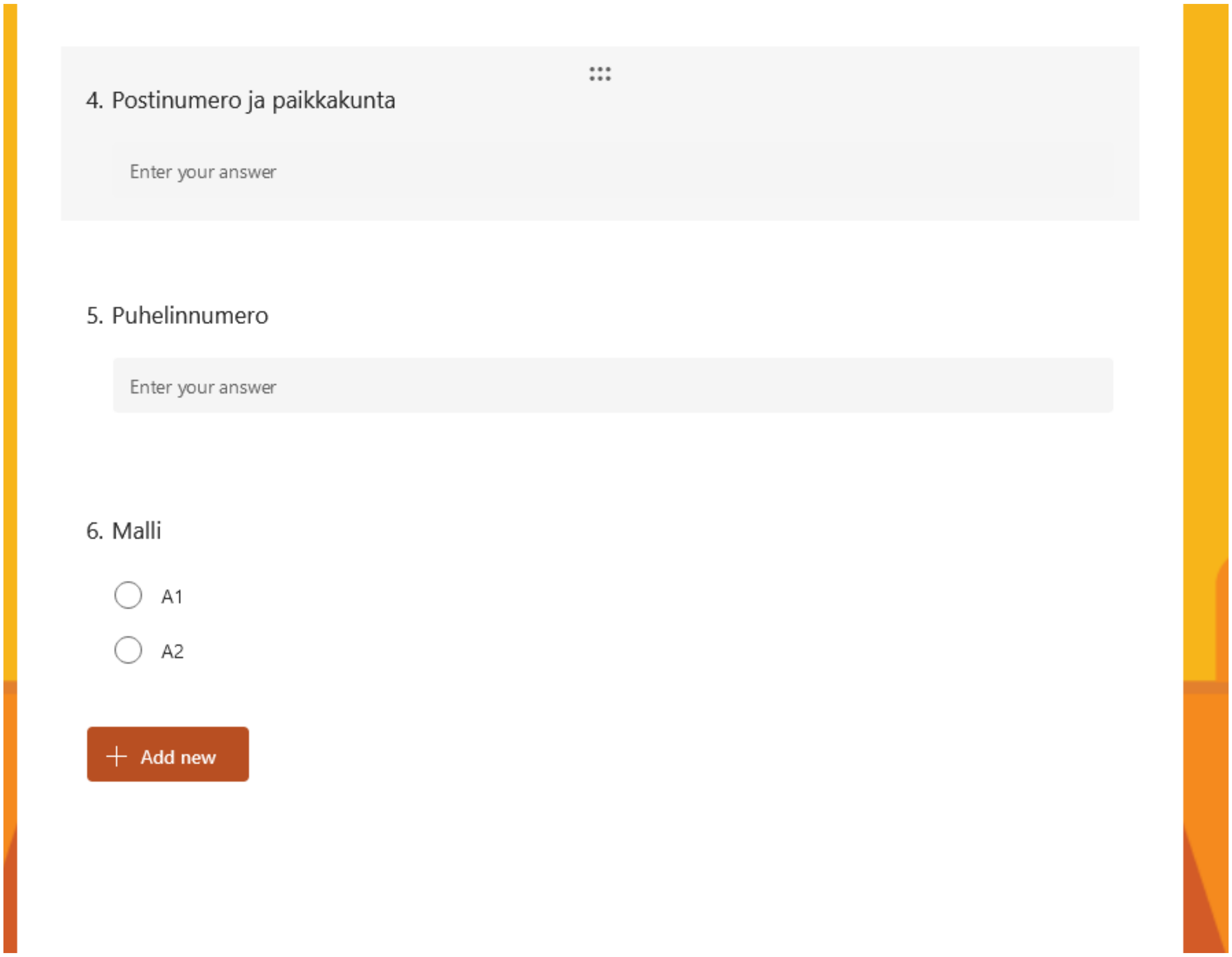

Figure 15: Bottom part of forms used

End of the Forms form is shown in (Figure 15).

1.5 Conclusion

By using Microsoft Power Automate and Forms, it was possible to automate customer project management process, saving time and increasing efficiency. The flow processes customer information and updates an Excel template and change log, ensuring that all customer information is easily accessible and up-to-date.