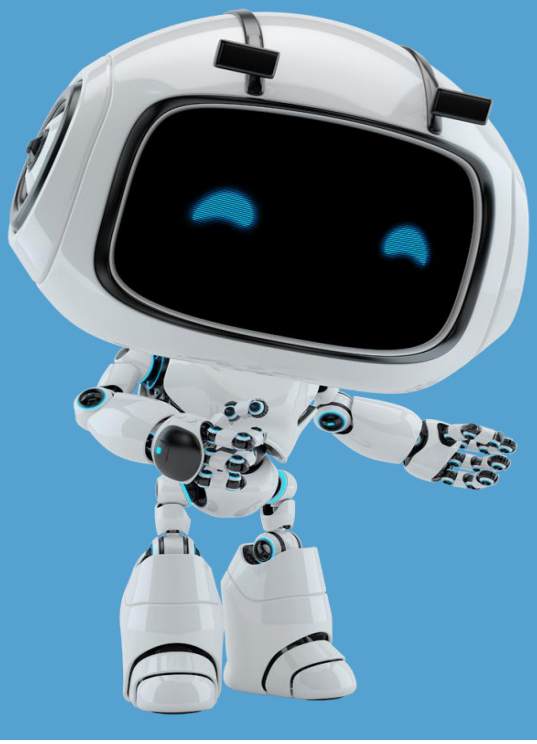
Robotic Process Automation



Recruitment_Task

Process Definition Document (PDD)

Table of Contents

1. Document Information

System Name	Collecting Most Popular Names in Poland 2018
Document Name	Process Definition Document (PDD)

Role	Name	Date
Developer	Konrad Bywald	18.08.2021

2. Purpose

This document describes the implementation of Recruitment_Task. Its purpose is to specify and explain the way in which automation has been created. In further chapters are described sequences and workflows used in the project. The Process Definition Document (PDD) is created for every business process that is automated using the RPA technology and document is filled by the RPA Solution Architect and RPA Developer who automates the business process.

3. Definitions

The following additional terms and abbreviations are used in this document:

Term	Definition
Master project	The overall output of the development, containing one or multiple projects that together cover the scope of the robotic process automation.
Project	An UiPath Studio project containing one or multiple workflow files. A project can be converted to a package and run independently, covering a particular scope within the master project.
Package	The output of compiling a project. A package can be deployed on the robot machine and be executed by the robot service. Only one package can be executed at a given time by a robot. The package is used when defining the running phase of the automation.
Workflow	A component of the package, the workflow encapsulates a part of the project logic. The workflow can be of type: sequence, flowchart or state machine. a workflow is saved as an .xaml file inside the project folder. A workflow file can be invoked from another workflow and by default there is an initial workflow file that will run when executing the package.
Activity	An action that the robot executes.
Sequence	A workflow where activities are executed one after another, in a sequential order.
Flowchart	A workflow where activities are connected by arrows and the logic of the workflow can be easily followed in a visual manner.

Term	Definition
State machine	A more advanced way of organizing a workflow, with higher capability of flow control. Similar to a flowchart.
State	A part of state machine representing state of application.
UiPath	Software provider for the automation.
ReFramework	Framework for building enterprise class automations in UiPath.
Robot	An execution host that runs processes built in UiPath Studio.
System exception	Exception caused by system failure.
Business rule exception	Exception caused by rules defined by the Business.

4. Master Project Details

Details filled in by the developer reflect the actual information for the master project.

Item	Details
Master project name and version	Recruitment_Task
Robot type	Unattended
Is Orchestrator used? (Yes/No)	No
Scalable? (Yes/No) (can the process be run by multiple robots in parallel)	No

4.1 List of packages

List of packages used in the automation.

Package Name	Version	Details
UiPath.Excel.Activities	2.11.0	The Excel activities package allow users to automate all aspects of Microsoft Excel
UiPath.System.Activities	21.6.1	The System Activities Pack contains all the basic activities used for creating automation projects
UiPath.UIAutomation.Activities	21.6.1	The UIAutomation activities package contains all the basic activities used for creating automation projects.

4.2 Runtime Details

Details of the automated process.

<table>
<tr><th></th><th></th></tr>
<tr><td>Prerequisites to run</td><td>- UiPath Robot installed
- UiPath Studio installed
- UiPath Chrome Extension has to be enabled in Google Chrome with allow incognito option enabled
- Access to the Internet</td></tr>
<tr><td>Input Data</td><td>Names of children extracted from the given websites.</td></tr>
<tr><td>Expected Output (output data)</td><td>Exported table to the excel file with top 10 names of boys and girls ordered descending by number of children with given name.
<table>
<tr><th>Imię</th><th>Płeć</th><th>Liczba dzieci</th><th>Miesiąc</th><th>Unicorn Name</th></tr>
<tr><td>Andrzej</td><td>M</td><td>264</td><td>September</td><td>Radiant Moon-Clover</td></tr>
</table></td></tr>
<tr><td>How to start the automated process?</td><td>- Manually from the Robot’s tray icon
- Manually from the UiPath Studio (recommended for this project)</td></tr>
<tr><td>Reporting (Kibana or another platform)</td><td>N/A (Robot does not send any reports after processing)</td></tr>
<tr><td>How to resume the process in case of error</td><td>Automation contains a mechanism which repeat process if exception has been thrown.</td></tr>
<tr><td>Stored Credentials: (Never hardcode credentials)</td><td>N/A (This process does not use any credentials from Orchestrator or credential manager)</td></tr>
<tr><td>List of Asset Names:</td><td>N/A (This process does not use any assets from Orchestrator)</td></tr>
<tr><td>Recommended resolution</td><td>1920x1080</td></tr>
</table>

5. User Requirements, Functional or Use Case Specification Document Reference

User Requirements Specification Document Name:	Zadanie rekrutacyjne.pdf

Celem ćwiczenia jest zbudowanie robota programowego za pomocą narzędzia UiPath, który wykona następujące czynności:

1. Pobranie ze stron: https://www.gov.pl/web/cyfryzacja/najpopularniejsze-imiona-dla-chlopcow-2018-ranking-ogolnopolski oraz https://www.gov.pl/web/cyfryzacja/najpopularniejsze-imiona-dla-dziewczynek-2018-ranking-ogolnopolski list z imionami dzieci jakie były nadane w 2018 roku.
2. Wyodrębnienie po 10 pierwszych imion z każdej ze stron.
3. Wygenerowanie dla każdego z imion losowej nazwy miesiąca.
4. Wygenerowanie na stronie https://www.rpasamples.com/unicornname Unicorn Name na podstawie imion i przypisanych im miesięcy.
5. Wyeksportowanie tablicy do arkusza kalkulacyjnego. Docelowy wygląd tabeli przedstawiony jest poniżej. Tabela powinna być posortowana malejąco po kolumnie „Liczba dzieci". Imiona powinny być zapisane w formacie - pierwsza wielka litera, pozostałe małe litery (przykład poniżej w tabeli).

Imię	Płeć	Liczba dzieci	Miesiąc	Unicorn Name
Andrzej	M	264	September	Radiant Moon-Clover

6. Wraz z kodem robota należy dołączyć również wypełniony dokument PDD zgodny ze standardami UiPath.

Zadanie będzie oceniane pod kątem realizacji wszystkich powyższych kroków oraz czystości i klarowności kodu.

5.1 Process Diagram

Diagram which represents flow of the process based on the requirements.

5.2 Requirements implementation

The requirements are implemented in the following way:

Step 1 – Download from websites list with names of children

Robot navigates to the given websites using URLs which are stored in the Config.xlsx file in settings called govWebsites_URL. Both URLs are assigned to one setting and they are separated

with # character, later in the code Robot use split method to separate URLs and use them in the Open Browser activity in order to navigate to the website. Once website is opened Robot use Get Text activity which purpose is to extract whole list of names from given website, as a result Robot receive text value which is later processed during automation.

Step 2 – Select top 10 names on each website

Robot uses regex regular expression in order to find match in the extracted text. Next he takes specify number of names using Take() method. This number can be set in the Config.xlsx file by changing value for setting called numberOfNamesToTake.

Step 3 – Generate random month for each name

Robot generates random month by using list of months to which he refers with random index generated by Random() method. Extracted value from list is assigned to the variable and added to the data table.

Step 4 – Generate Unicorn Name based on names and assigned months

In order to generate Unicorn Name robot take two variables – name and generated month. Next he navigates to the given website and enter data. After generating result value is extracted using Get Text activity and assigned directly to the appropriate variable.

Step 5 – Export table to the excel file. Table should be sorted descending by number of children. Names should start with capital letter but rest should be lowercase.

In the End Process state if no error occurred during processing robot invokes workflow which is responsible for write prepared sorted data table to the excel. First data table is sorted descending by column “Number of children”.

5.3 Configuration files

Following files are used as configuration sources for the project:

1. Project.json - this file contains the project metadata.

Variable name:	Description:
name	The title of the automation project.
description	The description of the project.
main	The entry point of the automation project. It consists of an .xaml file. The default name is “Main.xaml”.
dependencies	The activity packages used to create the automation project and their versions.
studioVersion	The version of Studio used to create the automation project.
projectVersion	The version of the project

2. Config.xlsx

   Main source of configuration data for the project. The purpose of each configuration variable is described inside the file.

   Spreadsheet is divided into 3 separate tabs:

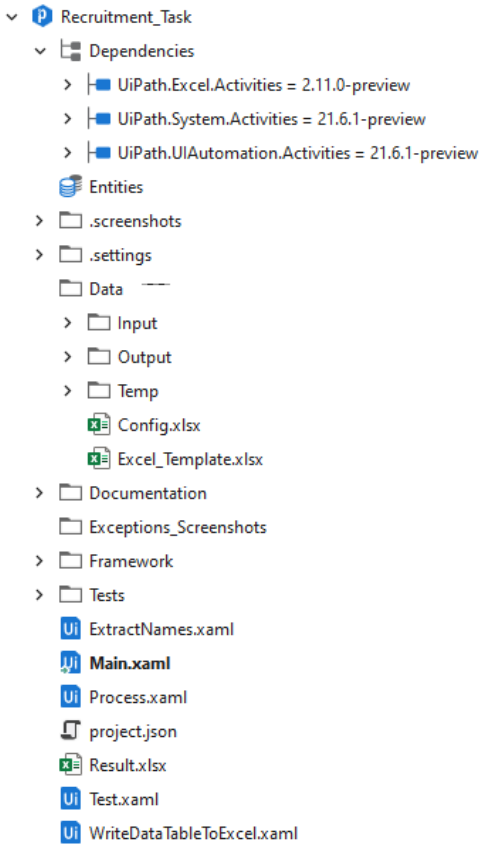
   - Settings: contains configuration variables for the project.
   - Constants: constants used around the project.
   - Assets: contains mappings between configuration key names and assets names on the Orchestrator. Assets are downloaded from the Orchestrator and added to the Dictionary<string, object> Config object. Assets will overdrive the configuration variables with the same names.

6. Solution Design

The Recruitment_Task is an automation which main purpose is to automate short process of downloading most popular names from the given websites. Moreover during execution robot also generates random month and unicorn name for each children name. All of those values are added to the data table and moved to the excel file as a result of execution. Project has been created based on the UiPath ReFramework with one additional state added after Initialization. This state is responsible for extracting data for processing and passing them to the Transaction Data collection.

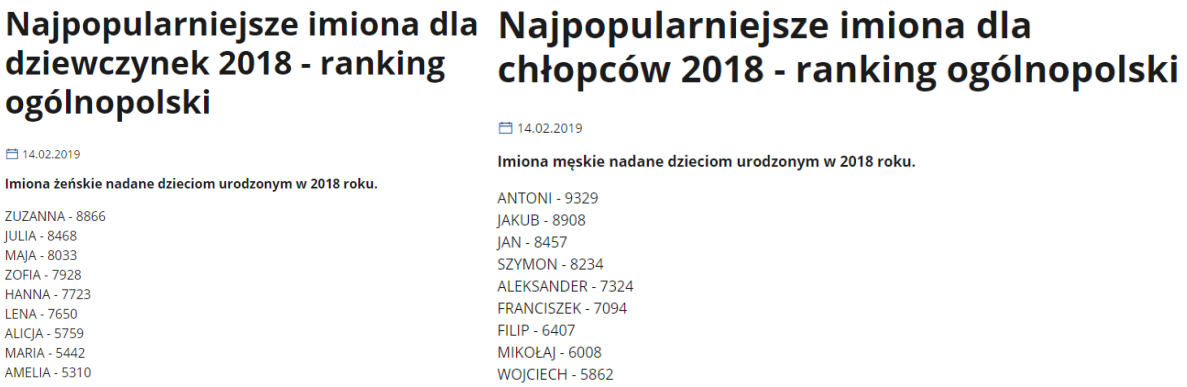
6.1 Project folders structure

The following folder structure is used in the project:



6.2 Input Data

Input for automation is extracted from websites specified in the requirements. Data are presented on the pages in very long list which robot needs to extract.



Najpopularniejsze imiona dla dziewczynek 2018 - ranking ogólnopolski

14.02.2019

Imiona żeńskie nadane dzieciom urodzonym w 2018 roku.

ZUZANNA - 8866
JULIA - 8468
MAJA - 8033
ZOFIA - 7928
HANNA - 7723
LENA - 7650
ALICJA - 5759
MARIA - 5442
AMELIA - 5310

Najpopularniejsze imiona dla chłopców 2018 - ranking ogólnopolski

14.02.2019

Imiona męskie nadane dzieciom urodzonym w 2018 roku.

ANTONI - 9329
JAKUB - 8908
JAN - 8457
SZYMON - 8234
ALEKSANDER - 7324
FRANCISZEK - 7094
FILIP - 6407
MIKOŁAJ - 6008
WOJCIECH - 5862

6.3 Output Data

Automation as the output produce an excel file which contains selected top 10 names for girls and boys. Table is sorted descending by column “Number of children”.

Name	Gender	Number of children	Month	Unicorn Name
Antoni	M	9329	February	Radiant Twinkle-Sweet
Jakub	M	8908	October	Sunbeam Feather-Wind
Zuzanna	F	8866	September	Bumblebee Moon-Clover
Julia	F	8468	July	Sunbeam Raindrop-Mist
Jan	M	8457	August	Sunbeam Summer-Dancer
Szymon	M	8234	October	Fancy Feather-Wind
Maja	F	8033	May	Delightful Snowflake-Dream
Zofia	F	7928	April	Bumblebee Nimble-Flower
Hanna	F	7723	November	Glimmer Starshine-Blazer
Lena	F	7650	January	Optimistic Crystal-Dazzler
Aleksander	M	7324	October	Radiant Feather-Wind
Franciszek	M	7094	March	Whirlwind Glitter-Blossom
Filip	M	6407	June	Whirlwind Cloud-Jumper
Mikołaj	M	6008	July	Delightful Raindrop-Mist
Wojciech	M	5862	January	Brilliant Crystal-Dazzler
Alicja	F	5759	February	Radiant Twinkle-Sweet
Maria	F	5442	November	Delightful Starshine-Blazer
Kacper	M	5437	September	Emerald Moon-Clover
Amelia	F	5310	February	Radiant Twinkle-Sweet
Oliwia	F	5058	June	Breezy Cloud-Jumper

7. Project Workflows

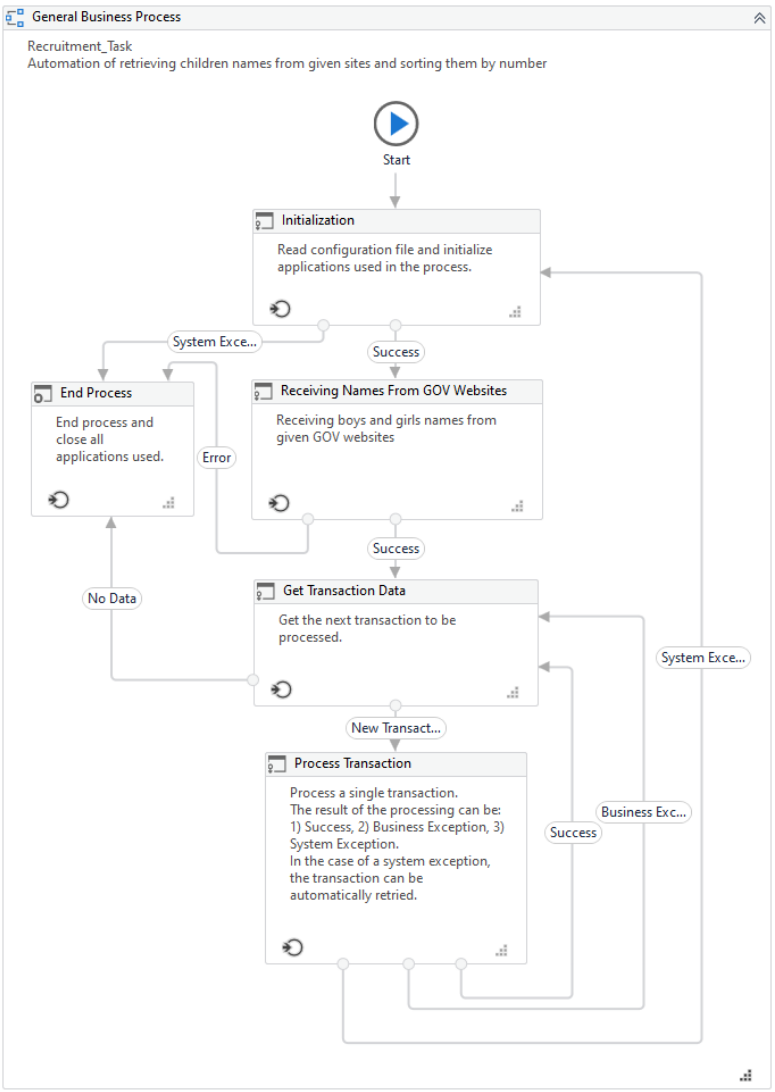
Define below all the workflow files (xaml files) used in the project, with the Input and Output data.

#	Workflow file name	Description	Input Argument	Output Argument
1	Main.xaml	Main file contains state machine which holds all states and is responsible for performing process. By default Main.xaml file is set as main in project.json	N/A	N/A
2	ExtractNames.xaml	This workflow is responsible for extracting names from given websites. In the for each loop robot navigates to the pages and on each of them he reads names using Get Text activity next result is checked with the regex expression and specified number of names is added to the TransactionData List<String> collection.	• io_TransactionData (List<String>) • in_WebsiteURLS String[] • in_Config Dictionary<String, Object>	• io_TransactionData (List<String>)
3	SortAndWriteDataTableToExcel.xaml	This workflow is used in the End State and it is responsible for sorting DataTable descending by number of children. Next sorted DataTable is moved to the excel file using Excel_Template.xlsx file which is stored in the Data folder	• in_ExcelDataTable Datatable • in_Config Dictionary<String, Object>	N/A
4	Process.xaml	Invoke major steps of the business process, which are usually implemented by multiple subworkflows.	• in_TransactionItem String • in_Config Dictionary<String, Object> • in_UnicornNameBrowser Browser • io_excelDataTable DataTable	• io_excelDataTable DataTable
5	GetTransactionData.xaml	Part of REFramework. Gets data from TransactionData collection. If no new data, set out_TransactionItem to nothing.	• in_TransactionNumber Int32 • in_Config Dictionary<String, Object> • out_TransactionItem String • io_TransactionData List<String>	• io_TransactionData List<String>
6	InitAllSettings.xaml	Part of REFramework. Outputs a settings Dictionary with key/value pairs to be used in the project. Settings are read from local config file then fetched from Orchestrator assets. Assets will overwrite the configuration file settings	• in_ConfigFile String • in_ConfigSheets String[]	• out_Config Dictionary<String, Object>
7	KillAllProcesses.xaml	Part of REFramework. Kills the working processes used in the automation. Processes killed by the workflow: saplogon.exe, chrome.exe	N/A	N/A
8	SetTransactionStatus.xaml	Part of REFramework. This workflow sets the TransactionStatus and Logs that status and details in extra Logging Fields. The flowchart branches out into the three possible Transaction Statuses: Success, Business Exception and Application Exception.	• in_Config Dictionary<String, Object> • in_TransactionItem String • in_TransactionData List<String> • in_BusinessException BusinessRuleException • in_Exception Exception	• io_TransactionNumber Int32 • io_RetryNumber Int32
9	TakeScreenshot.xaml	Part of REFramework. This workflow captures a screenshot and logs its name and location.	• in_Folder String	N/A
10	CloseAllApplications.xaml	Part of REFramework. Closes all working applications.	N/A	N/A

7.1 Main.xaml

Main.xaml file holds all states which belongs to the automation. In this case one additional state has been added which is responsible for collecting data for processing. At the beginning in the Initialization robot is reading Config file and preparing data table which will be filled later with all data and move to the result excel file. Below the structure of the data table is presented and Main.xaml file as well. Result file is saved in the project folder. We can specify his name by change value in config file for setting called excel_ResultFileName.

Name (String)	Gender (String)	Number of children (Int32)	Month (String)	Unicorn Name (String)

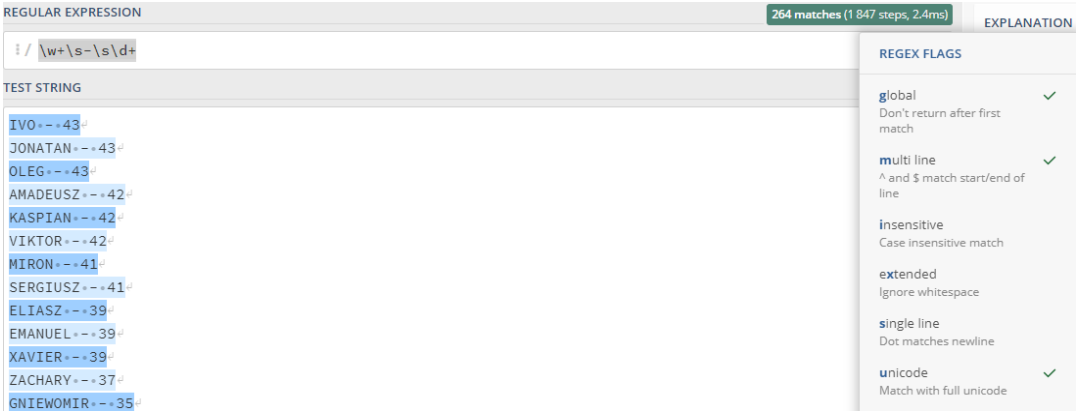


7.2 ExtractNames.xaml

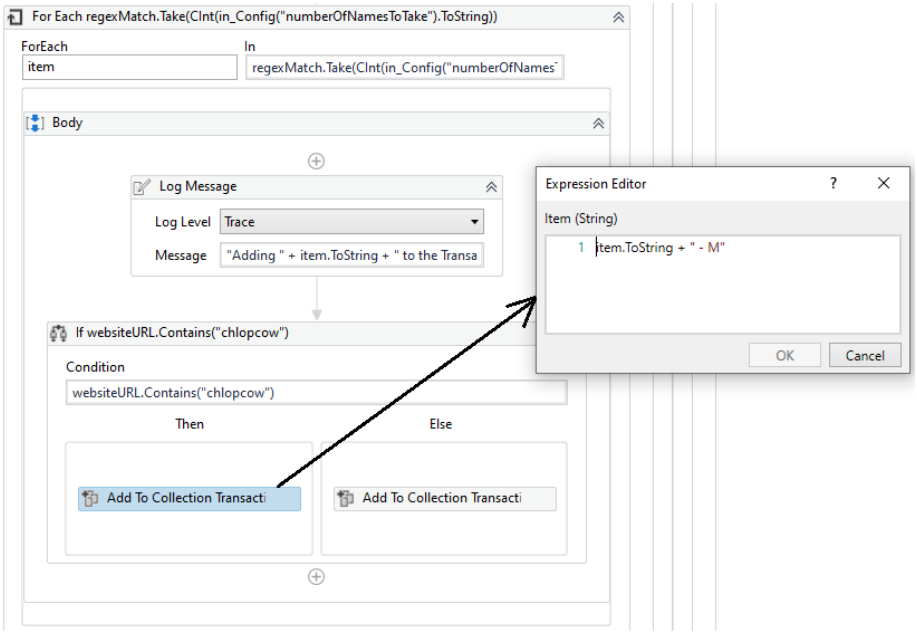
This workflow is invoked in state called Receiving Names From FOV Websites. It is responsible for navigate to the given websites using value passed via argument in_WebsiteURLS this value is splitted by # and passed to the For Each loop. Robot uses Google Chrome in incognito mode in order to open website and it cause that always after opening the page at the bottom of the page cookie popup appears, but it does not have any impact on the automation and it is not necessary to focus on it. At the beginning browse window is maximized and next Robot uses Get Text activity to extract whole list with the names.

Extracted value is check if regex found matches using pattern: \w+\s-\s\d+

Below example screenshot from regex101 website with flags sets as: global, multi-line and unicode.



Once matches were found Robot takes specified number of names by using value stored in the Config and adds them to the TransactionData collection in meantime checking if given name was received from boys page or no and based on that information Robot makes decision which gender should be assigned for this name.



7.3 Process.xaml

In this workflow robot performing actions according to the requirement document. Here the random month and Unicorn Name are generated.

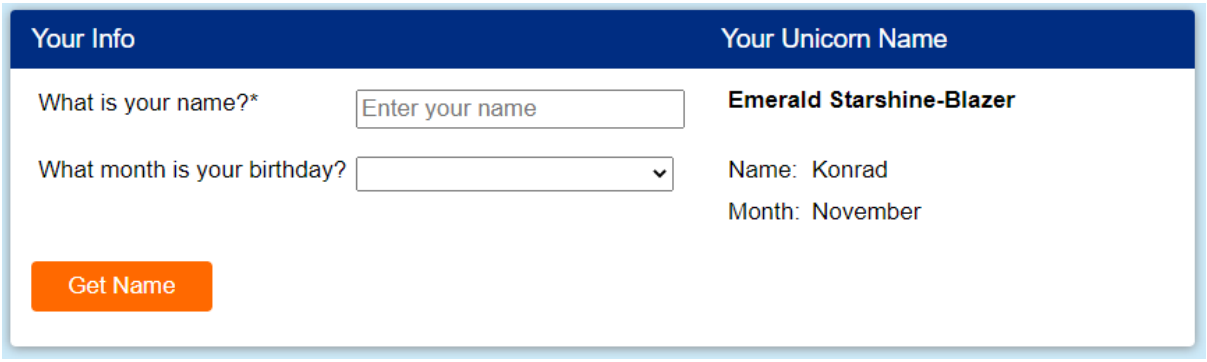
In order to generate random month robot is using list with months with the given structure:

```
New List(Of String) From {"January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"}
```

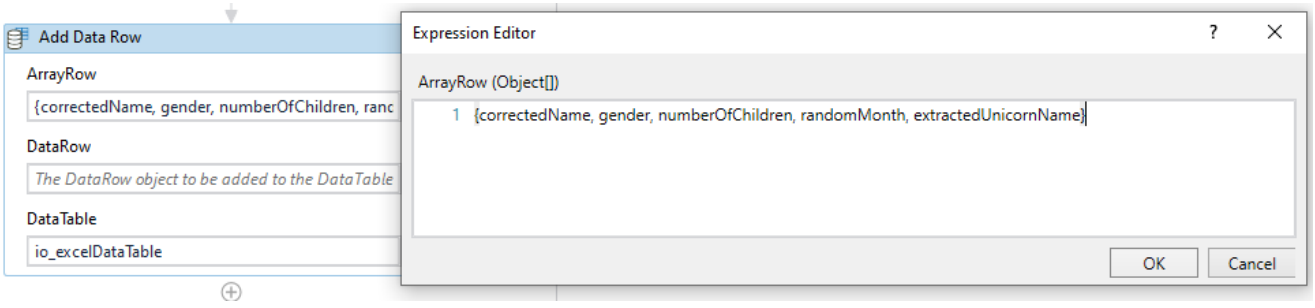
Random month is selected by assigning value from the list to the variable called randomMonth using Random().Next() method which is invoked as index of the list.

```
randomMonth = monthsList(cInt(new Random().Next(0,11)))
```

Unicorn Name is generated on the page specified in the requirements. Robot uses name and generated month next enters those values to the form in order to receive Unicorn Name which is extracted from the website using Get Text activity and result is assigned to the variable called extractedUnicornName.



At the end of Process.xaml all received and generated data are added to the Data Table which is set as in/out argument and in the End Process state will be sorted and saved to the excel file.



7.4 SortAndWriteDataTableToExcel.xaml

This workflow is invoked in the End Process state. The purpose of this workflow is to sort data table which is passed as input argument. Sorting is done descending by column “Number of children”. After that sorted data table is saved in the result excel file which name can be specified in the Config file by change value for setting excel_ResultFileName value. Name of the sheet used in the file can be specified in the Config as well. Excel file as a template is saved in the Data folder and it is always copied to the main project folder with option overwrite set to true.