|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| VILNIAUS KOLEGIJA  UNIVERSITY OF APPLIED SCIENCES  FACULTY OF ELECTRONICS AND INFORMATICS  Image result for viko logo | | |  | | | VILNIUS COLLEGE  Image result for viko logoFACULTY OF ELECTRONICS AND INFORMATICS |
|  | | |  | | |  |
| **INTRODUCTION TO INFORMATICS** | | |  | | | **INTRODUCTION TO INFORMATICS** |
| PRACTICAL ASSIGNMENT  Introduction to informatics work report  6531BX028 PI18E | | |  | | | PRACTICAL ASSIGNMENT  SPOTIFY USER MANUAL  6531BX028 PI18E |
| STUDENT | DŽIUGAS PEČIULEVIČIUS | STUDENT | | DŽIUGAS PEČIULEVIČIUS |
| (SIGNATURE)  1/3/2019 | | |  | | | LECTURER |
| LECTURER | VIRGILIJUS KUKLIERIUS | (SIGNATURE)  10/17/2018 | | VIRGILIJUS KUKLIERIUS |
| (SIGNATURE)  1/3/2019 | | |  | | | 2018 |

2019

# CONTENTS

[CONTENTS 2](#_Toc534893051)

[TITLE PAGE 7](#_Toc534893052)

[USER’S GUIDE 12](#_Toc534893053)[12](#_Toc534893054)

[**Title page** 12](#_Toc534893055)

[**Introduction to a software** 12](#_Toc534893057)

[**Downloading and installing the application** 12](#_Toc534893059)

[**Explaining how to get started on the new software and how to use it** 12](#_Toc534893060)

[**Adding the table of contents** 13](#_Toc534893061)

[WEB FORMAT PRESENTATION 14](#_Toc534893062)

[**Adding frames** 14](#_Toc534893063)

[**Adding information to your page** 18](#_Toc534893064)

[QUESTIONNAIRE 21](#_Toc534893065)

[**Adding basics to the questionnaire** 21](#_Toc534893068)

[**Adding Visual Basic for Applications (VBA) code** 26](#_Toc534893073)

[WORKS DONE OUT OF EXCEL 31](#_Toc534893074)

[**UserForm Database** 31](#_Toc534893076)

[**Writing the VBA code for the ‘Add’ button** 37](#_Toc534893077)

[**Writing the VBA code for the ‘Edit’ button** 41](#_Toc534893078)

[ACCESS 44](#_Toc534893079)

[**Picture 1 - A way of selecting custom margins** 6](#_Toc534893149)

[**Picture 2 - Selecting correct Margins** 7](#_Toc534893150)

[**Picture 3 - Entering Title page information and a logo** 7](#_Toc534893151)

[**Picture 4 - Selecting the Drop-Down list option** 8](#_Toc534893152)

[**Picture 5 - Drop-Down list** 8](#_Toc534893153)

[**Picture 6 - Content Control Properties** 9](#_Toc534893154)

[**Picture 7 - Work type, name and group information** 9](#_Toc534893155)

[**Picture 8 - Final touches for the title page** 10](#_Toc534893156)

[**Picture 9 – Click on File** 13](#_Toc534893157)

[**Picture 10 –Click on Options** 13](#_Toc534893158)

[**Picture 11 - Word Options Window** 14](#_Toc534893159)

[**Picture 12 - Click on Customize Ribbon** 14](#_Toc534893160)

[**Picture 13 - Choose Commands Not in the Ribbon and find Frames in the list** 15](#_Toc534893161)

[**Picture 14 - Add a New Group into a chosen category** 15](#_Toc534893162)

[**Picture 15 - Name it to Frames and add everything from the left Window to this** 15](#_Toc534893163)

[**Picture 16 - After adding all Frame options to this group click OK** 16](#_Toc534893164)

[**Picture 17 - All the Frame options are in the chosen Group** 16](#_Toc534893165)

[**Picture 18 - After you add frames to your document, you have a "skeleton" of your work** 16](#_Toc534893166)

[**Picture 19 - Naming a page** 17](#_Toc534893167)

[**Picture 20 - Create another word document in web format with the text you want** 17](#_Toc534893168)

[**Picture 21 - Click on the text and click Hyperlink** 18](#_Toc534893169)

[**Picture 22 - Click on Target Frame** 18](#_Toc534893170)

[**Picture 23 - Target a frame you want the information to appear and click OK** 18](#_Toc534893171)

[**Picture 24 - Find the document with the text you entered at first and click OK** 19](#_Toc534893172)

[**Picture 25 - After clicking on the text in main page, it will open the new page** 19](#_Toc534893173)

[**Picture 26 - Final look on a web page** 19](#_Toc534893174)

[**Picture 27 - Beginning of a questionnaire** 20](#_Toc534893175)

[**Picture 28 - Making an empty table** 20](#_Toc534893176)

[**Picture 29 - Writing down all the questions** 21](#_Toc534893177)

[**Picture 30 - Choosing an Option button to make possible answers** 21](#_Toc534893178)

[**Picture 31 - Add 4 or 5 option buttons for possible answers** 22](#_Toc534893179)

[**Picture 32 - For every single button right click and choose properties** 22](#_Toc534893180)

[**Picture 33 - Insert an answer in the Caption block and question number in the Group Name** 22](#_Toc534893181)

[**Picture 34 - Find the Command button option in the Developer tab and add it** 23](#_Toc534893182)

[**Picture 35 - Once Command button is added, go to buttons properties** 23](#_Toc534893183)

[**Picture 36 - Name your Command button** 24](#_Toc534893184)

[**Picture 37 - This is how Command buttons should look after you're done** 24](#_Toc534893185)

[**Picture 38 - Final look on the visuals** 25](#_Toc534893186)

[**Picture 39 - Right click on the RESET button and then click on View Code** 25](#_Toc534893187)

[**Picture 40 - You will get this empty field for your code** 26](#_Toc534893188)

[**Picture 41 - Setting values to each of the option of answers** 26](#_Toc534893189)

[**Picture 42 - Setting all the default values and colors for the text boxes** 27](#_Toc534893190)

[**Picture 43 - Resetting the SUBMIT button** 27](#_Toc534893191)

[**Picture 44 - Checking if there are empty lines** 28](#_Toc534893192)

[**Picture 45 - Setting text boxes up** 28](#_Toc534893193)

[**Picture 46 - Code for the option buttons. Making a choice for a correct one** 29](#_Toc534893194)

[**Picture 47 - Final bit of code for the questionnaire** 29](#_Toc534893195)

[**Picture 48 - Start of a database** 30](#_Toc534893196)

[**Picture 49 - Editing the 'Add' button** 30](#_Toc534893197)

[**Picture 50 - Adding Module** 31](#_Toc534893198)

[**Picture 51 - Adding UserForm** 31](#_Toc534893199)

[**Picture 52 - Linking User Form to the 'Modules" block** 31](#_Toc534893200)

[**Picture 53 - Making of User Form** 32](#_Toc534893201)

[**Picture 54 - Putting labels and text boxes into the user form** 32](#_Toc534893202)

[**Picture 55 - Adding a ComboBox** 33](#_Toc534893203)

[**Picture 56 - Creating a sheet for a Combo Box** 33](#_Toc534893204)

[**Picture 57 - Choose the rows you want to display from the previous sheet slide** 34](#_Toc534893205)

[**Picture 58 - Option Buttons** 34](#_Toc534893206)

[**Picture 59 - Final user form** 35](#_Toc534893207)

[**Picture 60 - Final steps of the user form** 35](#_Toc534893208)

[**Picture 61 - Writing a little bit of code for the 'Cancel' and 'Continue' buttons** 36](#_Toc534893209)

[**Picture 62 - Line Counter for the database** 36](#_Toc534893210)

[**Picture 63 - Adding a "starting" point** 37](#_Toc534893211)

[**Picture 64 - Adding a "starting" point** 37](#_Toc534893212)

[**Picture 65 - Targeting rows** 38](#_Toc534893213)

[**Picture 66 - Making a message box and an ID number** 39](#_Toc534893214)

[**Picture 67 - Code for checking if the FullName value already exists** 39](#_Toc534893215)

[**Picture 68 - Find User to edit** 40](#_Toc534893216)

[**Picture 69 - Create a new name in Name manager** 40](#_Toc534893217)

[**Picture 70 - Creating a dynamic name** 41](#_Toc534893218)

[**Picture 71 - Inserting a dynamic name into the user form combo box** 41](#_Toc534893219)

[**Picture 72 - Code for checking the number of the user in the database** 42](#_Toc534893220)

[**Picture 73 - Linking the edit for to the user form** 42](#_Toc534893221)

[**Picture 74 - Adding left over values** 42](#_Toc534893222)

[**Picture 75 - We rename all the values** 43](#_Toc534893223)

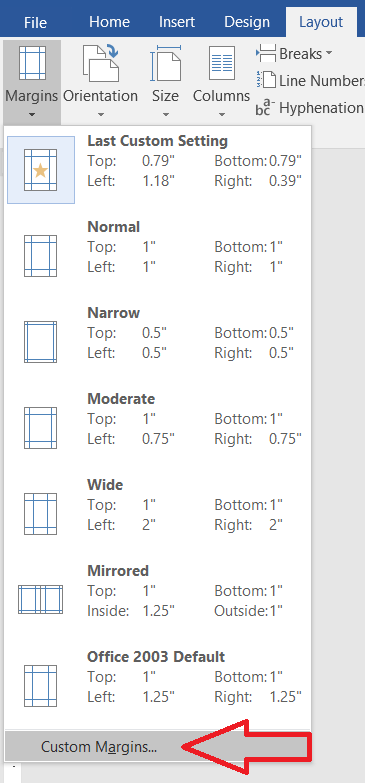
[**Picture 76- We also rename all the values on the userform** 43](#_Toc534893224)

# TITLE PAGE

Title page is needed for every single written work just like this. Title page is used in front of the work and it must contain the name of University, its logo, name of a subject for which the work is going to be presented to, work type and name, group name and code and most important student and lecturer names with signature boxes and dates that signatures have been placed. You can find this title page [here](Title%20page.docx).

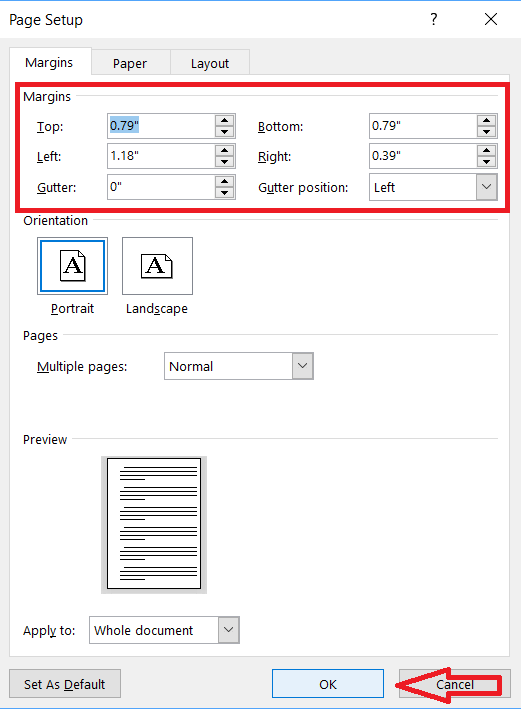
Firstly, for the title page the correct margins were selected which are 20mm at the top and the bottom, then 30mm are on the left and 10mm is on the right of the work. Also, the spacing between lines must be set to 1,5. All of this information on how to set it is displayed in pictures below:

1. Click on Layout tab, then Margins and then Custom Margins...



**Picture 1 - A way of selecting custom margins**

1. After clicking on Custom Margins a table will pop up where you need to select the correct measurments. After you‘re done, click OK.



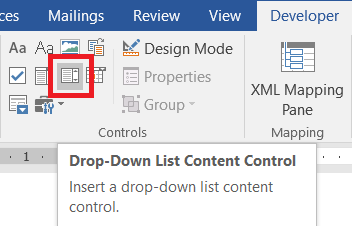
**Picture 2 - Selecting correct Margins**

1. After setting the correct margins, University information is entered followed by a logo. All information is entered in caps, Times New Roman shrift and in size 12.



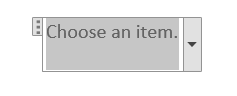
**Picture 3 - Entering Title page information and a logo**

1. After entering University information followed by a logo, subject name is chosen. To insert this to make a list of subjects to choose from, you can do by going to Developer tab and clicking on Controls option and then choosing the Drop-Down List Content Control to make a list of all the subjects and then to just click on this box and choose the subject you want to make a word document for.



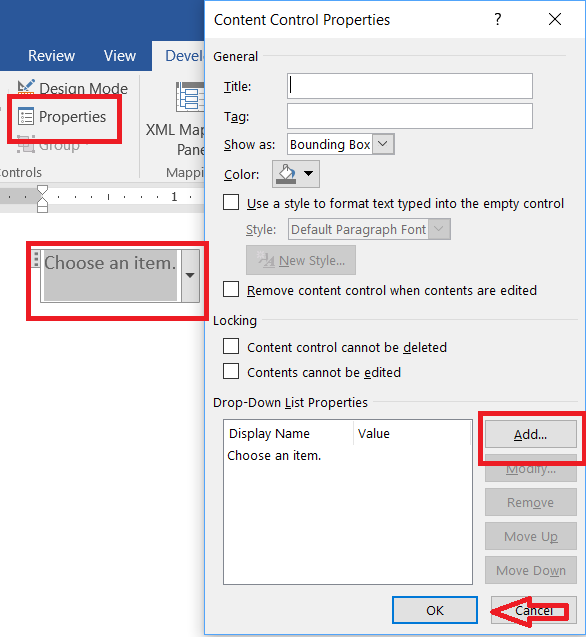
**Picture 4 - Selecting the Drop-Down list option**

1. After clicking on the Drop-Down List option, you will a small box looking like this:



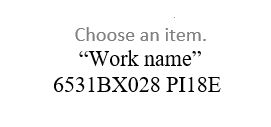
**Picture 5 - Drop-Down list**

1. To make a name and a list for this box, click on the box, then click on Properties and you’ll have a window open where for the Title you can name your Drop-Down list. To add components for the list, click Add and the just type the item you want. After you’re done, click OK.



**Picture 6 - Content Control Properties**

1. After making a Drop-Down list for all lecture names you need to make another Drop-Down list for the type of work you’re doing. After that you need to write the name of your work, group code and group name.



**Picture 7 - Work type, name and group information**

1. The last step of an introduction/title page is to insert student and lecturer names with signature boxes and dates which is done with Drop-Down boxes which is explained above.



**Picture 8 - Final touches for the title page**

# USER’S GUIDE

# Now about a user’s guide. There’s an already made a word format user guide on how to use music player application called Spotify. You can see these instructions [here](Spotify%20Instructions%20IT.docx). Now let’s talk about on how this guide was made.

## **Title page**

# Firstly, like every other word document it should start from a title page which is already explained. After that, you should include an empty page for a content table for all the information and pictures to put in after the work is done. (On how to add a content table, you can find an explanation right [here](#_And_the_table).)

## **Introduction to a software**

# On the first page of your user’s guide (on the page after a content table) it would be a pretty good idea to make an introduction to a software that you’re making a user’s guide for. Your introduction should be short, simple and include information on what does the software do, some short information about compatibility with an operating system of your computer and any other information that might be really important and useful to your users of that software.

## **Downloading and installing the application**

So to make sure your user base understands on how to download and install the application on their devices, you should make instructions on how to do it on every different platform. When writing a user’s guide on how to download an app for your users, you should also include some pictures with specific actions about the process on how to do it and where exactly to click. You should include links to smartphone app stores for the application to download.

If your application is compatible and works on a web based format, it would be great to include that too into your guide that there is a web format application and explain it later, just in case user’s device is not compatible with the software.

## **Explaining how to get started on the new software and how to use it**

Probably most important thing to explain in a ‘getting started’ guide, would be on how to create an account with different platforms, how to login and how to keep your account safe.

## **Adding the table of contents**

After you’re done with your user’s guide, you should include table of contents for your work so application users could easily interact and quickly find the information they need.

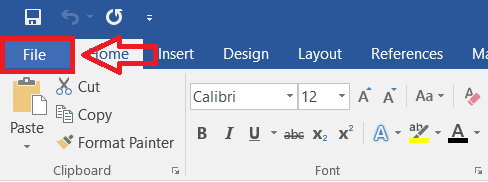
# WEB FORMAT PRESENTATION

This is web format word document, that when opened in an internet browser you can surf through all the pages and the information that were included. You can find this work [here](JAP%20CAR%20web/FinalWebPage.htm). On how this work has been done it’s explained below.

## **Adding frames**

So firstly, to begin with any web page format word document, you need to add frames to the Developer Tab. Let’s do this step-by-step right now:

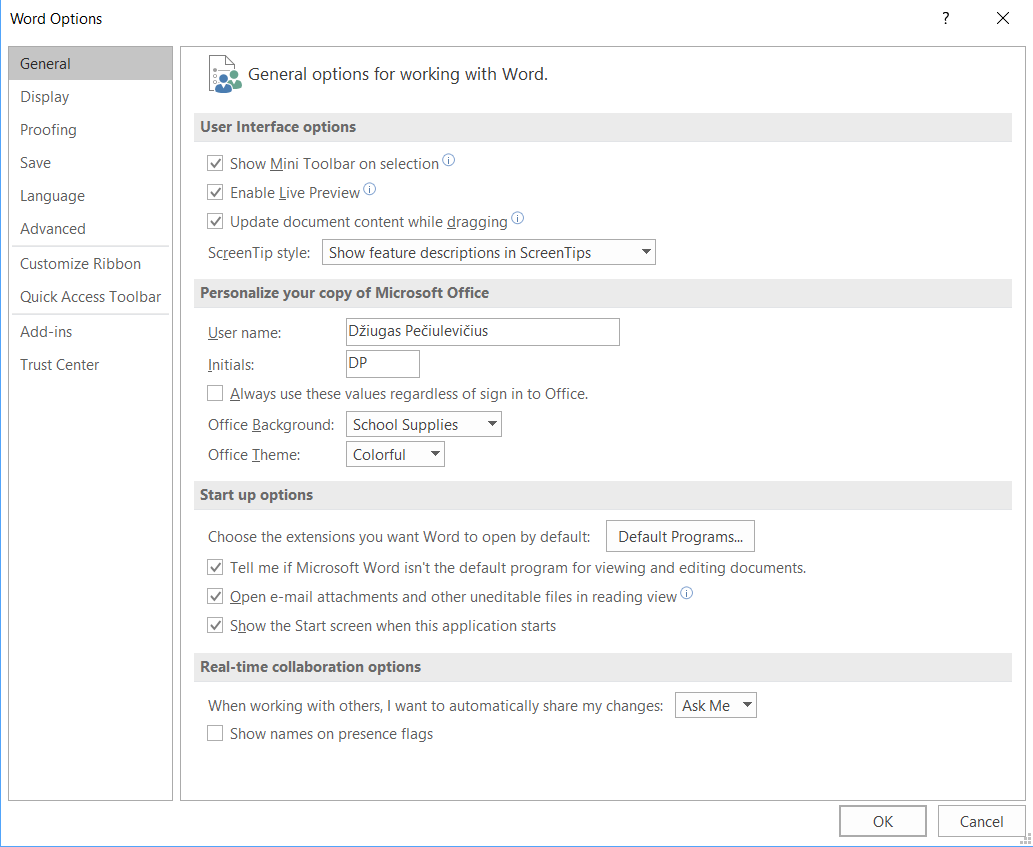
1. To add frames to the Developer Tab you need to click on **File** and then click on **Options** and then you’ll be displayed an options window.



**Picture 9 – Click on File**

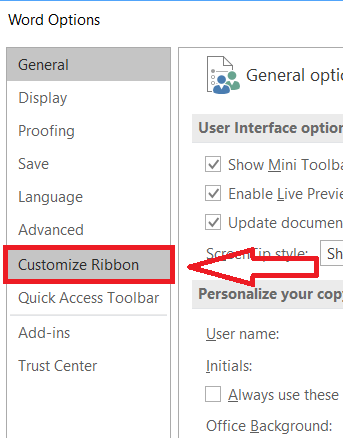


**Picture 10 –Click on Options**

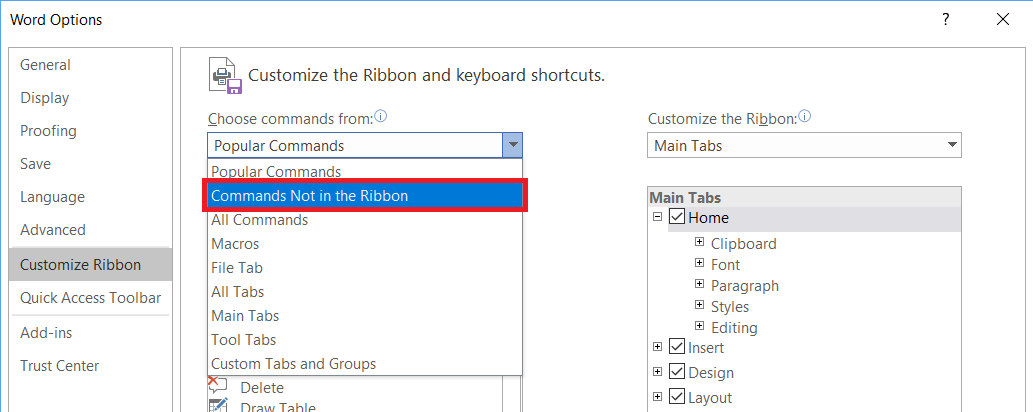


**Picture 11 - Word Options Window**

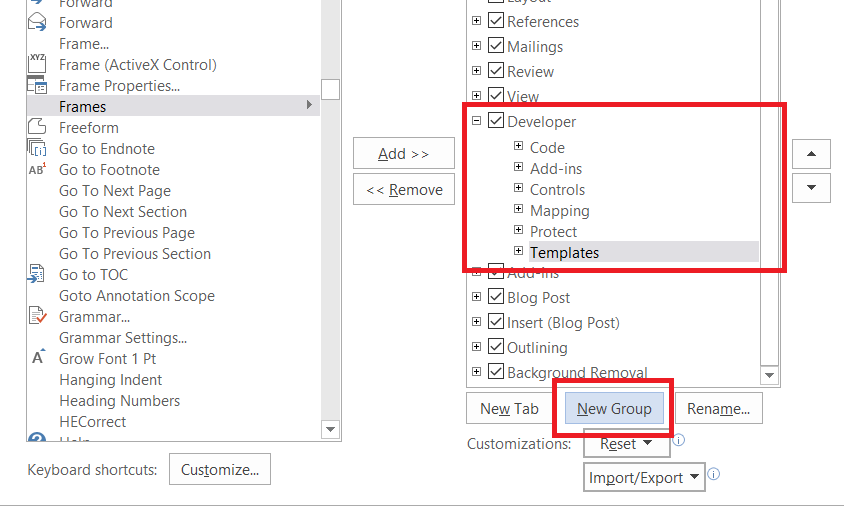
1. Once you got to Word Options window, click on **Customize Ribbon** and then on customize the **Ribbon** (**Choose commands from:**), choose Commands Not in the Ribbon and then find Frames in the list. Now on the other window on the Developer Tab Add a new group and name it Frames. Once you have done that, you can add all the Frame options from the left window to the right into your chosen group. All of this information is explained in pictures below:



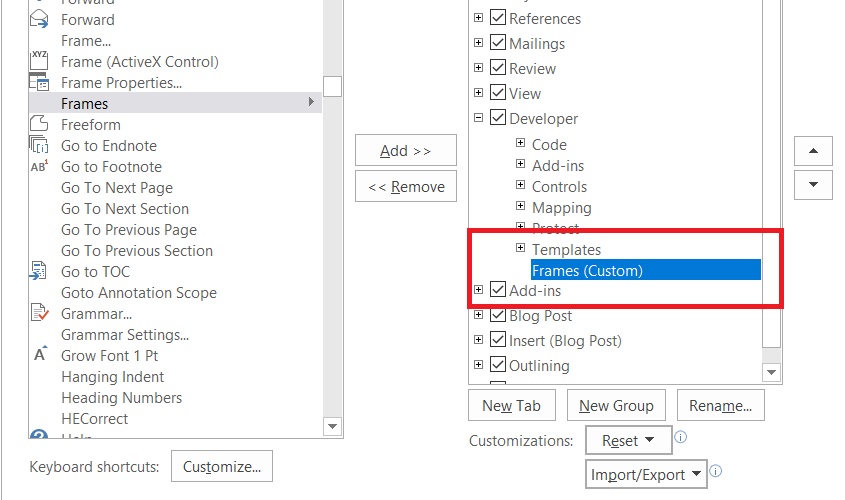
**Picture 12 - Click on Customize Ribbon**



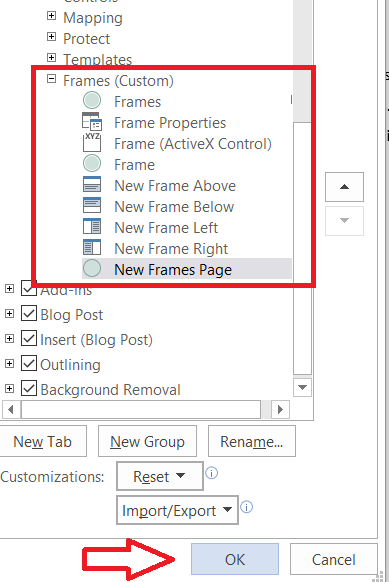
**Picture 13 - Choose Commands Not in the Ribbon and find Frames in the list**



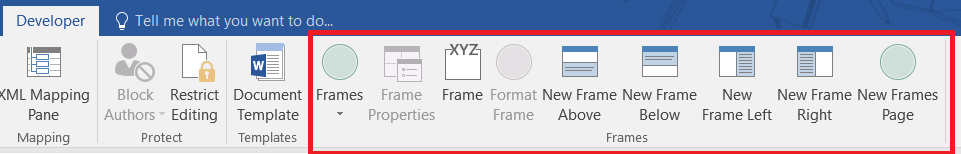
**Picture 14 - Add a New Group into a chosen category**



**Picture 15 - Name it to Frames and add everything from the left Window to this**

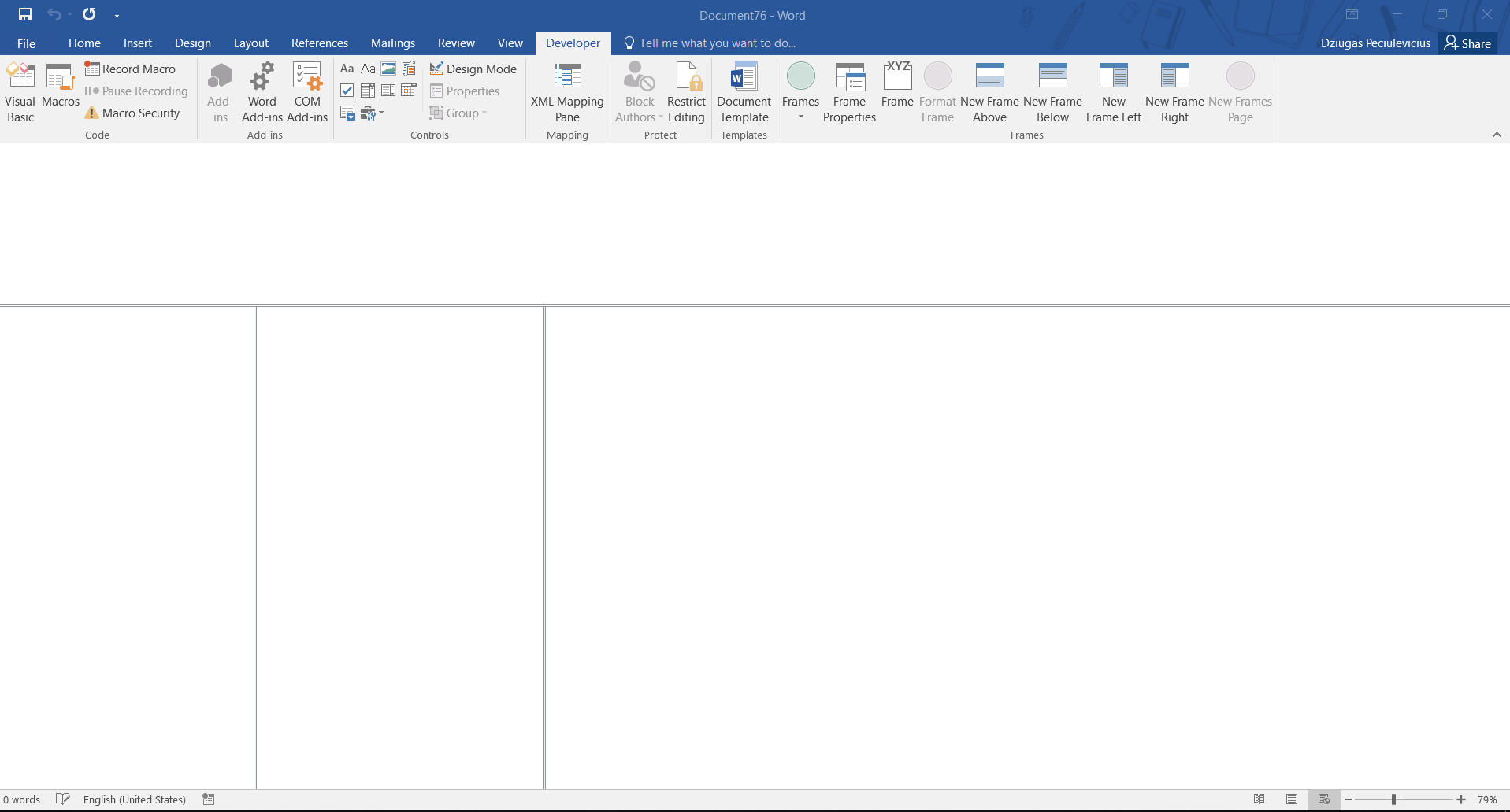


**Picture 16 - After adding all Frame options to this group click OK**



**Picture 17 - All the Frame options are in the chosen Group**

1. Now from the **View Tab** choose **Web Layout** option and then add frames onto the document itself. Then move frames all around how you’d like your page to look like. When you put all the frames how you like, you can start writing the page name on top and some hyperlinks if you like.

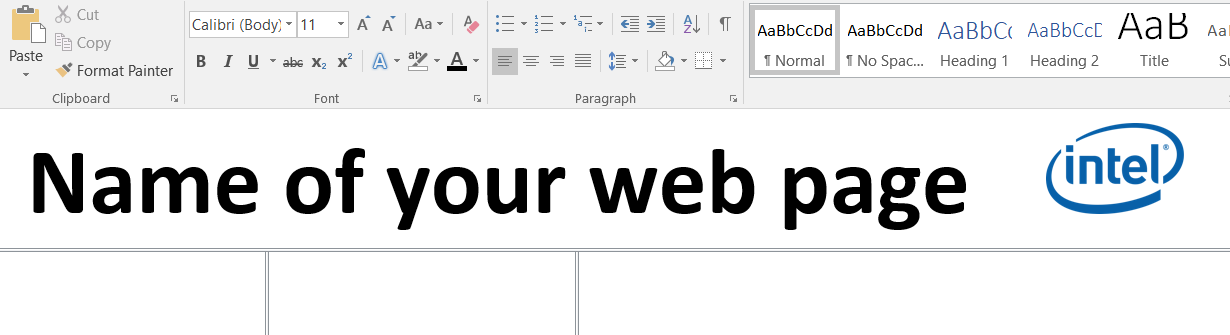


**Picture 18 - After you add frames to your document, you have a "skeleton" of your work**

## **Adding information to your page**

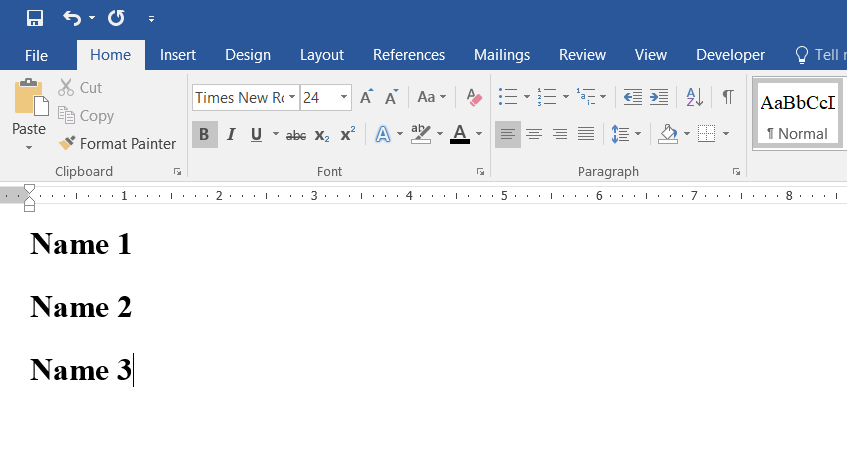
After you’re finished with frames, you can start adding information inside, add pictures if you like. I’ll show an example with a “family tree”.

1. First we’re adding a name to this web page and probably some kind of logo depended on the brand this web page is about.

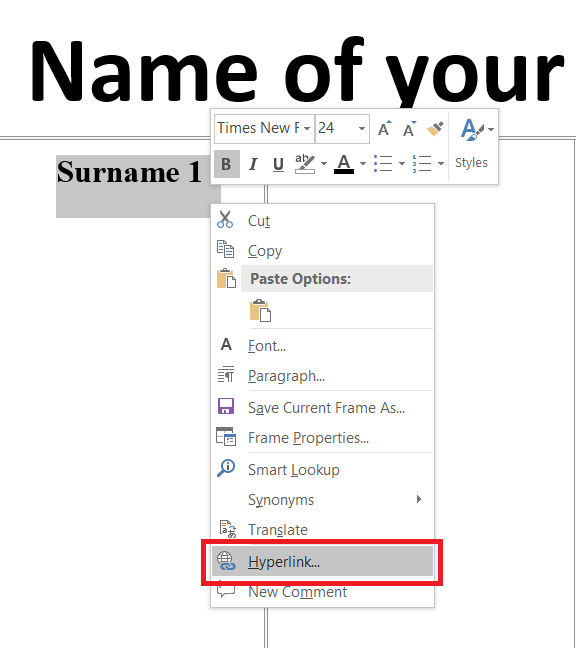


**Picture 19 - Naming a page**

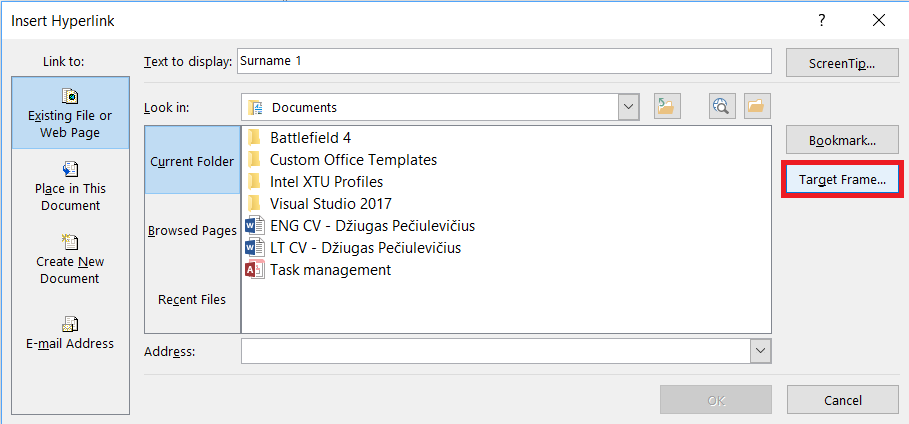
1. Afterwards, you might want to start entering some kind of information to your web page. If you want to get some kind of information on the other frame when you click on a word, this is what you want to do (additional word pages will be created, so it’s recommended to put all the information with pictures into one main folder):



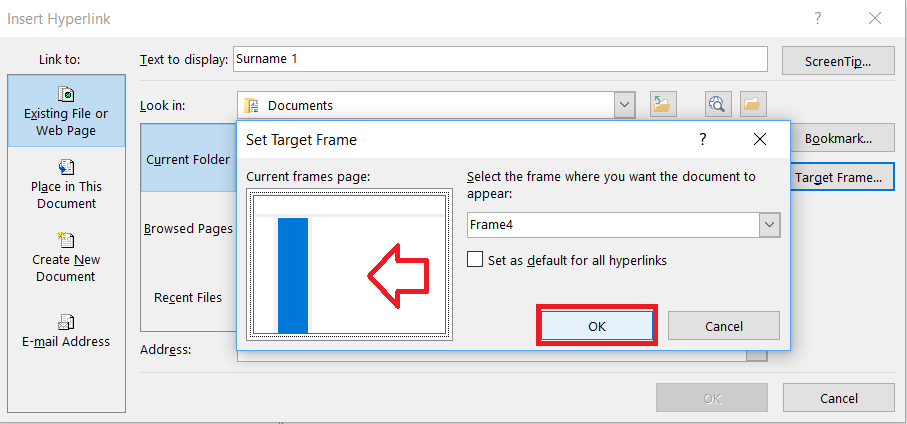
**Picture 20 - Create another word document in web format with the text you want**



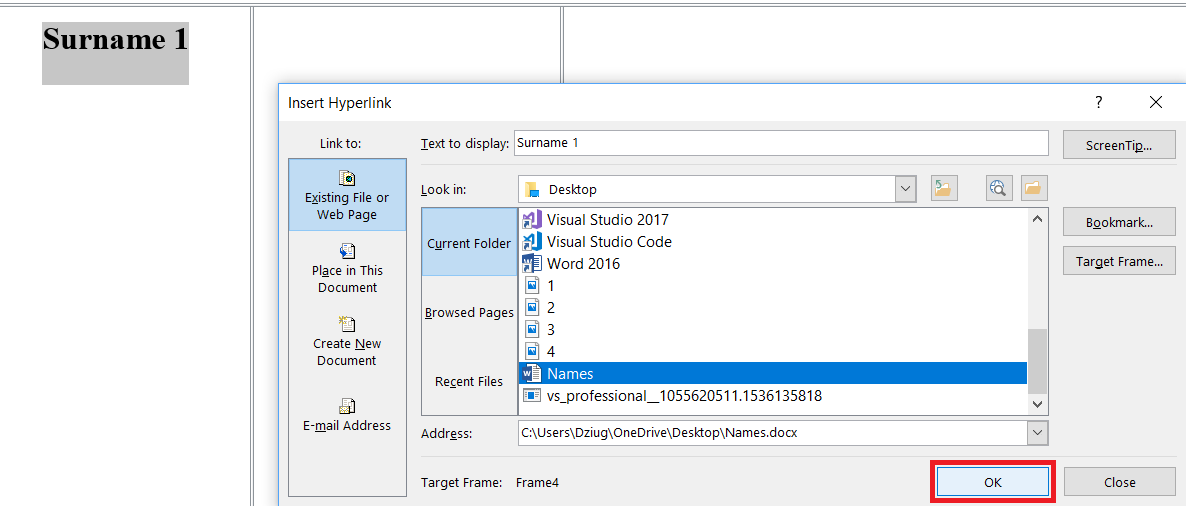
**Picture 21 - Click on the text and click Hyperlink**



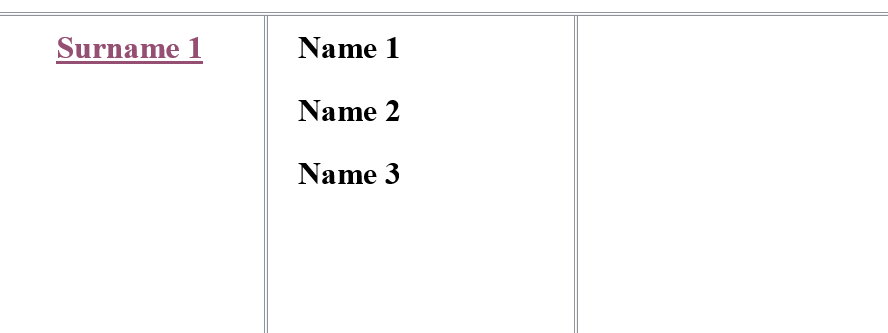
**Picture 22 - Click on Target Frame**



**Picture 23 - Target a frame you want the information to appear and click OK**

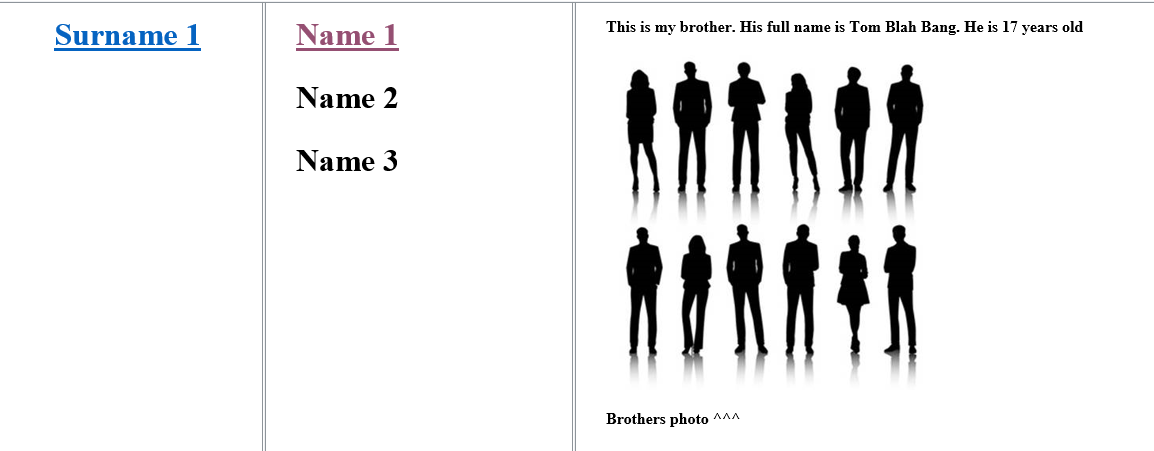


**Picture 24 - Find the document with the text you entered at first and click OK**



**Picture 25 - After clicking on the text in main page, it will open the new page**

1. And do this again with the rest. Just don’t forget to target different frames while adding a document.



**Picture 26 - Final look on a web page**

Don’t forget to put a **‘go back’** button and hyperlink it to a blank page on the same frame.

# QUESTIONNAIRE

# For the questionnaire you will have to use “Active” controls. On this kind of work, you want to make up 10 questions, 4 or 5 possible answers (one of them must be correct) and write them down. You can find an example of a mathematical test right [here](Mathematical%20test.dotm).

# And now let’s take a look on how to make this kind of questionnaire:

## **Adding basics to the questionnaire**

# Firstly, you want to open an empty word document and put a name for your test. Then make at least 3 text boxes for your name, surname and group, you can put an additional one for the score if you’d like (after adding these boxes, you need to right click them, click properties and name them to an easy name to make a code much easier to read). After you’re done with text boxes, then put one or two sentences about your test and how to complete it.

# C:\Users\Dziug\OneDrive\Desktop\Untitled.png

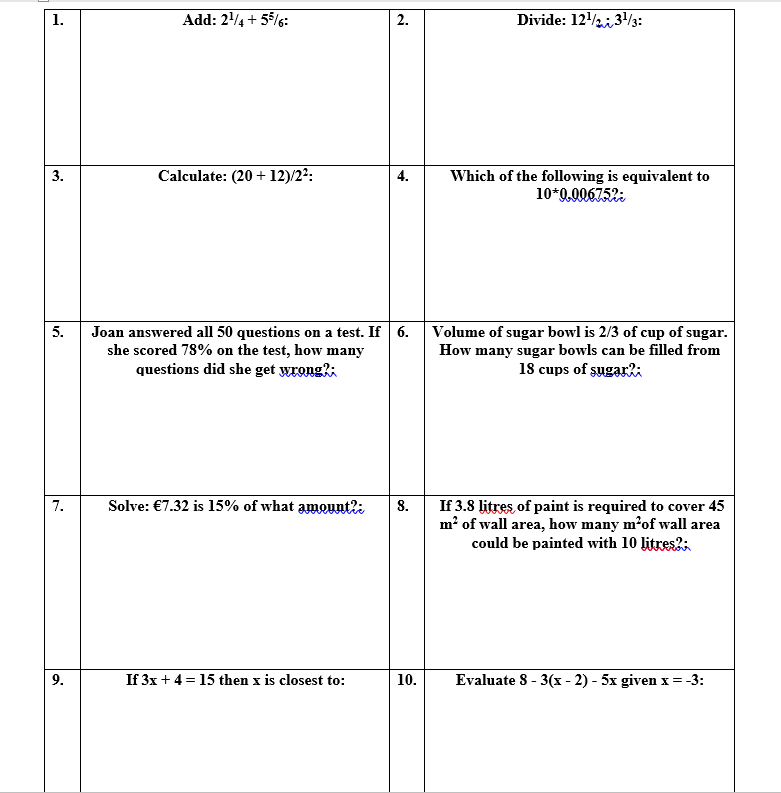
**Picture 27 - Beginning of a questionnaire**

# Next, you might want to create a table with numbers. Don’t forget to leave a bigger area because you’ll put a question and from 4 to 5 answers to that question. You might want to create a table just like that:

# C:\Users\Dziug\OneDrive\Desktop\Untitled.png

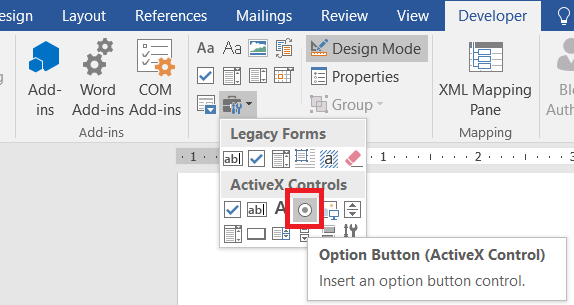
**Picture 28 - Making an empty table**

1. After that you will want to start making up your own questions for your questionnaire and write them down. You can see the example below:

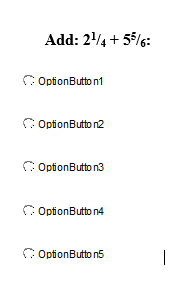


**Picture 29 - Writing down all the questions**

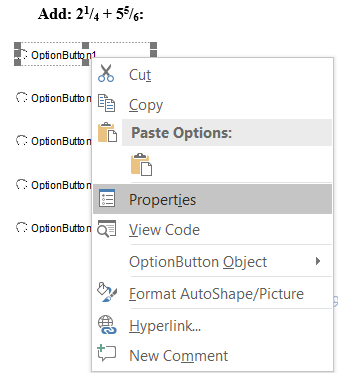
1. Now you want to add all the possible answers to your work. To do that you need to go to the **Developer Tab** and find the **Controls** group. In that group you might want to find a small circle with a dot on it (**Option Button**) and click on it. You might want to click on it for 5 times for 5 possible different answers to enter. Afterwards you need to name these option buttons that are matching for one question (you need to make the name of the group same to the other answers, otherwise it will not work.(**E.g. make a Group name for question 1 just Q1, for question 2 – Q2 and so on.** It will be a lot easier to understand the code since it will be a lot cleaner.)).



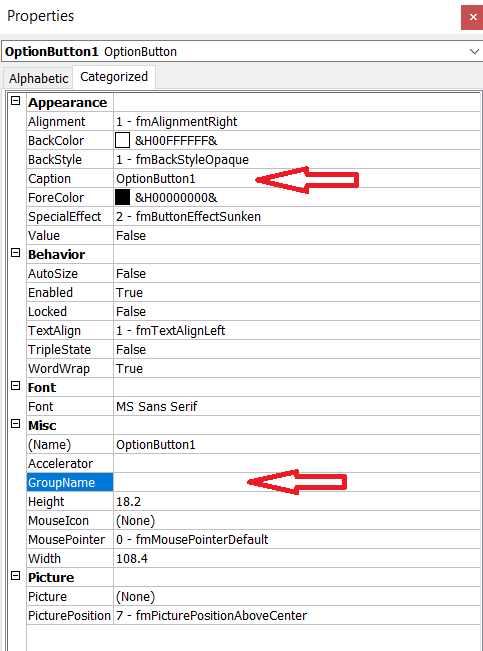
**Picture 30 - Choosing an Option button to make possible answers**



**Picture 31 - Add 4 or 5 option buttons for possible answers**

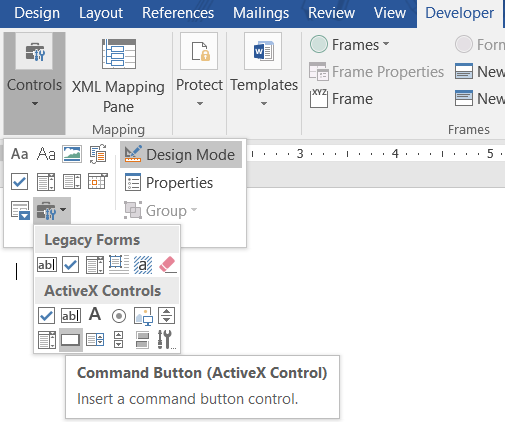


**Picture 32 - For every single button right click and choose properties**

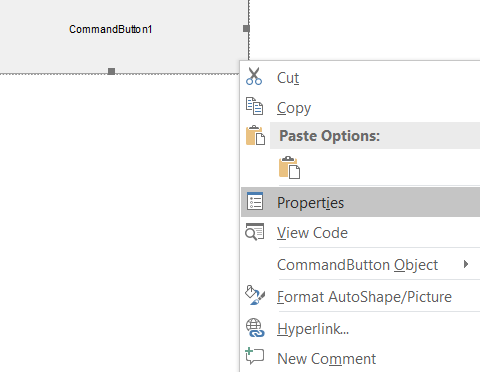


**Picture 33 - Insert an answer in the Caption block and question number in the Group Name**

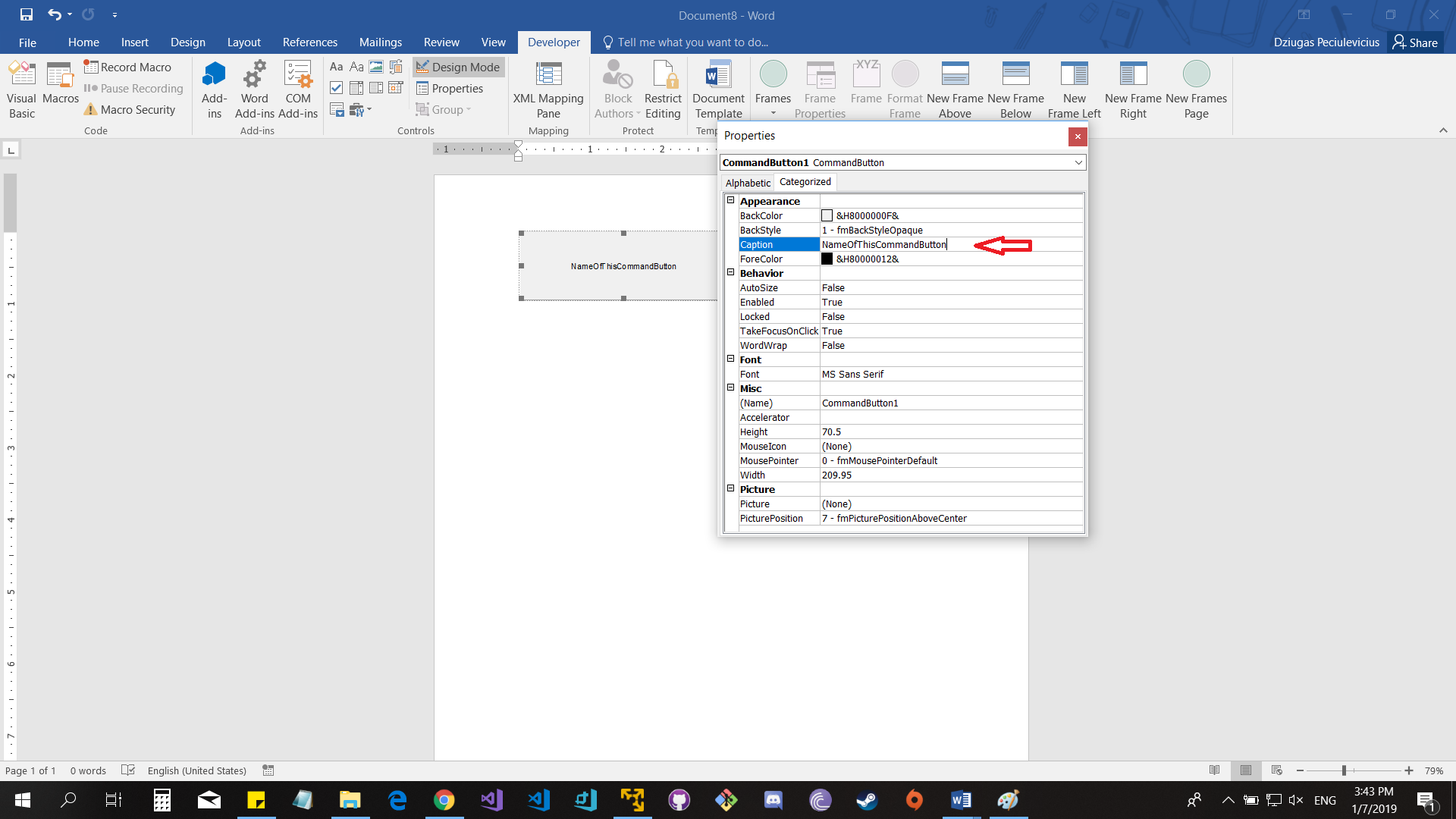
1. For the final visual touches, you will need to make at least 2 buttons: to submit and save your questionnaire and the second reset button – to clear the whole form and retake the test. To add these buttons, called Command buttons, you need to go to the **Developer Tab** and find the **Controls** group. In that group you might want to find an empty rectangle (**Command Button**) and click on it. You will need to at least click it twice: one for the **Submit** and another for **Reset** form. To name these boxes, you might want to do the same process that you have done with option buttons (you just don’t need to enter any group name for these buttons). Once you’ve done these, all you will have left is the VBA code.



**Picture 34 - Find the Command button option in the Developer tab and add it**

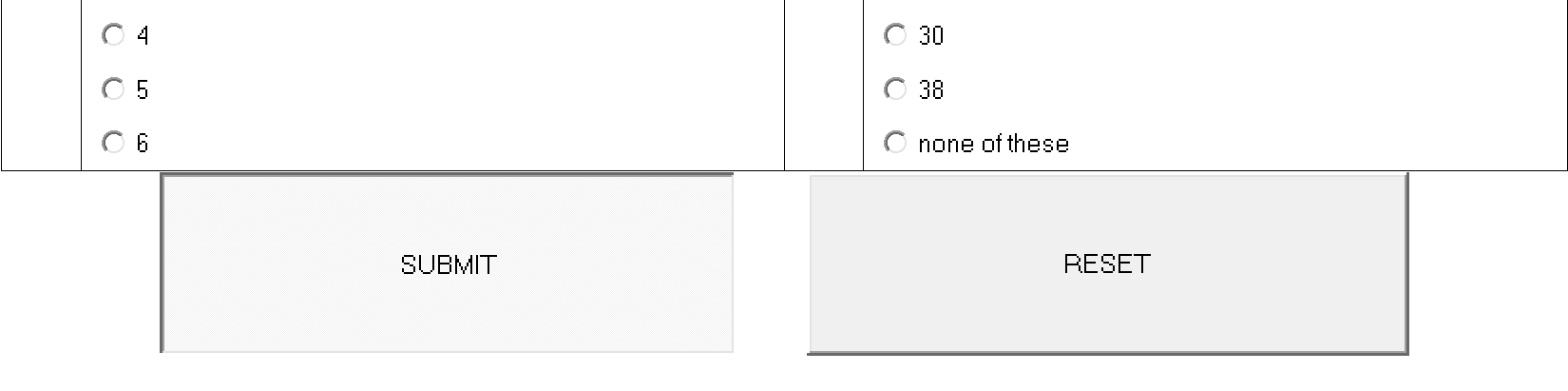


**Picture 35 - Once Command button is added, go to buttons properties**

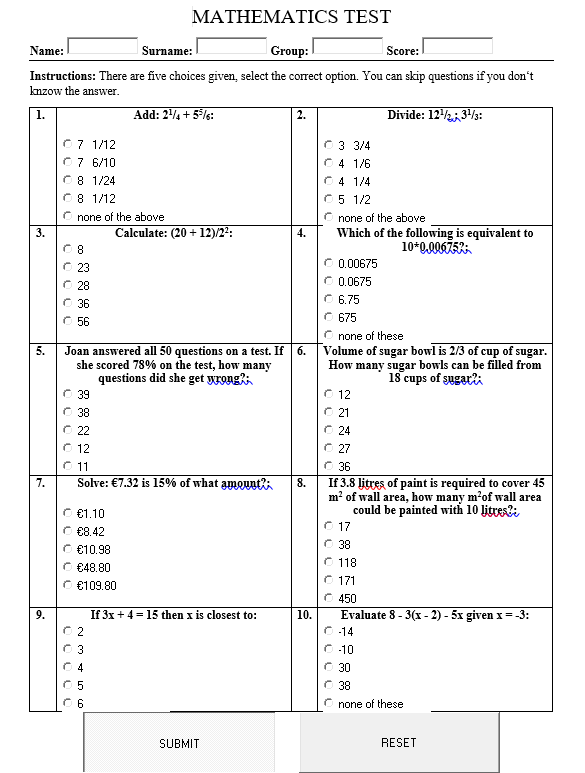


**Picture 36 - Name your Command button**

After you have added both buttons and name them as you like, you will get a look of something like this:



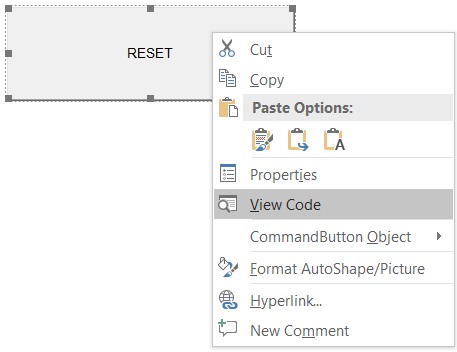
**Picture 37 - This is how Command buttons should look after you're done**



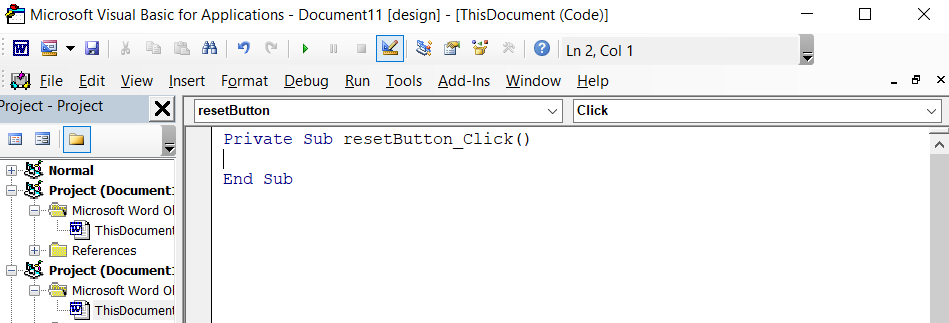
**Picture 38 - Final look on the visuals**

## **Adding Visual Basic for Applications (VBA) code**

1. So to begin with, it’s a little bit simpler to explain the **RESET** button first, since it involves a lot less code to be written into the Visual Basic. After finishing the test and if you want to do it again, you will need to reset your test to the previous state. Since we want to make a Reset button to reset all values to False (previous state) we will have to do in in the VBA. It is explained step-by-step below on how to write the VBA code for the **RESET** button:

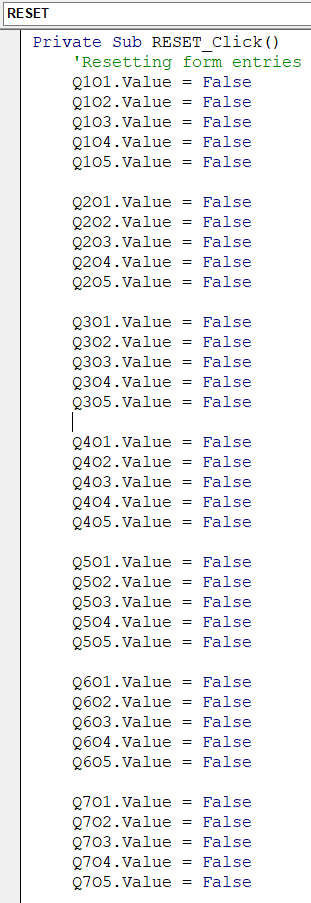


**Picture 39 - Right click on the RESET button and then click on View Code**



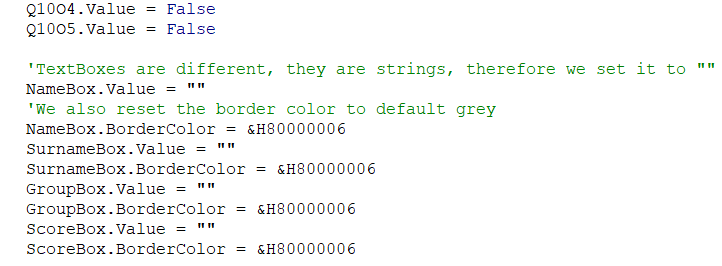
**Picture 40 - You will get this empty field for your code**

1. Now you want to start writing an actual code for your command button. To start with we know that we want to reset all values to what they were just before starting the test (from Q1O1 to Q10O5 (Question10Option5), so we set all values to **False.** Here’s an example:



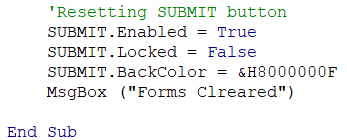
**Picture 41 - Setting values to each of the option of answers**

1. After it is done with setting all the option values to false, we will want to reset all the values of text boxes above and to set it to its default color which is grey. When we hit that **RESET** button, we will want for text boxes to be deleted as well so we set all the values of text boxes to **“ ”** (BoxName.Value = **“ “**). When we reset all the text boxes, we also want to set these box colors to grey, so we after every box we set the values to, we might want to set the color to grey, so we type BoxName.BorderColor = &H80000006 (this is a color code of grey). All this information is displayed below:



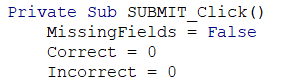
**Picture 42 - Setting all the default values and colors for the text boxes**

1. After setting all the values to default ones, the last step for the reset button code would be to write a code that resets the other command button which is a **SUBMIT** button. SUBMIT.Enabled = True and SUBMIT.Locked = False are the commands that actually lock or unlock the **SUBMIT** button. If the **RESET** button worked and the statement changed to True, then **SUBMIT** button will be enabled, otherwise – disabled.



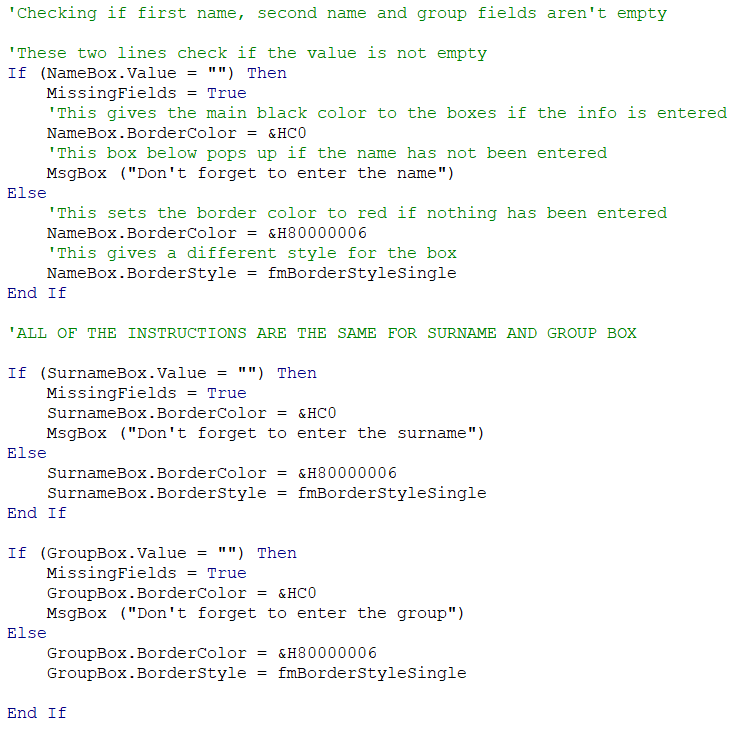
**Picture 43 - Resetting the SUBMIT button**

1. Now it’s all covered for the **RESET** button. Let’s move up to the button that saves the test, calculates the score and checks if all information is provided and displayed on the screen. It is called the **SUBMIT** button. So if we want to make the **SUBMIT** button properly work we need to specify if there are missing fields, to make it not work and count the score of this test.



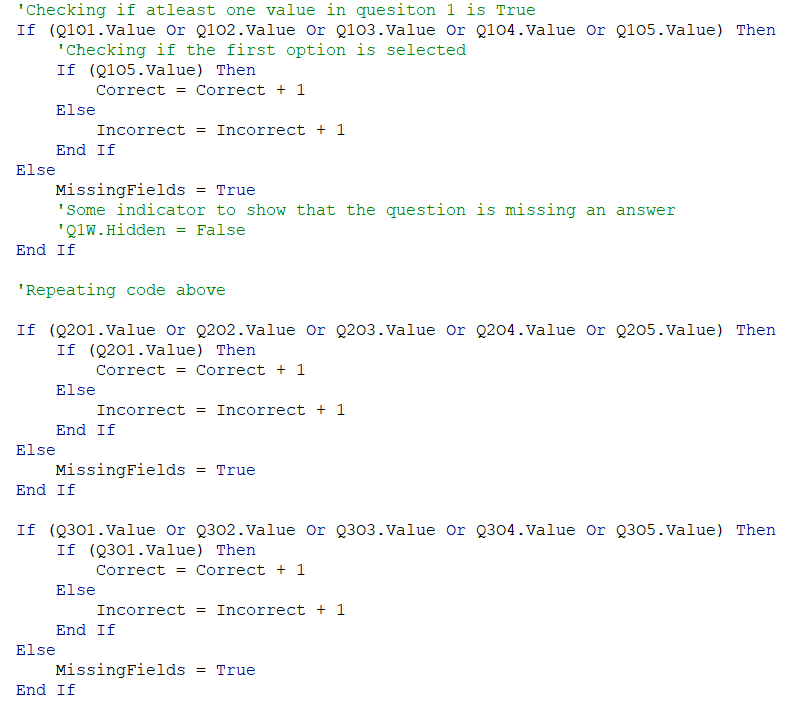
**Picture 44 - Checking if there are empty lines**

1. After writing that little code, we’ll move to a bigger one. In this code we want for **SUBMIT** button to check if the Name, Surname and Group text boxes are entered and are not empty. There’s some in-depth explanation going on in the code witch is displayed below:



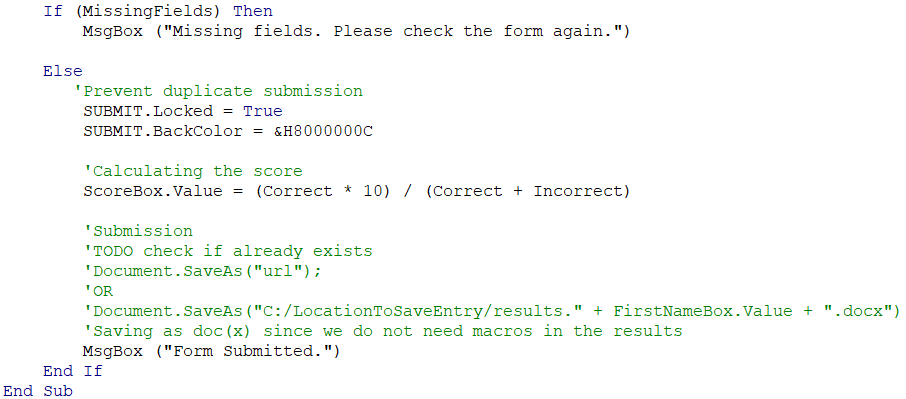
**Picture 45 - Setting text boxes up**

1. After writing a code for name, surname and group text boxes, we will want to start writing one for all the correct option buttons for each question just right below the code for setting up all the text boxes (in the same SUBMIT\_click code box). There’s an example for the code below, but only for a few questions. You will want to do the rest of the questions.



**Picture 46 - Code for the option buttons. Making a choice for a correct one**

1. There’s not much left now for the questionnaire. In the same box (SUBMIT\_click code box) all you want to write now is just to make a message box to pop up if there’s any missing fields. And if everything is okay and there are no missing boxes all you have to do is make the questionnaire lock itself and prevent it from duplicating itself. After clicking the **SUBMIT** button, you will also want to make the test show you the score of correct answers and to save and submit the results.



**Picture 47 - Final bit of code for the questionnaire**

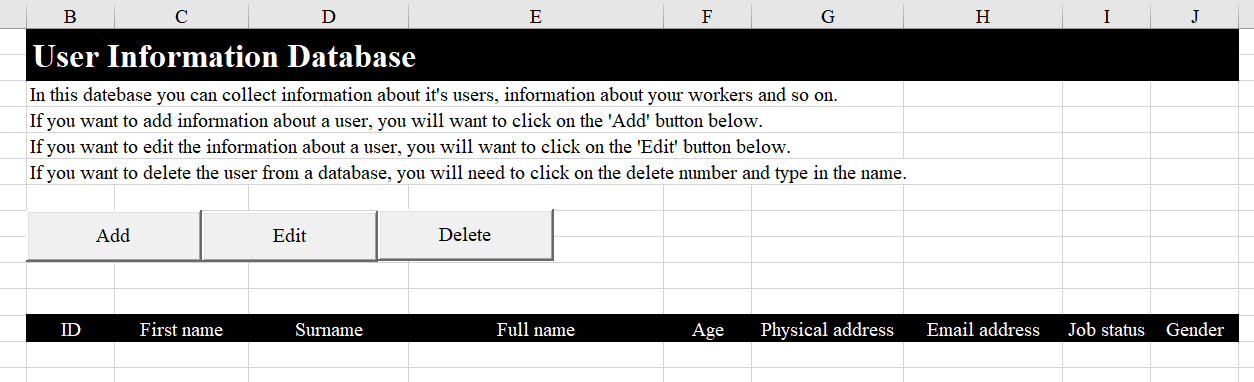
# WORKS DONE OUT OF EXCEL

# After all of these works above we came to probably one of the most important works of all. There are two jobs done with excel which is pretty simple so it not be explained here in this document, but I’ll drop a hyperlink [here](rolling%20dice%20excel/rolling%20dice.xlsx) so you can check it out if you’d like. So another one which is a user form based database is a lot more complicated and has a lot of work to be done. You can see it by clicking [here](Userformdatabase.xlsm).

## **UserForm Database**

So this work probably involved most of the work and it took the longest to complete (not finished yet). With this piece of work, you will be able to add users to a database, edit and delete them. In this document below it is explained how this database was done, so you’ll be able to maybe even do it yourself.

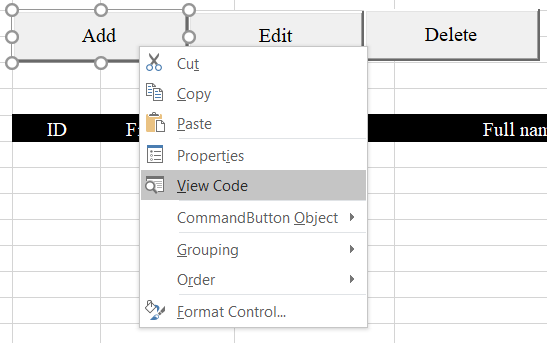
So firstly, you will want to create a simple “look” for your database with few buttons, here’s an example:



**Picture 48 - Start of a database**

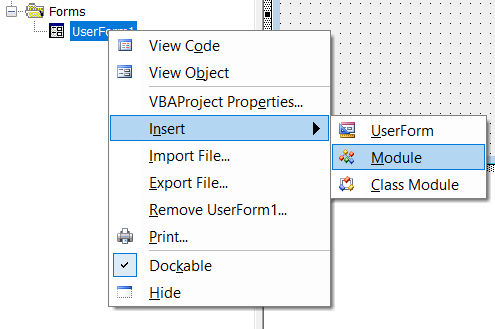
Now you will want to add a user form to the ‘Add’ button. To do that, you want to follow the instructions below:

1. Firstly, you want to right click on the ‘Add’ button and then click ‘View Code’

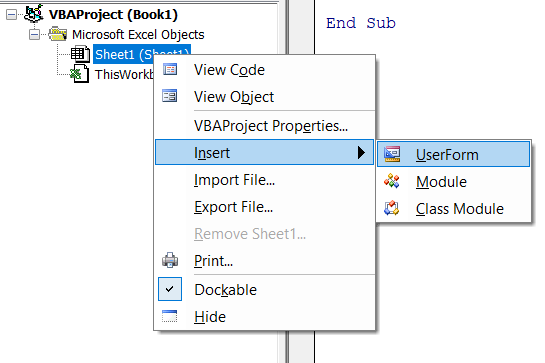


**Picture 49 - Editing the 'Add' button**

1. Now you want to insert both **Modules** and a **UserForm.**

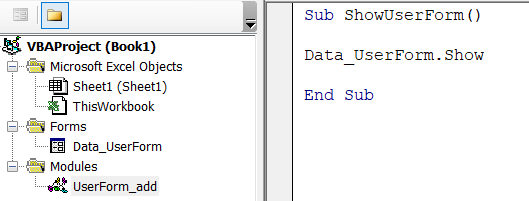


**Picture 50 - Adding Module**



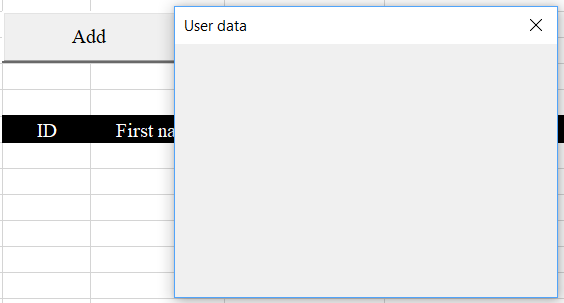
**Picture 51 - Adding UserForm**

1. Now you will want to link the user form to the modules, so firstly you will want to name the user form to something like Data\_UserForm. Once you’ve done that, you can link it to the modules just like this:

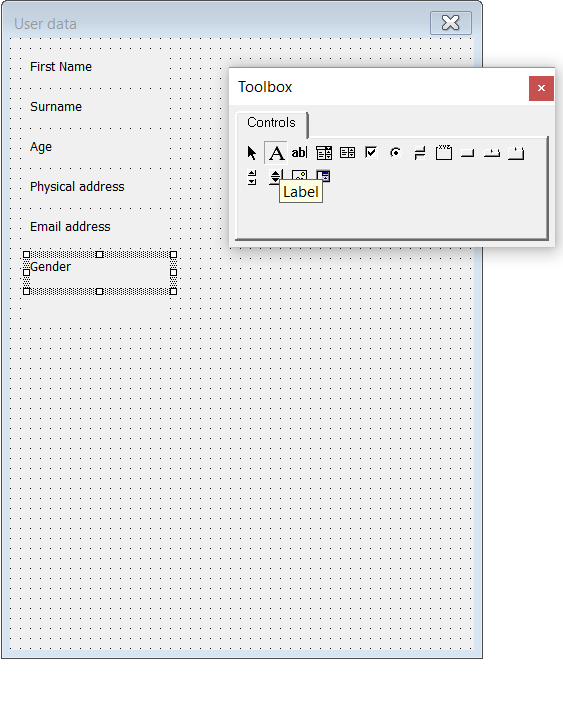


**Picture 52 - Linking User Form to the 'Modules" block**

1. So now since you linked your user form to the modules, you will want to click on the user form and start making text boxes and labels for it.

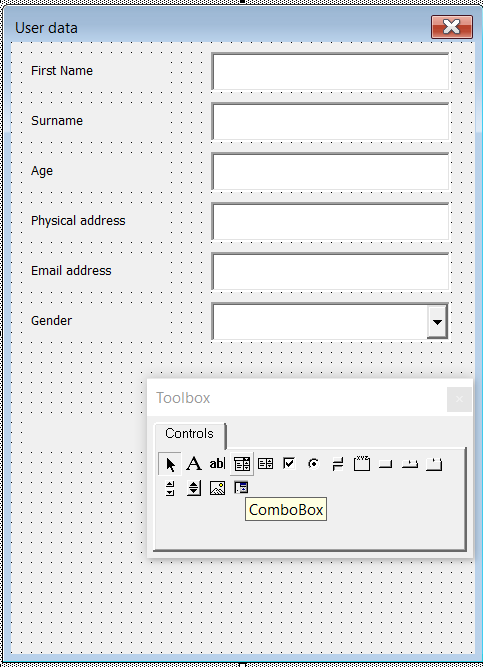


**Picture 53 - Making of User Form**



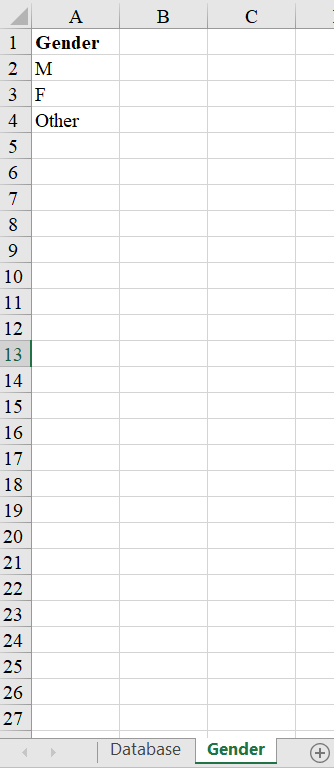
**Picture 54 - Putting labels and text boxes into the user form**

1. You will also want to make some ‘ComboBox’es” for your user form. To add a Combo Box, it’s pretty simple, all you need to do is just to follow these steps below:



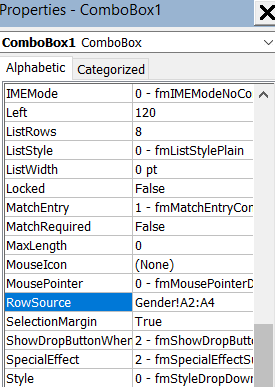
**Picture 55 - Adding a ComboBox**

For a combo box to work, you will want to make a different sheet for it. You will want to write all the choices there and then only choose the stats which values you only want to use in your user form.



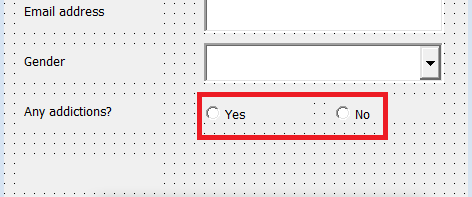
**Picture 56 - Creating a sheet for a Combo Box**

Now all you need to do is just to choose which boxes you want to display into your user form. Change your **‘RowSource’** to the lines you like to display.



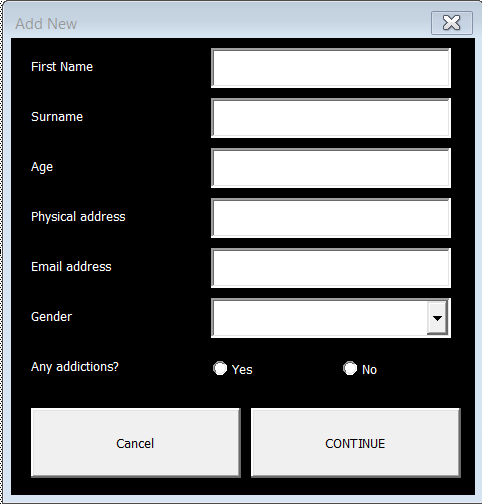
**Picture 57 - Choose the rows you want to display from the previous sheet slide**

1. You will also want to include some option buttons to your username. Don’t forget to name it to ‘Yes’/’No’ and define it to one group. You might also want to define one of these buttons to True for it to be already marked when opened and so user couldn’t just leave it blank.



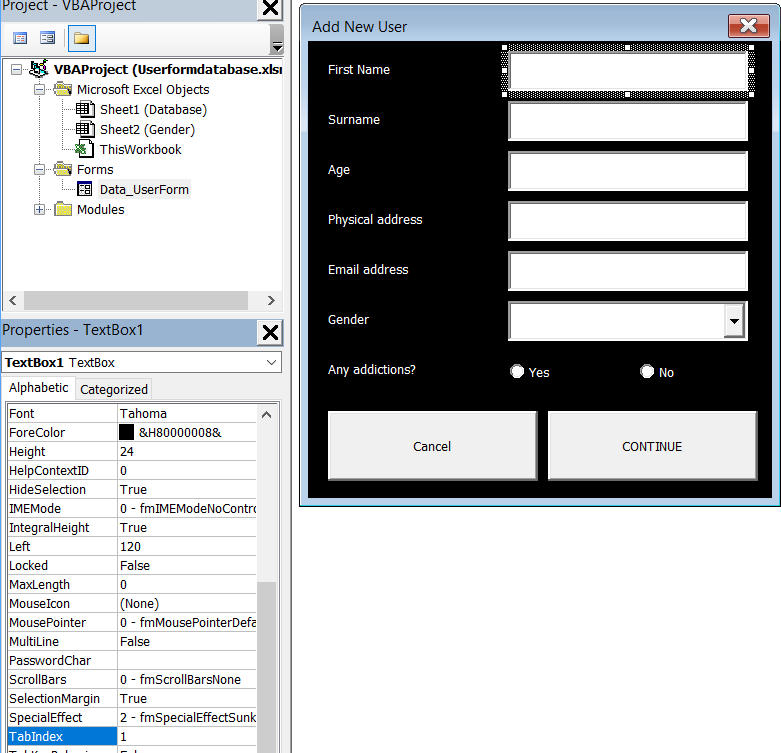
**Picture 58 - Option Buttons**

1. For the ending, you can put Command Buttons for ‘CANCEL’ and ‘CONTINUE’ keys and change the background color.



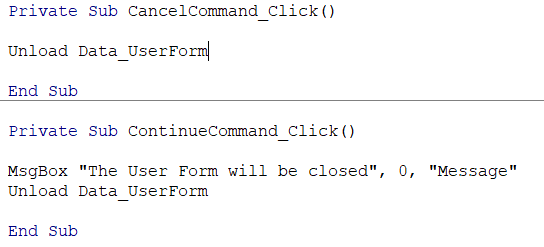
**Picture 59 - Final user form**

1. To make user’s life easier to navigate on the user form, we can make it, so it would work with keyboard. And we would want to make it go to the next box when we press **‘Tab’,** so to make that work, we will need to click on each white text box individually and then go to properties on the left and find **‘TabIndex’** and change it to the number depended on the queue of these boxes (1; 2; 3….).



**Picture 60 - Final steps of the user form**

1. And also, if you want to make these command buttons work, you need to write a code which is displayed here (it will change a little bit once code goes on). Unload Data\_UserForm closes the form. This message box below will be displayed to (“FullName” has been added **or** There’s already a user named “FullName”:



**Picture 61 - Writing a little bit of code for the 'Cancel' and 'Continue' buttons**

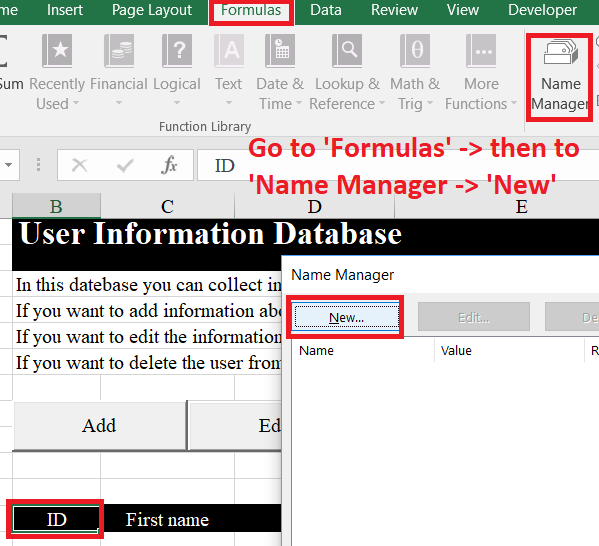
1. After writing this code, we will want to make some adjustments to see how many entries we have and also to make the the entered users go down. Make a new sheet and name it 'Engine'. After that you can go into the engine sheet and type this in:



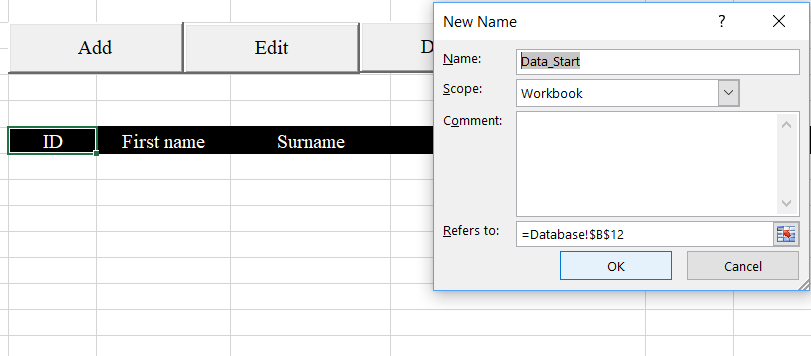
**Picture 62 - Line Counter for the database**

## **Writing the VBA code for the ‘Add’ button**

1. Now to start writing the code for the user form, we will need to make a data starting point, it will be an ID sign, but we will have to write some code for it. You can check all of these 5 pictures and see what code has been written before and so on.

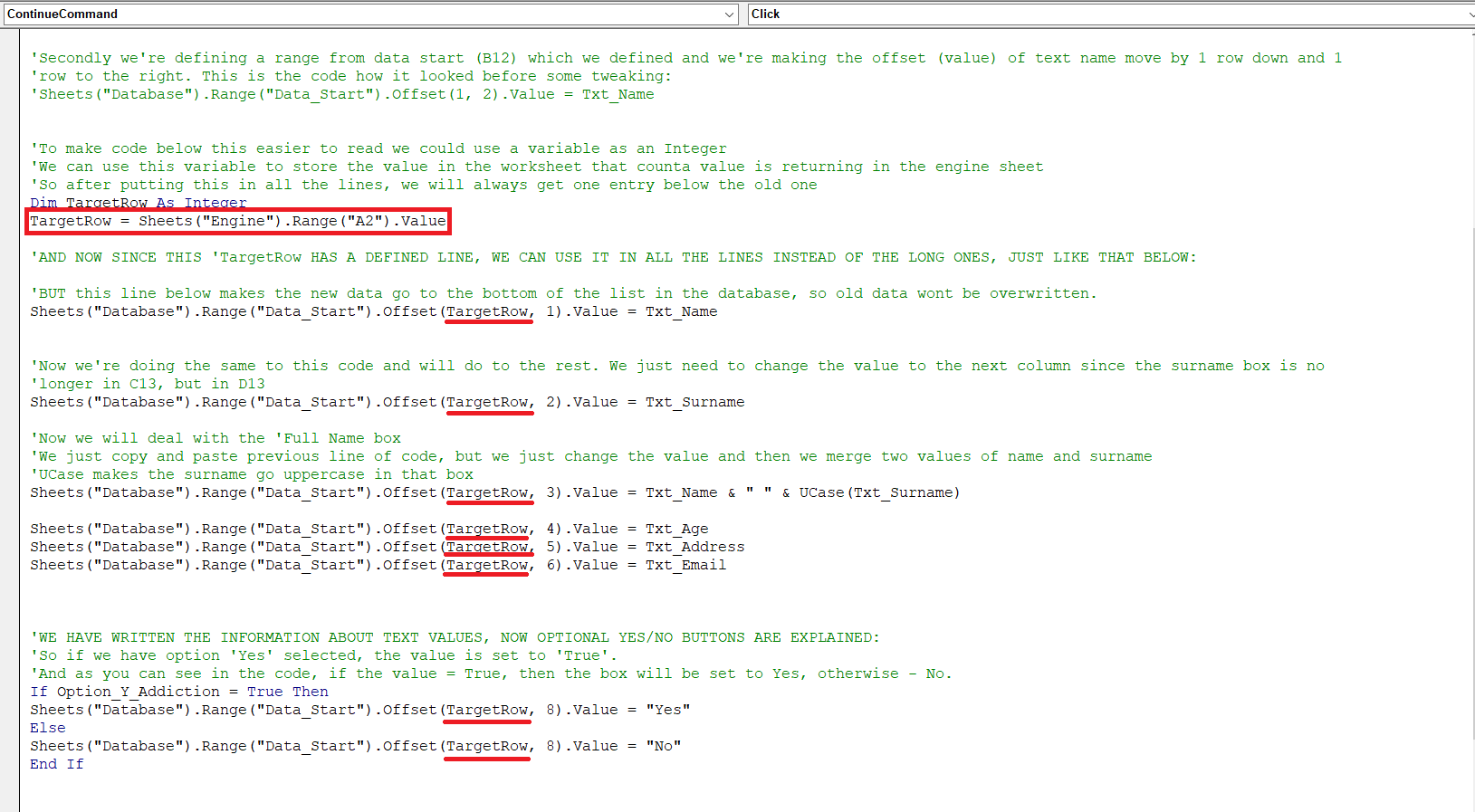


**Picture 63 - Adding a "starting" point**



**Picture 64 - Adding a "starting" point**

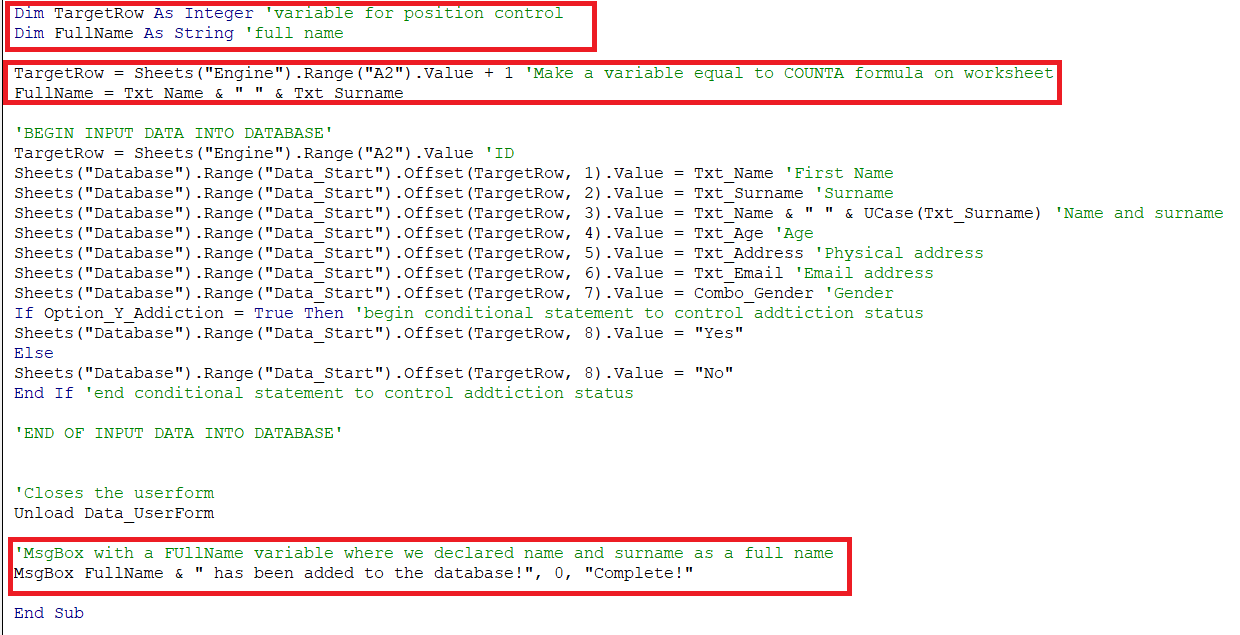
1. To save some time, you can check this picture out below what was the code before, what it is right now and how it works.:



**Picture 65 - Targeting rows**

Creating some user interaction and giving the user some information on what just happened. On this next picture, it is displayed what kind of code was written to make Full name appear at the end after clicking the **‘Continue’** button, so it would look something like this: “Name Surname has been added to the database”.

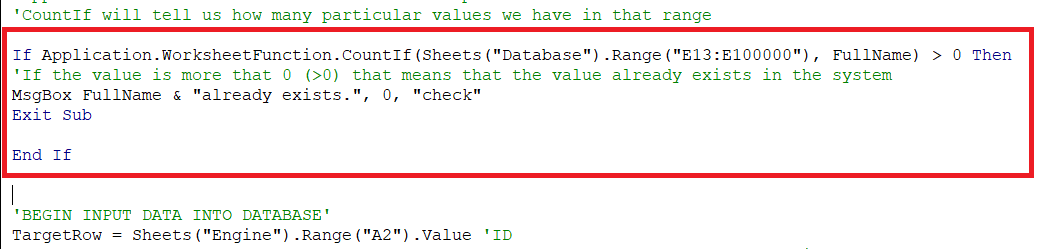
1. So for the position control we declared the TargetRow variable as integer on every row to make it go down once every time the user form has been produced and give as a number. And for the message box at the end we declared a value of FullName as a string. Then we want to assign name and surname values to the full name variable so we can just add full name value at the end. Here’s an example:



**Picture 66 - Making a message box and an ID number**

Now we will want to start writing a code for user form to check if the user already exists in the database after clicking the continue button. It’s pretty short and not really hard to understand.

1. So we already defined the full name variable at the top and assigned a value to that variable. Now we will want to check if the full name value appears in the FullName column on the database. So here’s an example and an explanation below the picture:

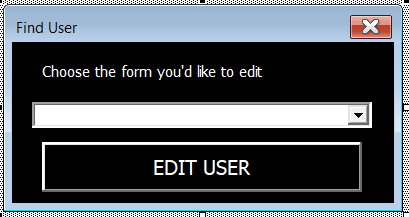


**Picture 67 - Code for checking if the FullName value already exists**

The **‘Application.WorksheetFunction’** is a pretty useful command in VBA. It allows us to access worksheet formally in the Visual Basic editor, so we don’t have to put any formulas in the worksheet. So we will want to add a main formula that we’re using to this application command which is **‘CountIf’.** So now we want to add a sheet, range and a value that we’re looking for. So in the database, we want for full name to appear in the range from E13, so that’s why the range starts from E13. And for the value, if there’s a character already inputted and value is greater than 1, that means that a user already appears in the database. And last, but not least, we will want to add a message box for our user.

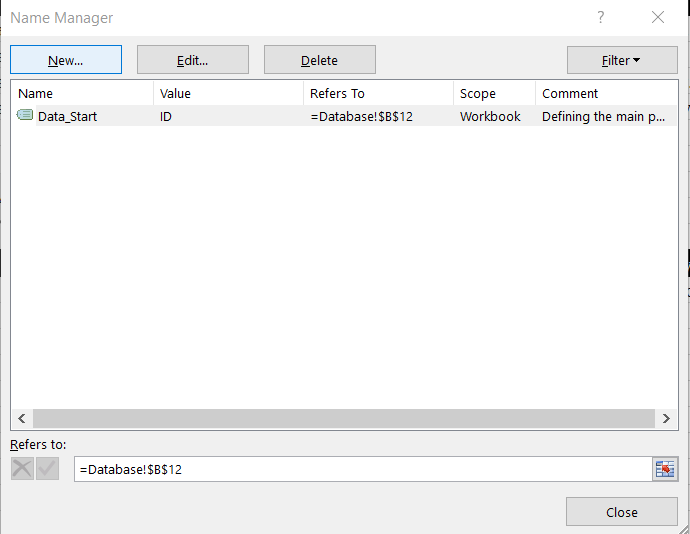
## **Writing the VBA code for the ‘Edit’ button**

1. Now we want to do the same with the **‘Edit’** button as we did with the **‘Add’** button at the beginning. For now, all we want to do is to add the button, click on it, link the user form to the modules, then assign it to macro and edit your user form to look something like this:

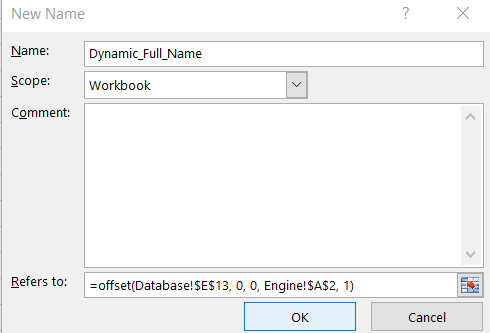


**Picture 68 - Find User to edit**

1. After this is done, you will want to name all the buttons and make the combo box search for the users that already exist in the database. So for a combo box to search for the entries, you will want to click on the combo box and then find **‘RowSource’** in the properties. And to make it not show any empty rows in the entry, we will want to do define a dynamic name. All the steps will be explained below.

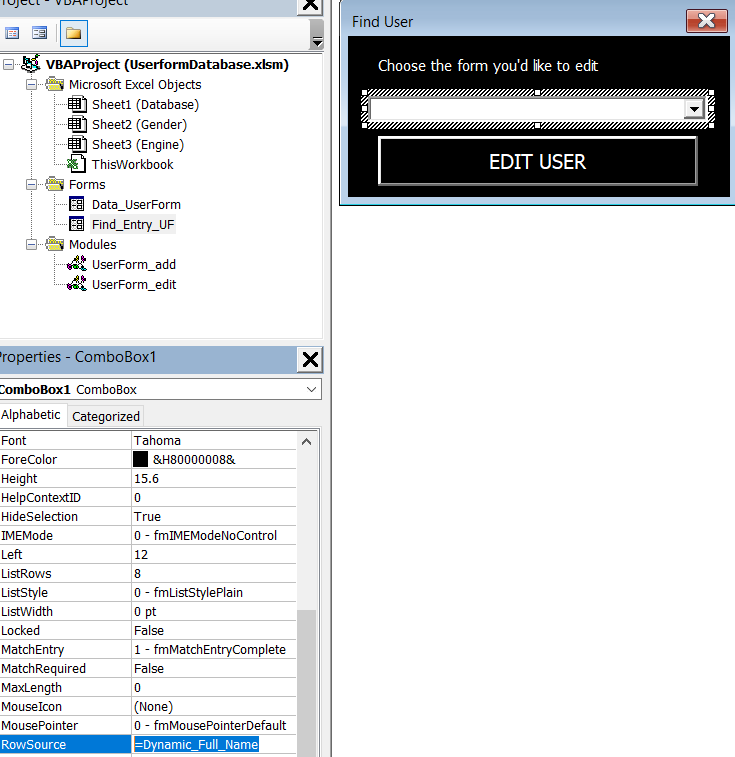


**Picture 69 - Create a new name in Name manager**



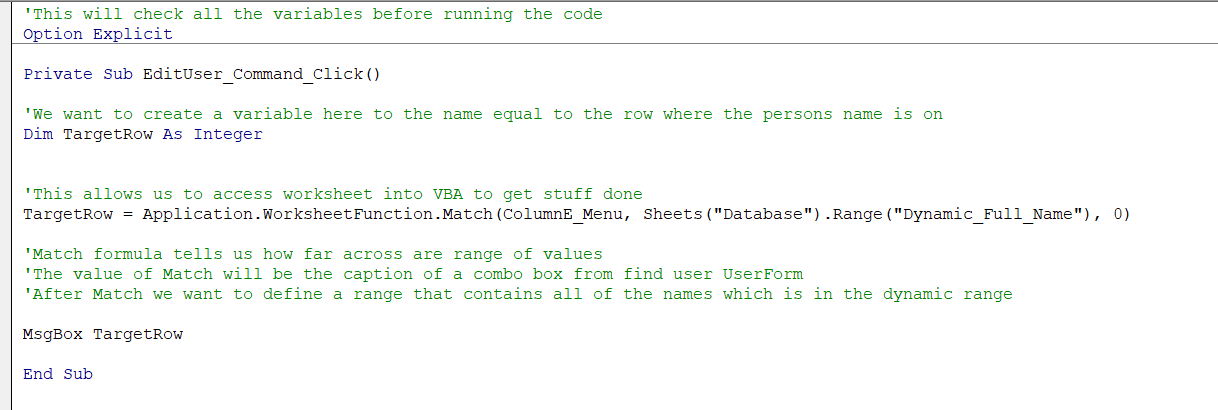
**Picture 70 - Creating a dynamic name**

1. For the offset command, will want to specify the full name column and then put zeros because we will not be changing any values. Then you will want to add a sheet and a box where your ‘counta’ command is and define it as 1, because we want its size to stay at 1.
2. Then we want to save this name and paste the name into the user form combo box properties, just like in the example below:



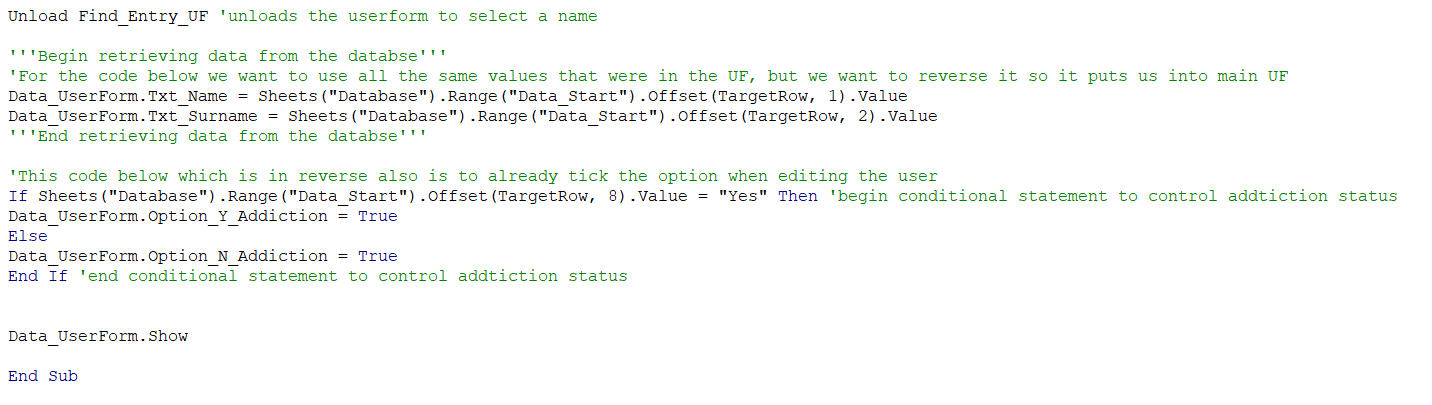
**Picture 71 - Inserting a dynamic name into the user form combo box**

1. So the code below we made was to check if the edit form sees the row number in the database which user that is:



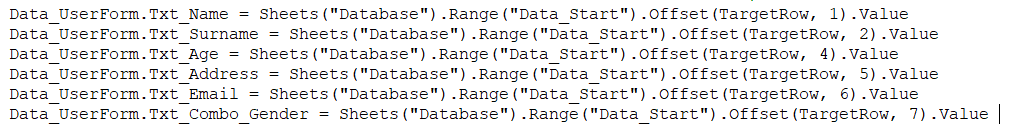
**Picture 72 - Code for checking the number of the user in the database**

1. So now we kinda want to make a user form box to pop up when we search for a user, soo we could edit existing contact. But we‘re not done yet, since it would add a new address.



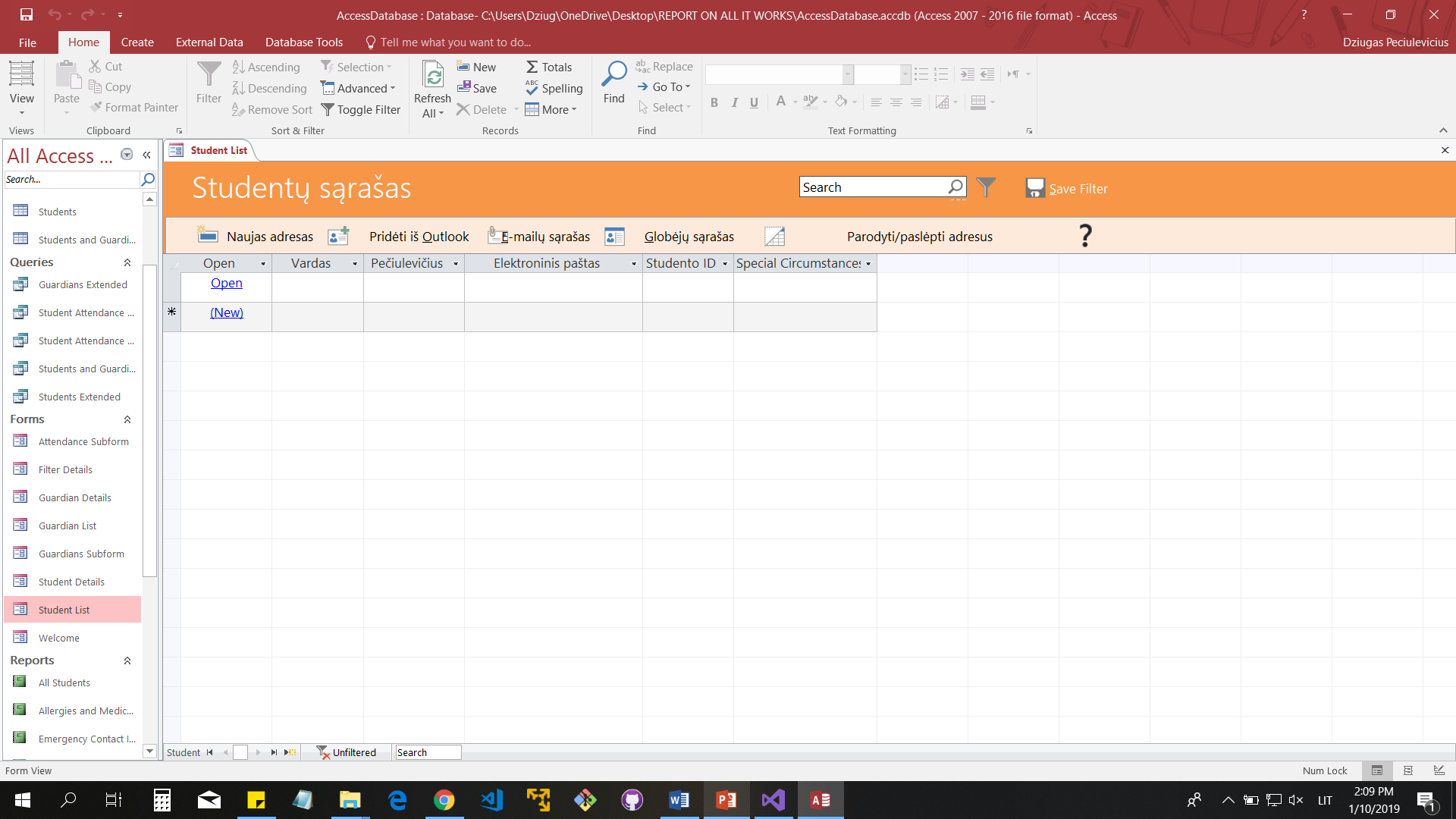
**Picture 73 - Linking the edit for to the user form**

1. We also want to finish the recent code and add all the values:

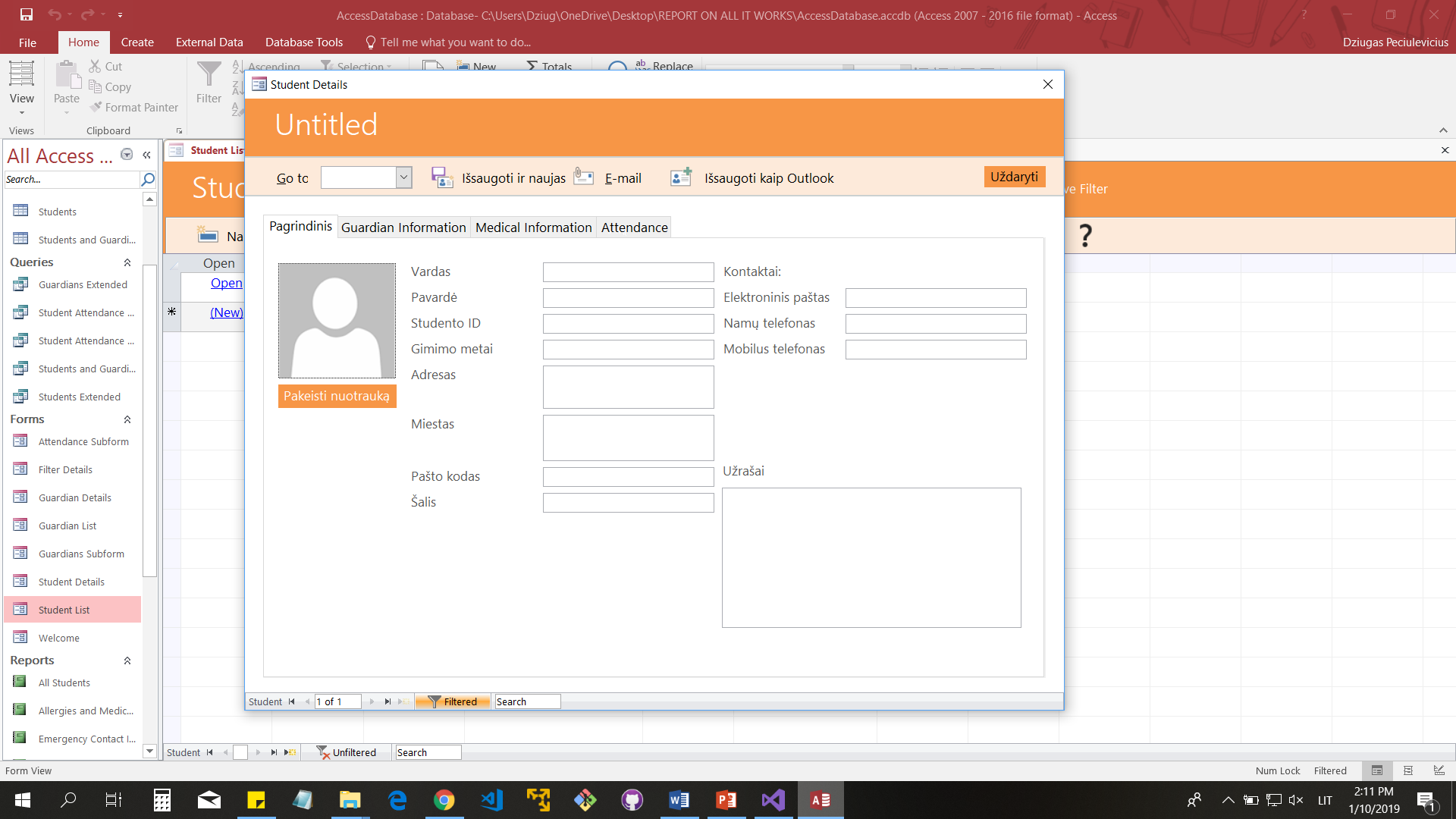


**Picture 74 - Adding left over values**

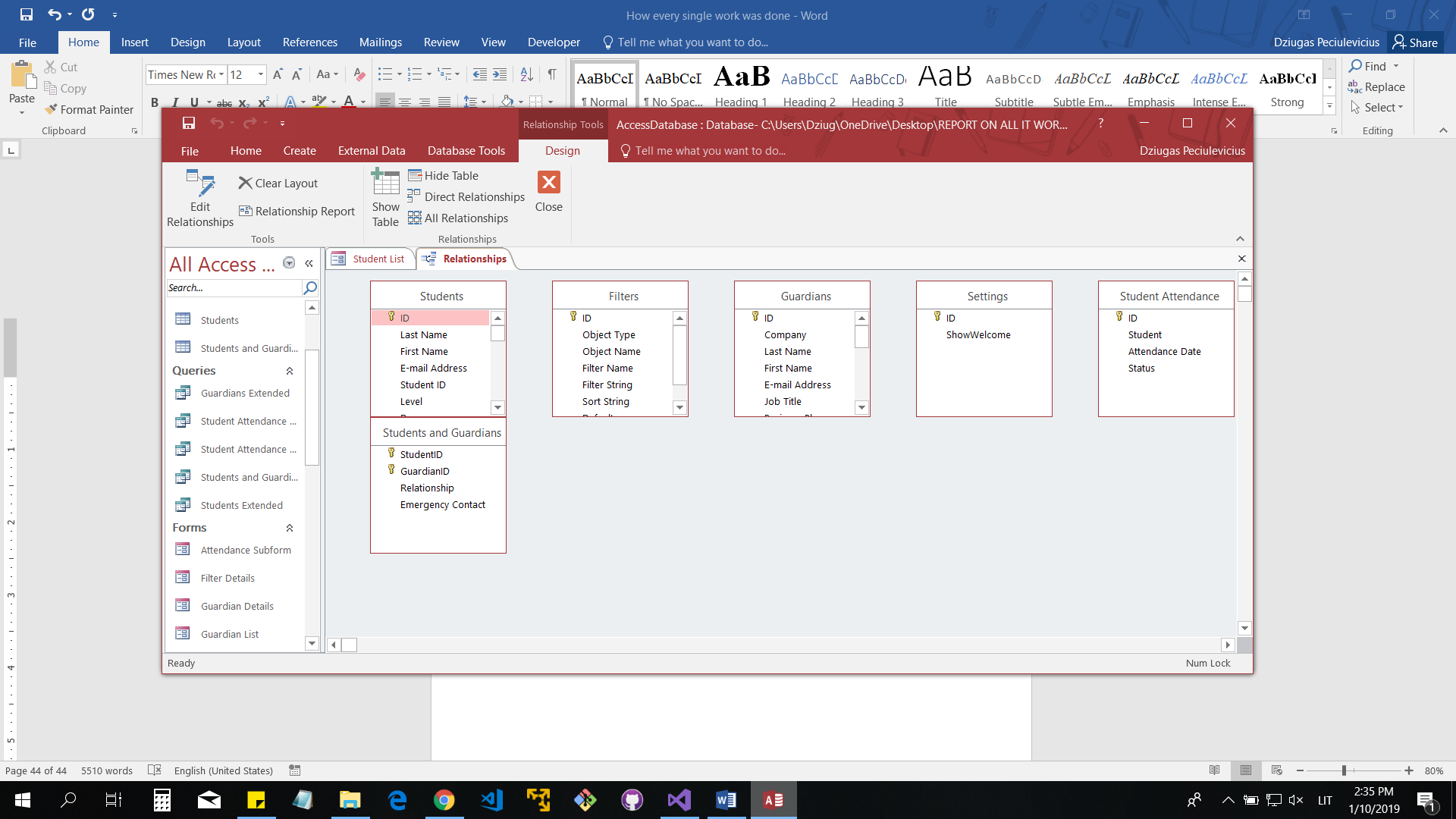
# ACCESS

For Microsoft Access all we did is take already existing template and edit it to Lithuanian words. 

**Picture 75 - We rename all the values**



**Picture 76- We also rename all the values on the user form**



**Picture 77 - This is the Relationship table, it was empty so I included.**