**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**GRADUATION RESEARCH 2**

**Excel Add-ins and Functions**

**PHAN DINH TUONG**

Tuong.pd164582@sis.hust.edu.vn

**School of Information and Communication Technology**

**Global ICT Program**

|  |  |
| --- | --- |
| **Supervisor:** | MSc. Nguyen Duc Tien  Signature of Supervisor |
| **Department:** | Computer Engineering |
| **School:** | Information and Communication Technology |
| **HANOI, 12/2020** | |

**RESEARCH TOPIC**

Write Excel add-ins and functions with custom functionalities, compile to dynamic link library ( .dll and .xll files) to run in Excel spreadsheet.

Supervisor

Signature and full name

**Acknowledgement**

For the second time, School of Information and Communication Technology, Hanoi University of Science and Technology, created another chance, the Graduation Research 2, for students to research on the interesting aspect of their study field. I truly appreciate that.

I would like to thank my supervisor, MSc. Nguyen Duc Tien, Department of Computer Engineering, School of Information and Communication Technology, Hanoi University of Science and Technology, for his guidance, although it is the mid-night, or his busy day with other works.

Once again, wish you all a happy and successful life!

**About research**

The research inludes 2 projects, ExcelAddins and ExcelFunction. The two projects provide the specific real problem of Exel add-ins and functions. The details are in the main content.

These two projects are written in C# programming language, IDE Microsoft Visual Studio Community 2019 (Version 16.7.5), with NuGet Package Manager, ResXManager, Github to control source code remotely.

Student

Signature and full name

Sinh viên thực hiện

Ký và ghi rõ họ tên

**TABLE OF CONTENTS**

[CHAPTER 1. MAIN PROGRESS 1](#_Toc59800038)

[1.1 Cell colorization 1](#_Toc59800039)

[1.1.1 About 1](#_Toc59800040)

[1.1.2 Detail 1](#_Toc59800041)

[1.2 QRCode generator function 3](#_Toc59800042)

[1.2.1 About 3](#_Toc59800043)

[1.2.2 Detail 3](#_Toc59800044)

[1.3 KipThi function 4](#_Toc59800045)

[1.3.1 About 4](#_Toc59800046)

[1.3.2 Detail 5](#_Toc59800047)

[1.4 InText function 5](#_Toc59800048)

[1.4.1 About 5](#_Toc59800049)

[1.4.2 Detail 5](#_Toc59800050)

[1.5 StringCellFormatter and StringFontFormatter functions 6](#_Toc59800051)

[1.5.1 About 6](#_Toc59800052)

[1.5.2 Detail 6](#_Toc59800053)

[1.6 Cortana voice recognition 6](#_Toc59800054)

[1.6.1 About 6](#_Toc59800055)

[1.6.2 Detail 6](#_Toc59800056)

[1.7 ArrowMouseScroll 6](#_Toc59800057)

[1.7.1 About 6](#_Toc59800058)

[1.7.2 Detail 7](#_Toc59800059)

[CHAPTER 2. HOW TO BUILD AND USE 8](#_Toc59800060)

[CHAPTER 3. CONCLUSION 12](#_Toc59800061)

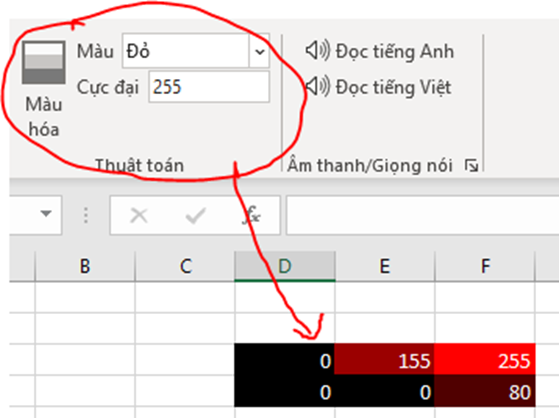
[REFERENCES 13](#_Toc59800062)

# MAIN PROGRESS

## Cell colorization

### About

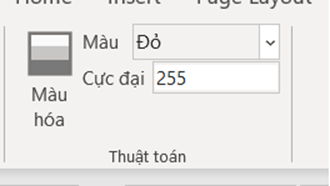
* This feature belongs to ExcelAddIn project.
* User chooses a range of cells, with selected color (red, green, blue) and saturation.
* User clicks on “Colorize”.
* The chosen cells will be colorized with corresponding color and saturation.



* Update: Add gray to color, change font color based on cell color (if saturation > 128, font color is black, otherwise, the font color is white), keep input text.

### Detail

* In group Algorithm (Thuật toán) on Add-ins tab, there are a button Colorize (Màu hóa), drop down ColorRGB (Màu), edit box editSaturationPeak (Cực đại) as the figure:



* In MyRibon.Designer class, buttonColorize\_Click is invoked whenever user click **Colorize** in MyRibon class with arguments color and saturation:

this.buttonColorize.Click += (sender,e) => this.buttonColorize\_Click(dropDownColorRGB.SelectedItem.OfficeImageId,editSaturationPeak.Text);

* In method buttonColorize\_Click in MyRibon class, the cells are selected:

//get selected cells

Range currentRange = (Range)Globals.ThisAddIn.Application.Selection as

Microsoft.Office.Interop.Excel.Range;

if (currentRange == null) return;

* The saturation value is checked if it is valid:

int saturationInt; //saturation value in integer

bool isNumeric = int.TryParse(saturation,out saturationInt); //boolean to check if saturation is a valid number

if (isNumeric) //check if the saturation is a number

if (saturationInt >= 0 && saturationInt <= 255) //check if the saturation value is in range 0 to 255

* If valid, the color to colorize is checked and the saturation value is assigned to the cells:

switch (color) // color to display with saturation

{

case "AppointmentColor1": currentRange.Interior.Color = Color.FromArgb(saturationInt, 0, 0); break; //red

case "AppointmentColor2": currentRange.Interior.Color = Color.FromArgb(0, 0, saturationInt); break; //green

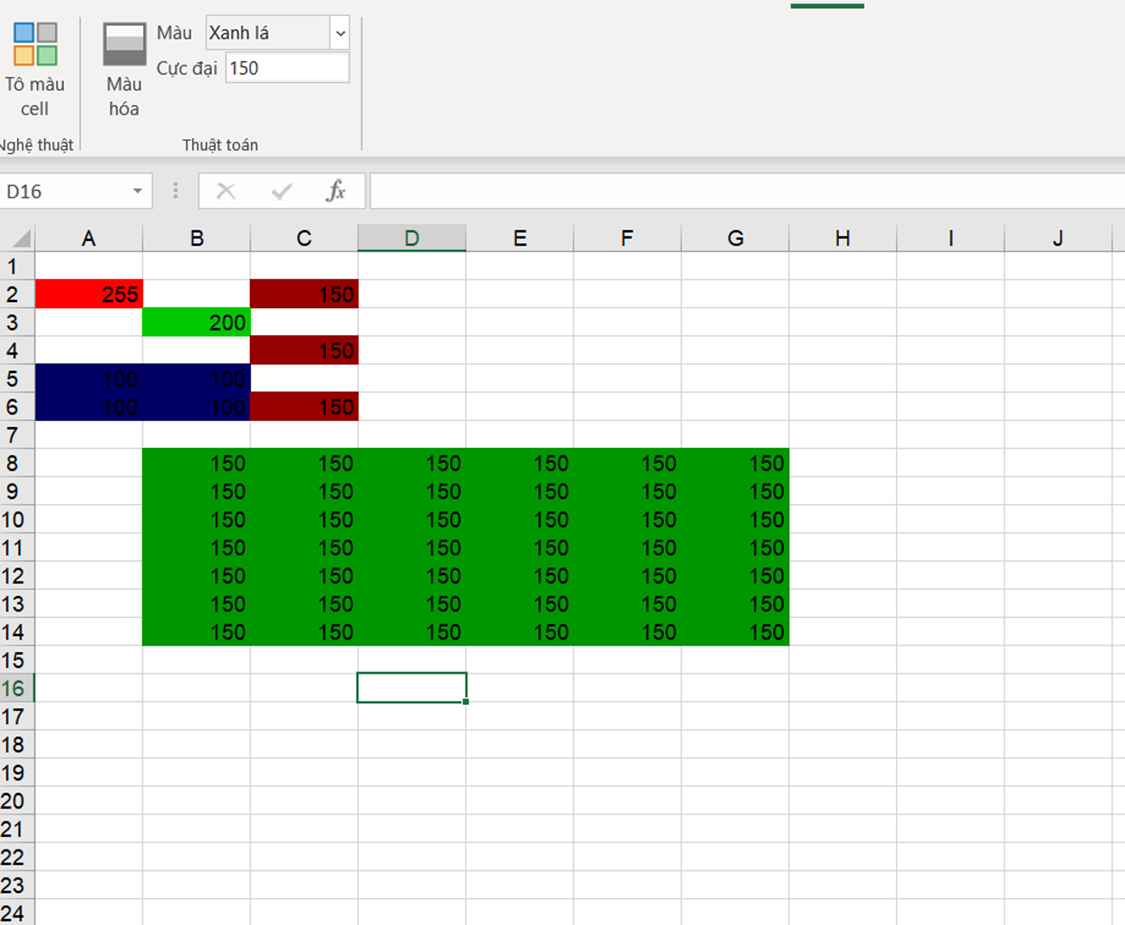
case "AppointmentColor3": currentRange.Interior.Color = Color.FromArgb(0, saturationInt, 0); break; //blue

default: break;

}

currentRange.Value = saturationInt; // cell value = saturation value

* Example:



* If invalid, the error messages will be displayed:

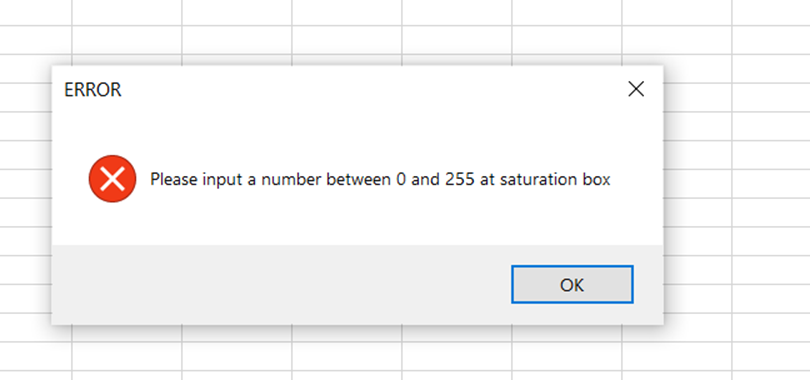
//saturation is not in range 0 to 255

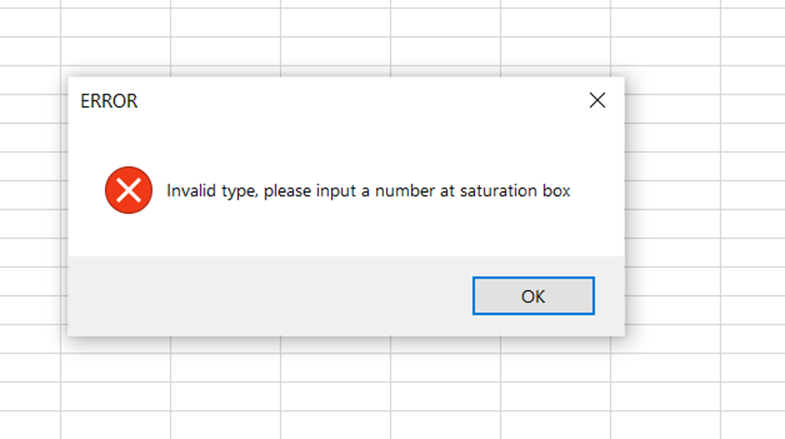
MessageBox.Show("Please input a number between 0 and 255 at saturation box", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

//saturation is not a valid number

MessageBox.Show("Invalid type, please input a number at saturation box", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

* Example:





## QRCode generator function

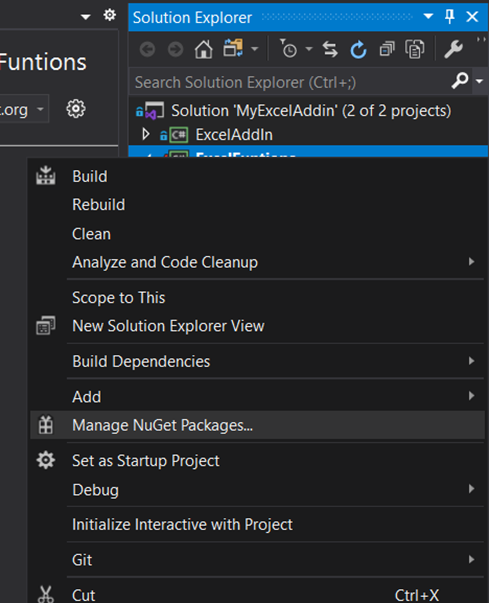
### About

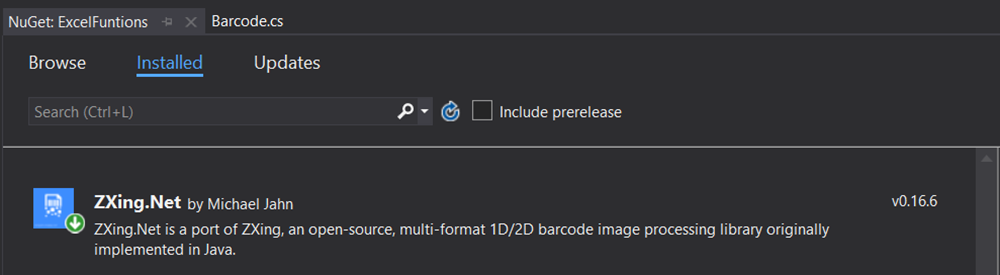
* This feature belongs to ExcelFunctions project.
* Using ZXing library (<https://github.com/zxing/zxing>) (instead of Google API) to generate QRCode image by input text.

### Detail

* ZXing library can be installed on Visual Studio as NuGet package:

On Solution Explorer, right click on one project, choose “Manage NuGet Packages…”, browse and install “ZXing.NET”:





* In Barcode class, an excel function QRCodeZ (Z stands for ZXing), with a string Text parameter, was created:

//Generate QRCode using ZXing library

[ExcelDna.Integration.ExcelFunction(Description = "QRCode generator by ZXing lib")]

public static object QRCodeZ(

[ExcelDna.Integration.ExcelArgument(Description = "Text to be transformed to QRCode. Example: \"hello\"")]

string Text)

* The Text string is read by ZXing library. Next step, it is transformed into QRCode image, saved to a file, then loaded to the current excel file.
* Update (12/10/2020): not necessary to save the QRCode image to a file, just save it to clipboard and paste to the worksheet.
* Update (23/10/2020): Add a parameter int option to the function QRCode, with option 1 to use ZXing library to generate QRCode image, option 2 to use Google API. The ZXing option uses Path.GetTempFileName() to save the QRCode image to local storage.

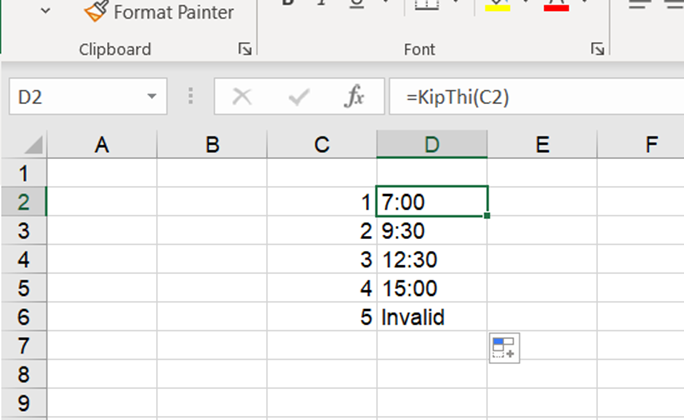
## KipThi function

### About

* This feature belongs to ExcelFunctions project.
* Function Kipthi(STT) to show the starting time of the exam:
* KipThi(1) -> 07:00
* KipThi(2) -> 09:30
* KipThi(3) -> 12:30
* KipThi(4) -> 15:00

### Detail

* In Hust class, a function KipThi was added with an int Kip parameter. The function will return the corresponding starting time of the exam:



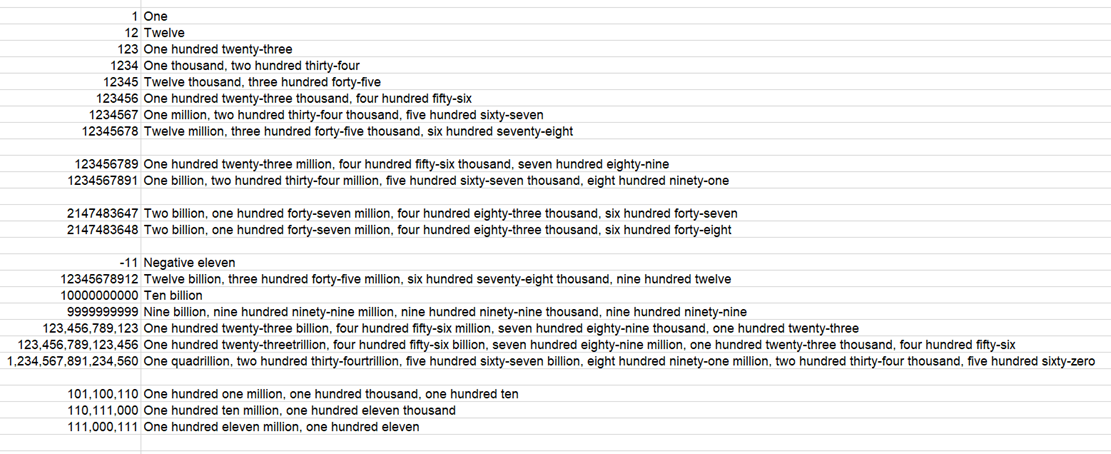
## InText function

### About

* This feature belongs to ExcelFunctions project.
* Function InText will transform number into text in English.

### Detail

* In FinanceTools class, a function InText was created with parameter int number, which will return the text value of the number in English.
* Another function ReadMoneyInText to divide the original number into smaller 3-digit numbers, and a function Read3DigitNumber to tranform these 3-digit numbers into text.
* TODO: lỗi tiếng việt, vd: 1001 đọc là “một nghìn không trăm linh một”. Done (check thêm ở trước có số không).
* Example:



## StringCellFormatter and StringFontFormatter functions

### About

* These features belong to ExcelFunction project.
* Function StringCellFormatter(text, CellBackgroundColor) will return the input text and change the background color of the cell.
* Function StringFontFormatter(text, FontName, FontSize, FontColor) will return the input text and change font type, font size, font color (type #3456FC for RGB).

### Detail

* The functions cannot access to the required properties. Therefore, these functions were canceled.

## Cortana voice recognition

### About

* This feature belongs to ExcelAddIn project.
* There is a button on ribbon bar, when user clicks that button, Cortana will return the text from voice recognition to a cell.

### Detail

* Added a button on ribbon bar.
* Used System.Speech.Recognition
* Recognized but not so accurate.
* Show progress of listening on the cell.
* This add-in is for research only because there is a better solution for voice recognition.

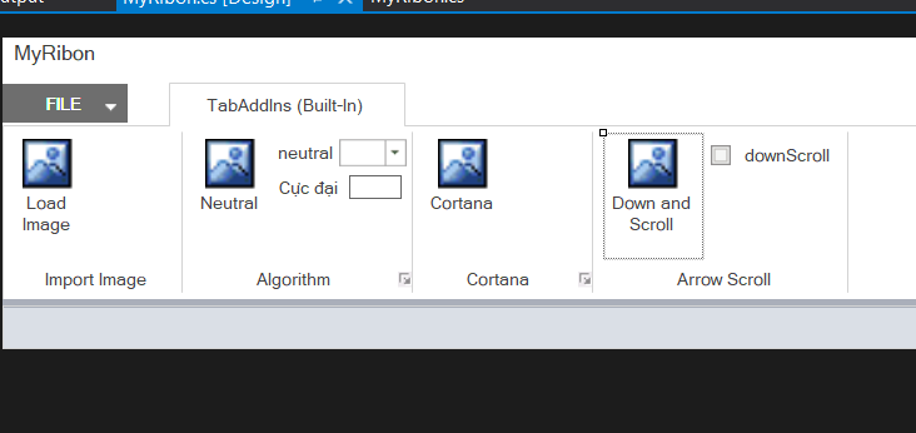
## ArrowMouseScroll

### About

* This feature belongs to ExcelAddIn project.
* When user presses down arrow, the workbook also scrolls.

### Detail

* A new group Arrow Scroll and a button Down and Scroll were created:



* In this button’s click event, this snippet of code was added:

SendKeys.Send("{Down}{SCROLLLOCK}{Down}");

* It will send the keys in the brackets, including 3 keys: Down (down arrow), SCROLLLOCK (Scroll Lock), and another Down.
* The first Down will make the pointer down 1 cell, scroll lock and the second down will make the workbook scroll one row.

# HOW TO BUILD AND USE

* Clone the source code, open it in Visual Studio. In the Solution Explorer, there are 2 projects, ExcelAddIn and ExcelFunctions:

Graphical user interface, text, application

Description automatically generated

* Right click to one project and choose Build to build and run it:

A screenshot of a computer screen

Description automatically generated

* See the result after building the project, pay attention the path of the result file:

Text

Description automatically generated

Text

Description automatically generated

* To run ExcelAddIn project, open one Excel workbook, on the ribbon bar, choose File -> Options, the Excel Options window will appear, in this window, choose Add-ins -> Manage, choose COM Add-ins, and Go:

Graphical user interface, text, application, email

Description automatically generated

* Add the Dynamic Link Library (.dll file) to the COM Add-ins windows, the path of this file is the build result of the ExcelAddIn project:

Graphical user interface, text, application, email

Description automatically generated

* Click OK, on the current workbook, the new tab Add-ins will appear on the ribbon bar(if not, you may have to restart the Excel workbook):

Application, table

Description automatically generated

* To run ExcelFunctions project, on the Manage section of the Excel Options window, choose Excel Add-ins and Go:

Graphical user interface, text, application, email

Description automatically generated

* Click Browse button on the new window Add-ins to lead to the path of the build result file (ExcelFunction-AddIn64-packed.xll) of the ExcelFunctions project:

Graphical user interface, text, application

Description automatically generated

* Click OK. To test, choose one cell on the workbook, on the Formula Bar, input =QRCode(<option>,”<shape>”,”<string to QRCode>”), for example: =QRCode(1,"shape1","testString").

# CONCLUSION

The overall result of the projects is successful, though some small parts are not as expected as from the beginning.

These DLL can be used in real cases, and in multiple version of Excel on different Microsoft Windows version.

After this research, I have gained much more knowledge and skills in coding in C#, writing new functions and add-ins for Excel, language in Microsoft Windows management.

# REFERENCES

* MSc. Nguyen Duc Tien’s github MyExcelAddin project repository: <https://github.com/neittien0110/MyExcelAddin>
* This research repository: <https://github.com/phandinhtuong/MyExcelAddin>
* Microsoft Visual Studio website: <https://visualstudio.microsoft.com/>
* ZXing library: <https://www.nuget.org/packages/ZXing.Net>
* ResXManager: <https://marketplace.visualstudio.com/items?itemName=TomEnglert.ResXManager>
* Windows speech recognition: <https://docs.microsoft.com/en-us/windows/uwp/design/input/speech-recognition>