|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| Developing Report  Simply Handy, West Lothian College |
| |  |  |  | | --- | --- | --- | | Juan Alvarez | 5/1/19 | HND Software Development 2018/9 | |

Contents

[1 Introduction 2](#_Toc7620169)

[2 Implementation of the solution 2](#_Toc7620170)

[2.1 Software used 2](#_Toc7620171)

[2.2 Business Model code listing 2](#_Toc7620172)

[2.3 View Model code listing 4](#_Toc7620173)

[2.4 Use of unfamiliar libraries/constructs 7](#_Toc7620174)

[2.5 Provision for error handling 8](#_Toc7620175)

[2.6 Internal Documentation 8](#_Toc7620176)

[3 Testing the solution 8](#_Toc7620177)

[4 Supporting User Documentation 8](#_Toc7620178)

[5 Conclusion 9](#_Toc7620179)

[Bibliography 10](#_Toc7620180)

# 1 Introduction

This report is a continuation of the process to develop an application to store all players details that Simply Rugby club commissioned from OOP Solutions.

It is a continuation of the Planning Report (Alvarez, Planning Report Juan Alvarez 2018-9, 2019) already handed in at the Analysis and Design stage of the project. Most of the resources used in this project is already referenced in that document.

I will attempt to describe the steps followed to achieve the goal of producing the software and what was done differently from the original plan explained in the Planning Report.

I will as well cover the testing of the product and will provide the User Documentation that will be used for training Simply Rugby’s employees.

# 2 Implementation of the solution

The first thing needed to be able to create the Application is the software used on the project.

## 2.1 Software used

* Visual Studio 2017
* Microsoft Word 365
* Microsoft Visio
* Sql Database embedded in Visual Studio 2017 (DotNetKida, 2017)
* MySql database downloaded from the internet (Oracle, 2019)

## 2.2 Business Model code listing

The complete solution is provided in the folder called Simply Handy, using either Visual Studio or notepad you can view the whole of the code and implementation of the solution. (Alvarez, Simply Handy Application, 2019)

The original plan (Alvarez, Planning Report Juan Alvarez 2018-9, 2019) was to installed a MySql database in the local machine where the application was going to be installed and run the server and did plenty of research regarding why using this database against other databases (Academind, 2018) and then went on to research how to install it and connect it with the solution. (MySql, 2019)

But when it came to the practical part of connecting it to the solution there were lots of constrains that made me change course. The main one was time and personal circumstances.

So I researched other possibilities and found a very neat solution that was at my fingertips, and that was to use the Sql database embedded into Visual Studio 2017 (DotNetKida, 2017).

There were lots of resources that I read, viewed, researched and used to be able to do the connection between the classes/application and the database (IAmTimCorey, 2017) (Microsoft C. a., 2016)

When I first played about with the solution once I had created the LoginIn Page and the Admin Page I had issues storing the data, as the database for some reason was creating the database in one place and reading it from another. After exhaustive research I found out it was an issue of how Visual Studio works and found a solution, stop using debug mode, and use Release mode instead. (Steve, 2014)

The biggest part of this stage of the project was ensuring the data entered into the text boxes is the correct one that the client wants saved into the database. When looking for the solutions to the problems that the Application presented, I always tried to use the simplest and more direct way to solve them.

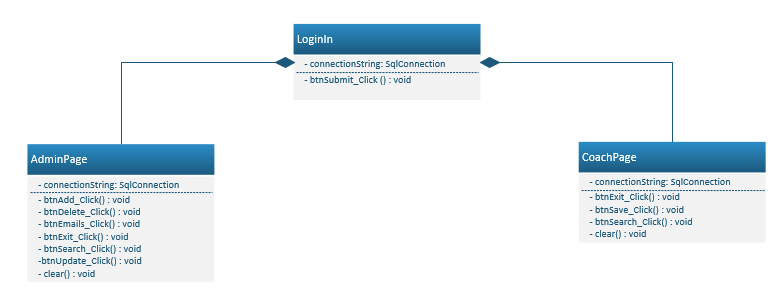
Some things I ensured the text boxes, date pickers and check boxes had, as data validation, are:

* Name field could not contain numbers (Ghazarian, 2014)
* SRU Number field could only contain numbers (Badirov, 2019)
* Ensuring the date picker is not left null (Agarwal, 2017)
* Ensuring the phone number are just numbers (Badirov, 2019) and that it can contain a maximum of 11 characters (Microsoft, TextBox.MaxLength Property, 2019), this last step was done using xaml
* Ensuring the email field must contain the characters “@” and “.” (Nickz2, 2015)
* Ensuring the Squad field is not left empty (Aɴᴀᴍ, 2018)
* How to check if the Parental Consent check box is checked or not (Shannon, 2009)
* Calculating if a player entered into the system is underage (Gallagher, 2016) (Hornby, 2010) and at the same time, when the player entered is underage been able to check if the Parental Consent is checked. (Shannon, 2009)
* In the Coach Page the fields Standard, Spin, Pop, Front, Rear, Side, Scrabble, Drop, Punt, Grabber and Goal have the same validation:  
  It is limited to just 1 character (Microsoft, TextBox.MaxLength Property, 2019)  
  The content can only be a number (Badirov, 2019)  
  And that the value has to be between 0 and 5 (Nair, 2010)

This probably was the bigger part of the project once I overcame the set up of the database and the connection of the database to the Simply Handy application.

Due to technical and time constrains and with the aim of simplifying the solution even more, eventually I ended up creating a different class diagram as I simplified it respect the original plan. (Alvarez, SimplyHandyClassDiagramV1.2, 2019)

In the end the class diagram looks like this:

  
Image 1

The LoginIn class is used to log into the system, I change in xaml the original page the class App uses to open the application, that is another thing I had to research. (CodAffection, How to Create Login Screen in Wpf with Sql DataBase, 2017)

AdminPage class provides all the functionality the user Admin requires:

* Searching for players
* Editing/Updating players’ details
* Adding new players to the database
* Deleting players
* Viewing a list of all the emails the database holds
* Exiting the application

Coach Page class provides all the functionality the user Coach requires:

* Searching for players
* Updating the Skills Development of existing players
* Exiting the application

## 2.3 View Model code listing

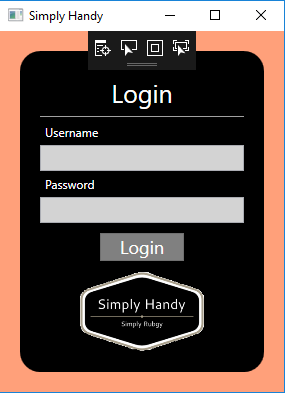
The first step in creating Simply Handy Application was to implement the User Interface using xaml code and the toolbox available in Visual Studio 2017. I learn a lot from a specific video regarding creating login page, not only it allowed me to follow a simple way to create a login page, but it as well gave me an opportunity to learn and understand much better xaml code, as we had not seen much of it in class. (CodAffection, How to Create Login Screen in Wpf with Sql DataBase, 2017)

That tutorial allowed me then to achieve several goals, one is learning something new and implementing it in this solution, improve the visuals of the application so it is more pleasing to the eye and implement neat solutions to some validation issues that I will cover latter on in the program.

The View Model was kept very faithfully to the originally planned on the section 2.2.1 Storyboard (Alvarez, Planning Report Juan Alvarez 2018-9, 2019) and the UI Visualisation logs already provided at the planning stage (Alvarez, UI Visualisation LoginScreen, 2019), (Alvarez, UI Visualisation AdminScreen, 2019) and (Alvarez, UI Visualisation CoachScreen, 2019).

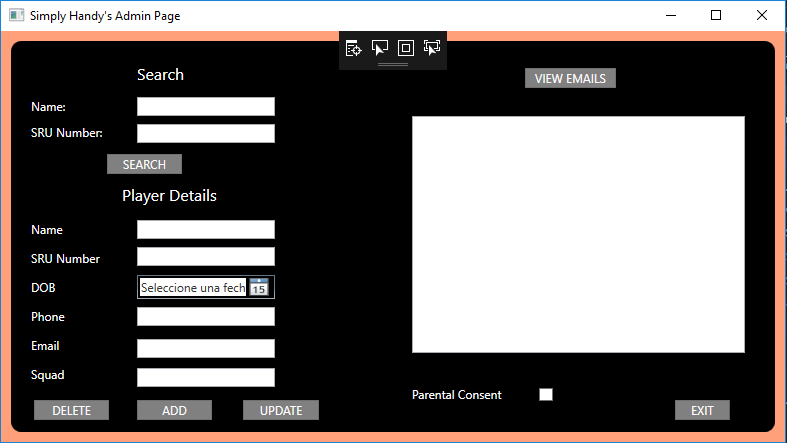
The thing I did update is the colours used on the pages, as well I used xaml to ensure the application opens centred in the screen so it is more comfortable and natural to use since I noted while I was developing the application that every time the application was opened it was opened in random positions around the screen making it uncomfortable to follow and made your positioning in front of the screen difficult and unnatural. (Petrotta, 2009)

The Login page ended looking like:

  
Image 2

This look was achieved thanks to the tutorial found online that taught me not only to create a Login page, but to use xaml to edit the look of the application (CodAffection, How to Create Login Screen in Wpf with Sql DataBase, 2017) and it includes the logo where both, the application’s name and the client’s name is clearly seen.

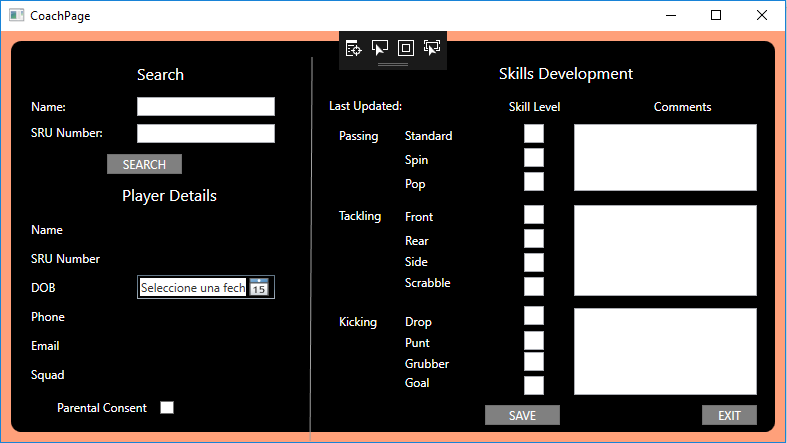
Once that was done it was simple to follow with the AdminPage:

  
Image 3

In the AdminPage the only difference with the original design is:

* The change of the DOB field from a text box to a date picker, much more fitting for its purpose.
* The change of the List box of emails for a text box, as Admin could not have selected all email and copy them into an email program when the need arise to send a group email. With it I had to add formatting to the textbox so the emails would appear in the format of a list. The IV Visualisation has been updated. (Alvarez, UI Visualisation AdminScreen V1-1, 2019)

And the CoachPage:

  
Image 4

The changes in the CoachPage are as well minimal:

* DOB has been changed from a textbox to a datepicker, much more fitting for its purpose
* The field Date has been changed to “Last Updated” as it will show the last date the user Coach updated this player’s records. If Coach never interacted with this particular player’s records this field will be blank.

This has been updated in the visualisation document. (Alvarez, UI Visualisation CoachScreen V1-1, 2019)

The basis of the Visual application are text boxes where the user can enter data and pressing buttons (both created with the toolbox provided with Visual Studio 2017) and then editing the xaml using the new skills learned in my research. (CodAffection, How to Create Login Screen in Wpf with Sql DataBase, 2017)

But the end result was faithful to the Story board originally provided with the only changes some realignment of some buttons, as I felt it was the simplest way to implement this solution, it would make it simple for the end users to learn and use the program, as it mirrors as much as possible the paper format that they were using until now. (Alvarez, Planning Report Juan Alvarez 2018-9, 2019)

The actual code for the View Model can be viewed in the Simply Handy folder using either Visual Studio 2017 or notepad. (Alvarez, Simply Handy Application, 2019)

## 2.4 Use of unfamiliar libraries/constructs

In the section 2.2 Business Model code listings I already show a lot of different things I had to research myself, like for example the use of xaml code to solve lots of issues that I found along the way (Microsoft, TextBox.MaxLength Property, 2019) as well as using xaml code to create a much more pleasing environment (CodAffection, How to Create Login Screen in Wpf with Sql DataBase, 2017).

But probably the biggest thing to research is the use of Sql database (DotNetKida, 2017) and how to connect it to the Application correctly (IAmTimCorey, 2017) (Microsoft C. a., 2016).

To make the connection work I had to use libraries like System.Data.SqlClient; (Microsoft, system.data.sqlclient, 2019) to ensure Simply Handy App would be able to connect and use the Sql database.

Another thing that I had to research is the use of SQL queries, but I was able to use knowledge acquired previously, so I just researched my personal notes regarding its workings.

In this project I used for the first time both check boxes and date pickers.

Another thing I learned is how to store the current date into the database without needed the user to input that data. (Zeeshan, 2014)

Finally, I had to learn how to retrieve data from a DataReader as part of the workings of making the database work. (Contributors, 2018)

## 2.5 Provision for error handling

In the section 2.2 Business Model code listings I already cover the validations provisions and research I took part on while developing the application.

Aside from it, in both the AdminPage and the CoachPage I used try catch to ensure that any possible error is flagged up and in the AdminPage I use a handling exception error code to give instructions to the user, in the case of the Admin trying to save an SRU Number that already exists on the application. (ILYAS, 2015)

## 2.6 Internal Documentation

In the folder Simply Handy (Alvarez, Simply Handy Application, 2019) the application can be viewed with Visual Studio 2017 and/or notepad to clearly see that all code is commented and explained.

# 3 Testing the solution

A lot of work went into the testing of the application, the resulting documentation is divided in 2 important group of testing documentation:

1 The Test Plan (Alvarez, TestPlanSimplyHandy, 2019) where all the testing process is explained and detailed.

2 The Test Logs (Alvarez, NavigationTestLogSimplyHandy, 2019), (Alvarez, FunctionalAcceptanceTestLogSimplyHandy, 2019), (Alvarez, EventValidationTestLogSimplyHandy-LoginIn, 2019), (Alvarez, EventValidationTestLogSimplyHandy-AdminPage, 2019), (Alvarez, EventValidationTestLogSimplyHandy-CoachPage, 2019), (Alvarez, DataValidationTestLogSimplyHandy-LoginIn, 2019), (Alvarez, DataValidationTestLogSimplyHandy-AdminPage, 2019) and (Alvarez, DataValidationTestLogSimplyHandy-CoachPage, 2019)

The bases of everything is in the Test Plan, where it exposes how the testing was done, the elements tested and the result of those tests.

# 4 Supporting User Documentation

The employees of Simply Rugby must be trained in the use of Simply Rugby. For that purpose, a user guide was produced to help conduct the training and for the employees to keep for future reference. (Alvarez, SimplyHandyUserGuide, 2019)

In that document the whole workings of the system is explained with screen shots of the different screens the users have access to.

# 5 Conclusion

Through the whole project I worked with a goal in mind, making it as simple as possible for the end user.

We are dealing with users that may not even had any computer experience in the past, since Simply Rugby uses a paper based system currently and don’t even have a computer at all, as I had to provide recommendations of what computer to use in the SRS handed in at the Planning stage (Alvarez, SRS Simply Handy Juan Alvarez V1-2, 2019).

So giving the most similar thing possible to a paper sheet was my main aim when developing this project.

With more time it is possible to work on the coding and improve it, but I do believe that as a first iteration, the proposed solution and prototype is the best possible considering the users that are going to be using Simply Handy day to day.

As time progresses, I expect to receive some requests for updating items on the system, changing some of the options or even adding features. By then the users will be more proficient in using computerized systems and the application, so we can then move forward to adding some features with extra complexity.

# Bibliography

6D, V. (06 de 12 de 2016). *# 1 MS Project 2016 ● Basics In 15 Minutes ● Easy*. Obtenido de youtube.com: https://www.youtube.com/watch?v=goX6N9RjGUs&feature=youtu.be

Academind. (25 de 07 de 2018). *SQL vs NoSQL or MySQL vs MongoDB*. Obtenido de youtube.com: https://www.youtube.com/watch?v=ZS\_kXvOeQ5Y

Agarwal, I. a. (11 de 09 de 2017). *check-if-datepicker-value-is-null*. Obtenido de stackoverflow: https://stackoverflow.com/questions/35957822/check-if-datepicker-value-is-null

Alvarez, J. (28 de 02 de 2019). ActivityDiagramAddingPlayer. Livingston, West Lothian, Scotland.

Alvarez, J. (01 de 03 de 2019). ActivityDiagramEditPlayer. Livingston, West Lothian, Scotland.

Alvarez, J. (01 de 03 de 2019). ActivityDiagramEditSkillsDevelopment. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). ActivityDiagramLoginIn. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). ActivityDiagramSearchingPlayer. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). ActivityDiagramViewEmails. Livingston, West Lothian, Scotland.

Alvarez, J. (02 de 03 de 2019). CRC\_AdminPage. Livingston, West Lothian, Scotland.

Alvarez, J. (02 de 03 de 2019). CRC\_CoachPage. Livingston, West Lothian, Scotland.

Alvarez, J. (02 de 03 de 2019). CRC\_LoginIn. Livingston, West Lothian, Scotland.

Alvarez, J. (02 de 03 de 2019). CRC\_SqlQueries. Livingston, West Lothian, Scotland.

Alvarez, J. (01 de 05 de 2019). DataValidationTestLogSimplyHandy-AdminPage. Livingston, West Lothian, Scotland.

Alvarez, J. (01 de 05 de 2019). DataValidationTestLogSimplyHandy-CoachPage. Livingston, West Lothian, Scotland.

Alvarez, J. (01 de 05 de 2019). DataValidationTestLogSimplyHandy-LoginIn. Livignston, West Lothian, Scotland.

Alvarez, J. (30 de 04 de 2019). EventValidationTestLogSimplyHandy-AdminPage. Livingston, West Lothian, Scotland.

Alvarez, J. (30 de 04 de 2019). EventValidationTestLogSimplyHandy-CoachPage. Livingston, West Lothian, Scotland.

Alvarez, J. (30 de 04 de 2019). EventValidationTestLogSimplyHandy-LoginIn. Livingston, West Lothian, Scotland.

Alvarez, J. (30 de 04 de 2019). FunctionalAcceptanceTestLogSimplyHandy. Livingston, West Lothian, Scotland.

Alvarez, J. (30 de 04 de 2019). NavigationTestLogSimplyHandy. Livingston, West Lothian, Scotland.

Alvarez, J. (2019). Planning Report Juan Alvarez 2018-9. Livingston, West Lothian, Scotland.

Alvarez, J. (04 de 03 de 2019). SequenceAddingPlayer. Livingston, West Lothian, Scotland.

Alvarez, J. (02 de 03 de 2019). SequenceDiagramLoginIn. Livingston, West Lothian, Scotland.

Alvarez, J. (2019). Simply Handy Application. Livingstone, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). SimplyHandyClassDiagramV1. Livingston, West Lothian, Scotland.

Alvarez, J. (02 de 03 de 2019). SimplyHandyClassDiagramV1.1. Livingston, West Lothian, Scotland.

Alvarez, J. (01 de 05 de 2019). SimplyHandyClassDiagramV1.2. Livingston, West Lothian, Scotland.

Alvarez, J. (30 de 04 de 2019). SimplyHandyUserGuide. Livingston, West Lothian, Scotland.

Alvarez, J. (09 de 01 de 2019). SimplyRugby-Juan-AlvarezV1.0. Livingston, West Lothian, Scotland.

Alvarez, J. (04 de 03 de 2019). SRS Simply Handy Juan Alvarez V1-2. Livingston, West Lothian, Scotland.

Alvarez, J. (10 de 01 de 2019). SRS Symply Handy Juan Alvarez. Livingston, West Lothian, Scotland.

Alvarez, J. (30 de 04 de 2019). TestPlanSimplyHandy. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). UI Visualisation AdminScreen. Livingston, West Lothian, Scotland.

Alvarez, J. (2019). UI Visualisation AdminScreen. Livingston, West Lothian, Scotland.

Alvarez, J. (01 de 05 de 2019). UI Visualisation AdminScreen V1-1. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). UI Visualisation CoachScreen. Livingston, West Lothian, Scotland.

Alvarez, J. (01 de 05 de 2019). UI Visualisation CoachScreen V1-1. Livingston, West Lothian, Scotland.

Alvarez, J. (07 de 03 de 2019). UI Visualisation LoginScreen. Livingston, West Lothian, Scotland.

Alvarez, J. (2019). UI Visualisation LoginScreen. Livingston, West Lothian, Scotland.

Alvarez, J. (08 de 01 de 2019). Use Case Diagram Juan Alvarez V1. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). Use Case Diagram Simply Handy V1-1. Livingston, West Lothian, Scotland.

Alvarez, J. (06 de 01 de 2019). User Stories Juan Alvarez V1. Livingstone, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). User Stories Juan Alvarez V1-1. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). WrittenUseCaseAddPlayers. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). WrittenUseCaseEditPlayers. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). WrittenUseCaseEditSkillsDevelopment. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). WrittenUseCaseLogingIn. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). WrittenUseCaseSearchandReviewSkillsDevelopment. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). WrittenUseCaseSearchPlayers. Livingston, West Lothian, Scotland.

Alvarez, J. (28 de 02 de 2019). WrittenUseCaseViewEmails. Livingston, West Lothian, Scotland.

Aɴᴀᴍ, F. (15 de 03 de 2018). *best-way-to-check-whether-a-textbox-is-empty-or-not*. Obtenido de stackoverflow: https://stackoverflow.com/questions/34298857/best-way-to-check-whether-a-textbox-is-empty-or-not/34299121

Badirov, T. (17 de 03 de 2019). *validating-whether-a-textbox-contains-only-numbers*. Obtenido de stackoverflow: https://stackoverflow.com/questions/15399323/validating-whether-a-textbox-contains-only-numbers/15399373

Bospear. (29 de 04 de 2017). *C# Excel Tutorial - #2 - Write and Save Excel Files*. Obtenido de youtube.com: https://www.youtube.com/watch?v=rSIzj2wbVrQ

CodAffection. (04 de 05 de 2017). *How to Create Login Screen in Wpf with Sql DataBase*. Obtenido de https://www.youtube.com/: https://www.youtube.com/watch?v=72366X1-heg

CodAffection. (04 de 05 de 2017). *How to Create Login Screen in Wpf with Sql DataBase*. Obtenido de www.youtube.com: https://www.youtube.com/watch?v=72366X1-heg

Contributors. (29 de 10 de 2018). *retrieving-data-using-a-datareader*. Obtenido de docs.microsoft.com: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/retrieving-data-using-a-datareader

Curry, C. (04 de 08 de 2017). *MongoDB in 18 Minutes - Intro to MongoDB*. Obtenido de youtube.com: https://www.youtube.com/watch?v=bKjH8WhSu\_E

D, V. (03 de 02 de 2013). *Create Login Window in C# step by step*. Obtenido de youtube.com: https://www.youtube.com/watch?v=tcmmCcMs8yU

DotNetKida. (3 de June de 2017). *Create Database Using Visual Studio 2017*. Obtenido de Youtube.com: https://www.youtube.com/watch?v=3lmHYKbuFIA

Gallagher, C. (28 de 07 de 2016). *check-if-age-is-18-datepicker-wpf*. Obtenido de stackoverflow: https://stackoverflow.com/questions/38634108/check-if-age-is-18-datepicker-wpf

Ghazarian, A. (18 de 01 de 2014). *how-could-i-check-if-a-textbox-has-text*. Obtenido de stackoverflow.com: https://stackoverflow.com/questions/21200279/how-could-i-check-if-a-textbox-has-text

Hornby, J. (10 de 02 de 2010). *how-can-i-calculate-age-by-datetimepicker*. Obtenido de stackoverflow: https://stackoverflow.com/questions/2237587/how-can-i-calculate-age-by-datetimepicker

IAmTimCorey. (28 de 02 de 2017). *How to connect C# to SQL (the easy way)*. Obtenido de Youtube.com: https://www.youtube.com/watch?v=Et2khGnrIqc

ILYAS, S. (20 de 07 de 2015). *how-can-i-catch-uniquekey-violation-exceptions-with-ef6-and-sql-server*. Obtenido de stackoverflow.com: https://stackoverflow.com/questions/31515776/how-can-i-catch-uniquekey-violation-exceptions-with-ef6-and-sql-server

Microsoft. (2019). *sql-server/*. Obtenido de microsoft.com: https://www.microsoft.com/en-US/sql-server/sql-server-2017?&OCID=AID739533\_SEM\_LBFN2I7a&MarinID=sLBFN2I7a\_324864632573\_sql%20server\_e\_c\_\_64776423093\_kwd-14998960\_

Microsoft. (2019). *system.data.sqlclient*. Obtenido de https://docs.microsoft.com: https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient?view=netframework-4.8

Microsoft. (2019). *TextBox.MaxLength Property*. Obtenido de https://docs.microsoft.com: https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.textbox.maxlength?view=netframework-4.8

Microsoft, C. a. (11 de 04 de 2016). *add-new-connections?view=vs-2019*. Obtenido de microsoft.com: https://docs.microsoft.com/en-us/visualstudio/data-tools/add-new-connections?view=vs-2019

Mongodb. (2019). *Mongodb*. Obtenido de mongodb.com: https://www.mongodb.com/

MySql. (2019). *MySQL*. Obtenido de mysql.com: https://www.mysql.com/

Nair, S. (02 de 11 de 2010). *Textbox-should-accept-numbers-from-0-to-366*. Obtenido de codeproject: https://www.codeproject.com/Questions/123858/Textbox-should-accept-numbers-from-0-to-366

Nickz2. (23 de 04 de 2015). *how-can-i-make-a-textbox-only-accept-a-valid-email*. Obtenido de stackoverflow: https://stackoverflow.com/questions/29820568/how-can-i-make-a-textbox-only-accept-a-valid-email

Oracle. (2019). *MySql.com*. Obtenido de mysql.com: https://dev.mysql.com/downloads/

Oracle. (2019). *technologies*. Obtenido de oracle.com: https://www.oracle.com/database/technologies/

Paysale. (2018). *PayScale*. Obtenido de payscale.com: https://www.payscale.com/research/UK/Certification=Project\_Management\_Professional\_(PMP)/Hourly\_Rate

PayScale. (2018). *PayScale*. Obtenido de payscale.com: https://www.payscale.com/research/UK/Job=Software\_Engineer/Salary

PayScale. (2018). *PayScale*. Obtenido de payscale.com: https://www.payscale.com/research/UK/Job=Software\_Tester/Salary

PayScale. (2018). *PayScale*. Obtenido de payscale.com: https://www.payscale.com/research/UK/Job=Personal\_Assistant/Salary

PayScale. (2018). *PayScale*. Obtenido de payscale.com: https://www.payscale.com/research/UK/Job=Software\_Developer/Salary

Petrotta, M. (1 de June de 2009). *how-to-center-a-wpf-app-on-screen*. Obtenido de stackoverflow: https://stackoverflow.com/questions/935599/how-to-center-a-wpf-app-on-screen

Shannon, B. C. (15 de 06 de 2009). *how-to-check-whether-a-checkbox-is-checked*. Obtenido de social.msdn.microsoft.com: https://social.msdn.microsoft.com/Forums/en-US/778db998-d92e-494d-83b1-5be66e0ebe5c/how-to-check-whether-a-checkbox-is-checked?forum=csharpgeneral

Simply Rugby. (01 de 12 de 2018). ProjectBrief. Livingston, West Lothian, Scotland.

Simply Rugby. (12 de 2018). Simply Rugby Paper Forms. Livingston, West Lothian, Scotland.

Steve. (06 de 05 de 2014). *data-not-saving-permanently-to-sql-table*. Obtenido de stackoverflow: https://stackoverflow.com/questions/23496393/data-not-saving-permanently-to-sql-table

tutorialspoint. (2018). *csharp\_binary\_files.htm*. Obtenido de tutorialspoint.com: https://www.tutorialspoint.com/csharp/csharp\_binary\_files.htm

Zeeshan, A. A. (16 de 02 de 2014). *how-do-i-get-the-current-system-date-and-time-in-c-sharp-and-insert-it-into-sql*. Obtenido de stackoverflow: https://stackoverflow.com/questions/21813711/how-do-i-get-the-current-system-date-and-time-in-c-sharp-and-insert-it-into-sql