**Database To Excel App**

**Documentation**

**(Windows Forms)**

**Table Of Contents**

1. **Introduction…………………………………………… 2**
   1. **Why Only SQLite Databases?.…………………………. 2**
   2. **Important Notes……………………………………………. 2**
   3. **Prerequisites…………………………………………..…… 2**
2. **Project Set-up……………………………………….... 3 - 5**
   1. **Creating A New Project…………………………………… 3 - 4**
   2. **Opening The Existing Project…………………………… 4 - 5**
      1. **Open With Visual Studio…………………………. 5**
      2. **Open With .exe…………………………………….. 5**
3. **Application………………………………………….…. 6 - 8**
   1. **How To Use…………………………………………………. 6 - 7**
   2. **Error Notes…………………………………………………. 8**
   3. **Custom Features……………………………..……………. 8**
4. **References…………………………………………….. 9 - 11**
   1. **General……………………………………………...………. 9**
      1. **View SQLite Database……………………………. 9**
      2. **Prerequisites………………………………………. 9**
      3. **Utilities……………………………………………… 9**
   2. **C# Coding………………………………………………..…. 9 - 10**
      1. **Excel……………………………………………..….. 9**
      2. **SQLite…………………………………………....….. 9**
      3. **Databases…………………………………………… 10**
      4. **General………………………………………………. 10**
   3. **C# Windows Forms…………………………………..……. 11**
      1. **Sections……………………………………………... 11**
      2. **Components/Tools……………………………….... 11**

**1. Introduction**

The Database To Excel App is a Windows Forms created project with the sole purpose of converting SQLite databases (.db, .mdf) into a Microsoft Excel Sheet file (.xls)

This mini project came about because of the constraints of using TIBCO Spotfire. As a brief summary, TIBCO Spotfire enables users to create impactful visualisations by providing a data source. However, database file extensions (Eg. .db and .mdf) are not supported.

Hence, the Database To Excel App aims to solve this problem by converting these database files into the supported file type in TIBCO Spotfire, specifically Microsoft Excel Sheet (.xls).

**1.1 Why Only SQLite Databases?**

When doing this project, the database files used were SQLite databases from prior Open House Databases. Henceforth, the usage will cater to only for SQLite databases instead of normal SQL Databases. Fortunately, due to the nature of the codes produced in this project, it is possible to adjust it to accept any SQLClient database type as the application reads the structure of the databases and only requires the correct references when handling other database types.

**1.2 Important Notes**

The project was coded with Visual Studio 2015. It was coded with the Windows Forms application provided by the platform. The programming language used is C#.

The .NET Framework version for Windows Forms used is Ver. 4.6. It is very important to take note of the .NET Framework version used because of the external Dynamic Link Library (DLL) files used.

When trying to recreate this project, it is important to do extensive research to identify the required DLL for the project as it can greatly enhance the production of the project. In this SQLiteToExcel\_FileConverter Application, I have used the SQLite DLL which enables users to be able to connect an SQLite Database, and manipulate the data.

To see the list of references and download links, please visit the last page of this documentation. All external references used are recorded in the last page of this documentation report.

**1.3 Prerequisites**

You require Visual Studios and Excel installed to create this Windows Forms Application.

Only Excel is required to use this Windows Forms Application.

**2. Project Set-up**

The Database To Excel App is a Windows Forms created project with the sole purpose of converting SQLite databases (.db, .mdf) into a Microsoft Excel Sheet file (.xls)

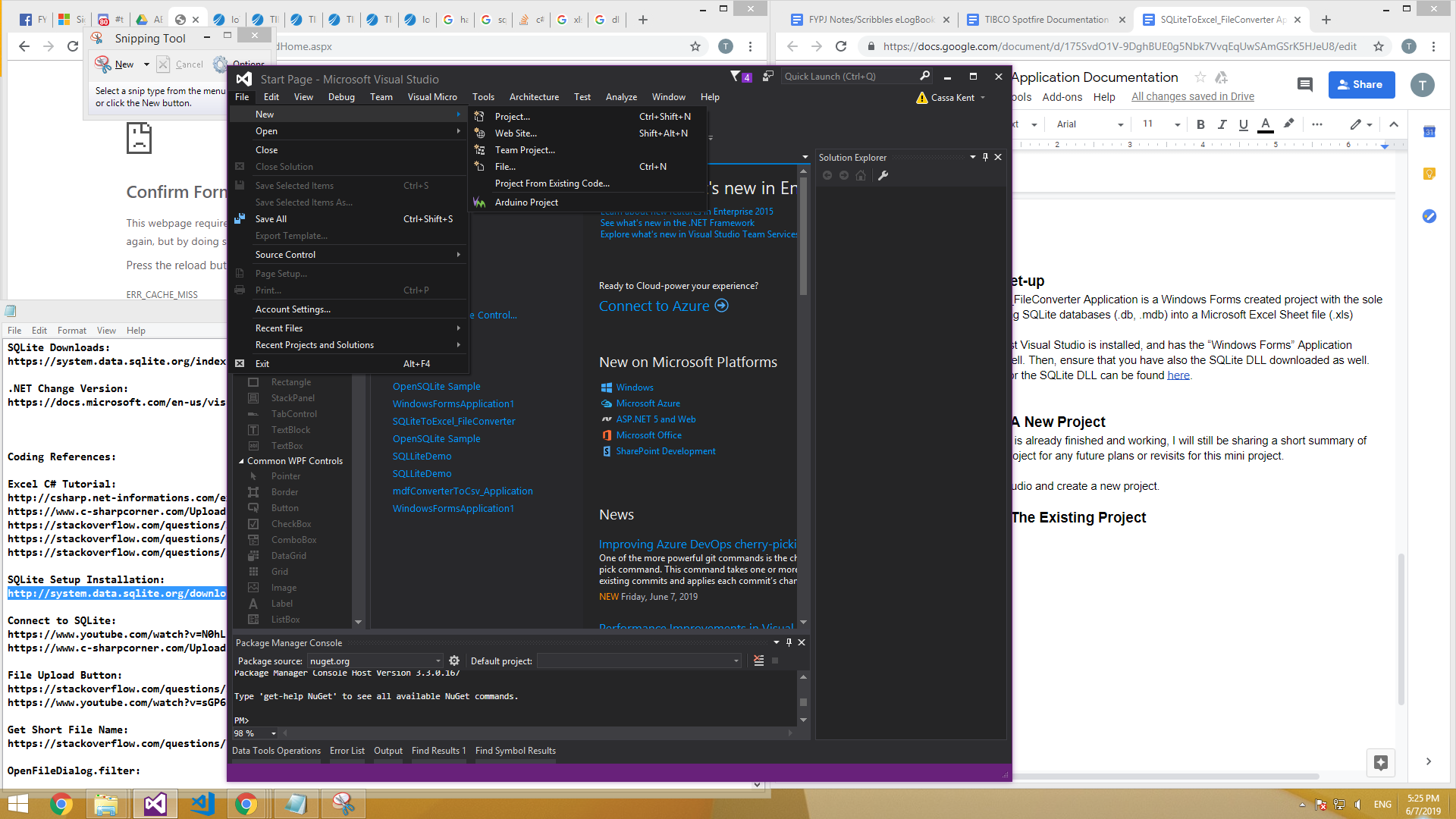
Ensure that the latest Visual Studio is installed and has the “Windows Forms” Application installed inside as well. Then, ensure that you have the correct version of SQLite DLL downloaded as well. The SQLite download link for .NET ver 4.6 can be found [here](http://system.data.sqlite.org/downloads/1.0.102.0/sqlite-netFx45-setup-bundle-x86-2012-1.0.102.0.exe).

**2.1 Creating A New Project**

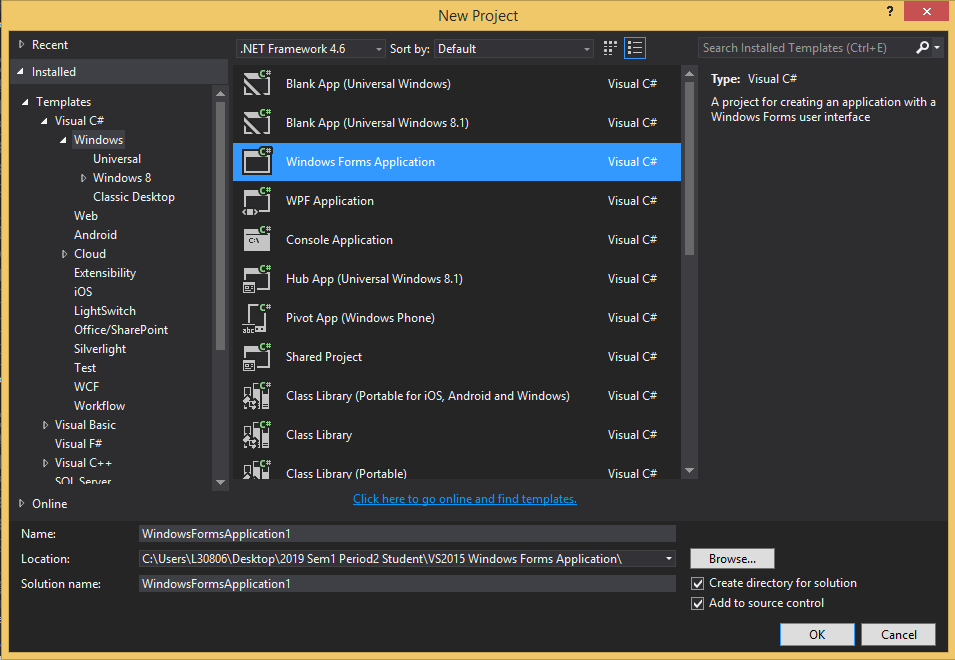
Although the project is already finished and working, I will still be sharing a short summary of how I created this project for any future plans or revisits for this mini project. With this said, I will at least expect you to already learn some basic fundamentals of using the Windows Forms Application. If not, you may refer to the reference page for useful links I used during this project.

Open your Visual Studio and create a new project.

File > New > Project.

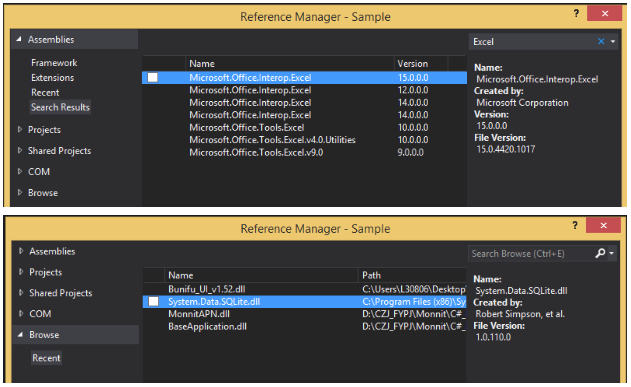
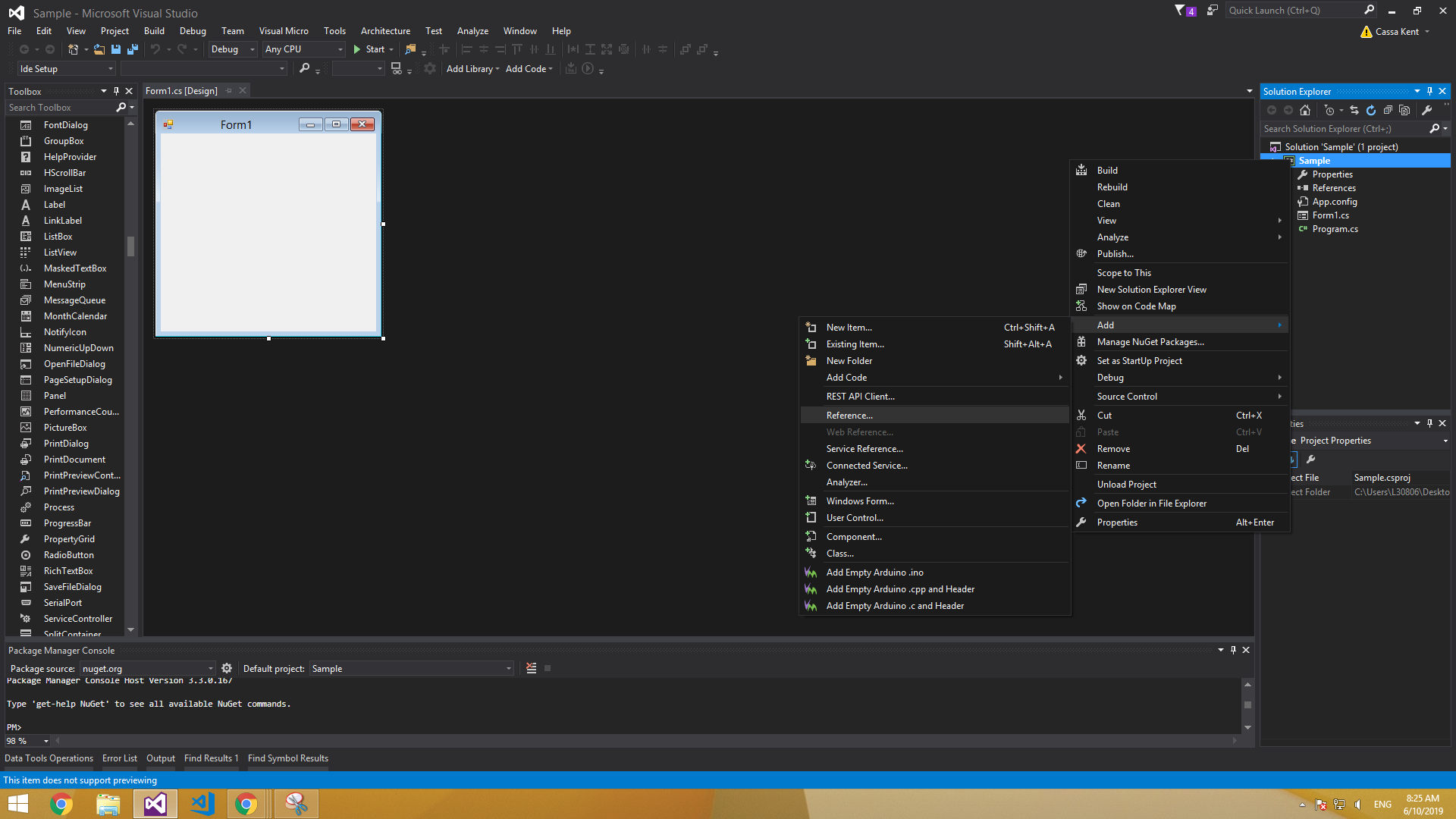


Then select on Windows Forms Application and create the project.



Once created, a pre-created template will appear. You may create or use this template. Next is to add the following references into the project.

To add a reference, right click the project then click on Add > Reference. Then add these two references; Microsoft.Office.Interop.Excel (Latest ver.), and System.Data.SQLite.dll (Ver. according to .NET Framework ver.).



The Microsoft.Office.Interop.Excel reference can be found inside “Assemblies”, and it will come together with Visual Studios. However, the System.Data.SQLite.dll will have to be downloaded separately. As previously explained, the download link for the SQLite for .NET ver 4.6 can be found [here](http://system.data.sqlite.org/downloads/1.0.102.0/sqlite-netFx45-setup-bundle-x86-2012-1.0.102.0.exe).

With the set-up complete, it is best to explore on your own to recreate the entire project. When in doubt, you may refer to the reference page of this documentation, which can be found on the last page.

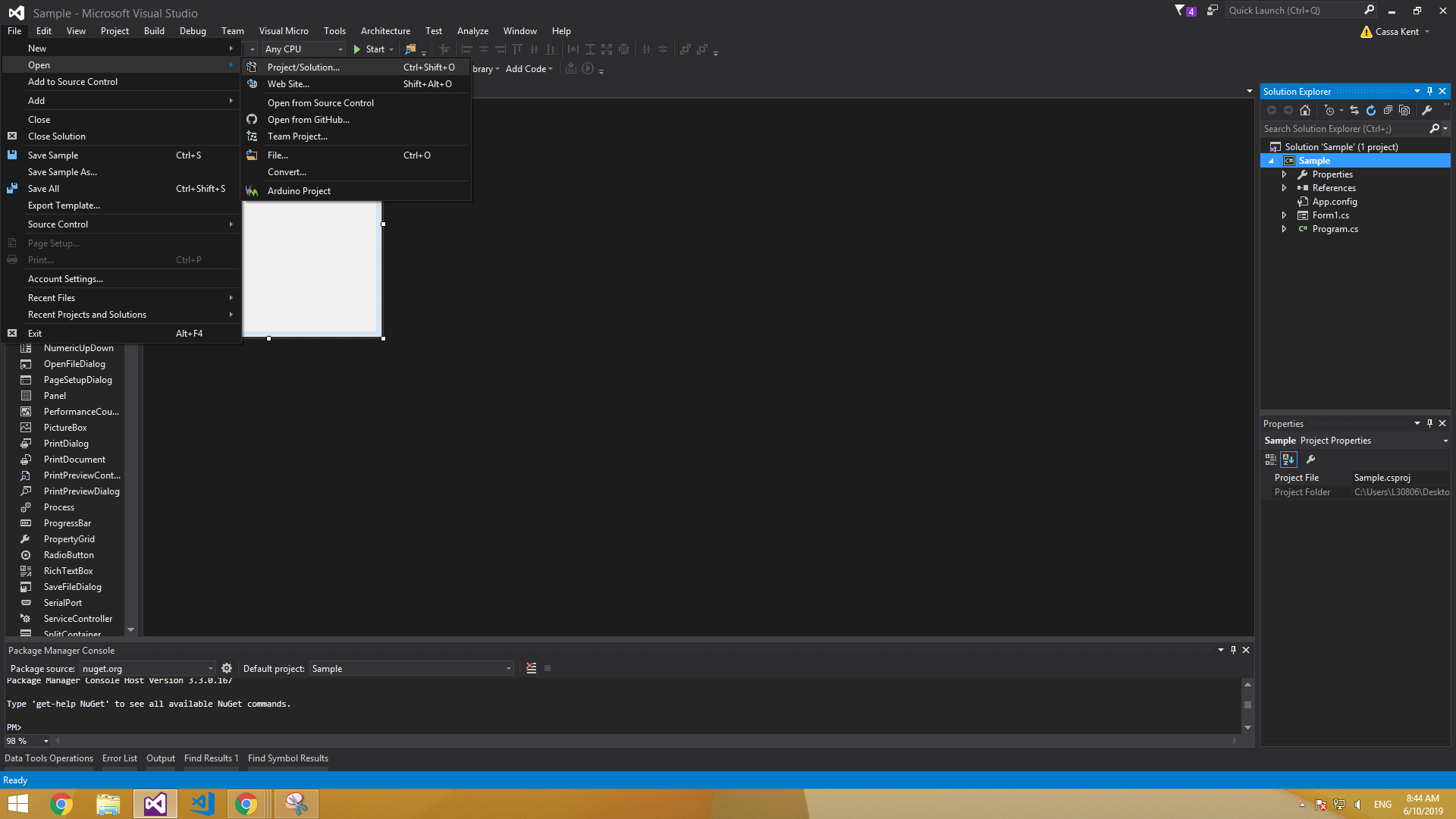
**2.2 Opening The Existing Project**

After downloading the zip folder of the project, extract the main directory folder onto your desired space. Recommended to extract it into desktop for easy access and prevent any errors.

The Database To Excel App can be opened by two different ways. It can be opened by either using Visual Studios, or it can be opened by running the .exe file located inside the extracted main directory folder.

**2.2.1 Open With Visual Studio**

To open the project with Visual Studio, first start up your Visual Studio normally. Then proceed to open the project by selecting File > Open > Project/Solution.

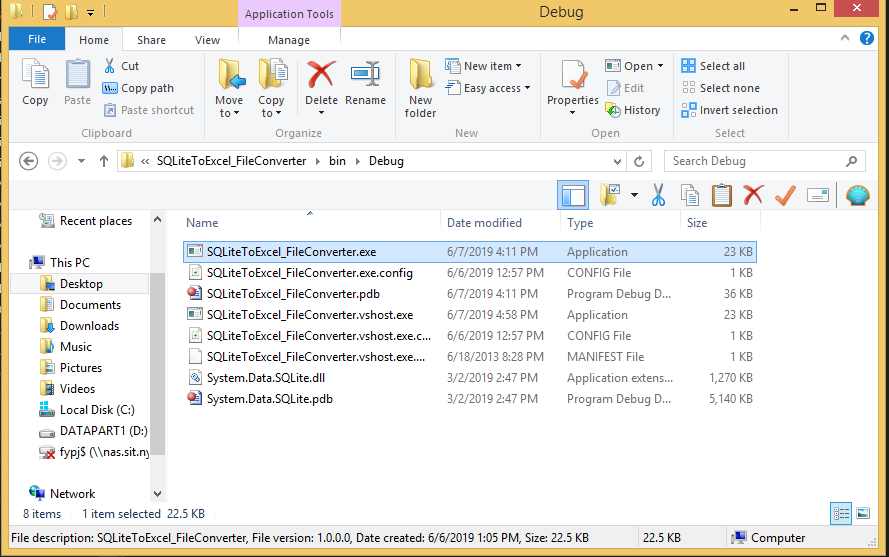


When clicked, you will now have to browse through your entire desktop to locate the solution file for the project. Inside the main project directory, locate the solution file and open with it. The solution file will have the .sin extension, and is the only file with that extension in the entire project folder.

**2.2.2 Open With .exe**

Alternatively, you may also run the .exe file to open the project. Inside the main project directory, navigate your way to the SQLiteToExcel\_FileConverter.exe file.

SQLiteToExcel\_FileConverter > bin > Debug > SQLiteToExcel\_FileConverter.exe

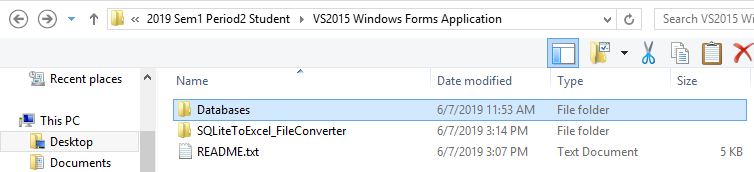


When the SQLiteToExcel\_FileConverter.exe file is clicked, it will immediately start the project.

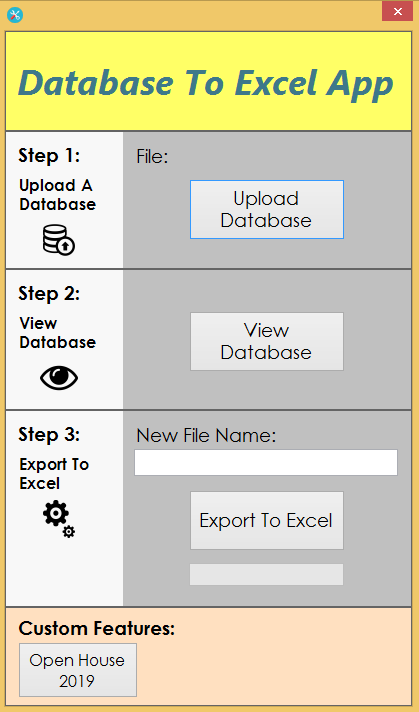
**3. Application**

**3.1 How To Use**

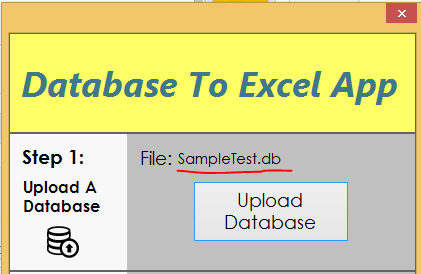
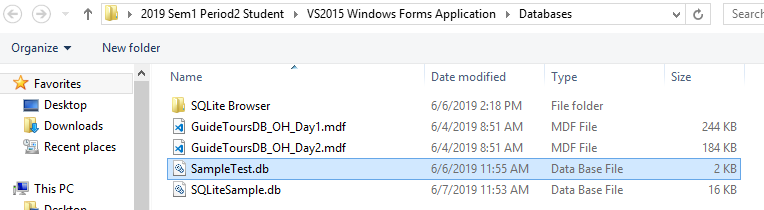
Inside the download file, I have included a “Databases” folder that consists of sample SQLite databases for testing. You may also use your own SQLite databases. SQLite databases accepted for this application must have the the .mdf or .db extension.



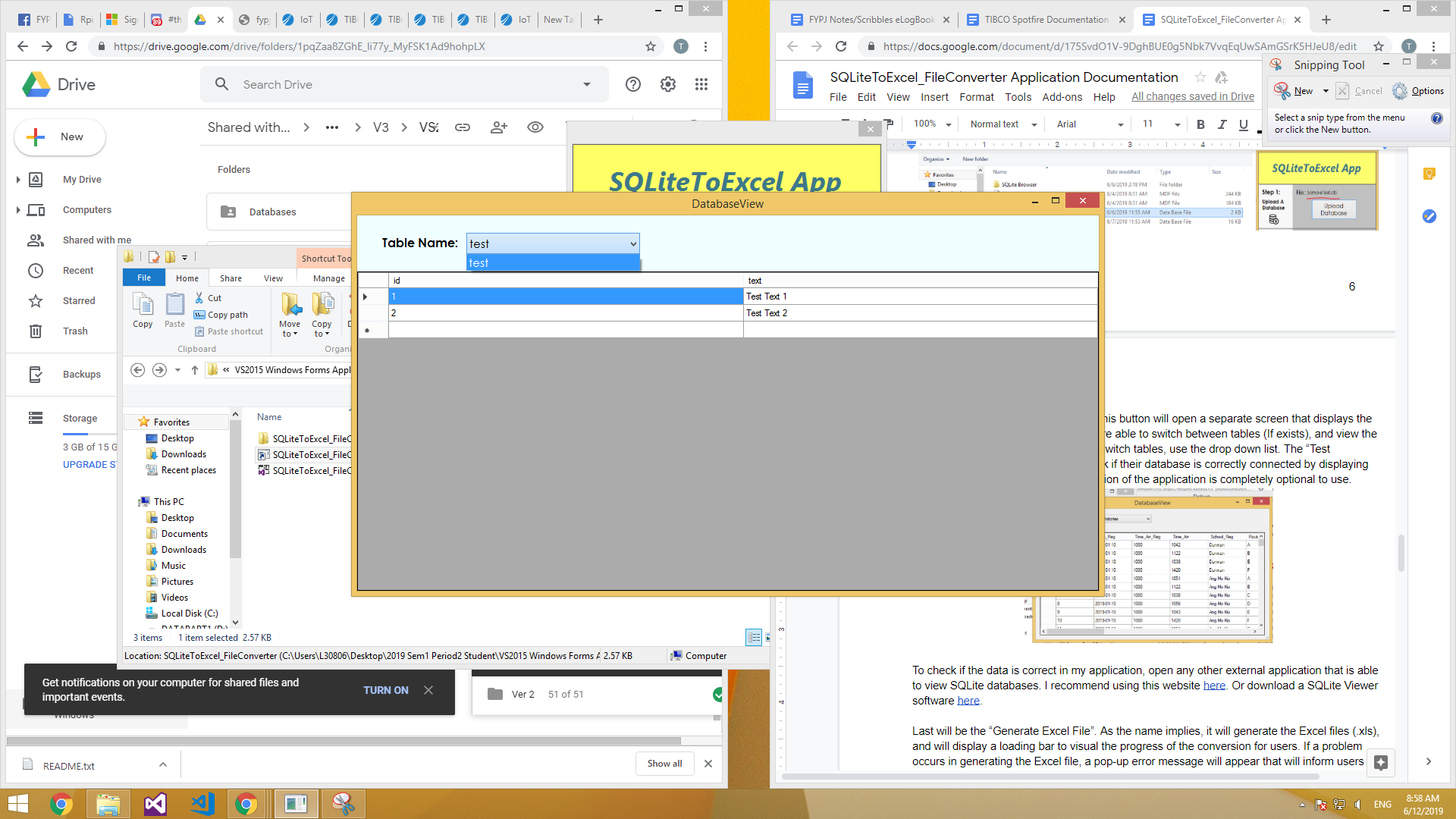
After opening the application in step 2.2, a screen will appear. This screen is the main view of the Database To Excel App.



To start using the application, click on “Upload Database” and select the SQLite database to be used for the conversion. You will have to browse through your entire desktop and locate the SQLite database. After providing a SQLite database file, the file name will be displayed below to ensure users that the file is connected to the application.

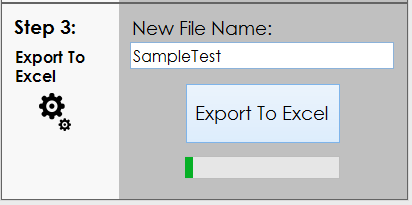


Next is to click on “Test Connection”. This button will open a separate screen that displays the entire SQLite database, where users are able to switch between tables (If exists), and view the data of each column in each table. To switch tables, use the drop down list. The “Test Connection” is meant for users to check if their database is correctly connected by displaying the data in the correct format. This portion of the application is completely optional to use.

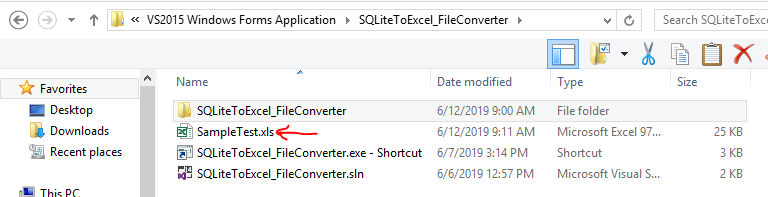


To check if the data is correct in my application, open any other external application that is able to view SQLite databases. I recommend using this website [here](https://inloop.github.io/sqlite-viewer/). Or download a SQLite Viewer software [here](https://sqlitebrowser.org/).

Last will be the “Export To Excel”. This section of the application will generate an Excel file (.xls), and will display a loading bar to visual the progress of the conversion for users. If a problem occurs in generating the Excel file, a pop-up error message will appear that will inform users of the error that the application encountered. Users are also able to add a new name for their Excel file.



When the application completes its Excel Sheet generation, it will be saved beside the project folder. It can be found inside the same directory as the project folder and solution file.

****

**3.2 Error Notes**

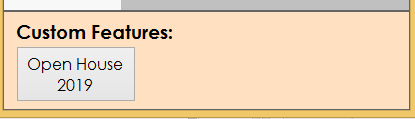
If the application is used in the wrong order, a corresponding error message will appear. Such as “Please connect a database first!”.

This application requires Microsoft Excel to be installed first, if not a pop-up error will appear to instruct user to install Microsoft Excel first.

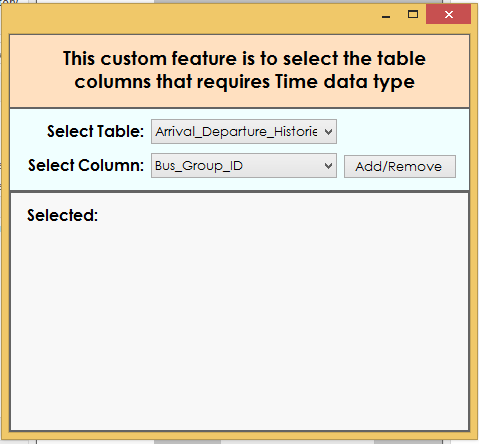
If a user double generates the Excel file, it will prompt the user if they would like to replace the file as it has already been generated, and exist. The user may click any of these options. If the user clicks “No” or “Cancel”, an error message will appear followed by the notification that the action was cancelled.

**3.3 Custom Features**

At the bottom of the application will be a custom feature section. This section is meant for future developments where specific requirements must be met.



In this instance, the “Open House 2019” custom feature allows users to select the table column to convert them from a “String” to “DateTime” data type.



**4 References**

**4.1 General**

**4.1.1 View SQLite Database**

Browser: <https://inloop.github.io/sqlite-viewer/>

Program: <https://sqlitebrowser.org/>

**4.1.2 Prerequisites**

SQLite DLL Downloads: <https://system.data.sqlite.org/index.html/doc/trunk/www/downloads.wiki>

.NET Change Version: <https://docs.microsoft.com/en-us/visualstudio/ide/how-to-target-a-version-of-the-dotnet-framework?view=vs-2019>

**4.1.3 Utilities**

Comment Shortcut: <https://community.dynamics.com/gp/b/gpdynland/archive/2017/06/10/changing-visual-studio-keyboard-shortcut-for-comment-and-uncomment>

Free Icons: <http://www.iconarchive.com/>

Database Formats: <https://fileinfo.com/filetypes/database>

**4.2 C# Coding**

**4.2.1 Excel**

<http://csharp.net-informations.com/excel/csharp-create-excel.htm>

<https://www.c-sharpcorner.com/UploadFile/bd6c67/how-to-create-excel-file-using-C-Sharp/>

<https://stackoverflow.com/questions/21847420/excel-worksheet-get-item>

<https://stackoverflow.com/questions/16705651/how-to-create-a-new-worksheet-in-excel-file-c>

<https://stackoverflow.com/questions/35368734/excel-change-cell-value-of-active-worksheet>

**4.2.2 SQLite**

Set-up: <http://system.data.sqlite.org/downloads/1.0.102.0/sqlite-netFx45-setup-bundle-x86-2012-1.0.102.0.exe>

Connect:

<https://www.youtube.com/watch?v=N0hL5sGkUSA>

<https://www.c-sharpcorner.com/UploadFile/5d065a/how-to-use-and-connect-sqlite-in-a-window-application/>

**4.2.3 Databases**

SQLite schema/table/data manipulation:

<https://stackoverflow.com/questions/3005095/can-i-get-name-of-all-tables-of-sql-server-database-in-c-sharp-application>

<https://stackoverflow.com/questions/31961970/how-can-i-count-the-number-of-tables-in-c-sharp>

<https://stackoverflow.com/questions/27699781/specified-cast-is-not-valid>

Database, Data manipulation:

<https://forums.asp.net/t/1962034.aspx?Get+entire+column+from+database+C+>

<https://stackoverflow.com/questions/32513018/retrieve-column-name-in-c-sharp>

<https://stackoverflow.com/questions/9660981/get-a-datatable-columns-datatype>

<https://stackoverflow.com/questions/31961970/how-can-i-count-the-number-of-tables-in-c-sharp>

<https://docs.microsoft.com/en-us/dotnet/api/system.data.datatable.tablename?view=netframework-4.8>

<https://stackoverflow.com/questions/3005095/can-i-get-name-of-all-tables-of-sql-server-database-in-c-sharp-application>

Display Database Table in C#:

<https://www.youtube.com/watch?v=C9s0H6yeFLQ>

**4.2.4 General**

Get Short File Name: <https://stackoverflow.com/questions/7793158/obtaining-only-the-filename-when-using-openfiledialog-property-filename>

Array List: <https://www.geeksforgeeks.org/c-sharp-get-or-set-the-number-of-elements-that-the-arraylist-can-contain/> <https://stackoverflow.com/questions/14260271/c-sharp-sort-arraylist-strings-alphabetical-and-on-length>

<https://www.tutorialsteacher.com/csharp/csharp-arraylist>

Dictionary: <https://www.tutorialsteacher.com/csharp/csharp-dictionary>

Foreach Loop: <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/foreach-in>

Substring C#:

<https://www.c-sharpcorner.com/UploadFile/mahesh/substring-in-C-Sharp/>

<https://www.c-sharpcorner.com/UploadFile/mahesh/add-remove-replace-strings-in-C-Sharp/>

DateTime Formats C#:

<https://www.c-sharpcorner.com/blogs/date-and-time-format-in-c-sharp-programming1>

String To Time C#:

<https://stackoverflow.com/questions/18919530/convert-string-to-time>

**4.3 C# Windows Forms**

**4.3.1 Sections**

File Upload Button:

<https://stackoverflow.com/questions/12968138/how-to-upload-a-file-in-window-forms>

<https://www.youtube.com/watch?v=sGP6u68k2hc>

File Upload OpenFileDialog.filter: <https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.filedialog.filter?view=netframework-4.8> <https://docs.microsoft.com/en-us/dotnet/api/microsoft.win32.filedialog.filter?view=netframework-4.8>

File Upload Multiple Filter: <https://stackoverflow.com/questions/4710881/multiple-file-extensions-in-openfiledialog>

Display SQLite Data Button:

<https://www.c-sharpcorner.com/UploadFile/5d065a/how-to-use-and-connect-sqlite-in-a-window-application/>

<https://www.youtube.com/watch?v=N0hL5sGkUSA>

Display SQLite Data Form:

<https://www.youtube.com/watch?v=GLCtoHagrtE>

Connect Forms:

<https://www.c-sharpcorner.com/UploadFile/5d065a/how-to-open-a-second-form-using-first-form-in-window-form/>

Passing Data Between Forms:

<https://www.c-sharpcorner.com/UploadFile/834980/how-to-pass-data-from-one-form-to-other-form-in-windows-form/>

Form Onload Event:

<https://www.youtube.com/watch?v=wNpuFsuto3w>

**4.3.1 Components/Tools**

ComboBox/DropDownList:

<https://www.youtube.com/watch?v=WOqho3SJtoc>

<https://stackoverflow.com/questions/7423911/how-to-populate-c-sharp-windows-forms-combobox>

Windows Forms ProgressBar:

<https://bytes.com/topic/c-sharp/answers/903766-how-do-i-reset-progress-bar-after-completes>

<https://www.youtube.com/watch?time_continue=234&v=b5pFzMo2wR0>

<https://stackoverflow.com/questions/12126889/how-to-use-winforms-progress-bar>

Windows Forms Drop Down List

<https://stackoverflow.com/questions/17519278/clearing-a-drop-down-list-in-c-sharp>

Windows Forms Scrollbar:

<https://stackoverflow.com/questions/730376/how-do-you-add-a-scrollbar-to-a-panel-control-with-many-controls-in-windows-form>

Windows Forms UI Design:

<https://www.youtube.com/watch?v=K9Ps66GoD-k>