

Final Exam

Term: Winter 2024

Professor: Gursharan Singh Tatla

Course: PROG32356 – .NET Tech. using C#

Email: gursharan.tatla@sheridancollege.ca

Please Be Advised That:

1. Exam must be completed as an individual effort. Do not collaborate with anyone or share it with any individual, party or entity.
2. Do not share this exam with anyone or any 3rd party without the written consent of the professor.
3. ZIP the project and upload it to SLATE by due date/time, mentioned in SLATE.
4. All online submissions will be done via SLATE (Email submissions will NOT be accepted).
5. Corrupt/incorrect submissions will be graded as zero.
6. This is a 3-hour exam which will be attempted over the period of 3-days. Therefore:
 - a. No late submissions will be accepted as it is an exam.
 - b. No 3-day grace period is allowed.
 - c. No extensions will be provided to students with accommodations.
7. Make sure your laptops are in good working condition. If something does not work, or any application or system crashes, you will be responsible to fix it and submit your work on time. No excuses will be accepted during the exam.
8. Make sure your Visual Studio is in good working condition. No excuses will be entertained regarding issues with Visual Studio.
9. Please refer to the [Academic Integrity Policy](#).
10. Refer to the School of Applied Computing's Academic Procedures for Evaluations for more details regarding missed work: [Procedures for Evaluations](#).
11. If you take code-snippets from external sources such as StackOverflow, make sure to provide references to the source's webpage as comments in your code.

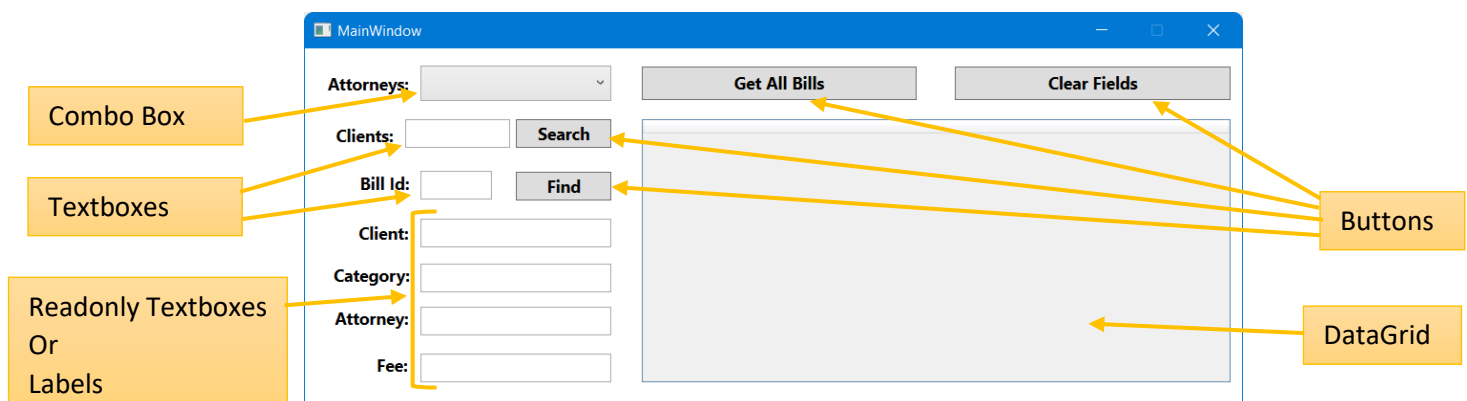
Copyright Disclaimer:

The materials provided in class and in SLATE are protected by copyright. They are intended for the personal, educational uses of students in this course and should not be shared externally or on websites such as Chegg, Course Hero or OneClass. Unauthorized distribution may result in copyright infringement and violation of Sheridan policies.

Gursharan Singh Tatla (Winter 2024)

Exam:

1. Make a **WPF App (.NET Framework)** in Visual Studio and name it as **FinalFirstnameLastname**.
2. This is the sample layout of the application. You can change the layout however you want.

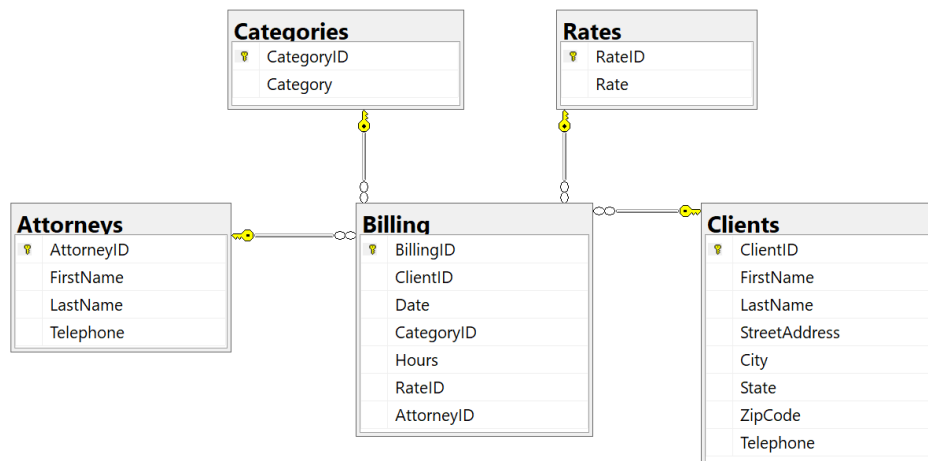


3. Download the **LawFirmDB.sql** file and use the SQL script it contains to create and populate the database.

- Download from SLATE → Content → Module 6 - Database Programming → Sample Databases.
- When creating a new database, it must be named **LawFirmDB**, otherwise, the script will not run properly.
- Use **LocalDB** as database source, unless you've got written permission from your professor to use a different data source such as SQL Server or MDF file.

Gursharan Singh Tatla (Winter 2024)

4. The following image shows the relationship between the tables:



5. In the WPF App, add **Entity Framework**.

6. Use **DB First** approach to generate the entities.

7. **Note:**

- *Implement the solution by only using Entity Framework and DB-First approach.*
- *Grade 0 will be given if any other approach or technique is used.*

Gursharan Singh Tatla (Winter 2024)

8. The window has:

- **ComboBox**: to display the list of attorneys.
- **Textbox**: to search for clients by name.
- Readonly **Textboxes** or **Labels**: to display the data for a single bill.
- **DataGrid**: to display the data from the Billings table.
- **Buttons**: three fetch data or clear the fields.

9. On the initial window load, populate the **Attorneys** **ComboBox** with their last names from the table **Attorneys**.

10. And populate the **DataGrid** with the data from the **Billing** table.

Gursharan Singh Tatla (Winter 2024)

Attorneys ComboBox:

1. On selecting an attorney's last name from the **Attorneys** **ComboBox**, fetch and display all the records from the **Billing** table in the **DataGrid** that correspond to the selected attorney.

Search Clients by Name:

1. User can type in the client's first name or last name and then click on the **Search** button.
2. Fetch and display the records from the **Clients** table in the **DataGrid**.
 - a. Partial matches should be sufficient to fetch the records.
 - b. Match the searched keyword with client's first name or last name.

Find Bill by Bill ID:

1. User can input the **Bill ID** in the *Bill Id TextBox* and then click on the **Find** button.
2. Fetch a record based on **Bill ID** and display the data in the respective readonly *TextBoxes* or *Labels*.
3. Concatenate client's and attorney's first names and last names, and then display.
4. Calculate the fee amount by multiplying hours with rate, and then display.
5. If a record is not found, display an appropriate message and clear the *TextBoxes* or *Labels*.

Gursharan Singh Tatla (Winter 2024)

Get All Bills Button:

1. This button fetches all the records from the **Billing** table and displays them in the *DataGrid*.
2. Optionally, to change the columns headings of the *DataGrid*, you can do something like this:

```
DataGridName.Columns[0].Header = "Attorney ID";  
DataGridName.Columns[1].Header = "First Name";  
DataGridName.Columns[2].Header = "Last Name";
```
3. Implement these changes after populating the *DataGrid*.

Clear Fields:

1. This button clears all the fields on the window.
2. The **Attorney ComboBox** must clear the displayed text but should not clear the list of attorney names.

Submission:

1. Once done, **ZIP the solution folder** and upload it to **Assignments** on SLATE.
 - a. Make sure to ZIP the whole folder, not just .SLN file.
 - b. If only .SLN file is zipped and submitted, Grade 0 will be given.
 - c. **Double-check your submission by downloading it and running it.**
2. You are to submit .ZIP and .TXT files, separately:
 - a. Upload the **.ZIP** file of your assignment to SLATE.
 - b. Upload the **.TXT** files to SLATE.
3. You must copy and paste all of the source code from the MainWindow.xaml.cs C# file into plain text file.
 - a. You can copy and paste the source code into Notepad.
 - b. You do not need to copy the XAML source code. Only copy/paste the .CS source code.
 - c. You do not need to copy the Entity (domain) classes code.
4. You don't have to format this code - it's used by TurnItIn (the originality checker in SLATE, which is a piece of software that checks your submission for plagiarism against other submissions in the college, in other colleges, from the web, and various other sources).
5. Submit the text file in addition to your assignment ZIP file.
 - a. DO NOT add it inside your zip/rar file – it must be a separate file.
 - b. It is used for TurnItIn (it won't examine the contents of zip/rar files).
6. **Note:**
 - a. If these submission instructions are not followed, Grade 0 will be granted.
 - b. If TXT files are not provided, Grade 0 will be granted.

Sample Output:

- Check out this video to see the sample output of this application:
<https://www.loom.com/share/43a46838b70a4987b91e305374034091>