**PROG8145**

**Software Development Techniques**

**Final Project**

**Olha Bahatiuk - 8001935**

**Pushpak - 8136590**

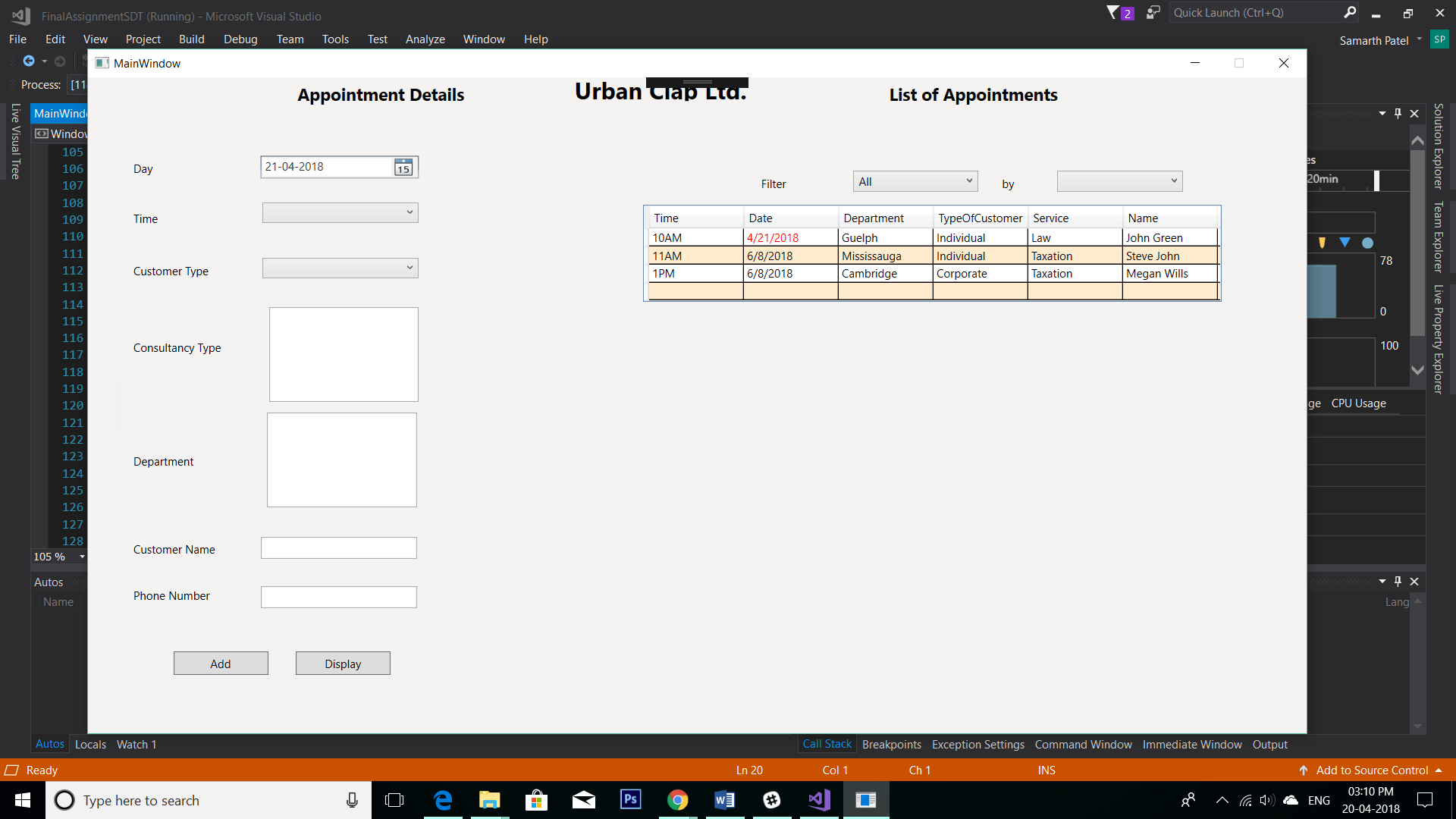
**Samarth Patel - 8146094**

**Igor Pustylnick**

**April 20,2018**

# **Overview**

We have designed an Appointment System for a Company called Urban Clap Ltd. The system schedules and manages the appointments for various services like Taxation, Lawyer and Investment. This system is used to add appointments, according to the customer type (i.e. corporate or individual), and what type of consultancy they want like lawyer, Taxation or Investment and then they can select the location they can visit nearby them. The appointment added can be seen in the list beside and also the appointments can be filtered by various fields.

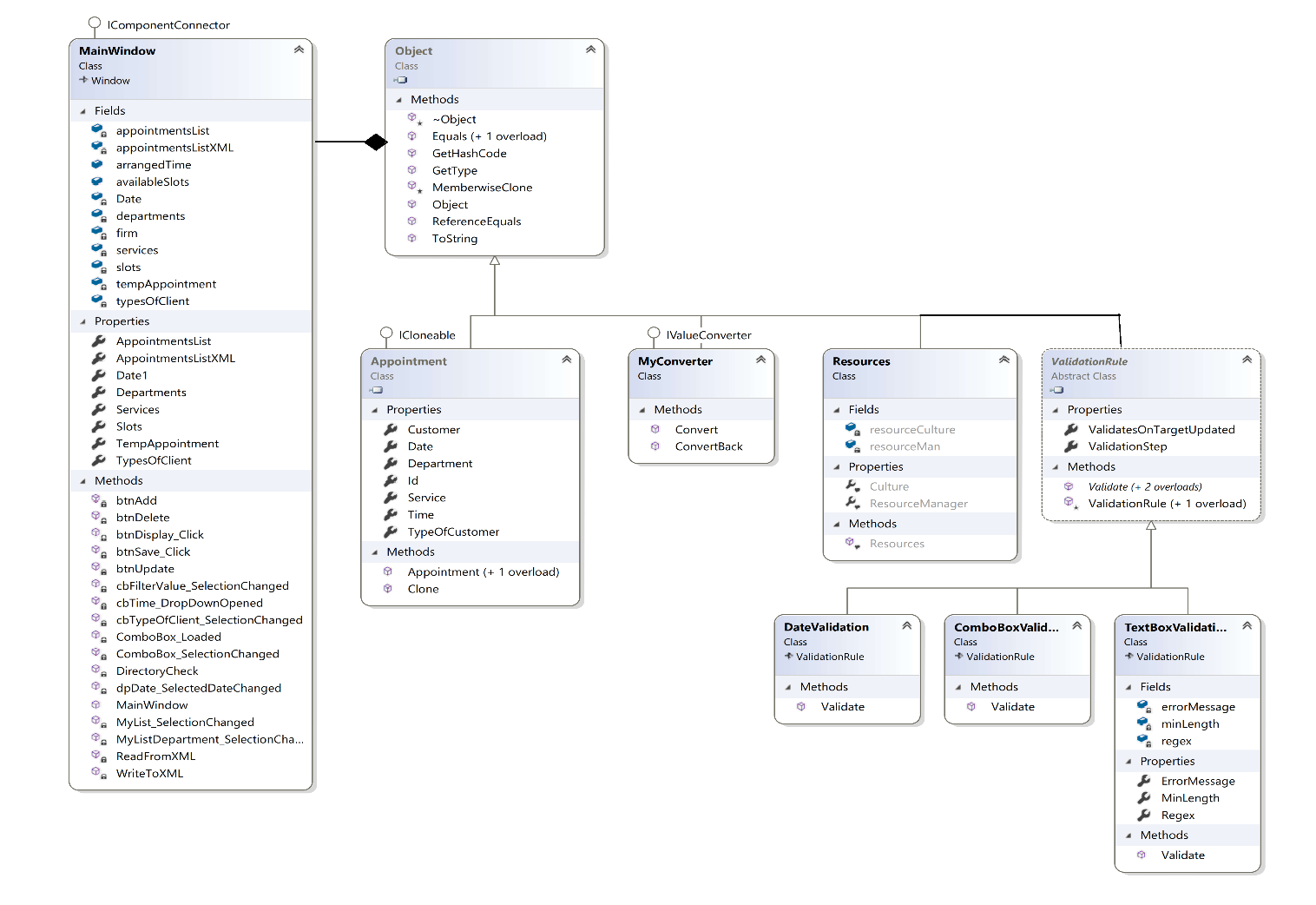


# **Feature of System**

This Appointment Scheduling System has features like:

* Allow the user to add an appointment
* User can check saved appointment in the list.
* User can filter the saved appointment to find a specific appointment.
* Appointment validation are used so that no two appointments are on same day at same time.
* All the fields have validation which prevents the user from entering wrong data.

# **Class Design**

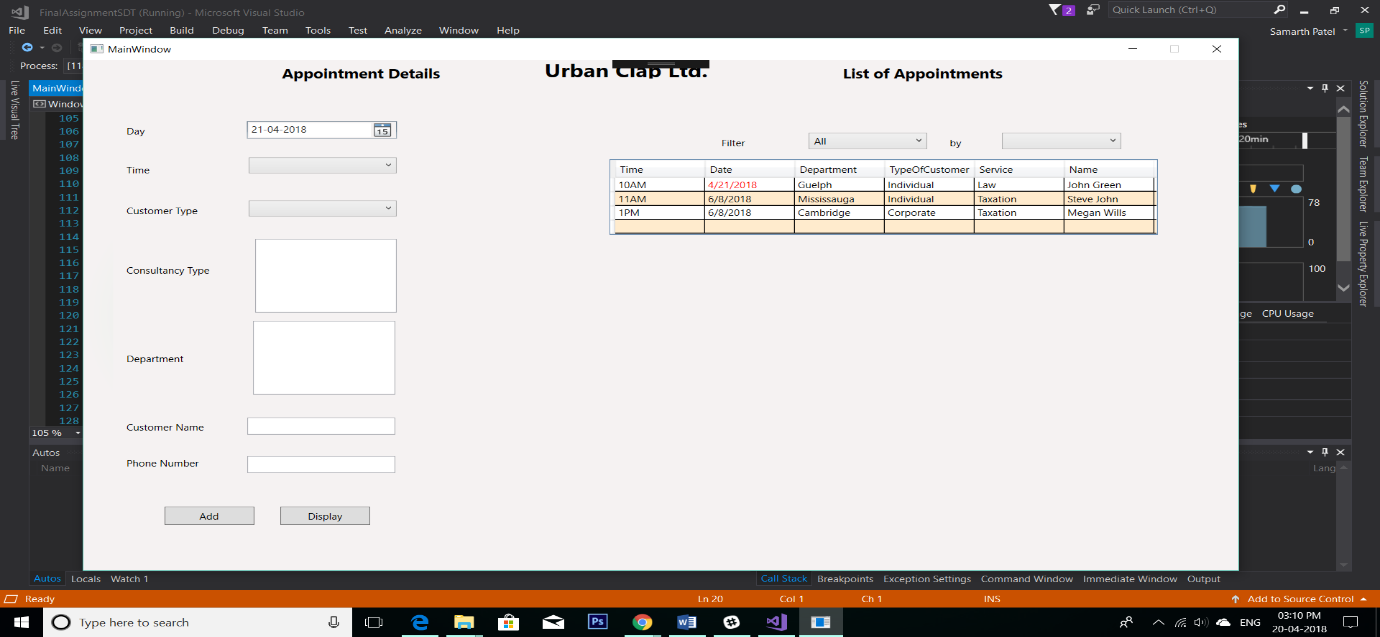


# **User Manual**

1. First the user has to select the date and time for which the appointment is to be scheduled. (Note: Also, the date and time for which there is already an appointment scheduled cannot be visible in the list, because two appointments cannot be at the same time.)
2. Now the user has to choose the type of customer form the drop-down list (i.e. Individual or Corporate).
3. Now select the Consultancy Type from the Data List, and the Department Location will be updated accordingly and can be selected, because the two list are bounded to each other.
4. And then you can enter the customer name and phone number. This system contains data validation and it will prompt if wrong data entered.
5. Click on the Add Button and the data will be saved and visible in the list beside.
6. You can also use the filter option available on the list of appointments and can be filtered fields and location.

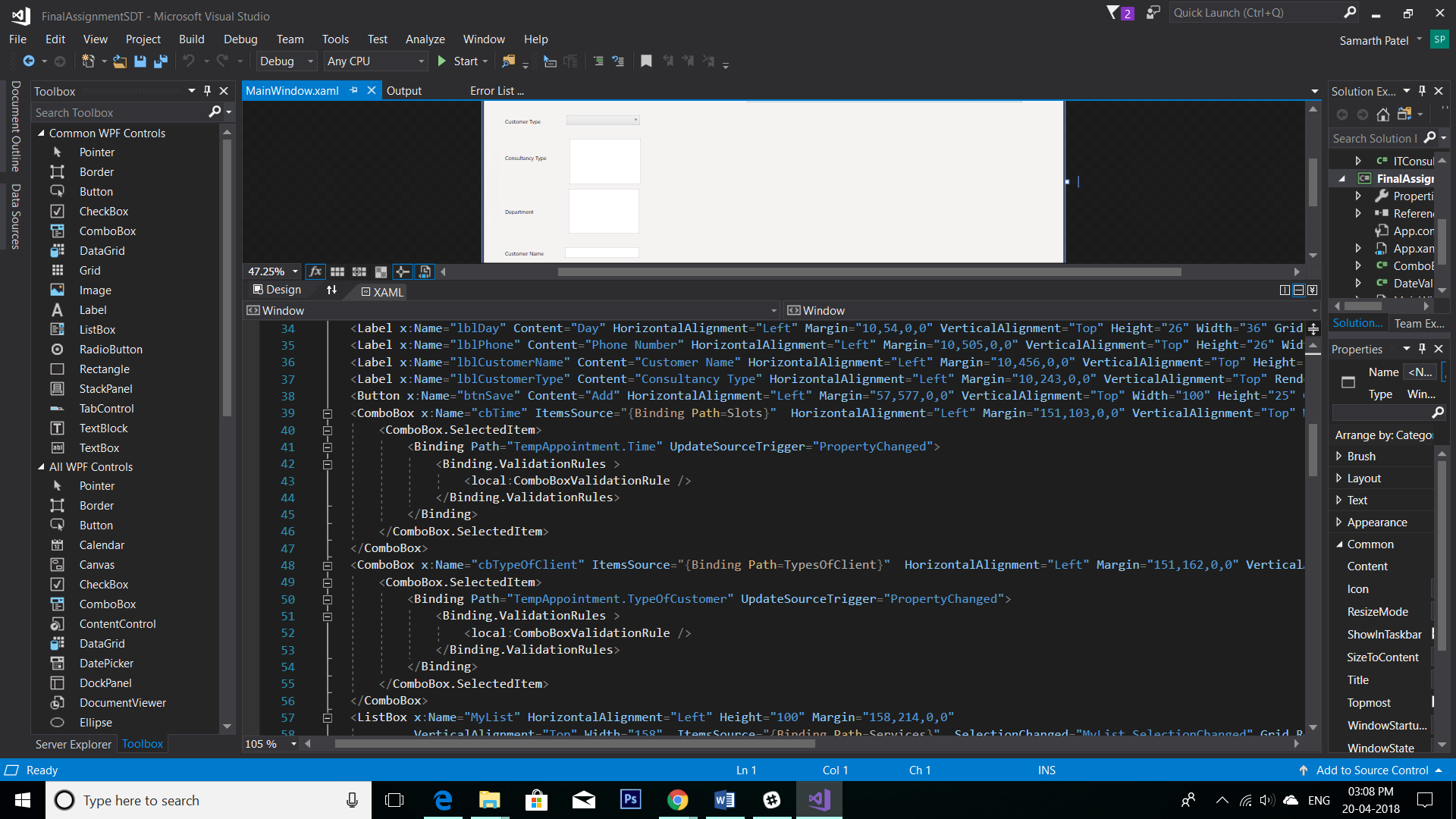
# **Rubric**

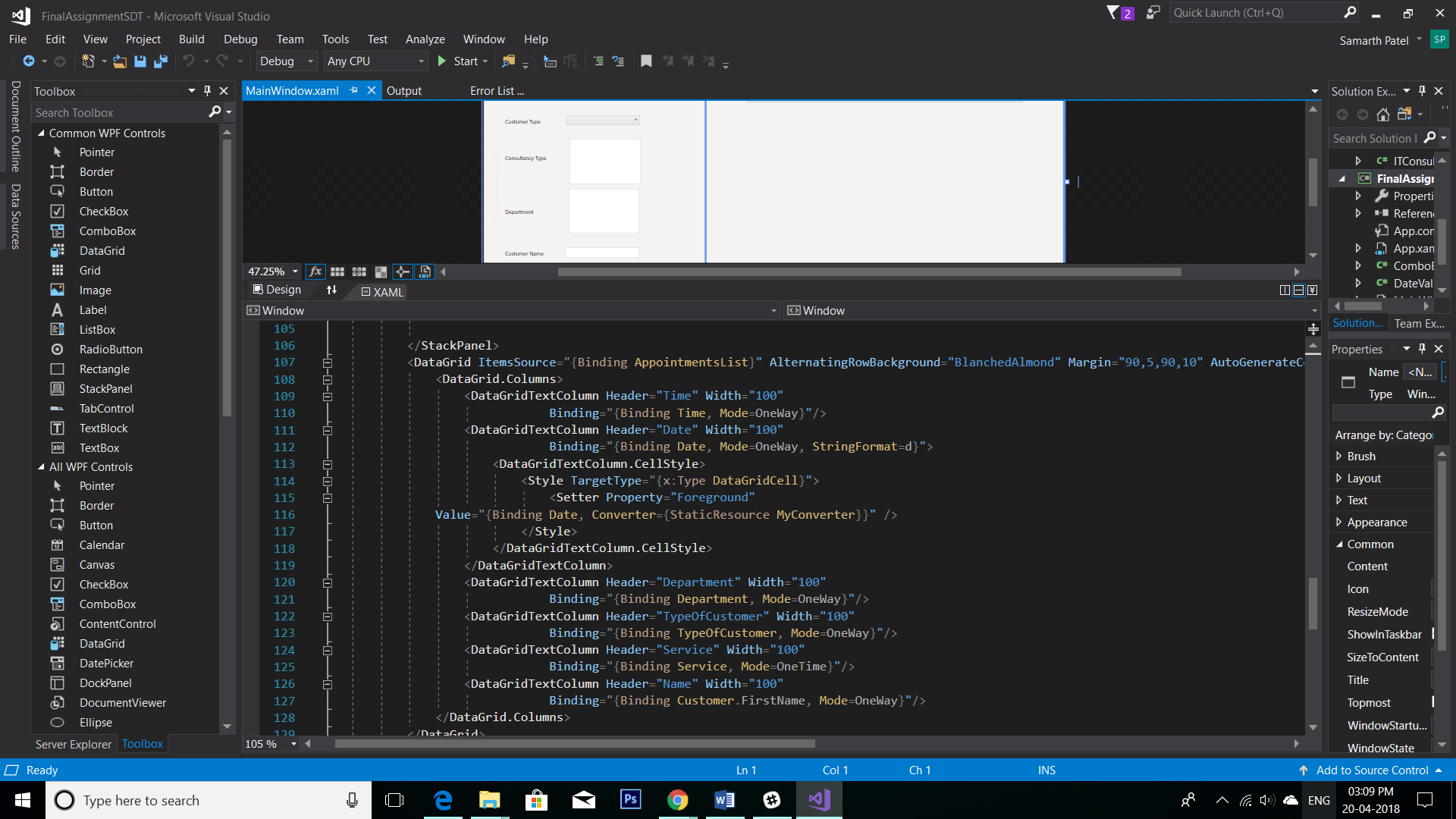
1. **GUI**
   1. **GUI-Screen Design / Grids:** Proper Screen design is developed using Grids. You can see it in the figure below.



* 1. **GUI-Binding:** All the paths are blinded to proper classes and properties.

Code: MainWindow.xaml – line (39-47, 48-56, 57-66, 70-74, 80-84, 87-96, 107-129)

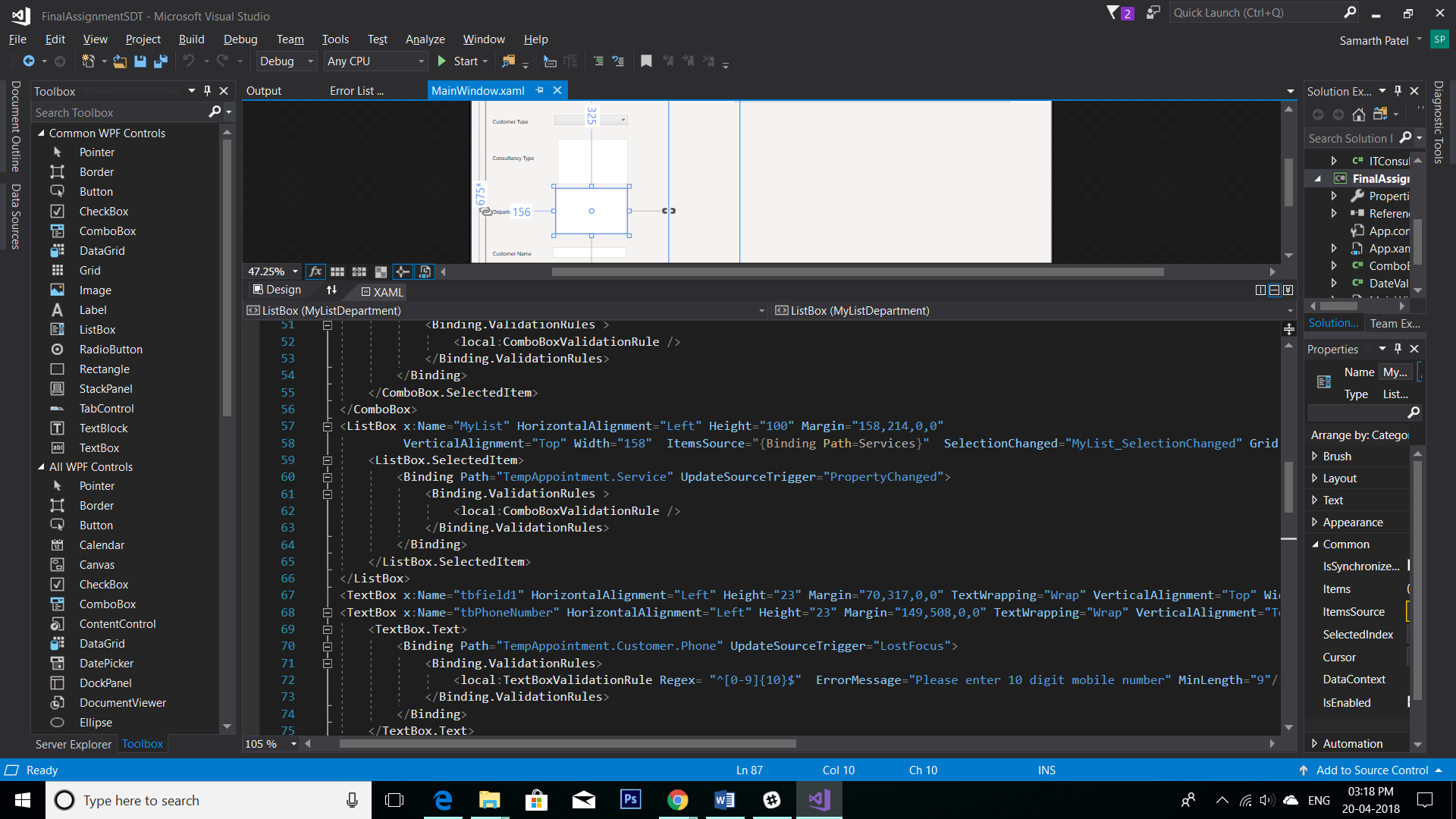


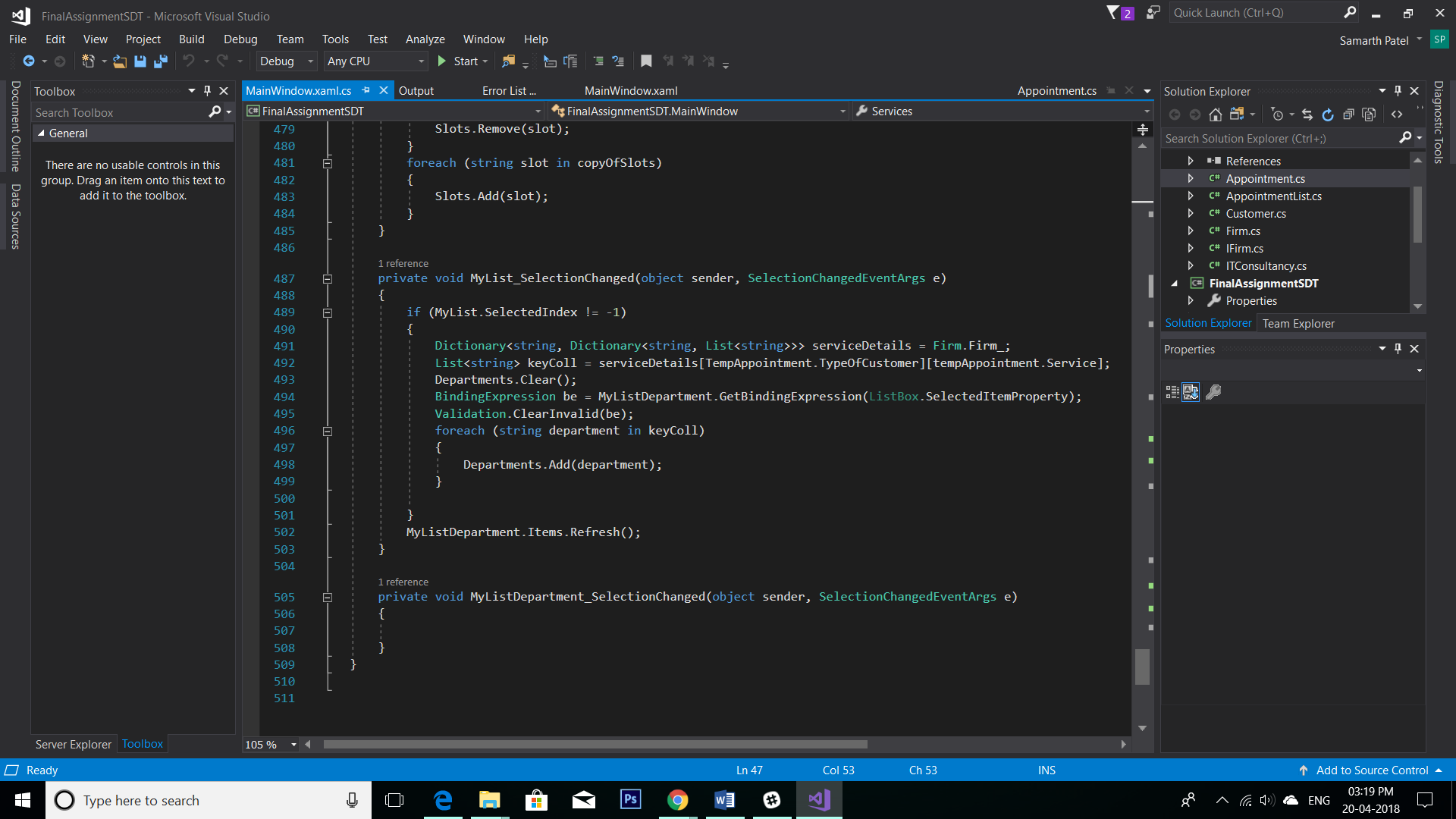


* 1. **GUI-Lists:** Two list are used in the system and bounded to each other. Below is the output and the code for List Selection Change.

Code: MainWindow.xaml – line (57-66, 87-96)

MainWindow.xaml.cs – line (487-503)





* 1. **GUI-Errors and Converters:** All the errors and converters are used properly. You can see the error is signalled as shown in figure below. Also, converter is used to convert the date to red if the appointment is less than 14 days.

Code: ComboBoxValidationRule.cs,

Date Validation.cs,

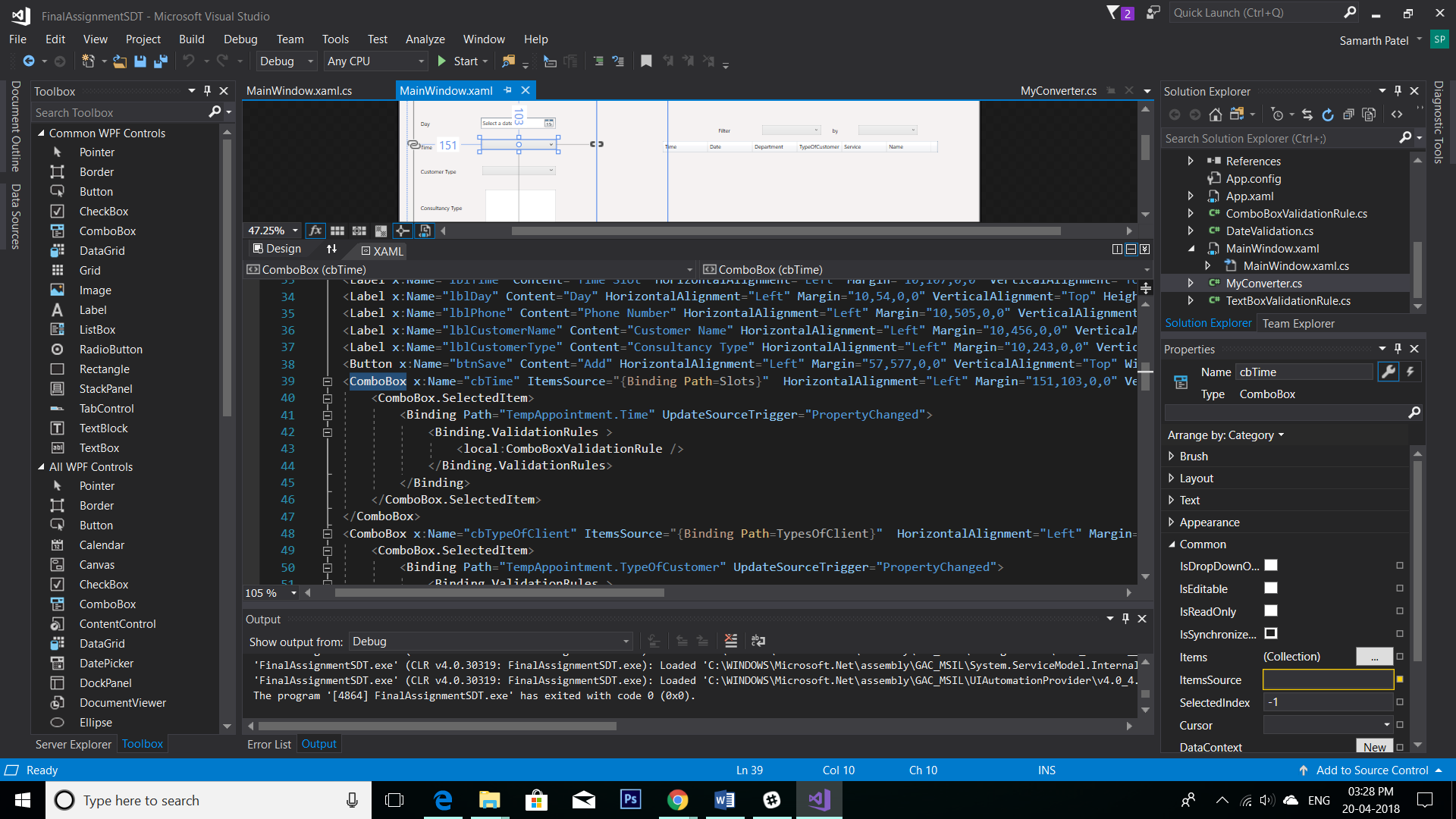
TextBoxValidation.cs,

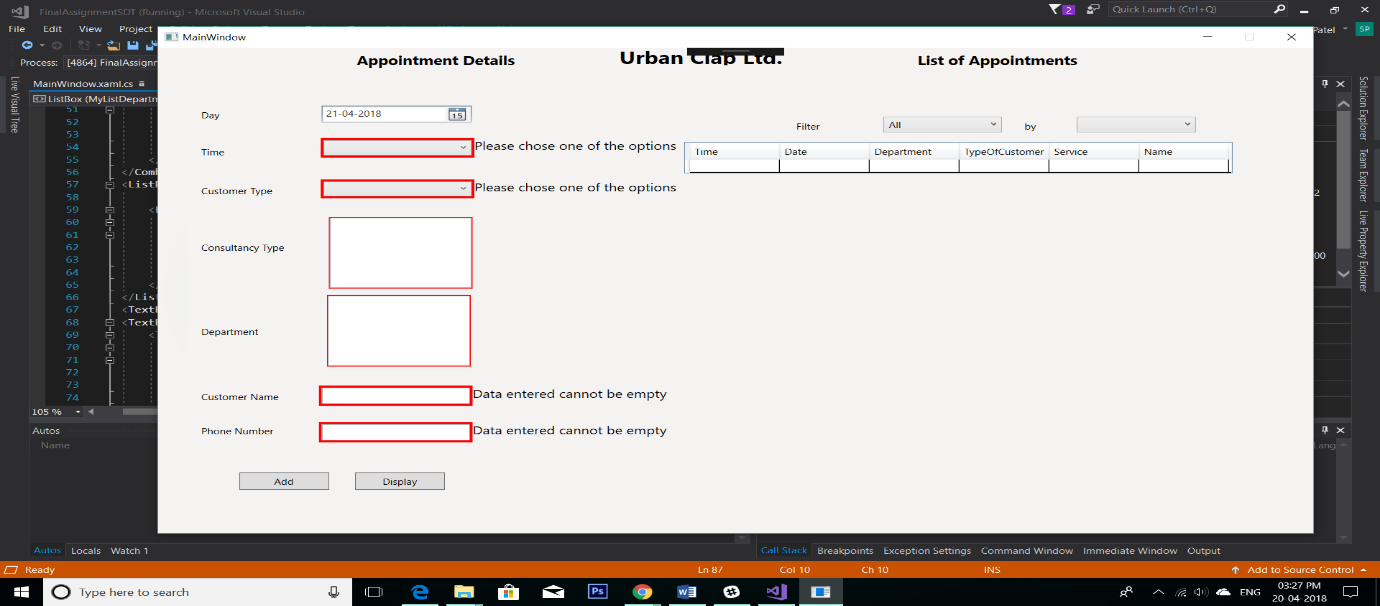
MyConverter.cs

MainWindow.xaml – line (39-47, 48-56, 57-66, 70-74, 80-84)

MainWindow.xaml.cs – line (182-233)

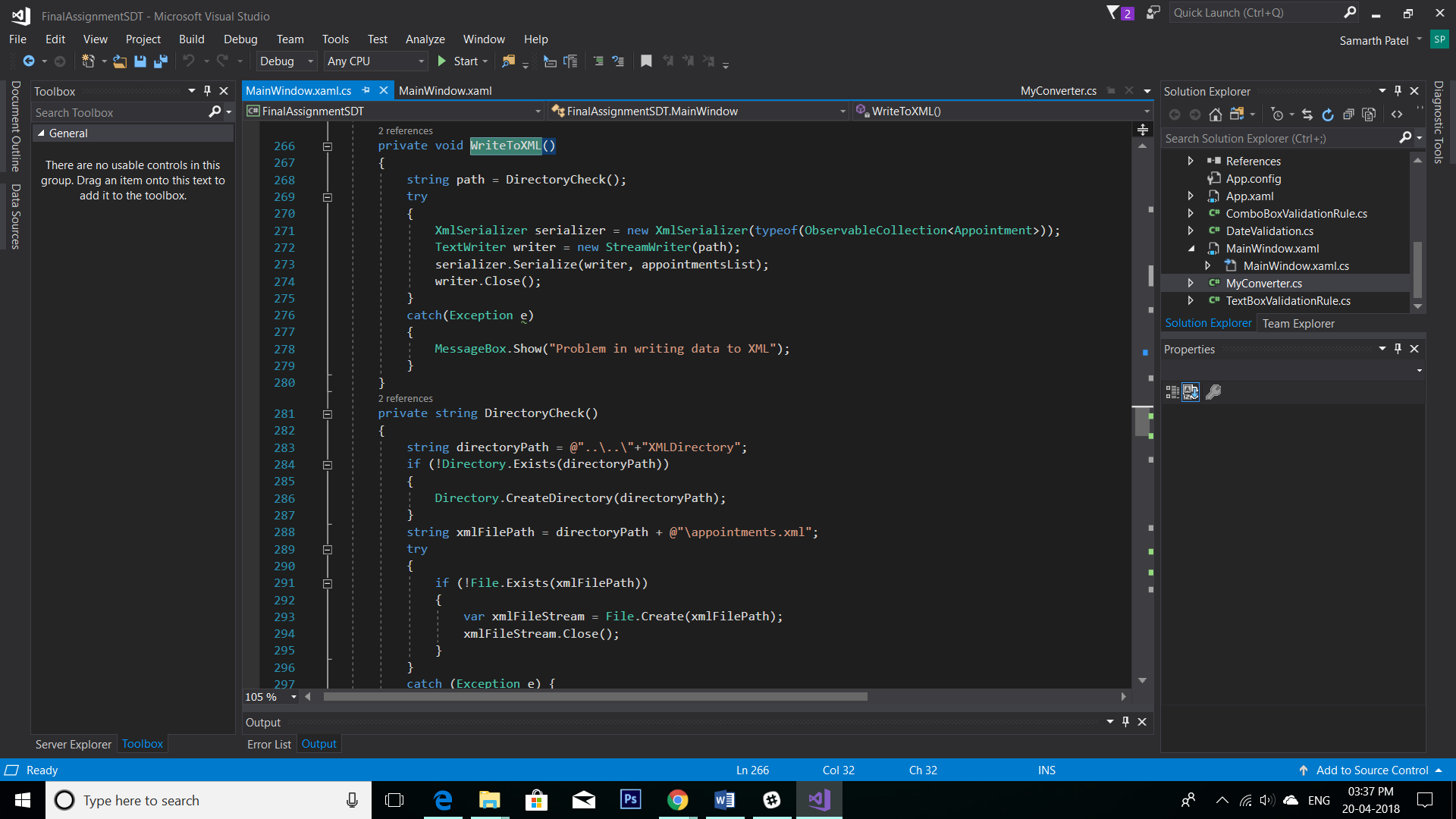
App.xaml – line(7-81)

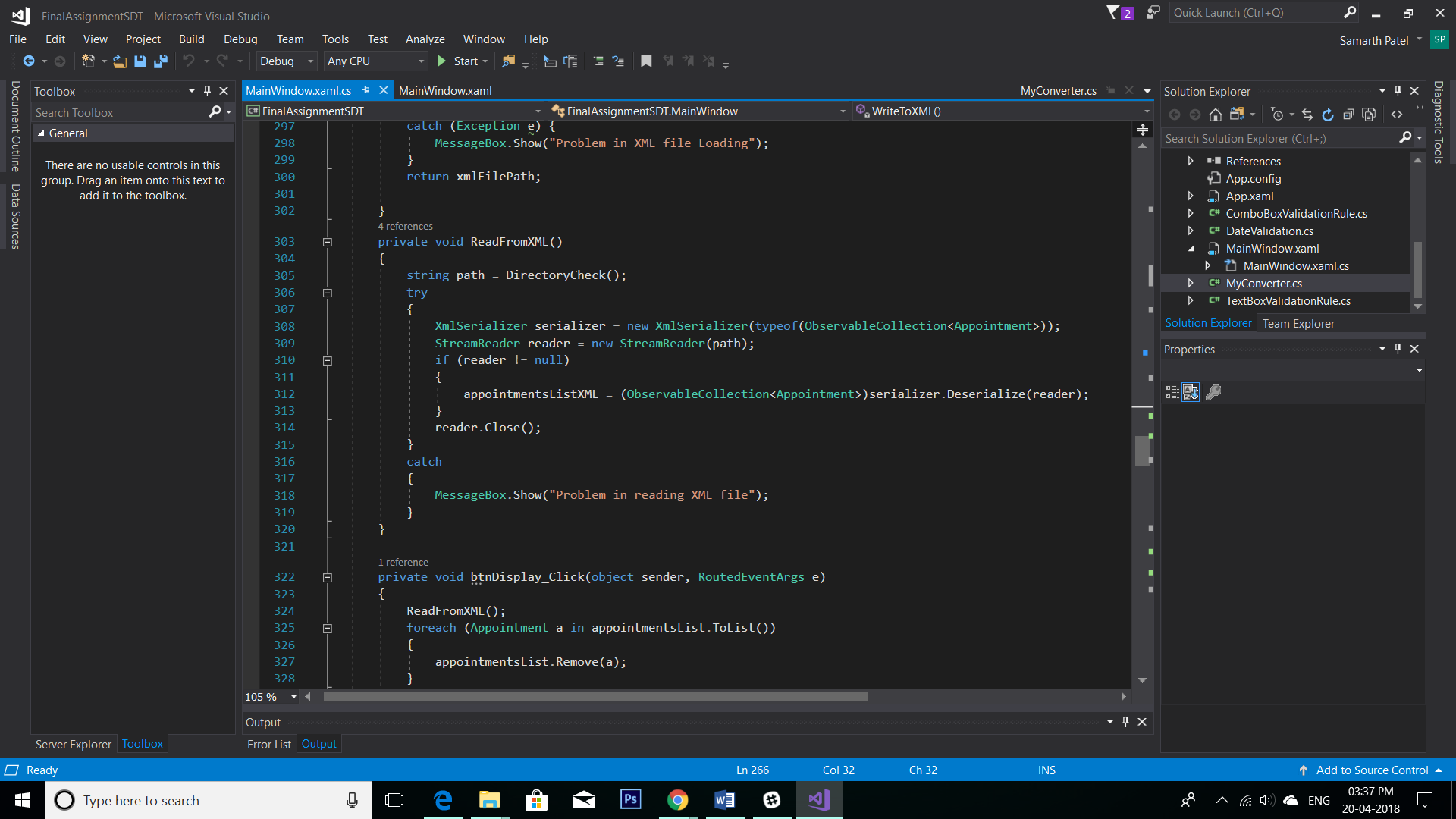




1. **File-Saving and Retrieving:** The system saves file in XML format and it uses XML Serialization and Deserialization to do so. You can see the code in the image below.

Code: MainWindow.xaml.cs – line (266-302, 303-320)





1. **LINQ – Queries:** We have used LINQ queries for filtering the data in the list.

Code: MainWindow.xaml.cs – line (392-456)

