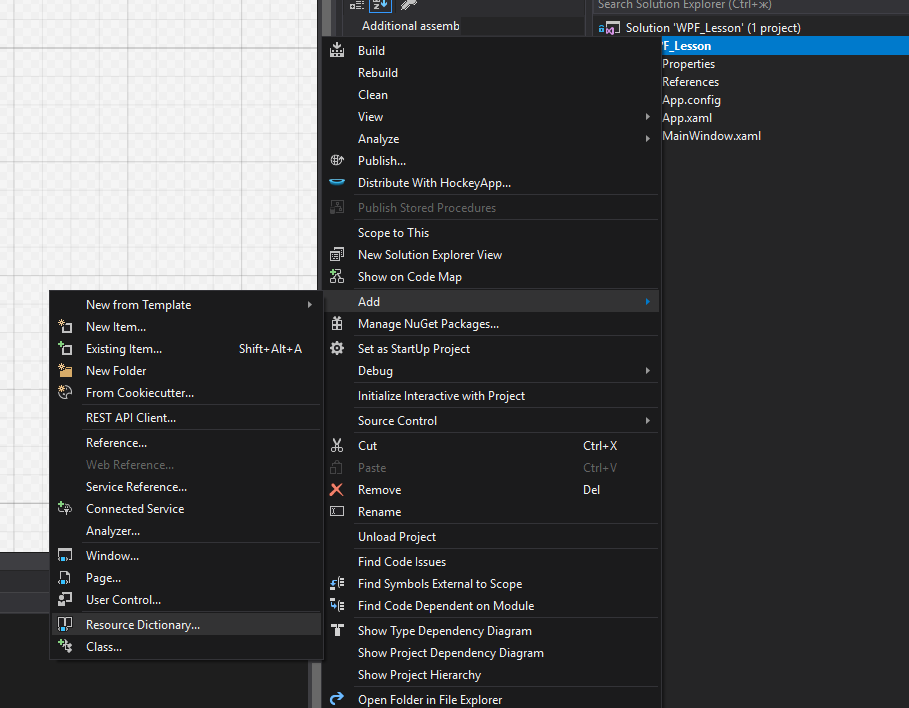
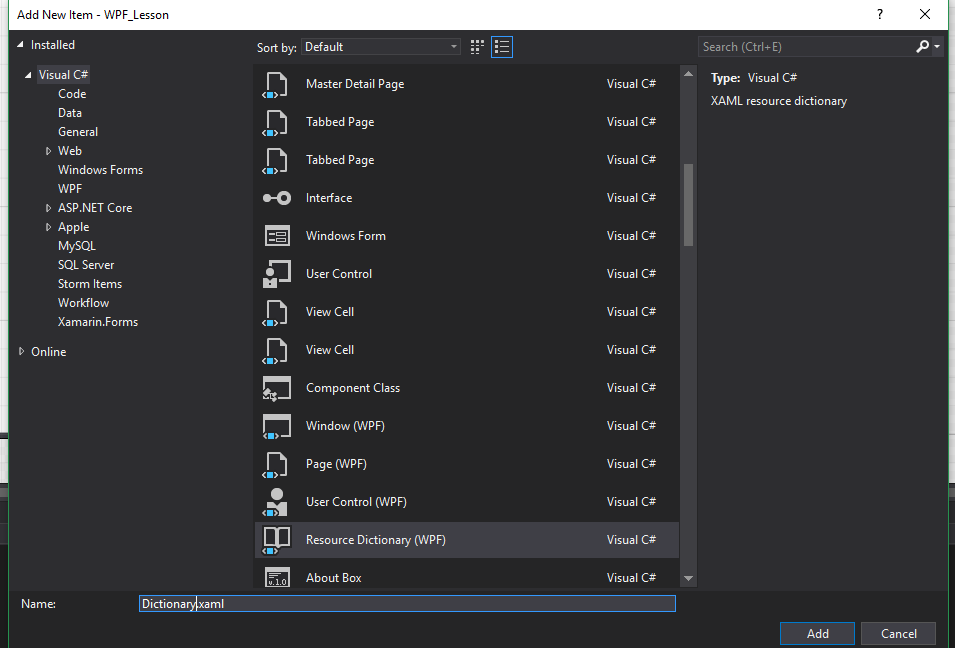
**ОФОРМЛЕНИЕ ПРИЛОЖЕНИЯ, ВВЕДЕНИЕ В ENTITY FRAMEWORK**

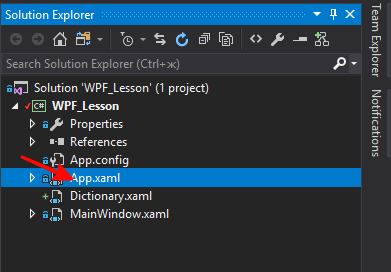
**Создание словаря стилей**



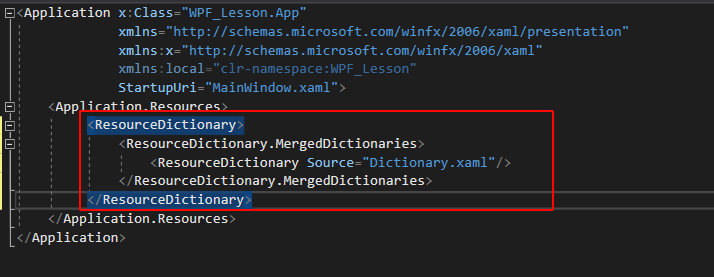
Укажем имя «Dictionary.xaml» и создадим его.



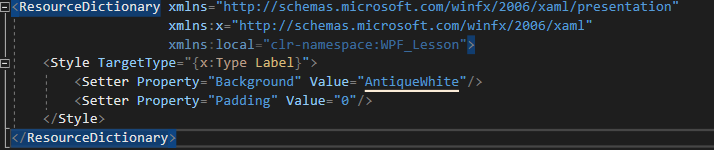
Теперь словарь необходимо подключить к проекту (чтобы компоненты могли обращаться к стилям) для этого откроем «App.xaml»

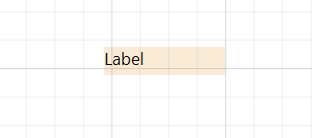


И впишем следующий код для подключения словаря.



После данных операций можно начинать писать стили и константы. Напишем первый стиль, который будет менять задний фон и внутренний отступ, и он применяться ко всем компонентам типа «Label»

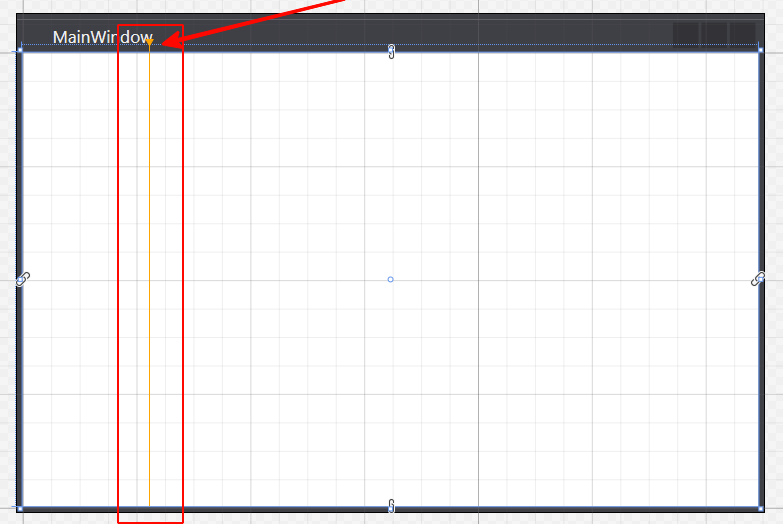




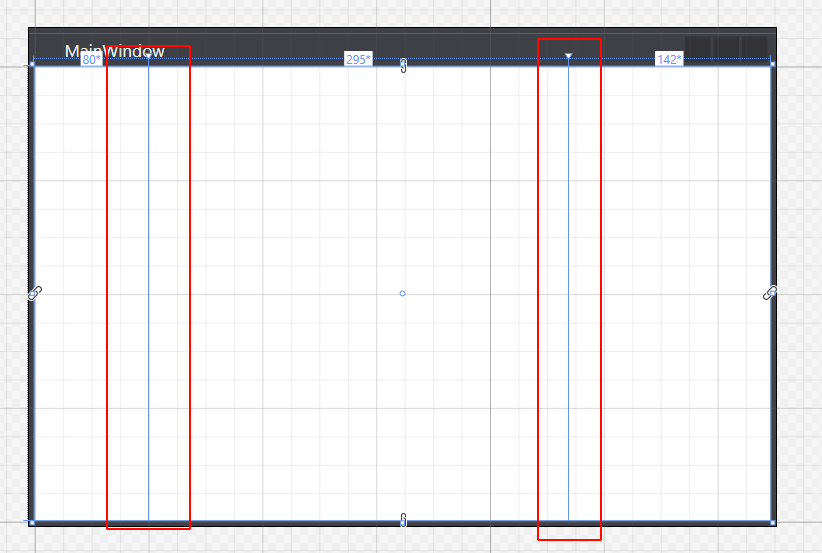
**Размещение контента по центру формы**

Откройте форму или страницу. Выберите компонент Grid, после чего по краям (сверху и слева) появится возможность добавлять строки и столбцы.

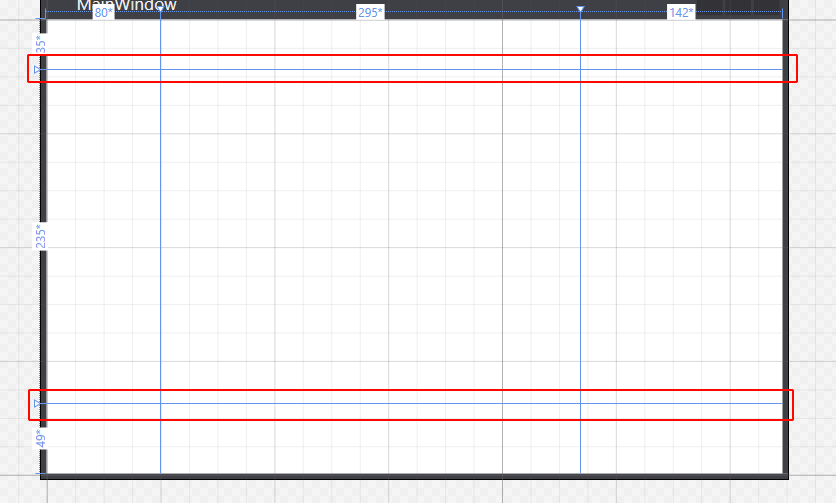
Добавим первый столбец.



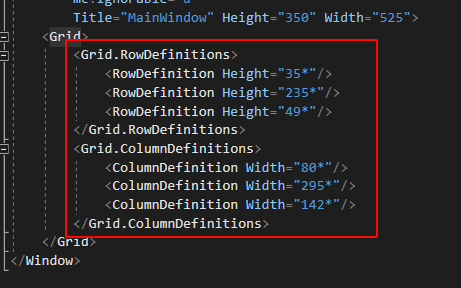
Добавим еще один столбец.



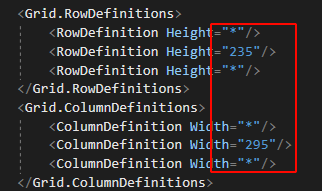
Таким же образом добавим две строки.



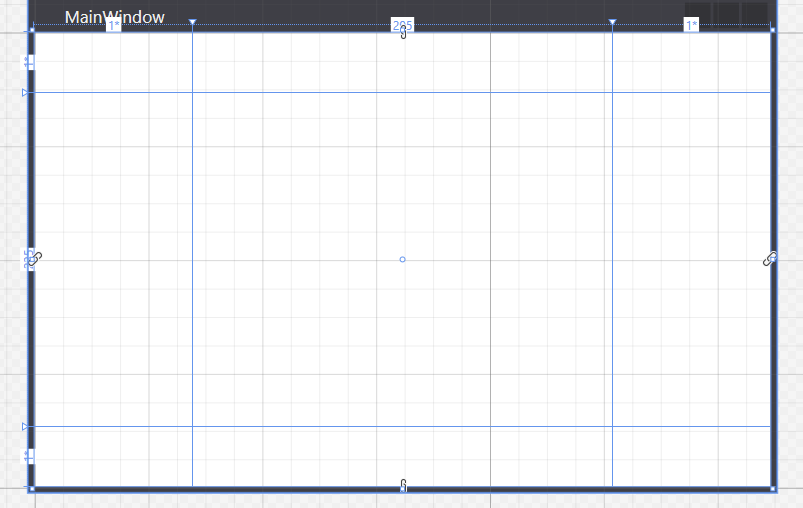
После всех добавление мышкой, мы можем увидеть следующий код XAML.



Так же столбцы и строки можно добавлять, редактируя код XAML. Здесь же можно редактировать размер строк и столбцов. Чтобы разместить элементы по центру (которые будут во второй строки и во втором столбце) необходимо задать размеры строкам и столбцам. Зададим столбцам и строкам следующие размеры.



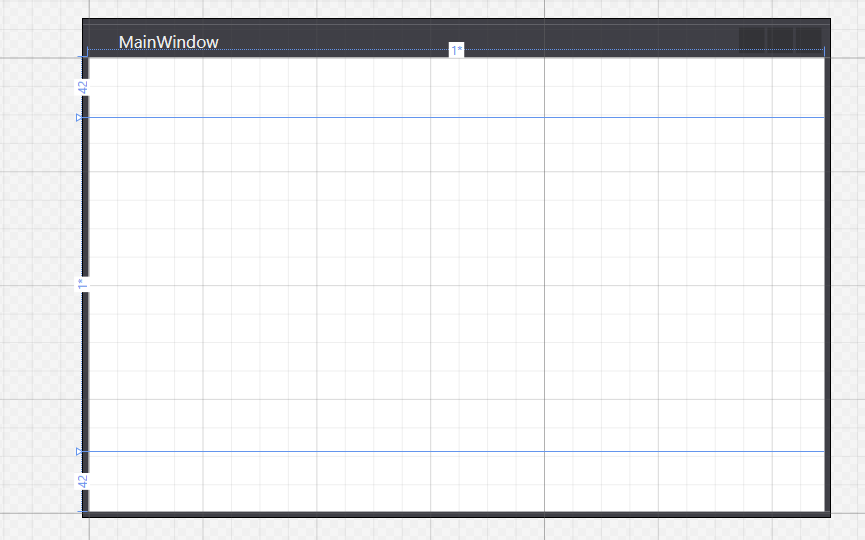
После чего на форме увидим следующее.

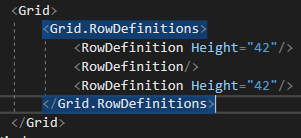


В результате средний столбец и средняя строка имеют фиксированный размер, в то время как последние и первые имеют динамический размер (меняется в зависимости от размера контейнера, окна).

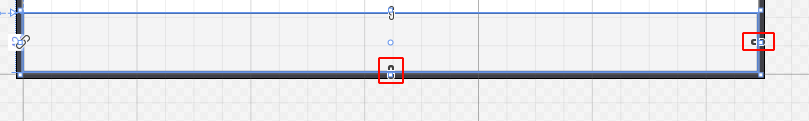
**Создание базовой формы**

Создадим несколько строк с фиксированными размерами.





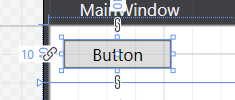
Добавим компонент «Rectangle» на форму и разместим его в 3 строке. Компонент необходимо растянуть на всю ширину и высоту строки, а затем закрепить высоту и ширину (чтобы он мог растягиваться)



Укажем у компонента «Rectangle» свойство «StrokeThickness» равное 0, это задаст размер границы прямоугольника.

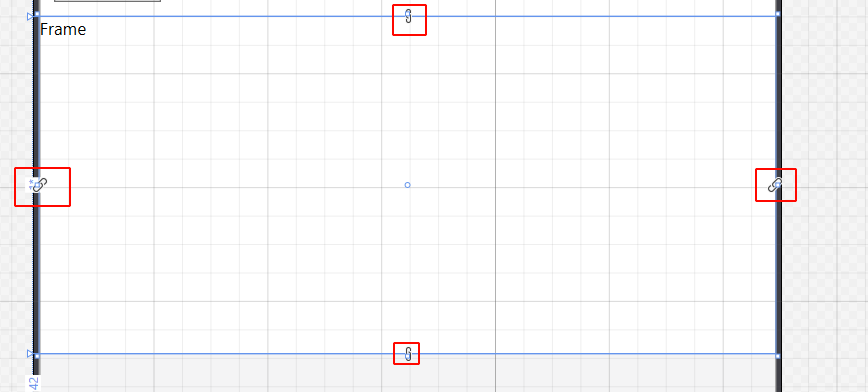


Добавим на форму кнопку и разместим в левом углу первой строки. И укажем имя кнопки «Назад», которая в последствии будет выполнять соответствующею функцию.



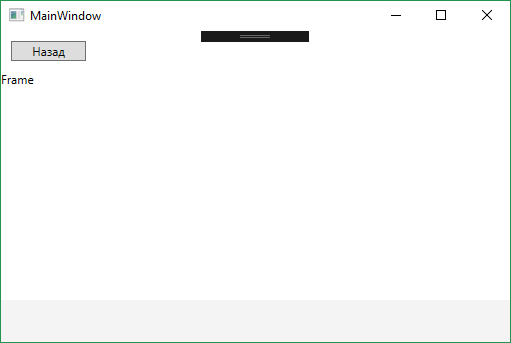


Добавим на форму компонент «Frame» во вторую строку, в него будут загружать страницы (Авторизация, Регистрация и другие). Закрепим компонент по высоте и ширине. Укажем имя ему «MainFrame»



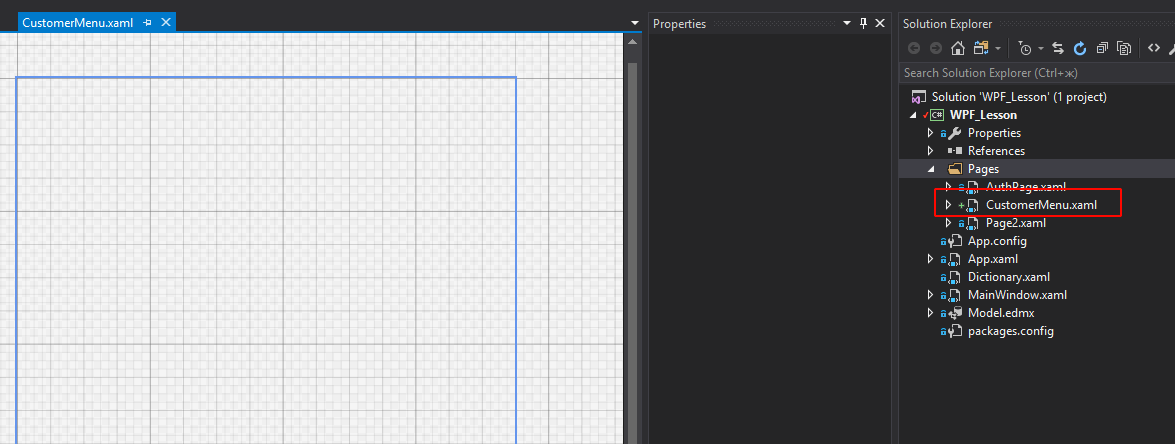


Запустим проект и увидим примерно следующее.

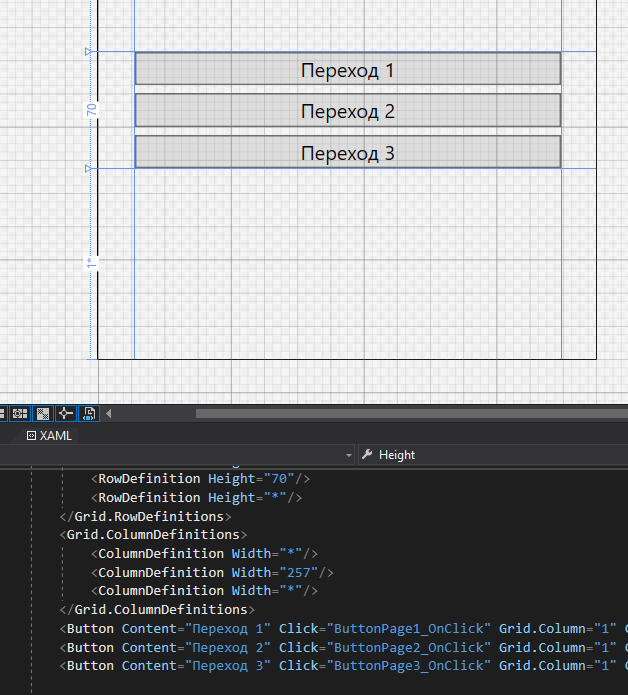


**СОЗДАНИЕ МЕНЮ ПОЛЬЗОВАТЕЛЯ**

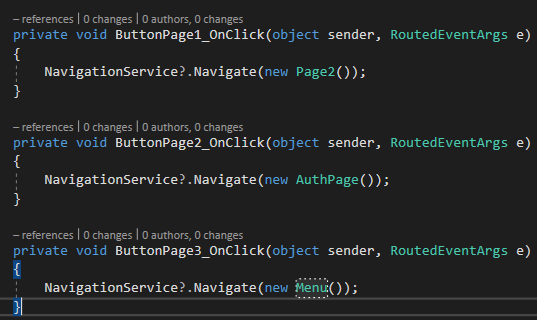
Создадим новую страницу, назовем ее CustomerMenu.



Добавим несколько кнопок, разместим их по центру и добавим им обработчик события клик.



И в коде каждому обработчику укажем свой переход



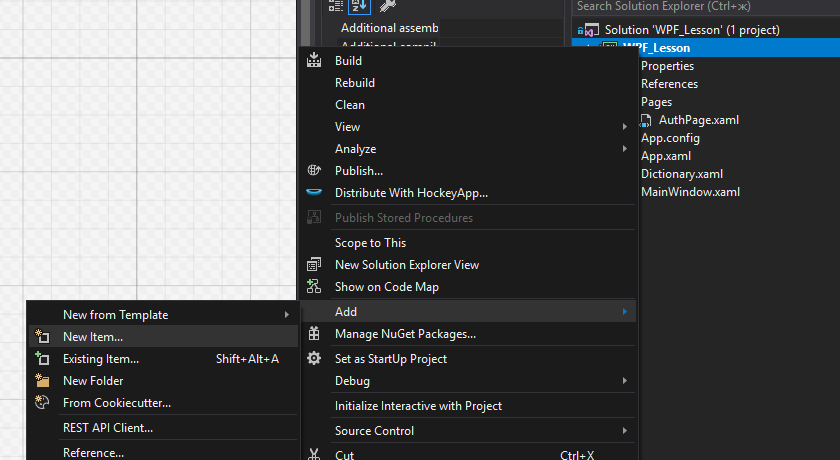
**Создание подключения к базе данных**

Entity Framework — это решение для работы с базами данных, которое используется в программировании на языках семейства.NET. Оно позволяет взаимодействовать с СУБД с помощью сущностей (entity), а не таблиц. Также код с использованием EF пишется гораздо быстрее.

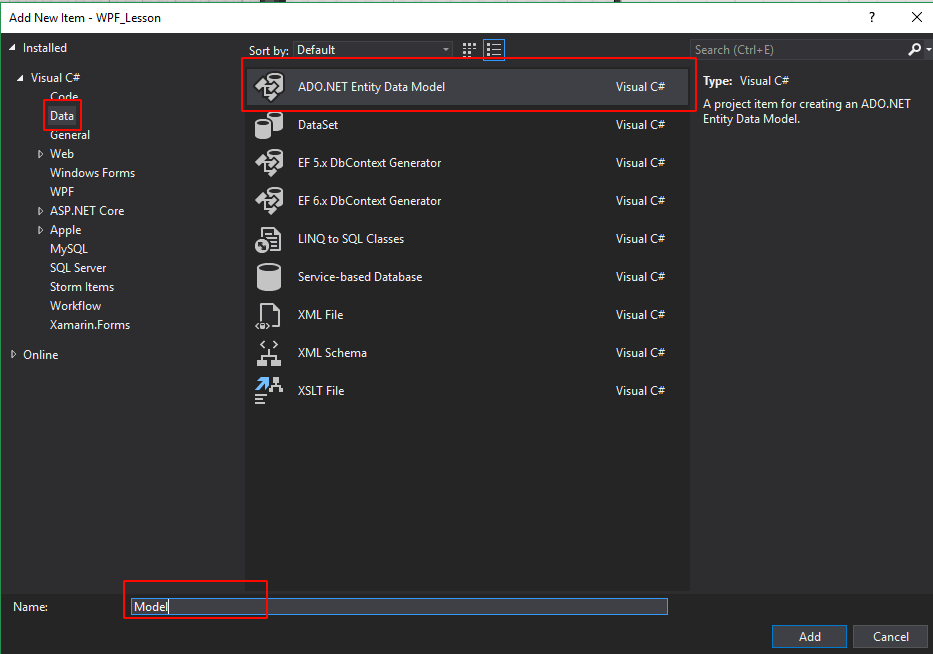
Например, работая с базами данных напрямую, разработчик должен беспокоиться о подключении, подготовке SQL и параметров, отправке запросов и транзакций. На Entity Framework всё это делается автоматически — программист же работает непосредственно с сущностями и только говорит EF, что нужно сохранить изменения.

В этой статье будут разобраны основы применения Entity Framework, для понимания которых нужно владеть ADO.NET — пользоваться базами данных, писать SQL-запросы и работать с подключениями.

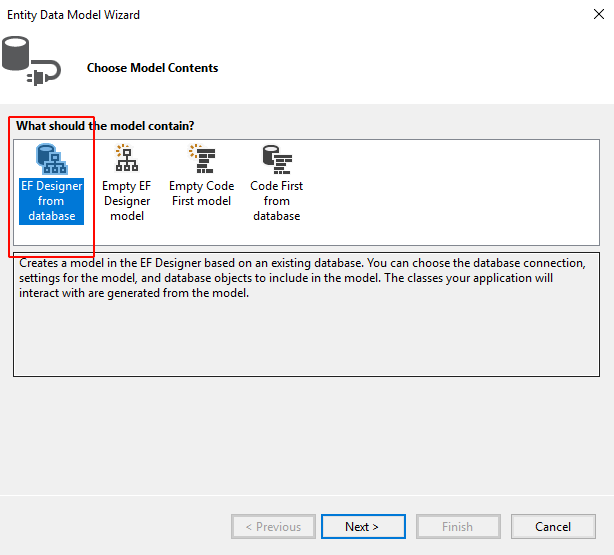
Добавим в проект новый объект



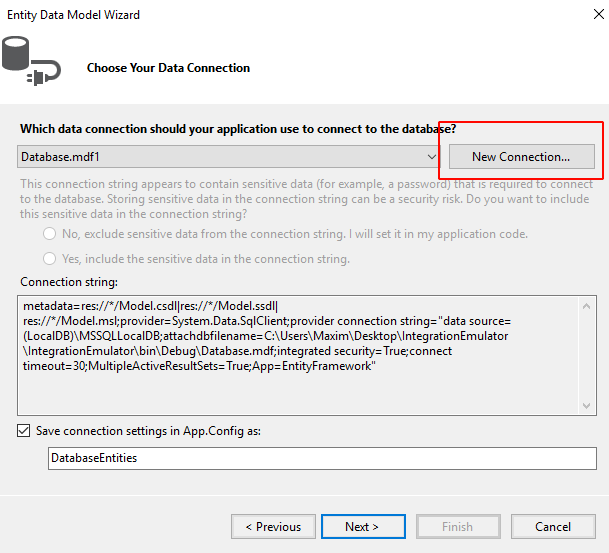
Выберем Data – ADO.NET Entity Data Model, назовем Model и добавим.



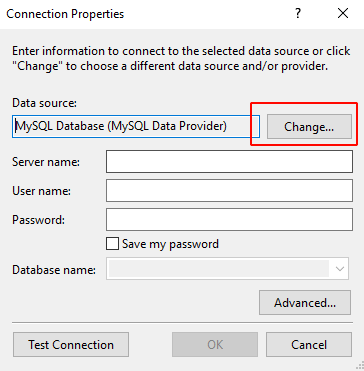
Появится диалоговое окно и выберем пункт «EF Designer from database»



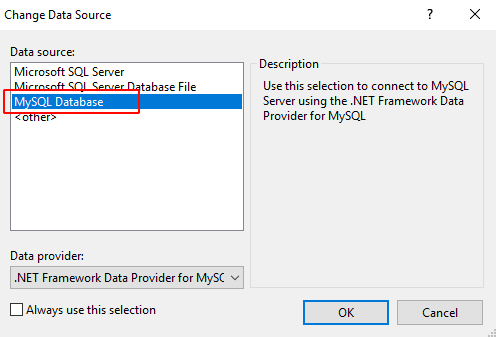
После появится следующее окно. Нажмем «New Connection»



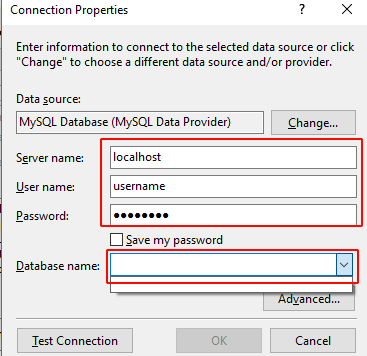
В данном окне выберем «Change».



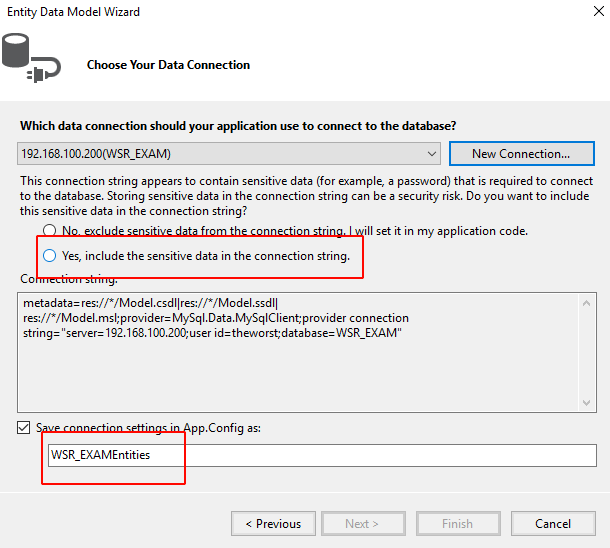
В следующем окне выберем «MSSQL Database» и нажмем «ОК».



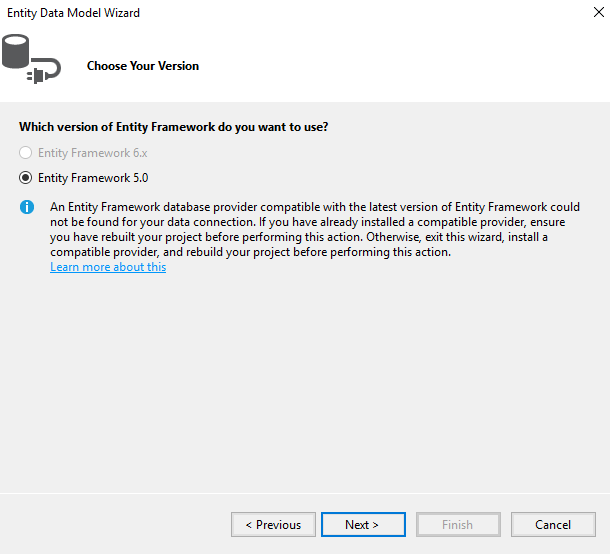
После выбора введем данные сервера: IP адрес, логин и пароль. После выберем нужную базу данных.



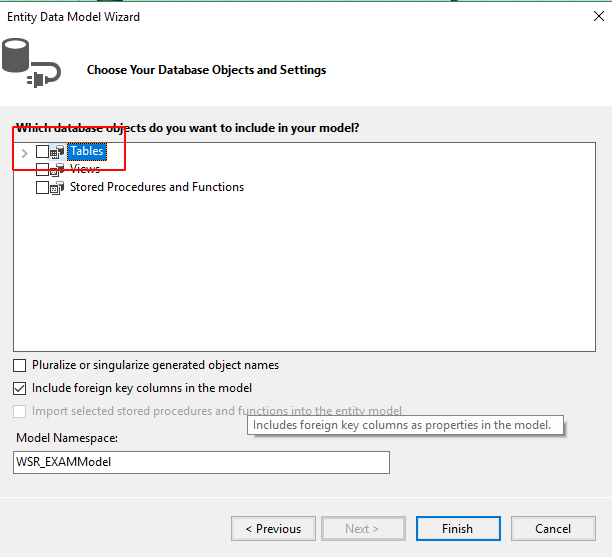
На следующем окне, выберем созданное подключение и поставим «Yes, include the sensitive data in the connection string», а также укажем имя нашего объекта базы данных «Entitites».



В следующем окне оставим по умолчанию:

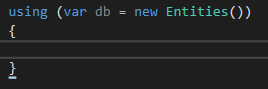


В этом окне выберем таблицы, которые хотим использовать в проекте:

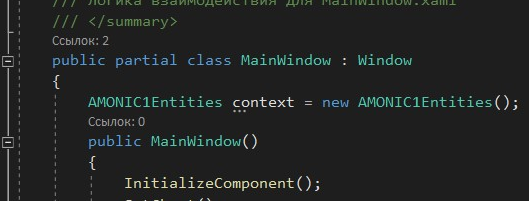


**Получение данных из базы данных**

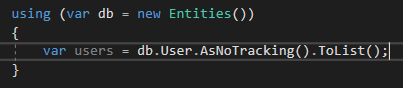
Для подключения к базе данных необходимо создать контекст.



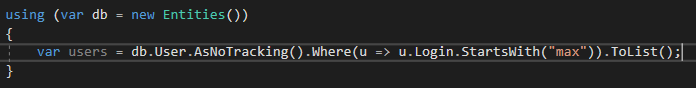
ПРИМЕЧАНИЕ: Подключить контекст можно и следующим способом, чтобы постоянно не прописывать using, соответственно объявив контекст в начале файла, вы просто прописываете код без using и фигурных скобок.



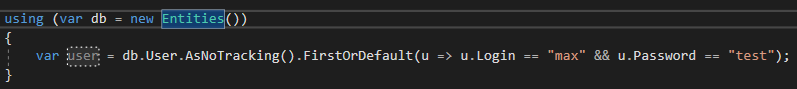
Теперь загрузим всю таблицу пользователей.



Получим пользователей по определенному критерию

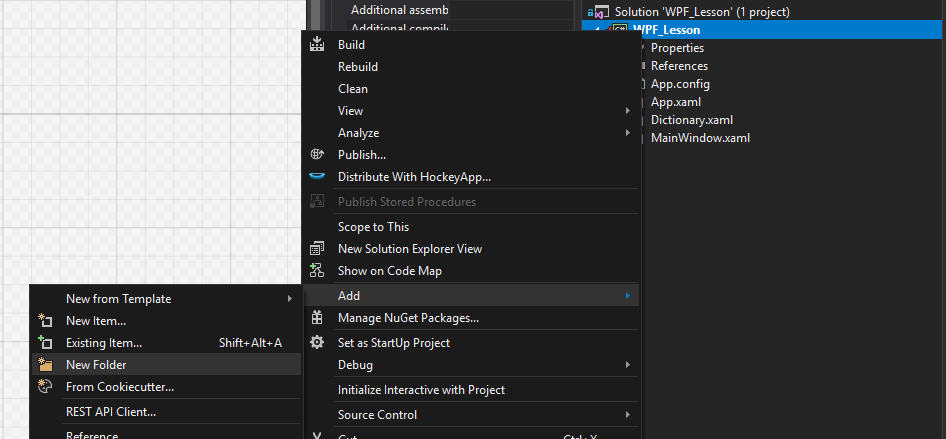


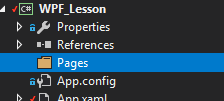
Получим пользователя по определенным критериям



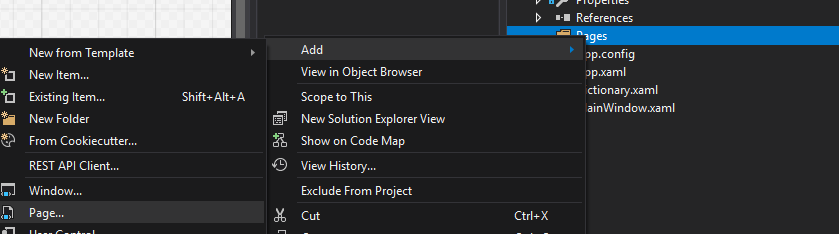
**Создание формы авторизации**

Добавим в проект новую папку и назовем ее «Pages», в этой папке будут находится страницы (авторизация, регистрация и другие)

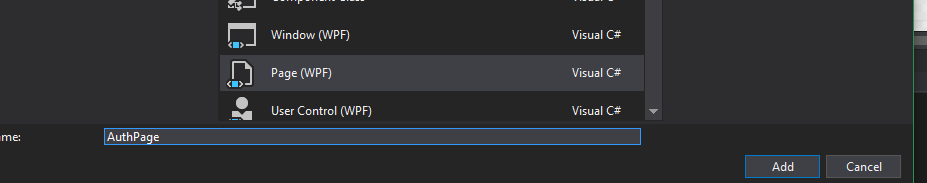


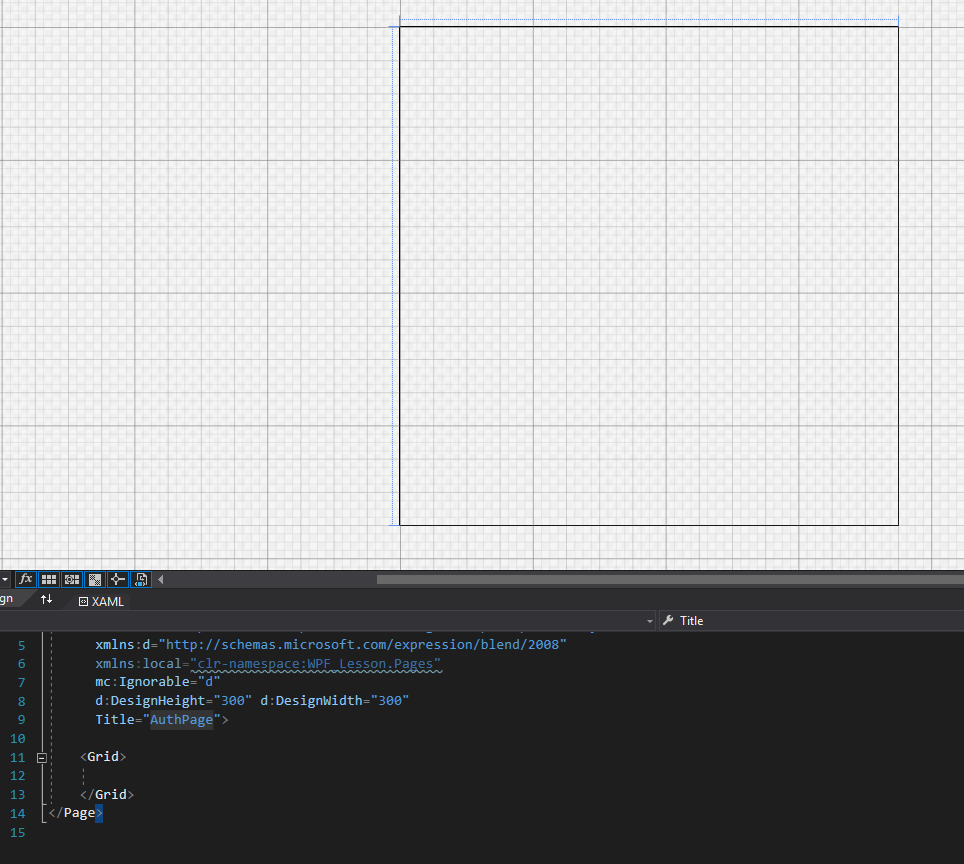


Добавим в папку новую страницу.

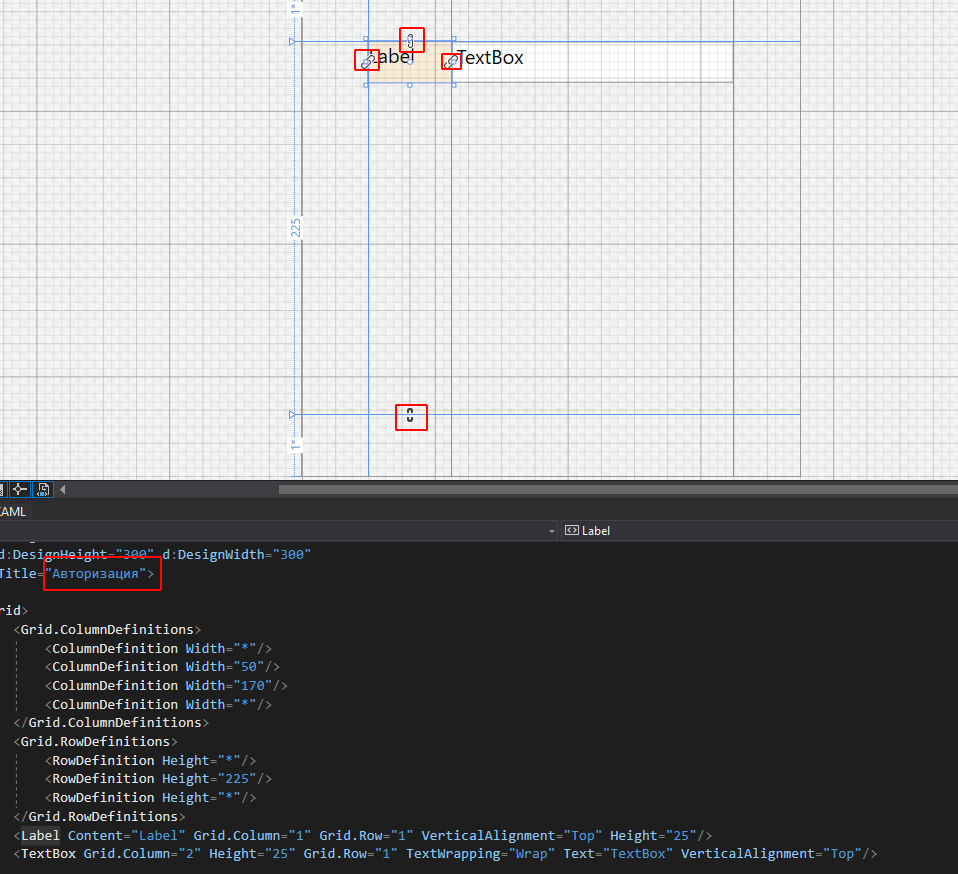


Назовем страницу AuthPage и создадим ее. После чего появится пустая страница.

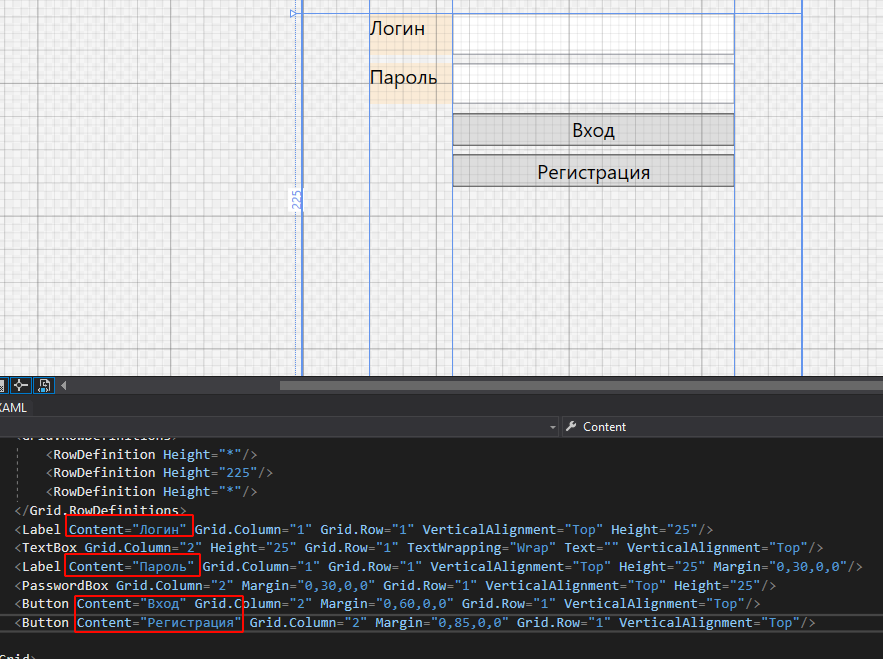




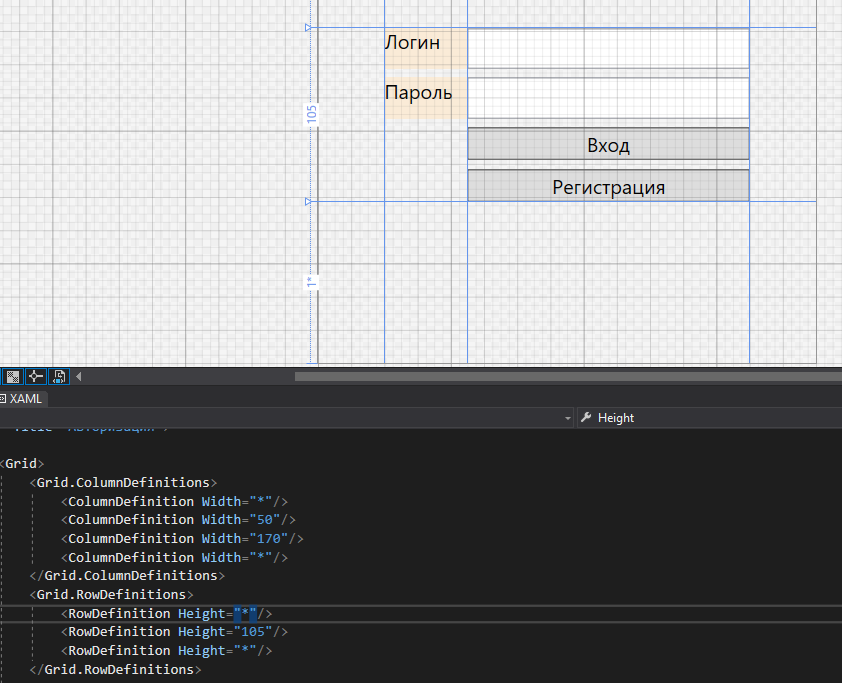
Добавим на форму компоненты «Label» и «TextBox», а затем от центруем их и добавим еще один столбец (для размещения лейблов). Также не забудем переименовать страницу, а затем закрепить лейбл и текстбокс, чтобы они растягивались по ширине.



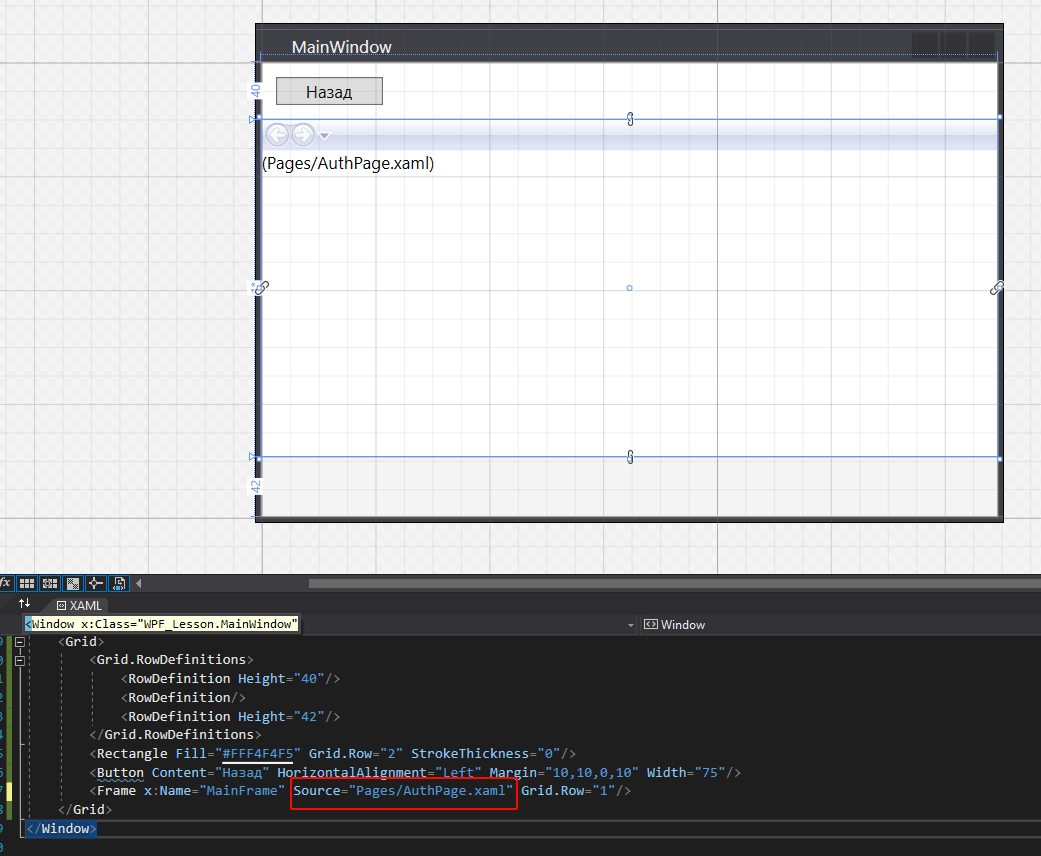
Добавим остальные компоненты по аналогии и переименуем их.



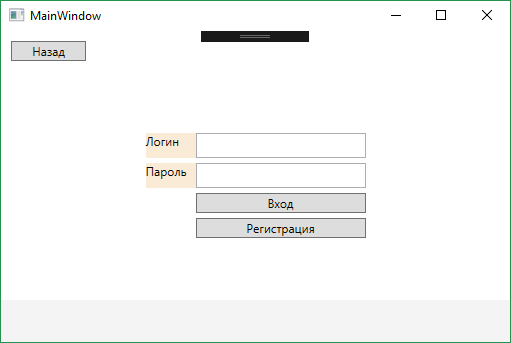
В результате получится примерно так.



Перейдем на главную форму и в компоненте «Frame» укажем в свойстве «Source» нашу страницу «AuthPage.xaml»

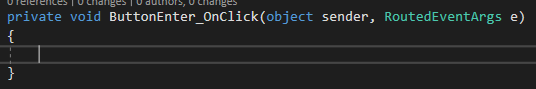


Запустим проект и увидим, что при запуске теперь отображается страница авторизации.

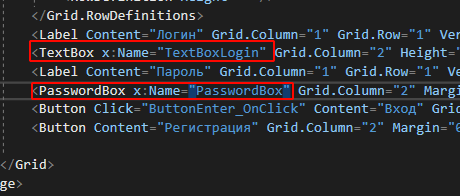


Теперь добавим функционал. Добавим обработчик события на кнопку вход.

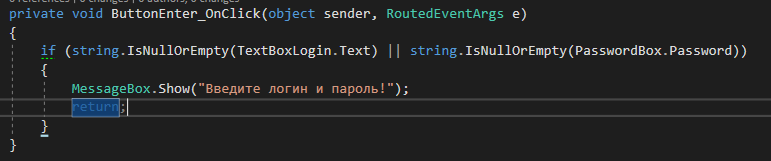




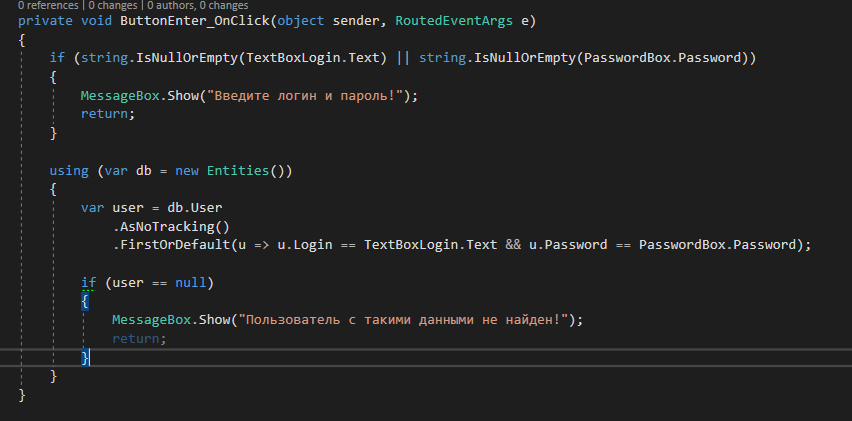
Добавим полям имена:



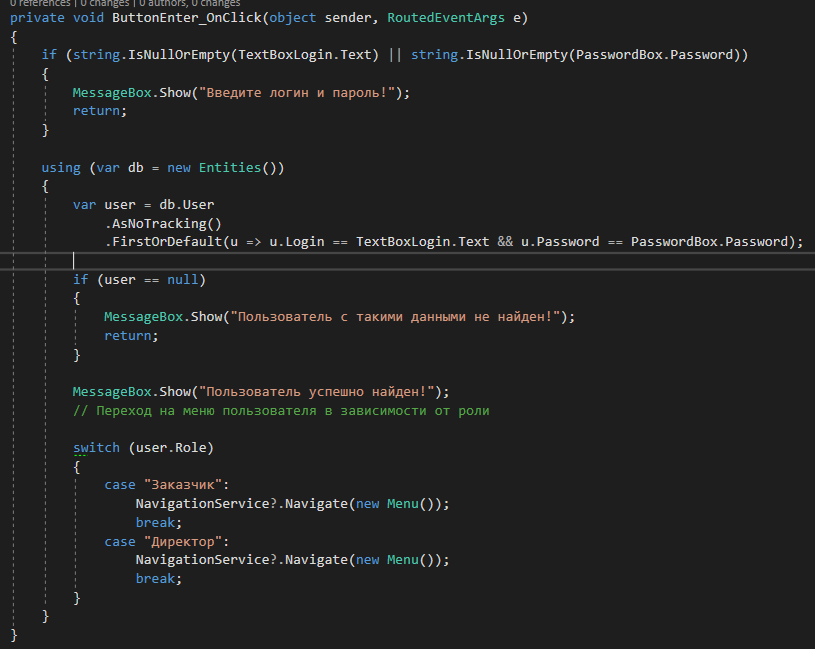
Добавим в код базовую проверку



Добавим запрос к базе данных:

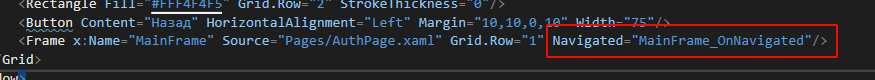


И теперь добавим переходы в зависимости от роли на меню пользователя (для этого необходимо создать страницы меню для каждого типа пользователя, CustomerMenu или DirectorMenu и тд)

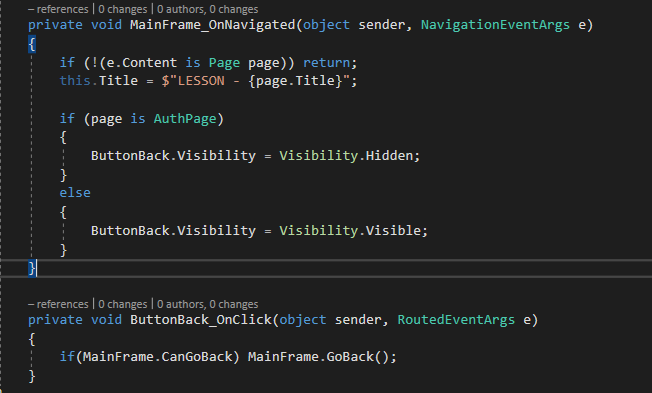


**Переход между страницами**

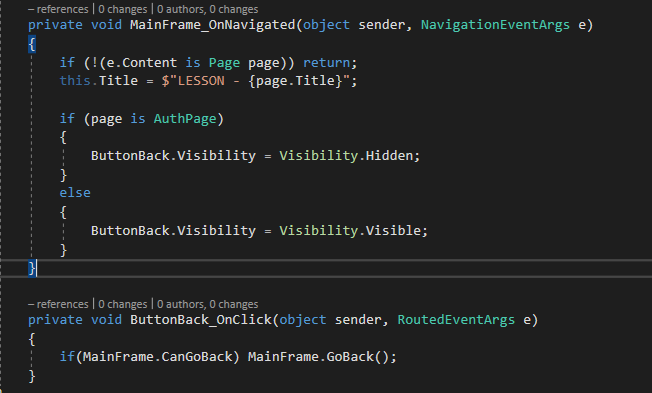
Перейдем на базовую форму и создадим обработчик события Navigated у Frame.



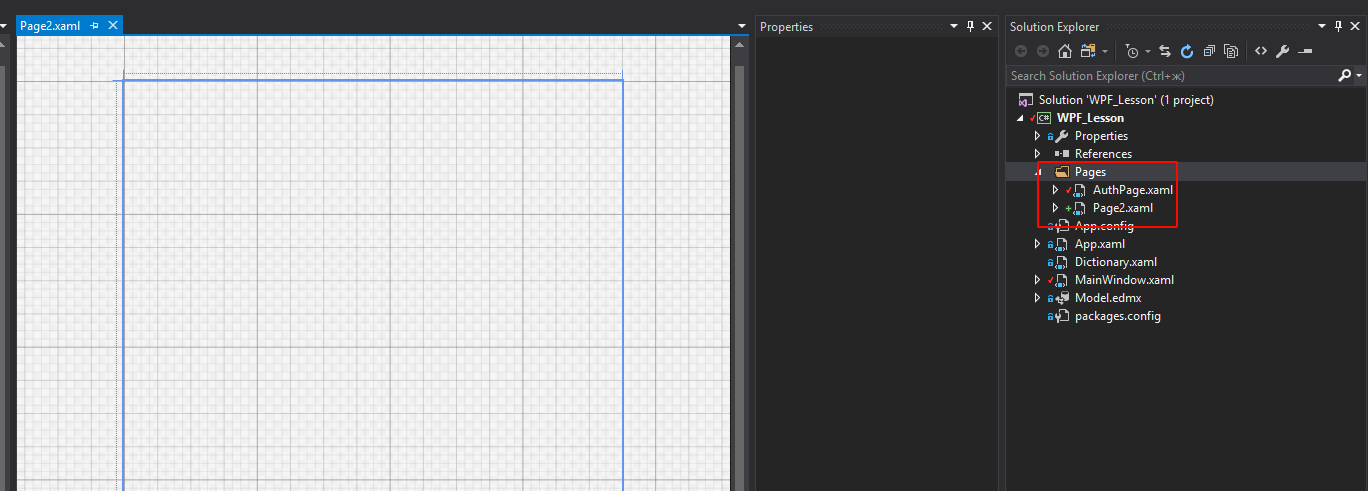
В обработчик события напишем следующий код. Сначала мы проверяем что получили ли мы страницу на вход, затем устанавливаем заголовок формы в соответствии с шаблоном, после в зависимости от страницы отображаем или скрываем кнопку «Назад».



А в обработчик события нажатия кнопки «Назад». При нажатии на кнопку будет выполнен переход назад, если такой возможен.



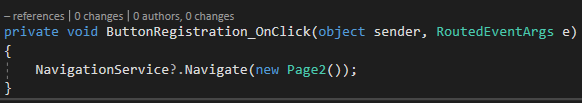
Добавим еще одну страницу.



Теперь выполним переход со страницы AuthPage на Page2. Перейдем на страницу AuthPage и добавим обрабочик события на кнопку «Регистрация»:



Напишем следующий код в обработчике:



После этого переход по кнопке будет осуществляться на страницу Page2, а по кнопке «Назад» обратно на AuthPage.