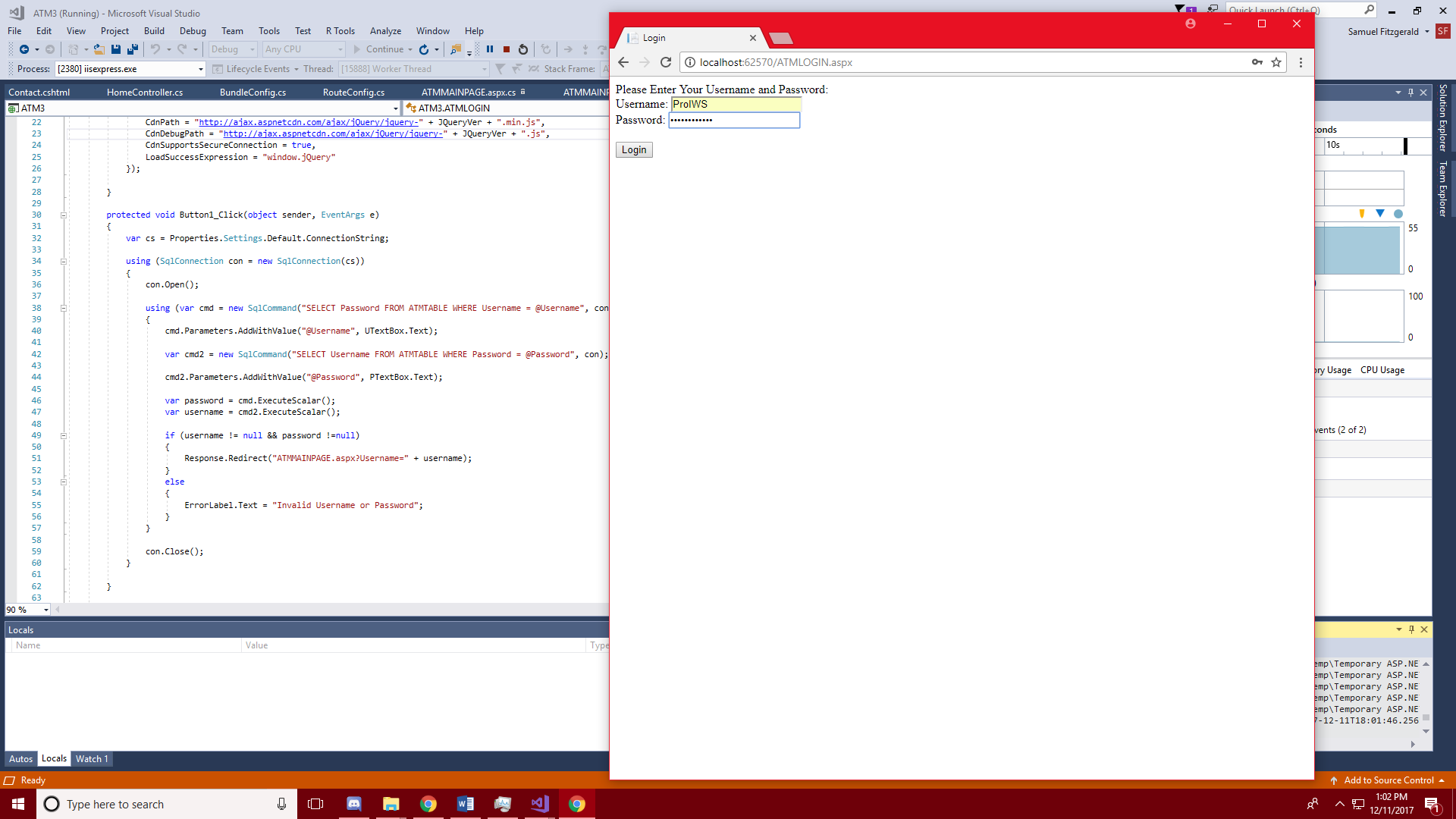
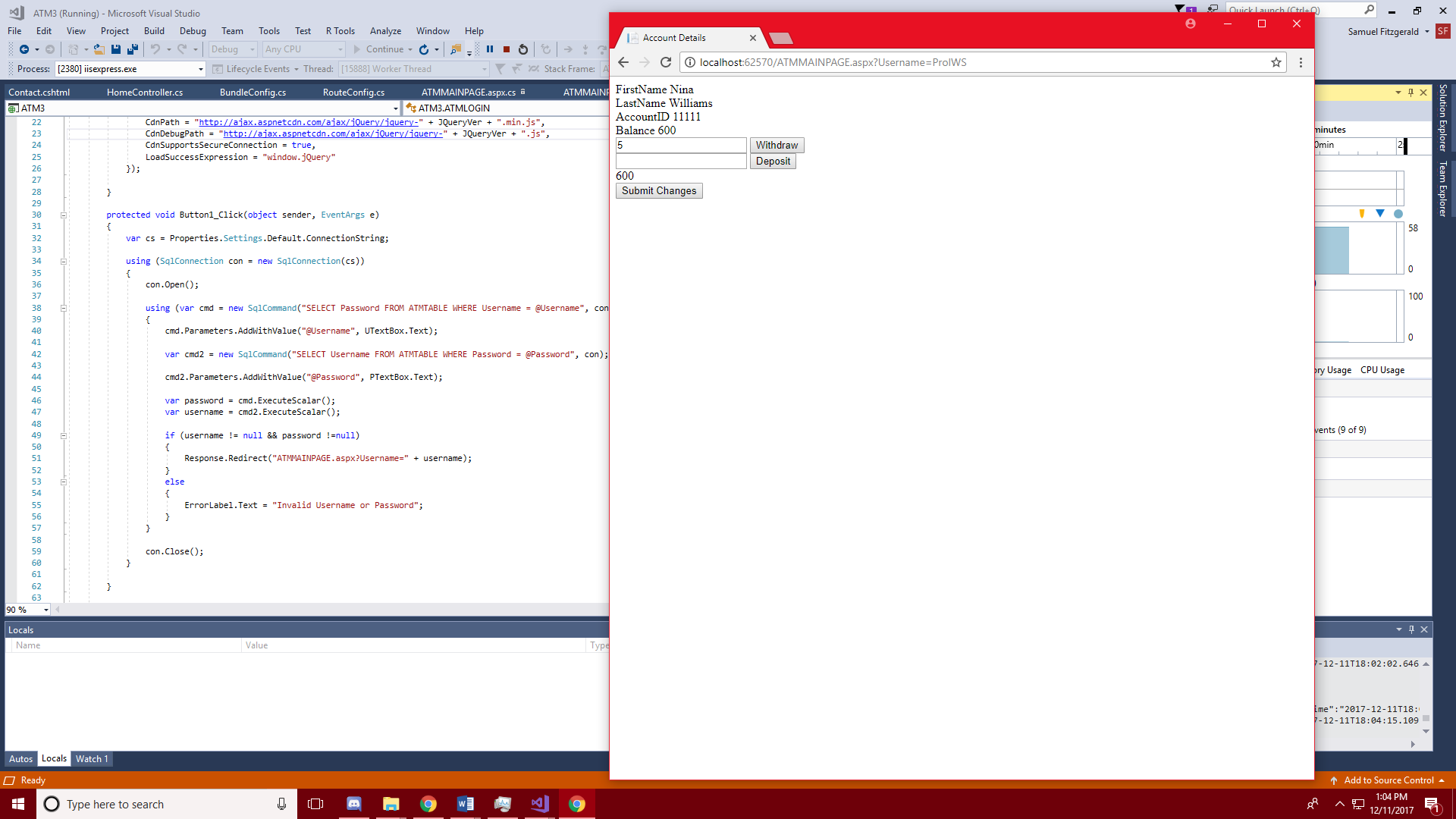
Midterm Using ASP.NET and MVC

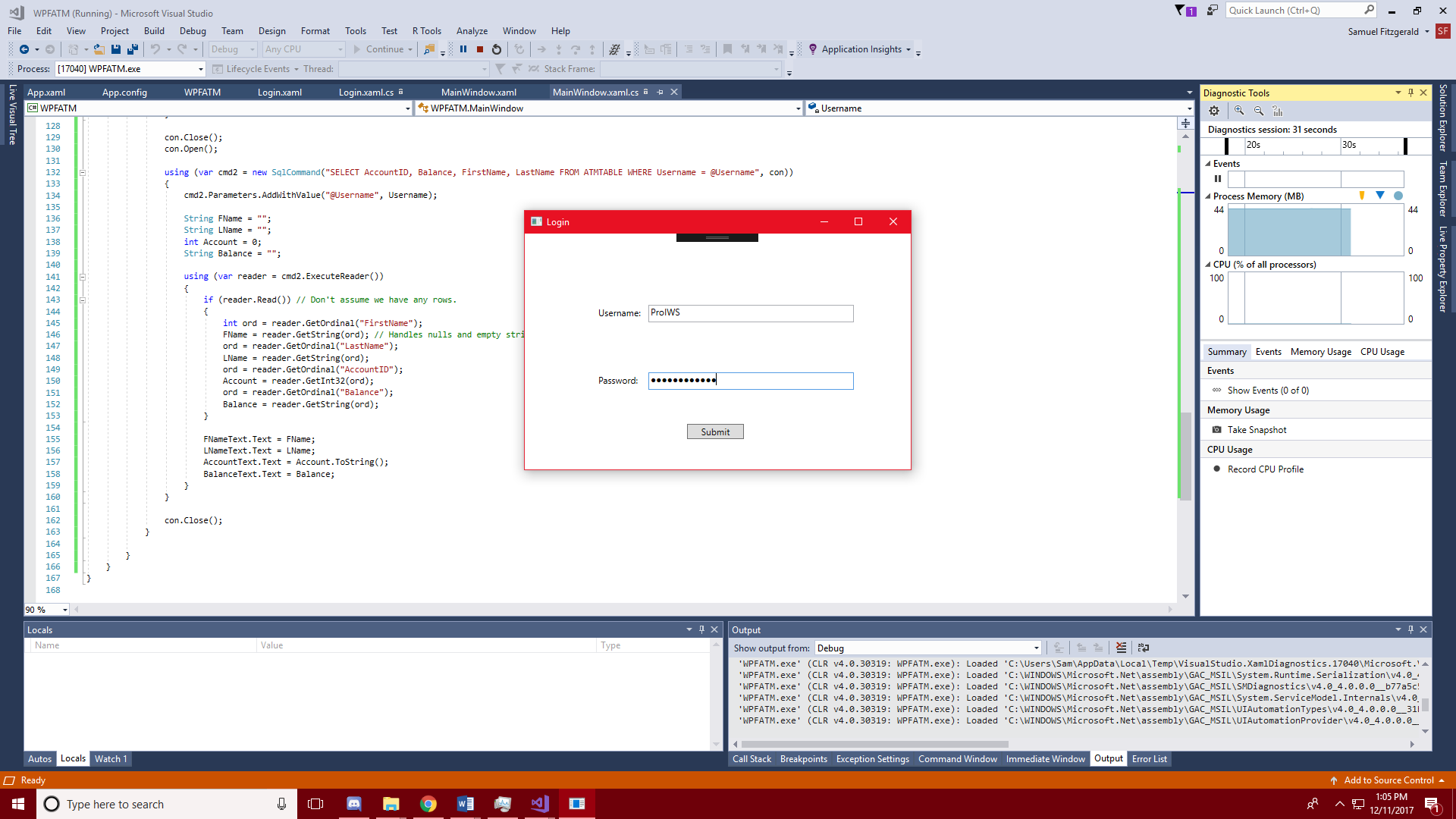


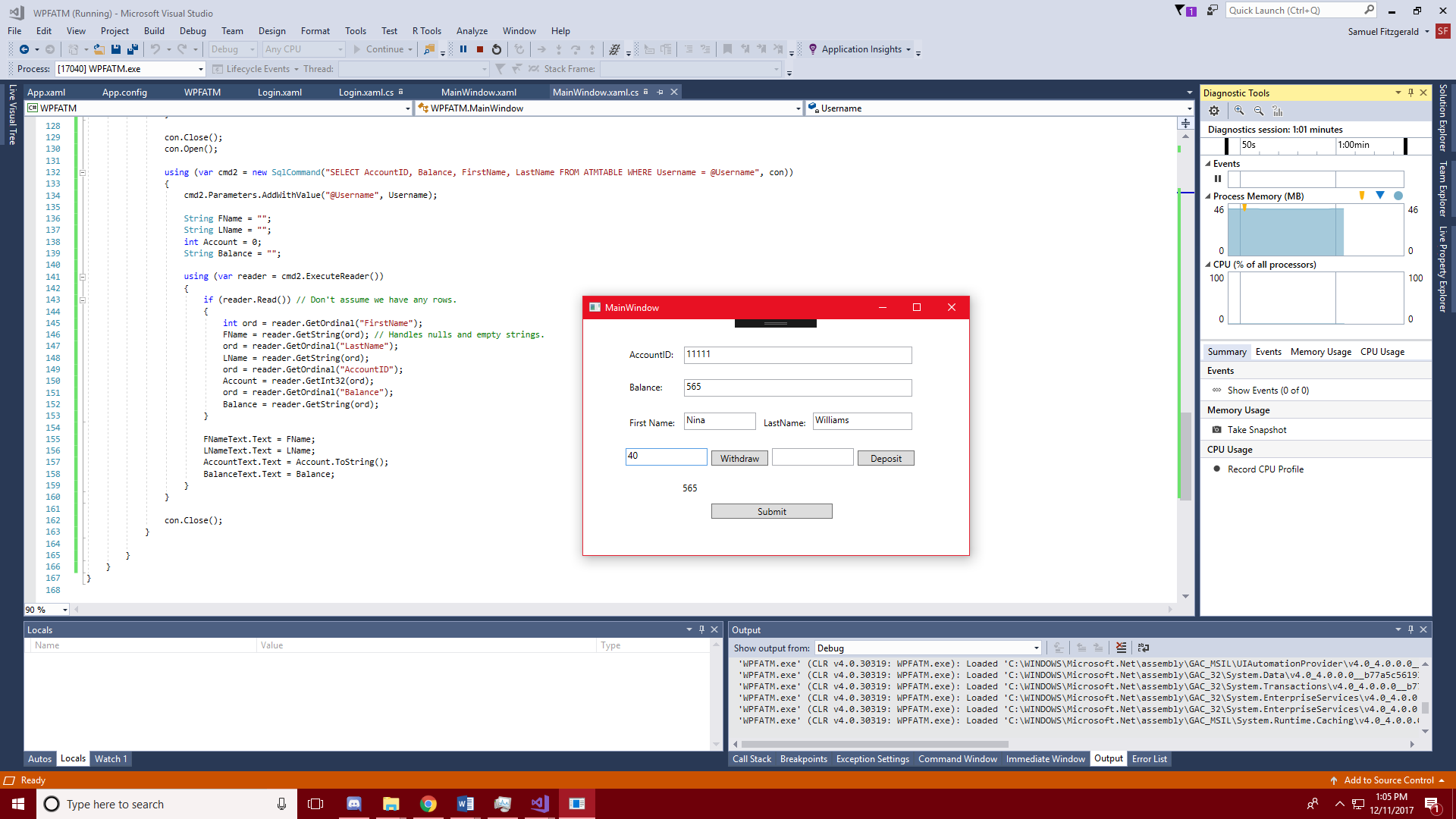


Objective: To create an interface mimicking an ATM interaction that allows a user to sign in with a valid username and password then allows them to view their account balance as well as lets them make deposits and withdraws using MVC.

Methodology: First I began by creating a table to store user data, next I built responsive webpages in asp.net. I validated the user’s credentials by running two SQL queries that checked that the username provided had a corresponding password and vice versa. Next, I pass the username to the main page through the url so that I can run SQL queries to populate the page with the user’s account information. The Withdraw and Deposit buttons compare the amount in the user’s balance and then throw an exception or add/subtract the desired amount and finally the submit button runs an update that provides the table with the new balance before refreshing the user’s account information.

Midterm Using WPF

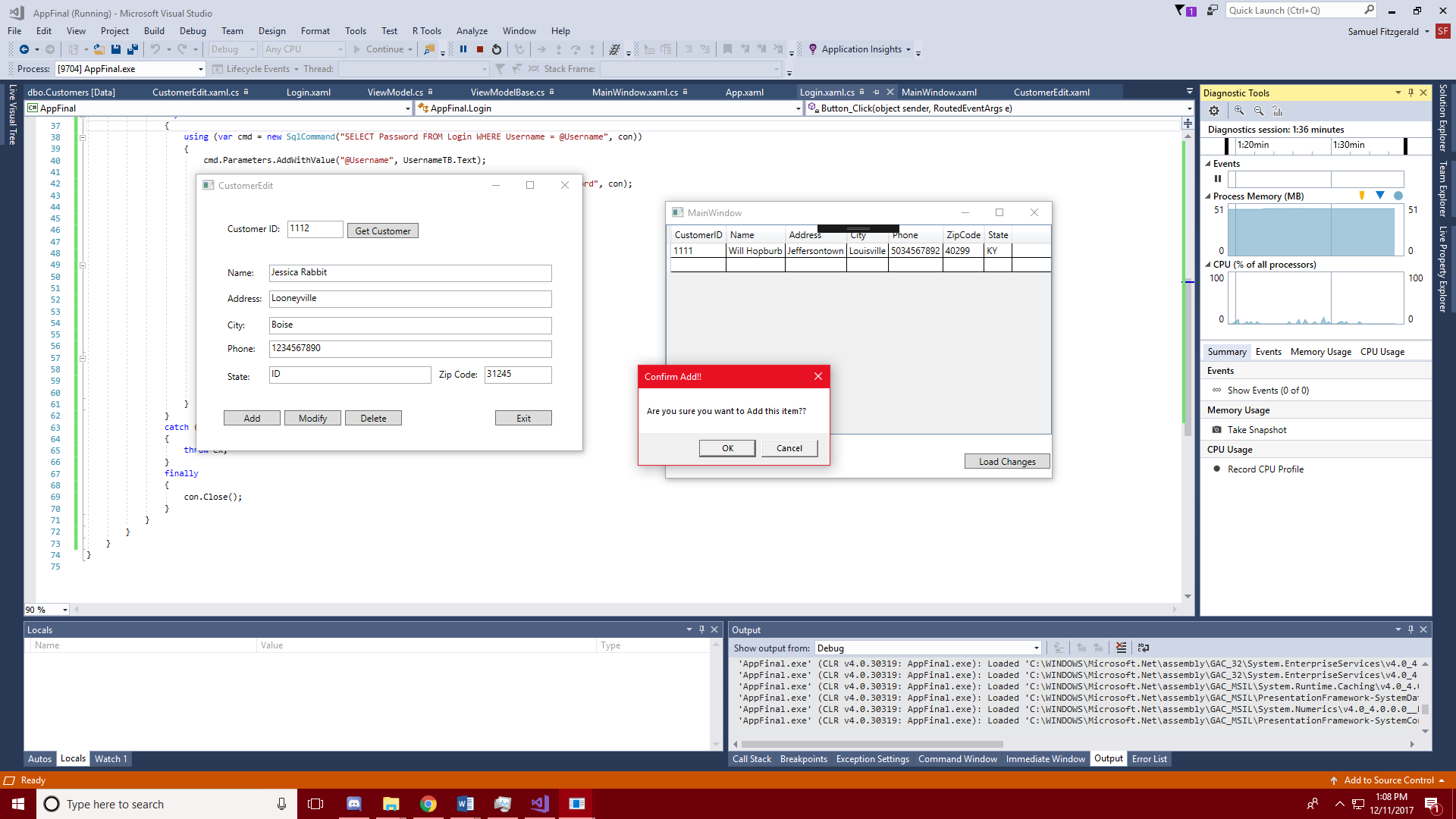
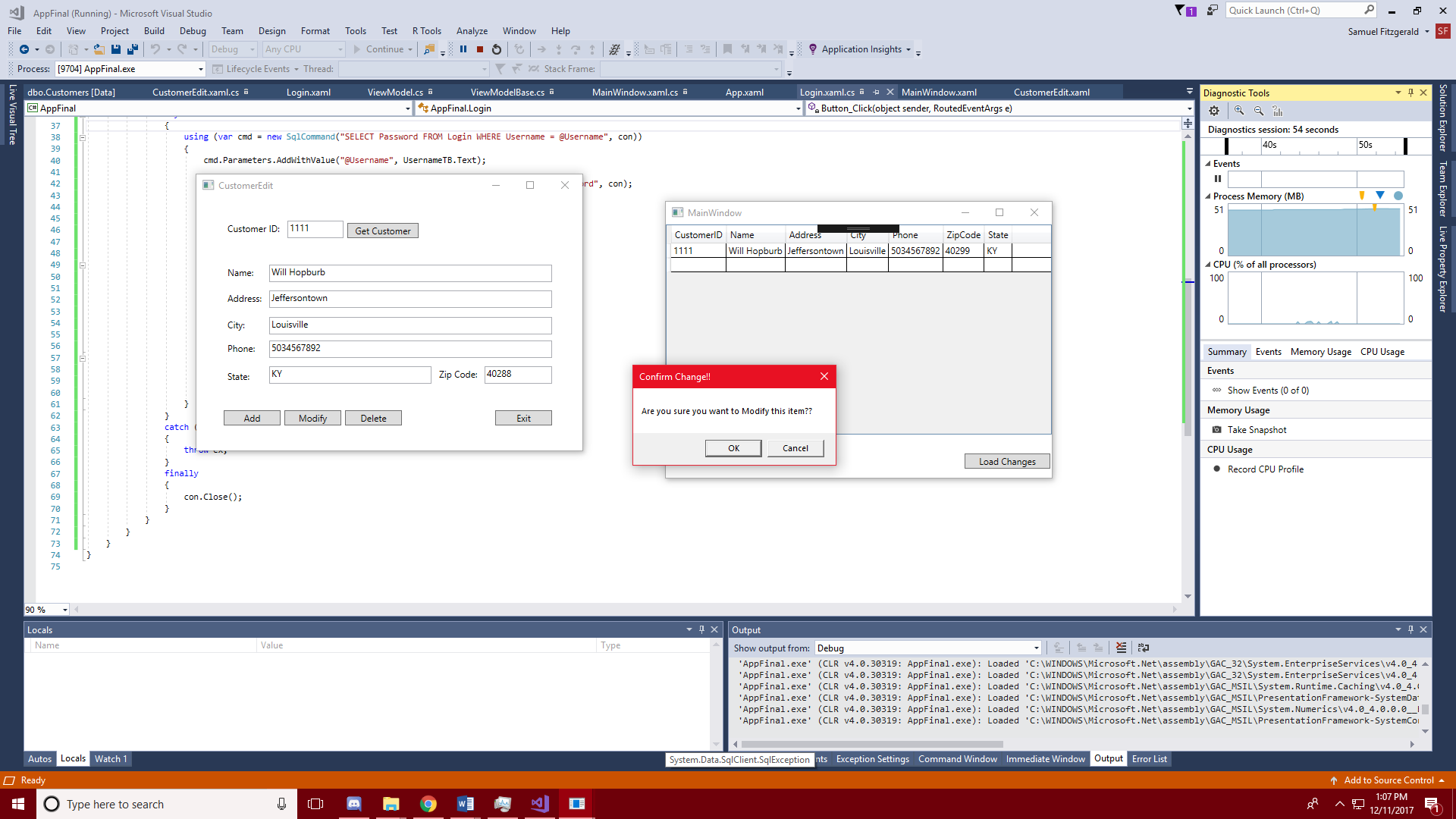
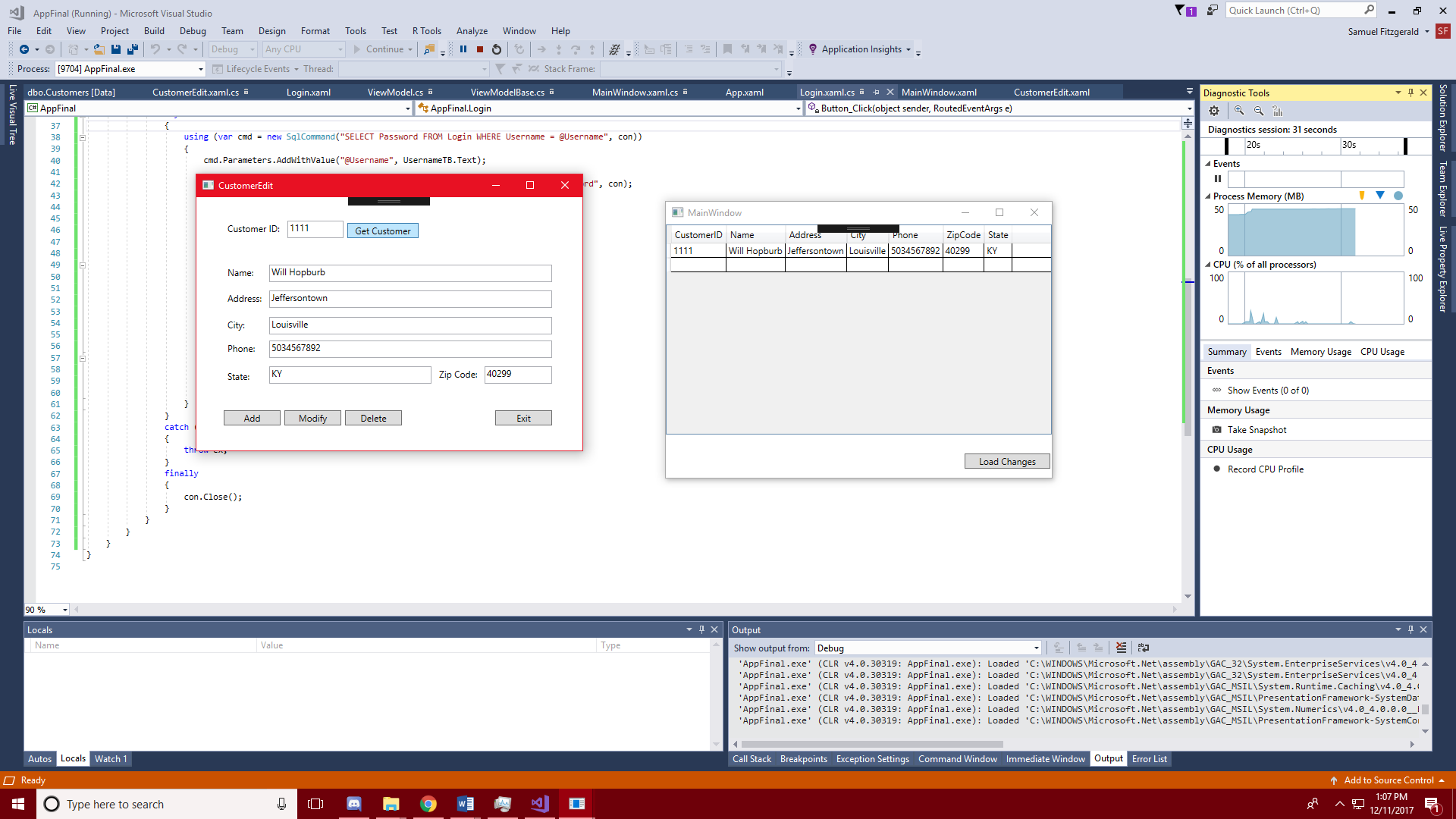
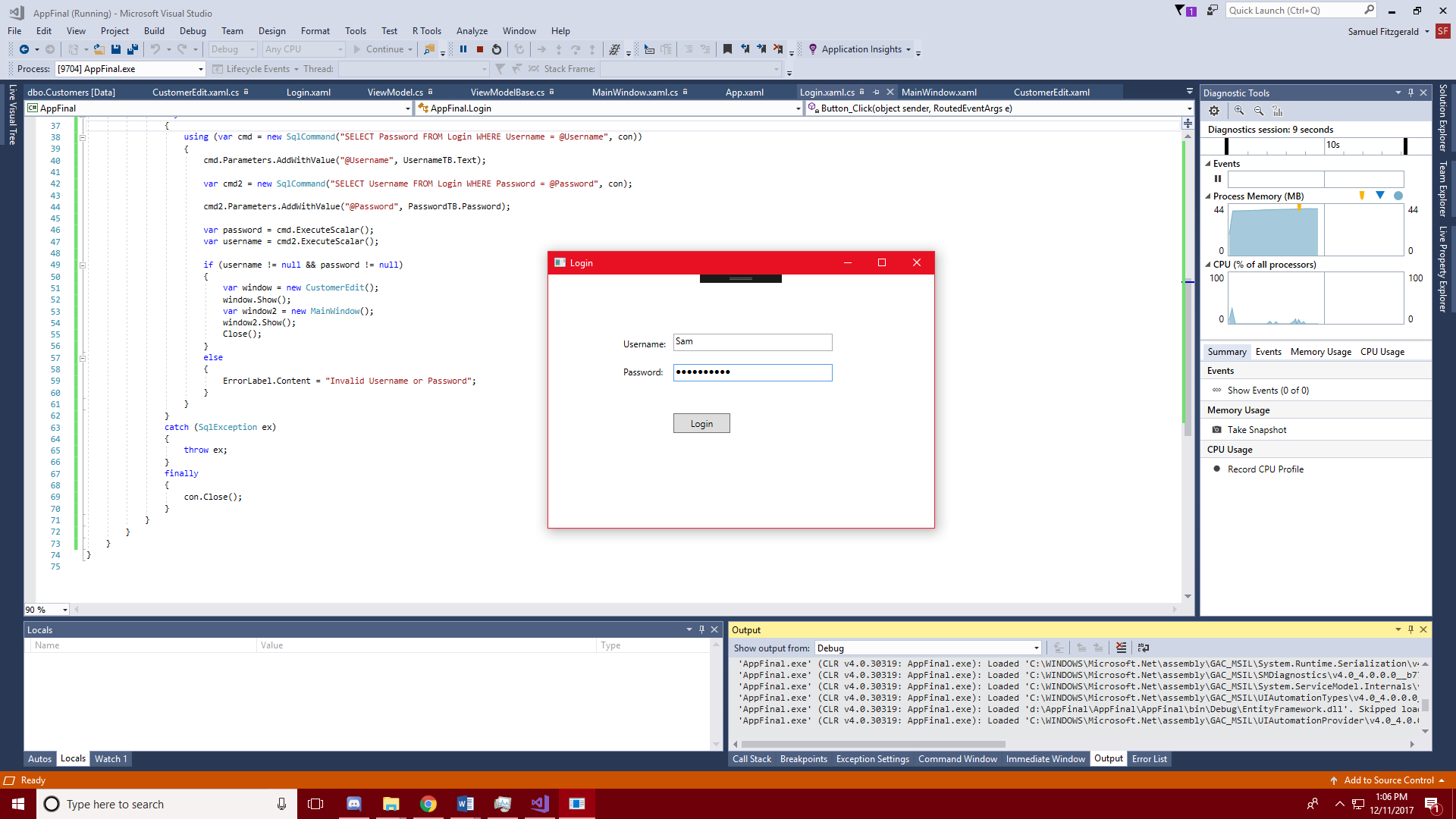


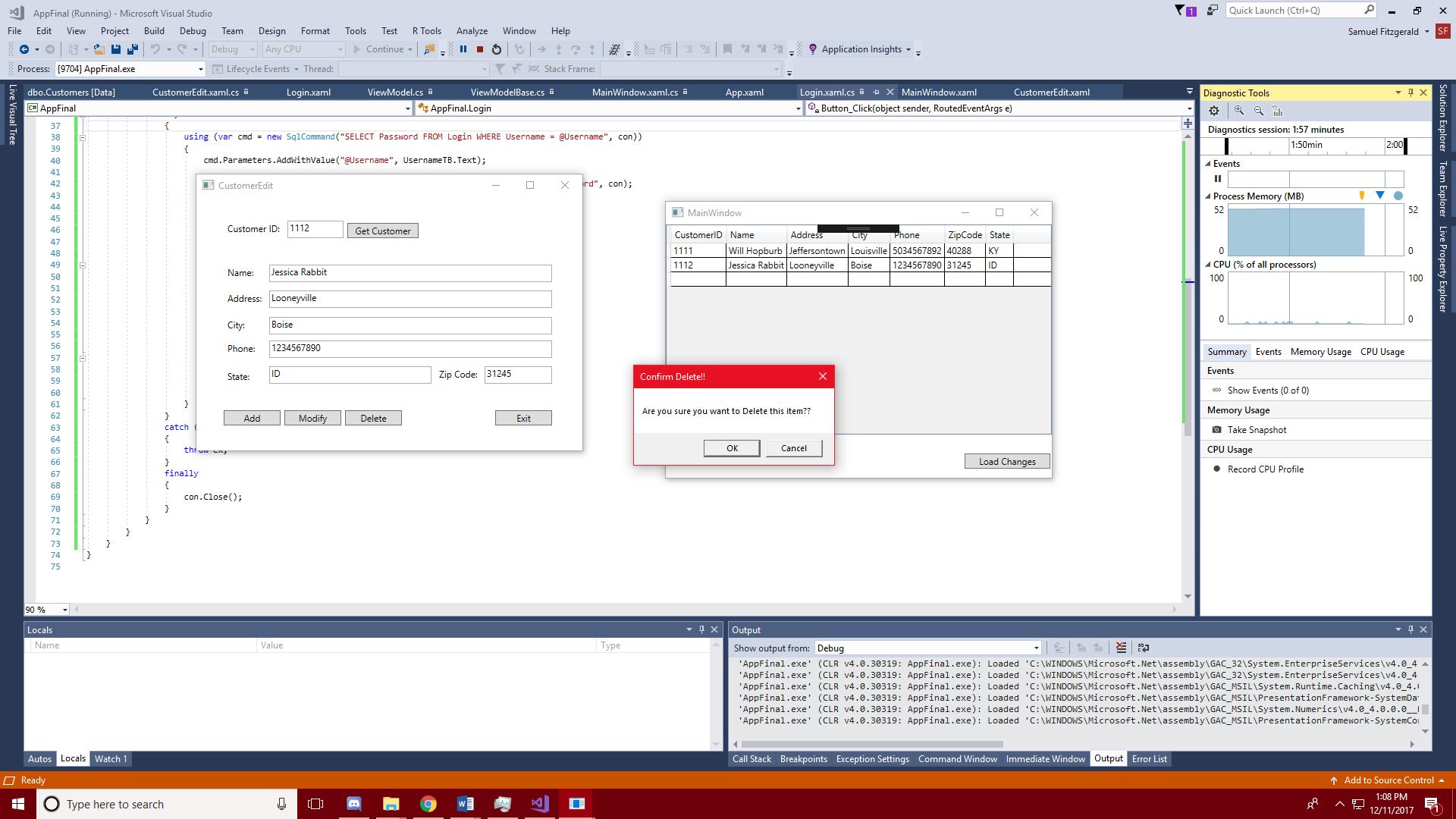


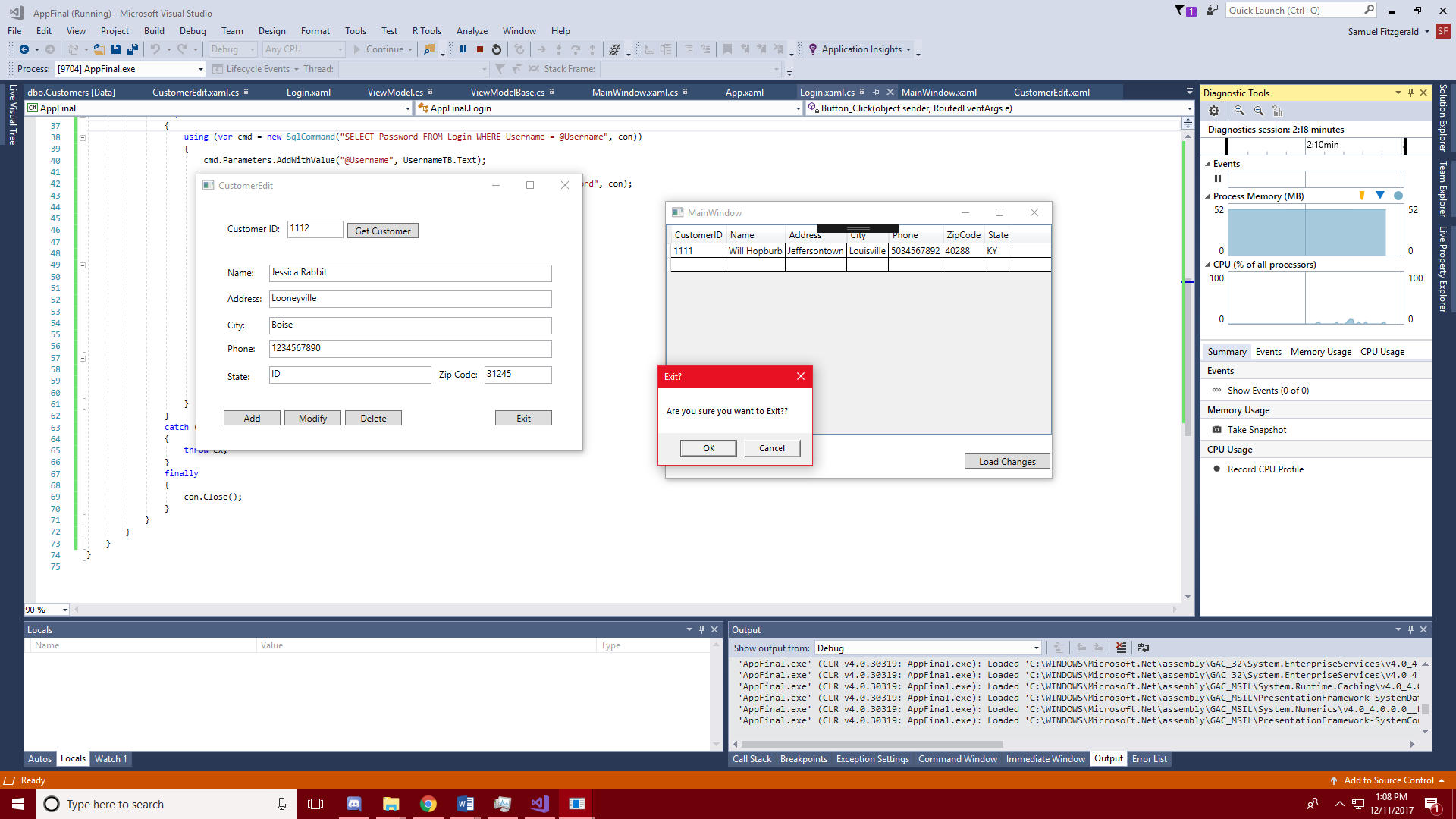
Objective: To create an interface mimicking an ATM interaction that allows a user to sign in with a valid username and password then allows them to view their account balance as well as lets them make deposits and withdraws using WPF.

Methodology: I started this one similar to how I started the MVC version with the login page first. I used the same methods to validate the username and password. To pass the username to the next form I added a parameter on the main method for the window to accept a String. Next the approach was almost identical to the MVC version, creating the form then using the same methods.

Final Using MVVM







Objective: To create a form that would allow users to sign in and then edit a form using a simple control window and then update the form to view the results using mvvm.

Methodology: Much like the midterm I began creating two separate tables, one for credentials and one for the customer database. Next I created the login form and used a very similar method to the ATM projects to validate the user. After I created the forms for the controller and the grid I began working on methods for each button making SQL queries to the Customers table to add, modify or remove data. Then I created a dataGrid linked to the Customers table that put all the entries into a list to be displayed as a grid. Finally I added confirmation prompts after the add, modify, delete, and exit buttons.