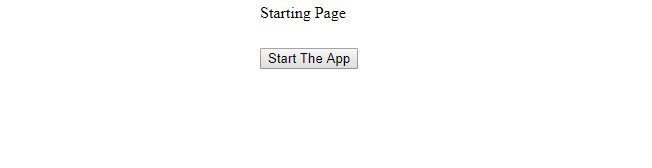
Final Report

# Description of requirements:

On this project I was assigned to make an application that is based on Azure Table Storage. I was tasked on making a front-end web form and another web form where all the functionality will be placed. The back-end web form will have all the basic functionality, such as connecting to the database, data storage, etc. I was also tasked on adding advanced features to the application to increase my grade further.

# Screenshots:

**Main Page**

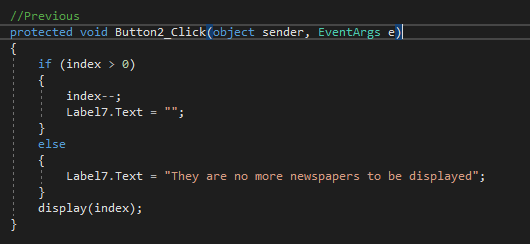
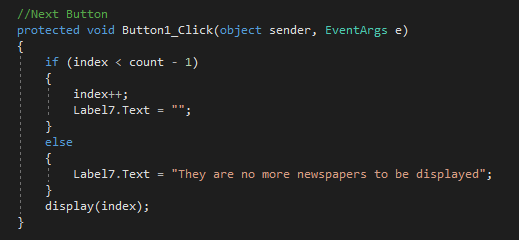


**Application Page**



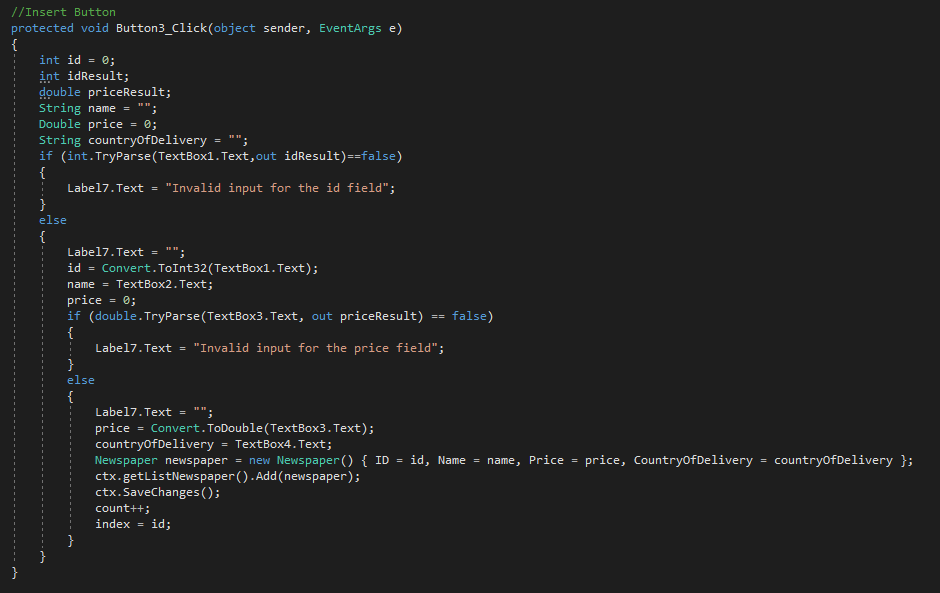
# Examples of Exception Handling:

**Exception Handling of Next & Previous Buttons**



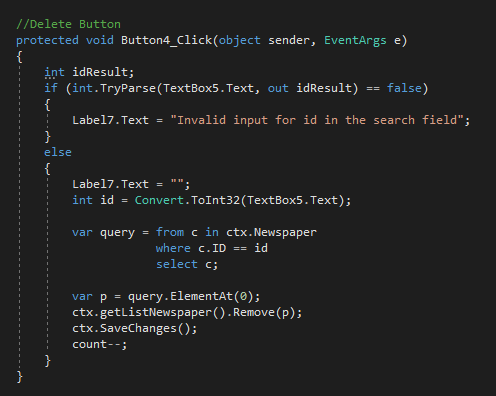
This exception is thrown when you try to get to the next/previous index when that index doesn’t exist.

**Exception Handling of the Insert Button**



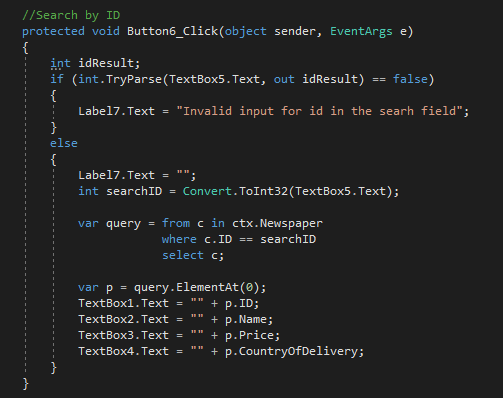
The first exception is thrown when the user inputs a value that isn’t an integer. The second exception is thrown when the user inputs a value that isn’t a double.

**Exception Handling of the Delete Button**



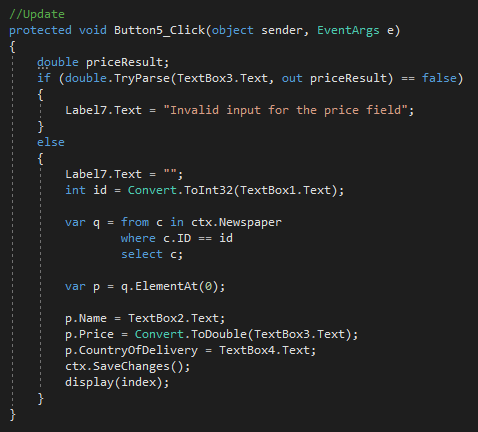
This exception is thrown when the id that the user is searching for is invalid.

**Exception Handling of the Search Button**



This exception is thrown when the id that the user is searching for is invalid

**Exception Handling of the Update Button**



This exception is thrown when the value entered in the price field is not a valid double.

# Table details and samples:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Price | Country of Delivery |
| 0 | The Irish Times | 2.50 | Ireland |
| 1 | Wall Street Journal | 2.00 | USA |
| 2 | The Sun | 1.50 | Ireland |
| 3 | The Daily Telegraph | 1.00 | United Kingdom |
| 4 | Daily Mirror | 1.00 | United Kingdom |
| 5 | Chicago Sun-Times | 2.00 | USA |
| 6 | The Denver Post | 3.00 | USA |

ID – The id of the newspaper (INT).

Name – The name of the newspaper (STRING).

Price – The price of the newspaper (DOUBLE).

Country of Delivery – The country of where the newspapers were delivered (STRING).

# Details of Problems encountered:

When making this application I had problems getting the update, insert, search and delete buttons. I was constantly running the code over and over to find out where the problem with my application was. Turns out for the update, insert and delete, I got the queries that would search for the id that was being searched, updated or delete were wrong. I had coded the conditions for the select query wrong. Previously I had the code written like this: c.ID.Equals(searchID). So, then I fixed it so that it would look like this: c.ID == searchID. The “Equals” command is used when looking for strings that matched not integers or doubles.

Other than that problem, I didn’t run into anymore problems because I was fine making the code print out a string when the inputs for certain fields were wrong.