John Cosentino

Nick Raynin

CSC 330

Project 2: Newspaper Delivery System

Test Plan

This program includes a database (with four tables), a gui with four forms, and multiple media through which we must edit. Therefore, a lot of troubleshooting will be involved.

For any textboxes that take in a numeric input, we must be certain to create an exception handle for format conversion. We cannot be certain that people will type in a number everytime. Assuming that someone will mistakenly enter an alphabetical character, we will create a FormatException. This will assure that accidental keystrokes will not crash the program. In the event that a string is entered, we will display the appropriate MessageBox. This will be done for the textbox that updates the IDs within carriers, and for the billing textbox. The IDs are all only six digits in length, therefore, we must limit the input to a max length of **six**. Since the IDs are of type Int32, we will also create a catch for OverflowException. This is not completely necessary due to the input limit, but it is a safety measure.

For any textboxes that edit the database, a MySqlException catch will be initiated. This will notify the user that the input is incorrect, while highlighting the offending row. The IDs will not be edited within the DataGridView, instead, they will be edited through a special textbox within Carriers. This textbox is always hidden until Carriers is selected in Form3. We will type in numeric characters, string, and SQL code into the textboxes to ensure that all tests pass.

For the login screen, we must hide the password field characters. Any input will turn its characters into \* characters. When logging in, the two textboxes’ inputs will be assigned to a username and password variable, respectively. Both will be compared to the SQL query results, and once **both** fields match, will the user be logged in. Once logged in, Form1 will hide and Form3 will display. If the user cannot login, a MessageBox will display. If the user tries to register a new user, the username will be checked against all of the usernames within the database. It will only register if that username is not taken.

In the SQL code itself, we must not concatenate strings to build the SQL code. This would open up the program to SQL injections. Therefore, there will be names used in place of the input strings. For example, @name would represent a variable name, and @name would be replaced in the SQL query. One such example would be:

cmd.Parameters.Add("@idH", MySqlDbType.Int32, 6, "idhouseholds");

The @idH will be replaced by “idhouseholds”, which will take in a textbox’s Text input. Convenience will be added by utilizing MySqlAdapter. This will give us the option to reuse a lot of code. The adapters have the ability to update, add, delete, etc. code using SQL queries. At the same time, redundancies will be avoided because of the unique IDs as SQL primary keys.