

Warehouse Management System

Students: Giovanni Hasbun-Prinstein, Juliana Pranckevicius, Megan Wee Tom

Mentor: Aydin Danaci, *Ukay Delivery*

Instructor: Masoud Sadjadi, *Florida International University*

CONNECTING TO DATABASE

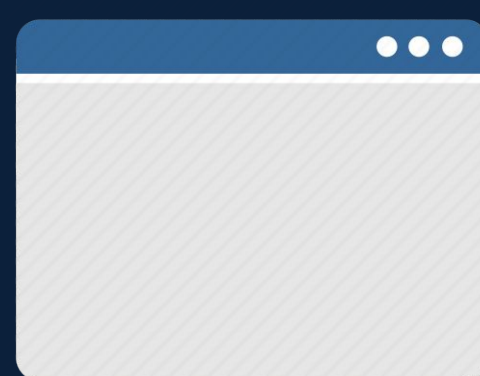
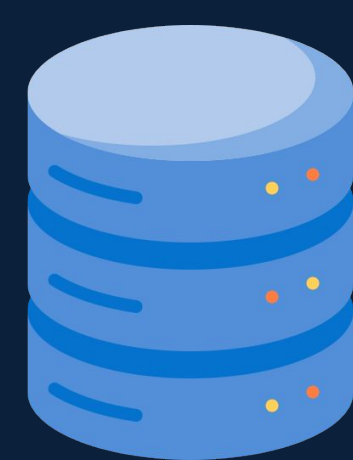
Custom Package:
Microsoft.Data.SqlClient;

```
private string connectionString = @"connectivity snippet";  
private SqlConnection connection;  
private DataTable dataTable;
```

SEARCH FUNCTIONS

```
try  
{  
    if (dataTable != null)  
    {  
        string searchTerm = textBoxSearch.Text.Trim();  
        if (!string.IsNullOrEmpty(searchTerm))  
        {  
            DataTable filteredTable = dataTable.Clone(); // Create a new table to hold the filtered data  
  
            foreach (DataRow row in dataTable.Rows)  
            {  
                foreach (DataColumn col in dataTable.Columns)  
                {  
                    // Check each cell for the search term  
                    if (row[col.ColumnName].ToString().IndexOf(searchTerm, StringComparison.OrdinalIgnoreCase) >= 0)  
                    {  
                        filteredTable.ImportRow(row); // Add the entire row if a match is found  
                        break; // Once a match is found in this row, move to the next row  
                    }  
                }  
            }  
            ...  
        }  
    }  
}
```

RESOURCES



An integral part of a database is retrieving information from the database to display in the front-end portion of the application. This code snippet creates a private connection to a local SQL database and opens the corresponding table when a user changes the category they are viewing. Without a connection, the application cannot retrieve or store data into the database.

CURRENT FUNCTIONS

- Basic level of security where the program verifies a matching username and password.
- Recover forgotten passwords. (No email API implementation as of yet)
- Generates quotes that can be later turned into invoices.
- Stores user inputs into an SQL database and displays the information in the application's UI.
- Users can delete rows upon highlighting and pressing a button.
- Dynamically changes information depending on which category the user is viewing.
- Can mark orders as shipped.
- Search for specific entries.

SCREENSHOTS

For the invoice the algorithms used were to take from a highlighted row of a 'DataGridView' tool from the 'invoiceManager' form and generate the fields - this form can have its code directly translated to creating quotes, purchase orders, sales order. The back-end algorithm uses a basic injection from a database connection string and corresponding commands in MSSQL.

This was a key part of our project as the scope was to create a system that can manage information through insertion, deleting, and documenting information. We achieved the ability to print the invoice of a shipment which ultimately says that a person will receive the item that the service was paid for and that the company will be sending the item over after having confirmed it in our own systems thus completing the Sending and Receiving portion of a Warehouse Management System.

ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by Ukay Delivery. We thank Aydin Danaci for his assistance and mentorship that I/we received throughout the senior design project.