System Design

Existing System—

The current system for shopping is to visit the store manually and from the available product choose the item customer want.

--It is less user-friendly
--User must go to a store and select products
--Description of the product is limited
--It is a time-consuming process
--Not safe during covid-19
--Not in reach od distant users.

Built system (Fashion Toast)—

With our system users can order product via online using credit cards.

Implementation

- Used C# .NET framework + WPF
- Used GitHub for version controlling (https://github.com/navintc/fashionToast-WPF)
- Used Microsoft SQL Server Management Studio 18 for maintaining MySQL Databases
- Used Microsoft Visual Studio 2019 to create WPF
- Adobe Photoshop.

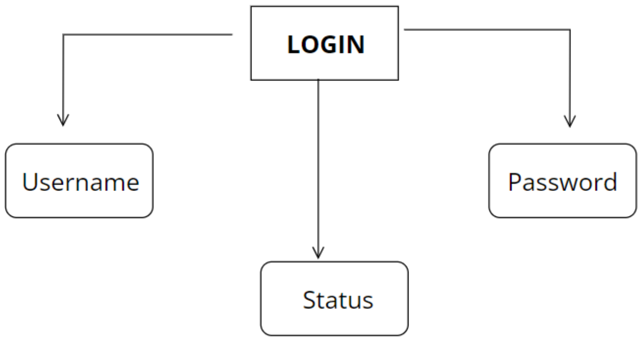
ADMIN

- Admin of the Fashion Toast App will have authorization to access the database anytime by authenticating his/her username and password.
- Only the admin is having the privilege to add, update, delete any item from and to the database.
- ADD Products—Admin can add new items to the existing system with all its details including an image.
- DELETE Products—Admin can delete the items based on the stock of that clothing item.
- UPDATE Products—Admin can update the name, size, color of an existing item in the database.
- Once the items are added, updated, or deleted the admin can logout and return to the login screen again.

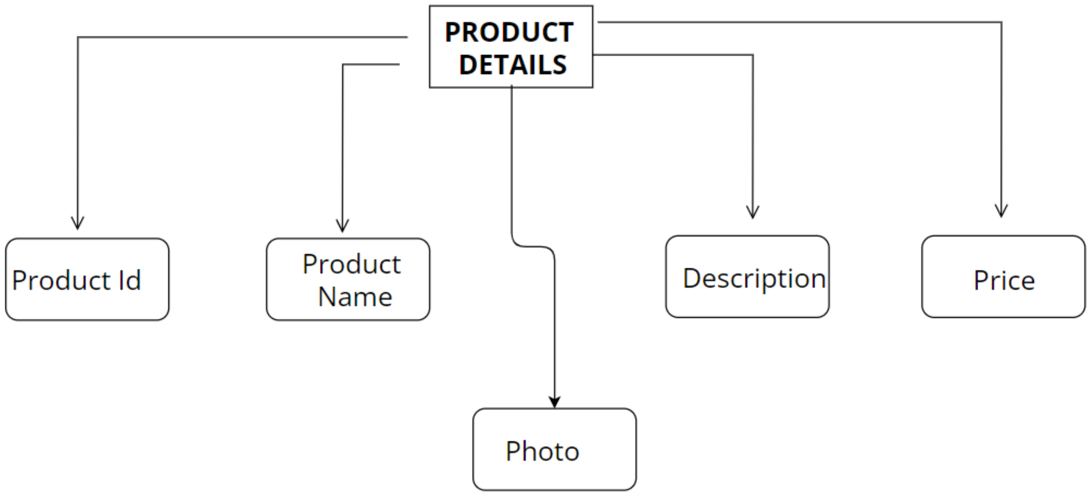
USER

- Registration—A new user will have to register in the system by providing essential details to view the products available in the system.

- Login—A user must login with his/her username and password to the system after registration.



- View Products—user can view the list of products based on their names after a successful login. A detailed description of a particular product with product name, product details, product image, price can be viewed by users.

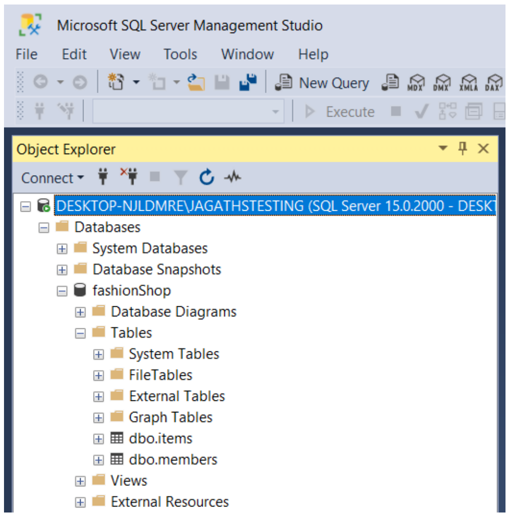


- Add to Cart—The user can add the desired product into his/her cart by clicking the icon on the top right-hand corner, all products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove. After confirming the items in the cart user can submit the cart and get directed to payments option.

Special Breakthroughs:

Account Management

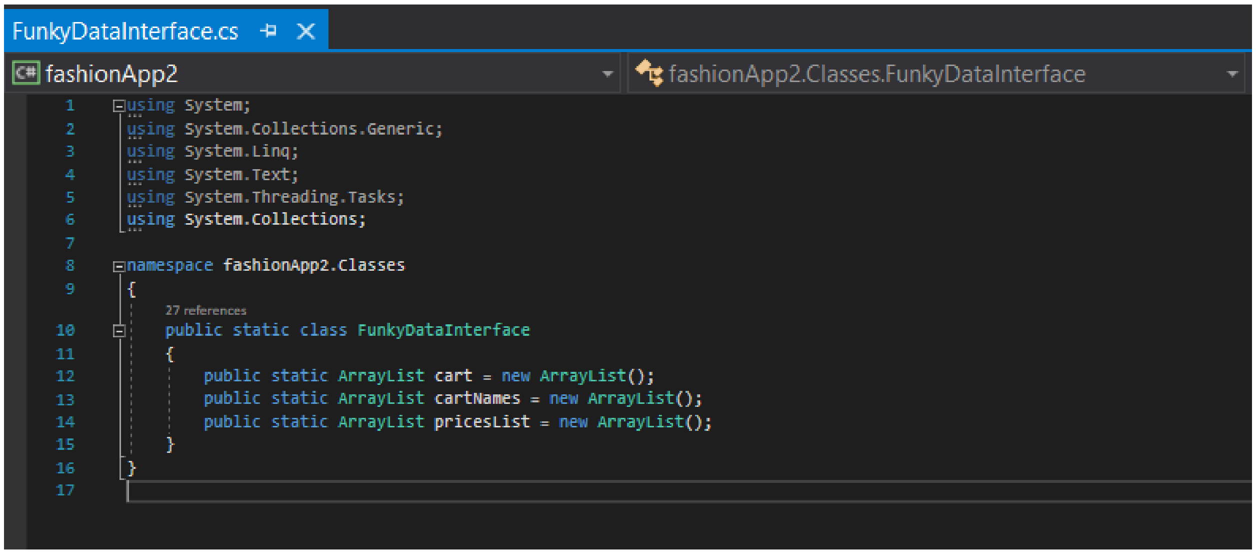
This program uses a single 3rd party database with **multiple tables** to store data of admin/users and products.


Microsoft SQL Server Management Studio

Object Explorer

```
DESKTOP-NJLDMRE\JAGATHSTESTING (SQL Server 15.0.2000 - DESK
  Databases
    System Databases
    Database Snapshots
    fashionShop
      Database Diagrams
      Tables
        System Tables
        FileTables
        External Tables
        Graph Tables
        dbo.items
        dbo.members
      Views
      External Resources
```

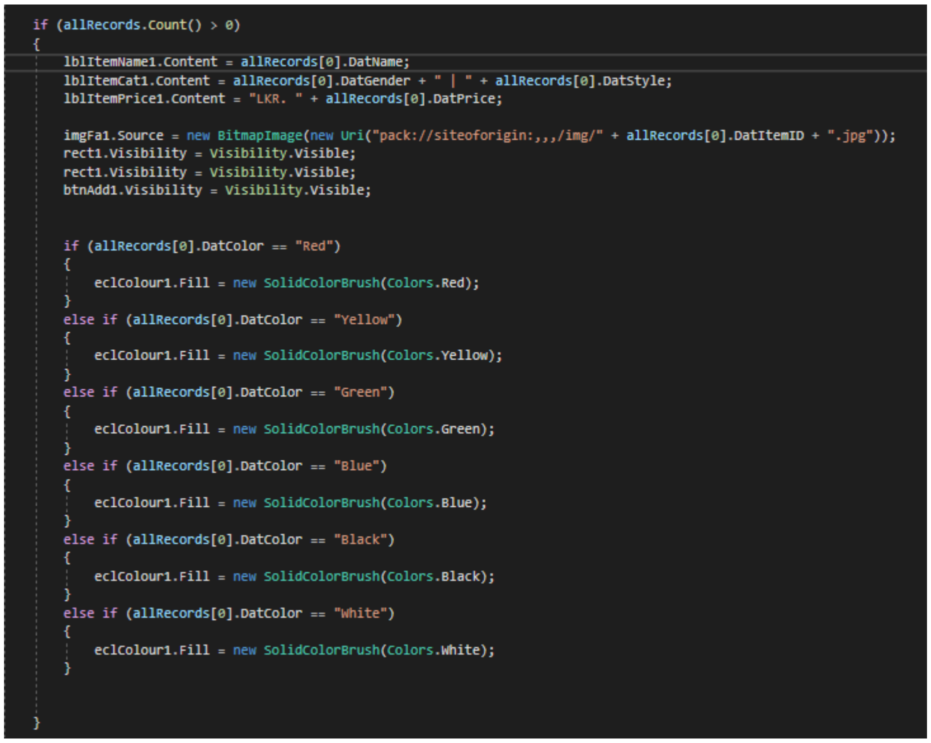
Separate class created as an interface to hold data (Interface classes)


FunkyDataInterface.cs

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Collections;

namespace fashionApp2.Classes
{
    public static class FunkyDataInterface
    {
        public static ArrayList cart = new ArrayList();
        public static ArrayList cartNames = new ArrayList();
        public static ArrayList pricesList = new ArrayList();
    }
}
```

A separate class, `FunkyDataInterface.cs` is created to hold data when moving between windows. By using Unlimited **ArrayList objects** we were able create the necessary storing capability for the Cart.

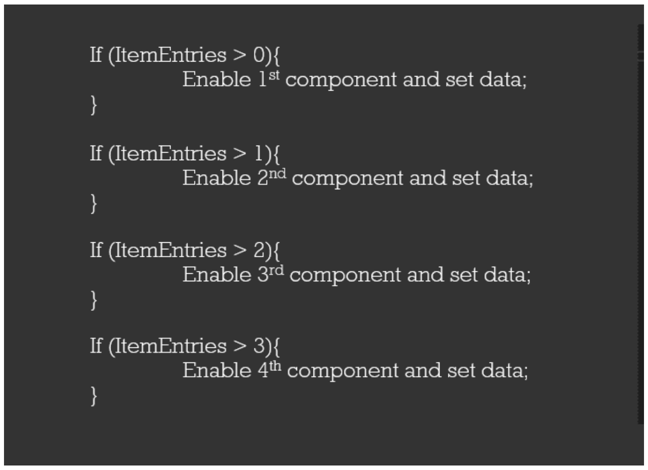
Auto-Generation of components



```
if (allRecords.Count() > 0)
{
    lblItemName1.Content = allRecords[0].DatName;
    lblItemCat1.Content = allRecords[0].DatGender + " | " + allRecords[0].DatStyle;
    lblItemPrice1.Content = "LKR. " + allRecords[0].DatPrice;

    imgFa1.Source = new BitmapImage(new Uri("pack://siteoforigin:,,,/img/" + allRecords[0].DatItemID + ".jpg"));
    rect1.Visibility = Visibility.Visible;
    rect1.Visibility = Visibility.Visible;
    btnAdd1.Visibility = Visibility.Visible;


    if (allRecords[0].DatColor == "Red")
    {
        eclColour1.Fill = new SolidColorBrush(Colors.Red);
    }
    else if (allRecords[0].DatColor == "Yellow")
    {
        eclColour1.Fill = new SolidColorBrush(Colors.Yellow);
    }
    else if (allRecords[0].DatColor == "Green")
    {
        eclColour1.Fill = new SolidColorBrush(Colors.Green);
    }
    else if (allRecords[0].DatColor == "Blue")
    {
        eclColour1.Fill = new SolidColorBrush(Colors.Blue);
    }
    else if (allRecords[0].DatColor == "Black")
    {
        eclColour1.Fill = new SolidColorBrush(Colors.Black);
    }
    else if (allRecords[0].DatColor == "White")
    {
        eclColour1.Fill = new SolidColorBrush(Colors.White);
    }


}
```

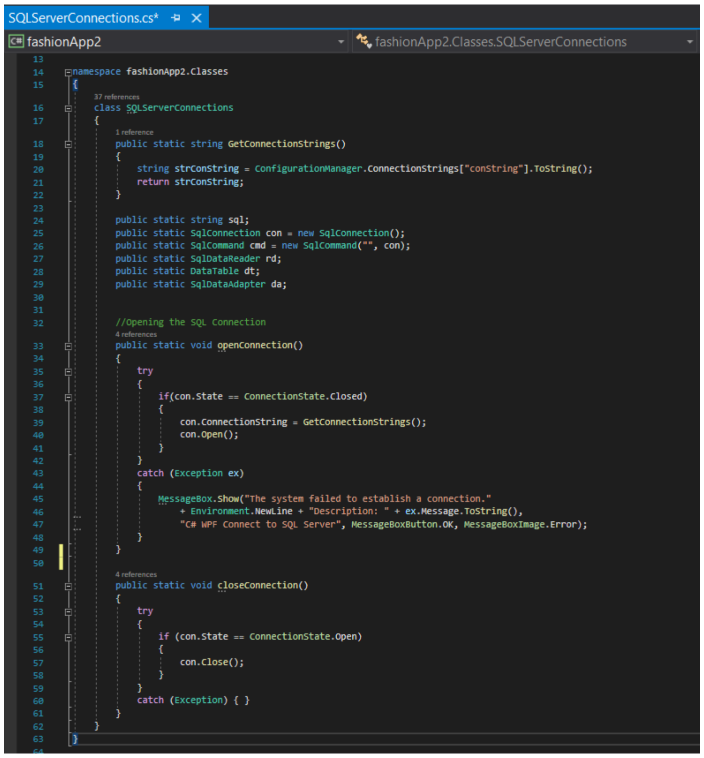
Our software can filter data and showing them in an auto generative matter. This was a trial-and-error experiment we did that, gave us successful results.

Pseudo Code (considering Item entries came from the data base):



```
If (ItemEntries > 0){
            Enable 1st component and set data;
}

If (ItemEntries > 1){
            Enable 2nd component and set data;
}

If (ItemEntries > 2){
            Enable 3rd component and set data;
}

If (ItemEntries > 3){
            Enable 4th component and set data;
}
```

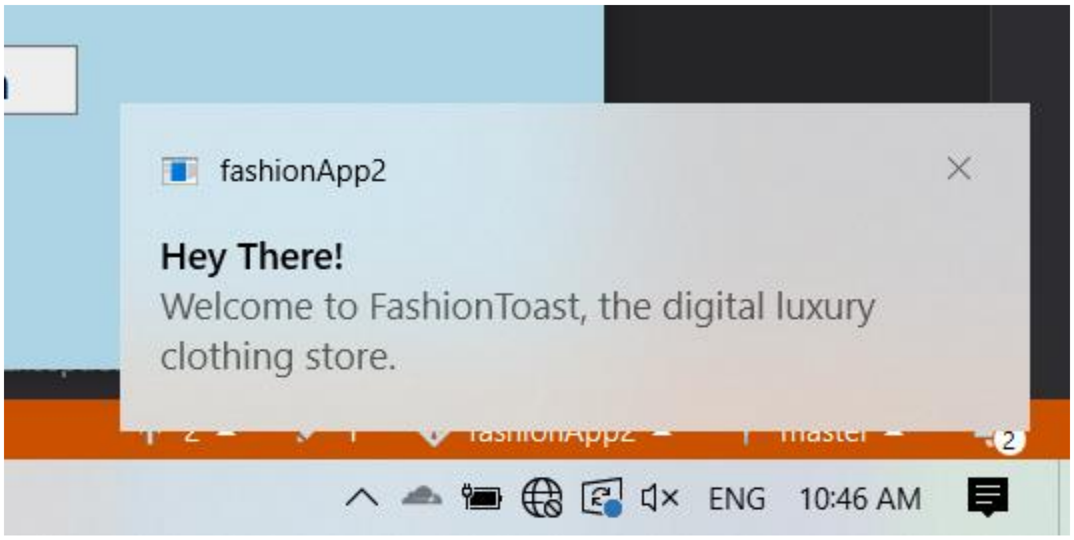
SQL Server Management



```
namespace fashionApp2.Classes
{
    37 references
    class SQLServerConnections
    {
        1 reference
        public static string GetConnectionStrings()
        {
            string strConString = ConfigurationManager.ConnectionStrings["conString"].ToString();
            return strConString;
        }

        public static string sql;
        public static SqlConnection con = new SqlConnection();
        public static SqlCommand cmd = new SqlCommand("", con);
        public static SqlDataReader rd;
        public static DataTable dt;
        public static SqlDataAdapter da;


        //Opening the SQL Connection
        4 references
        public static void openConnection()
        {
            try
            {
                if(con.State == ConnectionState.Closed)
                {
                    con.ConnectionString = GetConnectionStrings();
                    con.Open();
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("The system failed to establish a connection."
                    + Environment.NewLine + "Description: " + ex.Message.ToString(),
                    "C# WPF Connect to SQL Server", MessageBoxButton.OK, MessageBoxImage.Error);
            }
        }

        4 references
        public static void closeConnection()
        {
            try
            {
                if (con.State == ConnectionState.Open)
                {
                    con.Close();
                }
            }
            catch (Exception) { }
        }
    }
}
```

Separate class created for SQL Server Connection Management (Select, Add, Delete, Update) making the software more effective be more abstractive.

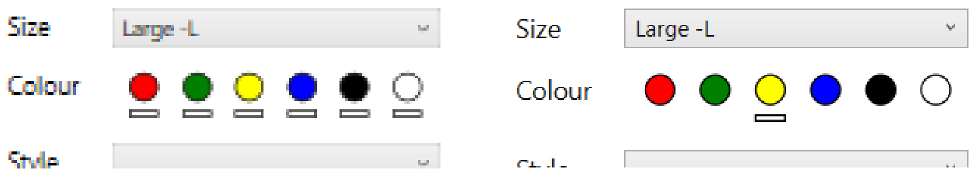
Windows Toast (Push Notifications)



Hey There!
Welcome to FashionToast, the digital luxury clothing store.

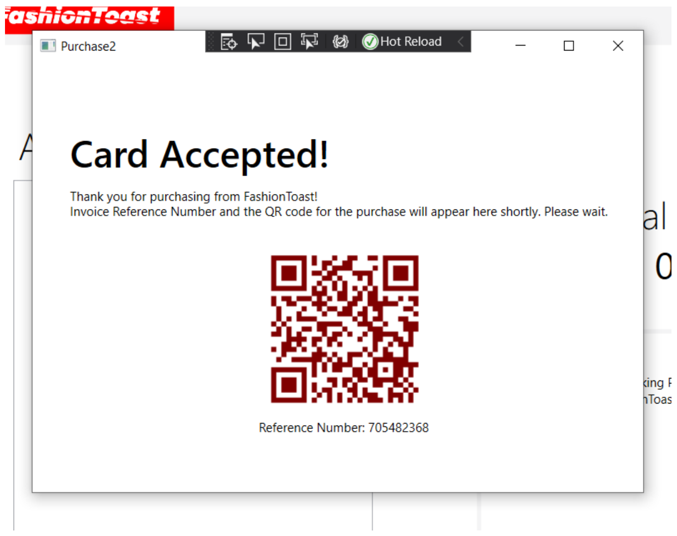
Fashion toast uses the `Microsoft.Toolkit.Uwp.Notifications;` nuget package to send windows push notifications to the user.

Dynamically changing UI elements to give a much better UX



FashionToast is capable of changing UI elements with inline editing of the xml content. This helps the user to understand how to use the program better.

QR Generation API (qrtag.net)



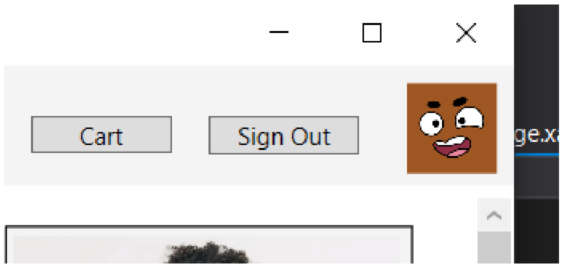
Card Accepted!

Thank you for purchasing from FashionToast!
Invoice Reference Number and the QR code for the purchase will appear here shortly. Please wait.

Reference Number: 705482368

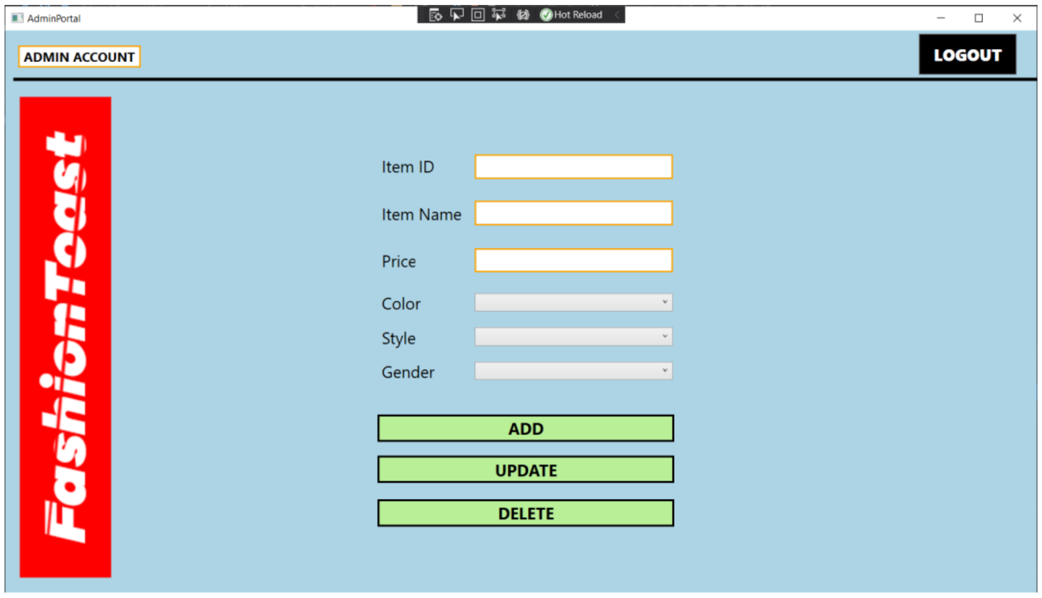
When a user purchases the products, a QR with the reference number will be shown. At the moment this reference number is a random number. This feature is created with qrtag.net API.

Randomly assigning Profile Pictures with an API



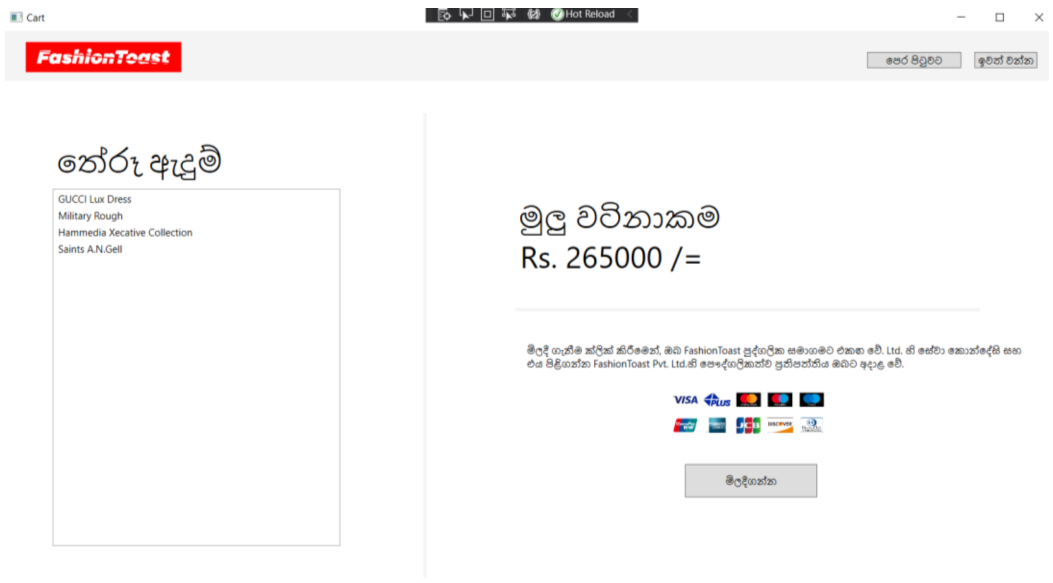
FashionToast is capable of assigning random profile pictures to its users with the help of avatars.dicebear.com API. But, since this project only has one user, the profile picture will be changed randomly when the window is initialized.

Fully Fledged CRUD



Fashion Toast can Perform Crud operations. This feature enables the admins to alter its item table.

Change Language to Sinhala



Cart

තෝරූ ඇදුම්

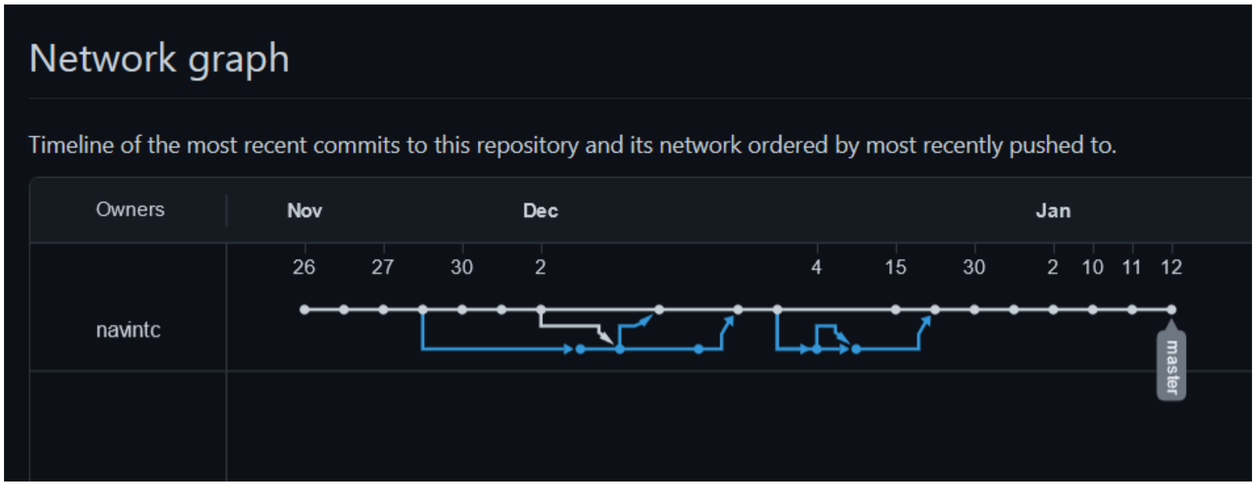
GUCCI Lux Dress
Military Rough
Hammedia Xecative Collection
Saints A.N.Gell

මුලු වටිනාකම
Rs. 265000 /=

මිලදී ගැනීම ක්ලික් කිරීමෙන්, ඔබ FashionToast පුද්ගලික සමාගමට එකඟ වේ. Ltd. හි සේවා කොන්දේසි සහ එය පිළිගන්න FashionToast Pvt. Ltd.හි පෞද්ගලිකත්ව ප්‍රතිපත්තිය ඔබට අදාළ වේ.

Another added feature of this app is changing its language. Users can use the app in both English and Sinhala.

Version Controlling



Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

- 23 Commits. Resolved 3 Merge Conflicts.
- More than 1500 original C# Lines were written and amended.
- More than 500 original XML Lines were written and amended.