PATHOLOGY LAB MANAGEMENT SYSTEM

BY

OMONZEJELE PRECIOUS O.,

OMOWUNMI and

FOLAKE

***SEMESTER*** 4 Project

FACULTY: MR NIYI.

©2017.

Table of Content

* About
* Acknowledgement
* Title of page

**CHAPTER ONE: INTRODUCTION**

1. Problem statement
2. Objective of the project
3. Expected contribution

**CHAPTER TWO**: **DESIGN AND METHODOLOGY**

1. Design and Methodology
2. Database structure

**CHAPTER THREE: IMPLEMENTATION**

1. UI Design
2. Source code

**About**

Pathology Lab Management System was built to tackle some pressing issues in the medical sector. Realising storage of records of tests, patients, doctors, items were major issues, something needed to be done about it. Pathology Lab Management (PLM) system tackles that, with the well-structured and normalised database, it makes sure all records are well stored and related when necessary. PLM System is very user friendly and easy to use with a simple user interface that focuses the relevant information to the end user.

**ACKNOWLEDGEMENT**

My Profound gratitude is expressed to some people and technologies who had supported and contributed to the successful completion of this project.

I thank and appreciate my everlasting Almighty Jehovah God for giving the ideas used in this project. I also want to thank my parents for their support and contributions to this project. And my faculty who gave me clues to finding answers to some confusing questions about this project.

**Critic:** Ope

**Adviser:** Ope

**Technology:** Microsoft, Visual studio, XAML, c#, MySQL, Oracle, PhpMyAdmin, WAMP Server.

Others: Haastrup Joke, Omotayo Akintola, Stack Overflow.

May God continue to be with you.

**CHAPTER ONE**

*INTRODUCTION*

1. Problem Statement:
2. **Life Line** is a pathology lab which performs different types of test. Earlier the lab use to perform selected test only but with the increasing response of the patient and doctor demand they have added new tests. So it becomes difficult to manage the reports of the patients and the details of the various tests. So the management of the lab decided to develop software for the lab management. So the software Pathology Lab MANAGEMENT SYSTEM was developed to automate the lab details. This includes information about all lab items, Doctors, patients, and employees. This system gives a generalized, concise and accurate information regarding billing, purchase details. Stock etc. This system provides any type of enquiry such as patient details, stock, purchase details, Patient details, billing etc.
3. Objective of the project:

* The application will have different tabs like staff registration, patient registration, through which staff should able to register themselves.
* Through patient registration tab page for patient registration will be available. It should also have a tab which will give information of the entire test which is conducted in the lab and the cost of each test.
* Registered staff should able to log in through there username and password.
* Each test, staff and patient should have a unique id.
* There should be an admin login who can update and delete the types of test there cost, staff details, stock detail and the doctors detail who have reference them to this lab.
* Details of the test performed on the patient and the reports of the patient should be maintained i.e. add, update and delete.
* Billing details of the test should be maintained.
* Stock of various things like chemicals, medicine, instruments like syringe and other which are required for the test, cotton etc should be maintained.
* Proper validation should be applied to run the application according to the requirement of the management like if the username or password is not the user should not able to login, user should not able to do any updating or deletion etc.

1. Expected Contribution:

PLM System is expected to make organizing, storing, and sorting of tests, reports, patients, items, doctors’ records easier and efficient.

**CHAPTER TWO**

*DESIGN AND METHODOLOGY*

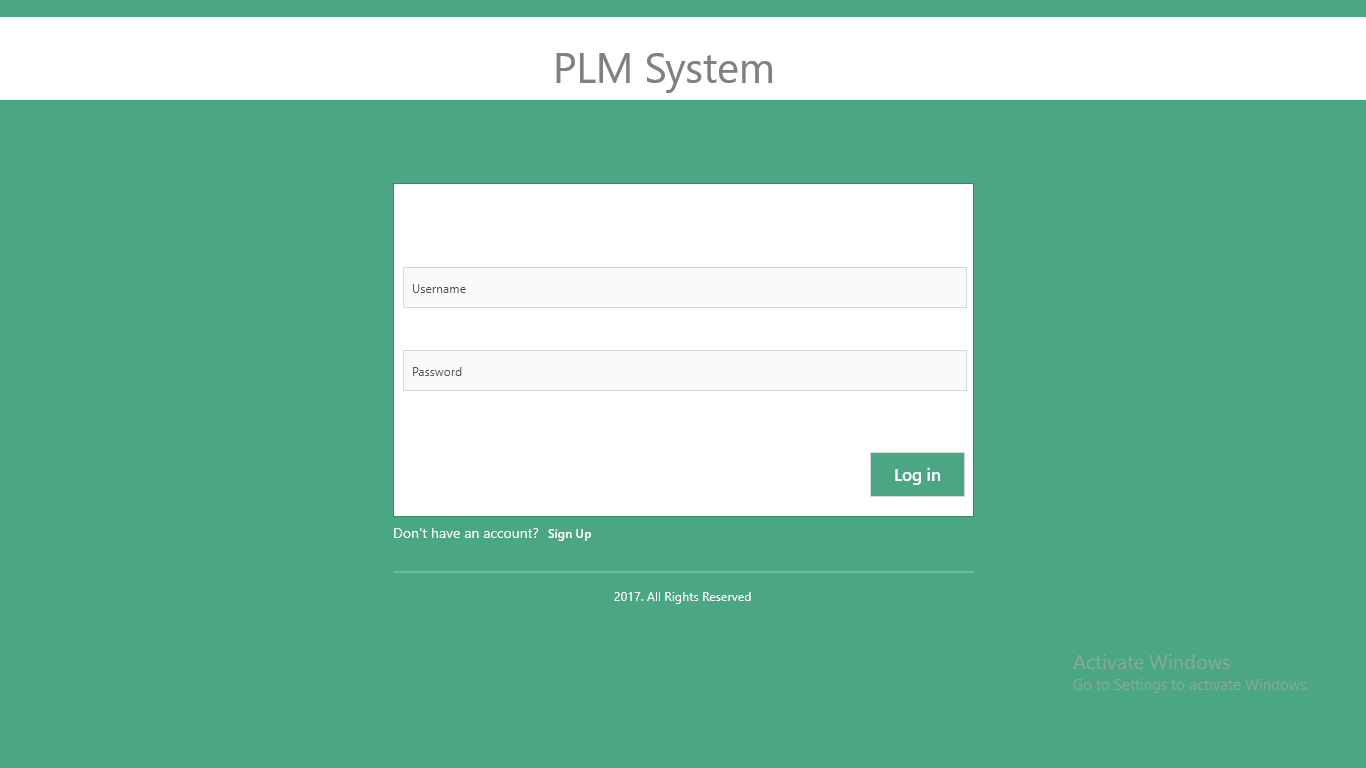
1. Design and Methodology:

**Welcome page**



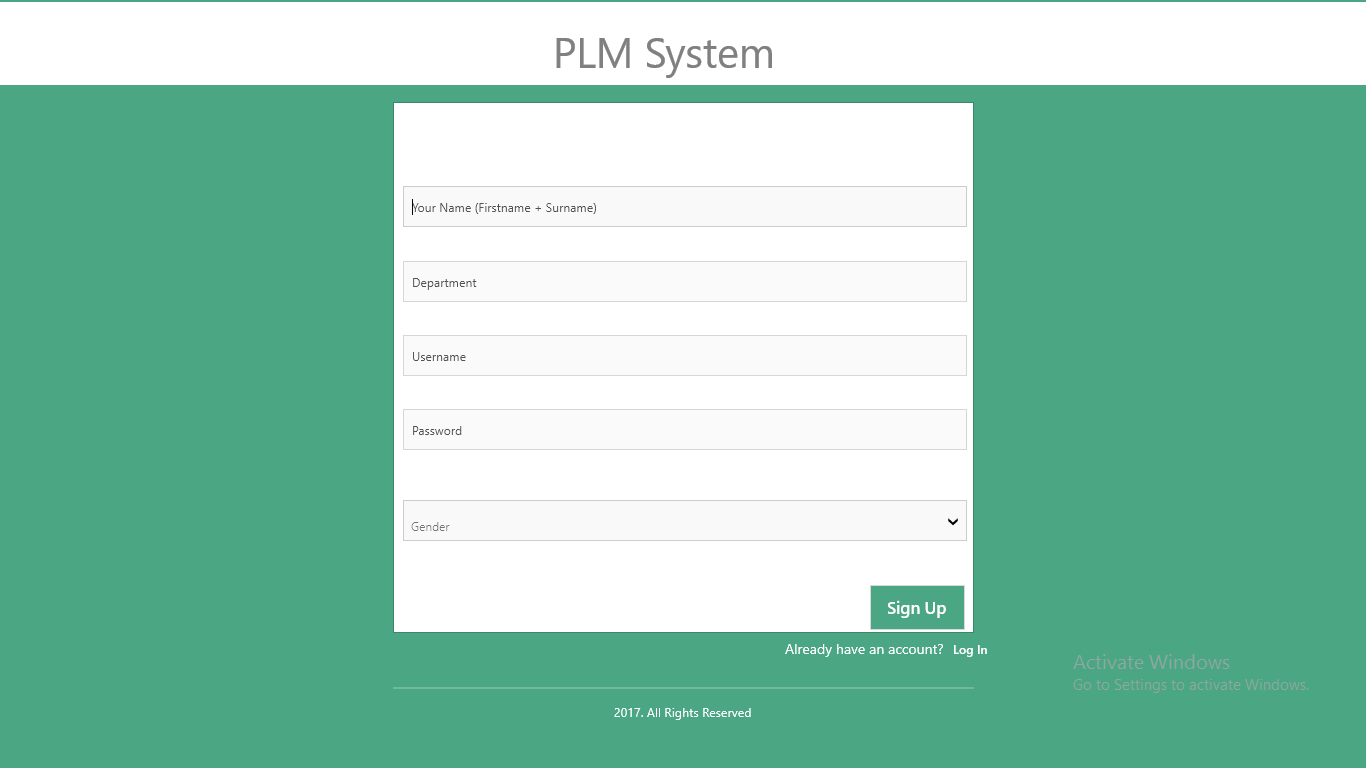
The starting page simply gives you access to either login or sign up.

**Login page**



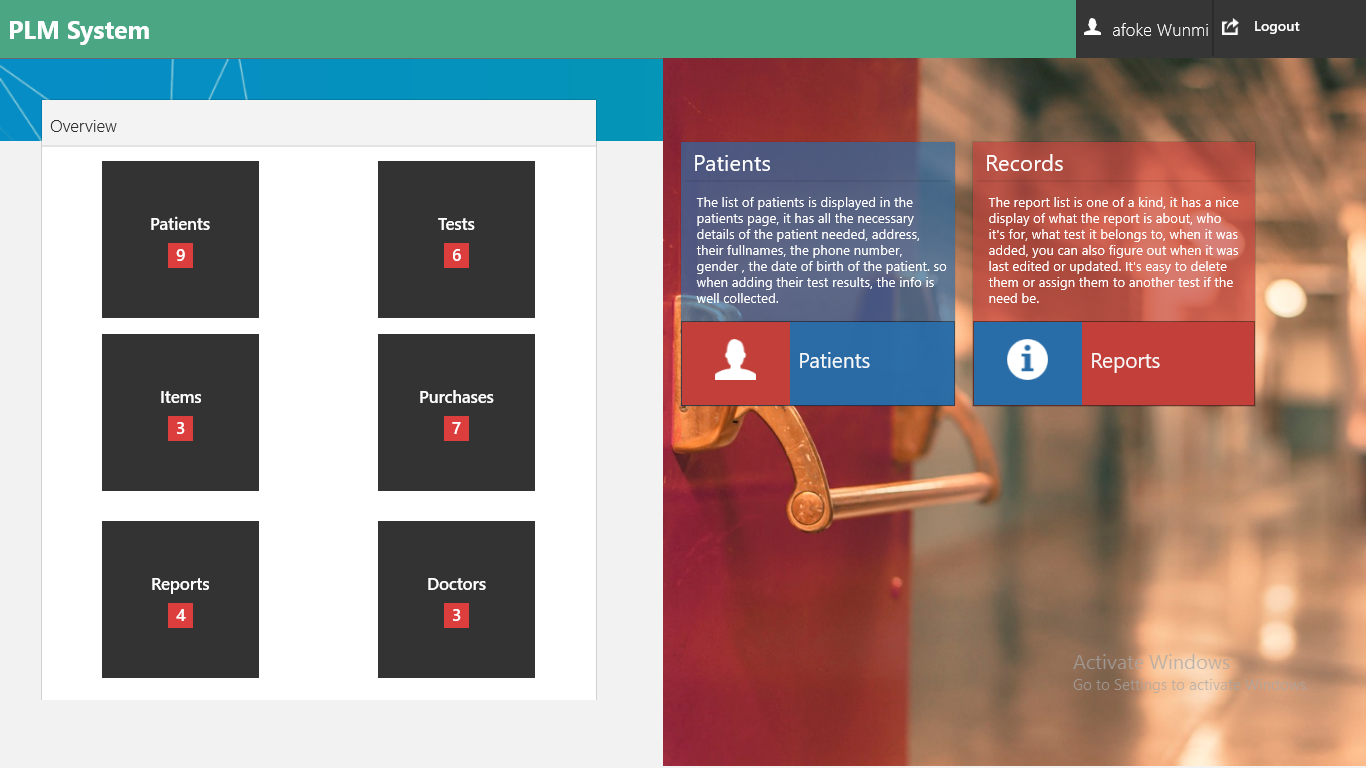
We all know what the login page does. It allows the admin to access their account, with their unique username and password, if not registered, it provides a quick link to the sign up page.

**Sign up page**



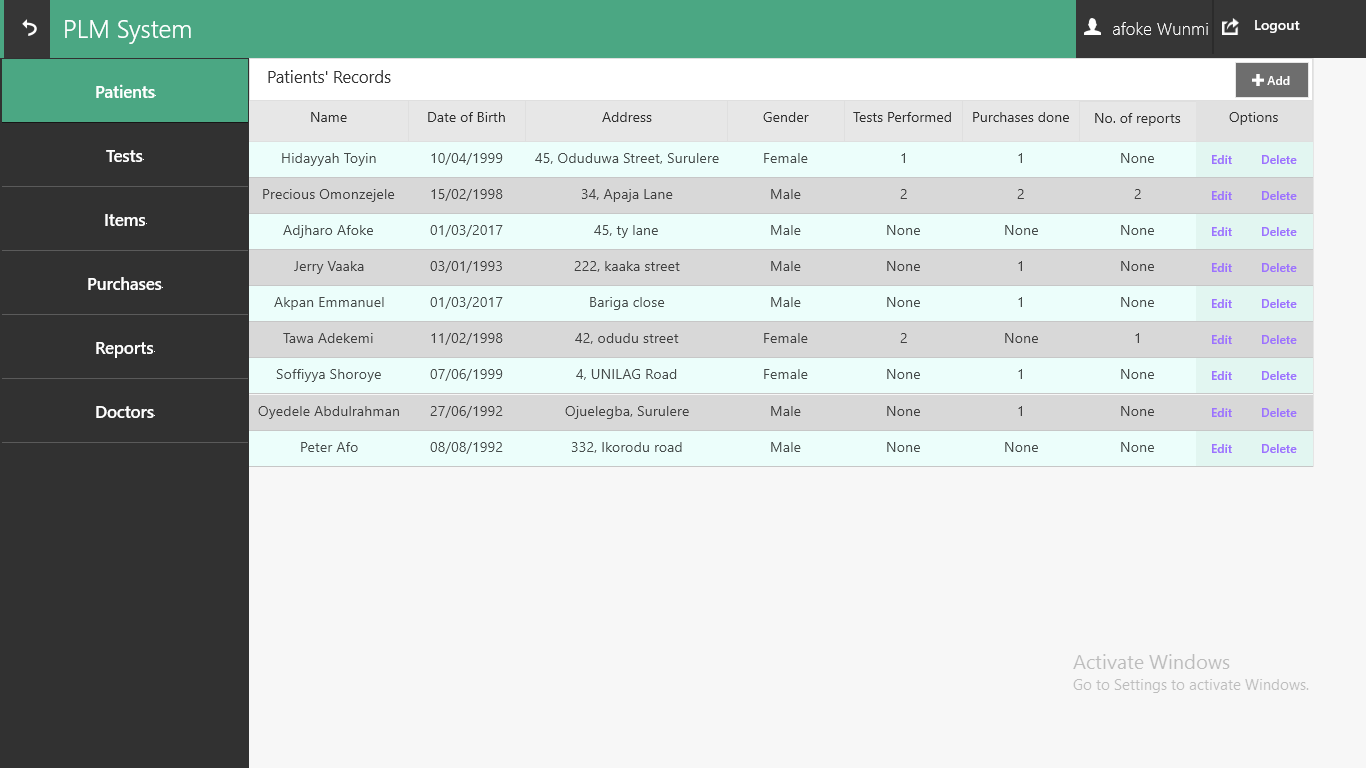
The sign up page allows the admin to sign up, asking for few relevant details to get them going, if already having an account, there’s a quick link to the login page.

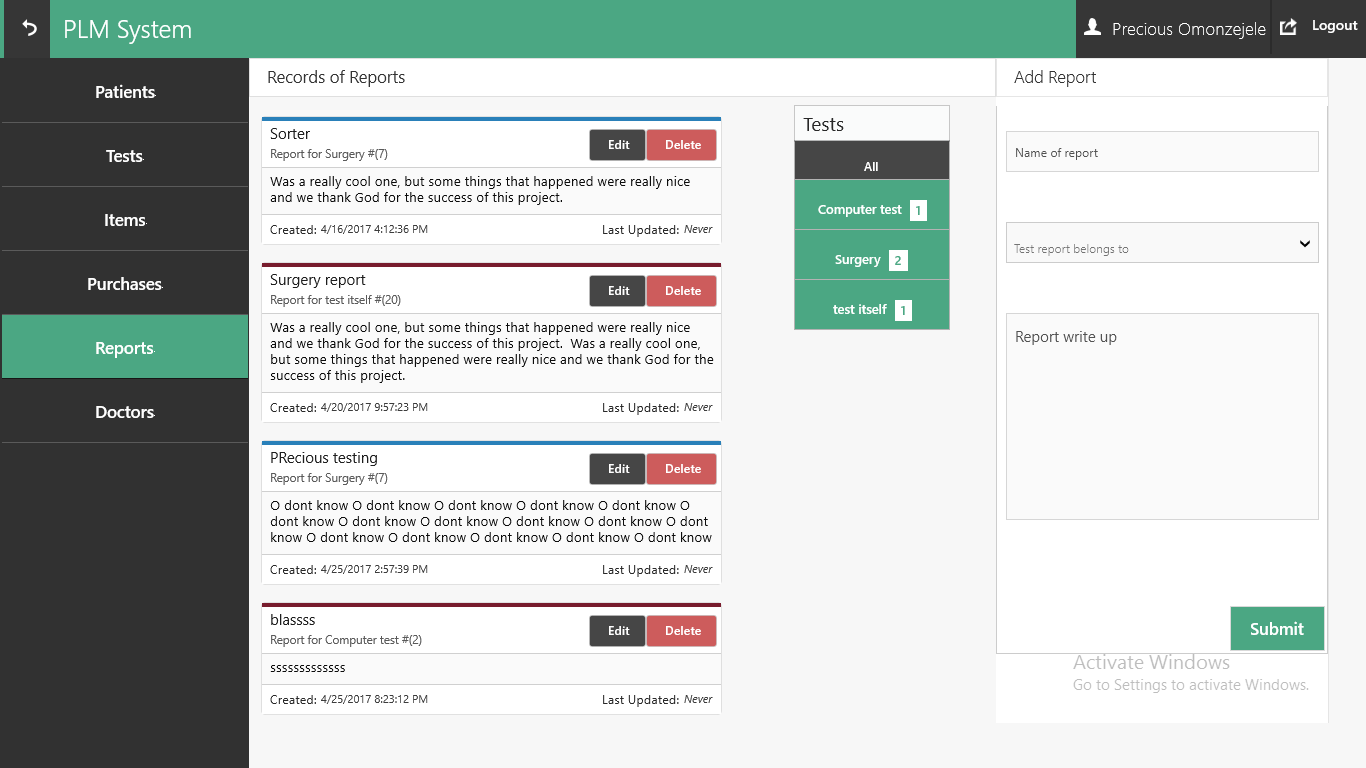
**Home page**



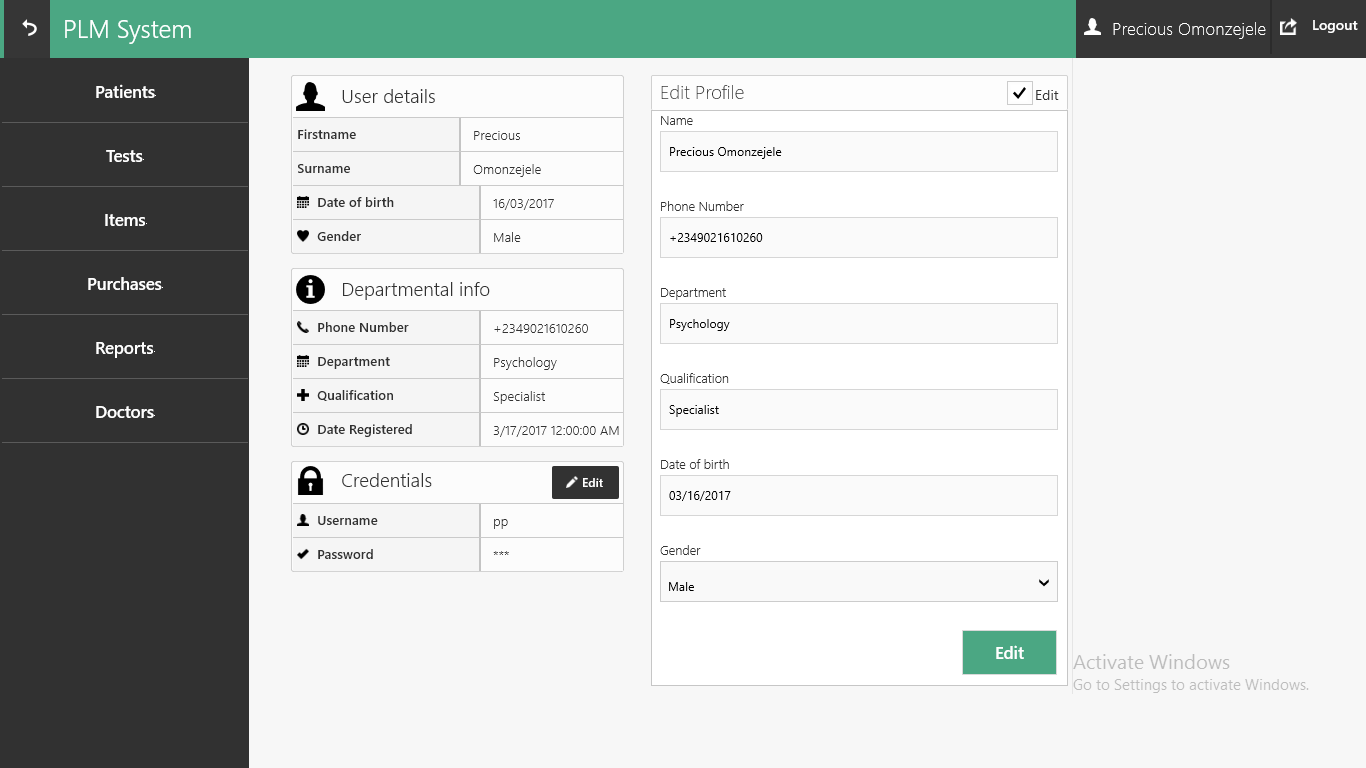
The home page takes the logged in admin straight to business, displaying an overview of records that were added and edited.

**Records page**





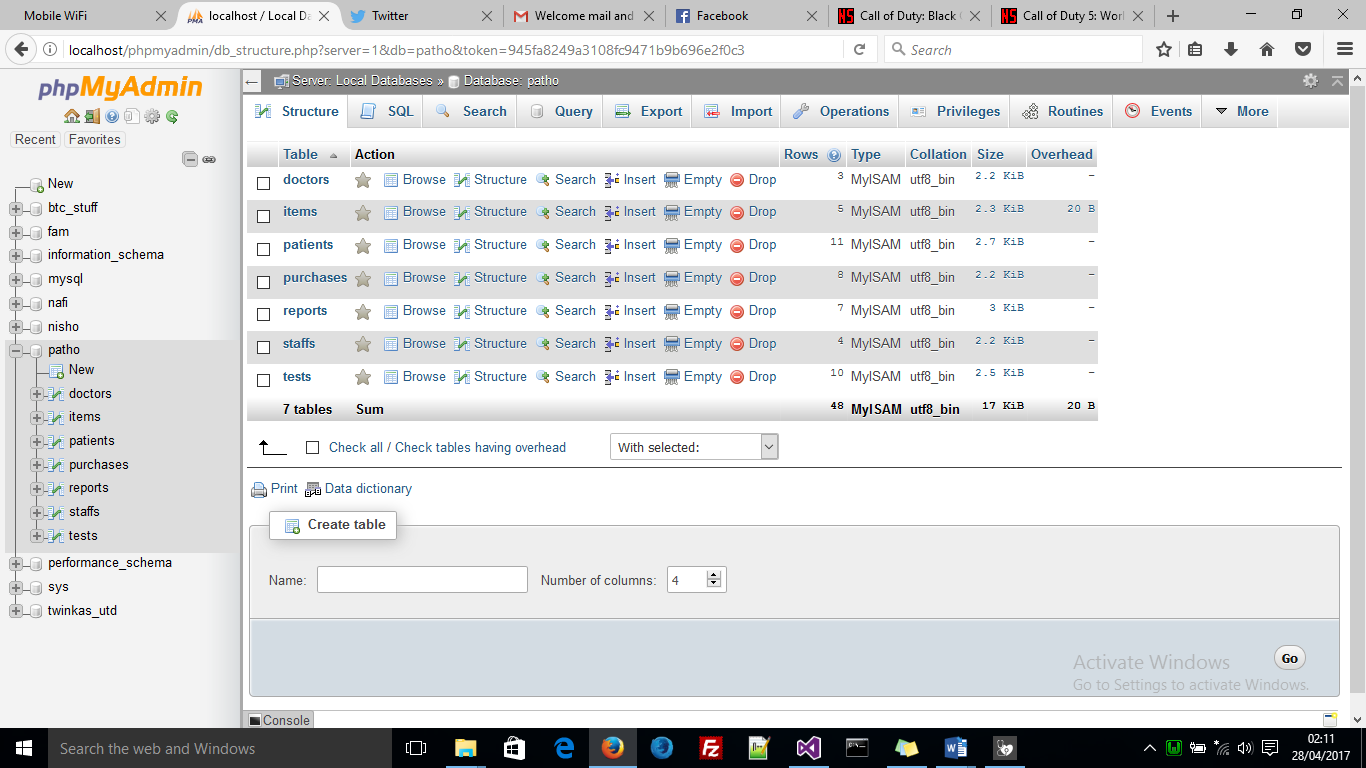
The record’s page shows the available records of the selected record category, patients, tests, reports, items and so on, and it allows the admin to edit, add, and delete them.

**Profile page** 

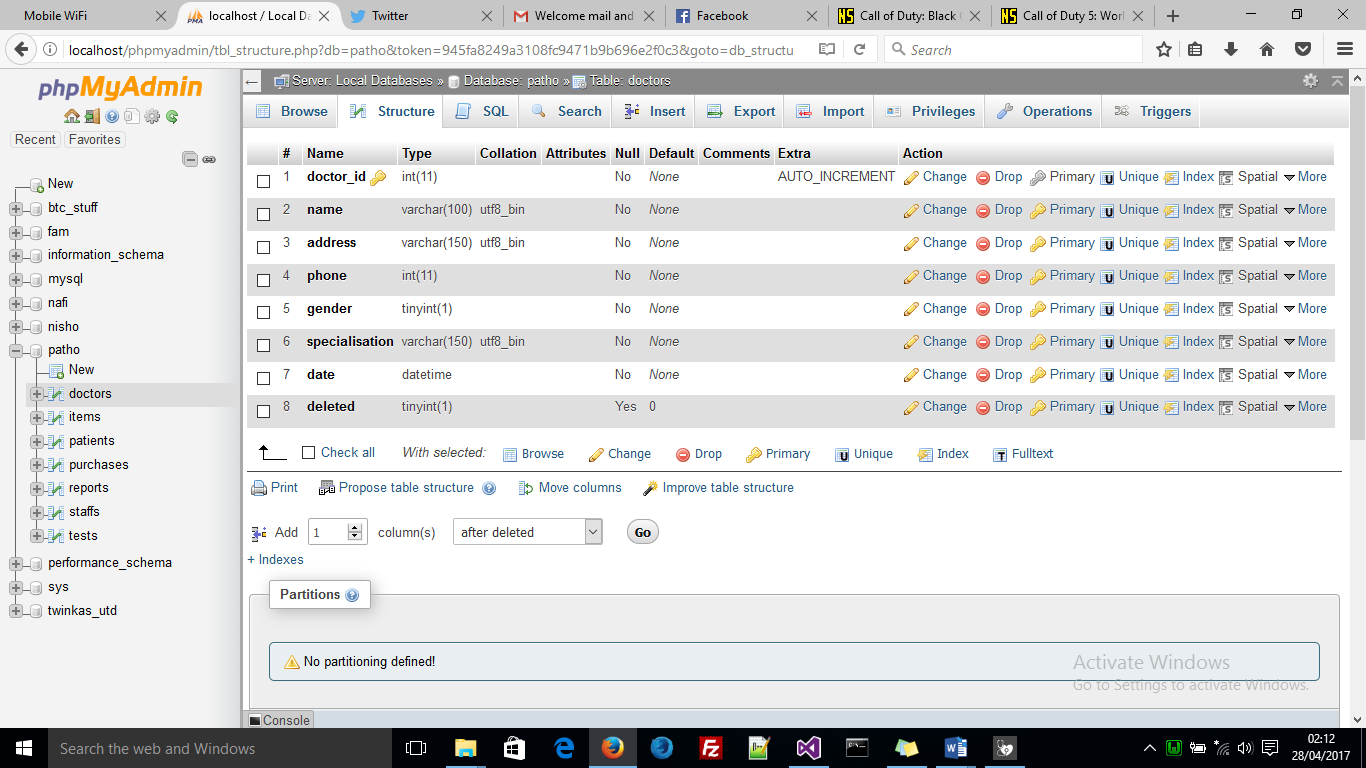
The profile page of the admin allows them to edit their info, credentials. It doesn’t allow anyone except the admin to edit this such that, before any detail can be entered, the password of the admin has to be re-entered.

1. Database structure:

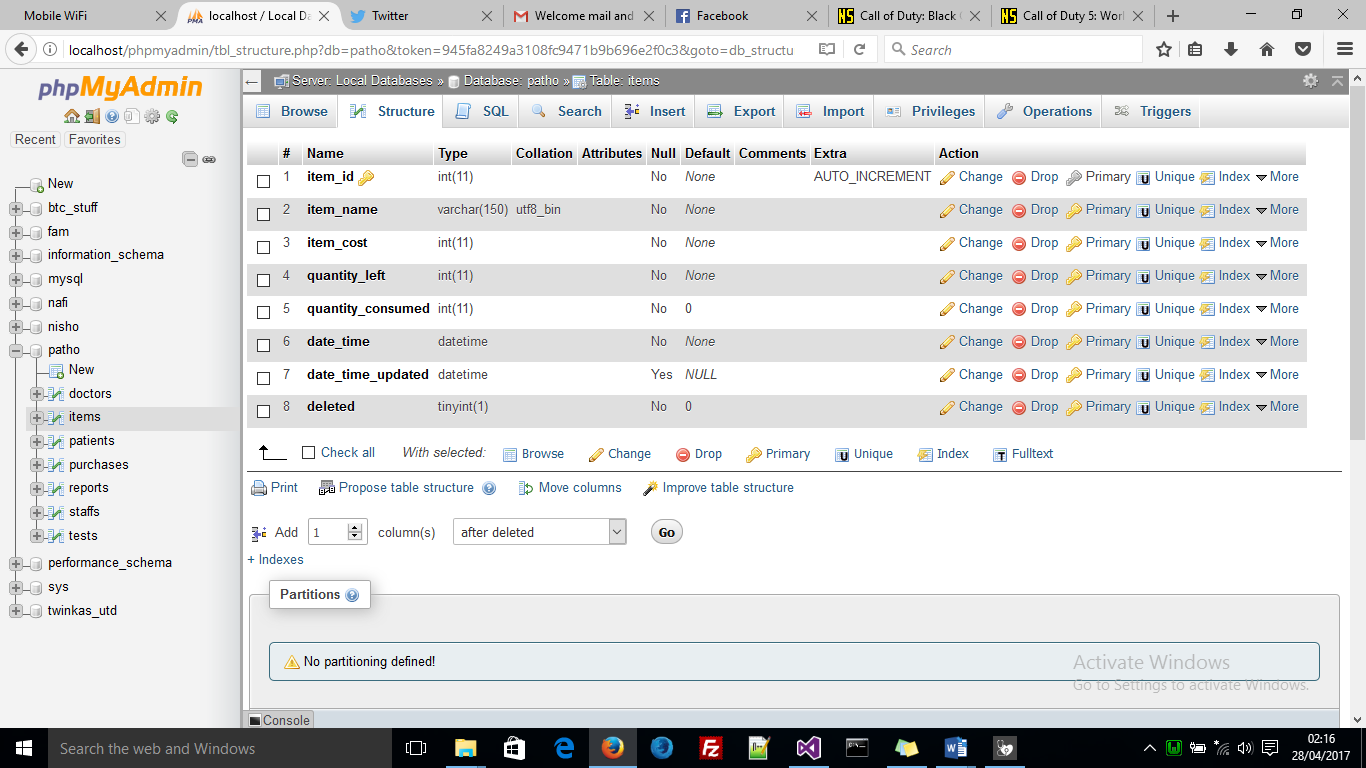
**Overview**



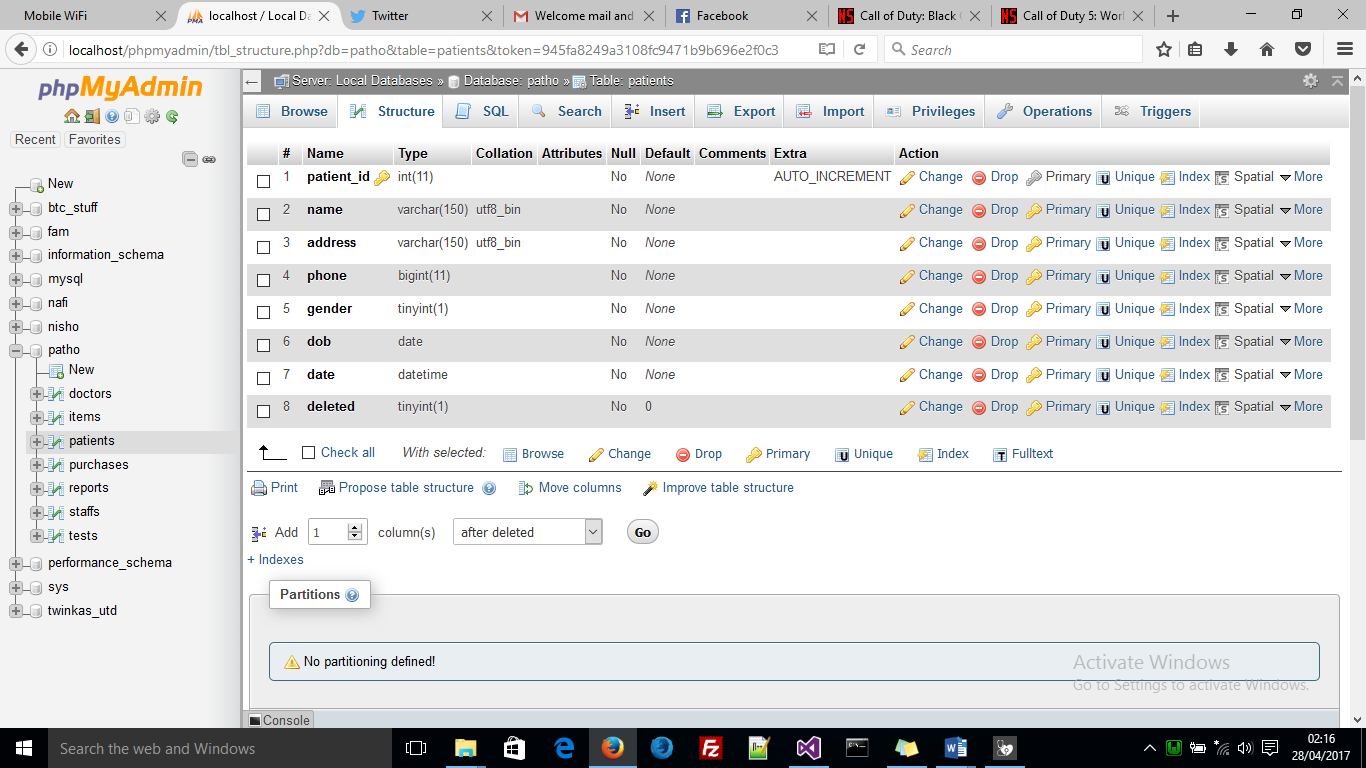
**Doctors Table**



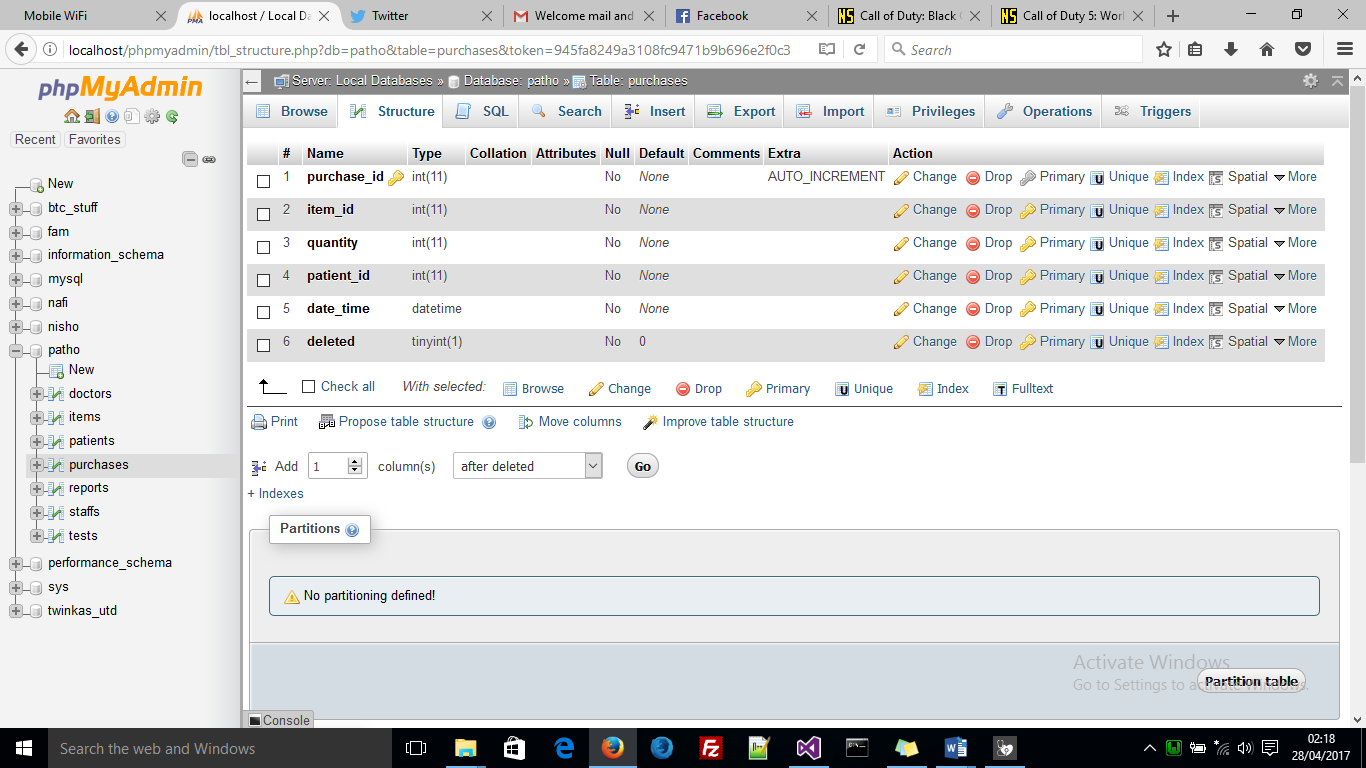
**Items Table**



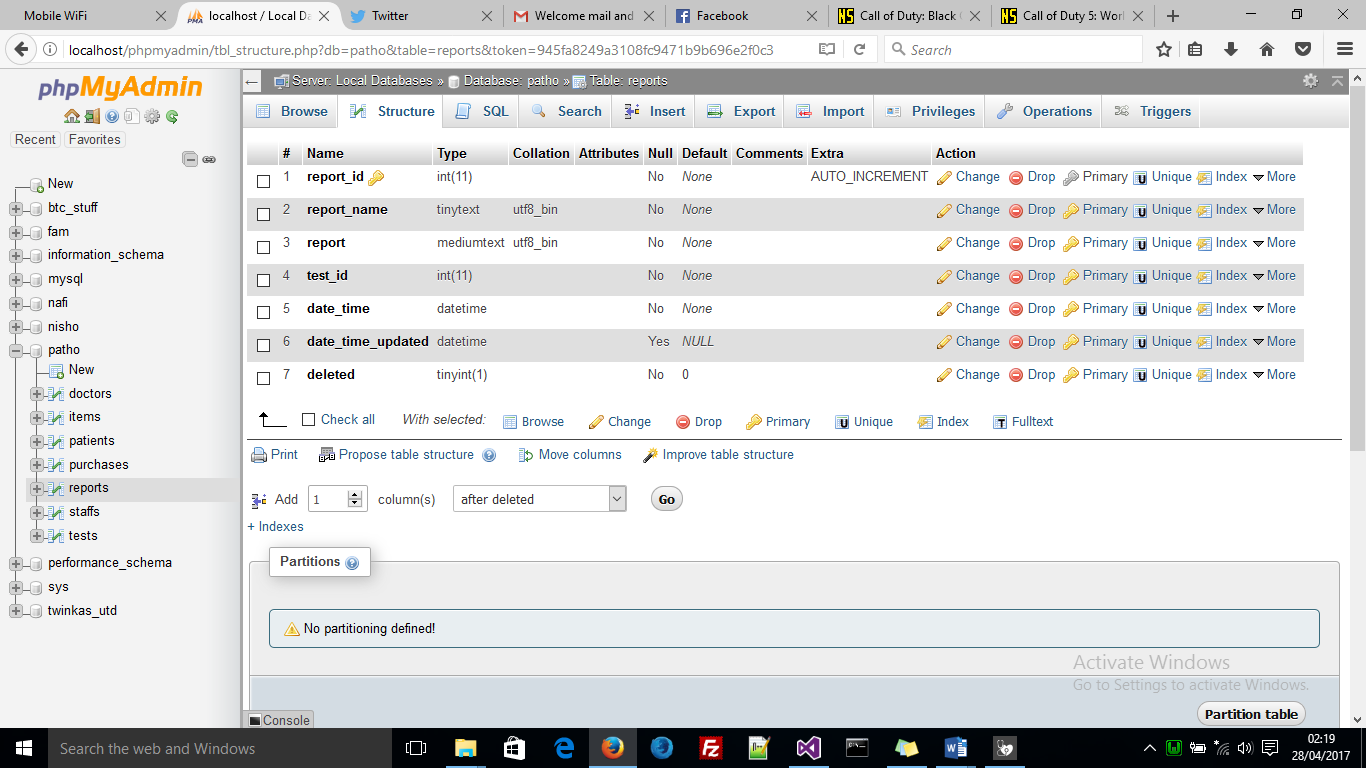
**Patients Table**



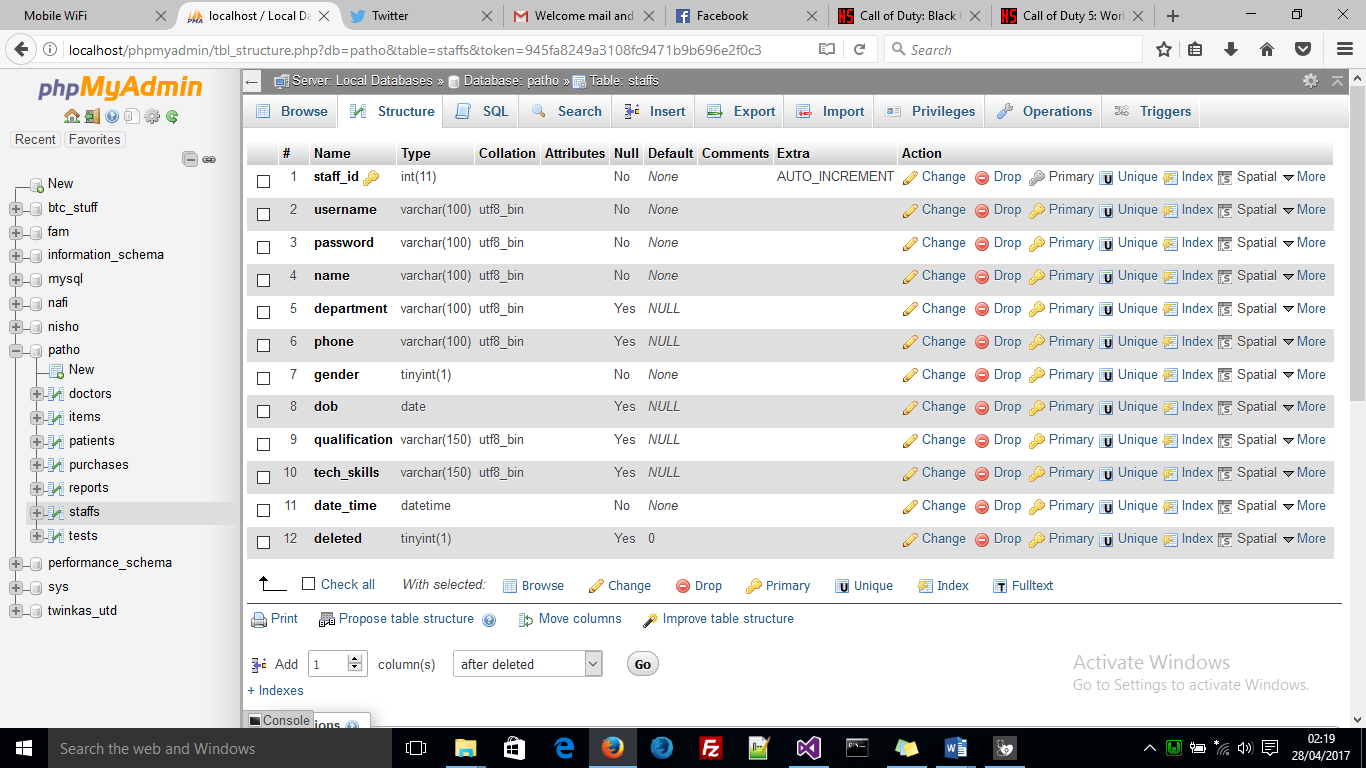
**Purchases Table**



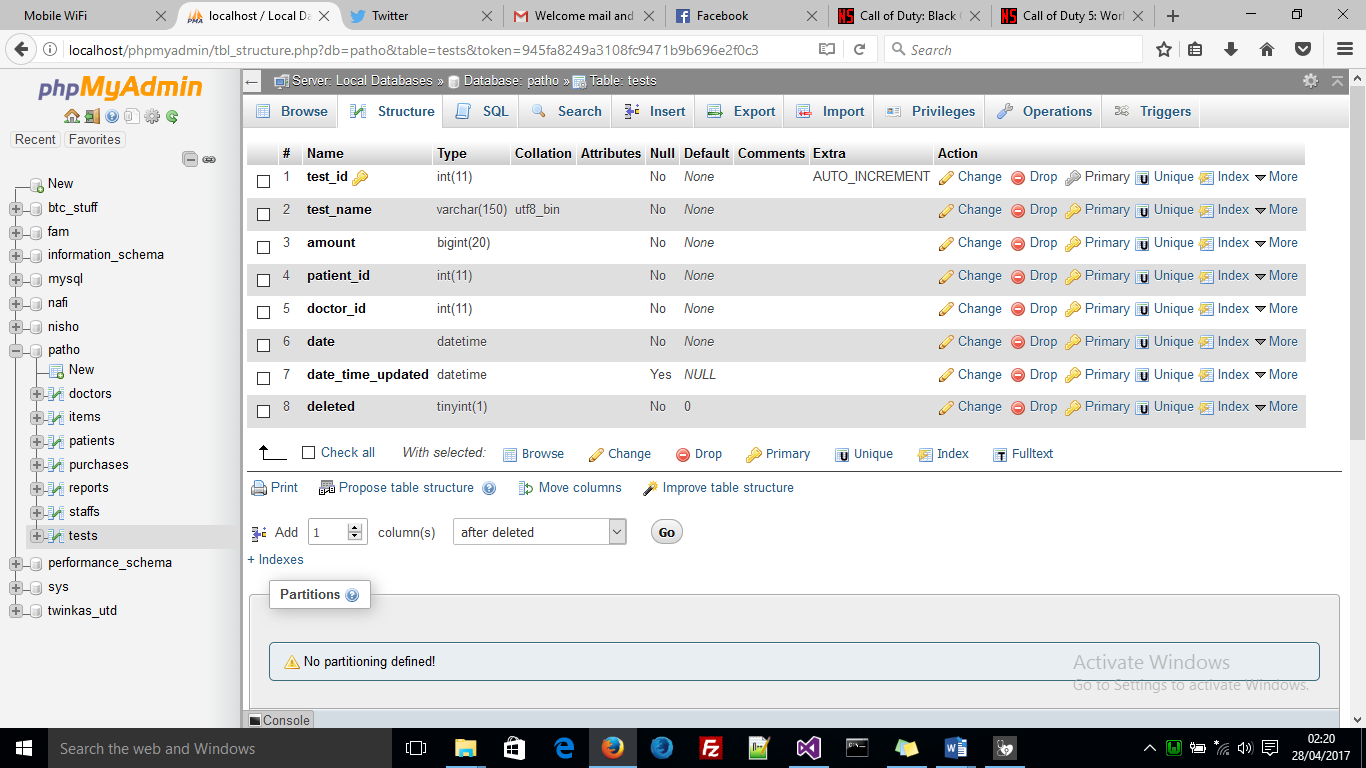
**Reports Table**



**Staffs Table**



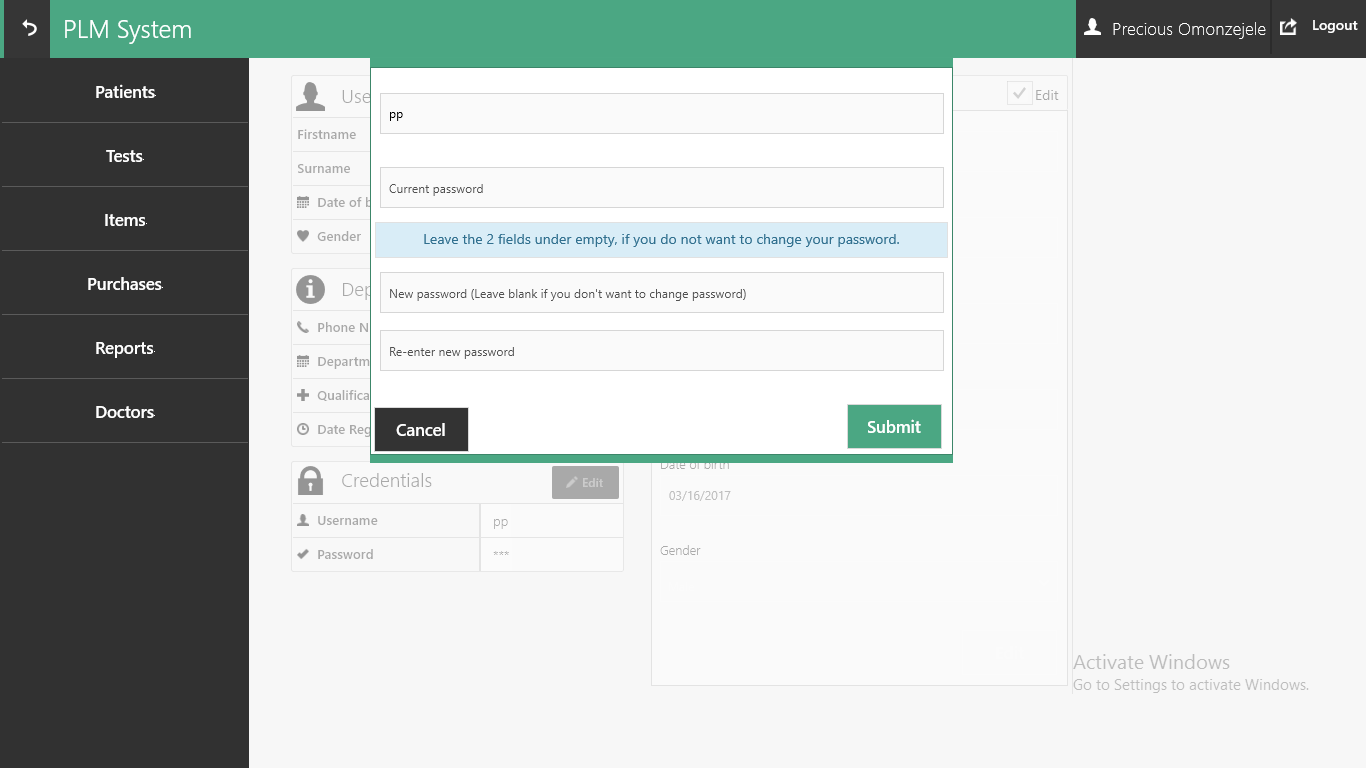
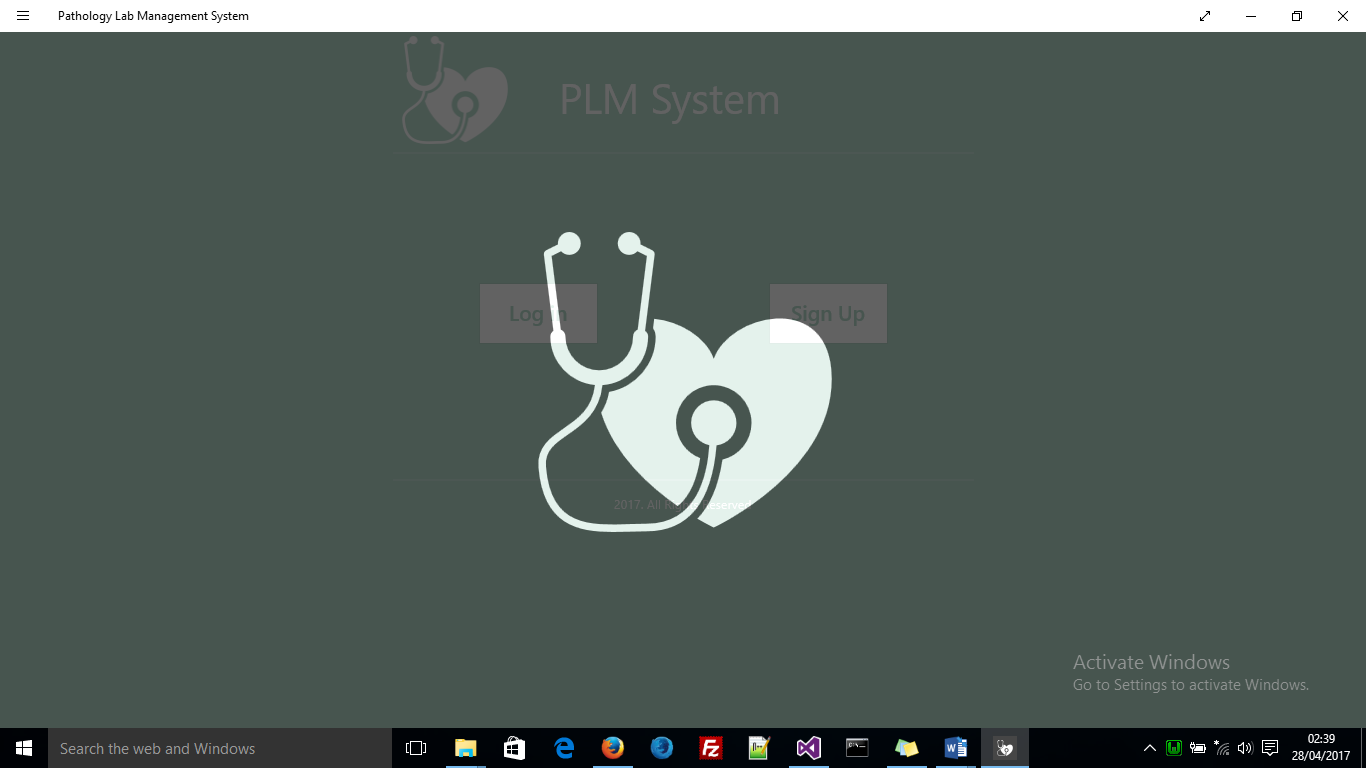
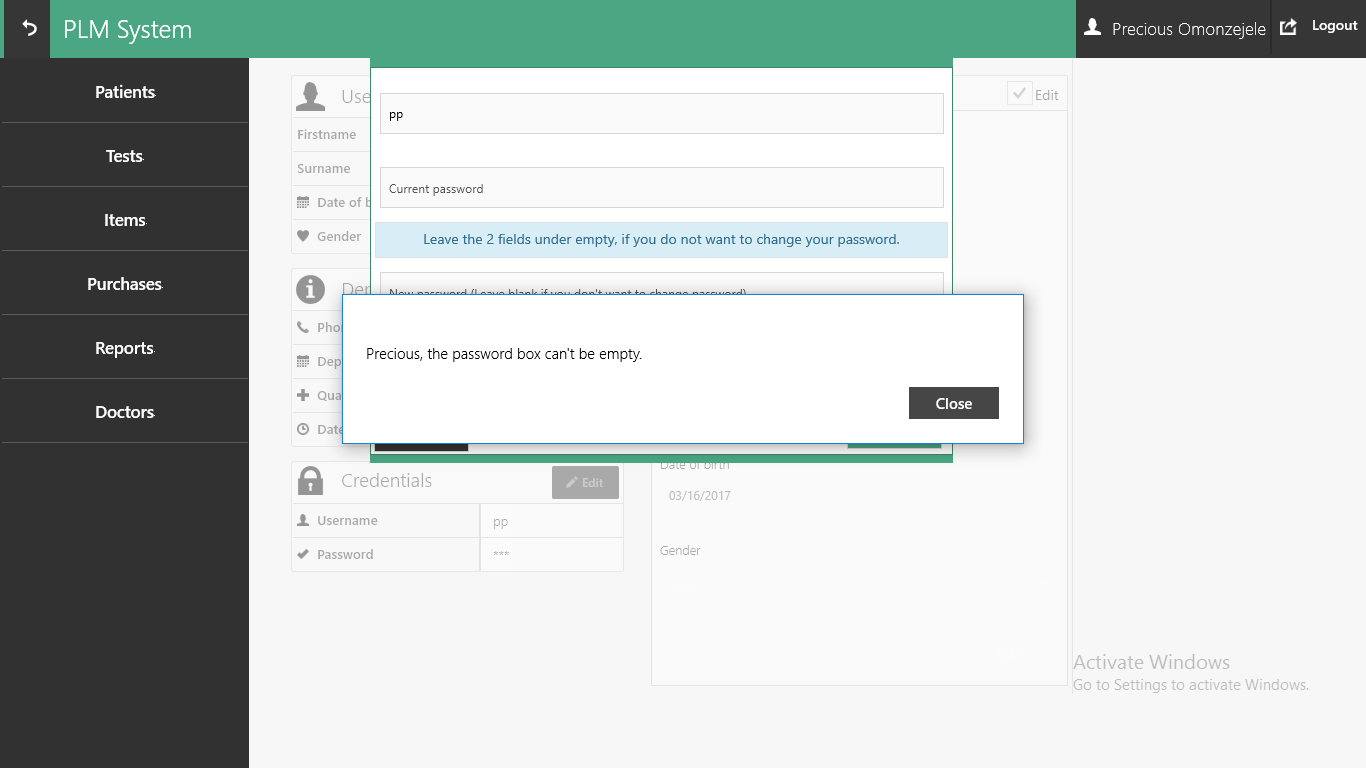
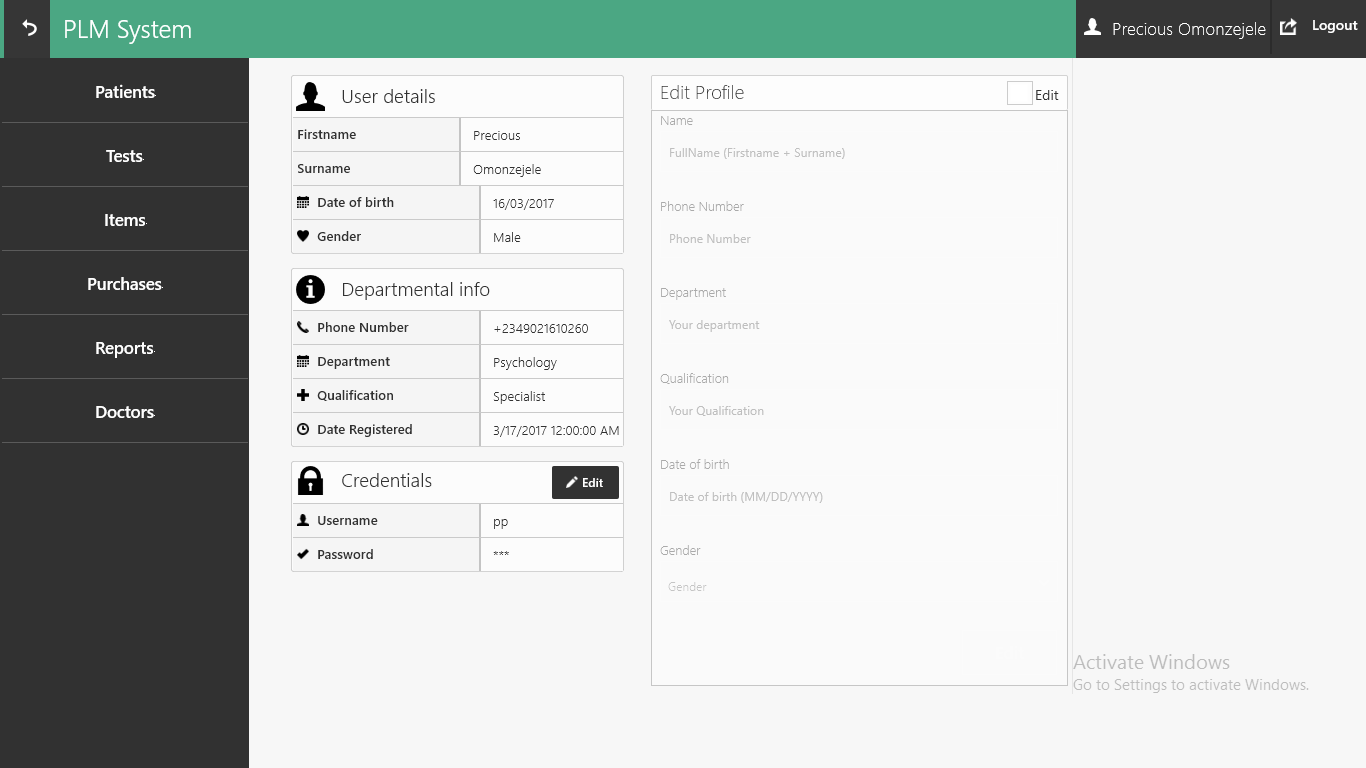
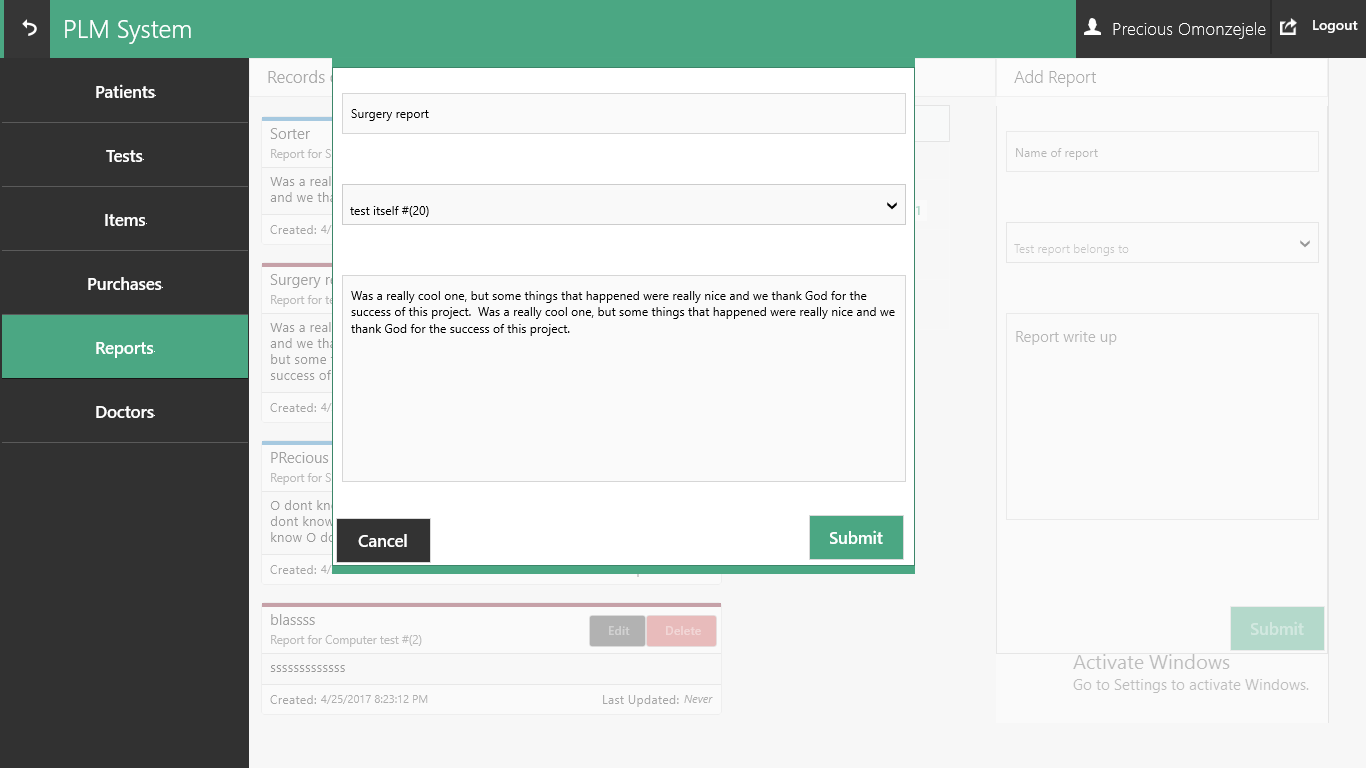
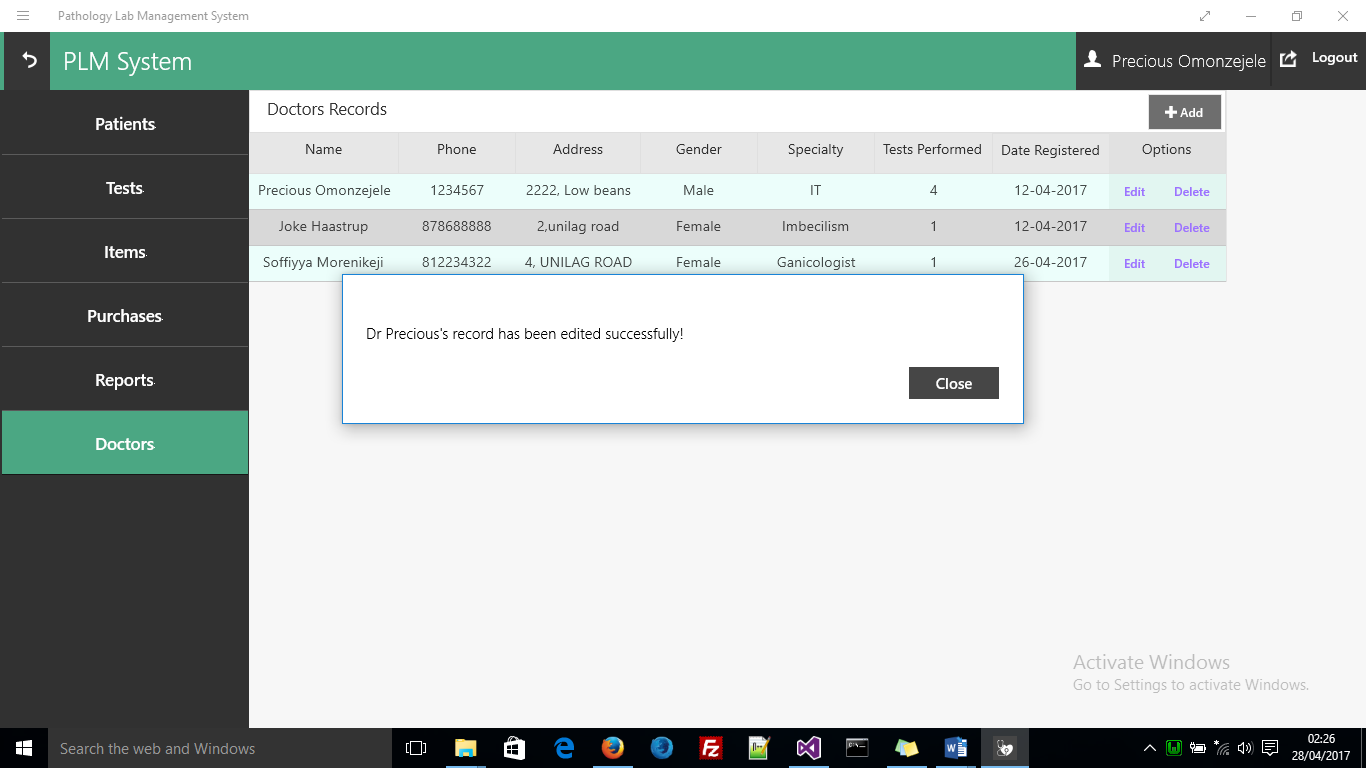
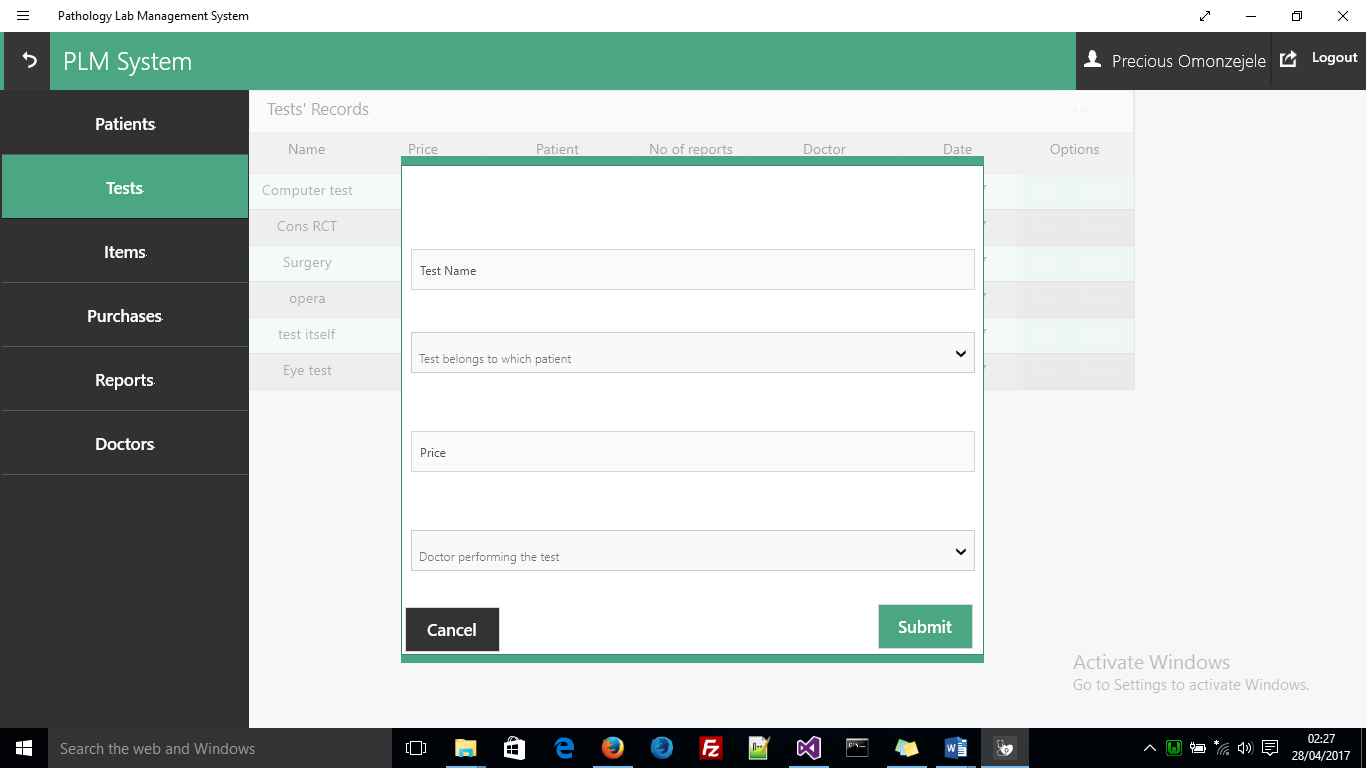
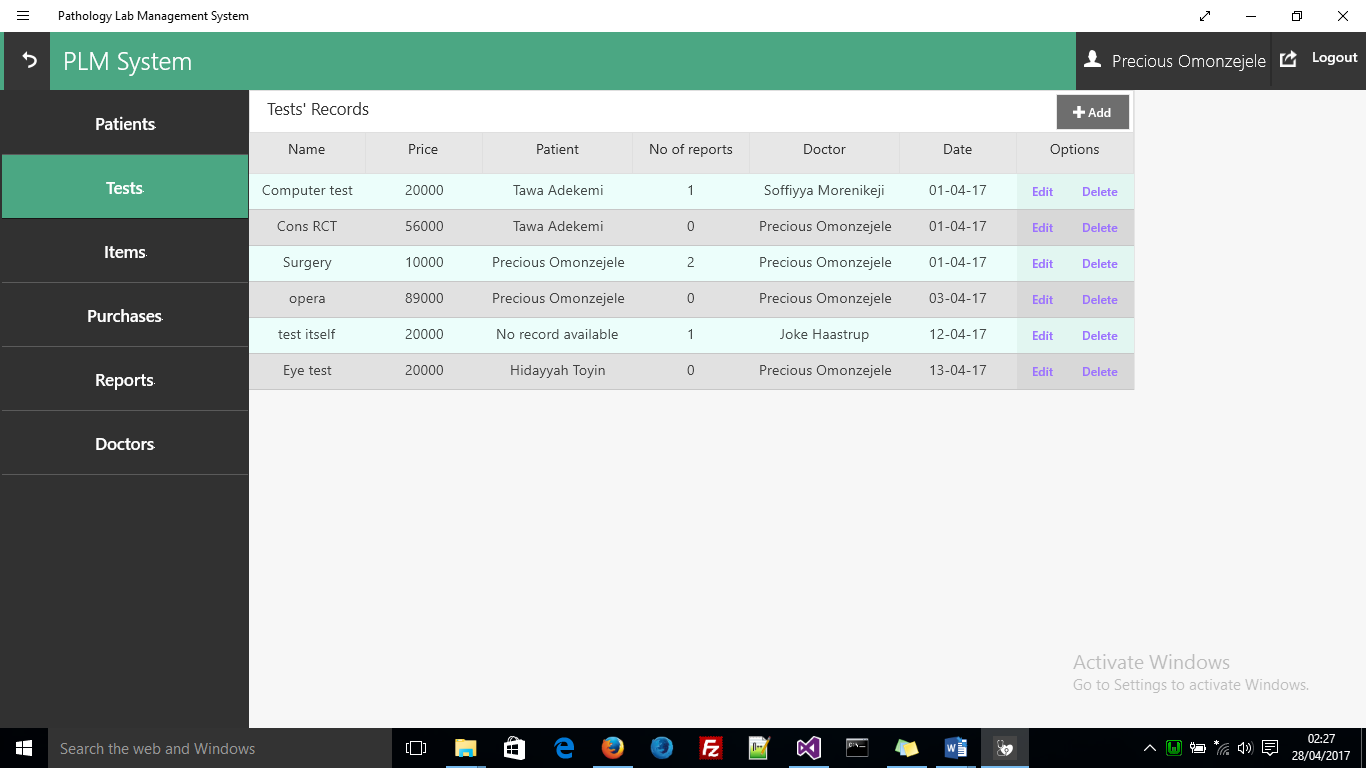
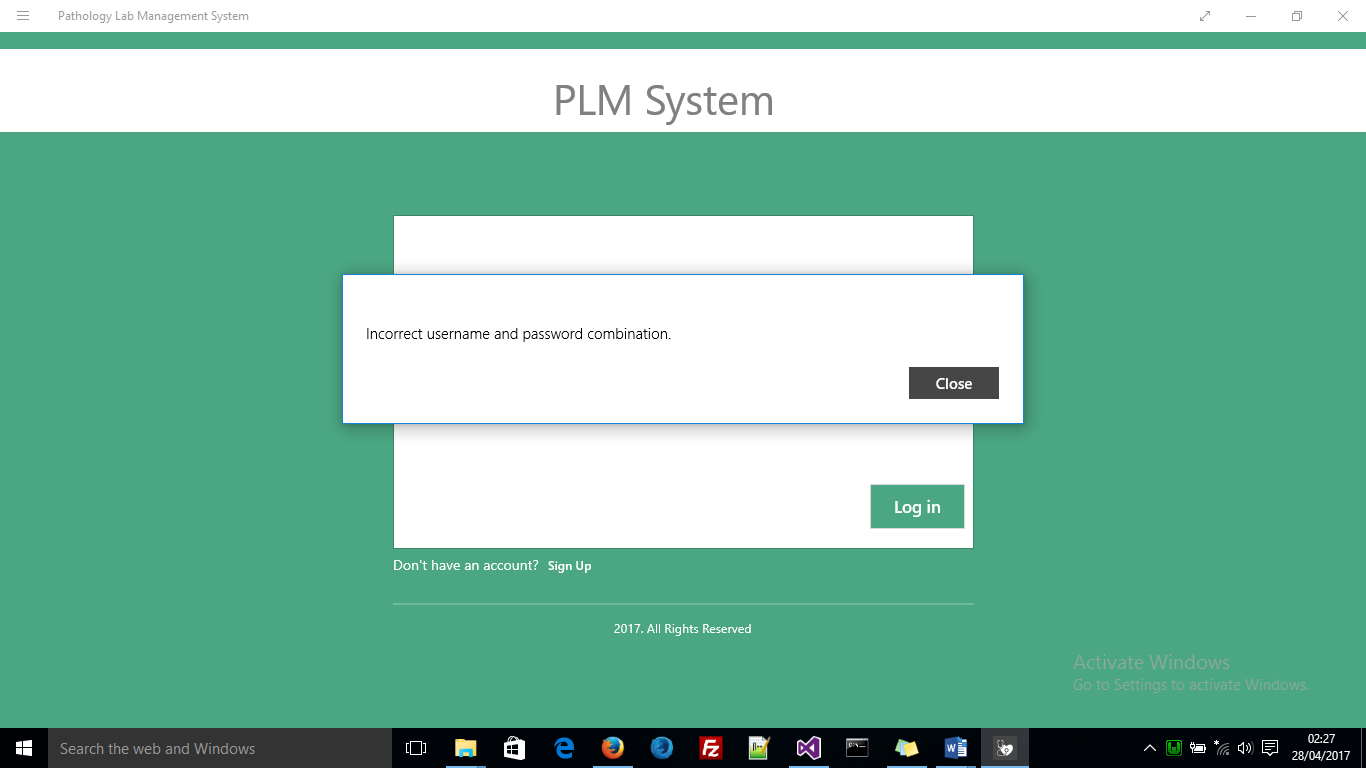
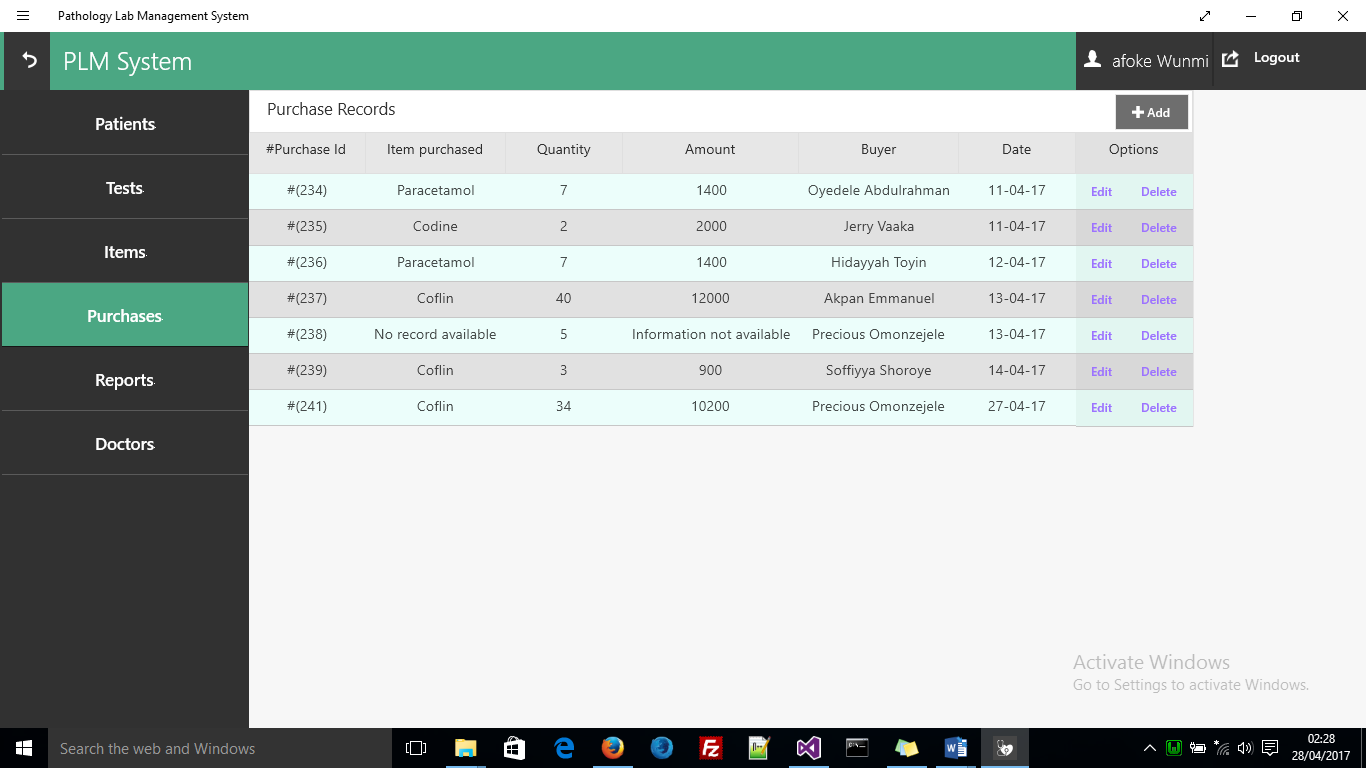
**Tests Table**



**CHAPTER THREE**

*IMPLEMENTATION*

1. UI Design:

1. Source code:

MainPage XAML

<Page

x:Class="Lpms.MainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:Lpms"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

<Grid Background="#4BA783">

<StackPanel>

<StackPanel x:Name="title" MaxWidth="700" Height="200" >

<StackPanel Orientation="Horizontal">

<Image Source="assets/icons/logo.png" Width="130" Height="130" Margin ="10 5 30 10"/>

<TextBlock Text="LPMS System" FontSize="50" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="30 35 55 10"/>

</StackPanel>

<Border BorderBrush="#33ffffff" BorderThickness="1"/>

</StackPanel>

<StackPanel Width="700" Orientation="Horizontal" Margin="0 100 0 10">

<Button x:Name="LogIn" Click="LogIn\_Click" Content="Log in" FontSize="25" Width="150" Height="80" Margin="100 0 0 0" BorderBrush="#26000000" BorderThickness="1" RenderTransformOrigin="1.018,0.693" Background="#ffffff" Foreground="#4BA783"/>

<Button x:Name="SignUp" Click="SignUp\_Click" Content="Sign Up" FontSize="25" Width="150" Height="80" Margin="200 0 0 0" Background="#ffffff" BorderBrush="#26000000" BorderThickness="1" Foreground="#4BA783"/>

</StackPanel>

<StackPanel Width="700" Margin="0 150 0 0">

<Border BorderBrush="#33ffffff" BorderThickness="1"/>

<TextBlock x:Name="copyright" Text="2017. All Rights Reserved" HorizontalAlignment="Center" FontSize="15" Padding="0 20 0 0"/>

</StackPanel>

</StackPanel>

</Grid>

</Page>

Mainpage C#

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

using Lpms.Classes;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace Lpms

{

/// <summary>

/// The cover page

/// </summary>

public sealed partial class MainPage : Page

{

public MainPage()

{

this.InitializeComponent();

}

private void LogIn\_Click(object sender, RoutedEventArgs e)

{

this.Frame.Navigate(typeof(Login), null);

}

private void SignUp\_Click(object sender, RoutedEventArgs e)

{

this.Frame.Navigate(typeof(SignUp), null);

}

}

}

Sign In page XAML

<Page

x:Class="Lpms.Login"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:Lpms"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

<Grid Background="#4BA783">

<StackPanel>

<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 20 0 0">

<TextBlock Text="PLM System" Foreground="Gray" FontSize="50" HorizontalAlignment="Center" Margin="10 35 55 10"/>

<Border BorderBrush="#cc000000" BorderThickness="1"/>

</StackPanel>

<Border BorderBrush="#33000000" BorderThickness="1" Width="700" Margin="0 100 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="Vertical" Margin="0 0 0 0">

<TextBox x:Name="username" Width="auto" PlaceholderText="Username" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 100 10 20"/>

<PasswordBox x:Name="pass" Width="auto" PlaceholderText="Password" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<Button x:Name="LogInBtn" Click="LogInBtn\_Click" HorizontalAlignment="Right" Content="Log in" FontSize="20" Width="120" Height="60" Margin="0 30 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

<StackPanel Orientation="Horizontal" Width="700">

<TextBlock Text="Don't have an account?" FontSize="17"/>

<HyperlinkButton x:Name="GoToSignUp" Foreground="#ffffff" Padding="0" Margin="5 -6 0 0" Click="GoToSignUp\_Click">

<HyperlinkButton.Content>

Sign Up

</HyperlinkButton.Content>

</HyperlinkButton>

</StackPanel>

<StackPanel Width="700" Margin="0 30 0 0">

<Border BorderBrush="#33ffffff" BorderThickness="1"/>

<TextBlock x:Name="copyright" Text="2017. All Rights Reserved" HorizontalAlignment="Center" FontSize="15" Padding="0 20 0 0"/>

</StackPanel>

</StackPanel>

</Grid>

</Page>

Sign in page C#

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

using Lpms.Classes;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace Lpms

{

/// <summary>

/// The Login Page

/// </summary>

public sealed partial class Login : Page

{

public int user\_token;

public Login()

{

this.InitializeComponent();

}

private void GoToSignUp\_Click(object sender, RoutedEventArgs e)

{

this.Frame.Navigate(typeof(SignUp),null);

}

private void LogInBtn\_Click(object sender, RoutedEventArgs e)

{

//CHECK IF THE NORMAL THINGS ARE OKAY

if (username.Text.Trim() == "")

{

PopUp p = new PopUp("Your username field is empty");

}

if (pass.Password.Trim() == "")

{

PopUp p = new PopUp("Please input a password");

}

else

{

//CALL IN THE QUERY STUFF

//Connection c = new Connection();

// c.ChooseDb("stuff");

Query q = new Query();

q.Get("staffs", "WHERE username = '" + username.Text.Trim() + "' And password = '" + pass.Password.Trim() + "' ");

//set the count

int count = 0;

//while (q.reader.Read())

//{

// user\_token = Convert.ToInt32(q.reader[0]);

// count += 1;

//}

q.Records();

while (count < q.CountRows())

{

user\_token = Convert.ToInt32(q.Results[0, 0]);

count += 1;

}

if (count > 0)//CHECK IF THE LOGIN DETAILS ARE VALID

{

(App.Current as App).User\_token = user\_token;

this.Frame.Navigate(typeof(Home), null);

}

else

{

PopUp p = new PopUp("Incorrect username and password combination.");

}

}

}

}

}

Sign up page XAML

<Page

x:Class="Lpms.SignUp"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:Lpms"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

<Grid Background="#4BA783">

<ScrollBar HorizontalAlignment="Left" VerticalAlignment="Top"/>

<StackPanel>

<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 2 0 0">

<TextBlock Text="PLM System" Foreground="Gray" FontSize="50" HorizontalAlignment="Center" Margin="10 35 55 10"/>

<Border BorderBrush="#cc000000" BorderThickness="1"/>

</StackPanel>

<Border BorderBrush="#33000000" BorderThickness="1" Width="700" Margin="0 20 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<TextBox x:Name="name" Width="auto" PlaceholderText="Your Name (Firstname + Surname)" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 100 10 20"/>

<TextBox x:Name="department" Width="auto" PlaceholderText="Department" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 20 10 20"/>

<TextBox x:Name="username" Width="auto" PlaceholderText="Username" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 20 10 20"/>

<PasswordBox x:Name="pass" Width="auto" PlaceholderText="Password" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 20 10 40"/>

<ComboBox x:Name="gender" Height="50" Padding="10 15 0 0" PlaceholderText="Gender" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 20 10 20">

<ComboBoxItem x:Name="male">Male</ComboBoxItem>

<ComboBoxItem x:Name="female">Female</ComboBoxItem>

</ComboBox>

<Button x:Name="SignUpBtn" Click="SignUpBtn\_Click" HorizontalAlignment="Right" Content="Sign Up" FontSize="20" Width="120" Height="60" Margin="0 30 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

<StackPanel Orientation="Horizontal" Width="700" HorizontalAlignment="Right">

<TextBlock Text="Already have an account?" FontSize="17"/>

<HyperlinkButton x:Name="GoToLogIn" Foreground="#ffffff" Padding="0" Margin="5 -6 0 0" Click="GoToLogIn\_Click">

<HyperlinkButton.Content>

Log In

</HyperlinkButton.Content>

</HyperlinkButton>

</StackPanel>

<StackPanel Width="700" Margin="0 30 0 0">

<Border BorderBrush="#33ffffff" BorderThickness="1"/>

<TextBlock x:Name="copyright" Text="2017. All Rights Reserved" HorizontalAlignment="Center" FontSize="15" Padding="0 20 0 0"/>

</StackPanel>

</StackPanel>

</Grid>

</Page>

Sign up page C#

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

using Lpms.Classes;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace Lpms

{

/// <summary>

/// An empty page that can be used on its own or navigated to within a Frame.

/// </summary>

public sealed partial class SignUp : Page

{

public int user\_id;

public SignUp()

{

this.InitializeComponent();

}

/// <summary>

/// when the sign up button is clicked

/// </summary>

private void SignUpBtn\_Click(object sender, RoutedEventArgs e)

{

Filter filter = new Filter();

if (filter.Trim(username.Text) == "" || filter.Trim(department.Text) == "" || filter.Trim(name.Text) == "" || filter.Trim(pass.Password) == "" || gender.SelectionBoxItem == null)

{

if (filter.Trim(username.Text) == "")

{

PopUp m = new PopUp("Username field is empty");

}

else if (filter.Trim(department.Text) == "")

{

PopUp m = new PopUp("department field is empty");

}

else if (filter.Trim(name.Text) == "")

{

PopUp m = new PopUp("Your name field is empty");

}

else if (filter.Trim(pass.Password) == "")

{

PopUp m = new PopUp("Password field is empty");

}

else if (gender.SelectionBoxItem == null)

{

PopUp m = new PopUp("Please Select a gender");

}

}

else

{

// THE FIELDS AREN'T EMPTY, SO CHECK FOR OTHER STUFF

Query q = new Query();

string user = filter.Trim(username.Text);

string password = filter.Trim(pass.Password);

string fullname = filter.Trim(name.Text);

string depart = filter.Trim(department.Text);

string gen = filter.Trim(gender.SelectionBoxItem.ToString());

if (!(user == "" || password == "" || fullname == "" || depart == "" || gen == ""))

{

//first check if the username hasn't been used already

q.Get("staffs", "Where username = '" + user + "'");

if (q.CountRows() > 0)

{

PopUp p = new PopUp("Sorry, username has already been taken. Try something else");

}

else

{

if (gen.ToLower() == "male")

gen = "1";

else if (gen.ToLower() == "female")

gen = "0";

//insert

//ADD THE VALUES IN AN ARRAY , CAUSE THAT'S HOW THE HOW METHOD CAN ACCEPT VALUES

string[] vals = { user, password, fullname, depart, gen, "NOW()" };

//q.ChooseDb("nisho");

// "INSERT into staffs(username,password,name,department,gender,date\_time) VALUES('user','password','fullname','depart','gen',NOW())"

q.successMessage = "Thank You " + filter.FirstName(fullname) + ", Your account has been created";

q.Add("staffs", vals, "username,password,name,department,gender,date\_time");

//LOG IN THE USER

q.Get("staffs", "Where username = '" + user + "'");

q.Record();

if (q.CountRows() == 1)

{

user\_id = Convert.ToInt32(q.Result[0]);

}

(App.Current as App).User\_token = user\_id;

//NAVIGATE

this.Frame.Navigate(typeof(Home), null);

}

}

else

{

PopUp p = new PopUp("sorry, an important field is missing");

}

}

}

private void GoToLogIn\_Click(object sender, RoutedEventArgs e)

{

this.Frame.Navigate(typeof(Login), null);

}

}

}

Home XAML

<Page

x:Class="Lpms.Home"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:Lpms"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

<Grid Background="#f2f2f2">

<Grid>

<Grid.Resources>

<Style x:Key="btn" TargetType="Button">

<Style.Setters>

<Setter Property="Width" Value="200"/>

<Setter Property="Height" Value="200"/>

<Setter Property="Background" Value="#cc000000"/>

<Setter Property="HorizontalAlignment" Value="Stretch" />

<Setter Property="Margin" Value="-3" />

<Setter Property="VerticalAlignment" Value="Stretch" />

<Setter Property="Margin" Value="0 5 0 20" />

</Style.Setters>

</Style>

<Style x:Key="inner\_btn" TargetType="StackPanel">

<Setter Property="Orientation" Value="Vertical"/>

</Style>

<Style x:Name="results" TargetType="StackPanel">

<Setter Property="Margin" Value="0 10 0 0" />

<Setter Property="Width" Value="30"/>

<Setter Property="Height" Value="30"/>

<Setter Property="Background" Value="#FFDC3E3E"/>

</Style>

<Style x:Key="result\_txt" TargetType="TextBlock">

<Setter Property="Foreground" Value="#ffffff"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontSize" Value="20"/>

</Style>

<Style x:Key="img" TargetType="Image">

<Setter Property="Width" Value="20"/>

<Setter Property="Height" Value="20"/>

</Style>

<Style x:Key="btn\_txt" TargetType="TextBlock">

<Setter Property="FontSize" Value="20"/>

</Style>

<Style x:Key="user\_details" TargetType="TextBlock">

<Setter Property="FontSize" Value="16"/>

<Setter Property="Padding" Value="10 3 5 0"/>

<Setter Property="TextTrimming" Value="CharacterEllipsis"/>

<Setter Property="HorizontalAlignment" Value="Left"/>

<Setter Property="MaxWidth" Value="310"/>

<Setter Property="FontWeight" Value="Light"/>

<Setter Property="Foreground" Value="White"/>

</Style>

</Grid.Resources>

<StackPanel>

<StackPanel x:Name="title" Width="auto" Height="70" Background="#4BA783" Margin="0 0 0 0">

<StackPanel Orientation="Vertical" Width="auto">

<TextBlock Text="PLM System" Foreground="#fff" FontSize="30" FontWeight="Bold" FontStretch="ExtraExpanded" HorizontalAlignment="Left" Margin="10,20,0,10"/>

<StackPanel HorizontalAlignment="Right" Width="350" Background="#FF363636" VerticalAlignment="Stretch" Margin="0 -69 0 0" Height="130">

<StackPanel Orientation="Horizontal" Height="70">

<StackPanel Orientation="Horizontal">

<Image Source="Assets/icons/big-user-white.png" Margin="10 0 0 0" Height="20" Width="20"/>

<HyperlinkButton Padding="-2" Margin="0" x:Name="Profile" Click="Profile\_Click">

<TextBlock Foreground="White" FontWeight="Light" MaxWidth="210" TextWrapping="Wrap" TextTrimming="WordEllipsis" x:Name="fullname" VerticalAlignment="Center" Padding="10 10 0 5" FontSize="21" Height="44" Margin="0,14,0,10"/>

</HyperlinkButton>

</StackPanel>

<Border BorderBrush="#33000000" BorderThickness="1"/>

<StackPanel Orientation="Horizontal">

<Image Source="Assets/icons/20-action-white.png" Margin="10 0 0 0" Height="20" Width="20"/>

<HyperlinkButton VerticalAlignment="Center" x:Name="Logout" Click="Logout\_Click" FontSize="17" Foreground="#ffffff" Content="Logout"/>

</StackPanel>

</StackPanel>

</StackPanel>

</StackPanel>

<Border BorderBrush="#cc000000" BorderThickness="1"/>

</StackPanel>

<StackPanel Height="100">

<Border BorderBrush="#90000000" BorderThickness="1">

<Image Source="Assets/Images/home-header.png" Margin="-1 0.2 0 0" Height="auto" Width="auto"/>

</Border>

</StackPanel>

</StackPanel>

<!-- BODY -->

<Grid Margin="0 70 0 10">

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

</Grid.ColumnDefinitions>

<Border Grid.Column="0" Grid.Row="0" Width="800" Margin="0 50 0 0" Padding="50 0 80 0" HorizontalAlignment="Left">

<Border BorderBrush="#23000000" BorderThickness="1" >

<StackPanel Background="White" Width="auto" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<StackPanel Background="#FFF3F3F3">

<TextBlock Text="Overview" Foreground="#cc000000" FontWeight="SemiLight" FontSize="20" Margin="10 20 0 10"/>

<Border BorderBrush="#11000000" BorderThickness="1" />

</StackPanel>

<Grid Margin="0 10 0 0">

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="\*" />

<RowDefinition Height="\*" />

<RowDefinition Height="\*" />

<RowDefinition Height="\*" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Button x:Name="PatientPage" Grid.Row="0" Grid.Column="0" Click="PatientPage\_Click" Style="{StaticResource btn}" HorizontalAlignment="Stretch" Margin="3" VerticalAlignment="Stretch">

<StackPanel Style="{StaticResource inner\_btn}">

<TextBlock Text="Patients" Style="{StaticResource btn\_txt}" />

<StackPanel Style="{StaticResource results}">

<TextBlock x:Name="patient\_r" Style="{StaticResource result\_txt}"/>

</StackPanel>

</StackPanel>

</Button>

<Button x:Name="TestPage" Grid.Row="0" Grid.Column="1" Click="TestPage\_Click" Style="{StaticResource btn}" Margin="3">

<StackPanel Style="{StaticResource inner\_btn}">

<TextBlock Text="Tests" Style="{StaticResource btn\_txt}" />

<StackPanel Style="{StaticResource results}">

<TextBlock x:Name="test\_r" Style="{StaticResource result\_txt}"/>

</StackPanel>

</StackPanel>

</Button>

<Button x:Name="ItemPage" Grid.Row="1" Grid.Column="0" Click="ItemPage\_Click" Style="{StaticResource btn}">

<StackPanel>

<TextBlock Text="Items" Style="{StaticResource btn\_txt}" />

<StackPanel Style="{StaticResource results}">

<TextBlock x:Name="item\_r" Style="{StaticResource result\_txt}"/>

</StackPanel>

</StackPanel>

</Button>

<Button x:Name="PurchasePage" Grid.Row="1" Grid.Column="1" Click="PurchasePage\_Click" Style="{StaticResource btn}">

<StackPanel>

<TextBlock Text="Purchases" Style="{StaticResource btn\_txt}" />

<StackPanel Style="{StaticResource results}">

<TextBlock x:Name="purchase\_r" Style="{StaticResource result\_txt}"/>

</StackPanel>

</StackPanel>

</Button>

<Button x:Name="ReportPage" Grid.Row="2" Grid.Column="0" Click="ReportPage\_Click" Style="{StaticResource btn}">

<StackPanel>

<TextBlock Text="Reports" Style="{StaticResource btn\_txt}" />

<StackPanel Style="{StaticResource results}">

<TextBlock x:Name="report\_r" Style="{StaticResource result\_txt}"/>

</StackPanel>

</StackPanel>

</Button>

<Button x:Name="DoctorPage" Grid.Row="2" Grid.Column="1" Click="DoctorPage\_Click" Style="{StaticResource btn}">

<StackPanel>

<TextBlock Text="Doctors" Style="{StaticResource btn\_txt}" />

<StackPanel Style="{StaticResource results}">

<TextBlock x:Name="Doctor\_r" Style="{StaticResource result\_txt}"/>

</StackPanel>

</StackPanel>

</Button>

</Grid>

</StackPanel>

</Border>

</Border>

<!--SECOND COLUMN-->

<StackPanel Grid.Row="0" Grid.Column="1" Margin="-1 100 0 0" VerticalAlignment="Top" Orientation="Horizontal">

<StackPanel.Resources>

<Style x:Key="head" TargetType="StackPanel">

<Setter Property="Margin" Value="5 7 5 0"/>

</Style>

<Style x:Key="head\_txt" TargetType="TextBlock">

<Setter Property="Foreground" Value="#ffffff"/>

<Setter Property="TextWrapping" Value="Wrap"/>

<Setter Property="FontSize" Value="27"/>

<Setter Property="Padding" Value="10 3"/>

</Style>

<Style x:Key="head\_write\_up" TargetType="TextBlock">

<Setter Property="Foreground" Value="#ffffff"/>

<Setter Property="TextWrapping" Value="Wrap"/>

<Setter Property="FontSize" Value="16"/>

<Setter Property="Padding" Value="14 16"/>

<Setter Property="Width" Value="330"/>

</Style>

<Style x:Key="head\_border" TargetType="Border">

</Style>

<Style x:Key="pic\_sep" TargetType="StackPanel">

<Setter Property="Background" Value="#FFC3403A"/>

<Setter Property="Orientation" Value="Horizontal"/>

</Style>

<Style x:Key="pic\_img" TargetType="Image">

<Setter Property="Height" Value="50"/>

<Setter Property="Width" Value="50"/>

<Setter Property="Margin" Value="0 20 0 0"/>

<Setter Property="VerticalAlignment" Value="Center"/>

</Style>

<Style x:Key="pic\_txt" TargetType="TextBlock">

<Setter Property="Foreground" Value="#f7f7f7"/>

<Setter Property="TextWrapping" Value="Wrap"/>

<Setter Property="FontSize" Value="25"/>

<Setter Property="Padding" Value="10 3"/>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="Margin" Value="0 30 0 0"/>

</Style>

</StackPanel.Resources>

<Canvas>

<Image Source="Assets/Images/home-bg.jpg" Margin="0 -100 0 0" Height="auto" Width="auto"/>

</Canvas>

<Border BorderBrush="#13000000" BorderThickness="1" MinHeight="300" Margin="20 0 20 0" CornerRadius="4">

<StackPanel Background="#aa286CA8" MinHeight="300" Width="330" VerticalAlignment="Stretch" HorizontalAlignment="Stretch">

<!-- HEAD PART-->

<StackPanel Style="{StaticResource head}">

<TextBlock Text="Patients" Style="{StaticResource head\_txt}"/>

<Border BorderBrush="#13000000" BorderThickness="1"/>

<TextBlock Text="The list of patients is displayed in the patients page, it has all the necessary details of the patient needed, address, their fullnames, the phone number, gender , the date of birth of the patient. so when adding their test results, the info is well collected." Style="{StaticResource head\_write\_up}"/>

</StackPanel>

<Border BorderBrush="#73000000" BorderThickness="1">

<StackPanel Style="{StaticResource pic\_sep}" Background="#f0286CA8">

<StackPanel Background="#C3403A" Width="130" Height="100">

<Image Source="Assets/Icons/big-user-white.png" Style="{StaticResource pic\_img}"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Patients" Style="{StaticResource pic\_txt}"/>

</StackPanel>

</StackPanel>

</Border>

</StackPanel>

</Border>

<!---/////////////////-->

<Border BorderBrush="#23000000" BorderThickness="1">

<StackPanel Background="#aaC3403A" MinHeight="300" Width="auto">

<!-- HEAD PART-->

<StackPanel Style="{StaticResource head}">

<TextBlock Text="Records" Style="{StaticResource head\_txt}"/>

<Border BorderBrush="#13000000" BorderThickness="1"/>

<TextBlock Text="The report list is one of a kind, it has a nice display of what the report is about, who it's for, what test it belongs to, when it was added, you can also figure out when it was last edited or updated. It's easy to delete them or assign them to another test if the need be." Style="{StaticResource head\_write\_up}"/>

</StackPanel>

<Border BorderBrush="#73000000" BorderThickness="1">

<StackPanel Style="{StaticResource pic\_sep}" Background="#FFC3403A">

<StackPanel Background="#286CA8" Width="130" Height="100">

<Image Source="Assets/Icons/big-info-white.png" Style="{StaticResource pic\_img}"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Reports" Style="{StaticResource pic\_txt}"/>

</StackPanel>

</StackPanel>

</Border>

</StackPanel>

</Border>

</StackPanel>

</Grid>

<!--//BODY-->

</Grid>

</Grid>

</Page>

Home C#

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.UI;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

using Lpms.Classes;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace Lpms

{

/// <summary>

/// An empty page that can be used on its own or navigated to within a Frame.

/// </summary>

public sealed partial class Home : Page

{

Loader l;

public Home()

{

this.InitializeComponent();

//PUT IN THE NAME OF THE USER

Query q = new Query();

q.Get("staffs","WHERE staff\_id = '"+(App.Current as App).User\_token+"'");

q.Records();

Filter f = new Filter();

fullname.Text += q.RecordExist(q.Results[0, 3], "Unavailable :(");

//NOW DISPLAY the rows

q.Get("patients");

patient\_r.Text = q.CountRows().ToString();

q.Get("items");

item\_r.Text = q.CountRows().ToString();

q.Get("tests");

test\_r.Text = q.CountRows().ToString();

int c = 0;

q.Get("Reports");

q.Records();

for (int i = 0; i < q.CountRows(); i++)

{

Query d = new Query();

d.Get("tests", "WHERE test\_id = '" + q.Results[i, 3] + "'");

if (d.CheckRecords() == true)

c += 1;

}

report\_r.Text = c.ToString();

q.Get("Doctors");

Doctor\_r.Text = q.CountRows().ToString();

q.Get("purchases");

purchase\_r.Text = q.CountRows().ToString();

}

/\*

\* when the respective buttons are clicked, they each carry a token

\* to the parent page, so that the parent page knows what to load

\*/

private void PatientPage\_Click(object sender, RoutedEventArgs e)

{

l = new Loader("patients");

this.Frame.Navigate(typeof(Parent), null);

}

private void TestPage\_Click(object sender, RoutedEventArgs e)

{

l = new Loader("tests");

this.Frame.Navigate(typeof(Parent), null);

}

private void ItemPage\_Click(object sender, RoutedEventArgs e)

{

l = new Loader("items");

this.Frame.Navigate(typeof(Parent), null);

}

private void DoctorPage\_Click(object sender, RoutedEventArgs e)

{

l = new Loader("Doctors");

this.Frame.Navigate(typeof(Parent), null);

}

private void ReportPage\_Click(object sender, RoutedEventArgs e)

{

l = new Loader("reports");

this.Frame.Navigate(typeof(Parent), null);

}

private void PurchasePage\_Click(object sender, RoutedEventArgs e)

{

l = new Loader("Purchases");

this.Frame.Navigate(typeof(Parent), null);

}

private void Logout\_Click(object sender, RoutedEventArgs e)

{

Auth auth = new Auth();

auth.Logout();

this.Frame.Navigate(typeof(MainPage),null);

}

private void Profile\_Click(object sender, RoutedEventArgs e)

{

l = new Loader("Profile");

this.Frame.Navigate(typeof(Parent), null);

}

}

}

Patients XAML

<Page

x:Class="Lpms.Patients"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:Lpms"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

<Page.Resources>

<!-- TABLE HEAD BORDER -->

<Style x:Key="table\_head\_border" TargetType="Border">

<Setter Property="BorderBrush" Value="#19000000" />

<Setter Property="BorderThickness" Value="0 0 1 1"/>

<!--<Setter Property="Background" Value="#E8F3F0"/>-->

<Setter Property="Padding" Value="0"/>

<Setter Property="Margin" Value="0 0 0 0 "/>

</Style>

<!-- table head-->

<Style x:Key="table\_head" TargetType="StackPanel">

<Setter Property="MinWidth" Value="140"/>

<Setter Property="Background" Value="#E7e7e7"/>

<Setter Property="Margin" Value="0 0 0 0"/>

<Setter Property="VerticalAlignment" Value="Stretch"/>

</Style>

<!--table head text -->

<Style x:Key="table\_head\_txt" TargetType="TextBlock">

<Setter Property="Foreground" Value="#cc000000"/>

<Setter Property="Margin" Value="10 10 10 10"/>

<Setter Property="FontSize" Value="17"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

<Setter Property="VerticalAlignment" Value="Center"/>

</Style>

</Page.Resources>

<Grid>

<!--PARENT STACKPANEL SO THAT THE STUPID ELEMENTS DON'T OVERLAP-->

<ContentControl x:Name="parent">

<StackPanel AreScrollSnapPointsRegular="True">

<!-- record title-->

<StackPanel>

<Border Style="{StaticResource record\_label\_border}">

<StackPanel Style="{StaticResource record\_label}">

<TextBlock Text="Patients' Records" Style="{StaticResource record\_label\_txt}"/>

<Button x:Name="Add" Click="Add\_Click" ClickMode="Press" Style="{StaticResource add\_btn}">

<StackPanel Orientation="Horizontal">

<Image Source="Assets/Icons/plus-white.png"/>

<TextBlock Text=" Add"/>

</StackPanel>

</Button>

</StackPanel>

</Border>

</StackPanel>

<!--TABLE HEAD-->

<Grid x:Name="patients\_r" Width="auto" HorizontalAlignment="Stretch" Margin="0 0 0 0">

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

</Grid.ColumnDefinitions>

<Border Grid.Row="0" Grid.Column="0" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Name" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="1" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Date of Birth" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="2" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Address" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="3" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Gender" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="4" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Tests Performed" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="5" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Purchases done" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="6" Style="{StaticResource table\_head\_border}" BorderThickness="0 1 0 1">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="No. of reports" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="7" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head\_option}">

<TextBlock Text="Options" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

</Grid>

<!--CONTENTS-->

</StackPanel>

</ContentControl>

<Popup x:Name="Adding" IsOpen="False" Width="700" Height="auto" Margin="0 40 0 0">

<StackPanel Background="#4BA783">

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1" Width="auto" Margin="0 10 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<TextBox x:Name="p\_name" Width="auto" PlaceholderText="Patient's Name (Firstname + Lastname)" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 100 10 20"/>

<TextBox x:Name="p\_address" Width="auto" PlaceholderText="Patient's address" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<TextBox x:Name="p\_phone" Width="auto" PlaceholderText="Phone number" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<ComboBox x:Name="p\_gender" Width="auto" PlaceholderText="Patient's Gender" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select Patient's Gender</ComboBoxItem>

<ComboBoxItem IsEnabled="True">Male</ComboBoxItem>

<ComboBoxItem IsEnabled="True">Female</ComboBoxItem>

</ComboBox>

<TextBox x:Name="p\_dob" Width="auto" PlaceholderText="Date of birth (MM/DD/YYYY)" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<Button x:Name="Cancel\_Add" Click="Cancel\_Click" HorizontalAlignment="Left" Content="Cancel" FontSize="20" Width="120" Height="60" Margin="0 0 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#cc000000" Foreground="#ffffff"/>

<Button x:Name="AddRecord" Click="AddRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" Margin="0 -67 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</Popup>

<!-- EDITING POPUP-->

<Popup x:Name="Editing" IsOpen="False" Width="700" Height="auto" Margin="0 40 0 0">

<StackPanel Background="#4BA783">

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1" Width="auto" Margin="0 10 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<TextBox x:Name="p\_name\_e" Width="auto" PlaceholderText="Patient's Name (Firstname + Lastname)" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 100 10 20"/>

<TextBox x:Name="p\_address\_e" Width="auto" PlaceholderText="Patient's address" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<TextBox x:Name="p\_phone\_e" Width="auto" PlaceholderText="Phone number" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<ComboBox x:Name="p\_gender\_e" Width="auto" PlaceholderText="Patient's Gender" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select Patient's Gender</ComboBoxItem>

<ComboBoxItem IsEnabled="True">Male</ComboBoxItem>

<ComboBoxItem IsEnabled="True">Female</ComboBoxItem>

</ComboBox>

<TextBox x:Name="p\_dob\_e" Width="auto" PlaceholderText="Date of birth(MM/DD/YYYY)" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<Button x:Name="Cancel\_Edit" Click="Cancel\_Click" HorizontalAlignment="Left" Content="Cancel" FontSize="20" Width="120" Height="60" Margin="0 0 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#cc000000" Foreground="#ffffff"/>

<Button x:Name="EditRecord" Click="EditRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" Margin="0 -67 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</Popup>

</Grid>

</Page>

Patients C#

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.UI.Xaml;

using Windows.UI;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

using Lpms.Classes;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace Lpms

{

/// <summary>

/// The page which displays the patient's records

/// </summary>

public sealed partial class Patients : Page

{

//FOR EDITING

EditTracker tracker;

private string p\_id = "";

public Patients()

{

this.InitializeComponent();

//start displaying the patients records

Query q = new Query();

Query inner\_q = new Query();

q.GetExtra("SELECT name,dob,address,Gender FROM patients");

q.Records();

int patients\_row = q.CountRows();

Query patient\_id = new Query();

patient\_id.Get("Patients");

patient\_id.Records();

//AMOUNT OF COLUMN DEFINITIONS

int col\_num = Convert.ToInt32(patients\_r.ColumnDefinitions.Count);

Filter f = new Filter();

bool check\_records = q.CheckRecords();

// RowDefinition[] rd = new RowDefinition[q.CountRows()];

// RowDefinition rd;

//STACKPANEL TO HOLD THE ROWS

// StackPanel parent\_row\_panel;

//WE'RE ADDING +1 CAUSE THERE'S ALREADY A ROW DEFINED WHICH IT'S VALUE WOULD BE ZERO

//SO IF WE DONT INCREMENT IT, IT STARTS WITH 0, WHICH MIGHT BE A BAD IDEA :(.

//int i = 0;

//now check the number of rows

//keep adding rows as long as it doesn't exceed the row limit from the stupid database.

if(check\_records){

int i = 0;

for (i = 0; i < (patients\_row); i++)

{

RowDefinition rd = new RowDefinition();

//rd.Height = new GridLength(50);

patients\_r.RowDefinitions.Add(rd);

/\*

\* COLUMN DEFINITION

\* CREATE COLUMNS ACCORDING TO THE PARENT COLUMN OF THE FIRST ROW, WHICH IS col\_num.

\*/

//////////////////////////////////

int k = 0;

for (k = 0; k < col\_num; k++)

{

ColumnDefinition cd = new ColumnDefinition();

//cd.Width = new GridLength();

//cd.MinWidth = 150;

patients\_r.ColumnDefinitions.Add(cd);

//reset column def so that when it starts looping the for again, it creates a new set of column definition

//column\_def = 0;

//////////////////////////////////////////////////////////

Border border = new Border();

TextBlock n = new TextBlock();

n.FontSize = 17.3;

n.Foreground = new SolidColorBrush(Color.FromArgb(180, 0, 0, 0));

///////////////////////////////////////

//stackpanel

StackPanel pa = new StackPanel();

// pa.Style = new Style();

//StackPanel m\_pa = new StackPanel();

// m\_pa.Name = "p";// + i.ToString();

StackPanel row\_holder = new StackPanel();

//row\_holder.Orientation = new Orientation();

// Orientation orie = row\_holder1.Orientation;

//add

//pa.

pa.Children.Add(n);

pa.Margin = new Thickness(10);

//pa.MaxWidth = 115;

pa.HorizontalAlignment = HorizontalAlignment.Center;

//border.Style = new Windows.UI.Xaml.Style();

//border.Style.TargetType = typeof(Border);

border.BorderThickness = new Thickness(0, 0, 0, 1);

border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

//calculate odd and even ish

int calc = i % 2;

//border.MaxWidth = 115;

// border.MinHeight = 61;

border.Child = pa;

row\_holder.Children.Add(border);

int calcu = col\_num - k;

//CHECK WHAT TO INPUT

if (k == 1)

{

n.Text = f.Date(q.Results[i, k], "dd/MM/yyyy");

}

else if (k == 3)

{

n.Text = f.Gender(q.Results[i, k]);

}

else if (k == 4)

{//tests results

inner\_q.Get("tests", "WHERE patient\_id = '" + patient\_id.Results[i, 0] + "'");

n.Text = f.Numbers(inner\_q.CountRows());

}

else if (k == 5)

{//purchases

inner\_q.Get("purchases", "Where patient\_id = '" + patient\_id.Results[i, 0] + "'");

n.Text = f.Numbers(inner\_q.CountRows());

}

else if (k == 6)

{//NO OF REPORTS

//report is related to test\_id, so we have to get the test id results first, before, we know the patient

inner\_q.Get("tests", "Where patient\_id = '" + patient\_id.Results[i, 0] + "'");

int tests\_no = inner\_q.CountRows();

//hold the value of reports gotten

int report\_count = 0;

int report\_only = 0;

int l = 0;

//LOOP THROUGH THE QUERY

inner\_q.Records();

for (l = 0; l < tests\_no; l++)

{

//get the number of reports for the tests returned

inner\_q.Get("reports", "Where test\_id = '" + inner\_q.Results[l, 0] + "'");

//n.Text = q.CountRows().ToString();

if(inner\_q.CheckRecords() == true)

report\_only += inner\_q.CountRows();

}

//HOW MANY TIMES THE LOOP EXCECUTED SHOWS THE NUMBER OF TESTS VALID FOR THAT PATIENT

report\_count = report\_only;

n.Text = f.Numbers(report\_count);

}

else if (calcu == 1)

{//LAST VALUE

}

else

{

n.Text = q.Results[i, k];

}

//TextBlock

//////////////OPTION

StackPanel option\_panel = new StackPanel();

Border option\_border = new Border();

//////////////////////////////////////////////////////

////FOR CHANGING THE ROW BACKGROUND.

if (calc == 1)

{

border.Background = new SolidColorBrush(Color.FromArgb(250, 215, 215, 215));

option\_border.Background = new SolidColorBrush(Color.FromArgb(250, 215, 215, 215));

}

else

{

border.Background = new SolidColorBrush(Color.FromArgb(250, 236, 254, 251));

option\_border.Background = new SolidColorBrush(Color.FromArgb(250, 226, 246, 241));

}

//DETERMINE WHEN TO ADD THE OPTION STUFF

if (calcu == 1)

{//CREATE THE OPTIONS STUFF

StackPanel option\_inner\_panel = new StackPanel();

//TO MAKE THINGS EASY JARE,LOOP IT THROUGH THE ARRAY

string[] stuff\_text = { "Edit", "Delete" };

string[] stuff\_image = { "", "" };

HyperlinkButton[,] link = new HyperlinkButton[patients\_row, stuff\_text.Length];

int reduce = i;

if (i == patients\_row)

{

reduce = i - 1;

}

for (int t = 0; t < stuff\_text.Length; t++)

{

option\_border.BorderThickness = new Thickness(0, 0, 0, 1);

option\_border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

TextBlock option\_txt = new TextBlock();

option\_txt.Text = stuff\_text[t];

option\_txt.Foreground = new SolidColorBrush(Color.FromArgb(150, 0, 0, 0));

option\_txt.FontSize = 15;

option\_txt.Margin = new Thickness(10, 0 ,0 ,0);

option\_txt.VerticalAlignment = VerticalAlignment.Center;

// Button button = new Button();

link[i, t] = new HyperlinkButton();

//HyperlinkButton[] link = new HyperlinkButton[tests\_row];

link[i, t].Content = option\_txt.Text;

// link.Click = RoutedEventHandler.Combine(sender, RoutedEventArgs e);

//button.ClickMode = ClickMode.Hover;

//CLICK EVENT

int reduce\_t = t;

if (t == stuff\_text.Length)

{

reduce\_t = t - 1;

}

//link[i].Click += new RoutedEventHandler(EditRecord\_Click("E",));

link[i, t].Click += (s, e) =>

{

if ((string)link[reduce, reduce\_t].Content == stuff\_text[0])

{//IT's edit

if (!Editing.IsOpen)

{

//OPEN

this.Frame.Opacity = .4;

//this.Opacity = .4;

parent.IsEnabled = false;

//SET THE EDIT TRACKER

//I DID (i-1) BECAUSE IT WAS INCREASE BY ONE, THEREFORE OUT OF BOUND ISH.

tracker = new EditTracker("Patients", "patient\_id", patient\_id.Results[(reduce), 0]);

//track the row.

tracker.Track();

this.p\_id = tracker.TrackResult[0];//set the id to the global var, so that we know what to edit.

p\_name\_e.Text = tracker.TrackResult[1];

p\_address\_e.Text = tracker.TrackResult[2];

p\_phone\_e.Text = tracker.TrackResult[3];

p\_dob\_e.Text = f.Date(tracker.TrackResult[5],"MM/dd/yyyy");

//CHECK GENDER

if (tracker.TrackResult[4].ToLower() == "true")

{//male

p\_gender\_e.SelectedIndex = 1;

//p\_gender\_e.SelectedItem = "Male";

}

else if (tracker.TrackResult[4].ToLower() == "false")

{//female

p\_gender\_e.SelectedIndex = 2;

}

Editing.IsOpen = true;

}

}

else if ((string)link[reduce, reduce\_t].Content == stuff\_text[1])

{//it's the delete.

//CALL THE DELETE QUERY

//track first

tracker = new EditTracker("Patients", "patient\_id", patient\_id.Results[(reduce), 0]);

//track

tracker.Track();

//CALL THE DELETE QUERY

Query que = new Query();

que.successMessage = "Patient " + f.FirstName(tracker.TrackResult[1]) + " has been deleted successfully.";

que.Remove("patients", "where patient\_id = '" + tracker.TrackResult[0] + "'");

this.Frame.Navigate(typeof(Patients), null);

}

};

////ADDING

option\_inner\_panel.Children.Add(link[i, t]);

}

option\_panel.Children.Add(option\_inner\_panel);

option\_inner\_panel.Orientation = Orientation.Horizontal;

option\_panel.VerticalAlignment = VerticalAlignment.Center;

option\_border.Child = option\_panel;

Grid.SetColumn(option\_border, k);

Grid.SetRow(option\_border, i + 1);

patients\_r.Children.Add(option\_border);

}

else

{

Grid.SetRow(row\_holder, i + 1);

Grid.SetColumn(row\_holder, k);

patients\_r.Children.Add(row\_holder);

}

}

}

}

else{

//

Messages msg = new Messages();

msg.Note("No Patients Record Available.");

patients\_r.Children.Add(msg.\_Container);

RowDefinition rd = new RowDefinition();

// rd.Height = new GridLength(50);

patients\_r.RowDefinitions.Add(rd);

Grid.SetColumnSpan(msg.\_Container, col\_num);

Grid.SetColumn(msg.\_Container, 0);

Grid.SetRow(msg.\_Container, 1);

}

}

//ADD BTN

private void Add\_Click(object sender, RoutedEventArgs e)

{

if (!Adding.IsOpen)

{

//OPEN

// (App.Current as App).ParentOpacity = .4;

this.Frame.Opacity = .4;

//this.Opacity = .4;

parent.IsEnabled = false;

Adding.IsOpen = true;

}

}

//Cancel BTN

private void Cancel\_Click(object sender, RoutedEventArgs e)

{

if (Adding.IsOpen)

{

//Close

// (App.Current as App).ParentOpacity = .4;

this.Frame.Opacity = 1;

//this.Opacity = .4;

parent.IsEnabled = true;

Adding.IsOpen = false;

}

if(Editing.IsOpen){

this.Frame.Opacity = 1;

//this.Opacity = .4;

parent.IsEnabled = true;

Editing.IsOpen = false;

}

}

//add record click btn

private void AddRecord\_Click(object sender, RoutedEventArgs e)

{

//CHECK EVERYTHING

PopUp p;

Filter f = new Filter();

string patient\_name = f.Trim(p\_name.Text);

string patient\_phone = f.Trim(p\_phone.Text);

string patient\_address = f.Trim(p\_address.Text);

string patient\_dob = f.Trim(p\_dob.Text);

string patient\_gender = f.Trim(p\_gender.SelectedIndex.ToString());

if (!(patient\_name == "" || patient\_address == "" || patient\_phone == "" || patient\_dob == "" || (patient\_gender == "" || patient\_gender == "0")))

{

//GOOD TO GO

//check for phone number

Query q = new Query();

if (patient\_gender == "1")

{//male

patient\_gender = "1";

}

else if(patient\_gender == "2")

{//female

patient\_gender = "0";

}

patient\_dob = f.Date(patient\_dob, "yyyy-MM-dd");

string[] val = { patient\_name, patient\_address, patient\_phone, patient\_gender, patient\_dob, "Now()" };

q.successMessage = patient\_name+"'s record has been added successfully!";

q.Add("patients", val, "name,address,phone,gender,dob,date");

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Patients), null);

}

else

{

if (patient\_name == "")

{

p = new PopUp("Please enter the name of the patient");

}

else if (patient\_address == "")

{

p = new PopUp("Please enter the patient's address");

}

else if (patient\_phone == "")

{

p = new PopUp("Please enter the patient's phone number");

}

else if (patient\_dob == "")

{

p = new PopUp("Please enter the patient's date of birth");

}

else if (patient\_gender == "" || patient\_gender == "0")

{

p = new PopUp("Please select the patient's gender");

}

else

{

p = new PopUp("Sorry, an error occured. The patient couldn't be added. Please try again later");

}

}

}

//FOR EDITING

private void EditRecord\_Click(object sender, RoutedEventArgs e)

{

// tracker.Track();

// t\_price.Text = tracker.TrackResult.Length.ToString();

//CHECK EVERYTHING

PopUp p;

Filter f = new Filter();

string patient\_name = f.Trim(p\_name\_e.Text);

string patient\_phone = f.Trim(p\_phone\_e.Text);

string patient\_address = f.Trim(p\_address\_e.Text);

string patient\_dob = f.Trim(p\_dob\_e.Text);

string patient\_gender = f.Trim(p\_gender\_e.SelectedIndex.ToString());

if (!(patient\_name == "" || patient\_address == "" || patient\_phone == "" || patient\_dob == "" || (patient\_gender == "" || patient\_gender == "0")))

{

//GOOD TO GO

Query q = new Query();

if (patient\_gender == "1")

{//male

patient\_gender = "1";

}

else if (patient\_gender == "2")

{//female

patient\_gender = "0";

}

patient\_dob = f.Date(patient\_dob, "yyyy-MM-dd");

string[] val = { patient\_name, patient\_address, patient\_phone, patient\_gender,patient\_dob};

string[] col = { "name","address", "phone", "gender","dob"};

q.Get("Patients", "Where patient\_id = '" + p\_id + "'");

q.Records();

q.successMessage = f.FirstName(q.Results[0,1])+"'s record has been edited successfully!";

//MAKE THINGS EASIER JARE, LOOP THROUGH THE VAL, TO INSERT DYNAMICALLY

//I CANNOT COME AND BE WOUNDING MYSELF :-|

string condition = " Where patient\_id = '" + this.p\_id + "'";

q.Change("Patients","name,address,phone,gender,dob",val, condition);

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Patients), null);

}

else

{

if (patient\_name == "")

{

p = new PopUp("Please enter the name of the patient");

}

else if (patient\_address == "")

{

p = new PopUp("Please enter the patient's address");

}

else if (patient\_phone == "")

{

p = new PopUp("Please enter the patient's phone number");

}

else if (patient\_dob == "")

{

p = new PopUp("Please enter the patient's date of birth");

}

else if (patient\_gender == "" || patient\_gender == "0")

{

p = new PopUp("Please select the patient's gender");

}

else

{

p = new PopUp("Sorry, an error occured. The patient couldn't be added. Please try again later");

}

}

}

}

}

Tests XAML

<Page

x:Class="Lpms.Tests"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:Lpms"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

<Page.Resources>

<!-- TABLE HEAD BORDER -->

<Style x:Key="table\_head\_border" TargetType="Border">

<Setter Property="BorderBrush" Value="#19000000" />

<Setter Property="BorderThickness" Value="0 0 1 1"/>

<!--<Setter Property="Background" Value="#E8F3F0"/>-->

<Setter Property="Padding" Value="0"/>

<Setter Property="Margin" Value="0 0 0 0 "/>

</Style>

<!-- table head-->

<Style x:Key="table\_head" TargetType="StackPanel">

<Setter Property="MinWidth" Value="140"/>

<Setter Property="Background" Value="#E7e7e7"/>

<Setter Property="Margin" Value="0 0 0 0"/>

<Setter Property="VerticalAlignment" Value="Stretch"/>

</Style>

<!--table head text -->

<Style x:Key="table\_head\_txt" TargetType="TextBlock">

<Setter Property="Foreground" Value="#cc000000"/>

<Setter Property="Margin" Value="10 10 10 10"/>

<Setter Property="FontSize" Value="17"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

<Setter Property="VerticalAlignment" Value="Center"/>

</Style>

</Page.Resources>

<Grid>

<!--PARENT STACKPANEL SO THAT THE STUPID ELEMENTS DON'T OVERLAP-->

<StackPanel>

<ContentControl x:Name="parent">

<StackPanel>

<!-- record title-->

<StackPanel>

<Border Style="{StaticResource record\_label\_border}">

<StackPanel Style="{StaticResource record\_label}" Orientation="Vertical">

<TextBlock Text="Tests' Records" Style="{StaticResource record\_label\_txt}"/>

<Button x:Name="Add" Click="Add\_Click" ClickMode="Press" Style="{StaticResource add\_btn}">

<StackPanel Orientation="Horizontal">

<Image Source="Assets/Icons/plus-white.png"/>

<TextBlock Text=" Add"/>

</StackPanel>

</Button>

</StackPanel>

</Border>

</StackPanel>

<!--TABLE HEAD-->

<Grid x:Name="tests\_r" Width="auto" HorizontalAlignment="Stretch" Margin="0 0 0 0">

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

</Grid.ColumnDefinitions>

<Border Grid.Row="0" Grid.Column="0" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Name" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="1" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Price" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="2" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Patient" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="3" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="No of reports" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="4" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Doctor" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="5" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Date" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="6" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Options" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

</Grid>

</StackPanel>

</ContentControl>

<!--//end of records-->

</StackPanel>

<Popup x:Name="Adding" IsOpen="False" Width="700" Height="auto" Margin="0 80 0 0">

<StackPanel Background="#4BA783">

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1" Width="auto" Margin="0 10 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<TextBox x:Name="t\_name" Width="auto" PlaceholderText="Test Name" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 100 10 20"/>

<ComboBox x:Name="t\_patients" Width="auto" PlaceholderText="Test belongs to which patient" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select a Patient</ComboBoxItem>

</ComboBox>

<TextBox x:Name="t\_price" Width="auto" PlaceholderText="Price" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<ComboBox x:Name="t\_doctors" Width="auto" PlaceholderText="Doctor performing the test" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select a Doctor</ComboBoxItem>

</ComboBox>

<!-- <ComboBox x:Name="t\_doctors" ItemsSource="{Binding q.Results}" Width="auto" PlaceholderText="Doctor performing the test" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40">

<ComboBox.ItemTemplate>

<DataTemplate>

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select a Doctor</ComboBoxItem>

</DataTemplate>

</ComboBox.ItemTemplate>

</ComboBox> -->

<!--<Button Content="Price"

Background="ForestGreen" VerticalAlignment="Bottom" HorizontalAlignment="Center" Margin="309,0,324,35">

<Button.Flyout>

<MenuFlyout>

<MenuFlyoutItem Text="#4,000" />

<MenuFlyoutSeparator/>

<ToggleMenuFlyoutItem Text="#300" IsChecked="{Binding IsShuffleEnable, Mode=TwoWay}"/>

<ToggleMenuFlyoutItem Text="#1,000" IsChecked="{Binding IsShuffleEnable, Mode=TwoWay}"/>

</MenuFlyout>

</Button.Flyout>

</Button>-->

<Button x:Name="Cancel\_Add" Click="Cancel\_Click" HorizontalAlignment="Left" Content="Cancel" FontSize="20" Width="120" Height="60" Margin="0 0 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#cc000000" Foreground="#ffffff"/>

<Button x:Name="AddRecord" Click="AddRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" Margin="0 -67 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</Popup>

<!-- EDITING POPUP-->

<Popup x:Name="Editing" IsOpen="False" Width="700" Height="auto" Margin="0 80 0 0">

<StackPanel Background="#4BA783">

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1" Width="auto" Margin="0 10 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<TextBox x:Name="t\_name\_e" Width="auto" PlaceholderText="Test Name" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 100 10 20"/>

<ComboBox x:Name="t\_patients\_e" Width="auto" PlaceholderText="Test belongs to which patient" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select a Patient</ComboBoxItem>

</ComboBox>

<TextBox x:Name="t\_price\_e" Width="auto" PlaceholderText="Price" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<ComboBox x:Name="t\_doctors\_e" Width="auto" PlaceholderText="Doctor performing the test" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select a Doctor</ComboBoxItem>

</ComboBox>

<!--<Button Content="Price"

Background="ForestGreen" VerticalAlignment="Bottom" HorizontalAlignment="Center" Margin="309,0,324,35">

<Button.Flyout>

<MenuFlyout>

<MenuFlyoutItem Text="#4,000" />

<MenuFlyoutSeparator/>

<ToggleMenuFlyoutItem Text="#300" IsChecked="{Binding IsShuffleEnable, Mode=TwoWay}"/>

<ToggleMenuFlyoutItem Text="#1,000" IsChecked="{Binding IsShuffleEnable, Mode=TwoWay}"/>

</MenuFlyout>

</Button.Flyout>

</Button>-->

<Button x:Name="Cancel\_edit" Click="Cancel\_Click" HorizontalAlignment="Left" Content="Cancel" FontSize="20" Width="120" Height="60" Margin="0 0 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#cc000000" Foreground="#ffffff"/>

<Button x:Name="EditRecord" Click="EditRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" Margin="0 -67 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</Popup>

</Grid>

</Page>

Tests C#

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.UI.Xaml;

using Windows.UI;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

using Lpms.Classes;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace Lpms

{

/// <summary>

/// An empty page that can be used on its own or navigated to within a Frame.

/// </summary>

public sealed partial class Tests : Page

{

private string t\_id = "";

public Tests()

{

this.InitializeComponent();

//FOR EDITING

EditTracker tracker;

//start displaying the tests records

Query q = new Query();

q.GetExtra("SELECT test\_name,amount,patient\_id,doctor\_id,date FROM tests");

q.Records();

////

Query test\_id = new Query();

test\_id.GetExtra("SELECT test\_id,doctor\_id from tests");

//GET THE RECORDS

test\_id.Records();

///////////////////

int tests\_row = q.CountRows();

//AMOUNT OF COLUMN DEFINITIONS

int col\_num = Convert.ToInt32(tests\_r.ColumnDefinitions.Count);

Filter f = new Filter();

bool check\_records = q.CheckRecords();

// RowDefinition[] rd = new RowDefinition[q.CountRows()];

// RowDefinition rd;

//STACKPANEL TO HOLD THE ROWS

// StackPanel parent\_row\_panel;

//WE'RE ADDING +1 CAUSE THERE'S ALREADY A ROW DEFINED WHICH IT'S VALUE WOULD BE ZERO

//SO IF WE DONT INCREMENT IT, IT STARTS WITH 0, WHICH MIGHT BE A BAD IDEA :(.

//int i = 0;

//now check the number of rows

//keep adding rows as long as it doesn't exceed the row limit from the stupid database.

int i = 0;

//CHECK IF THERE'S ANY RECORD

if (check\_records) {

for (i = 0; i < (tests\_row); i++)

{

RowDefinition rd = new RowDefinition();

// rd.Height = new GridLength(50);

tests\_r.RowDefinitions.Add(rd);

/\*++

\* COLUMN DEFINITION

\* CREATE COLUMNS ACCORDING TO THE PARENT COLUMN OF THE FIRST ROW, WHICH IS col\_num.

\*/

//////////////////////////////////

int k = 0;

int ke = 1;

for (k = 0; k < col\_num; k++)

{

ColumnDefinition cd = new ColumnDefinition();

//cd.Width = new GridLength();

//cd.MinWidth = 150;

tests\_r.ColumnDefinitions.Add(cd);

//reset column def so that when it starts looping the for again, it creates a new set of column definition

//column\_def = 0;

//calculate odd and even ish

int calc = i % 2;

//////////////////////////////////////////////////////////

Border border = new Border();

TextBlock n = new TextBlock();

n.FontSize = 17.3;

n.Foreground = new SolidColorBrush(Color.FromArgb(180, 0, 0, 0));

//TextBlock

///////////////////////////////////////

//stackpanel

StackPanel pa = new StackPanel();

// pa.Style = new Style();

//StackPanel m\_pa = new StackPanel();

// m\_pa.Name = "p";// + i.ToString();

StackPanel row\_holder = new StackPanel();

//add

//pa.

pa.Children.Add(n);

pa.Margin = new Thickness(10);

//pa.MaxWidth = 115;

pa.HorizontalAlignment = HorizontalAlignment.Center;

pa.VerticalAlignment = VerticalAlignment.Center;

border.BorderThickness = new Thickness(0, 0, 0, 1);

border.BorderBrush = new SolidColorBrush(Color.FromArgb(50,0,0, 0));

//border.MaxWidth = 115;

//border.MinHeight = 60.5;

border.Child = pa;

row\_holder.Children.Add(border);

//CHECK WHAT TO INPUT

Query c = new Query();

//check if it's the last column

int calcu = (col\_num - k);

if (k == 2)

{

//GET THE PATIENT'S NAME

c.Get("patients", "WHERE patient\_id = '" + q.Results[i, k] + "'");

//check if the patient still exists, if not, no record available

c.Records();

n.Text = c.RecordExist(c.Results[0, 1]);

}

else if (k == 3)

{//get the amount of reports

c.Get("reports", "WHERE test\_id = '" + test\_id.Results[i, 0] + "'");

n.Text = c.CountRows().ToString();

}

else if (k == 4)

{//get the doctor who performed the test

c.Get("doctors", "Where doctor\_id = '" + test\_id.Results[i, 1] + "'");

c.CheckRecords();

c.Records();

n.Text = c.RecordExist(c.Results[0, 1]);

}

else if (k == 5)

{//DATE

n.Text = f.Date(q.Results[i, 4], "dd-MM-yy");

}

else if (k == 6)

{

}

else

{

n.Text = q.Results[i, k];

}

//////////////OPTION

StackPanel option\_panel = new StackPanel();

Border option\_border = new Border();

//////////////////////////////////////////////////////

////FOR CHANGING THE ROW BACKGROUND.

if (calc == 1)

{

border.Background = new SolidColorBrush(Color.FromArgb(250, 225, 225, 225));

option\_border.Background = new SolidColorBrush(Color.FromArgb(250, 215, 215, 215));

}

else

{

border.Background = new SolidColorBrush(Color.FromArgb(250, 236, 254, 251));

option\_border.Background = new SolidColorBrush(Color.FromArgb(250, 226, 246, 241));

}

//DETERMINE WHEN TO ADD THE OPTION STUFF

if (calcu == 1)

{//CREATE THE OPTIONS STUFF

StackPanel option\_inner\_panel = new StackPanel();

//TO MAKE THINGS EASY JARE,LOOP IT THROUGH THE ARRAY

string ed = "Edit";

string del = "Delete";

string[] stuff\_text = { ed, del };

string[] stuff\_image = { "", "" };

HyperlinkButton[,] link = new HyperlinkButton[tests\_row,stuff\_text.Length];

int reduce = i;

if (i == tests\_row)

{

reduce = i - 1;

}

for (int t = 0; t < stuff\_text.Length; t++)

{

option\_border.BorderThickness = new Thickness(0, 0, 0, 1);

option\_border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

TextBlock option\_txt = new TextBlock();

option\_txt.Text = stuff\_text[t];

option\_txt.Foreground = new SolidColorBrush(Color.FromArgb(150, 0, 0, 0));

option\_txt.FontSize = 15;

option\_txt.Margin = new Thickness(10,0,3,0);

// Button button = new Button();

link[i,t] = new HyperlinkButton();

//HyperlinkButton[] link = new HyperlinkButton[tests\_row];

link[i,t].Content = option\_txt.Text;

// link.Click = RoutedEventHandler.Combine(sender, RoutedEventArgs e);

// button.ClickMode = ClickMode.Hover;

//CLICK EVENT

int reduce\_t = t;

if (t == stuff\_text.Length)

{

reduce\_t = t - 1;

}

//link[i].Click += new RoutedEventHandler(EditRecord\_Click("E",));

link[i,t].Click += (s,e) =>{

if ((string)link[reduce,reduce\_t].Content == ed)

{//IT's edit

if (!Editing.IsOpen)

{

//OPEN

this.Frame.Opacity = .4;

//this.Opacity = .4;

parent.IsEnabled = false;

//SET THE EDIT TRACKER

//I DID (i-1) BECAUSE IT WAS INCREASE BY ONE, THEREFORE OUT OF BOUND ISH.

tracker = new EditTracker("Tests", "test\_id", test\_id.Results[(reduce),0]);

//track the row.

tracker.Track();

this.t\_id = tracker.TrackResult[0];//set the id to the global var, so that we know what to edit.

t\_name\_e.Text = tracker.TrackResult[1];

t\_price\_e.Text = tracker.TrackResult[2];

//get values for the combo boxes.

c.Get("patients", "Where patient\_id = '" + q.Results[reduce, 2] + "'");

c.Records();

t\_patients\_e.SelectedIndex = f.SelectedInverse(c.Results[0, 0], "patients", null, 1);

c.Get("doctors", "Where doctor\_id = '" + q.Results[reduce, 3] + "'");

c.Records();

t\_doctors\_e.SelectedIndex = f.SelectedInverse(c.Results[0, 0], "doctors", null, 1);

Editing.IsOpen = true;

}

}

else if ((string)link[reduce,reduce\_t].Content == del)

{//it's the delete.

//track first

tracker = new EditTracker("Tests","test\_id", test\_id.Results[(reduce),0]);

//track

tracker.Track();

//CALL THE DELETE QUERY

Query que = new Query();

que.successMessage = "Test "+tracker.TrackResult[1]+" has been deleted.";

que.Remove("Tests","where test\_id = '"+tracker.TrackResult[0]+"'");

this.Frame.Navigate(typeof(Tests),null);

}

};

//private void R\_C(object sender, EventArgs e){

//}

////ADDING

option\_inner\_panel.Children.Add(link[i,t]);

//ADD THE EDIT, DELETE ICON

//Image img = new Image();

//img.Source = new ;

}

option\_panel.Children.Add(option\_inner\_panel);

option\_inner\_panel.Orientation = Orientation.Horizontal;

option\_panel.VerticalAlignment = VerticalAlignment.Center;

option\_border.Child = option\_panel;

Grid.SetColumn(option\_border, k);

Grid.SetRow(option\_border, i+1);

tests\_r.Children.Add(option\_border);

}

else

{

Grid.SetRow(row\_holder, i + 1);

Grid.SetColumn(row\_holder, k);

tests\_r.Children.Add(row\_holder);

}

///i've even forgotten what this does. :(

ke++;

}

}

}

else

{//NO RECORD, SHOW SOMETHING

Messages msg = new Messages();

msg.Note("No Tests record available.");

tests\_r.Children.Add(msg.\_Container);

RowDefinition rd = new RowDefinition();

// rd.Height = new GridLength(50);

tests\_r.RowDefinitions.Add(rd);

Grid.SetColumnSpan(msg.\_Container, col\_num);

Grid.SetColumn(msg.\_Container, 0);

Grid.SetRow(msg.\_Container, 1);

}

////////////////////////////////////////////////

//fill some stuff in the form

Query query = new Query();

query.Get("Patients");

int row\_count = query.CountRows();

query.Records();

//ComboBox pp = t\_patients;

// object[] selected\_value = new object[row\_count];

//string[] selected\_value\_path = new string[row\_count];

//t\_patients.SelectedValue = new object();

//Object[] p = new object[que.CountRows()];

for (int r = 0; r < row\_count; r++)

{

object p = query.Results[r, 1];

// t\_patients.Items.Add(p[r]);

t\_patients.Items.Add(p);

// t\_patients.SelectedValuePath = new string[row\_count];

// t\_patients.SelectedValue = new object[que.CountRows()];

// t\_patients.SelectedValue[r] = "s";

// t\_patients.SelectedValuePath = query.Results[r, 1];

// t\_patients.SelectedValue = t\_patients.SelectedValuePath;

// selected\_value[r] = t\_patients.SelectedValuePath;

//editing box

t\_patients\_e.Items.Add(p);

}

query.Get("Doctors");

query.Records();

for (int r = 0; r < query.CountRows(); r++)

{

Object d = query.Results[r, 1];

t\_doctors.Items.Add(d);

//editing box

t\_doctors\_e.Items.Add(d);

// t\_doctors.SelectedIndex

}

}

//ADD BTN

private void Add\_Click(object sender, RoutedEventArgs e)

{

if (!Adding.IsOpen)

{

//OPEN

// (App.Current as App).ParentOpacity = .4;

this.Frame.Opacity = .4;

//this.Opacity = .4;

parent.IsEnabled = false;

Adding.IsOpen = true;

}

}

//Cancel BTN

private void Cancel\_Click(object sender, RoutedEventArgs e)

{

if (Adding.IsOpen)

{

//Close

// (App.Current as App).ParentOpacity = .4;

this.Frame.Opacity = 1;

//this.Opacity = .4;

parent.IsEnabled = true;

Adding.IsOpen = false;

}

if (Editing.IsOpen)

{

//Close

this.Frame.Opacity = 1;

//this.Opacity = .4;

parent.IsEnabled = true;

Editing.IsOpen = false;

}

}

//add record click btn

private void AddRecord\_Click(object sender, RoutedEventArgs e)

{

//CHECK EVERYTHING

PopUp p;

Filter f = new Filter();

string test\_name = f.Trim(t\_name.Text);

string test\_price = f.Trim(t\_price.Text);

string test\_patient = f.Trim(t\_patients.SelectedIndex.ToString());

string test\_doctor = f.Trim(t\_doctors.SelectedIndex.ToString());

if (!(test\_name == "" || f.Trim(test\_price) == "" || (test\_patient == "" || test\_patient == "0") || (test\_doctor == "" || test\_doctor == "0")))

{

//GOOD TO GO

string[] t\_p = f.Selected(t\_patients.SelectedIndex, "Patients", null, 1);//.CopyTo(t, 0);

string[] t\_d = f.Selected(t\_patients.SelectedIndex, "Doctors", null, 1);//.CopyTo(t, 0);

Query q = new Query();

test\_patient = t\_p[0];

test\_doctor = t\_d[0];

string[] val = { test\_name, test\_price, test\_patient, test\_doctor, "Now()" };

q.successMessage = "Test record has been added successfully!";

q.Add("tests", val, "test\_name,amount,patient\_id,doctor\_id,date");

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Tests), null);

}

else

{

if (test\_name == "")

{

p = new PopUp("Please enter the test name");

}

else if (test\_price == "")

{

p = new PopUp("Please enter a Price for the test");

}

else if (test\_patient == "" || test\_patient == "0")

{

p = new PopUp("Please select a patient");

}

else if (test\_doctor == "" || test\_doctor == "0")

{

p = new PopUp("Please select a doctor");

}

else

{

p = new PopUp("Sorry, an error occured. The test couldn't be added. Please try again later");

}

}

}

//FOR EDITING

private void EditRecord\_Click(object sender, RoutedEventArgs e)

{

// tracker.Track();

// t\_price.Text = tracker.TrackResult.Length.ToString();

//CHECK EVERYTHING

PopUp p;

Filter f = new Filter();

string test\_name = f.Trim(t\_name\_e.Text);

string test\_price = f.Trim(t\_price\_e.Text);

// string[] t = f.Selected("0", "Tests", "test\_id");

string test\_patient = f.Trim(t\_patients\_e.SelectedIndex.ToString());

string test\_doctor = f.Trim(t\_doctors\_e.SelectedIndex.ToString());

if (!(test\_name == "" || test\_price == "" || (test\_patient == "" || test\_patient == "0") || (test\_doctor == "" || test\_doctor == "0")))

{

//GOOD TO GO

string[] t\_p = f.Selected(t\_patients\_e.SelectedIndex, "Patients", null, 1);//.CopyTo(t, 0);

string[] t\_d = f.Selected(t\_doctors\_e.SelectedIndex, "Doctors", null, 1);//.CopyTo(t, 0);

Query q = new Query();

q.GetExtra("SELECT test\_name from tests WHERE test\_id = '"+t\_id+"'");

q.Record();

test\_patient = t\_p[0];

test\_doctor = t\_d[0];

string[] val = { test\_name, test\_price, test\_patient, test\_doctor, "Now()" };

q.successMessage = "Test ("+q.Result[0]+")'s record has been updated successfully!";

q.Change("tests","test\_name,amount,patient\_id,doctor\_id,date\_time\_updated",val,"WHERE test\_id = '"+t\_id+"'");

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Tests), null);

}

else

{

if (test\_name == "")

{

p = new PopUp("Please enter the test name");

}

else if (test\_price == "")

{

p = new PopUp("Please enter a Price for the test");

}

else if (test\_patient == "" || test\_patient == "0")

{

p = new PopUp("Please select a patient");

}

else if (test\_doctor == "" || test\_doctor == "0")

{

p = new PopUp("Please select a doctor");

}

else

{

p = new PopUp("Sorry, an error occured. The test couldn't be added. Please try again later");

}

}

}

}

}

Items XAML

<Page

x:Class="Lpms.Items"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:Lpms"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

<Grid>

<!--PARENT STACKPANEL SO THAT THE STUPID ELEMENTS DON'T OVERLAP-->

<ContentControl x:Name="parent">

<StackPanel>

<!-- record title-->

<StackPanel>

<Border Style="{StaticResource record\_label\_border}">

<StackPanel Style="{StaticResource record\_label}">

<TextBlock Text="Item Records" Style="{StaticResource record\_label\_txt}"/>

<Button x:Name="Add" Click="Add\_Click" ClickMode="Press" Style="{StaticResource add\_btn}">

<StackPanel Orientation="Horizontal">

<Image Source="Assets/Icons/plus-white.png"/>

<TextBlock Text=" Add"/>

</StackPanel>

</Button>

</StackPanel>

</Border>

</StackPanel>

<!--TABLE HEAD-->

<Grid x:Name="items\_r" Width="auto" HorizontalAlignment="Stretch" Margin="0 0 0 0">

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

</Grid.ColumnDefinitions>

<Border Grid.Row="0" Grid.Column="0" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Name of Item" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="1" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Price" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="2" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Quantity Left" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="3" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Quantity Consumed" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="4" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Date added" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="5" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head\_option}">

<TextBlock Text="Options" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

</Grid>

<!--CONTENTS-->

</StackPanel>

</ContentControl>

<Popup x:Name="Adding" IsOpen="False" Width="700" Height="auto" Margin="0 40 0 0">

<StackPanel Background="#4BA783">

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1" Width="auto" Margin="0 10 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<TextBox x:Name="i\_name" Width="auto" PlaceholderText="Item Name" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 100 10 20"/>

<TextBox x:Name="i\_cost" Width="auto" PlaceholderText="How much is this item" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<TextBox x:Name="i\_quantity" Width="auto" PlaceholderText="The quantity of the item" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<Button x:Name="Cancel\_Add" Click="Cancel\_Click" HorizontalAlignment="Left" Content="Cancel" FontSize="20" Width="120" Height="60" Margin="0 0 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#cc000000" Foreground="#ffffff"/>

<Button x:Name="AddRecord" Click="AddRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" Margin="0 -67 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</Popup>

<!-- EDITING POPUP-->

<Popup x:Name="Editing" IsOpen="False" Width="700" Height="auto" Margin="0 40 0 0">

<StackPanel Background="#4BA783">

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1" Width="auto" Margin="0 10 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<TextBox x:Name="i\_name\_e" Width="auto" PlaceholderText="Item Name" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 100 10 20"/>

<TextBox x:Name="i\_cost\_e" Width="auto" PlaceholderText="How much is this item" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<TextBox x:Name="i\_quantity\_e" Width="auto" PlaceholderText="The quantity of the item" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<Button x:Name="Cancel\_Edit" Click="Cancel\_Click" HorizontalAlignment="Left" Content="Cancel" FontSize="20" Width="120" Height="60" Margin="0 0 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#cc000000" Foreground="#ffffff"/>

<Button x:Name="EditRecord" Click="EditRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" Margin="0 -67 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</Popup>

</Grid>

</Page>

Items C#

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.UI;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

using Lpms.Classes;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace Lpms

{

/// <summary>

/// The items page

/// </summary>

public sealed partial class Items : Page

{

//EDITING DATA

EditTracker tracker;

private string i\_id = "";

public Items()

{

this.InitializeComponent();

//start displaying the items records

Query q = new Query();

q.GetExtra("SELECT item\_name,item\_cost,quantity\_left,quantity\_consumed,date\_time FROM items");

q.Records();

int items\_row = q.CountRows();

Query item\_id = new Query();

item\_id.Get("items");

item\_id.Records();

//AMOUNT OF COLUMN DEFINITIONS

int col\_num = Convert.ToInt32(items\_r.ColumnDefinitions.Count);

Filter f = new Filter();

// RowDefinition[] rd = new RowDefinition[q.CountRows()];

// RowDefinition rd;

bool check\_records = q.CheckRecords();

//STACKPANEL TO HOLD THE ROWS

// StackPanel parent\_row\_panel;

//WE'RE ADDING +1 CAUSE THERE'S ALREADY A ROW DEFINED WHICH IT'S VALUE WOULD BE ZERO

//SO IF WE DONT INCREMENT IT, IT STARTS WITH 0, WHICH MIGHT BE A BAD IDEA :(.

//int i = 0;

//now check the number of rows

//keep adding rows as long as it doesn't exceed the row limit from the stupid database.

int i = 0;

if (check\_records)

{

for (i = 0; i < (items\_row); i++)

{

RowDefinition rd = new RowDefinition();

//rd.Height = new GridLength(50);

items\_r.RowDefinitions.Add(rd);

/\*

\* COLUMN DEFINITION

\* CREATE COLUMNS ACCORDING TO THE PARENT COLUMN OF THE FIRST ROW, WHICH IS col\_num.

\*/

//////////////////////////////////

int k = 0;

for (k = 0; k < col\_num; k++)

{

ColumnDefinition cd = new ColumnDefinition();

//cd.Width = new GridLength();

//cd.MinWidth = 150;

items\_r.ColumnDefinitions.Add(cd);

//reset column def so that when it starts looping the for again, it creates a new set of column definition

//column\_def = 0;

//////////////////////////////////////////////////////////

Border border = new Border();

TextBlock n = new TextBlock();

n.FontSize = 17.3;

n.Foreground = new SolidColorBrush(Color.FromArgb(180, 0, 0, 0));

///////////////////////////////////////

//stackpanel

StackPanel pa = new StackPanel();

// pa.Style = new Style();

//StackPanel m\_pa = new StackPanel();

// m\_pa.Name = "p";// + i.ToString();

StackPanel row\_holder = new StackPanel();

//row\_holder.Orientation = new Orientation();

// Orientation orie = row\_holder1.Orientation;

//add

//pa.

pa.Children.Add(n);

pa.Margin = new Thickness(10);

//pa.MaxWidth = 115;

pa.HorizontalAlignment = HorizontalAlignment.Center;

//border.Style = new Windows.UI.Xaml.Style();

//border.Style.TargetType = typeof(Border);

border.BorderThickness = new Thickness(0, 0, 0, 1);

border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

//calculate odd and even ish

int calc = i % 2;

//border.MaxWidth = 115;

// border.MinHeight = 61;

border.Child = pa;

row\_holder.Children.Add(border);

int calcu = col\_num - k;

//CHECK WHAT TO INPUT

if (calcu == 1)

{//LAST VALUE

}

else if (k == 2 || k == 3)

{

if (q.Results[i, k] == "0")

{

n.Text = "None";

}

else

{

n.Text = f.Numbers(Convert.ToInt32(q.Results[i, k]));

}

}

else if (k == 4)

{

n.Text = f.Date(q.Results[i, k], "dd/MM/yyyy");

}

else

{

n.Text = q.Results[i, k];

}

//TextBlock

//////////////OPTION

StackPanel option\_panel = new StackPanel();

Border option\_border = new Border();

//////////////////////////////////////////////////////

////FOR CHANGING THE ROW BACKGROUND.

if (calc == 1)

{

border.Background = new SolidColorBrush(Color.FromArgb(250, 215, 215, 215));

option\_border.Background = new SolidColorBrush(Color.FromArgb(250, 215, 215, 215));

}

else

{

border.Background = new SolidColorBrush(Color.FromArgb(250, 236, 254, 251));

option\_border.Background = new SolidColorBrush(Color.FromArgb(250, 226, 246, 241));

}

//DETERMINE WHEN TO ADD THE OPTION STUFF

if (calcu == 1)

{//CREATE THE OPTIONS STUFF

StackPanel option\_inner\_panel = new StackPanel();

//TO MAKE THINGS EASY JARE,LOOP IT THROUGH THE ARRAY

string[] stuff\_text = { "Edit", "Delete" };

string[] stuff\_image = { "", "" };

HyperlinkButton[,] link = new HyperlinkButton[items\_row, stuff\_text.Length];

int reduce = i;

if (i == items\_row)

{

reduce = i - 1;

}

for (int t = 0; t < stuff\_text.Length; t++)

{

option\_border.BorderThickness = new Thickness(0, 0, 0, 1);

option\_border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

TextBlock option\_txt = new TextBlock();

option\_txt.Text = stuff\_text[t];

option\_txt.Foreground = new SolidColorBrush(Color.FromArgb(150, 0, 0, 0));

option\_txt.FontSize = 15;

option\_txt.Margin = new Thickness(10, 0, 0, 0);

option\_txt.VerticalAlignment = VerticalAlignment.Center;

// Button button = new Button();

link[i, t] = new HyperlinkButton();

//HyperlinkButton[] link = new HyperlinkButton[tests\_row];

link[i, t].Content = option\_txt.Text;

// link.Click = RoutedEventHandler.Combine(sender, RoutedEventArgs e);

//button.ClickMode = ClickMode.Hover;

//CLICK EVENT

int reduce\_t = t;

if (t == stuff\_text.Length)

{

reduce\_t = t - 1;

}

//link[i].Click += new RoutedEventHandler(EditRecord\_Click("E",));

link[i, t].Click += (s, e) =>

{

if ((string)link[reduce, reduce\_t].Content == stuff\_text[0])

{//IT's edit

if (!Editing.IsOpen)

{

//OPEN

this.Frame.Opacity = .4;

//this.Opacity = .4;

parent.IsEnabled = false;

//SET THE EDIT TRACKER

//I DID (i-1) BECAUSE IT WAS INCREASE BY ONE, THEREFORE OUT OF BOUND ISH.

tracker = new EditTracker("items", "item\_id", item\_id.Results[(reduce), 0]);

//track the row.

tracker.Track();

this.i\_id = tracker.TrackResult[0];//set the id to the global var, so that we know what to edit.

i\_name\_e.Text = tracker.TrackResult[1];

i\_cost\_e.Text = tracker.TrackResult[2];

i\_quantity\_e.Text = tracker.TrackResult[3];

Editing.IsOpen = true;

}

}

else if ((string)link[reduce, reduce\_t].Content == stuff\_text[1])

{//it's the delete.

tracker = new EditTracker("items", "item\_id", item\_id.Results[(reduce), 0]);

//track the row

tracker.Track();

//CALL THE DELETE QUERY

Query que = new Query();

que.successMessage = "Item " + tracker.TrackResult[1] + " with id("+tracker.TrackResult[0]+") has been deleted.";

que.Remove("Items", "Where item\_id = '" + item\_id.Results[(reduce), 0] + "'");

this.Frame.Navigate(typeof(Items), null);

}

};

////ADDING

option\_inner\_panel.Children.Add(link[i, t]);

}

option\_panel.Children.Add(option\_inner\_panel);

option\_inner\_panel.Orientation = Orientation.Horizontal;

option\_panel.VerticalAlignment = VerticalAlignment.Center;

option\_border.Child = option\_panel;

Grid.SetColumn(option\_border, k);

Grid.SetRow(option\_border, i + 1);

items\_r.Children.Add(option\_border);

}

else

{

Grid.SetRow(row\_holder, i + 1);

Grid.SetColumn(row\_holder, k);

items\_r.Children.Add(row\_holder);

}

}

}

}

else

{

//NO RECORD

Messages msg = new Messages();

msg.Note("No items record available.");

items\_r.Children.Add(msg.\_Container);

RowDefinition rd = new RowDefinition();

// rd.Height = new GridLength(50);

items\_r.RowDefinitions.Add(rd);

Grid.SetColumnSpan(msg.\_Container, col\_num);

Grid.SetColumn(msg.\_Container, 0);

Grid.SetRow(msg.\_Container, 1);

}

}

//ADD BTN

private void Add\_Click(object sender, RoutedEventArgs e)

{

if (!Adding.IsOpen)

{

//OPEN

// (App.Current as App).ParentOpacity = .4;

this.Frame.Opacity = .4;

//this.Opacity = .4;

parent.IsEnabled = false;

Adding.IsOpen = true;

}

}

//Cancel BTN

private void Cancel\_Click(object sender, RoutedEventArgs e)

{

if (Adding.IsOpen)

{

//Close

// (App.Current as App).ParentOpacity = .4;

this.Frame.Opacity = 1;

//this.Opacity = .4;

parent.IsEnabled = true;

Adding.IsOpen = false;

}

if (Editing.IsOpen)

{

this.Frame.Opacity = 1;

//this.Opacity = .4;

parent.IsEnabled = true;

Editing.IsOpen = false;

}

}

//add record click btn

private void AddRecord\_Click(object sender, RoutedEventArgs e)

{

//CHECK EVERYTHING

PopUp p;

Filter f = new Filter();

string item\_name = f.Trim(i\_name.Text);

string item\_quantity = f.Trim(i\_quantity.Text);

string item\_cost = f.Trim(i\_cost.Text);

if (!(item\_name == "" || item\_quantity == "" || item\_cost == "" ))

{

//GOOD TO GO

Query q = new Query();

string[] val = { item\_name,item\_cost,item\_quantity, "Now()" };

q.successMessage = "Item " + item\_name + " has been added successfully!";

q.Add("items", val, "item\_name,item\_cost,quantity\_left,date\_time");

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Items), null);

}

else

{

if (item\_name == "")

{

p = new PopUp("Please enter the name of the item");

}

else if (item\_cost == "")

{

p = new PopUp("Please enter the price of the item");

}

else if (item\_quantity == "")

{

p = new PopUp("Please enter the quantity of the item.");

}

else

{

p = new PopUp("Sorry, an error occured. The item couldn't be added. Please try again later");

}

}

}

//FOR EDITING

private void EditRecord\_Click(object sender, RoutedEventArgs e)

{

PopUp p;

Filter f = new Filter();

// tracker.Track();

// t\_price.Text = tracker.TrackResult.Length.ToString();

//CHECK EVERYTHING

string item\_name = f.Trim(i\_name\_e.Text);

string item\_quantity = f.Trim(i\_quantity\_e.Text);

string item\_cost = f.Trim(i\_cost\_e.Text);

if (!(item\_name == "" || item\_quantity == "" || item\_cost == ""))

{

//GOOD TO GO

Query q = new Query();

string[] val = { item\_name, item\_cost, item\_quantity, "Now()" };

q.successMessage = "Item " + item\_name + " with id #"+i\_id+" has been updated successfully!";

q.Change("items","item\_name,item\_cost,quantity\_left,date\_time\_updated", val,"Where item\_id = '"+i\_id+"'");

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Items), null);

}

else

{

if (item\_name == "")

{

p = new PopUp("Please enter the name of the item");

}

else if (item\_cost == "")

{

p = new PopUp("Please enter the price of the item");

}

else if (item\_quantity == "")

{

p = new PopUp("Please enter the quantity of the item.");

}

else

{

p = new PopUp("Sorry, an error occured. The item couldn't be added. Please try again later");

}

}

}

}

}

Purchases XAML

<Page

x:Class="Lpms.Purchases"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:Lpms"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

<Grid>

<!--PARENT STACKPANEL SO THAT THE STUPID ELEMENTS DON'T OVERLAP-->

<ContentControl x:Name="parent">

<StackPanel>

<!-- record title-->

<StackPanel>

<Border Style="{StaticResource record\_label\_border}">

<StackPanel Style="{StaticResource record\_label}">

<TextBlock Text="Purchase Records" Style="{StaticResource record\_label\_txt}"/>

<Button x:Name="Add" Click="Add\_Click" ClickMode="Press" Style="{StaticResource add\_btn}">

<StackPanel Orientation="Horizontal">

<Image Source="Assets/Icons/plus-white.png"/>

<TextBlock Text=" Add"/>

</StackPanel>

</Button>

</StackPanel>

</Border>

</StackPanel>

<!--TABLE HEAD-->

<Grid x:Name="purchases\_r" Width="auto" HorizontalAlignment="Stretch" Margin="0 0 0 0">

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

</Grid.ColumnDefinitions>

<Border Grid.Row="0" Grid.Column="0" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="#Purchase Id" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="1" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Item purchased" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="2" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Quantity" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="3" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Amount" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="4" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Buyer" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="5" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Date" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="6" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head\_option}">

<TextBlock Text="Options" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

</Grid>

<!--CONTENTS-->

</StackPanel>

</ContentControl>

<Popup x:Name="Adding" IsOpen="False" Width="700" Height="auto" VerticalAlignment="Center">

<StackPanel Background="#4BA783">

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1" Width="auto" Margin="0 10 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<ComboBox x:Name="p\_items" Width="auto" SelectionChanged="p\_items\_SelectionChanged" PlaceholderText="Item to be bought" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 30">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select Item</ComboBoxItem>

</ComboBox>

<TextBlock x:Name="info" FontSize="14" FontWeight="SemiLight" Foreground="#bb000000" Margin="10 30 0 5"/>

<TextBox x:Name="p\_quantity" Width="auto" PlaceholderText="quantity to be bought" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 5 10 30"/>

<ComboBox x:Name="p\_patients" Width="auto" PlaceholderText="which patient is buying this item" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select a Patient</ComboBoxItem>

</ComboBox>

<Button x:Name="Cancel\_Add" Click="Cancel\_Click" HorizontalAlignment="Left" Content="Cancel" FontSize="20" Width="120" Height="60" Margin="0 0 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#cc000000" Foreground="#ffffff"/>

<Button x:Name="AddRecord" Click="AddRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" Margin="0 -67 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</Popup>

<!-- EDITING POPUP-->

<Popup x:Name="Editing" IsOpen="False" Width="700" Height="auto" VerticalAlignment="Center">

<StackPanel Background="#4BA783">

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1" Width="auto" Margin="0 10 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<ComboBox x:Name="p\_items\_e" Width="auto" SelectionChanged="p\_items\_SelectionChanged" PlaceholderText="Item to be bought" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 30">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select Item</ComboBoxItem>

</ComboBox>

<TextBlock x:Name="info\_e" FontSize="14" FontWeight="SemiLight" Foreground="#bb000000" Margin="10 30 0 5"/>

<TextBox x:Name="p\_quantity\_e" Width="auto" PlaceholderText="quantity to be bought" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 5 10 30"/>

<ComboBox x:Name="p\_patients\_e" Width="auto" PlaceholderText="which patient is buying this item" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select a Patient</ComboBoxItem>

</ComboBox>

<Button x:Name="Cancel\_Edit" Click="Cancel\_Click" HorizontalAlignment="Left" Content="Cancel" FontSize="20" Width="120" Height="60" Margin="0 0 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#cc000000" Foreground="#ffffff"/>

<Button x:Name="EditRecord" Click="EditRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" Margin="0 -67 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</Popup>

</Grid>

</Page>

Purchases C#

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.UI;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

using Lpms.Classes;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace Lpms

{

/// <summary>

/// The purchases page

/// </summary>

public sealed partial class Purchases : Page

{

string p\_id = "";

public Purchases()

{

this.InitializeComponent();

//FOR EDITING

EditTracker tracker;

//start displaying the purchases records

Query q = new Query();

q.GetExtra("SELECT purchase\_id,item\_id,quantity,patient\_id,date\_time FROM purchases");

q.Records();

////

Query purchase\_id = new Query();

purchase\_id.Get("purchases");

//GET THE RECORDS

purchase\_id.Records();

///////////////////

int purchases\_row = q.CountRows();

//AMOUNT OF COLUMN DEFINITIONS

int col\_num = Convert.ToInt32(purchases\_r.ColumnDefinitions.Count);

Filter f = new Filter();

bool check\_records = q.CheckRecords();

// RowDefinition[] rd = new RowDefinition[q.CountRows()];

// RowDefinition rd;

//STACKPANEL TO HOLD THE ROWS

// StackPanel parent\_row\_panel;

//WE'RE ADDING +1 CAUSE THERE'S ALREADY A ROW DEFINED WHICH IT'S VALUE WOULD BE ZERO

//SO IF WE DONT INCREMENT IT, IT STARTS WITH 0, WHICH MIGHT BE A BAD IDEA :(.

//int i = 0;

//now check the number of rows

//keep adding rows as long as it doesn't exceed the row limit from the stupid database.

int i = 0;

//CHECK IF THERE'S ANY RECORD

if (check\_records)

{

for (i = 0; i < (purchases\_row); i++)

{

RowDefinition rd = new RowDefinition();

// rd.Height = new GridLength(50);

purchases\_r.RowDefinitions.Add(rd);

/\*++

\* COLUMN DEFINITION

\* CREATE COLUMNS ACCORDING TO THE PARENT COLUMN OF THE FIRST ROW, WHICH IS col\_num.

\*/

//////////////////////////////////

int k = 0;

int ke = 1;

for (k = 0; k < col\_num; k++)

{

ColumnDefinition cd = new ColumnDefinition();

//cd.Width = new GridLength();

//cd.MinWidth = 150;

purchases\_r.ColumnDefinitions.Add(cd);

//reset column def so that when it starts looping the for again, it creates a new set of column definition

//column\_def = 0;

//calculate odd and even ish

int calc = i % 2;

//////////////////////////////////////////////////////////

Border border = new Border();

TextBlock n = new TextBlock();

n.FontSize = 17.3;

n.Foreground = new SolidColorBrush(Color.FromArgb(180, 0, 0, 0));

//TextBlock

///////////////////////////////////////

//stackpanel

StackPanel pa = new StackPanel();

// pa.Style = new Style();

//StackPanel m\_pa = new StackPanel();

// m\_pa.Name = "p";// + i.ToString();

StackPanel row\_holder = new StackPanel();

//add

//pa.

pa.Children.Add(n);

pa.Margin = new Thickness(10);

//pa.MaxWidth = 115;

pa.HorizontalAlignment = HorizontalAlignment.Center;

pa.VerticalAlignment = VerticalAlignment.Center;

border.BorderThickness = new Thickness(0, 0, 0, 1);

border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

//border.MaxWidth = 115;

//border.MinHeight = 60.5;

border.Child = pa;

row\_holder.Children.Add(border);

//CHECK WHAT TO INPUT

Query c = new Query();

//check if it's the last column

int calcu = (col\_num - k);

if (k == 0)

{

n.Text = "#(" + q.Results[i, k] + ")";

//n.FontWeight = FontWeight.23;

}

else if (k == 1)

{

//GET THE item bought

c.Get("items", "WHERE item\_id = '" + q.Results[i, k] + "'");

if (c.CheckRecords() == true)

{

c.Records();

n.Text = c.Results[0, 1];

}

else

{

n.Text = "No record available";

}

}

else if (k == 3)

{//get the price of the item

//calculate, price \* quanity

c.Get("items", "WHERE item\_id = '" + q.Results[i, 1] + "'");

c.Records();

if (c.CheckRecords() == true)

{

int amount = Convert.ToInt32(c.Results[0, 2]) \* Convert.ToInt32(q.Results[i, 2]);

n.Text = f.Numbers(amount,"No value");

}

else

{

n.Text = "Information not available";

}

}

else if (k == 4)

{//get the buyer

c.Get("patients", "Where patient\_id = '" + q.Results[i, 3] + "'");

if (c.CheckRecords())

{

c.Records();

n.Text = c.Results[0, 1];

}

else

{

n.Text = "Patient's record not available";

}

}

else if (k == 5)

{//DATE

n.Text = f.Date(q.Results[i, 4], "dd-MM-yy");

}

else if (k == 6)

{

}

else

{

n.Text = q.Results[i, k];

}

//////////////OPTION

StackPanel option\_panel = new StackPanel();

Border option\_border = new Border();

//////////////////////////////////////////////////////

////FOR CHANGING THE ROW BACKGROUND.

if (calc == 1)

{

border.Background = new SolidColorBrush(Color.FromArgb(250, 225, 225, 225));

option\_border.Background = new SolidColorBrush(Color.FromArgb(250, 215, 215, 215));

}

else

{

border.Background = new SolidColorBrush(Color.FromArgb(250, 236, 254, 251));

option\_border.Background = new SolidColorBrush(Color.FromArgb(250, 226, 246, 241));

}

//DETERMINE WHEN TO ADD THE OPTION STUFF

if (calcu == 1)

{//CREATE THE OPTIONS STUFF

StackPanel option\_inner\_panel = new StackPanel();

//TO MAKE THINGS EASY JARE,LOOP IT THROUGH THE ARRAY

string ed = "Edit";

string del = "Delete";

string[] stuff\_text = { ed, del };

string[] stuff\_image = { "", "" };

HyperlinkButton[,] link = new HyperlinkButton[purchases\_row, stuff\_text.Length];

int reduce = i;

if (i == purchases\_row)

{

reduce = i - 1;

}

for (int t = 0; t < stuff\_text.Length; t++)

{

option\_border.BorderThickness = new Thickness(0, 0, 0, 1);

option\_border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

TextBlock option\_txt = new TextBlock();

option\_txt.Text = stuff\_text[t];

option\_txt.Foreground = new SolidColorBrush(Color.FromArgb(150, 0, 0, 0));

option\_txt.FontSize = 15;

option\_txt.Margin = new Thickness(10, 0, 3, 0);

// Button button = new Button();

link[i, t] = new HyperlinkButton();

//HyperlinkButton[] link = new HyperlinkButton[purchases\_row];

link[i, t].Content = option\_txt.Text;

// link.Click = RoutedEventHandler.Combine(sender, RoutedEventArgs e);

// button.ClickMode = ClickMode.Hover;

//CLICK EVENT

int reduce\_t = t;

if (t == stuff\_text.Length)

{

reduce\_t = t - 1;

}

//link[i].Click += new RoutedEventHandler(EditRecord\_Click("E",));

link[i, t].Click += (s, e) =>

{

if ((string)link[reduce, reduce\_t].Content == ed)

{//IT's edit

if (!Editing.IsOpen)

{

//OPEN

this.Frame.Opacity = .4;

//this.Opacity = .4;

parent.IsEnabled = false;

//SET THE EDIT TRACKER

//I DID (i-1) BECAUSE IT WAS INCREASE BY ONE, THEREFORE OUT OF BOUND ISH.

tracker = new EditTracker("purchases", "purchase\_id", purchase\_id.Results[(reduce), 0]);

//track the row.

tracker.Track();

p\_id = tracker.TrackResult[0];

p\_quantity\_e.Text = tracker.TrackResult[2];

c.Get("patients", "Where patient\_id = '" + q.Results[reduce, 3] + "'");

c.Records();

p\_patients\_e.SelectedIndex = f.SelectedInverse(c.Results[0,0], "patients", null, 1);

c.Get("items", "Where item\_id = '" + q.Results[reduce, 1] + "'");

c.Records();

p\_items\_e.SelectedIndex = f.SelectedInverse(c.Results[0, 0], "items", null, 1);

//t\_price\_e.Text = tracker.TrackResult[2];

Editing.IsOpen = true;

}

}

else if ((string)link[reduce, reduce\_t].Content == del)

{//it's the delete.

//CALL THE DELETE QUERY

//track first

tracker = new EditTracker("Purchases", "purchase\_id", purchase\_id.Results[(reduce), 0]);

//track

tracker.Track();

//CALL THE DELETE QUERY

Query que = new Query();

que.successMessage = "Purchase #(" + tracker.TrackResult[0] + ") has been deleted successfully.";

que.Remove("purchases", "where purchase\_id = '" + tracker.TrackResult[0] + "'");

this.Frame.Navigate(typeof(Purchases), null);

}

};

//private void R\_C(object sender, EventArgs e){

//}

////ADDING

option\_inner\_panel.Children.Add(link[i, t]);

//ADD THE EDIT, DELETE ICON

//Image img = new Image();

//img.Source = new ;

}

option\_panel.Children.Add(option\_inner\_panel);

option\_inner\_panel.Orientation = Orientation.Horizontal;

option\_panel.VerticalAlignment = VerticalAlignment.Center;

option\_border.Child = option\_panel;

Grid.SetColumn(option\_border, k);

Grid.SetRow(option\_border, i + 1);

purchases\_r.Children.Add(option\_border);

}

else

{

Grid.SetRow(row\_holder, i + 1);

Grid.SetColumn(row\_holder, k);

purchases\_r.Children.Add(row\_holder);

}

///i've even forgotten what this does. :(

ke++;

}

}

}

else

{//NO RECORD, SHOW SOMETHING

Messages msg = new Messages();

msg.Note("No Purchases available.");

purchases\_r.Children.Add(msg.\_Container);

RowDefinition rd = new RowDefinition();

// rd.Height = new GridLength(50);

purchases\_r.RowDefinitions.Add(rd);

Grid.SetColumnSpan(msg.\_Container, col\_num);

Grid.SetColumn(msg.\_Container, 0);

Grid.SetRow(msg.\_Container, 1);

}

////////////////////////////////////////////////

//fill some stuff in the form

Query query = new Query();

query.Get("Patients");

int row\_count = query.CountRows();

query.Records();

p\_patients.SelectedValue = new object();

for (int r = 0; r < row\_count; r++)

{

Object p = new object[query.CountRows()];

p = query.Results[r, 1];

p\_patients.Items.Add(p);

// t\_patients.SelectedValuePath = new string[row\_count];

// selected\_value\_path[r] =

// t\_patients.Selec

// p\_patients.SelectedValuePath = query.Results[r, 1];

// selected\_value[r] = t\_patients.SelectedValuePath;

// p\_patients.SelectedValue = p\_patients.SelectedValuePath = query.Results[r,1];

//editing box

p\_patients\_e.Items.Add(p);

//p\_patients\_e.SelectedValue = "a";

//p\_patients\_e.SelectedValuePath = query.Results[r, 0];

}

query.Get("items");

query.Records();

for (int r = 0; r < query.CountRows(); r++)

{

Object d = query.Results[r, 1];

p\_items.Items.Add(d);

//editing box

p\_items\_e.Items.Add(d);

// p\_items\_e.SelectedValue = "a";

// p\_items\_e.SelectedValuePath = query.Results[r, 0];

// t\_doctors.SelectedIndex

}

}

//ADD BTN

private void Add\_Click(object sender, RoutedEventArgs e)

{

if (!Adding.IsOpen)

{

//OPEN

// (App.Current as App).ParentOpacity = .4;

this.Frame.Opacity = .4;

//this.Opacity = .4;

parent.IsEnabled = false;

Adding.IsOpen = true;

}

}

//Cancel BTN

private void Cancel\_Click(object sender, RoutedEventArgs e)

{

if (Adding.IsOpen)

{

//Close

// (App.Current as App).ParentOpacity = .4;

this.Frame.Opacity = 1;

//this.Opacity = .4;

parent.IsEnabled = true;

Adding.IsOpen = false;

}

if (Editing.IsOpen)

{

//Close

this.Frame.Opacity = 1;

//this.Opacity = .4;

parent.IsEnabled = true;

Editing.IsOpen = false;

}

}

//add record click btn

private void AddRecord\_Click(object sender, RoutedEventArgs e)

{

//CHECK EVERYTHING

PopUp p;

Filter f = new Filter();

string purchase\_quantity = f.Trim(p\_quantity.Text);

//string[] t = f.Selected("0", "purchases");

string purchase\_patient = f.Trim(p\_patients.SelectedIndex.ToString());

string purchase\_item = f.Trim(p\_items.SelectedIndex.ToString());

if (!(purchase\_quantity == "" || (purchase\_item == "" || purchase\_item == "0") || (purchase\_patient == "" || purchase\_patient == "0")))

{

//GOOD TO GO

string[] p\_p = f.Selected(p\_patients.SelectedIndex, "patients", null, 1);

string[] p\_i = f.Selected(p\_items.SelectedIndex, "Items", null, 1);

purchase\_patient = p\_p[0];

purchase\_item = p\_i[0];

Query q = new Query();

//now edit the item quantity

// the quantity left and the quantity consumed has to be update

//using simple maths

q.Get("items","Where item\_id = '"+p\_i[0]+"'");

q.Records();

//check if the quantity requested is not more than the available quantity

int quant = Convert.ToInt32(p\_quantity.Text);//quantity to be consumed

int item\_quant = Convert.ToInt32(q.Results[0,3]);//the quantity left in the item

int quant\_consumed = Convert.ToInt32(q.Results[0,4]);//the quantity already consumed.

//get the number of quantity left in the item selected

string[] s\_item = f.Selected(p\_items.SelectedIndex, "Items", null, 1);

int the\_item = Convert.ToInt32(s\_item[3]);

//##########################################################

if (quant <= item\_quant && the\_item > 0)

{//within range ,allow

//NOW DO SOME CALCULATIONS

int quant\_left = item\_quant - quant;

int quant\_cons = quant\_consumed + quant;

string[] q\_val = { quant\_left.ToString(), quant\_cons.ToString(), "now()" };

string[] val = { purchase\_item, purchase\_patient, purchase\_quantity, "Now()" };

q.successMessage = null;

q.Change("items", "quantity\_left,quantity\_consumed,date\_time\_updated", q\_val, "Where item\_id = '" + p\_i[0] + "'");

q.successMessage = "purchase record has been added successfully!";

q.Add("purchases", val, "item\_id,patient\_id,quantity,date\_time");

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Purchases), null);

}

else

{

if (the\_item > 0)

{

p = new PopUp("Quantity to be bought is more than the available quantity which is " + item\_quant);

}

else

{

p = new PopUp("Sorry, the item to be bought has no quantity left to be purchased.");

}

}

}

else

{

if (purchase\_quantity == "")

{

p = new PopUp("Please enter a quantity for the item being purchased");

}

else if (purchase\_patient == "" || purchase\_patient == "0")

{

p = new PopUp("Please select a patient");

}

else if (purchase\_item == "" || purchase\_item == "0")

{

p = new PopUp("Please select an item for purchase");

}

else

{

p = new PopUp("Sorry, an error occured. The purchase couldn't be added. Please try again later");

}

}

}

//FOR EDITING

private void EditRecord\_Click(object sender, RoutedEventArgs e)

{

//CHECK EVERYTHING

PopUp p;

Filter f = new Filter();

int add\_sub = 0;

string purchase\_quantity = f.Trim(p\_quantity\_e.Text);

//string[] t = f.Selected("0", "purchases");

string purchase\_patient = f.Trim(p\_patients\_e.SelectedIndex.ToString());

string purchase\_item = f.Trim(p\_items\_e.SelectedIndex.ToString());

if (!(purchase\_quantity == "" || (purchase\_item == "" || purchase\_item == "0") || (purchase\_patient == "" || purchase\_patient == "0")))

{

//GOOD TO GO

string[] p\_p = f.Selected(p\_patients\_e.SelectedIndex, "patients", null, 1);

string[] p\_i = f.Selected(p\_items\_e.SelectedIndex, "Items", null, 1);

purchase\_patient = p\_p[0];

purchase\_item = p\_i[0];

Query q = new Query();

//now edit the item quantity

// the quantity left and the quantity consumed has to be update

//using simple maths

q.Get("items","Where item\_id = '"+p\_i[0]+"'");

q.Records();

//check if the quantity requested is not more than the available quantity

int quant = Convert.ToInt32(p\_quantity\_e.Text);//quantity to be consumed

int item\_quant = Convert.ToInt32(q.Results[0, 3]);//the quantity left in the item

int quant\_consumed = Convert.ToInt32(q.Results[0, 4]);//the quantity already consumed.

if (p\_id.Trim() != "")

{

//check if it's the same item, same patient and same quantity, so as not to keep changing

//the quantity left, in case the user just submitted the edit form without change anything

q.Get("purchases", "WHERE purchase\_id = '" + p\_id + "'");

q.Record();

//get some stuff from different tables, chai.

int selected\_patient = f.SelectedInverse(q.Result[3], "patients", null, 1);//get the selected index of the edit.

int selected\_item = f.SelectedInverse(q.Result[1], "items", null, 1);//get the selected index of the edit.

int initial\_quant = Convert.ToInt32(q.Result[2]);

if (purchase\_quantity == q.Result[2] && p\_patients\_e.SelectedIndex == selected\_patient && p\_items\_e.SelectedIndex == selected\_item)

{//no value was changed

p = new PopUp("Nothing in purchase record #(" + p\_id + ") was edited. click cancel or edit a value and submit");

}

else

{

//get the number of quantity left in the item selected

string[] s\_item = f.Selected(p\_items\_e.SelectedIndex, "Items", null, 1);

int the\_item = Convert.ToInt32(s\_item[3]);

//check if it's the initial item selected, so that we know what to show

Query d = new Query();

d.GetExtra("SELECT item\_id from purchases WHERE purchase\_id = '" + p\_id + "'");

d.Record();

bool cond = false;

bool yes\_or\_no = false;

int greater = 0;

if (p\_i[0] != d.Result[0])

{//it's not the initial selected item.

//if (the\_item > 0)

//{

// cond = true;

//}

// else

// {

// yes\_or\_no = false;

// }

if (the\_item > 0)

{

greater = 1;

}

cond = true;

}

else

{//its the same item chosen

yes\_or\_no = true;

cond = false;

}

//##########################################################

if ((quant <= item\_quant && cond == true && greater == 1) ||(yes\_or\_no == true))

{//within range ,allow

//NOW DO SOME CALCULATIONS

int quant\_left = 0;

int quant\_cons = 0;

//check for some conditions

//CHECK IF THE ITEM WASN'T CHANGED BUT THE QUANTITY WAS CHANGED

//so we minus what was changed, or add what was changed

if(purchase\_quantity != q.Result[2] && p\_items\_e.SelectedIndex == selected\_item)

{//the same item, but the quantity was changed, so carefully calculate

//if the text box is greater than the initial quant, add what was added to

//quantity consumed and subtract what was subtracted

if (quant > initial\_quant)

{//what was entered is higher than initial so add to quantity consumed and minus from quantity left.

add\_sub = quant - initial\_quant;

quant\_left = item\_quant - add\_sub;

quant\_cons = quant\_consumed + add\_sub;

}

else

{//what was entered is lower.

add\_sub = initial\_quant - quant;

quant\_left = item\_quant + add\_sub;

quant\_cons = quant\_consumed - add\_sub;

}

}

else if(purchase\_quantity == q.Result[2] && p\_patients\_e.SelectedIndex != selected\_patient &&(p\_items\_e.SelectedIndex == selected\_item))

{//the quantity wasn't changed, but the patient was changed, so leave the quantity as it is

//

quant\_left = item\_quant;

quant\_cons = quant\_consumed;

}

else if ((purchase\_quantity == q.Result[2] || purchase\_quantity != q.Result[2]) && p\_items\_e.SelectedIndex != selected\_item)

{//it's a different item selected, update the quantity of the initial one

Query c = new Query();

c.Get("items", "WHERE item\_id = '" + q.Result[1] + "'");

c.Record();

int left = Convert.ToInt32(c.Result[3]) + initial\_quant;//add to the quantity left

int cons = Convert.ToInt32(c.Result[4]) - initial\_quant;//subtract from the quantity consumed.

//

quant\_left = item\_quant - quant;

quant\_cons = quant\_consumed + quant;

string[] v = {left.ToString(),cons.ToString(),"NOW()"};

q.Change("items", "quantity\_left,quantity\_consumed,date\_time\_updated", v,"WHERE item\_id = '"+q.Result[1]+"'");

}

else

{//its something else, so act normally.

quant\_left = item\_quant - quant;

quant\_cons = quant\_consumed + quant;

}

string[] q\_val = { quant\_left.ToString(), quant\_cons.ToString(), "now()" };

string[] val = { purchase\_item, purchase\_patient, purchase\_quantity };

if (yes\_or\_no == true)

{//THE ITEM IS THE SAME

//Query h = new Query();

// h.GetExtra("SELECT quantity\_left from items WHERE item\_id = '"++"'");

// h.Record();

//int former\_quantity = Convert.ToInt32(h.Result[0]);

if (!(quant > initial\_quant && add\_sub > item\_quant))

{

q.successMessage = null;

q.Change("items", "quantity\_left,quantity\_consumed,date\_time\_updated", q\_val, "Where item\_id = '" + p\_i[0] + "'");

q.successMessage = "purchase record #(" + p\_id + ") has been edited successfully!";

q.Change("purchases", "item\_id,patient\_id,quantity", val, "WHERE purchase\_id = '" + p\_id + "'");

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Purchases), null);

}

else

{

p = new PopUp("Sorry, the quantity you added is exceeding the quantity left of the item selected.");

}

}

else

{//THE ITEM WAS CHANGED

q.successMessage = null;

q.Change("items", "quantity\_left,quantity\_consumed,date\_time\_updated", q\_val, "Where item\_id = '" + p\_i[0] + "'");

q.successMessage = "purchase record #(" + p\_id + ") has been edited successfully!";

q.Change("purchases", "item\_id,patient\_id,quantity", val, "WHERE purchase\_id = '" + p\_id + "'");

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Purchases), null);

}

}

else

{

if (the\_item > 0)

{

p = new PopUp("Quantity to be bought is more than the available quantity which is " + item\_quant);

}

else

{

p = new PopUp("Sorry, the item to be bought has no quantity left to be purchased.");

}

}

}

}

else

{

p = new PopUp("Internal error, the record couldn't be edited. Try again later.");

}

}

else

{

if (purchase\_quantity == "")

{

p = new PopUp("Please enter a quantity for the item being purchased");

}

else if (purchase\_patient == "" || purchase\_patient == "0")

{

p = new PopUp("Please select a patient");

}

else if (purchase\_item == "" || purchase\_item == "0")

{

p = new PopUp("Please select an item for purchase");

}

else

{

p = new PopUp("Sorry, an error occured. The purchase couldn't be added. Please try again later");

}

}

}

private void p\_items\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{//update the quantity of item when choosing, i don tire jare.

Filter f = new Filter();

if (Adding.IsOpen)

{

string[] p\_i = f.Selected(p\_items.SelectedIndex, "Items", null, 1);

int quant = Convert.ToInt32(p\_i[3]);

if (quant > 0)

info.Text = "Max. quantity that can be bought: " + p\_i[3];

else

info.Text = "No quantity available";

}

if (Editing.IsOpen)

{//its the editing,

//check if it's the initial item selected, so that we know what to show

Query q = new Query();

q.GetExtra("SELECT item\_id from purchases WHERE purchase\_id = '" + p\_id + "'");

q.Record();

string[] p\_i = f.Selected(p\_items\_e.SelectedIndex, "Items", null, 1);

int quant = Convert.ToInt32(p\_i[3]);

if (p\_i[0] != q.Result[0])

{//show something, it's not the initial selected item.

if (quant > 0)

info\_e.Text = "Max. quantity that can be bought: " + p\_i[3];

else

info\_e.Text = "No quantity available";

}

else

{

info\_e.Text = "";

}

}

}

}

}

Reports XAML

<Page

x:Class="Lpms.Reports"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:Lpms"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition MaxWidth="900"/>

<ColumnDefinition Width="400"/>

</Grid.ColumnDefinitions>

<!--PARENT STACKPANEL SO THAT THE STUPID ELEMENTS DON'T OVERLAP-->

<StackPanel Grid.Column="0" Grid.Row="0">

<ContentControl x:Name="parent">

<StackPanel x:Name="reports\_r">

<!-- record title-->

<StackPanel Width="900">

<Border Style="{StaticResource record\_label\_border}">

<StackPanel Style="{StaticResource record\_label}" Orientation="Horizontal">

<TextBlock Text="Records of Reports" Style="{StaticResource record\_label\_txt}"/>

<TextBlock x:Name="sort\_txt" Style="{StaticResource record\_label\_txt}"/>

</StackPanel>

</Border>

</StackPanel>

<!--body-->

</StackPanel>

</ContentControl>

</StackPanel>

<!-- second column-->

<StackPanel Grid.Column="1" Grid.Row="0" Background="#ffffff">

<StackPanel>

<Border Style="{StaticResource record\_label\_border}">

<StackPanel Style="{StaticResource record\_label}" Orientation="Vertical" Margin="0,0,0,0">

<TextBlock Text="Add Report" Style="{StaticResource record\_label\_txt}" FontWeight="SemiLight" Foreground="#cc000000"/>

</StackPanel>

</Border>

</StackPanel>

<StackPanel >

<StackPanel>

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1 0 1 1" Width="auto" Margin="0 10 0 10">

<StackPanel Width="auto" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<TextBox x:Name="r\_name" Width="auto" PlaceholderText="Name of report" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 30"/>

<ComboBox x:Name="r\_tests" Width="auto" PlaceholderText="Test report belongs to" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 30">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select Test</ComboBoxItem>

</ComboBox>

<TextBox FontSize="18" TextWrapping="Wrap" x:Name="r\_content" PlaceholderText="Report write up" Height="250" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<ScrollBar>

<ScrollBar.MaxHeight>

60

</ScrollBar.MaxHeight>

</ScrollBar>

<Button x:Name="AddRecord" Click="AddRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</StackPanel>

</StackPanel>

<Popup x:Name="Editing" IsOpen="False" Width="700" Height="auto" VerticalAlignment="Top">

<StackPanel Background="#4BA783">

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1" Width="auto" Margin="0 10 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<TextBox x:Name="r\_name\_e" Width="auto" PlaceholderText="Name of report" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 30"/>

<ComboBox x:Name="r\_tests\_e" Width="auto" PlaceholderText="Test report belongs to" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 30">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select Test</ComboBoxItem>

</ComboBox>

<TextBox x:Name="r\_content\_e" TextWrapping="Wrap" Width="auto" PlaceholderText="Report write up" Height="250" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<Button x:Name="Cancel\_Edit" Click="Cancel\_Click" HorizontalAlignment="Left" Content="Cancel" FontSize="20" Width="120" Height="60" Margin="0 0 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#cc000000" Foreground="#ffffff"/>

<Button x:Name="EditRecord" Click="EditRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" Margin="0 -67 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</Popup>

</Grid>

</Page>

Reports C#

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.UI;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

using System.Drawing;

using Lpms.Classes;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace Lpms

{

/// <summary>

/// The reports page

/// </summary>

public sealed partial class Reports : Page

{

string r\_id = "";

EditTracker tracker;

/// <summary>

/// the current test report being shown,holds the id of the test report being shown

/// </summary>

string chosen\_test = ((App.Current as App).chosen\_test\_for\_report == null ? "" : (App.Current as App).chosen\_test\_for\_report);

Filter f = new Filter();

/// <summary>

/// the theme color

/// </summary>

SolidColorBrush theme\_color = new SolidColorBrush(Color.FromArgb(255, 75, 167, 131));

/// <summary>

/// the selected test color

/// </summary>

SolidColorBrush selected\_color = new SolidColorBrush(Color.FromArgb(255, 70, 70, 70));

public Reports()

{

this.InitializeComponent();

//start displaying the items records

Query q = new Query();

//show according to what is needed

if (this.chosen\_test == "")

{

q.Get("reports");

}

else

{

q.Get("reports", "Where test\_id = '" + this.chosen\_test + "'");

}

q.Records();

int reports\_row = q.CountRows();

bool check\_records = q.CheckRecords();

int i = 0;

if (check\_records == true)

{

//create 2 parts, one that shows the test to select

Grid main\_g = new Grid();

RowDefinition rd = new RowDefinition();

StackPanel main\_p = new StackPanel();

Border side\_p\_border = new Border();

StackPanel side\_p = new StackPanel();

Border side\_p\_inner\_border = new Border();

side\_p\_inner\_border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

side\_p\_inner\_border.BorderThickness = new Thickness(1);

side\_p\_inner\_border.HorizontalAlignment = HorizontalAlignment.Stretch;

side\_p\_inner\_border.Margin = new Thickness(0, 0, 0, 4.0);

StackPanel side\_p\_head = new StackPanel();

side\_p\_head.Background = new SolidColorBrush(Color.FromArgb(255, 250, 251, 251));

TextBlock side\_p\_head\_txt = new TextBlock();

side\_p\_head\_txt.Text = "Tests";

side\_p\_head\_txt.FontSize = 23;

side\_p\_head\_txt.Padding = new Thickness(10, 9, 5, 5);

side\_p\_head\_txt.Foreground = new SolidColorBrush(Color.FromArgb(255, 40, 40, 40));

//design

main\_p.Margin = new Thickness(10, 7, 25, 5);

side\_p\_border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

//side\_p\_border.BorderThickness = new Thickness(1);

// side\_p\_border.Background = theme\_color;

// side\_p\_border.CornerRadius = new CornerRadius(3);

side\_p\_border.Margin = new Thickness(5, 10, 5, 1);

side\_p\_border.HorizontalAlignment = HorizontalAlignment.Right;

///////////////////////////////

for (int j = 0; j < 2; j++)

{//create 2 columns

ColumnDefinition cd = new ColumnDefinition();

if (j == 1)

{

cd.Width = new GridLength(250);

}

else

{

cd.Width = new GridLength(600);

}

main\_g.ColumnDefinitions.Add(cd);

}

Grid.SetColumn(main\_p, 0);

Grid.SetRow(main\_p, 0);

//for the side menu.

Grid.SetColumn(side\_p\_border, 1);

Grid.SetRow(side\_p\_border, 0);

main\_g.RowDefinitions.Add(rd);

//

side\_p\_head.Children.Add(side\_p\_head\_txt);

side\_p\_inner\_border.Child = side\_p\_head;

side\_p.Children.Add(side\_p\_inner\_border);

side\_p\_border.Child = side\_p;

main\_g.Children.Add(side\_p\_border);

main\_g.Children.Add(main\_p);

reports\_r.Children.Add(main\_g);

//design the side bar that shows the list of tests that have reports

Query s\_t = new Query();

s\_t.GetExtra("Select test\_id,test\_name from tests");

s\_t.Records();

int counter = s\_t.CountRows();

//get the number of tests that have reports

Query single = new Query();

single.GetExtra("Select report\_name,test\_id FROM reports");

int single\_count = single.CountRows();

single.Records();

string test\_checker = single.Results[0, 1];

for (int m = 0; m < single\_count; m++)

{

//check if the test isn't deleted already

Query g = new Query();

g.GetExtra("SELECT test\_id FROM tests WHERE test\_id = '" + single.Results[m, 1] + "'");

if (g.CheckRecords() == true)

{

//check if they belong to more than one report

if (test\_checker != single.Results[m, 1])

{//there's another test, so it's more than one

//lets show all

Button a\_btn = new Button();

a\_btn.Content = "All";

a\_btn.BorderBrush = new SolidColorBrush(Color.FromArgb(23, 255, 255, 255));

a\_btn.BorderThickness = new Thickness(1, 1, 1, 1);

a\_btn.HorizontalAlignment = HorizontalAlignment.Stretch;

a\_btn.Background = theme\_color;

a\_btn.Padding = new Thickness(15, 20, 15, 5);

a\_btn.BorderBrush = new SolidColorBrush(Color.FromArgb(70, 0, 0, 0));

a\_btn.Margin = new Thickness(-3, -10, -3, 1);

if (this.chosen\_test == "")

{

//its all that was chosen, change the color

a\_btn.Background = selected\_color;

a\_btn.Foreground = new SolidColorBrush(Colors.White);

}

a\_btn.Click += (s, e) =>

{

//show the selected test reports

//set the value of chosen\_test to empty;

(App.Current as App).chosen\_test\_for\_report = "";

this.Frame.Navigate(typeof(Reports), null);

};

side\_p.Children.Add(a\_btn);

break;

}

}

}

Button[] btn = new Button[counter];

for (int k = 0; k < counter; k++)

{//now lets check for tests that have reports ,so that we don't display all/

Query t = new Query();

t.GetExtra("Select report\_name FROM reports where test\_id = '" + s\_t.Results[k, 0] + "' ");

if (t.CheckRecords() == true)

{//there are records

int o = 0;

//start designing

int calc = counter - k;

btn[k] = new Button();

StackPanel panel = new StackPanel();

panel.Orientation = Orientation.Horizontal;

panel.Margin = new Thickness(15, 20, 15, 5);

panel.HorizontalAlignment = HorizontalAlignment.Center;

StackPanel count\_p = new StackPanel();

count\_p.Background = new SolidColorBrush(Colors.White);

TextBlock count\_txt = new TextBlock();

count\_p.Margin = new Thickness(10, 0, 0, 0);

count\_txt.Text = t.CountRows().ToString();

count\_txt.FontSize = 14.5;

count\_txt.Foreground = theme\_color;

count\_txt.Padding = new Thickness(7, 3, 7, 3);

TextBlock txt = new TextBlock();

txt.Text = s\_t.Results[k, 1];

txt.FontSize = 15.5;

btn[k].Foreground = new SolidColorBrush(Colors.White);

btn[k].HorizontalAlignment = HorizontalAlignment.Stretch;

btn[k].BorderBrush = new SolidColorBrush(Color.FromArgb(70, 0, 0, 0));

btn[k].Background = theme\_color;

//check if it's the only result,make sure it's not a deleted test.

int amount = 0;

for (int h = 0; h < q.CountRows(); h++)

{

Query g = new Query();

g.GetExtra("SELECT test\_id from tests WHERE test\_id = '" + q.Results[h, 3] + "'");

g.Record();

if (g.CheckRecords() == true)

{//now check if h has looped once

if (h > 0)

{//now check if the previous value is the same, if it's only one value, they have to be the same :-)

if(q.Results[h,3] != q.Results[h-1,3])

{

amount += 1;

}

}

}

}

if (amount > 0 && amount <= 1)

{//only 1 value

//its the one that's chosen, change the color

btn[k].Foreground = new SolidColorBrush(Colors.WhiteSmoke);

btn[k].Background = selected\_color;

count\_txt.Foreground = selected\_color;

sort\_txt.Text = "| Showing result for " + s\_t.Results[k, 1];

sort\_txt.FontSize = 13;

sort\_txt.Foreground = new SolidColorBrush(Color.FromArgb(255, 120, 120, 120));

sort\_txt.VerticalAlignment = VerticalAlignment.Center;

}

else

{//ts more than one

if (this.chosen\_test == s\_t.Results[k, 0])

{

//its the one that's chosen, change the color

btn[k].Foreground = new SolidColorBrush(Colors.WhiteSmoke);

btn[k].Background = selected\_color;

count\_txt.Foreground = selected\_color;

sort\_txt.Text = "| Showing result for " + s\_t.Results[k, 1];

sort\_txt.FontSize = 13;

sort\_txt.Foreground = new SolidColorBrush(Color.FromArgb(255, 120, 120, 120));

sort\_txt.VerticalAlignment = VerticalAlignment.Center;

}

}

btn[k].Margin = new Thickness(-3, -6.6, -3, 0);

if (calc == 1)

{

btn[k].BorderThickness = new Thickness(0, 0, 0, 0);

}

else

{

btn[k].BorderThickness = new Thickness(1, 0, 1, 1);

}

count\_p.Children.Add(count\_txt);

panel.Children.Add(txt);

panel.Children.Add(count\_p);

btn[k].Content = panel;

//click event

//btn[k].Click +=Cancel\_Click;

//lets solve the annoying issue that's common, if bla is the last value, bla - 1

int reduce = k;

if (k == counter)

{

//the last value

reduce = k - 1;

}

btn[k].Click += (s, e) =>

{

//show the selected test reports

//set the value of chosen\_test to the selected test id

(App.Current as App).chosen\_test\_for\_report = s\_t.Results[reduce, 0];

this.Frame.Navigate(typeof(Reports), null);

};

side\_p.Children.Add(btn[k]);

}

}

int ot = 0;//for switching designs.

for (i = 0; i < (reports\_row); i++)

{

//get test info

Query c = new Query();

c.GetExtra("Select test\_id,test\_name FROM tests Where test\_id = '" + q.Results[i, 3] + "'");

c.Record();

if (c.CheckRecords())

{

//start generating designs

//get color that will be used

SolidColorBrush text\_color = new SolidColorBrush(Color.FromArgb(255, 12, 12, 12));

SolidColorBrush head\_bg = new SolidColorBrush(Color.FromArgb(255, 255, 255, 255));

SolidColorBrush body\_bg = new SolidColorBrush(Color.FromArgb(255, 250, 250, 250));

SolidColorBrush footer\_bg = head\_bg;

SolidColorBrush f\_text\_color = new SolidColorBrush(Color.FromArgb(255, 40, 40, 40));

SolidColorBrush f\_text\_color2 = new SolidColorBrush(Color.FromArgb(255, 60, 60, 60));

double footer\_txt\_size = 15;

//pattern

/\*

\* <border>

\* <stackpanel>

\* <stackpanel head>

\* <Textblock/>

\* <border/>

\* </stackpanel head>

\* <stackpanel body>

\* <Textblock/>

\* </stackpanel body>

\* <stackpanel footer>

\* <textblock/>

\* <textblock/>

\* </stackpanel footer>

\* </stackpanel>

\* </border>

\*/

Border parent\_border = new Border();

parent\_border.Margin = new Thickness(5, 15, 5, 5);

parent\_border.CornerRadius = new CornerRadius(5);

StackPanel parent\_panel = new StackPanel();

//#################### HEAD

StackPanel head\_panel = new StackPanel();

head\_panel.MinHeight = 60;

Border head\_border\_t = new Border();

head\_border\_t.BorderThickness = new Thickness(2);

head\_border\_t.BorderBrush = new SolidColorBrush(Color.FromArgb(255, 41, 128, 185));

StackPanel inner\_head = new StackPanel();

inner\_head.Margin = new Thickness(10, 6, 5, 5);

StackPanel inner\_head1 = new StackPanel();

StackPanel inner\_head2 = new StackPanel();

inner\_head2.HorizontalAlignment = HorizontalAlignment.Right;

inner\_head2.Orientation = Orientation.Horizontal;

inner\_head2.Margin = new Thickness(1, -60, 0, 0);

TextBlock head\_txt = new TextBlock();

head\_txt.FontSize = 20;

head\_txt.Foreground = new SolidColorBrush(Color.FromArgb(255, 30, 30, 30));

TextBlock head\_txt\_sub = new TextBlock();

head\_txt\_sub.FontSize = 16;

head\_txt\_sub.Margin = new Thickness(0, 5, 0, 10);

head\_txt\_sub.Foreground = new SolidColorBrush(Color.FromArgb(255, 65, 65, 65));

////////////////////////

parent\_border.BorderThickness = new Thickness(1);

parent\_border.BorderBrush = new SolidColorBrush(Color.FromArgb(30, 0, 0, 0));

////////////////////////

//design

//inner\_head.Orientation = Orientation.Horizontal;

head\_txt.Text = q.Results[i, 1] + " ";//name of report

string test\_name = c.RecordExist(c.Result[1], "");

string test\_id = c.Result[0];

head\_txt\_sub.Text = c.RecordExist("Report for " + test\_name + " #(" + test\_id + ")", "Test Record not available");

head\_txt\_sub.Foreground = new SolidColorBrush(Color.FromArgb(255, 90, 90, 90));

head\_txt\_sub.FontSize = 14;

head\_txt.FontSize = 18;

head\_panel.Background = head\_bg;

head\_panel.Height = 50;

//Menu flyout

//MenuFlyout menu = new MenuFlyout();

//Button tn = new Button();

//MenuFlyoutItemBase b ;

//ADDING THE EDIT AND DELETE STUFF

string[] stuff\_text = { "Edit", "Delete" };

string[] stuff\_image = { "", "" };

HyperlinkButton[] link = new HyperlinkButton[stuff\_text.Length];

Border[] brd = new Border[stuff\_text.Length];

int reduce = i;

if (i == counter)

{

reduce = i - 1;

}

for (int t = 0; t < stuff\_text.Length; t++)

{

// option\_border.BorderThickness = new Thickness(0, 0, 0, 1);

//option\_border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

Image img = new Image();

ImageBrush source = new ImageBrush();//new Uri("Assets/icons/edit-black.png");

//source.ImageSource = source.

//img.source =

TextBlock option\_txt = new TextBlock();

option\_txt.Foreground = new SolidColorBrush(Color.FromArgb(150, 0, 0, 0));

option\_txt.FontSize = 15;

option\_txt.Foreground = new SolidColorBrush(Colors.White);

option\_txt.Text = stuff\_text[t];

option\_txt.Margin = new Thickness(5, 0, 0, 0);

option\_txt.VerticalAlignment = VerticalAlignment.Center;

// Bitm

// Button button = new Button();

link[t] = new HyperlinkButton();

brd[t] = new Border();

brd[t].Background = selected\_color;

if (option\_txt.Text == stuff\_text[1])

brd[t].Background = new SolidColorBrush(Colors.IndianRed);

brd[t].BorderBrush = new SolidColorBrush(Color.FromArgb(53,0,0,0));

brd[t].BorderThickness = new Thickness(1);

link[t].Foreground = new SolidColorBrush(Colors.White);

link[t].Content = option\_txt;

link[t].Margin = new Thickness(0);

brd[t].CornerRadius = new CornerRadius(3);

// brd[t].Padding = new Thickness(-5);

brd[t].MaxWidth = 85;

brd[t].MaxHeight = 38;

link[t].Click += (s, e) =>

{

if (option\_txt.Text == stuff\_text[0])

{//editing time

//call the editing stuff

if (!Editing.IsOpen)

{

//OPEN

this.Frame.Opacity = .4;

parent.IsEnabled = false;

//SET THE EDIT TRACKER

//I DID (i-1) BECAUSE IT WAS INCREASE BY ONE, THEREFORE OUT OF BOUND ISH.

tracker = new EditTracker("reports", "report\_id", q.Results[(reduce), 0]);

//track the row.

tracker.Track();

this.r\_id = tracker.TrackResult[0];//set the id to the global var, so that we know what to edit.

r\_name\_e.Text = tracker.TrackResult[1];

r\_content\_e.Text = tracker.TrackResult[2];

Query a = new Query();

a.Get("tests", "Where test\_id = '" + q.Results[reduce, 3] + "'");

a.Record();

r\_tests\_e.SelectedIndex = f.SelectedInverse(a.Result[0], "tests", null, 1);

Editing.IsOpen = true;

}

}

else if (option\_txt.Text == stuff\_text[1])

{//delete

//CALL THE DELETE QUERY

//track first

tracker = new EditTracker("Reports", "report\_id", q.Results[(reduce), 0]);

//track

tracker.Track();

//CALL THE DELETE QUERY

Query que = new Query();

que.successMessage = "Report has been deleted successfully.";

que.Remove("reports", "where report\_id = '" + tracker.TrackResult[0] + "'");

(App.Current as App).chosen\_test\_for\_report = "";

this.Frame.Navigate(typeof(Reports), null);

}

else

{

PopUp p = new PopUp("Yes");

}

};

//add

brd[t].Child = link[t];

inner\_head2.Children.Add(brd[t]);

}

// inner\_head2.Children.Add();

//######################## /head

//######################## body

Border body\_border = new Border();

body\_border.BorderThickness = new Thickness(0, 1, 0, 1);

body\_border.BorderBrush = new SolidColorBrush(Color.FromArgb(40, 0, 0, 0));

// body\_border.Margin = new Thickness(0, 5, 0, 0);

StackPanel body\_panel = new StackPanel();

body\_panel.Background = body\_bg;

TextBlock body\_txt = new TextBlock();

body\_txt.Text = q.Results[i, 2];

body\_txt.Foreground = text\_color;

body\_txt.FontSize = 16;

body\_txt.Padding = new Thickness(10, 7, 8, 9);

body\_txt.CharacterSpacing = 12;

// body\_txt.TextTrimming = TextTrimming.CharacterEllipsis;

body\_txt.TextWrapping = TextWrapping.Wrap;

body\_txt.IsTextSelectionEnabled = true;

//######################## /body

//######################## footer

StackPanel footer\_panel = new StackPanel();

footer\_panel.Background = footer\_bg;

StackPanel inner\_footer\_panel1 = new StackPanel();

inner\_footer\_panel1.Margin = new Thickness(10, 10, 5, 7);

inner\_footer\_panel1.Orientation = Orientation.Horizontal;

StackPanel inner\_footer\_panel2 = new StackPanel();

inner\_footer\_panel2.Margin = new Thickness(5, -25, 10, 7);

inner\_footer\_panel2.Orientation = Orientation.Horizontal;

inner\_footer\_panel2.HorizontalAlignment = HorizontalAlignment.Right;

TextBlock footer\_txt1 = new TextBlock();

TextBlock footer\_txt2 = new TextBlock();

TextBlock footer\_txt3 = new TextBlock();

TextBlock footer\_txt4 = new TextBlock();

footer\_txt1.Text = "Created: ";

footer\_txt1.Foreground = f\_text\_color;

footer\_txt1.FontSize = footer\_txt\_size;

footer\_txt1.CharacterSpacing = 12;

footer\_txt2.Text = f.Date(q.Results[i, 4], "");

footer\_txt2.Foreground = f\_text\_color2;

footer\_txt2.FontSize = 13;

footer\_txt2.CharacterSpacing = 12;

footer\_txt2.Margin = new Thickness(5, 0, 0, 0);

footer\_txt3.Text = "Last Updated: ";

footer\_txt3.Foreground = f\_text\_color;

footer\_txt3.FontSize = footer\_txt\_size;

footer\_txt3.CharacterSpacing = 12;

footer\_txt4.Text = f.CheckEmpty(f.Date(q.Results[i, 5], ""), "Never");

footer\_txt4.FontStyle = Windows.UI.Text.FontStyle.Italic;

footer\_txt4.Foreground = f\_text\_color2;

footer\_txt4.FontSize = 13;

footer\_txt4.CharacterSpacing = 12;

footer\_txt4.Margin = new Thickness(5, 0, 0, 0);

//######################## /footer

int calc = ot % 2;

if (calc == 1)

{

head\_border\_t.BorderBrush = new SolidColorBrush(Color.FromArgb(255, 120, 28, 45));

}

ot++;

//adding

inner\_head1.Children.Add(head\_txt);

inner\_head1.Children.Add(head\_txt\_sub);

inner\_head.Children.Add(inner\_head1);

inner\_head.Children.Add(inner\_head2);

head\_panel.Children.Add(head\_border\_t);

head\_panel.Children.Add(inner\_head);

//head\_panel.Children.Add(head\_border\_b);

body\_panel.Children.Add(body\_txt);

body\_border.Child = body\_panel;

//adding footer stuff

inner\_footer\_panel1.Children.Add(footer\_txt1);

inner\_footer\_panel1.Children.Add(footer\_txt2);

inner\_footer\_panel2.Children.Add(footer\_txt3);

inner\_footer\_panel2.Children.Add(footer\_txt4);

footer\_panel.Children.Add(inner\_footer\_panel1);

footer\_panel.Children.Add(inner\_footer\_panel2);

parent\_panel.Children.Add(head\_panel);

parent\_panel.Children.Add(body\_border);

parent\_panel.Children.Add(footer\_panel);

parent\_border.Child = parent\_panel;

main\_p.Children.Add(parent\_border);

}

}

}

else

{

//NO RECORD

Messages msg = new Messages();

msg.Note("No Reports record available.");

reports\_r.Children.Add(msg.\_Container);

}

////////////////////////////////////////////////

//fill some stuff in the form

Query query = new Query();

query.Get("Tests");

int row\_count = query.CountRows();

query.Records();

for (int r = 0; r < row\_count; r++)

{

object p = query.Results[r, 1]+" #("+query.Results[r,0]+")";

// t\_patients.Items.Add(p[r]);

r\_tests.Items.Add(p);

r\_tests\_e.Items.Add(p);

}

}

//Cancel BTN

private void Cancel\_Click(object sender, RoutedEventArgs e)

{ if (Editing.IsOpen)

{

//Close

this.Frame.Opacity = 1;

//this.Opacity = .4;

parent.IsEnabled = true;

Editing.IsOpen = false;

}

}

//add record click btn

private void AddRecord\_Click(object sender, RoutedEventArgs e)

{

//CHECK EVERYTHING

PopUp p;

Filter f = new Filter();

string report\_content = f.Trim(r\_content.Text);

string report\_name = f.Trim(r\_name.Text);

string report\_test = f.Trim(r\_tests.SelectedIndex.ToString());

if (!(report\_content == "" || (report\_test == "" || report\_test == "0") || report\_name == "" ))

{

//GOOD TO GO

string[] r\_t = f.Selected(r\_tests.SelectedIndex, "Tests", null, 1);

report\_test = r\_t[0];

Query q = new Query();

string[] val = { report\_test, report\_name, report\_content, "Now()" };

q.successMessage = "Report "+report\_name+" has been added successfully!";

q.Add("reports", val, "test\_id,report\_name,report,date\_time");

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Reports), null);

}

else

{

if (report\_content == "")

{

p = new PopUp("Please say something about the report");

}

else if (report\_name == "")

{

p = new PopUp("Please give the report a name");

}

else if (report\_test == "" || report\_test == "0")

{

p = new PopUp("Please select a test the report belongs to");

}

else

{

p = new PopUp("Sorry, an error occured. The report couldn't be added. Please try again later");

}

}

}

private void EditRecord\_Click(object sender, RoutedEventArgs e)

{

//CHECK EVERYTHING

PopUp p;

Filter f = new Filter();

string report\_content = f.Trim(r\_content\_e.Text);

string report\_name = f.Trim(r\_name\_e.Text);

string report\_test = f.Trim(r\_tests\_e.SelectedIndex.ToString());

if (!(report\_content == "" || (report\_test == "" || report\_test == "0") || report\_name == ""))

{

//GOOD TO GO

string[] r\_t = f.Selected(r\_tests\_e.SelectedIndex, "Tests", null, 1);

report\_test = r\_t[0];

Query q = new Query();

//##################################CHECKING IF ANYTHING WAS EDITED

q.GetExtra("SELECT report\_name,test\_id,report FROM reports WHERE report\_id = '" + r\_id + "'");

q.Record();

string former\_name = q.Result[0];

string former\_test = q.Result[1];

string former\_content = q.Result[2];

//check if anything was edited

if(report\_name == former\_name && report\_test == former\_test && report\_content == former\_content)

{//the same thing,nothing was edited.

p = new PopUp("Nothing in this report was edited. click cancel or edit a value and submit");

}

else{

string[] val = { report\_test, report\_name, report\_content, "Now()" };

q.successMessage = "Report " + report\_name + " has been edited successfully!";

q.Change("reports", "test\_id,report\_name,report,date\_time\_updated",val,"WHERE report\_id = '"+r\_id+"'");

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Reports), null);

}

}

else

{

if (report\_content == "")

{

p = new PopUp("Please say something about the report");

}

else if (report\_name == "")

{

p = new PopUp("Please give the report a name");

}

else if (report\_test == "" || report\_test == "0")

{

p = new PopUp("Please select a test the report belongs to");

}

else

{

p = new PopUp("Sorry, an error occured. The report couldn't be added. Please try again later");

}

}

}

}

}

Doctors XAML

<Page

x:Class="Lpms.Doctors"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:Lpms"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

<Page.Resources>

<!-- TABLE HEAD BORDER -->

<Style x:Key="table\_head\_border" TargetType="Border">

<Setter Property="BorderBrush" Value="#19000000" />

<Setter Property="BorderThickness" Value="0 0 1 1"/>

<!--<Setter Property="Background" Value="#E8F3F0"/>-->

<Setter Property="Padding" Value="0"/>

<Setter Property="Margin" Value="0 0 0 0 "/>

</Style>

<!-- table head-->

<Style x:Key="table\_head" TargetType="StackPanel">

<Setter Property="MinWidth" Value="140"/>

<Setter Property="Background" Value="#E7e7e7"/>

<Setter Property="Margin" Value="0 0 0 0"/>

<Setter Property="VerticalAlignment" Value="Stretch"/>

</Style>

<!--table head text -->

<Style x:Key="table\_head\_txt" TargetType="TextBlock">

<Setter Property="Foreground" Value="#cc000000"/>

<Setter Property="Margin" Value="10 10 10 10"/>

<Setter Property="FontSize" Value="17"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

<Setter Property="VerticalAlignment" Value="Center"/>

</Style>

</Page.Resources>

<Grid>

<!--PARENT STACKPANEL SO THAT THE STUPID ELEMENTS DON'T OVERLAP-->

<ContentControl x:Name="parent">

<StackPanel>

<!-- record title-->

<StackPanel>

<Border Style="{StaticResource record\_label\_border}">

<StackPanel Style="{StaticResource record\_label}">

<TextBlock Text="Doctors Records" Style="{StaticResource record\_label\_txt}"/>

<Button x:Name="Add" Click="Add\_Click" ClickMode="Press" Style="{StaticResource add\_btn}">

<StackPanel Orientation="Horizontal">

<Image Source="Assets/Icons/plus-white.png"/>

<TextBlock Text=" Add"/>

</StackPanel>

</Button>

</StackPanel>

</Border>

</StackPanel>

<!--TABLE HEAD-->

<Grid x:Name="doctors\_r" Width="auto" HorizontalAlignment="Stretch" Margin="0 0 0 0">

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="auto"/>

</Grid.ColumnDefinitions>

<Border Grid.Row="0" Grid.Column="0" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Name" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="1" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Phone" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="2" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Address" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="3" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Gender" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="4" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Specialty" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="5" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Tests Performed" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="6" Style="{StaticResource table\_head\_border}" BorderThickness="0 1 0 1">

<StackPanel Style="{StaticResource table\_head}">

<TextBlock Text="Date Registered" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

<Border Grid.Row="0" Grid.Column="7" Style="{StaticResource table\_head\_border}">

<StackPanel Style="{StaticResource table\_head\_option}">

<TextBlock Text="Options" Style="{StaticResource table\_head\_txt}"/>

</StackPanel>

</Border>

</Grid>

<!--CONTENTS-->

</StackPanel>

</ContentControl>

<Popup x:Name="Adding" IsOpen="False" Width="700" Height="auto" Margin="0 40 0 0">

<StackPanel Background="#4BA783">

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1" Width="auto" Margin="0 10 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<TextBox x:Name="d\_name" Width="auto" PlaceholderText="Doctor's Name (Firstname + Surname)" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 100 10 20"/>

<TextBox x:Name="d\_address" Width="auto" PlaceholderText="Doctor's address" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<TextBox x:Name="d\_phone" Width="auto" PlaceholderText="Phone number" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<ComboBox x:Name="d\_gender" Width="auto" PlaceholderText="Doctor's Gender" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select Doctor's Gender</ComboBoxItem>

<ComboBoxItem IsEnabled="True">Male</ComboBoxItem>

<ComboBoxItem IsEnabled="True">Female</ComboBoxItem>

</ComboBox>

<TextBox x:Name="d\_spec" Width="auto" PlaceholderText="Speciality" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<Button x:Name="Cancel\_Add" Click="Cancel\_Click" HorizontalAlignment="Left" Content="Cancel" FontSize="20" Width="120" Height="60" Margin="0 0 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#cc000000" Foreground="#ffffff"/>

<Button x:Name="AddRecord" Click="AddRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" Margin="0 -67 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</Popup>

<!-- EDITING POPUP-->

<Popup x:Name="Editing" IsOpen="False" Width="700" Height="auto" Margin="0 40 0 0">

<StackPanel Background="#4BA783">

<!--<StackPanel x:Name="title" Width="auto" Height="100" Background="#ffffff" Margin="0 5 0 0">

<TextBlock Text="Add Tests" Foreground="Gray" FontSize="30" HorizontalAlignment="Center" Margin="10 0 55 10"/>

</StackPanel>-->

<Border BorderBrush="#33000000" BorderThickness="1" Width="auto" Margin="0 10 0 10">

<StackPanel Background="#ffffff" Width="700" MinHeight="400" Orientation="vertical" Margin="0 0 0 0">

<TextBox x:Name="d\_name\_e" Width="auto" PlaceholderText="Doctors's Name (Firstname + Surname)" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 100 10 20"/>

<TextBox x:Name="d\_address\_e" Width="auto" PlaceholderText="Doctors's address" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<TextBox x:Name="d\_phone\_e" Width="auto" PlaceholderText="Phone number" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<ComboBox x:Name="d\_gender\_e" Width="auto" PlaceholderText="Doctor's Gender" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40">

<ComboBoxItem Foreground="#33000000" IsEnabled="False">Select Doctor's Gender</ComboBoxItem>

<ComboBoxItem IsEnabled="True">Male</ComboBoxItem>

<ComboBoxItem IsEnabled="True">Female</ComboBoxItem>

</ComboBox>

<TextBox x:Name="d\_spec\_e" Width="auto" PlaceholderText="Speciality" Height="50" Padding="10 15 0 0" BorderBrush="#33000000" Background="#f9f9f9" BorderThickness="1" Margin="10 30 10 40"/>

<Button x:Name="Cancel\_Edit" Click="Cancel\_Click" HorizontalAlignment="Left" Content="Cancel" FontSize="20" Width="120" Height="60" Margin="0 0 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#cc000000" Foreground="#ffffff"/>

<Button x:Name="EditRecord" Click="EditRecord\_Click" HorizontalAlignment="Right" Content="Submit" FontSize="20" Width="120" Height="60" Margin="0 -67 10 0" BorderBrush="#26000000" BorderThickness="1" Background="#4ba783" Foreground="#ffffff"/>

</StackPanel>

</Border>

</StackPanel>

</Popup>

</Grid>

</Page>

Doctors C#

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.UI;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

using Lpms.Classes;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace Lpms

{

/// <summary>

/// An empty page that can be used on its own or navigated to within a Frame.

/// </summary>

public sealed partial class Doctors : Page

{

string d\_id = string.Empty;

EditTracker tracker;

public Doctors()

{

this.InitializeComponent();

//start displaying the doctors records

Query q = new Query();

Query inner\_q = new Query();

q.GetExtra("SELECT name,phone,address,gender,specialisation,date FROM doctors");

q.Records();

int doctors\_row = q.CountRows();

Query doctor\_id = new Query();

doctor\_id.Get("doctors");

doctor\_id.Records();

//AMOUNT OF COLUMN DEFINITIONS

int col\_num = Convert.ToInt32(doctors\_r.ColumnDefinitions.Count);

Filter f = new Filter();

bool check\_records = q.CheckRecords();

// RowDefinition[] rd = new RowDefinition[q.CountRows()];

// RowDefinition rd;

//STACKPANEL TO HOLD THE ROWS

// StackPanel parent\_row\_panel;

//WE'RE ADDING +1 CAUSE THERE'S ALREADY A ROW DEFINED WHICH IT'S VALUE WOULD BE ZERO

//SO IF WE DONT INCREMENT IT, IT STARTS WITH 0, WHICH MIGHT BE A BAD IDEA :(.

//int i = 0;

//now check the number of rows

//keep adding rows as long as it doesn't exceed the row limit from the stupid database.

if (check\_records)

{

int i = 0;

for (i = 0; i < (doctors\_row); i++)

{

RowDefinition rd = new RowDefinition();

//rd.Height = new GridLength(50);

doctors\_r.RowDefinitions.Add(rd);

/\*

\* COLUMN DEFINITION

\* CREATE COLUMNS ACCORDING TO THE PARENT COLUMN OF THE FIRST ROW, WHICH IS col\_num.

\*/

//////////////////////////////////

int k = 0;

for (k = 0; k < col\_num; k++)

{

ColumnDefinition cd = new ColumnDefinition();

//cd.Width = new GridLength();

//cd.MinWidth = 150;

doctors\_r.ColumnDefinitions.Add(cd);

//reset column def so that when it starts looping the for again, it creates a new set of column definition

//column\_def = 0;

//////////////////////////////////////////////////////////

Border border = new Border();

TextBlock n = new TextBlock();

n.FontSize = 17.3;

n.Foreground = new SolidColorBrush(Color.FromArgb(180, 0, 0, 0));

///////////////////////////////////////

//stackpanel

StackPanel pa = new StackPanel();

// pa.Style = new Style();

//StackPanel m\_pa = new StackPanel();

// m\_pa.Name = "p";// + i.ToString();

StackPanel row\_holder = new StackPanel();

//row\_holder.Orientation = new Orientation();

// Orientation orie = row\_holder1.Orientation;

//add

//pa.

pa.Children.Add(n);

pa.Margin = new Thickness(10);

//pa.MaxWidth = 115;

pa.HorizontalAlignment = HorizontalAlignment.Center;

//border.Style = new Windows.UI.Xaml.Style();

//border.Style.TargetType = typeof(Border);

border.BorderThickness = new Thickness(0, 0, 0, 1);

border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

//calculate odd and even ish

int calc = i % 2;

//border.MaxWidth = 115;

// border.MinHeight = 61;

border.Child = pa;

row\_holder.Children.Add(border);

int calcu = col\_num - k;

//CHECK WHAT TO INPUT

if(k == 3)

{//gender

n.Text = f.Gender(q.Results[i, k]);

}

else if (k == 5)

{//tests results

inner\_q.Get("tests", "WHERE doctor\_id = '" + doctor\_id.Results[i, 0] + "'");

n.Text = f.Numbers(inner\_q.CountRows());

}

else if (k == 6)

{//DATE REGISTERED

n.Text = f.Date(q.Results[i,5],"dd-MM-yyyy");

}

else if (calcu == 1)

{//LAST VALUE

}

else

{

n.Text = q.Results[i, k];

}

//TextBlock

//////////////OPTION

StackPanel option\_panel = new StackPanel();

Border option\_border = new Border();

//////////////////////////////////////////////////////

////FOR CHANGING THE ROW BACKGROUND.

if (calc == 1)

{

border.Background = new SolidColorBrush(Color.FromArgb(250, 215, 215, 215));

option\_border.Background = new SolidColorBrush(Color.FromArgb(250, 215, 215, 215));

}

else

{

border.Background = new SolidColorBrush(Color.FromArgb(250, 236, 254, 251));

option\_border.Background = new SolidColorBrush(Color.FromArgb(250, 226, 246, 241));

}

//DETERMINE WHEN TO ADD THE OPTION STUFF

if (calcu == 1)

{//CREATE THE OPTIONS STUFF

StackPanel option\_inner\_panel = new StackPanel();

//TO MAKE THINGS EASY JARE,LOOP IT THROUGH THE ARRAY

string[] stuff\_text = { "Edit", "Delete" };

string[] stuff\_image = { "", "" };

HyperlinkButton[,] link = new HyperlinkButton[doctors\_row, stuff\_text.Length];

int reduce = i;

if (i == doctors\_row)

{

reduce = i - 1;

}

for (int t = 0; t < stuff\_text.Length; t++)

{

option\_border.BorderThickness = new Thickness(0, 0, 0, 1);

option\_border.BorderBrush = new SolidColorBrush(Color.FromArgb(50, 0, 0, 0));

TextBlock option\_txt = new TextBlock();

option\_txt.Text = stuff\_text[t];

option\_txt.Foreground = new SolidColorBrush(Color.FromArgb(150, 0, 0, 0));

option\_txt.FontSize = 15;

option\_txt.Margin = new Thickness(10, 0, 0, 0);

option\_txt.VerticalAlignment = VerticalAlignment.Center;

// Button button = new Button();

link[i, t] = new HyperlinkButton();

//HyperlinkButton[] link = new HyperlinkButton[tests\_row];

link[i, t].Content = option\_txt.Text;

// link.Click = RoutedEventHandler.Combine(sender, RoutedEventArgs e);

//button.ClickMode = ClickMode.Hover;

//CLICK EVENT

int reduce\_t = t;

if (t == stuff\_text.Length)

{

reduce\_t = t - 1;

}

//link[i].Click += new RoutedEventHandler(EditRecord\_Click("E",));

link[i, t].Click += (s, e) =>

{

if ((string)link[reduce, reduce\_t].Content == stuff\_text[0])

{//IT's edit

if (!Editing.IsOpen)

{

//OPEN

this.Frame.Opacity = .4;

//this.Opacity = .4;

parent.IsEnabled = false;

//SET THE EDIT TRACKER

//I DID (i-1) BECAUSE IT WAS INCREASE BY ONE, THEREFORE OUT OF BOUND ISH.

tracker = new EditTracker("doctors", "doctor\_id", doctor\_id.Results[(reduce), 0]);

//track the row.

tracker.Track();

this.d\_id = tracker.TrackResult[0];//set the id to the global var, so that we know what to edit.

d\_name\_e.Text = tracker.TrackResult[1];

d\_address\_e.Text = tracker.TrackResult[2];

d\_phone\_e.Text = tracker.TrackResult[3];

d\_spec\_e.Text = tracker.TrackResult[5];

//CHECK GENDER

if (tracker.TrackResult[4].ToLower() == "true" || tracker.TrackResult[4].ToLower() == "1")

{//male

d\_gender\_e.SelectedIndex = 1;

}

else if (tracker.TrackResult[4].ToLower() == "false" || tracker.TrackResult[4].ToLower() == "0")

{//female

d\_gender\_e.SelectedIndex = 2;

}

Editing.IsOpen = true;

}

}

else if ((string)link[reduce, reduce\_t].Content == stuff\_text[1])

{//it's the delete.

//CALL THE DELETE QUERY

//track first

tracker = new EditTracker("doctors", "doctor\_id", doctor\_id.Results[(reduce), 0]);

//track

tracker.Track();

//CALL THE DELETE QUERY

Query que = new Query();

que.successMessage = "Doctor " + f.FirstName(tracker.TrackResult[1]) + " has been deleted successfully.";

que.Remove("doctors", "where doctor\_id = '" + tracker.TrackResult[0] + "'");

this.Frame.Navigate(typeof(Doctors), null);

}

};

////ADDING

option\_inner\_panel.Children.Add(link[i, t]);

}

option\_panel.Children.Add(option\_inner\_panel);

option\_inner\_panel.Orientation = Orientation.Horizontal;

option\_panel.VerticalAlignment = VerticalAlignment.Center;

option\_border.Child = option\_panel;

Grid.SetColumn(option\_border, k);

Grid.SetRow(option\_border, i + 1);

doctors\_r.Children.Add(option\_border);

}

else

{

Grid.SetRow(row\_holder, i + 1);

Grid.SetColumn(row\_holder, k);

doctors\_r.Children.Add(row\_holder);

}

}

}

}

else

{

//

Messages msg = new Messages();

msg.Note("No Doctors Record Available.");

doctors\_r.Children.Add(msg.\_Container);

RowDefinition rd = new RowDefinition();

// rd.Height = new GridLength(50);

doctors\_r.RowDefinitions.Add(rd);

Grid.SetColumnSpan(msg.\_Container, col\_num);

Grid.SetColumn(msg.\_Container, 0);

Grid.SetRow(msg.\_Container, 1);

}

}

//ADD BTN

private void Add\_Click(object sender, RoutedEventArgs e)

{

if (!Adding.IsOpen)

{

//OPEN

// (App.Current as App).ParentOpacity = .4;

this.Frame.Opacity = .4;

//this.Opacity = .4;

parent.IsEnabled = false;

Adding.IsOpen = true;

}

}

//Cancel BTN

private void Cancel\_Click(object sender, RoutedEventArgs e)

{

if (Adding.IsOpen)

{

//Close

// (App.Current as App).ParentOpacity = .4;

this.Frame.Opacity = 1;

//this.Opacity = .4;

parent.IsEnabled = true;

Adding.IsOpen = false;

}

if (Editing.IsOpen)

{

this.Frame.Opacity = 1;

//this.Opacity = .4;

parent.IsEnabled = true;

Editing.IsOpen = false;

}

}

//add record click btn

private void AddRecord\_Click(object sender, RoutedEventArgs e)

{

//CHECK EVERYTHING

PopUp p;

Filter f = new Filter();

string doctor\_name = f.Trim(d\_name.Text);

string doctor\_phone = f.Trim(d\_phone.Text);

string doctor\_address = f.Trim(d\_address.Text);

string doctor\_spec = f.Trim(d\_spec.Text);

string doctor\_gender = f.Trim(d\_gender.SelectedIndex.ToString());

if (!(doctor\_name == "" || doctor\_address == "" || doctor\_phone == "" || doctor\_spec == "" || (doctor\_gender == "" || doctor\_gender == "0")))

{

//GOOD TO GO

//check for phone number

Query q = new Query();

if (doctor\_gender == "1")

{//male

doctor\_gender = "1";

}

else if (doctor\_gender == "2")

{//female

doctor\_gender = "0";

}

//doctor\_dob = f.Date(doctor\_dob, "yyyy-MM-dd");

string[] val = { doctor\_name, doctor\_address, doctor\_phone, doctor\_gender, doctor\_spec, "Now()" };

q.successMessage = "Dr "+f.FirstName(doctor\_name) + "'s record has been added successfully!";

q.Add("doctors", val, "name,address,phone,gender,specialisation,date");

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Doctors), null);

}

else

{

if (doctor\_name == "")

{

p = new PopUp("Please enter the name of the doctor");

}

else if (doctor\_address == "")

{

p = new PopUp("Please enter the doctor's address");

}

else if (doctor\_phone == "")

{

p = new PopUp("Please enter the doctor's phone number");

}

else if (doctor\_spec == "")

{

p = new PopUp("Please enter the doctor's area of specialty");

}

else if (doctor\_gender == "" || doctor\_gender == "0")

{

p = new PopUp("Please select the doctor's gender");

}

else

{

p = new PopUp("Sorry, an error occured. The doctor couldn't be added. Please try again later");

}

}

}

//FOR EDITING

private void EditRecord\_Click(object sender, RoutedEventArgs e)

{

// tracker.Track();

// t\_price.Text = tracker.TrackResult.Length.ToString();

//CHECK EVERYTHING

PopUp p;

Filter f = new Filter();

string doctor\_name = f.Trim(d\_name\_e.Text);

string doctor\_phone = f.Trim(d\_phone\_e.Text);

string doctor\_address = f.Trim(d\_address\_e.Text);

string doctor\_spec = f.Trim(d\_spec\_e.Text);

string doctor\_gender = f.Trim(d\_gender\_e.SelectedIndex.ToString());

if (!(doctor\_name == "" || doctor\_address == "" || doctor\_phone == "" || doctor\_spec == "" || (doctor\_gender == "" || doctor\_gender == "0")))

{

//GOOD TO GO

Query q = new Query();

if (doctor\_gender == "1")

{//male

doctor\_gender = "1";

}

else if (doctor\_gender == "2")

{//female

doctor\_gender = "0";

}

string[] val = { doctor\_name, doctor\_address, doctor\_phone, doctor\_gender, doctor\_spec };

string[] col = { "name", "address", "phone", "gender", "specialisation" };

q.Get("doctors", "Where doctor\_id = '" + d\_id + "'");

q.Records();

q.successMessage = "Dr "+f.FirstName(q.Results[0, 1]) + "'s record has been edited successfully!";

//MAKE THINGS EASIER JARE, LOOP THROUGH THE VAL, TO INSERT DYNAMICALLY

//I CANNOT COME AND BE WOUNDING MYSELF :-|

string condition = " Where doctor\_id = '" + this.d\_id + "'";

q.Change("doctors", "name,address,phone,gender,specialisation", val, condition);

this.Frame.Opacity = 1;

this.Frame.Navigate(typeof(Doctors), null);

}

else

{

if (doctor\_name == "")

{

p = new PopUp("Please enter the name of the doctor");

}

else if (doctor\_address == "")

{

p = new PopUp("Please enter the doctor's address");

}

else if (doctor\_phone == "")

{

p = new PopUp("Please enter the doctor's phone number");

}

else if (doctor\_spec == "")

{

p = new PopUp("Please enter the doctor's area of specialty");

}

else if (doctor\_gender == "" || doctor\_gender == "0")

{

p = new PopUp("Please select the doctor's gender");

}

else

{

p = new PopUp("Sorry, an error occured. The doctor couldn't be added. Please try again later");

}

}

}

}

}

© CopyRight All Rights Reserved.