MOBILE DEVICES PROJECT

Fun App

Service

Students: Atanasiu Justin

Ion Costin-Andrei

Gorgan Constantin

Stanciu Tudor

Untescu Mihai

Group: 1232E

Politehnica University of Bucharest

January 2015

**Introduction**

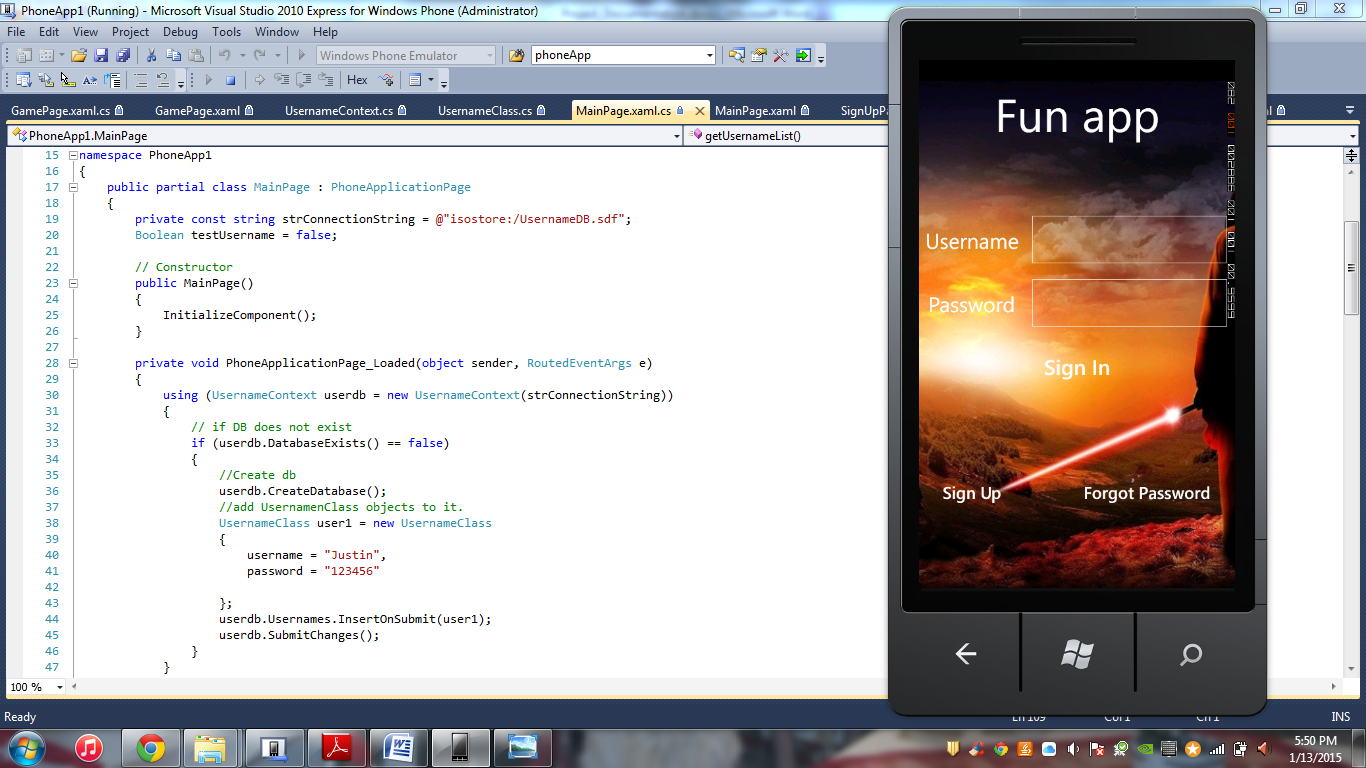
Fun App is an application built for Windows Phone operating system that allows users to check the weather in Constanta and Bucharest, Romania. The app displays the minimum and maximum temperature for the current and the next day.

The idea is simple, everyone who wants to use the app must first create an account with a user and a password. The credentials are saved into a local database which is created on to the Isolated Storage Container of the phone.

The information regarding the weather is obtained in the following way: a request is sent to http://weather.yahooapis.com/forecastrss?w= , the website then sends an xml file back to the phone. Finally the information on the xml file is parsed to usable information.

**How does it work**

**Page 1: Main Page**



As it can be seen, there is nothing really happening in the main page. The user can sign in by introducing his credentials. If the user does not have any he can click the "Sign Up" which sends him to SignUpPage.xaml, as follows:

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

{

NavigationService.Navigate(new Uri("/SignUpPage.xaml", UriKind.Relative));

}

If the user already has an account he can introduce his username and password in the text assigned text fields. The application checks for errors before signing in:

* if the account exists
* if the username is correct
* if the password matches the username
* if information is introduced in all the fields

Code:

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

{

IList<UsernameClass> UsernameList = this.getUsernameList();

testUsername = false;

// test if the textboxes are correct.

if (nameTextBox.Text == "" || passwordBox.Password == "")

{

MessageBoxResult message = MessageBox.Show("Please enter information in all fields", "Incorrect fields", MessageBoxButton.OK);

}

else

{

UsernameClass newUser = new UsernameClass

{

username = nameTextBox.Text.ToString(),

password = passwordBox.Password.ToString()

};

foreach (UsernameClass user in UsernameList)

{

// test if the username is already in the DB

if (user.username.ToString().Equals(newUser.username.ToString()))

{

testUsername = true;

if(user.password.ToString().Equals(newUser.password.ToString()))

{

NavigationService.Navigate(new Uri("/GamePage.xaml", UriKind.Relative));

}

else

{

MessageBoxResult message = MessageBox.Show("Wrong password", "Wrong password", MessageBoxButton.OK);

}

}

}

if (testUsername == false)

{

MessageBoxResult message = MessageBox.Show("Wrong username", "Wrong username", MessageBoxButton.OK);

}

}

}

**Page 2: Sign Up Page**



In this page the user can create an account in order to sign into the app. There are two fields: one for the username and one for the password. Also, in the bottom right corner of the page there is a "Back" button that sends you to the Main Page.

The username and the password must both have a minimum of six characters, otherwise the app displays error messages.

Another error message appears if you try to create an account with an already existent username.

Code:

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

{

IList<UsernameClass> UsernameList = this.getUsernameList();

if (nameTextBox.Text == "" || passwordBox.Password == "")

{

MessageBoxResult message = MessageBox.Show("Please enter information in all fields", "Incorrect fields", MessageBoxButton.OK);

}

else if(nameTextBox.Text.Length < 6)

{

MessageBoxResult message = MessageBox.Show("Username must be longer than 5 characters", "Short username", MessageBoxButton.OK);

}

else if(passwordBox.Password.Length < 6)

{

MessageBoxResult message = MessageBox.Show("Password must be longer than 5 characters", "Short password", MessageBoxButton.OK);

}

else

{

using (UsernameContext userdb = new UsernameContext(strConnectionString))

{

// make a new object of type username class with values from textbox

UsernameClass newUsername = new UsernameClass

{

username = nameTextBox.Text.ToString(),

password = passwordBox.Password.ToString(),

};

foreach (UsernameClass user in UsernameList)

{

// test if the username is already in the DB

if (user.username.ToString().Equals(newUsername.username.ToString()))

{

addBoolean = false;

}

}

if (addBoolean == false)

{ // if the username is in the DB already, a message box pops

MessageBox.Show("This username already exists");

}

else if(addBoolean == true)

{ // else, it submits the username.

userdb.Usernames.InsertOnSubmit(newUsername);

userdb.SubmitChanges();

MessageBoxResult result = MessageBox.Show("Username succesfully added", "Username added", MessageBoxButton.OK);

if (result == MessageBoxResult.OK)

{

NavigationService.Navigate(new Uri("/MainPage.xaml", UriKind.Relative));

}

}

// refresh addboolean

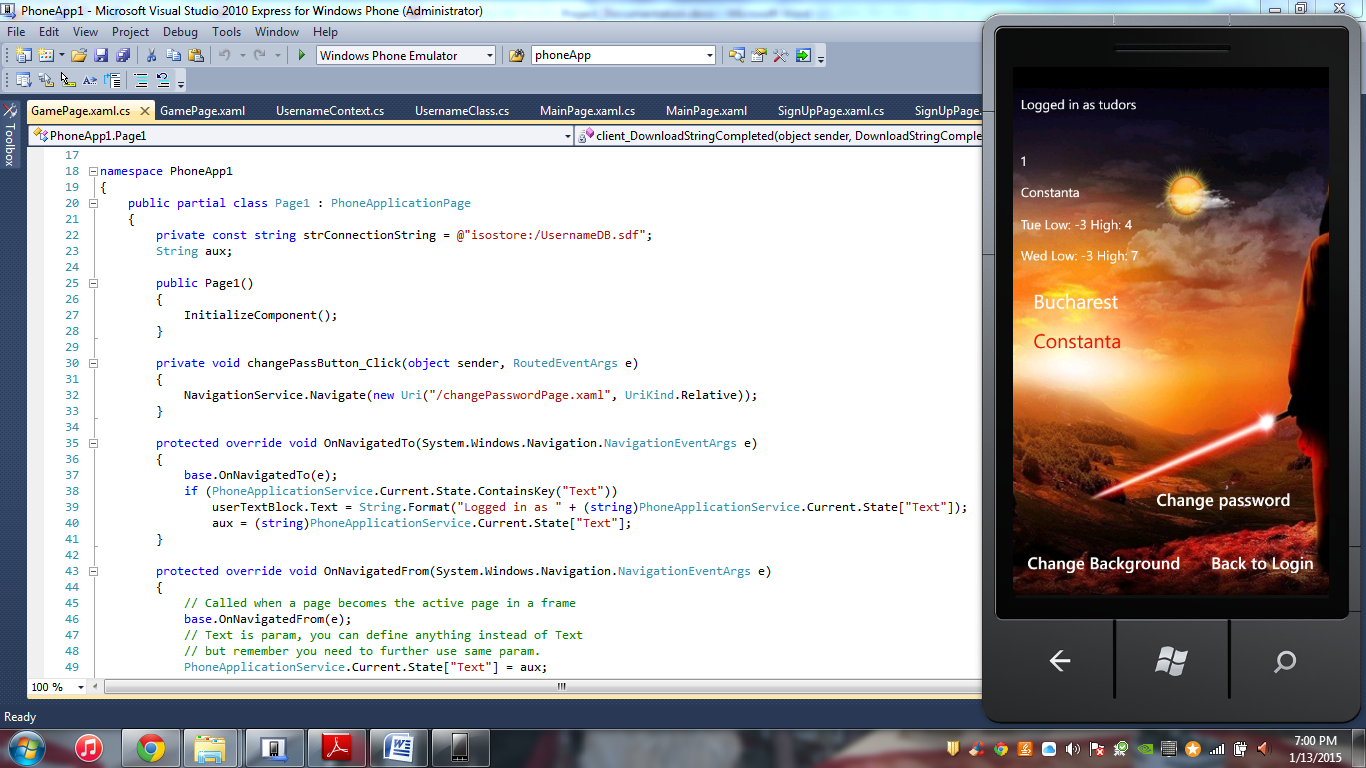
addBoolean = true;

}

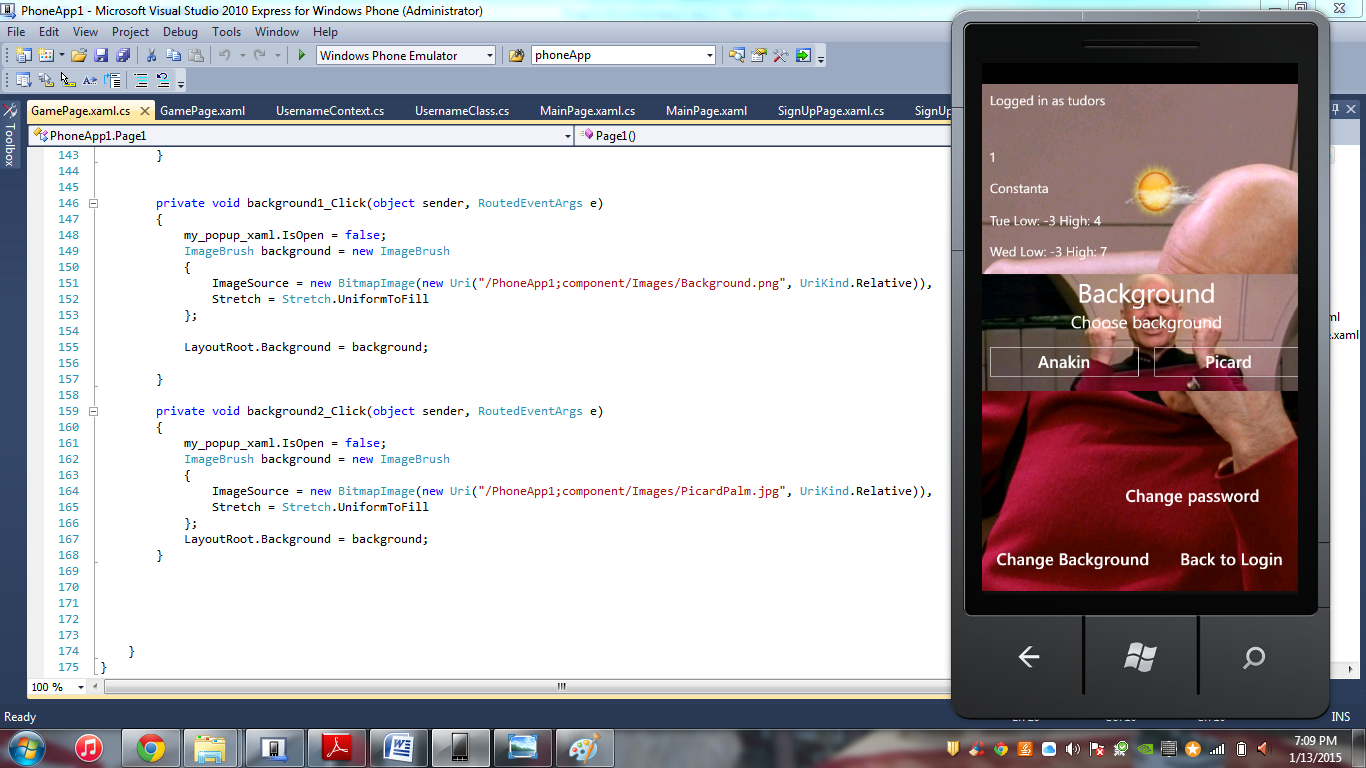
}

}

**Page 3: Game Page**



This is the most complex page of the application. As functionality, you can see the weather from either Bucharest or Constanta. The user can select the city. At that moment the app shows the highest and the lowest value of the temperature for the current and the following day and also a descriptive picture.

In the bottom left corner there is a "Change Background" button which the user can use in order to, you guessed it, change the background for the current page.

Code:

private void background1\_Click(object sender, RoutedEventArgs e)

{

my\_popup\_xaml.IsOpen = false;

ImageBrush background = new ImageBrush

{

ImageSource = new BitmapImage(new Uri("/PhoneApp1;component/Images/Background.png", UriKind.Relative)),

Stretch = Stretch.UniformToFill

};

LayoutRoot.Background = background;

}

private void background2\_Click(object sender, RoutedEventArgs e)

{

my\_popup\_xaml.IsOpen = false;

ImageBrush background = new ImageBrush

{

ImageSource = new BitmapImage(new Uri("/PhoneApp1;component/Images/PicardPalm.jpg", UriKind.Relative)),

Stretch = Stretch.UniformToFill

};

LayoutRoot.Background = background;

}

}

**User guide**

In terms of how to use it, the application is pretty much straight forward. There is an easy way to navigate between all the pages, as there are additional buttons implemented for this purpose, other than the ones of the phone itself.

The name of the buttons are also self-describing, which helps the app to be user-friendly.

**Conclusions**

As it is implemented in this moment, the application works correctly, delivering the user the pieces of information that he needs. It is obvious that there is a lot of room for improvement, but by further adding functionalities, it can become a decent Windows Store app.