Digital Marketing Project

N van der Linde

Project submitted for the degree *Baccalaureus Scientiae* in Information Technology at the North-West University

Supervisor: Dr. JT Janse van Rensburg
Co-supervisor: Mr. Z. Boonzaaier

Student number: 25909932

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1 INTRODUCTION AND OVERVIEW

1.1 Introduction

This project is about designing an application for a South African digital marketing company. They have expressed the need for a web application that serves as an online platform allowing users to store, view and share photos.

Users should be able to create an account and login before utilising the platform. Once logged in, users should be able to view their own photos as well as the photos shared with them. The user should be able to create and manage the metadata recorded for each photo that the user has access to. The ability to share content with other registered users also needs to be built into the web application. Users should have the ability to download photos that they have access to.

The web application should be designed as consistent, intuitive, easy to use and easy to navigate without needing to provide excessive user guide documentation. All information and media items that are stored should be stored securely and in accordance with POPI.

1.2 Highlights

The highlights for this project was the ERD diagram as well as the Use Case diagram. Drawing these diagrams was valuable. It helps to stay updated on what Use Case as well as ERD diagrams need to look like or contain.

The weekly Client Sessions was another highlight. To be able to interact with someone working in this type of area was beneficial. It was a very good learning opportunity. She provided a lot of solutions and help with this project.

The research for this project was another highlight. For example, to search online for what is meant by a certain aspect or guidelines on how to do something was really advantageous, because now I will remember how to do it again going forward. There were interesting approaches and guidelines that I never knew existed.

1.3 Challenges

A challenge for this project was to choose a technology stack to build this program. There are so many to choose from. Another challenge was the all the different ways this project could have been approached; there are so many ways to complete it.

1.4 System Overview

Users within the digital marketing company currently send the photos back and forth over email. The users with whom the photos are shared are included in the recipients list of the email, with any additional information included in the body of the email. The additional information does not follow any specific format and is often excluded completely (or lost between versions of mail trails).

The current process sometimes creates confusion and miscommunication about which photos are the latest versions and which users the photos have been shared with. There is currently no defined and standardised way of recording any additional information relevant to the photos. There are also size limits when sending emails. These size limits are not often breached but when they are, the limitation needs to be mitigated. By switching to a centralised online platform, that all users have access to, the highlighted challenges can be mitigated and resolved.

The biggest risk to the project would be any limitations that a technology stack would present. Any technology limitations that prohibit the functional implementation of core requirements would need to be mitigated using a technology stack (or intermediate workaround) that alleviates the blocker or limitation.

The recommended approach for this business case is to pursue the development of an online platform that allows users to create, manage and share content with other users, with the ability to capture the applicable additional information (in the form of metadata). The content to be uploaded onto the platform should be grouped using albums and any content that has metadata recorded should be searchable by its metadata.

There are different technology stacks that need to be implemented for this project to run successfully. The following need to be used:

- GIT: this is used for the source control. GitHub was used for this project's source control.
- Backend (API): .NET Core was used for the backend of this project.
- Backend (Database): SQL server (a relational database in Oracle) was used to design the database.
- Frontend (Design): C# (ASP.NET) and CSS was used for the frontend.
- Cloud Technology: Heroku was used to host this project.
- Diagrams: it is required to draw diagrams (ERD, Flow and Use Case diagrams) for this project. Any tool could have been used. For this project, draw.io was used to design the diagrams.

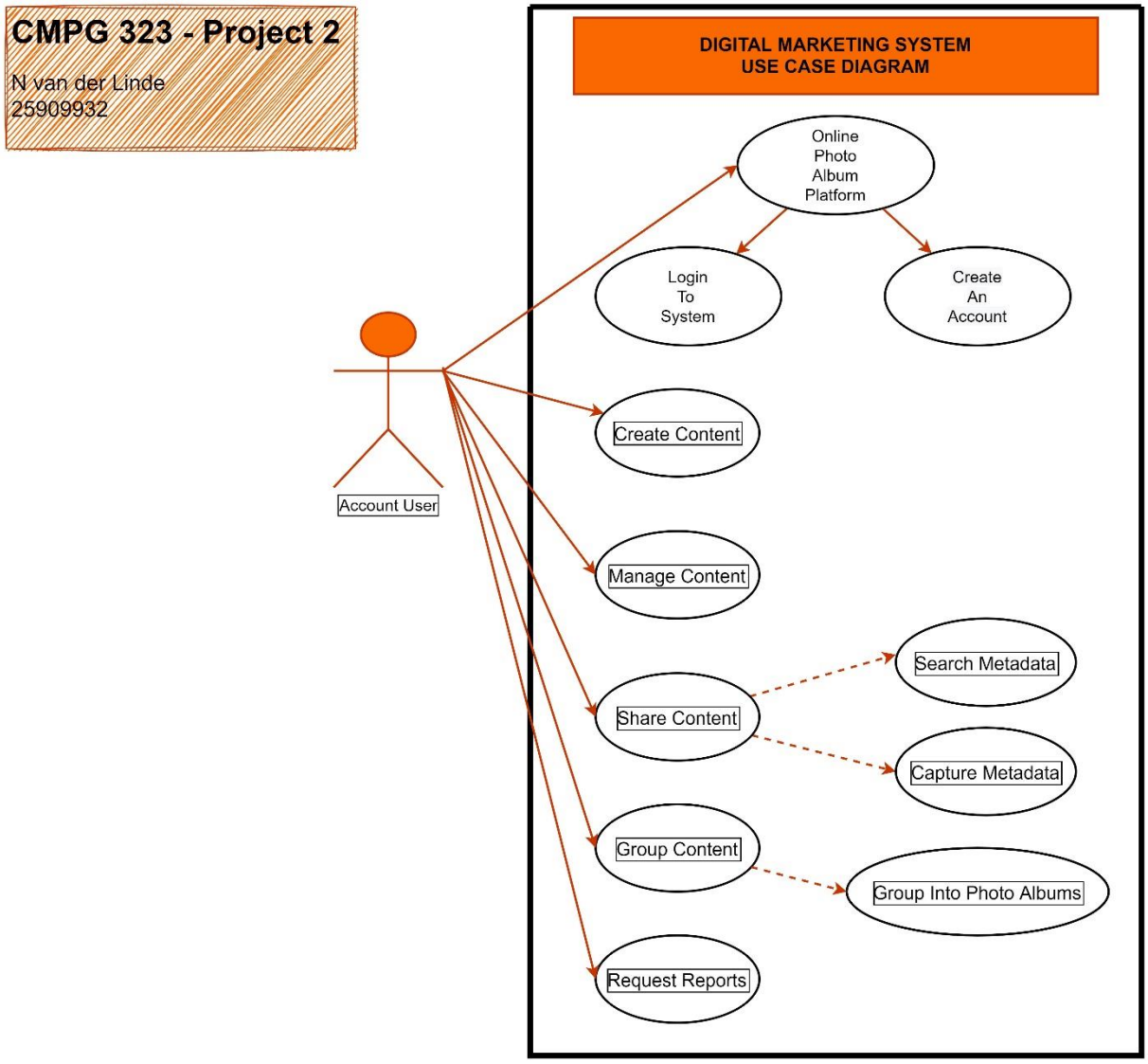
- Logging: the application will need logging. A logging framework was implemented.
- Code coverage: this is a type of tool, library, plugin, or framework that needs to be used to check the project's code coverage. A code coverage framework was implemented.

CHAPTER 2 USE CASE DIAGRAM

2.1 The Use of UML Diagrams

The Use Case Diagram is a diagram that depicts the interactions between the system and external systems and users. It graphically describes who will use the system and in what ways the user expects to interact with the system. It is also a set of modelling conventions that is used to specify or describe a software system in terms of objects. The use-case narrative is used in addition to textually describe the sequence of steps of each interaction.

Figure 2-1: Use Case Diagram



From the abovementioned diagram for the Digital Marketing System, the following can be derived:

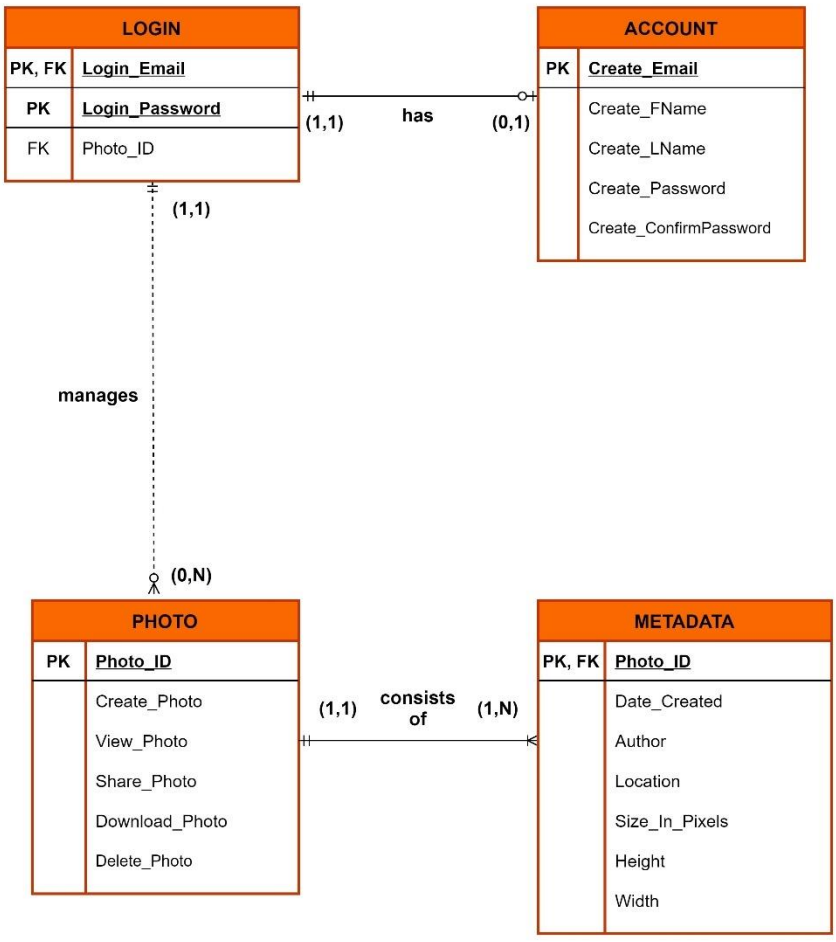
- Online Photo Album System: the account user will either login to his or her account, using their unique username and password, or create an account.
- Create Content: the user can create photos, with the ability to capture the applicable additional information (in the form of metadata) and any content that has metadata recorded should be searchable by its metadata.
- Manage Content: the user can manage photos, with the ability to capture the applicable additional information (in the form of metadata) and any content that has metadata recorded should be searchable by its metadata.
- Share Content: the user can share photos, with the ability to capture the applicable additional information (in the form of metadata) and any content that has metadata recorded should be searchable by its metadata.
- Group Content: the account user makes use of this so that the content that was created, viewed, or stored can be uploaded onto the platform and grouped using photo albums.
- Request Reports: an extra Use Case added. This is for the user to see what content was created, viewed, or stored. It is also to show the metadata created.

CHAPTER 3 ERD DIAGRAM

3.1 The Use of ERD Diagrams

The ERD diagram is a model or blueprint representing the technical implementation of the database. Therefore, the technical implementation for the Digital Marketing System is shown below:

Figure 3-1: ERD Diagram



From the abovementioned diagram for the Digital Marketing System, the following can be derived:

- Login: the Login entity is for the user to login into his or her account. The username, consisting of the user's email, and password have to be entered to login. The photo_id is for the user to enter that will take the user directly to their photos on the system.
- Account: this is if a user does not have an account on the Digital Marketing System. Then an account can be created by entering the new user's email, followed by their first name, last name and finally a password also must be created and confirmed.
- Photo: here the user can either choose to create photos, view photos, delete photos, share photos with other users or download photos.
- Metadata: in this entity all photos' metadata can be changed or altered accordingly. Anything from the date that the photo was created, or the author, or the location, or the size in pixels, or the height and the width of the photo can be reformed.

CHAPTER 4 FLOW DIAGRAMS

4.1 The Use of Flow Diagrams

The flow diagram below demonstrates the different application layering for the Digital Marketing System.

First off there is a Welcome Page. Followed by a Login Page. Then a Create Account Page comes next. Finally, a Photos Page will be the last page for this web application.

Below is a proper explanation of the flow diagram designed for the Digital Marketing System:

1. The Welcome Page:
    a. The user must choose to either login to their account or to create an account if one does not have an account.
    b. From there, the next page is loaded.

2. The Login Page:
    a. This page loads if the user already has an account with the Digital Marketing System.
    b. Here the user enters his or her credentials consisting of a username, which is the user's email, and a strong password.
    c. Then when that is finished, the next page is loaded.

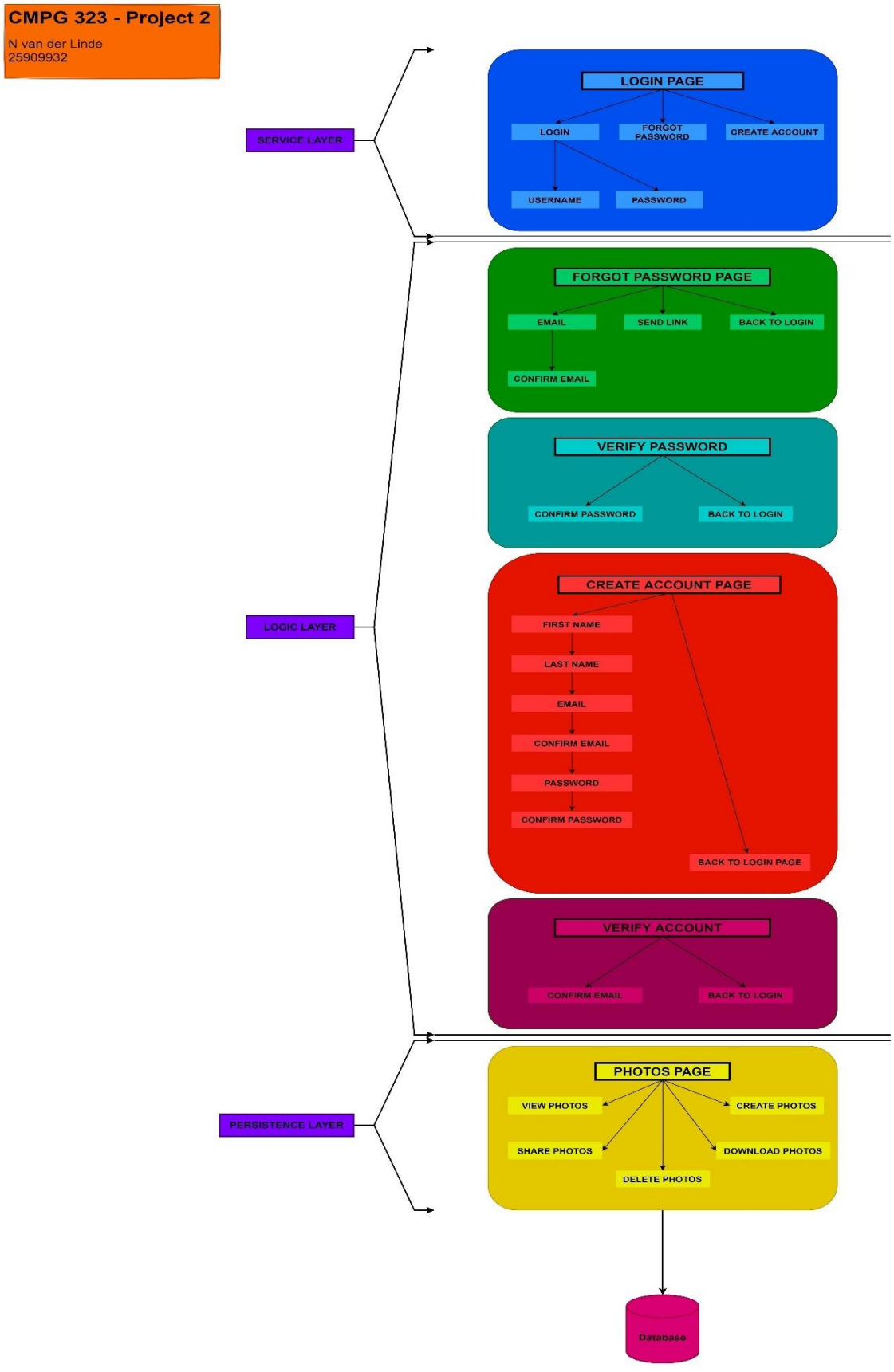
3. The Create Account Page or Register Page:
    a. This page loads if the user does not yet have an account with the Digital Marketing System.
    b. Here the user fills out the form. Their first name, last name, email as well as a strong password must be entered. The password also has to be confirmed by entering it again.
    c. Then when that is finished, the next page is loaded.

4. The Photos Page:
    a. This page loads once the user has logged in to their account.
    b. Here the user can either choose to create photos, view photos, share photos with other registered users or download their photos.

c. On this page the user can also manage the metadata of their photos on their account. They can either create new metadata for a photo or change the already existing metadata of a specific photo.

Lastly, all the services, layers, and pages connect to a Database.

Figure 4-1: Flow Diagram



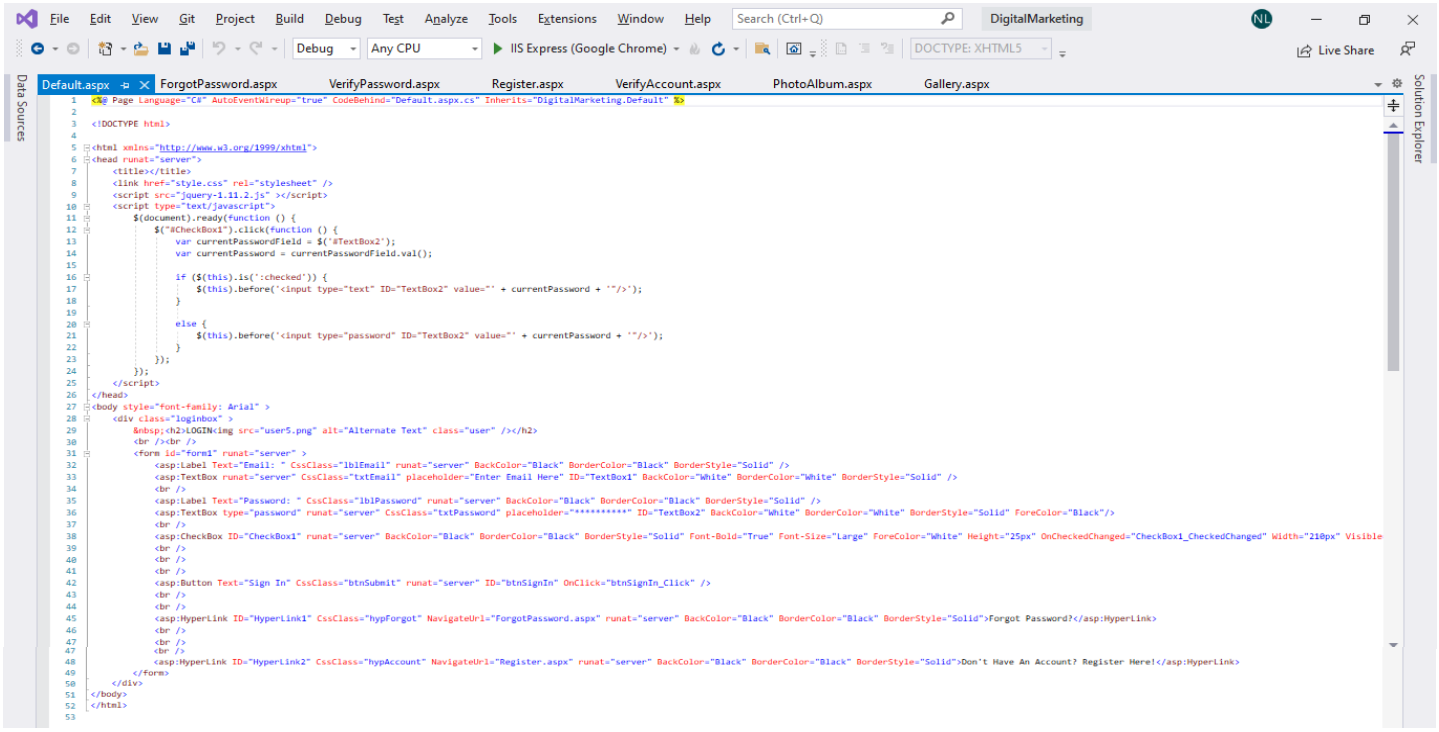
CHAPTER 5 THE USER GUIDE

5.1 The Default Page

The Default Web Form Page:

- This C# Class is the main page. When a user runs this system, this is the first page they will see. The user can login to his or her account by entering their email address as well as their password. The password will not be shown when the application is running; javascript was used to hide the user's password. The source code for this page is specified below:

Figure 5-1: The Default Web Form Page

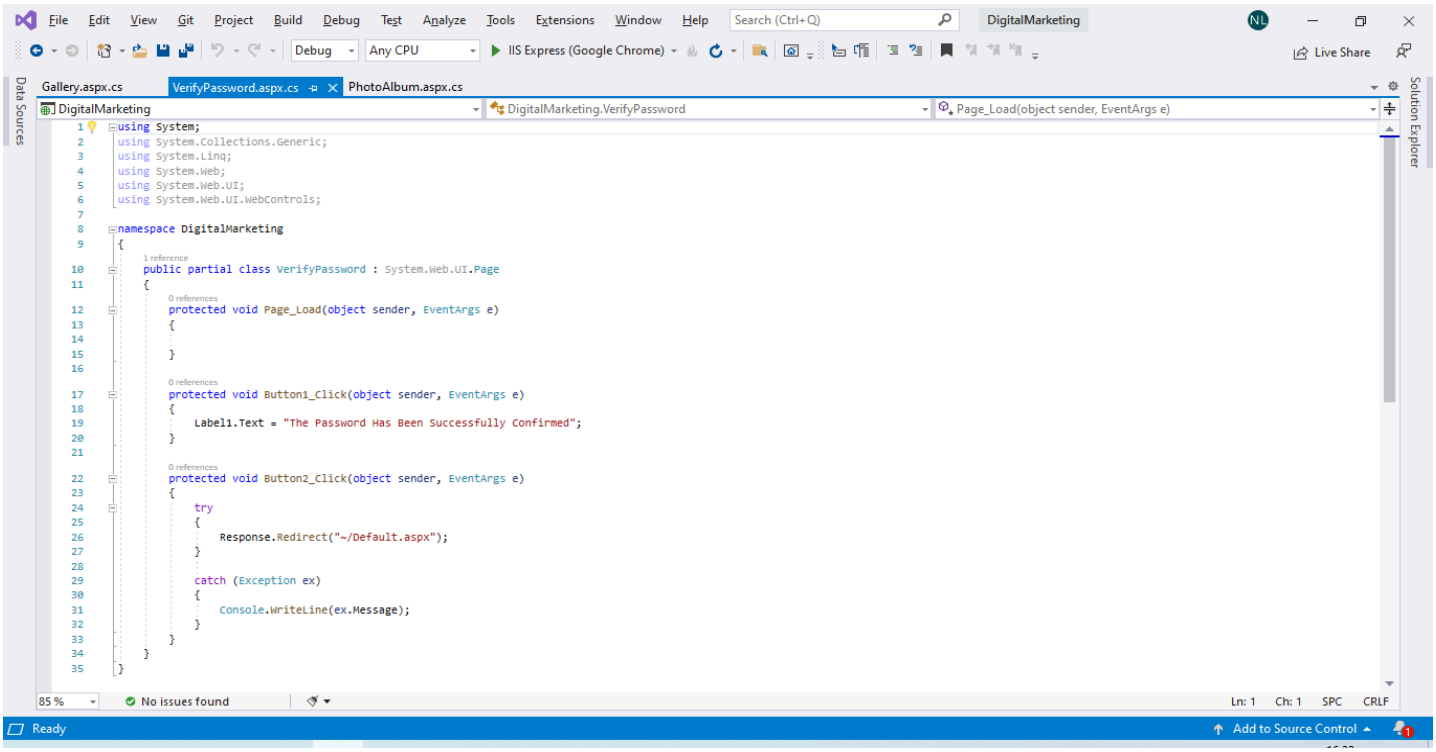


5.2 The Verify Password Page

The VerifyPassword Web Form Page:

- This page is for when the user cannot remember his or her password. On the Default Page, as can be seen in *Section 5.1.*, there is a hyperlink that redirects the user to this page. They then enter their new password and confirm it as well. The verification of the password is printed out in a label, stating that the password was successfully confirmed. The code for this page can be seen below:

Figure 5-2: The VerifyPassword Web Form Page

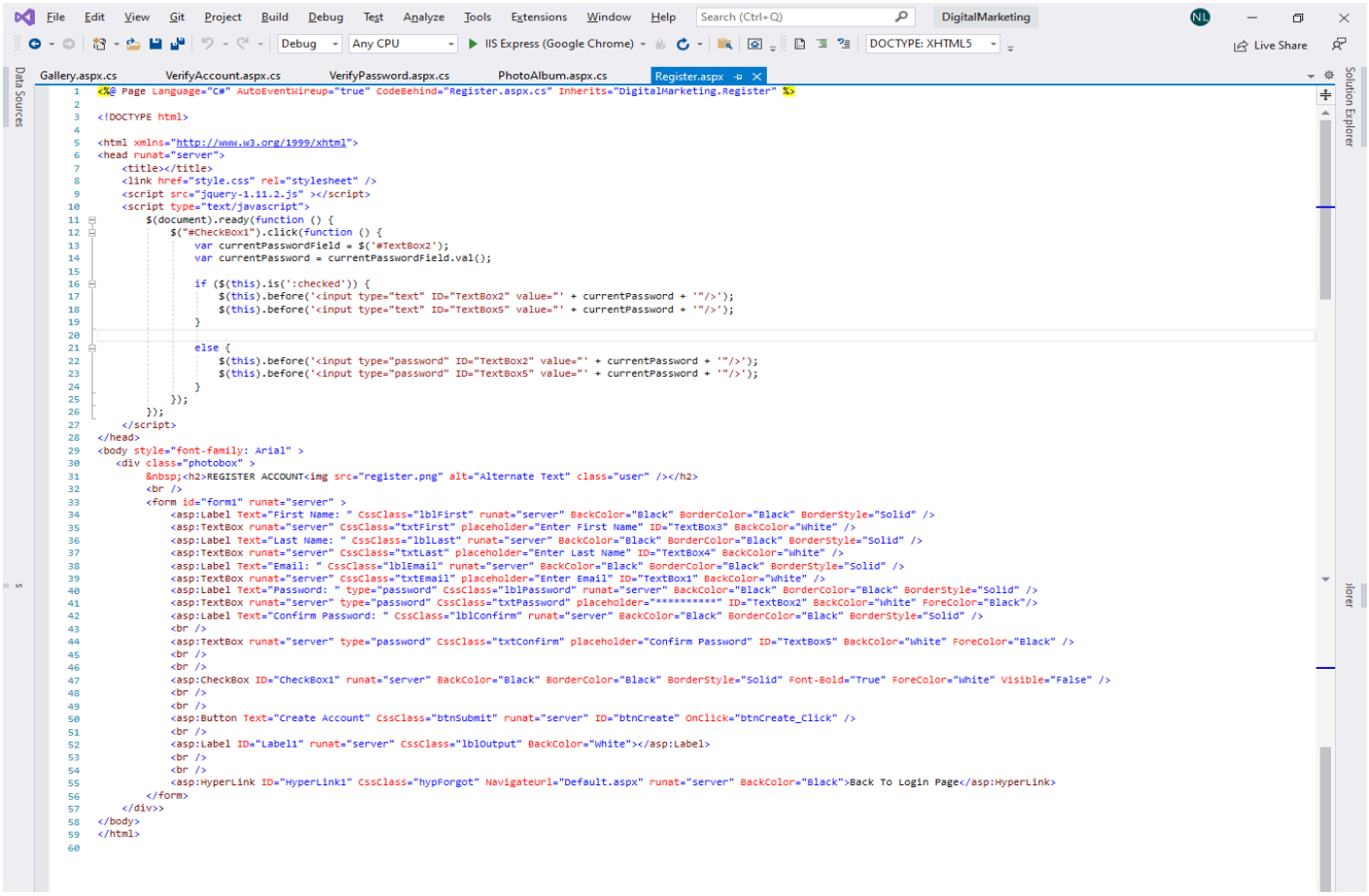


5.3 The Register Page

The Register Web Form Page:

- This page is for when the user does not have an account on the Digital Marketing System. They then enter their name, surname, email and create a password. They are redirected to a page, named the *VerifyAccount.aspx* page, where they must confirm their email. The output is also printed out in a label. Then they are redirected to the Default Page, to login with their new credentials. The source code is shown below:

Figure 5-3: The Register Web Form Page

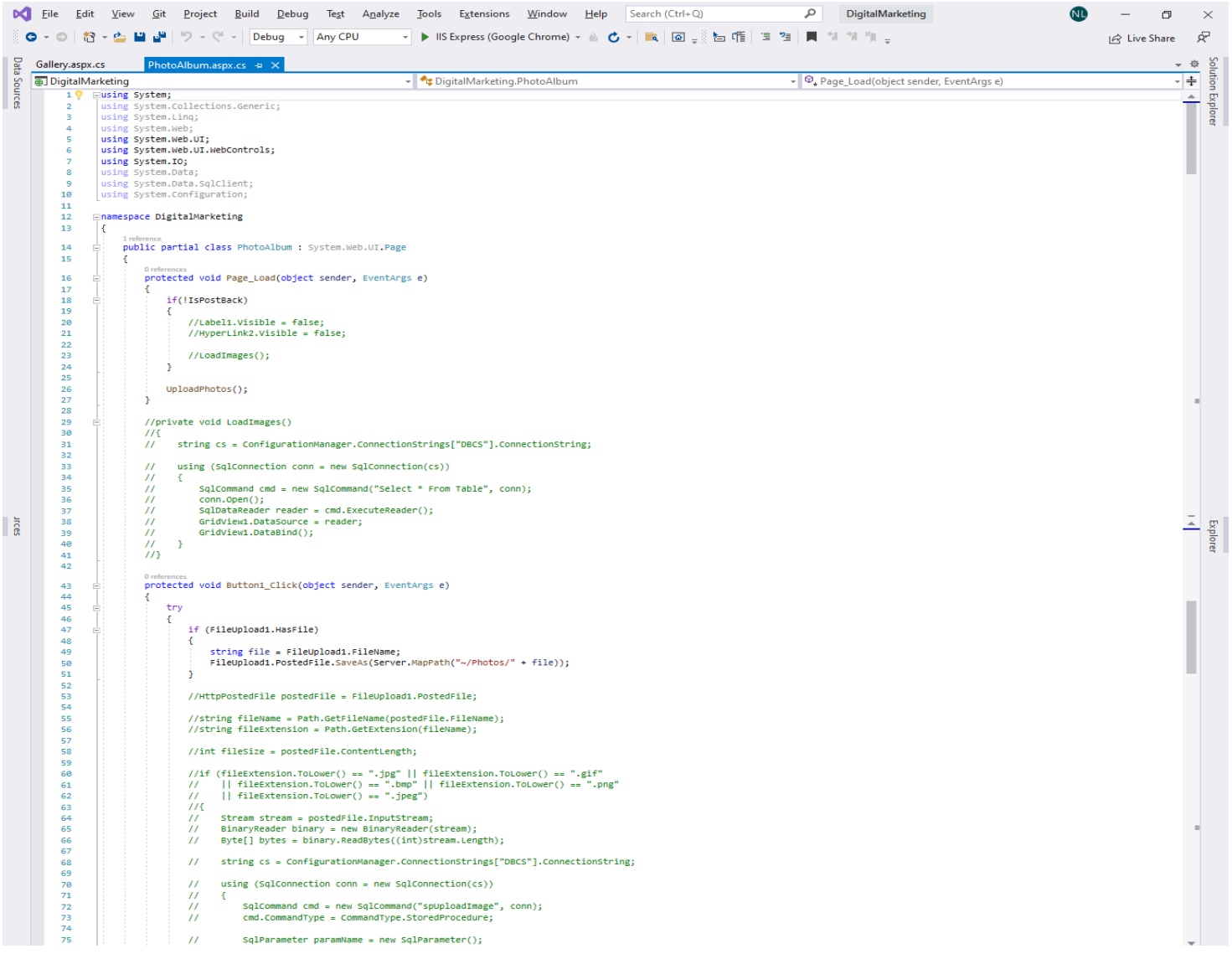


5.4 The Photo Album Page

The PhotoAlbum Class:

- This page uploads and deletes the photos created by the user. The user chooses a photo from a folder on their device and then uploads it. All the photos uploaded by the user will be grouped in a panel. The user can also delete a photo in this panel. The code for this page is given below:

Figure 5-4: The PhotoAlbum Web Form Page




```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace DigitalMarketing
{
    1 reference
    public partial class PhotoAlbum : System.Web.UI.Page
    {
        0 references
        protected void Page_Load(object sender, EventArgs e)
        {
            if(!IsPostBack)
            {
                //Label1.Visible = false;
                //HyperLink2.Visible = false;

                //LoadImages();
            }

            UploadPhotos();
        }

        //private void LoadImages()
        //{
        //    string cs = ConfigurationManager.ConnectionStrings["DBCS"].ConnectionString;

        //    using (SqlConnection conn = new SqlConnection(cs))
        //    {
        //        SqlCommand cmd = new SqlCommand("Select * From Table", conn);
        //        conn.Open();
        //        SqlDataReader reader = cmd.ExecuteReader();
        //        GridView1.DataSource = reader;
        //        GridView1.DataBind();
        //    }
        //}

        0 references
        protected void Button1_Click(object sender, EventArgs e)
        {
            try
            {
                if (FileUpload1.HasFile)
                {
                    string file = FileUpload1.FileName;
                    FileUpload1.PostedFile.SaveAs(Server.MapPath("~/Photos/" + file));
                }

                //HttpPostedFile postedFile = FileUpload1.PostedFile;

                //string fileName = Path.GetFileName(postedFile.FileName);
                //string fileExtension = Path.GetExtension(fileName);

                //int fileSize = postedFile.ContentLength;

                //if (fileExtension.ToLower() == ".jpg" || fileExtension.ToLower() == ".gif"
                //    || fileExtension.ToLower() == ".bmp" || fileExtension.ToLower() == ".png"
                //    || fileExtension.ToLower() == ".jpeg")
                //{
                //    Stream stream = postedFile.InputStream;
                //    BinaryReader binary = new BinaryReader(stream);
                //    Byte[] bytes = binary.ReadBytes((int)stream.Length);

                //    string cs = ConfigurationManager.ConnectionStrings["DBCS"].ConnectionString;

                //    using (SqlConnection conn = new SqlConnection(cs))
                //    {
                //        SqlCommand cmd = new SqlCommand("spUploadImage", conn);
                //        cmd.CommandType = CommandType.StoredProcedure;

                //        SqlParameter paramName = new SqlParameter();
```

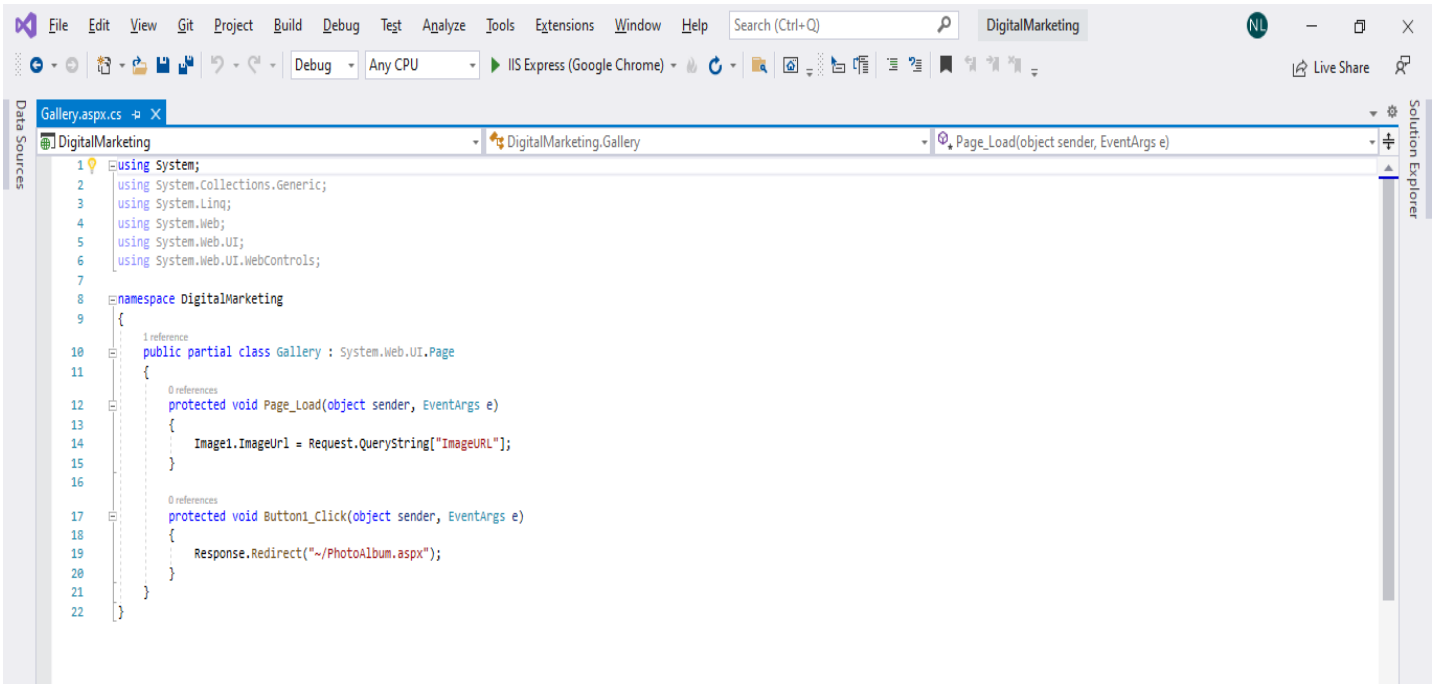
```
                //        {
                //            ParamaterName = @"Name";
                //            Value = fileName;
                //        };
                //        cmd.Parameters.Add(paramName);

                //        SqlParameter paramSize = new SqlParameter();
                //        {
                //            ParamaterName = @"Size";
                //            Value = fileSize;
                //        };
                //        cmd.Parameters.Add(paramSize);

                //        SqlParameter paramImageData = new SqlParameter();
                //        {
                //            ParamaterName = @"ImageData";
                //            Value = bytes;
                //        };
                //        cmd.Parameters.Add(paramImageData);

                //        SqlParameter paramImageData = new SqlParameter();
                //        {
                //            ParamaterName = @"NewId";
                //            Value = -1;
                //            Direction = ParameterDirection.Output;
                //        };
                //        cmd.Parameters.Add(paramNewId);

                //        conn.Open();
                //        cmd.ExecuteNonQuery();
                //        conn.Close();

                //        Label1.Visible = true;
                //        Label1.Text = "Upload Successful!";
                //        HyperLink2.Visible = true;
                //        HyperLink2.NavigateUrl = "~/Gallery.aspx?Id=" + cmd.Parameters["@NewId"].Value.ToString();

                //        LoadImages();
                //    }
                //}

                //else
                //{
                //    Label1.Visible = true;
                //    Label1.Text = "Only Images With The Correct Extension (.bmp, .gif, .jpg, .jpeg, .png) Can Be Uploaded!";
                //    HyperLink2.Visible = false;
                //}

            }

            catch (FileNotFoundException ex)
            {
                Console.WriteLine("Please make sure that {0} exists.", ex.FileName);
            }

            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
                Console.WriteLine();
                Console.WriteLine();
                Console.WriteLine(ex.StackTrace);
            }
        }

        1 reference
        private void UploadPhotos()
        {
            foreach (string stringFile in Directory.GetFiles(Server.MapPath("~/Photos/")))
            {
                ImageButton image = new ImageButton();
                FileInfo info = new FileInfo(stringFile);
                image.ImageUrl = "~/Photos/" + info.Name;
                image.Width = Unit.Pixel(300);
                image.Height = Unit.Pixel(300);
                image.Style.Add("padding", "5px");
                image.Click += new ImageClickEventHandler(image_Click);
                Panel1.Controls.Add(image);
            }
            Panel1.Visible = true;
        }

        1 reference
        void image_Click(object sender, EventArgs e)
        {
            try
            {
                Response.Redirect("~/Gallery.aspx?ImageURL=" + ((ImageButton)sender).ImageUrl);
            }

            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }
        }

        0 references
        protected void Button2_Click(object sender, EventArgs e)
        {
            Panel1.Visible = false;
        }
    }
}
```

5.5 The Gallery Page

The Gallery Web Form Page:

- This page is for the user to download and view the photos he or she uploaded onto the Digital Marketing System. The user clicks on one of the photos uploaded in the panel onto the *PhotoAlbum.aspx* page, then they are then redirected to this page. Here, they will see that the selected photo is enlarged for them to view. They can either head back to the *PhotoAlbum.aspx* page or download it by right clicking and saving the image to a folder. The code is given below:

Figure 5-5: The Gallery Web Form Page



CHAPTER 6 CODE COVERAGE

6.1 Code Coverage Report

The code coverage report does not want to show. In the command prompt of where the project is stored, the following must be typed to show a detailed description of the code coverage report:

- dotnet test –collect:”XPlat Code Coverage”

Figure 6-1 The Code Coverage Report In Command Prompt