Bahria University,

Karachi Campus

COURSE: CSL-210 OBJECT ORIENTED PROGRAMMING
TERM: SPRING 2025, CLASS: BSE- 2 (A)

PROJECT NAME

Gym And Fitness
(C# Desktop App)

Submitted By:

AAMIR RAFIQUE

02-131242-019

Submitted to:

Engr.Muhammad Faisal/ Engr. Saniya Sarim

Signed **Remarks:** **Score:**

Table of Content

Contents

1. INTRODUCTION

1.1. Problem Statement:

Many people struggle to stay consistent with fitness due to a lack of simple, guided tools. Existing apps are often too complex, while traditional methods like notebooks aren't practical. This project introduces a user-friendly C# Gym And Fitness app designed for everyday users, offering basic features like workout plans, BMI calculation, and daily routines to help build healthy habits without the complexity of full management systems.

1.2. Provided Solution:

The Gym And Fitness Application is a Full-stack WinForms C# fitness application designed to assist users in achieving their health and workout goals. It provides personalized workout plans, nutritional guidance, and progress tracking. Whether you're aiming to build muscle, lose weight, or enhance overall fitness, this application offers a structured approach to support your journey. Our mission is to make fitness accessible, convenient, and enjoyable for everyone. Stay fit, stay healthy!

1.3. Tools & Technologies:

- Programming Language: C#
- IDE: Visual Studio
- Database: SQL Server
- Framework: .NET Framework (4.7.2)
- Deployment: Deployed using ClickOnce

1.4. System Requirements:

- **Platform:** Windows 10/11
- **Framework:** .NET Framework 4.7.2+
- **Database:** SQL Server LocalDB 2022
- **Users:** Any user (since it's a personal fitness application)

2. OOP CONCEPTS USED IN PROJECT

2.1. Class & Objects:

- User, UserDataManager, Features and forms etc.

2.2. Inheritance:

- Some forms like AboutForm, MainForm, DashboardForm, DietPlansForm, WorkoutPlansForm, ProfileForm, BMICalculatorForm are inheriting common properties like SlidePanel and RibbonControl etc. from BaseForm.cs.

2.3. Encapsulation:

- Classes like User, WorkoutPlan, UserDataManager, encapsulate data (properties like Name, Age, etc.) and behavior (methods).
- Access modifiers like *public, private* are used to *hide implementation details* and expose only what's needed.

2.4. Abstraction:

- Abstracted away logic like user data handling into classes like UserDataManager, hiding database interaction details from the UI.
- Implementation details (e.g., saving/loading data, API interactions, calculating BMI) are abstracted into classes like:
- UserDataManager.cs and Features.cs etc.

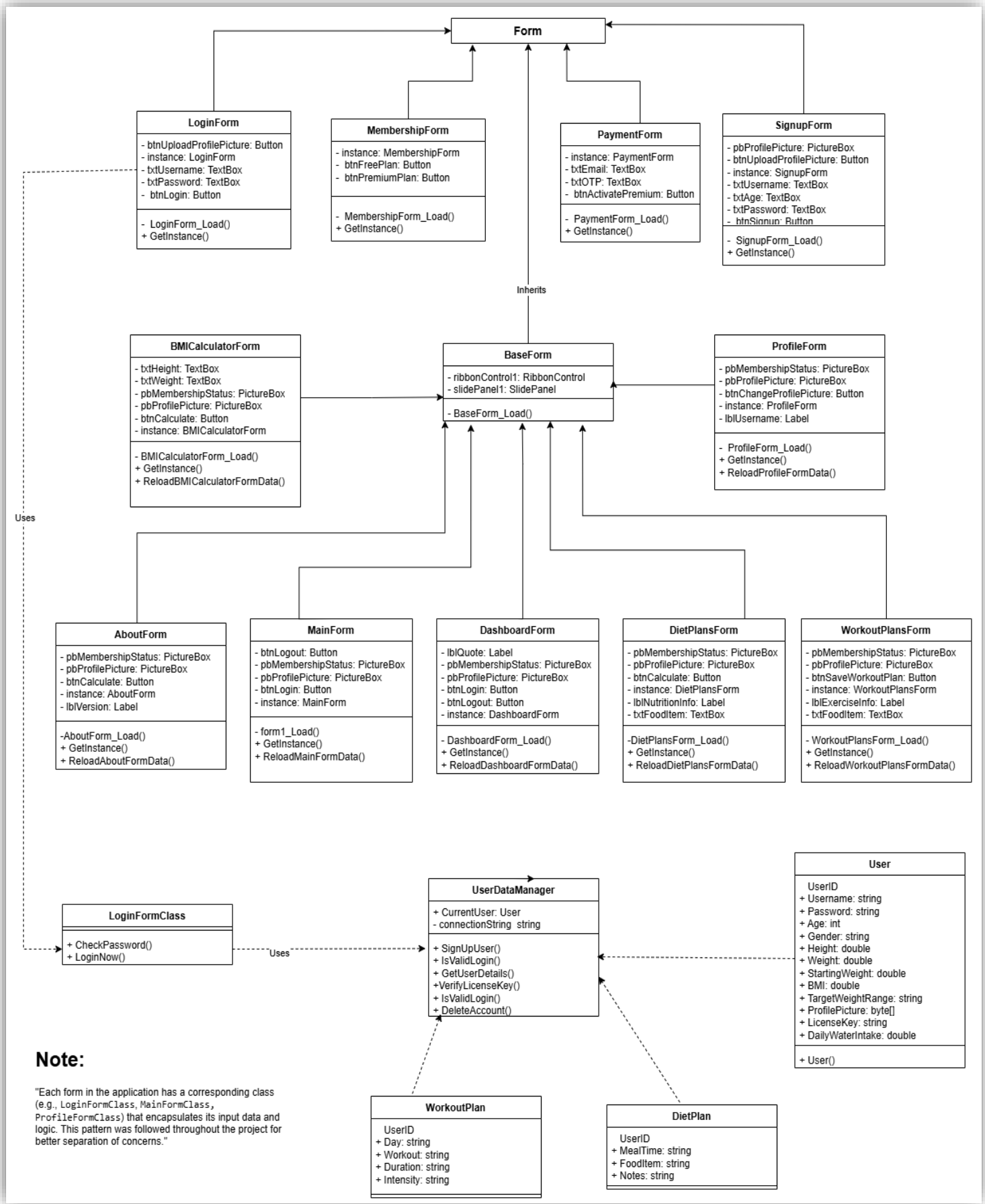
2.5. Singleton Design Pattern:

- To ensure the Forms have only one instance throughout the application.
- Benefits include: Less memory Consumption and better performance.

2.6. Static Classes / Methods:

- Made Static UserDataManager.cs class to share a single copy of user info (for e.g. *public static User CurrentUser)* throughout the whole application, and for access without instantiation.

3. UML CLASS DIAGRAM



Note:

"Each form in the application has a corresponding class (e.g., `LoginFormClass`, `MainFormClass`, `ProfileFormClass`) that encapsulates its input data and logic. This pattern was followed throughout the project for better separation of concerns."

4. FEATURES OF YOUR PROJECT

- User-Friendly Interface
- BMI Calculator Module
- Personal Fitness Profile
- Workout Module
- Nutrition Tracking
- Diet Plans (Customizable)
- Water Intake Tracking
- Fitness Progress Monitoring
- Free / Premium versions available
- Email verification for premium upgrade

5. CODE (ONLY MAIN CLASSES AND FORMS)

5.1. MainForm.cs

```
using GymAndFitness.Classes;
using GymAndFitness.Forms;
using System;
using System.Windows.Forms;

namespace GymAndFitness
{
    public partial class MainForm : BaseForm
    {
        private static MainForm instance;
        private MainForm()
        {
            InitializeComponent();
        }

        public static MainForm GetInstance()
        {
            if (instance == null || instance.IsDisposed)
            {
                instance = new MainForm();
            }
            return instance;
        }

        //method to refresh premium features. in the forms.. after it is changed in profile form...
        public void RefreshPremiumFeaturesMainForm()
        {
            pbMembershipStatus.Image = Features.MembershipStatusPic();
        }

        //helper method for refreshing profile picture...
        public void RefreshProfilePictureInForms()
        {
            UserDataManager.ApplyProfilePicture(pbProfilePicture);
        }

        //form1 i.e. main
        private void Form1_Load(object sender, EventArgs e)
        {
            MainFormClass.MainFormLoadEvents(pbProfilePicture, btnLogout, btnLogin, pbMembershipStatus);
        }
```

```
        public void ReloadMainFormData()
        {
            MainFormClass.MainFormLoadEvents(pbProfilePicture, btnLogout, btnLogin, pbMembershipStatus);
        }

        private void btnLogout_Click(object sender, EventArgs e)
        {
            Features.BtnLogout();
        }

        private void btnLogin_Click_1(object sender, EventArgs e)
        {
            Features.OpenLoginForm();
            this.Hide();
        }

        private void MainForm_FormClosed(object sender, FormClosedEventArgs e)
        {
            Features.FormClosedEvent();
        }

        private void pbProfilePicture_Click(object sender, EventArgs e)
        {
            Features.OpenProfileForm();
            this.Hide();
        }

        private void pbProfilePicture_MouseEnter(object sender, EventArgs e)
        {
            Features.TooltipProfilePic(toolTip1, pbProfilePicture);
        }

    }
}
```

5.2. MainFormClass.cs

```
using System.Windows.Forms;

namespace GymAndFitness.Classes
{
    internal class MainFormClass
    {
        public static void MainFormLoadEvents(PictureBox pbProfilePicture, Button btnLogout, Button
btnLogin, PictureBox pbMembershipStatus)
        {
            if (UserDataManager.CurrentUser != null)
            {
                UserDataManager.ApplyProfilePicture(pbProfilePicture);
                btnLogout.Visible = true;
                btnLogout.Enabled = true;
                btnLogin.Visible = false;
                btnLogin.Enabled = false;

                //load membership plan pics
                pbMembershipStatus.Image = Features.MembershipStatusPic();
            }
            else
            {
                btnLogout.Visible = false;
                btnLogout.Enabled = false;
                btnLogin.Visible = true;
                btnLogin.Enabled = true;

                MessageBox.Show("No user is logged in.", "Attention", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
            }
        }

    }
}
```

5.3. LoginForm.cs

```
using GymAndFitness.Classes;
using System;
using System.Windows.Forms;

namespace GymAndFitness
{
    public partial class LoginForm : Form
    {
        private static LoginForm instance;
        public LoginForm()
        {
            InitializeComponent();
        }
        public static LoginForm GetInstance()
        {
            if (instance == null || instance.IsDisposed)
            {
                instance = new LoginForm();
            }
            return instance;
        }

        private void chkShowPassword_CheckedChanged(object sender, EventArgs e)
        {
            LoginFormClass.CheckPassword(chkShowPassword, txtPassword);
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
            LoginFormClass.LoginNow(txtUsername, txtPassword, error);

            if (UserDataManager.CurrentUser != null)
            {
                // Navigate to MainForm
                Features.OpenMainForm();
                this.Hide();
            }
        }

        private void lblCreateAccount_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
        {
            Features.OpenSignUpForm();
            this.Hide();
        }

        private void txtUsername_KeyPress(object sender, KeyPressEventArgs e)
        {
            error.Clear();
        }

        private void txtPassword_KeyPress(object sender, KeyPressEventArgs e)
        {
            error.Clear();
        }

        private void lblGuest_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
        {
            Features.OpenMainForm();
            this.Hide();
        }

        private void LoginForm_VisibleChanged(object sender, EventArgs e)
        {
            txtUsername.Clear();
            txtPassword.Clear();
            txtUsername.Focus();
        }

        private void txtPassword_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter) // Check if Enter key was pressed
            {
```

```
                e.SuppressKeyPress = true; // Prevent the default behavior (e.g., beep sound)
                btnLogin.PerformClick(); // Trigger the button's click event
            }
        }

        private void LoginForm_FormClosed(object sender, FormClosedEventArgs e)
        {
            Features.FormClosedEvent();
        }

    }
}
```

5.4. LoginFormClass.cs

```
using System;
using System.Windows.Forms;

namespace GymAndFitness.Classes
{
    internal class LoginFormClass
    {
        public static void CheckPassword(CheckBox chkShowPassword, TextBox txtPassword)
        {
            bool check = chkShowPassword.Checked;

            switch (check)
            {
                case true:
                    txtPassword.UseSystemPasswordChar = false;
                    break;
                case false:
                    txtPassword.UseSystemPasswordChar = true;
                    break;
            }
        }
        //validate login info
        public static void LoginNow(TextBox txtUsername, TextBox txtPassword, ErrorProvider error)
        {
            try
            {
                if (string.IsNullOrEmpty(txtUsername.Text))
                {
                    txtUsername.Focus(); //isi pr focus!
                    error.SetError(txtUsername, "Please Enter your Username ");
                }
                else if (string.IsNullOrEmpty(txtPassword.Text))
                {
                    txtPassword.Focus(); //isi pr focus!
                    error.SetError(txtPassword, "Please Enter your Password");
                }
                else
                {
                    string username = txtUsername.Text;
                    string password = txtPassword.Text;

                    if (UserDataManager.IsValidLogin(username, password))
                    {
                        MessageBox.Show("Login Successful!", "Success", MessageBoxButtons.OK,
MessageBoxIcon.Information);

                        User user = UserDataManager.GetUserDetails(txtUsername.Text);
                        if (user != null && user.Password == txtPassword.Text)
                        {
                            // calling ReloadFormData(); in btnLogin in order to reload form data, at
the time when a user logs in..
                            AboutForm.GetInstance().ReloadAboutFormData();
                            BMICalculatorForm.GetInstance().ReloadBMICalculatorFormData();
                            DashboardForm.GetInstance().ReloadDashboardFormData();
                            DietPlansForm.GetInstance().ReloadDietPlansFormData();
```

```
                            MainForm.GetInstance().ReloadMainFormData();
                            ProfileForm.GetInstance().ReloadProfileFormData();
                            WorkoutPlansForm.GetInstance().ReloadWorkoutPlansData();

                            UserDataManager.CurrentUser = user;
                            MessageBox.Show($"Welcome {UserDataManager.CurrentUser.Username}!", " ",
MessageBoxButtons.OK, MessageBoxIcon.Information);

                            //opening main form and close login form in forms...btnlogin method event..
                        }
                    }
                    else
                    {
                        MessageBox.Show("Invalid username or password.", "Login Failed",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
            }
            catch (Exception e)
            {
                MessageBox.Show(e.ToString());
            }
        }

    }
}
```

5.5. User.cs

```
using System.Collections.Generic;

namespace GymAndFitness
{
    public class User
    {
        public int UserID { get; set; } //primary key
        public List<WorkoutPlan> WorkoutPlans { get; set; }
        public List<DietPlan> DietPlans { get; set; }
        public string Username { get; set; }
        public string Password { get; set; }
        public int Age { get; set; }
        public string Gender { get; set; }
        public double Height { get; set; }
        public double StartingWeight { get; set; }
        public double CurrentWeight { get; set; }
        public double BMI { get; set; }
        public string TargetWeightRange { get; set; }
        public double TargetWeight { get; set; }
        public string FitnessGoal { get; set; }
        public string FitnessLevel { get; set; }
        public byte[] ProfilePicture { get; set; }
        public double DailyWaterIntake { get; set; }
        public string MembershipStatus { get; set; }
        public string LicenseKey { get; set; }

        public User()
        {
            DietPlans = new List<DietPlan>();
            WorkoutPlans = new List<WorkoutPlan>();
        }

    }
}
```

6. INTERFACES (OUTPUT SCREESHOTS)

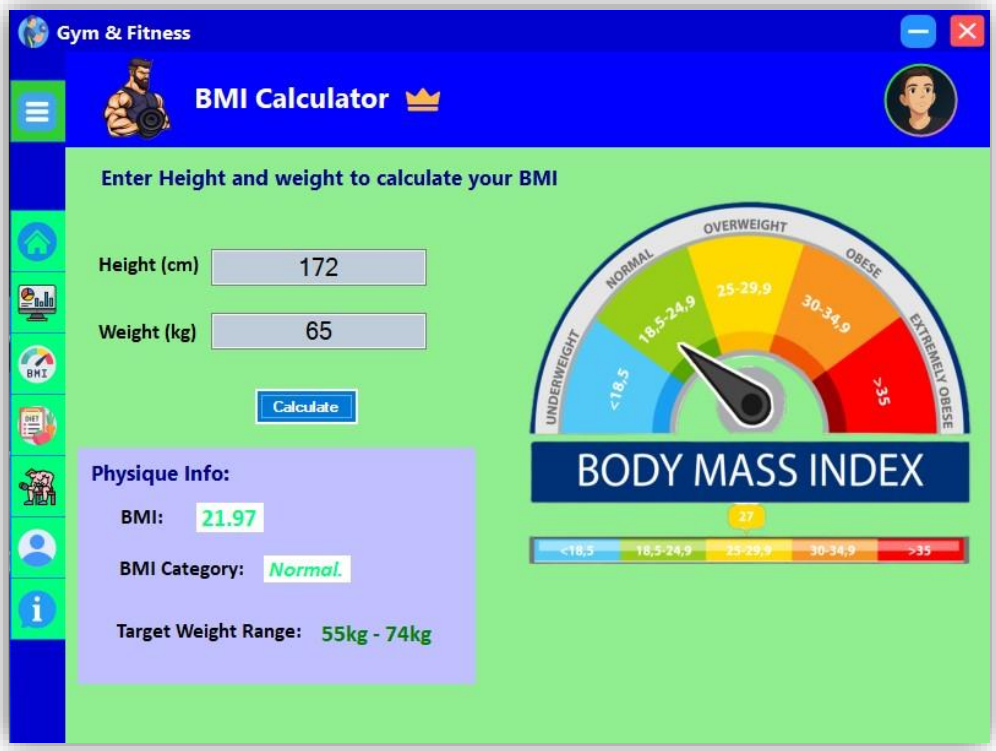
6.1. LoadingForm

6.2. LoginForm

6.3. MainForm

6.4. DashboardForm

6.5. BMICalculatorForm



6.6. ProfileForm

6.7. AboutForm

7. CONCLUSION

- The project **"Gym And Fitness App"** has been successfully developed and implemented using **C#, .NET Framework, SQL Server**. The main goal of this project was to develop a user-friendly C# Gym And Fitness desktop app, which has been effectively achieved.

- During the development of this project, we have got a deeper understanding of key **Object-Oriented Programming (OOP)** concepts such as *inheritance, encapsulation, abstraction, and polymorphism,* and learned how to apply them in a real-world software application. Additionally, we became more proficient in using ***Windows Forms*** *for UI design* and ***SQL*** *for database management.*

- Throughout the development process, we have encountered challenges such as creating a user-friendly design, database connection issues, errors, exceptions while testing and somehow difficulty in deploying the app using ***Visual Studio ClickOnce Deployment*** which we overcame by continuous efforts. These experiences enhanced our problem-solving skills, team management and increased the confidence in software development.
- In the future, this project can be improved by adding:
    - Integrate AI features
    - Regular updates for improved performance
    - Continuous user Support

- Overall, this project provided us with valuable hands-on experience in software development and has strengthened our foundation for future projects and learning.