Project Title:

UNICOMTIC Management System

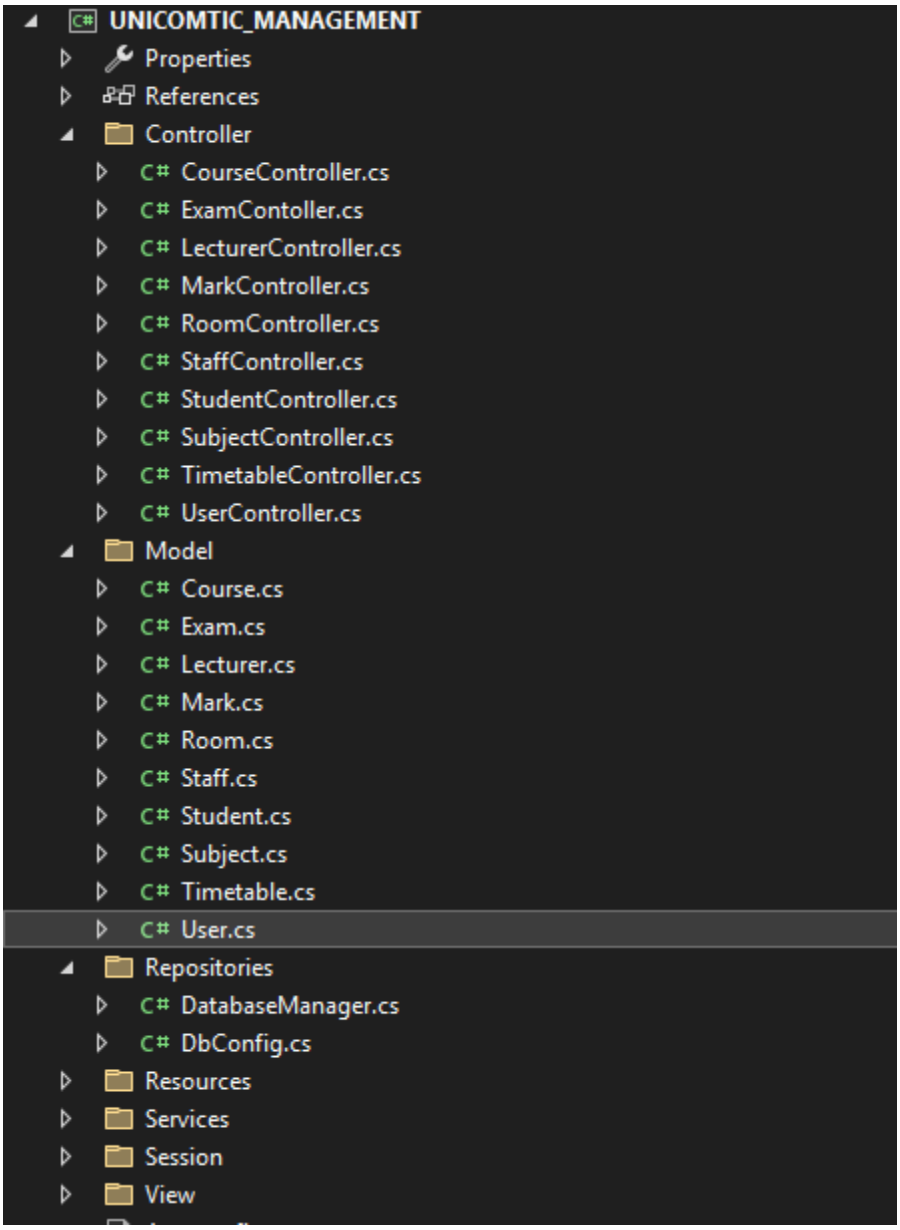
Submitted By:

Thanujan

[6/24/2025]

1. Introduction

The **UNICOMTIC Management System** is a desktop application designed to manage different roles in an academic institution, such as Students, Lecturers, and Staff. This report focuses on the **User Login Module**, which authenticates users based on their credentials and roles and then redirects them to their respective dashboards.



2. Functional Requirements

- Authenticate users based on role (Student, Lecturer, Staff)
- Redirect users to their respective interfaces on successful login
- Validate input fields
- Manage sessions for logged-in users

3. Module Description

4.1 User Login Form (`Userlogin.cs`)

- Collects input: username, password, and role
- Validates for empty fields
- Sends login request to the appropriate controller method based on selected role
- Shows success or error messages
- Navigates to the appropriate form

4.2 Controller (`UserController.cs`)

Handles the authentication logic for each role:

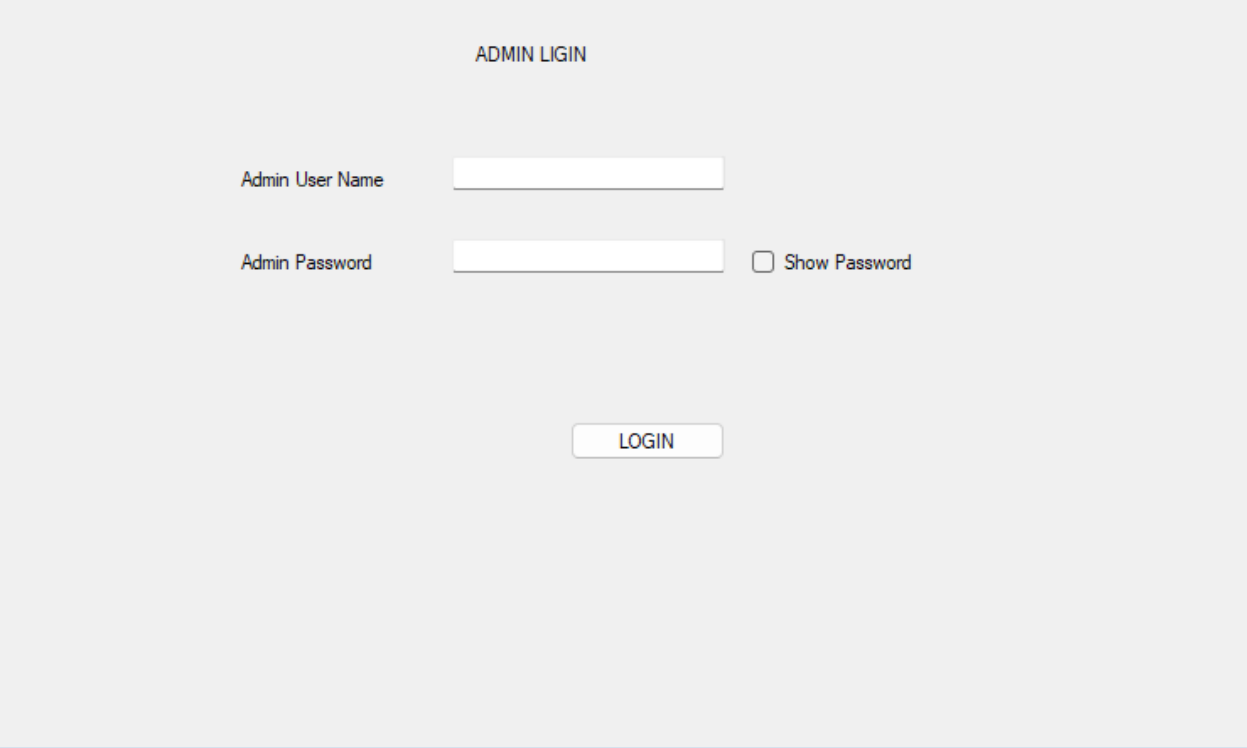
- `LoginStudent()` – Verifies credentials from `Students` table
- `LoginLecturer()` – Verifies credentials from `Lecturers` table
- `LoginStaff()` – Verifies credentials from `Staff` table
- `GetUsers()` – (Optionally) checks credentials from the `Users` table for generic roles

4.3 Session Management (`UserSession.cs`)

- Static class that stores login session information:
  - `UserID`, `UserName`, `Password`, `Role`
- Allows access to session values throughout the application

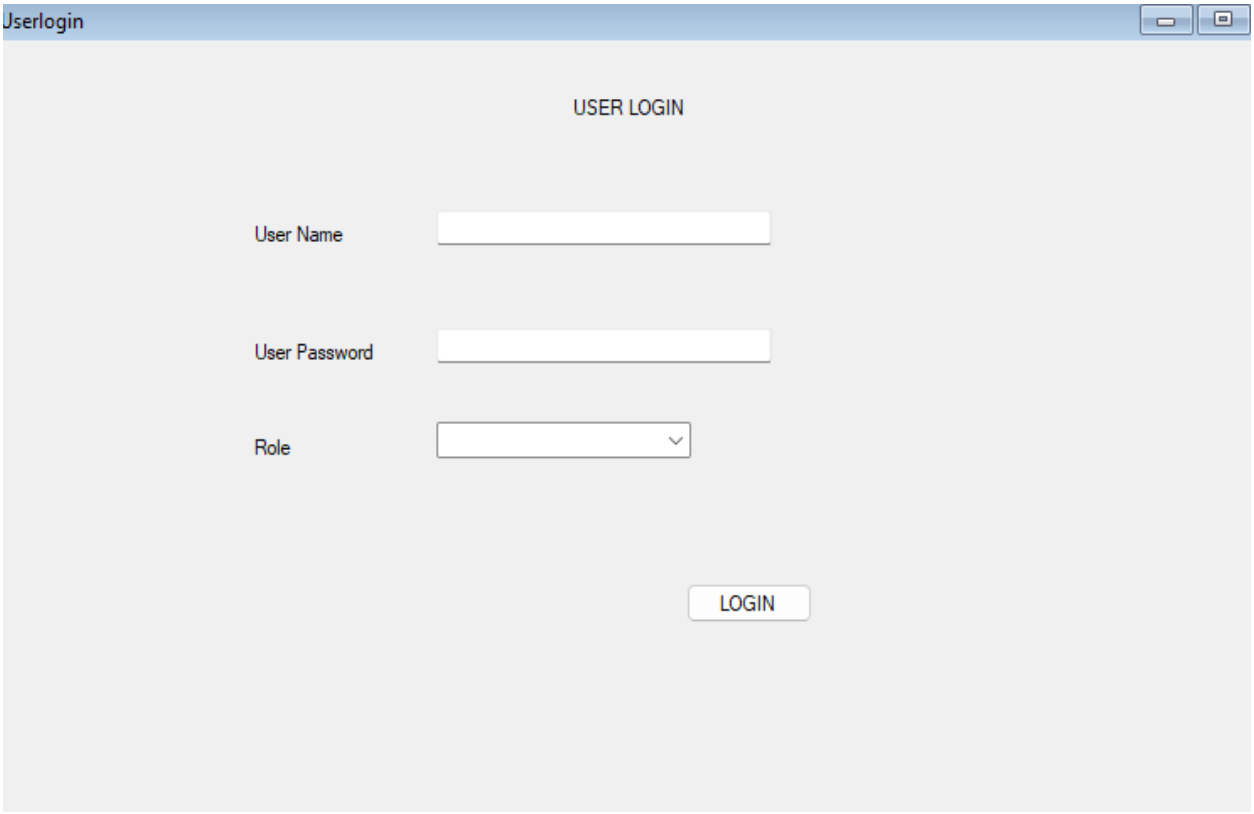
5. Sample Screenshots

Admin Login UI



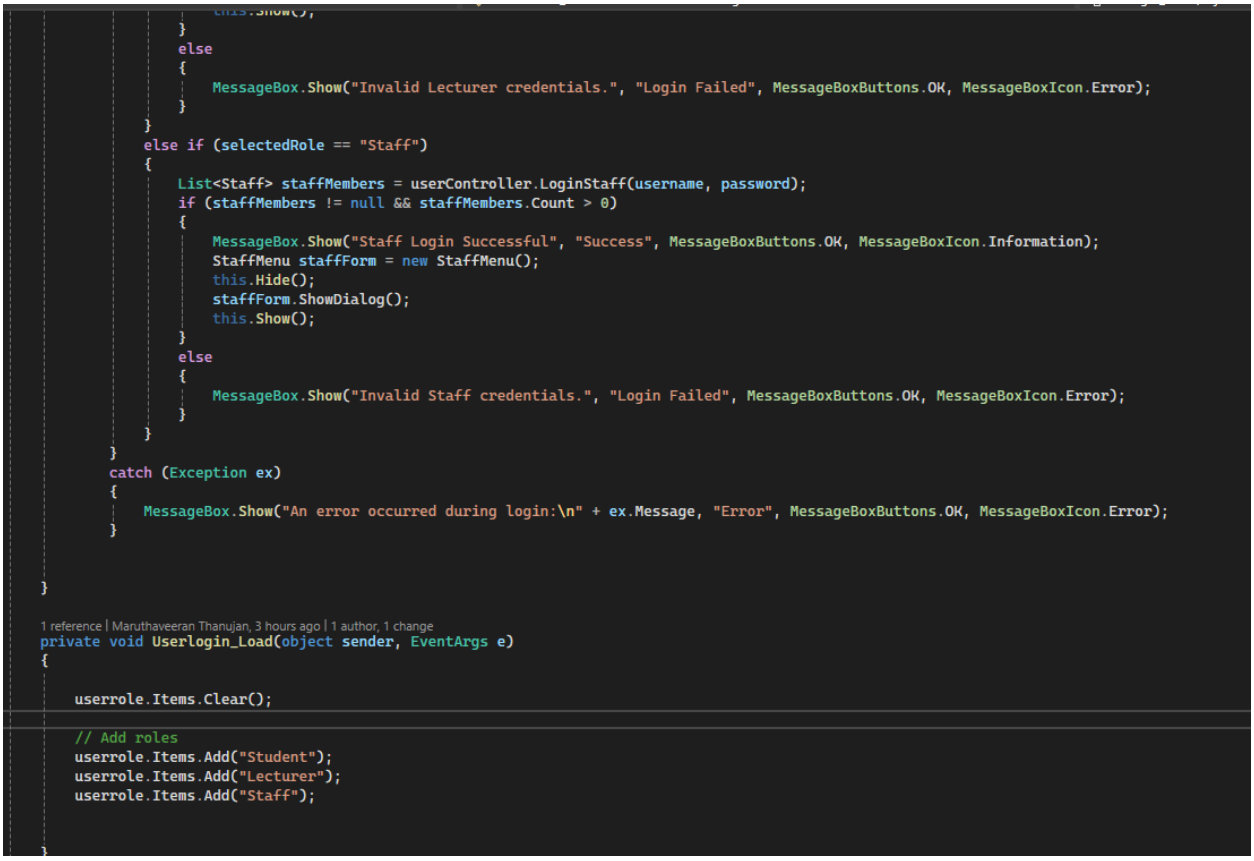

```
public AdminLogin()
{
    InitializeComponent();
}

1 reference | Maruthaveeran Thanujan, 2 hours ago | 1 author, 3 changes
private void btnadminlogin_Click(object sender, EventArgs e)
{
    try
    {
        if (string.IsNullOrWhiteSpace(adminusername.Text))
        {
            MessageBox.Show("Please enter a username.", "Input Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            adminusername.Focus();
            return;
        }
        if (adminusername.Text == "Admin" && adminpassword.Text == "Admin@123")
        {

            MessageBox.Show("SuccessFully Logged in");

            this.Hide();
            AdminMenu adminForm = new AdminMenu();
            adminForm.ShowDialog();
        }

        else
        {
            MessageBox.Show("Invalid UserName or Password");
        }

    }
    catch (Exception ex)
    {
        // Log or handle unexpected errors
        MessageBox.Show("An unexpected error occurred: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
```

Admin Login Back End



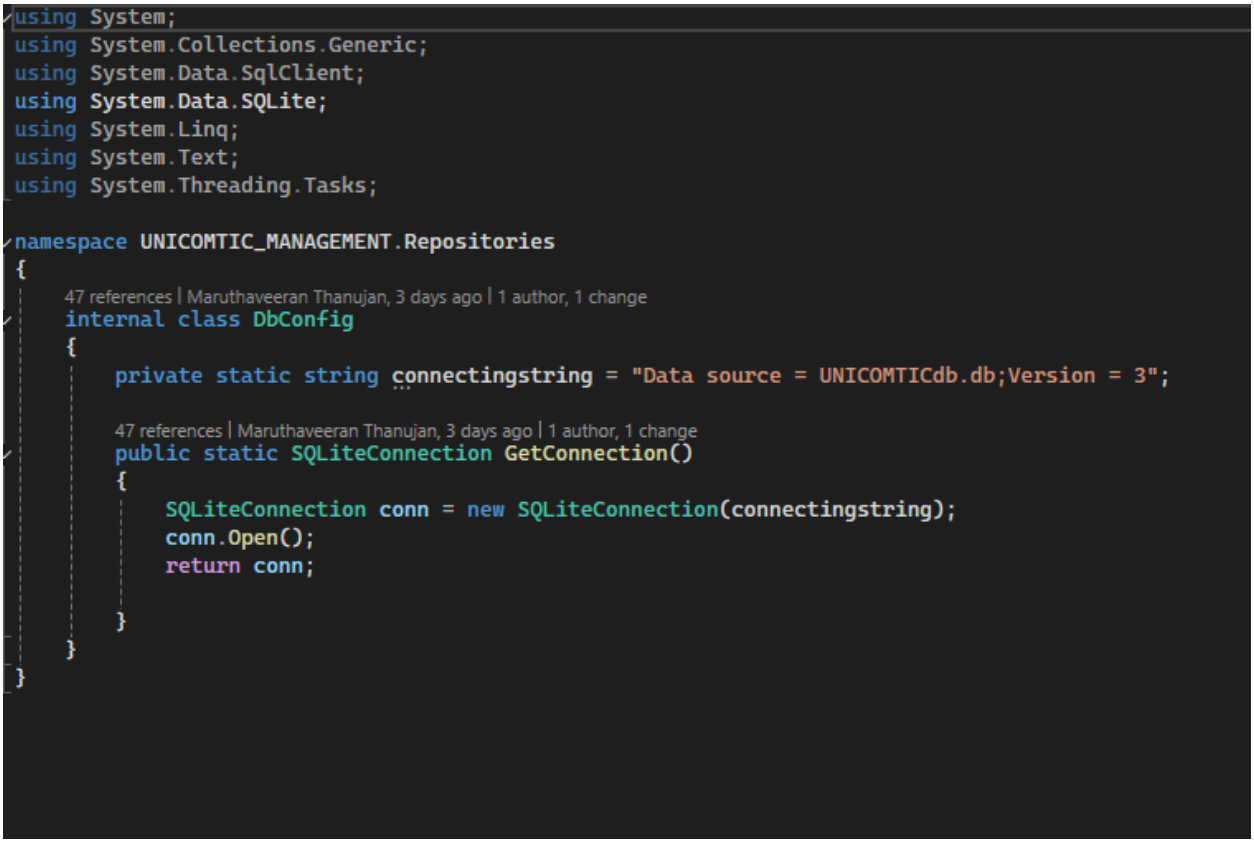
USER LOGIN UI

```
try
{
    string username = txtusername.Text.Trim();
    string password = txtpassword.Text;
string selectedRole = userrole.SelectedItem.ToString();


UserSession.UserName = username;
UserSession.Role = selectedRole;
UserSession.Password = password;

    if (string.IsNullOrEmpty(username) || string.IsNullOrEmpty(password))
    {
        MessageBox.Show("Please enter both Username and Password.", "Validation", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }



    if (selectedRole == "Student")
    {

        List<Student> students = userController.LoginStudent(username, password);
        if (students != null && students.Count > 0)
        {
            MessageBox.Show("Student Login Successful", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
            StudentViewForm studentForm = new StudentViewForm();
            this.Hide();
            studentForm.ShowDialog();
            this.Show();
        }
        else
        {
            MessageBox.Show("Invalid Student credentials.", "Login Failed", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else if (selectedRole == "Lecturer")
    {
        List<Lecturer> lecturers = userController.LoginLecturer(username, password);
        if (lecturers != null && lecturers.Count > 0)
        {
            MessageBox.Show("Lecturer Login Successful", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
        LactuerViewForm lecturerForm = new LactuerViewForm();
            this.Hide();
            lecturerForm.ShowDialog();
            this.Show();
```



```
                this.Show();
            }
            else
            {
                MessageBox.Show("Invalid Lecturer credentials.", "Login Failed", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        else if (selectedRole == "Staff")
        {
            List<Staff> staffMembers = userController.LoginStaff(username, password);
            if (staffMembers != null && staffMembers.Count > 0)
            {
                MessageBox.Show("Staff Login Successful", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
                StaffMenu staffForm = new StaffMenu();
                this.Hide();
                staffForm.ShowDialog();
                this.Show();
            }
            else
            {
                MessageBox.Show("Invalid Staff credentials.", "Login Failed", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("An error occurred during login:\n" + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }


}

1 reference | Maruthaveeran Thanujan, 3 hours ago | 1 author, 1 change
private void Userlogin_Load(object sender, EventArgs e)
{

    userrole.Items.Clear();

    // Add roles
    userrole.Items.Add("Student");
    userrole.Items.Add("Lecturer");
    userrole.Items.Add("Staff");


}
```

- User Login Back End



```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data.SQLite;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace UNICOMTIC_MANAGEMENT.Repositories
{
    47 references | Maruthaveeran Thanujan, 3 days ago | 1 author, 1 change
    internal class DbConfig
    {
        private static string connectingstring = "Data source = UNICOMTICdb.db;Version = 3";

        47 references | Maruthaveeran Thanujan, 3 days ago | 1 author, 1 change
        public static SQLiteConnection GetConnection()
        {
            SQLiteConnection conn = new SQLiteConnection(connectingstring);
            conn.Open();
            return conn;

        }
    }
}
```

Data Base Connection

```
CREATE TABLE IF NOT EXISTS Users (
    UserID INTEGER PRIMARY KEY AUTOINCREMENT,
    Username TEXT NOT NULL UNIQUE,
    Password TEXT NOT NULL,
    Role TEXT NOT NULL
    );

CREATE TABLE IF NOT EXISTS Courses (
    CourseID INTEGER PRIMARY KEY AUTOINCREMENT,
    CourseName TEXT NOT NULL
);


CREATE TABLE IF NOT EXISTS Subjects (
    SubjectID INTEGER PRIMARY KEY AUTOINCREMENT,
    SubjectName TEXT NOT NULL,
    CourseID INTEGER,
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)
);


CREATE TABLE IF NOT EXISTS Students (
    StudentID INTEGER PRIMARY KEY AUTOINCREMENT,
    Name TEXT NOT NULL,
    Address TEXT,
    Username TEXT NOT NULL,
    Password TEXT NOT NULL,
    CourseID INTEGER,
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)

);


CREATE TABLE IF NOT EXISTS Staff (
    StaffID INTEGER PRIMARY KEY AUTOINCREMENT,
    Name TEXT NOT NULL,
    Address TEXT,
    Username TEXT NOT NULL,
    Password TEXT NOT NULL
);


CREATE TABLE IF NOT EXISTS Lecturers (
    LecturerID INTEGER PRIMARY KEY AUTOINCREMENT,
    Name TEXT NOT NULL,
```

CREATE TABLE

Thanks ....