# Unicom TIC Management system

## Inventory

* Introduction
* Technologies used
* Code Highlights
* Conclusion

**By:-k.Lingaruban**

**UT:-010534**

Introduction

The **Unicom TIC Management System (UMS)** is a desktop application designed to manage essential school operations with a focus on simplicity and usability. The system is aimed at automating and streamlining the day-to-day management of a school's academic structure, including course management, student records, exam handling, and timetable organization.

The primary goal of the system is to provide a seamless experience for different user roles, including **Admins**, **Staff**, **Students**, and **Lecturers**. Each user role is granted different levels of access, ensuring that only authorized users can perform specific tasks. The application utilizes a **Model-View-Controller (MVC)** architecture to separate the user interface from the business logic, making it easy to maintain and extend.

The system's core modules include:

1. **Course & Subject Management**: Allows the administration to manage courses and subjects, while providing all roles the ability to view them.
2. **Student Management**: Enables the addition, editing, and deletion of student records by Admins, with Students being able to view their own data.
3. **Exam & Marks Management**: Facilitates the creation of exams, and adding and viewing marks, with specific actions available to Admins, Staff, and Lecturers.
4. **Timetable Management & Room Allocation**: Admins can assign rooms (e.g., labs, halls) to timetables, while all roles can view the timetable.
5. **Login System**: A secure login mechanism enables users to sign in with their credentials and ensures role-based access control to limit actions based on the user's role.

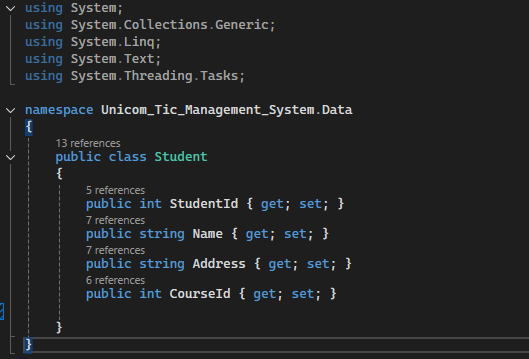
The **Unicom TIC Management System** was developed using **C# WinForms** for the user interface, paired with a **SQLite** database for persistent data storage. The system uses **asynchronous programming** techniques to ensure smooth performance while interacting with the database. This application is aimed at providing a beginner-friendly yet robust solution for managing a small school's academic and administrative needs.

This project also serves as a learning exercise for understanding **basic C# programming**, creating **database-driven applications**, and working with design patterns like **MVC** to keep the application structure clean and maintainable.

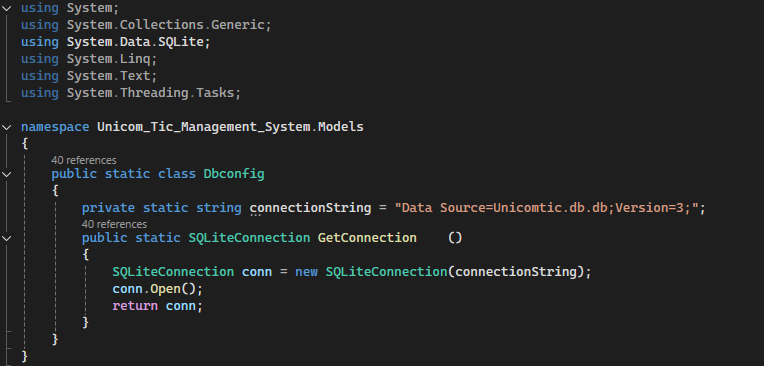
Technoligies used

The **Unicom TIC Management System** was built using several modern technologies and tools, each chosen for their ability to handle specific requirements of the project. The following technologies were used in the development of the application:

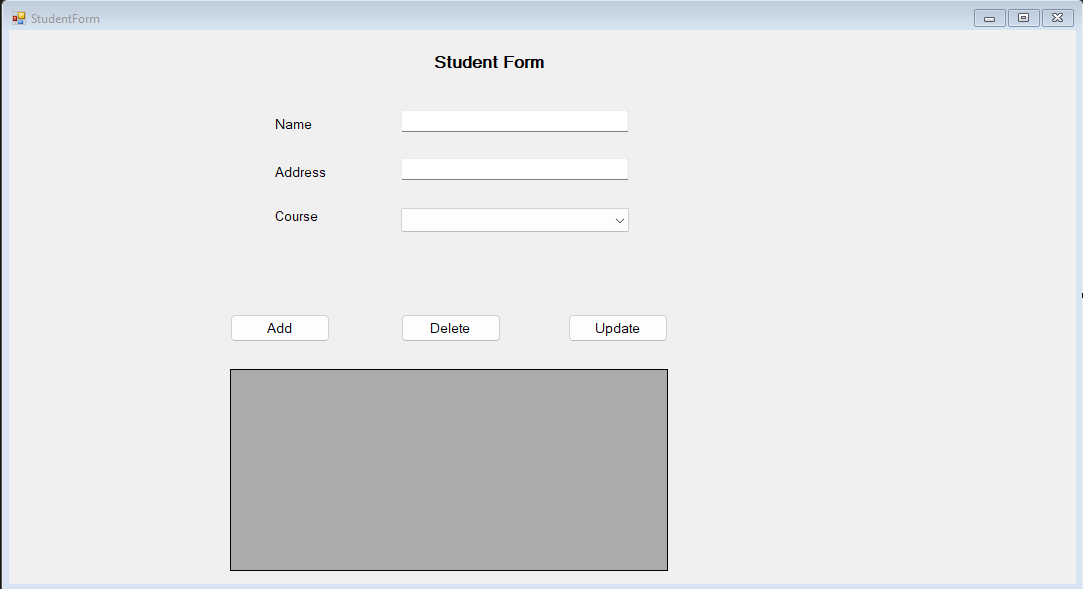
1. **C# (C-Sharp)**
   * The core programming language used to develop the application. C# is widely used in desktop application development and is part of the .NET ecosystem, offering robust support for object-oriented programming, event-driven programming, and asynchronous operations.
2. **WinForms (Windows Forms)**
   * A graphical user interface (GUI) framework for building desktop applications in C#. WinForms was used to create the application's user interface, including forms, buttons, textboxes, labels, and data grids. It provides a simple, efficient way to develop desktop applications with minimal complexity.
3. **SQLite**
   * A lightweight, serverless relational database used to store application data. SQLite was chosen for its simplicity and ability to embed the database directly within the application, eliminating the need for a separate database server. It is ideal for small to medium-sized applications that require efficient, local data storage.
4. **System.Data.SQLite**
   * A .NET library used to interact with the SQLite database. This library allows the application to execute SQL queries, handle database connections, and manage data. It integrates seamlessly with C# and is essential for database operations such as adding, updating, and deleting records.
5. **Model-View-Controller (MVC) Design Pattern**
   * The project follows the **MVC** architectural pattern to separate concerns and organize code into three distinct components:
     + **Model**: Represents the data and business logic (e.g., Course, Student, Exam models).
     + **View**: The user interface components (forms and controls).
     + **Controller**: Manages user interactions, communicates with the model, and updates the view accordingly.
   * The MVC structure promotes code reusability, scalability, and maintainability.
6. **DataGridView (WinForms Control)**
   * A powerful WinForms control used to display tabular data, such as the list of students, courses, or timetable entries. It provides features like sorting, filtering, and editing, allowing the user to interact with data seamlessly.
7. **Buttons and ComboBoxes (WinForms Controls)**
   * Buttons were used for actions like login, adding records, and navigating between forms. ComboBoxes were used to allow users to select options like courses, subjects, and rooms (for timetable entries), providing a smooth user experience.

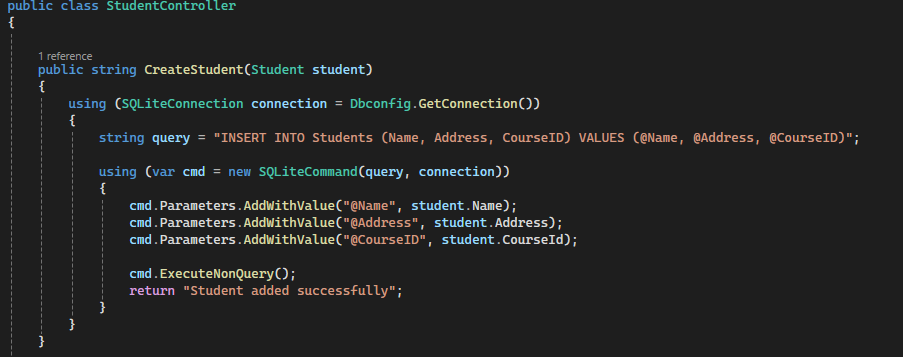
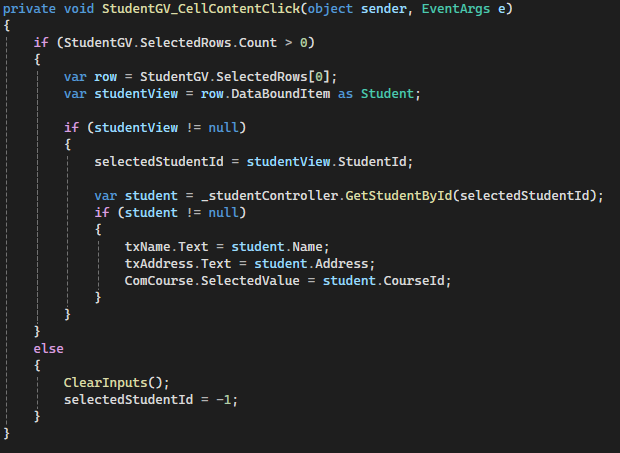
Code Highlights

* Basic student model class.
* Using Encapsulation



* + **Data base connection**
  + **Auto opening**

****

* **Student form design**
* Add student Function
* **When I click the space on the side in the grid view, the name, address, etc., should be shown in the text box**.

Conclusion

The Unicom TIC Management System (UMS) is a beginner-friendly C# WinForms application built with the MVC pattern and SQLite database to help manage core school functions such as courses, subjects, students, exams, marks, and timetables. It introduces basic desktop application development concepts including user login with role-based access, data management, form controls, and database integration.

By developing UMS, students gain hands-on experience in:

* Creating a multi-role login system (Admin, Staff, Student, Lecturer)
* Performing CRUD operations in a database using System.Data.SQLite
* Structuring applications using the MVC architecture
* Designing responsive interfaces with WinForms components
* Implementing real-world school scenarios such as assigning labs and halls to classes

This project serves as a practical foundation in C# development, object-oriented programming, UI design, and database handling. It prepares learners for building more advanced systems by focusing on simplicity, usability, and code structure.