Unicom TIC Management System

**Developed Using: C#, WinForms, SQLite**

**Submitted by:** **A.Annet**

**UT Number:** **UT010188**

# Introduction

The **Unicom TIC Management System (UMS)** is a basic yet comprehensive desktop application designed to help manage daily academic operations within an educational institution. This beginner-friendly project aims to help understand essential concepts in C#, Windows Forms, MVC architecture, database management using SQLite, and rolebased access systems.

The Unicom TIC Management System is a desktop application designed to manage basic operations of a school. This project aims to provide a simple and efficient way to handle courses, subjects, students, exams, marks, and timetables.

# Acknowledgement

I would like to thank Unicom TIC and My Lecturers,Mentors and my friends whose guidance and encouragement helped me complete this project. I also appreciate the support from my peers and the resources provided by Microsoft and SQLite documentation that greatly aided the development process.

# Objective of the Project

* To build a user-friendly, responsive desktop application using **C# WinForms**.
* To understand and apply the **MVC design pattern** for better organization of application logic.
* To manage data using **SQLite** with create, read, update, and delete (CRUD) operations.
* To implement **role-based login access** for Admin, Staff, Lecturers, and Students.
* To allow basic scheduling through **timetable management** with room allocation (labs/halls).
* To demonstrate form design and data-binding techniques with proper validation and error handling.
* Extra Features

## Role-Based Login System

* A secure login system with different roles

**Admin** – Full access: can add/edit/delete all data including courses, students, exams, marks, timetables, and attendance.

**Staff** – Can manage exams and marks and view timetables.

**Lecturer** – Can view timetable and mark/edit student attendance and marks. **Student** – Can view their own profile, timetable, marks, and attendance.

* Role-based dashboards hide or show features depending on the user’s access level.

## Room Allocation (Lab/Hall) in Timetable

* Admin can assign a **Room (Lab or Hall)** when adding a timetable entry.
* Rooms are stored in a dedicated table with type (Lab, Hall).
* ComboBox is used to select room.

## Form View and Hide (Navigation)

* After login, the **LoginForm is hidden**, and the **MainForm is shown**.
* Forms are shown or hidden dynamically using Show() and Hide() methods.
* On logout, current form closes and **LoginForm is reopened**.

## Pop-up Messages

• Important alerts and confirmations using MessageBox.Show():

* Login success/failure.
* Data saved or updated (e.g., “Timetable saved!”). o Validation errors (e.g., “Please select a room”).

## Error Logging

• Errors such as failed logins, database issues, or form validation failures are **logged to a text file** (errorlog.txt).

## Technical Details

|  |  |
| --- | --- |
| **Feature** | **Technology Used** |
| GUI | Windows Forms (WinForms) |
| Language | C# (.NET Framework) |
| Database | SQLite (Local database) |
| **Feature** | **Technology Used** | |
|  | System.Data.SQLite via NuGet | |
| Architecture | MVC (Model-View-Controller) | |
| UI Elements | DataGridView, ComboBox, Buttons,  TextBoxes | |
| Data Storage | .db file for SQLite | |

## Budget Plan

**Estimated Cost**

**Item**

**(USD)**

$0 (Free version

Visual Studio Community

used)

SQLite $0 (Open-source)

System.Data.SQLite $0 (Installed via

Library NuGet)

Hardware (Laptop/PC) $0 (Personal use)

**$0 (No cost**

Total

**incurred)**

## Design

Use Case Diagram ER Diagram

Architecture

* DTO mapper entity
* Interfacerepository
* Repository
* Interfaceservice
* Sercice
* Controller
* Viewform

### References

* Microsoft Docs
* SQLite Official Site
* C# Tutorials: Tamil Programmer C#
* System.Data.SQLite: https://system.data.sqlite.org/
* YouTube Tutorials on WinForms & MVC
* With folder structure controllers,services,repositories ,DTos,Enums,Mappers
* Stack Overflow for debugging and best practices
* Get help from Chatgpt

Challenges

* + Table query is not written correctly
  + Runtime error
  + Role base Login
  + Exam Marks management
  + Timetable Management Time slot
  + Data is not getting saved in DataGridView
  + Repository (Database Manager)

Overcome

* + I ran each form and noted down the errors or missing parts...
  + I corrected each form accordingly, and then checked whether the data was being saved in the database...
  + I also took help from ChatGPT.

### 2. Code Samples (Screenshots)

* Loading enums items into combo box and inserting form into panel by selecting
* Login Main Form Run
* CRUD functions for staff,course,students ,lecturers,subjects ,etc
* Converting entity to DTOs and Viveversa
* Correctly using interface ; declaring signature of methods

