A

Project Report On

“UnicomTIC Management System”

Miss. M. Emima Niromy

UT010334

Guidance of

Mr. Kathir

Date: 24.06.2025

**Table of Contents**

• Introduction

• Acknowledgement

• Current system

• Problem definition

• Customer requirements

• Technology used and feasibility studies

• Project schedule

• ER diagram

• Code screen shot and explanation

• References

• Conclusion

***Introduction***

The UnicomTIC Management System is a desktop-based application developed using C# (Windows Forms) and SQLite to streamline academic and administrative tasks within an educational institution. The system is designed to support role-based access for different users, such as administrator, students, lecturers, and staff, allowing them to access only the features relevant to their specific roles.

This project aims to improve efficiency in handling student records, exam mark entries, timetable management, and subject/course allocation. By automating these processes, the system reduces manual errors, improves data accuracy, and saves time. The user-friendly interface ensures that all users can interact with the system easily without requiring advanced technical skills.

The application is built following object-oriented programming principles and adopts a modular architecture using the MVC (Model-View-Controller) design pattern. This structure ensures better code organization, scalability, and ease of maintenance, making the system suitable for real-world deployment in academic environments.

***Acknowledgement***

I owe a great many thanks to a great many people who helped and supported me during the making of this project

Firstly, I am truly grateful to my project supervisor, Mr. Kathir, for his constant support, clear instructions, and valuable feedback throughout the development of this system. His guidance helped me understand both theoretical and practical aspects of software development, and his encouragement kept me motivated at every stage.

I also extend my appreciation to the administrative staff and lab instructors for providing the necessary resources, infrastructure, and learning environment that played a crucial role in the success of this project.

Special thanks go to my classmates and friends for their input and support during testing and brainstorming phases. Lastly, I am deeply thankful to my family for their unwavering support, patience, and encouragement during the entire duration of this project.

I would also thank the institution and the faculty members without whom this project would have been a distant reality. I also extend my heartfelt thanks to well-wishers.

***Current System***

Currently, many educational institutions manage student, staff, and academic information manually using tools such as paper records, Excel spreadsheets, or disconnected software systems. These traditional methods are often time-consuming, error-prone, and inefficient.

For instance, exam results are recorded manually, timetables are shared through printed notices or informal channels, and student details are maintained in scattered files. This leads to several issues.

Students often face challenges in checking their marks, attendance, or class schedules without approaching administrative staff directly. Likewise, lecturers and administrative personnel spend excessive time managing routine academic tasks, which affects productivity.

This project was undertaken to address these challenges and introduce an efficient, centralized, and user-friendly system that digitizes and simplifies academic operations. So, I aimed to build a system that reduces manual effort and enhances the overall academic management process.

***Problem Definition***

Most educational institutions still use manual methods or spreadsheets to manage students, staff, exams, and timetables. This leads to:

• Data entry errors

• Delays in accessing information

• No role-based access control

• Difficulty for students to view marks or timetables

• Poor data security

To solve these issues, an automated system is needed that offers secure, role-based access, faster data handling, and easy access to academic information. The UnicomTIC Management System is designed to meet this need efficiently.

***Customer Requirements***

To ensure the UnicomTIC Management System effectively serves its users, the following requirements were identified:

1. **Functional Requirements**

**▪ Secure Login & Role Assignment**

• Each user (Admin, Lecturer, Staff, Student) must log in using a username and password.

• Users are granted access based on their assigned roles.

**▪ Customized Dashboards**

• Admin: Full access to manage users, subjects, courses, marks, exams, timetables, and rooms.

• Lecturer: Can view and manage only their assigned subjects and class schedules.

• Staff: Access to specific administrative data relevant to their duties.

• Student: Limited access to their own marks, room allocation and timetable.

***Data Management Features***

• Ability to add, update, delete, and view information for:

• Students, Staff, Lecturers

• Courses and Subjects

• Exam Marks and Timetables

• Staff and Lecturer profiles include essential fields like name, role, ID, and contact info.

• Restricted Access Views

• Students and Lecturers can only view (not edit) academic data assigned to them.

1. **Non-Functional Requirements**

▪ Security

• Role-based access control to protect data and prevent unauthorized actions.

▪ System Performance

• The system should respond quickly and perform smoothly even when multiple users are logged in simultaneously….

|  |  |
| --- | --- |
| ***Technology Used*** | |
| **Components** | **Technology** |
| **Language** | **C#** |
| **Framework** | **.NET(WinForms)** |
| **Database** | **SQLite** |
| **Architecture** | **MVC** |
| **Tools** | **Visual Studio, DB for SQLite** |

***Project Schedule***

The development of the UnicomTIC Management System was planned and completed in the following stages:

1. Requirement Analysis

• Duration: 3-5 days

• Activities: Understanding user needs, defining system features, and identifying key modules.

1. System Design

• Duration: 2 Day

• Activities: Designing database structure, UI layout, and module flow using the MVC model.

3.Development

• Duration: 2–3 Weeks

▪ Activities:

• Coding login and user role

• Implementing student, lecturer, course, subject, and timetable modules

5. Final Report

• Duration: 3 Days

• Activities: Preparing documentation, screenshots

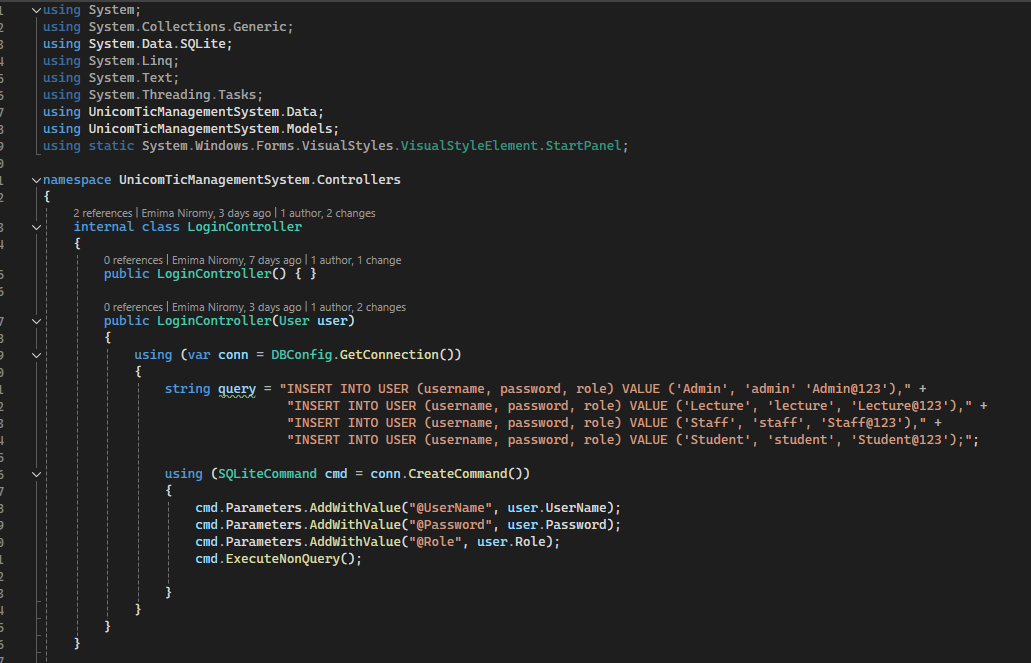
***ER Diagram (Text Version)***

***Entities & Attributes***

* **User**
* Id
* Role
* Username
* Password
* **Student**
* Id
* First Name
* Last Name
* Address
* Stream
* **Staff**
* Name
* Address
* **Lecture**
* Id
* Name
* Address
* **Course**
* Id
* Name
* Start Date
* End Date
* **Subject**
* Id
* Name
* Course Id
* **Exam**
* Id
* Name
* Subject Id
* **Timetable**
* Id
* Time Slot
* Subject Id
* Room Id
* **Room**
* Id
* Name
* Type

***Code Screenshots and Explanation***

* Login Controller

******

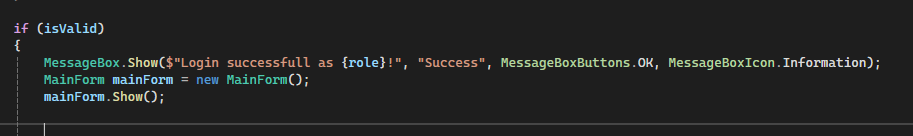
Inside the constructor, an SQL query is created to insert four different user types into the user table. These users are:

1. Admin
2. Lecture
3. Staff
4. Student

“The code inserts four different users into the database with predefined usernames, passwords, and roles (Admin, Lecture, Staff, and Student)”.

* Login Form .cs





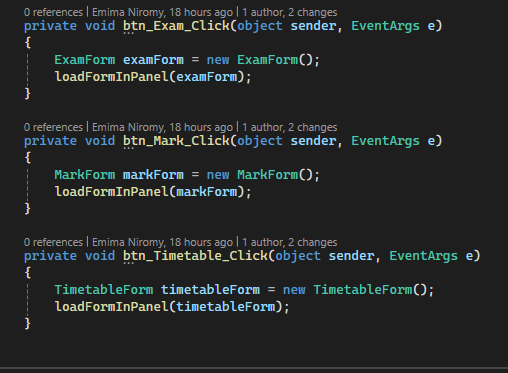
This code is part of a **login validation** for a user

This code is part of a **login validation** for a user

If the login is successful (isValid is true), it displays a success message and opens the Main Form:

If the login fails (i.e., no match found), the system doesn't handle it explicitly here, but you can easily add a failure message if desired

* Admin Dashboard Form



This method takes any Form as an argument (ChildForm) and dynamically loads it into panel3.

Making the child form occupy the entire panel3 by setting its Dock property to Fill.

Whenever any of the buttons are clicked, a new form (e.g., Login Form, Student Form, etc.) is created and loaded into panel3.

The panel3 acts as a container for all these forms. Instead of opening each form as a new window, all the forms are shown inside panel3

***References***

**•** C# Programming Guide **– Microsoft ,**

**•** Microsoft Docs **– Windows Forms Documentation ,**

**•** SQLite Documentation**,**

**•** W3Schools C# and SQL Tutorials **,**

**•** Support from Mr.Kathir(Lecturer) and Lab instructor**,**

**•** ChatGPT.

***Conclusion***

The UnicomTic Management System effectively addresses the challenges faced by educational institutions in managing academic operations such as session scheduling, lecture and lab hall allocation, and subject management. By digitizing these processes, the system reduces manual effort, minimizes human errors, and ensures data consistency across departments.

Built using C# with a WinForms interface and SQLite as the backend, the system offers a userfriendly platform that allows Admins, Lecturers, and Students to access and manage relevant information according to their roles. The modular design and role-based access control improve both usability and security.

Through this project, we demonstrated how automation can improve the efficiency and transparency of university academic planning. The system is scalable and can be enhanced in the future with additional features like attendance analytics, notification systems, and mobile app integration, making it a comprehensive solution for university management needs