**MMCM Computer Laboratory Manager**

A Hands-on Project in IT102 Object-Oriented Programming

In Partial Fulfillment of:

BS in Computer Science

Term 1, School Year 2024-2025



Date of Submission:

November 6, 2024

Course Name:

IT102 – Object Oriented Programming

Team Members:

Aque, Kit Janbren

Magtibay, Zuriel

Yleaña, John Cyrus

Submitted to:

Clyde Balaman | ccrbalaman@mcm.edu.ph

**Table of Contents**

**1. Introduction** 3

1.1. Needs

1.2. Benefits

1.3. Solution

1.4. Differentiation

**2. Objectives** 4

2.1. General Objective

2.2. Specific Objectives

2.2.1. Specific Objective 1

2.2.2. Specific Objective 2

2.2.3. Specific Objective 3

**3. Object-Oriented Programming Discussion** 5

3.1. Understanding OOP

3.2. Implementation in the Project

**4. Diagrams** 6

4.1. Conceptual Diagram

4.2. Class Diagram

4.3. Entity-Relationship Diagram

**5. User Manual** 7

5.1. User Interface Components

5.2. Database Environment

**6. Test Plan** 8

6.1. Test Cycles

**7. Results and Discussion** 9

**8. Members Curriculum Vitae** 10

**1. Introduction**

1.1 Needs

This application addresses the school computer laboratory’s need for an efficient time recording software. Our application provides a secure and centralized system to track student logins, monitor usage duration, and manage access control, ensuring that each session is accurately recorded. By automating login validation, session tracking, and account management, this solution not only enhances the security of lab resources but also allows administrators to streamline oversight of computer usage.

1.2 Benefits

For **professors**, this application brings improved oversight and ease of access, allowing them to reserve computer lab time, view student usage, and promote responsible digital conduct within the lab. Professors can monitor student engagement in real-time, enabling them to ensure students are focused on academic tasks and to intervene if misuse occurs.

For **students,** the system offers a seamless and fair experience, with clear time-tracking that ensures equitable access to lab computers, particularly during peak usage times. Students can view their own login history, fostering accountability and time management, while benefiting from an efficient login/logout process.

1.3 Solution

The **solution** provided by this application is a comprehensive computer lab management system designed to efficiently monitor and manage student activity within the school computer lab. It addresses key issues by allowing students to securely log in using their student ID and password, tracking their usage time and automatically recording their session duration. Professors and administrators can monitor active sessions from a central dashboard on the server, where they can see which computers are in use and view individual session details, such as login history and usage hours, for each student.

The system also includes features to improve security and accountability, such as account locking after multiple failed login attempts, and administrative controls for unlocking accounts and creating new ones directly in the application. Additionally, it provides a clear log-out mechanism to ensure that student sessions end properly, updating both the client-side and server records.

1.4 Differentiation

This application distinguishes itself by providing the school’s first structured system for managing computer lab usage and tracking student activity in real-time. Unlike existing solutions**—which are largely manual or non-existent**—this application automates critical aspects of lab management, such as time-tracking, login authentication, session monitoring, and account security.

**2. Objectives**

2.1 General Objective

The general objective of this project is to develop a comprehensive computer laboratory management system that streamlines the monitoring, usage, and security of computer resources in a school setting. This system aims to automate time-tracking, user authentication, and session management, providing real-time insights and secure access control to ensure efficient and accountable use of computer lab facilities by students.

2.2 Specific Objectives

1. To implement a secure login and account management system

Develop a login system that allows students to authenticate with unique IDs and passwords, tracks login attempts, and prevents unauthorized access. [Specific objective 2]

2. To create a session tracking system

Design and implement a timer-based session management feature that records each student’s usage time. This system should track login and logout times, calculate session durations, and log this data in a centralized database.

3. Streamline User Access and Logout Processes

Design and implement login and logout workflows that are quick and user-friendly, supporting a smooth experience and accurate session tracking for all users accessing the computer lab systems.

**3. Object-Oriented Programming Discussion**

3.1 Understanding OOP

Object-oriented programming (OOP) is built around four core principles: encapsulation, inheritance, polymorphism, and abstraction. Encapsulation bundles data and methods into a class while restricting access to protect the integrity of the data, promoting modularity. Inheritance allows a subclass to inherit properties and behaviors from a superclass, enabling code reuse and hierarchical relationships. Polymorphism enables objects of different classes to be treated as instances of a common superclass, allowing methods to operate on various types of objects flexibly. Abstraction simplifies interaction with complex systems by hiding implementation details and exposing only essential features, typically through abstract classes or interfaces. Together, these principles foster the development of robust, scalable, and maintainable software systems.3.2 Implementation in the Project

In my Computer Management project, I applied the principles of object-oriented programming (OOP) as follows:

**Encapsulation:** Encapsulation in SF\_dashboardForm and ClientForm1: Both forms encapsulate properties and methods specific to their purpose. For instance, the database connection and TCP communication handling methods (ConnectToDatabase, ConnectToServer) are private methods, restricting direct access from outside the class. Variables such as connection and listener in SF\_dashboardForm and client in ClientForm1 are private, limiting their accessibility. The private methods HandleClientConnection and UpdateLoginTime provide controlled access to the underlying database and connection operations, encapsulating those details.

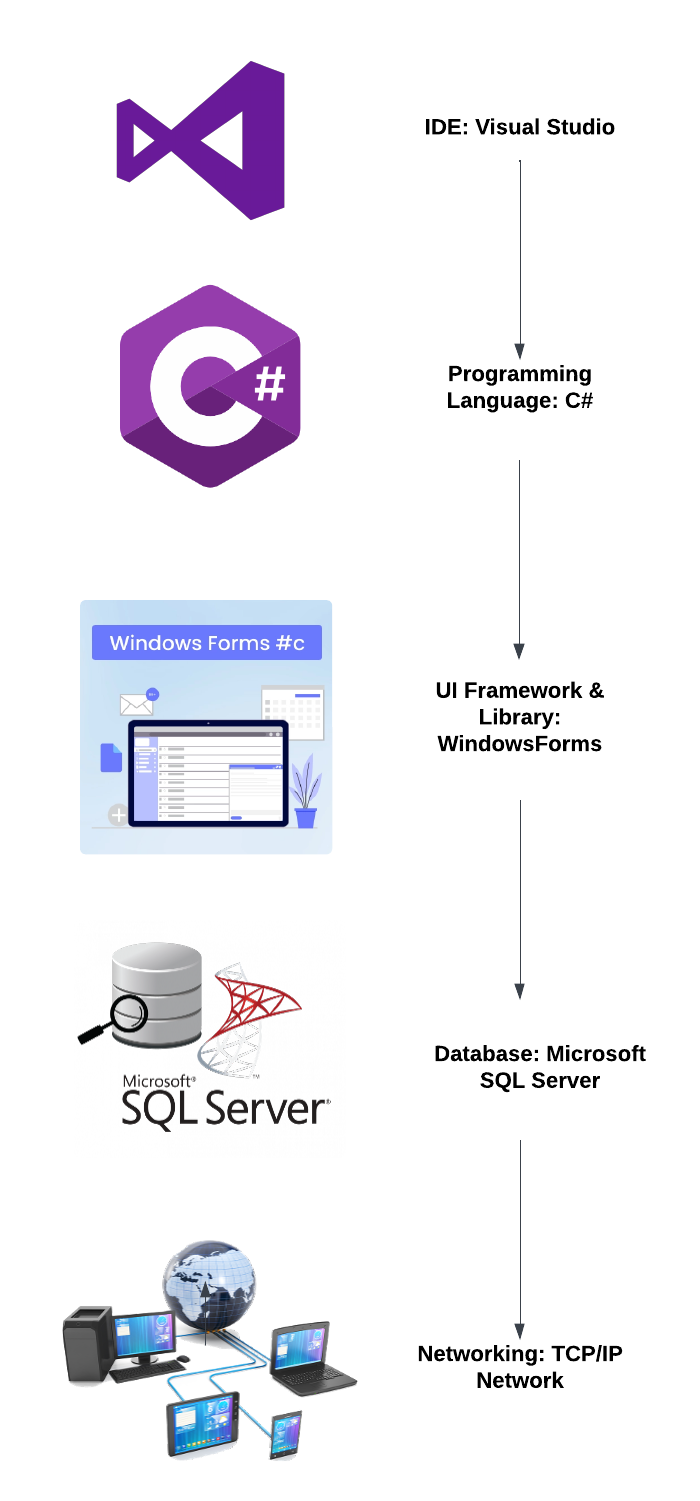
**Inheritance:** Both SF\_dashboardForm and ClientForms inherit from Form, allowing them to extend the base functionality provided by Windows Forms. This inheritance is fundamental to C# Windows Forms applications, enabling these classes to use UI components and event handling from the Form class.

**Polymorphism:** Event Handlers as Polymorphic Methods: In ServerForm.cs, methods like SF\_createaccountButton\_Click\_1, SF\_lockedaccountsButton\_Click\_1, and button2\_Click exhibit polymorphism as they follow a common signature (object sender, EventArgs e). These event handlers are called based on specific events in the UI, illustrating runtime polymorphism.

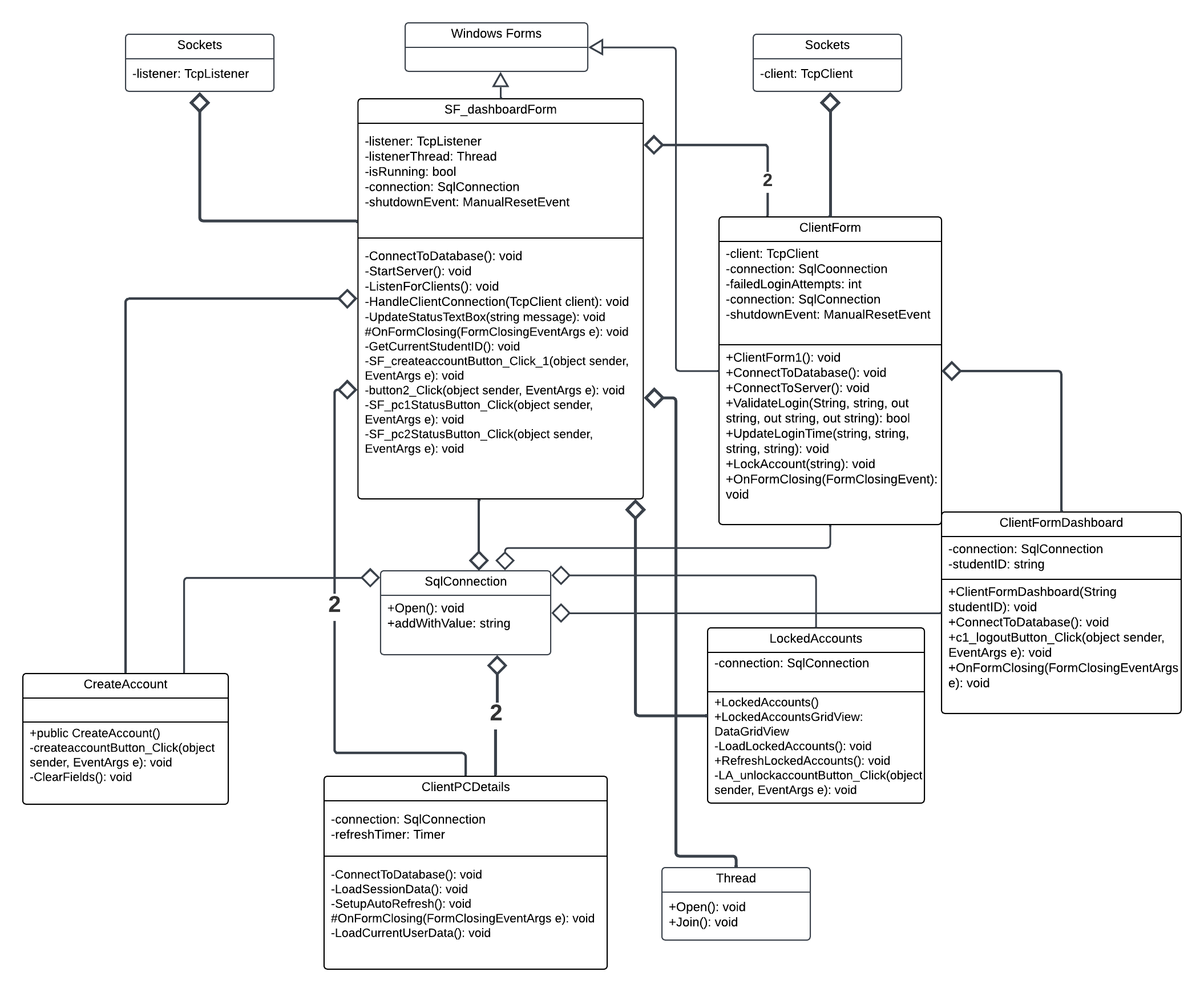
**Abstraction:** ConnectToDatabase and ConnectToServer abstract the connection details. The end-user or another part of the application only calls these methods to establish connections without needing to know the underlying SQL connection strings or TCP/IP protocols.

**4. Diagrams**

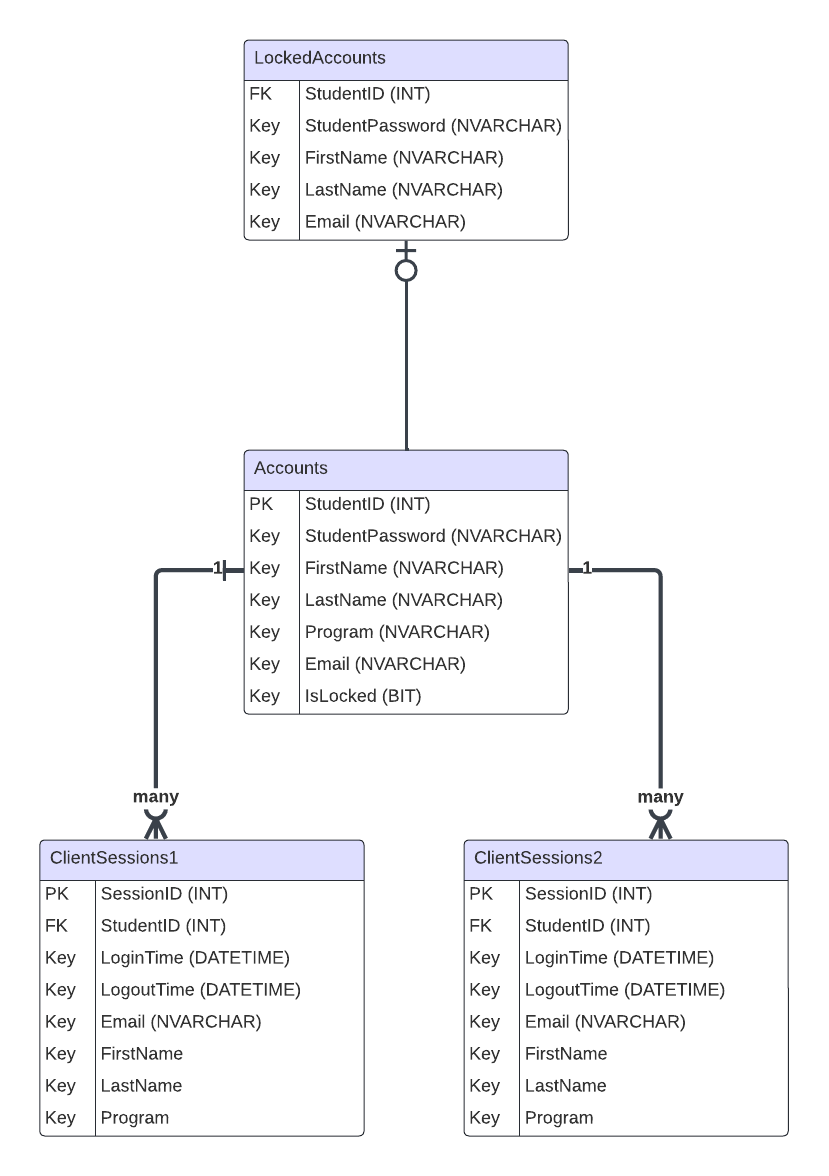
4.1 Conceptual Diagram



4.2 Class Diagram



4.3 Entity-Relationship Diagram



**5. User Manual**

5.1 User Interface Components

**Client PC:**

**ClientForm1.cs & ClientForm2.cs**

A login screen with a logo and text

Description automatically generatedA login screen with a logo and text

Description automatically generated

Provides the user login interface for both of client computers. It has the StudentID and Password textbox which is connected to the Accounts table. ClientForm1 login button is connected to the ClientSessions1 table while ClientForm2 login button is connected to the ClientSessions2 table. It’s security involves locking the account when a user fails to enter the corresponding password to the valid student ID. The account is appended to the LockedAccounts table in SQL Server.

**ClientForm1Dashboard.cs & ClientForm2Dashboard.cs**

**A screenshot of a computer error

Description automatically generatedA screenshot of a computer

Description automatically generated**

Provides the information about the user that logged in on that particular PC. It lists the StudentID on the label and the logout button is connected to the ClientSessions ‘LogoutTime’ table.

**Server PC:**

**ServerForm.cs**

**A screenshot of a computer

Description automatically generated**

Is composed of a read-only textbox where it is updated whenever a ClientForm.cs instance is active. It has view option to show the ClientPCDetails1 and ClientPCDetails2 forms and CreateAccount.cs and LockedAccount.cs buttons.

**ClientPCDetails1.cs & ClientPCDetails2.cs**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Is connected to the ClientSessions1 and ClientSessions2 table in SQL. It consists of two ‘Data grid view’ boxes. The first gridview shows the current user that is connected in the corresponding client PC while the second gridview shows the history of the logins

**CreateAccount.cs**

A screenshot of a computer

Description automatically generated

Handles the account creation for login. It takes in a First name, Last name, Program, School Email, Student ID, and password confirmation. The “create account” button is connected to the Accounts sql table where a new account is appended whenever it is pressed.

**LockedAccounnt.cs**

**A screenshot of a computer

Description automatically generated**

The LockedAccount.cs form shows the current locked accounts which have failed to authenticate the correct password for three (3) times in the ClientForm. The server PC dashboard has the means to unlock these locked accounts by taking them from the LockedAccounts table back into the Accounts table.

5.2 Database Environment

The database name is ‘NewComputerManagementDB’ with a user which has an ‘owner’ status which named ‘admin1’. This ensures that the client PCs can access to the database hosted by the server PC without using the default ‘Windows Authentication’ by the SQL manager.

A screenshot of a computer

Description automatically generated 

dbo.Accounts:

Handles the Accounts that are created and eligible to log-in on the application. StudentID is int and Primary Key, StudentPassword, FirstName, Lastname, Program, and Email are nvarchar and isLocked is in bit (1 for locked).

A screenshot of a computer

Description automatically generated

dbo.ClientSessions1:

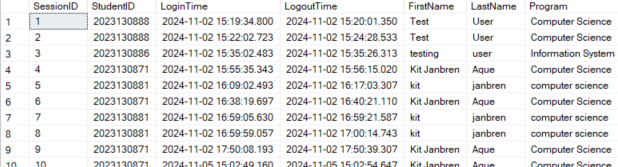
Records the accounts that were logged in on ClientForm1.cs and were logged out on ClientDashboard1.cs. It has a many-to-one relationship with dbo.Accounts which means an account can have as many sessions as it can log in on the ClientForm1.cs

A screenshot of a computer

Description automatically generated

dbo.ClientSessions2:

Records the accounts that were logged in on ClientForm2.cs and were logged out on ClientDashboard2.cs. It has a many-to-one relationship with dbo.Accounts which means an account can have as many sessions as it can log in on the ClientForm2.cs



dbo.LockedAccounts:



Keeps the accounts that are locked when they fail to authenticate the correct password in ClientForms. It has a zero-to-one relationship with dbo.Accounts.

**6. Test Plan**

6.1 Test Cycles

|  |
| --- |
| **TEST CYLE 1** |
| The goal of the first test case was to create a simple outline for the structure of the entire program. It was quite ambitious at first especially for the ‘ClientTimer’ which we think would track the user session of the client PCs. |
| Client-side login form |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Case ID** | **Test Case Name** | **Objective** | **Preconditions** | **Test Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** | | TC001 | ClientForm | Shows the Client Login form on the Client PC only | Application must be running on Client PC | 1. Open the application on Client PC 2. The Login form should cover the whole screen | The user is prompted to enter a student ID and password with a form that covers the screen |  | Fail | | TC002 | ClientLoginAttempt | Limit login attempts to 3 before locking the account | Application running on Client PC, user has an account | 1. Enter an invalid student ID 2. Repeat until 3 failed attempts 3. Try logging in again | After 3 failed attempts, the user is locked out and prompted to contact admin |  | Fail | | TC003 | AdminUnlockAccount | Admin unlocks the locked client account | A client account is locked after 3 failed login attempts | 1. Run dashboard on Server PC 2. Navigate “Locked accounts” 3. Click “Unlock account” | The locked account is unlocked, and the client can log in again |  | Fail | | TC004 | ClientTimer | Records the duration of the client’s login state | The client is logged in on client PC | 1.Login as client   1. Stay logged in for a certain amount of time 2. Log out 3. View recorded time | The system records only the duration of the client’s active login session |  | Fail | | TC005 | ClientLogOut | Client logs out, and the login form covers the entire screen again | The client is logged in | 1. Click “Logout” 2. Confirm logout action | Client logs out and the login form covers the entire screen again |  | Fail | |
| Server-Side Dashboard |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | TC006 | ServerDashboard | Displays active client PCs on the Server dashboard | Server application must be running on Server PC | 1.Open the server application on the Server PC  2. Check the dashboard for logged-in clients | The servers shows two active client PCS, other PCs displayed as images and labels |  | Fail | | TC007 | ViewClientProfile | Admin can view client profile including login history | A client has logged into the system, data exists in database | 1.Click on a logged-in client in the server dashboard  2. Select “View Profile” | Admin views the client's profile, including login history and hours used |  | Fail | | TC008 | CreateAccount | Create a new client account for login/logout tracking in SQL | Server Application running with access to the database | 1. Admin navigates to the “Create Account” button 2. Enter client details (Student ID, course, password) 3. Save | The account is created on the SQL database |  | Fail | | TC009 | NetworkConnection | Ensure server and client communicate over a network | Server and client are connected via network | 1.Start server application  2.Start client application  3.Try logging in clients  4. The server dashboard would see the client PC connected | Both clients communicate with each other (Client PC logs in which is now detected by the Server dashboard) |  | Fail | |

|  |
| --- |
| **TEST CYCLE 2** |
| For the Test Case 2, the ClientForms were shaping up but we ran into some trouble with the ClientTimer. We found out that it was very complicated and way above our skill level so we decided to drop it on Test Case 2. |
| Client-side login form |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Case ID** | **Test Case Name** | **Objective** | **Preconditions** | **Test Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** | | TC001 | ClientForm | Shows the Client Login form on the Client PC only | Application must be running on Client PC | 1. Open the application on Client PC 2. The Login form should cover the whole screen | The user is prompted to enter a student ID and password with a form that covers the screen | Client is prompted to enter student ID and password with a form that covers the entire screen | Pass | | TC002 | ClientLoginAttempt | Limit login attempts to 3 before locking the account | Application running on Client PC, user has an account | 1. Enter an invalid student ID 2. Repeat until 3 failed attempts 3. Try logging in again | After 3 failed attempts, the user is locked out and prompted to contact admin | Client account is locked and cannot proceed to dashboard | Pass | | TC003 | AdminUnlockAccount | Admin unlocks the locked client account | A client account is locked after 3 failed login attempts | 1. Run dashboard on Server PC 2. Navigate “Locked accounts” 3. Click “Unlock account” | The locked account is unlocked, and the client can log in again | Client can now use the locked account | Pass | | TC004 | ClientLogOut | Client logs out, and the login form covers the entire screen again | The client is logged in | 1. Click “Logout” 2. Confirm logout action | Client logs out and the login form covers the entire screen again | Client is logged out | Pass | | TC005 | ClientDashboard | After client successfully validates their student id, a client dashboard opens with their ID | Client logged in (validated studentID) | 1. Client clicks login button after entering student ID and password | ClientDashboard is in the screen with their studentID | ClientDashboard is now on the screen | Pass | |
|  |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | TC006 | ServerDashboard | Displays active client PCs on the Server dashboard | Server application must be running on Server PC | 1.Open the server application on the Server PC  2. Check the dashboard for logged-in clients | The servers shows two active client PCS, other PCs displayed as images and labels | ServerDashboard opens up and displays two pictureboxes with PCs | Pass | | TC007 | ViewClientProfile | Admin can view client profile including login history | A client has logged into the system, data exists in database | 1.Click on a logged-in client in the server dashboard  2. Select “View Profile” | Admin views the client's profile, including login history and hours used | ServerDashboard can see the client profile with login history | Pass | | TC008 | CreateAccount | Create a new client account for login/logout tracking in SQL | Server Application running with access to the database | 1. Admin navigates to the “Create Account” button 2. Enter client details (Student ID, course, password) 3. Save | The account is created on the SQL database | Account now exists in dbo.Account table in SQL | Pass | | TC009 | NetworkConnection | Ensure server and client communicate over a network | Server and client are connected via network | 1.Start server application  2.Start client application  3.Try logging in clients  4. The server dashboard would see the client PC connected | Both clients communicate with each other (Client PC logs in which is now detected by the Server dashboard) | Server dashboard detects the existence of the two client PCs | Pass |   Server-side Dashboard |
|  |

|  |
| --- |
| **TEST CYCLE 3**  For test case 3, the TCP/IP protocols and SQL connections were established. We ran into some trouble about the TCP/IP Protocol and the Client connection to the SQL server due to firewall settings. At the end, we successfully established a connection from client to the server hosting the SQL database. We added the SQLConnection (TC002) Test case and DeleteAccount (TC010) Test case. |
| Client-side Dashboard |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Case ID** | **Test Case Name** | **Objective** | **Preconditions** | **Test Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** | | TC001 | ClientForm | Shows the Client Login form on the Client PC only | Application must be running on Client PC | 1. Open the application on Client PC 2. The Login form should cover the whole screen | The user is prompted to enter a student ID and password with a form that covers the screen | Client is prompted to enter student ID and password with a form that covers the entire screen | Pass | | TC002 | ClientLoginAttempt | Limit login attempts to 3 before locking the account | Application running on Client PC, user has an account | 1. Enter an invalid student ID 2. Repeat until 3 failed attempts 3. Try logging in again | After 3 failed attempts, the user is locked out and prompted to contact admin | Client account is locked and cannot proceed to dashboard | Pass | | TC002 | SQLConnection | Client is connected to the SQL Database of the Server PC | Client should have both of their ClientForms open | 1. Run the ClientForm.exe file | 1.ClientForm is connected to the Server SQL Database | Client is connected to the SQL database after opening the ClientForm exe file | Pass | | TC004 | AdminUnlockAccount | Admin unlocks the locked client account | A client account is locked after 3 failed login attempts | 1. Run dashboard on Server PC 2. Navigate “Locked accounts” 3. Click “Unlock account” | The locked account is unlocked, and the client can log in again | Client can now use the locked account | Pass | | TC005 | ClientLogOut | Client logs out, and the login form covers the entire screen again | The client is logged in | 1. Click “Logout” 2. Confirm logout action | Client logs out and the login form covers the entire screen again | Client is logged out | Pass | | TC006 | ClientDashboard | After client successfully validates their student id, a client dashboard opens with their ID | Client logged in (validated studentID) | 1. Client clicks login button after entering student ID and password | ClientDashboard is in the screen with their studentID | ClientDashboard is now on the screen | Pass |   Server-side Dashboard |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | TC006 | ServerDashboard | Displays active client PCs on the Server dashboard | Server application must be running on Server PC | 1.Open the server application on the Server PC  2. Check the dashboard for logged-in clients | The servers shows two aDasboarctive client PCS, other PCs displayed as images and labels | ServerDashboard opens up and displays two pictureboxes with PCs | Pass | | TC007 | ViewClientProfile | Admin can view client profile including login history | A client has logged into the system, data exists in database | 1.Click on a logged-in client in the server dashboard  2. Select “View Profile” | Admin views the client's profile, including login history and hours used | ServerDashboard can see the client profile with login history | Pass | | TC008 | CreateAccount | Create a new client account for login/logout tracking in SQL | Server Application running with access to the database | 1. Admin navigates to the “Create Account” button 2. Enter client details (Student ID, course, password) 3. Save | The account is created on the SQL database | Account now exists in dbo.Account table in SQL | Pass | | TC009 | DeleteAccount | Delete account in the database | Server Application must be running to access to the sql database | 1.Run ServerForm.exe file  2.Open the DeleteAccount form | Account is deleted from the SQL Database using the ServerForm dashboard | Account is now deleted from dbo.Accounts in SQL table | Pass | | TC010 | NetworkConnection | Ensure server and client communicate over a network | Server and client are connected via network | 1.Start server application  2.Start client application  3.Try logging in clients  4. The server dashboard would see the client PC connected | Both clients communicate with each other (Client PC logs in which is now detected by the Server dashboard) | Server dashboard detects the existence of the two client PCs | Pass | |
|  |

|  |
| --- |
| **TEST CYCLE 4** |
| For the final test case, it was more of UI and SQL integration. We managed to connect tha data grid view for the ClientSessions table to the ClientPCDetails. We added ClientSessions (TC007) and DataGridView(TC013) Test cases. |
| Client-side login form |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Case ID** | **Test Case Name** | **Objective** | **Preconditions** | **Test Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** | | TC001 | ClientForm | Shows the Client Login form on the Client PC only | Application must be running on Client PC | 1. Open the application on Client PC 2. The Login form should cover the whole screen | The user is prompted to enter a student ID and password with a form that covers the screen | Client is prompted to enter student ID and password with a form that covers the entire screen | Pass | | TC002 | ClientLoginAttempt | Limit login attempts to 3 before locking the account | Application running on Client PC, user has an account | 1. Enter an invalid student ID 2. Repeat until 3 failed attempts 3. Try logging in again | After 3 failed attempts, the user is locked out and prompted to contact admin | Client account is locked and cannot proceed to dashboard | Pass | | TC003 | SQLConnection | Client is connected to the SQL Database of the Server PC | Client should have both of their ClientForms open | 1.Run the ClientForm.exe file | 1.ClientForm is connected to the Server SQL Database | Client is connected to the SQL database after opening the ClientForm exe file | Pass | | TC004 | AdminUnlockAccount | Admin unlocks the locked client account | A client account is locked after 3 failed login attempts | 1. Run dashboard on Server PC 2. Navigate “Locked accounts” 3. Click “Unlock account” | The locked account is unlocked, and the client can log in again | Client can now use the locked account | Pass | | TC005 | ClientLogOut | Client logs out, and the login form covers the entire screen again | The client is logged in | 1. Click “Logout” 2. Confirm logout action | Client logs out and the login form covers the entire screen again | Client is logged out | Pass | | TC006 | ClientDashboard | After client successfully validates their student id, a client dashboard opens with their ID | Client logged in (validated studentID) | 1.Client clicks login button after entering student ID and password | ClientDashboard is in the screen with their studentID | ClientDashboard is now on the screen | Pass | | TC007 | ClientSessions | Client PC ‘records’ the accounts that are logged in their respective databases in SQL table | Account should exist in dbo.Accounts | 1.user types in with the valid student id and password  2.click login button | Client PC will record the time of the account login and account logout into the ClientSessions table in SQL | ClientSessionsTable is updated with the current login and logout history of the client PCs | Pass | |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | TC008 | ServerDashboard | Displays active client PCs on the Server dashboard | Server application must be running on Server PC | 1.Open the server application on the Server PC  2. Check the dashboard for logged-in clients | The servers shows two active client PCS, other PCs displayed as images and labels | ServerDashboard opens up and displays two pictureboxes with PCs | Pass | | TC009 | ViewClientProfile | Admin can view client profile including login history | A client has logged into the system, data exists in database | 1.Click on a logged-in client in the server dashboard  2. Select “View Profile” | Admin views the client's profile, including login history and hours used | ServerDashboard can see the client profile with login history | Pass | | TC010 | CreateAccount | Create a new client account for login/logout tracking in SQL | Server Application running with access to the database | 1. Admin navigates to the “Create Account” button 2. Enter client details (Student ID, course, password) 3. Save | The account is created on the SQL database | Account now exists in dbo.Account table in SQL | Pass | | TC011 | DeleteAccount | Delete account in the database | Server Application must be running to access to the sql database | 1.Run ServerForm.exe file  2.Open the DeleteAccount form | Account is deleted from the SQL Database using the ServerForm dashboard | Account is now deleted from dbo.Accounts in SQL table | Pass | | TC012 | NetworkConnection | Ensure server and client communicate over a network | Server and client are connected via network | 1.Start server application  2.Start client application  3.Try logging in clients  4. The server dashboard would see the client PC connected | Both clients communicate with each other (Client PC logs in which is now detected by the Server dashboard) | Server dashboard detects the existence of the two client PCs | Pass | | TC013 | DashboardGridView | Display a message when the Client PCs are connected to the Server | ClientPC forms are opened and connected to the TCP connection | 1.Open the ClientForm.exe files | A message Is sent to the dashboard gridview that client PC are connected | Gridview is updated with a message that the client PCs are connected | Pass | | TC014 | PCDetailsGridView | Display the current user and history of the client PCs on their grid view | Server PC should be started up | 1.Open ServerForm.exe  2. Click “view” on ClientPC1 or ClientPC2 picturebox | Two gridviews will display the current user and the history of the particular PC | Current user gridview is updated with the current user while the history gridview has the data of the logged in user sessions | Pass |   Server-side Dashboard |

**7. Results and Discussion**

**Client-side login:**

1. **User enters password**

**A login screen with a logo and text

Description automatically generated**

1. **User has three tries (3) to enter the correct password**

A screenshot of a computer screen

Description automatically generated

**A screenshot of a computer screen

Description automatically generatedA screenshot of a computer

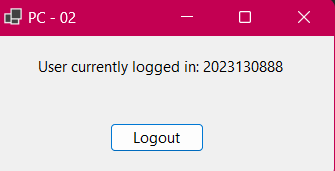
Description automatically generated**

1. **If the Student ID entered doesn’t exist**

**A screenshot of a computer screen

Description automatically generated**

1. **If the account is validated, the user has now logged into the client PC, directed to the client dashboard and is now connected to the SQL ClientSession database.**

****

1. **Logout button to logout and record the logout time of the ClientSession**

**A screenshot of a computer

Description automatically generated**

**Server client:**

1. **ServerForm dashboard automatically opens:**

A screenshot of a computer

Description automatically generated

1. **ClientPCDetails1 shows the ClientPC1 current user and the history of logins/logouts**

**A screenshot of a computer

Description automatically generated**

1. **ClientPC2 shows the ClientPC2 current user and the history of logins/logouts**

**A screenshot of a computer

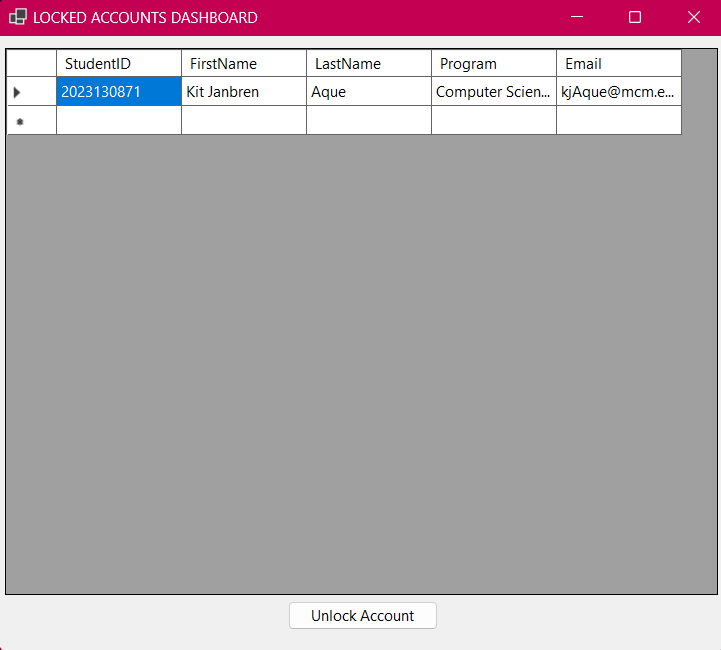
Description automatically generated**

1. **CreateAccount.cs form**

A screenshot of a computer

Description automatically generated

1. **LockedAccount.cs form**



**8. Members Curriculum Vitae**

8.1 Kit Janbren M. Aque

A person in a suit smiling

Description automatically generated

8.2 Education:

2023 – 2026 Bachelor of Science in Computer Science

Mapua Malayan Colleges of Mindanao

McArthur Highway, Matina, Davao City

8.3 Skills:

*Technical Skills*

My knowledge of programming languages such as Python, Java, and C# allow me to perform and write code in various languages and platforms. I leverage this knowledge to become a flexible team member in a development environment.

*Intrapersonal Skills*

I thrive in a collaborative environment. I prioritize communication with my colleagues, and I value active listening. I prefer a laidback working relationship ensuring that my colleagues are comfortable with each other while performing various company tasks.

8.4 Role:

Main C# Developer