



KIU
Faculty Of Computer Science
Bachelor Of (Honors) in Software Engineering

Continuous Assessment 4 (CA4) - Visual Programming

Module Code: SEN3306

Module Title: Visual Programming

Student Name: M.N.F. Nusra

Student ID: 11554

Submission Date: 14.05.2025

Part A: Testing Report

Test Case Table

Test Case ID	Test Description	Expected Output	Actual Output	Status (Pass/Fail)
TC001	Add valid student Details (Name, Age, ID)	Added Successfully	Student Added Successfully	Pass
TC002	Enter invalid student age (negative number: -5)	Display Error message: "Invalid Age"	Display Error message: "Invalid Age"	Pass
TC003	Leave "Student Name" field blank and try to add student	Display Error message: "Name cannot be empty"	Display Error message: "Name cannot be empty"	Pass
TC004	Update an existing student's name	Display Student details updated successfully Message	Display Student details updated successfully Message	Pass
TC005	Add 1000 students to test performance	Application responds within 3 seconds	Application responds within 2.5 seconds	Pass
TC006	Delete an existing student entry	Display Student deleted successfully Message	Display Student deleted successfully Message	Pass
TC007	Try deleting a student that does not exist	Display Error message: "Student not found"	Display Error message: "Student not found"	Pass

Part B: Documentation

User Manual

1. Application Overview

The Student Management System is a desktop-based application developed in C# and Windows Forms using the .NET Framework. It is designed to perform simple student record operations such as add, update, and delete student records with validation and feedback to the user. The system supports bulk entries to test the performance and includes error handling for missing or invalid data.

2. Installation Instructions

1. Download and extract the application folder.
2. Open in Visual Studio.
3. Build Solutions
4. Click Start to run the application.

3. Features and Functionalities

Features	Functionalities
Add Student Details	Add New Student Details.
View Students Details	Display All Students Records.
Update Students Details	Edit Existing Student Information.
Delete Students Details	Delete a Student Record.
Search Student	Find Student by ID
Show Error Messaging	Display When Enter the Students Details Incorrect.

4. Step-by-Step Guide on Using the Application

1. Add Student Details: Enter Student ID, Name, Age → Click Add.
2. View Students: All records shown in the DataGridView.
3. Update Student: Select a student from the list → Edit details → Click Update.
4. Delete Student: Select a student → Click Delete.
5. Search Student: Enter Student ID in the search box → Click Search.

5. Troubleshooting Tips

Issue	Solution
App not opening	Check if .NET Framework is installed
Cannot add student	Ensure all fields are filled correctly
Slow performance	Avoid entering large datasets at once
Update/Delete not working	Ensure a student record is selected

Technical Documentation

1. System Requirements

- Hardware Requirements
Processor: Intel i3
RAM: Minimum 4GB
Hard Disk: Minimum 100MB of Free Space
- Software Requirements
Software: Visual Studio 2019 or 2022
OS: Windows 8,10 or Later
.NET Framework: Version 4.7.2 or Later
Language: C#

2. Software Architecture Overview

Presentation Layer: Windows Forms (UI)

Logic Layer: Event Handlers (CRUD operations)

Data Storage: In-memory List Collection

3. Explanation of Key Components

Forms: Main Form (MainForm.cs) with DataGridView for displaying students.

Controls: TextBoxes (for input), Buttons (Add, Update, Delete), Labels, and DataGridView.

Event Handling: Button Click events trigger methods like AddStudent(), UpdateStudent(), and DeleteStudent ().

4. Code Snippets

Update Student Functionality:

```
private void btnUpdate_Click(object sender, EventArgs e)
{
    string id = txtID.Text;

    var student = studentList.FirstOrDefault(s => s.ID == id);

    if (student == null)
    {
        MessageBox.Show("Student not found");

        return;
    }
}
```

```
}

    student.Name = txtName.Text;
}

int age;

if (int.TryParse(txtAge.Text, out age) && age >= 0)
    student.Age = age;

RefreshStudentList();

MessageBox.Show("Student details updated successfully");
}
```