Lab Assignment 12

Course: CS202 Software Tools and Techniques for CSE
Lab Topic: Event-driven Programming for Windows Forms Apps. in C#
Date: 17th April 2025

Objective

This lab introduces students to event-driven programming in **C# Windows Forms Apps.** using **Visual Studio**. The focus is on understanding the event-driven paradigm and understanding how control flows in response to occurrence of events generated by user or by a particular state of the application under development.

Learning Outcomes

By the end of this lab, students will be able to:

- Develop and analyze Windows Forms Apps. using C# as the development language.
- Understand the benefit of the Visual Studio Toolbox.
- Why/how/when control flow through the application due to user interaction or triggered by a certain state of the application during runtime.

Pre-Lab Requirements

- Operating System: Windows
- Software: Visual Studio 2022 (Community Edition), Visual Studio with .NET SDK
- Programming Language: C# (latest stable version)

Lab Activities:

1. Write a C# console application to accept time from user in HH:MM:SS format and keep on checking it with current system time. As soon as the user supplied time and current system time becomes same, raise a user-defined event raiseAlarm which will call the function Ring_alarm() to display appropriate message. Follow the publisher/subscriber model. Note that all input/output is through the console (terminal) only.
2. Modify the C# console app. (in the previous task) to perform all input/output activities only through one Windows Form[1]. The user should enter the time (valid[2] input in HH:MM:SS format) in a textbox and then click on the start button. Thereafter, every second, the background color of the Form should change. On reaching the target time, the color should stop changing, and the appropriate message should be displayed through a message box. There should be no usage of the console for this task.

Note: Proper screenshots of both C# Code as well as the Graphical User Interface (GUI) should be able to illustrate the end-to-end event-driven workflow for both tasks.

Resources

- Lecture 11 Slides
- https://learn.microsoft.com/en-us/dotnet/csharp

[1] Recall what a Form is for Windows Form based C# applications.
[2] Validation is left as an exercise.

- https://en.wikipedia.org/wiki/Event-driven_programming
- https://quix.io/blog/what-why-how-of-event-driven-programming