Department of
Computer Science and Engineering

Title: Graphical User Interface design for the given project

Integrated Design Project II

CSE 406

Green University of Bangladesh

1 Objective(s)

- To define the use of Graphical User Interface of a system.
- To design component and elements for a specific project.
- To find out Graphical User Interface design activities for the given project

2 Problem analysis

User Interface Design is one of the reasons that your website will start to see an influx in traffic. It draws people in and keeps them there. It is what makes people recommend your site and become loyal customers. Regardless of what anyone says, it should never be overlooked. One thing that takes people by surprise is how big of an impact even the smallest adjustments in UI Design can have. For instance, did you know that the shape of a button could determine whether or not someone knows how to accomplish a task? It is a crazy phenomenon to think about, but it's true. People have certain instincts based on visual details that play into how they interact with a website and web and software development companies need to take these into consideration. Design speaks to people and should be taken seriously if you want to be successful

In addition to basic design changes like shapes of buttons and color schemes, chances are people won't stay on your site if it's difficult to interact with. When people visit your website, their user experience should be one of your top priorities. When people have a good experience on your site, the conversion rates are higher and they tend to tell more people about it. This means more chances to get your sales up and even higher opportunities to grow your customer base. When people have a bad experience on your site, the chances of your product or company being rejected increases exponentially. This is especially true if you rely on Internet marketing to get in touch with most of your target audience. Simply put, UI Design is important because without it you'd probably have to resort to printing flyers and making cold calls. People make snap judgments and when they visit your site, they'd rather spend 30 seconds opening a new site than meddling around on a difficult one. If your customer base finds your website too complicated, confusing or difficult to use/navigate, an otherwise extraordinary product can easily fail1. People want to be able to understand things easily, which is why most physical products can be figured out without digging through a 1,000-page manual. Making your site's UI Design understandable without a manual is the difference between complete success and a potentially discouraging failure. Simply put, User Interface Design is important because it can make or break your customer base. It creates fewer problems, increases user involvement, perfects functionality and creates a strong link between your customers and your website.

3 Graphical User Interface

User interface is the front-end application view to which user interacts in order to use the software. User can manipulate and control the software as well as hardware by means of user interface. Today, user interface is found at almost every place where digital technology exists, right from computers, mobile phones, cars, music players, airplanes, ships etc.

User interface is part of software and is designed such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human-computer interaction.

UI can be graphical, text-based, audio-video based, depending upon the underlying hardware and software combination. UI can be hardware or software or a combination of both.

The software becomes more popular if its user interface is:

- Attractive
- Simple to use
- Responsive in short time
- Clear to understand
- Consistent on all interfacing screens

3.1 How Does a Graphical User Interface Work?

Graphical user interface design principles conform to the model–view–controller software pattern, which separates internal representations of information from the manner in which information is presented to the user, resulting in a platform where users are shown which functions are possible rather than requiring the input of command codes. Users interact with information by manipulating visual widgets, which are designed to respond in accordance with the type of data they hold and support the actions necessary to complete the user's task.

The appearance, or "skin," of an operating system or application software may be redesigned at will due to the nature of graphical user interfaces being independent from application functions. Applications typically implement their own unique graphical user interface display elements in addition to graphical user interface elements already present on the existing operating system. A typical graphical user interface also includes standard formats for representing graphics and text, making it possible to share data between applications running under common graphical user interface design software.

Graphical user interface testing refers to the systematic process of generating test cases in order to evaluate the functionality of the system and its design elements. Graphical user interface testing tools, which are either manual or automated and typically implemented by third-party operators, are available under a variety of licenses and are supported by a variety of platforms. Popular examples include: Tricentis Tosca, Squish GUI Tester, Unified Functional Testing (UFT), Maveryx, Appium, and eggPlant Functional.

3.2 Graphical User Interface Examples

Sketchpad, believed to be the first graphical computer-aided design program, was developed in 1962 by Ivan Sutherland while he was at MIT, and consisted of a light pen that enabled users to create and manipulate objects in engineering drawings in real time with coordinated graphics.

Modern operating systems and graphical user interfaces are incorporated into nearly every interactive application, such as ATMs, self-service checkouts, airline self-ticketing and check-in, video games, smartphones, and desktops. Some popular, modern graphical user interface examples include Microsoft Windows, macOS, Ubuntu Unity, and GNOME Shell for desktop environments, and Android, Apple's iOS, BlackBerry OS, Windows 10 Mobile, Palm OS-WebOS, and Firefox OS for smartphones.

4 GUI Elements

GUI provides a set of components to interact with software or hardware.

Every graphical component provides a way to work with the system. A GUI system has following elements such as:

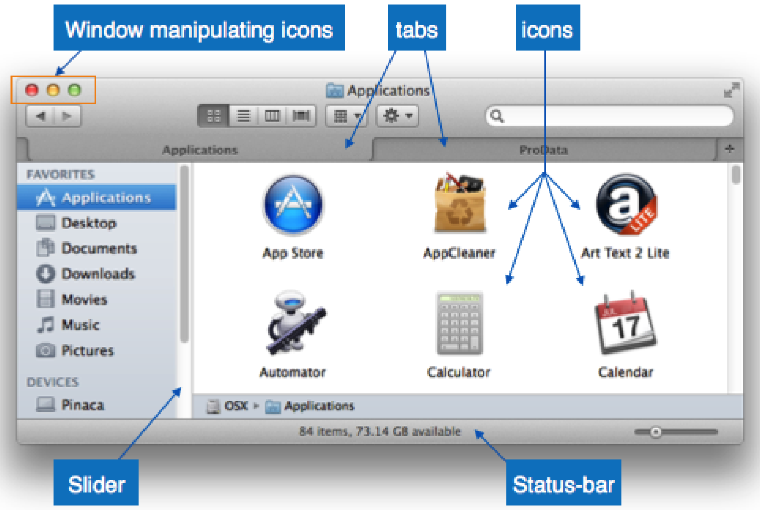


Figure 1: GUI Elements

Window - An area where contents of application are displayed. Contents in a window can be displayed in the form of icons or lists, if the window represents file structure. It is easier for a user to navigate in the file system in an exploring window. Windows can be minimized, resized or maximized to the size of screen. They can be moved anywhere on the screen. A window may contain another window of the same application, called child window.

Tabs - If an application allows executing multiple instances of itself, they appear on the screen as separate windows. Tabbed Document Interface has come up to open multiple documents in the same window. This interface also helps in viewing preference panel in application. All modern web-browsers use this feature.

Menu - Menu is an array of standard commands, grouped together and placed at a visible place (usually top) inside the application window. The menu can be programmed to appear or hide on mouse clicks.

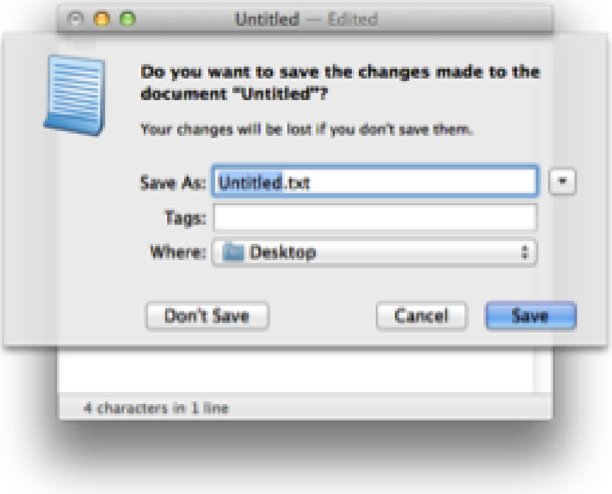
Icon - An icon is small picture representing an associated application. When these icons are clicked or double clicked, the application window is opened. Icon displays application and programs installed on a system in the form of small pictures.

Cursor - Interacting devices such as mouse, touch pad, digital pen are represented in GUI as cursors. On screen cursor follows the instructions from hardware in almost real-time. Cursors are also named pointers in GUI systems. They are used to select menus, windows and other application features.

4.1 Application specific GUI components

A GUI of an application contains one or more of the listed GUI elements:

- **Application Window -** Most application windows uses the constructs supplied by operating systems but many use their own customer created windows to contain the contents of application.
- **Dialogue Box -** It is a child window that contains message for the user and request for some action to be taken. For Example: Application generate a dialogue to get confirmation from user to delete a file.



Untitled — Edited

Do you want to save the changes made to the document "Untitled"?

Your changes will be lost if you don't save them.

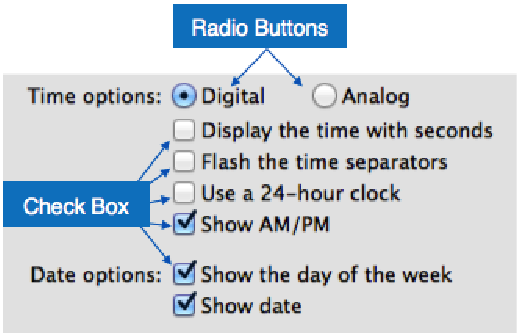
Save As: Untitled.txt

Tags:

Where: Desktop

4 characters in 1 line

- **Text-Box -** Provides an area for user to type and enter text-based data.
- **Buttons -** They imitate real life buttons and are used to submit inputs to the software.



Time options: ● Digital ○ Analog

□ Display the time with seconds

□ Flash the time separators

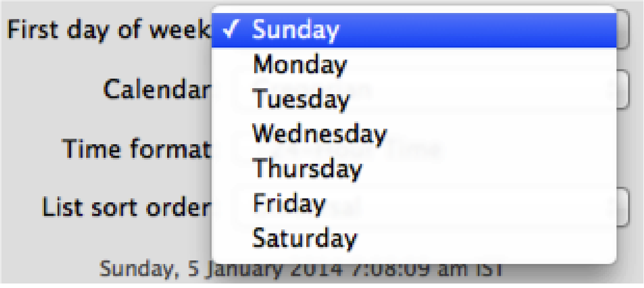
□ Use a 24-hour clock

☑ Show AM/PM

Date options: ☑ Show the day of the week

☑ Show date

- **Radio-button -** Displays available options for selection. Only one can be selected among all offered.
- **Check-box -** Functions similar to list-box. When an option is selected, the box is marked as checked. Multiple options represented by check boxes can be selected.
- **List-box -** Provides list of available items for selection. More than one item can be selected



First day of week ✓ Sunday

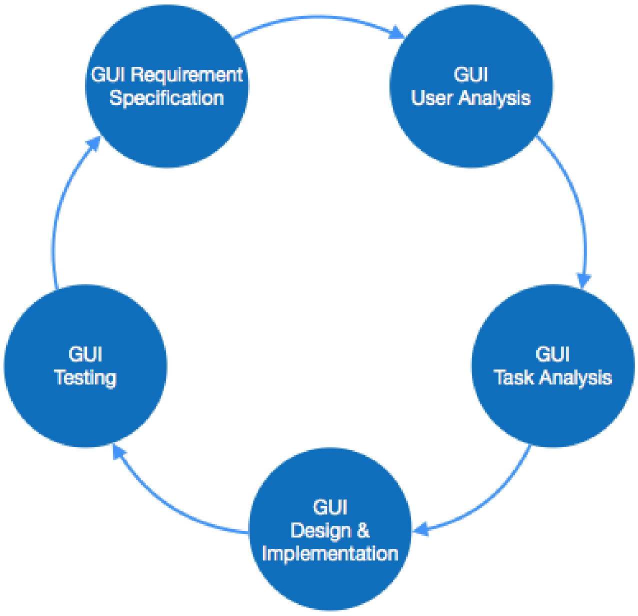
Other impressive GUI components are:

- Sliders
- Combo-box
- Data-grid
- Drop-down list

5 User Interface Design Activities

There are a number of activities performed for designing user interface. The process of GUI design and implementation is alike SDLC. Any model can be used for GUI implementation among Waterfall, Iterative or Spiral Model.

A model used for GUI design and development should fulfill these GUI specific steps.



GUI Requirement Gathering - The designers may like to have list of all functional and non-functional requirements of GUI. This can be taken from user and their existing software solution.

User Analysis - The designer studies who is going to use the software GUI. The target audience matters as the design details change according to the knowledge and competency level of the user. If user is technical savvy, advanced and complex GUI can be incorporated. For a novice user, more information is included on how-to of software.

Task Analysis - Designers have to analyze what task is to be done by the software solution. Here in GUI, it does not matter how it will be done. Tasks can be represented in hierarchical manner taking one major task and dividing it further into smaller sub-tasks. Tasks provide goals for GUI presentation. Flow of information among sub-tasks determines the flow of GUI contents in the software.

GUI Design implementation - Designers after having information about requirements, tasks and user environment, design the GUI and implements into code and embed the GUI with working or dummy software in the background. It is then self-tested by the developers.

Testing - GUI testing can be done in various ways. Organization can have in-house inspection, direct involvement of users and release of beta version are few of them. Testing may include usability, compatibility, user acceptance etc.

6 GUI Implementation Tools

There are several tools available using which the designers can create entire GUI on a mouse click. Some tools can be embedded into the software environment (IDE).

GUI implementation tools provide powerful array of GUI controls. For software customization, designers can change the code accordingly.

There are different segments of GUI tools according to their different use and platform.

Example Mobile GUI, Computer GUI, Touch-Screen GUI etc. Here is a list of few tools which come handy to build GUI:

- FLUID
- AppInventor (Android)
- LucidChart
- Wavemaker
- Visual Studio

7 User Interface Golden rules

The following rules are mentioned to be the golden rules for GUI design, described by Shneiderman and Plaisant in their book (Designing the User Interface).

Strive for consistency - Consistent sequences of actions should be required in similar situations. Identical terminology should be used in prompts, menus, and help screens. Consistent commands should be employed throughout.

Enable frequent users to use short-cuts - The user's desire to reduce the number of interactions increases with the frequency of use. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.

Offer informative feedback - For every operator action, there should be some system feedback. For frequent and minor actions, the response must be modest, while for infrequent and major actions, the response must be more substantial.

Design dialog to yield closure - Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and this indicates that the way ahead is clear to prepare for the next group of actions.

Offer simple error handling - As much as possible, design the system so the user will not make a serious error. If an error is made, the system should be able to detect it and offer simple, comprehensible mechanisms for handling the error.

Permit easy reversal of actions - This feature relieves anxiety, since the user knows that errors can be undone. Easy reversal of actions encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.

Support internal locus of control - Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.

Reduce short-term memory load - The limitation of human information processing in short-term memory requires the displays to be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

8 Implementation

In this tutorial, you'll create a Visual Basic application that has a Windows Forms user interface. The Visual Studio integrated development environment (IDE) includes all the tools you need to create a Windows Forms app.

In this tutorial, you learn how to:

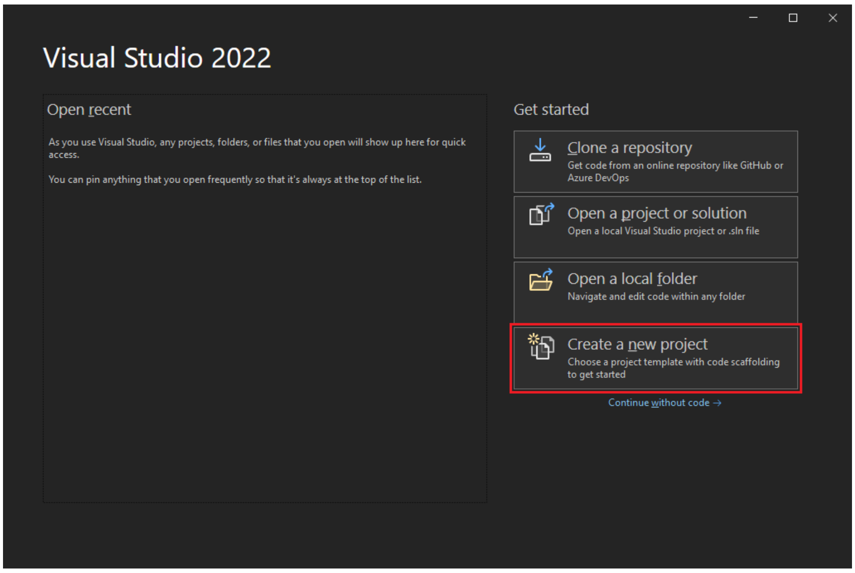
- Create a project
- Add a button to the form
- Add a label and code
- Run the application

Create a project

Create a Visual Basic application project. The project type comes with all the template files you'll need, before you've even added anything.

1. Open Visual Studio.

2. On the start window, choose Create a new project.



Visual Studio 2022

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

Get started

Clone a repository
Get code from an online repository like GitHub or Azure DevOps

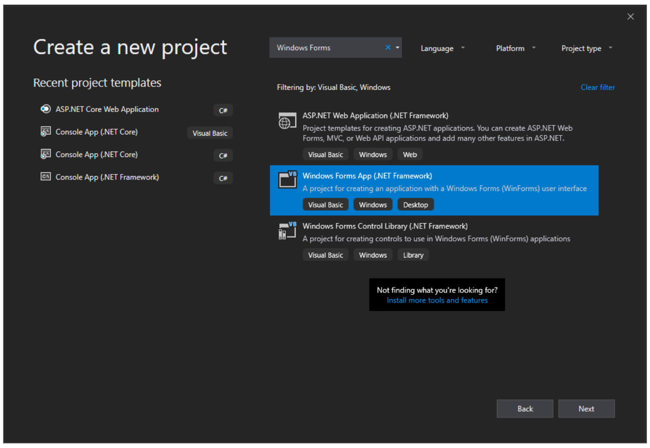
Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

3. On the Create a new project window, select the Windows Forms App (.NET Framework) template for Visual Basic.

You can refine your search to quickly get to the template you want. For example, enter Windows Forms App in the search box. Next, select Visual Basic from the Language list, and then Windows from the Platform list.



Create a new project

Windows Forms

Recent project templates

ASP.NET Core Web Application	C#
Console App (.NET Core)	Visual Basic
Console App (.NET Core)	C#
Console App (.NET Framework)	C#

Filtering by: Visual Basic, Windows

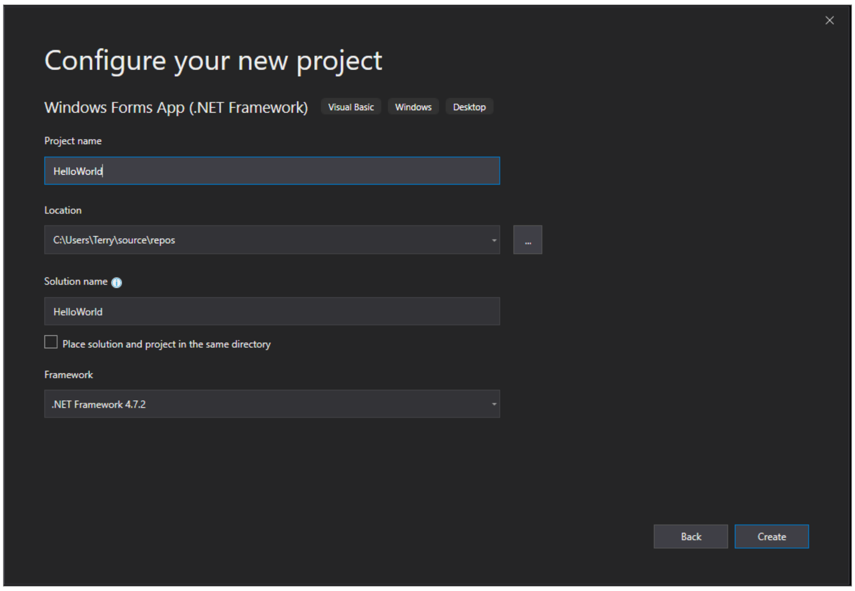
ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
Visual Basic · Windows · Web

Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface
Visual Basic · Windows · Desktop

Windows Forms Control Library (.NET Framework)
A project for creating controls to use in Windows Forms (WinForms) applications
Visual Basic · Windows · Library

Not finding what you're looking for?
Install more tools and features

4. In the Configure your new project window, enter HelloWorld as the Project name. Then, select Create.



Configure your new project

Windows Forms App (.NET Framework) Visual Basic Windows Desktop

Project name

HelloWorld

Location

C:\Users\Terry\source\repos

Solution name

HelloWorld

□ Place solution and project in the same directory

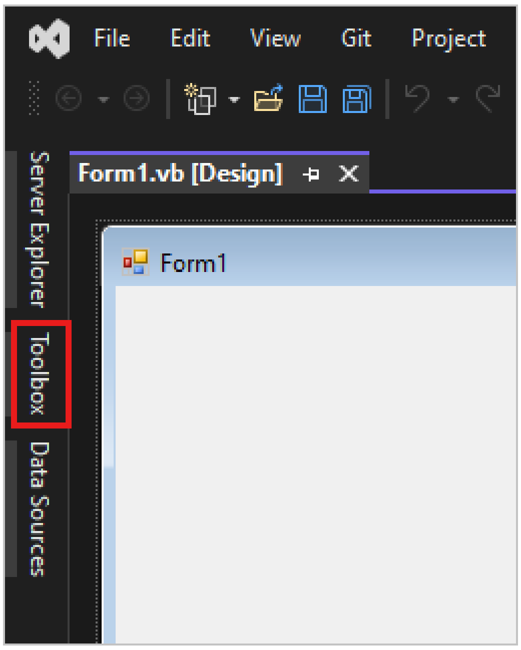
Framework

.NET Framework 4.7.2

Add a button to the form

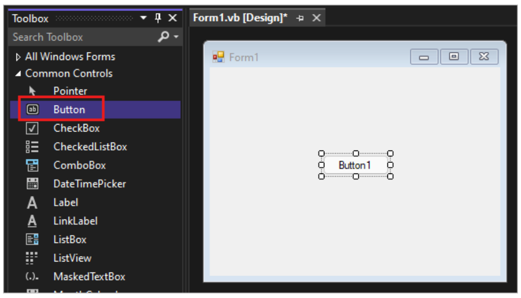
After you select your Visual Basic project template and name your file, Visual Studio opens a form for you. A form is a Windows user interface. You'll create a "Hello World" application by adding controls to the form.

1. On the left side of the Visual Studio IDE, select the Toolbox tab. If you don't see it, select View > Toolbox from the menu bar or Ctrl+Alt+X.

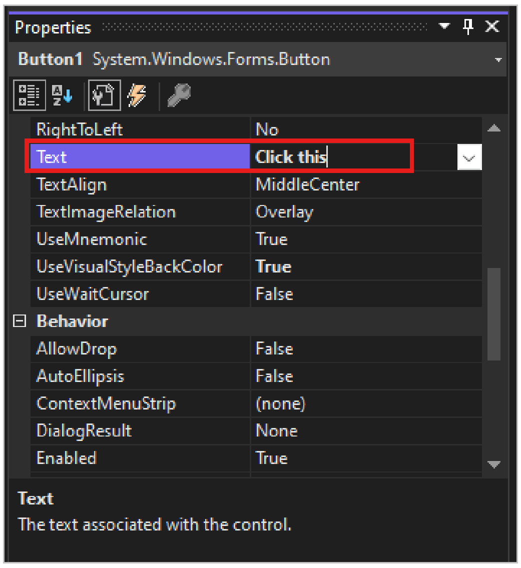


If you want, select the Pin icon to dock the Toolbox window.

2. Select the Button control and then drag it onto the form.



3. In the Appearance section of the Properties window, for Text, type Click this, and then press Enter.



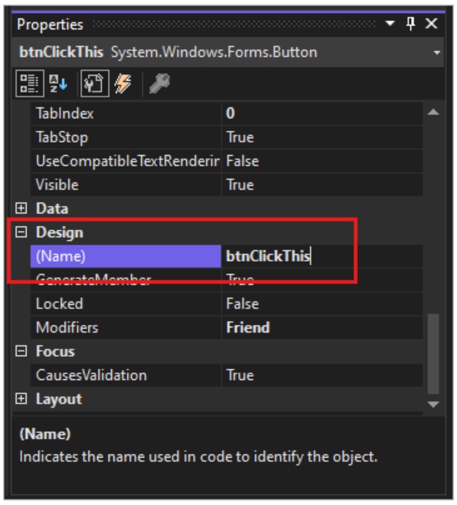
Button1 System.Windows.Forms.Button

RightToLeft	No
Text	Click this
TextAlign	MiddleCenter
TextImageRelation	Overlay
UseMnemonic	True
UseVisualStyleBackColor	True
UseWaitCursor	False
Behavior	
AllowDrop	False
AutoEllipsis	False
ContextMenuStrip	(none)
DialogResult	None
Enabled	True

Text
The text associated with the control.

If you don't see the Properties window, you can open it from the menu bar. Select View > Properties Window or press F4.

4. In the Design section of the Properties window, change the name from Button1 to btnClickThis, and then press Enter.



btnClickThis System.Windows.Forms.Button

TabIndex	0
TabStop	True
UseCompatibleTextRenderir	False
Visible	True
Data	
Design	
(Name)	btnClickThis
GenerateMember	True
Locked	False
Modifiers	Friend
Focus	
CausesValidation	True
Layout	

(Name)
Indicates the name used in code to identify the object.

Add a label and code

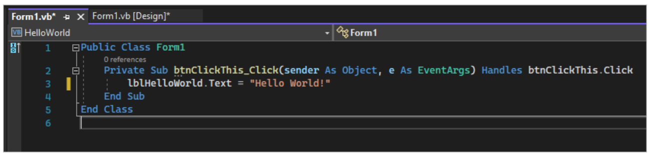
Now that you've added a button control to create an action, add a label control to send text to.

1. Select the Label control in the Toolbox window, and then drag it onto the form. Place it beneath the Click this button.

2. In either the Design section or the (DataBindings) section of the Properties window, change the name Label1 to lblHelloWorld, and then press Enter.

3. In the Form1.vb [Design] window, double-click the Click this button to open the Form1.vb window. Another option is to expand Form1.vb in Solution Explorer, and then select Form1.

4. In the Form1.vb window, between the Private Sub and End Sub lines, enter lblHelloWorld.Text = "Hello World!" as shown in the following screenshot:

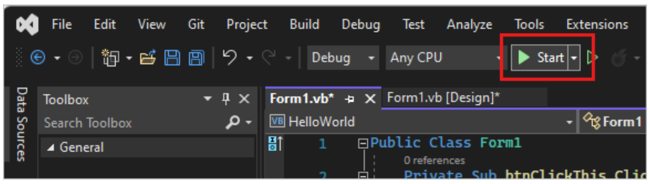


```
Public Class Form1
    Private Sub btnClickThis_Click(sender As Object, e As EventArgs) Handles btnClickThis.Click
        lblHelloWorld.Text = "Hello World!"
    End Sub
End Class
```

Run the application

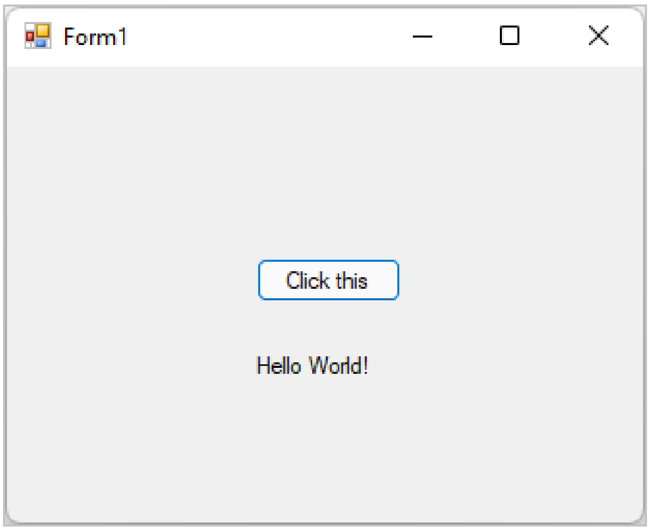
Your application is ready to build and run.

1. Select Start to run the application.



Several things happen. In the Visual Studio IDE, the Diagnostics Tools window opens, and an Output window opens. Outside of the IDE, a Form1 dialog box appears. It includes your Click this button and text that says Label1.

2. Select the Click this button in the Form1 dialog box.



The Label1 text changes to Hello World!.

3. Close the Form1 dialog box to stop running the app.

9 Discussion & Conclusion

After reading this tutorial on C++ GUI, you would have understood how to create a project and configure visual studio to run C++ GUI application and the Creation of windows form application. You will also learn how to create a C++ GUI application.

10 Lab Task (Please implement yourself and show the output to the instructor)

1. Draw a GUI interface of the given project ‘Library Management System’.
2. Find out the User interface design activities of the given project ‘Library Management System’.

10.1 Problem analysis

A library database system is an infrastructure that allows users to search books and book content, add/remove, and download selected books. The problem faced is that library users require an efficient method to find a specific book or keyword(s) within a book given a continuously expanding library. Some scenarios of the 'Library Management System' are as follows:

1. User who registers himself as a new user initially is regarded as staff or student for the library system.

(i) For the user to get registered as a new user, registration forms are available that is needed to be fulfilled by the user.

(ii) After registration, a library card is issued to the user by the librarian. On the library card, an ID is assigned to cardholder or user.

2. After getting the library card, a new book is requested by the user as per there requirement.

3. After, requesting, the desired book or the requested book is reserved by the user that means no other user can request for that book.

4. Now, the user can renew a book that means the user can get a new due date for the desired book if the user has renewed them.

5. If the user somehow forgets to return the book before the due date, then the user pays fine. Or if the user forgets to renew the book till the due date, then the book will be overdue and the user pays fine.

6. User can fill the feedback form available if they want to.

7. Librarian has a key role in this system. Librarian adds the records in the library database about each student or user every time issuing the book or returning the book, or paying fine.

8. Librarian also deletes the record of a particular student if the student leaves the college or passed out from the college. If the book no longer exists in the library, then the record of the particular book is also deleted.

9. Updating database is the important role of Librarian.

11 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.