CSE446 Software Integration and Engineering

Spring 2019

Instructor: Dr. Yinong Chen

Project 1 (100 Points)

Assignment 1 Due: Saturday, January 26, 2019 by 11:59pm

Assignment 2 Due: Saturday, February 2, 2019 by 11:59pm

Plus one day grace period for each part

Introduction

The aim of this project is to make sure that you have read the lecture slides and text, and you have understood the concepts covered in the lectures and in the text, including basic WCF services, advanced WCF services, and RESTful services. By the end of this project, you should have a good understanding of these concepts and have applied these concepts in developing operational programs.

Preparation and Practice Exercises (No submission required)

1. Read Textbook Chapter 3 if you have not taken the course CSE445 Distributed Software Development.
2. Read Textbook Chapter 7 to prepare for this project.
3. Download Visual Studio 2017 Professional/Enterprise Edition, in MyASU → MyApps. It is fine if you have VS 2015. You may also use the computer in CIDSE lab in BYENG 114, where the required software is installed. Note, when you install VS, choose ALL the options to install.
4. Get familiar with the service hosting. Follow textbook chapter 3 to develop the basic-three service. You can choose to use IIS Express to host the service, which will provide a localhost port number. To test if your project works on a different computer, send the zipped project file to another computer, unzip the file and test the project on that machine. Service oriented software is distributed. It may have saved different parts in different locations. You need to make sure that you submit all the parts required.
5. Practice exercises: Questions in Textbook Chapter 7, Section 7.6 multiple choice questions. You can find the solution key in the course web page. These questions help you to prepare the quizzes and the multiple choice questions in the exam.
6. Follow Text Section 7.1.1 and repeat the development process of the self-hosting service given in the section. If the Visual Studio Version used in the text is different from the version you use, try to find the differences and complete the process. You may find the updated instruction from MSDN library online.
7. Follow Text Section 7.1.2 and repeat the development process of the console application that invokes the service developed in Section 7.1.1.

8. Follow Text Sections 7.3.3 and 7.3.4, and repeat the development process of the RESTful services
9. Follow Text Section 7.3.5 and repeat the development process of the clients that consume the RESTful services.

Assignment 1 WCF Services and Application (Submission Required)

In this part of the project, you will implement the number guessing services and number guessing games using different techniques. You are given the class definition below as the basic functions required in the number guessing services.

```
public class NumberGuess {
   public int SecretNumber(int lower, int upper) {
          DateTime currentDate = DateTime.Now;
          int seed = (int)currentDate.Ticks;
          Random random = new Random(seed);
          int sNumber = random.Next(lower, upper);
          return sNumber;
   }
     public string checkNumber(int userNum, int SecretNum) {
               if (userNum == SecretNum)
                    return "correct";
               else
                    if (userNum > SecretNum)
                         return "too big";
                    else return "too small";
     }
}
```

1. Based on the above given class, write a WCF .svc service with two operations on localhost of .Net IIS Express server. You may want to review Text Section 3.2 before completing this assignment. [10 points]

2. Create an ASP .Net Windows Forms Application to consume the .svc service in question 1. VS 2017 project option is shown in Figure 1.

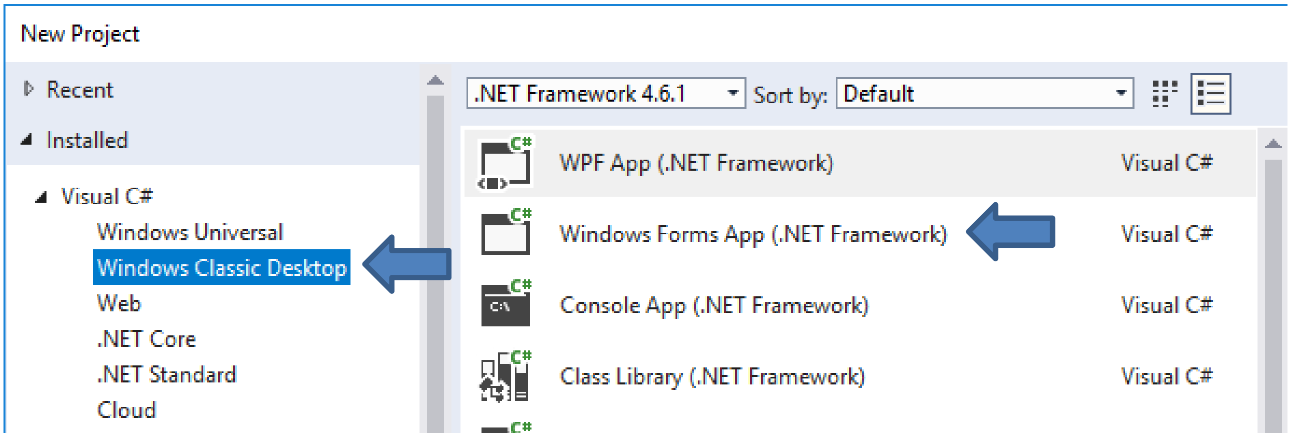

New Project
Recent
Installed
Visual C#
Windows Universal
Windows Classic Desktop
Web
.NET Core
.NET Standard
Cloud
.NET Framework 4.6.1
Sort by: Default
WPF App (.NET Framework) Visual C#
Windows Forms App (.NET Framework) Visual C#
Console App (.NET Framework) Visual C#
Class Library (.NET Framework) Visual C#

Figure 1. Creating a Windows Forms App project in VS 2017

You can host the service and the application on localhost or IIS Express. The application GUI must contain at least the following items, as shown in Figure 2. [15 points]

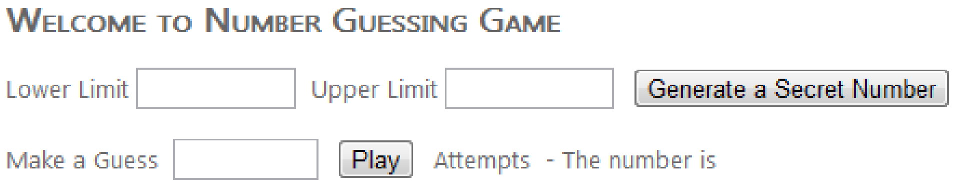


Figure 2. Number Guessing Game Basic GUI Design

- The first two textboxes are used for a player to enter the lower and upper limits (integers) of the random number to be generated;
- The code behind the button “Generate a Secret Number” will generate a secret number and save it as a state;
- The last textbox allows the player to enter an integer to make a guess of the secret number.
- A label (attempts) that displays how many attempts have been made after a secret number is generated.
- A label (The number is) that displays if the given number is too small, too big, or a correct guess.

3. Create a self-hosing service to implement the NumberGuess class, with the SecretNumber and CheckNumber operations. The program must include the service and the hosting code. [20 points]

4. Write a console application to call the self-hosting service and to display the output of the service. The application must use svcutil.exe to generate the proxy. [5 points]

Assignment 1 submission requirement:

Questions 1 and 2 must be in one solution, and questions 3 and 4 must be in another solution. Each solution folder must contain all the files. Put the two solution folders into another folder. Zip the folder (a single file) for submission. Testing results (with screenshots) must be submitted in a Word document or in PDF. You must indicate in the document the Visual Studio version (2017 or 2015) and the template that you use, so that the TA can use the same tools to test your program.

Blackboard submission notice: The assignment consists of multiple distributed projects and components. They may be stored in different locations on your computer when you create them. You must copy these projects into the submission folder for blackboard submission. To make sure that you have all the files included in the zip file and they work together, you must test them on a different computer before submission. You must also download your own submission from the blackboard. Unzip the file on a different location or computer, and test your assignment and see if you can run the solution in a different location, because the TA will test your application on a different machine.

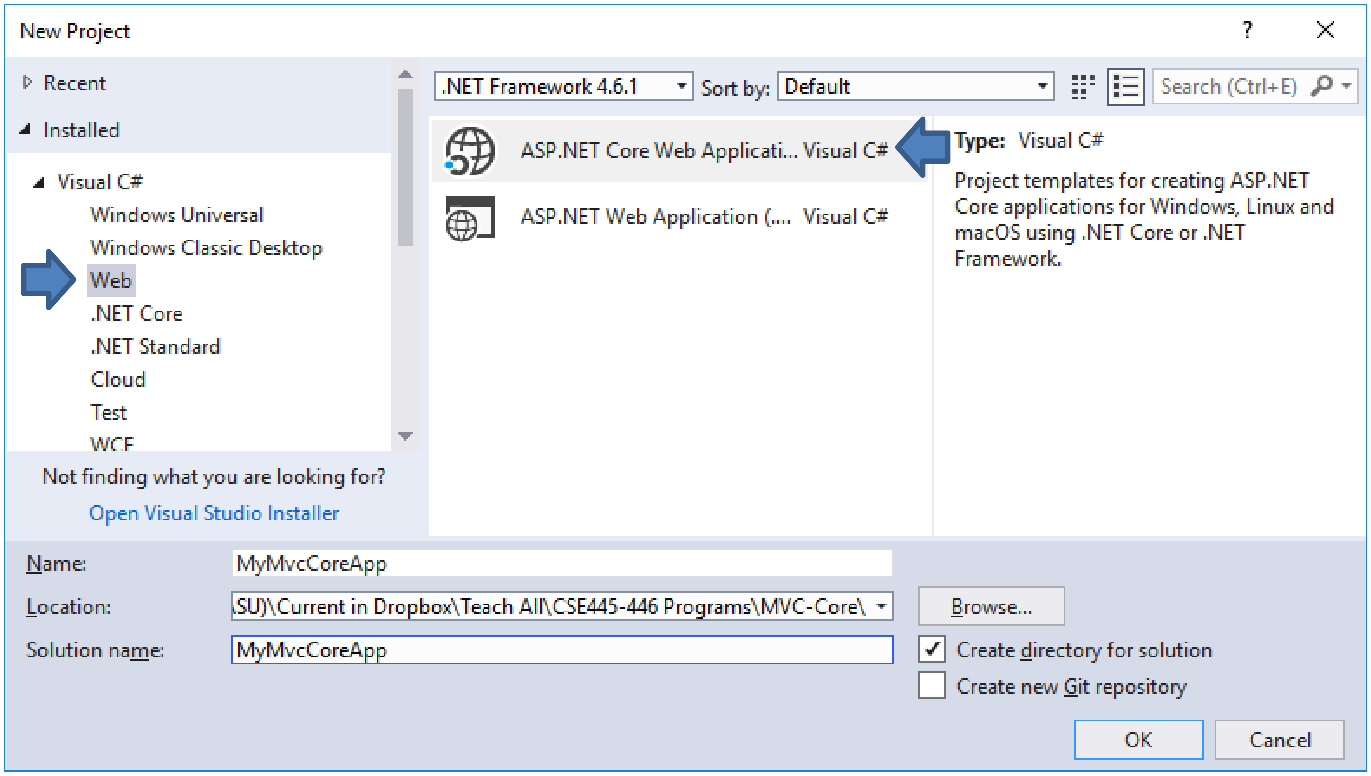
Assignment 2 Advanced Service and Application Architecture (Submission Required)

In this assignment, you will continue to work on the project that you created in assignment 1.

5. Re-implement the NumberGuess service as a RESTful service. Host the service on IIS Express server. For each service operation in the service developed in the question above, add UriTemplate in the attribute [WebGet], so that the service operation names are not exposed to the callers of the service. [20 points]

6. Choose one of the following assignment questions to implement your application architecture that calls your service. [30 points]

6.1 Create an MVC Web Application (Please read Text Section 5.8.2 and Lecture 1-6 slides) to consume the RESTful service in question 5. You must create at least one asynchronous call to access the service using BeginGetResponse and a callback function. Please read text section 7.3.7 for an example. The following diagrams in Figure 3 and Figure 4 show some of the steps of creating a MVC Web application in VS 2017. If you use a different version, you may need to use different variation of the steps following MSDM documents.



.NET Framework 4.6.1

Sort by: Default

Type: Visual C#

Project templates for creating ASP.NET Core applications for Windows, Linux and macOS using .NET Core or .NET Framework.

Name: MyMvcCoreApp

Location: SU)\Current in Dropbox\Teach All\CSE445-446 Programs\MVC-Core\

Solution name: MyMvcCoreApp

☑ Create directory for solution

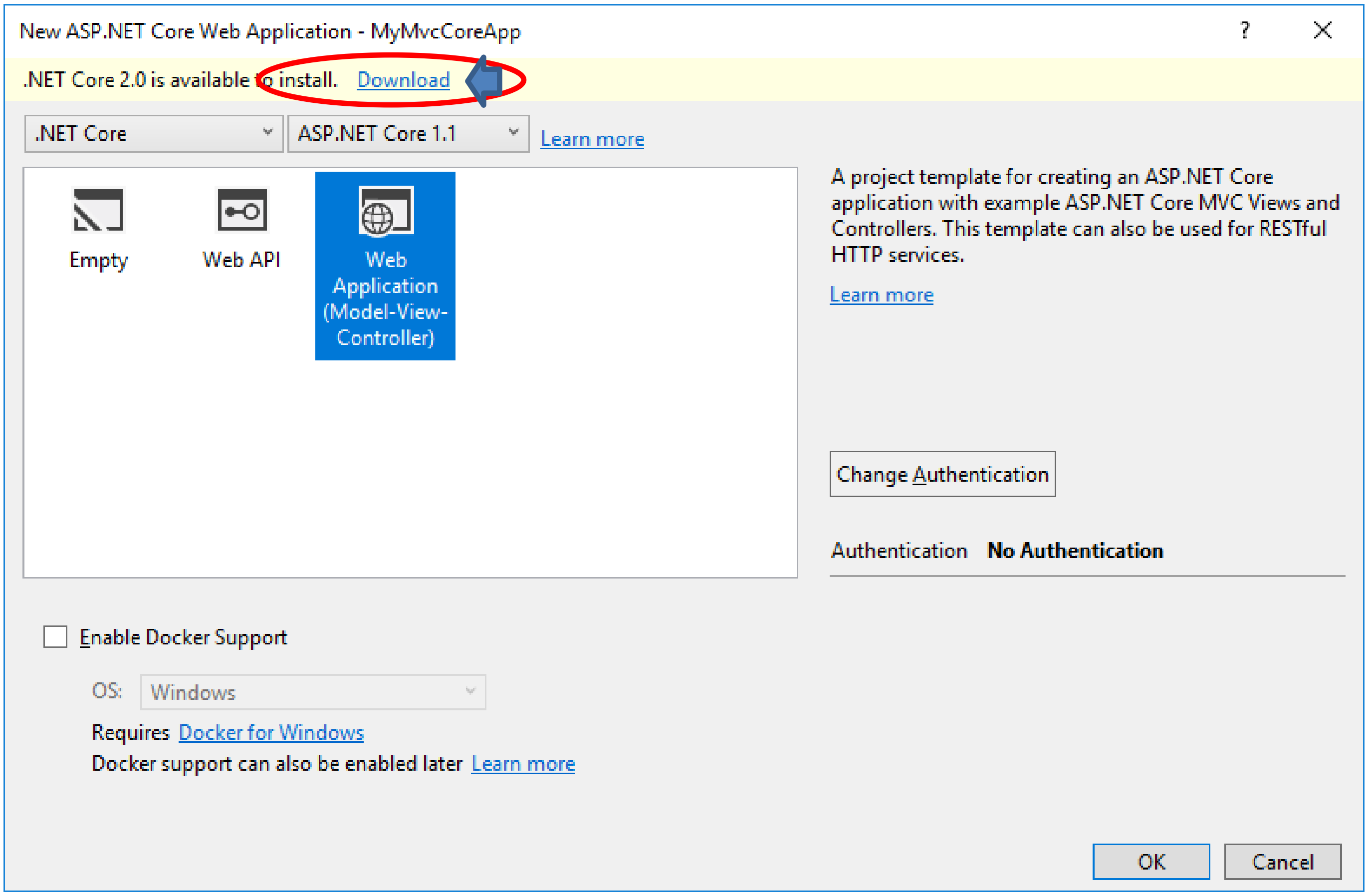
□ Create new Git repository

Figure 3. Creating a Windows Forms App project in VS 2017

You may need to download .Net Core 2.0, as shown in the following diagram.

To get started with .NET Core 2.0 in Visual Studio 2017 follow the steps

- Install .NET Core 2.0 - https://www.microsoft.com/net/core
- Restart Visual Studio 2017



New ASP.NET Core Web Application - MyMvcCoreApp

.NET Core 2.0 is available to install. Download

.NET Core | ASP.NET Core 1.1 | Learn more

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Learn more

Authentication **No Authentication**

□ Enable Docker Support

OS: Windows

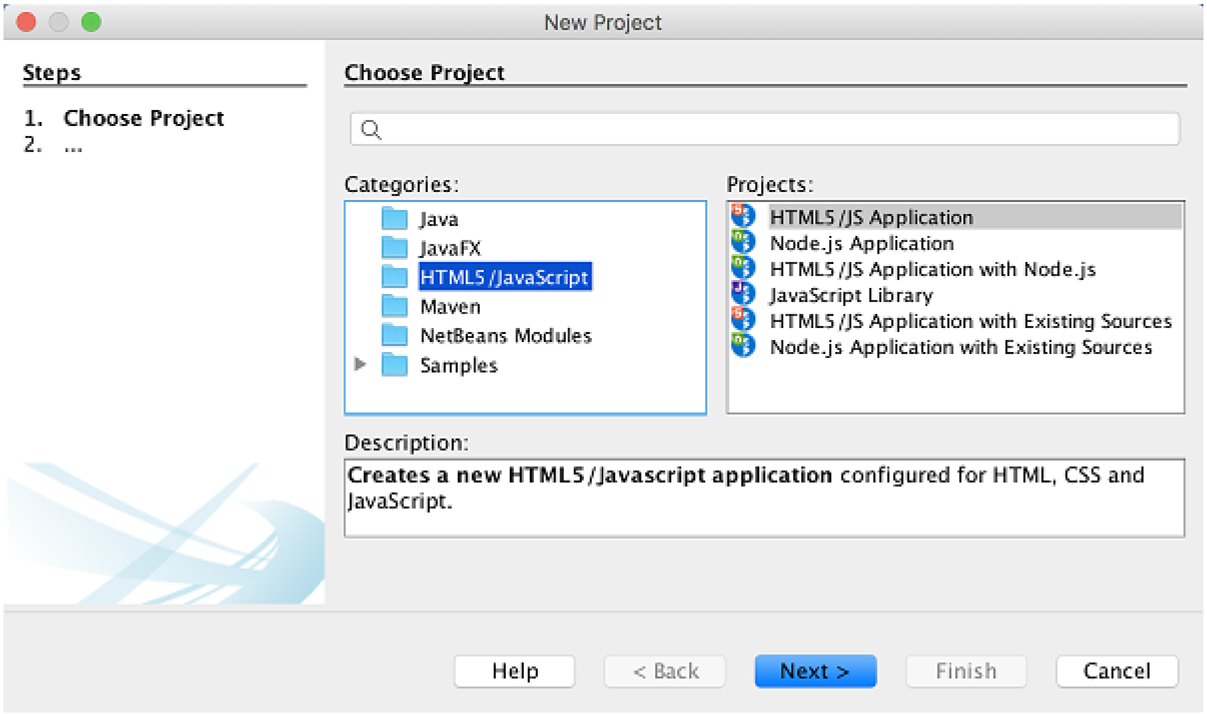
Requires Docker for Windows

Docker support can also be enabled later Learn more

You can host the application on IIS Express. The application GUI must contain at least the items that are shown in Figure 2.

6.2 Create an HTML 5 Web Application to consume the RESTful service in question 5. You are required to create graphic user interface. Creating animation is optional.
In this question, you will read additional reference materials to learn HTML 5 programming model and HTML 5 graphics libraries. There is a short introduction in Text Section 4.6.5. A sample code is given that shows the creation of graphic user interface and animation in HTML 5.

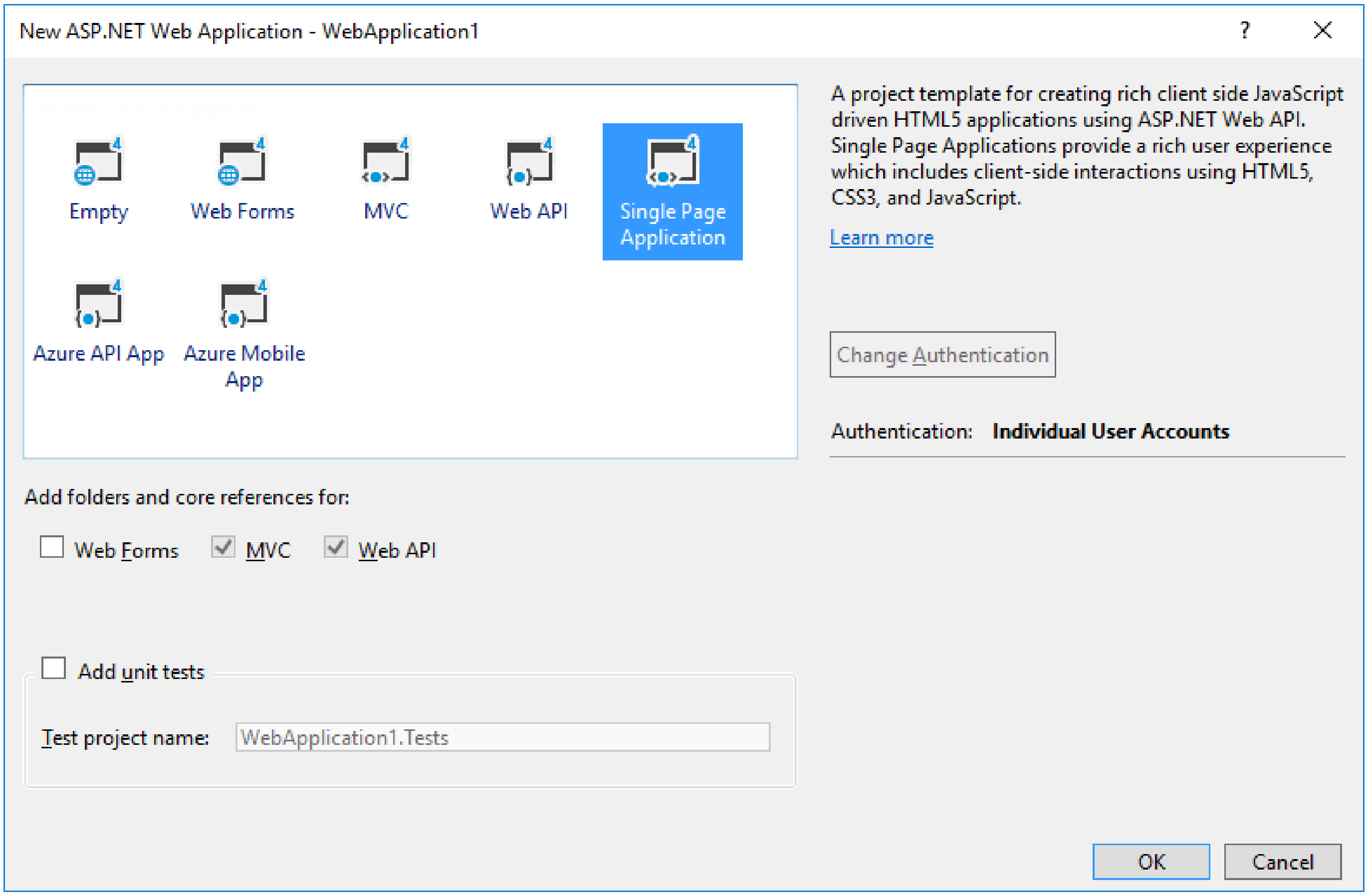
You can use different platforms to create HTML Web application. For example, you can follow the tutorial given at the link:
https://netbeans.org/kb/docs/webclient/html5-gettingstarted.html



Description:

Creates a new HTML5/Javascript application configured for HTML, CSS and JavaScript.

You can also use Visual Studio to implement this question. Choose New Project…, Web, and then Single Page Application template:



A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication: **Individual User Accounts**

Add folders and core references for:

☐ Web Forms ☑ MVC ☑ Web API

☐ Add unit tests

Test project name: WebApplication1.Tests

Assignment 2 submission requirement:

Questions 5 and 6.1 should be in one solution. If you choose question 6.2, it should be a separate project. Each solution and project must contain all the files. Put your solution and project into another folder. Zip the folder (a single file) for submission. Testing results (with screenshots) must be submitted in a Word document or in a PDF document. You must indicate in the document the Visual Studio version and the template that you use, so that the TA can use the same tools to test your program.

General Submission Requirement

Submit the solution folder containing all the files so that the solution can be tested on the TA's machine. If multiple solutions must be submitted, all the solutions must put into another "root" folder. Zip the root folder to submit a single zip file.

Submission notice: The assignment consists of multiple distributed projects. They may be stored in different locations when you create them. You must copy these projects into a single folder for blackboard submission. To make sure that you have all the files included in the zip file and they work together, download your own submission from the blackboard. Unzip the file on a different machine, and test it and see if you can run the solution in a different location because the TA will test your application on a different machine.

Late submission deduction policy:

- No penalty for late submissions that are received within 24 hours of the given deadline (grace period);
- 1% grade deduction for every hour after the first 24 hours!
- No submission will be allowed after Tuesday, 11:59pm.

General Grading Policy

We will grade your programs following these steps:

(1) We will read your program and give points based on the points allocated to each component, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementations of each function.

(2) Compile the code. If it does not compile, 40% of the points given in (1) will be deducted. For example, if you are given 20 points in step (1), your points will become 12 if the program fails to compile.

(3) If the code passes the compilation, we will execute and test the code. If, for any reason, the program gives an incorrect output or crashes for any input, 20% of the points given in (1) will be deducted.

Please notice that we will not debug your program to figure out how big or how small the error is. You may lose 40% or 20% of your points for a small error such as missing a comma or a space!