

## Assignment 6: ToDo Reminder

### 1. Objectives

The main objectives are:

- Working more with **enum**, constructors, properties and **List<T>**.
- Using some new common and useful controls, e.g. **MenuStrip**, ToolTip and Timer control, as well as the **MessageBox**.
- Save data to files.

You are expected to solve this assignment using your experiences from the previous assignments. Your code must be well-structured, documented using proper comments and that you use properties, constructors and encapsulation in a correct way as you have learned throughout the course.

### 2. Description

Apu is having a hard time getting organized with all the tasks that he has to do, and remember to do them on time. He should not of course miss birthdays, shopping, getting a haircut and so on. Your assignment is now to write a C# program for him (and all others having the same problem) that keeps track of all to-dos.

The program should allow the user to specify a task, select a task date and time, and the program should display the data in a list (ListBox, or ListView) on the GUI.

Create a class **Task** to store data and handle operation for a to-do (a task item) and a container class, **TaskManager**, containing a List<Task> object to manage a list of the tasks inputted by the user. Use an enum for Priorities. Figure 2 shows a sample design for the user interface.

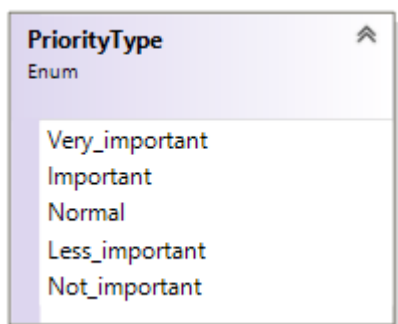
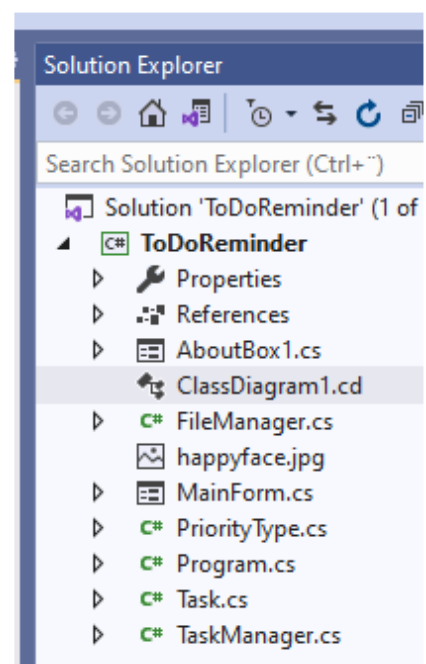


Figure 1: The Project



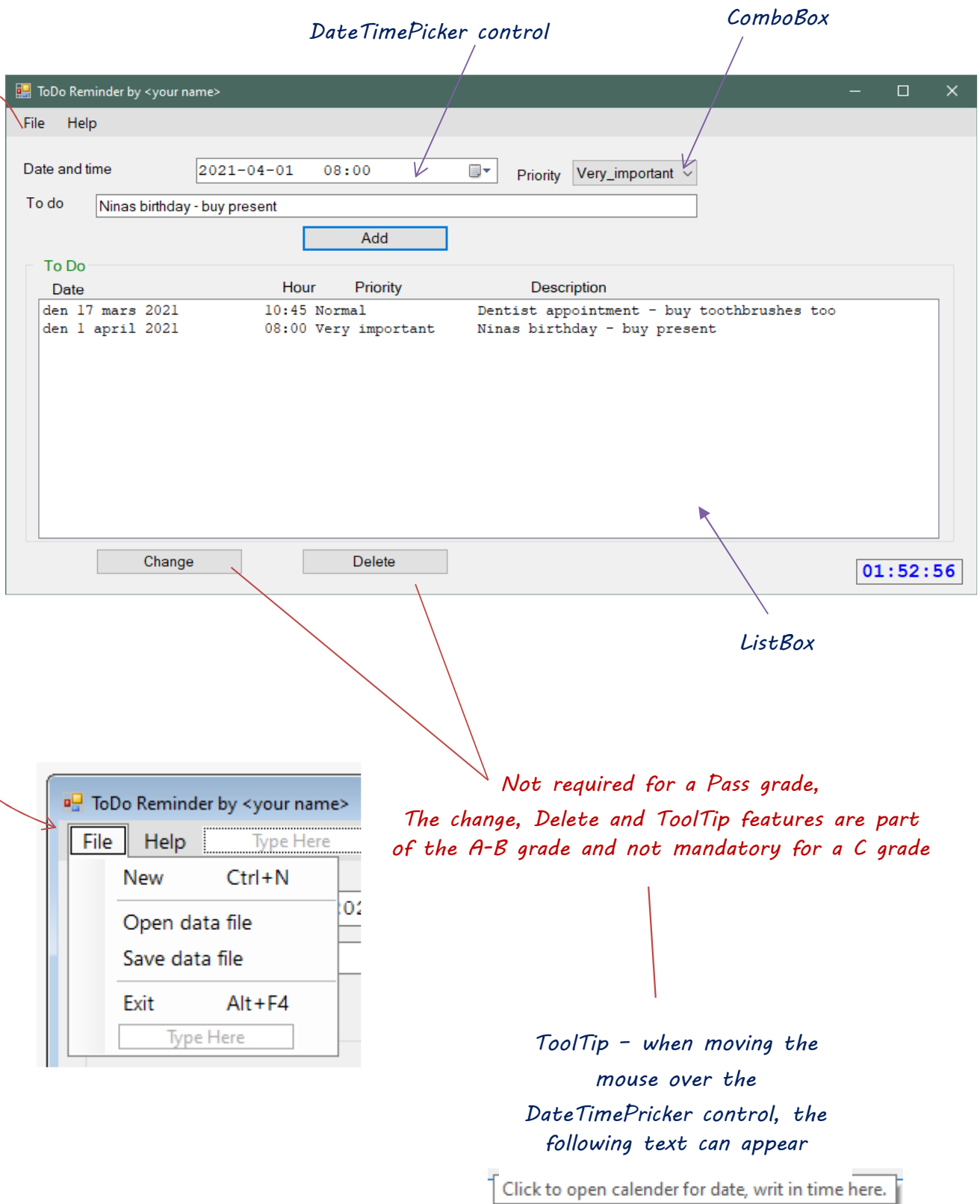


Figure 2: The GUI

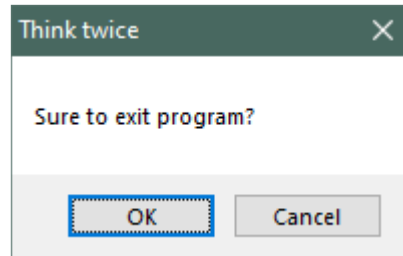
### 3. Features and requirements for a Pass grade (C)

The application should be functioning well and it should control the user input so it does not crash or give unexpected output for invalid input.

- 3.1 The user should be able to select a date, time and priority and add a description for the task.
- 3.2 At program start, the Form should be clear of all design-time texts (Form1 as the title of the form is not accepted).
- 3.3 **Add a new Task:** The user clicks the Add-button after inputting a date a time, and giving a description (or name) of the task, to save the data into the ListBox (see the preceding image).
- 3.4 The GUI should have a main menu bar consisting of the menus File and Help and the submenus as demonstrated in the previous figure. You may obviously have more menu items and sub-menu items.
- 3.5 You can approach file handling in either of the following two alternatives:
  - 3.5.1 Use a single-file with a path relative to the EXE-folder and a hard-coded name (for ex. "reminders.txt". This file will be created the first time data is saved and will be rewritten the subsequent times when the user chooses to save data. The program can then read the same file when the user chooses to open the file.  
  
**Note:** The file name (**reminders.txt**) can be hard-coded but the path to this file should not be hard-coded (for ex **C:\Data\reminders.txt** is not allowed). It will work on your computer but not in other computers.
  - 3.5.2 Use Windows Forms **SaveFileDialog** control to let the user select the device, folder and the file name to save data. The user can select an existing or a new file every time. Use the **OpenFileDialog** control to open a file from saved by the program in much the same way.
- 3.6 The **File** menu should have the following sub-menus:
  - 3.6.1 **New** to reset the program exactly as at start-up. The program should not terminate. All you have to do is to call the **InitilizeGUI** method, to initialize the MainForm and also re-create the manager..
  - 3.6.2 **Open data file:** open the data from a text file that you have saved the data previously. The file must exist.
  - 3.6.3 **Save data file:** save the data to a text file.
  - 3.6.4 **Exit** to close the program after a confirmation message given to the user, when the user clicks the OK button in the message box. Do not exit the

application if the user presses the Cancel button; nothing should change in this case.

- 3.6.5 If the user selects the Cancel button, the program should not exit and the execution should continue as before.



A step by step guidance for the whole solution for saving and reading data using the single-file option is provided in a separate document on the module.

## 4. Requirements and feature for a higher grade (A and B)

Skip this section if you are not going for a higher than a C grade.

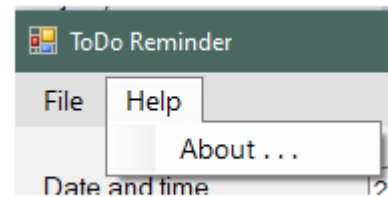
In addition to the requirements listed above, the following requirements are also to be fulfilled:

### For B Grade:

- 4.1 **Tooltip:** When the mouse is over the **DateTimePicker** control, a help message is to be displayed as the run example (Figure 2) given earlier.
- 4.2 **Timer:** Use an object of the **Timer** control to show the time on a Label (as displayed in Figure 2).
- 4.3 **Change a Task:** The user should be able to change data for a selected item in the listbox.
- 4.4 **Delete a Task:** The user highlights an item in the ListBox and then presses the Delete button. Your program shows a message box letting the user have a chance to confirm or cancel deletion.
- 4.5 The **change** and **delete** buttons must be disabled when the ListBox is empty, or when no item in the ListBox is selected (highlighted) by the user.

### For A Gade: In addition to above B-requirements:

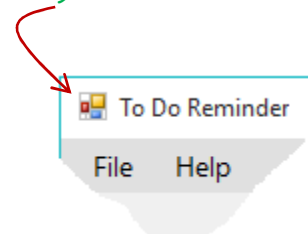
- 4.6 **About Box:** When the user selects the sub-menu **Help - About...**, an About Box is to be displayed. Open the Properties >> **AssemblyInfo.cs** file and change the Description text and other relevant data. Change also the image to one of your own one, your photo, an icon or an image from the Internet. **Remember:** use a **PictureBox** control as a placeholder for the image.



**Optional:** Create an icon for your application and replace the VS's default icon. You will need to draw both a 16 \* 16 and a 32 \* 32 Pixels icon. Read the document "Working with Resources in VS" that is available in the module and get help from MSDN or other resources on the Internet.

**Optional:** You can also test print-controls **PrintDialog**, **PrintDocument**, etc. in the Toolbox in VS to print out values to the printers connected to the computer.

*Default icon -  
Change this to  
your own icon*



## 5. Help and Guidance

Although it is expected that you solve this application by using your experiences from the previous assignments, a help document is available in the Module

## 6. Submission

Submit your assignment in the same way as the previous one.

Good Luck.

*Farid Naisan*

Course Coordinator and teacher