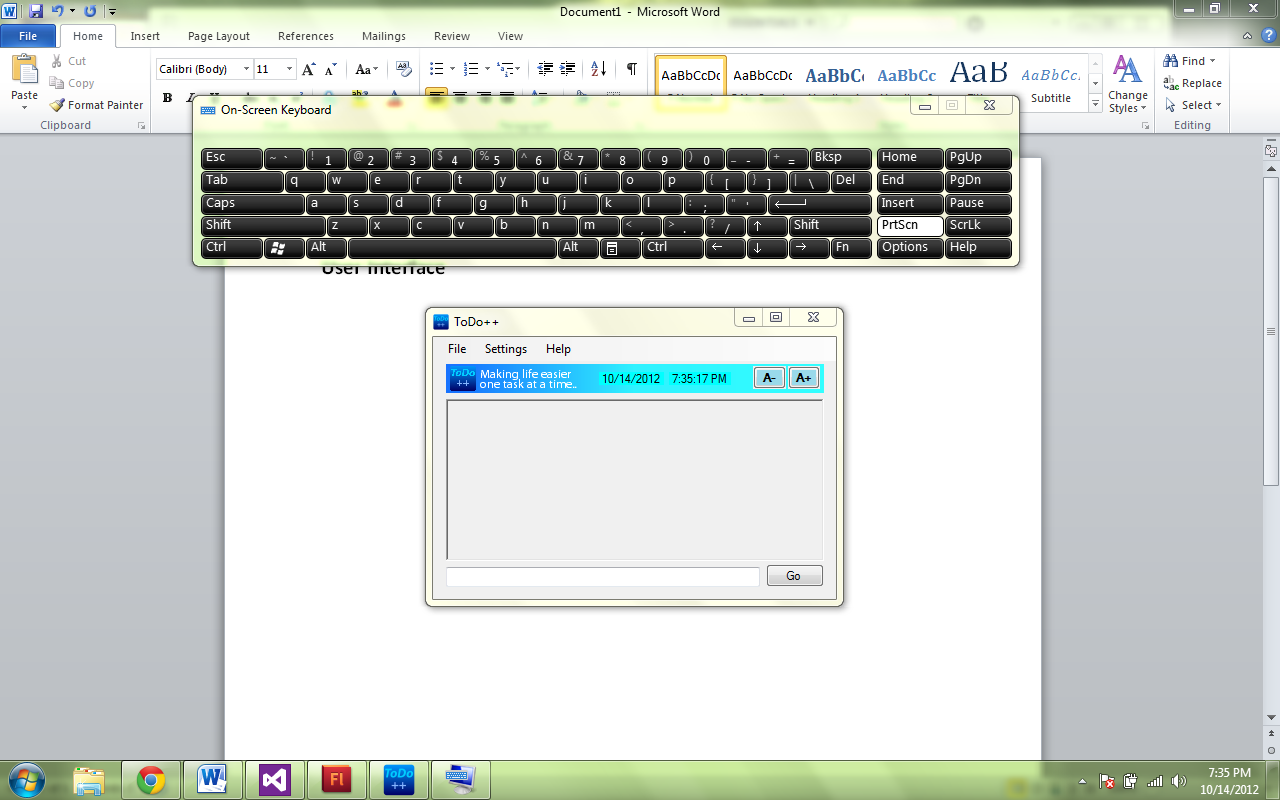
**Main User Interface**

Developer Guide for User Interface

**Controls/Interfaces**



**textInput**

Place where user inputs commands

**goButton**

ToDo++ executes the command. Enter may be hit too

**TimeDateControl** (TimeDateControl.cs)

Displays Date and Time

**increaseSizeButton**

Increase Size of Text

**decreaseSizeButton**

Decrease Size of Text

**OutputBox Control** (Menu.cs)

Requires an instance SettingsManager to be passed in

**Menu Control** (Menu.cs)

Requires an instance of UI and SettingsManager to be passed in

**Overview**

The User interface class is what the User see’s when the program is open. It is where all the magic begins. In order to function, this class contains an instance of SettingsManager (see below) and Logic (see below). SettingsManager contains all program settings that are loaded when ToDo++ is opened. Logic is where user input is passed in and processed. Listed below is a class diagram showing the relations between the objects.

**Main User Interface (UI.cs)**

What the User sees when opening ToDo++

Attributes

|  |  |
| --- | --- |
| Variables | Explanation |
| private Hotkeys.GlobalHotkey ghk; | Global Hotkey to Minimize to System Tray |
| SettingsManager mainSettingsManager; | Settings Manager stores all settings data, including Flexi-Commands |
| Logic logic; | Instance of Logic that handles Data structure and File Operations |

Constructor UI()

|  |  |
| --- | --- |
| Methods | Explanation |
| PrepareSystemTray(); | Loads Code to place App in System Tray |
| PrepareSettingsManager(); | Loads initial Settings of App and applies the settings |
| PrepareMenu(); | Loads the menu strip |
| PrepareOutputBox(); | Loads Output Box |
| PrepareLogic(); | Creates instance of Logic to be used by Text Processing |

Win32Functions

|  |  |
| --- | --- |
| Methods | Explanation |
| private void PrepareSystemTray() | Registers the Hotkey (Alt+Q) to be used for Minimizing/Maximizing to/from System Tray |
| private void MinimiseMaximiseTray() | Minimize/Maximizes App from System Tray |
| private void RegisterInStartup(bool isChecked) | Sets whether ToDo++ is to be open at startup or not |

PrepareFunctions

|  |  |
| --- | --- |
| SettingsManager | Explanation |
| private void PrepareSettingsManager() | Creates an Instance of Settings Manager |
| private void MinimiseToTrayWhenChecked() | Minimizes App to System tray if true |
| private void RegisterLoadOnStartupWhenChecked() | Sets the Load on Startup Status |
| Menu | Explanation |
| private void PrepareMenu() | Prepare the Menu Bar. Pass an instance of settings manager into it so it can interact with it. Also loads the settings into menu |
| OutputBox | Explanation |
| private void PrepareOutputBox() | Prepare the Output Box. Pass an instance of settings manager into it so it can interact with it. Also load the settings into outputBox |
| Logic | Explanation |
| private void PrepareLogic() | Creates an instance of Logic |

TextInput/KeyboardCommands

|  |  |
| --- | --- |
| Methods | Explanation |
| private void ProcessText() | Passes an the user text to Logic, which processes it and returns an output to be displayed |
| protected override bool ProcessCmdKey(ref Message msg, Keys keyData) | New Keyboard shortcuts may be entered here. There is no need to use this function directly. Simply place all Keyboard shortcut keys here. Do not use ALT+Q as that is used for minimizing ToDo++ to the system tray. |

**Menu Control (Menu.cs)**

Menu Bar

|  |  |
| --- | --- |
| Methods | Explanation |
| public void SetSettingsManager(SettingsManager passedSettingsManager) | Set the settingsManager of the Menu Control, so interaction is possible |
| public void LoadSettingsIntoMenu() | Currently loads the check status of Load On Startup and Get Minimized Status |

**OutputBox Control (Menu.cs)**

Output Box that shows User and ToDo++ commands

|  |  |
| --- | --- |
| Methods | Explanation |
| public void SetSettingsManager(SettingsManager passedSettingsManager) | Set the settingsManager of the OuputBox Control, so interaction is possible |
| public void SetOutputSize(int size) | Sets the Text Size of the OutputBox directly |
| public void DecreaseSizeOfOutput() | Decrease the Text Size by 1 unit, while modifying the settingsManager |
| public void IncreaseSizeOfOutput() | Increase the Text Size by 1 unit, while modifying the settingsManager |
| public void DisplayCommand(string userInput,string systemOutput) | This method displays both ToDo++ output and input with correct formatting |

**Settings User Interface**

Developer Guide for User Interface

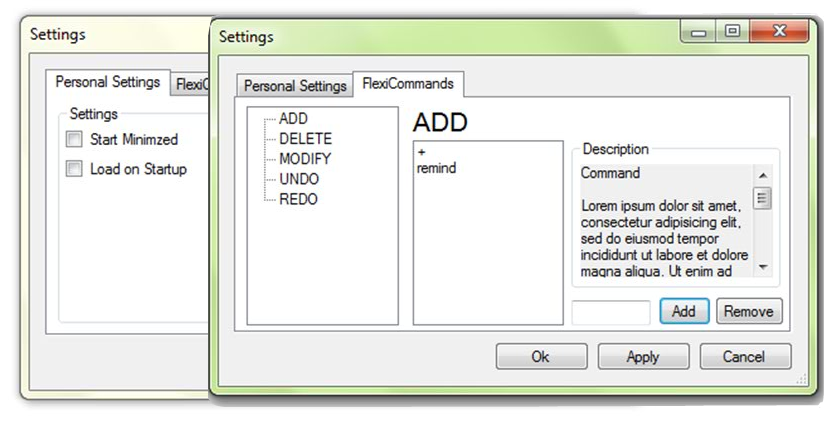
**Controls/Interfaces**

**commandPreview**

TextBox that shows the current command

**listOfCommands**

Lists all user Commands



**commandDescription**

Description of selected Command

**okayButton**

**applyButton**

**cancelButton**

**CommandTree**

This is where all Commands that are modifiable are added

**minimisedCheckbox**

**loadOnStartupCheckbox**

**TimeDateControl** (TimeDateControl.cs)

Displays Date and Time

**increaseSizeButton**

Increase Size of Text

**decreaseSizeButton**

Decrease Size of Text

**OutputBox Control** (Menu.cs)

Requires an instance SettingsManager to be passed in