# PROG 2500 Windows PRogramming - Assignment 2

# UWP Applications – Local Note

Assignment Value: 12.5% of overall course mark.

Due Dates: **See due date on A2 dropbox – Marked by code review during scheduled class time**

Late submissions will receive the standard late submission penalty as stated in the course outline. (5% overall deduction per day late, and 0% after assignment handed back to the class.)

#### Assignment Instructions:

In Visual Studio, use XAML/C# to create a UWP application as described by the requirements of this assignment.

#### Submissions:

Submission of work will be achieved through your GitHub repository. Please ensure that the commit you wish to have marked is labelled with an appropriate Commit Comment in your repo and that you upload a text file with your commit ID to Brightspace. Unless otherwise specified, your most recent commit prior to the assignment deadline will be the one chosen for evaluation.

#### Evaluation:

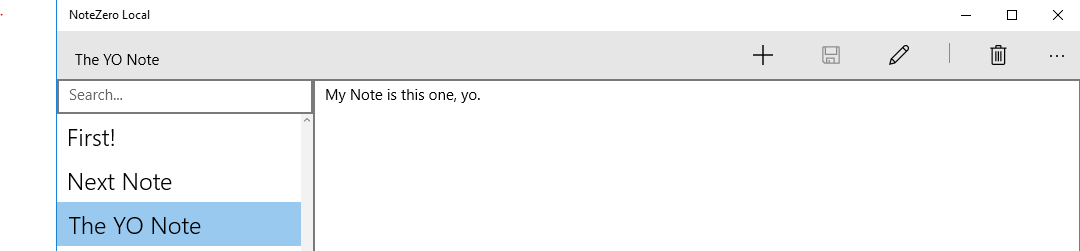
To insure the greatest chance of success on this assignment, be sure to check the marking rubrics at the end of this document or in Brightspace. The rubrics contain the criteria your instructor will be assessing when marking your assignment.

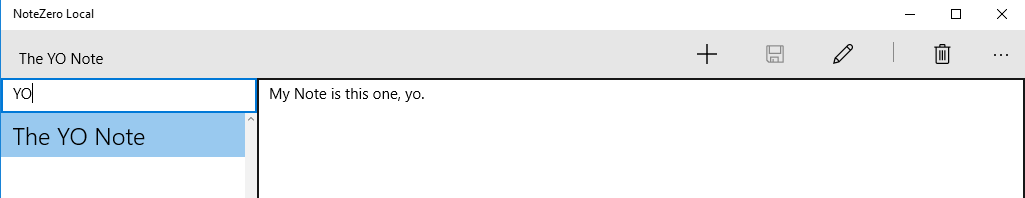
## Program – LocalNote

## Design and write a program allows users to create, view, store and retrieve notes as local text files. Think a simple version of OneNote or EverNote.

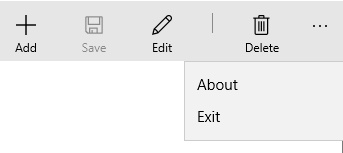
* The text files are retrieved, saved, edited and deleted asynchronously using the Windows.Storage.StorageFolder and StorageFile classes.
* Existing notes are displayed in a list on the bottom left that uses data-binding to get the list.
* Filter the note titles using a textbox immediately above this list (also bound to the data).
* Menu buttons and command bars implement appropriate ICommand scheme throughout.
* Selecting a note displays its name in a label on the command bar but as read-only in the main window.
* The **Add** button deselects current note and blanks main text area for entry.
* Click the **Edit** button to make the text editable.
* If no note is selected you can start entering text and then press **Save**.
* The **Save** button displays an asynchronous dialog to enter the name.
* **Validation**:
  + Cannot have two notes with the same name.
  + If a note is deleted, the text should be removed from main edit section and no note selected.
  + In Edit mode, the **Edit** button is disabled.
  + In Read-Only mode, the **Save** button is disabled.

Sample Screenshots

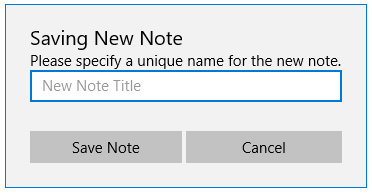
When An Existing Note selected 

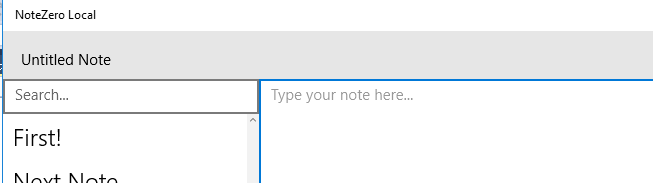
Using the Search 

Commands & Menu



Save Note Dialog



After hitting Add to enter a new note.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Application – LocalNote** | |  |  |  |  |
| **Criteria** | **Insufficient (0 pts)** | **Sufficient (1 pt)** | **Excellent (2 pts)** | **Mark** | **X** |
| **Storage Scheme** | Multiple requirements or elements not working as expected, or missing. | At least one requirement or element does not work as expected, or is missing. | Windows Storage classes used asynchronously and work as expected. |  |  |
| **Data-Binding** | Insufficient or no data-binding used. Much direct value setting occurs. | Data-binding mostly used, but not entirely as expected. Some direct value setting occurs. | Data-binding used throughout application, is demonstrated in XAML, and works as expected. Minimal to no direct value setting occurs. |  | 3 |
| **Layout** | All or too much of application does not scale when viewed at different sizes. | Application mostly scales properly when viewed at diff sizes. Improvements could be made. | Entirety of application should scale properly when viewed at different sizes. |  |  |
| **Command & Actions** | Functionality has multiple errors or omissions. | Functionality has one error or omission. | All menu buttons/commands use proper ICommand methods as expected. |  | 2 |
| **Search Filtering** | Functionality has multiple errors or omissions. | Functionality has one error or omission. | Search filtering of notes works completely as expected. |  |  |
| **Read-Only Mode** | Functionality has multiple errors or omissions. | Functionality has one error or omission. | Selecting a note displays its name in a label on the command bar but as read-only in the main window. |  |  |
| **Add Button** | Functionality has multiple errors or omissions. | Functionality has one error or omission. | The Add button deselects current note and blanks main text area for entry. |  |  |
| **Edit Mode** | Functionality has multiple errors or omissions. | Functionality has one error or omission. | Clicking the Edit button makes the text editable as expected. |  |  |
| **No-Selection Entry** | Functionality has multiple errors or omissions. | Functionality has one error or omission. | If no note is selected, user can start entering text and then press Save. |  |  |
| **Save Dialog** | Functionality has multiple errors or omissions. | Functionality has one error or omission. | The Save button displays an asynchronous dialog to enter the note title. |  |  |
| **Validation** | Functionality has multiple errors or omissions. | Functionality has one error or omission. | Cannot have two notes with the same name. If a note is deleted, the text should be removed from main edit section and no note selected. In Edit mode, the Edit button is disabled. In Read-Only mode, the Save button is disabled. |  | 2 |
| **Exception Handling** | No try-catch blocks implemented or used appropriately. | At least one try-catch block used appropriately. | Try-catch blocks implemented correctly, and cover all reasonably foreseeable exceptions. |  |  |
|  |  |  | **Total:** |  | **/32** |