# PROG 2500 Windows PRogramming - Assignment 4

# UWP Applications – Local Note, Part III

Assignment Value: 12.5% of overall course mark.

Due Dates: **See due date on A4 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, Part III

## Continue the development of the LocalNote app created in Assignment Two & Three, with the additions listed below.

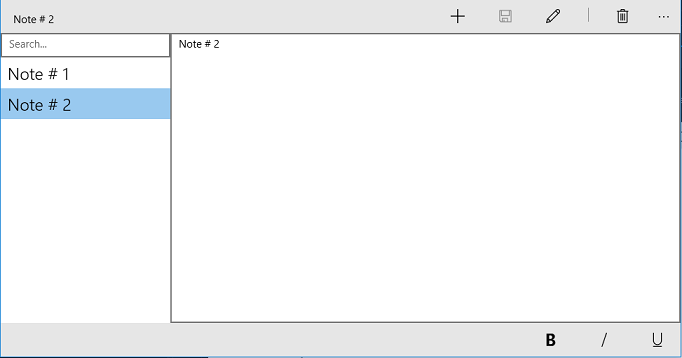
**Note:** All primary application functionality from the previous assignment is expected to work properly, which means errors/omissions from the previous submission are required to have been fixed. This doesn’t mean every cosmetic issue or small-scale bug must be fixed, but you will lose marks on this version of the app if primary elements of the app are missing or broken.

**Additional Functionality:**

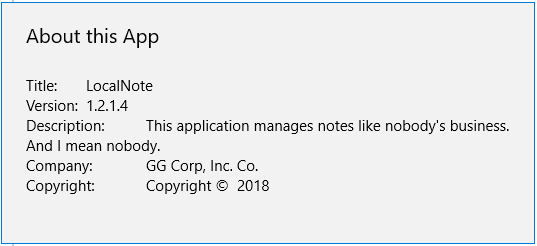
* Functionality from previous app versions should continue to work as expected.
* Refactor the backend of your existing application to read and write all Note data from a **local** SQLite database (in place of saving text files to the file system). This enhancement should cover all reading, saving, editing and deleting functionality from the prior assignments.
* While in a note’s Edit Mode, add a new command bar to allow some Rich Text Format (RTF) formatting of note content. At minimum, functionality for bold, underlined and italicized text must be added.
* Change the About screen to be a XAML page that displays more app-specific data, retrieved from the application’s settings. The new XAML page should have a Back button to return to the main screen.
* Accessibility improvements must be added to all major UI elements in the app, such as Help Text and tooltips.
* Coverage for all new non-UI functionality should be added to the Unit Testing scheme, with a comment to show differences from unit tests for the previous assignment.
* Final polish, best practices and robustness of the app will be evaluated, to ensure the app has a friendly and appealing user experience, generally follows coding best practices, is bug-free and uses a reasonable amount of exception handling.

Sample Screenshots

Command bar with some basic RTF commands, while in Edit Mode for a selected note



New About Screen as new page, retrieving data from app’s assembly information.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Application – LocalNote, Part III** | |  |  |  |  |
| **Criteria** | **Insufficient (0 pts)** | **Sufficient (1 pt)** | **Excellent (2 pts)** | **Mark** | **X** |
| **Prior Functionality** | Multiple requirements or elements not working as expected, or missing. | At least one requirement or element does not work as expected, or is missing. | All primary functionality from previous assignment requirements present and working as expected |  |  |
| **Database Implementation** | Multiple errors or omissions, or is not implemented. | At least one error or omission. | Application properly uses a database for all note-based transactions. Notes are no longer saved to the file system. All data is persisted between sessions. |  | 3 |
| **RTF Functionality** | Multiple errors or omissions, or is not implemented. | At least one error or omission. | In a note’s Edit Mode, a command bar is displayed to allow B-I-U rich text functionality, which works correctly and as expected. RTF changes are properly persisted in the note data. |  |  |
| **About Screen Enhancements** | Multiple errors or omissions, or is not implemented. | At least one error or omission. | About screen is now a separate page, displays application-specific data from the app’s settings, and has a Back navigation button. Everything works as expected. |  |  |
| **Unit Testing** | Unit Tests not present, has insufficient coverage or contain multiple errors. | Unit Tests present but without required coverage, or contains some errors. | Unit Tests present and cover all non-UI program functionality, including both old and new functionality. New tests commented to show differences from previous versions. |  |  |
| **Accessibility Additions** | Multiple errors or omissions, or is not implemented. | At least one error or omission. | All major UI elements in the application conform to good Accessibility practices – help text and tooltips. |  |  |
| **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. |  |  |
| **App Polish** | Little to no effort made to create a polished and usable UI or follow best practices. | Some improvements could be made. | App’s UI is well-designed and visually appealing. Coding best practices (both C# and XAML) are generally followed. Functionality and layout is user-friendly and polished. |  |  |
|  |  |  | **Total:** |  | **/20** |