**ACS560 Fall 2018 Project**

# Problem Description

You are an avid reader of both digital and classic media. You find yourself having trouble keeping track of every chapter, issue and volume you are in, along with release dates. You want a single user application to track and manage all your reading material regardless of what type (Comic book, novel, manga, light novel, Short Story, Graphic Novel, etc) along with all the relevant information. This includes the publisher, current issue in series, latest issue in series, and next release date (if such information is available).

# Software Requirements

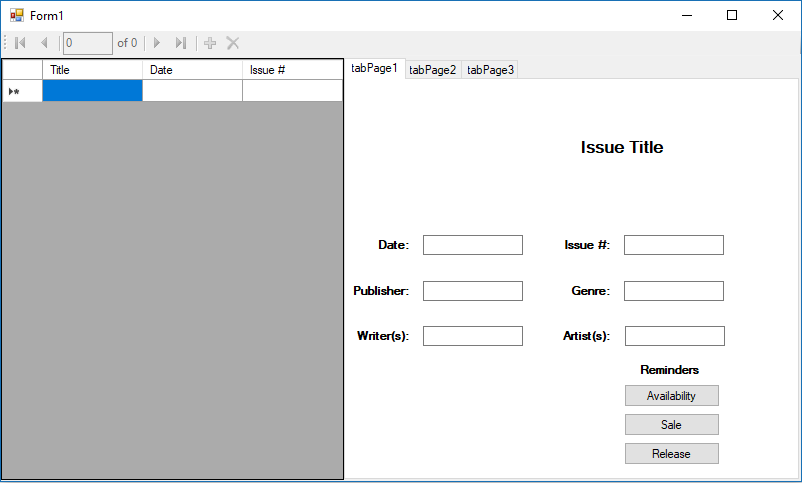
## **Definitions, Acronyms, and Abbreviations**

* Category: This refers to the different type of digital or physical literature that can be managed with this application. Definitions bellow include the default list of categories.
* Comic book: A publication that consists of comic art in the form of sequential panels that represent individual scenes. Panels are often accompanied by brief descriptive prose and written narrative, usually dialog contained in word balloons emblematic of the comics art form.
* Issue: In comic books, an issue is the equivalent of a chapter. Issues are released periodically.
* Manga: Comics created in Japan or by creators in the Japanese language
* Light Novel: A style of Japanese novel primarily targeting high school and middle school students (young adult demographic).
* Short Story: A piece of prose fiction that typically can be read in one sitting and focuses on a self-contained incident or series of linked incidents, with the intent of evoking a "single effect" or mood.

### **Overall Description**

The system will provide the user with the ability to view, add, and manage their reading material which will be displayed in a table format separating them by categories in the form of tabs. The system will also allow the user to view, add and remove each category (along with ever record in it). When adding a category, the user will have to specify the name and the type of reading material that will be in it. When adding a book the user will have to specify the name and year of publishing, the system will then try to determine all other relevant information based on the name. If the system cannot figure out the extra information, it will request the user to input the data manually. The table in each category will display essential information (name, current issue/book in series reading, next release date). When double clicking a specific entry, a window with all the information related to that entry will pop-up.

Preliminary Interface Design:



### **User Characteristics**

Users of this application will be comfortable with the general look of a windows application and be able to read English language text.

### **Constraints**

* Get the appropriate APIs to retrieve relevant book information for autofill.
* Limited to the user of the two programming languages and the one type of database management system specified by professor.
* our lack of knowledge in the two of the three technologies specified by the professor (Golang and MySQL.)
* We are limited to the timeframe of one semester.
* Both team member have different operating systems in their development machines.

### **Specific Requirements**

### **Functional Requirements**

* R1: The system shall provide a mean for the user to create, edit and remove different book categories.
* R2: The system shall provide a mean for the user to add, edit and remove book entries.
* R3: The system shall provide a mean for the user to sort the books in each category.
* R4: The system shall provide a means for the user to enable release date reminder notification.
* R5: The system shall notify the user of the upcoming enabled reminder upon startup.
* R6: The system shall provide a means for the user to disable startup reminders.

### **Non-Functional Requirements**

* R1: The program must be written using C# for the front-end, Golang/Python for the back-end, Socket for network communication and SQLite for the database.
* R2: The program must be finished within the current fall semester.

**Must-Haves**

In addition to the functional requirements

**Nice-to-Haves**

Time permitting, we would like to add the following features:

* Have the ability to retrieve data from the web for different categories.
* Have multiple types of reminders.
* To be able to include other media (ie: Music, Movies, etc.)