# C Project Report: Reading record

## Abstract

As we all know, improve civilian’s reading is very hot these years, especially in China. But it’s not easy for us to know how much we actually read and what we have read. How could we deal with this problem?

It reminds me that we often use video websites to watch videos. It’s very convenient for it can always remember what we watched and when did we watched them which helps us to continue the watching and remind us the things we watched. So maybe I can create something in same method.

So, I decided to create a program which can save all of our reading trail just like the video websites and it can let us know whether we have done well or not in the past several days.

This is my motivation for starting the project Reading Record.

## Introduction/Problem Statement

There were few APPs of the similar purpose in iOS system several months ago (seems that more and more APPs of this kind are appearing in the Apple Store there days) and the function of those APPs cannot fully match my need. I didn’t find any program of this kind in PC system but I take PC as a better place to keep these reading records for I was thinking about saving the reading report into the program at the time.

For these reasons I decided to create a computer program to meet my own needs (which are also things I think most of the readers want). I used C language to create a project which can record the books the user owns and output the book list for user to view them, record everyday reading trail, and statistics the pages read each day and give a brief summary.

It is true that most of the APPs beat my project from various aspects. Because the time is limited and I am making all of these on my own, my project doesn’t have pretty interface and comprehensive functions. But I think that my project has a better statistics system than the APPs.

## **Group Division**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | student ID | College | Proportion |
| 邬冰蕊 | 11170911 | Physics | 100% |

### Member1\_邬冰蕊

I am responsible for all the parts of the project.

* Analyze requirement to find out what the want is and what the real need is.
* Explorer possible solutions to archive the goal (real need of the software).
* Finger out the best experience user interface to delight end user.
* Give detail design.
* Give complete test cases.
* Make sure the final software can be delivered on schedule.

## Analysis

The goal of the software is helping reader to read book as plan by visualized chart.

To read book as plan, reader needs to current status and how he/she read hard in past days, past week, past month. To show reading page number of books, reader has to save reading status daily. Because reader can read multiple books in a week, the reading record should be for each book.

So, the software should have below major functions:

* Reader can add a new book into the software;
* Reader can save reading page number of a book into the software;
* Read can see statistic of reading any time.

Reading record can be the initial version of this software. In future, more features could be added. So, when reader is adding a book into the software, details of a book will be saved. To save reader’s time, these details can be searched from Internet by software itself. These details can be included: abstract, author, publisher, publish date, price, and so on.

Any book can be have reading status: new, reading, and finished. Status ‘New’ means that book is bought, but read is not starting to read it. Status ‘reading’ means that reader is reading the book. Status ‘finished’ means that reader has read this book. Of course, the status ‘finished’ can be changed to ‘reading’ when reader wants to read this book again.

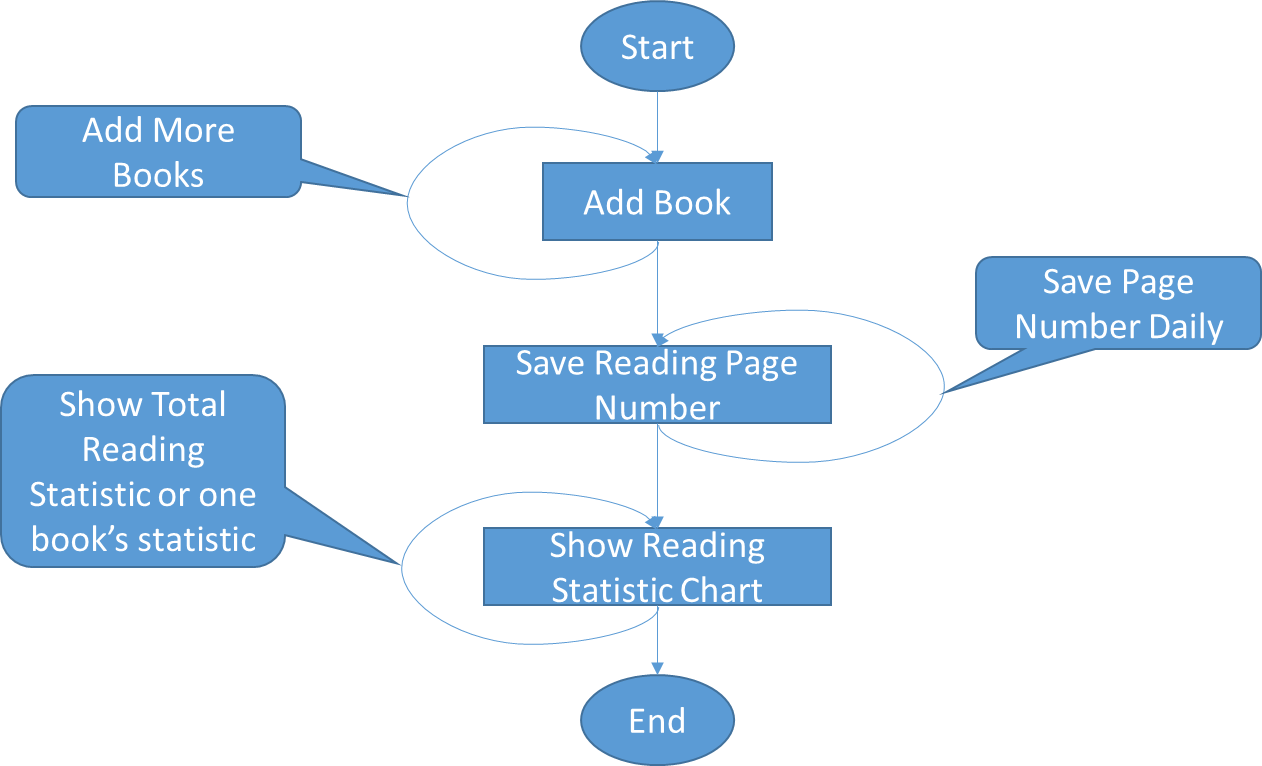
The visualized chart will show total read page number in past weeks/months in every day. The chart also can show each book’s reading record. Meanwhile, maximum reading page number and book will be showed.

When reader starts to use this software, I assumes that will be in this total workflow. Major steps are adding book, save reading page number daily, and showing total books / one book’s statistic chart.

Step 1: adding book. Reader not only can add one book but also can add multiple books in this step.

Step 2: Saving reading page number. As same as step ‘adding book’, reader can save/update reading book page number for one or more books.

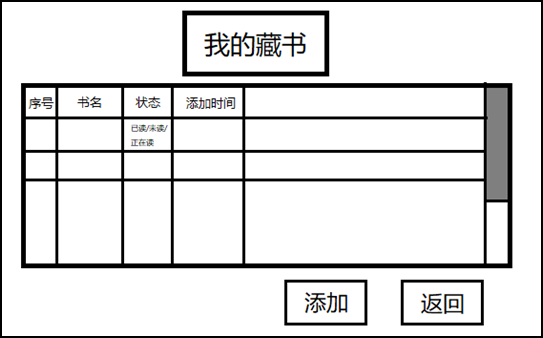
Step 3: Showing reading statistic. Reader can compare different books’ statistic.



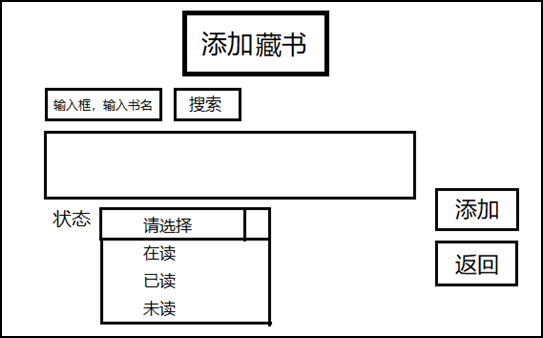
The software UI will be looked like as below.

* Add a book into software.

To avoid adding book duplicated, reader can check book list before adding a new book. When the book list is too long, he/she can delete some books from list (low priority, will be done in next version). Book list will be list at first that UI will be looked like as below.



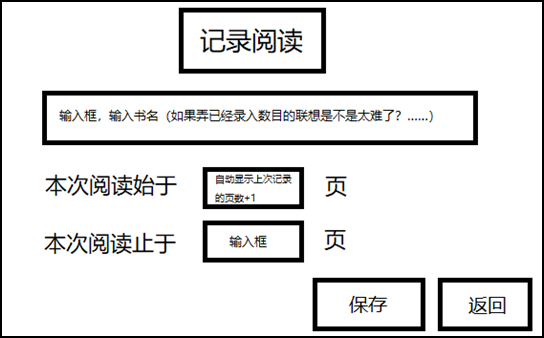
Reader can add a new book by clicking button ‘Add’. A new dialog will be popped up as below.



After reader input book’s name and clicked button ‘search’, the software will search books’ detail information from ‘www.dangdang.com’ or ‘www.amazon.cn’. The related information will be dug out from web page. On this page, reader can set reading status. Because reader may read a book before using this software, reader can set status to ‘new’, ‘reading’, and ‘finished’.

* Save reading page number.

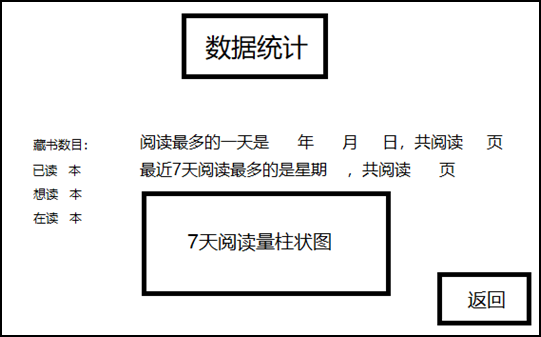
In this UI, reader can save/update reading page number for one or more books. It will be looked like as below.



Because operating system has clock, reader needn’t input date on it. Reader just need to provide book’s name, start page, end page. Then, click button ‘save’ to save it. Reader can save books’ reading page again because reader may continue to read the book after save today’s reading page number. Reader can read some chapters of book, Reader can save start page number directly. Of course, the software can show an initial page number base on previous record.

* Show books’ statistic

Here, reader can learn his/her reading status in a visualized chart that will be looked like as below.



Meanwhile, some comments will be gave. So, reader can have more clear data of reading statistic.

In future version, other readers’ reading statistic can be showed here. So, they can have reading competition.

Finally, these 3 major steps should be combined together. The UI will be looked like as below.



Because this software is helping reader to check his/her reading statistic any time, the statistic may be showed on this UI directly in future version. So, when the software is launched up, reader can see statistic immediately.

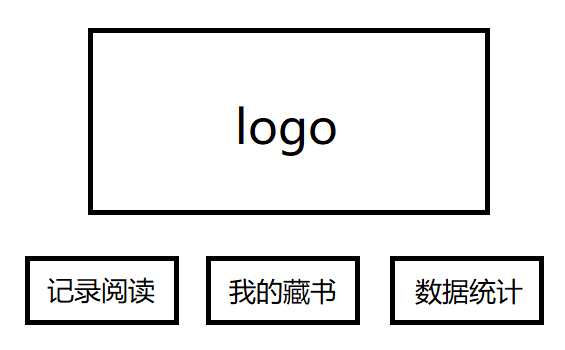
## Design

To show visualized chart for reading statistic, a Windows Desktop application will be developed. Because reader will add books, save reading page number, a dialog application is a best choice. So, the software will be an dialog application that will be programmed with C/C++.

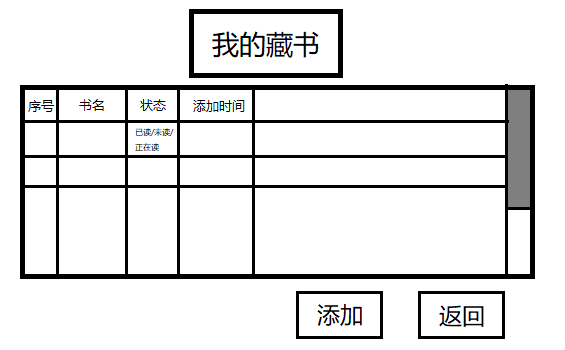
### Interface design

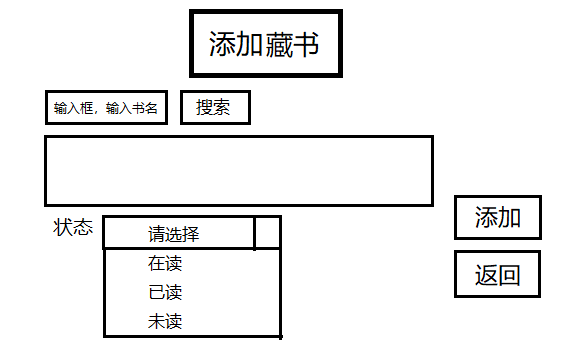
The software needs to achieve a lot of functions, so the interface is combined of several pages. Each page is a dialog window. The functions are divided into three groups and show in different pages.

Main page combines these three groups together. Reader can access these groups from this page.

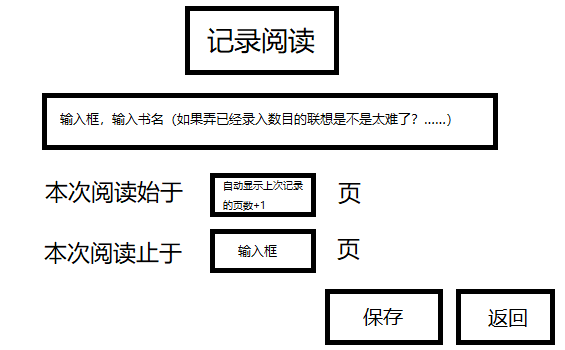


The first group is adding book into software. It combines two page: book list and add new book page.

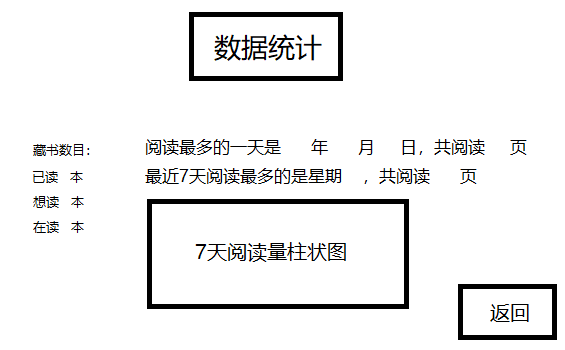




The second group is saving reading page number. It has only one page.



The third group is showing reading statistic. It is one page too.



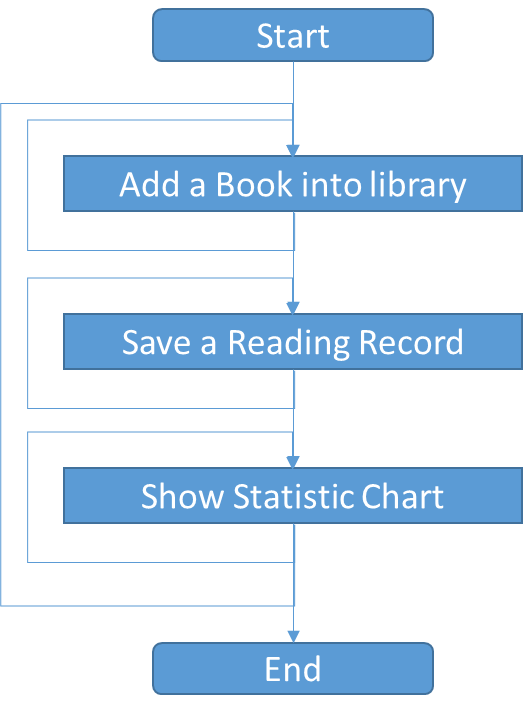
### Main Page

The final design for main page is the same as the first edition. Reader can click the three buttons below to enter each specialized page.

Because saving reading page number is the most frequent task, the button ‘record’ for saving reading page number is at the first position. Then, the library is switched to the second position.

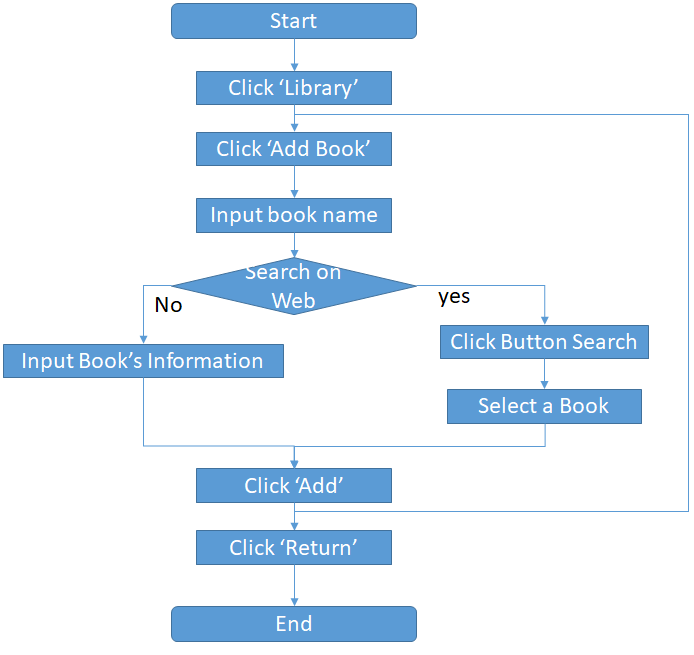


For reader, he/she can access record, library, statistics in below workflow.

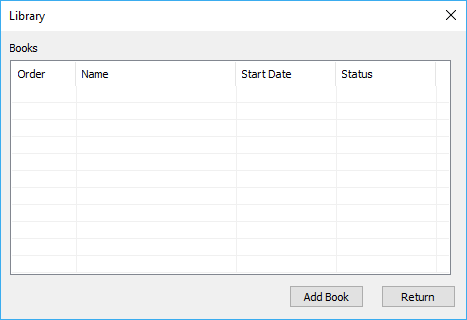


### Add a book

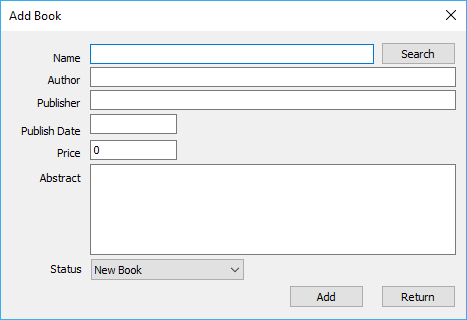
Library Page is where we can view the books we own. It’s very convenient if reader want to add a book on this page. The book can be added in two ways, search the book from www.dangdang.com by the book name that reader enters, other way is that reader enters all book information by himself. The overall steps are showed as below graph.



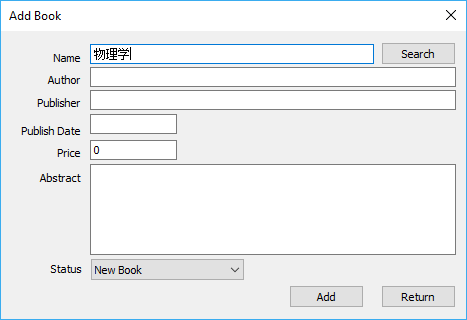
According to design, the screen shot of adding book is.



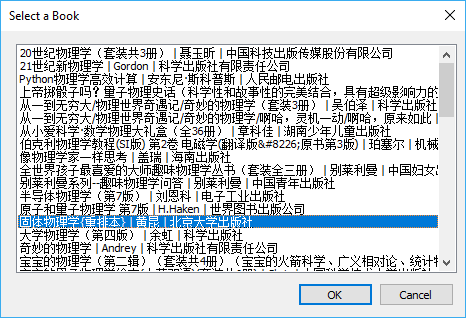
Show all books on this library dialog. When reader luanches the software first time, no book is showed. After reader clicked button ‘Add Book’, below dialog is popped up.



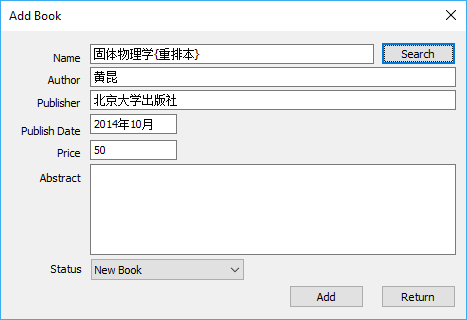
Now, reader can input book name in the first edit control. After input book name, reader can continue to input other information in different edit controls. Or, reader can click button ‘Search’ beside book name edit control.



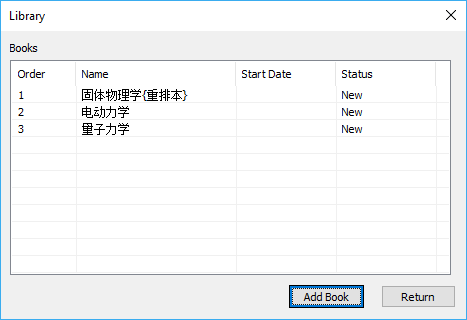
If software got search result from www.dangdang.com，then below dialog will be popped up. Reader can select correct book from the list.



After reader select a book, software will try to get details about this book and show in ‘Add book’ dialog as same as below.

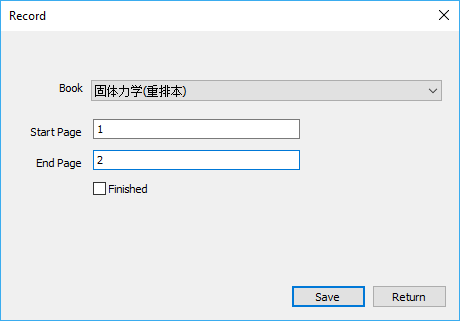


After reader added all books, he/she can click button ‘Return’ to return to library dialog. Now, library dialog shows reader’s books. It could be looked like as below. By default, status for new book is ‘New’.

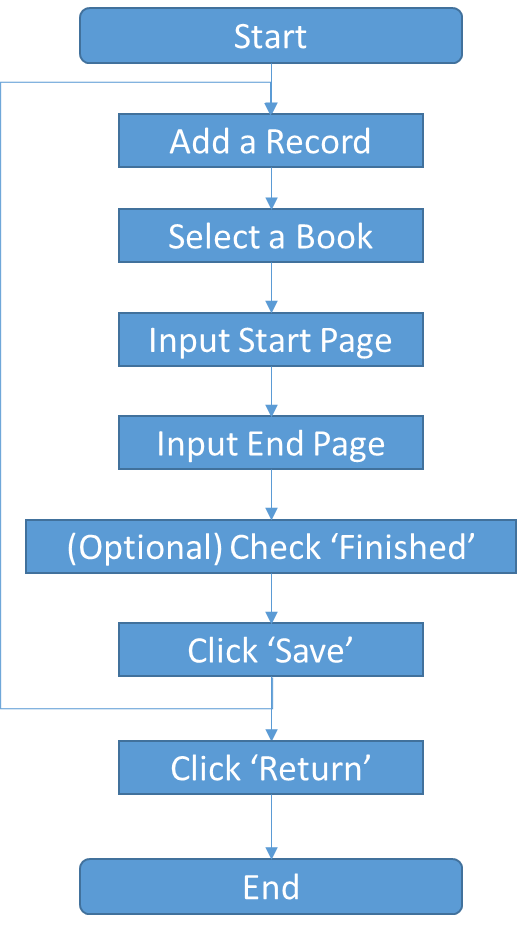


### Add Reading Record

On main page, reader clicks button ‘Record’. Record dialog will be showed as below. Books’ name is showed in drop down control. Reader can select the book who is reading. After reader selected a book, software will give a suitable start/end page here. Meanwhile, reader can input start/end page number. If reader has read this book, he/she can click the checkbox ‘Finished’. If reader clicked this checkbox, the book status will be set to ‘Finished’. Of course, reader can change it to reading any time when he/she reads the book again. For start/end page, software will not verify page number either because reader can read any chapter again and again. After finished all items, reader can click button ‘Save’ to save the record. The dialog will not be closed after clicked button ‘Save’ because reader may read other book in same day. So, reader can save other book reading record.

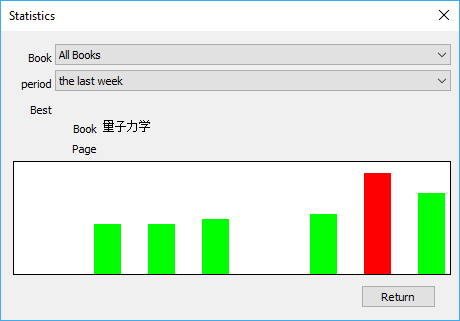


According above description, the workflow can be showed as below graph.

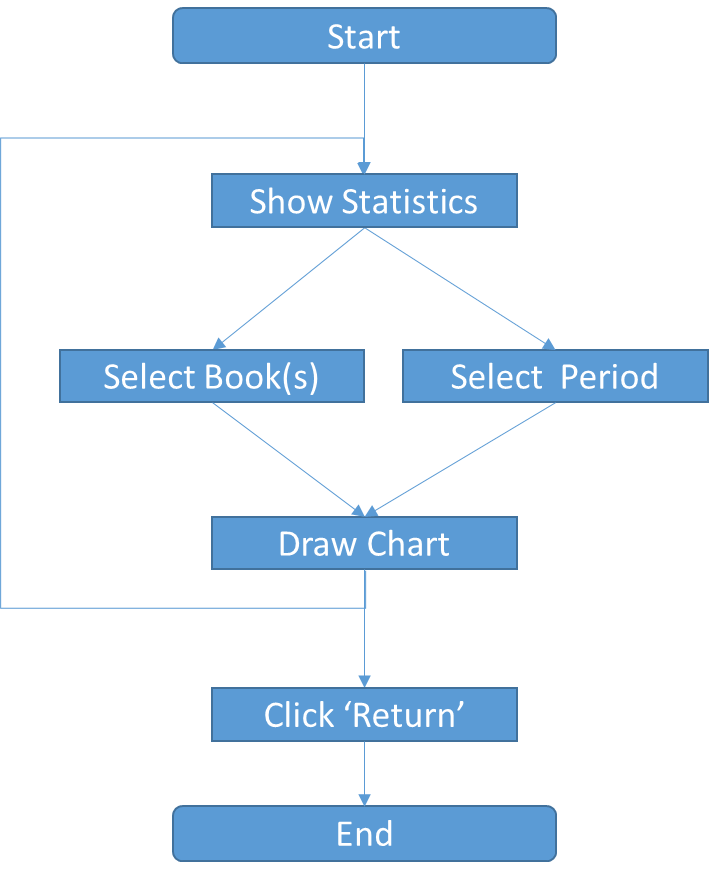


### Show Statistics

Showing reading statistic is the most important in this software. Reader can make adjustment based on this statistics. The dialog is showed as below.

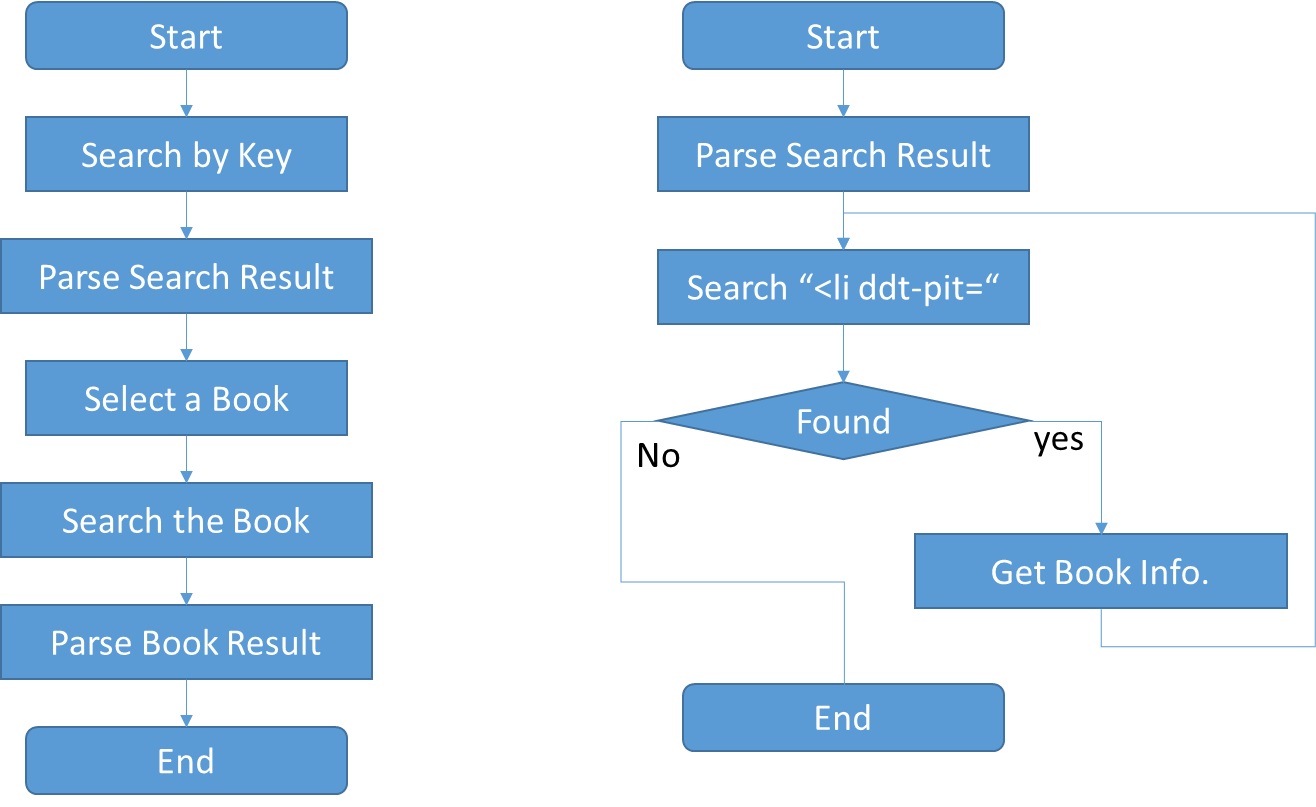


Reader can learn recent reading status about all books, or just one of them by selecting book on the first drop down control. Reader can select statistics’ period. So, he/she can learn reading status in past week, two weeks, month. So, chart will be updated when book or period is changed.



### Search a Book on www.dangdang.com

According research of www.dangdang.com, book details will be retrieved in two steps. At first step, searching books with book name. The second step, download web page of a special book.



## Implementation

### Framework of the software

Create the software with Visual Studio Wizard. After Wizard is finished, the software framework is ready. Major files are:

* ReadingRecord.h/cpp: application launch entry code. This is very generic code that will not be changed in whole development cycle.
* ReadingRecordDlg.h/cpp: the main page code. Three button and logo picture will be showed by this code.
* ReadingRecord.rc: all dialog elements are saved in this file that will be modified by Visual Studio itself. We will not touch it in development cycle directly.
* resource.h: dialog elements’ identify, message id, and other resource id are saved in this file that is maintained by Visual Studio. It means that we will not edit it manually.

Showing logo on main page.

* Add a bitmap in Visual Studio Resource View under item ‘Bitmap’.
* Add a control variable (m\_staticLog) in class CReadingRecordDlg.
* In function CReadingRecordDlg::OnInitDialog(), add below code to show the picture.

|  |
| --- |
| HBITMAP hBm = ::LoadBitmap(::AfxGetApp()->m\_hInstance, MAKEINTRESOURCE(IDB\_BITMAP\_LOGO));  m\_staticLog.ModifyStyle(SS\_ENHMETAFILE, SS\_BITMAP);  m\_staticLog.SetBitmap(hBm); |

* Add button ‘Record’, ‘Library’, and ‘Statistics’ on dialog ‘ReadingRecord’.
* Add related process functions by double click each button on dialog design view.

### Adding book.

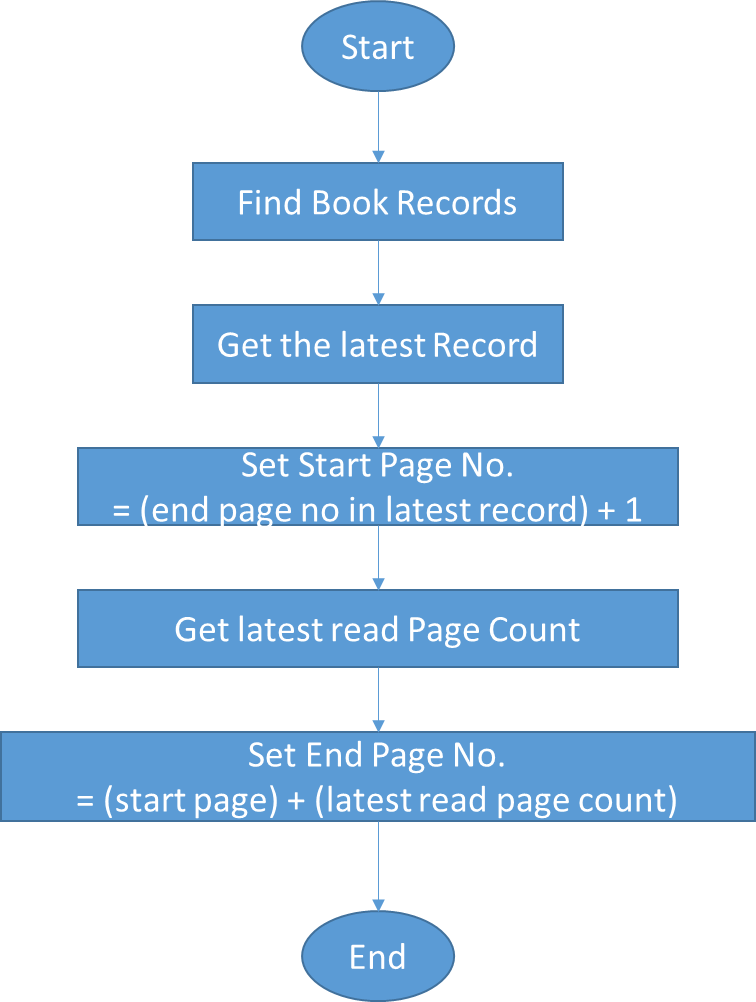
* Add new dialogs on Resource View under ‘Dialog’. Dialogs are ‘library dialog’, ‘add book dialog’, ‘select a book dialog’.
* On dialog ‘library’, add list control and two buttons (‘Add Book’, ‘Return’).
* Add variable for list control. Then, filling book list code will be implemented in dialog source code.
* Add two related process functions for ‘Add Book’ and ‘Return’ by double click the button on dialog design view.
* In same way, add a new dialog for ‘Add Book’ and ‘Select a book’.
* Add code to dig book details from web page (www.dangdang.com). The implement will be described in section 5.5.

Major files:

* AddDlg.h/cpp: the dialog of ‘Add book’. It calls functions to search books by name, shows search result on popped up dialog, show details on related edit controls, and save book details.
* HttpClient.h/cpp: write in C language. Calling Win32 API WinHttpxxx to retrieve web page from www.dangdang.com.
* SearchBookWeb.h/cpp: calling function in HttpClient.h/cpp to retrieve we b page, and parse retrieved web page to get book list and book’s details.
* SelectBookDlg.h/cpp: reader selects one book from this dialog.

### Adding reading page number

* Add a new dialog on Resource View under ‘Dialog’.
* Add all controls on the dialog design view.
* Add related function to respond user input.
* To easy use, reader can select a book from drop down list control. So, reader needn’t input character by keyboard.
* When reader selected a book, the code will read book’s records, find out the last record, and fill the start page number with the last record end page add 1. The code also calculate last read page number and give today’s end page by start page add last read page number. So, reader can modify few character during saving reading page number. The workflow is showed as below.

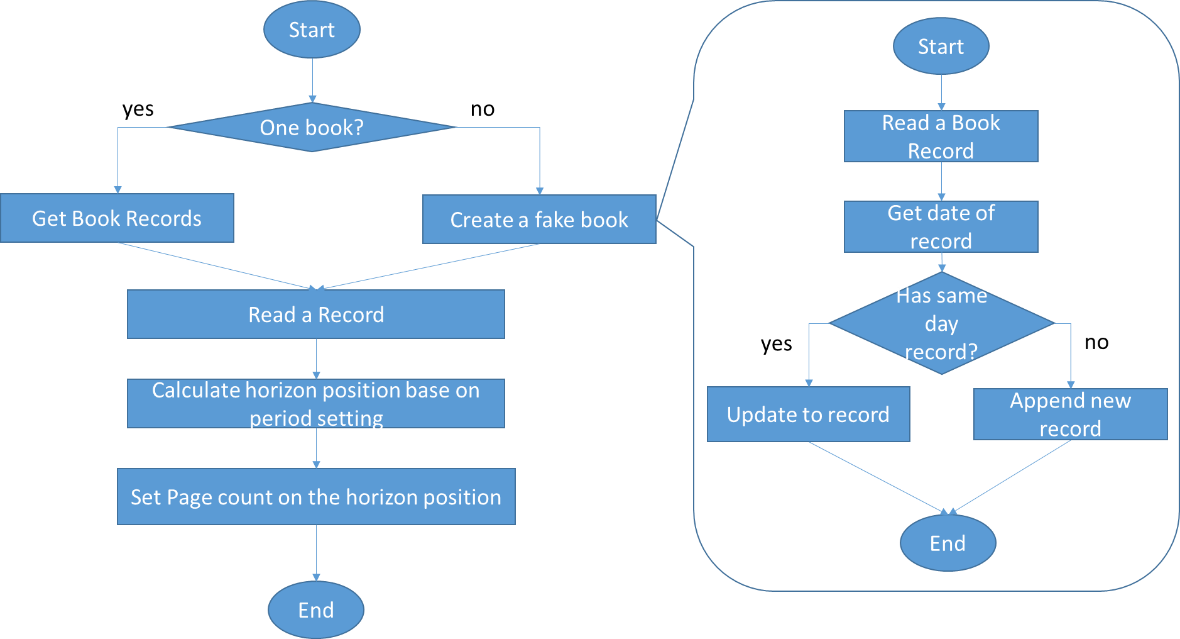


Major files:

* RecordDlg.h/cpp: show dialog and save record.

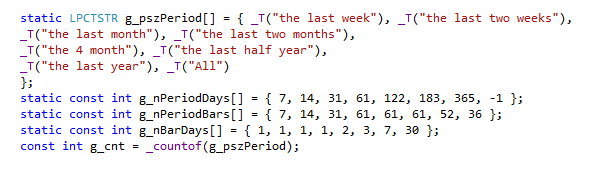
### Show Reading Statistics.

* Add new dialog on Resource View.
* Add two drop list controls for book selection and period selection. For book selection, all book name will be showed in this control. For period selection, reader can learn week, month, and so on.
* Add static controls to show best reading. One is for showing book name, other is for showing most reading page number.
* Add a big static control to show chart.
* To show a chart, creating a new class CStaticChart that is derived from CWnd. The implement of CStaticChart will be described in section 5.6.
* Add two process functions for two drop list control when reader is changed selection. In these functions, function DrawStatisticChart() is called to prepare book reading statistics data and call CStaticChart to show result.
* In function DrawStatisticChart(), the workflow is showed as below.

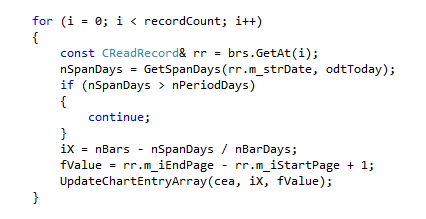


To show total reading statistics, all books’ record must be synthetized first. To save synthetized result, a fake book is created to store daily reading record. Reader can’t grant to read book every day. Inside book object, records is organized as an array and based on date. When a record of book is updated to fake book, software will search existed records of fake book first. If the record is found, then end page number will be increased by reading page count. Else, a new record will be appended into fake book record array. Because horizon position on visualized chart is calculated from record date, the record array needn’t be sorted after a new record is appended.

To show a book’s statistics chart, chart’s x/y value will be calculated before input data to class CStaticChart. Horizon value (x) is calculated from record date. The original x is the first day of the period. To support longer period, the data can be summed up from several days. So, x value is days from the first date of chart and divided by summed up days that is predefined in source code as below.



The calculation is below.



The vertical value (y) is reading page count of each day(s). Because only start/end page number are saved in record, the page count will be calculated with (end – start + 1) page number. NOTE: page count is not end page number subtract start page number. For example, the start page number is 1, the end page number is 2. Reader reads 2 (=2-1+1) pages, not 1 (=2-1) page.

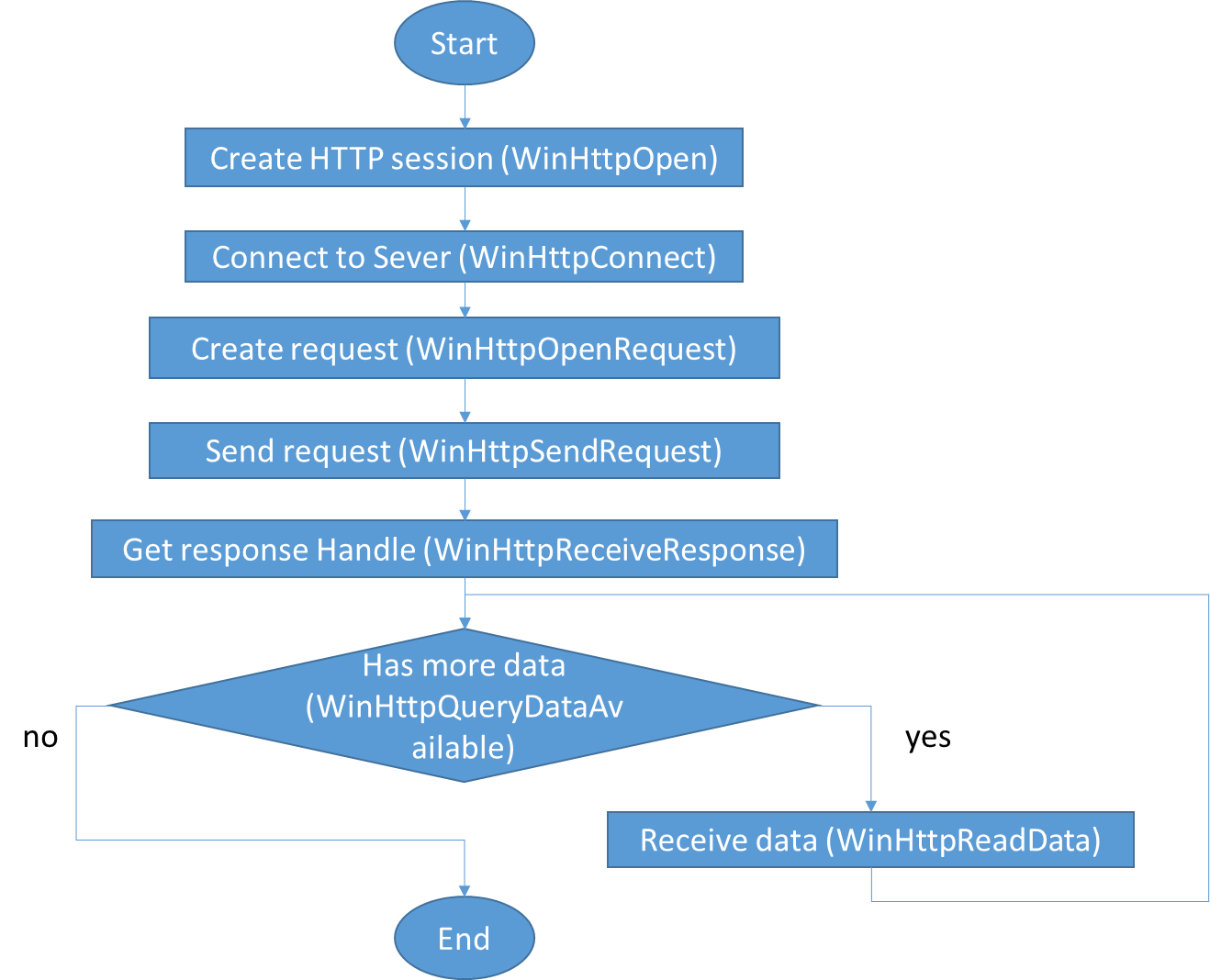
Major files:

* StatisticsDlg.h/cpp: show the dialog. It calls CStaticChart to show the chart.
* StaticChart.h/cpp: show chart on any control.

### Search book detail from web site.

* HTTP request/response.

HTTP is based on TCP that has special protocol between client (application) and web server. It is real complex system. Fortunately, Win32 has related APIs that can help to simply program. But, the process is still complex. So, a new class is created to wrap them up. For any HTTP request, the workflow with Win32 API is in below way.



* Parse HTTP response (HTML Page).

HTML is tagged pure text file. Each document elements are marked with different tag. Each tag may have different id and class. Software is searching special tag/id/class to distinguish content and pick up related information.



Basically, any html is combined up with comments and html element. Html element is combined by head and body. Head is meta data of this web page, includes text encoding and other information. Body contains web page showing content and some JavaScripts.

* Search book’s detail on [www.Dangdang.com](http://www.Dangdang.com).

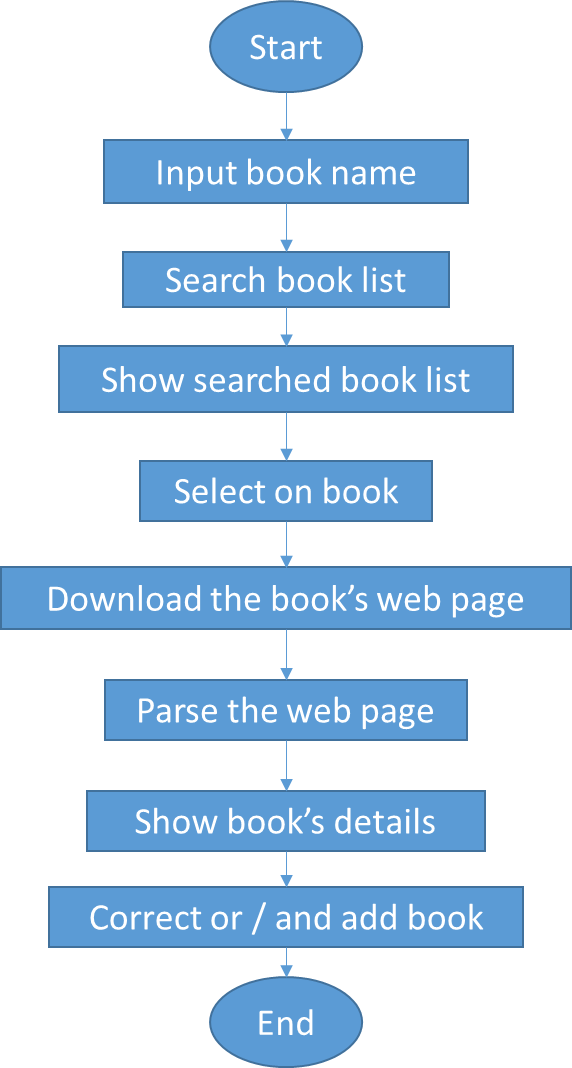
To search book’s details, open web page of [www.dangdang.com](http://www.dangdang.com) search result page as below.



After read several search result web pages, ‘<li ddt-pit="b" class=\"’ in red rectangle is the special identifier that can help software to get book basic information. Meanwhile, hyper link in green rectangle is book’s web page has detail information.

So, to find book’s detail information, the first step is searching book with keyword. The second step is opening special book’s web page. For software, after reader inputs book name (complete name or uncomplete name) and click button ‘search’, software will search books. Because there are many books will be found, a popup dialog will be showed. Reader can select one of them. Then, software will retrieve this book’s web page. In this page, abstract, author, publisher, publish date, price, and other information will be collected and show on ‘add book’ dialog. If the software doesn't collect information in correct way, reader still has chance to modify/input related information.

The major workflow is showed as below.



Major files:

* HttpClient.h/cpp: write in C language. Calling Win32 API WinHttpxxx to retrieve web page from www.dangdang.com.
* SearchBookWeb.h/cpp: calling function in HttpClient.h/cpp to retrieve we b page, and parse retrieved web page to get book list and book’s details.

### Drawing chart.

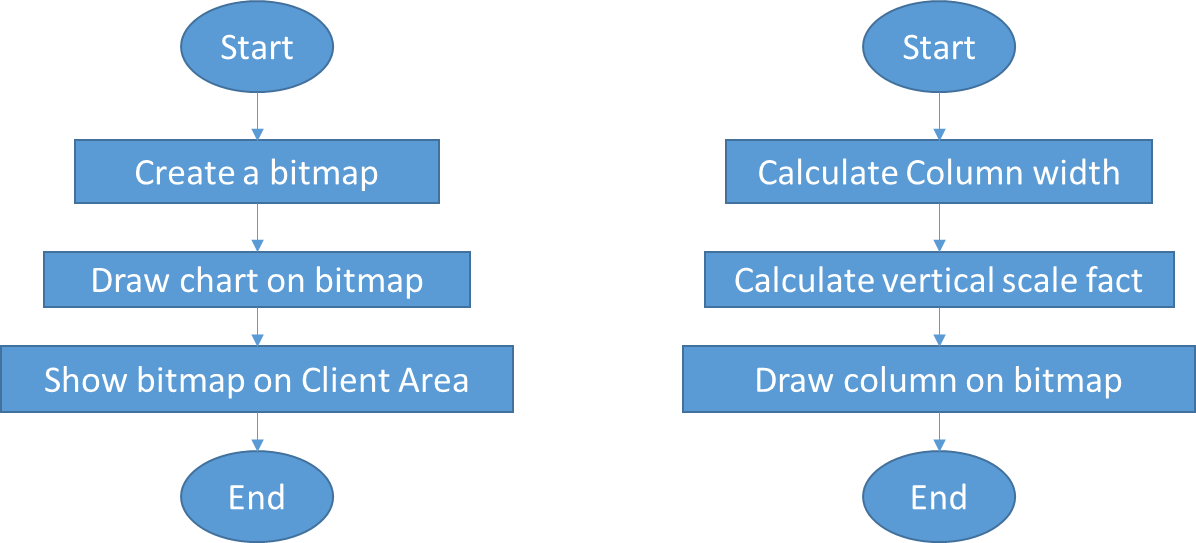
Draw chart is a very generic function that is decoupled from this project. To simpify implement, here only support column chart for one data set. To show a basic chart, caller need provide chart’s x/y value.

To show whole chart, column width is cacluated with chart width and total column number. The gap between two column is half width of a column. So, column width can be gave. The column high will be decided by vertical (y) value. At first, maxiun and minumin will be found out. Scale fact will be cacluated base of vertical abslute value and chart height.

After these two key scale fact is cacluated, the chart can be draw.

To avoid draw chart from raw data again and again when application is hidden/restore or other case. The chart will be draw into a bitmap. CStaticChart will show the chart by bitlt the bitmap. Because chart is bitblt from bitmap on whole client area, CWnd needn’t earse background again.

So, CStaticChar is derived from CWnd and implemented two virtual function OnEraseBkgnd() and OnPaint().



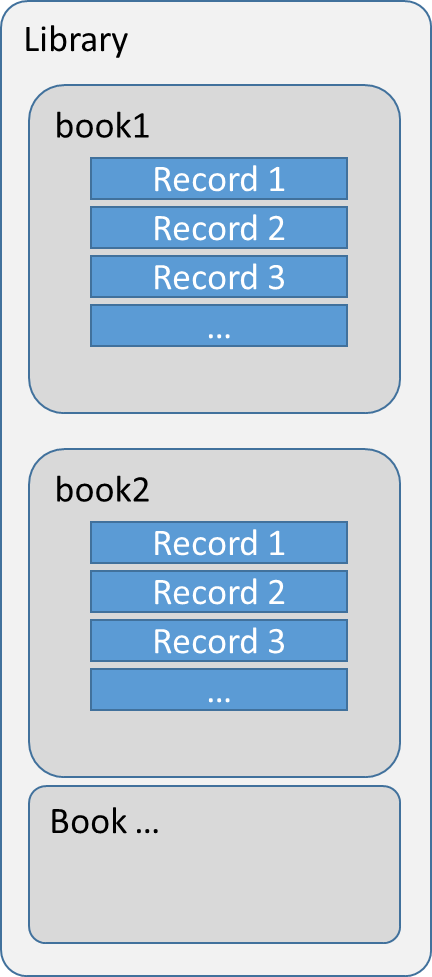
Major files:

* StaticChart.h/cpp: show chart on any control.

### Persistence of book information

All reader input data must be stored on disk properly. Else, after software shutdown, all data will be lost. Of course, statistics can’t be showed.

According requirement, the data is organized in this way. Reader has a virtual library. In this library, there are many books. For each book, it has some reading records. This structure can be described as below chart.

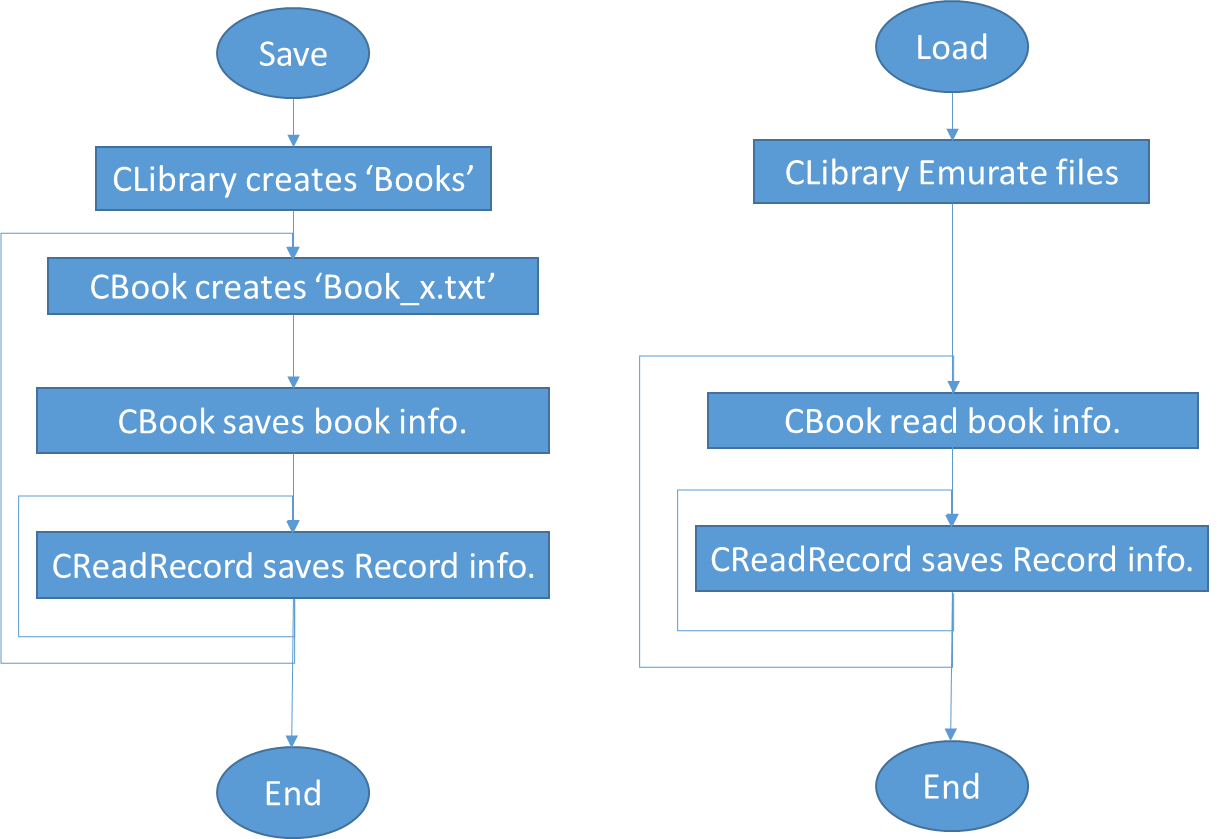


Class CLibrary is designed to store books information. Caller can save books information to disk by calling function save() and load books information from disk by calling function load().

Class CBook is designed to store book’s information. Book’s information is saved/loaded by this class. In real implement, CLibrary calls CBook to save/load infoamiton.

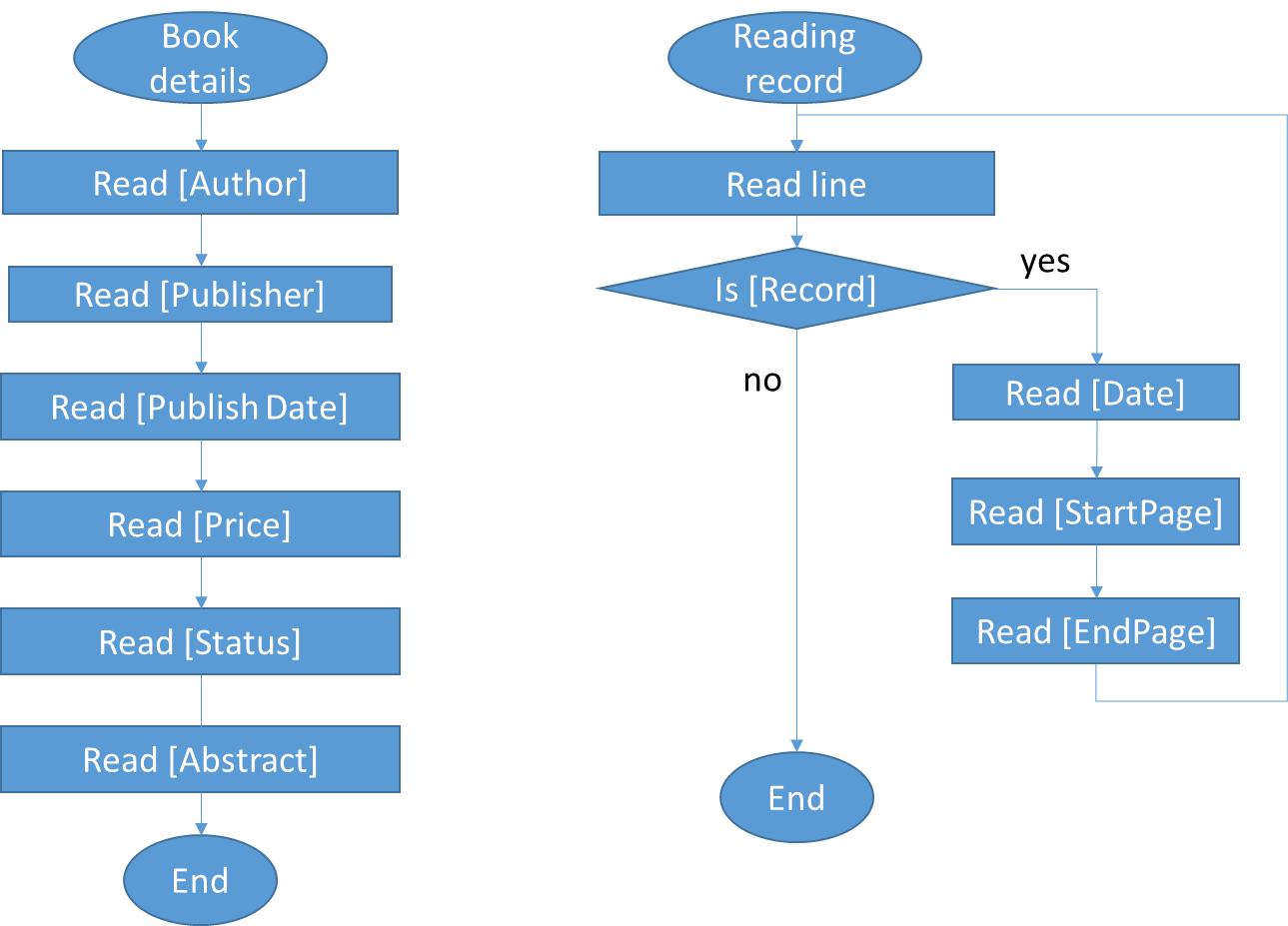
Class CReadRecord is designed to store one book reading record. So, book’s record is saved/loaded by this class. But, CBook calls its function.

So, book information saving/loading workflow is showed as below.

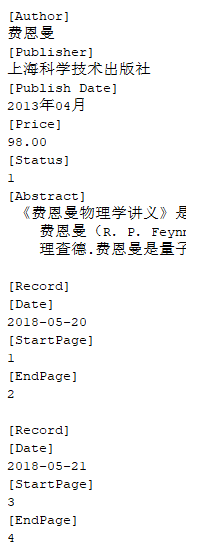


To save book information, class CLibrary creates a subdirectory “Books” under software running directory. Then, class CLibrary saves the book information into a file under this subdirectory. The file name is ‘Book\_+<book id>+.txt’. Book id is an integer from zero. So, the first book’s file name is ‘Book\_0.txt’. This file contains two major parts – book information and reading records. For book information, there are several element include name, author, publisher, publish date, price, status, abstract. For reading records, it is saved as an array. For each record, it is started with tag ‘[Record]’. Between book information and records, there is an empty line is inserted. Between each record, an empty line is inserted too.

Saving books’ information is very straight. Loading is not luck. Software has to read text line one by one line and check tags carefully. For loading, there are two stage, for the first stage, book details will be load. For the second stage, reading record will be loaded. In the first stage, after tag ‘[Abstract]’ is loaded, an empty line is detected, it means the first stage (book details) is loaded. The workflow is showed as below.



Below is an example of book information file.

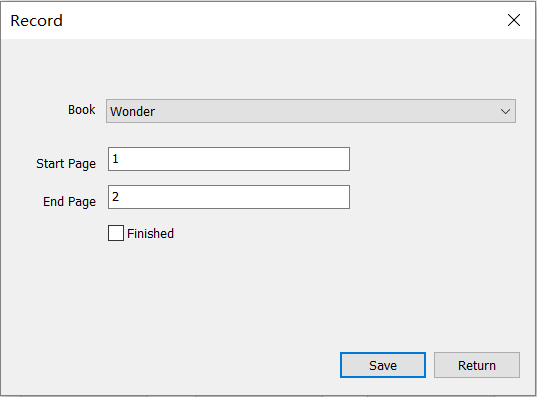


Major files:

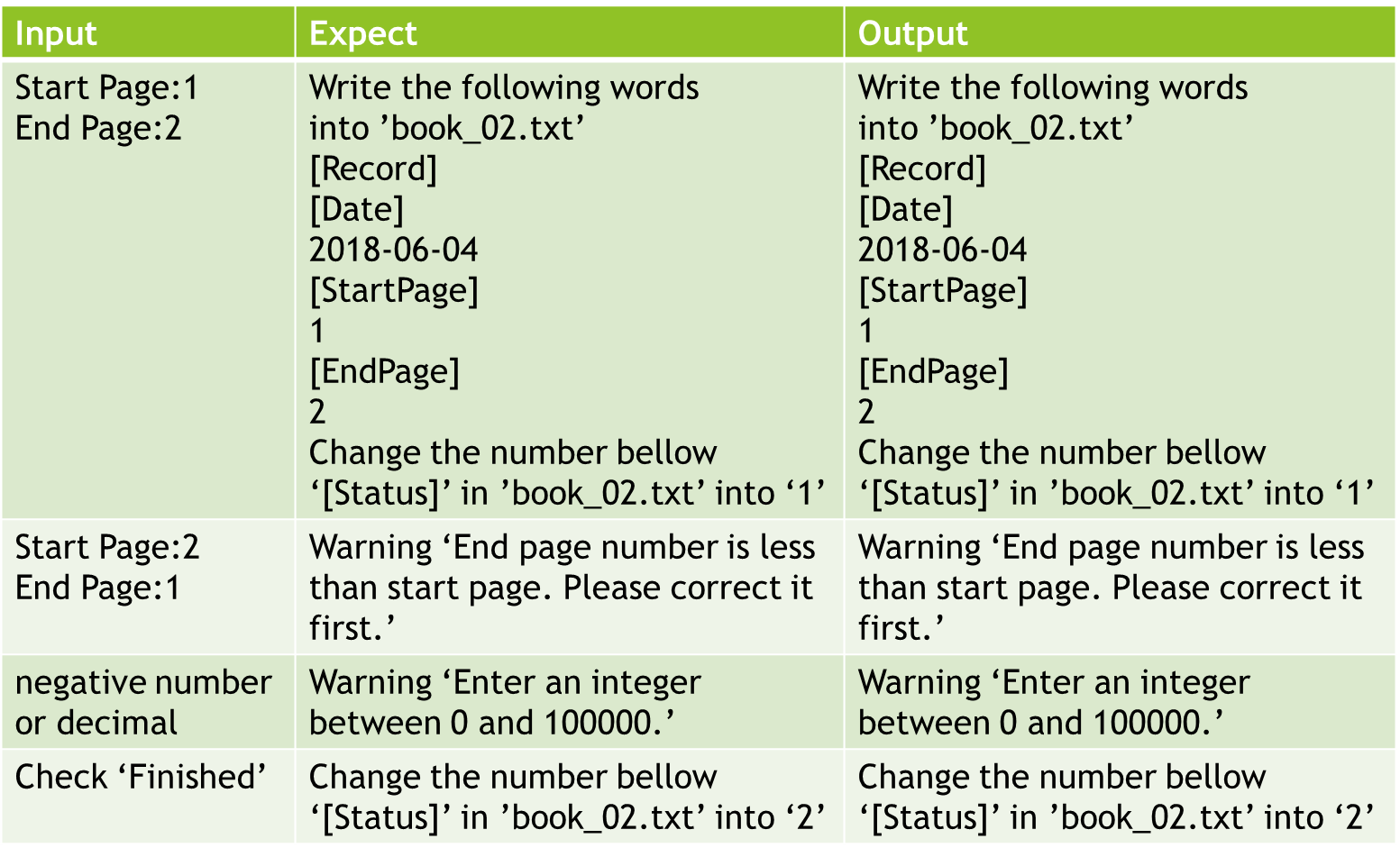
* Library.h/cpp: represent a virtual library of the reader. All books information is stored in this class. It calls class CBook to store a special book’s information.
* Book.h/cpp: represent a book. It contains two type information – book’s details and reading record. It provides not only save/load functions but also add/find/remove book functions. Inside class member function save/load, it will call class CReadRecord function to save/read reading record information.
* ReadRecord.h/cpp: represent a reading record.

## Testing and Debugging

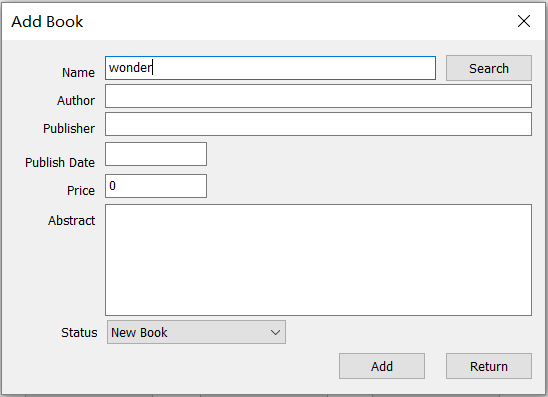
### Record page



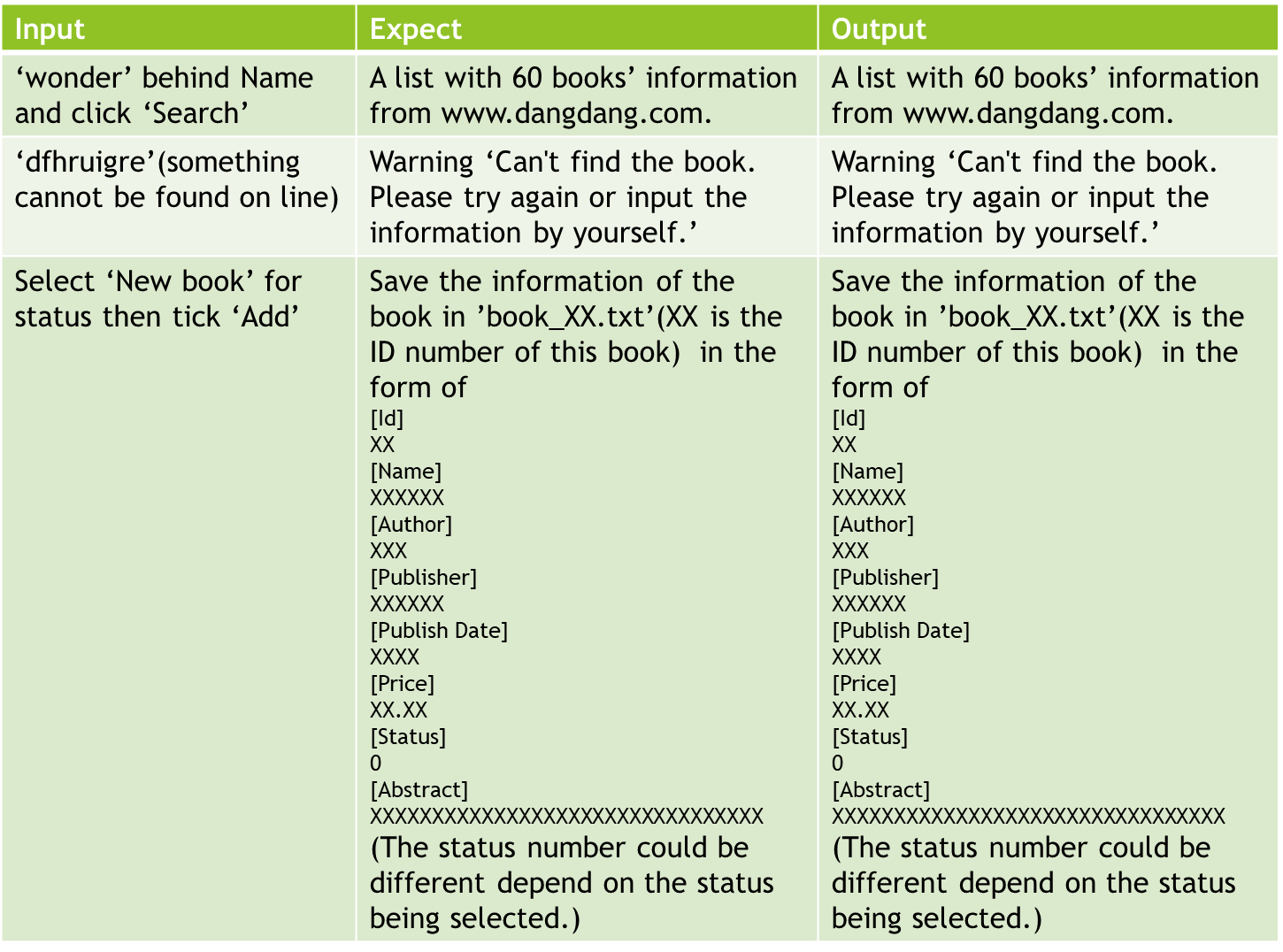
This is the interface where we enter the page number of the book which we have read just now. There is no debt that the page number should be natural number and the start page should be no more than the end page. So the testing situations are as follow.



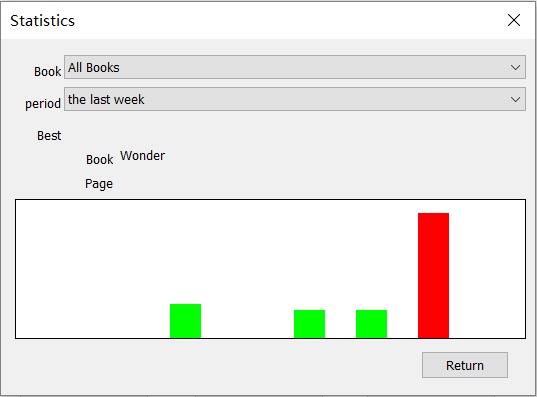
### Add Book page



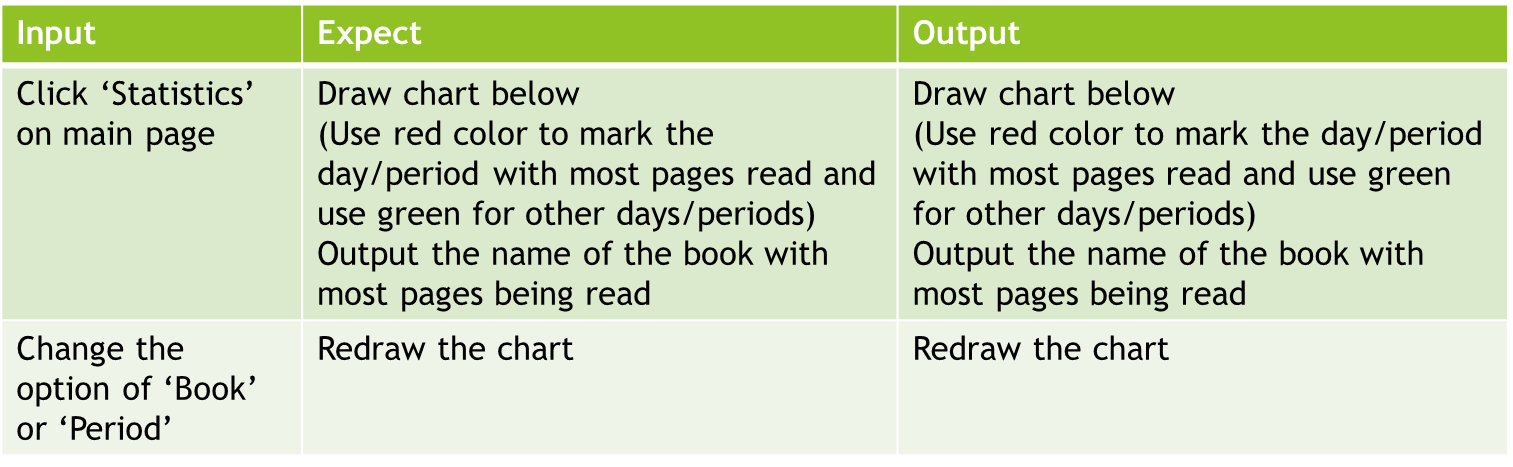
This is the interface where we enter the information of the books we own. There are two ways to enter the information as we have already discussed. For entering the information all by ourselves is exactly the same as the second half section of searching information on dangdang.com so we can just test the ‘searching’ way. There are several buttons and input fields on this page so the testing situation is a bit complicated.



### Statistic page



This is the interface where we can see the statistic for the reading trail we made in the past few days. By selecting different options of ‘book’ and ‘period’ we can select the exact range for the system to choose the records which need to be counted in. To make the bar chart more intuitionistic I use red color to mark the day/period with most pages read and use green for other days/periods.



## Result & Conclusion

The project can now achieve the following functions:

·Add book into library

Search book’s information from web site to save input time

Book’s initial status can be set

·Save reading record

Save records into file

Give reading record page number base on last record

Can change reading status manually

·Show statistics

Show all books’ reading record

Show one single book’s reading record

Show chart for different period

Show when reading maximum page, which book for all books’ statistic

The debugging status had been succeeded quite successfully, so there is few programming mistakes for me to correct.

Now, it can already meet my needs and achieved the main propose ‘record the books the user owns and output the book list for user to view them, record everyday reading trail, and statistics the pages read each day and give a brief summary’.

But there are still lots more functions that can be add to the project.

This project is just a basic one that can only record the reading trail. In fact, reading record could include more things. So the project could develop into a complete system to manage all the things about our reading. From record the book buying wish list, set the reading plan from the beginning to store and share the book report after finish the reading (which I haven’t found any independent APP could fully achieve). So I think this can be the future for my Reading Record: to build up a system for our lifelong reading.