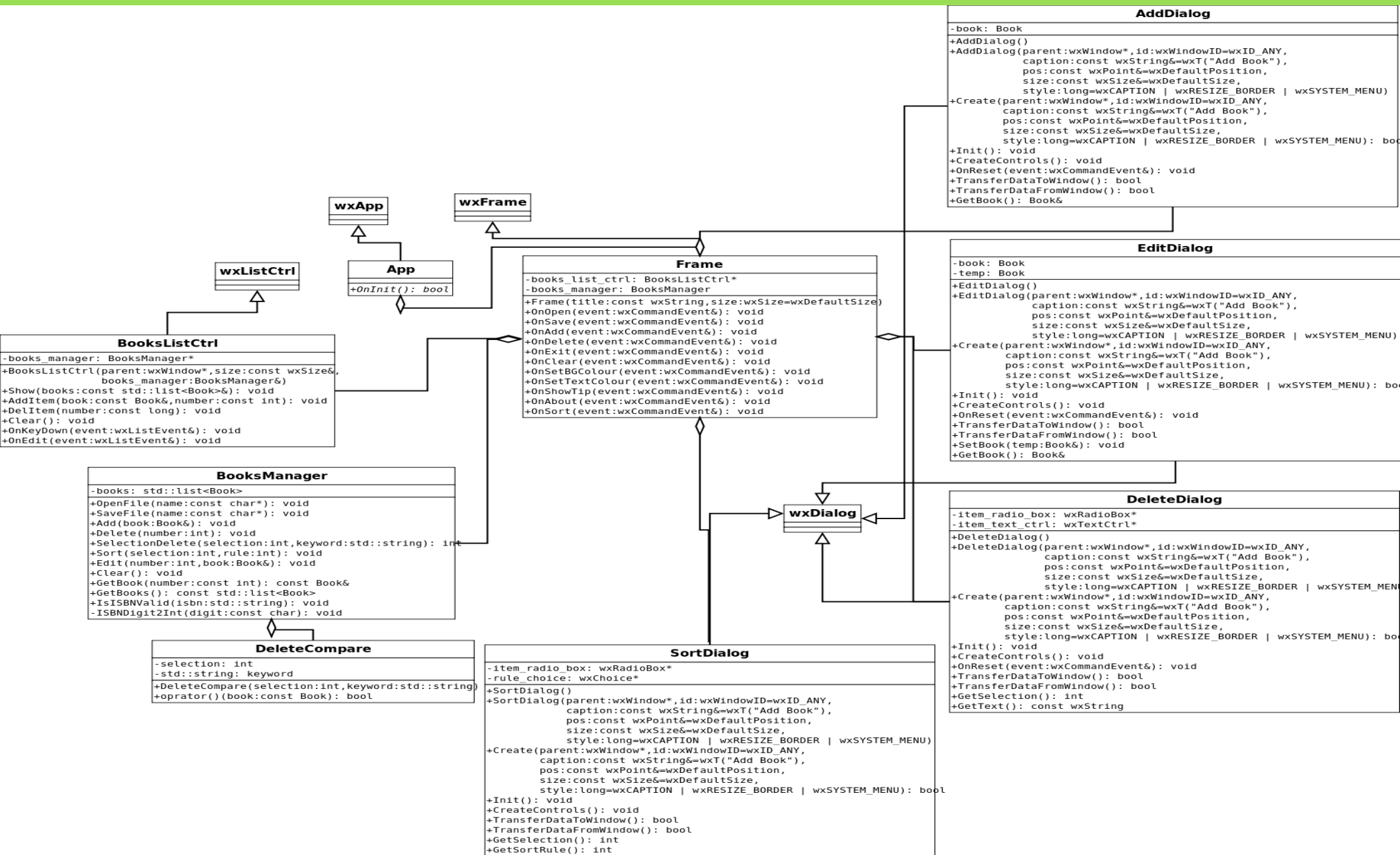


程式架構說明

- 整個程式分成 GUI、Dialog、Database。
- GUI：App、Frame。
- Dialog：Add、Delete、Edit、Sort。
- Database：BooksManager（這層可以換成常見的 MySQL 之類的資料庫）。



UML

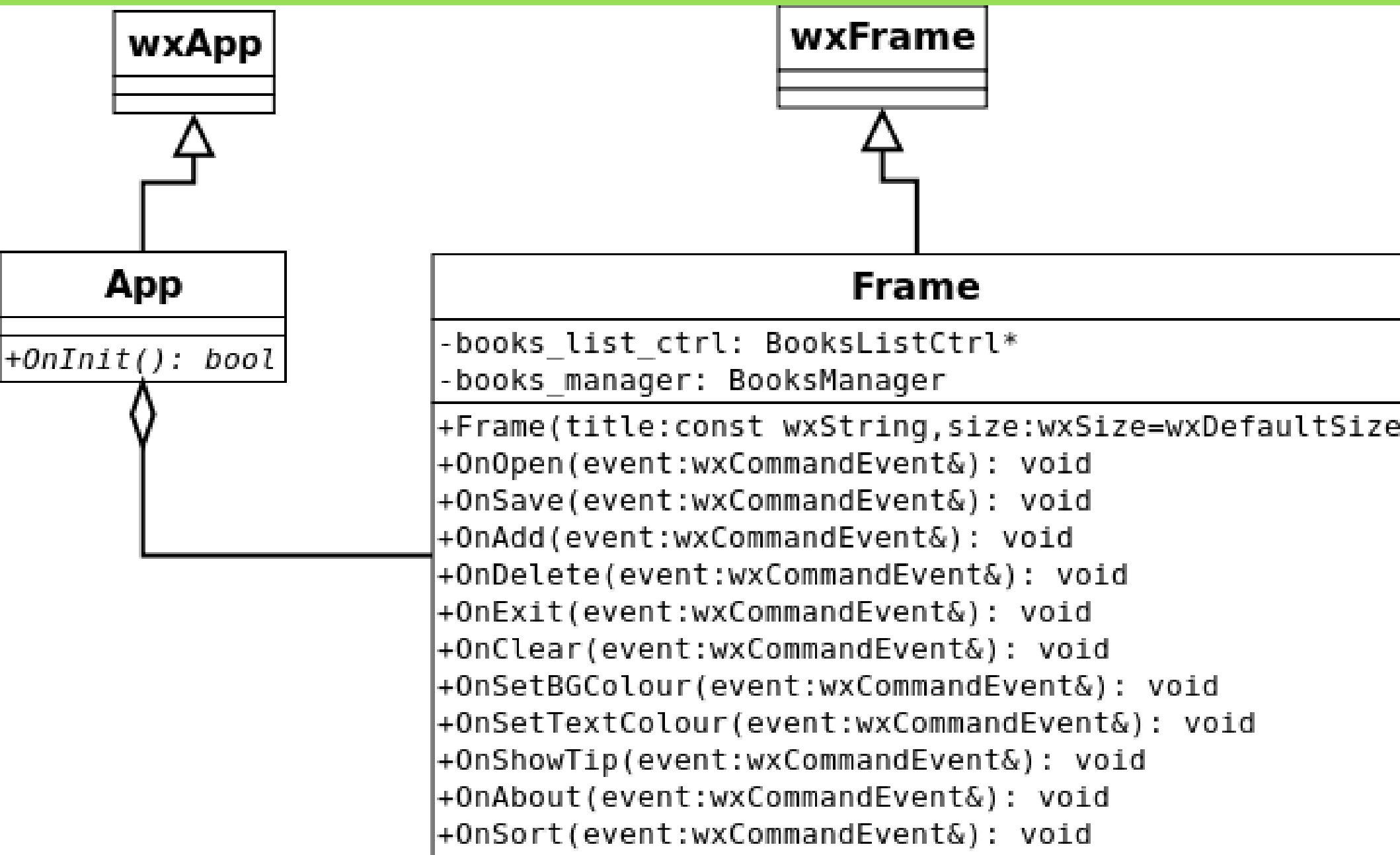


整體說明

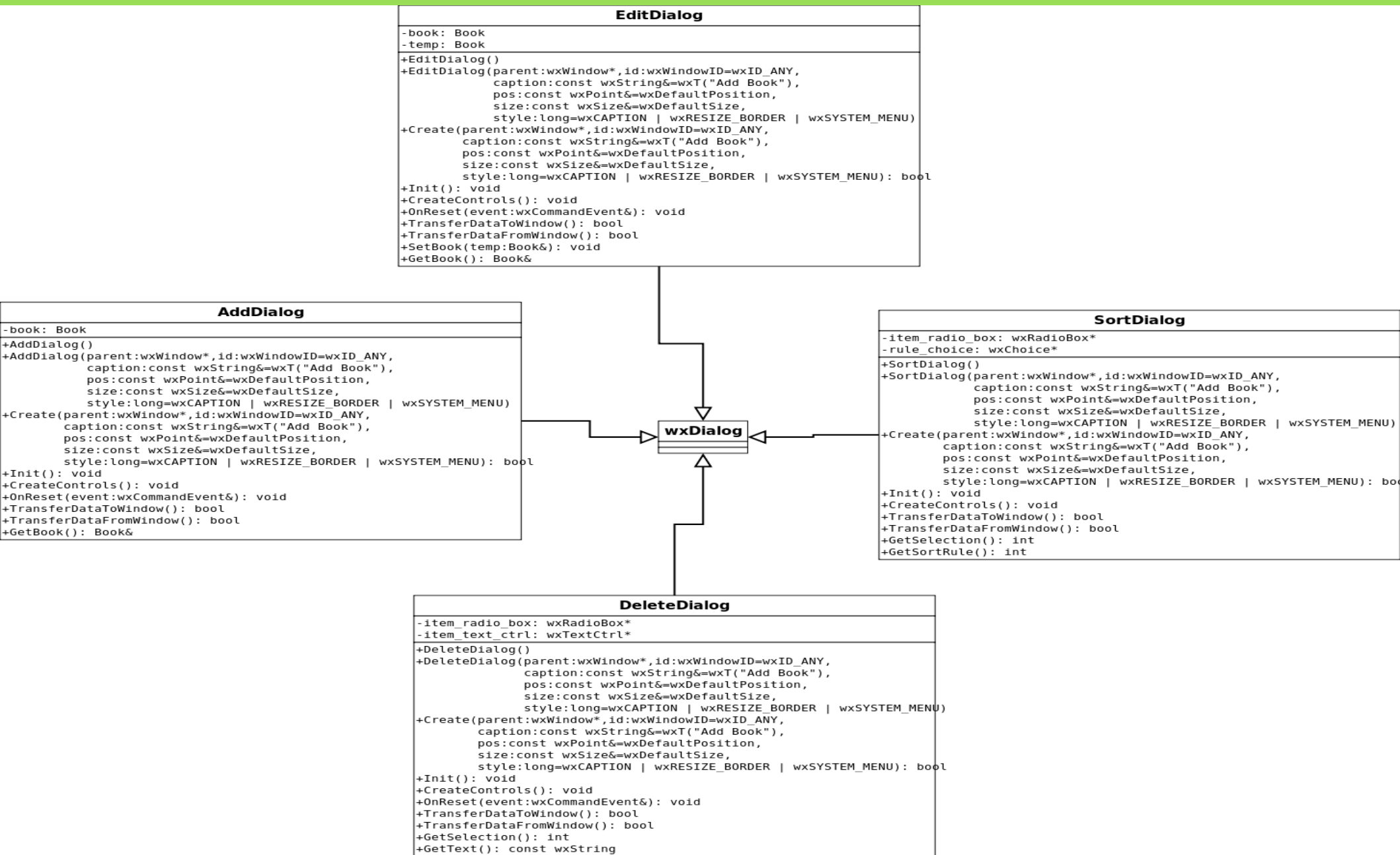
- 實體化 App 時會先校驗是否系統有相應的 dynamic library，接著實體化 Frame 把視窗以及 UI 進行 Layout 完成視窗。
- Frame 包含 BooksManager 以及各個 Dialog，當 Frame 被實體化時 BooksManager 以及各個 Dialog 會被初始化。



GUI



Dialog

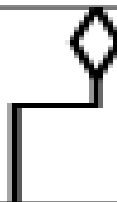


BooksManager

DeleteCompare 是使用選擇刪除時進行比對的類別。

BooksManager

```
- books: std::list<Book>
+ OpenFile(name:const char*): void
+ SaveFile(name:const char*): void
+ Add(book:Book&): void
+ Delete(number:int): void
+ SelectionDelete(selection:int,keyword:std::string): int
+ Sort(selection:int,rule:int): void
+ Edit(number:int,book:Book&): void
+ Clear(): void
+ GetBook(number:const int): const Book&
+ GetBooks(): const std::list<Book>
+ IsISBNValid(isbn:std::string): void
- ISBNDigit2Int(digit:const char): void
```



DeleteCompare

```
- selection: int
- std::string: keyword
+ DeleteCompare(selection:int,keyword:std::string)
+ oprator()(book:const Book): bool
```

BooksListCtrl

BooksListCtrl 是顯示條目 GUI 的類別。



BooksListCtrl

```
-books_manager: BooksManager*
+BooksListCtrl(parent:wxWindow*,size:const wxSize&
               books_manager:BooksManager&)
+Show(books:const std::list<Book>&): void
+AddItem(book:const Book&,number:const int): void
+DelItem(number:const long): void
+Clear(): void
+OnKeyDown(event:wxListEvent&): void
+OnEdit(event:wxListEvent&): void
```