Yeo Tong - LiBerry Project Portfolio

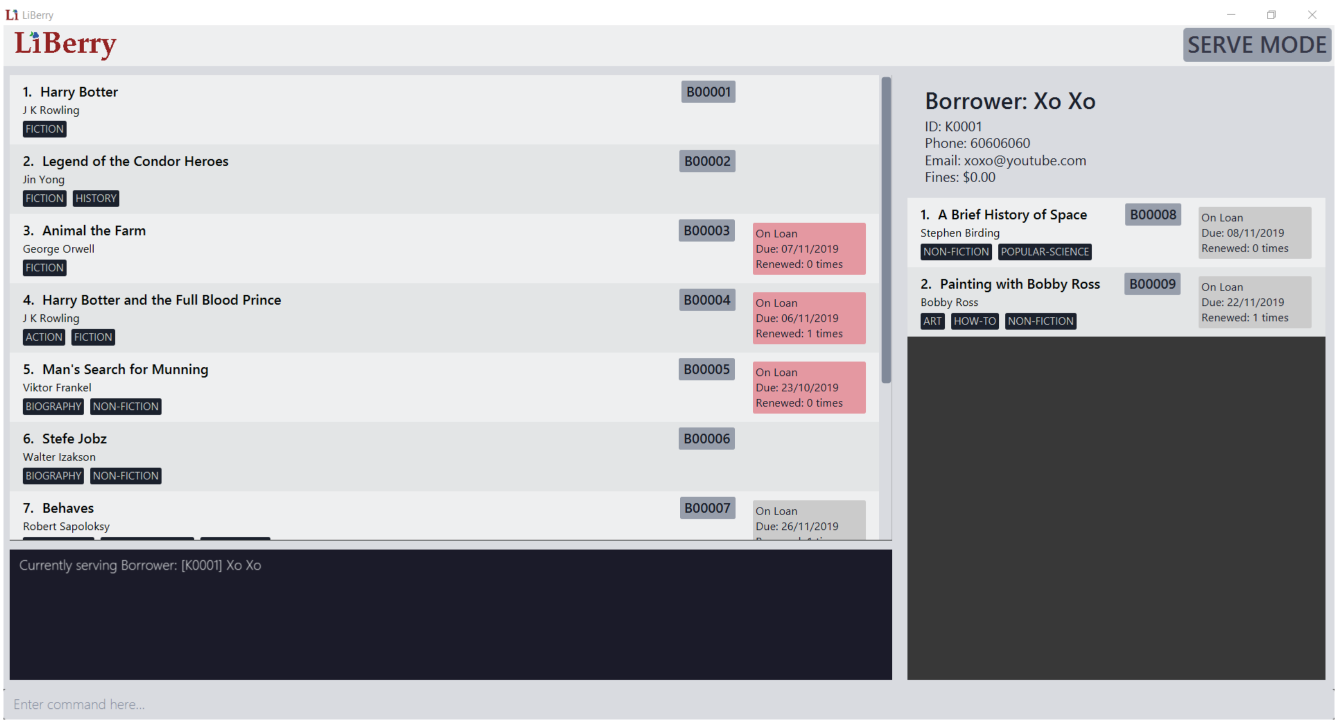
Introduction

Purpose

This portfolio is written to document my role as the **User Interface** (UI) designer and my contributions to our project, **LiBerry**. The following sections describe our project and the enhancements that I added in detail, together with relevant documentation from the user guide and developer guide.

About the project

LiBerry is a **free**, **single-user**, and **lightweight** library management system, that is designed to manage small community and private libraries. It does not require any internet connection to function and is optimized for librarians who prefer to work with a **Command Line Interface** (CLI). The **Graphical User Interface** (GUI) of **LiBerry** is created with **JavaFX** while the entire software is written in **Java**.



LiBerry

SERVE MODE

1. Harry Botter — J K Rowling — FICTION — B00001

2. Legend of the Condor Heroes — Jin Yong — FICTION HISTORY — B00002

3. Animal the Farm — George Orwell — FICTION — B00003 — On Loan, Due: 07/11/2019, Renewed: 0 times

4. Harry Botter and the Full Blood Prince — J K Rowling — ACTION FICTION — B00004 — On Loan, Due: 06/11/2019, Renewed: 1 times

5. Man's Search for Munning — Viktor Frankel — BIOGRAPHY NON-FICTION — B00005 — On Loan, Due: 23/10/2019, Renewed: 0 times

6. Stefe Jobz — Walter Izakson — BIOGRAPHY NON-FICTION — B00006

7. Behaves — Robert Sapoloksy — B00007 — On Loan, Due: 26/11/2019

Borrower: Xo Xo
ID: K0001
Phone: 60606060
Email: xoxo@youtube.com
Fines: $0.00

1. A Brief History of Space — Stephen Birding — NON-FICTION POPULAR-SCIENCE — B00008 — On Loan, Due: 08/11/2019, Renewed: 0 times

2. Painting with Bobby Ross — Bobby Ross — ART HOW-TO NON-FICTION — B00009 — On Loan, Due: 22/11/2019, Renewed: 1 times

Currently serving Borrower: [K0001] Xo Xo

Enter command here...

Figure 1. Screenshot of LiBerry user interface

The figure above shows the graphical user interface of our application, LiBerry.

Motivations

LiBerry is created to promote the setting up of private and community libraries, so as to improve literacy rates. One of the main reasons why literacy rates is low due to the lack of accessibility of knowledge. This is more prevalent in rural communities due to the lack of sources of knowledge.

Hence, in order to breach this gap in knowledge, we created **LiBerry**, which is a free, connectionless, easy to use library management system so as to encourage more people to set up their own community libraries.

Main Features

The main features of **LiBerry** are:

- Adding/Removing of books
- Registering/Unregistering of borrowers
- Loaning/Returning of books
- Searching of books

These basic features would allow librarians to manage their libraries effectively.

Key symbols

The following symbols and formatting are used in this document:

TIP	Denotes useful tips.
IMPORTANT	Denotes important details to take note of.

Summary of Contributions

This section describes the contributions that I have made to **LiBerry**.

Here is the link to the code that I contributed to the project: Code contributed: RepoSense Report

Major enhancement - Undo/Redo feature:

Description:

This feature enables users to undo all previous commands one at a time. Undone commands can be reversed one at a time by using the redo command.

Justification:

This feature makes the application much more user friendly. This is because this feature provides a convenient method for users to rectify any immediate mistakes that was made.

Highlights:

This enhancement was implemented in a way that minimises the memory usage required. In addition, this enhancement is extensible and can be applied to future commands with little overheads. This required an in-depth analysis of design alternatives and the consideration of the target audience of our application.

Secondary enhancement - Set User Settings feature:

Description:

This feature enables users to set customisable settings to our application. Some of the settings includes loan period, which is the number of days a book can be loaned out, and renew period, which is the number of the days a loan can be extended. These user settings are then stored locally so that users do not have to set them every time they open the application.

Justification:

As every library have different loan policies, this feature allows libraries to customise their own loan policies which is essential for the operation of the library. This feature also allows users to set different policies for different type of books which makes the application more flexible.

Highlights:

The most challenging part of this enhancement is making it persistent as it requires some knowledge of teh storage system.

Minor enhancement:

Added a help command that opens up a window to show the users a summary of commands available.

Other contributions:

- Project management:
  - Managed release `v1.2.1` on GitHub
- Enhancements to existing features:
  - Updated the GUI color scheme (Pull requests #33, #34)
  - Wrote additional tests for existing features to increase coverage from 88% to 92% (Pull requests #36, #38)
- Documentation:
  - Updated UI component of the Developer Guide (Pull requests #158)
- Community:
  - PRs reviewed (with non-trivial review comments): #12, #32, #19, #42

◦ Contributed to forum discussions (examples: 1, 2, 3, 4)

◦ Reported bugs and suggestions for other teams in the class (examples: 1, 2, 3)

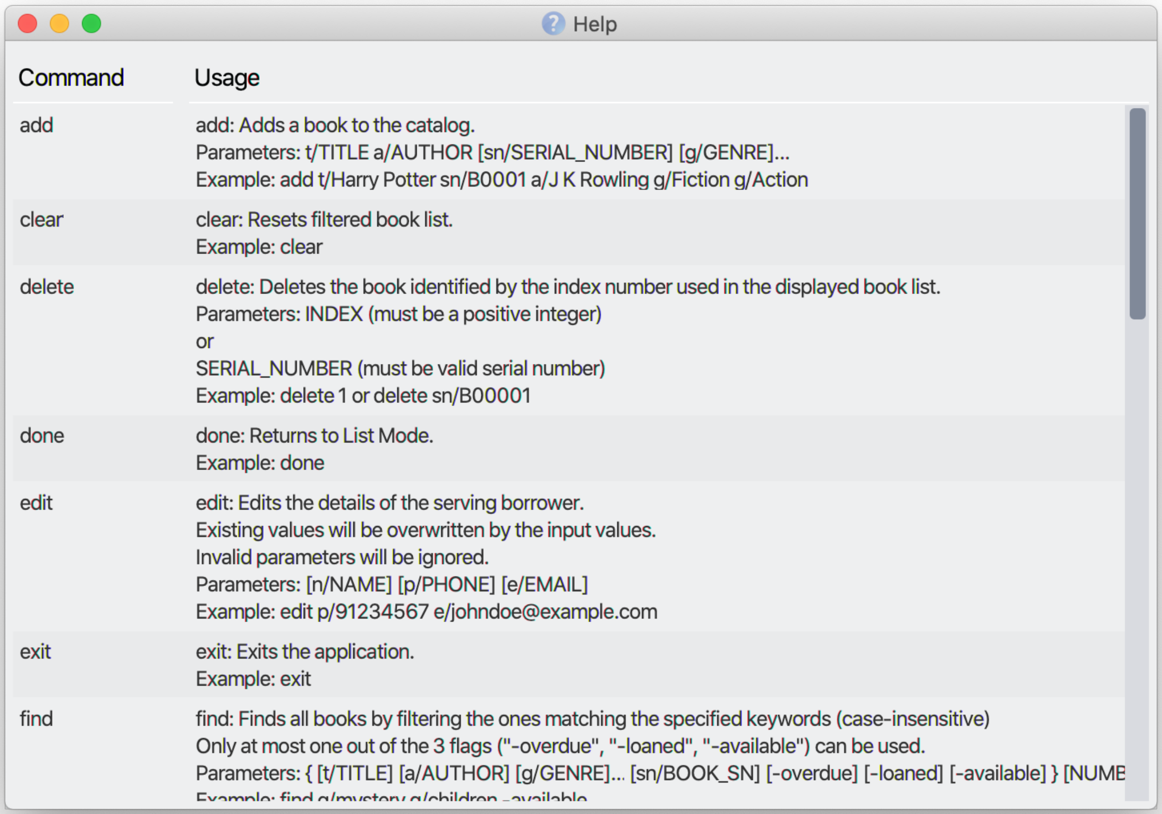
◦ Some parts of the history feature I added was adopted by several other class mates (1, 2)

Contributions to the User Guide

Given below are sections I contributed to the User Guide. They showcase my ability to write documentation targeting end-users.

Viewing help : `help`

Opens up a help window, which shows a summary of the list possible commands that you can execute.



Help

Command	Usage
add	add: Adds a book to the catalog. Parameters: t/TITLE a/AUTHOR [sn/SERIAL_NUMBER] [g/GENRE]... Example: add t/Harry Potter sn/B0001 a/J K Rowling g/Fiction g/Action
clear	clear: Resets filtered book list. Example: clear
delete	delete: Deletes the book identified by the index number used in the displayed book list. Parameters: INDEX (must be a positive integer) or SERIAL_NUMBER (must be valid serial number) Example: delete 1 or delete sn/B00001
done	done: Returns to List Mode. Example: done
edit	edit: Edits the details of the serving borrower. Existing values will be overwritten by the input values. Invalid parameters will be ignored. Parameters: [n/NAME] [p/PHONE] [e/EMAIL] Example: edit p/91234567 e/johndoe@example.com
exit	exit: Exits the application. Example: exit
find	find: Finds all books by filtering the ones matching the specified keywords (case-insensitive) Only at most one out of the 3 flags ("-overdue", "-loaned", "-available") can be used. Parameters: { [t/TITLE] [a/AUTHOR] [g/GENRE]... [sn/BOOK_SN] [-overdue] [-loaned] [-available] } [NUME Example: find g/mystery g/children -available

Figure 2. Screenshot of Help Window

The figure above shows a screenshot of the help window. The left side of the window shows the command while the right side of the window gives a description of the command and a usage example.

Format: `help`

Undoing mistakes: `undo`

Undoes the previous command/action. This command only works for commands that modifies the catalog, loan records, borrower records or user settings.

Undoable Commands:

- `add`, `delete`, `edit`, `loan`, `register`, `renew`, `return`, `set`, `toggleui` and `unregister`.

IMPORTANT	After every `serve`, `done` or `pay` command, all previous state would be cleared. This means that you would not be able to undo to the state before the `serve`, `done` or `pay` command.

TIP	Check the commands you made and ensure that they are correct before entering a `serve`, `done` or `pay` command.

Format: `undo`

Redoing undone commands : redo

Redoes the most recent command that was undone. This command only works if there are undone commands.

IMPORTANT	Once a new undoable command is entered, you may not redo previously undone commands.

Format: `redo`

Setting User Settings: `set`

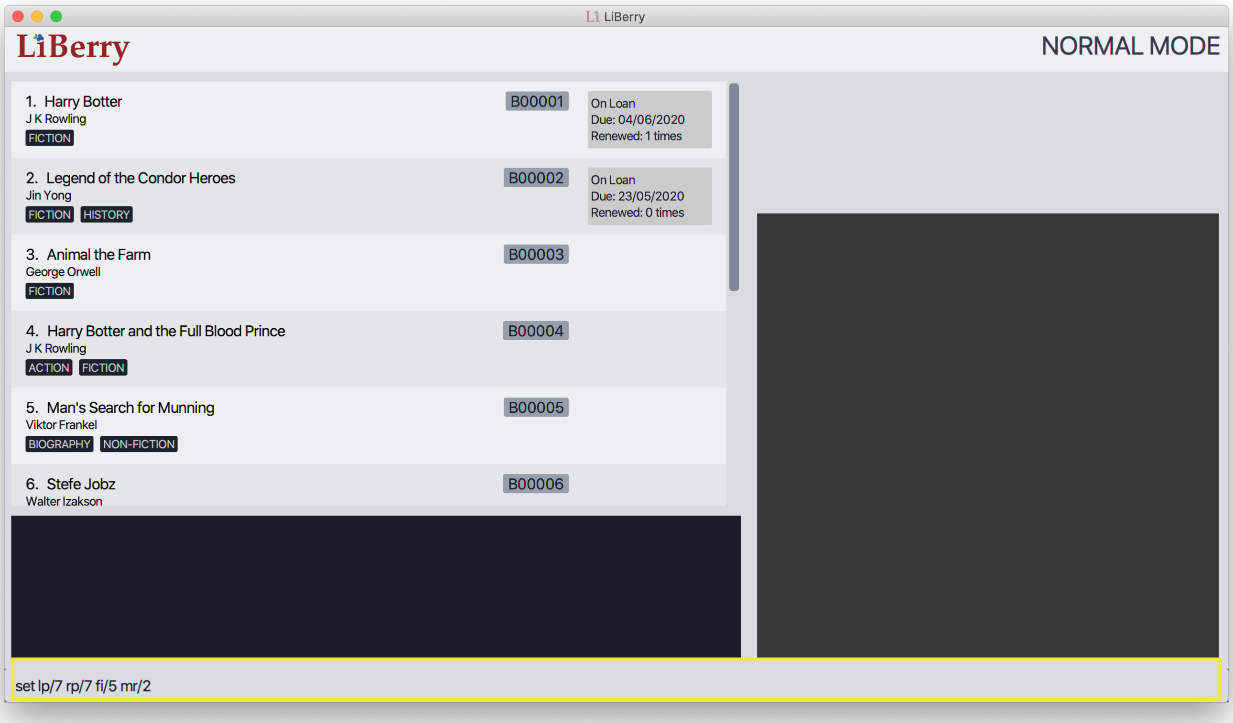
Sets the user settings for loan period (in days), renew period (in days), fine increment (in cents) and maximum renews allowed.

Format: `set [lp/LOAN_PERIOD] [rp/RENEW_PERIOD] [fi/FINE_INCREMENT] [mr/MAX_RENEWS] `

- Updates the user settings with the specified `LOAN_PERIOD`, `RENEW_PERIOD`, `FINE_INCREMENT` and `MAX_RENEW`.
- All the fields that are specified must be a positive integer.
- If none of the fields are specified, the current user settings would be displayed.
- `LOAN_PERIOD` refers to the number of days that a book can be loaned out for.
- `RENEW_PERIOD` refers to the number of days that the loan can be extended for.
- `FINE_INCREMENT` refers to the amount of cents charged per day for each overdue book.
- `MAX_RENEWS` refers to the maximum amount renewals that can be made per loaned out book.

Examples:

- `set`
  Shows the current user settings.
- `set lp/7 rp/7 fi/5 mr/2`
  Sets the loan period to 7 days, renew period to 7 days, fine increment to 5 cents per day and maximum renews allowed to 2.



LiBerry

LiBerry NORMAL MODE

1. Harry Botter
J K Rowling
FICTION
B00001
On Loan
Due: 04/06/2020
Renewed: 1 times

2. Legend of the Condor Heroes
Jin Yong
FICTION HISTORY
B00002
On Loan
Due: 23/05/2020
Renewed: 0 times

3. Animal the Farm
George Orwell
FICTION
B00003

4. Harry Botter and the Full Blood Prince
J K Rowling
ACTION FICTION
B00004

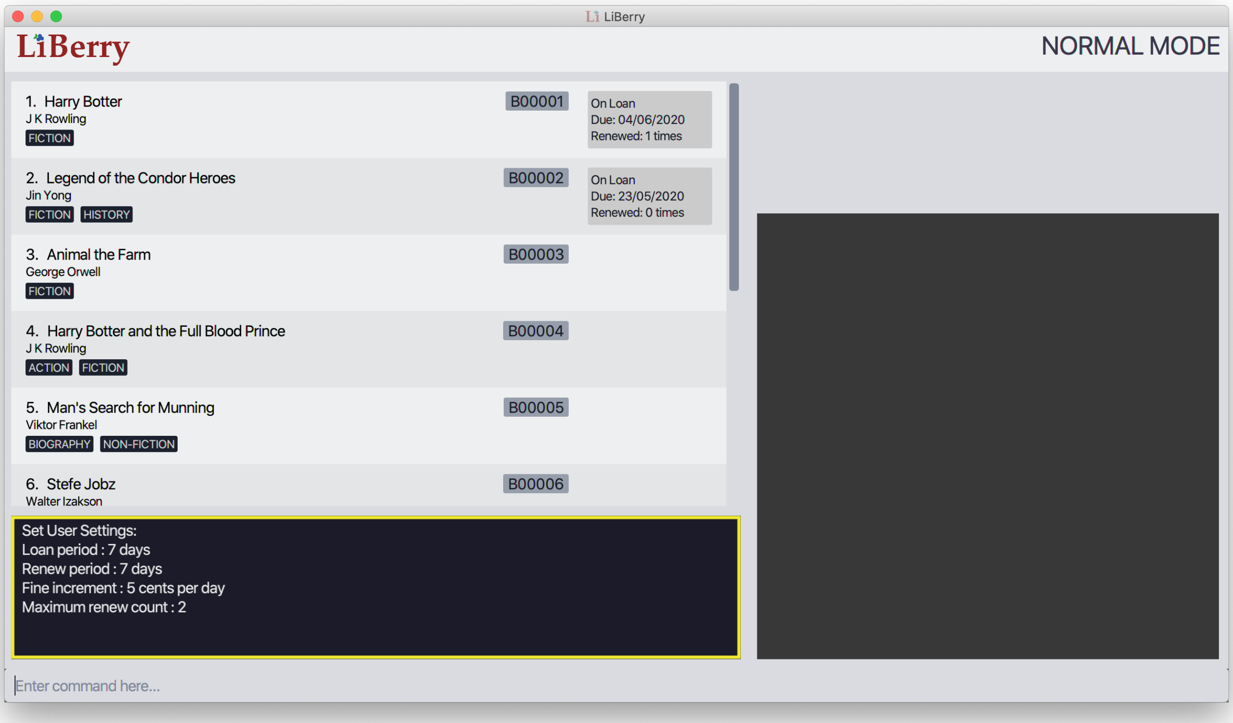
5. Man's Search for Munning
Viktor Frankel
BIOGRAPHY NON-FICTION
B00005

6. Stefe Jobz
Walter Izakson
B00006

```
set lp/7 rp/7 fi/5 mr/2
```

Figure 3. User interface before set command is executed.

The figure above shows the user interface before the set command is executed. The yellow box shows the set command that is being entered.



```
Set User Settings:
Loan period : 7 days
Renew period : 7 days
Fine increment : 5 cents per day
Maximum renew count : 2
```

Figure 4. User interface after set command is executed.

After entering the set command, the user settings would be updated. The updated user settings will then be displayed in the result display as shown in the figure above.

Contributions to the Developer Guide

Given below are sections I contributed to the Developer Guide. They showcase my ability to write technical documentation and the technical depth of my contributions to the project.