Lin Yuting - Project Portfolio Page

LiBerry

Overview

With the motivation to raise literacy rate in the world and encourage more setups of community libraries, my team has developed Liberry - a desktop software for managing community and private libraries. The target users are mainly librarians looking to set up or are already running private or community libraries, especially those in under-developed communities who cannot afford commercial versions of such software.

LiBerry can execute the basic functions of library management:

- Addition and deletion of books in the catalog
- Registering Borrowers and editing their particulars
- Loan, return, renew and pay fine functions
- Searching through books in the catalog by filters

As part of the requirements of the project, we were given the source code for a common AddressBook application and were tasked to morph it to something else that is useful. We were also limited to designing software that takes in the majority of user input through a Command Line Interface (CLI), though displays information through a Graphical User Interface (GUI).

This is what the GUI looks like:

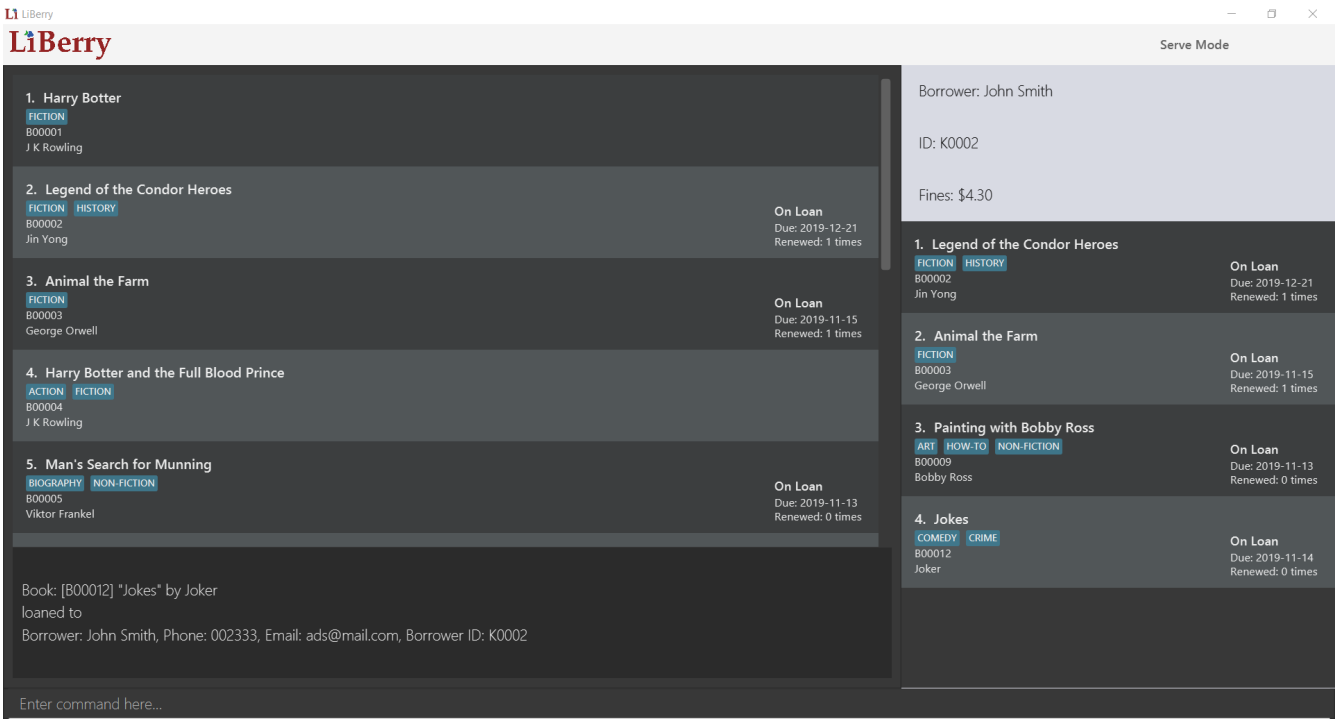


Figure 1. Graphical User Interface (GUI) for **LiBerry**

My role was to:

1. configure the basic AddressBook software to create a `Borrower` class and its related classes.
2. implement borrower-related commands such as the **Register Command**, **Serve Command**, **Done Command**,
EditBorrowerCommand and **UnregisterCommand**.

Summary of Contributions

- **Major enhancement**: added **the ability to unregister a borrower**
    - What it does: allows the user to unregister previously registered borrowers.
    - Justification: This feature improves the product significantly because a borrower can be no longer using the library and the app should unregister him or her from the library system.
    - Highlights: This enhancement affects existing commands and commands to be added in future. It required an in-depth analysis of design alternatives. The implementation too was challenging as it required changes to many data in the storage.
- **Minor enhancement**: added a Borrower ID generator that auto generates a borrower's ID.
- **Code contributed**:
[RegisterCommandParser Class]
[RegisterCommand Class]
[ServeCommandParser Class]
[ServeCommand Class]
[Borrower Class]
[Name Class]
[BorrowerId Class]
[Phone Class]
[Email Class]
[BorrowerIdGenerator Class]
- **Other contributions**:
    - Project management
        - Managed the release of `v1.2` on GitHub. In `v1.2`, we have the following features:
            - Set default user settings
            - Exit the application
            - Register a borrower into the library System
            - Setting the application to serving mode
            - Exiting the serving mode
            - Loaning a book
            - Returning a book
            - Deleting a book
    - Enhancements to existing features:

        - Wrote additional tests for existing features to increase coverage from 88% to 92% (Pull requests #36, #38)
    - Documentation:
        - Improve user stories of the User Guide: #103
        - Updating Developer Guide: #165, #183
    - Community:
        - PRs reviewed (with non-trivial review comments): #12, #32, #19, #42
        - Contributed to forum discussions (examples: 1, 2, 3, 4)

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.

Registering a new borrower: `register`

Registers a new borrower to the library records. A unique ID associated with the borrower will automatically be generated and displayed. Borrowers are expected to know his ID in order for loans to be processed.
Format: `register n/NAME p/PHONE_NUMBER e/EMAIL`

Example:

- `register n/matt p/83938249 e/matt@damon.com`
  Registers a new borrower called "matt", with phone number "83938249" and email "matt@damon.com" to LiBerry.

Unregistering a borrower: `unregister`

Unregisters and removes a borrower with the given borrower ID from the library records.
Format: `unregister id/BORROWER_ID`

Example:

- `unregister id/K0001`
  Deletes the borrower with the borrower ID `id/K0001`

Using Serve Mode

The Serve Mode is for librarians to serve borrowers. All commands in Serve Mode are done on a specific borrower currently served by the librarian. All commands in Normal Mode can be used in Serve Mode too.

Entering Serve Mode: `serve`

Enters Serve Mode. All commands/actions will be done on this specific borrower. A list of the borrower's currently loaned books and their serial numbers will be displayed.
Format: `serve id/BORROWER_ID`

Example:

- `serve id/K0001`
  Enters save mode to serve a borrower with the ID `K0001`

Exiting Serve Mode: `done`

Exits Serve Mode.
Format: `done`

Editing a borrower: `edit`

Edit borrower's particulars.
Format: `edit { [n/NAME] [p/PHONE_NUMBER] [e/email] }`

> - Edits the currently serving borrower's particulars.
> - At least one of the optional fields must be provided.
> - Existing values will be updated to the input values.

Examples:

- `edit p/91234567 e/jane@austen.com`
  Edits the phone number and borrower's email address to be `91234567` and `jane@austen.com` respectively.
- `edit n/Betsy Crower`
  Edits the name of the borrower to be `Betsy Crower`.

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.

Register borrower feature

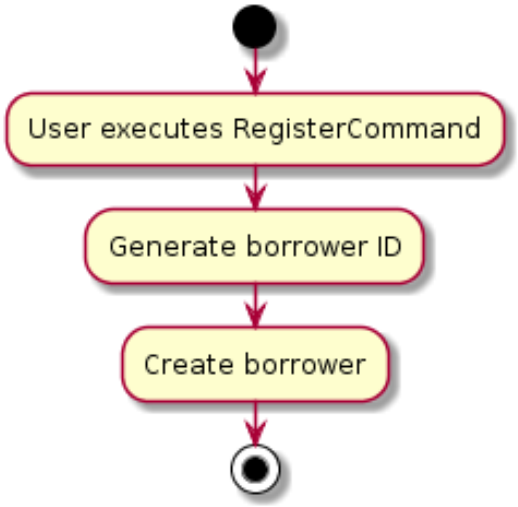
Implementation

The `register` borrower feature is facilitated by `BorrowerRecords`. The BorrowerRecords stores a list of borrowers, representing the borrowers registered into the library system. The command to register a borrower into the library system is as followed:

`register n/NAME p/PHONE_NUMBER e/EMAIL`

Given below is an activity diagram of a borrower being registered into the Borrower Records of the library.

Activity Diagram for registering a borrower



Given below is a class diagram of a book.

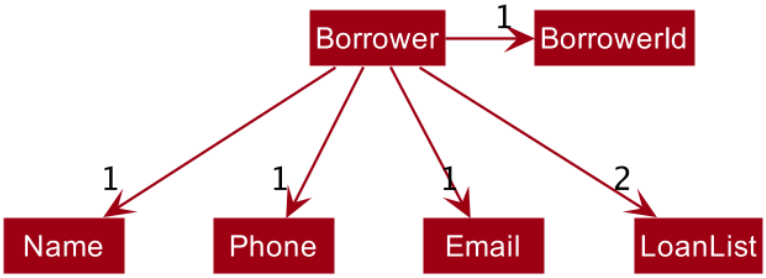


Figure 2. Class Diagram for `Borrower`

Given below is the object diagram of a newly registered borrower.

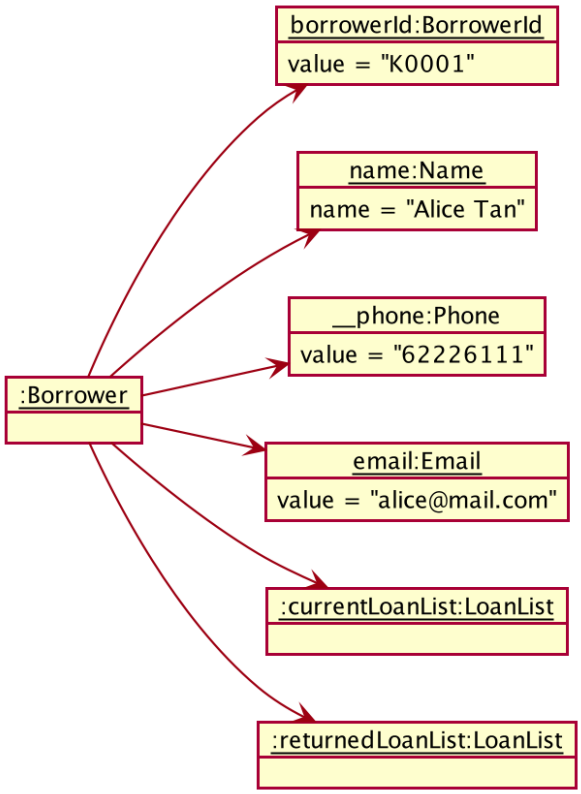


Figure 3. Object Diagram for `Borrower`

Design Considerations

Aspect: Generating a unique borrower ID

Every time a new borrower is being registered, the system will automatically generate a borrower ID for the borrower which the borrower will have to use every time the borrower borrows books from the library. Initially, what we proposed is that, every time a new borrower is being registered into the system, we find the size of the list of borrowers, we add 1 and set it as the borrower ID of the new borrower.

Eg: There are 100 borrowers in the system. The new borrower's ID will be "K0101".

However, we decided to implement a new function, which is to allow borrowers to be removed from the library system. Therefore, this method does not work anymore. So we decided to change to generate the new ID based on the first-found available ID.