Public Library System handbook

**For librarians**

**Catalog**

The catalog object (an object of the class Catalog) contains a dictionary of all authors and books. It also contains a search function for customers and librarians.

**Adding new books**

Books are saved as bookItem objects in a dictionary inside the catalog object. In order to add a new book you will have to add it to the dictionary. This dictionary takes the title of the book as the key, and the object of class ‘Book’ as a value. So in order to add it, this format will have to be followed:

catalog.books[str title] = Book(str author, str country, str imagelink, str language, str link, int pages, str title, int year)

Catalog.books[str title].addCopy()

The last line to add copies would be repeated as many times as necessary.

Or if there is an already existing book object (in this case the object is referenced under the variable existingbook)

Catalog.books[existingbook.title] = existingbook

Catalog.books[str title].addCopy()

For example, to add the book Hamlet with 3 copies:

catalog.books[‘Hamlet’] = Book(‘William Shakespeare’, ‘England’,’ images/hamlet.jpg’, ’English’, ‘<https://en.wikipedia.org/wiki/Hamlet>’, 432,’Hamlet’,1603)

Catalog.books[‘Hamlet’].addCopy()

Catalog.books[‘Hamlet’].addCopy()

Catalog.books[‘Hamlet’].addCopy()

**Adding new customers**

Customers are saved as Customer objects inside a dictionary called allCustomers in the object called loan\_administration, with the key being the full name of the customer, and the value being an object of the class Customer. To add it, this format will have to be followed:

loan\_administration.addNewCustomer(str number, str gender, str firstName, str surName, str streetadress, str zipcode, str city, str emailAddress, str username, str telephonenumber)

For example:

loan\_administration.addNewCustomer(str(len(loan\_administration.allCustomers)), ‘male’, ‘Ronan’, ‘van Kessel’, ‘Nes 4’, ‘2642 LA’, ‘Pijnacker’, ‘0972466@hr.nl’, ‘Ronan2019’, ‘06-1235678’)

**Making a backup**

To make a backup of the system a function the loan\_adminstration object will have to be accessed:

loan\_administration.makeBackup()

**Loading the backup**

To load a backup a function the loan\_adminstration object will have to be accessed:

loan\_administration.restoreFromBackup()

**For customers**

**Becoming a customer**

This part of the guide is for customers that are already registered in the loan\_administration. If you are not, please refer to the ‘Adding new customers’ section of the for Librarians part to be registered, or ask a librarian.

**Loaning a book**

To loan a book you will have to use a function that takes a book object as argument. This function will check if there are any copies available first. This function is accessed through the customer object, which is saved in a dict in the loan\_administration object. So, this format will have to be followed:

loan\_administration.allCustomers[str fullName].loanBook(catalog.books[str title])

For example, for Hisham Altink to loan Hamlet, this will have to be typed

loan\_administration.allCustomers['Hisham Altink'].loanBook(catalog.books['Hamlet'])

**Checking current loaned books**

To see a list of the objects of the books a customer has loaned, use this format:

loan\_administration.allCustomers[str fullName].myLoans

**Searching for books**

Searching for books is done through the catalog object. Searching for a book is not case sensitive. You can use multiple search arguments at once. To search use the following format:

catalog.search(str or int \*value)

Where \*value can be any number of values. For example, to look for all english books written by a ‘William’, this would be what you write and what you get in return:  
  
catalog.search('English','William')

Console shows:  
Title: The Sound and the Fury

Author: William Faulkner

Language: English

Year 1929

Pages: 326

Available copies: 3

Title: King Lear

Author: William Shakespeare

Language: English

Year 1608

Pages: 384

Available copies: 1

Title: Othello

Author: William Shakespeare

Language: English

Year 1609

Pages: 314

Available copies: 3

Title: Absalom, Absalom!

Author: William Faulkner

Language: English

Year 1936

Pages: 313

Available copies: 1

Title: Hamlet

Author: William Shakespeare

Language: English

Year 1603

Pages: 432

Available copies: 2

**Returning books**

To return a book, another function in the customer object needs to be accessed. The function takes the loanItem object as an argument, which can be found in the customer objects myLoans list. This format will have to be followed.

loan\_administration.allCustomers[str fullName].returnBook(loan\_administration.allCustomers[str fullName].myLoans[int book number in myLoans list])

For example, if we want to return book 0 from Hisham Altinks list of books (in his variable myLoans), we would type:

loan\_administration.allCustomers['Hisham Altink'].returnBook(loan\_administration.allCustomers['Hisham Altink'].myLoans[0])