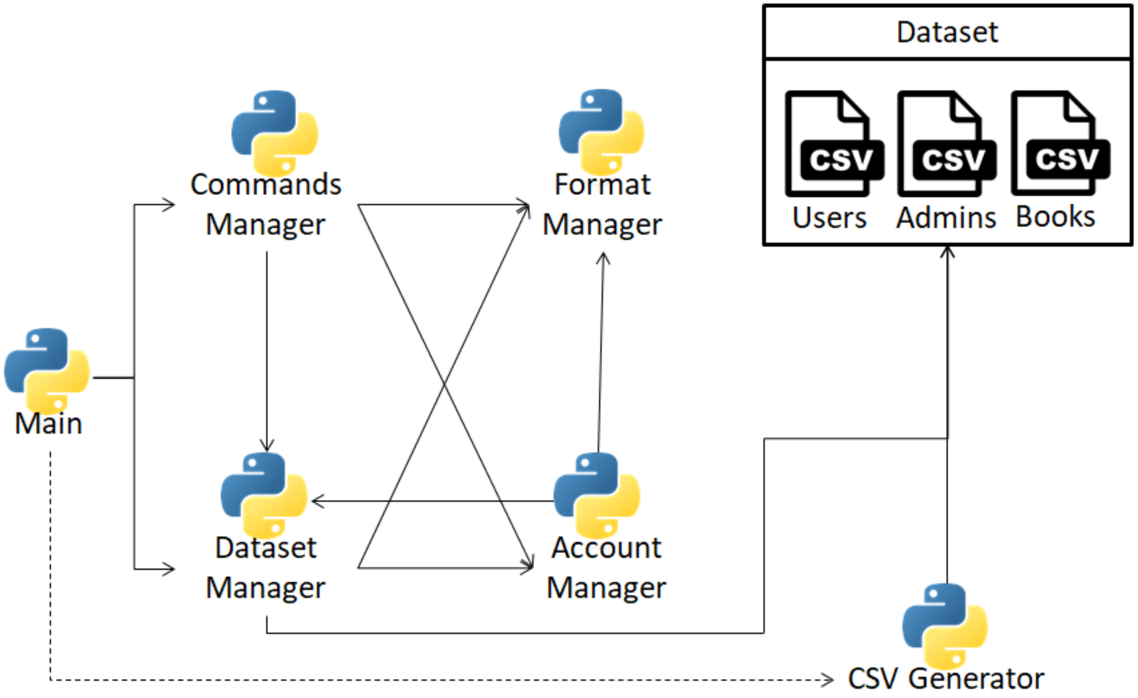
Library Management Software - Python

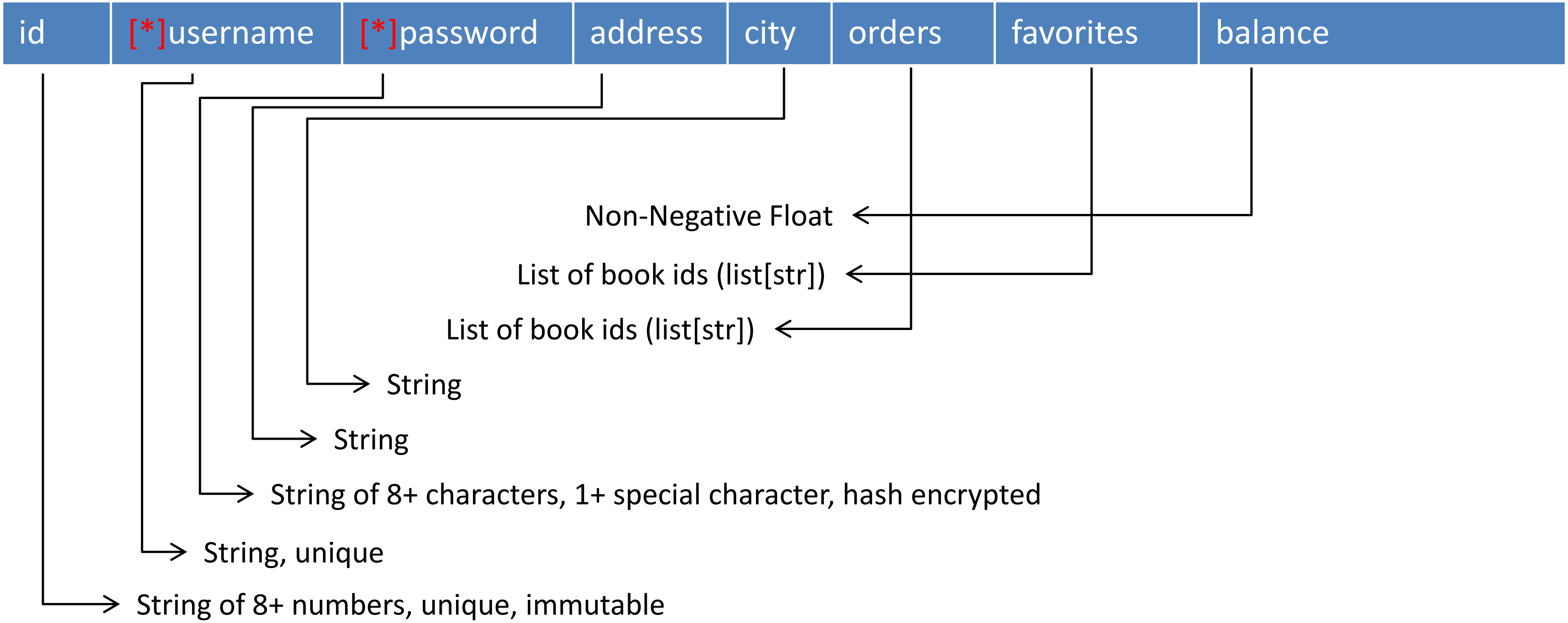
This project has been split up to 6 different files, each for a different purpose. 4 out 6 are the managers, the 5th file is the main function itself, and the 6th file is not used during the program's execution phase, and is instead a dataset creation tool that was used for the creation of the project's datasets.

The main shape of this project is the diagram on the right, which shows the files and how the communicate with each other. The arrows indicate which files the current manager is taking functions from. We can also clearly see that the Commands Manager takes methods from all other managers, whereas the Format Manager is only providing its functions to the other files.



The Dataframes this project is based on are structured like so:

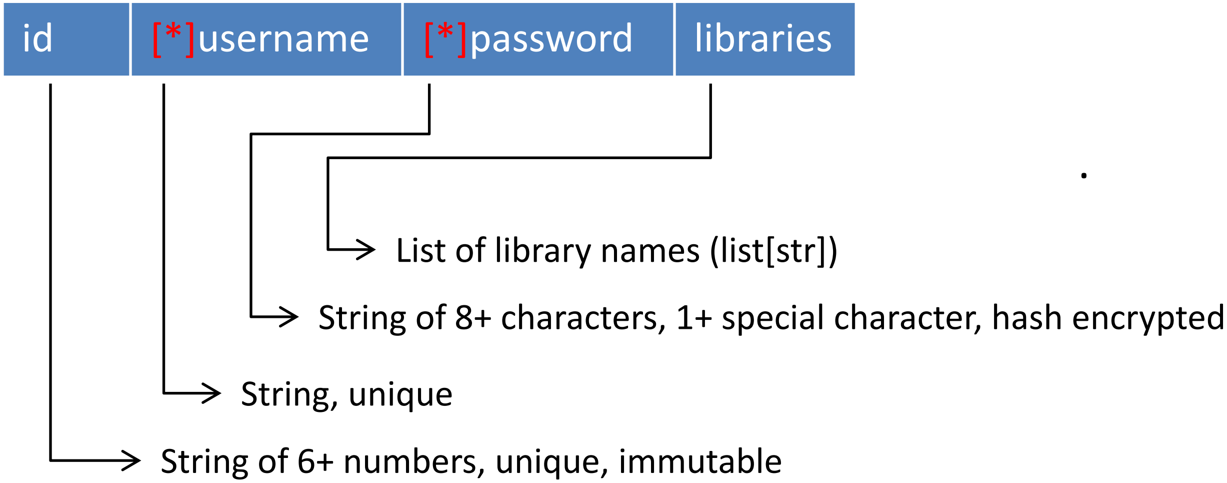
Users



id	[*]username	[*]password	address	city	orders	favorites	balance
String of 8+ numbers, unique, immutable	String, unique	String of 8+ characters, 1+ special character, hash encrypted	String	String	List of book ids (list[str])	List of book ids (list[str])	Non-Negative Float

*The fields with [*] are the mandatory fields during the creation of an account. All other fields can either be left blank, or are filled out automatically (like the id and balance)

Admins



id	[*]username	[*]password	libraries

List of library names (list[str])

String of 8+ characters, 1+ special character, hash encrypted

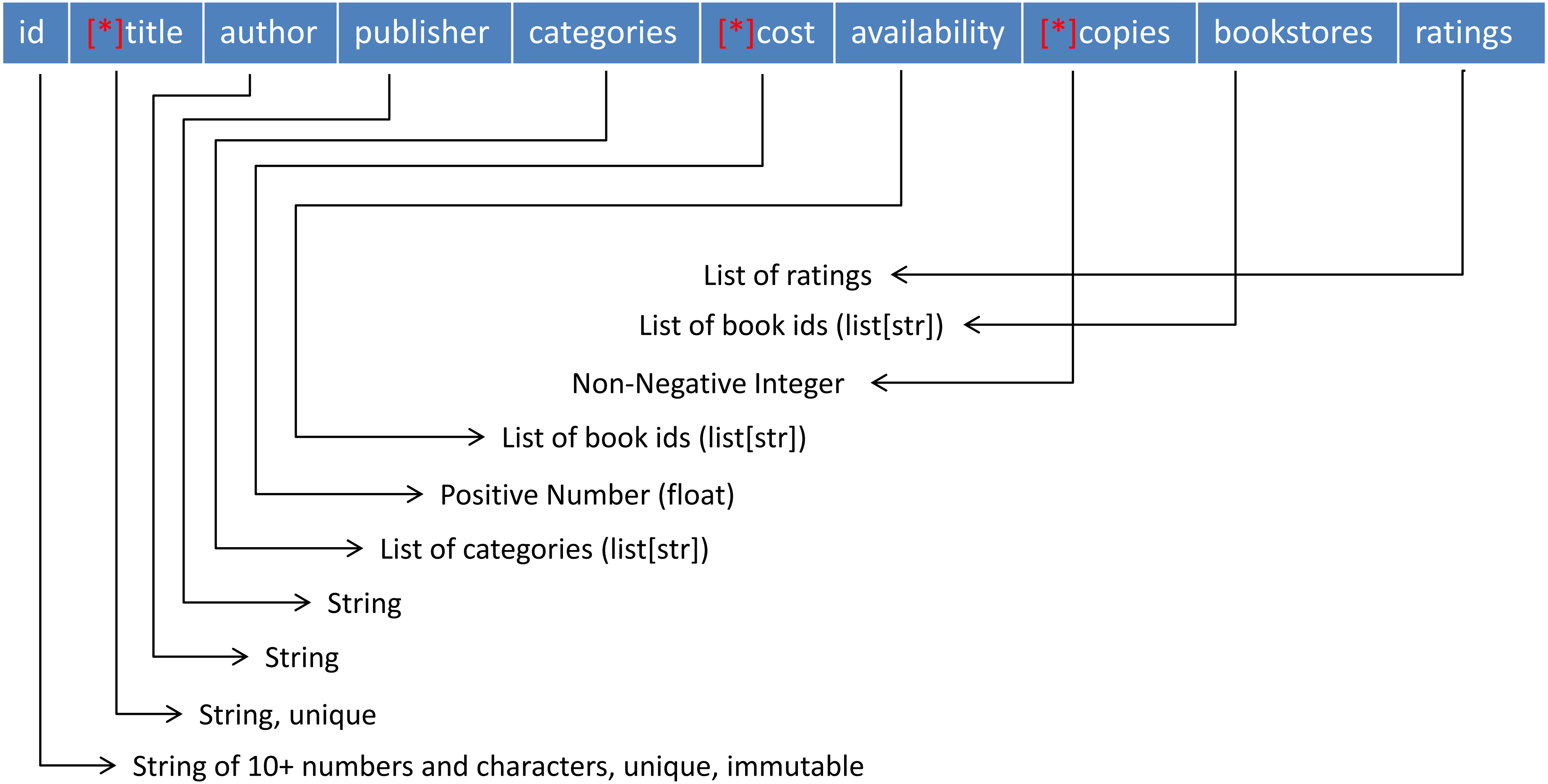
String, unique

String of 6+ numbers, unique, immutable

Books

commenter	last edited by	rating	comment1	comment2	…
user id	user/admin id or ‘0’	float[0, 10]	string	string	…

ratings format



id	[*]title	author	publisher	categories	[*]cost	availability	[*]copies	bookstores	ratings

List of ratings

List of book ids (list[str])

Non-Negative Integer

List of book ids (list[str])

Positive Number (float)

List of categories (list[str])

String

String

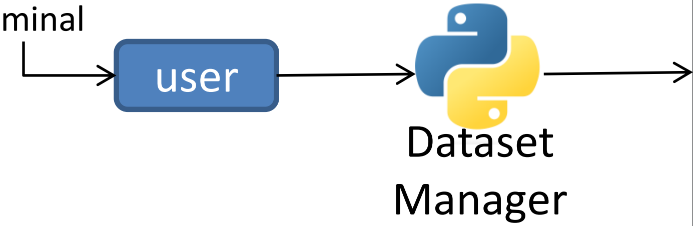
String, unique

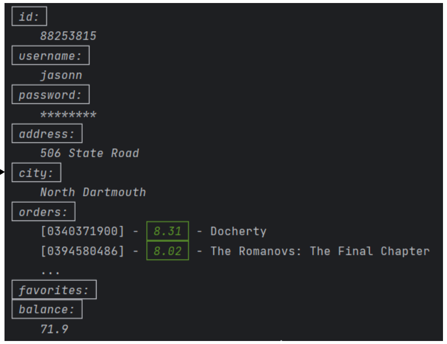
String of 10+ numbers and characters, unique, immutable

During the start of the program, the Dataset Manager imports all the dataframes used by the program, checking for duplicate values or invalid formatting.

In addition to a few utility functions, the Dataset Manager is responsible for everything that has to do with said datasets, like:

- Saving / Loading the dataframes
- Formatting Entity Values on the terminal
- Importing Entities
- Retrieving Entities
- Deleting Entities
- Updating values in an entity's field



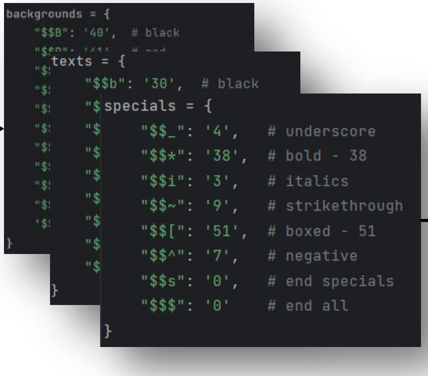


```
id:
    88253815
username:
    jasonn
password:
    ********
address:
    506 State Road
city:
    North Dartmouth
orders:
    [0340371900] - 8.31 - Docherty
    [0394580486] - 8.02 - The Romanovs: The Final Chapter
    ...
favorites:
balance:
    71.9
```

The Format Manager has 2 methods used all across the program. These methods are enhanced versions of the print and input commands, allowing for special codes inside the text in order to change its appearance. A few examples are as follows:

"$$[Boxed "
"$$F $$b $$i Header"
"$$g Colored"

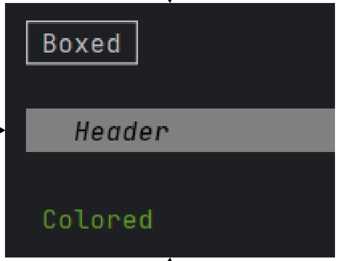
Format Manager



```
backgrounds = {
    "$$B": '40',  # black
    "$$R": '41',  # red
    "$
    "$
    "$
    "$
    "$
    "$
    "$
    "$
    '$
}
```

```
texts = {
    "$$b": '30',  # black
    "$
    "$
    "$
    "$
    "$
    "$
    "$
    "$
}
```

```
specials = {
    "$$_": '4',   # underscore
    "$$*": '38',  # bold - 38
    "$$i": '3',   # italics
    "$$~": '9',   # strikethrough
    "$$[": '51',  # boxed - 51
    "$$^": '7',   # negative
    "$$s": '0',   # end specials
    "$$$": '0'    # end all
}
```



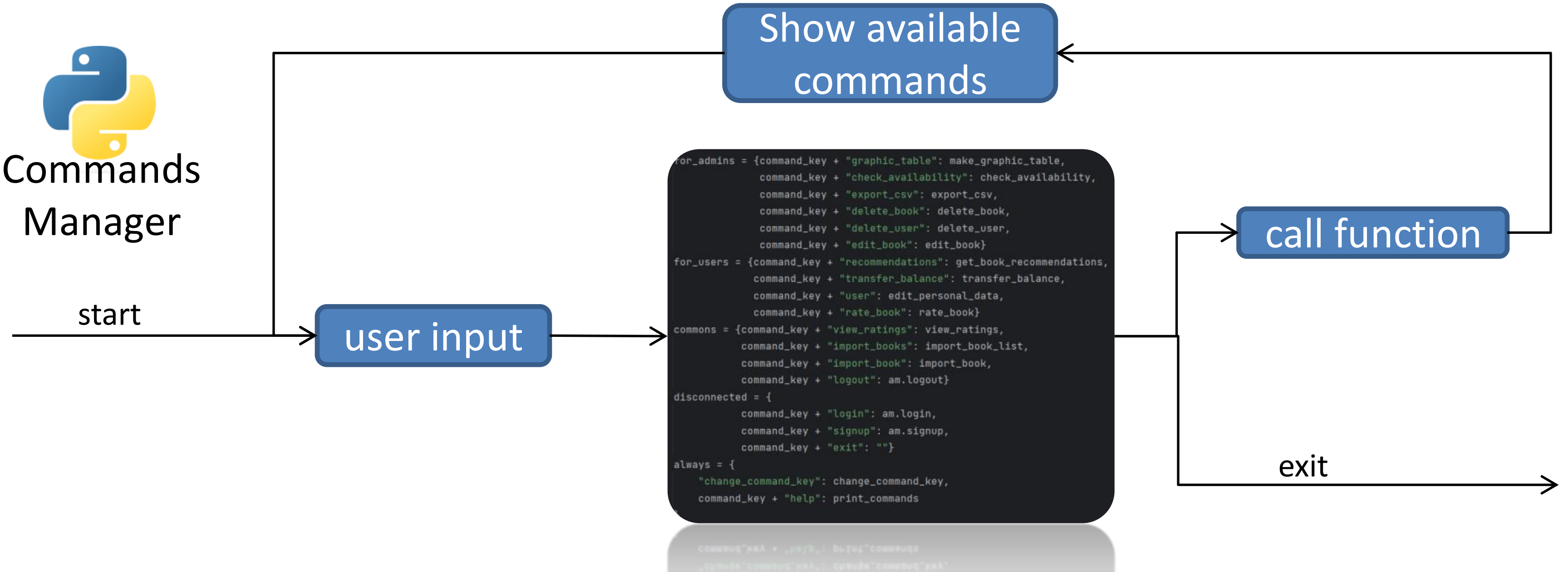
```
Boxed

    Header

Colored
```

The Account Manager stores all information regarding the logged in user, keeping track of their id, and a copy of their entry in the dataset for quick access. It also houses the following methods and commands:

- Signup
- Login
- Logout
- Password Verification

Account Manager

The Commands Manager is the largest file, with well over a thousand lines of code. It has methods for almost all available commands.
It also has 5 different dictionaries with the command names, and the functions they call once used. Each category has different visibility, depending on the logged in user



```
for_admins = {command_key + "graphic_table": make_graphic_table,
              command_key + "check_availability": check_availability,
              command_key + "export_csv": export_csv,
              command_key + "delete_book": delete_book,
              command_key + "delete_user": delete_user,
              command_key + "edit_book": edit_book}
for_users = {command_key + "recommendations": get_book_recommendations,
             command_key + "transfer_balance": transfer_balance,
             command_key + "user": edit_personal_data,
             command_key + "rate_book": rate_book}
commons = {command_key + "view_ratings": view_ratings,
           command_key + "import_books": import_book_list,
           command_key + "import_book": import_book,
           command_key + "logout": am.logout}
disconnected = {
           command_key + "login": am.login,
           command_key + "signup": am.signup,
           command_key + "exit": ""}
always = {
    "change_command_key": change_command_key,
    command_key + "help": print_commands
```

Next up: The list of commands. Each command's visibility and accessibility are separated into 5 groups: always, disconnected, commons, for users, for admins.

The Command Manager is responsible for updating these commands and getting the user's input for which one to call. Here's all the commands the program holds, as well as their categories:

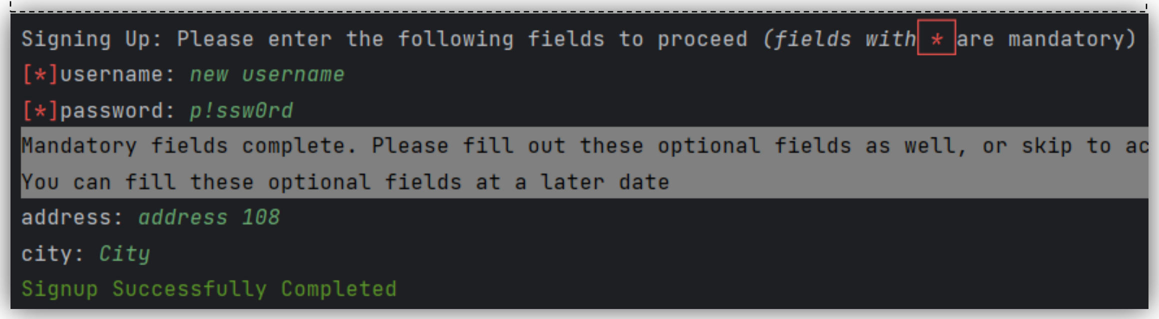
Disconnected Commands:

- Signup
- Login
- Exit

Disconnected Commands are only shown prior of logging in, or after logging out. Any time there is no user logged in to the system, these commands are active.

The signup process, requesting a unique username, and a password of 8+ characters plus at least one special character from this list "!@#$%^&*()-+?_=,<>/"

This command can only sign up users. Admins must be manually inserted into the system through their respective database file.



```
Signing Up: Please enter the following fields to proceed (fields with * are mandatory)
[*]username: new username
[*]password: p!ssw0rd
Mandatory fields complete. Please fill out these optional fields as well, or skip to ac
You can fill these optional fields at a later date
address: address 108
city: City
Signup Successfully Completed
```

Allows for 3 tries to enter a correct username and password, allowing both users and admins to connect using this.

Through the code itself, the function can have a variable amount of tries, and even autofill the username and password fields. This is only possible through calling the function manually through the code however, so it is merely a debugging functionality meant to quickly allow access to the system without having to always type out these fields in the console.

Terminates the "run console" function of the Console Manager, allowing the Dataset Manager to then save the updated datasets before shutting down the program.

For Users:

- Recommendations
- Transfer Balance
- User
- Rate Book

This list of commands is only available to users, and will not be shown to admins or prior of logging in

Allows viewing and editing of personal info, such as the user's username, or orders list. ID and Balance are also visible, but cannot be edited by the user. The user's password is shown as eight asterisks instead.

Attempting to change the password requires them to enter their old password first, with 3 failed attempts kicking them out of the system. Exiting the change-password function and re-entering does not reset their tries counter.

Changing their orders list updates the user's balance, and books that the user does not have sufficient money to purchase are declined. During this editing period, extra commands are also available to the user, such as rating the book, clearing their list, and creating a new book entry to add to their list / replace another book in their list.

Allows the user to "transfer money from their card to their account's balance", effectively giving them more money that they can use within the system.

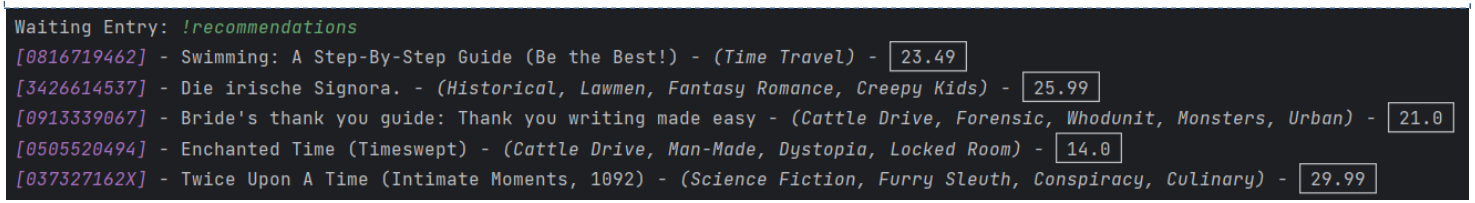
Rating a book prompts the user to select one of the books in their orders list. They can then either edit their already existing rating, or create one. Only one rating per book per user can exist.

Users are the only ones able of changing their numeric rating, whereas admins can only edit the comment section of that rating, or delete the entire rating.

Finds the most common category/categories in the user's favorites list, and compiles a list of books, first with only those categories, and if the selection is insufficient, with at least one of those categories.

In the case the list is still too small, or the user has no favorites in their list, the entirety of the database is used as well, regardless of categories.

From that list, 5 are randomly selected and shown to the user, including their IDs for easy searching, their categories, and their cost if they want to add that book in their orders list.



```
Waiting Entry: !recommendations
[0816719462] - Swimming: A Step-By-Step Guide (Be the Best!) - (Time Travel) - 23.49
[3426614537] - Die irische Signora. - (Historical, Lawmen, Fantasy Romance, Creepy Kids) - 25.99
[0913339067] - Bride's thank you guide: Thank you writing made easy - (Cattle Drive, Forensic, Whodunit, Monsters, Urban) - 21.0
[0505520494] - Enchanted Time (Timeswept) - (Cattle Drive, Man-Made, Dystopia, Locked Room) - 14.0
[037327162X] - Twice Upon A Time (Intimate Moments, 1092) - (Science Fiction, Furry Sleuth, Conspiracy, Culinary) - 29.99
```

For Admins:

- Graph Table
- Check Availability
- Export CSV
- Delete Book
- Delete User
- Edit Book

This list of commands is only available to admins, and will not be shown to users or prior of logging in

Allows the export of user/admin/book entries to an output file given by the admin. Exporting user/admin entries replaces all passwords with dashes, so that not even the hashed passwords can be extracted using this process.

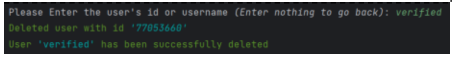
Only deletes the book from the libraries accessible to the admin. If all libraries can be accessed, the entry itself is removed from the database.

Only applicable to books that the admin has access to at least one library the book is housed in. All fields are editable (except id).

While editing the bookstores or availability sections, only the libraries the admin has access to can be added/removed. Removing all libraries the admin has access from the book's bookstores kicks the admin out of the editing book command.

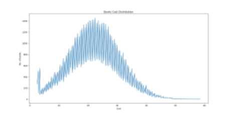
Updating a book's bookstores also automatically updates the book's availability section, so that it is not available to bookstores it is no longer a part of.

When given a user's ID or username, deletes them from the database.

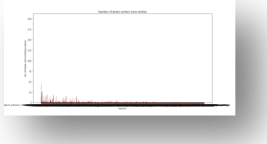


```
Please Enter the user's id or username (Enter nothing to go back): verified
Deleted user with id '77053660'
User 'verified' has been successfully deleted
```

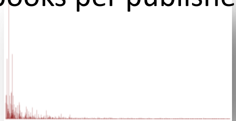
book cost



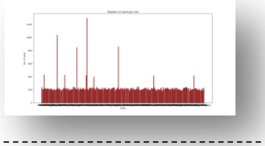
books per author



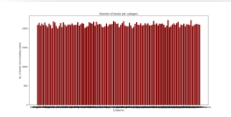
books per publisher



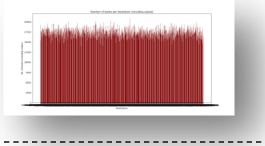
users per city



books per category

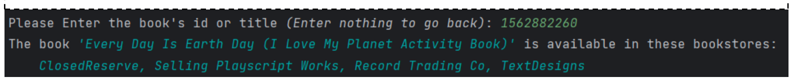


books per bookstore



*The above graphs were made using 1k admins, 50k users, and 100k books for higher definition

The Check Availability command asks for either the ID of the book, or the title of the book the admin wants to see its availability options.
Once a valid id/title has been provided, the list of bookstores the book is available in will be shown



```
Please Enter the book's id or title (Enter nothing to go back): 1562882260
The book 'Every Day Is Earth Day (I Love My Planet Activity Book)' is available in these bookstores:
    ClosedReserve, Selling Playscript Works, Record Trading Co, TextDesigns
```

Commons:

- View Ratings
- Import Books
- Import Book
- Logout

This category is for commands available to both users and admins, and are accessible as long as a user is logged in. Users and Admins may have slightly different versions of these commands.

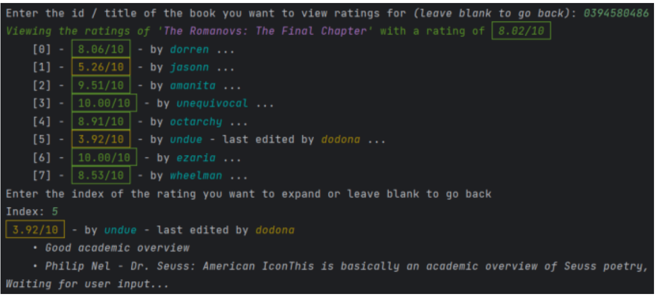
Prompts the user/admin to select one csv file from which books will be imported. Users have these books enter their favorites list, whereas admins simply have them inserted into the system.

Logs out the user

Allows the creation of a book entry, asking for title, author, publisher, ect. The only mandatory fields are the title and cost. Admins can also specify the book's libraries, availability, and copies.

Allows the viewing of a book's ratings. It first displays a compacted version, only showing the numeric rating, the user who submitted it, and the user/admin who last edited it (if any). Selecting one of them then shows the full rating.

Admins also have two extra commands during this step, allowing for quick editing of that specific rating they are viewing, or deleting said rating, provided they have access to at least one of that book's libraries.



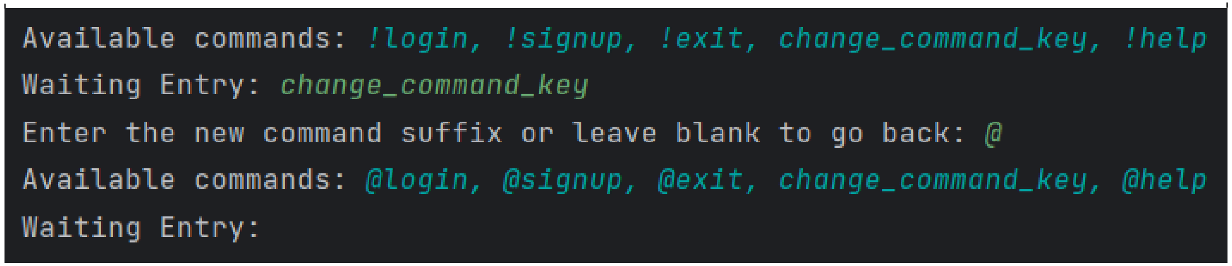
```
Enter the id / title of the book you want to view ratings for (leave blank to go back): 0394580486
Viewing the ratings of 'The Romanovs: The Final Chapter' with a rating of 8.02/10
    [0] - 8.06/10 - by dorren ...
    [1] - 5.26/10 - by jasonn ...
    [2] - 9.51/10 - by amanita ...
    [3] - 10.00/10 - by unequivocal ...
    [4] - 8.91/10 - by octarchy ...
    [5] - 3.92/10 - by undue - last edited by dodona ...
    [6] - 10.00/10 - by ezaria ...
    [7] - 8.53/10 - by wheelman ...
Enter the index of the rating you want to expand or leave blank to go back
Index: 5
3.92/10 - by undue - last edited by dodona
    • Good academic overview
    • Philip Nel - Dr. Seuss: American IconThis is basically an academic overview of Seuss poetry,
Waiting for user input...
```

Always:

- Change Command Key
- Help

These functions are always available, to both users and admins, logged in or out.

Changes the command key that's used by all commands as a suffix. By default this is the '!' key. This command is the only exception, and does not require a command key to activate.



```
Available commands: !login, !signup, !exit, change_command_key, !help
Waiting Entry: change_command_key
Enter the new command suffix or leave blank to go back: @
Available commands: @login, @signup, @exit, change_command_key, @help
Waiting Entry:
```

Prints out the list of commands available.

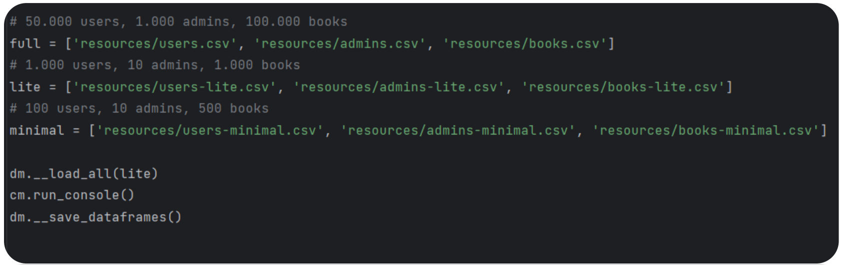
There are 3 different models that have been set up: Minimal – Lite – Full

Due to size constraints, only Minimal and Lite have been included.

Minimal: 100 users – 10 admins – 500 books
Lite: 1.000 users – 10 admins – 1.000 books
Full: 50.000 users – 1.000 admins – 100.000 books

*the diagrams generated and shown prior were made using the Full model

Main

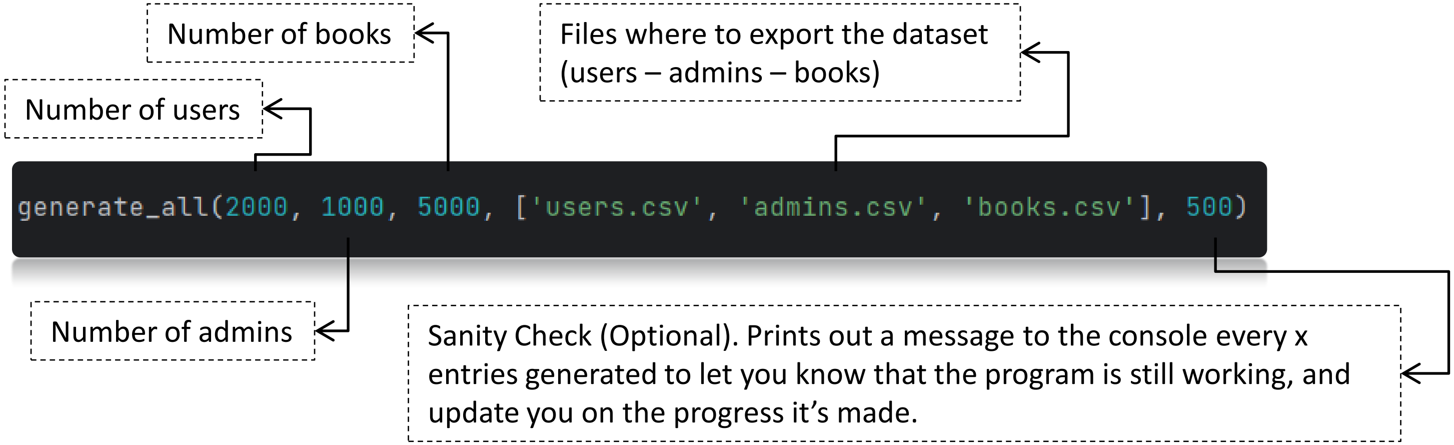


```
# 50.000 users, 1.000 admins, 100.000 books
full = ['resources/users.csv', 'resources/admins.csv', 'resources/books.csv']
# 1.000 users, 10 admins, 1.000 books
lite = ['resources/users-lite.csv', 'resources/admins-lite.csv', 'resources/books-lite.csv']
# 100 users, 10 admins, 500 books
minimal = ['resources/users-minimal.csv', 'resources/admins-minimal.csv', 'resources/books-minimal.csv']

dm.__load_all(lite)
cm.run_console()
dm.__save_dataframes()
```

The Main file can control which version of the model is loaded.

If the csv_generator is imported, it can also be used to generate new models by calling its generate_all method.

CSV Generator



```
generate_all(2000, 1000, 5000, ['users.csv', 'admins.csv', 'books.csv'], 500)
```

Number of books

Number of users

Files where to export the dataset (users – admins – books)

Number of admins

Sanity Check (Optional). Prints out a message to the console every x entries generated to let you know that the program is still working, and update you on the progress it's made.