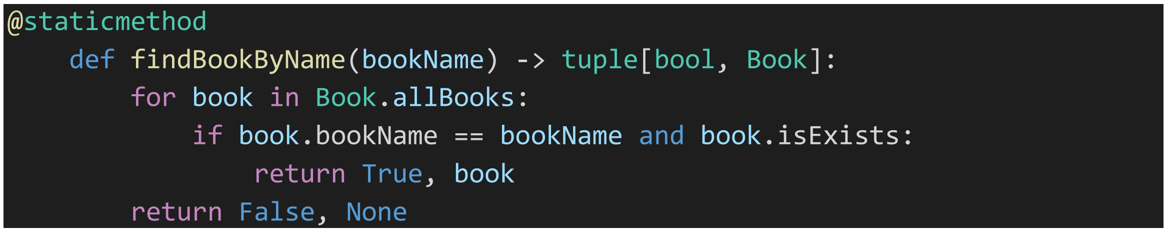
Design Principles used in Library Management System

1. DRY – Functions are defined to prevent repetition of code. For example –



```
@staticmethod
    def findBookByName(bookName) -> tuple[bool, Book]:
        for book in Book.allBooks:
            if book.bookName == bookName and book.isExists:
                return True, book
        return False, None
```

2. KISS – Hooks are passed in all functions instead of objects. Librarian and Admin are singletons.

3. YAGNI – Features which are not utilized are removed.

4. SOC – All class related functions for Admin, Librarian, Student, Book and BookCopy remain in their corresponding class.

5. TDA – Followed throughout while accessing object data.

6. S (Single responsibility) – Admin class is separate, does not inherit User and performs a single task of adding accounts. Book and BookCopy are separated.

7. O (Open close principle) – Attempted to follow as much as possible.

8. D (Dependency Inversion) – Login System features are added as composition in the users that require it. The composition is dependent on a Login System Interface rather than the Login system class.

Validations added in Library Management System

1. No user can perform an operation without logging in with the correct credentials.
2. All create operations for objects with unique hooks check if the hook is already present in the list of all objects
3. Issuing book checks availability of books.
4. Return book checks if the book is present in the first place.
5. Get books by author names checks the presence of author in book db.
6. Code Related –
    a. All functions follow strict typing for return values.