

The “system-to-be”: YorkU Library Management Java App

“YorkU Library Management Team” is now seeking a new system that can help them provide better online services to their clients (e.g., students, faculty members, non-faculty staff, and visitors). The system is supposed to be a **GUI-based Java application**. The basic requirements of the system (from an interview with their management teams) are as follows:

- **Req1:** Any client should be able to register as a user of the system with a unique/valid email and strong password (i.e., a combination of uppercase letters, lowercase letters, numbers, and symbols). The system currently allows four types of clients to be registered, i.e., students, faculty members, non-faculty staff, and visitors, while it’s open for new types in the future. If a client registers as a student, a faculty member or a non faculty staff, her/his registration requires a further validation from the management teams.
- **Req2:** Using the system, any registered client can rent a physical item (i.e., books, magazines, CDs), open an online book, or subscribe to an online university-provided newsletter (e.g. NY Times), etc. Each physical item has 20 copies in the library. Penalty will be applied if a book is overdue (i.e., 0.5\$ a day). A user can borrow up to 10 physical items and can keep an item for at most 1 month. All physical items borrowed from the library (books, magazines, CDs) count toward the total of 10 items. A user will lose his borrowing privileges if he has more than 3 items overdue. Books that are 15 days overdue will be considered lost.
- **Req3:** After login, the system should show a list of hardcover books that a user is currently renting and the due date for returning the books. It should also prompt warnings about any book that is not returned yet and it is approaching (less than 24 hours until the due date) or past the due date.
- **Req4:** The system should allow a user to subscribe and read a paid-for newsletter via its interface, such as the NY Times. This can be done by opening a frame within the system where the NY Times website can be loaded. A user can decide at any time to cancel a newsletter subscription.
- **Req5:** A user can search for a book using the application. For a book a user is searching, the app should also show recommendations of similar other books (based on the text similarity of book titles).
- **Req6:** If a user is a faculty, the app can keep track of the courses the user is teaching and the textbooks the user has previously used. The app then offers notifications to the user when a new edition of the textbook is available. If a textbook is not available, the app should notify the library management team of this, so that they could consult with the user to procure the book.
- **Req7:** Each item has a unique identification number and other details including its location in the library and whether the item can be purchased, which will help with the navigation for clients. Managers of the system can add, enable (can be rented), or disable (cannot be rented) an item.
- **Req8:** If a user is a student, the textbooks of a given course the student is taking, the app should make virtual copies of the textbooks available to the account of the user for the duration of the course. After that, the app should remove the virtual copies from the student account.
- **Req9:** A user can request for a new book. A request can be of two types, i.e., textbooks for course teaching and self-improvement, etc. Depending on the type, the request will need to be prioritized by the app and the user should be notified of the priority accordingly. Often, textbooks for course teaching will be given higher priority.
- **Req10:** The system could also offer discounted purchases of items via its special agreements with publishers, whose books/DVD are not normally freely available via the usual library management system. For this, the system needs to provide payment options like debit, credit, mobile wallet, etc.
- **Req11:** System data is stored in a database, we will use CSV/Excel files to simulate this process.