Functional Requirements (Functions/Services)

User:

Creating an account: adds users with name, email, password, auto assigned Id and lets you pick your role - member, admin or librarian

Login: asks for the email and password you provided when you made the account - or the hardcoded logons in the code for testing.

Admin:

View Users: display all users Id, name, email and account type.

View Books: show all books Id and title

Member:

View books: display all books through librarian .

Borrow book: borrow up to 5 books and updating status to borrowed recording in borrowing

records.

Reserve book: reserve up to 5 books and updating their status reserved

Return book: return borrowed or reserved books and updating status to available

Librarian:

Add book: adds a book with Id and title and gives it the default status of available

Remove book: delete a book by Id.

Edit status: update a book’s status - available, borrowed or reserved

Show books: List all books Id, title and its status.

Generate report: Show borrowed books with its Id and title.

Book Management:

Dosnt allow multiple user to borrow / reserve the same book

And it blocks it it when they do try, it blocks the request

System:

Starts with 3 users and 10 hardcoded books

Exit: it deletes created users and created books

Non-Functional Requirements

usability: very simple command menu, clear responses from the choices.

Performance: can deal with up to moderate use. - login and borrowing are instant.

Reliability: stops invalid inputs, stops multiple requests on the same book

Security: logs in with email and password, role lets you access specific manus and permissions

Maintainability: easy to read design, and very editable (once i change the variable names back)

Portability: if you access the github it ‘should’ run the same from pc to pc (assuming you're using a c++ engine / windows)

.

Memory Management: Clean up on exit where it delete any users or books you add when it saves in a vector (except the hard coded ones)