

# **Library Management Application\***

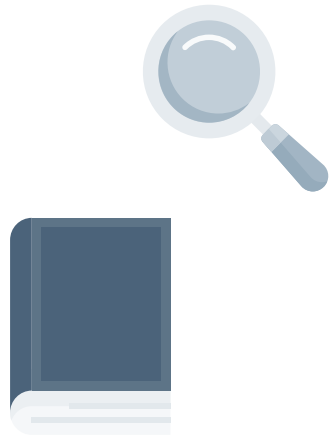
Jose Garcia-Esparza  
Ricardo Sanchez-Macias

\* Name TBD

# System Description

Application with administrator users and end users

- For administrator, provide tasks to easily manage a library (e.g. adding, deleting, and editing books)
- For end user, provide tasks to easily utilize library resources (e.g. check-out books, search books)



# Planned Implementation\*

- Database
  - Supported by SQLAlchemy
- Backend
  - Python Flask
- Front-end
  - HTML and CSS

SQLAlchemy



\* Subject to change

# Key Use Cases

## For librarian and end user:

- To search books by different traits (author name, title, genre)



## For end user:

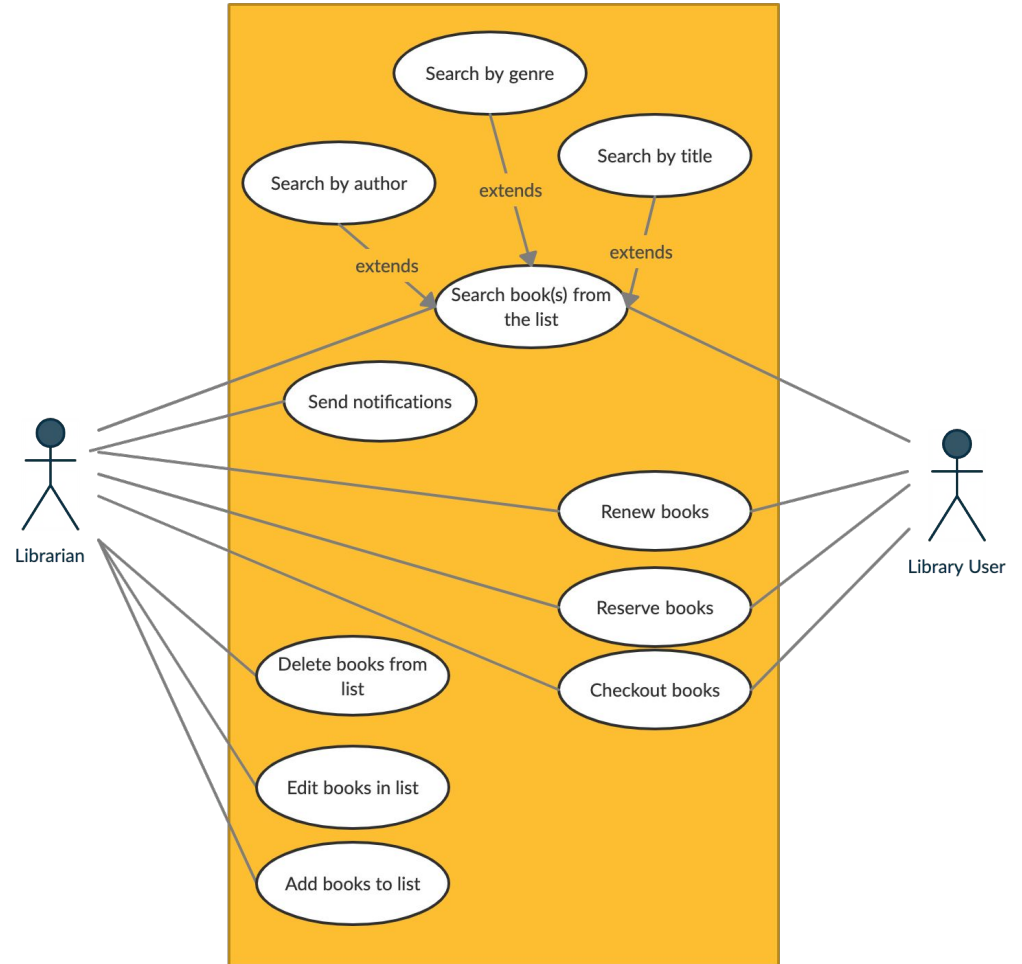
- Check-out books
- Reserve books if not available

## For librarian:

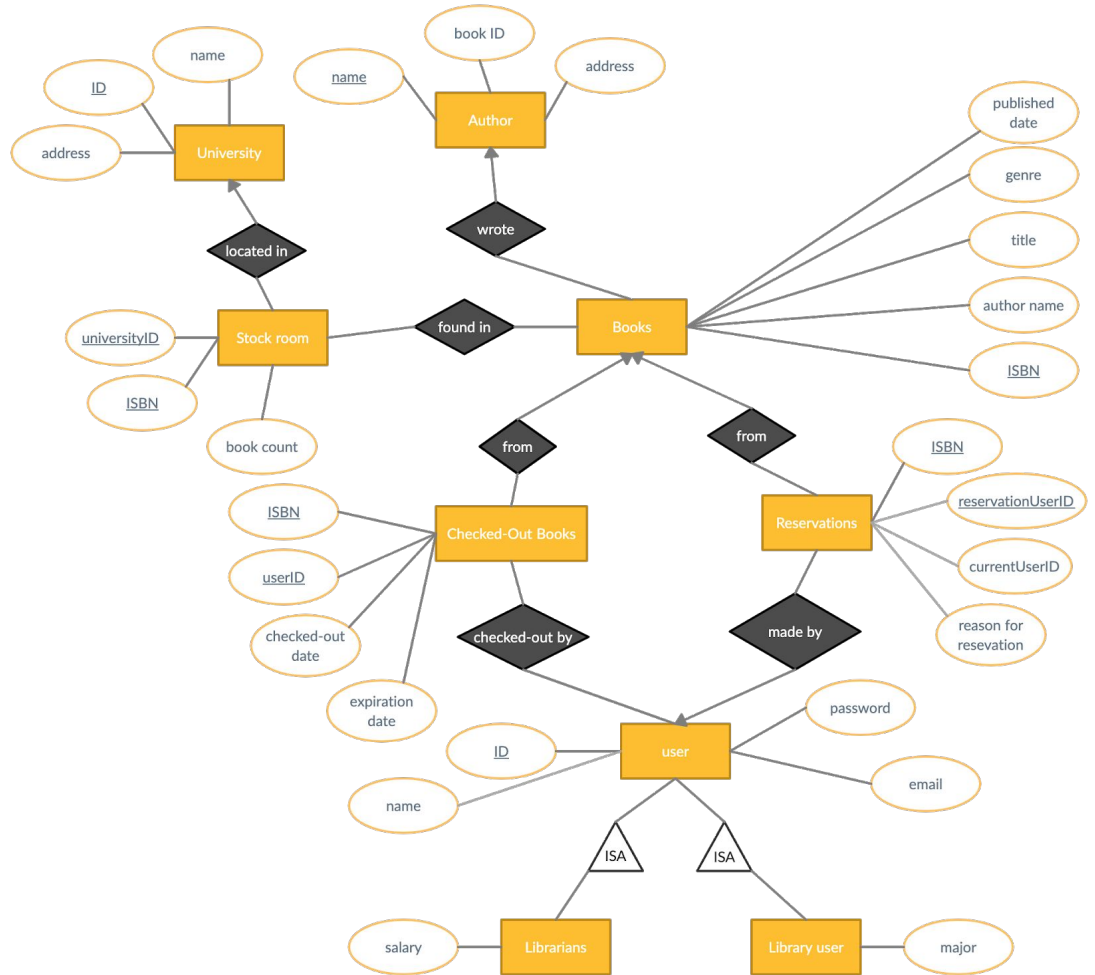
- Add, delete and edit books



# User-Case Diagram



# E/R Diagram



# Relational Schema

## Books:

- b\_isbn, b\_authurname, b\_title, b\_genre, b\_publisheddate

## Checked-out Books:

- cb\_isbn, cb\_userid, cb\_checkeddate, cb\_expirationdate

## Reservations:

- r\_isbn, r\_fromuserid, r\_currentuserid, r\_reason

## Users:

- u\_userid, u\_name, u\_email, u\_password

## Librarians

- l\_userid, l\_salary

## Library users:

- lu\_userid, lu\_major

## Authors:

- a\_name, a\_isbn, a\_address

## Stock room:

- s\_universidtyid, s\_isbn, s\_bookcount

## University:

- uni\_id, uni\_name, uni\_address

## Checked-by:

- cby\_isbn, cby\_userid

# Thanks!

Any questions?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik** and illustrations by **Stories**