

General Description of the Project (red means I am deleting it)

- My project will help people keep track of their home library. They will be able to access their list from their mobile phone and computer.
- Users will be able to search for books and add them to their own personal library.
- I am playing with the idea of connecting to other systems where a user could download an ebook or audiobook.
- The user will be able to add notes to the books and rate the books. Maybe even connect to Goodreads.
- I am looking at multiple API's for this app:
 - **The Open Library API** - this catalog will allow me to connect lots of older books to my system and then, possibly, download a version for themselves.
 - <https://openlibrary.org/developers/api>
 - **Google Books** - allows developers to fetch Google Books data for their applications. With this API, applications can conduct full-text searches for books. Developers can then get information related to ratings, reviews, and author information.
 - <https://developers.google.com/books>
 - **Audible** - A Python low-level interface to communicate with the non-publicly Audible API. It enables Python developers to create their own Audible services. Asynchronous communication with the Audible API is supported.
 - <https://audible.readthedocs.io/en/latest/>
 - **Goodreads** - Let members connect to their Goodreads accounts, and you'll have full access to the books in their shelves, their ratings, their reviews, and their friends – the social reading graph. Use this to personalize an ecommerce store, power recommendations, show a widget of a member's favorite books, build a mobile or desktop client app, and more.
 - <https://www.goodreads.com/api>
 - **Google Drive** - The Google Drive API lets you create apps that leverage Google Drive cloud storage. You can develop applications that integrate with Drive, and create robust functionality in your application using the Drive API.
 - <https://developers.google.com/drive/api/guides/about-sdk>
- While researching GUIs, I landed on Tkinter. From what I have read it is one of the most common, and since I am new to this I am going to start there.
- Lastly, I think it is important to export your library to keep as a backup on their system. Could be as simple as an excel doc that sits in your drive.

Task Vignettes (User activity "flow")

- Adding books into their personal library.
 - Poe is out and about doing some shopping. He notices a book he really likes so he decides to buy it. He finishes his day of shopping and then heads home. When he gets around to putting the book away he decides he wants to add it to

his in home library app. He pulls out his phone and opens the app. From here he has to login (google or login/create an account), he selects the “Add Book” option. When this option is selected he has to decide if he wants to search the database(Google Book API) for the book, manually enter the book, or scan a barcode on the book. He decides to search the database. He types in the title of the book and presses search. He selects the book from the search results and is presented a form with metadata that has auto populated (pages, goodreads rating, description, etc) then he decides to add the book to his TBR shelf (Shelves are personalized categories in the app to help users filter. For example, To Be Read, Recommendations, Living Room, Work, etc). He selects that shelf from a dropdown then scrolls to the bottom of the form and selects “Save Entry”. This adds the book to his personal library on the app.

- Poe needs to loan a book to a friend and keep track of who he is loaning books to and when.
 - Poe frequently loans books to his friends and forgets who he lent them to. A friend has asked to borrow The Little Prince from him, so Poe gives the book to his friend, they have a few drinks and then the friend leaves with the book. Poe opens his app and searches his personal library for The Little Prince and clicks on that cover. He then selects the “Loaned Book” option. This automatically marks the book as loaned and opens a text entry box for Poe to type information about who he loaned it to. The app will also store when he loaned it out in case he wants to reference that. The default notification will remind Poe in 1 month that he lent the book, but he has the option to update that reminder to be shorter, longer, or nonexistent.
- Flows for above tasks
 - <https://www.figma.com/board/pgcHL4HIIWiv6UImA9od6W/Flows-for-my-Project-Specs?node-id=0-1&t=jZ3xwbCCaXcGR751-1>
- Figma file with my very rough first iteration of screens
 - <https://www.figma.com/proto/UZ0SHBozncHVlg3EtUSMLp/First-Draft-Book-App?page-id=0%3A1&node-id=2-2&viewport=643%2C365%2C0.19&t=Fc60m3GyieVLNhc-1&scaling=scale-down>

Technical “Flow”

- Logging in
 - The user should have the option to login
 - This could be the time they connect their Google account to store the backups
 - User verifies account and that they will use the backup
- Adding a book
 - User would push the “Add Book” button
 - They could search for a Title or Author OR enter the information manually.
 - Add what type of book it is (audiobook, ebook, or paper)

- This is where we would use the Book APIs to search for books and connect them to your library.
- Rating a book
 - Open the book record, scroll down to the Rating field and add your star rating.
 - This is where the API for Goodreads will be used.
 - There will need to be some sort of authorization to login to the user's Goodreads account.
- Adding a note
 - Open the book record, scroll down to the Notes field and add whatever notes you would like.