Book Shelf Software

By: Brady Cash, Christian Flores, Emmanuel Samuel, and AJ Prosser

Project Objective

We will be creating a book shelf software that can manage all your books in one place, including being able to read them, annotate them, and sort them into different categories.

Cost Estimation

Hardware

Since the average size of a book is only around 2 MB, with 4 TB of storage we could hold roughly 2 million books in our repository. Building a small business server with no storage would cost around $2000. Add in four 1TB SSDs for $200 each, and the final price for our server will only be **$2800**.

Costs Estimation

- Most of our costs will come from personnel:
  - Per levels.fyi, the median software engineer salary in Dallas, Texas is **$124,000** and the median software engineering manager salary in Dallas, Texas is **$210,095.** I estimate that we could build and maintain our app with a team of 7 engineers and 1 manager which would lead to a cost of personnel of **$1,078,095**

- Almost all of our technology is free and open source which allows our team to save a lot of money.
  - MariaDB - Database Software
  - React Native - Cross-Platform Framework
  - Spring Boot - Java Backend Framework
  - Figma - UI/UX Design
  - Trello - Team Management
- If our team was to want to move our application to the cloud MariaDB charges as you go:

STORAGE PER NODE	PRICE / MONTH	PRICE / HOUR
AWS io1 Storage / GB	$0.125	$0.0001712329
AWS io1 I/O / IOPS	$0.065	$0.0000890411
AWS GP2 / GB	$0.100	$0.0001369863
GCP SSD / GB	$0.170	$0.0002328767
AWS S3 / GB	$0.023	$0.0000315068
GCP Cloud Storage	$0.026	$0.0000356164

Effort Estimation

	Function Category	Count	Complexity			Count x Complexity
			Simple	Avg	Cmplx	
1	# of User Input	1 (Notation) + 1 (Categories + 1 (Search Text) + 1 (Adding Bookmark) + 1 (View settings) = 5 inputs	3	4	6	4 * 5 = 20
2	# of User Output	1 (Displaying Text W/ Correct View Settings) + 1 (Displaying a Notation) = 2	4	5	7	2 * 5 = 10
3	# of User Queries	2 (Displaying Notations per book/per page) + 1 (Text Search Results) + 1 (Displaying books within a category) = 4	3	4	6	4 * 4 = 16
4	# of Data Files and Relational Tables	1 (Type of book file), 1 (Table to categorize all the books), 1 (Relational table to connect the first and second) = 3	7	10	15	Complexity of Relational table is simple, the other two are average 3 * 9 = 27
5	# of External Interfaces	1 (Touchscreen)	5	7	10	1 * 7 = 7
					GFP	80

PC1 = 1
PC2 = 1
PC3 = 0
PC4 = 1
PC5 = 4
PC6 = 1
PC7 = 0
PC8 = 0
PC9 = 2
PC10 = 1
PC11 = 1
PC12 = 3
PC13 = 3
PC14 = 5

Sum(PC) = 20
PCA = 0.65 + .01(20) = 0.85

Function Point = GFP x PCA
Function Point = 80 * 0.85
Function Point = 68
Estimated effort = FP / productivity = 68 / 60 = 1.13333
D = 1.13333 / 7 = .162 weeks

Assuming that one programmer has a productivity equivalent to 60 function points per week, it would take roughly .162 weeks to finish this project with 7 employees.

Project Timeline

Initial Start Date: May 1, 2023

Initial End Date: May 8, 2023

Using the function point algorithm it was estimated that it will take less than 1 week to program our bookshelves functional capabilities, as seen in the next question. Thus, we will spend a couple extra days sorting out bugs, which is why I gave it a week for the initial release. The number of workers per day is estimated to be 8 hours a day per person only on weekdays, and we would have 7 employees. Furthermore, after the initial release we plan to operate and update the software for at least 25 years.

Functional Requirements

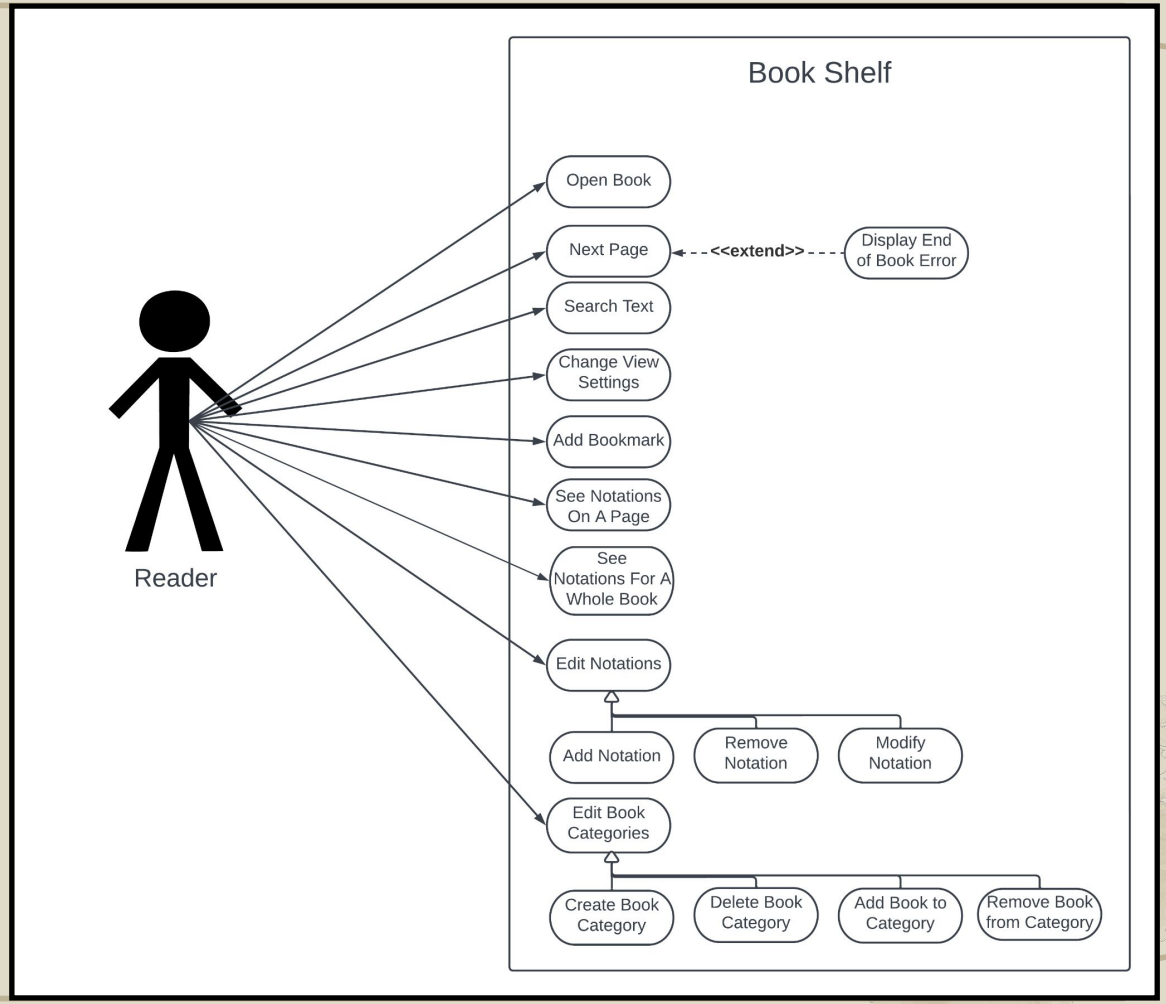
User should be able to:

1. Load and delete books
2. Create and maintain categories
3. Read books
4. Search book text
5. Add bookmarks
6. Change visual settings (dark mode, text display size, etc.)
7. Add and manage notations
    a. See all notations per page or per book

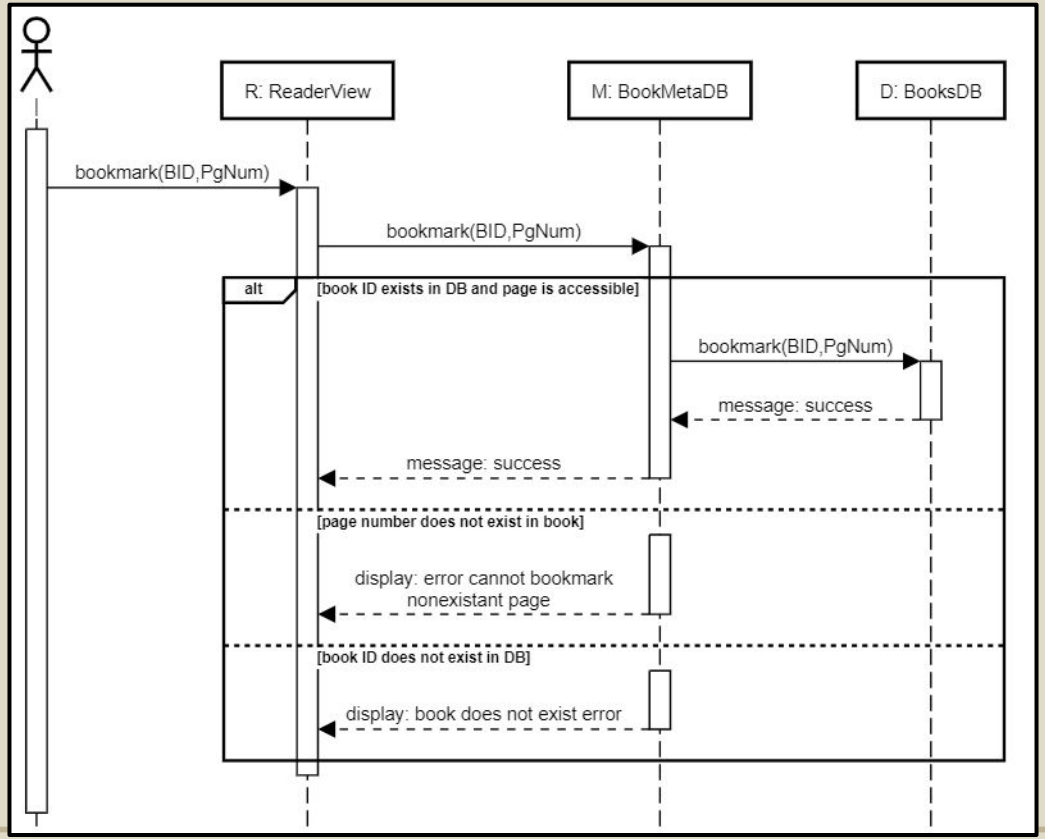
Non-functional Requirements

- Usability requirement: The software should support English, Spanish, and Chinese languages.
- Performance requirement: It should not take more than 5 seconds to load a book.
- Space requirement: The software should not be larger than 10 GB.
- Dependability requirement: There should be multiple clouder servers that have backed up information for users to offer redundancy if one of the servers fails.
- Security requirement: It should require the user to set up 2-factor authentication.
- Environmental requirement: The servers that we use to store user data should run purely on renewable electricity.
- Operational requirement: The basic local functionalities of the software should be able to continue to run when internet connection is lost.
- Development requirement: There should be an update to the software once a month to fix any reported bugs or issues.
- Regulatory requirement: The software should require age verification before allowing children to read explicit books. (Assuming there is a regulation for showing explicit material to children)
- Ethical requirement: User data should not be shared with any third-parties without the user's authorization.

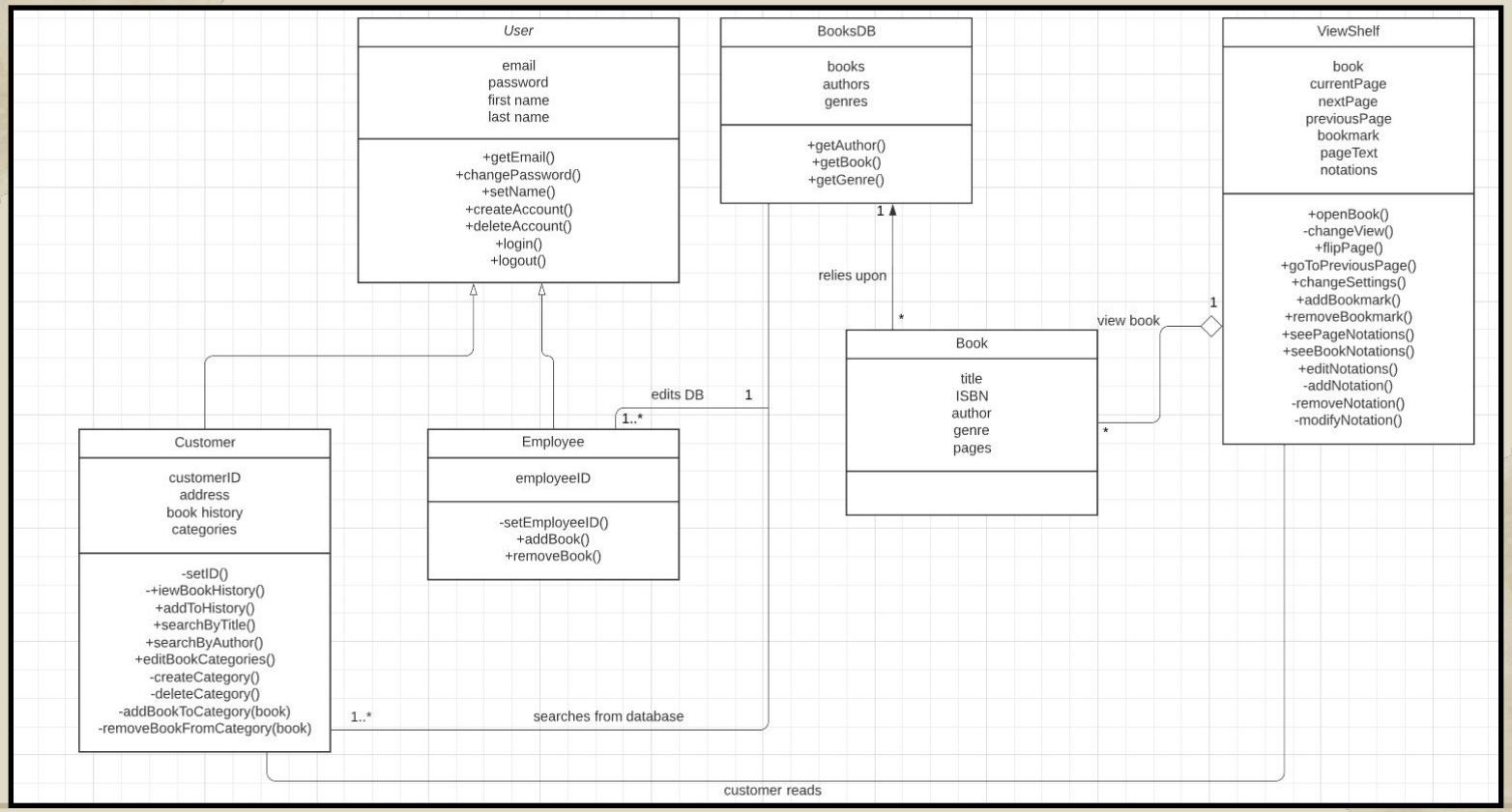
Use Case Diagram



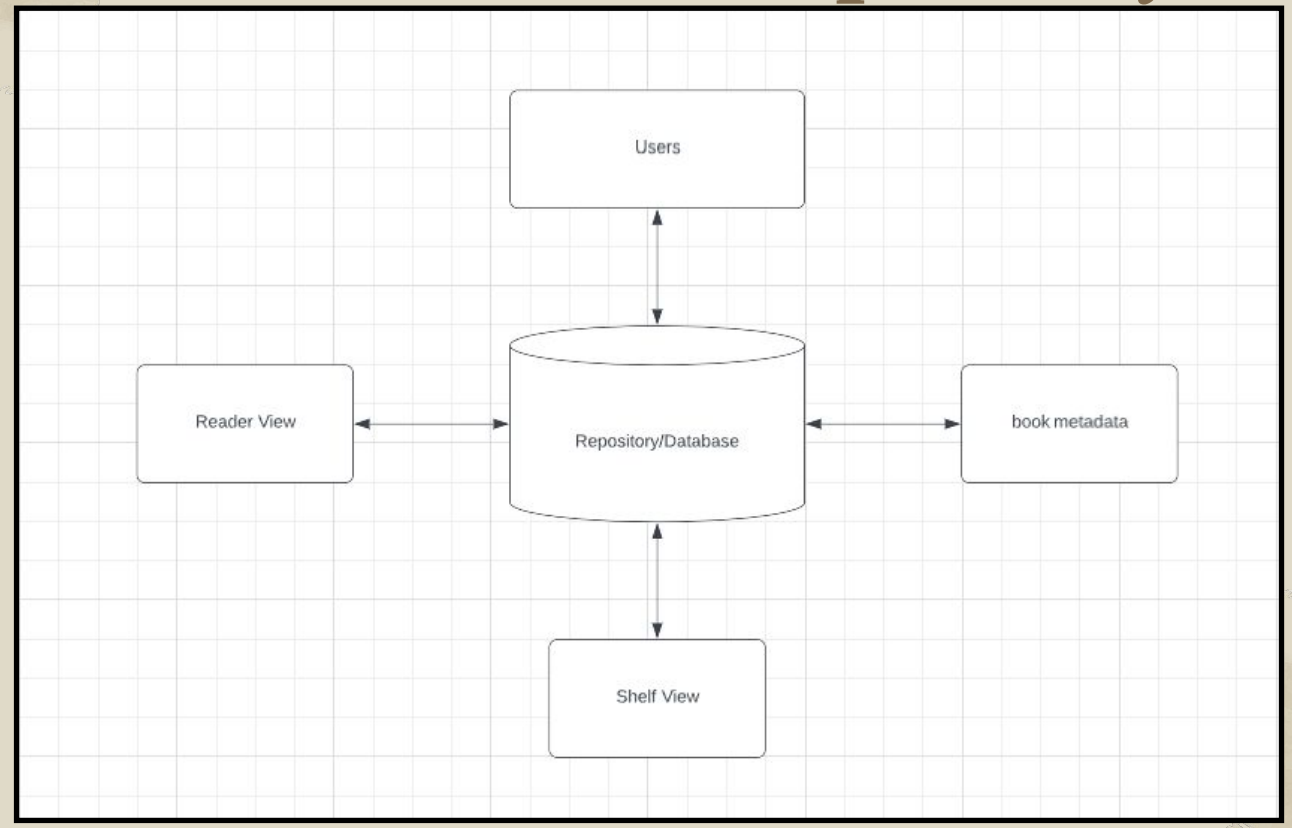
Sequence Diagram Example - Add Bookmark



Class Diagram

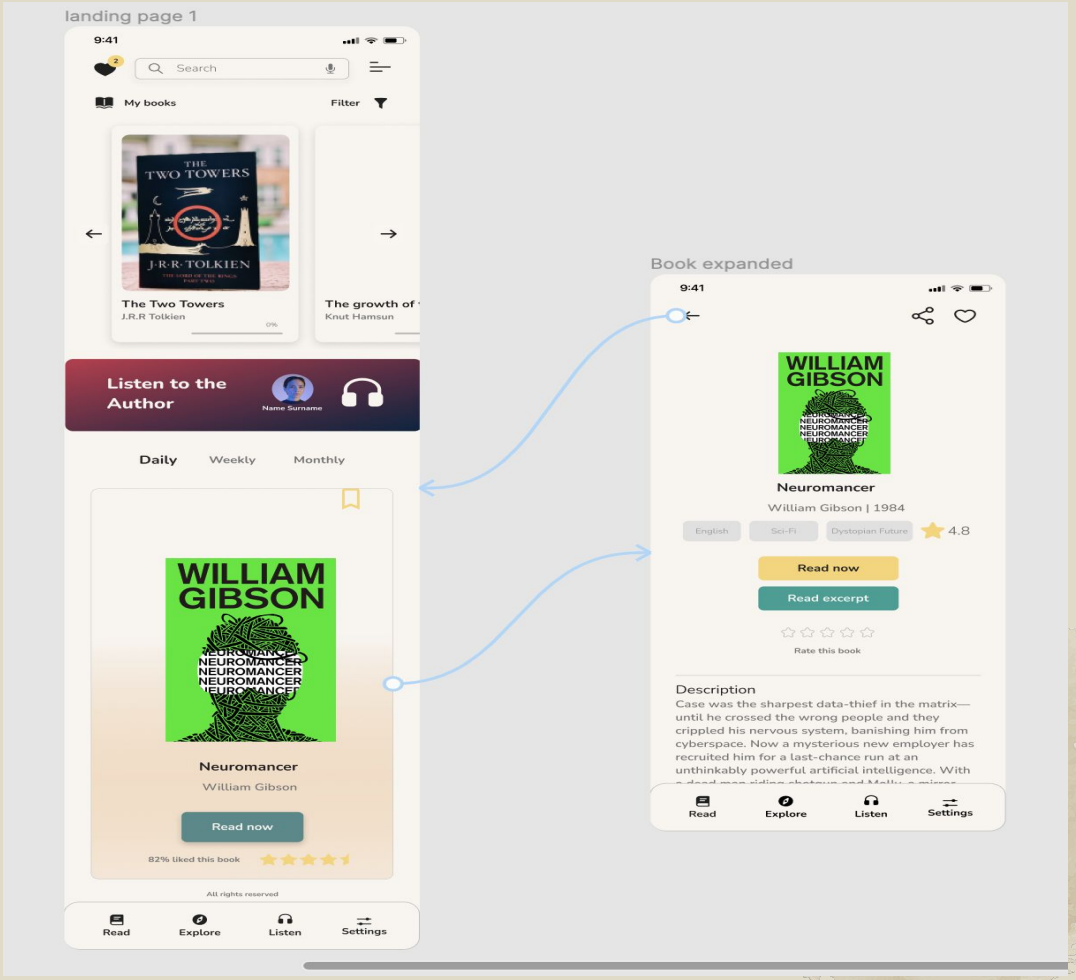


Architecture - Repository



UI Mockup - Shelf View

Created using Figma



UI Mockup - Book View

- The book scrolls as you read just as any normal app would. This is demonstrated in the 1st and 2nd images
- It also keeps track of your current chapter