CS496 Software Project:
Karson Institute Digital Library

Vilnis Jatnieks, Aidan Marshall

2026-02-11

1 Client Information

By sharing this client information and the rest of this document, you are stating that this client has provided this project as something they want (not something you created and asked if they wanted), and that they are interested in having you complete this project for your capstone.

- Client name:
- Client title:
- Client email address:
- Client employer:
- How you know the client:

2 Project Description

2.1 Overview

[Add a few paragraphs describing your project succinctly. What problem are you trying to solve, what is the purpose of your project? Why does your client want this project?]

2.2 Key Features

[At this point you should have a basic understanding of your client's needs. List out the key features of the software system the client wants you to build.]

2.3 Why this Project is Interesting

[Why did you decide this project was interesting enough to you to be a capstone project? What about this project is enticing? Why should anyone care?]

2.4 Areas of CS required

[What subfields of computer science seem most likely to be relevant to your project? A capstone must involve multiple.]

2.5 Potential Concerns and Questions

[Is there any aspect of this project that makes you unsure if it will work, either due to your own interests/background, or that you aren't sure if it fits the requirements? Are there questions you have about this project that you want instructor feedback about?]

2.6 Summary of Efforts to Find a Project

(Not necessary for 482) [Briefly list out when/how you've discussed with this client, and if you've discussed with other clients who either didn't work out or didn't respond. If you considered a different project and it didn't work out, why didn't it work out?]

[Most CS495 projects end here. The sections below are for CS482 and CS496 software projects].

2.7 Comparison to Draft

[For CS496 only, focus on highlighting the major differences between the draft proposal in CS495 and this one here. If there are no major differences, you can remove this subsection.]

3 Requirements

3.1 Non-Functional Requirements

[Non-functional requirements are just as important as functional requirements. Dont forget to specify them.]

Table 1 presents the NFRs for this software project.

ID	NFR Title	Category	Description
NFR0	Restrict checkout	System	Users with overdue books cannot check out additional books
NFR1	Password strength	Security	Passwords must have 8+ characters with 1 capital letter, 1 number, and 1 special character
NFR2	2 Factor Authentication	Security	Allow users to add additional security to their account
NFR3	SSO	Security	Allow users to use Loyola SSO
NFR4	Notifications	UI	Bell icon + number of notifications
NFR5	Limit Review Posters	System	Only users who have checked out a book can leave a review for that book
NFR6	Comment filter	System	Filter out profanities and unwanted language
NFR7	Log In / Log Out	System	Ability to log in and log out to your personal account

Table 1: Non-Functional requirements

3.2 Functional Requirements (User Stories)

Table 2 presents the functional requirements for this software project.

ID	Story Title	Points	Description
A0	Users CRUD	8	As a/an Admin, I want to CRUD other users, so that I can assign users to have staff permissions and delete bot accounts
S1	Book CRUD	8	As a/an Staff, I want to CRUD books, so that I can maintain the collection of books
S2	Fill book info via ISBN	2	As a/an Staff, I want an alternative way to fill book information with an ISBN number, so that make logging a book easier
S3	Scan barcode	5	As a/an Staff, I want to add book by scanning barcode, so that make logging even easier
U4	Search for book	2	As a/an User, I want to search for a book, so that I can find what I want
S5	Check out/in book	1	As a/an Staff, I want to check out/in a book, so that I can update the current status of the collection

U6	Reserve book	2	As a/an User, I want to reserve book, so that I can let people know to not take it
U7	Reminder email	2	As a/an User, I want to receive reminder emails when due date is approaching, so that I can return my book on time
U8	View books	2	As a/an User, I want to view the books, so that I can find a book that I'm
U9	Filter the view of the books	1	As a/an User, I want to filter book, so that I can easily change view
U10	Recommended books	5	As a/an User, I want to see recommended books, so that I can find something that is interesting to me
S11	View checked out books	2	As a/an Staff, I want to see all books that have been checked out, so that I know which books are not currently in the library
U12	Sort the view of books	1	As a/an User, I want sort books by when they were published, so that I can see the newest published books
U13	Upvote/downvote comments	2	As a/an User, I want to react to comments, so that I can show my approval
S14	Tags CRUD	8	As a/an Staff, I want to CRUD tags for books and other works, so that I can organize my collection how I would like
S15	Late users	2	As a/an Staff, I want to view users who have overdue books, so that I have an easy way to follow up with students who need to return a book
U16	User view my history	2	As a/an User, I want to see previous books I checked out, so that I can remember what books I previously read
S17	Staff view book history	2	As a/an Staff, I want to see a book's history, so that I can see where a book has went
U18	Popular books	1	As a/an User, I want to see which books have been checked out the most, so that I can read what other books people find interesting
U19	Password reset	2	As a/an User, I want to reset password, so that I can log if I forget my password
U20	View book covers	2	As a/an User, I want to view book covers, so that I can see the cover artwork
U21	Rate book	3	As a/an User, I want to rate a book I read, so that I can let other people know my opinion on the book
U22	Contact staff	1	As a/an User, I want to send emails to the staff, so that I can ask questions about the collection
U23	Book preview	2	As a/an User, I want to preview the first pages of a book, so that I can read a snippet of the book before checking out
U24	Request extension	1	As a/an User, I want to request an extension of a book, so that I can keep the book for longer
S25	Approve extension	1	As a/an Staff, I want to approve an extension of a book, so that I can allow people to keep books for longer
U26	Create account	2	As a/an User, I want to create an account , so that I can log back in
U27	User get notifications	3	As a/an User, I want to get notifications about overdue books, new books added, receive notis from admin, etc., so that I know what actions need to be taken
A28	Admin send notifications	2	As a/an Admin, I want to push notifications to staff, so that I can send information
U29	Comments CRUD	8	As a/an User, I want to comment reviews on books, so that I can share my opinion about the book
U30	Report comments	1	As a/an User, I want to report comments, so that I can ensure a safe space

U31	Log in/log out	2	As a/an User, I want to log in and log out, so that I can access my account
U32	Popular books by genre	1	As a/an User, I want to see which books have been checked out the most in each genre, so that I can see what other people like
	Total:	89	

Table 2: Functional requirements as User Stories.

3.2.1 AI User Story Analysis

Model: Claude Opus 4.5

Prompt: The attached .pptx is the project my partner and I am working on for our Senior Capstone, and the .xlsx are the current user stories that we have.

Are there any important/valuable user stories we are missing? If yes, what is it and explain why you believe it is needed.

Response: I'll review your capstone proposal and current user stories to identify any gaps.

Check the skill file for working with xlsx files

Read the user stories spreadsheet

Now let me extract the key information from your PowerPoint to understand the full project scope:

Read the capstone proposal slides to understand the project scope

Based on my analysis, here's the most important missing user story:

Missing User Story: View Book Availability/Location

As a User, I want to see whether a book is available and where it's located, so that I know if I can get it and where to find it on the shelves.

Why this is critical:

Your proposal slides explicitly frame the core problem: a student comes to the Karson Institute, searches through shelves, can't find a book, asks staff, and then discovers it's already checked out. Your current user stories have:

U4: Search for book

U8: View books

S11: Staff view checked out books

But no user-facing story that shows availability status or physical location. The student still has to ask staff or wander the shelves—which is the exact pain point your project is supposed to solve.

This story directly addresses the "Finding a Book is Hard" problem from slides 3-4 and should be Sprint 1 or 2, High priority since it's central to your value proposition.

Secondary gap worth noting:

You have S2 (ISBN lookup) and S3 (barcode scan) for adding books, but no story for bulk import of the existing 1600+ book collection. That's a real operational concern for launch—your client won't manually enter 1600 books. Consider whether that needs to be scoped.

Analysis: The first story that Claude proposed is a great idea that we overlooked. We did specifically mention that there would be a feature to find the location of a book. For the second user story, Claude is right that it is inconvenient to have to scan 1,600 books individually. However, we could not think of a reliable way to bulk import books accurately. The only solution to ensure that each book is imported is through a manual individual process. We will not include this story in our functional requirements.

4 System Design

4.1 Architecture

We will be following a Client-Server architecture. Since we are creating a Next.js project, our software will follow their typical repository structure. The main modules are "app" (where all the pages and API routes live), "components" (where the reusable React components that are used throughout the pages live), and "lib" (establish a database connection). Additionally, we will use a RESTful architecture in order for our frontend to be able to communicate with our backend. Some of the modules throughout our program are Users (admin, staff, or normal user), Works (will mostly be books, but could be speeches or other types of written works), Comments (written by users on posts), Notifications (sent to users for various reasons, can be sent by system or admin), Author (of a work), Tags (attached to works to organize collection).

4.2 Diagrams

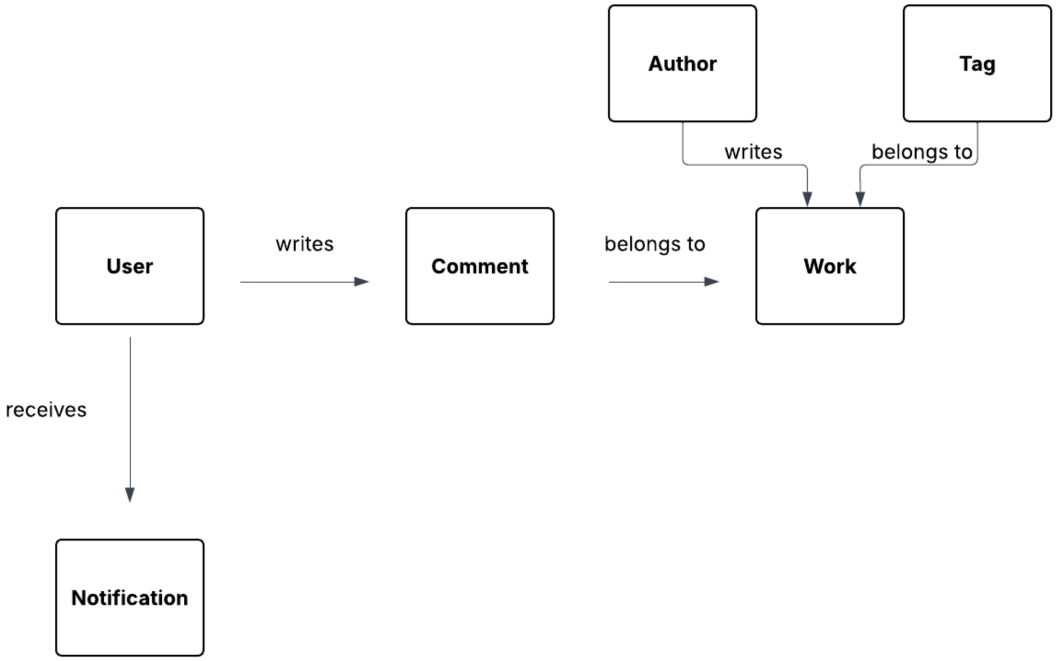


Figure 1: Simplified Class Diagram

4.3 Technology

Technology	Function	Justification
Next.js	Frontend framework	Easier routing
React	Frontend framework	Reusable components, hooks
Tailwind	CSS library	Easier CSS
ShadCN	Component library	Pre-made components
Node.js	JS Runtime	Use one programming language across frontend and backend
TypeScript	Programming language	Strongly typed, self-documenting
ESLint	Linter	Enforce code style
Jest	Test framework	Easier unit testing
Postman	Backend testing	Easier manual testing, backend development
PostgreSQL	Database	Relational database, speed

4.4 Coding Standards

We will be following standard JavaScript naming conventions for this repository. We will use camelCase for variables, SCREAMING_SNAKE_CASE for constants. PascalCase for components, and kebab-case for files. Features need to be created on new branches. All new branches must be created off of the dev branch. The naming convention for branches will look like ”<user-story-id>-<user-story-name>”. An example of this branch naming convention is ”A0-users-crud”. Pull requests must be reviewed by each other before being merged to the dev branch. GitHub Actions will also be set up to automatically run test cases. A pull request cannot be merged unless all test cases have passed. We will merge to the dev branch during sprints, and merge to the main branch at the end of each sprint (hotfixes are an exception). We will enforce a minimum of 80% line coverage for our repository.

4.5 Data

Data will be stored in a relational database. The main structure of the data is defined by the Entity Relationship Diagram. We chose to use an SQL database over a NoSQL database because our data has clear relationships (like the relationships between work, checkout, and user).

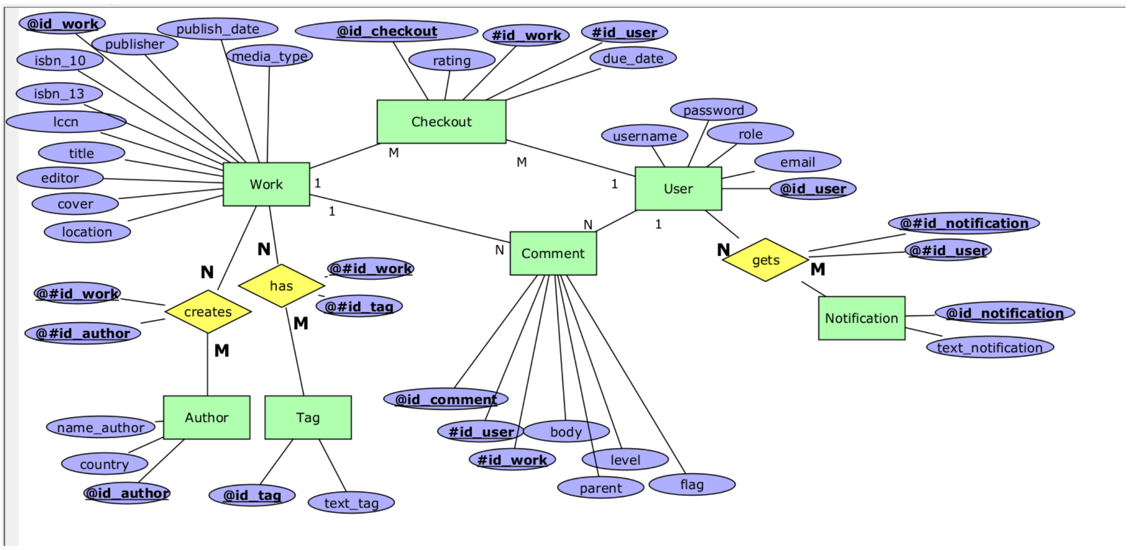


Figure 2: Entity Relationship Diagram

4.6 UI Mocks

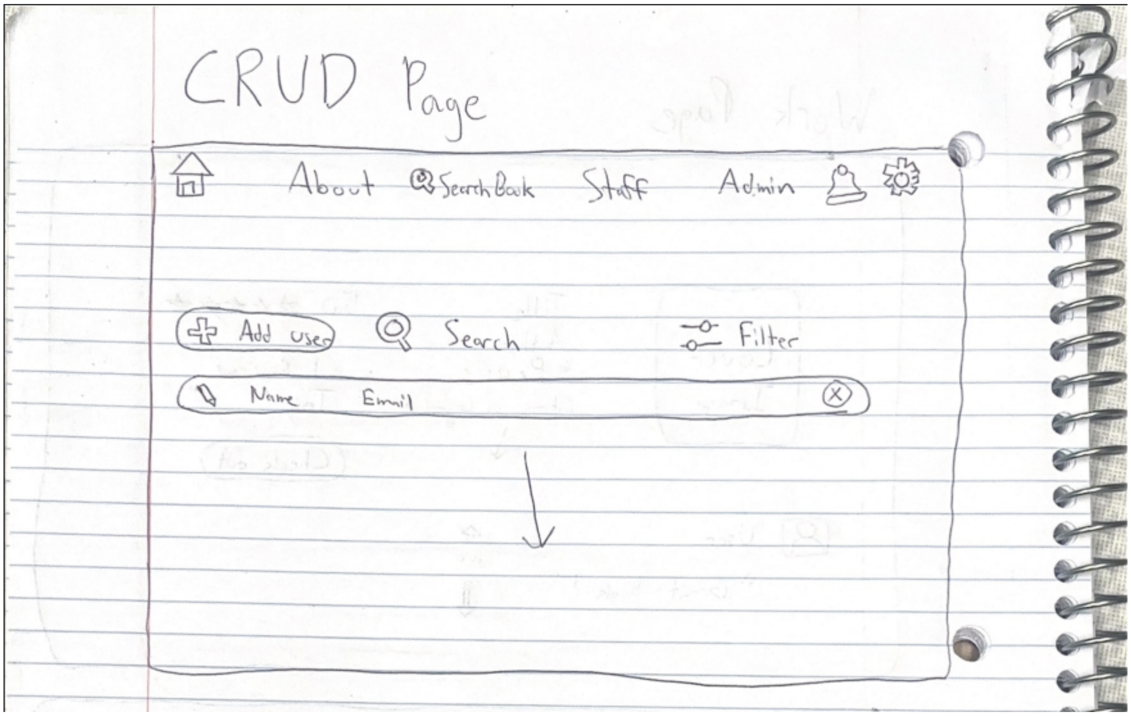


Figure 3: CRUD Mockup (Users and Works)

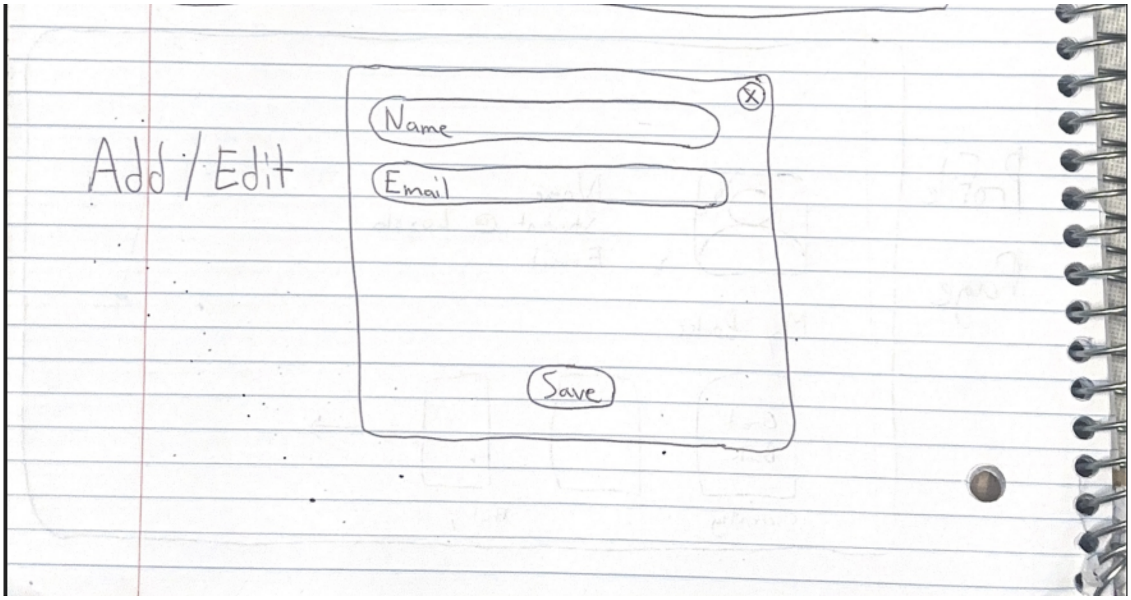


Figure 4: CRUD Modal

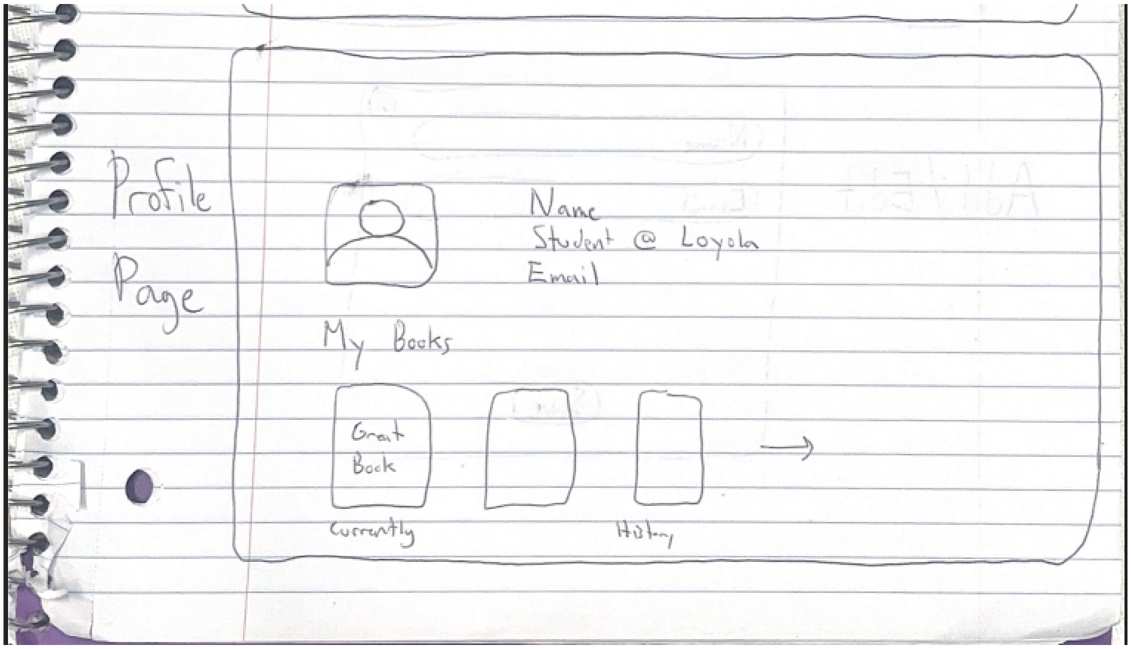


Figure 5: Profile Page

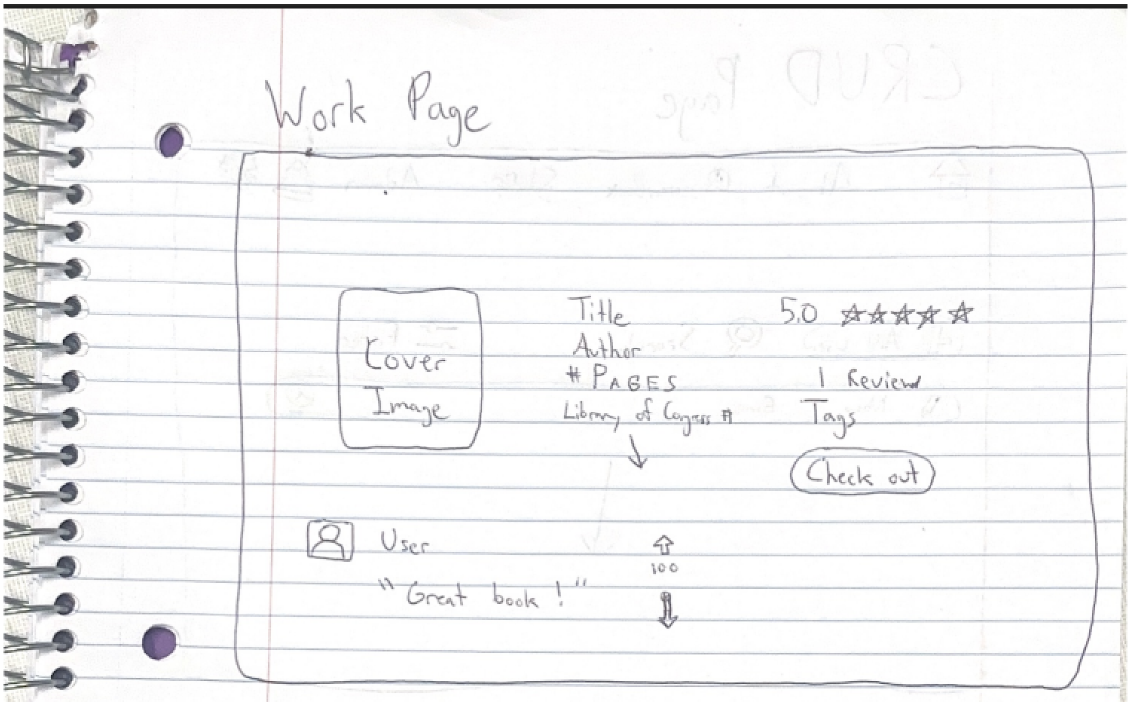


Figure 6: Work Page

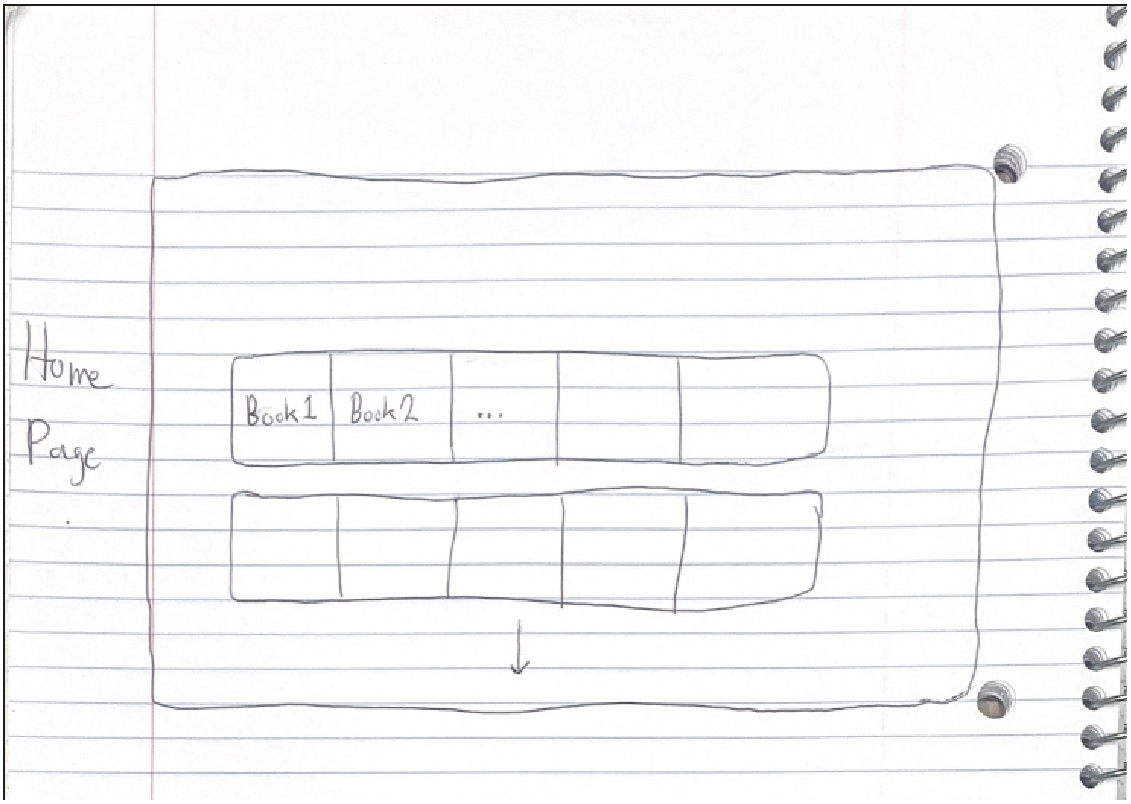


Figure 7: Home Page

5 Iterations

5.1 Iteration Planning

[In CS496, you plan all iterations beforehand. In CS482, you update the planning here at each iteration.]

Table 3 shows the iteration planning.

			Points	
It.	**Dates**	**Stories**	**Planned**	**Done**
1	01/20 - 02/10	A0 Users CRUD, S1 Book CRUD, U4 Search for book	18	
2	02/11 - 02/24	U8 View books, U26 Create account, U7 Reminder email, S11 View checked out books, U31 Log in/log out, S5 Check out/in book, S2 Fill book info via ISBN, S15 Late users, U9 Filter the view of the books, U12 Sort the view of books	17	
3	02/25 - 03/17	S14 Tags CRUD, S3 Scan barcode, U21 Rate book, U24 Request extension, S25 Approve extension	18	
4	03/18 - 03/31	U20 View book covers, U6 Reserve book, U29 Comments CRUD, U27 User get notifications, A28 Admin send notifications, U18 Popular books, U32 Popular books by genre	19	
5	04/01 - 04/14	U10 Recommended books, U23 Book preview, U13 Upvote/downvote comments, U19 Password reset, U16 User view my history, S17 Staff view book history, U30 Report comments, U22 Contact staff	17	
		Total:	89	0

Table 3: Iteration Planning for Incremental Deliveries

5.2 Iteration/Sprint 1

5.2.1 Planning

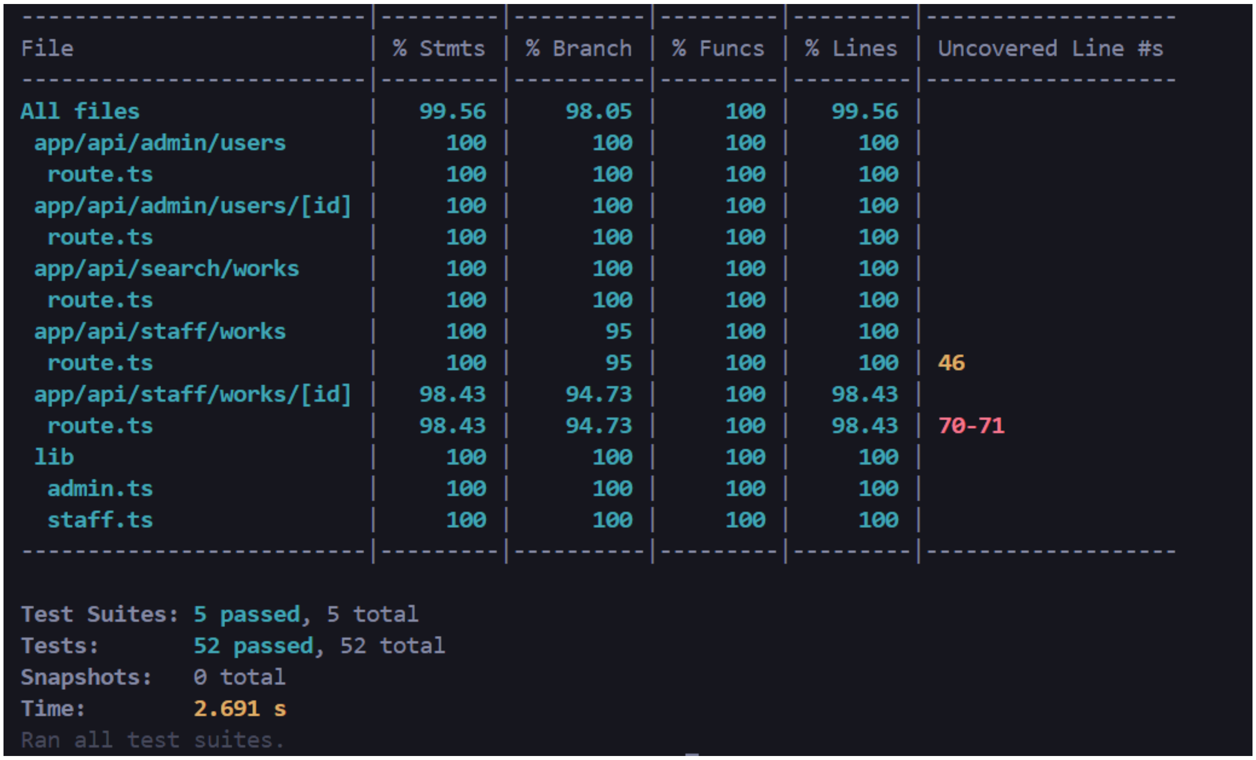
The stories we originally planned for were A0 Users CRUD (8 points), S1 Book CRUD (8 points), U4 Search for book (2 points). We planned for a total of 18 points. The reason why we selected the Users and Book CRUDs were because they were the most important CRUDs of the app. Almost everything else in our app builds on top of these CRUDs. We also added the ability for users to search for books in iteration 1 since it is also a basic requirement that should not be too difficult after setting up a books CRUD.

5.2.2 Work Done

Aidan completed A0 Users CRUD and U26 Create account for a total of 10 points. Vilnis completed S1 Book CRUD and U4 Search for book for a total of 10 points. We completed all of our stories and did not leave any partial work for next sprint. We added U26 to this sprint so that we could test the authentication of the CRUDs (although we didn't originally plan it).

5.2.3 Testing Coverage

Figure 8 shows the test coverage



```
---------------------------|---------|----------|---------|---------|-------------------
File                       | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
---------------------------|---------|----------|---------|---------|-------------------
All files                  |   99.56 |    98.05 |     100 |   99.56 |
 app/api/admin/users       |     100 |      100 |     100 |     100 |
  route.ts                 |     100 |      100 |     100 |     100 |
 app/api/admin/users/[id]  |     100 |      100 |     100 |     100 |
  route.ts                 |     100 |      100 |     100 |     100 |
 app/api/search/works      |     100 |      100 |     100 |     100 |
  route.ts                 |     100 |      100 |     100 |     100 |
 app/api/staff/works       |     100 |       95 |     100 |     100 |
  route.ts                 |     100 |       95 |     100 |     100 | 46
 app/api/staff/works/[id]  |   98.43 |    94.73 |     100 |   98.43 |
  route.ts                 |   98.43 |    94.73 |     100 |   98.43 | 70-71
 lib                       |     100 |      100 |     100 |     100 |
  admin.ts                 |     100 |      100 |     100 |     100 |
  staff.ts                 |     100 |      100 |     100 |     100 |
---------------------------|---------|----------|---------|---------|-------------------

Test Suites: 5 passed, 5 total
Tests:       52 passed, 52 total
Snapshots:   0 total
Time:        2.691 s
Ran all test suites.
```

Figure 8: Iteration 1 test coverage report

We were very pleased with this testing coverage. We included testing for almost every branch in our codebase. Every file has 98% or more statement coverage. One thing our test coverage lacks is explicit front-end testing. However, since we covered all the api endpoints and lib (where our authentication functions live), we are very confident in presenting this iteration.

5.2.4 Retroespective & Reflection

Aidan - A personal pitfall that I faced was using PostgreSQL. I was familiar with everything in our web stack except for this. I am more used to using NoSQL databases like MongoDB. I was excited to use PostgreSQL after finishing CS*485 Database Management Systems at Loyola. Vilnis and I spent time designing the attributes and relationships for some of our key entities. It was nice having more control over our entities.

Vilnis - First, I questioned whether I should have included author as an attribute in the Works table to make the iteration 1 product more presentable. I did not include it since we will make a separate table for Authors and have a relation between it and Works. However, if we were to present iteration 1 to our client, I wonder if they would ask where the author is displayed on the items in our Works CRUD. Second, I faced a challenge programming the search function for users. Initially, I was using something very similar to the GET work API for staff. I realized that this would not look very good because you have to wait to fully type in a search to return something, and if it is not exact, you would get nothing. Therefore, I implemented a search feature that actively updates as the user types in the search bar using SQL ILIKE. I was happy with this solution.

5.3 Iteration/Sprint 2

5.3.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.3.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.3.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.3.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

5.4 Iteration/Sprint 3

5.4.1 Planning

We planned to complete A0 Users CRUD, S1 Book CRUD, U4 Search for Book for a total of 18 story points. We wanted to start with some of the basic CRUDs first because everything else in our app builds on top of them. We added the ability for users to search for books in iteration 1 since it is also a basic requirement that should not be too difficult after setting up a books CRUD.

5.4.2 Work Done

We fully completed all stories in this iteration and U31 Log in/log out. Aidan tackled A0 Users CRUD, U31 Log in/log out and initializing the repo while Vilnis handled S1 Book CRUD and U4 Search for Book.

5.4.3 Testing Coverage

5.4.4 Retrospective & Reflection

5.5 Iteration/Sprint 4

[CS496 has 5 sprints. CS482 only has only 3 sprints (remove Iterations 4 and 5 from this doc if you are writing a doc for 482]

5.5.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.5.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.5.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.5.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

5.6 Iteration/Sprint 5

5.6.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.6.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.6.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.6.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

6 Final Remarks

6.1 Overall Progress

[Have you completed everything? If so, present evidence on how you brought value to your client, and the overall client satisfaction. Otherwise, estimate how much progress you done and how long it would take to finish this project. Be concrete about your progress, you know how many story points your software is, how many points you completed (this shows your progress). You also how many points your team delivers at each iteration, therefore you can estimate how many more iterations it would take to finish the leftover points (show the math).]

6.2 Project Reflection

[Your personal reflection on the project. What lessons did you learned. What would you have done differently? How can you do better work in future projects? You may write this as a team or per person (or both — if all your iterations were team reflections, then it would be better to write individual reflections here)]

Appendix

[Appendix section if needed]