CP476 Project: Limited Books

Quinn Brimley (170781000), Pauline Gwozdz (160393760), Steven Tran(170574740)

2021-04-10

1 Introduction

We wanted to make an online book management system where users can view and purchase their favourite books from the comfort of their own home. The idea was inspired by the pandemic; where people are stuck inside with nothing to do. What better way to spend your quarantine, then sitting back and enjoying a nice book! 😊

2 Problem solving and algorithms

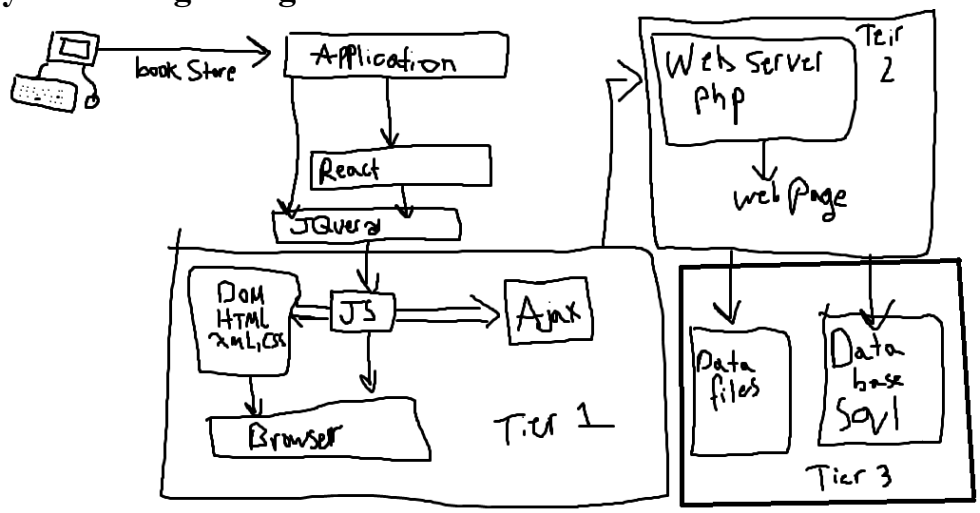
1. **Application data collection**
   We are going to use the **Google Books API** → http://googleapis.com/books/v1/volumes?q= query results of book information. The queried results return a JSON formatted book, that contains the title of the book, the author, the genre, and the cover photo of the image as well.

2. **Storing the books**
   We are going to be storing the books on the mySQL database if the user chooses to save the queried book into their favourites list. The database will keep track of both the user login information, along with the information/settings attatched to the accounts.

3. **Displaying the book information on the page**
   Once the query to the API is successful, the returned JSON contains a URL to the cover of the book –> We would only need to make another HTTP request using AJAX to fetch the image from the web URL and display it to the book result screen. As for the remainder of the information, the returned JSON contains all of the neccessary information, and we need only to parse the JSON string and display the correctly formated information onto the client page.

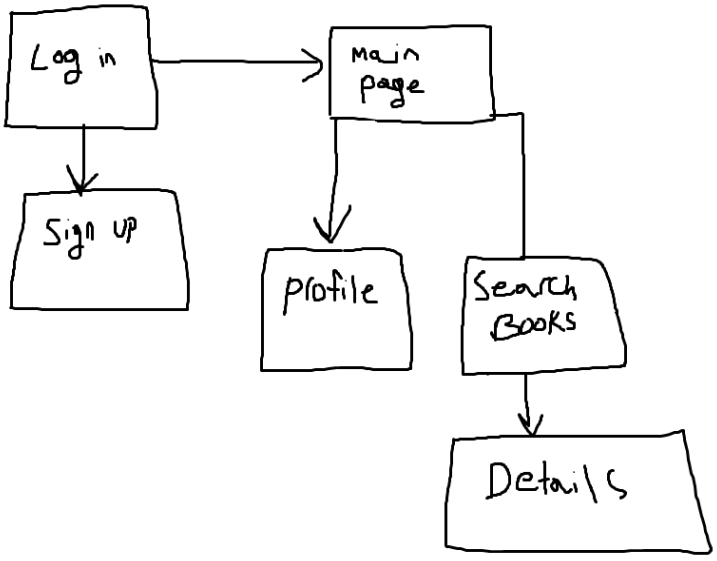
3 System Design

1. **System Design Diagram:**



- Once the web application is launched, the system initialize the DOM elements and make JQuery calls to the APIs to fetch the infomation
- The **1st tier** of the system design consists of what the user can see:
- The user can interact with the HTML webpage through their browser (while under the hood, the system is communicating through JavaScript and AJAX to talk to the server)
- The **2nd tier** of the system design is where the heavy calculations and workload is done
- The system will contain a backend made up of php scripts that launch the correct webpages and send and receive API calls using JQuery and JavaScript
- The **3rd tier** of the system design contains the local storage * After receiving the information from the web, the information is then stored onto the local storage of the application where the user can retreive these files when neccessary (stored onto mySQL database)

2. **Architecture Diagram:**



- Upon launching the application, the user is greeted with a **login/sign up page** where they can enter their credentials and login to their account
- If login/sign up is successful, then the user is taken to the **search page** where they can continue to use the site
- The **search page** contains a button to query their book, and an interactive UI where the user can select one of these book to view in detail
- The search page is where the API calls will be made, where when the user searches for books that they are interested in, the system makes a call to the API and gets the information from the *Google Books API* * Upon clicking on a book, then the system will **load the necessary information onto the webpage** and allow the user to start a **discussion** for selected book, add to their **favourites list**, or click on a button where they can purchase the book (will navigate to the Google Books web application to buy thier book if it's available)

3. **Features to implement: _ Search for books** - Will be using a web API to make HTTP requests to the Google Books API and get book information _ **Redirect Users to where they can buy the books** _ Will implement a button located on the details page that makes another http call to the Google Books API, so you can buy your favourite book! :) _ **Create discussions for books** _ Talk about the plot and thoughts about the book! _ Recommend books with a touch of button (selecting users who are registerd to recommend the book to)

4. **Tools to be used in the project:**

- **CGI programming:**
  - php - Create a mySQL database with tables to store the user information
  - AJAX - Will be used to implement many of the features

- **Server side:**
  - local mySQL database to store:
    - The favourited books
    - User login information
    - Discussions
    - Most Previously viewed book
- **Client side:**
  - JavaScript, JQuery, HTML, CSS
  - API call (Google Books) Will be used to make HTTP requests to get book information and display it to the user

4 Milestones & schedule

All the tasks will be worked on throughout the remainder of the course by everyone. The leader of each milestone meeting will cooridinate what is to be done before the start of the next milestone meeting and check up on everyone's progress thus far.

Task ID	Description	Due date	Lead
1	Project research & team up	Wednesday of week 9	Steve
2	Project proposal	Sunday of week 10	Pauline
3	Get the layout of HTML and PHP scripts working	Friday of Week 10	Quinn
4	Set up datbases and connect everything together	Friday of Week 11	Quinn
5	Project documentation creation	Thursday of week 13	Pauline
6	Project submission	Saturday of week 13	Steve

5 References

1. OpenLibrary Book API
2. Indigo
3. Google Book API
4. Closed Book Image
5. Open Book Image

6 Appendices

1. Password Encryption (Built-in php hashing) - To securely store the password in the database
2. Will implement HTTPS for safer transactions