

## Problem

The college library maintains books of various departments like Computer engineering, Electronics engineering, etc. The library staff maintains records of all books, return and issue of books to students and teachers. It was becoming tedious for teachers and students to search for a book in library. The librarian decided to provide online service for students and teachers to search availability of the books. The system will search for the availability of the given book and display if the book is available/not available with its shelf number(location of the book).Your task is to design a software for the requirement stated by librarian, using appropriate data structure and algorithm.

### Your design for the above problem-

Phases	Your answer
1. Understand the problem from multiple perspectives	<p>The diagram(model) of the system drawn-</p> <pre> graph LR     Librarian -- "inserts which book is available" --&gt; books[books data structure]     books -- "checks the availability of books" --&gt; Server     Server -- "returns the availability of book" --&gt; books     Students[Students/teachers] -- "enters credentials" --&gt; login[login page]     login -- "Allows student to access" --&gt; Server     login -- "checks whether credentials are correct" --&gt; login     Students -- "searches whether the books are available" --&gt; Server   </pre> <p>GoJS 1.7 evaluation (c) 1996-2017 Northwoods Software Not for distribution or production use nwoods.com</p> <p><b>Entities-</b>books data structure ,Students/teachers ,login page ,Server ,Librarian ,  <b>Relations-</b> returns the availability of book,checks the availability of books,inserts which book is available,searches whether the books are available,Allows student to access ,checks whether credentials are correct,enters credentials,</p>

<b>2. Formulate the problem</b>	<p><b>Broad Goal</b>-To make it easy for students to find the book in library The students and teachers of the college So that students can see whether the book they want is available in the library</p> <p><b>Data</b>-data_storage ;;users ;;login ;;server ;;librarian ;;</p> <p><b>Operations</b>-available() ;;search(section, book_name) ;;credential_check(username, password) ;;insert(section, book_name, quantity) ;;</p>
<b>3.1 Solution map</b>	<p>The attribute listing map you generated-</p> <p>The diagram illustrates the data structure and operations for a library system. It shows a hierarchy starting from 'Data Items' which branches into 'Data Storage' and 'Login'. 'Data Storage' further branches into 'Section' and 'Data Items'. 'Section' branches into 'Books', 'Electronics', and 'Mechanics'. Each of these categories has a 'shelf_no' (node 2) and 'quantity' (node 3). 'Login' branches into 'username' (node 1) and 'password' (node 2). 'Operations' branches into 'available()', 'search(section, book_name)', 'credential_check(username, password)', and 'insert(section, book_name, quantity)'. Each operation is linked to a 'Write Algorithm' or 'Write A' task.</p>
<b>3.2 Generate Solutions</b>	<p><b>Solution 1</b> - +data_storage -&gt;tree, users -&gt;none, login -&gt;linked list, server -&gt;none, librarian -&gt;none, +-&gt;, +available() -&gt;if the book is available, it will give shelf_no as output, search(section, book_name) -&gt;it takes in the section and book_name, traverses through the section and then goes to the appropriate book by using book name, credential_check(username, password) -&gt;It will check the username and password in the linked list, insert(section, book_name, quantity) -&gt;it takes in the section and book_name, traverses through the section, goes to the appropriate book by using book name and then uploads the quantity, +-&gt; ,</p>
<b>4. Analyze solutions</b>	<p><b>Criteria used to analyze solutions-</b></p>
<b>5. Identify constraints</b>	<p><b>Constraints identified for the given problem-</b></p> <p><b>1. Criteria-</b></p> <p><b>Constraint-</b> Execution time of search&amp;Update(item quantity) operation should be low</p>

	<b>Justify-</b> to avoid wastage of timing of the user <b>2. Criteria-</b> <b>Constraint-</b> Execution time of login should be low <b>Justify-</b> for a user friendly experience <b>3. Criteria-</b> <b>Constraint-</b> Searching should be less time consuming <b>Justify-</b> the whole website is about searching
<b>6. Evaluate and select</b>	<b>The selected solution which is acheiving the goal/sub-goals and constraints is-</b> Justified as-

You have designed the solution by selecting appropriate data structure and algorithms for solving the above problem.

Now proceed to implement the solution using appropriate programming language.