Dominic Latty
February 9, 2007
CS 190

Book Locator Specifications

I. Description

There is no doubt that Brown University's Rockefeller Library possesses an immense wealth of information. Many students at Brown are eager to use the Rock's resources to aid them in their journey for knowledge, but are often intimidated by the seemingly daunting task of finding a single book among the hundreds of shelves stacked throughout the facility. While Brown's call number system ameliorates this to some extent, it is only certain in its ability to reduce a text's location to a specific floor. It is often still quite difficult to find the exact shelf without wandering aimlessly throughout the floor.

Book Locator will eliminate this problem for Brown University students forever. Whenever a student looks for a text at the Rock, they will now have the option of viewing a floor map that highlights the book's shelf location accompanied by walking directions. With Book Locator, Brown University students will no longer have reason to fear easily retrieving the texts they need from the Rock.

II. Feature Set

Primary:

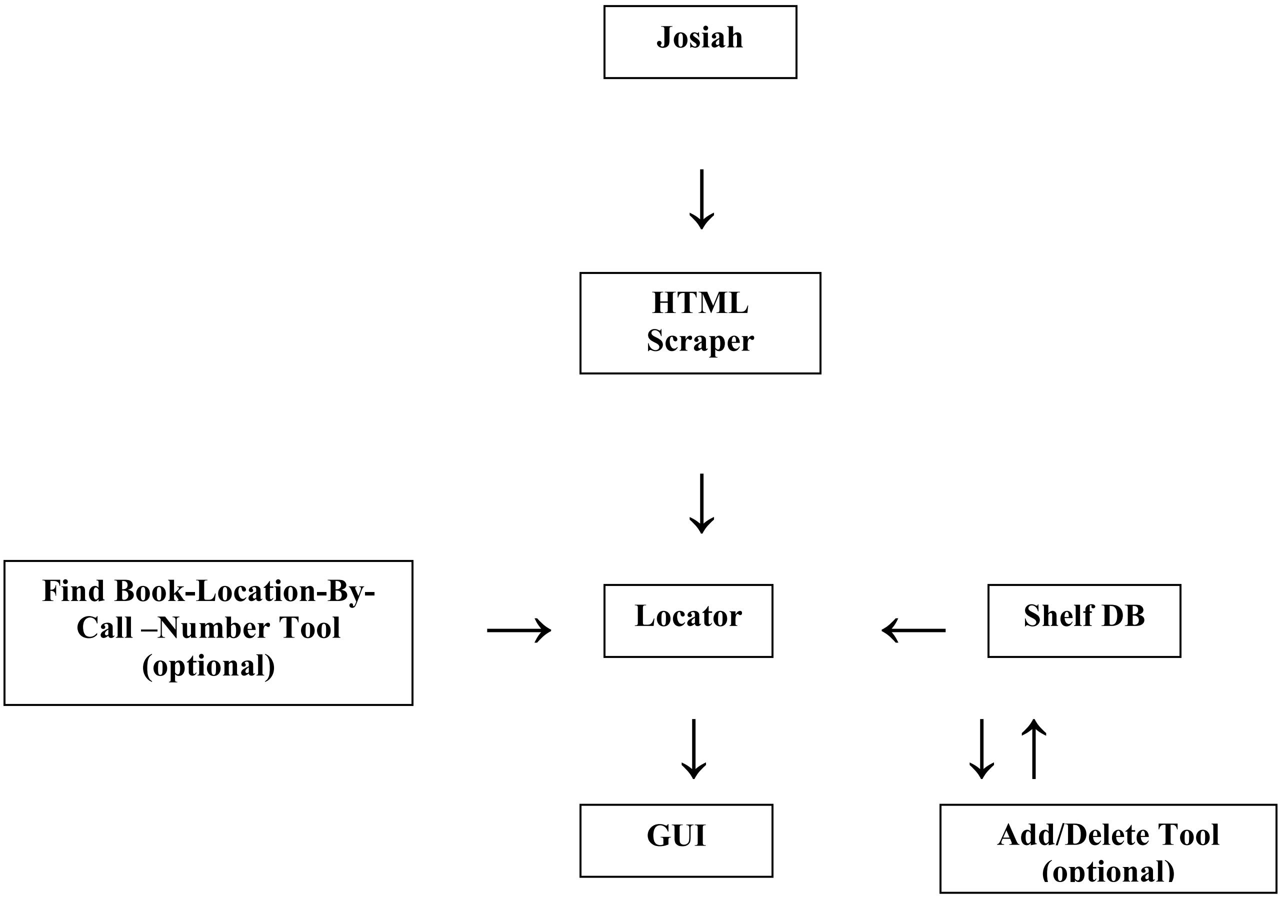
When a user searches for a text in the Rockefeller Library on Josiah:

- The user will see a floor map displaying all the shelves on that given floor.
- The general area where the book is located will be highlighted. At this stage, it is not certain whether the specific shelf will be highlighted, or if the book's location can be reduced to two or three adjacent shelves.
- Walking directions to the book's location can also be retrieved.
- System based on a database that keeps track of all stacks in the Rock

Optional:

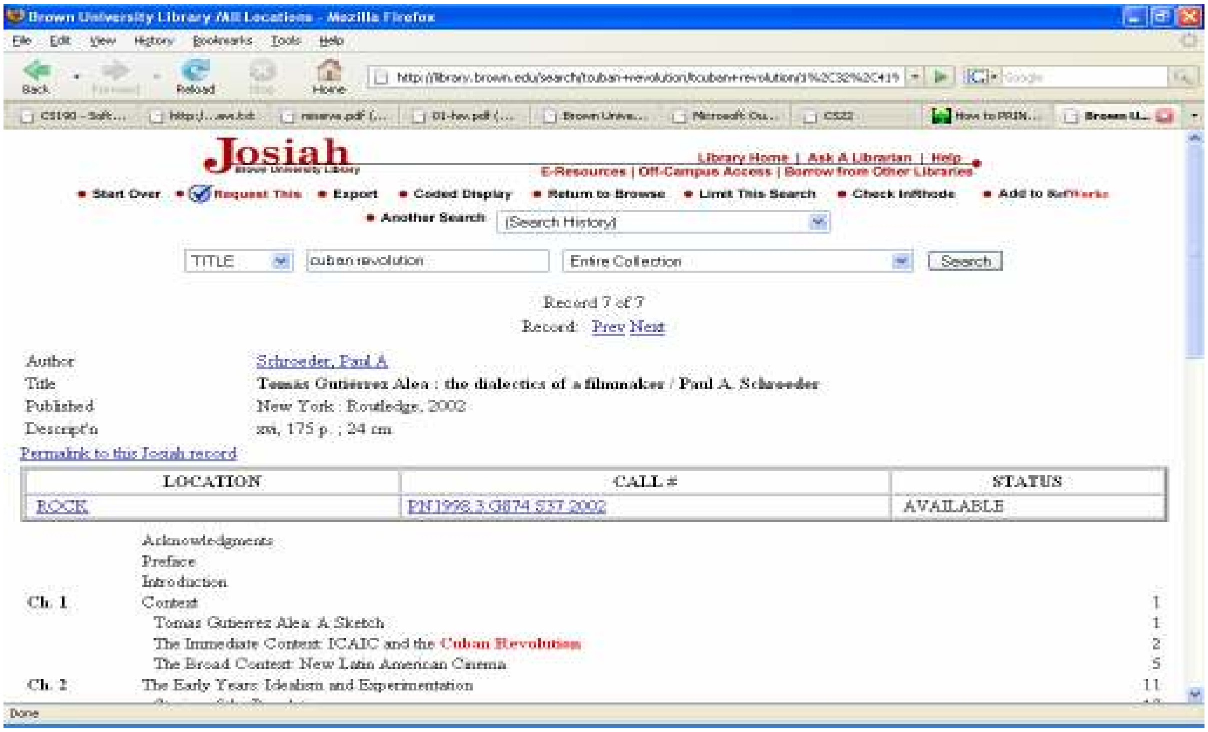
- Users can manually enter a book's call number to retrieve the book's highlighted location with walking directions.
- An add/delete shelf feature for use by librarians. This feature would consist of a GUI that allows librarians to add and remove shelves from sections of the library.

III. System Model Diagram


Josiah
HTML Scraper
Find Book-Location-By-Call –Number Tool (optional)
Locator
Shelf DB
GUI
Add/Delete Tool (optional)

IV. User Interface Diagram

- Beginning page with button that generates the “Result Page”:



cuban revolution

Record 7 of 7
Record: Prev Next

Author	Schroeder, Paul A
Title	Tomas Gutierrez Alea : the dialectics of a filmmaker / Paul A. Schroeder
Published	New York : Routledge, 2002
Descript'n	xvi, 175 p. ; 24 cm

Permalink to this Josiah record

LOCATION	CALL #	STATUS
ROCK	PN1998.3.G874 S37 2002	AVAILABLE

	Acknowledgments	
	Preface	
	Introduction	
Ch. 1	Context	1
	Tomas Gutierrez Alea: A Sketch	1
	The Immediate Context: ICAIC and the Cuban Revolution	2
	The Broad Context: New Latin American Cinema	5
Ch. 2	The Early Years: Idealism and Experimentation	11

Description:

From this page, one in which a specific text has been found, there will be an added button (probably below the 3X2 table) that, once, clicked, will go a floor map (see February 5, 2007 presentation) is displayed and the book’s location (with walking directions) is highlighted. The floor map can also be accessed by a simple optional tool consisting of a text field and button, where the user enters the book’s call number.

V. Non-functional Requirements

i. Performance/Reliability

a) Because Book Locator will be used by countless Brown students, it should be very reliable with no noticeable hiccups in performance.

b) The algorithm for finding the quickest path (in the Locator component) must run efficiently (preferably milliseconds, absolutely < ~5 seconds)

ii. Testing

a) When testing the optional database front end GUI, the database should be updating itself as the user adds or deletes shelves and these changes should be shown in the GUI.

iii. Ease of Use

a) Should extremely easy to use.

b) Accessible to any Brown student, computer-savvy or not

iv. Documentation

a) Minimal documentation is required, as it is a simple search tool

v. Dependencies on other systems

a) Josiah network (HTML scraper)

VI. Updated requirements

i. Book Locator must have a component that scrapes HTML code off the search result page on Josiah in order to read the book's call number.

ii. The stacks must be stored in the database per floor, per call number range, and are each divided into two sides.

a) Each stack contains books whose call numbers fall within a certain range. The stack should be represented in the database using the bounds of the call number range.

b) Many different types of data structures can be used to store the stacks in the Rockefeller Library (i.e. hash table, tree, vector)

iii. The system's database should keep track of how the special 1-size (>28cm) and 2-size (>40cm) books are stored in each section.

a) A tricky situation: some stacks store the special sized books separately while have regular and special sized books mixed together.

b) The system must be aware of the different ways in which special sized books are stored, per stack.

iv. ***optional*** The user must be able to enter a valid call number directly through a simple tool

a) Must follow XXX#### format

v. ***optional*** Requirements for the add/delete shelf feature to function efficiently:

a) Each floor of the Rock must be divided into sections.

b) The sections can be perhaps ordered by orientation. In other words, sections will either consist of only vertically-aligned or horizontally-aligned shelves.

c) Each section can store a maximum amount of shelves (later to be determined).

d) Librarians who need to add or delete shelves will simply select the floor section and increase or decrease the amount of shelves in that section.

e) Through this tool, the shelf numbers that correspond to the range of call numbers that can be found in the shelf can be altered.

f) Through this tool, the shelf length can be increased or decreased.

VII. Divisibility

Because of the nature of Book Locator's components, the project is very divisible. The components can be divided into: the database of shelves; the HTML scraper that collects the book's call number; the "locator algorithm" that calculates walking directions to the book's location; the UI that displays the book's location along with the walking directions; the optional find-location-by-call-number feature; and the optional front end add/delete tool.

VIII. Specific Challenges

The adding and deleting of shelves may pose a significant challenge. It may be difficult to provide a GUI that allows for librarians to add and delete shelves, as well as increase or decrease each shelf's size while still maintaining an accurate portrayal of the floor layouts of the Rock.

One feasible idea is to graphically represent the floor as a grid with sections where shelves cannot be possibly stored (i.e. study rooms, bathrooms, stair wells, elevators) blocked off. The user can then add shelves by clicking the start coordinate of the grid and dragging the mouse until the shelf (represented by a line) is of the desired size. Removing the shelves could be done by clicking the desired shelf and pressing a key. Utilizing a grid could also come in handy when designing the algorithms to draft walking directions to the book's location, considering that each "square" in the grid would correspond to an actual length.