



Library Book Borrowing System

Basic CRUD Operations with MongoDB

Project Overview

This presentation introduces a straightforward library system, enabling students to efficiently borrow and return books.



Student-Centric

Designed for ease of use by students.



Book Management

Comprehensive tracking of book availability.



Core Data Collections

Manages Students, Books, BorrowRecords, and Librarians.



CRUD Functionality

Supports fundamental Create, Read, Update, and Delete operations.

Key Objectives

Our system focuses on maintaining accurate records and streamlining the borrowing process.

Maintain Accurate Book Records

Ensure all book details are current and easily accessible.

Track Student Borrowings

Monitor which student has borrowed which book at any given time.

Record Dates

Store precise borrow and return dates for all transactions.

Prevent Duplicate Borrowings

Implement checks to ensure a student cannot borrow the same book twice simultaneously.

Clean Database Structure

Maintain an organised and efficient database for optimal performance.

System Features

The system provides a suite of features to manage library operations effectively.

Add New Students

Register new student profiles with essential contact information.

Add & Update Books

Easily add new titles or modify existing book details and availability.

Borrowing Books

Streamlined process for students to check out books.

Returning Books

Effortless procedure for students to return borrowed items.

View All Borrow Records

Access a comprehensive log of all borrowing transactions.

MongoDB Collections Overview

Our system utilises four distinct MongoDB collections to store and manage library data.



Students

Stores student details:

`studentId, name, class,
phone.`



Books

Contains book information:

`bookId, title, author,
genre, availableCopies.`



BorrowRecords

Tracks borrowing activities:

`recordId, studentId,
bookId, borrowDate,
returnDate, status.`



Librarians

Librarian profiles:

`librarianId, name, phone.`



Core CRUD Operations

Understanding the fundamental MongoDB operations that power our library system.

1

Create

Insert new documents into collections (e.g., adding a new student or book).

2

Read

Use the `find()` method to [retrieve specific data](#) (e.g., searching for a book).

3

Update

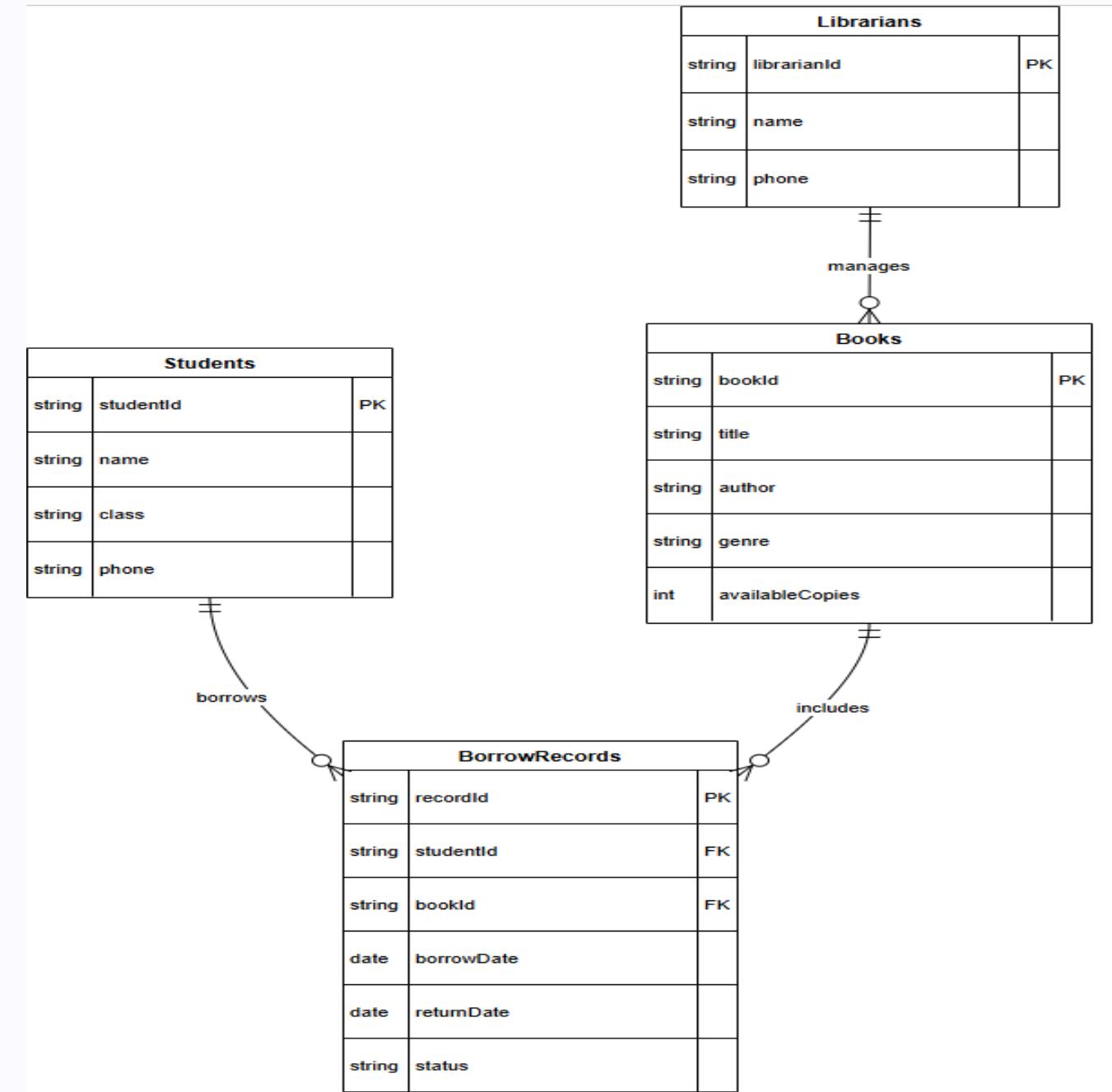
Employ `$set` to modify fields or `$inc` to increment values like available copies.

4

Delete

Utilise `deleteOne()` to [remove documents](#), such as a completed borrow record.

Entity-Relationship Diagram



This diagram illustrates the relationships between the different entities within our library database, highlighting how students interact with books through borrowing records, and how librarians manage the book collection.

Flow of Borrowing a Book

A step-by-step process outlining how a student can successfully borrow a book from the system.

Student Selects Book

The student identifies the desired book.

Check Availability

The system verifies if the book has available copies.

Create Borrow Record

A new record is generated, linking the student and the book.

Reduce Available Copies

The count of available copies for the book is decremented by one.



```
7  const MongoClient = require('mongodb').MongoClient;
19  const url = "mongodb://127.0.0.1:27017";
20  const client = new MongoClient(url);
21  client.connect();
22  const db = client.db('library');
23  const booksCollection = db.collection('books');
24  const borrowRecordsCollection = db.collection('borrowRecords');
25  const usersCollection = db.collection('users');
26  const genresCollection = db.collection('genres');
27  const authorsCollection = db.collection('authors');
28  const availableCopies = 5;
29  const availableCopies2 = 3;
30  const availableCopies3 = 1;
31  const availableCopies4 = 2;
32  const availableCopies5 = 4;
33  const availableCopies6 = 3;
34  const availableCopies7 = 2;
35  const availableCopies8 = 1;
36  const availableCopies9 = 0;
37  const availableCopies10 = 1;
38  const availableCopies11 = 0;
39  const availableCopies12 = 1;
40  const availableCopies13 = 0;
41  const availableCopies14 = 1;
42  const availableCopies15 = 0;
43  const availableCopies16 = 1;
44  const availableCopies17 = 0;
45  const availableCopies18 = 1;
46  const availableCopies19 = 0;
47  const availableCopies20 = 1;
48  const availableCopies21 = 0;
49  const availableCopies22 = 1;
50  const availableCopies23 = 0;
51  const availableCopies24 = 1;
52  const availableCopies25 = 0;
53  const availableCopies26 = 1;
54  const availableCopies27 = 0;
55  const availableCopies28 = 1;
56  const availableCopies29 = 0;
57  const availableCopies30 = 1;
58  const availableCopies31 = 0;
59  const availableCopies32 = 1;
60  const availableCopies33 = 0;
61  const availableCopies34 = 1;
62  const availableCopies35 = 0;
63  const availableCopies36 = 1;
64  const availableCopies37 = 0;
65  const availableCopies38 = 1;
66  const availableCopies39 = 0;
67  const availableCopies40 = 1;
68  const availableCopies41 = 0;
69  const availableCopies42 = 1;
70  const availableCopies43 = 0;
71  const availableCopies44 = 1;
72  const availableCopies45 = 0;
73  const availableCopies46 = 1;
74  const availableCopies47 = 0;
75  const availableCopies48 = 1;
76  const availableCopies49 = 0;
77  const availableCopies50 = 1;
78  const availableCopies51 = 0;
79  const availableCopies52 = 1;
80  const availableCopies53 = 0;
81  const availableCopies54 = 1;
82  const availableCopies55 = 0;
83  const availableCopies56 = 1;
84  const availableCopies57 = 0;
85  const availableCopies58 = 1;
86  const availableCopies59 = 0;
87  const availableCopies60 = 1;
88  const availableCopies61 = 0;
89  const availableCopies62 = 1;
90  const availableCopies63 = 0;
91  const availableCopies64 = 1;
92  const availableCopies65 = 0;
93  const availableCopies66 = 1;
94  const availableCopies67 = 0;
95  const availableCopies68 = 1;
96  const availableCopies69 = 0;
97  const availableCopies70 = 1;
98  const availableCopies71 = 0;
99  const availableCopies72 = 1;
100 const availableCopies73 = 0;
101 const availableCopies74 = 1;
102 const availableCopies75 = 0;
103 const availableCopies76 = 1;
104 const availableCopies77 = 0;
105 const availableCopies78 = 1;
106 const availableCopies79 = 0;
107 const availableCopies80 = 1;
108 const availableCopies81 = 0;
109 const availableCopies82 = 1;
110 const availableCopies83 = 0;
111 const availableCopies84 = 1;
112 const availableCopies85 = 0;
113 const availableCopies86 = 1;
114 const availableCopies87 = 0;
115 const availableCopies88 = 1;
116 const availableCopies89 = 0;
117 const availableCopies90 = 1;
118 const availableCopies91 = 0;
119 const availableCopies92 = 1;
120 const availableCopies93 = 0;
121 const availableCopies94 = 1;
122 const availableCopies95 = 0;
123 const availableCopies96 = 1;
124 const availableCopies97 = 0;
125 const availableCopies98 = 1;
126 const availableCopies99 = 0;
127 const availableCopies100 = 1;
128 const availableCopies101 = 0;
129 const availableCopies102 = 1;
130 const availableCopies103 = 0;
131 const availableCopies104 = 1;
132 const availableCopies105 = 0;
133 const availableCopies106 = 1;
134 const availableCopies107 = 0;
135 const availableCopies108 = 1;
136 const availableCopies109 = 0;
137 const availableCopies110 = 1;
138 const availableCopies111 = 0;
139 const availableCopies112 = 1;
140 const availableCopies113 = 0;
141 const availableCopies114 = 1;
142 const availableCopies115 = 0;
143 const availableCopies116 = 1;
144 const availableCopies117 = 0;
145 const availableCopies118 = 1;
146 const availableCopies119 = 0;
147 const availableCopies120 = 1;
148 const availableCopies121 = 0;
149 const availableCopies122 = 1;
150 const availableCopies123 = 0;
151 const availableCopies124 = 1;
152 const availableCopies125 = 0;
153 const availableCopies126 = 1;
154 const availableCopies127 = 0;
155 const availableCopies128 = 1;
156 const availableCopies129 = 0;
157 const availableCopies130 = 1;
158 const availableCopies131 = 0;
159 const availableCopies132 = 1;
160 const availableCopies133 = 0;
161 const availableCopies134 = 1;
162 const availableCopies135 = 0;
163 const availableCopies136 = 1;
164 const availableCopies137 = 0;
165 const availableCopies138 = 1;
166 const availableCopies139 = 0;
167 const availableCopies140 = 1;
168 const availableCopies141 = 0;
169 const availableCopies142 = 1;
170 const availableCopies143 = 0;
171 const availableCopies144 = 1;
172 const availableCopies145 = 0;
173 const availableCopies146 = 1;
174 const availableCopies147 = 0;
175 const availableCopies148 = 1;
176 const availableCopies149 = 0;
177 const availableCopies150 = 1;
178 const availableCopies151 = 0;
179 const availableCopies152 = 1;
180 const availableCopies153 = 0;
181 const availableCopies154 = 1;
182 const availableCopies155 = 0;
183 const availableCopies156 = 1;
184 const availableCopies157 = 0;
185 const availableCopies158 = 1;
186 const availableCopies159 = 0;
187 const availableCopies160 = 1;
188 const availableCopies161 = 0;
189 const availableCopies162 = 1;
190 const availableCopies163 = 0;
191 const availableCopies164 = 1;
192 const availableCopies165 = 0;
193 const availableCopies166 = 1;
194 const availableCopies167 = 0;
195 const availableCopies168 = 1;
196 const availableCopies169 = 0;
197 const availableCopies170 = 1;
198 const availableCopies171 = 0;
199 const availableCopies172 = 1;
200 const availableCopies173 = 0;
201 const availableCopies174 = 1;
202 const availableCopies175 = 0;
203 const availableCopies176 = 1;
204 const availableCopies177 = 0;
205 const availableCopies178 = 1;
206 const availableCopies179 = 0;
207 const availableCopies180 = 1;
208 const availableCopies181 = 0;
209 const availableCopies182 = 1;
210 const availableCopies183 = 0;
211 const availableCopies184 = 1;
212 const availableCopies185 = 0;
213 const availableCopies186 = 1;
214 const availableCopies187 = 0;
215 const availableCopies188 = 1;
216 const availableCopies189 = 0;
217 const availableCopies190 = 1;
218 const availableCopies191 = 0;
```

Simple MongoDB Example

Illustrative code snippets showcasing basic MongoDB operations for our library system.

Adding New Books

```
db.books.insertMany([
  { bookId: "B001", title: "The Great Novel", author: "A. Author", genre: "Fiction", availableCopies: 5 },
  { bookId: "B002", title: "Tech Guide", author: "B. Coder", genre: "Technology", availableCopies: 3 }
]);
```

Creating a Borrow Record

```
db.borrowRecords.insertOne({
  recordId: "BR001", studentId: "S101", bookId: "B001",
  borrowDate: ISODate("2023-10-26"), returnDate: null, status: "borrowed"
});
```

Updating Available Copies after borrowing:

```
db.books.updateOne( { bookId: "B001" }, { $inc: { availableCopies: -1 } } );
```



Conclusion

This system lays a robust foundation for a scalable library management application.

Basic Working System

Successfully implements core library functionalities.

CRUD Operations Completed

Full Create, Read, Update, and Delete capabilities are in place.

Expandable for Future Growth

Ready for additional features and full application development.