



Báo cáo bài tập môn

Cơ sở dữ liệu tiên tiến

Xây dựng hệ thống e-Library

phân tán nhiều cơ sở



Giảng viên hướng dẫn: TS. Nguyễn Duy Hải

Nhóm học viên số 10: 1. Trương Tuấn Nghĩa

2. Phạm Mạnh Thắng

3. Lưu Anh Tú

Thư viện số e-Library

Mượn sách mọi lúc, mọi nơi!

01	Tổng quan về hệ thống
02	Phân tích và thiết kế hệ thống
03	Cài đặt và đánh giá

01

Tổng quan về hệ thống

Hệ thống thư viện số



Hà Nội



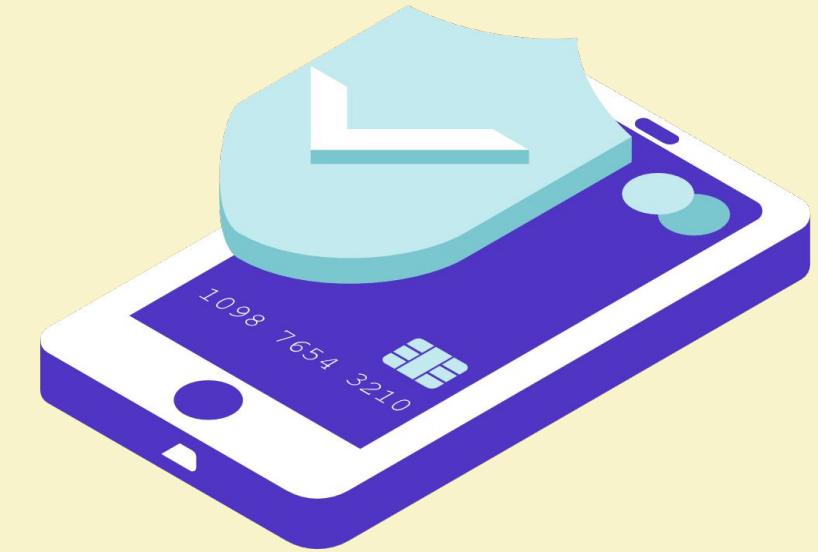
Đà Nẵng



TP.Hồ Chí Minh



Bối cảnh và giải pháp



Bối cảnh:

- Thời đại số hóa
- Dữ liệu lớn, yêu cầu nhất quán về dữ liệu
- Phạm vi quản lý rộng
- Tính thống nhất về mặt nghiệp vụ
- Sự tiện nghi, tối ưu thời gian tìm kiếm và chi phí quản lý nhân sự



Giải pháp:

- Tiết kiệm thời gian quản lý sách, mượn - trả sách.
- Phân tán và đồng bộ:
 - + lưu trữ dữ liệu lớn
 - + giảm thời gian tải thao tác đọc ghi
 - + sẵn sàng dữ liệu
- Thống nhất về mặt nghiệp vụ quản lý sách ở nhiều cơ sở
- Sự tiện nghi, tối ưu thời gian tìm kiếm và chi phí quản lý nhân sự

Các yêu cầu hệ thống

Hệ thống e-Library phân tán nhiều cơ sở

1. Quản lý sách, người mượn, lượt mượn, lịch sử giao dịch.
2. Thiết kế hệ thống phân tán 3 node (Hà Nội – Đà Nẵng – TP. Hồ Chí Minh).
3. Tối ưu truy vấn tìm kiếm bằng index + text search NoSQL.
4. Đồng bộ dữ liệu tự động khi có giao dịch mượn/trả.
5. Yêu cầu: MongoDB Replica Set + Full-text Search.

Các quy trình nghiệp vụ

1. Đăng nhập

- Người dùng đăng nhập vào ứng dụng với Username và password
- Phân quyền với giao diện tương ứng
- Lưu phiên đăng nhập và thời hạn 24h

2. Tìm kiếm sách

- Vào danh sách sách
- Nhập tìm kiếm theo tên sách, thể loại, tác giả

3. Quản lý người dùng

- Admin quản lý người dùng, xem các thông tin người dùng, lịch sử giao dịch

4. Quản lý sách

- Admin từ trung tâm sách thực hiện: xem, tìm kiếm, thêm mới, sửa, cập nhật thông tin và trạng thái hoạt động của sách
- Admin từ chi nhánh thực hiện: xem, tìm kiếm, báo cáo cập nhật số lượng sách bị thất thoát

5. Mượn, Trả sách

- TH1: Đơn mượn sách -> thanh toán -> nhận sách -> trả sách (paid -> success -> returned)
- TH2: Đơn mượn sách -> thanh toán -> hủy đơn (paid -> cancelled)

Phân tích và thiết kế hệ thống

Quy trình mượn, trả sách

1. Tạo đơn mượn

- (1) Sinh viên chọn sách vào giỏ hàng
- (2) Xác nhận mượn và thanh toán
- (3) Đơn mượn được tạo thành công -> hiển thị trong mục “quản lý đơn mượn” của nhân viên thư viện (paid)

2. Hủy đơn mượn

- (1) Sinh viên có thể hủy đơn mượn khi chưa đến nhận sách
- (2) Đơn bị hủy biến mất khỏi mục quản lý đơn mượn phía nhân viên thư viện (cancelled)

3. Xác nhận đơn mượn sách

- (1) Sinh viên đến thư viện nhận sách từ thủ thư
- (2) Thủ thư cập nhật trên hệ thống đơn mượn thành “đang mượn” (success)

4. Trả sách

- (1) Sinh viên đến trả sách
- (2) Nhân viên thư viện nhận sách, kiểm tra tình trạng sách
- (3) Cập nhật trạng thái “đã trả” của đơn mượn trên hệ thống (returned)

02

Phân tích và thiết kế hệ thống

Phân tích và thiết kế hệ thống

Các yêu cầu chức năng

Bảng 2.1: Danh sách yêu cầu chức năng

STT	Chức năng	Actor	Mô tả
1	Đăng ký tài khoản	Customer	Tạo account mới
2	Đăng nhập/Đăng xuất	All	JWT authentication
3	Quản lý sách (CRUD)	Admin	Thêm/Sửa/Xóa sách
4	Tìm kiếm sách	All	Full-text search
5	Quản lý người dùng	Admin	CRUD users
6	Giỏ hàng	Customer	Thêm/Xóa sách
7	Đặt mượn sách	Customer	Tạo đơn mượn
8	Lịch sử mượn	Customer	Xem orders
9	Dashboard thống kê	Admin	Charts, reports
10	Đồng bộ dữ liệu	System	Sync branches

Phân tích và thiết kế hệ thống

Thiết kế CSDL

1. users - Thông tin người dùng
2. books - Danh mục sách
3. carts - Giỏ hàng
4. orders - Đơn mượn sách

Bảng: Mối quan hệ giữa các collection

Collection	Quan hệ	Ý nghĩa
users– carts	1:1	Một người dùng chỉ có một giỏ hàng
users– orders	1:N	Một người dùng có thể tạo nhiều đơn mượn sách. Mỗi đơn mượn lưu user_id để truy vấn ngược.
carts– books	N:M	Một giỏ hàng có thể chứa nhiều sách, và một đầu sách có thể xuất hiện trong giỏ hàng của nhiều người dùng cùng lúc (embedded trong items[]).
orders– books	N:M	Một đơn mượn có thể chứa nhiều sách, và một đầu sách có thể xuất hiện trong nhiều đơn mượn khác nhau (embedded trong items[])

Phân tích và thiết kế hệ thống

Thiết kế CSDL

Collection users - Thông tin người dùng

```
▶ _id: ObjectId('694fce9831f0f2802b0b8ab6')
  username : "ducpm"
  display_name : "ducpm"
  password : "$2y$10$wlA81rf1NQvS6TFaNtMMseDa3ml5CnyqnYIGCkgFQqjnGCgoSXB/y"
  role : "customer"
  balance : 232000
  branch_id : "Hà Nội"
  created_at : 1999-12-31T17:00:00.000+00:00
```

Phân tích và thiết kế hệ thống

Thiết kế CSDL

Collection books - Danh mục sách

```
_id: ObjectId('694fa6518fe3dc31379be3d3')
bookCode : "00001"
bookGroup : "Khoa học"
bookName : "Vật lý lượng tử nhập môn"
location : "Hà Nội"
quantity : 50
pricePerDay : 2000
borrowCount : 0
status : "active"
created_at : 2025-12-27T00:00:00.000+00:00
```

Phân tích và thiết kế hệ thống

Thiết kế CSDL

Collection carts - Giỏ hàng

```
_id: ObjectId('694ff92c2e25f8c3fb0012e8')
user_id : ObjectId('694fd29a749704794c09dba6')
- items : Array (empty)
  updated_at : 2025-12-27T15:27:21.364+00:00
```

```
▶ _id: ObjectId('694ff9792e25f8c3fb0012ea')
  user_id : ObjectId('694fd2be749704794c09dba9')
  - items : Array (3)
    ▶ 0: Object
      book_id : ObjectId('694faa08d4871065baed1eec')
      bookCode : "373"
      bookName : "Bí mật ngôi sao nhỏ"
      pricePerDay : 3000
      quantity : 1
    ▶ 1: Object
    ▶ 2: Object
  updated_at : 2025-12-27T15:27:45.404+00:00
```

Phân tích và thiết kế hệ thống

Thiết kế CSDL

Collection orders - Đơn mượn sách

```
_id: ObjectId('694fd90448b6f0b5630da30f')
user_id : ObjectId('694fce9831f0f2802b0b8ab6')
username : "ducpm"
total_per_day : 13000
total_amount : 13000
total_quantity : 5
- items : Array (5)
  ▶ 0: Object
  ▶ 1: Object
  ▶ 2: Object
  ▶ 3: Object
  □ 4: Object
    book_id : ObjectId('694faa05d4871065baed1dd6')
    bookCode : "00002"
    bookName : "Cơ học cổ điển và hiện đại"
    pricePerDay : 2000
    quantity : 1
    rent_days : 1
    subTotal : 2000
    status : "returned"
    branch_id : "HN"
    synced : false
    created_at : 2025-12-27T13:03:00.699+00:00
    confirmed_at : 2025-12-27T13:10:21.905+00:00
    confirmed_by : "adminHN"
    returned_at : 2025-12-27T13:10:33.056+00:00
```

Phân tích và thiết kế hệ thống

Tối ưu truy vấn tìm kiếm bằng index + text search NoSQL

- Thay vì scan toàn bộ dữ liệu, MongoDB truy vấn trực tiếp qua index
- Index dùng cho tìm kiếm chính xác, Text search dùng cho tìm kiếm gần đúng

Ưu điểm:

- Nhằm giảm thời gian truy vấn, giảm quét toàn bộ collection,
- Và tăng hiệu năng hệ thống khi dữ liệu lớn.

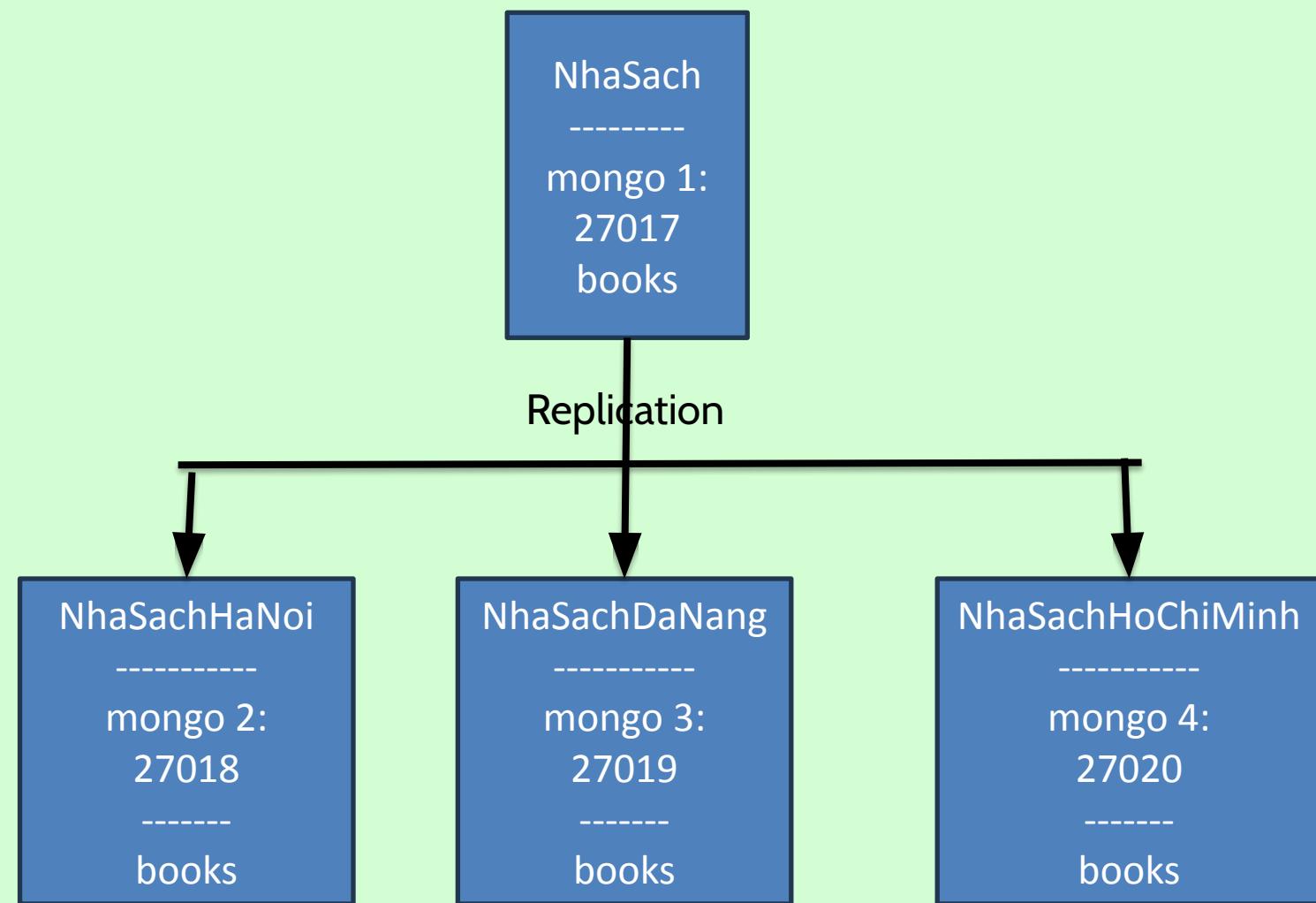
Name & Definition	Type	Size	Usage	Properties	Status	
> _id_	REGULAR ⓘ	24.6 kB	9 (since Sat Jan 17 2026)	UNIQUE ⓘ	READY	Đảm bảo mã sách duy nhất
> idx_bookCode_unique	REGULAR ⓘ	24.6 kB	0 (since Sat Jan 17 2026)	UNIQUE ⓘ	READY	Tìm sách theo khu vực
> idx_location_bookName_unique	REGULAR ⓘ	28.7 kB	0 (since Sat Jan 17 2026)	UNIQUE ⓘ COMPOUND ⓘ	READY	Lọc theo nhóm sách
> idx_bookGroup	REGULAR ⓘ	24.6 kB	0 (since Sat Jan 17 2026)		READY	Chỉ lấy sách đang hoạt động
> idx_location	REGULAR ⓘ	24.6 kB	4 (since Sat Jan 17 2026)		READY	Xếp hạng sách mượn nhiều
> idx_status	REGULAR ⓘ	24.6 kB	0 (since Sat Jan 17 2026)		READY	
> idx_borrowCount_desc	REGULAR ⓘ	24.6 kB	0 (since Sat Jan 17 2026)		READY	Tìm nhanh theo vị trí + tên sách
✓ idx_books_text_search	TEXT ⓘ	36.9 kB	0 (since Sat Jan 17 2026)	COMPOUND ⓘ	READY	Full-text search tên & mô tả

Phân tích và thiết kế hệ thống
Các truy vấn mẫu

1

Phân tích và thiết kế hệ thống phân tán

Master Data



Danh mục sách được đồng bộ xuống các cơ sở
giúp: hệ thống nhất quán được các loại sách

Tại NhaSach – center master data

```
Collection: books {  
    "bookCode": "00001",  
    "bookGroup": "Khoa học",  
    "bookName": "Vật lý lượng tử nhập môn",  
    "location": "Hà Nội",  
    "quantity": 50,  
    "pricePerDay": 2000,  
    "borrowCount": 0,  
    "status": "active",  
    "created_at": {  
        "$date": "2025-12-27T00:00:00.000Z"  
    },  
    "updated_at": {  
        "$date": "2025-12-27T15:14:09.491Z"  
    },  
    "synced": false  
}
```

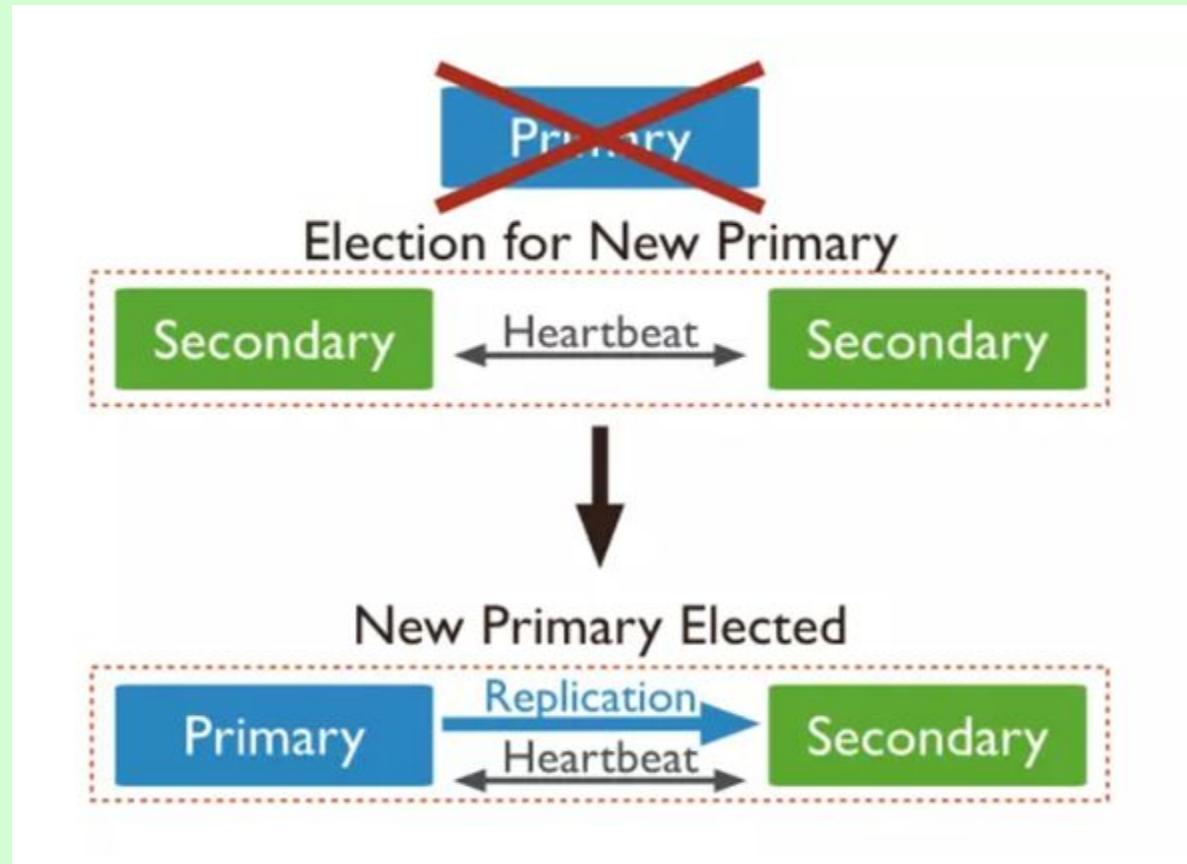
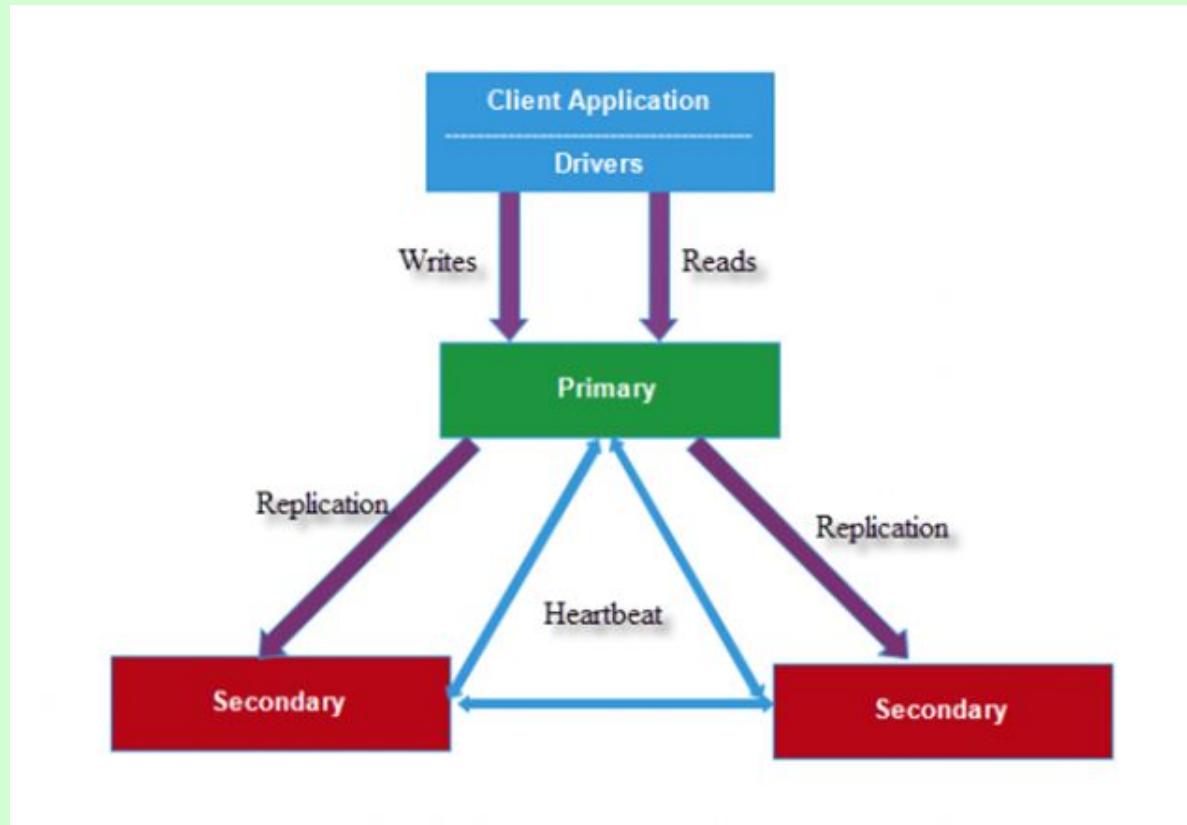
Tại NhaSachHaNoi, NhaSachHoChiMinh, NhaSachDaNang -
branch master

```
Collection: books {  
    "bookCode": "00002",  
    "bookGroup": "Khoa học",  
    "bookName": "Cơ học cổ điển và hiện đại",  
    "borrowCount": 0,  
    "created_at": {  
        "$date": "2025-12-27T09:42:29.879Z"  
    },  
    "location": "Hà Nội",  
    "pricePerDay": 2000,  
    "quantity": 48,  
    "status": "active",  
    "updated_at": {  
        "$date": "2025-12-27T15:14:09.491Z"  
    },  
    "synced": false  
}
```

MongoDB Architecture: Hybrid (1 Standalone + 3-node Replica Set)

- mongo1 (port 27017): Nhasach - STANDALONE
- mongo2 (port 27018): NhasachHaNoi - PRIMARY (rs0)
- mongo3 (port 27019): NhasachDaNang - SECONDARY (rs0)
- mongo4 (port 27020): NhasachHoChiMinh - SECONDARY (rs0)
- **# MongoDB Hybrid Setup for Distributed e-Library System**
- **# =====**
- **# Architecture:**
- **# - mongo1 (port 27017): Nhasach (Central Hub) - STANDALONE**
- **# - mongo2 (port 27018): NhasachHaNoi (Hanoi Branch) - PRIMARY in rs0**
- **# - mongo3 (port 27019): NhasachDaNang (Da Nang Branch) - SECONDARY in rs0**
- **# - mongo4 (port 27020): NhasachHoChiMinh (Ho Chi Minh Branch) - SECONDARY in rs0**
- **#**
- **# Replica Set (rs0) synchronizes ORDERS only across branches (HaNoi, DaNang, HCM)**
- **# Books and Users remain independent per branch**
-

Phân tích và thiết kế hệ thống phân tán Replica Set



(1) Mọi thao tác ghi dữ liệu (thêm, sửa, xóa, cập nhật) được thực hiện ở node primary

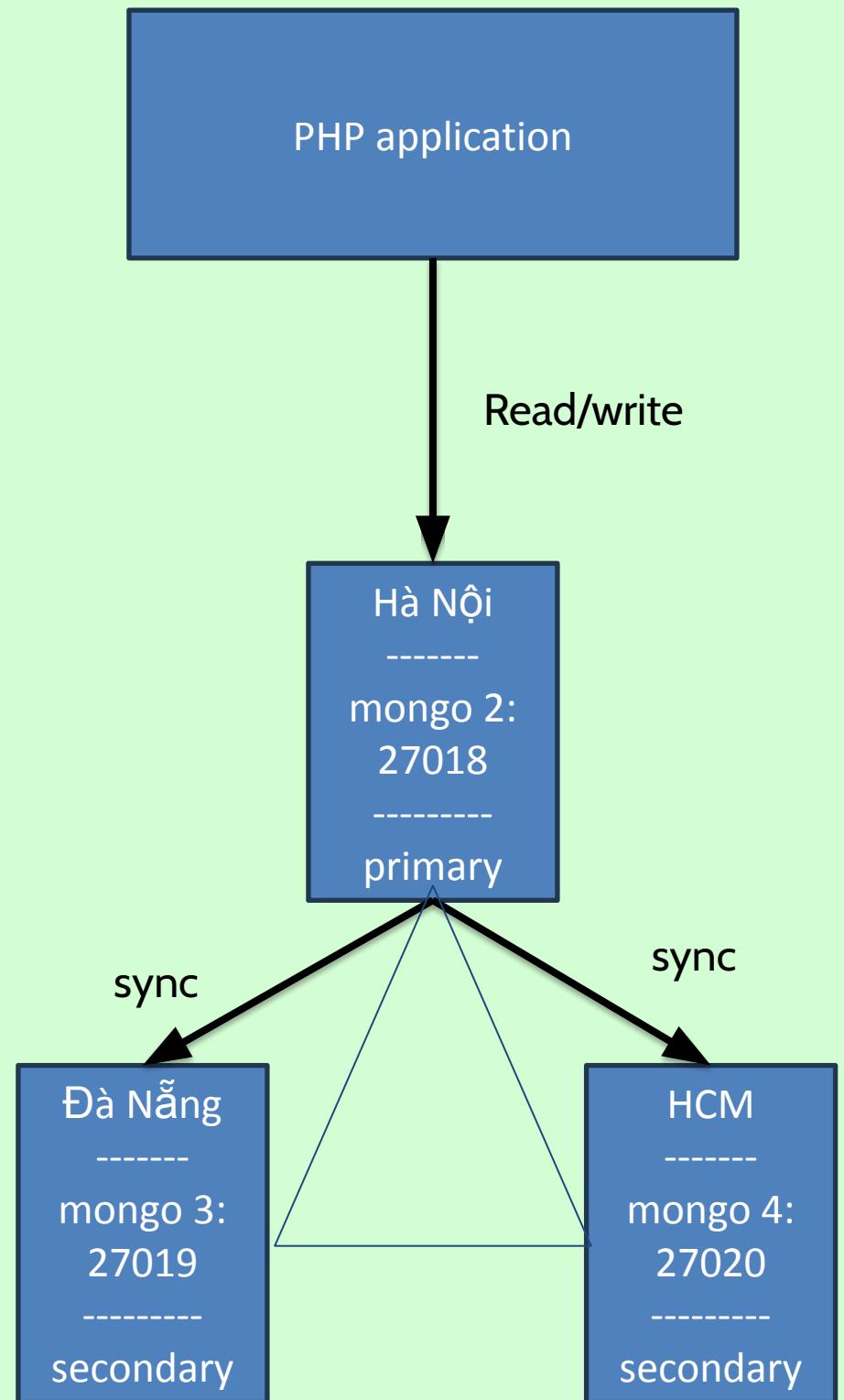
(2) Ghi thành công trên primary → mongoDB sẽ tự động sao chép và đồng bộ dữ liệu đó sang các node secondary

(3) Các node secondary chỉ phục vụ truy vấn đọc (tìm kiếm, xem): giúp giảm tải và tăng tốc độ truy vấn

(4) Trường hợp node primary gặp sự cố: 1 trong các node secondary sẽ được bầu làm primary

Phân tích và thiết kế hệ thống phân tán

Sơ đồ phân tán - Replica Set



03 Cài đặt và đánh giá

Cài đặt

Xây dựng giao diện , API kết nối giao diện với CSDL với PHP

Lưu trữ và phân tán: CSDL mongoDB, mongoDB compass; mongo driver,

Docker Compose cho MongoDB Replica Set

Đánh giá sản phẩm : giao diện và dữ liệu

Các kịch bản kiểm thử

Kịch bản 1: Kiểm thử đăng nhập, phân quyền, mã hóa mật khẩu, JWT

Các kịch bản kiểm thử

Kịch bản 2: Kiểm thử hiển thị dữ liệu

3.6.1 Kịch bản 1: Kiểm thử hiển thị dữ liệu

Mục đích: Đảm bảo dữ liệu hiển thị đúng tại mỗi chi nhánh.

Kết quả:

31

Bảng 3.4: Kết quả kiểm thử hiển thị dữ liệu

Chi nhánh	Port	Sách	Người dùng	Đơn mượn
Central Hub	8001	509	42	111
Hà Nội	8002	200	13	46
Dà Nẵng	8003	163	12	16
TP.HCM	8004	146	11	14
Tổng		1.018	78	187

Đánh giá: **PASS** - Dữ liệu hiển thị đúng theo từng database.

Các kịch bản kiểm thử

Kịch bản 3: Kiểm thử tìm kiếm full-text search

Các kịch bản kiểm thử

Kịch bản 4: Kiểm thử thêm, cập nhật đơn mượn sách

- Kịch bản 4.1:

1. Sinh viên đặt đơn yêu cầu mượn sách (paid)
2. Sinh viên hủy đơn mượn sách (cancelled)

- Kịch bản 4.2:

1. Sinh viên đặt đơn yêu cầu mượn sách (paid)
2. Nhân viên thư viện cập nhật đơn mượn sách (success)
3. Nhân viên thư viện cập nhật đơn trả sách (returned)

Các kịch bản kiểm thử

Kịch bản 5: Kiểm thử ghi và đồng bộ

3.6.2 Kịch bản 2: Kiểm thử ghi và đồng bộ

Mục đích: Đảm bảo dữ liệu đồng bộ từ PRIMARY sang SECONDARY.

Các bước:

1. Thêm sách mới tại Central Hub
2. Kiểm tra sách xuất hiện tại mongo2, mongo3
3. Đo replication lag

Kết quả:

- Ghi vào PRIMARY: Thành công
- Replication lag: 50-200ms
- Dữ liệu nhất quán: OK

Đánh giá: **PASS**

Các kịch bản kiểm thử

Kịch bản 6: Kiểm thử Failover

3.6.3 Kịch bản 3: Kiểm thử Failover

Mục đích: Đảm bảo hệ thống tự động phục hồi khi PRIMARY gặp sự cố.

Các bước:

```
1 # 1. Check current status
2 docker exec mongo1 mongosh --eval "rs.status().members.map(m => m.
   stateStr)"
3
4 # 2. Stop PRIMARY
5 docker stop mongo1
6
7 # 3. Wait for election (10-15s)
8 sleep 15
9
10 # 4. Check new PRIMARY
11 docker exec mongo2 mongosh --eval "rs.status().members.map(m =>
   stateStr)"
12
13 # 5. Restart old PRIMARY
14 docker start mongo1
```

Listing 3.8: Script kiểm thử Failover

Kết quả:

- Phát hiện node hỏng: ~10 giây
- Bầu chọn PRIMARY mới: ~5 giây
- Tổng thời gian gián đoạn: **10-15 giây**
- Hệ thống tiếp tục hoạt động: OK

Dánh giá: **PASS**

Các kịch bản kiểm thử

Kịch bản 7: Kiểm thử hiệu năng truy vấn

3.6.4 Kịch bản 4: Đo lường hiệu năng truy vấn (Benchmark)

Để đánh giá hiệu năng thực tế của hệ thống, nhóm đã xây dựng một bộ công cụ benchmark với 10 kịch bản truy vấn khác nhau, bao gồm cả các thao tác đọc và ghi. Mỗi kịch bản được thực hiện lặp lại 50 lần (iterations) trên dữ liệu thực của hệ thống để đảm bảo độ chính xác của kết quả đo.

Môi trường thử nghiệm:

- Phiên bản MongoDB: 8.0.16 (Community Edition)
- Dữ liệu thử nghiệm: 509 cuốn sách, 42 người dùng trong cơ sở dữ liệu Nhasach
- Số lần lặp: 50 iterations cho mỗi test case
- Chế độ kết nối: Standalone (localhost:27017)

Đánh giá

*Nhóm 10 xin cảm ơn
thầy và các bạn đã lắng nghe
phản báo cáo sản phẩm!*