|  |  |
| --- | --- |
| Logo_FPT_University_doc | **MINISTRY OF EDUCATION AND TRAINING**  **FPT UNIVERSITY** |

**DESIGN DOCUMENT**

**BMA1**

**Software Design Specification**

|  |  |
| --- | --- |
| **GR03** | |
| **Group Member** | HieuNXFX08030 – Nguyễn Xuân Hiếu |
| **Supervisor** | Xuân Hiếu |
| **Ext. Supervisor** |  |
| **Project Code** | BMA1 |

- Hanoi, 10/2020 -

**Table of Contents**

[1](#_gjdgxs) ARCHITECTURE DESIGN 3

[1.1](#_30j0zll) Choice of Architecture Design 3

[1.2](#_1fob9te) Architectural Presentation 3

[1.3](#_3znysh7) Component/Package Design 3

[2](#_2et92p0) DETAILED DESIGN 3

[2.1](#_tyjcwt) Common Design 3

[2.2](#_3dy6vkm) <UC xx- Use case name> 3

[2.2.1 Class Diagram 3](#_1t3h5sf)

[2.2.2 Class Description 3](#_4d34og8)

[2.2.3 Screen Design 5](#_2s8eyo1)

[2.2.4 Sequence Diagram 6](#_17dp8vu)

# ARCHITECTURE DESIGN

## Choice of Architecture Design

Các yếu tố:

* Kiểu ứng dụng của hệ thống phần mềm: ứng dụng quản lý người dùng và tài nguyên, ở đây là sách.
* Môi trường triển khai: môi trường Internet.
* Đối tượng người dùng: tất cả các đối tượng người dùng.
* Lĩnh vực kinh doanh: Thông tin, tức tức, giải trí.

Sau khi phân tích các yêu cầu từ phía khách hàng về phần mềm, thiết kế kiến trúc được lựa chọn là Layer và MVC

* Ưu điểm
  + Mô hình quen thuộc với nhiều lập trình viên, là một mô hình khá đơn giản và các lập trình viên đều có thể hiểu được.
  + Dễ dàng phát triển ứng dụng cũng như bảo trì. Các thành viên có thể làm các chức năng tách biệt nhau cùng 1 lúc.
  + Dễ dàng tìm kiếm các lỗi phát sinh, do ứng dụng có nhiều lớp khác nhau (Layer).
  + Dễ dàng thêm mới hay sửa đổi các chức năng, do ứng dụng chia làm nhiều module cũng như các lớp hoặc các service để thực hiện nhiều chức năng khác nhau, phân tách rõ ràng các phần như Model, giao diện, cơ sở dữ liệu, nghiệp vụ …
* Nhược điểm:
  + Mô hình MVC có thể hơi cồng kềnh đối với dự án vừa và nhỏ như dự án này.
  + Có những quy chuẩn nghiêm ngặt do sự quy định của mô hình MVC cũng như Framework được chọn là Ruby on Rails, mặc dù dễ tiếp cận tuy nhiên có thể gây khó khăn ban đầu trong việc tạo nền tảng cho dự án.

Để khắc phục các nhược điểm ở trên, nhóm có những giải pháp sau:

* Các thành viên sẽ luôn luôn review code của nhau sai khi các chức năng được hoàn thành (peer-review) để đảm bảo chất lượng code được tốt nhất.
* Những lập trình viên giàu kinh nghiệm với Framework sẽ đưa ra các tiêu chuẩn để các thành viên sau đó có thể tuân theo, các tiêu chuẩn này sẽ được tài liệu hóa để dễ dàng tham khảo và tra cứu.
* Đơn giản hóa cũng như tăng tính tái sử dụng của các service và các module để giảm sự cồng kềnh vốn có của mô hình MVC.

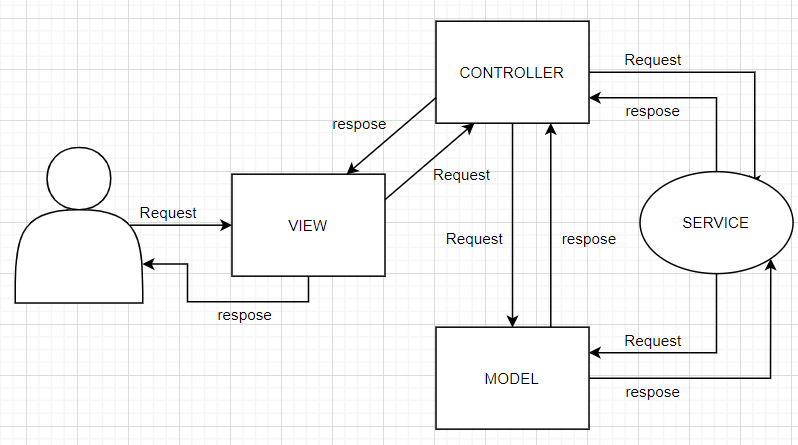
## Architectural Presentation

Mô hình MVC sẽ có các thành phần cơ bản sau:

* **M**odel: model là lớp làm việc trực tiếp với CSDL, tương tác trực tiếp với hệ cơ sở dữ liệu, như ứng dụng BMA này sẽ là PosgreSQL. Model sẽ không làm việc trực tiếp với View là giao diện người dùng mà sẽ nhận các request từ phía Controller, đôi khi là Service để thực hiện các lệnh truy vấn đến CSDL. Thành phần này sẽ bao gồm các class / function xử lý việc kết nối CSDL. Thêm, sửa. xóa … dữ liệu
* **V**iew: đây là thành phần giúp người dùng có thể tương tác được với ứng dụng, hay nói cách khác là giao diện người dùng, là nơi hiển thị các dữ liệu trả lại từ ứng dụng cũng như đón nhận các hành động hay dữ liệu của người dùng với với hệ thống.
* **C**ontroller: là nơi tiếp nhận những yêu cầu xử lý được gửi từ phía người dùng. Ở đây sẽ có những class / function xử lý những nhiệm vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ Model cung cấp. Đôi khi trong BMA Controller sẽ không trực tiếp gọi đến Model để lấy dữ liệu mà sẽ thông qua một lớp các Service nữa với những nghiệp vụ có logic phức tạp.

Các thành phần khác:

* Service: một thành phần không nhất thiết xuất hiện trong mọi sự tương tác giữa Controller và Model, thành phần này chứa các class / function xử lý các nghiệp vụ phức tạp nhưng đều để thực hiện mục đích hay chức năng nhất định có thể được sử dụng nhiều lần, mà việc đưa những logic hay chức năng này sẽ làm Controller trở nên nặng nề và khó bảo trì hay tìm lỗi.



Các thành phần trong mô hình MVC sẽ tương tác với nhau như mô hình trên. Người dùng sẽ thực hiện các yêu cầu (request) thông qua view, các request này sẽ được xử lý ở Controller, tùy từng loại request mà Controller sẽ xử lý và gửi các request này đến Service hoặc Model để xử lý, nếu như dữ liệu được gửi đến Service, Service sẽ lại tiếp tục xử lý các request này và gửi đến Model để nhận dữ liệu trả về cho Controller. Cuối cùng dữ liệu sẽ gửi Controller gửi về cho người dùng thông qua View, hiển thị cho người dùng.

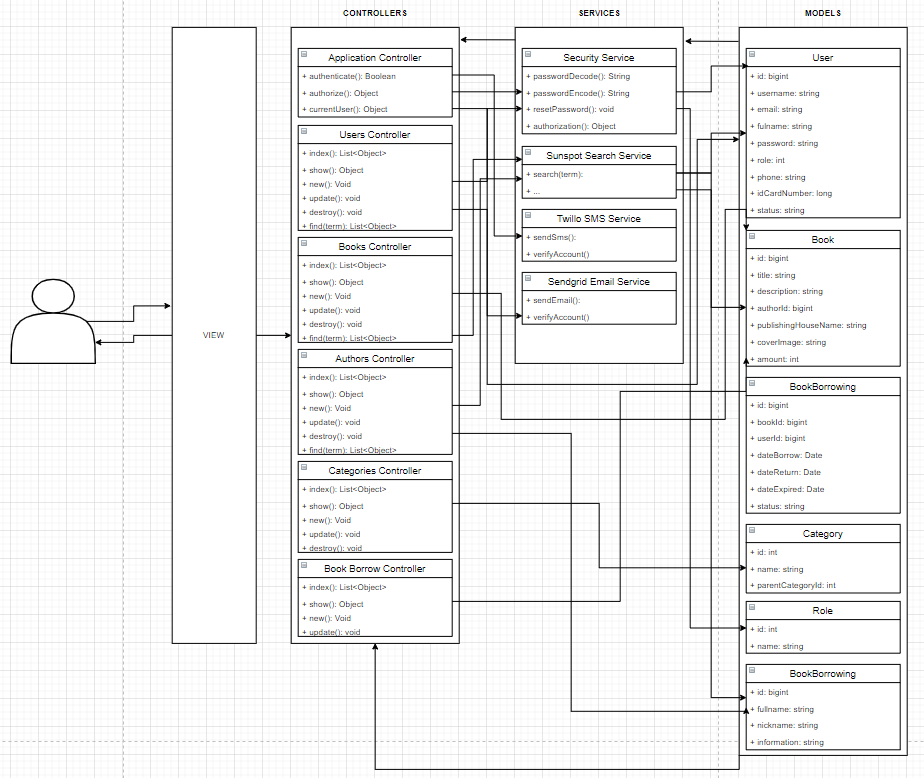
Các công nghệ sử dụng:

* Back-end: Ruby
* Front-end: HTML, CSS, JavaScript
* Library: jQuery
* Framework: Bootstrap 4, Material Design, Ruby on Rails.

Một số từ viết tắt trong tài liệu:

* BMA: Book Management Application
* CSDL: Cơ sở dữ liệu
* NDTT: Người dùng thông thường
* NQL: Người quản lý

## Component/Package Design



Khái quát kiến trúc của hệ thống dưới góc nhìn về components

# DETAILED DESIGN

## Common Design

Các chức năng của ứng dụng:

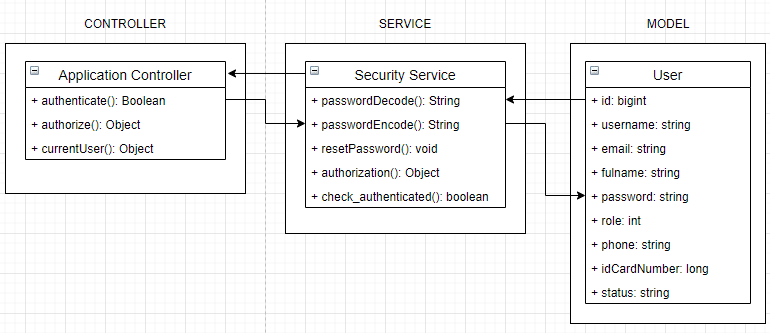
* Xác thực và ủy quyền người dùng
* Quản lý người dùng:
  + Tạo mới người dùng: đăng ký người dùng mới
  + Sửa thông tin người dùng: thay đổi thông tin cá nhân, thay đổi tình trạng người dùng (khóa người dùng)
  + Xóa người dùng khỏi hệ thống.
  + Xem danh sách người dùng, xem chi tiết về từng người dùng
* Quản lý sách:
  + Thêm mới đầu sách vào hệ thống
  + Thay thôi thông tin về sách trong hệ thống: tăng giảm số lượng sách, các thông tin về loại sách, thay đổi thông tin về thể loại
  + Xem danh sách các đầu sách về chi tiết về từng đầu sách trong hệ thống
  + Xóa đầu sách ra khỏi hệ thống
* Quản lý tác giả:
  + Thêm mới tác giả
  + Chỉnh sửa thông tin về các tác giả
  + Thêm các đầu sách vào tác giả tương ứng
  + Xóa tác giả khỏi hệ thống
* Quản lý thể loại:
  + Thêm mới thể loại sách
  + Chỉnh sửa tên thể loại sách
* Quản lý các đơn mượn sách:
  + Tạo mới đơn mượn sách: NDTT có thể tạo mới một đơn mượn sách và chờ phê duyệt.
  + Thay đổi thông tin đơn mượn sách:
    - Phê duyệt đơn mượn sách (đã cho mượn sách)
    - Hủy đơn mượn sách (từ phía NQL hoặc NDTT)
    - Thay đổi trạng thái đơn mượn sách (đã trả lại sách)
  + Xem danh sách các đơn mượn sách từ phía NDTT và NQL

Các thành phần chung:

|  |  |
| --- | --- |
| Tên thành phần | Mô tả thiết kế |
| User | Chứa các thông tin về người dùng của hệ thống, có thể vừa là NDTT (người mượn sách) hoặc NQL (người quản lý) |
| Role | Vai trò của người dùng, có hai loại là Admin hoặc User |
| Book | Chứa thông tin về các sách đang có trong hệ thống |
| Author | Chứa thông tin về các tác giả |
| Category | Thông tin về các thể loại sách |
| bookborrowing | Chứa thông tin về các đơn mượn sách của NDTT |

## BMA001 – Đăng nhập

### Class Diagram



Sơ đồ quan hệ giữa các lớp của chức năng

### Class Description

#### Application Controller Class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Application Controller** | | | |
| **Description** | Lớp này là một Base Class của cả ứng dụng. Nhưng controller khác của ứng dụng sẽ kế thừa class này và kế thừa cả những hàm của class này. Đây là class mà tất cả các request của ứng dụng sẽ đều đi qua đầu tiên. | | | |
| **Base Class** | ActionController::Base | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | Application\_controller.rb | | | |
| **Namespace** | app.controller | | | |
| **Attributes** | **Name** | **Type** | **Description** | |
| @current\_user | Object | Đây là object chứa thông tin về người dùng hiện tại đã đăng nhập về hệ thống, sẽ trả về nill (nil trong Ruby) nếu như người dùng chưa đăng nhập | |
| @logged\_in | Boolean | Trả về true nếu đang có người dùng đăng nhập, false nếu ngược lại. | |
| @role | Object | Vai trò của người dùng hiện tại, có thể là Admin (tương ứng với NQL) hoặc User (tương ứng với NDTT) | |
| **Methods** | **Name** | **Input** | **Output** | **Description** |
| authenticate | Username / Email: String  Password: String | boolean | Hàm này sẽ nhận vào giá trị là username hoặc email cùng với password, gán biến @current\_user ở trên với Object tương ứng với User đang đăng nhập vào hệ thống nếu password và email / username chính xác. Nếu @current\_user tồn tại sẽ trả về true, ngược lại trả về false |
| return\_role | N/A | void | Trả về object tương ứng với vai trò (Role) của người dùng đang đăng nhập và gán vào biến @role |

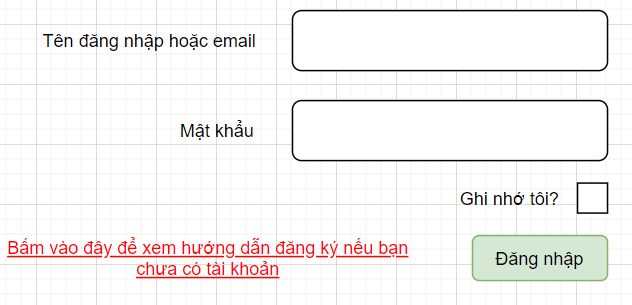
#### Security Service Class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Security Service Class** | | | |
| **Description** | Đây là service chứa những hàm liên quan đến xác thực và phân quyền người dùng | | | |
| **Base Class** | SecurityService::Base | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | security\_service.rb | | | |
| **Namespace** | app.services | | | |
| **Attributes** | **Name** | **Type** | **Description** | |
| @username | String | Có thể là username hoặc email người dùng nhập vào để đăng nhập | |
| @password | String | Mật khẩu người dùng sử dụng để đăng nhập | |
| @passwordEncode | String | Mật khẩu sau khi đã được mã hóa | |
| @role | Object | Vai trò của người dùng hiện tại, có thể là Admin (tương ứng với NQL) hoặc User (tương ứng với NDTT) | |
| **Methods** | **Name** | **Input** | **Output** | **Description** |
| passwordEncode | String | void | Hàm này có chức năng mã hóa mật khẩu thông thường thành dạng mã hóa |
| check\_authenticated | passwordEncode: String  username / email: String | boolean | Hàm này nhận vào usẻname hoặc email của người dùng cùng với mật khẩu đã mã hóa. Trả về giá trị boolean là true nếu người dùng có tồn tại trong hệ thống, false nếu sai mật khẩu hoặc username / email |

#### User Model Class

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **User** | | |
| **Description** | Lớp này là Model của đối tượng người dùng | | |
| **Base Class** | ApplicationRecord | | |
| **Constructor** | N/A | | |
| **Prototype** | N/A | | |
| **Source File** | user.rb | | |
| **Namespace** | app.model | | |
| **Attributes** | **Name** | **Type** | **Description** |
| id | bigint | ID của người dùng |
| username | Text | Username của người dùng |
| email | Text | Email của người dùng đăng ký trong hệ thống |
| fullname | Text | Tên đầy đủ của người dùng |
| password | Text | Mật khâỉ dạng mã hóa |
| role | Number | Khóa ngoại của bảng vai trò người dùng |
| phone | Text | Số điện thoại của người dùng |
| id\_card\_number | Text | Số chứng minh thư của người dùng |
| status | Text | Trạng thái của người dùng, có thể là:   * + - active: hoạt động     - inactive: người dùng đã bị khóa tài khoản |

### Screen Design



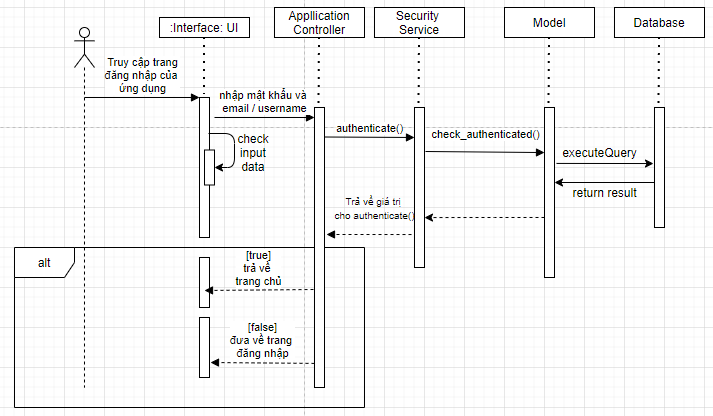
Hình 2.2.3-1: Screen Design của màn hình đăng nhập của ứng dụng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Mandatory | Max Length | Description |
| 1 | Email | Text | Yes | 40 | Email của người dùng muốn dùng để đăng ký |
| 2 | Xác nhận mật khẩu | Text | Yes | 20 | Hiển thị “\*” thay thế cho ký tự |
| 3 | Số chứng minh thư | Numbers | Yes | 10 | Số chứng minh thư hoặc số hộ chiếu của thành viên muốn đăng ký tài khoản |
| 4 | Hủy | Button |  |  | Bấm button này để dừng thao tác tạo mới thành viên và quay trở về trang chủ |
| 5 | Tạo mới thành viên | Button |  |  | Xác nhận việc tạo mới thành viên |

Bảng 2.2.3-1: định nghĩa màn hình đăng nhập của ứng dụng

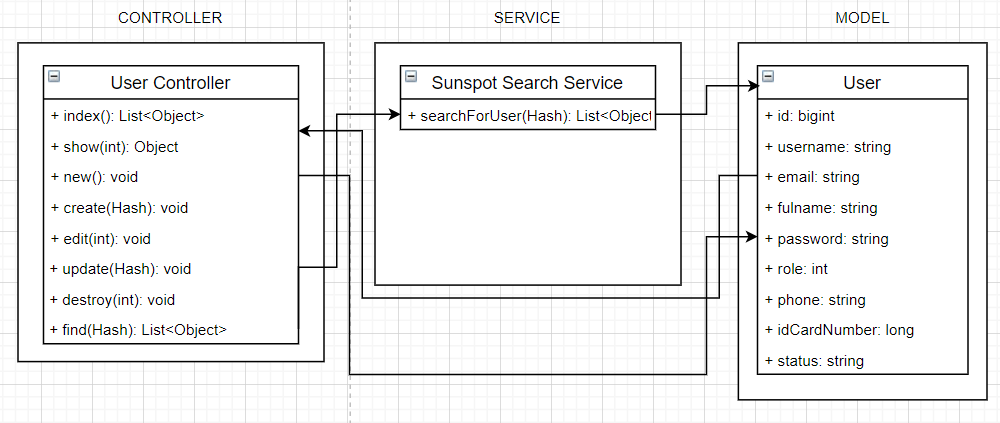
### Sequence Diagram

#### BMA001 – Use case đăng nhập



## BMA002 – Tạo mới thành viên (NDTT)

### Class Diagram



Sơ đồ quan hệ giữa các lớp chức năng

### Class Description

#### Users Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Users Controller** | | | |
| **Description** | Lớp này là Controller xử lý các request và trả về các response liên quan đến đối tượng là người dùng | | | |
| **Base Class** | Application Controller | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | users\_controller.rb | | | |
| **Namespace** | app.controller | | | |
| **Methods** | **Name** | **Input** | **Output** | **Description** |
| index | N/A | string | Trả về trang danh sách các users |
| show | Int | string | Trả về trang thông tin của detail user tương ứng |
| new | N/A | void | Trả về trang thêm mới người dùng |
| create | Hash (một dạng Object của Ruby) | void | Thêm mới người dùng vào CSDL |
| edit | int | string | Trả về một địa chỉ URL để người dùng có thể update thông tin của người dùng có ID tương ứng |
| update | Hash | void | Cập nhật thông tin một người dùng trong CSDL |
| destroy | int | void | Xóa người dùng có ID tương ứng trong CSDL |
| find | Hash | List<Object> | Hàm này sẽ nhận vào một Hash, dạng key - value tương ứng với trường và giá trị dùng để tìm kiếm một User, trả về List<Object> sau khi tìm kiếm |

#### 

### Screen Design

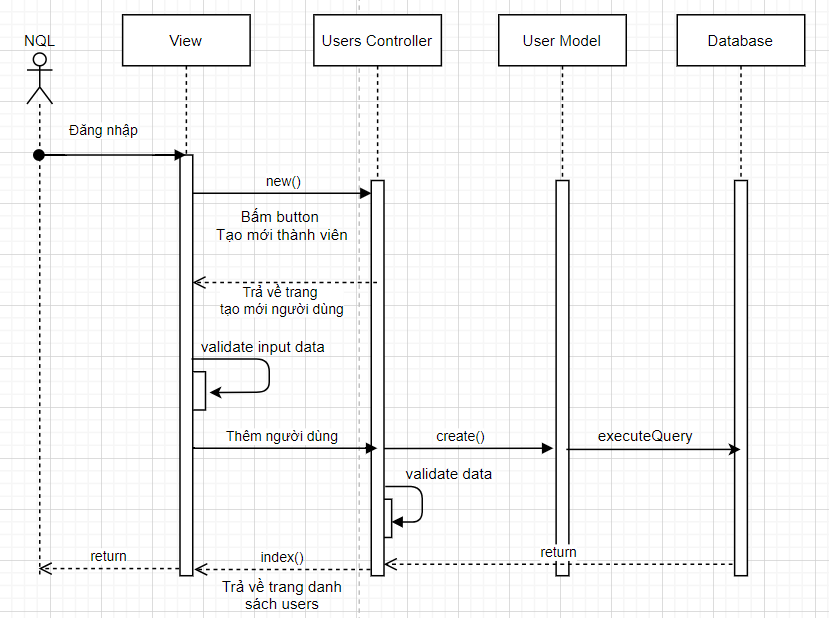


Hình 2.3.3-1: Screen Design của màn hình tạo mới thành viên

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Mandatory | Max Length | Description |
| 1 | Email | Text | Yes | 40 | Email của người dùng muốn dùng để đăng ký |
| 2 | Xác nhận mật khẩu | Text | Yes | 20 | Hiển thị “\*” thay thế cho ký tự |
| 3 | Số chứng minh thư | Numbers | Yes | 10 | Số chứng minh thư hoặc số hộ chiếu của thành viên muốn đăng ký tài khoản |
| 4 | Hủy | Button |  |  | Bấm button này để dừng thao tác tạo mới thành viên và quay trở về trang chủ |
| 5 | Tạo mới thành viên | Button |  |  | Xác nhận việc tạo mới thành viên |

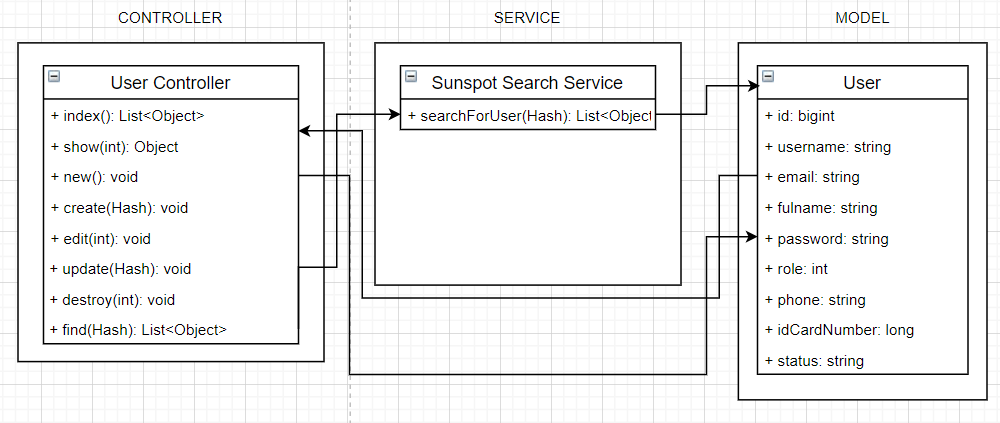
Bảng 2.3.3-1: Screen Definition của màn hình tạo mới thành viên

### Sequence Diagram



## BMA003 - Hiển thị danh sách các NDTT từ phía NQL

### Class Diagram



### Class Description

* Tương tự với Class Description của BMA002

### Screen Design

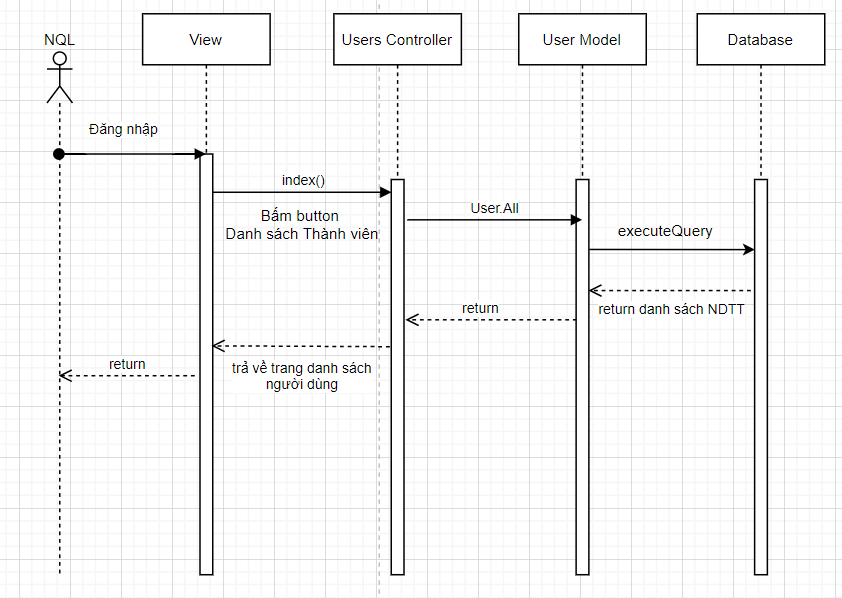


Hình 2.4.2-1: Screen Design của màn hình hiển thị danh sách các NDTT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Mandatory | Max Length | Description |
| 1 | Tìm kiếm người dùng | Text Field | No |  | Có thể là Username hoặc email hoặc Số chứng minh thư để tìm kiếm NDTT có trong hệ thống. |
| 2 | Tìm kiếm | Button |  |  | Submit để tìm kiếm người dùng |
| 3 | Tạo mới NDTT | Button |  |  | Đưa NQL đến trang để tạo mới một người dùng ở BMA002 |
| 4 | Bảng thông tin người dùng | Table |  |  | Bảng chứa các thông tin người dùng, mỗi cột là thông tin của người dùng tương ứng với username của hàng đó. |
| 5 | Username link của mỗi hàng | Link |  |  | Đường link dẫn đến trang thông tin chi tiết của người dùng có username tương ứng |
| 6 | Cho mượn sách | Button |  |  | Đưa NQL đến trang để cho người dùng tương ứng mượn sách |
| 7 | Nhận lại sách | Button |  |  | Đưa NQL đến trang để thu hồi lại sách. Nút này sẽ bị làm mờ (disabled) nếu như người dùng tương ứng đang không mượn sách nào của hệ thống |
| 8 | Vô hiệu hóa người dùng | Button |  |  | Đưa NQL đến trang để vô hiệu hóa hoặc xóa người dùng |
| 9 | Các button phân trang | Button |  |  | Chuyển trang hiển thị danh sách các thành viên. |

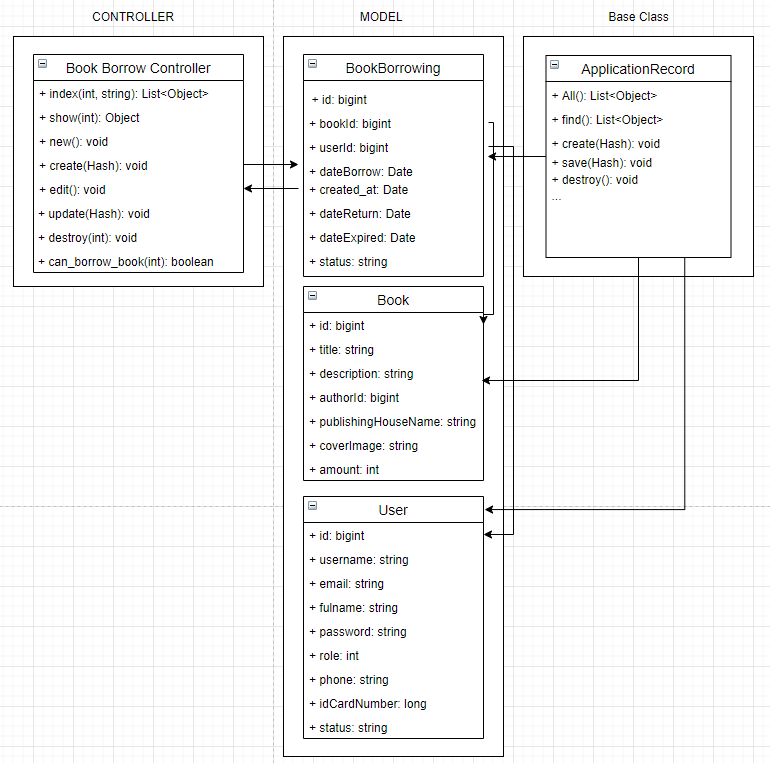
Bảng 2.4.3-1: Screen Definition của màn hình hiển thị danh sách các NDTT

### Sequence Diagram



## BMA004 - Hiển thị danh sách các sách đang mượn, đã mượn và đã trả của NDTT

### Class Diagram



### Class Description

#### Book Borrow Controller Class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Book Borrow Controller** | | | |
| **Description** | Lớp này là Controller xử lý các request cũng như trả về các response liên quan đến xử lý các đơn đăng ký mượn sách của người dùng. Có thể là tạo mới đơn đăng ký, cập nhật trạng thái đơn mượn sách (hủy, xác nhận cho mượn, xác nhận trả sách ...), kiểm tra xem người dùng có đủ điều kiện mượn sách không … | | | |
| **Base Class** | ActionController::Base | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | Application\_controller.rb | | | |
| **Namespace** | app.controller | | | |
| **Methods** | **Name** | **Input** | **Output** | **Description** |
| index | User ID: integer  Status: String | List<Object> | Hàm này sẽ nhận vào giá trị là ID của user cùng với status của sách muốn xem, các giá trị có thể là:  - All: xem tất cả các đơn mượn sách mà người dùng đã mượn.  - Cancel: Các đơn trả sách đã hủy  - Returned: Các đơn trả sách mà sách đã được trả lại bởi người dùng  - Borrowed: Các đơn mà sách đang được mượn  - Expired: các đơn mượn sách mà người dùng đang mượn và chưa trả lại sách cho thư viện |
| show | int | Object | Trả về chi đơn mượn sách với ID tương ứng với tham số truyền vào |
| new | N/A | void | Trả về trang tạo mới một đơn mượn sách |
| create | Hash | void | Tạo mới một đơn mượn sách, với các giá trị được lưu vào CSDL tương ứng với tham số truyền vào. |
| edit | bigint | void | Trả về trang sửa thông tin của một đơn mượn sách, nhận vào tham số là ID của đơn mượn sách cần cập nhật thông tin hoặc trạng thái |
| update | Hash | void | Cập nhật trạng thái của một đơn mượn sách và các thông tin khác |
| destroy | bigint | void | Xóa một đơn mượn sách |
| can\_borrow\_book | bigint | boolean | Kiểm tra một user có thể mượn sách của thư viện hay không, trả về giá trị True nếu có thể là False nếu ngược lại. |

#### 

#### ApplicationRecord Class

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **ApplicationRecord** | | |
| **Description** | Active Record là **M** trong mô hình MVC của frameword Ruby on Rails. Đây là layer của ứng dụng chịu trách nhiệm việc thể hiện mối quan hệ của các Object với các bảng trong CSDL, hay còn gọi là Object Reation Mapping (ORM). Có thể | | |
| **Base Class** | N/A | | |
| **Constructor** | N/A | | |
| **Prototype** | N/A | | |
| **Source File** | N/A | | |
| **Namespace** | N/A | | |

#### BookBorrowing Model Class

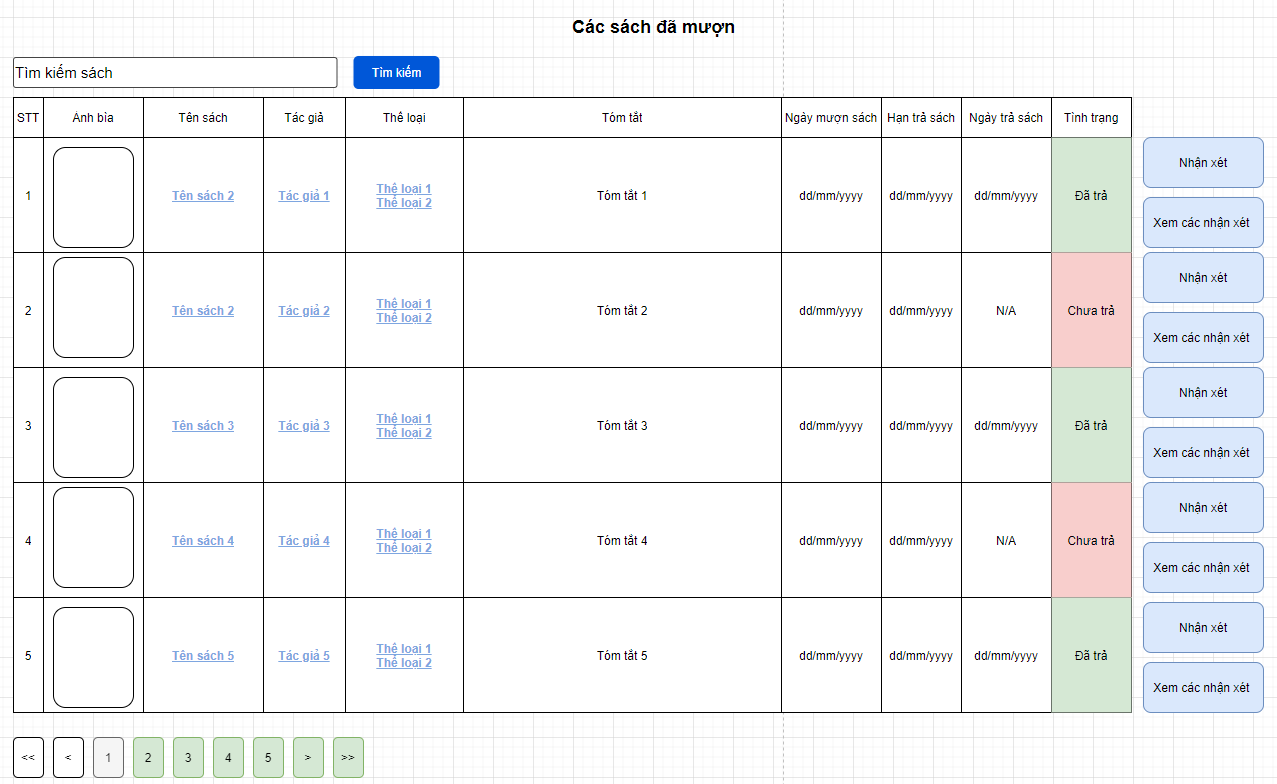
|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **BookBorrowing Model** | | |
| **Description** | Lớp này chịu trách nhiệm đại diện cho các Object các đơn mượn sách của người dùng. Lớp này có mối quan hệ Một - Nhiều với hai Model khác là Book và User. | | |
| **Base Class** | ApplicationRecord | | |
| **Constructor** | N/A | | |
| **Prototype** | N/A | | |
| **Source File** | book\_borrowing.rb | | |
| **Namespace** | app.model | | |
| **Attributes** | **Name** | **Type** | **Description** |
| id | bigint | ID của đơn mượn sách |
| bookId | bigint | ID của sách mà người dùng mượn |
| userId | bigint | ID của NDTT mượn sách |
| created\_at | Date | Ngày người dùng tạo đơn mượn sách |
| dateBorrow | Date | Ngày người dùng nhận được sách |
| dateReturn | Date | Ngày mười dùng trả lại sách |
| dateExpired | Date | Ngày tới hạn trả sách của đơn |
| status | Text | Trạng thái của đơn mượn sách. |

#### Book Model Class

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Book Model** | | |
| **Description** | Lớp này chịu trách nhiệm đại diện cho các Object tương ứng với các đầu sách đang có trong thư viện | | |
| **Base Class** | ApplicationRecord | | |
| **Constructor** | N/A | | |
| **Prototype** | N/A | | |
| **Source File** | book.rb | | |
| **Namespace** | app.model | | |
| **Attributes** | **Name** | **Type** | **Description** |
| id | bigint | ID của sách |
| title | Text | Tên của đầu sách |
| description | Text | Mô tả về tựa sách, các tóm tắt về sách để người đọc tham khảo. |
| authorId | bigint | ID của tác giả đầu sách |
| publishingHouseName | String | Tên nhà xuất bản |
| coverImage | String | Đường link của ảnh bìa sách trong Cloudinary |
| amount | int | Số lượng các sách này trong thư viện |

#### 

### Screen Design

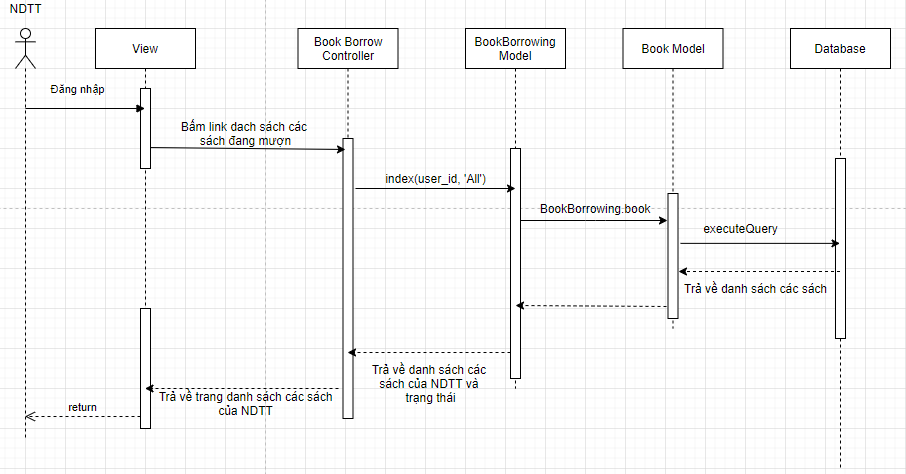


Hình 2.5.3-1: Screen Design của màn hiển thị các sách đã mượn, đã trả, đang mượn của NDTT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Mandatory | Max Length | Description |
| 1 | Tìm kiếm sách | Text Field | No |  | Tìm kiếm theo tên hoặc tác giả của sách. |
| 2 | Tìm kiếm | Button |  |  | Submit để tìm kiếm sách tương ứng với từ khóa |
| 3 | Cột Ảnh bìa | Ảnh |  |  | Ảnh bìa của mỗi sách |
| 4 | Cột Tên sách | Link |  |  | Tên của sách đã mượn, được hiển thị dưới dạng đường link, bấm vào link sẽ dẫn NDTT đến trang chi tiết về sách đó |
| 5 | Cột Tác giả | Link |  |  | Tên tác giả được hiển thị dưới dạng Link, bấm vào sẽ ra trang tiểu sử của tác giả cùng với các sách của tác giả đó |
| 6 | Cột Thể loại | Link |  |  | Tên của thể loại được hiển thị dưới dạng link, bấm vào sẽ đưa người dùng đến với trang chi tiết về các sách của thư viện có cùng thể loại đó |
| 7 | Cột Tóm tắt | Text |  |  | Tóm tắt về ấn phẩm |
| 8 | Cột Ngày mượn sách | Ngày tháng |  |  | Ngày mượn sách của người dùng |
| 9 | Cột Hạn trả sách | Ngày tháng |  |  | Hạn trả sách của người dùng |
| 10 | Cột Ngày trả sách | Ngày tháng / Text |  |  | Ngày người dùng trả lại sách, nếu người dùng chưa trả sách sẽ hiển thị **N/A** |
| 11 | Cột Tình trạng | Text |  |  | Tình trạng trả / mượn sách |
| 12 | Nhận xét | Button |  |  | Button nếu người dùng bấm vào sẽ đưa người dùng đến trang để nhận xét về tác phẩm |
| 13 | Xem các nhận xét | Button |  |  | Button nếu người dùng bấm vào sẽ đưa người dùng đến trang xem các nhận xét của các NDTT khác về sách |

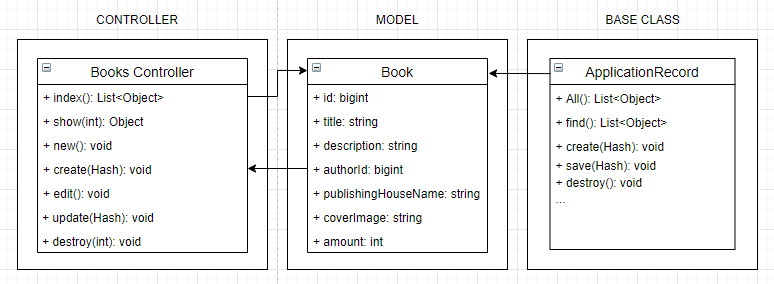
Bảng 2.5.3-2: Screen Definition của màn hình hiển thị các sách người dùng đã mượn, đã trả hoặc đang mượn

### Sequence Diagram



## BMA005 - Hiển thị danh sách các sách có trong thư viện

### Class Diagram



### Class Description

#### Books Controller Class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Books Controller** | | | |
| **Description** | Lớp này là Controller xử lý các request cũng như trả về các response liên quan đến các đầu sách hoặc các thông tin chi tiết về sách có trong hệ thống. | | | |
| **Base Class** | ActionController::Base | | | |
| **Constructor** | N/A | | | |
| **Prototype** | N/A | | | |
| **Source File** | books\_controller.rb | | | |
| **Namespace** | app.controller | | | |
| **Methods** | **Name** | **Input** | **Output** | **Description** |
| index | N/A | List<Object> | Hàm này sẽ trả về trang danh sách các sách có trong thư viện từ CSDL. Có thể sẽ nhận thêm các tham số như thể loại các trang hiển thị từ các tham số trên URL để hiển thị các sách tương ứng. |
| show | int | Object | Trả về ctrang thông tin chi tiết về một đầu sách có tham số là ID của đầu sách muốn lấy thông tin chi tiết |
| new | N/A | void | Trả về trang tạo mới một sách trong hệ thống |
| create | Hash | void | Tạo mới một đầu sách vào thư viện, với các giá trị được lưu vào CSDL tương ứng với tham số truyền vào. |
| edit | bigint | void | Trả về trang sửa thông tin của một đầu sách, nhận vào tham số là ID của đơn mượn sách cần cập nhật thông tin hoặc trạng thái |
| update | N/A | void | Cập nhật thông tin của một đầu sách |
| destroy | bigint | void | Xóa một đầu sách có trong thư viện |

#### 

### Screen Design

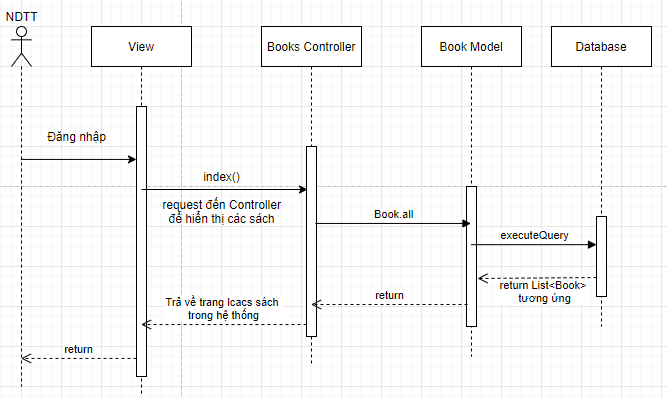


Hình 2.6.3-1: Screen Design của màn hình hiển thị các sách có trong hệ thống

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Mandatory | Max Length | Description |
| 1 | Tìm kiếm sách | Text Field | No |  | Tìm kiếm theo tên hoặc tác giả của sách. |
| 2 | Tìm kiếm | Button |  |  | Submit để tìm kiếm sách tương ứng với từ khóa |
| 3 | Cột Ảnh bìa | Ảnh |  |  | Ảnh bìa của mỗi sách |
| 4 | Cột Tên sách | Link |  |  | Tên của sách đã mượn, được hiển thị dưới dạng đường link, bấm vào link sẽ dẫn NDTT đến trang chi tiết về sách đó |
| 5 | Cột Tác giả | Link |  |  | Tên tác giả được hiển thị dưới dạng Link, bấm vào sẽ ra trang tiểu sử của tác giả cùng với các sách của tác giả đó |
| 6 | Cột Thể loại | Link |  |  | Tên của thể loại được hiển thị dưới dạng link, bấm vào sẽ đưa người dùng đến với trang chi tiết về các sách của thư viện có cùng thể loại đó |
| 7 | Cột Tóm tắt | Text |  |  | Tóm tắt về ấn phẩm |
| 8 | Cột Ngày xuất bản | Ngày tháng |  |  | Ngày sách được xuất bản |
| 9 | Cột Nhà xuất bản | Text |  |  | Tên nhà xuất bản |
| 10 | Cột Giá bìa | Numbers |  |  | Giá bìa của sách |
| 11 | Cột Tình trạng | Text |  |  | Tình trạng sách có sẵn để cho mượn hay không |
| 12 | Đăng ký mượn sách | Button |  |  | Button nếu người dùng bấm vào sẽ đưa người dùng đến trang để xác nhận mượn sách |
| 13 | Xem các đánh | Button |  |  | Button nếu người dùng bấm vào sẽ đưa người dùng đến trang xem các nhận xét của các NDTT khác về sách |
| 14 | Danh sách thể loại | Link |  |  | Các đường link bấm vào sẽ hiển thị các sách với thể loại tương ứng |
| 15 | Xem danh sách các sách đã mượn | Link |  |  | Bấm vào sẽ đưa người dùng đến trang hiển thị các sách mà người dùng đã mượn được dùng trong Use Case BMA004 |
| 16 | Các phím phân trang | Button |  |  | Bấm vào để chuyển đến trang hiển thị tương ứng |

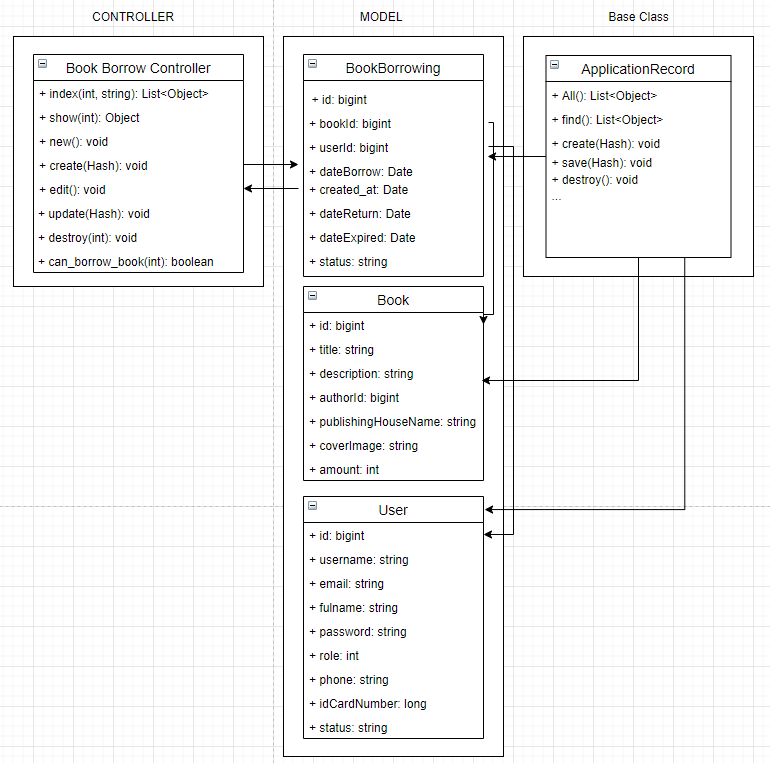
Bảng 2.6.3-2: Screen Definition của màn hình hiển thị các sách có trong hệ thống

### Sequence Diagram



## BMA006 - Đăng ký mượn sách

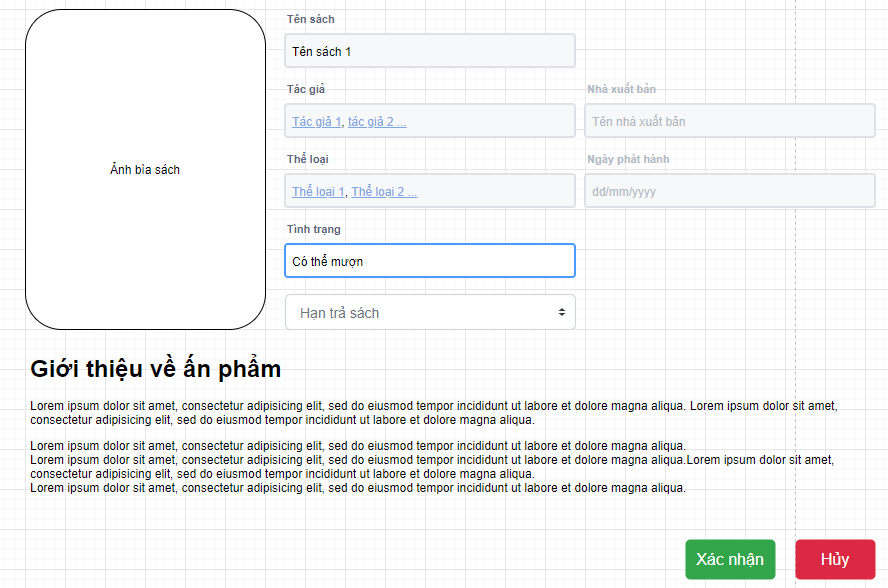
### Class Diagram



### Class Description

* Tương tự với Class Description của Use Case BMA004

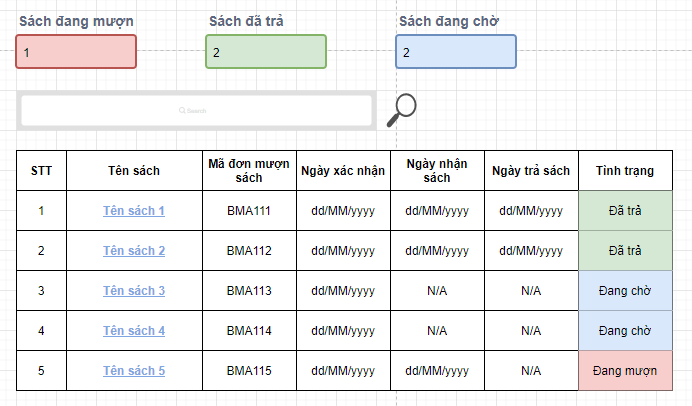
### Screen Design



Hình 2.7.3-1: Screen Design của màn hình hiển thị tạo đơn xác nhận đăng ký mượn sách

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Mandatory | Max Length | Description |
| 1 | Ảnh bìa sách | Image |  |  | Ảnh bìa của sách mà người dùng muốn mượn |
| 2 | Tên sách | Label |  |  |  |
| 3 | Tên sách Input | Text |  |  | Tên của sách muốn mượn |
| 4 | Tác giả | Label |  |  |  |
| 5 | Tên tác giả | Link |  |  | Tên tác giả được hiển thị dưới dạng Link, bấm vào sẽ ra trang tiểu sử của tác giả cùng với các sách của tác giả đó |
| 6 | Thể loại | Label |  |  |  |
| 7 | Tên thể loại | Link |  |  | Tên thể loại được hiển thị dưới dạng link, bấm vào sẽ vào trang danh sách các sách có trong thư viện tương ứng với thể loại vừa chọn |
| 8 | Tình trạng | Label |  |  |  |
| 9 | Tình trạng sách | Text |  |  | Tình trạng sách có sẵn để cho mượn hay không, nếu có thì hiển thị **Có thể mượn**, nếu không sẽ hiển thị **Ấn phẩm không có sẵn.** |
| 10 | Nhà xuất bản | Label |  |  |  |
| 11 | Tên nhà xuất bản | Text |  |  | Tên của nhà xuất bản |
| 12 | Ngày phát hành | Label |  |  |  |
| 13 | dd/MM/yyyy | Ngày tháng |  |  | Ngày sách được phát hành |
| 14 | Hạn trả sách | Selection | Yes |  | Nơi người dùng chọn thời gian mượn sách. Các giá trị có sẵn là  - 1 tuần  - 2 tuần  - 1 tháng  - 2 tháng |
| 15 | Giới thiệu ấn phẩm | Header |  |  |  |
| 16 | Nội dung của giới thiệu ấn phẩm | Paragraph |  |  | Hiển thị tóm tắt nội dung của ấn phẩm |
| 17 | Xác nhận | Button |  |  | Xác nhận việc mượn sách |
| 18 | Hủy | Button |  |  | Hủy bỏ việc mượn sách |

Bảng 2.7.3-2: Screen Definition của màn hình hiển thị tạo đơn xác nhận đăng ký mượn sách

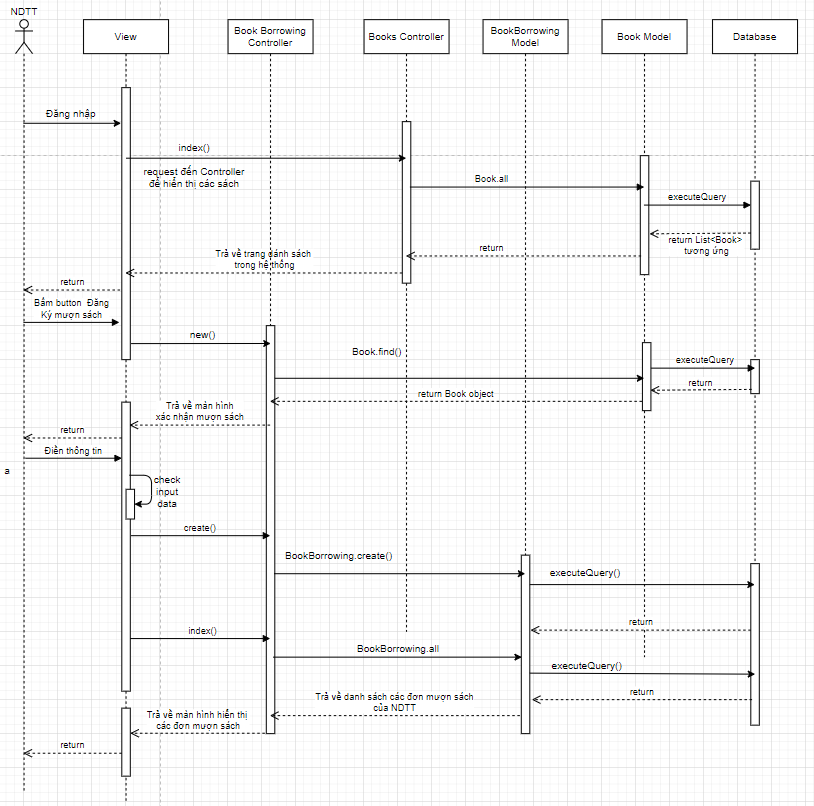


Hình 2.7.3-3: Screen Design của màn hình hiển thị các đơn mượn sách của người dùng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Mandatory | Max Length | Description |
| 1 | Sách đang mượn | Label |  |  |  |
| 2 | Sách đang mượn | Number |  |  | Cho biết số lượng sách mà NDTT đang mượn, chưa trả lại cho thư viện |
| 3 | Sách đã trả | Label |  |  |  |
| 4 | Sách đã trả textbox | Number |  |  | Cho biết số sách mà NDTT đã mượn và đã trả |
| 5 | Sách đang chờ | Label |  |  |  |
| 6 | Sách đang chờ textbox | Number |  |  | Cho biết số lượng sách mà NDTT đã xác nhận mượn tuy nhiên chưa nhận sách từ thư viện |
| 7 | Cột STT | Number |  |  | Số thứ tự |
| 8 | Cột Tên sách | Text |  |  | Tên các sách đang mượn, đã mượn và đã đăng ký mượn |
| 9 | Cột Mã đơn mượn sách | Text |  |  | Đơn mượn sách, mỗi một đơn sẽ có một mã khác nhau. |
| 10 | Cột Ngày xác nhận | Date |  |  | Ngày NDTT xác nhận việc mượn sách |
| 11 | Cột Ngày nhận sách | Date |  |  | Ngày NDTT nhận được sách từ thư viện. Nếu NDTT chưa nhận được sách thì cột sẽ có giá trị là N/A |
| 12 | Cột Ngày trả sách | Date |  |  | Ngày NDTT trả lại sách cho thư viện. Nếu người dùng chưa trả lại sách hoặc sách chưa đến tay người mượn giá trị sẽ là N/A |
| 13 | Cột Tình trạng | Text |  |  | Tình trạng của sách đang mượn |
| 14 | Search textbox | Input | No |  | Tìm sách đã mượn / đang mượn / sắp mượn theo tên hoặc mã đơn |

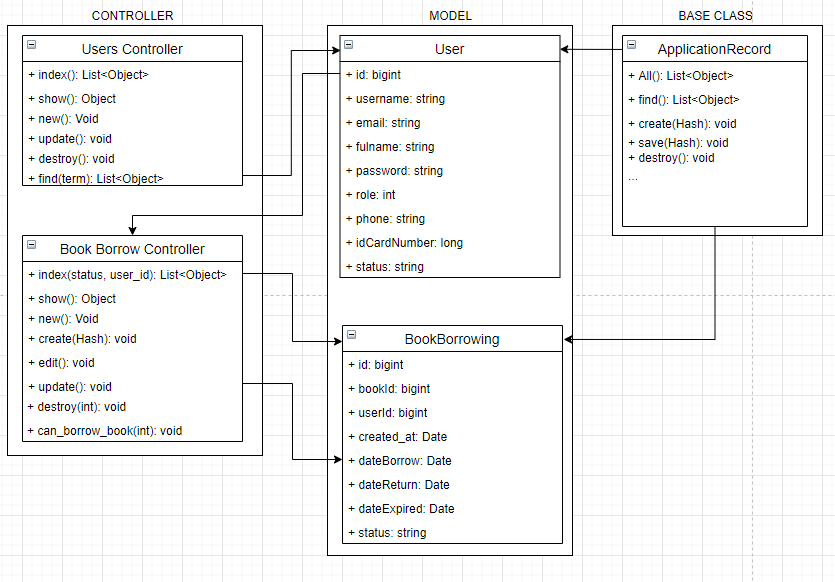
Bảng 2.7.3-4: Screen Definition của màn hình hiển thị các đơn đăng ký mượn sách

### Sequence Diagram



## BMA007 - Xác nhận cho người dùng mượn sách từ phía NQL

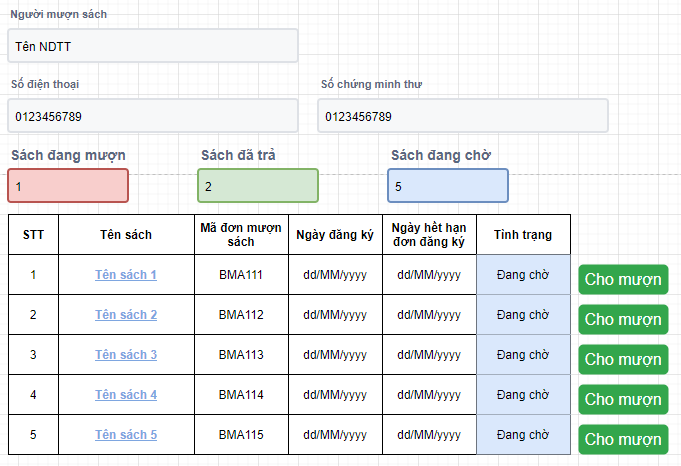
### Class Diagram



### Class Description

* Tương tự với Class Description của các Use Case trên

### Screen Design



Hình 2.8.3-1: Screen Design của màn hình cho người dùng mượn sách

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Mandatory | Max Length | Description |
| 1 | Người mượn sách | Label |  |  |  |
| 2 | Tên NDTT | Text |  |  | Tên của NDTT có nhu cầu mượn sách. Trường này không sửa được (disabled) |
| 3 | Số điện thoại | Label |  |  |  |
| 4 | Số điện thoại – Input Text | Numbers |  |  | Số điện thoại của người mượn sách. Trường này không sửa được (disabled) |
| 5 | Số chứng minh thư | Label |  |  |  |
| 6 | Số chứng minh thư – Text input | Numbers |  |  | Số chứng minh thư cua người mượn sách. Trường này không sửa được (disabled) |
| 7 | Sách đang mượn | Label |  |  |  |
| 8 | Sách đang mượn – Text Input | Number |  |  | Số lượn các sách mà người dùng đang mượn nhưng chưa trả lại cho thư viện. Nhiều nhất sẽ là 5 sách. Trường này không sửa được (disabled) |
| 8 | Sách đã trả | Label |  |  |  |
| 9 | Sách đã trả - Text Input | Number |  |  | Số lượng các sách mà người dùng đã trả lại cho thư viện. Trường này không sửa được (disabled) |
| 10 | Sách đang chờ | Label |  |  |  |
| 11 | Sách đang chờ - Text Input | Numbers |  |  | Số lượng các sách mà người dùng đã đăng ký mượn từ thư viện. Trường này không sửa được (disabled) |
| 12 | Cột STT | Number |  |  | Số thư tự của các đơn đăng ký mượn sách |
| 13 | Cột Tên sách | Text |  |  | Tên các sách mà người dùng có nhu cầu mượn |
| 14 | Cột Mã đơn mượn | Text |  |  | Mã đơn mượn sách |
| 15 | Cột Ngày đăng ký | Date |  |  | Ngày người dùng tạo đơn mượn sách |
| 16 | Cột Ngày hết hạn đăng ký | Date |  |  | Ngày đơn đăng ký mượn sách hết hạn, có giá trị là 1 tuần kể từ ngày đăng ký mượn sách |
| 17 | Cột Tình trạng | Text |  |  | Tình trạng của đơn đăng ký mượn sách, có thể là:  - Đang chờ: người dùng chưa mượn sách và vẫn còn nhu cầu mượn sách  - Hết hạn: người dùng chưa mượn sách tuy nhiên đã hết hạn mượn sách (hơn 1 tuần kể từ ngày đăng ký mượn sách)  - Hủy: người dùng đã hủy đơn mượn sách |
| 18 | Cho mượn | Button |  |  | Xác nhận việc cho mượn sách từ phía NQL. Button này sẽ không hiển thị với những đơn mượn sách bị hết hạn hoặc người dùng đã hủy đơn mượn sách này. |

Bảng 2.8.4-2: Screen Definition của màn hình cho mượn sách của NQL

### Sequence Diagram

