# Khung sườn

Khung sườn khóa luận kết thúc 6 năm luyện ngục của nvmnghia.

Độ dài yêu cầu: 40-50 trang.

## 0. Bìa & các mục liên quan

* Bìa
* Phụ bìa
* Cam đoan không sao chép
* Phê chuẩn của giảng viên hướng dẫn
* Lời cảm ơn
* Tóm tắt
* Nhờ Internet, truyền hình và điện ảnh, văn hóa truyện tranh đã trở nên phổ biến toàn cầu, đặc biệt là trong giới trẻ. Sự phát triển của truyện tranh trên trong thời đại số thúc đẩy sự phát triển của các trang web truyện tranh, thỏa mãn nhu cầu đọc sớm nhất có thể sau khi truyện được xuất bản. Tuy nhiên, nguồn truyện tranh của những trang web này có chất lượng không cao. Một bộ phận người đọc kĩ tính chọn đọc và lưu trữ những file truyện được số hóa chất lượng cao, thường ở dạng file nén đuôi cbr và cbz. Xuất phát từ yêu cầu này, tôi muốn viết ứng dụng yacv có thể đọc các file truyện nén trên điện thoại Android. Nội dung của khóa luận sẽ trình bày những đặc điểm cơ bản của kiến trúc ứng dụng Android - với cốt lõi là mẫu thiết kế MVVM - được khuyên dùng bởi Google, các tính năng trong yacv, và các ca sử dụng của phần mềm.
* Từ khóa: MVVM, Jetpack, Android, cbr, cbz, comic
* Mục lục
* Danh sách bảng
* Danh sách hình
* Danh sách kí hiệu, chữ viết tắt
  + MVVM

## 1. Chương 1: Giới thiệu

### 1.1. Đặt vấn đề

Tại Đông Á và Đông Nam Á, văn hóa truyện tranh gốc Á, nhất là truyện tranh Nhật (manga), được đón nhận khá tích cực, đặc biệt trong giới trẻ. Thế hệ những người dưới 40 tuổi hiện nay được tiếp xúc với truyện tranh từ sớm, thông qua những cuốn truyện truyền tay và phim hoạt hình dựa trên truyện tranh, và tiếp tục đọc dù đã qua tuổi thiếu niên. Một số tác phẩm manga còn có lượng người đọc lớn trên toàn cầu như Doraemon, One Piece. Ở bên kia bán cầu, với sự thành công của vũ trụ điện ảnh Marvel và DC, truyện tranh phương Tây (comic) cũng được hồi sinh phần nào sau một thập kỷ thiếu sáng tạo và suy giảm doanh số sách in. Các bộ truyện siêu anh hùng, vốn trước đây chỉ phổ biến ở Hoa Kỳ, nay đang trên đường trở thành một phần của văn hóa đại chúng như vị thế của manga. Có thể nói, văn hóa truyện tranh nói chung đang ở thời kì phát triển nhất, xét theo tiêu chí về độ phổ biến và thái độ đón nhận của xã hội.

Hiện nay, hầu hết mọi người đọc truyện qua các trang web tổng hợp truyện tranh. Những trang web này có hai ưu điểm chính:

* Số lượng: chỉ cần truy cập một trang là có thể đọc ít nhất hàng nghìn đầu truyện.
* Tốc độ: tốc độ ra truyện rất nhanh. Với các bộ truyện nổi tiếng, thường chỉ trong vòng một vài giờ sau khi chương mới ra mắt, bản dịch đã xuất hiện.

Tuy vậy, nhược điểm chính của những trang web này là chất lượng ảnh của truyện. Để giảm thời gian tải và tránh tốn băng thông, hình ảnh của truyện thường được nén khá nhiều, gây vỡ hình, mờ nhòe. Một bộ phận người đọc, hoặc kĩ tính, hoặc muốn sưu tầm truyện, thường chọn đọc những tệp truyện chất lượng cao, thường có đuôi .cbz hoặc .cbr. Bản chất tệp truyện này là các tệp nén zip bình thường, bên trong có các tệp ảnh thông dụng như .jpg, .png. Tuy nhiên, do được tải hẳn về máy rồi mới đọc, những tệp truyện này không bị giới hạn về băng thông hay thời gian, do đó hình ảnh trong tệp có thể có chất lượng rất cao.

Trong khóa luận này, tôi viết một ứng dụng Android nhằm phục vụ số ít người dùng có nhu cầu đọc truyện tranh chất lượng cao đã giới thiệu ở trên. Tên của ứng dụng là “yacv”, viết tắt của cụm từ tiếng Anh “Yet Another Comic Viewer”, tạm dịch là “Lại một ứng dụng xem truyện tranh nữa”. Hai tính năng chính duy nhất của ứng dụng là đọc và quản lí cơ bản (tìm kiếm, xóa) tệp truyện tranh có sẵn trên điện thoại.

Cần chú ý rằng ứng dụng yacv chỉ bao gồm các tính năng liên quan đến đọc truyện ngoại tuyến, đọc các tệp truyện có sẵn trên điện thoại người dùng. Ứng dụng không phải là ứng dụng khách cho các trang đọc truyện hiện có, hay có máy chủ tập trung riêng để cung cấp truyện.

### 1.2. Ứng dụng tương tự

Hiện có nhiều ứng dụng đọc truyện tranh ngoại tuyến như yacv trên chợ ứng dụng Google Play. Hai ứng dụng phổ biến nhất trong số này là [ComicScreen](https://play.google.com/store/apps/details?id=com.viewer.comicscreen&hl=en&gl=US) và [Astonishing Comic Reader](https://play.google.com/store/apps/details?id=com.aerilys.acr.android&hl=en&gl=US). ComicScreen là ứng dụng nổi tiếng hơn. Các tính năng của ComicScreen giống với các tính năng của yacv, tuy nhiên ComicScreen có thêm nhiều chức năng phụ, đáng kể nhất là khả năng đọc từ mạng FTP/SMB và khả năng sửa ảnh trong file. Astonishing Comic Reader cũng có chức năng tương tự yacv, không hơn, tuy nhiên giao diện khá trau chuốt. Cả hai đều miễn phí và có quảng cáo, được cập nhật có thể nói là thường xuyên.

Một ngoại lệ đáng kể ở đây là ứng dụng mã nguồn mở [Tachiyomi](https://github.com/tachiyomiorg/tachiyomi). Ứng dụng này có hệ thống phần mở rộng, cho phép đọc truyện ở các trang web truyện tranh. Khi web truyện tranh thay đổi, hoặc hỗ trợ thêm trang mới, chỉ cần tải về phần mở rộng tương ứng ở dạng ứng dụng .APK. Tính năng này cùng với mô hình mã nguồn mở khiến Tachiyomi mạnh hơn, và chau chuốt hơn, cập nhật nhanh hơn toàn bộ các ứng dụng đã có và sẽ có. Tuy nhiên, Tachiyomi lại không thể được đưa lên Play Store, vì chính tính năng phần mở rộng đã [vi phạm chính sách](https://github.com/tachiyomiorg/tachiyomi/issues/1745) của Play Store.

Một điểm khác biệt quan trọng của yacv với các ứng dụng có sẵn là việc hỗ trợ tìm kiếm metadata của tệp truyện tranh, do các ứng dụng có sẵn trên Play Store đa số bỏ qua thông tin này trong tệp truyện. Một trong số rất ít những ứng dụng hỗ trợ tính năng này là [Kuro Reader](https://play.google.com/store/apps/details?id=br.com.kurotoshiro.leitor_manga&hl=en&gl=US), tuy nhiên đây là một tính năng trả phí.

### 1.3. Kết quả đạt được

Ứng dụng có các tính năng đủ dùng theo mục đích đã đề ra:

* Đọc file truyện .cbz
* Tìm kiếm truyện theo metadata

Tính năng đọc tệp truyện .cbr hiện mới chỉ được cài đặt một phần, do khó khăn trong việc tích hợp thư viện đọc định dạng này.

### 1.4. Cấu trúc khóa luận

Các phần còn lại của khóa luận có cấu trúc như sau:

* Chương 2 - Kiến thức nền tảng: giới thiệu sơ lược về ba nền tảng của ứng dụng, gồm hệ điều hành Android, ngôn ngữ lập trình Kotlin, và mẫu thiết kế MVVM; định dạng tệp nén .zip cũng được giới thiệu vì liên quan trực tiếp đến ứng dụng.
* Chương 3 - Phân tích yêu cầu: phân tích nhu cầu và ca sử dụng để có đặc tả yêu cầu.
* Chương 4 - Lập trình & Kiểm thử: một số cài đặt và ca kiểm thử trong ứng dụng sẽ được nêu một cách có chọn lọc.
* Chương 5 - Kết luận: kết thúc khóa luận.

## 2. Chương 2: Kiến thức nền tảng

Chương này giới thiệu sơ qua về các nền tảng trong quá trình xây dựng ứng dụng.

* Hai nền tảng đầu tiên liên quan đến nhau, là tiền đề cho toàn bộ ứng dụng sẽ được giới thiệu trước, gồm hệ điều hành Android và ngôn ngữ lập trình Kotlin.
* Tiếp theo, lựa chọn về kiến trúc tổng quan, liên quan đến giao diện của ứng dụng được trình bày.
* Sau đó, cơ sở dữ liệu một số phần mở rộng của nó dùng trong ứng dụng sẽ được nhắc qua.
* Cuối cùng là thông tin về .cbz - định dạng tệp tin mà ứng dụng đọc, gồm hai phần: sơ lược về định dạng .zip mà .cbz dựa trên, và các trường metadata trong tệp tin .cbz - nguồn thông tin quan trọng để quản lí truyện.

### 2.1. Hệ điều hành Android

#### 2.1.1. Android Jetpack

### 2.2. Ngôn ngữ lập trình Kotlin

#### 2.2.1. Coroutine

### 2.3. Mẫu thiết kế MVVM và Kiến trúc khuyên dùng bởi Google

### 2.4. SQLite

#### 2.4.1. Full-text Search

#### 2.4.2. Room

### 2.5. Định dạng tệp nén .zip và .cbz

#### 2.5.1. Định dạng tệp nén .zip

#### 2.5.2. Định dạng tệp nén .cbz

## 3. Chương 3: Phân tích yêu cầu & Thiết kế

Chương này phân tích yêu cầu để lập ra đặc tả yêu cầu, là bộ khung cho quá trình phát triển ứng dụng.

### 3.1. Mô tả chung

#### 3.1.1. Người dùng

Ứng dụng yacv tập trung vào một số ít người dùng, là một trong hai nhóm sau:

* Người dùng sưu tầm truyện
* Người dùng có yêu cầu đọc truyện với chất lượng hình ảnh cao

Cả hai nhóm có điểm chung là kĩ tính, yêu cầu cao về trải nghiệm đọc truyện. Cũng do kĩ tính, nên cả hai nhóm không cần nhiều chức năng, tuy nhiên có yêu cầu cao về từng chức năng. Nhóm người dùng sưu tầm truyện còn có yêu cầu về xem thông tin (metadata) của tệp truyện.

#### 3.1.2. Mục đích

Trước khi đi vào chi tiết yêu cầu ở mục tiếp theo, tôi muốn làm rõ mục đích của sản phẩm đã nhắc ở [mục 1.1](#P1.1-background).

* Ứng dụng yacv chỉ bao gồm các tính năng liên quan đến đọc **truyện tranh** ngoại tuyến (tức đọc các tệp truyện có sẵn trên điện thoại người dùng).
* Ứng dụng *không phải* là ứng dụng khách cho các trang đọc truyện hiện có, hay có máy chủ tập trung riêng để cung cấp truyện.
* Ứng dụng *không có* khả năng đọc truyện đuôi .pdf, cùng với các định dạng truyện thiên về chữ khác như .txt, .epub.

Các giới hạn này nhằm tránh cho phần mềm quá phức tạp với tôi, đồng thời phù hợp (không thừa thiếu chức năng) so với nhu cầu của nhóm người dùng mục tiêu đã nêu ở [mục 3.1.1](#P3.1.1-users).

### 3.2. Yêu cầu đặt ra

#### 3.2.1. Yêu cầu chức năng

Ứng dụng có các chức năng chính sau:

* Quét các tệp truyện trên thiết bị
* Hiển thị danh sách truyện
* Đọc truyện
* Xem metadata truyện
* Tìm kiếm truyện
* Xóa truyện

#### 3.2.2. Yêu cầu phi chức năng

Ứng dụng cần đạt một số tiêu chí sau:

* Phản hồi nhanh: Các thao tác cần có thời gian phản hồi nhanh. Phản hồi nhanh không nhất thiết là thời gian thực thi ngắn, mà là luôn có các thông báo tiến độ cho người dùng.
  + Luôn hiện thông báo chờ khi làm việc gì đó lâu
  + Nếu có nhiều kết quả tìm kiếm, hiển thị từ từ, đưa những kết quả đã biết lên trước
* Tốc độ xem truyện chấp nhận được: Tốc độ xem truyện gồm tốc độ mở truyện, tốc độ duyệt trang tới-lui đều cần phải có tốc độ chấp nhận được để cải thiện trải nghiệm.
* Chiếm dụng ít bộ nhớ: Bộ nhớ chiếm dụng của ứng dụng gồm hai phần: bộ nhớ RAM và bộ nhớ tạm, cả hai cần sử dụng ít dung lượng nhất có thể. Đây là một yêu cầu đáng cân nhắc, lí do vì kích cỡ từng tệp truyện thường rất lớn (từ vài chục đến hơn một trăm megabyte), tuy nhiên cần chú ý cân bằng yêu cầu này với yêu cầu về tốc độ (đánh đổi không gian-thời gian).
* Giao diện đơn giản, trực quan: Người dùng hướng đến có thể xếp vào nhóm người dùng “say mê” (enthusiast), do đó giao diện chỉ cần đơn giản rõ ràng, không màu mè, tập trung vào tính năng.

### 3.3 Phân tích yêu cầu

Mỗi yêu cầu đã xác định trong [mục 3.2.1.](#P3.2.1-functional-requirements) được coi là một ca sử dụng, được trình bày trong các tiểu mục dưới đây. Người dùng duy nhất trong các ca sử dụng là người đọc, do đó hai cụm từ này sẽ được dùng hoán đổi cho nhau. Do ứng dụng hoàn toàn ngoại tuyến, người đọc cũng không có tương tác với nhau.

#### 3.3.1. Quét các tệp truyện trên thiết bị

* **Mô tả ngắn gọn**:
* Người đọc chọn một thư mục trong điện thoại làm thư mục gốc. Ứng dụng sẽ quét thư mục này và tìm các tệp truyện, rồi hiển thị những thư mục chứa tệp truyện cho người đọc duyệt.
* **Mô tả từng bước**:
  1. Người đọc bật ứng dụng
  2. Người đọc ấn vào nút thay đổi thư mục gốc.
  3. Trình chọn thư mục của Android hiện ra, cho phép người đọc chọn thư mục làm thư mục gốc.
  4. Các thư mục chứa truyện trong thư mục gốc được hiển thị lên màn hình theo tiến độ, ứng dụng quét đến đâu hiển thị đến đấy. Ảnh đại diện cho thư mục là bìa một truyện bất kì tìm được trong thư mục đó. Tới đây người đọc có thể duyệt theo thư mục rồi xem truyện, hoặc thực hiện các ca sử dụng khác.
  5. Nếu người đọc đã chọn một thư mục gốc, động tác chọn thư mục gốc này sẽ thay thế thư mục gốc cũ bằng thư mục được chọn, và quá trình quét được kích hoạt lại như lần chọn đầu tiên.
  6. Nếu có lỗi trong bản thân quá trình chọn thư mục (gồm thư mục gốc không có truyện, thư mục không tìm được, thiếu quyền), cần gợi ý người đọc chọn lại. Lỗi trong quá trình quét cần phải giảm thiểu và giấu khỏi người đọc nếu có.

Đây là ca sử dụng đầu tiên khi người đọc khởi động ứng dụng lần đầu. Các tệp truyện sẽ được quét từ thư mục gốc, rồi được gom lại theo thư mục theo mô tả ở [ca sử dụng tiếp theo](#P3.3.2-show-library).

Màn hình đầu tiên khi người đọc bật lên gọi là Màn hình Thư viện (Library screen). Các thư mục chứa truyện, hoặc thông báo lỗi liên quan đến bản thân quá trình chọn truyện (đã miêu tả trong bước 6 ở trên) sẽ được hiển thị ở màn hình này.

#### 3.3.2. Hiển thị danh sách truyện

* **Mô tả ngắn gọn**:
* Người đọc duyệt truyện theo thư mục, rồi chọn truyện và xem.
* **Mô tả từng bước**:
  1. Người đọc bật ứng dụng, đã chọn thư mục gốc, đã quét được ít nhất một thư mục chứa truyện (đang ở Màn hình Thư viện).
  2. Ứng dụng hiển thị *danh sách thư mục* chứa truyện cho người đọc xem và chọn.
  3. Người đọc chọn một thư mục.
  4. Ứng dụng hiển thị *danh sách truyện* trong thư mục đó cho người đọc xem và chọn. Danh sách truyện cần thể hiện được tiến độ đọc của từng truyện, và đánh giá yêu thích của người đọc.

Đây là một trong hai ca sử dụng chính của ứng dụng, bên cạnh (và là tiền điều kiện cho) [ca sử dụng đọc truyện](#P3.3.3-read-comic) sẽ được miêu tả tiếp theo.

Màn hình khi người đọc chọn một thư mục gọi là Màn hình Thư mục (Directory screen). Cũng giống như Màn hình Thư viện, ảnh bìa và tên của truyện được hiển thị để người đọc chọn.

Tong ứng dụng, truyện sẽ được quản lí và duyệt theo thư mục. Có hai lí do cho lựa chọn thiết kế này:

* Giảm độ phức tạp khi lập trình
* Các phương pháp duyệt khác không trực quan
  + Các phương pháp duyệt khác chỉ bao gồm duyệt theo metadata, tức duyệt theo các thông tin đi kèm như Tác giả, Nhân vật, Bộ truyện,… thì yêu cầu truyện phải có đủ metadata. Trên thực tế, không phải tệp truyện nào cũng có đủ thông tin này, do vậy sẽ có trường hợp rất nhiều truyện bị gom vào mục “Không đủ thông tin”. Hơn nữa, giả sử truyện có đi kèm metadata, ta xem xét tiếp trường hợp dưới.
  + Giả sử ta quản lí theo Nhân vật: Vậy để trực quan, yacv phải hiển thị ảnh nhân vật. Hiện nay, việc nhận diện và cắt đúng ảnh phần mặt nhân vật ra để tạo ảnh đại diện có thể nói là bất khả thi. Do vậy, khi duyệt theo Nhân vật, người đọc chỉ có thể thấy tên, không thấy một hình ảnh gợi ý nào khác, dẫn đến khó khăn khi sử dụng. Lập luận tương tự có thể dùng với các cách xếp khác.
  + Một cách xếp có thể nói là tốt là xếp theo Bộ truyện, tuy nhiên ta lại quay về vấn đề thiếu metada.

Hơn nữa, các thư mục cần được “làm phẳng”. “Làm phẳng” có nghĩa là hiển thị thư mục con (cháu,…) ngang hàng với thư mục gốc. Ví dụ sau cho thấy cách yacv làm phẳng cây thư mục:

| Cây thư mục gốc | yacv đã làm phẳng |  
|-----------------------------------|---------------------------|  
| thư mục gốc | thư mục gốc |  
| ├── Original Sin #1.cbz | └── Original Sin #1.cbz |  
| └── House of M | House of M |  
| ├── House of M #1.cbz | ├── House of M #1.cbz |  
| ├── House of M #3.cbz | └── House of M #3.cbz |  
| └── Tie-ins | Tie-ins |  
| └── Black Panther #7.cbz | └── Black Panther #7.cbz |

Bảng 1: Cách yacv làm phẳng thư mục

Theo như cột phải Bảng 1, các màn hình trong yacv được tổ chức như sau:

* Màn hình Thư viện: có 3 thư mục:
  + thư mục gốc
  + House of M
  + Tie-ins
* Khi chọn “House of M”: chuyển sang Màn hình Thư mục tương ứng, không có thư mục con, và có 2 tệp truyện:
  + House of M #1.cbz
  + House of M #3.cbz
* Tương tự với các thư mục khác.

Có ba lí do cho lựa chọn thiết kế này:

* Giảm độ phức tạp khi lập trình.
* Người đọc không phải đi qua nhiều tầng thư mục để đến được tệp truyện cần đọc.
* Không có ca sử dụng có ý nghĩa cho thư mục lồng nhau:
* Trường hợp hợp lí nhất cho việc có thư mục lồng nhau là khi lưu các tệp truyện liên quan đến một bộ truyện (tie-ins), như cột trái Bảng 1:
  + Thư mục cha (House of M) chứa tệp truyện trong bộ truyện cùng tên và thư mục tie-ins.
  + Thư mục Tie-ins chứa các tệp truyện tie-in.
* Tuy nhiên, bản thân các tệp tie-in lại là tệp truyện thông thường trong một bộ truyện khác, do đó nếu tổ chức thư mục như thế này sẽ dẫn đến tình trạng lặp tệp truyện, là điều không mong muốn ngay cả với máy tính.

#### 3.3.3. Đọc truyện

* **Mô tả ngắn gọn**:
* Người đọc chọn một truyện để xem.
* **Mô tả từng bước**:
  1. Người đọc bật ứng dụng, đã chọn thư mục gốc, đã quét được ít nhất một thư mục chứa truyện, đã chọn một thư mục (đang ở Màn hình Thư mục).
  2. Ứng dụng hiển thị danh sách truyện trong thư mục đó cho người đọc xem và chọn.
  3. Người đọc chọn một truyện và đọc.
  4. Người đọc vuốt qua lại theo phương ngang để chuyển trang, có thể đánh dấu trang truyện, có thể lưu lại trang truyện ở dạng ảnh.
  5. Nếu người đọc đã đọc truyện:
     + Ứng dụng cần đưa về chính trang truyện đang đọc dở.
     + Nếu đã đọc đến trang cuối, tức đã đọc xong, ứng dụng cần đưa về trang đầu tiên.

Đây là một trong hai ca sử dụng chính của ứng dụng, bên cạnh (và là mục đích của) [ca sử dụng hiển thị danh sách truyện](#P3.3.2-show-library) đã được miêu tả ở trên.

Màn hình khi người đọc đọc một truyện gọi là Màn hình Đọc truyện. Màn hình này cho phép người đọc duyệt các trang truyện theo phương ngang, có thêm hai tính năng đánh dấu và lưu trang truyện như miêu tả ở bước 4. Mục tiêu là thiết kế màn hình này sao cho có trải nghiệm gần giống nhất với ứng dụng Thư viện ảnh (Gallery) tích hợp trong mọi điện thoại Android.

#### 3.3.4. Xem metadata truyện

* **Mô tả ngắn gọn**:
* Trong Màn hình Đọc truyện, người đọc ấn nút để xem metadata.
* **Mô tả từng bước**:
  1. Người đọc bật ứng dụng, chọn một truyện để vào đến Màn hình Đọc truyện.
  2. Người đọc ấn nút Xem metadata.
  3. Ứng dụng hiển thị mọi metadata, bao gồm cả những trường bị thiếu. Ảnh bìa của truyện cũng được hiển thị kèm.
  4. Người dùng có thể đánh giá truyện bằng nút Yêu thích trong màn hình này, hoặc ngược lại (bỏ đánh giá Yêu thích).

Đây là một ca sử dụng phụ, có thể được kích hoạt khi người dùng đang ở Màn hình Đọc truyện.

Màn hình khi người đọc xem metadata gọi là Màn hình Metadata. Màn hình này có thể có chức năng sửa metadata, tùy theo tiến độ khóa luận để xem xét có cài đặt không.

Hệ thống đánh giá của ứng dụng chỉ ở mức cơ bản, gồm duy nhất tính năng Yêu thích. Tính năng này cũng chỉ phục vụ hai mục đích là thể hiện sự đánh giá của người dùng và lọc nhanh truyện về mặt thị giác (đã nhắc đến trong phần Mô tả từng bước của [ca sử dụng hiển thị danh sách truyện](#P3.3.2-show-library)).

Các tính năng nâng cao hơn như gợi ý không xuất hiện, do một số lí do sau:

* Giảm độ phức tạp khi lập trình.
* Người dùng không có nhu cầu: nhóm người dùng hướng đến có đặc điểm hiểu biết về truyện tranh, do đó việc gợi ý có thể coi là thừa thãi.
* Thiểu thông tin gợi ý: việc gợi ý chỉ có hiệu quả khi có một cơ sở dữ liệu về các bộ truyện liên quan, hoặc lựa chọn các truyện liên quan của cộng đồng người đọc, trong khi yacv là một ứng dụng hoàn toàn ngoại tuyến.

#### 3.3.5. Tìm kiếm truyện

* **Mô tả ngắn gọn**:
* Trong Màn hình Thư viện, người đọc ấn nút để tìm kiếm truyện.
* **Mô tả từng bước**:
  1. Người đọc bật ứng dụng.
  2. Người đọc ấn nút Tìm kiếm, và gõ từ khóa cần tìm, và ấn nút Enter.
  3. Ứng dụng hiển thị kết quả tìm kiếm theo metadata và tên tệp truyện
     + Kết quả tìm kiểm cần được gom theo nhóm dựa vào trường metadata tìm thấy được.
     + Nếu có thể, hiển thị ảnh bìa của truyện.

Đây là một ca sử dụng phụ, có thể được kích hoạt khi người dùng đang ở Màn hình Thư viện.

Màn hình khi người đọc xem kết quả tìm kiếm gọi là Màn hình Kết quả. Màn hình này chỉ hiện ra khi người dùng ấn nút Enter để chính thức tìm kiếm.

Màn hình kết quả phải nhóm kết quả theo trường metadata mà kết quả tìm thấy được. Lấy ví dụ, người dùng tìm kiếm “Watchmen” sẽ nhận được Màn hình Kết quả gần như sau:

Truyện  
- Watchmen #1.cbz  
- Watchmen #2.cbz  
Bộ truyện  
- Watchmen

Hình 1. Mô tả giao diện Màn hình Kết quả khi tìm kiếm truyện.

Tương tác của người đọc với Màn hình Kết quả trên diễn ra như sau:

* Khi ấn vào một mục trong danh sách “Truyện”, người đọc được đưa đến thẳng Màn hình Đọc truyện của truyện đó (và hiển thị ở trang đọc dở như đã mô tả ở trong [ca sử dụng đọc truyện](#P3.3.3-read-comic)).
* Khi ấn vào một mục trong danh sách “Bộ truyện”, người đọc được đưa đến màn hình chứa danh sách những truyện trong bộ truyện đã chọn. Màn hình này cần giống với Màn hình Thư mục. Sau đó, người dùng chọn một truyện để đọc như bình thường.

Đây chỉ là ví dụ về một từ khóa có kết quả khi tìm theo tên tệp truyện và bộ truyện. Các trường metadata khác nếu có kết quả phù hợp cũng sẽ thể hiện theo hình thức trên.

Chú ý rằng bìa truyện luôn được thể hiện khi có thể. Trong ví dụ trên, chắc chắn phải có bìa truyện cho mọi mục con trong danh sách “Truyện”.

Nếu không có kết quả, cần thể hiện rõ cho người dùng biết.

Khi người dùng gõ từ khóa để tìm kiếm, một số thông tin gợi ý tìm kiếm (từ khóa cũ, từ khóa liên quan) có thể hiện ra; tính năng này tùy theo tiến độ khóa luận để xem xét có cài đặt không. Chỉ khi người dùng ấn Enter, quá trình tìm kiếm mới bắt đầu, và hiển thị kết quả, chứ không hiển thị kết quả theo quá trình người dùng gõ phím. Lí do cho lựa chọn này như sau:

* Giảm độ phức tạp khi lập trình.
* Không quá cần thiết: ví dụ, bộ máy tìm kiếm của Google cũng chỉ hiện gợi ý khi người dùng nhập từ khóa tìm kiếm, phải đến khi ấn Enter thì quá trình tìm kiếm mới diễn ra và kết quả chi tiết được hiển thị.

Với độ phức tạp dự kiến của việc hiển thị ảnh bìa truyện, đây có thể được xem là đánh đổi hợp lí để tăng hiệu năng ứng dụng.

#### 3.3.6. Xóa truyện

* **Mô tả ngắn gọn**:
* Người dùng chọn một số truyện trong một màn hình chứa danh sách truyện để xóa.
* **Mô tả từng bước**:
  1. Người dùng truy cập vào một màn hình chứa danh sách truyện (là Màn hình Thư mục hoặc Màn hình Kết quả).
  2. Người dùng ấn và giữ vào một truyện.
  3. Màn hình đó sẽ chuyển sang chế độ xóa, báo hiệu bằng biểu tượng Thùng rác trên màn hình, và ô đánh dấu để xóa ở cạnh mỗi truyện. Truyện mà người dùng ấn giữ phải được đánh dấu xóa ngay.
  4. Người dùng có thể chọn thêm truyện để xóa nếu muốn.
  5. Người dùng ấn nút xóa để xóa truyện.
  6. Ứng dụng hiện ra hộp thoại xóa, ghi rõ rằng truyện sẽ được xóa khỏi bộ nhớ điện thoại, số truyện sẽ xóa, và hỏi người dùng có thực sự muốn xóa không.
  7. Nếu người dùng ấn vào nút Đồng ý xóa, truyện sẽ được xóa khỏi bộ nhớ điện thoại và tắt hộp thoại, nếu không thì tắt hộp thoại.
  8. Sau khi tắt hộp thoại, màn hình trở về chế độ thông thường, biểu tượng thùng rác cũng biến mất.

Ca sử dụng này không có màn hình riêng biệt, mà sử dụng một chế độ của các màn hình hiển thị danh sách truyện.

## 4. Chương 4: Thiết kế

### 4.1. Thiết kế hướng đối tượng

### 4.2. Thiết kế CSDL

### 4.3. Thiết kế giao diện

### 4.4. Tối ưu

## 5. Chương 5: Lập trình & Kiểm thử

### 5.1. Lập trình

### 5.2. Kiểm thử

#### 5.2.1 Unit test. E2E test, auto + manual test

#### 4.2.2 Kiểm thử phi chức năng

### 5.3 Đánh giá người dùng

## 6. Chương 6: Kết luận

## 7. Tài liệu tham khảo