

NGUYỄN VĂN ĐÔ

KỸ THUẬT PHẦN MỀM

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KỸ THUẬT PHẦN MỀM

XÂY DỰNG ỨNG DỤNG HỖ TRỢ ĐỌC TRUYỆN SỬ DỤNG
REACT NATIVE

GVHD	: TS. Hà Mạnh Đào
Sinh viên	: Nguyễn Văn Đô
Mã sinh viên	: 2019600676

MỤC LỤC

MỞ ĐẦU.....	7
CHƯƠNG 1. TỔNG QUAN VỀ ỨNG DỤNG HỖ TRỢ ĐỌC TRUYỆN	9
1.1 Giới thiệu về truyện.....	9
1.2 Giới thiệu về ứng dụng di động.....	9
1.2.1 Ứng dụng di động là gì ?	9
1.2.2 Quy trình phát triển ứng dụng di động.....	11
1.3 Ngôn ngữ lập trình Javascript.....	13
1.4 Framework React Native	14
1.6 Giới thiệu về Firebase	15
1.8 Tóm tắt chương.....	17
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ ỨNG DỤNG HỖ TRỢ ĐỌC TRUYỆN.....	18
2.1 Giới thiệu ứng dụng đọc truyện	18
2.1.1 Mô tả ứng dụng	18
2.1.2 Các chức năng chính	18
2.1.3 Các yêu cầu phi chức năng	19
2.2 Biểu đồ Usecase tổng quát.....	19
2.3 Mô tả chi tiết các usecase	20
2.3.1 Mô tả UC “Đăng nhập”	20
2.3.2 Mô tả UC “Đăng kí”	20
2.3.3 Mô tả UC “Tìm kiếm truyện”	21
2.3.4 Mô tả UC “Xem chi tiết truyện”	21
2.3.5 Mô tả UC “Yêu thích”	22
2.3.6 Mô tả UC “ Bình luận”	22
2.3.7 Mô tả UC “ Thêm tác giả “	23
2.3.8 Mô tả UC ”Xóa tác giả”	23
2.3.9 Mô tả UC “Thêm thể loại”.....	24
2.3.10 Mô tả UC “Thêm truyện”	24
2.3.11 Mô tả UC “ Xóa truyện”	25
2.3.12 Mô tả UC “Xóa thể loại”	25
2.4 Biểu đồ trình tự	26

2.4.1 Biểu đồ trình tự UC “Đăng nhập”	26
2.4.2 Biểu đồ trình tự UC “Đăng kí”	27
2.4.3 Biểu đồ trình tự UC “Tìm kiếm truyện”	28
2.4.4 Biểu đồ trình tự UC “Yêu thích truyện”	29
2.4.5 Biểu đồ trình tự UC “Bình luận”	30
2.4.6 Biểu đồ trình tự UC “”Thêm tác giả”	31
2.4.7 Biểu đồ trình tự UC “Xóa tác giả”	32
2.4.8 Biểu đồ trình tự UC “Xóa thể loại”	33
2.4.9 Biểu đồ trình tự UC “Thêm thể loại”	34
2.4.10 Biểu đồ trình tự UC “Xóa truyện”	35
2.4.11 Biểu đồ UC “Thêm truyện”	36
2.5 Thiết kế cơ sở dữ liệu	37
2.5.1 Biểu đồ thực thể liên kết.....	37
2.5.2 Chi tiết các bảng.....	37
2.6 Thiết kế giao diện.....	39
2.6.1.Giao diện đăng nhập.....	39
2.6.2 Giao diện đăng kí	40
2.6.3 Giao diện trang chủ	41
2.6.4 Giao diện Xem chi tiết truyện.....	42
2.6.5 Giao diện Thêm mới thể loại	43
2.6.6 Giao diện Thêm mới tác giả	44
2.6.7 Giao diện thêm mới Thêm truyện	45
2.7 Tóm tắt chương.....	45
CHƯƠNG 3 .THỬ NGHIỆM VÀ ĐÁNH GIÁ.....	46
3.1 Mô hình thử nghiệm.....	46
3.2 Kiểm thử ứng dụng hỗ trợ đọc truyện.....	46
3.2.1 Kế hoạch kiểm thử	46
3.2.2 Mục tiêu kiểm thử	46
3.2.3 Chiến lược kiểm thử.....	47
3.2.4 Kịch bản kiểm thử	47
3.2.5 Báo cáo kiểm thử.....	48
3.3 Tóm Tắt Chương.....	48
KẾT LUẬN.....	49
TÀI LIỆU THAM KHẢO	50

DANH MỤC CÁC KÍ HIỆU, CÁC CHỮ VIẾT TẮT

STT	Tên Viết tắt	Tiếng Anh	Nghĩa tiếng Việt
1	UC	Use case	Trường hợp sử dụng
2	JS	Javascript	Ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác.
3	VSCode	Visual Studio Code	Một trình soạn thảo mã nguồn.
4	UI	User InterFace	Giao diện người dùng
5	UX	User Experience	Trải nghiệm người dùng
6	API	Application Programming Interface	Giao diện lập trình ứng dụng

DANH MỤC BẢNG BIỂU

Bảng 1 Chi tiết bảng user.....	37
Bảng 2 Chi tiết bảng story.....	37
Bảng 3 Chi tiết bảng Author	38
Bảng 4 Chi tiết bảng Favorite.....	38
Bảng 5 Chi tiết bảng comment.....	38
Bảng 6 Chi tiết bảng Genre.....	39
Bảng 7 Báo cáo kiểm thử.....	48

DANH MỤC HÌNH VẼ

Hình 2.1 Biểu đồ usecase tổng quát.....	19
Hình 2.2 Biểu đồ trình tự UC đăng nhập.....	26
Hình 2.3 Biểu đồ trình tự UC đăng kí.....	27
Hình 2.4 Biểu đồ trình tự UC tìm kiếm truyện.....	28
Hình 2.5 Biểu đồ trình tự UC yêu thích.....	29
Hình 2.6 Biểu đồ trình tự UC Bình Luận.....	30
Hình 2.7 Biểu đồ trình tự UC Thêm Tác Giả.....	31
Hình 2.8 Biểu đồ trình tự UC xóa tác giả.....	32
Hình 2.9 Biểu đồ trình tự UC xóa thể loại.....	33
Hình 2.10 Biểu đồ trình tự UC Thêm thể loại.....	34
Hình 2.11 Biểu đồ trình tự UC xóa truyện.....	35
Hình 2.12 Biểu đồ trình tự UC thêm truyện.....	36
Hình 2.13 Biểu đồ thực thể liên kết.....	37
Hình 2.14 Giao diện màn hình đăng nhập.....	39
Hình 2.15 Giao diện màn hình đăng kí.....	40
Hình 2.16 Giao diện trang chủ.....	41
Hình 2.17 Giao diện xem chi tiết truyện.....	42
Hình 2.18 Giao diện thêm mới thể loại.....	43
Hình 2.19 Giao diện thêm tác giả	44
Hình 2.20 Giao diện thêm truyện.....	45
Hình 3.1 Mô hình thử nghiệm.....	46
Hình 3.2 Mô tả testcase đăng kí.....	47
Hình 3.3 Mô tả testcase đăng nhập.....	47
Hình 3.4 Mô tả testcase bình luận.....	47
Hình 3.5 Mô tả Testcase yêu thích.....	48
Hình 3.6 Mô tả testcase tìm kiếm.....	48

MỞ ĐẦU

1. Lý do chọn đề tài

Những năm gần đây, các dịch vụ Internet và viễn thông phát triển rất mạnh. Điều này tác động không nhỏ tới cách thức, hành vi giải trí của mọi người. Với sự sẵn có, tiện lợi của Internet, mọi người thường đọc truyện trực tuyến thay vì tìm sách báo như ngày xưa. Cho nên vấn đề xử lý sao cho việc đọc, tìm kiếm truyện được dễ dàng tiện lợi, có hiệu quả bằng các ứng dụng, nền tảng đọc truyện trực tuyến là cấp thiết hơn bao giờ hết. Trong đó ứng dụng di động đọc truyện có những tính năng vượt trội hơn so với việc đọc trên website như đọc offline, lưu truyện dễ dàng nên đề án quyết định xây dựng ứng dụng đọc truyện bằng React Native.

Hiện nay nhiều ứng dụng hay website cung cấp hàng ngàn bộ truyện, đầu truyện cho tất cả mọi người có thể tham khảo và đọc. Để tạo ra một nơi cung cấp truyện phong phú cũng như cộng đồng độc giả có thể giao lưu với nhau nên đề án xây dựng ứng dụng đọc truyện online và offline cho mọi người.

2. Mục đích của đề tài

- Nghiên cứu, khảo sát hiện trạng đọc truyện trực tuyến qua mạng Internet hiện nay, các hệ thống ứng dụng đọc truyện phổ biến.
- Nghiên cứu và làm chủ framework React Native.
- Xây dựng một ứng dụng hỗ trợ đọc truyện bằng React Native.
- Cài đặt, triển khai thử nghiệm ứng dụng.

3. Đối tượng

- Nghiên cứu về truyện.
- Nghiên cứu, phân tích và thiết kế hệ thống đọc truyện trực tuyến.
- Nghiên cứu về framework React Native và các thực hoạt động.

4. Phạm vi nghiên cứu

- Tổng quan tình hình nghiên cứu trong và ngoài nước đã thực hiện liên quan đến đề tài.

- Nghiên cứu một số ứng dụng và website đọc truyện.
- Nghiên cứu ứng dụng React Native xây dựng thử nghiệm ứng dụng đọc truyện trực tuyến.
- Ứng dụng được đưa vào thử nghiệm đối với một số người dùng.

5. Ý nghĩa khoa học và thực tiễn của đề tài

- Đưa ra một ứng dụng đọc truyện trực tuyến mới bằng React Native.
- Hệ thống đáp ứng được tình hình hiện tại và có thể áp dụng lâu dài.

6. Kết quả đạt được

- Ứng dụng đọc truyện bằng React Native.
- Báo cáo đồ án tốt nghiệp.

7. Bố cục báo cáo

- Chương 1. Tổng quan về ứng dụng đọc truyện bằng React Native : nêu ra đặc điểm, mô hình và công cụ phát triển của ứng dụng.
- Chương 2. Phân tích và thiết kế ứng dụng đọc truyện sử dụng React Native : trình bày mô tả hệ thống, usecase và các biểu đồ phân tích, giao diện ứng dụng.
- Chương 3. Thử nghiệm và đánh giá : đưa ra các kịch bản thử nghiệm và đánh giá .

CHƯƠNG 1. TỔNG QUAN VỀ ỨNG DỤNG HỖ TRỢ ĐỌC TRUYỆN

1.1 Giới thiệu về truyện

Truyện là một thể loại văn học dùng để kể lại một câu chuyện thông qua việc sắp xếp các sự kiện, tình tiết, và nhân vật. Truyện thường được viết dưới nhiều hình thức khác nhau, bao gồm truyện ngắn, tiểu thuyết, truyện tranh.

Câu chuyện trong một truyện có thể là hư cấu hoặc dựa trên sự thật, và có thể chứa đựng nhiều yếu tố như tình yêu, hành động, giả tưởng, hài hước, và kinh dị. Mục đích của việc viết truyện thường là giải trí, giáo dục, hoặc để thể hiện các ý tưởng, giá trị, hoặc tầm nhìn của tác giả.

Truyện thường được phân phối qua nhiều phương tiện, từ sách in, tạp chí, đến các ứng dụng và trang web trực tuyến. Độ dài của một câu chuyện có thể từ một vài trang đến hàng trăm trang, tùy thuộc vào thể loại và mục đích của tác phẩm.

1.2 Giới thiệu về ứng dụng di động

1.2.1 Ứng dụng di động là gì ?

Ứng dụng di động (mobile application hay mobile app) là một phần mềm được thiết kế để chạy trên các thiết bị di động như điện thoại thông minh và máy tính bảng. Ứng dụng di động có thể được cài đặt và sử dụng trên các hệ điều hành di động như Android của Google, iOS của Apple, và một số hệ điều hành khác như Windows Phone và Blackberry OS.

Hiện nay ứng dụng di động được chia làm 2 loại là : **Native Mobile App** và **Hybird Mobile App**.

1.2.1.1 Giới thiệu về Native Mobile App

Native Mobile App là loại ứng dụng di động được phát triển đặc biệt cho một hệ điều hành cụ thể như Android hoặc iOS. Các ứng dụng này được viết bằng các ngôn ngữ lập trình và sử dụng các công cụ phát triển riêng của hệ

điều hành đó. Chúng có thể tận dụng tối đa các tính năng phần cứng và phần mềm của thiết bị, như camera, GPS, cảm biến, và thông báo đẩy.

Ưu điểm :

- Tích hợp sâu với hệ điều hành: Native Mobile App có thể tích hợp sâu với các dịch vụ và tính năng của hệ điều hành, giúp cung cấp trải nghiệm người dùng tốt hơn.
- Hiệu suất tối ưu: Nhờ được viết và tối ưu hóa cho hệ điều hành cụ thể, Native Mobile App thường chạy mượt mà và nhanh chóng.
- Trải nghiệm người dùng tốt: Native Mobile có thể cung cấp giao diện và trải nghiệm người dùng nhất quán với hệ điều hành, làm cho người dùng cảm thấy quen thuộc và dễ sử dụng.
- Thông báo đẩy : Dễ dàng tích hợp và quản lý thông báo đẩy, giúp tương tác với người dùng hiệu quả hơn.

Nhược điểm :

- Chi phí phát triển cao: Việc phát triển Native Mobile App cho nhiều nền tảng khác nhau (Android và iOS) đòi hỏi phải viết mã và duy trì hai ứng dụng riêng biệt, dẫn đến chi phí cao hơn.
- Thời gian phát triển dài: Quá trình phát triển Native Mobile App thường mất nhiều thời gian hơn do cần phải phát triển, kiểm thử và tối ưu hóa cho từng hệ điều hành riêng biệt.
- Cập nhật phức tạp: Mỗi khi có cập nhật mới, cần phải thực hiện và quản lý các bản cập nhật cho từng nền tảng riêng biệt.

1.2.1.2 Giới thiệu về Hybrid Mobile App

Hybrid Mobile App là loại ứng dụng di động kết hợp giữa native app và ứng mobile web app. Chúng được phát triển bằng các công nghệ web như HTML5, CSS, và JavaScript, sau đó đóng gói và triển khai trên các nền tảng di động thông qua một framework như React Native hoặc Xamarin. Hybrid Mobile App cho phép chúng chạy trên nhiều nền tảng khác nhau mà không cần viết mã riêng biệt cho từng hệ điều hành.

Ưu điểm :

- Phát triển nhanh: Việc sử dụng mã nguồn chung giúp giảm thời gian phát triển.
- Tiết kiệm chi phí: Chi phí phát triển và bảo trì thấp hơn do chỉ cần duy trì một mã nguồn.
- Cập nhật dễ dàng: Khi cần cập nhật hoặc sửa lỗi, chỉ cần thực hiện một lần cho tất cả các nền tảng.
- Truy cập tính năng thiết bị: Các framework như React Native và Flutter cho phép truy cập các tính năng phân cứng của thiết bị

Nhược điểm :

- Hiệu suất thấp hơn: Hybrid App thường có hiệu suất thấp hơn so với Native App do không sử dụng được các API như Native App
- Trải nghiệm người dùng không tối ưu: Giao diện và trải nghiệm người dùng có thể không mượt mà như Native App, đặc biệt là với các ứng dụng phức tạp hoặc yêu cầu hiệu suất cao.
- Giới hạn về tính năng: Một số tính năng phân cứng và phần mềm đặc thù của hệ điều hành có thể khó truy cập hoặc không thể sử dụng đầy đủ.

1.2.2 Quy trình phát triển ứng dụng di động

Quy trình phát triển ứng dụng di động có thể mô tả bằng các bước sau :

- Xác định ý tưởng và nghiên cứu thị trường : Hiểu rõ mục tiêu của ứng dụng, xác định vấn đề cần giải quyết. Phân tích thị trường và nghiên cứu các ứng dụng cạnh tranh. Lên danh sách các tính năng chính của ứng dụng và xác định phạm vi dự án.
- Lập kế hoạch và phân tích : Xác định lịch trình phát triển, các giai đoạn quan trọng, và phân bổ nguồn lực. Tạo tài liệu yêu cầu chi tiết cho ứng dụng, bao gồm yêu cầu chức năng và phi chức năng. Tạo wireframe (bản phác thảo) và prototype (mẫu thử nghiệm) để mô phỏng giao diện và luồng người dùng.

- **Thiết kế giao diện người dùng :** Thiết kế chi tiết giao diện người dùng và trải nghiệm người dùng, đảm bảo tính thẩm mỹ và dễ sử dụng. Lập kế hoạch cho kiến trúc ứng dụng, bao gồm kiến trúc front-end và back-end, cơ sở dữ liệu, và tích hợp API.
- **Phát triển:** Viết mã cho phần giao diện người dùng, sử dụng các ngôn ngữ lập trình và framework phù hợp. Thiết kế và phát triển máy chủ, cơ sở dữ liệu, và các API cần thiết. Kết nối front-end và back-end, tích hợp các dịch vụ của bên thứ ba nếu cần.
- **Kiểm thử :** Đảm bảo tất cả các tính năng hoạt động đúng như yêu cầu. Đo lường và tối ưu hóa hiệu suất của ứng dụng. Xác định và khắc phục các lỗi hỏng bảo mật. Đảm bảo ứng dụng hoạt động tốt trên nhiều loại thiết bị và hệ điều hành khác nhau.
- **Triển khai :** Tạo tài khoản nhà phát triển trên các cửa hàng ứng dụng (Google Play Store, Apple App Store), và chuẩn bị các tài liệu cần thiết như mô tả ứng dụng, ảnh chụp màn hình, video giới thiệu. Đưa ứng dụng lên các cửa hàng ứng dụng và theo dõi quá trình phê duyệt.
- **Duy trì và cập nhật :** Sử dụng các công cụ phân tích để theo dõi hiệu suất và hành vi người dùng. Sửa lỗi và cập nhật ứng dụng để cải thiện hiệu suất và bảo mật. Thêm tính năng mới và cập nhật để giữ cho ứng dụng luôn mới mẻ và phù hợp với nhu cầu người dùng.
- **Thu thập phản hồi và cải tiến.** Lắng nghe phản hồi từ người dùng qua các đánh giá và bình luận trên cửa hàng ứng dụng. Phân tích dữ liệu sử dụng để hiểu rõ hơn về hành vi và nhu cầu của người dùng. Dựa trên phản hồi và phân tích, liên tục cải tiến và tối ưu hóa ứng dụng để đáp ứng nhu cầu thay đổi của người dùng.

1.3 Ngôn ngữ lập trình Javascript

JavaScript là một ngôn ngữ lập trình đa năng, được phát triển bởi Brendan Eich tại Netscape vào năm 1995. JavaScript hiện nay là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới, được sử dụng rộng rãi không chỉ trong phát triển web mà còn trong phát triển di động, máy chủ (với Node.js), trò chơi và nhiều lĩnh vực khác.

Ưu điểm :

- Dễ hiểu, dễ học hơn so với các ngôn ngữ lập trình khác.
- Dễ dàng phát hiện và khắc phục, sửa chữa lỗi.
- Hoạt động được trên nhiều nền tảng trình duyệt.
- Hỗ trợ web tương tác hiệu quả với người truy cập.
- Tiết kiệm lưu lượng tải dữ liệu cho máy chủ (server) do người dùng sẽ xác nhận Input (đầu vào) rồi mới gửi web đến máy chủ.
- Ngôn ngữ lập trình JS cho phép tạo nhiều giao diện khác nhau.
- Phản hồi tức thời đến người truy cập.

Nhược điểm :

- Quản lý bộ nhớ không hiệu quả: JavaScript có một hệ thống thu gom rác tự động, nhưng quản lý bộ nhớ vẫn có thể trở thành một vấn đề đối với các ứng dụng lớn và phức tạp. Việc không giải phóng bộ nhớ một cách hiệu quả có thể dẫn đến rò rỉ bộ nhớ và hiệu suất kém.
- Khả năng xử lý không đồng bộ: JavaScript chủ yếu là một ngôn ngữ đơn luồng (single-threaded), điều này có nghĩa là nó chỉ có thể thực hiện một công việc tại một thời điểm. Điều này có thể tạo ra vấn đề với các tác vụ đòi hỏi xử lý không đồng bộ, như gọi API hoặc thực hiện các tác vụ I/O dài hạn, làm chậm quá trình thực thi.
- Bảo mật: JavaScript có thể gặp phải các vấn đề bảo mật như cross-site scripting (XSS) nếu không được viết mã cẩn thận và không tuân thủ các nguyên tắc bảo mật. Điều này có thể dẫn đến việc xâm nhập và đánh cắp thông tin người dùng.

1.4 Framework React Native

React Native là một framework phát triển ứng dụng di động đa nền tảng được phát triển bởi Facebook. Nó cho phép bạn sử dụng JavaScript và React để xây dựng các ứng dụng di động cho cả iOS và Android. Với React Native, bạn có thể viết mã một lần và sử dụng lại trên cả hai hệ điều hành này, giúp tiết kiệm thời gian và công sức phát triển.

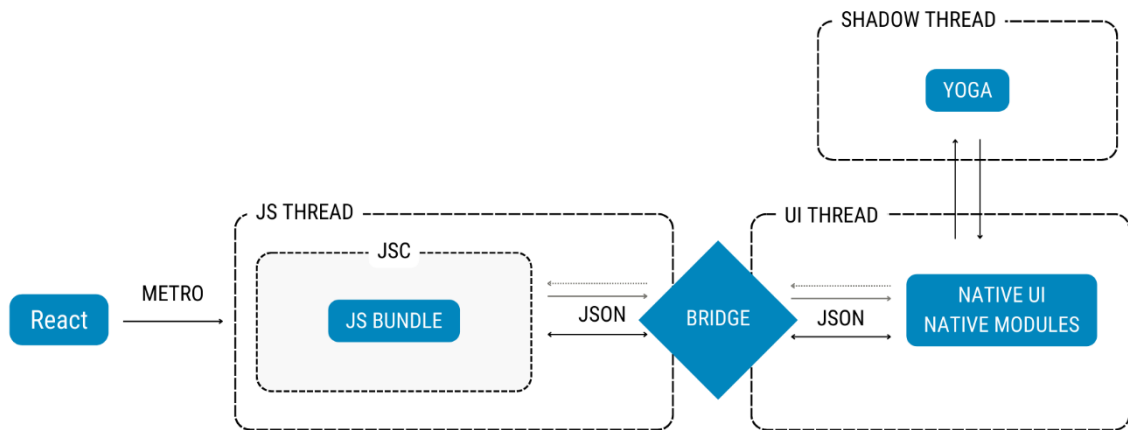
Ưu điểm :

- **Phát triển đa nền tảng :** Viết một lần, chạy trên cả iOS và Android. Điều này giúp giảm thiểu thời gian và công sức cần thiết cho việc phát triển ứng dụng di động.
- **Cộng đồng lớn và hỗ trợ mạnh mẽ :** React Native có một cộng đồng lớn và nhiều tài liệu hướng dẫn, thư viện và công cụ hỗ trợ, giúp việc phát triển trở nên dễ dàng hơn.
- **Tái sử dụng mã :** Bạn có thể tái sử dụng một phần lớn mã giữa các nền tảng khác nhau, giúp giảm thời gian và công sức cần thiết cho việc phát triển.

Nhược điểm :

- **Hiệu năng kém hơn so với Native App.**
- **Bảo mật chưa thật sự tốt do dùng JavaScript.**
- **Tùy biến chưa thật sự tốt ở một số module.** Không thích hợp cho các ứng dụng cần năng lực tính toán cao
- **Hạn chế về UI/UX:** Mặc dù React Native cung cấp các thành phần UI gốc của hệ điều hành, nhưng đôi khi bạn có thể gặp phải hạn chế về tùy chỉnh và thiết kế phức tạp.
- **Phụ thuộc vào bridge:** Tương tác giữa JavaScript và mã gốc thông qua bridge có thể tạo ra độ trễ trong một số trường hợp, đặc biệt là khi xử lý các tác vụ đòi hỏi hiệu suất cao

1.5 Mô hình kiến trúc của React Native



Hình 1.1 : Mô hình kiến trúc React Native

Mô hình kiến trúc của React Native gồm các thành phần chính :

- Native Thread / UI Thread : đây là luồng chính được khởi tạo ngay sau khi khởi chạy ứng dụng. Luồng này giao tiếp với Javascript Thread bất cứ khi nào có thay đổi về giao diện hoặc truy cập các hàm native.
- Javascript Thread: đây là nơi tất cả các code Javascript được đọc và biên dịch, là nơi xử lý hầu hết các logic nghiệp vụ của ứng dụng.
- Shadow Thread: là nơi mà React Native sẽ tính toán giao diện. Nó sử dụng Layout Engine riêng của Facebook có tên là Yoga nhằm tính toán flexbox layout, sau đó gửi kết quả về phía Native Thread .
- Bridge: Bridge là cơ chế cho phép tương tác giữa mã JavaScript và mã native của hệ điều hành. Khi mã JavaScript cần gọi các hàm hoặc phương thức native, nó sẽ thông qua bridge để gửi yêu cầu đến mã native và nhận kết quả trả về.

1.6 Giới thiệu về Firebase

Firebase là một nền tảng phát triển ứng dụng di động và web do Google cung cấp. Nó cung cấp một loạt các dịch vụ và công cụ cho việc xây dựng, triển khai và quản lý các ứng dụng, giúp các nhà phát triển tập trung vào việc xây

dụng trải nghiệm người dùng tốt nhất. Dưới đây là một số dịch vụ chính mà Firebase cung cấp:

- **Realtime Database:** Cơ sở dữ liệu thời gian thực cho phép bạn lưu trữ và đồng bộ dữ liệu giữa các client một cách tức thì.
- **Cloud Firestore:** Một cơ sở dữ liệu mạnh mẽ với tính năng truy vấn linh hoạt và đồng bộ dữ liệu thời gian thực.
- **Authentication:** Dịch vụ xác thực người dùng, hỗ trợ xác thực bằng email/password, số điện thoại, Google, Facebook, Twitter, vv.
- **Storage:** Lưu trữ đám mây cho các tệp tin, hình ảnh, video và dữ liệu của ứng dụng.
- **Cloud Functions:** Cho phép bạn viết các hàm backend dựa trên sự kiện mà không cần phải quản lý máy chủ.
- **Hosting:** Dịch vụ hosting web tĩnh và động, cho phép triển khai và quản lý trang web của bạn một cách dễ dàng.
- **Analytics:** Cung cấp thông tin phân tích về người dùng và hoạt động của ứng dụng để bạn có thể hiểu rõ hơn về cách người dùng sử dụng ứng dụng của mình.
- **Crashlytics:** Cung cấp thông tin về các lỗi và sự cố trong ứng dụng của bạn để bạn có thể sửa chữa chúng nhanh chóng.
- **Cloud Messaging:** Cho phép bạn gửi thông báo đến người dùng trên các nền tảng khác nhau (iOS, Android, web).
- **Remote Config:** Cho phép bạn cấu hình và tùy chỉnh ứng dụng của mình từ xa mà không cần phải cập nhật ứng dụng.

1.7 Giới thiệu công cụ Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở và miễn phí được phát triển bởi Microsoft. Nó được thiết kế để làm việc với nhiều

ngôn ngữ lập trình và các công nghệ phát triển phổ biến. Dưới đây là một số điểm nổi bật của Visual Studio Code:

- Đa nền tảng: Visual Studio Code có sẵn trên Windows, macOS và Linux, giúp cho các nhà phát triển có thể làm việc trên nhiều hệ điều hành một cách thuận tiện.
- Mở rộng và Tùy chỉnh: VS Code có một hệ thống mở rộng mạnh mẽ, cho phép bạn cài đặt các extension để mở rộng chức năng của nó và tùy chỉnh theo nhu cầu cá nhân.
- Hỗ trợ ngôn ngữ đa dạng: VS Code cung cấp hỗ trợ cho nhiều ngôn ngữ lập trình và công nghệ, bao gồm JavaScript, TypeScript, Python, Java, C++, HTML, CSS, và nhiều ngôn ngữ khác.
- Debugging tích hợp: Nó cung cấp các công cụ debugging tích hợp cho nhiều ngôn ngữ, giúp bạn dễ dàng theo dõi và sửa lỗi trong mã nguồn của mình.
- Hỗ trợ Git tích hợp: VS Code tích hợp sâu vào hệ thống kiểm soát phiên bản Git, cung cấp các tính năng như xem và so sánh thay đổi, commit, push, pull và nhiều tính năng khác.
- Live Share: Tính năng Live Share cho phép bạn chia sẻ màn hình và làm việc cùng nhau trực tuyến với các thành viên trong đội của bạn.
- IntelliSense: VS Code cung cấp IntelliSense, một tính năng tự động hoàn thành mã và gợi ý cú pháp, giúp tăng năng suất khi viết mã.
- Terminal tích hợp: Trình quản lý Terminal tích hợp cho phép bạn chạy các lệnh hệ thống và các công cụ dòng lệnh trực tiếp từ VS Code.

1.8 Tóm tắt chương

Chương này đã giới thiệu về tổng quan của ứng dụng, mô hình, các công nghệ và công cụ để ứng dụng vào phát triển ứng dụng đọc truyện. Những kiến thức chương này chính là cơ sở cho các chương sau.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ ỨNG DỤNG HỖ TRỢ ĐỌC TRUYỆN

2.1 Giới thiệu ứng dụng đọc truyện

2.1.1 Mô tả ứng dụng

Ứng dụng đọc truyện sử dụng React Native là ứng dụng được xây dựng hỗ trợ việc đọc truyện, tìm truyện mà qua đó những tác giả có tài năng viết xuất sắc thể đến được với độc giả nhờ các bộ truyện được đưa lên mạng, các trao đổi tương tác giữa các độc giả với nhau.

Ứng dụng cung cấp cho độc giả những bộ truyện hay có khả năng hỗ trợ thêm khả năng giao tiếp, nhạy bén trong nhận thức, phát triển trí tuệ cảm xúc, độc giả có thể giải quyết vấn đề sáng tạo, đồng cảm, dễ tiếp thu, linh hoạt ứng dụng, tích cực, phán đoán hợp lý.

2.1.2 Các chức năng chính

Ứng dụng được xây dựng cần đáp ứng đầy đủ các chức năng như sau :

Người dùng :

- Đăng kí / Đăng nhập: Cho phép độc giả đăng kí tài khoản mới và đăng nhập vào ứng dụng.
- Xem danh sách truyện: cho phép độc giả có thể xem tất cả truyện theo danh mục.
- Tìm truyện: Cho phép độc giả tìm kiếm tên truyện mà mình mong muốn.
- Xem chi tiết truyện: Cho phép độc giả đọc toàn bộ nội dung truyện.
- Bình luận: Cho phép độc giả đưa ra ý kiến của mình về truyện.
- Yêu thích: Cho phép độc giả yêu thích truyện.

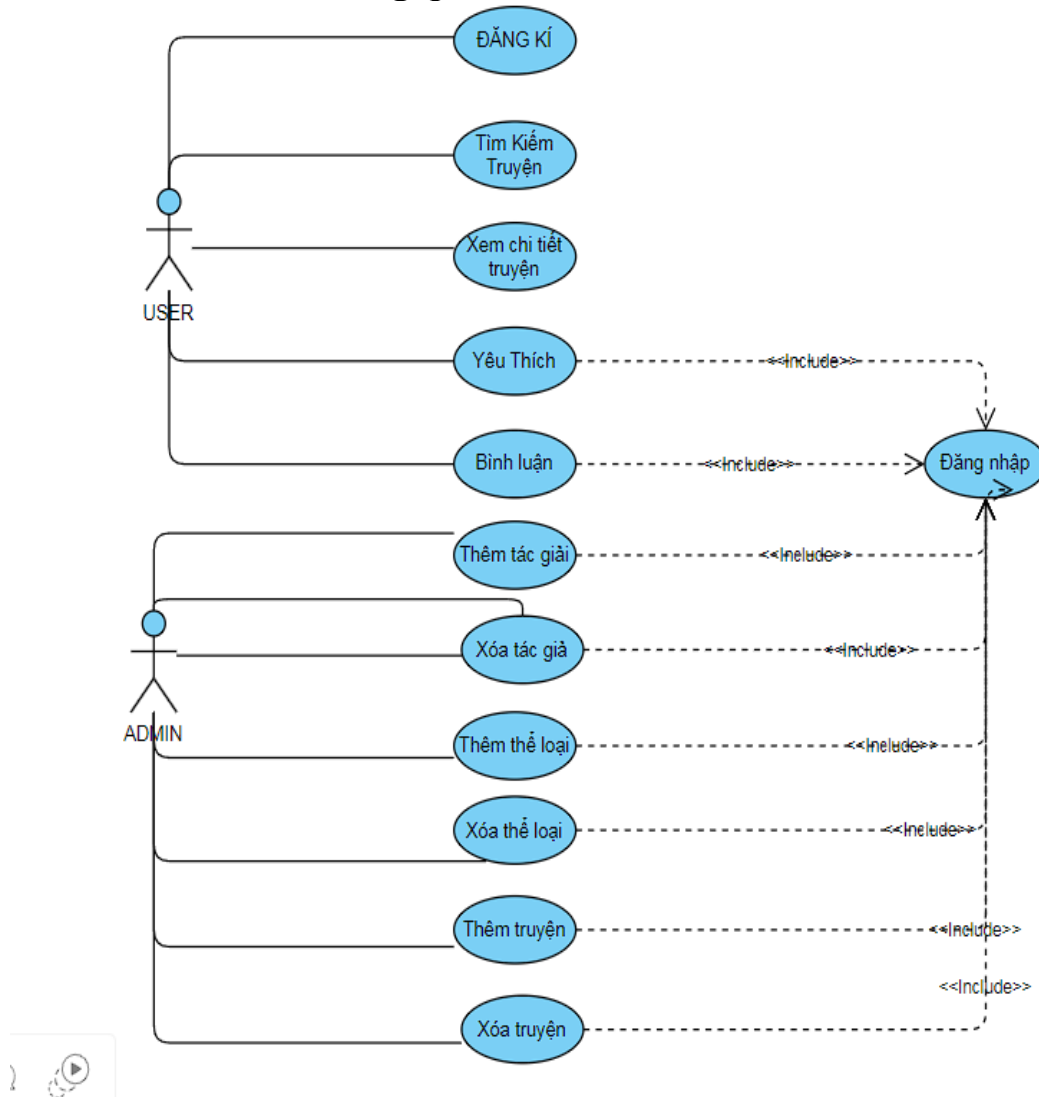
Quản Trị Viên

- Thêm,sửa, xóa tác giả: người quản trị có thể cập nhật tác giả.
- Thêm sửa xóa thể loại: người quản trị có thể cập nhật thể loại.
- Thêm sửa xóa truyện: người quản trị có thể cập nhật truyện.

2.1.3 Các yêu cầu phi chức năng

- Giao diện thân thiện với độc giả và dễ dàng sử dụng.
- Ứng dụng có thể chạy trên nhiều hệ điều hành và các thiết bị khác nhau.
- Ứng dụng hoạt động ổn định và thông báo lỗi nếu có.
- An toàn, bảo mật: Đảm bảo an toàn thông tin độc giả.

2.2 Biểu đồ Usecase tổng quát



Hình 2.1: Biểu đồ UC tổng quát

Biểu đồ usecase tổng quát mô tả về hệ thống ứng dụng có 2 tác nhân: người dùng và quản trị viên. Người dùng và quản trị viên phải đăng nhập để có thể trải nghiệm các chức năng còn lại.

2.3 Mô tả chi tiết các usecase

2.3.1 Mô tả UC “Đăng nhập”

- Mô tả vắn tắt: UC này cho phép người dùng đăng nhập vào ứng dụng để sử dụng các chức năng còn lại.
- Luồng sự kiện
 - Luồng cơ bản
 - 1) Usecase bắt đầu khi người dùng chọn vào phần đăng nhập của hệ thống. Hệ thống sẽ trả về form đăng nhập.
 - 2) Người dùng điền tài khoản và mật khẩu của mình vào và chọn đăng nhập. Hệ thống sẽ kiểm tra tính hợp lệ và hiển thị màn hình chính của ứng dụng.
 - Luồng rẽ nhánh
 - 1) Tại bước 2 của luồng cơ bản, nếu người dùng nhập sai toàn khoản hoặc mật khẩu hệ thống sẽ bắt người dùng nhập lại.
 - 2) Tại bất kì thời điểm nào trong quá trình thực hiện usecase , nếu không kết nối được CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và usecase kết thúc.

2.3.2 Mô tả UC “Đăng kí”

- Mô tả vắn tắt: UC này cho phép người dùng đăng kí tài khoản mới.
- Luồng sự kiện
 - Luồng cơ bản
 - 1) Usecase bắt đầu khi người dùng chọn vào phần đăng kí của hệ thống. Hệ thống sẽ trả về một form đăng kí.
 - 2) Người dùng điền thông tin tài khoản và mật khẩu của mình vào và chọn đăng kí. Hệ thống sẽ tạo tài khoản và lưu trong hệ thống.
 - Luồng rẽ nhánh :

- 1) Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được với CSDL thì hệ thống một bảng báo lỗi và kết thúc.

2.3.3 Mô tả UC “Tìm kiếm truyện”

- Mô tả vắn tắt :UC này cho phép người dùng thao tác tìm kiếm truyện.
- Luồng sự kiện :
 - Luồng cơ bản :
 - 1) Usecase bắt đầu khi người dùng ấn vào ô tìm kiếm.Hệ thống sẽ hiển thị ra ô tìm kiếm cho người dùng nhập thông tin.
 - 2) Người dùng điền nội dung cần tìm kiếm.Hệ thống sẽ lấy danh sách các truyện phù hợp với yêu cầu của người dùng.
 - Luồng rẽ nhánh:
 - 1) Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không kết nối được CSDL thì hệ thống hiển thị một bảng báo lỗi và kết thúc.
- Tiền điều kiện: Người dùng đã đăng nhập trước đó.

2.3.4 Mô tả UC “Xem chi tiết truyện”

- Mô tả vắn tắt:UC này cho phép người dùng xem chi tiết truyện.
- Luồng sự kiện
 - Luồng cơ bản
 - 1) Usecase bắt đầu khi người dùng ấn vào truyện muốn đọc. Hệ thống sẽ hiển thị ra chi tiết truyện cho người dùng.
 - Luồng rẽ nhánh
 - 1) Tại bất kì thời điểm nào trong quá trình thực hiện usecase , nếu không có kết nối được CSDL thì hệ thống hiển thị một bảng báo lỗi và kết thúc.

2.3.5 Mô tả UC “Yêu thích”

- Mô tả vắn tắt: UC này cho phép người dùng muốn thêm truyện đang đọc vào mục yêu thích.
- Luồng sự kiện :
 - Luồng cơ bản :
 - 1) Usecase bắt đầu khi người dùng tích chọn vào yêu thích truyện. Hệ thống sẽ lưu truyện yêu thích vào cơ sở dữ liệu.
 - Luồng rẽ nhánh
 - 1) Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không có kết nối được CSDL thì usecase kết thúc.
- Tiên điều kiện :

Người dùng đã xem chi tiết truyện.

2.3.6 Mô tả UC “ Bình luận”

- Mô tả vắn tắt: UC này cho phép người dùng bình luận về truyện đang đọc.
- Luồng sự kiện :
 - Luồng cơ bản :
 - 1) Usecase bắt đầu khi người dùng bấm vào ô bình luận. Hệ thống sẽ hiển thị ra ô bình luận để người dùng nhập bình luận.
 - 2) Người dùng nhập bình luận và ấn gửi. Hệ thống cập nhật bình luận vào cơ sở dữ liệu.
 - Luồng rẽ nhánh :
 - 1) Tại bất kì thời điểm nào trong quá trình thực hiện usecase , nếu không có kết nối với CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và kết thúc.

- Tiền điều kiện :Người dùng phải đăng nhập trước đó.

2.3.7 Mô tả UC “ Thêm tác giả “

- Mô tả vắn tắt: Uc này cho phép quản trị viên thêm tác giả.
- Luồng sự kiện :
 - Luồng cơ bản :
 - 1) Usecase bắt đầu khi người quản trị bấm vào phần thông tin tác giả.Hệ thống sẽ hiển thị các tác giả đã có.
 - 2) Quản trị viên thao tác thêm tác giả.Hệ thống cập nhật thông tin mới của tác giả sau khi thêm.
 - Luồng rẽ nhánh :
 - 1) Tại bất kì thời điểm nào trong quá trình thực hiện usecase, nếu không có kết nối CSDL thì hệ thống sẽ hiển thị một bảng báo lỗi và kết thúc.
- Tiền điều kiện : Quản trị viên phải đăng nhập trước đó.

2.3.8 Mô tả UC ”Xóa tác giả”

- Mô tả vắn tắt : UC này cho phép quản trị viên xóa tác giả.
- Luồng sự kiện :
 - Luồng cơ bản :
 - 1) Usecase bắt đầu khi người quản trị bấm vào phần tác giả. Hệ thống sẽ hiển thị ra danh sách tác giả đang có.
 - 2) Quản trị viên ấn thao tác xóa.Hệ thống sẽ cập nhật dữ liệu và hiển thị ra danh sách tác giả mới.
 - Luồng rẽ nhánh :
 - 1) Tại bất kì thời nào điểm trong quá trình thực hiện usecase, nếu không thể kết nối tới CSDL, hệ thống sẽ hiển thị ra màn

hình báo lỗi và usecase kết thúc.

- Tiên điều kiện: Quản trị viên phải đăng nhập trước đó.

2.3.9 Mô tả UC “Thêm thẻ loại”

- Mô tả vắn tắt: UC này cho phép quản trị viên thêm thẻ loại mới.
- Luồng sự kiện :
 - Luồng cơ bản :
 - 1) Usecase bắt đầu khi người quản trị bấm vào phần thẻ loại. Hệ thống sẽ hiển thị ra danh sách thẻ loại đang có.
 - 2) Quản trị viên ấn thao tác thêm. Hệ thống sẽ hiển thị form nhập thẻ loại , quản trị viên nhập thẻ loại mới. Hệ thống sẽ lưu và hiển thị ra danh sách thẻ loại mới.
 - Luồng rẽ nhánh :
 - 1) Tại bất kì thời nào điểm trong quá trình thực hiện usecase, nếu không thể kết nối tới CSDL, hệ thống sẽ hiển thị ra màn hình báo lỗi và usecase kết thúc.
- Tiên điều kiện : Quản trị viên phải đăng nhập trước đó.

2.3.10 Mô tả UC “Thêm truyện”

- Mô tả vắn tắt: UC này cho phép quản trị viên thêm truyện mới.
- Luồng sự kiện :
 - Luồng cơ bản :
 - 1) Usecase bắt đầu khi người quản trị bấm vào phần truyện. Hệ thống sẽ hiển thị ra danh sách truyện đang có.
 - 2) Quản trị viên ấn thao tác thêm. Hệ thống sẽ cập nhật dữ liệu và hiển thị ra danh sách truyện mới.
 - Luồng rẽ nhánh :

- 1) Tại bất kì thời nào điểm trong quá trình thực hiện usecase, nếu không thể kết nối tới CSDL, hệ thống sẽ hiển thị ra màn hình báo lỗi và usecase kết thúc.

– Tiên điều kiện : Quản trị viên phải đăng nhập trước đó.

2.3.11 Mô tả UC “ Xóa truyện”

– Mô tả vắn tắt: UC này cho phép quản trị viên xóa truyện.

– Luồng sự kiện :

○ Luồng cơ bản :

- 1) Usecase bắt đầu khi người quản trị bấm vào phần truyện. Hệ thống sẽ hiển thị ra danh sách truyện đang có.
- 2) Quản trị viên ấn thao tác xóa. Hệ thống sẽ cập nhật dữ liệu và hiển thị ra danh sách truyện mới.

○ Luồng rẽ nhánh :

- 1) Tại bất kì thời nào điểm trong quá trình thực hiện usecase, nếu không thể kết nối tới CSDL, hệ thống sẽ hiển thị ra màn hình báo lỗi và usecase kết thúc.

– Tiên điều kiện : Quản trị viên phải đăng nhập trước đó.

2.3.12 Mô tả UC “Xóa thể loại”

– Mô tả vắn tắt: UC này cho phép quản trị viên xóa thể loại.

– Luồng sự kiện :

○ Luồng cơ bản :

- 1) Usecase bắt đầu khi người quản trị bấm vào phần thể loại. Hệ thống sẽ hiển thị ra danh sách thể loại đang có.
- 2) Quản trị viên ấn thao tác xóa. Hệ thống sẽ cập nhật dữ liệu và hiển thị ra danh sách thể loại mới.

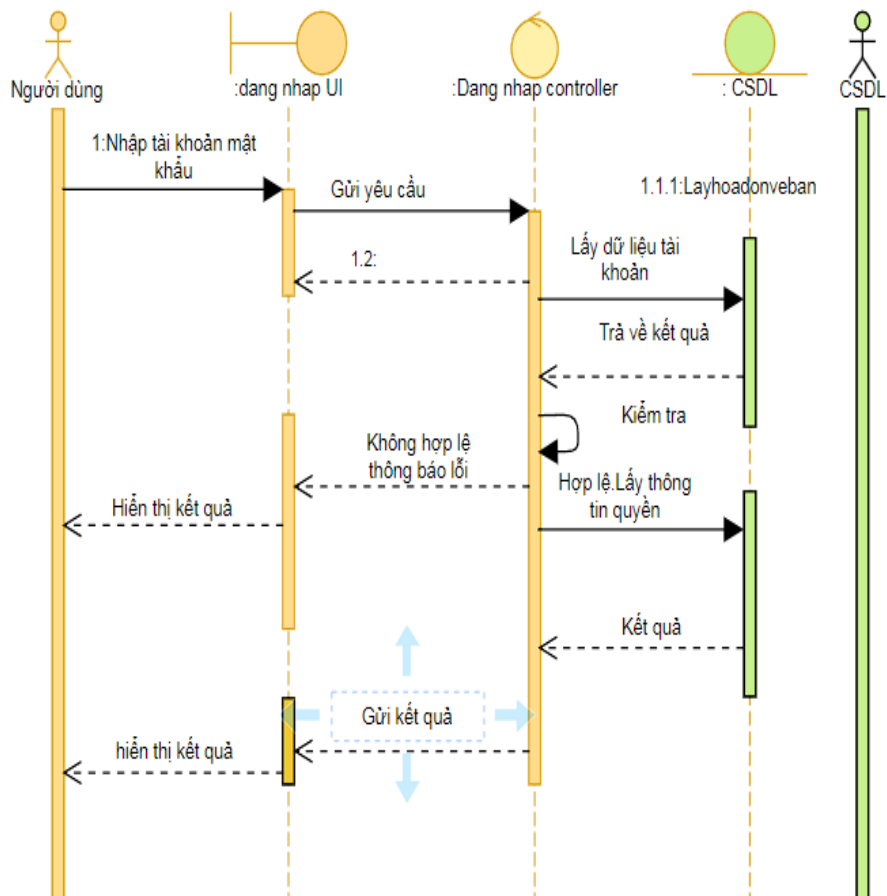
○ Luồng rẽ nhánh :

- 1) Tại bất kì thời nào điểm trong quá trình thực hiện usecase, nếu không thể kết nối tới CSDL, hệ thống sẽ hiển thị ra màn hình báo lỗi và usecase kết thúc.

– Tiên điều kiện : Quản trị viên phải đăng nhập trước đó.

2.4 Biểu đồ trình tự

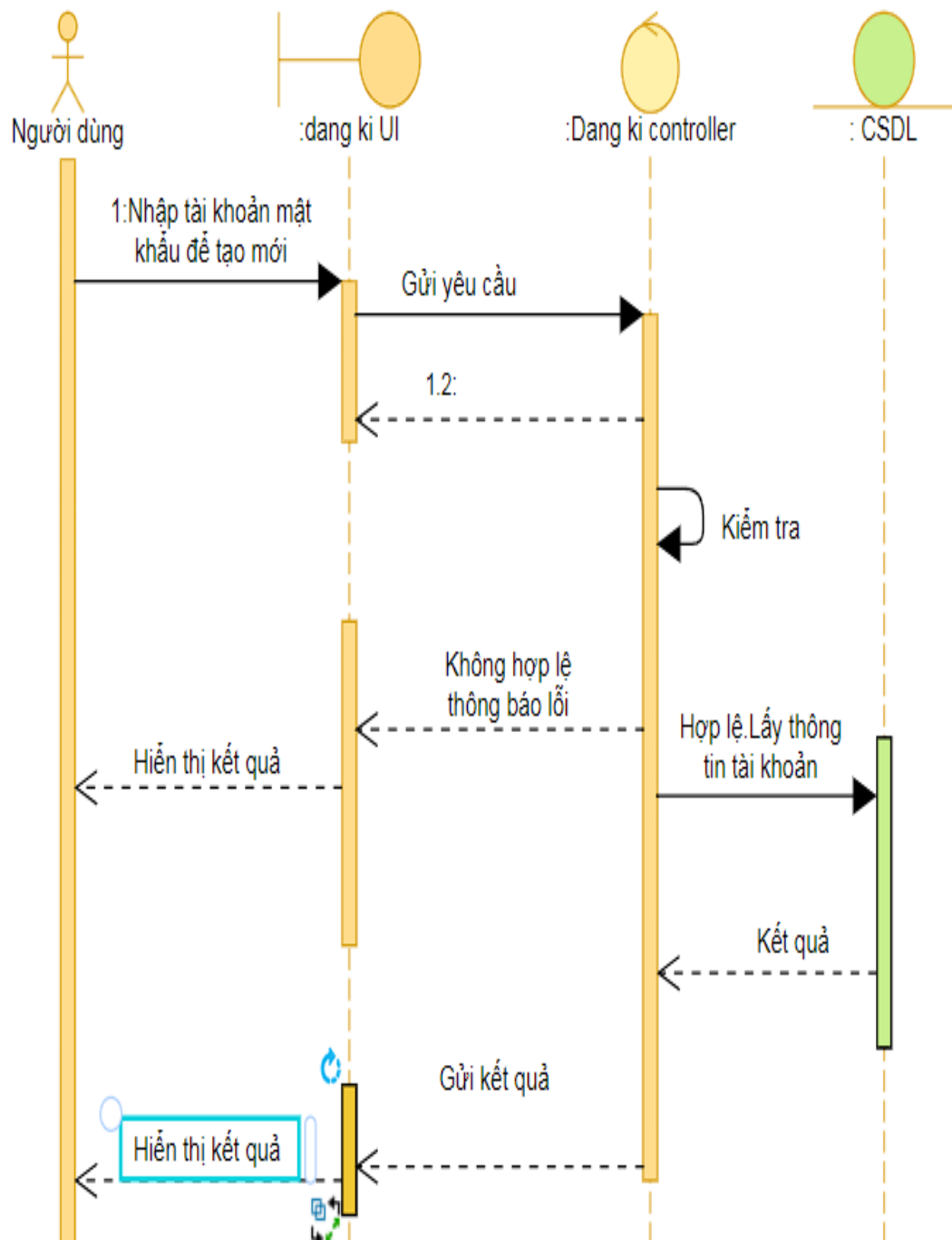
2.4.1 Biểu đồ trình tự UC “Đăng nhập”



Hình 2.2 : Biểu đồ trình tự UC đăng nhập

Người dùng đăng nhập vào ứng dụng qua màn hình đăng nhập, thông tin sẽ được gửi lên hệ thống và so sánh với dữ liệu đã lưu, sau đó kết quả được trả về và hiển thị giao diện. Nếu thông tin không hợp, hệ thống thông báo cho người dùng nhập lại.

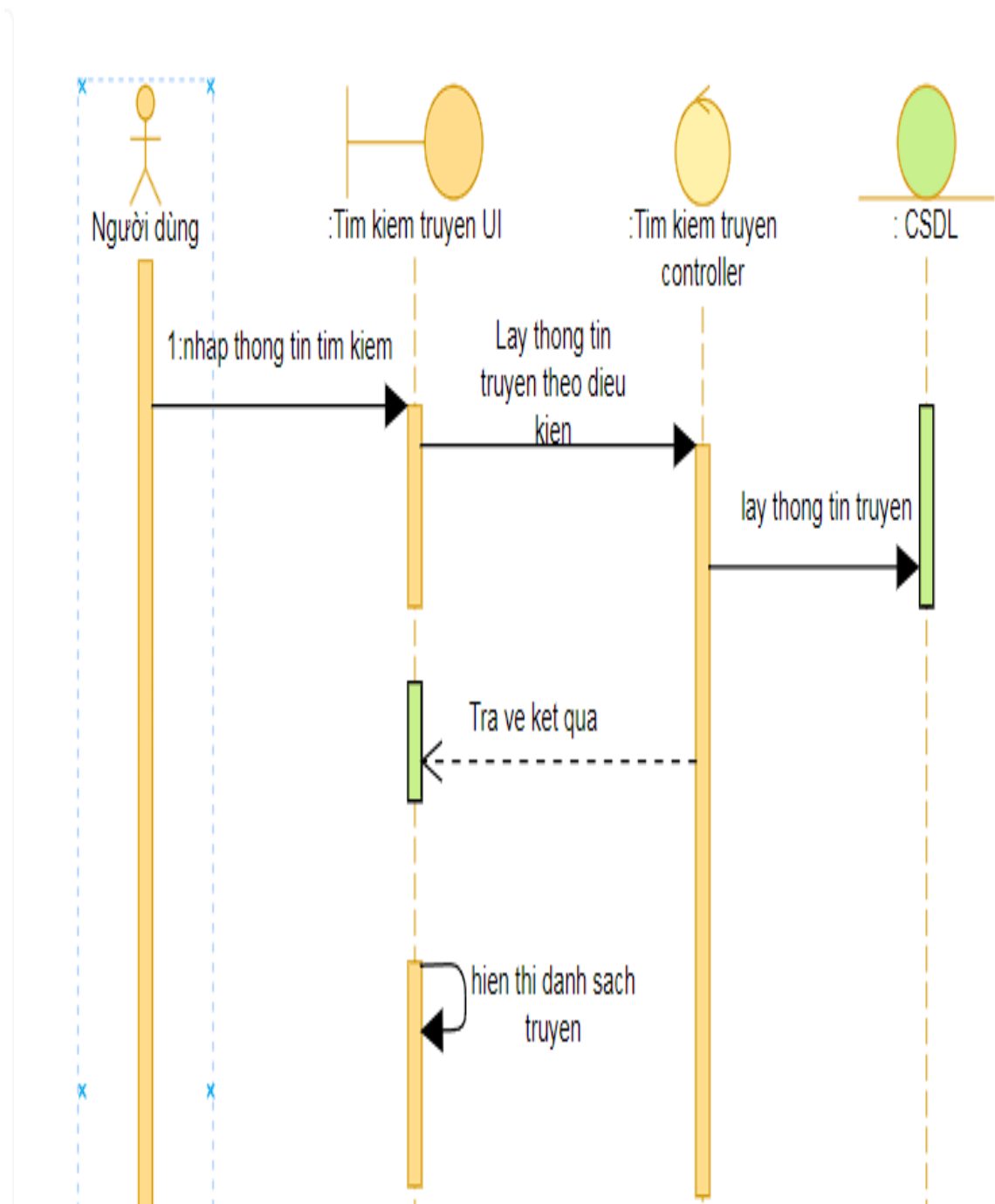
2.4.2 Biểu đồ trình tự UC “Đăng kí”



Hình 2.3 : Biểu đồ trình tự UC đăng kí

Người dùng đăng nhập vào ứng dụng qua màn hình đăng nhập, thông tin sẽ được gửi lên hệ thống và so sánh với dữ liệu đã lưu, sau đó kết quả được trả về và hiển thị giao diện. Nếu thông tin không hợp lệ hoặc không đúng, hệ thống thông báo cho người dùng nhập lại.

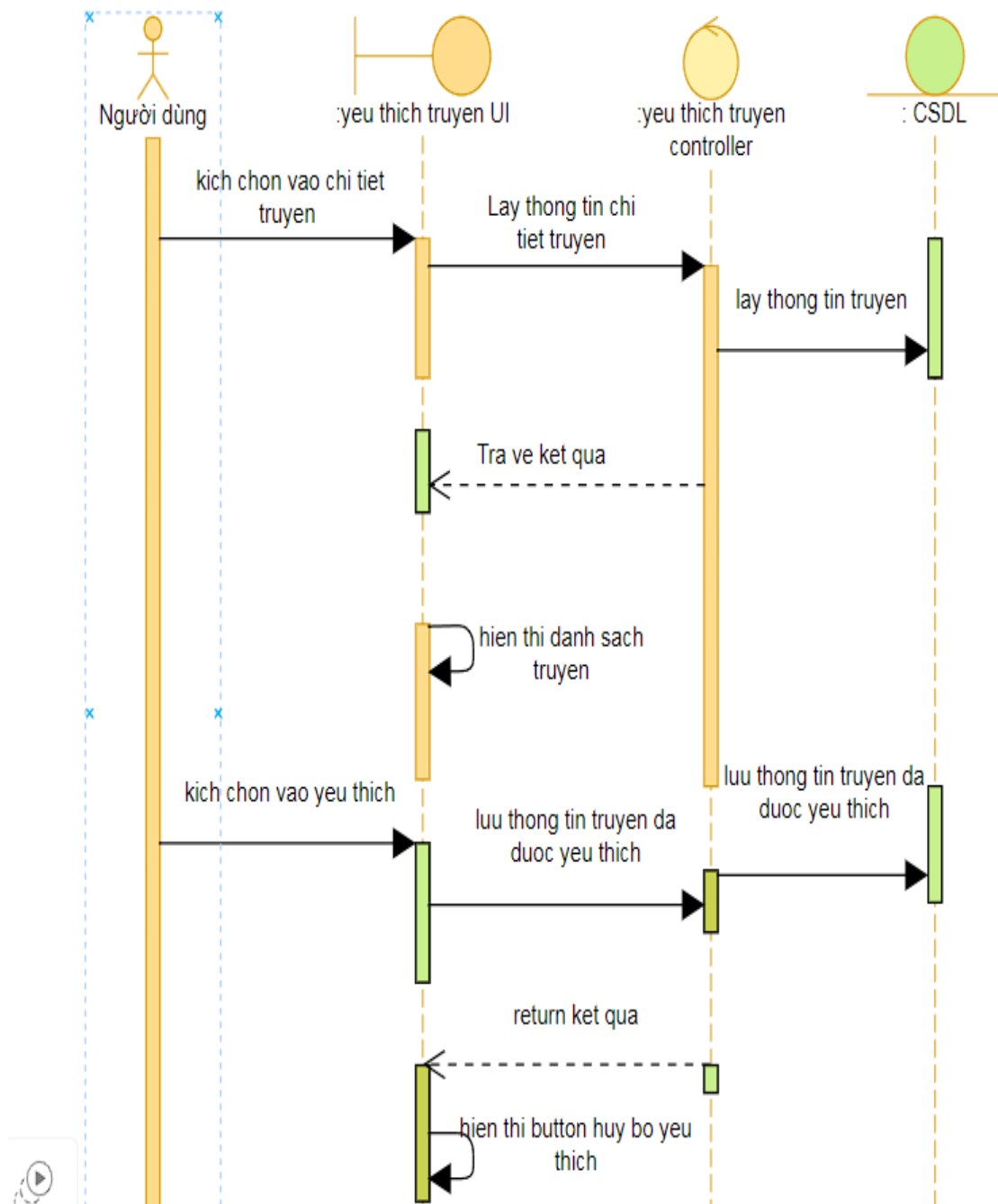
2.4.3 Biểu đồ trình tự UC “Tìm kiếm truyện”



Hình 2.4 : Biểu đồ trình tự UC tìm kiếm truyện

Khi muốn tìm kiếm một truyện bất kì người dùng nhấn vào thanh tìm kiếm truyện. Người dùng nhập tên truyện muốn tìm kiếm. Hệ thống sẽ lấy thông tin truyện và hiển thị ra màn hình cho người dùng. Nếu không thể tìm thấy truyện thì hệ thống báo không có dữ liệu.

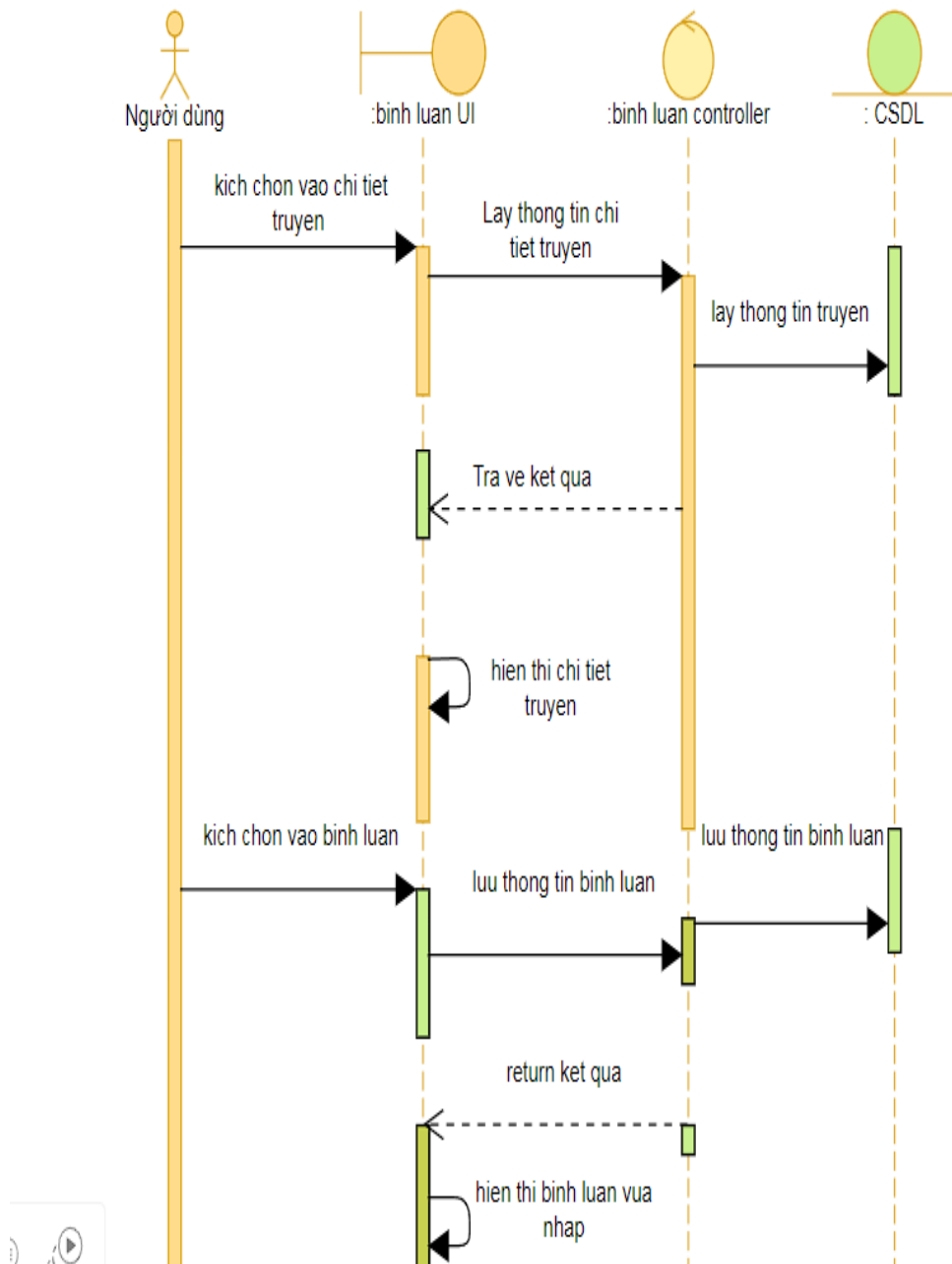
2.4.4 Biểu đồ trình tự UC “Yêu thích truyện”



Hình 2.5: Biểu đồ trình tự UC yêu thích

Người dùng chọn vào xem chi tiết truyện. Hệ thống sẽ hiển thị chi tiết truyện như tên truyện, tác giả, thể loại, nút yêu thích, nội dung truyện, bình luận. Người dùng nhấn vào nút yêu thích, hệ thống sẽ lưu lại truyện được yêu thích đồng thời nút yêu thích sẽ hiển thị thành nút hủy bỏ.

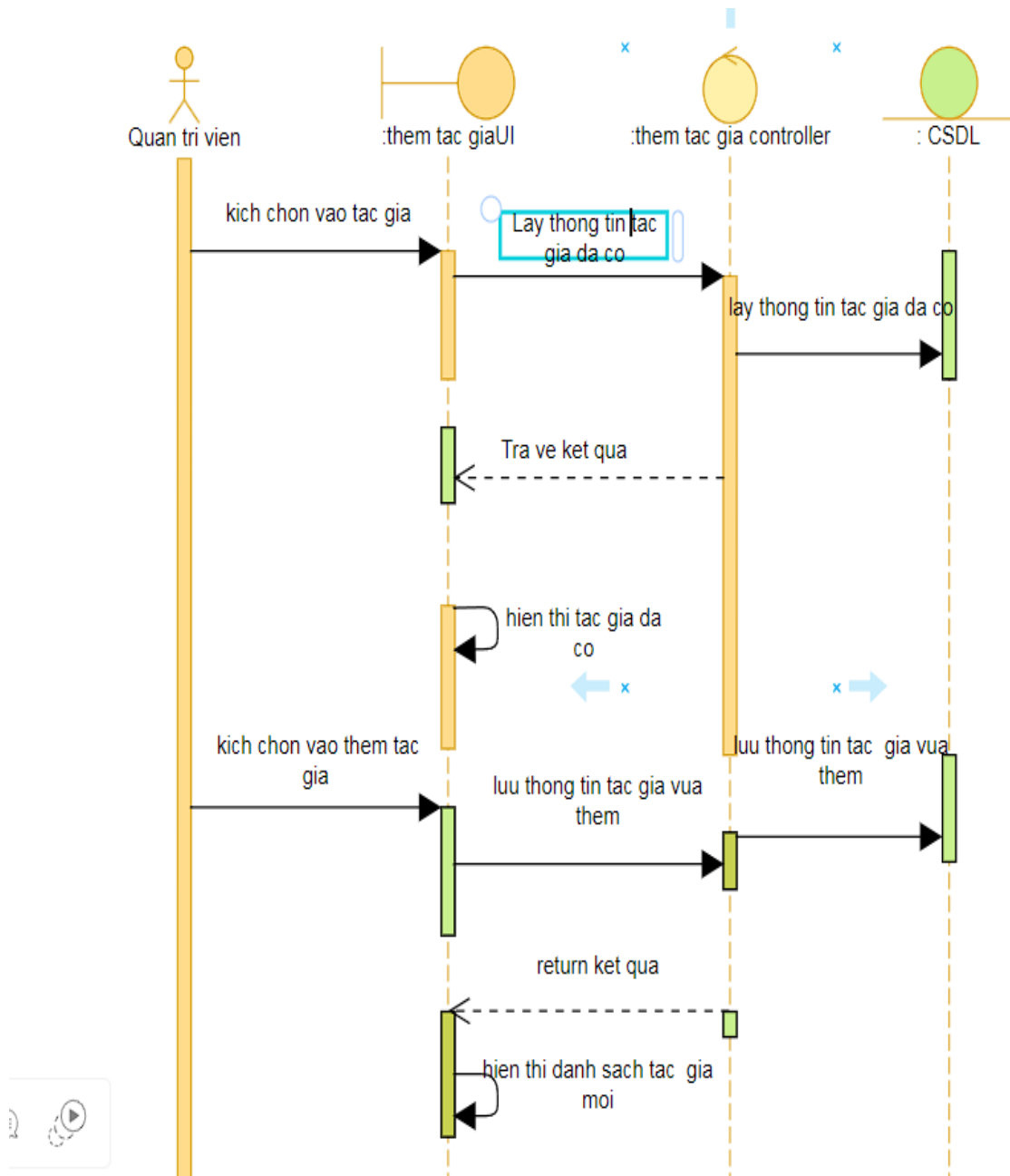
2.4.5 Biểu đồ trình tự UC“Bình luận”



Hình 2.6: Biểu đồ trình tự UC Bình Luận

Khi ở phần giao diện của chi tiết truyện nếu người muốn bình luận thì kéo xuống phần bình luận, người dùng nhập bình luận vào ô bình luận, sau đó ấn gửi bình luận. Hệ thống sẽ hiển thị lại bình luận vừa nhập.

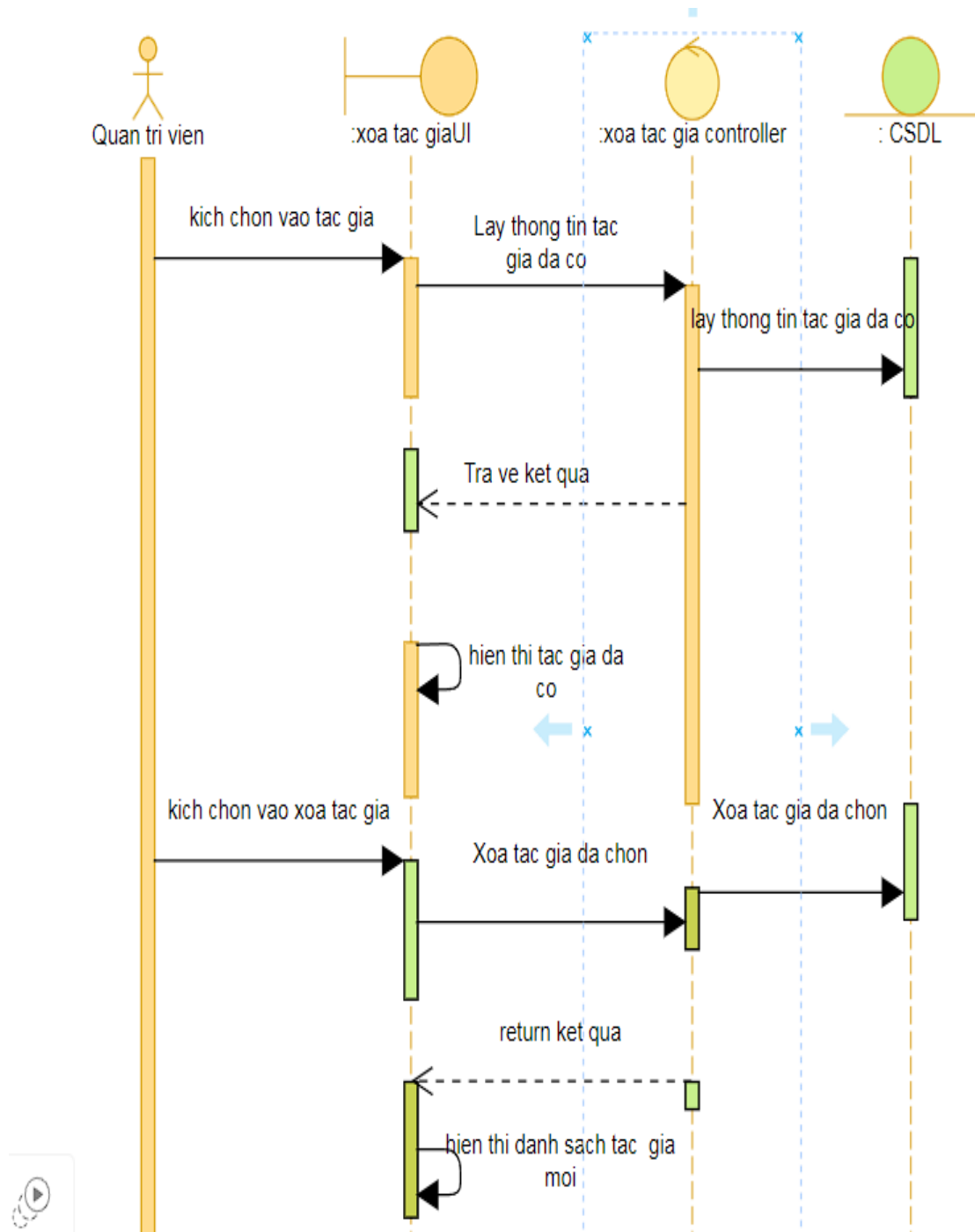
2.4.6 Biểu đồ trình tự UC “Thêm tác giả”



Hình 2.7 Biểu đồ trình tự UC Thêm Tác Giả

Quản trị viên chọn vào phần quản lý tác giả, hệ thống sẽ hiển thị ra các tác giả đã có. Quản trị viên nhấn vào nút thêm, hệ thống sẽ hiển thị ra form nhập tên tác giả, quản trị viên nhập tên tác giả, ấn vào thêm mới, hệ thống sẽ lưu dữ liệu và hiển thị ra danh sách tác giả mới.

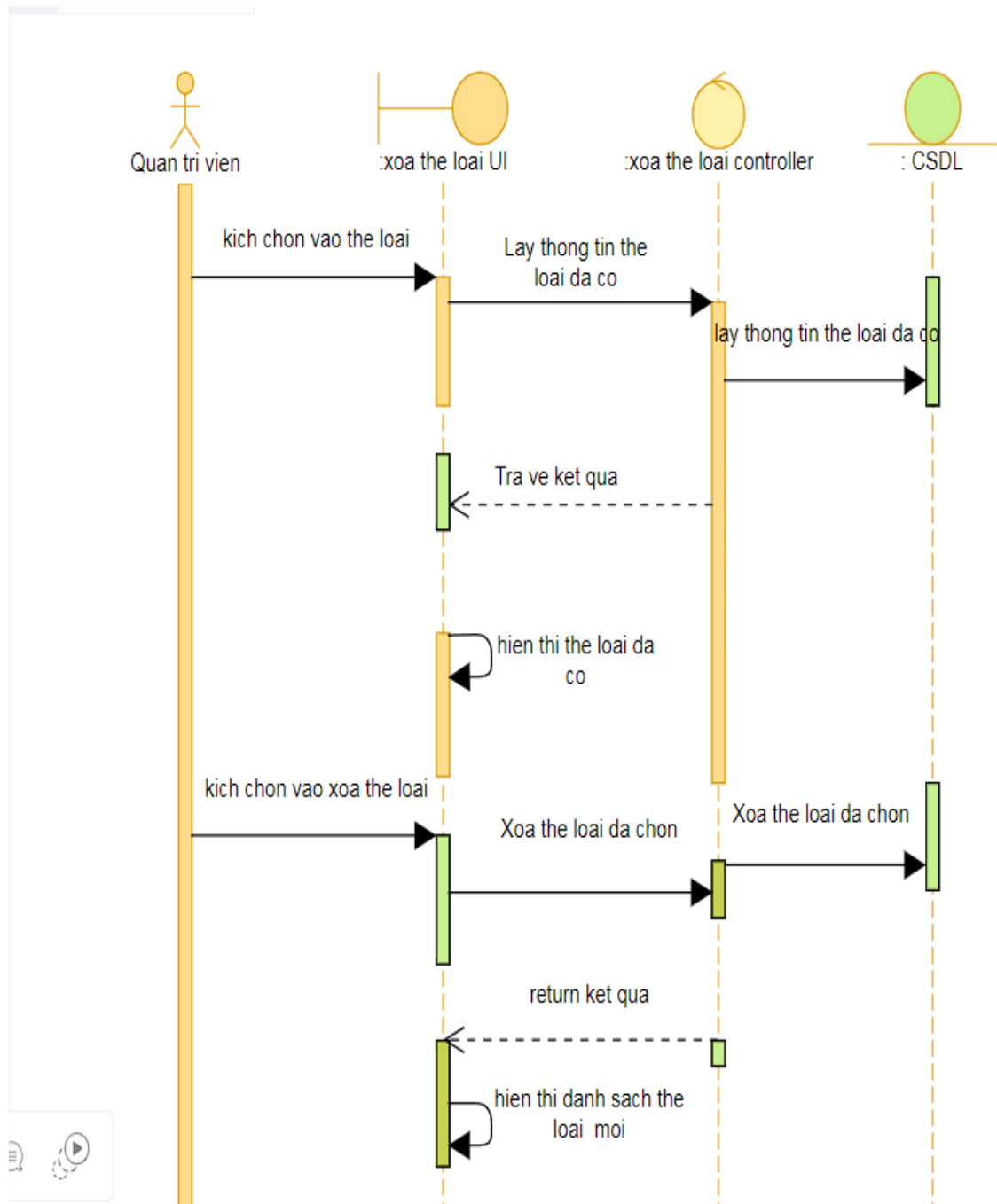
2.4.7 Biểu đồ trình tự UC “Xóa tác giả”



Hình 2.8 Biểu đồ trình tự UC xóa tác giả

Khi quản trị viên muốn xóa tác giả, quản trị viên chọn vào phần tác giả, hệ thống sẽ hiển thị ra các tác giả đã có. Quản trị viên nhấn vào nút xóa, hệ thống xóa tác giả đã chọn, hệ thống sẽ lưu dữ liệu và hiển thị ra danh sách tác giả mới sau xóa.

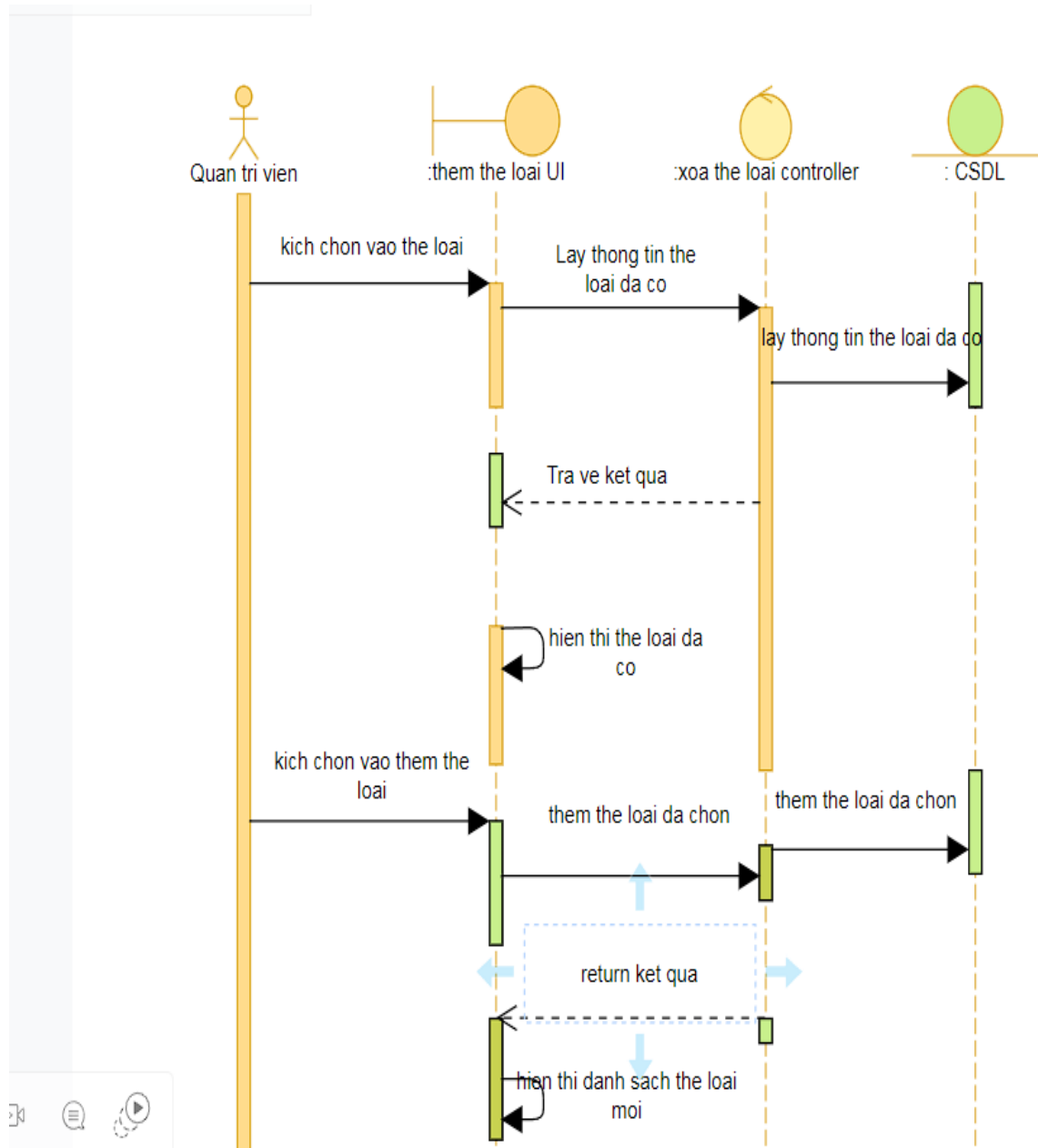
2.4.8 Biểu đồ trình tự UC “Xóa thẻ loại”



Hình 2.9 Biểu đồ trình tự UC xóa thẻ loại

Khi quản trị viên muốn xóa thẻ loại, quản trị viên chọn vào phần quản lý thẻ loại, hệ thống sẽ hiển thị ra các thẻ loại đã có. Quản trị viên nhấn vào nút xóa, hệ thống xóa thẻ loại đã chọn, hệ thống sẽ lưu dữ liệu và hiển thị ra danh sách thẻ loại mới sau xóa.

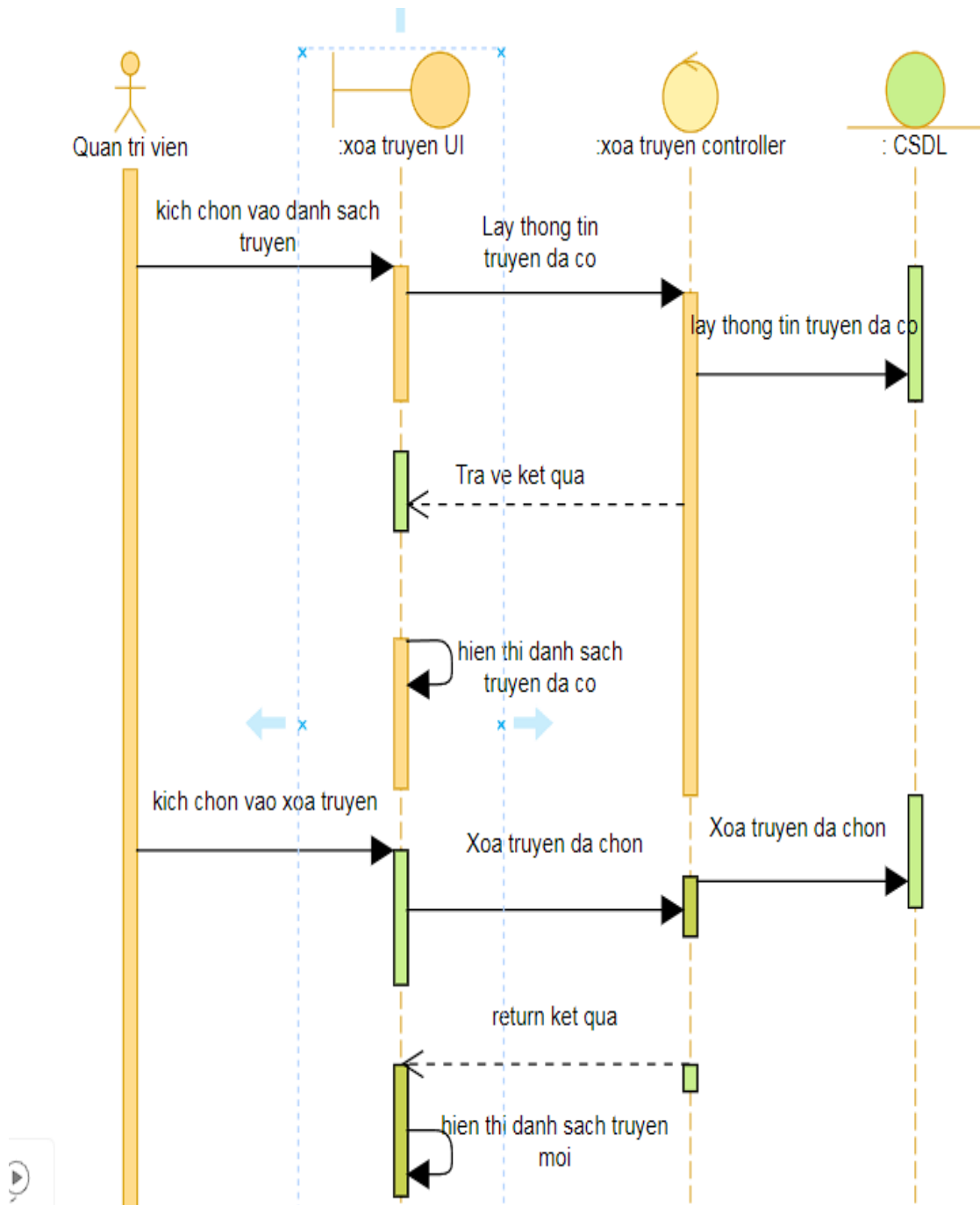
2.4.9 Biểu đồ trình tự UC “Thêm thẻ loại”



Hình 2.10 Biểu đồ trình tự UC Thêm thẻ loại

Khi quản trị viên muốn thêm thẻ loại, quản trị viên chọn vào phần quản lý thẻ loại, hệ thống sẽ hiển thị ra các thẻ loại đã có. Quản trị viên nhấn vào nút thêm, hệ thống sẽ hiển thị ra form nhập tên thẻ loại, quản trị viên nhập tên thẻ loại, ấn vào thêm mới, hệ thống sẽ lưu dữ liệu và hiển thị ra danh sách thẻ loại mới.

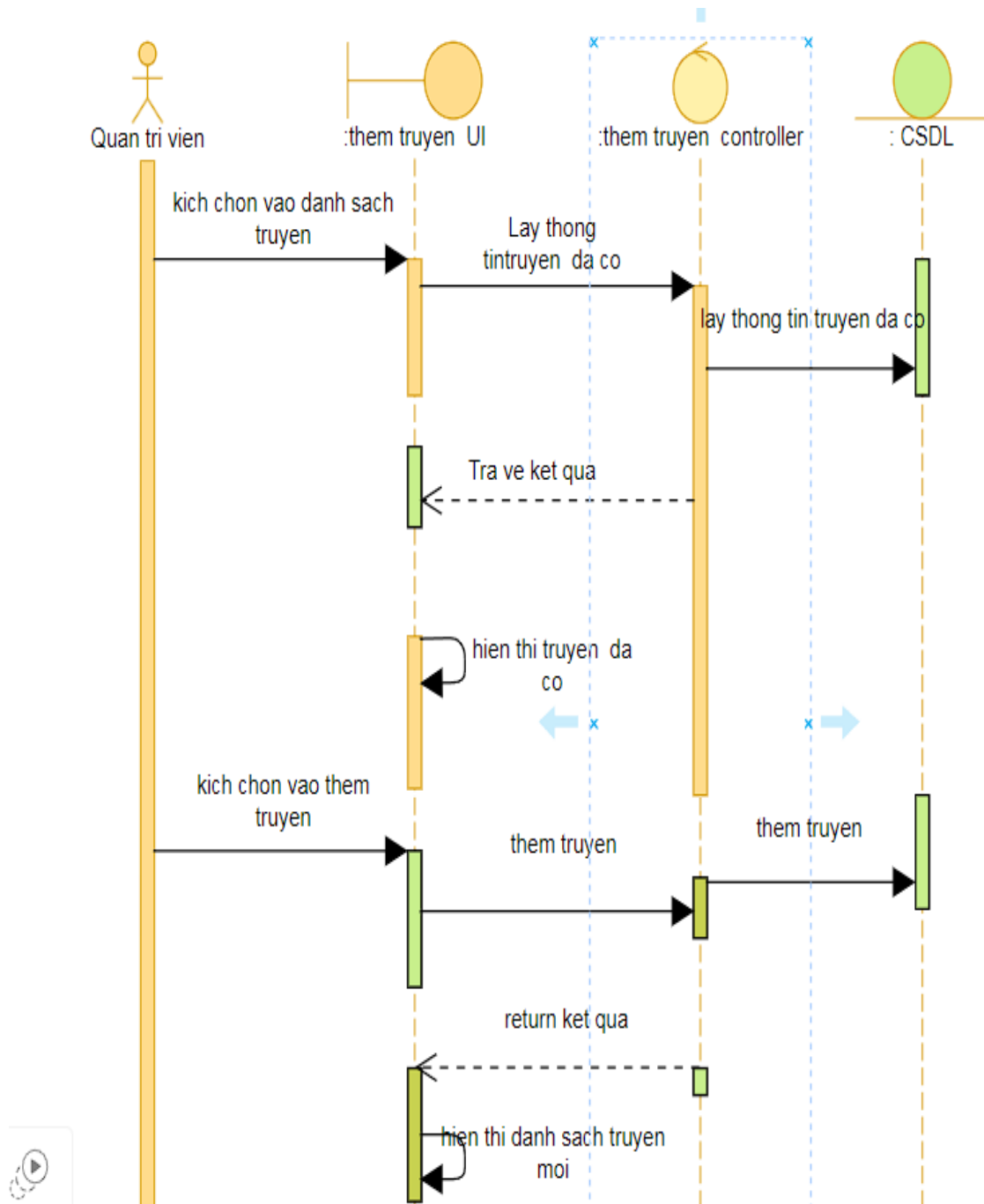
2.4.10 Biểu đồ trình tự UC “Xóa truyện”



Hình 2.11: Biểu đồ trình tự UC xóa truyện

Khi quản trị viên muốn xóa truyện, quản trị viên chọn vào phần quản lý truyện, hệ thống sẽ hiển thị ra các truyện đã có. Quản trị viên nhấn vào nút xóa, hệ thống xóa truyện đã chọn, hệ thống sẽ lưu dữ liệu và hiển thị ra danh sách truyện mới sau xóa.

2.4.11 Biểu đồ UC “Thêm truyện”

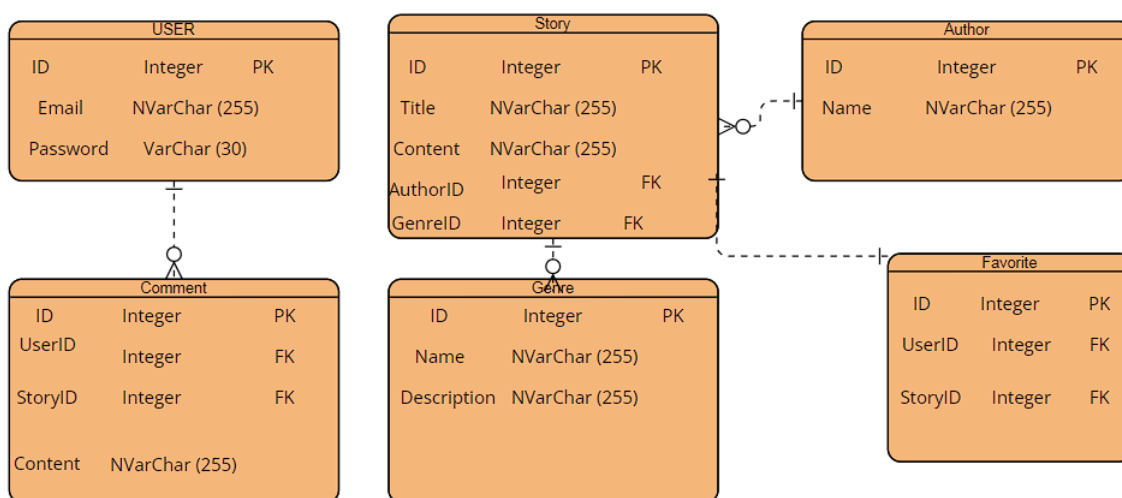


Hình 2.12 Biểu đồ trình tự UC thêm truyện

Khi quản trị viên muốn thêm truyện, quản trị viên chọn vào phần quản lý truyện, hệ thống sẽ hiển thị ra các truyện đã có. Quản trị viên nhấn vào nút thêm, hệ thống sẽ hiển thị ra form nhập, quản trị viên nhập tên truyện, ảnh truyện, chọn tác giả, chọn thể loại. Hệ thống sẽ lưu dữ liệu và hiển thị ra danh sách truyện mới.

2.5 Thiết kế cơ sở dữ liệu

2.5.1 Biểu đồ thực thể liên kết



Hình 2.13 : Biểu đồ thực thể liên kết

2.5.2 Chi tiết các bảng

Bảng 1: Chi tiết bảng user

STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	ID	Integer	Primary Key	Mã người dùng
2	Email	Nvarchar(255)	No	Địa chỉ email
3	Password	Varchar(30)	No	Mật khẩu

Bảng 2. Chi tiết bảng Story

STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	Integer	Primary key	Mã truyện
2	Title	NVarchar(255)	No	Tên truyện
3	Description	NVarchar(255)	No	Mô tả

4	AuthorID	Interger	No	Mã tác giả
5	GenreID	Interger	No	Mã thể loại

Bảng 3. Chi tiết bảng Author

STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	Integer	Primary key	Mã truyện
2	Name	NVarchar(255)	No	Tên tác giả

Bảng 4. Chi tiết bảng Favorite

STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	Integer	Primary key	Mã yêu thích
2	UserID	Integer	Foreign Key	Mã người dùng
3	StoryID	Integer	Foreign Key	Mã truyện

Bảng 5. Chi tiết bảng Comment

STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	Integer	Primary key	Mã bình luận
2	UserID	Integer	FK	Mã người dùng
3	StoryID	Integer	FK	Mã Truyện

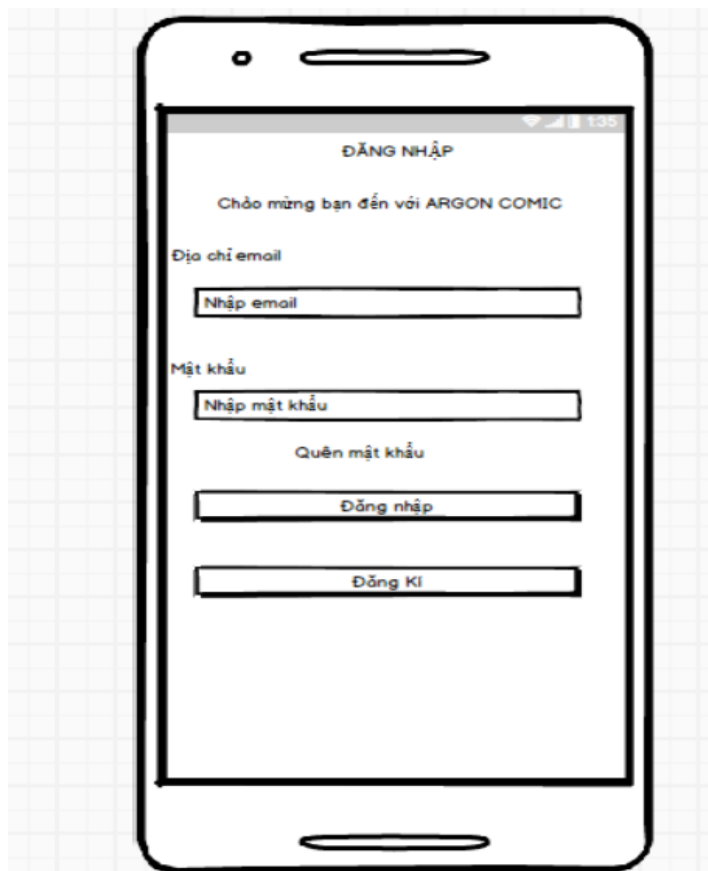
4	Content	Nvarchar(255)	No	Nội dung
---	---------	---------------	----	----------

Bảng 6. Chi tiết bảng Genre

STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	Integer	Primary key	Mã thể loại
2	Name	Nvarchar(255)	No	Tên thể loại
3	Description	Nvarchar(255)	No	Mô tả

2.6 Thiết kế giao diện

2.6.1. Giao diện đăng nhập



Hình 2.14 : Giao diện màn hình đăng nhập

2.6.2 Giao diện đăng kí

ĐĂNG KÍ

Địa chỉ email

Nhập email

Số điện thoại

Số điện thoại

Họ Và Tên

Nhập họ và tên

Mật khẩu

Nhập mật khẩu

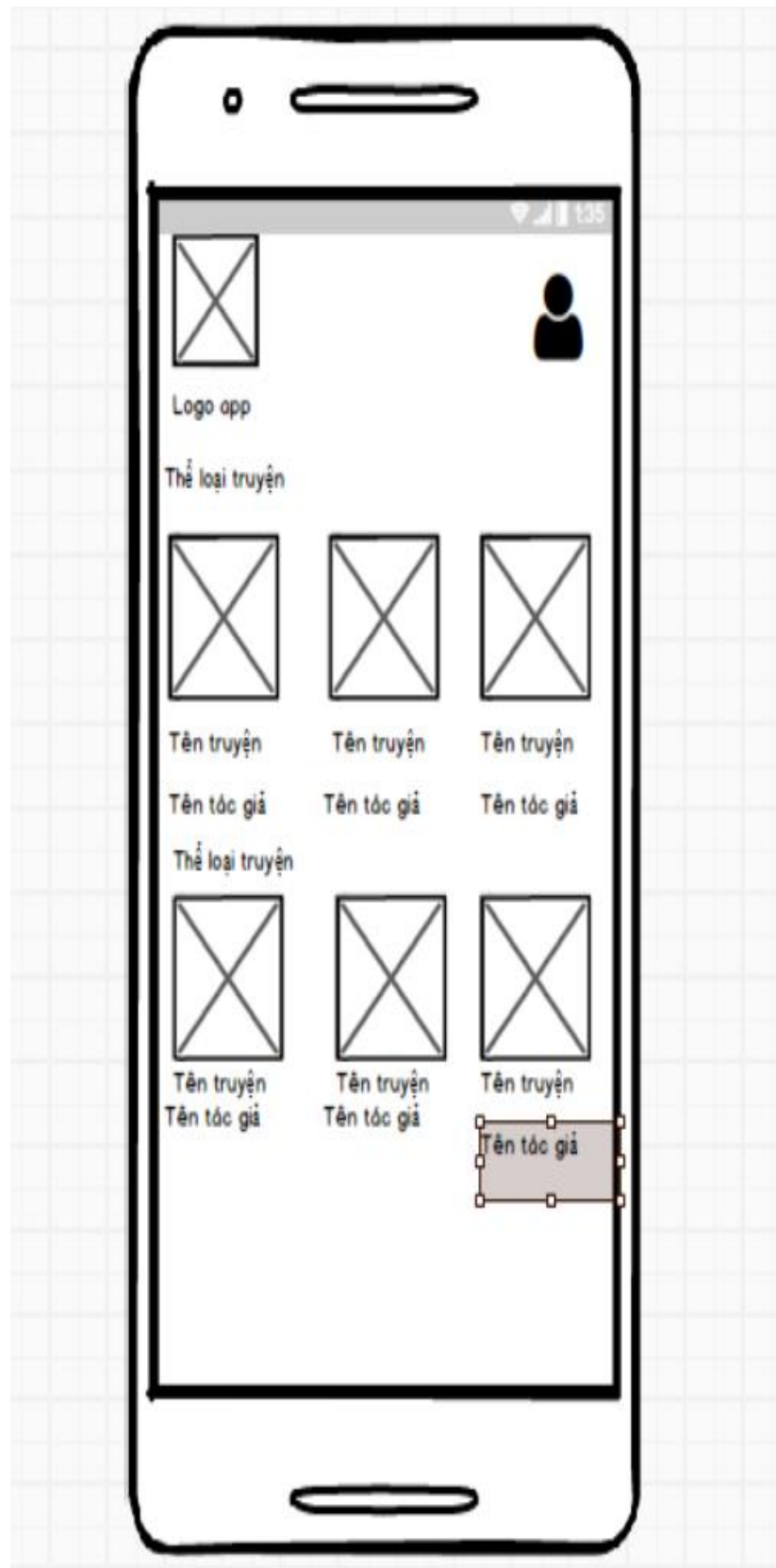
Xác nhận Mật khẩu

Nhập lại mật khẩu

Đăng Kí Ngay

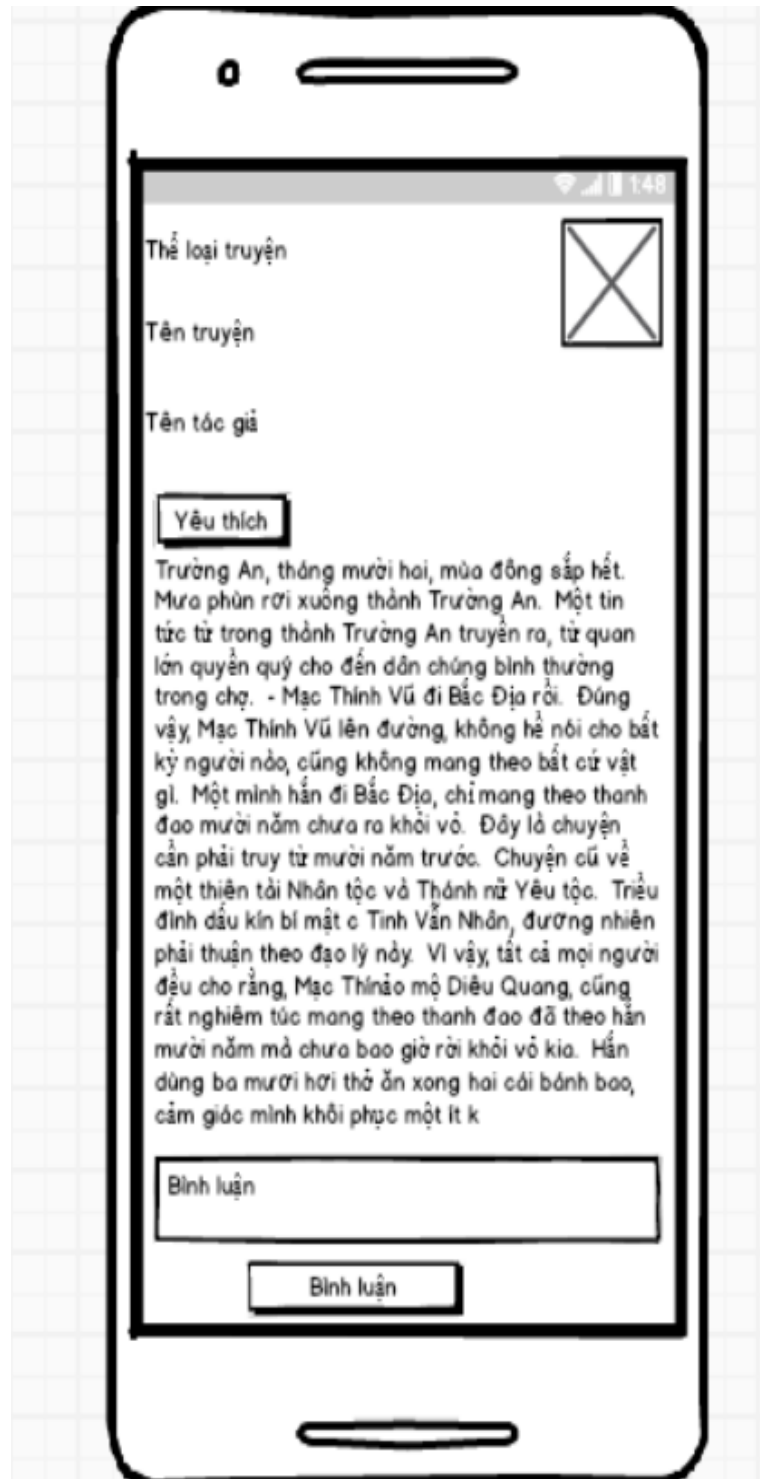
Hình 2.15: Giao diện màn hình đăng kí

2.6.3 Giao diện trang chủ



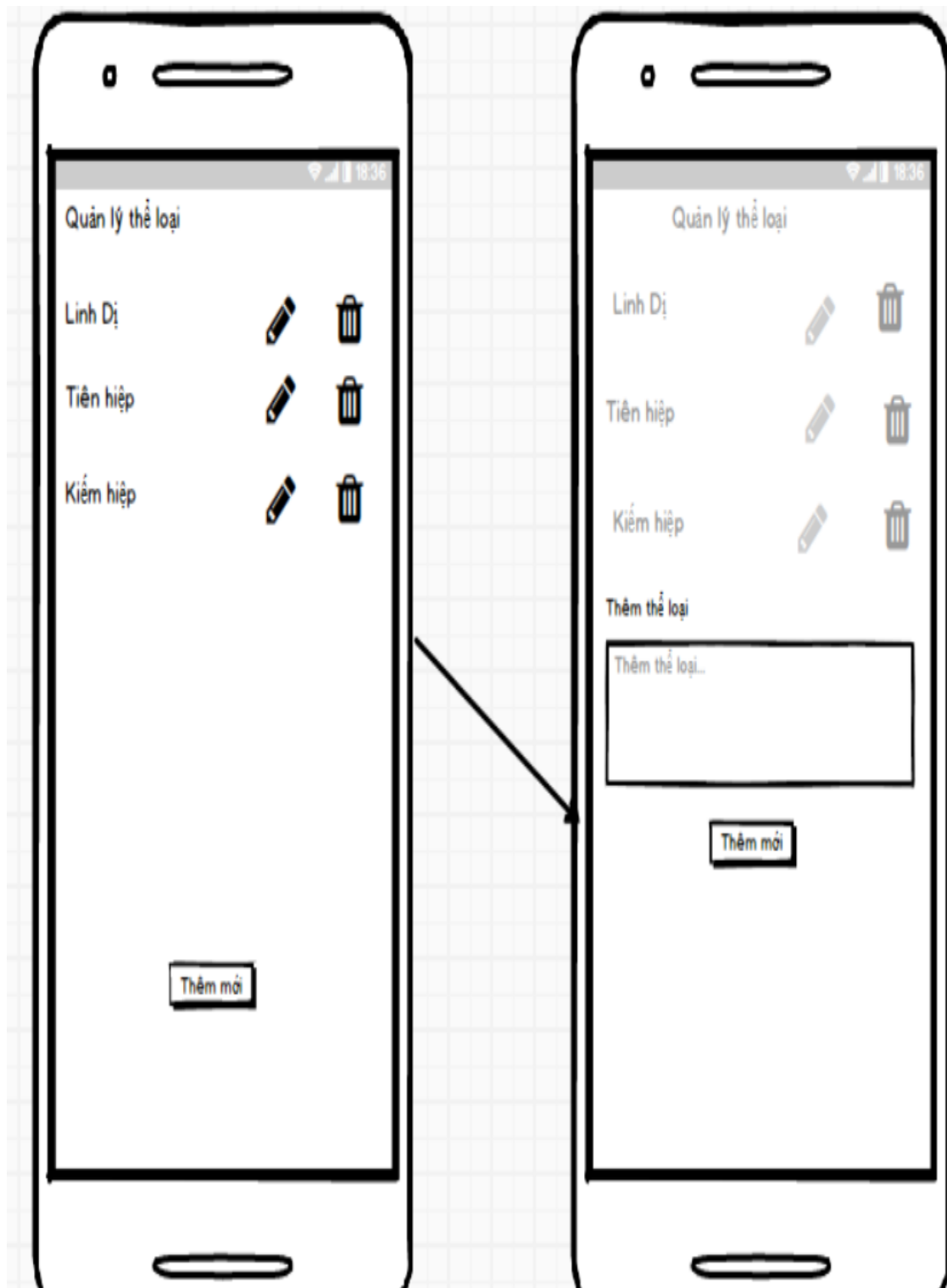
Hình 2.16 : Giao diện trang chủ

2.6.4 Giao diện Xem chi tiết truyện



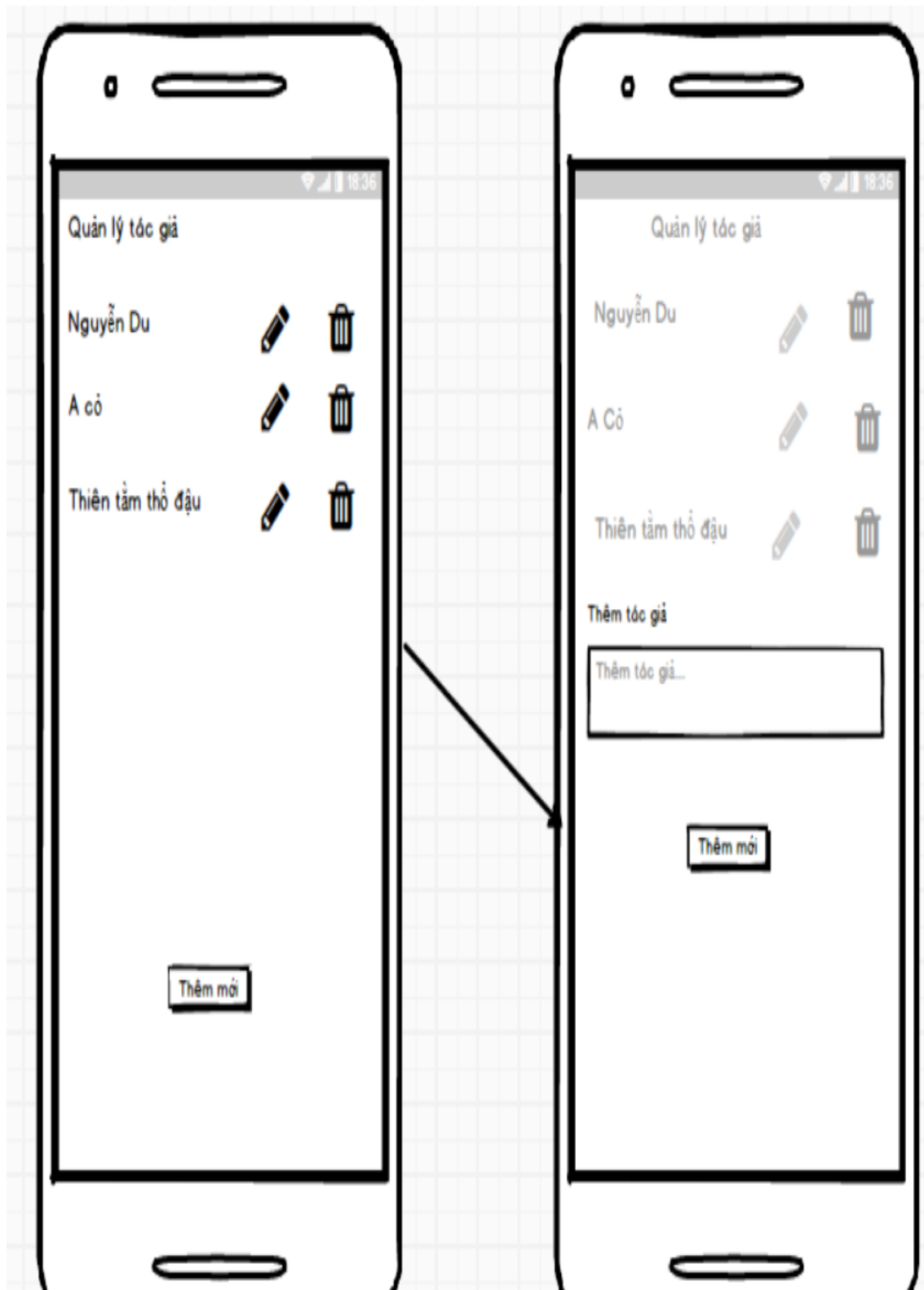
Hình 2.17: Giao diện xem chi tiết truyện

2.6.5 Giao diện Thêm mới thẻ loại



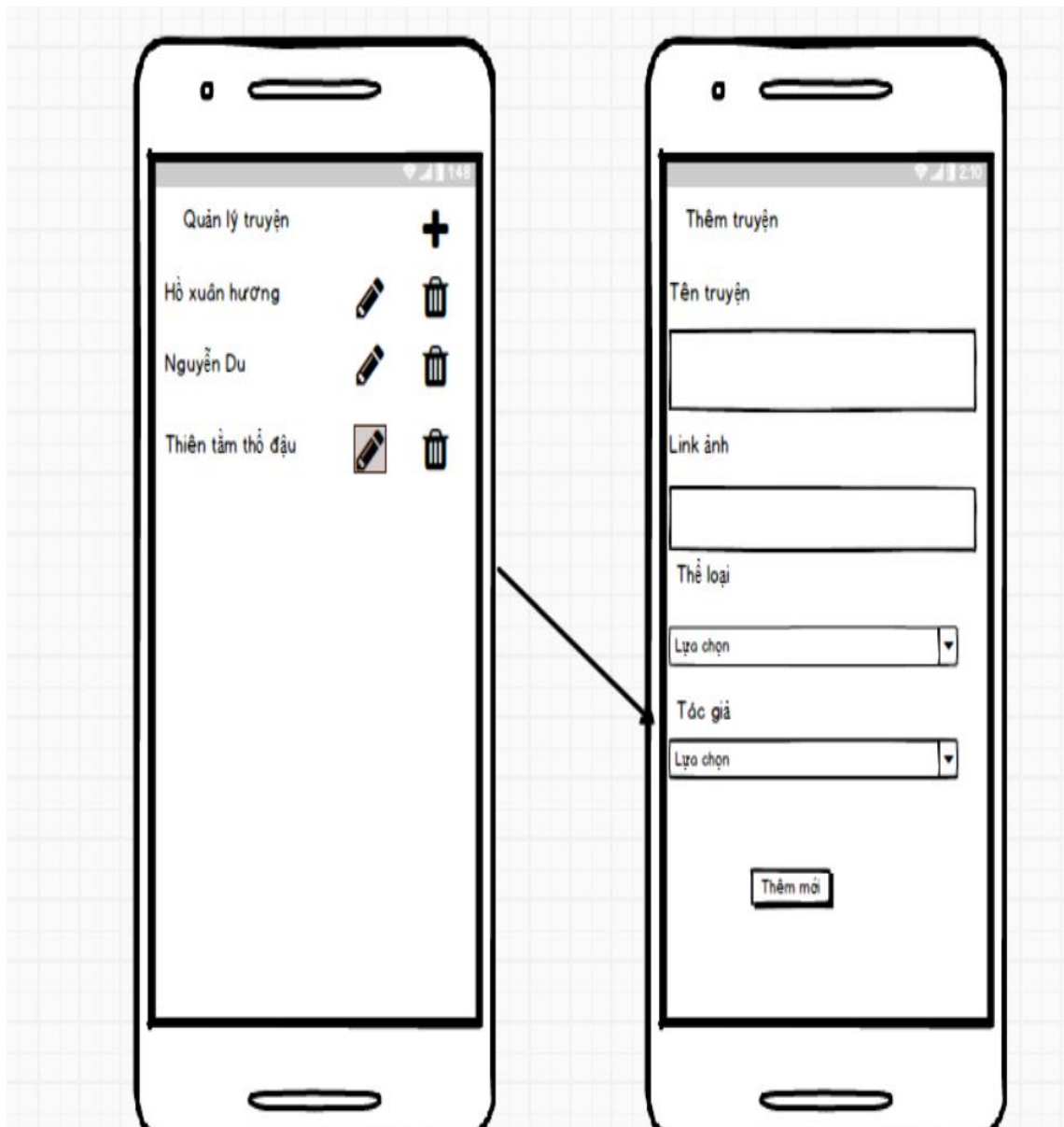
Hình 2.18 : Giao diện thêm mới thẻ loại

2.6.6 Giao diện Thêm mới tác giả



Hình 2.19: Giao diện thêm tác giả

2.6.7 Giao diện thêm mới Thêm truyện



Hình 2.20 Giao diện thêm truyện

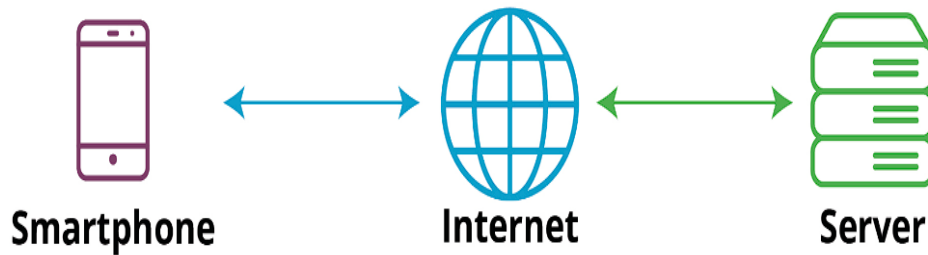
2.7 Tóm tắt chương

Chương này đã đề cập đến các chức năng của ứng dụng, phân tích yêu cầu, mô tả các biểu đồ UC, sơ đồ lớp, biểu đồ trình tự và thiết kế giao diện của từng chức năng của ứng dụng đọc truyện. Những kiến thức của chương này chính là cơ sở cho các chương sau.

CHƯƠNG 3 .THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1 Mô hình thử nghiệm

Mô hình client-sever là một kiến trúc phần mềm phổ biến được chia thành 2 phần chính: Client(người dùng) và Server. Hai thành phần này tương tác với nhau thông qua mạng Lan hoặc Internet để cung cấp dịch vụ và tài nguyên cho người dùng.



Hình 3.1 Mô hình thử nghiệm

3.2 Kiểm thử ứng dụng hỗ trợ đọc truyện

3.2.1 Kế hoạch kiểm thử

Thực hiện tạo ca kiểm thử với các chức năng của ứng dụng đọc truyện.

- Phía người dùng:
 - 2) Đăng kí /Đăng nhập
 - 3) Xem chi tiết truyện
 - 4) Bình luận
 - 5) Yêu thích
 - 6) Tìm kiếm
- Phía quản trị viên :
 - Quản lý tác giả
 - Quản lý thể loại
 - Quản lý truyện

3.2.2 Mục tiêu kiểm thử

Tập trung vào kiểm tra chức năng cơ bản với hạng mục kiểm tra sau:

- Kiểm tra chức năng
- Kiểm tra giá trị biên

3.2.3 Chiến lược kiểm thử

- Tạo các ca kiểm thử dựa vào các kịch bản
- Kỹ thuật kiểm thử được áp dụng để tạo các ca kiểm thử là phân tích giá trị biên

3.2.4 Kịch bản kiểm thử

a) Kiểm thử chức năng đăng kí

TestCase#	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
DK1	Kiểm tra chức năng đăng kí	Email phải đúng định dạng	1) điền định dạng mail không hợp lệ	abbgmail@com	Hệ thống báo lỗi "Email không hợp lệ"	Hệ thống báo lỗi "Email không hợp lệ"	Pass
DK2	Kiểm tra chức năng đăng kí	Email chưa được đăng kí	1) điền email đã đăng kí	abc@gmail.com	Hệ thống báo lỗi "Email đã được sử dụng"	Hệ thống báo lỗi "Email đã được sử dụng"	Pass
DK3	Kiểm tra chức năng đăng kí	Mật khẩu không được ít hơn 8 kí tự	1) Điền mật khẩu ít hơn 8 kí tự	123456	Hệ thống báo lỗi "Mật khẩu phải có ít nhất 8 kí tự"	Hệ thống báo lỗi "Mật khẩu phải có ít nhất 8 kí tự"	Pass
DK4	Kiểm tra chức năng đăng kí	Đăng kí tài khoản thành công	1) Điền đúng định dạng mail 2) Điền mật khẩu nhiều hoặc bằng 8 kí tự	1233455678	Hiện thị thông báo "Đăng kí thành công" và chuyển đến màn hình đăng nhập	Hiện thị thông báo "Đăng kí thành công" và chuyển đến màn hình đăng nhập	Pass
DK5	Kiểm tra chức năng đăng kí	Email không được để trống	Để trống trường email		Hệ thống báo lỗi "Không được để trống trường email"	Hệ thống báo lỗi "Không được để trống trường email"	Pass

Hình 3.2: Mô tả testcase Đăng Kí

b) Kiểm thử chức năng đăng nhập

TestCase#	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
DN1	Kiểm tra chức năng đăng nhập	Email phải đúng định dạng	1) điền định dạng mail không hợp lệ	abbgmail@com	Hệ thống báo lỗi "Email không hợp lệ"	Hệ thống báo lỗi "Email không hợp lệ"	Pass
DN2	Kiểm tra chức năng đăng nhập	Mật khẩu không được ít hơn 8 kí tự	1) Điền mật khẩu ít hơn 8 kí tự	123456	Hệ thống báo lỗi "Mật khẩu phải có ít nhất 8 kí tự"	Hệ thống báo lỗi "Mật khẩu phải có ít nhất 8 kí tự"	Pass
DN3	Kiểm tra chức năng đăng nhập	Đăng nhập tài khoản thành công	1) Điền đúng định dạng mail 2) Điền mật khẩu nhiều hoặc bằng 8 kí tự	1233455678	Hiện thị thông báo "Đăng nhập thành công" và chuyển đến màn hình trang chủ	Hiện thị thông báo "Đăng nhập thành công" và chuyển đến màn hình trang chủ	Pass

Hình 3.3: Mô tả testcase Đăng Nhập

c) Kiểm thử chức năng bình luận

TestCase#	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
BL1	Kiểm tra chức năng bình luận	Bình luận không được để trống	1) Để trống bình luận 2) Nhấn gửi bình luận		Hệ thống báo lỗi "Vui lòng nhập bình luận"	Hệ thống báo lỗi "Vui lòng nhập bình luận"	Pass
BL3	Kiểm tra chức năng bình luận	Bình luận thành công	1) Điền 1 bình luận	Truyện hay	Hệ thống hiển thị bình luận mới	Hệ thống hiển thị bình luận mới	Pass

Hình 3.4: Mô tả testcase Bình Luận

d) Kiểm thử chức năng yêu thích

TestCase#	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
YT1	Kiểm tra chức năng Yêu thích	Yêu thích thành công	1) Nhấp vào button yêu thích		Hệ thống báo "yêu thích truyền thành công"	Hệ thống báo "yêu thích truyền thành công"	Pass

Hình 3.5: Mô tả testcase Yêu Thích

e) Kiểm thử chức năng Tìm kiếm

TestCase#	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
TK1	Kiểm tra chức năng Tìm Kiếm	Tìm kiếm không được để trống	1) Nhấp vào ô tìm kiếm 2) Để trống và ấn tìm kiếm		Hệ thống báo "Vui lòng nhập thông tin để tìm kiếm"	Hệ thống báo "Vui lòng nhập thông tin để tìm kiếm"	Pass
TK2	Kiểm tra chức năng Tìm Kiếm	Tìm kiếm truyền bất kì	1) Nhấp vào ô tìm kiếm 2) Nhập tên truyện và ấn tìm kiếm	Thư kiếm trường truyện phù hợp An	Hệ thống hiển thị ra tên	Hệ thống không hiển thị ra truyện phù hợp	Fail

Hình 3.6: Mô tả testcase Tìm Kiếm

3.2.5 Báo cáo kiểm thử

Bảng 7: Báo cáo kiểm thử

	Tổng số testcase	Số testcase Pass	Số TCs Fail
Đăng nhập	3	3	0
Đăng kí	5	5	0
Bình luận	2	2	0
Yêu thích	1	1	0
Tìm kiếm	2	1	1
Tổng	13	12	1

3.3 Tóm Tắt Chương

Chương này đưa ra mô hình kiểm thử, kế hoạch kiểm thử, mục tiêu chiến lược, các kịch bản kiểm thử và báo cáo kiểm thử từ đó đánh giá được chất lượng tính năng của ứng dụng.

KẾT LUẬN

Sau một thời gian tìm hiểu và xây dựng đề tài “**Xây dựng ứng dụng hỗ trợ đọc truyện sử dụng React Native**”, nhiều kinh nghiệm và kiến thức bổ ích đã được tích lũy. Những kết quả nổi bật mà đề tài đạt được bao gồm :

- Về công nghệ :
 - Nắm vững về quá trình xây dựng ứng dụng di động với framework React Native.
 - Sử dụng thành thạo Firebase để tạo cơ sở dữ liệu.
 - Có thêm kiến thức xây dựng mã nguồn, tối ưu hóa , bảo mật...
- Về sản phẩm :
 - Xây dựng thành công ứng dụng di động đọc truyện Argon Comic với các tính năng theo yêu cầu mô tả , thiết kế rõ ràng, mô tả chi tiết và dễ hình dung.
 - Giao diện ứng dụng dễ nhìn, dễ dàng thao tác phù hợp với nhiều đối tượng.
 - Ứng dụng có bảo mật, an toàn cho người dùng.

Hướng phát triển :

- Hoàn thiện ứng dụng, sửa lỗi phát sinh và cải thiện tính bảo mật.
- Tích hợp thêm các chức năng như chế độ ban đêm, thêm nhiều thể loại, chia sẻ ứng dụng cho bạn bè, xây dựng nơi trò chuyện cho mọi người.
- Đẩy ứng dụng lên lên chợ ứng dụng như Google Play, App Store.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1]. Hoàng Quang Huy (2016), *Giáo trình cơ sở dữ liệu*, NXB Giáo dục VN.
- [2]. Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thuý, Phạm Thị Kim Phụng (2011), *Giáo trình phân tích thiết kế hệ thống*, NXB Giáo dục VN.
- [3]. Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thuý (2011), *Giáo trình cơ sở dữ liệu*, NXB Giáo dục VN.
- [4]. Nguyễn Bá Nghiênn (2020), *Giáo trình Phân tích thiết kế hướng đối tượng*, Nhà xuất bản Thống kê, Hà Nội.
- [5]. Frank Zammetti(2018), Pratical React Native.
- [6]. Adam Boduch(2020), React and React Native, 5th Edition.
- [7]. Nicholas Brown(2016), React Native Tutorial.

PHỤ LỤC :Hướng dẫn cài đặt và sử dụng

1.Cài đặt môi trường

- *Cài đặt Visual Studio Code (VSCode)*
 - Bước 1 : Truy cập vào link :<https://code.visualstudio.com/>
 - Bước 2 : Chọn phiên bản phù hợp với hệ điều hành, tải xuống và cài đặt
- *Cài đặt Android Studio*
 - Bước 1: Truy cập trang chủ của Android Studio
<https://developer.android.com>
- *Chạy chương trình*
 - Bước 1 : Giải nén tệp zip chứa mã nguồn
 - Bước 2: Mở folder bằng VS Code
 - Bước 3: Mở Terminal , Gõ lần lượt các dòng lệnh để chạy cài đặt: “npm run install”, “npm run build”, Chương trình sẽ được khởi chạy.
 - Bước 4 : Mở Android Studio chạy máy ảo