TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**

**---------🙞🙜🕮🙞🙜---------**



ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**XÂY DỰNG ỨNG DỤNG PODCAST CHO TRUYỆN MLIS**

Sinh viên thực hiện: Lê Hoàng Dương

Giảng viên hướng dẫn: TS. Nguyễn Thọ Thông

HÀ NỘI, 2023

# MỤC LỤC

# DANH MỤC HÌNH ẢNH

# DANH MỤC BẢNG BIỂU

# DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH THUẬT NGỮ

# TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT

## Tổng quan

### Đặt vấn đề

Truyện là một chuỗi các sự kiện được kết nối với nhau và được kể thông qua lời nói, hình ảnh, ngôn ngữ cơ thể, biểu diễn, âm nhạc hoặc bất kỳ hình thức giao tiếp nào khác. Một câu truyện có thể kể về bất cứ điều gì, các sự kiện được mô tả có thể là thực hoặc tưởng tượng. Chúng ta có thể tiếp thu nội dung câu chuyện bằng nhiều cách khác nhau, người ta có thể đọc truyện thông qua sách, nghe kể truyện qua người nói hoặc các phương thức kể truyện qua mạng. nhìn và nghe toàn bộ nội dung truyện bằng cả hình ảnh và âm thanh qua phim ảnh, biểu diễn trực tiếp.

Việc đọc sách truyện qua hình thức nghe mang một số ưu điểm so với đọc sách truyện trực tiếp như:

* Nghe truyện giúp người nghe có thể tiết kiệm thời gian và công sức so với đọc truyện trực tiếp, người nghe có thể nghe truyện khi làm các công việc khác như lái xe, tập thể dục, làm việc nhà, hay thậm chí là đang chuẩn bị ngủ
* Nghe truyện có thể tiếp cận tới các đối tượng bị mù hay khuyết tật về thị giác, giúp họ trải nghiệm nội dung câu chuyện một cách thuận lợi hơn

Ngoài ra về mặt phương tiện truyền thông, ta có podcast – một loại truyền thông giới thiệu, giải trí và giáo dục dưới dạng file âm thanh hoặc video có thể tải về để nghe hoặc xem. Podcast cho phép người dùng theo dõi các chương trình âm thanh hoặc video trực tuyến thông qua các ứng dụng podcast hoặc trình duyệt web. Người dùng có thể sử dụng các ứng dụng podcast hoặc trình duyệt web để nghe hoặc xem podcast trên thiết bị của họ

Với sự phát triển về công nghệ thông tin và ứng dụng di động, việc phát triển một ứng dụng podcast cho truyện giúp người dùng có thể nghe podcast trên chính thiết bị di động của mình

## Cơ sở lý thuyết

### Tổng quan về lập trình Android

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Android.Inc với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005 và ra mắt vào năm 2007.[1]

Lập trình Android là việc lập trình viên xây dựng ứng dụng trên nền tảng Android

Thông thường thì các lập trình viên sử dụng 2 ngôn ngữ chính là Java và Kotlin. Ngoài ra còn có thể sử dụng các ngôn ngữ khác như Javascript, Dart, C#,... Java từ lâu đã được coi là ngôn ngữ mặc định cho việc lập trình các ứng dụng trên nền tảng Android còn Kotlin là ngôn ngữ lập trình mới được phát triển và giới thiệu bởi Google và đang dần trở nên phổ biến với cộng đồng lập trình Android. Công cụ thường dùng cho việc lập trình Android đó là Android Studio, một công cụ được phát triển bởi Google nhằm hỗ trợ cho việc xây dựng ứng dụng cũng như là quản lý nó. Android Studio cung cấp đầy đủ các tính năng cần thiết cho việc lập trình Android như debug, thiết kế giao diện ứng dụng, quản lý, …

Thường thì một ứng dụng Android sẽ gồm 4 component chính đó là: Activity, Service, Broadcast Receiver, Content Provider.

Activity: Là thành phần quan trọng, nó được sử dụng cho việc tạo màn hình giao diện người dùng. Mỗi một màn hình trong ứng dụng đều được hiển thị thông qua Activity. Mỗi một Activity đều có vòng đời riêng. Về cơ bản, một Activity gồm có các trạng thái sau: onCreate, onStart, onResume, onPause, onStop, onDestroy. Với mỗi trạng thái, lập trình viên có thể thực hiện các công việc khác nhau như: khởi tạo, hiển thị, lưu trạng thái hoặc giải phóng tài nguyên,…

Service: Là thành phần chạy nền và không có giao diện người dùng. Ta cần phải khai báo service trong file AndroidManifest.

Có 3 loại service

* Foreground: Thực hiện các thao tác mà người dùng có thể chú ý, nhận biết. Foreground bắt bộc phải hiển thị một thông báo. Thường thấy là Foreground Service phát nhạc
* Background: Thực hiện thao tác mà người dùng không cần thiết phải để ý tới, từ Android 8.0, service chạy trong background bị giới hạn
* Bound: cung cấp 1 giao diện Client - Server cho phép các thành phần tương tác với nó: gửi yêu cầu, nhận kết quả và thậm chí là IPC (inter-process communication) - giao tiếp qua nhiều tiến trình

Broadcast Receiver: Là thành phần cho phép ứng dụng lắng nghe và phản hồi lại các sự kiện được gửi từ hệ thống hoặc từ các ứng dụng khác trên thiết bị của người dùng. Ví dụ như: khi thiết bị mất mạng, pin yếu, tin nhắn đến, … Nó có thể hiện thông báo, cập nhật giao diện và nhiều tác vụ khác. Khi muốn sử dụng Broadcast Receiver, ta cần khai báo trong file AndroidManifest bằng cách thêm thành phần <receiver> vào trong <application>

Content Provider: là một thành phần để quản lý truy cập dữ liệu, nó cung cấp các phương thức khác nhau để các ứng dụng có thể truy cập dữ liệu từ một ứng dụng khác. Content Provider hoạt động rất giống với một cơ sở dữ liệu, ta có thể truy vấn, chỉnh sửa nội dung, cũng như là thêm xóa các nội dung

Ngoài các component nêu trên thì còn có các component khác như: Intent, Fragment, Resource …

### Tổng quan về Java SpringBoot

Spring Boot là một framework Java dùng để phát triển ứng dụng web và các dịch vụ dựa trên Spring Framework. Nó giúp đơn giản hóa việc xây dựng và triển khai các ứng dụng Java Enterprise bằng cách cung cấp các cấu hình mặc định và tối ưu hóa quy trình phát triển. Spring Boot được thiết kế để giúp các nhà phát triển tạo ra các ứng dụng Java một cách nhanh chóng và dễ dàng, bằng cách loại bỏ những công việc cần thiết phải làm khi thiết lập một dự án Spring truyền thống. Nó cung cấp các thiết lập tự động và cấu hình mặc định, giúp giảm bớt lượng mã nguồn cần viết và tăng tốc quá trình phát triển.

Để xây dựng một ứng dụng Spring Boot, ta có thể sử dụng các công cụ như Eclipse, NetBean hay Intellij Idea

### Tổng quan về API

API là viết tắt của "Application Programming Interface" (Giao diện Lập trình ứng dụng), là một tập hợp các quy tắc và định dạng mà các ứng dụng sử dụng để giao tiếp với nhau. API cho phép hai ứng dụng hoặc hệ thống khác nhau tương tác và trao đổi dữ liệu một cách an toàn và hiệu quả. API được hoạt động theo 4 cách như sau:

* API SOAP: Máy chủ và máy khách trao đổi thông điệp bằng XML.
* API RPC: Gọi thủ tục từ xa và trả về kết quả.
* API Websocket: Dùng các đối tượng JSON để chuyển dữ liệu.
* API Rest:  Máy khách gửi yêu cầu đến máy chủ dưới dạng dữ liệu. Máy chủ dùng dữ liệu đầu vào từ máy khách này để bắt đầu các hàm nội bộ và trả lại dữ liệu đầu ra cho máy khách.

Rest API (Representational State Transfer) sử dụng các giao thức HTTP như GET, POST, PUT, … đến một địa chỉ URL để xử lý. Để sử dụng, ta cần nêu rõ mục đích thông qua các phương thức tương ứng được hỗ trợ như:

* GET: Trả về một resource hoặc một danh sách resource.
* POST: Tạo resource mới.
* PUT : Cập nhật thông tin cho resource.

Rest API gồm status code. Sau đây là một vài status code phổ biến:

* 200 OK: Trả về thành công.
* 400 Bad Request: Request không hợp lệ hoặc sai request.
* 403 Forbidden: Request bị từ chối không cho phép
* 404 Not Found: Không tìm thấy resource từ URL
* 204: No Content. Hoàn thành hoạt động nhưng sẽ không trả về nội dung gì.
* 500: Internal server error.

Và còn nhiều status code khác nữa.

## Công nghệ sử dụng

Đề tài sử dụng các công nghệ:

* Ngôn ngữ lập trình: Java phiên bản 8
* Cơ sở dữ liệu: Sử dụng hai cơ sở dữ liệu là MongoDB và MySQL

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Đặc tả hệ thống

Ứng dụng “Xây dựng ứng dụng Podcast cho truyện Mlis” là ứng dụng di động chạy trên nền tảng hệ điều hành Android, cho phép người dùng có thể nghe đọc truyện trên chính thiết bị di động của mình.

Ứng dụng sẽ cung cấp cho người dùng một giao diện đơn giản và thân thiện, giúp họ có thể tìm kiếm và nghe những mẩu truyện một cách dễ dàng

Người dùng có thể sử dụng ứng dụng để nghe truyện, nhưng nếu muốn sử dụng các chức năng nâng cao hơn thì người dùng phải đăng ký tài khoản qua email hoặc đăng nhập bằng các phương thức có sẵn

Phía quản lý ứng dụng sẽ có một giao diện riêng xây dựng dựa trên ứng dụng website để quản lý các danh sách truyện, xem các thống kê chi tiết về lượt truy cập

### Phân tích yêu cầu phần mềm

Người quản lý có thể quản lý truyện và các danh sách truyện với các công việc như:

* Thêm sửa xóa các truyện, tập truyện trong cơ sở dữ liệu, thêm truyện, tập truyện vào danh sách truyện tương ứng
* Thêm sửa xóa các danh sách truyện trong cơ sở dữ liệu
* Cập nhật lại ảnh và các file âm thanh được lưu trữ trên server

Người sử dụng ứng dụng có thể xem, tìm kiếm và nghe truyện, tạo tài khoản và đăng nhập để sử dụng các tính năng nâng cao hơn:

* Tìm kiếm truyện theo từ khóa đã nhập
* Xem các thông tin của truyện như tên tác giả, mô tả truyện
* Nghe truyện ngay cả khi thiết bị di động đã tắt màn hình, tua truyện, hẹn giờ tắt truyện
* Tạo tài khoản bằng địa chỉ email hoặc đăng nhập bằng tài khoản Google

Nếu người sử dụng ứng dụng đã đăng nhập tài khoản, họ có thể lưu lại truyện vào danh sách yêu thích để nghe vào những lần sau mà không tốn công tìm lại:

* Tạo, sửa đổi, xóa danh sách yêu thích của bản thân
* Thêm truyện vào danh sách yêu thích của bản thân

### Phân rã chức năng

Xác định các tác nhân hệ thống:

* Người dùng: Người sử dụng ứng dụng đã đăng nhập tài khoản
* Người dùng khách: Người sử dụng ứng dụng chưa đăng nhập tài khoản
* Quản lý: Người quản trị hệ thống

Xác định UC:

* Đăng nhập
* Đăng ký
* Đăng xuất
* Quản lý truyện

+ Thêm truyện

+ Sửa đổi thông tin truyện

+ Xóa truyện

* Quản lý danh sách truyện

+ Tạo danh sách truyện

+ Sửa đổi thông tin danh sách truyện

+ Xóa danh sách truyện

* Quản lý danh sách yêu thích

+ Tạo danh sách yêu thích

+ Thêm truyện vào danh sách yêu thích

+ Xóa truyện khỏi danh sách yêu thích

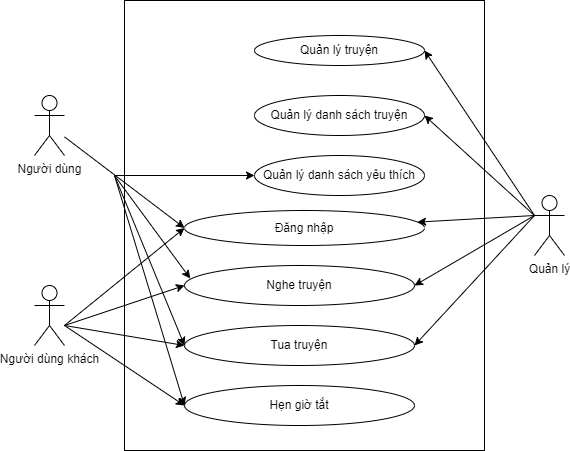
+ Xóa danh sách yêu thích

* Xem báo cáo lượt truy cập
* Xem báo cáo lịch sử sửa đổi
* Nghe truyện
* Tua truyện
* Hẹn giờ tắt

## Các biểu đồ

### Biểu đồ UC

#### Biểu đồ UC tổng quát



Biểu đồ UC tổng quát

#### Biểu đồ UC chi tiết cho Actor Quản lý

#### Biểu đồ UC chi tiết cho Actor Người dùng

#### Biểu đồ UC chi tiết cho Actor Người dùng khách

### Biểu dồ lớp

### Kịch bản

### Biểu đồ hoạt động

### Thiết kế cơ sở dữ liệu

## Cơ sở dữ liệu

# THỰC NGHIỆM VÀ ĐÁNH GIÁ SO SÁNH

## Môi trường thử nghiệm

## Kết quả thử nghiệm

### Tính ổn định

## Đánh giá và so sánh

**KẾT LUẬN**

# TÀI LIỆU THAM KHẢO

[1] Android(Hệ điều hành)

https://vi.wikipedia.org/wiki/Android\_(h%E1%BB%87\_%C4%91i%E1%BB%81u\_h%C3%A0nh)