

TP. HCM

2022

# BỘ THÔNG TIN VÀ TRUYỀN THÔNG

DƯƠNG PHẠM THANH TÒNG MSSV: N18DCCN189 CHUYÊN NGÀNH: Công Nghệ Phần Mềm 2018-2023 Lớp: D18CQCP02-N 2018 -2023

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

BÁO CÁO ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**Đề tài: “Xây dựng website nghe và tìm nhạc phẩm**

**theo cảm xúc dựa trên phương pháp học sâu”**

## Người hướng dẫn : ThS.NGUYỄN NGỌC DUY Sinh viên thực hiện : DƯƠNG PHẠM THANH TÒNG

**Mã số sinh viên : N18DCCN0189**

**Lớp : D18CQCP02-N**

**Khóa : 2018-2023**

**Hệ : ĐẠI HỌC CHÍNH QUY**

**TP.HCM, tháng 12/2022**



**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

BÁO CÁO ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**Đề tài: “Xây dựng website nghe và tìm nhạc phẩm**

**theo cảm xúc dựa trên phương pháp học sâu”**

## Người hướng dẫn : ThS.NGUYỄN NGỌC DUY Sinh viên thực hiện : DƯƠNG PHẠM THANH TÒNG

**Mã số sinh viên : N18DCCN0189**

**Lớp : D18CQCP02-N**

**Khóa : 2018-2023**

**Hệ : ĐẠI HỌC CHÍNH QUY**

**TP.HCM, tháng 12/2022**

BM4-TN

|  |  |
| --- | --- |
| HỌC VIỆN | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG | **Độc lập – Tự do – Hạnh phúc** |
| CƠ SỞ TẠI TP. HỒ CHÍ MINH |  |
| **Khoa: CNTT2** | *TP. Hồ Chí Minh, ngày tháng năm .* |

PHIẾU ĐĂNG KÝ ĐỀ TÀI THỰC HIỆN

ĐỒ ÁN/KHOÁ LUẬN TỐT NGHIỆP

***Kính gửi***: Lãnh đạo khoa CNTT2

1. **Họ và tên sinh viên: Dương Phạm Thanh Tòng Mã SV: N18DCCN189**

Lớp: D18CQCP02-N. Ngành: Công Nghệ Thông Tin. Hình thức đào tạo: ĐHCQ.

1. **Tên đồ án tốt nghiệp: Xây dựng website nghe và tìm nhạc phẩm theo cảm xúc dựa trên phương pháp học sâu.**
2. **Đề cương:**

* **Lý thuyết:**
* Nghiên cứu các phương pháp học sâu.
* Nghiên cứu xây dựng cơ sở tri thức cho hệ thống học sâu về cảm xúc mà âm nhạc mang lại cho người nghe.
* Tìm hiểu các công nghệ xây dựng website nghe nhạc trực tuyến.
* **Thực hành:**
* Xây dựng cơ sở tri thức về nhạc và cảm xúc khi nghe nhạc.
* Thiết kế cơ sở dữ liệu cho website nghe nhạc trực tuyến.
* Xây dựng hệ thống học sâu cho mục đích phân tích cảm xúc.
* Xây dựng website nghe nhạc trực tuyến với chức năng tìm kiếm nhạc theo cảm xúc với các chức năng:
* Chọn bài hát khi biết tựa đề
* Tìm bài hát phù hợp cảm xúc.

1. **Giáo viên hướng dẫn: ThS. Nguyễn Ngọc Duy**
2. **Thời gian thực hiện:**

Từ ngày: …… tháng …… năm ……… đến ngày: …… tháng …… năm ……..

|  |  |  |
| --- | --- | --- |
| TRƯỞNG BỘ MÔN | GIÁO VIÊN HƯỚNG DẪN | SINH VIÊN ĐĂNG KÝ |
|  |  |  |

# LỜI CẢM ƠN

Kính thưa thầy cô và các bạn!

Thời gian học tập tại mái trường Học viện Công nghệ Bưu chính Viễn thông cơ sở tại Tp. Hồ Chí Minh của bản thân em sắp sửa kết thúc .Qua thời gian học tập, em đã tiếp thu cho mình một số kiến thức cơ bản dưới sự giảng dạy của thầy cô tại Học Viện Công Nghệ Bưu Chính Viễn thông cơ sở tại TP.HCM. Để gắn với thực tiễn, lấy phương châm “học đi đôi với thực hành” em đã chọn và làm đề tài “**Xây dựng website nghe và tìm nhạc phẩm theo cảm xúc dựa trên phương pháp học sâu**”. Trong suốt thời gian thực hiện đề tài, em đã nhận được rất nhiều sự quan tâm, giúp đỡ, ủng hộ của thầy cô, bạn bè và gia đình. Chính điều đó là niềm khích lệ, động viên lớn để giúp em thực hiện đề tốt tài thực tập tốt nghiệp này.

Trước tiên, em xin gửi lời tri ân sâu sắc đến các thầy cô trường Học Viện Công Nghệ Bưu Chính Viễn thông cơ sở tại TP.HCM đã tận tình dẫn dắt và truyền đạt cho em rất nhiều kiến thức quý báu trong các học kỳ vừa qua.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến ThS. Nguyễn Ngọc Duy, thầy đã hướng dẫn tận tình, truyền đạt kiến thức, chỉ bảo cho em trong suốt thời gian học tập và thực hiện đề tài. Kính chúc thầy và gia đình nhiều sức khoẻ và thành công trong cuộc sống.

Con xin gửi lòng biết ơn và sự kính trọng đến cha mẹ, gia đình đã luôn động viên, tạo điều kiện tốt nhất cho con học tập.

Và xin chân thành cảm ơn tất cả các bạn đã luôn sát cánh, giúp đỡ, động viên mình trong những thời điểm khó khăn, tiếp thêm động lực và ý chí giúp vượt qua khó khăn trong suốt quá trình học tập tại trường.

Tuy nhiên, vì thời gian và kiến thức còn hạn chế em đã cố gắng rất nhiều để hoàn thành đề tài này, nhưng trong quá trình làm bài em không tránh khỏi những thiếu sót, kính mong được lời nhận xét và góp ý của thầy cô. Một lần nữa, em xin chân thành cảm ơn !

TP.HCM, tháng 12 năm 2022 Sinh viên thực hiện

Dương Phạm Thanh Tòng

**MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc121446041)

[LỜI MỞ ĐẦU viii](#_Toc121446042)

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 1](#_Toc121446043)

[1.1. Tổng quan: 1](#_Toc121446044)

[1.2. Mục tiêu, yêu cầu của đề tài: 1](#_Toc121446045)

[1.3. Mục tiêu của đề tài 2](#_Toc121446046)

[1.4. Yêu cầu về lý thuyết 2](#_Toc121446047)

[1.5. Yêu cầu về thực hành 2](#_Toc121446048)

[1.6. Công Nghệ sử dụng 3](#_Toc121446049)

[1.7. Công Cụ sử dụng 3](#_Toc121446050)

[1.8. Lĩnh Vực 3](#_Toc121446051)

[1.9. Phương Pháp tiếp cận 3](#_Toc121446052)

[1.10. Chức năng chính 3](#_Toc121446053)

[CHƯƠNG 2: CƠ SỞ XÂY DỰNG HỆ THỐNG 4](#_Toc121446054)

[2.1. Giới Thiệu Công Cụ Xây Dựng Hệ Thống 4](#_Toc121446055)

[2.1.1. Giới thiệu IntelliJ 4](#_Toc121446056)

[2.1.2. Giới thiệu Visual Studio Code 4](#_Toc121446057)

[2.1.3. Giới thiệu về Microsoft SQL Server Management 4](#_Toc121446058)

[2.1.4. Giới thiệu PyCharm 5](#_Toc121446059)

[2.2. Giới Thiệu Công Nghệ Xây Dựng Hệ Thống 5](#_Toc121446060)

[2.2.1. Tổng quan về Python 5](#_Toc121446061)

[*2.2.2.* Tổng quan về Java 6](#_Toc121446062)

[2.2.3. Tổng quan về Spring Framework 9](#_Toc121446063)

[2.2.4. Tổng quan về Spring Boot. 10](#_Toc121446064)

[2.2.5. Tổng quan về Học máy. 11](#_Toc121446065)

[2.2.6. Tổng quan về RESTful API 12](#_Toc121446066)

[2.2.7. Tổng quan về ReactJS 13](#_Toc121446067)

[CHƯƠNG 3: CƠ SỞ LÍ THUYẾT 14](#_Toc121446068)

[3.1. Giới thiệu về xử lí ngôn ngữ tự nhiên 14](#_Toc121446069)

[3.2. Giới thiệu về cơ sở tri thức 15](#_Toc121446070)

[3.2.1. Khái niệm 15](#_Toc121446071)

[3.2.2. Cấu trúc của hệ cơ sở tri thức 15](#_Toc121446072)

[3.2.3. Các Đặc tính của hệ cơ sở tri thức 16](#_Toc121446073)

[3.2.4. Các hệ thống liên quan 16](#_Toc121446074)

[3.2.5. Ứng dụng 16](#_Toc121446075)

[3.2.6. Quá trình xây dựng cơ sở tri thức cho hệ thống học sâu về cảm xúc mà âm nhạc mang lại cho người xem 16](#_Toc121446076)

[3.3. Tổng quan về Deep Learning 18](#_Toc121446077)

[3.3.1. Deep Learning là gì 18](#_Toc121446078)

[3.3.2. Mạng neural nhân tạo 18](#_Toc121446079)

[3.3.3. Các mạng neural trong xử lý ngôn ngữ tự nhiên 19](#_Toc121446080)

[3.3.3.2. Recurrent Neural Networks (RNNs) 21](#_Toc121446081)

[3.4. Tổng quan về word2vec 25](#_Toc121446082)

[3.4.1. Mô Hình Skip-Gram 25](#_Toc121446083)

[3.4.2. Mô Hình Túi từ liên Tục (CBOW) 27](#_Toc121446084)

[CHƯƠNG 4: KHẢO SÁT HỆ THỐNG 29](#_Toc121446085)

[4.1. Mô hình hoạt động 29](#_Toc121446086)

[4.1.1. Chuẩn bị dữ liệu cho mô hình (Data Preparation) 29](#_Toc121446087)

[4.1.2 Xây dựng mô hình (Data Modeling) 29](#_Toc121446088)

[4.1.3 Kiểm định mô hình (Validation model) 29](#_Toc121446089)

[4.1.4 Sử dụng mô hình (Model Usage) 29](#_Toc121446090)

[4.2. Yêu cầu khi cài đặt thuật toán 29](#_Toc121446091)

[4.3. Phân tích – Xây dựng kho dữ liệu cho hệ thống 30](#_Toc121446092)

[4.4. Khảo sát hệ thống website nghe và tìm nhạc phẩm theo cảm xúc dựa trên phương pháp học sâu 31](#_Toc121446093)

[4.5. Các chức năng của hệ thống. 31](#_Toc121446096)

[4.5.1. Yêu cầu chức năng của hệ thống: 31](#_Toc121446097)

[4.5.2. Yêu cầu phi chức năng: 32](#_Toc121446101)

[CHƯƠNG 5: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 33](#_Toc121446102)

[5.1. Mô hình chức năng nghiệp vụ (BFD – Business Function Diagram) 33](#_Toc121446103)

[5.2. Mô hình luồng dữ liệu (DFD – Data Flow Diagram) 34](#_Toc121446104)

[5.3. Mô hình cộng tác (Collaboration Diagram) 36](#_Toc121446105)

[5.4. Thiết lập mô hình thực thể ERD (ERD - Entity Relationship Diagram) 36](#_Toc121446106)

[5.4.1. Xác định các thực thể 36](#_Toc121446107)

[5.4.2. Mô hình thực thể kết hợp MS (ERD - Entity Relationship Diagram 37](#_Toc121446108)

[5.5. Thiết kế cơ sở dữ liệu 37](#_Toc121446109)

[5.5.1. Mô hình Diagram 37](#_Toc121446110)

[5.5.2. Từ điển dữ liệu 38](#_Toc121446112)

[CHƯƠNG 6: CHƯƠNG TRÌNH 41](#_Toc121446113)

[6.1. Giới Thiệu Chương Trình 41](#_Toc121446114)

[6.1.1. Mục Tiêu: 41](#_Toc121446115)

[6.2. Thiết kế Web Service 41](#_Toc121446116)

[6.2.1. Giao tiếp Web Service 41](#_Toc121446117)

[6.2.2. Kiến trúc của Web service 42](#_Toc121446118)

[6.2.3. Authorize và Authentication 42](#_Toc121446119)

[6.3. Thiết kế Web cho khách hàng 43](#_Toc121446120)

[6.3.1. Mô Hình sử dụng: 43](#_Toc121446121)

[6.4. Thiết kế giao diện chức năng: 43](#_Toc121446122)

[ Giao diện Đăng nhập 43](#_Toc121446123)

[ Giao diện Menu 44](#_Toc121446124)

[ Giao diện Tìm nhạc theo cảm xúc 44](#_Toc121446125)

[ Giao diện Nghe Bài hát 45](#_Toc121446126)

[ Giao diện Bài hát 45](#_Toc121446127)

[CHƯƠNG 7: KẾT LUẬN 47](#_Toc121446128)

[7.1. Kết quả đạt được và hạn chế 47](#_Toc121446129)

[7.2. Hướng phát triển 47](#_Toc121446131)

[DANH MỤC TÀI LIỆU THAM KHẢO 48](#_Toc121446132)

|  |  |
| --- | --- |
| API: Application Programming  Interface | Giao diện lập trình ứng dụng |
| App: Application | Ứng dụng |
| DFD: Data Flow Diagram | Mô hình luồng dữ liệu |
| ERD: Entity Relationship Diagram | Mô hình thực thể kết hợp |
| HTTP: HyperText Transfer  Protocol | Giao thức truyền tải siêu văn bản |
| IDE: Integrated development  environment | Môi trường phát triển tích hợp |
| IoT: Internet of Things | Internet vạn vật |
| MVC: Model – View – Controller | Một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính |
| RAM: Random Access Memory | Một loại bộ nhớ khả biến cho phép  truy xuất đọc ghi ngẫu nhiên |
| RDBMS: Relational Database  Management System | Hệ quản trị cơ sở dữ liệu quan hệ |
| REST: Representational State  Transfer | Một tiêu chuẩn dùng trong việc thiết kế  API cho các ứng dụng web |
| IIS: Internet Information Services | Một phần mềm máy chủ web mở rộng  được tạo bởi Microsoft |
| UI: User Interface  CSTT  CNNs | Giao diện người dùng  Cở sở tri thức  Convolutional Neural Network |

[Hình 1: Mô Hình Tổng quát ứng dụng 2](#_Toc121444727)

[Hình 2 Luồng thực thi lập trình Java 7](#_Toc121444728)

[Hình 3 : Mô Hình Cấu Tạo một neural 18](file:///C:\Users\ad\Desktop\CP_189_DuongPhamThanhTong.docx#_Toc121444729)

[Hình 4: Minh hoạ mạng neural CNN cho xữ lý ngôn ngữ tự nhiên 20](#_Toc121444730)

[Hình 5: Minh hoạ ma trận max-pooling trong CNN 21](file:///C:\Users\ad\Desktop\CP_189_DuongPhamThanhTong.docx#_Toc121444731)

[Hình 6: Mô hình mạng neural RNN 21](#_Toc121444732)

[Hình 7: Sự lặp lại kiến trúc module trong mạng RNN chứa một tầng 23](#_Toc121444733)

[Hình 8: Sự lặp lại kiến trúc module trong mạng LSTM chứa 4 tầng ẩn (3 sigmoid và 1 tanh) tương tác 23](#_Toc121444734)

[Hình 9: Diễn giải các kí hiệu trong đồ thị mạng nơ ron (áp dụng chung cho toàn bộ bài) 23](#_Toc121444735)

[Hình 10: Kiến trúc mạng Neural [1] 24](#_Toc121444736)

[Hình 11: Kiến trúc mạng Neural [2] 24](file:///C:\Users\ad\Desktop\CP_189_DuongPhamThanhTong.docx#_Toc121444737)

[Hình 12 Mô tả kho dữ liệu 30](#_Toc121444738)

[Hình 13: . Mô hình phân rã chức năng được biểu diễn thành hình cây phân cấp 33](#_Toc121444739)

[Hình 14: Các đối tượng cơ bản sử dụng để thiết kế biểu đồ luồng dữ liệu 34](file:///C:\Users\ad\Desktop\CP_189_DuongPhamThanhTong.docx#_Toc121444740)

[Hình 15: Mô hình Data Flow Diagram tổng quát 35](#_Toc121444741)

[Hình 16: Mô hình Data Flow Diagram mức 0 35](#_Toc121444742)

[Hình 17: Mô hình Data Flow Diagram mức 1 **Error! Bookmark not defined.**](#_Toc121444743)

[Hình 18: Mô hình cộng tác 36](#_Toc121444744)

[Hình 19: Mô hình ERD 37](#_Toc121444745)

[Hình 20: Database Diagram 37](#_Toc121444746)

[Hình 21: Giao tiếp Toàn Hệ Thống 41](#_Toc121444747)

[Hình 22: Mô Hình Three-Tier Architecture 42](#_Toc121444748)

[Hình 23: Authorize và Authentication 43](#_Toc121444749)

[Hình 24: Mô Hình MVC 43](#_Toc121444750)

[Hình 25: Giao diện đăng nhập 44](#_Toc121444751)

[Hình 26: Giao diện Menu 44](#_Toc121444752)

[Hình 27: Giao diện Tìm kiếm và kết quả xử lí câu 45](#_Toc121444753)

[Hình 28: Giao diện Nghe bài hát 45](#_Toc121444754)

[Hình 29: Giao diện bài hát 46](#_Toc121444755)

# LỜI MỞ ĐẦU

**Lý do chọn đề tài**

Ngày nay, con người luôn quan tâm chú trọng hơn tới việc giải trí sau những giờ làm việc căng thẳng mệt mỏi, đi đôi với việc thời đại Công nghệ thông tin ngày càng phát triển, việc chăm sóc sức khỏe tinh thần cũng được phát triển hơn khi được ứng dụng các công nghệ tiên tiến về phần mềm cũng như phần cứng, giúp việc giải trí từ đó trở nên hiệu quả, tiết kiệm thời gian, công sức và tiền bạc hơn.

Nghe nhạc là phương pháp phổ biến nhất được mọi người tìm đến khi muốn giải tỏa áp lực và giải trí. Lúc này, người dùng sẽ mong muốn được nghe những bài nhạc yêu thích và được giải tỏa cảm xúc, người dùng sẽ tìm kiếm những bài hát theo tên bài hát mà họ yêu thích. Tuy nhiên, trí nhớ của con người sẽ không thể nhớ hết được và số lượng bài hát thì rất là nhiều, người dùng muốn nghe một bài hát mới, một bài hát phù hợp với tâm trạng nhưng nếu không biết tên bài hát, thì làm sao có thể tìm được, làm sao có thể giải tỏa mong muốn giải trí được.

Chính vì điều băn khoăn này, riêng bản thân em cũng đã từng có những lần gặp phải vấn đề này, và theo em khảo sát ở những nhóm bạn nhỏ, những đồng nghiệp, người thân bạn bè. Họ cũng có những mong muốn tìm kiếm những bài nhạc không bằng tên bài hát mà bằng cảm xúc. Vì vậy, em đã chọn đề tài “**Xây dựng website nghe và tìm nhạc phẩm theo cảm xúc dựa trên phương pháp học sâu”** với mục đích có thể giải quyết vấn đề nêu trên. Cụ thể, đề tài dựa trên những số liệu rất thực tế đã thu thập được, sau đó thông qua phương pháp học sâu để có thể đề xuất cho người dùng những bài nhạc hay, phù hợp tâm trạng, ứng dụng kết quả này vào chương trình website nghe nhạc để giải trí với những chức năng cơ bản.

Với kiến thức còn hạn chế, chắc chắn không thể tránh khỏi những sai sót, vì vậy em rất mong nhận được sự góp ý của các thầy cô

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## Tổng quan:

Trí tuệ nhân tạo (Artificial intelligence – AI) là một ngành thuộc lĩnh vực khoa học máy tính. Là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người. AI là một bộ phận của khoa học máy tính và do đó nó phải được đặt trên những nguyên lý lý thuyết vững chắc, có khả năng ứng dụng được nhiều lĩnh vực.

Trong lĩnh vực giải trí, việc dùng AI để phân tích cảm xúc người dùng và gợi ý nhạc vẫn còn nhiều hạn chế, song đã có nhiều ứng dụng hữu ích. Chẳng hạn, Youtube và Spotify đã bước đầu có thể gợi ý cho khách hàng những video và những bài hát dựa theo xu hướng của người dùng, cũng như là những bình luận về bài hát mà từ đó những ứng dụng này đề xuất những nội dung giải trí mới lạ cho người dùng. Việc dùng AI để gợi ý theo cảm xúc hay theo chủ đề đang phát triển và được các chuyên gia dự đoán sẽ tăng trưởng vượt bậc, là xu hướng cho tương lai của tất cả các phần mềm giải trí và thương mại.

Học sâu là một chức năng của trí tuệ nhân tạo (AI), bắt chước hoạt động của bộ não con người trong việc xử lí dữ liệu và tạo ra các mẫu để sử dụng cho việc ra quyết định.

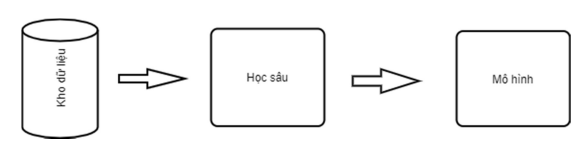
Học sâu là tập con của học máy trong AI, có các mạng lưới có khả năng "học" mà không bị giám sát từ dữ liệu không có cấu trúc hoặc không được gắn nhãn.

Đề tài “**Xây dựng website nghe và tìm nhạc phẩm theo cảm xúc dựa trên phương pháp học sâu”** sử dụng phương pháp học sâu để nhận diện cảm xúc người dùng dựa vào những gì người dùng gõ vào thanh tìm kiếm. Người dùng truy cập vào website và gõ vào thanh tìm kiếm, từ đó sẽ nhận được những bài nhạc đề xuất phù hợp. Website sẽ giúp người dùng tìm kiếm những bài nhạc mới mà có thể chưa nghe lần nào, giảm những tìm kiếm theo tên bài hát, việc đề xuất này sẽ góp phần giảm tải các bước xử lí tìm kiếm cho hệ thống, làm hệ thống hoạt động nhanh hơn.

Do quỹ thời gian và nguồn tài liệu còn hạn chế, việc tham khảo chuyên môn từ các chuyên gia âm nhạc là rất khó khăn. Vì vậy, đề tài tập trung hướng đến nghiên cứu xây dựng một mô hình hoạt động linh hoạt, dựa trên các dữ liệu là những bình luận của người dùng trên những trang web giải trí lớn như youtube, từ đó sẽ tối ưu chương trình để có thể dễ dàng đề xuất cho người dùng.

## Mục tiêu, yêu cầu của đề tài:

Các phương pháp học sâu đã chứng minh được hiệu quả trong rất nhiều lĩnh vực như kinh tế, tài chính, sinh học, … Giải trí là một lĩnh vực phức tạp nhưng cũng có nhiều hệ thống được xây dựng dựa trên trí tuệ nhân tạo để phân tích hình ảnh để gợi ý, phân tích. Các hệ thống này ngoài việc sử dụng trí tuệ nhân tạo còn cần rất nhiều nghệ sĩ đầu ngành cố vấn. Để có thể xây dựng được một hệ thống ứng dụng học sâu cần nhiều công sức cho các giai đoạn thu thập, xử lý và phân tích dữ liệu để có thể rút trích ra những tri thức từ đó. Vì vậy, trong phạm vi làm Đồ án tốt nghiệp, đề tài xây dựng ứng dụng học sâu trong gợi ý những bài nhạc theo cảm xúc mới chỉ ở mức độ tiếp cận về mặt phương pháp thu thập và xử lý dữ liệu dựa trên thuật toán CNN để đưa ra những bài nhạc được gợi ý theo mô hình. Các phân tích đưa ra mang tính hỗ trợ phân loại cảm xúc ban đầu của người dùng nhập vào. Mô hình tổng quát của hệ thống được trình bày ở hình 1.



**Hình 1: Mô hình tổng quát ứng dụng**

Từ câu input ban đầu do người dùng gõ vào, hệ thống sẽ phân loại cảm xúc: Yêu đời, thất tình, từ đó đưa ra những bài nhạc thích hợp theo cảm xúc.

Để phù hợp với mục đích giải trí, ứng dụng được xây dựng dựa trên môi trường Internet – website – để đối tượng người dùng có thể tiếp cận dễ dàng.

* 1. **Mục tiêu của đề tài**

• Hệ thống mang tính ứng dụng cao, phù hợp với những nhu cầu khác nhau.

• Xây dựng hệ thống gợi ý nhạc theo cảm xúc dựa vào dữ liệu đầu vào, phù hợp với những người thích nghe nhạc, dựa vào bộ dữ liệu được thu thập phù hợp với thực tế.

• Hệ thống dễ sử dụng, có các chức năng cơ bản như: Tìm nhạc theo tên và gợi ý nhạc theo cảm xúc. Sử dụng thuật toán CNN để phân loại cảm xúc người dùng.

* 1. **Yêu cầu về lý thuyết**
* Nghiên cứu các phương pháp học sâu.
* Nghiên cứu xây dựng cơ sở tri thức cho hệ thống học sâu về cảm xúc mà âm nhạc mang lại cho người nghe.
* Tìm hiểu các công nghệ xây dựng website nghe nhạc trực tuyến.
  1. **Yêu cầu về thực hành**
* Xây dựng cơ sở tri thức về nhạc và cảm xúc khi nghe nhạc.
* Thiết kế cơ sở dữ liệu cho website nghe nhạc trực tuyến.
* Xây dựng hệ thống học sâu cho mục đích phân tích cảm xúc.
* Xây dựng website nghe nhạc trực tuyến với chức năng tìm kiếm nhạc theo cảm xúc với các chức năng:
* Chọn bài hát khi biết tựa đề
* Tìm bài hát phù hợp cảm xúc**.**
  1. **Công Nghệ sử dụng**
* Python
* Java
* Microsoft SQL
* Javascript
  1. **Công cụ sử dụng**
* JetBrains Pycharm 2019.2.6
* Python 3.8
* Intellij Ultimate
* Visual Studio Code
  1. **Lĩnh vực**
* Chuyên nghành: Công Nghệ Phần Mềm
* Chuyên môn: Xây dựng website nghe và tìm nhạc phẩm theo cảm xúc,sử dụng ngôn ngữ lập trình Java và Javascript dựa trên giải thuật CNN, sử dụng ngôn ngữ lập trình Python.
* Lĩnh vực liên quan: Giải trí
  1. **Phương pháp tiếp cận**

Việc xây dựng một kho dữ liệu làm tri thức cho hệ thống dùng mô hình học sâu cho lĩnh vực giải trí là rất phức tạp và cần rất nhiều chi phí về thời gian và kinh phí. Vấn đề chọn chuyên gia làm cố vấn chuyên môn cũng là một bài toán phức tạp, nhất là đối với đối tượng sinh viên.

Để khắc phục những khó khăn trên, em đã chọn giải pháp thu thập dữ liệu từ những bình luận về cảm xúc sau khi nghe bài hát trên những trang web lớn, nhiều người dùng như youtube và spotify.

Sử dụng ngôn ngữ lập trình Python và kết hợp giải thuật học sâu bằng thuật toán CNN để thiết kế hệ thống đã phân tích cảm xúc từ 1 câu input do người dùng nhập vào.

Sử dụng ngôn ngữ lập trình Java để xây dựng backend cho website nghe nhạc và kết hợp sử dụng ReactJs để xây dựng giao diện website.

* 1. **Chức năng chính**
* Xây dựng được một hệ thống dựa trên giải thuật học sâu để hiểu được đặc trưng của các bình luận về cảm xúc mà người dùng có được khi nghe bài hát.
* Xây dựng Website nghe nhạc đơn giản để có thể trực quan hóa việc tìm kiếm nhạc theo cảm xúc.

# CHƯƠNG 2: CƠ SỞ XÂY DỰNG HỆ THỐNG

## 2.1. Giới Thiệu Công Cụ Xây Dựng Hệ Thống

### 2.1.1. Giới thiệu IntelliJ

IntelliJ là 1 công cụ hỗ trợ lập trình mã nguồn mở được phát triển bởi JetBrains.

IntelliJ như một môi trường phát triển Java tích hợp (IDE), với IntelliJ chúng ta có thể mở rộng hơn mã nguồn bằng cách chèn thêm các plugins cho project (PDE- Plug-in Development Environment). Mặc dù IntelliJ được viết bằng ngôn ngữ lập trình Java, nhưng việc sử dụng nó không hạn chế chỉ cho ngôn ngữ Java. Ví dụ, Eclipse hỗ trợ sẵn hoặc có thể cài thêm các plugins để hỗ trợ cho các ngôn ngữ lập trình như C/C + + và COBOL. Ngoài ra, còn rất nhiều ngôn ngữ khác như PHP, Groovy, ...

IntelliJ còn hỗ trợ cho lập trình viên code theo các mô hình phát triển như MVC, tạo thêm các lib hỗ trợ phát triển phần mềm.

### 2.1.2. Giới thiệu Visual Studio Code

VSCode là một công cụ soạn thảo mã nguồn (Code Editor –  nôm na là chỗ để viết code) do Microsoft phát triển, được giới thiệu lần đầu năm 2015 và chính thức phát hành năm 2016. VSCode có thể cài đặt và sử dụng trên cả Windows, MacOS và Linux. VSCode là mã nguồn mở và nó hoàn toàn miễn phí.

Ưu Điểm của VS Code:

* Đa dạng ngôn ngữ lập trình giúp người dùng thỏa sức sáng tạo và sử dụng như HTML, CSS, JavaScript, C++,…
* Ngôn ngữ, giao diện tối giản, thân thiện, giúp các lập trình viên dễ dàng định hình nội dung.
* Các tiện ích mở rộng rất đa dạng và phong phú.
* Tích hợp các tính năng quan trọng như tính năng bảo mật (Git), khả năng tăng tốc xử lý vòng lặp (Debug),…
* Đơn giản hóa việc tìm quản lý hết tất cả các Code có trên hệ thống.

### 2.1.3. Giới thiệu về Microsoft SQL Server Management

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mạng máy tính hoạt động theo mô hình khách chủ cho phép đồng thời cùng lúc có nhiều người dùng truy xuất đến dữ liệu, quản lý việc truy nhập hợp lệ và các quyền hạn của từng người dùng trên mạng.

Microsoft SQL hỗ trợ sao lưu tự động dữ liệu qua JOB. Việc làm này có thể được tự động hóa một cách dễ dàng thông qua SQL Server Agent. Đây là module trong.

Microsoft SQL Server cung cấp các loại đồng bộ dữ liệu bao gồm Transactional replication, Merge replication và Snapshot replication nhằm mục đích đảm bảo an toàn dữ liệu và giải quyết bài toán nhiều người truy cập cùng một thời điểm. Giới thiệu về Spring Framework.

* + 1. **Giới thiệu PyCharm**

PyCharm là môi trường phát triển tích hợp đa nền tảng (IDE) được phát triển bởi Jet Brains và được thiết kế đặc biệt cho Python. PyCharm có mặt trên cả 3 nền tảng Windows, Linux và Mac OS.

Pycharm là một giải pháp phù hợp cho Python developers vì IDE này hỗ trợ nhiều extensions, môi trường ảo (Virtual Environment), nhiều tính năng thông minh như bộ code completion, tự động thụt lề, phát hiện văn bản trùng lặp và kiểm tra lỗi. Ngoài ra còn có các tính năng tìm kiếm mã nguồn thông minh để tìm kiếm từng từ một trong nháy mắt.

## Giới Thiệu Công Nghệ Xây Dựng Hệ Thống

### Tổng quan về Python

* + - 1. **Giới Thiệu về ngôn ngữ lập trình Python**

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng. Ngôn ngữ lập trình Python được tạo bởi Guido van Rossum và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

* + - 1. **Tính năng của Python**

Chúng ta có thể kể đến những tính năng của Python:

* **Ngôn ngữ lập trình đơn giản, dễ học*:*** Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.
* **Miễn phí, mã nguồn mở*:*** Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng cho mục đích thương mại. Vì là mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.
* **Khả năng di chuyển*:*** Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.
* **Khả năng mở rộng và có thể nhúng*:*** Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, bạn có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của bạn những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.
* **Ngôn ngữ thông dịch cấp cao*:*** Không giống như C/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Bạn không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào.
* **Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến*:*** Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code. Ví dụ: Bạn cần kết nối cơ sở dữ liệu MySQL trên Web server? Bạn có thể nhập thư viện MySQLdb và sử dụng nó. Những thư viện này được kiểm tra kỹ lưỡng và được sử dụng bởi hàng trăm người. Vì vậy, bạn có thể chắc chắn rằng nó sẽ không làm hỏng code hay ứng dụng của mình.
* **Hướng đối tượng*:*** Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.
  + 1. **Tổng quan về Java**

**2.2.2.1. Giới thiệu về Java**

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

Java được tạo ra với tiêu chí “**Viết (code) một lần, thực thi khắp nơi**” (Write Once, Run Anywhere  – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

**2.2.2.2. Đặc điểm của ngôn ngữ lập trình Java**

* **Tương tự C++, hướng đối tượng hoàn toàn**

Trong quá trình tạo ra một ngôn ngữ mới phục vụ cho mục đích chạy được trên nhiều nền tảng, các kỹ sư của Sun MicroSystem muốn tạo ra một ngôn ngữ dễ học và quen thuộc với đa số người lập trình. Vì vậy họ đã sử dụng lại các cú pháp của C và C++.

Tuy nhiên, trong Java thao tác với con trỏ bị lược bỏ nhằm đảo bảo tính an toàn và dễ sử dụng hơn. Các thao tác overload, goto hay các cấu trúc như struct và union cũng được loại bỏ khỏi Java.

* **Độc lập phần cứng và hệ điều hành**

Một chương trình viết bằng ngôn ngữ Java có thể chạy tốt ở nhiều môi trường khác nhau. Gọi là khả năng “cross-platform”. Khả năng độc lập phần cứng và hệ điều hành được thể hiện ở 2 cấp độ là cấp độ mã nguồn và cấp độ nhị phân.

Ở cấp độ mã nguồn: Kiểu dữ liệu trong Java nhất quán cho tất cả các hệ điều hành và phần cứng khác nhau. Java có riêng một bộ thư viện để hỗ trợ vấn đề này. Chương trình viết bằng ngôn ngữ Java có thể biên dịch trên nhiều loại máy khác nhau mà không gặp lỗi.

Ở cấp độ nhị phân: Một mã biên dịch có thể chạy trên nhiều nền tảng khác nhau mà không cần dịch lại mã nguồn. Tuy nhiên cần có Java Virtual Machine để thông dịch đoạn mã này.

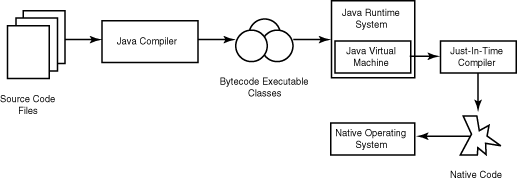
* **Ngôn ngữ thông dịch**

Ngôn ngữ lập trình thường được chia ra làm 2 loại (tùy theo các hiện thực hóa ngôn ngữ đó) là ngôn ngữ thông dịch và ngôn ngữ biên dịch.

* Thông dịch (Interpreter) : Nó dịch từng lệnh rồi chạy từng lệnh, lần sau muốn chạy lại thì phải dịch lại.
* Biên dịch (Compiler): Code sau khi được biên dịch sẽ tạo ra 1 file thường là .exe, và file .exe này có thể đem sử dụng lại không cần biên dịch nữa.

Ngôn ngữ lập trình Java thuộc loại ngôn ngữ thông dịch. Chính xác hơn, Java là loại ngôn ngữ vừa biên dịch vừa thông dịch. Cụ thể như sau

Khi viết mã, hệ thống tạo ra một tệp .java. Khi biên dịch mã nguồn của chương trình sẽ được biên dịch ra mã byte code. Máy ảo Java (Java Virtual Machine) sẽ thông dịch mã byte code này thành machine code  (hay native code) khi nhận được yêu cầu chạy chương trình.



**Hình 2 Luồng thực thi lập trình Java**

* + - Ưu điểm : Phương pháp này giúp các đoạn mã viết bằng Java có thể chạy được trên nhiều nền tảng khác nhau. Với điều kiện là JVM có hỗ trợ chạy trên nền tảng này.
    - Nhược điểm : Cũng như các ngôn ngữ thông dịch khác, quá trình chạy các đoạn mã Java là chậm hơn các ngôn ngữ biên dịch khác (tuy nhiên vẫn ở trong một mức chấp nhận được).
* **Cơ chế thu gom rác tự động**

Khi tạo ra các đối tượng trong Java, JRE sẽ tự động cấp phát không gian bộ nhớ cho các đối tượng ở trên heap.

Với ngôn ngữ như C \ C++, bạn sẽ phải yêu cầu hủy vùng nhớ mà bạn đã  cấp phát, để tránh việc thất thoát vùng nhớ. Tuy nhiên vì một lý do nào đó, bạn không hủy một vài vùng nhớ, dẫn đến việc thất thoát và làm giảm hiệu năng chương trình.

Ngôn ngữ lập trình Java hỗ trợ cho bạn điều đó, nghĩa là bạn không phải  tự gọi hủy các vùng nhớ. Bộ thu dọn rác của Java sẽ theo vết các tài nguyên đã được cấp. Khi không có tham chiếu nào đến vùng nhớ, bộ thu dọn rác sẽ tiến hành thu hồi vùng nhớ đã được cấp phát.

* **Đa luồng**

Java hỗ trợ lập trình đa tiến trình (multithread) để thực thi các công việc đồng thời. Đồng thời cũng cung cấp giải pháp đồng bộ giữa các tiến trình (giải pháp sử dụng priority…).

* **Tính an toàn và bảo mật**
* Tính an toàn

Ngôn ngữ lập trình Java yêu cầu chặt chẽ về kiểu dữ liệu.

Dữ liệu phải được khai báo tường minh.

Không sử dụng con trỏ và các phép toán với con trỏ.

Java kiểm soát chặt chẽ việc truy nhập đến mảng, chuỗi. Không cho phép sử dụng các kỹ thuật tràn. Do đó các truy nhập sẽ không vượt quá kích thước của mảng hoặc chuỗi.

Quá trình cấp phát và giải phóng bộ nhớ được thực hiện tự động.

Cơ chế xử lý lỗi giúp việc xử lý và phục hồi lỗi dễ dàng hơn.

* Tính bảo mật

Java cung cấp một môi trường quản lý chương trình với nhiều mức khác nhau.

Mức 1 : Chỉ có thể truy xuất dữ liệu cũng như phương phức thông qua giao diện mà lớp cung cấp.

Mức 2 : Trình biên dịch kiểm soát các đoạn mã sao cho tuân thủ các quy tắc của ngôn ngữ lập trình Java trước khi thông dịch.

Mức 3 : Trình thông dịch sẽ kiểm tra mã byte code xem các đoạn mã này có đảm bảo được các quy định, quy tắc trước khi thực thi.

Mức 4: Java kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống.

### Tổng quan về Spring Framework

Spring framework, hay ngắn hơn là Spring, là một cấu trúc dùng để xây dựng chương trình ứng dụng mã nguồn mở dành cho ngôn ngữ lập trình Java. Phiên bản đầu tiên của nó do Rod Johnson viết và đã được ông xuất bản cùng với quyển sách đầu tay Expert One-on-One JavaEE Design and Development (Nhà xuất bản Wrox Press, Tháng 10 năm 2002).

Kiến trúc của Spring framework được ra mắt công chúng lần đầu tiên hồi tháng 6 bnăm 2003 dưới Giấy phép Apache - phiên bản 2.0. Phiên bản 1.0 đánh dấu mốc thành đạt đầu tiên được xuất bản vào tháng 3 năm 2004 và tiếp đó vào tháng 9 năm 2004, tháng 3 năm 2005. Tuy Spring framework không bắt buộc người ta phải tuân theo một mô hình lập trình cụ thể nào, song nó lan truyền rộng rải trong cộng đồng những người viết chương trình dùng Java, như một hình thức chủ yếu thay thế cho mô hình Enterprise Java Bean.

Theo thiết kế, bộ framework này giải phóng lập trình viên dùng Java, cho phép họ nhiều quyền tự do hơn và đồng thời cung cấp một giải pháp tiện lợi, đầy đủ dẫn chứng bằng tài liệu, dễ dàng sử dụng, phù hợp với những thực hành thông dụng trong công nghệ phần mềm.

Bên cạnh những đặc trưng nền tảng của Spring framework là những cái có thể dùng được trong bất cứ một chương trình ứng dụng Java nào, rất nhiều các mở rộng và tiến bộ trong việc kiến tạo các trình ứng dụng dành cho nền tảng mạng web (web-based application) dựa trên nền Java Enterprise cũng tồn tại nữa. Spring framework nổi tiếng cũng một phần do chính đặc thù kể trên và được giới thương mại công nhận như một nền tảng kiến trúc có tầm quan trọng trong chiến lược kiến tạo phần mềm.

**Lợi ích khi dùng Spring Framework.**

* + - * Spring cho phép lập trình viên sử dụng POJOs. Việc sử dụng POJOs giúp bạn không phải làm việc với EJB, ứng dụng, các luồng chạy, cấu hình… đơn giản hơn rất nhiều.
      * Spring được tổ chức theo kiểu mô đun. Số lượng các gói và các lớp khá nhiều, nhưng bạn chỉ cần quan tâm đến những gì bạn cần và không cần quan tâm đến phần còn lại.
      * Spring hỗ trợ sử dụng khá nhiều công nghệ như ORM Framework, các logging framework, JEE, các thư viện tạo lịch trình (Quartz và JDK timer)…
      * Module Web của Spring được thiết kế theo mô hình MVC nên nó cung cấp đầy đủ các tính năng giúp thay thế các web framework khác như Struts.

### Tổng quan về Spring Boot.

Spring Boot là một dự án phát triển bởi JAV (ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng.

* + - 1. **Giai đoạn phát triển ứng dụng web.**

Theo thiết kế, bộ framework này giải phóng lập trình viên dùng Java, cho phép họ nhiều quyền tự do hơn và đồng thời cung cấp một giải pháp tiện lợi, đầy đủ dẫn chứng bằng tài liệu, dễ dàng sử dụng, phù hợp với những thực hành thông dụng trong công nghệ phần mềm.

* Giai đoạn 1: Tạo ra một project sử dụng Maven với các dependency cần thiết của Spring MVC và Servlet API.
* Giai đoạn 2: Một tập tin web.xml để khai báo DispatcherServlet của Spring MVC.
* Giai đoạn 3: Một tập tin có cấu hình của Spring MVC.
* Giai đoạn 4: Một class Controller trả về trang “Hello World” khi có request gửi đến.
* Giai đoạn 5: Cuối cùng, phải có một web server dùng triển khai ứng dụng lên để chạy.

Trong 5 công đoạn trên, chỉ có công đoạn tạo class Controller thì có thể khác cho các ứng dụng khác nhau. Bởi vì mỗi ứng dụng sẽ có một yêu cầu khác. Còn những công đoạn khác thì đều như nhau.

* + - 1. **Tính năng của Spring Boot.**

Một số tính năng nổi bật của Spring Boot đó là:

* Tạo ra các ứng dụng Spring mang tính độc lập.
* Nhúng trực tiếp Tomcat, Jetty hoặc Undertow mà không cần phải triển khai ra file WAR.
* Starter dependency giúp cho việc chạy cấu hình Maven trở nên đơn giản hơn.
* Tự động chạy cấu hình Spring khi cần thiết.
* Không sinh code cấu hình, đồng thời không yêu cầu phải cấu hình bằng XML.
  + - 1. **Đặc điểm của Spring Boot.**

Được phát triển tối ưu sao cho cấu hình XML trở nên đơn giản nhanh chóng và dễ dàng nhất trong Spring.

* Gia tăng được năng suất trong quá trình lập trình.
* Giảm ở mức tối thiểu thời gian lập trình.
* Giúp người dùng mặc dù không có nhiều kiến thức lập trình vẫn có thể xây dựng được một ứng dụng

Một số tính năng rất quan trọng của Spring Boot là:

* SpringApplication: Khi lập trình xong và bạn chỉ muốn chạy thử nghiệm thì nên làm thế nào? Bạn chỉ cần gọi run() là được, vì Spring Boot được thiết kế theo dạng “just run”. Giúp cho các lập trình viên chỉ cần cấu hình ít Spring nhất, phần còn lại Spring Boot sẽ lo liệu.
* Externalized Configuration: Bạn mong muốn tạo nên một ứng dụng có thể chạy trên nhiều loại môi trường khác nhau? Spring Boot sẽ giúp bạn config cấu hình từ ngoài và ứng dụng của bạn sẽ có thể chạy thoải mái.
* Profiles: Nếu có nhiều config khác nhau, bạn có thể sử dụng Profile để phân chia từng loại cho từng môi trường để dễ dàng quản lý hơn.
* Logging: Tính năng này được sử dụng cho toàn bộ chức năng log trong phạm vi nội bộ và nó được quản lý mặc định.

Bên cạnh những tính năng quan trọng nói trên, còn một số tính năng tương tự khác như: Security, Messaging, Developing web Applications, Working with SQL Technologies, Caching, Sending Email, Validation, Calling rest Services with RestTemplate/WebClient,…

### Tổng quan về Học máy.

Học máy (Machine Learning) là một ngành khoa học nghiên cứu các thuật toán cho phép máy tính có thể học được các khái niệm (concept).

Phân loại: Có hai loại phương pháp học máy chính

* Phương pháp quy nạp: Máy học/phân biệt các khái niệm dựa trên dữ liệu đã thu thập được trước đó. Phương pháp này cho phép tận dụng được nguồn dữ liệu rất nhiều và sẵn có.
* Phương pháp suy diễn: Máy học/phân biệt các khái niệm dựa vào các luật. Phương pháp này cho phép tận dụng được các kiến thức chuyên ngành để hỗ trợ máy tính.

Hiện nay, các thuật toán đều cố gắng tận dụng được ưu điểm của hai phương pháp này. Các ngành khoa học liên quan:

* Lý thuyết thống kê: các kết quả trong xác suất thống kê là tiền đề cho rất nhiều phương pháp học máy. Đặc biệt, lý thuyết thống kê cho phép ước lượng sai số của các phương pháp học máy.
* Các phương pháp tính: các thuật toán học máy thường sử dụng các tính toán số thực/số nguyên trên dữ liệu rất lớn. Trong đó, các bài toán như: tối ưu có/không ràng buộc, giải phương trình tuyến tính v.v… được sử dụng rất phổ biến.
* Khoa học máy tính: là cơ sở để thiết kế các thuật toán, đồng thời đánh giá thời gian chạy, bộ nhớ của các thuật toán học máy.

Các nhóm giải thuật học máy:

* Học có giám sát: Máy tính được xem một số mẫu gồm đầu vào (input) và đầu ra (output) tương ứng trước. Sau khi học xong các mẫu này, máy tính quan sát một đầu vào mới và cho ra kết quả.
* Học không giám sát: Máy tính chỉ được xem các mẫu không có đầu ra, sau đó máy tính phải tự tìm cách phân loại các mẫu này và các mẫu mới.
* Học nửa giám sát: Một dạng lai giữa hai nhóm giải thuật trên.
* Học tăng cường: Máy tính đưa ra quyết định hành động (action) và nhận kết quả phản hồi (response/reward) từ môi trường (environment). Sau đó máy tính tìm cách chỉnh sửa cách ra quyết định hành động của mình.

Ứng dụng của học máy: Học máy có ứng dụng rộng khắp trong các ngành khoa học/sản xuất, đặc biệt những ngành cần phân tích khối lượng dữ liệu khổng lồ. Một số ứng dụng thường thấy

* Xử lý ngôn ngữ tự nhiên (Natural Language Processing): xử lý văn bản, giao tiếp người – máy, …
* Nhận dạng (Pattern Recognition): nhận dạng tiếng nói, chữ viết tay, vân tay, thị giác máy (Computer Vision) …
* Tìm kiếm (Search Engine)
* Chẩn đoán trong y tế: phân tích ảnh X-quang, các hệ chuyên gia chẩn đoán tự động.
* Tin sinh học: phân loại chuỗi gene, quá trình hình thành gene/protein
* Vật lý: phân tích ảnh thiên văn, tác động giữa các hạt …
* Phát hiện gian lận tài chính (financial fraud): gian lận thẻ tỉn dụng
* Phân tích thị trường chứng khoán (stock market analysis)
* Chơi trò chơi: tự động chơi cờ, hành động của các nhân vật ảo
* Robot: là tổng hợp của rất nhiều ngành khoa học, trong đó học máy tạo nên hệ thần kinh/bộ não của người máy.

### Tổng quan về RESTful API

REST là viết tắt của Representational State Transfer. RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API

### Tổng quan về ReactJS

React (Hay ReactJS, React.js) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi Jordan Walke, một kỹ sư phần mềm tại Facebook. Người bị ảnh hưởng bởi XHP (Một nền tảng thành phần HTML cho PHP). React lần đầu tiên được triển khai cho ứng dụng Newsfeed của Facebook năm 2011, sau đó được triển khai cho Instagram.com năm 2012. Nó được mở mã nguồn (open-sourced) tại JSConf US tháng 5 năm 2013.

**Tính năng React**

* **JSX** – JSX là phần mở rộng cú pháp JavaScript. Không nhất thiết phải sử dụng JSX trong phát triển React, nhưng nó được khuyến khích.
* **Các thành phần** – React là tất cả về các thành phần. Bạn cần nghĩ mọi thứ như một thành phần. Điều này sẽ giúp bạn duy trì mã khi làm việc trên các dự án quy mô lớn hơn.
* **Luồng dữ liệu một chiều và Flux** – React triển khai luồng dữ liệu một chiều giúp bạn dễ dàng suy luận về ứng dụng của mình. Flux là một mẫu giúp giữ dữ liệu của bạn không theo hướng.
* **Giấy phép** – React được cấp phép theo Facebook Inc. Tài liệu được cấp phép theo CC BY 4.0.

**Ưu điểm của React**

* Sử dụng DOM ảo là một đối tượng JavaScript. Điều này sẽ cải thiện hiệu suất ứng dụng, vì DOM ảo JavaScript nhanh hơn DOM thông thường.
* Có thể được sử dụng trên máy khách và máy chủ cũng như với các khuôn khổ khác.
* Các mẫu thành phần và dữ liệu cải thiện khả năng đọc, giúp duy trì các ứng dụng lớn hơn.

**Giới hạn phản ứng**

* Chỉ bao gồm lớp xem của ứng dụng, do đó bạn vẫn cần chọn các công nghệ khác để có được bộ công cụ hoàn chỉnh để phát triển.
* Sử dụng khuôn mẫu nội tuyến và JSX, điều này có vẻ khó xử đối với một số nhà phát triển.

# CHƯƠNG 3: CƠ SỞ LÍ THUYẾT

* 1. **Giới thiệu về xử lí ngôn ngữ tự nhiên**

Xử lý ngôn ngữ tự nhiên là một nhánh của Trí tuệ nhân tạo, tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người, dưới dạng tiếng nói (speech) hoặc văn bản (text). Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

Xử lý ngôn ngữ tự nhiên ra đời từ những năm 40 của thế kỷ 20, trải qua các giai đoạn phát triển với nhiều phương pháp và mô hình xử lý khác nhau. Có thể kể tới các phương pháp sử dụng ô-tô-mát và mô hình xác suất (những năm 50), các phương pháp dựa trên ký hiệu, các phương pháp ngẫu nhiên (những năm 70), các phương pháp sử dụng học máy truyền thống (những năm đầu thế kỷ 21), và đặc biệt là sự bùng nổ của học sâu trong thập kỷ vừa qua.

Xử lý ngôn ngữ tự nhiên có thể được chia ra thành hai nhánh lớn, không hoàn toàn độc lập, bao gồm xử lý tiếng nói (speech processing) và xử lý văn bản (text processing). Xử lý tiếng nói tập trung nghiên cứu, phát triển các thuật toán, chương trình máy tính xử lý ngôn ngữ của con người ở dạng tiếng nói (dữ liệu âm thanh). Các ứng dụng quan trọng của xử lý tiếng nói bao gồm nhận dạng tiếng nói và tổng hợp tiếng nói. Nếu như nhận dạng tiếng nói là chuyển ngôn ngữ từ dạng tiếng nói sang dạng văn bản thì ngược lại, tổng hợp tiếng nói chuyển ngôn ngữ từ dạng văn bản thành tiếng nói. Xử lý văn bản tập trung vào phân tích dữ liệu văn bản. Các ứng dụng quan trọng của xử lý văn bản bao gồm tìm kiếm và truy xuất thông tin, dịch máy, tóm tắt văn bản tự động, hay kiểm lỗi chính tả tự động. Xử lý văn bản đôi khi được chia tiếp thành hai nhánh nhỏ hơn bao gồm hiểu văn bản và sinh văn bản. Nếu như hiểu liên quan tới các bài toán phân tích văn bản thì sinh liên quan tới nhiệm vụ tạo ra văn bản mới như trong các ứng dụng về dịch máy hoặc tóm tắt văn bản tự động.

Xử lý văn bản bao gồm 4 bước chính sau:

* **Phân tích hình vị:** là sự nhận biết, phân tích, và miêu tả cấu trúc của hình vị trong một ngôn ngữ cho trước và các đơn vị ngôn ngữ khác, như từ gốc, biên từ, phụ tố, từ loại, v.v. Trong xử lý tiếng Việt, hai bài toán điển hình trong phần này là tách từ (word segmentation) và gán nhãn từ loại (part-of-speech tagging).
* **Phân tích cú pháp:** là quy trình phân tích một chuỗi các biểu tượng, ở dạng ngôn ngữ tự nhiên hoặc ngôn ngữ máy tính, tuân theo văn phạm hình thức. Văn phạm hình thức thường dùng trong phân tích cú pháp của ngôn ngữ tự nhiên bao gồm Văn phạm phi ngữ cảnh (Context-free grammar – CFG), Văn phạm danh mục kết nối (Combinatory categorial grammar – CCG), và Văn phạm phụ thuộc (Dependency grammar – DG). Đầu vào của quá trình phân tích là một câu gồm một chuỗi từ và nhãn từ loại của chúng, và đầu ra là một cây phân tích thể hiện cấu trúc cú pháp của câu đó.
* **Phân tích ngữ nghĩa:** là quá trình liên hệ cấu trúc ngữ nghĩa, từ cấp độ cụm từ, mệnh đề, câu và đoạn đến cấp độ toàn bài viết, với ý nghĩa độc lập của chúng. Nói cách khác, việc này nhằm tìm ra ngữ nghĩa của đầu vào ngôn từ. Phân tích ngữ nghĩa bao gồm hai mức độ: Ngữ nghĩa từ vựng biểu hiện các ý nghĩa của những từ thành phần, và phân biệt nghĩa của từ; Ngữ nghĩa thành phần liên quan đến cách thức các từ liên kết để hình thành những nghĩa rộng hơn.
* **Phân tích diễn ngôn:** là phân tích văn bản có xét tới mối quan hệ giữa ngôn ngữ và ngữ cảnh sử dụng (context-of-use). Phân tích diễn ngôn, do đó, được thực hiện ở mức độ đoạn văn hoặc toàn bộ văn bản thay vì chỉ phân tích riêng ở mức câu.
  1. **Giới thiệu về cơ sở tri thức**
     1. **Khái niệm**

Hệ cơ sở tri thức (CSTT) là chương trình máy tính được thiết kế để mô hình hóa khả năng giải quyết vấn đề của chuyên gia con người.

Hệ CSTT là hệ thống dựa trên tri thức, cho phép mô hình hóa các tri thức của chuyên gia, dùng tri thức này để giải quyết vấn đề phức tạp thuộc cùng lĩnh vực.

Hai yếu tố quan trọng trong hệ CSTT là: tri thức chuyên gia và lập luận, tương ứng với hệ thống có hai khối chính là CSTT và động cơ suy diễn.

Hệ Chuyên gia là một loại cơ sở tri thức được thiết kế cho một lĩnh vực ứng dụng cụ thể. Hệ Chuyên gia làm việc như một chuyên gia thực thụ và cung cấp các ý kiến dựa trên kinh nghiệm của chuyên gia con người đã được đưa vào Hệ Chuyên gia.

Cơ sở tri thức: chứa các tri thức chuyên sâu về lĩnh vực như chuyên gia. Cơ sở tri thức bao gồm: các sự kiện, các luật, các khái niệm và các quan hệ.

Động cơ suy diễn: bộ xử lý tri thức theo mô hình hóa theo cách lập luận của chuyên gia. Động cơ hoạt động trên thông tin về vấn đề đang xét, so sánh với tri thức lưu trong cơ sở tri thức rồi rút ra kết luận.

Kỹ sư tri thức (Knowledge Engineer): người thiết kế, xây dựng và thử nghiệm Hệ Chuyên gia.

* + 1. **Cấu trúc của hệ cơ sở tri thức**
* Giao diện người dùng (User Interface): thực hiện giao tiếp giữa Hệ Chuyên gia và User. Nhận các thông tin từ User và đưa ra các câu trả lời.
* Bộ giải thích (Explanation system): Giải thích các hoạt động khi có yêu cầu.
* Động cơ suy diễn (Inference Engine): Quá trình trong Hệ Chuyên gia cho phép khớp các sự kiện trong vùng nhớ làm việc với các tri thức về lĩnh vực trong cơ sở tri thức, để rút ra các kết luận về vấn đề đang giải quyết.
* Bộ tiếp nhận tri thức (Knowledge editor): Làm nhiệm vụ thu nhận tri thức từ chuyên gia con người (human expert), từ kỹ sư tri thức và User thông qua các yêu cầu và lưu trữ vào cơ sở tri thức.
* Cơ sở tri thức: Lưu trữ, biểu diễn các tri thức mà hệ đảm nhận, làm cơ sở cho các hoạt động của hệ. Cơ sở tri thức bao gồm các sự kiện (facts) và các luật (rules).
* Vùng nhớ làm việc (Working memory): một phần của Hệ Chuyên gia chứa các sự kiện của vấn đề đang xét.
  + 1. **Các Đặc tính của hệ cơ sở tri thức**

CSTT là hệ thống lưu trữ dựa trên tri thức của chuyên gia, tập trung nhiều nguồn dữ liệu từ các chuyên gia.

CSTT là hệ thống lập luận xử lý trên logic đã được thiết lập, có thể tính toán được rủi ro cũng như lập luận xử lý với các thông tin không đầy đủ.

CSTT chứa khả năng suy diễn, mức độ sâu sắc cao, có khả năng tự kiểm tra, cũng như khả năng tự học trong suốt quá trình suy luận.

* + 1. **Các hệ thống liên quan**
* Hệ hỗ trợ ra quyết định

Khái niệm hệ hỗ trợ ra quyết định được đề xuất bởi Michael S. Scott Morton vào những năm 1970. Hệ hỗ trợ ra quyết định có:

Phần mềm máy tính; Chức năng hỗ trợ ra quyết định; Làm việc với các bài toán có cấu trúc yếu; Hoạt động theo cách tương tác với người dùng; Được trang bị nhiều mô hình phân tích và mô hình dữ liệu.

* Hệ học
* Hệ điều khiển mờ
  + 1. **Ứng dụng**
* Diễn giải (Interpretation)
* Dự báo (Prediction)
* Thiết kế (Design)
* Chẩn đoán (Diagnosis)
* Lên kế hoạch (Planning)
* Dẫn dắt (Monotoring)
* Gỡ rối (Debugging)
* Giảng dạy (Instruction)
* Điều khiển (Control)

### Quá trình xây dựng cơ sở tri thức cho hệ thống học sâu về cảm xúc mà âm nhạc mang lại cho người xem

Để xây dựng một cơ sở tri thức gồm 5 bước:

Bước 1: Xác định mục đích của cơ sở tri thức

Cần xác định mục đích của CSTT cho hệ thống học sâu là để phục vụ cho việc nghiên cứu cảm xúc mà âm nhạc ảnh mang lại cho người nghe và liệt kê một số mục tiêu cụ thể nó cần đạt được. Ví dụ, cơ sở tri thức cho hệ thống học sâu cần phải:

* Cung cấp những câu trả lời cho các câu hỏi về cảm nhận sau khi nghe nhạc của người nghe
* Cung cấp những bài viết chuyên sâu về mảng đánh giá nhận xét khách quan về những bài hát.
* Cung cấp thông tin về những bài hát.
* Dù mục đích thế nào, CSTT của vẫn phải đáp ứng những gì mà người dùng mong đợi.

Bước 2: Tạo lập nội dung cho cơ sở tri thức

Để tạo lập nội dung cho cơ sở tri thức trước tiên cần tham khảo về các chuyên gia. Tổ chức các cuộc họp với mọi người trong các vai trò người dùng tin tưởng khác nhau (các nhà phê bình âm nhạc, báo chí, ca sĩ, nghệ sĩ …) để hỏi họ về cảm nhận, đánh giá về những gì mà bài hát mang lại. Với sự giúp đỡ của chuyên gia, các câu hỏi truy vấn của người dùng tin tưởng sẽ nhận được câu trả lời. Những câu trả lời của chuyên gia cho danh sách các cảm nhận tạo thành nội dung đầu tiên cho cơ sở tri thức.

Đây là một quá trình có thể lặp đi lặp lại để liên tục làm phong phú CSTT cho hệ thống.

Bước 3: Xác định cấu trúc nội dung của cơ sở tri thức

Một CSTT đòi hỏi phải có cấu trúc nội dung chặt chẽ. Nếu không người dùng sẽ không thể tìm thấy được thứ mà họ cần một cách nhanh chóng. Cấu trúc rõ ràng và sắp xếp tất cả thông tin đã có là chìa khóa thành công cho một CSTT.

Cần sử dụng những hiểu biết về đặc điểm người dùng tin để thiết kế một cấu trúc phù hợp và hữu ích nhất cho họ. Bằng cách xác định nhu cầu của người dùng tin, ta sẽ bắt đầu phát triển kiến trúc thông tin cơ bản và cấu trúc phân loại cho CSTT. Đảm bảo rằng các nội dung của CSTT được cấu trúc một cách khoa học, rõ ràng và phù hợp với nhu cầu của khách hàng.

Cấu trúc nội dung của CSTT dựa vào phân loại các nội dung đã có.

Bước 4: Thu thập thông tin phản hồi

Khi CSTT đi vào hoạt động, đôi khi mọi thứ có thể xảy ra. Một số nhận xét đánh giá có thể không tốt như những cái khác, cấu trúc có thể gây nhầm lẫn cho một số khách hàng, … Vì vậy, để giữ cho CSTT luôn phù hợp và hữu ích, cần phải thu thập và phân tích các thông tin phản hồi từ khách hàng.

Bước 5: Cập nhật thường xuyên

Để giữ cho CSTT hữu ích về lâu dài, cần thường xuyên cập nhật CSTT để kịp thời có những thông tin phù hợp đáp ứng nhu cầu của người dùng.

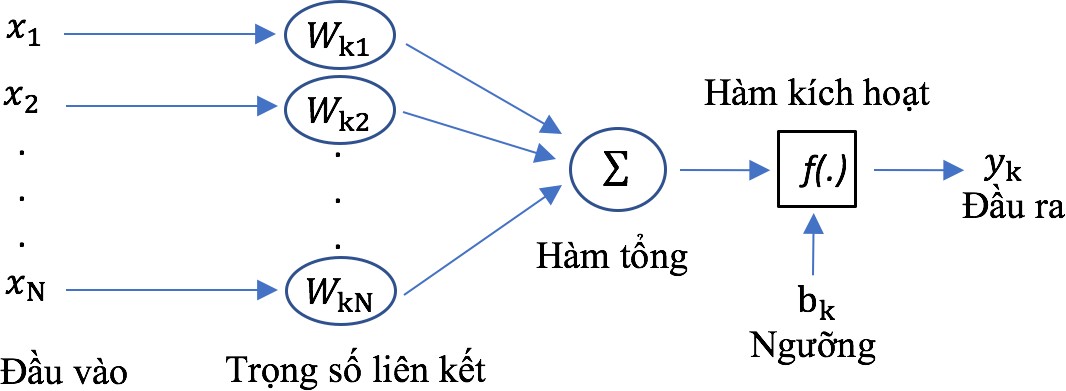
## Tổng quan về Deep Learning

* + 1. **Deep Learning là gì**

Deep Learning là một kỹ thuật máy học (Machine Learning) mạnh mẽ đang được nhiều người trong ngành biết đến và nghiên cứu. Với khả năng biểu diễn thông tin (represent problem/feature engineering) và học (learning). Bên cạnh các lĩnh vực đã gặt hái được nhiều thành công như xử lý ảnh số và video số, hay xử lý tiếng nói, và được áp dụng vào trong xử lý ngôn ngữ tự nhiên.

### Mạng neural nhân tạo

Mạng neural nhân tạo (hay còn gọi là mạng thần kinh nhân tạo) Artificial Neural Network (ANN) là một hệ thống các chương trình và cấu trúc dữ liệu mô phỏng cách xử lý thông tin của mạng neural sinh học. Nó được tạo nên từ một số lượng lớn các phần tử neural kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng neural nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu, ...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các neural.

Cấu tạo neural nhân tạo:

**Hình 3 : Mô hình cấu tạo một neural**

### Các mạng neural trong xử lý ngôn ngữ tự nhiên

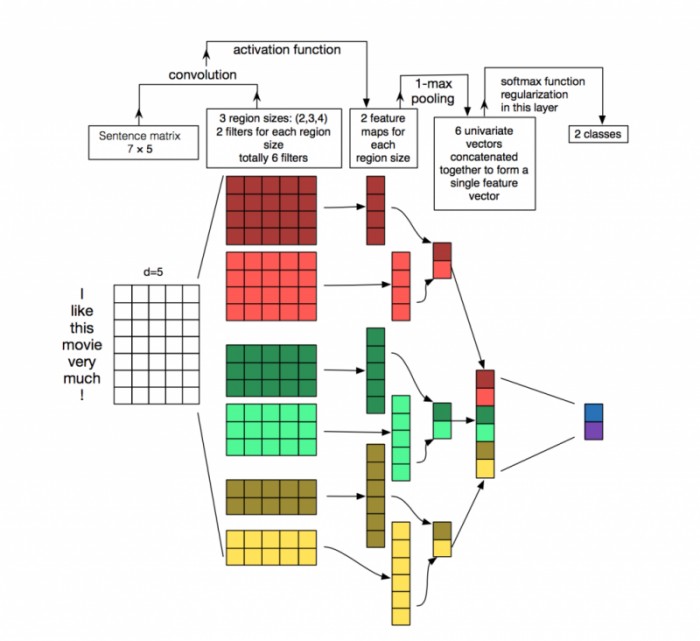
* + - 1. **Convolutional Neural Networks (CNNs)**
         1. **Định nghĩa**

Convolutional Neural Network (Mạng neural tích chập) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. Thường được sử dụng trong tín hiệu số (Signal Processing), phân lớp ảnh (Image Classification).

* + - * 1. ***Mô hình mạng neural tích chập (CNNs)***

**Mô hình mạng neural tích chập cho xữ lý ngôn ngữ tự nhiên**

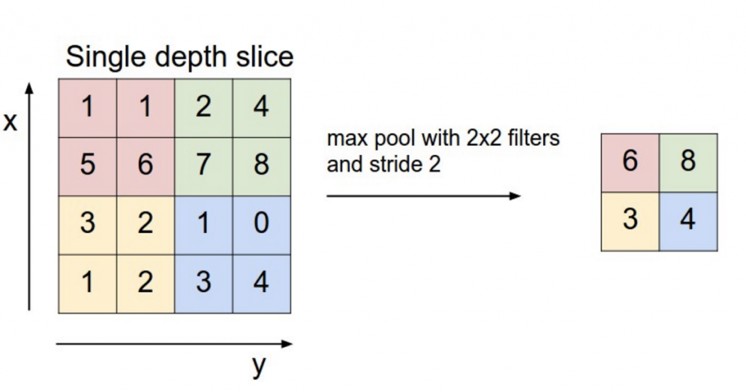
Thay vì sử dụng các pixel như trong xử lý ảnh, trong xử lý ngôn ngữ tự nhiên (NLP), ta biến đổi các câu hay văn bản thành một ma trận đầu vào. Mỗi dòng của ma trận tương ứng với một token (một từ trong câu, nhưng cũng có thể là một ký tự – character). Nghĩa là, mỗi dòng là một vector đại diện cho một từ. Giả sử ta có một câu gồm 10 từ được biểu diễn thành word embedding vector gồm 100 chiều, ta sẽ có đầu vào là một ma trận 10×100, tương ứng trong xử lý ảnh, ta có một bức ảnh 10×100 pixel.



**Hình 4: Minh hoạ mạng neural CNN cho xữ lý ngôn ngữ tự nhiên**

Duyệt qua từng từ một. Nghĩa là, ta sẽ duyệt qua từng dòng của ma trận đầu vào. Khi đó, chiều rộng của filter sẽ bằng với chiều rộng của ma trận đầu vào (trong ví dụ trên, chiều rộng bằng 10). Chiều cao (region size) của filter có thể tùy ý thay đổi sao cho phù hợp, thường là từ 2 đến 5 từ (như 2 đến 5-grams trong n-grams).

Quan sát hình 4, ta có đầu vào là một câu gồm 7 từ (kể cả ký tự!) “I like this movie very much!”. Các từ này được chuyển đổi thành word embedding có số chiều d = 5. Khi đó, ta có một ma trận đầu vào là 7×5. Ở layer tiếp theo, ta sử dụng các filter với region size khác nhau gồm 2, 3, và 4. Gấp đôi số lượng từng dạng filter này lên, ta có tổng cộng 6 filter. Tiếp đến, ta thực hiện nhân tích chập tương ứng với từng filter và cho qua activate function (softmax hay tanh) để thu được các vector đầu ra. Sau đó, ta sẽ cho các vector này đi qua filter max-pooling (average-pooling hay các

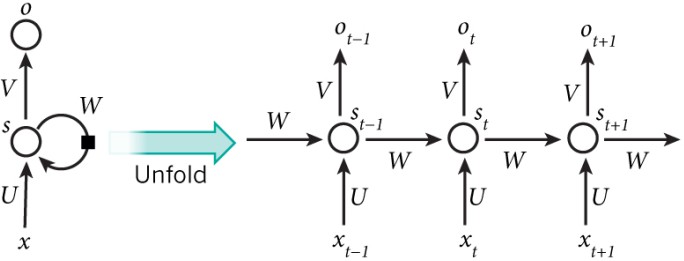
phép toán tương tự khác), mục đích chính là để chuẩn hóa đầu ra các vector sao cho chúng có số chiều là như nhau. Ở layer cuối cùng, tùy theo bài toán của chúng ta là Phan loại văn bản (Text Classification), Phân tích cảm xúc (Sentiment Analysis), phát hiện spam (Spam Detection), Phân loại đề tài (Topic Categorization) mà ta sẽ giới hạn số chiều của vector phân lớp này thành 2 hay nhiều chiều tương ứng

**Hình 5: Minh hoạ ma trận max-pooling trong CNN**

CNNs phù hợp với các tác vụ phân lớp như Phân tích cảm xúc (Sentiment Analysis), Phát hiện spam (Spam Detection) hay Phân loại chủ đề (Topic Categorization). Các phép toán convolution và max-pooling làm thất thoát thông tin về thứ tự cục bộ giữa các từ (local order of words). Do đó, các tác vụ liên quan đến gán nhãn cho chuỗi (sequence tagging) như POS Tagging, Entity Extraction sẽ không phù hợp khi sử dụng kiến trúc của CNNs.

## Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (hay còn gọi là mạng neural tái phát) là một trong những mô hình Deep Learning được đánh giá có nhiều ưu điểm trong các tác vụ xử lý ngôn ngữ tự nhiên. và các ứng dụng nhận diện giọng nói, các chuỗi thông tin đóng vai trò rất quan trọng. Ví dụ, một từ xuất hiện trong câu sẽ phụ thuộc vào từ trước đó. Kiến trúc recurrent neural network giải quyết vấn đề này được minh họa:



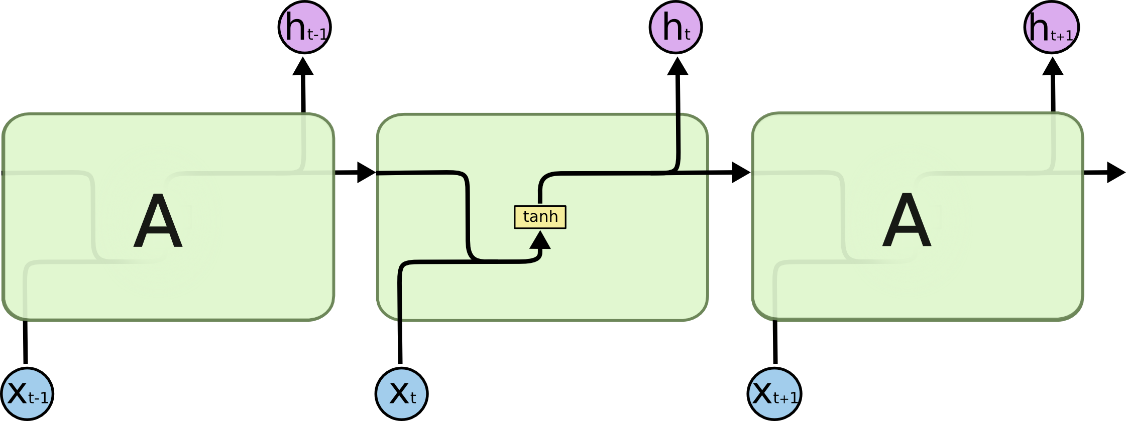
**Hình 6: Mô hình mạng neural RNN**

Recurrent có nghĩa là thực hiện lặp lại cùng một tác vụ cho mỗi thành phần trong chuỗi. Trong đó, kết quả đầu ra tại thời điểm hiện tại phụ thuộc vào kết quả tính toán của các thành phần ở những thời điểm trước đó. Nói cách khác, RNNs là một mô hình có trí nhớ (memory), có khả năng nhớ được thông tin đã tính toán trước đó. Không như các mô hình Neural Network truyền thống đó là thông tin đầu vào (input) hoàn toàn độc lập với thông tin đầu ra (output). Về lý thuyết, RNNs có thể nhớ được thông tin của chuỗi có chiều dài bất kì, nhưng trong thực tế mô hình này chỉ nhớ được thông tin ở vài bước trước đó.

* + - 1. **Long short-term memory (LSTM) có trí nhớ (memory)**

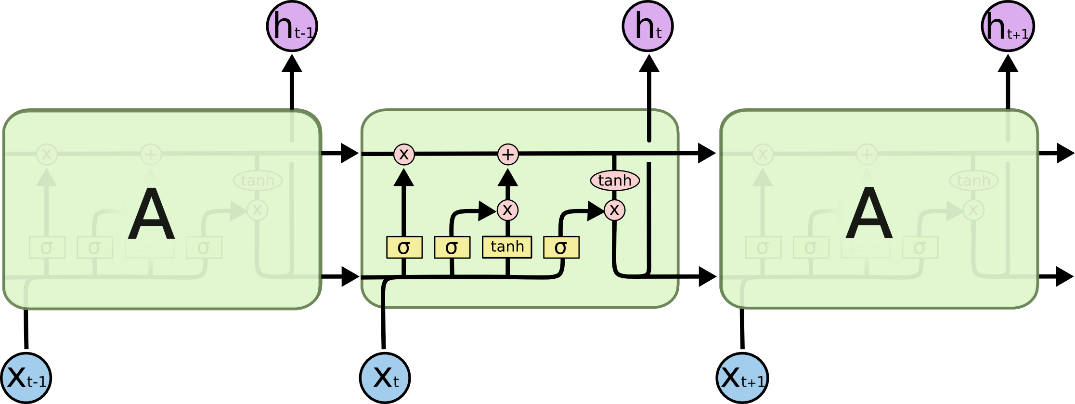
Mạng *trí nhớ ngắn hạn định hướng dài hạn* còn được viết tắt là LSTM làm một kiến trúc đặc biệt của RNN có khả năng học được sự phục thuộc trong dài hạn (*long-term dependencies*) được giới thiệu bởi [Hochreiter & Schmidhuber (1997)](http://www.bioinf.jku.at/publications/older/2604.pdf). Kiến trúc này đã được phổ biến và sử dụng rộng rãi cho tới ngày nay. LSTM đã tỏ ra khắc phục được rất nhiều những hạn chế của RNN trước đây về triệt tiêu đạo hàm. Tuy nhiên cấu trúc của chúng có phần phức tạp hơn mặc dù vẫn dữ được tư tưởng chính của RNN là sự sao chép các kiến trúc theo dạng chuỗi.

Một mạng RNN tiêu chuẩn sẽ có kiến trúc rất đơn giản chẳng hạn như đối với kiến trúc gồm một tầng ẩn là hàm tanh như bên dưới.



**Hình 7: Sự lặp lại kiến trúc module trong mạng RNN chứa một tầng**

LSTM cũng có một chuỗi dạng như thế nhưng phần kiến trúc lặp lại có cấu trúc khác biệt hơn. Thay vì chỉ có một tầng đơn, chúng có tới 4 tầng ẩn (3 sigmoid và 1 tanh) tương tác với nhau theo một cấu trúc đặc biệt.



**Hình 8: Sự lặp lại kiến trúc module trong mạng LSTM chứa 4 tầng ẩn (3 sigmoid và 1 tanh) tương tác**

Các kí hiệu có thể diễn giải như sau:

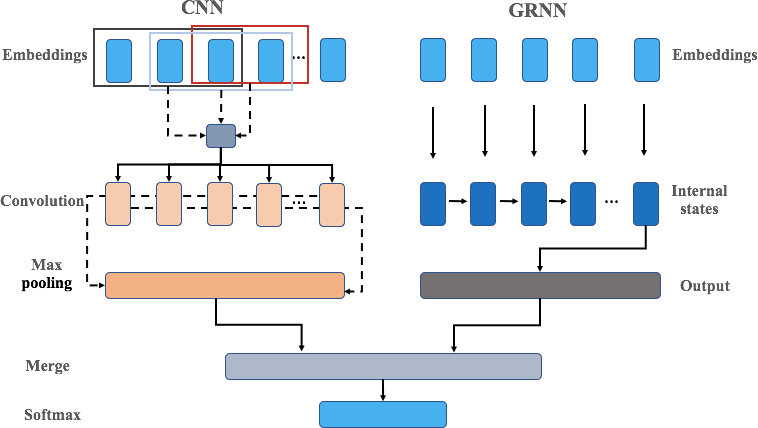


**Hình 9: Diễn giải các kí hiệu trong đồ thị mạng nơ ron (áp dụng chung cho toàn bộ bài)**

Trong sở đồ tính toán trên, mỗi một phép tính sẽ triển khai trên một véc tơ. Trong đó hình tròn màu hồng biểu diễn một toán tử đối với véc tơ như phép cộng véc tơ, phép nhân vô hướng các véc tơ. Màu vàng thể hiện hàm activation mà mạng nơ ron sử dụng để học trong tầng ẩn, thông thường là các hàm phi tuyến sigmoid và tanh. Kí hiệu 2 đường thẳng nhập vào thể hiện phép chập kết quả trong khi kí hiệu 2 đường thẳng rẽ nhánh thể hiện cho nội dung véc tơ trước đó được sao chép để đi tới một phần khác của mạng nơ ron.

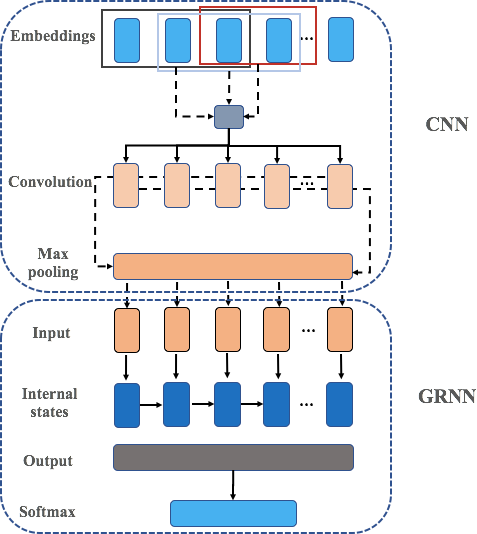
* + - 1. **Kiến trúc mạng Neural được xem xét nghiên cứu**

Mô hình mạng được trình bày trong hình 9 bao gồm việc thiết kế hai mạng neural được chạy song song với nhau, và cuối cùng kết quả của hai mạng được gộp lại với nhau để cho ra 1 softmax, là tập dùng cho việc dự đoán các kết quả tiếp theo.



**Hình 10: Kiến trúc mạng Neural [1]**

Và ở hình 10, mô hình này được từ ý tưởng liên kết (combine) hai mô hình mạng CNN và RNN. Thông qua việc đưa kết quả max-pooling của mạng CNN thành input cho mạng RNN, và output sẽ chuyển thành softmax để dự đoán cho các dữ liệu kiểm thử. Như vậy thì kết quả của CNN sẽ được chuyển tiếp cho RNN xử lý.



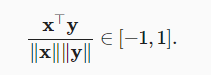
**Hình 11: Kiến trúc mạng Neural [2]**

## Tổng quan về word2vec

Ngôn ngữ tự nhiên là một hệ thống phức tạp mà con người sử dụng để diễn đạt ngữ nghĩa. Trong hệ thống này, từ là đơn vị cơ bản của ngữ nghĩa. Như tên gọi của nó, một vector từ (*word vector*) là một vector được sử dụng để biểu diễn một từ. Vector từ cũng có thể được xem là vector đặc trưng của một từ. Kỹ thuật ánh xạ từ ngữ sang vector số thực còn được gọi là kỹ thuật embedding từ (*word embedding*). Trong vài năm gần đây, embedding từ dần trở thành kiến thức cơ bản trong xử lý ngôn ngữ tự nhiên.

Chúng ta sử dụng vector one-hot để đại diện cho từ (thực chất là ký tự). Nhớ lại rằng khi giả sử số lượng các từ riêng biệt trong từ điển (tức kích thước từ điển) là N, mỗi từ có thể tương ứng một-một với các số nguyên liên tiếp từ 0 đến N−1, được gọi là chỉ số của từ. Giả sử chỉ số của một từ là i. Để thu được biểu diễn vector one-hot của từ đó, ta tạo một vector có N phần tử có giá trị là 0 và đặt phần tử thứ i bằng 1. Theo đó, mỗi từ được biểu diễn dưới dạng vector có độ dài N có thể được trực tiếp đưa vào mạng nơ-ron.

Mặc dù rất dễ xây dựng các vector one-hot, nhưng chúng thường không phải là lựa chọn tốt. Một trong những lý do chính là các vector one-hot không thể biểu diễn một cách chính xác độ tương tự giữa các từ khác nhau, chẳng hạn như độ tương tự cô-sin mà ta thường sử dụng. Độ tương tự cô-sin của hai vectors x,y∈ là giá trị cô-sin của góc giữa chúng:



Do độ tương tự cô-sin giữa các vector one-hot của bất kỳ hai từ khác nhau nào đều bằng 0, nên rất khó sử dụng vector one-hot để biểu diễn độ tương tự giữa các từ khác nhau.

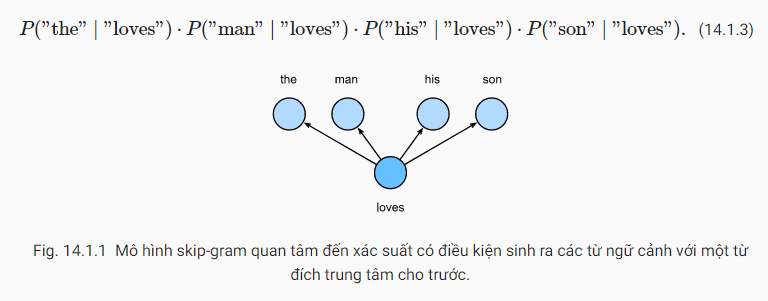
[**Word2vec**](https://code.google.com/archive/p/word2vec/) là một công cụ được phát minh để giải quyết vấn đề trên. Nó biểu diễn mỗi từ bằng một vector có độ dài cố định và sử dụng những vector này để biểu thị tốt hơn độ tương tự và và các quan hệ loại suy (*analogy relationship*) giữa các từ. Công cụ Word2vec gồm hai mô hình: skip-gam  và túi từ liên tục (*continuous bag of words* – CBOW) . Tiếp theo, ta sẽ xem xét hai mô hình này và phương pháp huấn luyện chúng.

* + 1. **Mô Hình Skip-Gram**

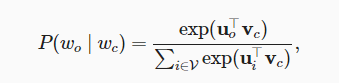
Mô hình skip-gam giả định rằng một từ có thể được sử dụng để sinh ra các từ xung quanh nó trong một chuỗi văn bản. Ví dụ, giả sử chuỗi văn bản là “the”, “man”, “loves”, “his” và “son”. Ta sử dụng “loves” làm từ đích trung tâm và đặt kích thước cửa sổ ngữ cảnh bằng 2. Như mô tả , với từ đích trung tâm “loves”, mô hình skip-gram quan tâm đến xác suất có điều kiện sinh ra các từ ngữ cảnh (“the”, “man”, “his” và “son”) nằm trong khoảng cách không quá 2 từ:



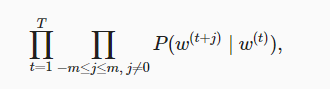
Ta giả định rằng, với từ đích trung tâm cho trước, các từ ngữ cảnh được sinh ra độc lập với nhau. Trong trường hợp này, công thức trên có thể được viết lại thành



Trong mô hình skip-gam, mỗi từ được biểu diễn bằng hai vector d-chiều để tính xác suất có điều kiện. Giả sử chỉ số của một từ trong từ điển là i, vector của từ được biểu diễn là vi∈Rd khi từ này là từ đích trung tâm và là ui∈Rd khi từ này là một từ ngữ cảnh. Gọi cc và oo lần lượt là chỉ số của từ đích trung tâm wc và từ ngữ cảnh wo trong từ điển. Có thể thu được xác suất có điều kiện sinh ra từ ngữ cảnh cho một từ đích trung tâm cho trước bằng phép toán softmax trên tích vô hướng của vector:



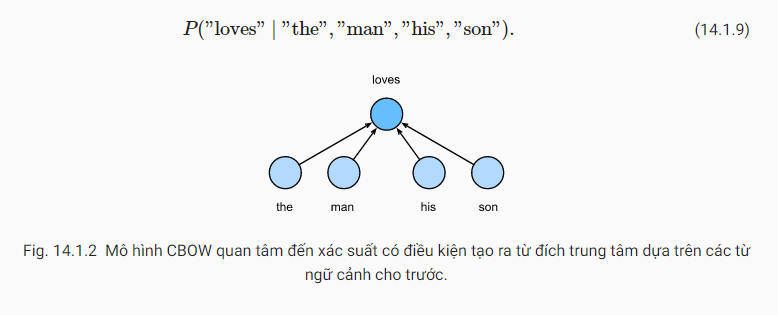
trong đó, tập chỉ số trong bộ từ vựng là V={0,1,…,|V|−1}V={0,1,…,|V|−1}. Giả sử trong một chuỗi văn bản có độ dài T, từ tại bước thời gian t được ký hiệu là w(t) Giả sử rằng các từ ngữ cảnh được sinh độc lập với từ trung tâm cho trước. Khi kích thước cửa sổ ngữ cảnh là m, hàm hợp lý (*likelihood*) của mô hình skip-gam là xác suất kết hợp sinh ra tất cả các từ ngữ cảnh với bất kỳ từ trung tâm cho trước nào



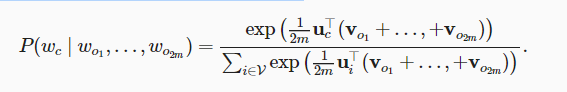
Ở đây, bất kỳ bước thời gian nào nhỏ hơn 1 hoặc lớn hơn T đều có thể được bỏ qua.

* + 1. **Mô Hình Túi từ liên tục (CBOW)**

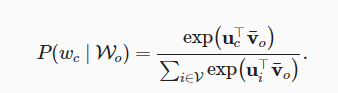
Mô hình túi từ liên tục (*Continuous bag of words* - CBOW) tương tự như mô hình skip-gram. Khác biệt lớn nhất là mô hình CBOW giả định rằng từ đích trung tâm được tạo ra dựa trên các từ ngữ cảnh phía trước và sau nó trong một chuỗi văn bản. Với cùng một chuỗi văn bản gồm các từ “the”, “man”, “loves”, “his” và “son”, trong đó “love” là từ đích trung tâm, với kích thước cửa sổ ngữ cảnh bằng 2, mô hình CBOW quan tâm đến xác suất có điều kiện để sinh ra từ đích “love” dựa trên các từ ngữ cảnh “the”, “man”, “his” và “son” như sau:



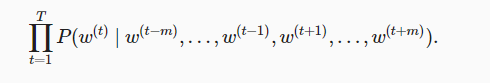
Vì có quá nhiều từ ngữ cảnh trong mô hình CBOW, ta sẽ lấy trung bình các vector từ của chúng và sau đó sử dụng phương pháp tương tự như trong mô hình skip-gram để tính xác suất có điều kiện. Giả sử vi∈Rd và ui∈Rd là vector từ ngữ cảnh và vector từ đích trung tâm của từ có chỉ số ii trong từ điển (lưu ý rằng các ký hiệu này ngược với các ký hiệu trong mô hình skip-gram). Gọi cc là chỉ số của từ đích trung tâm wc,và o1,…,o2 là chỉ số các từ ngữ cảnh wo1,…,wo2m trong từ điển.

Do đó, xác suất có điều kiện sinh ra từ đích trung tâm dựa vào các từ ngữ cảnh cho trước là

Để rút gọn, ký hiệu Wo={wo1,…,wo2m} }, và v¯o=(vo1+…,+vo2m)/(2m). Phương trình trên được đơn giản hóa thành



Cho một chuỗi văn bản có độ dài TT, ta giả định rằng từ xuất hiện tại bước thời gian tt là w(t)w(t), và kích thước của cửa sổ ngữ cảnh là mm. Hàm hợp lý của mô hình CBOW là xác suất sinh ra bất kỳ từ đích trung tâm nào dựa vào những từ ngữ cảnh.



* + 1. **Tóm Tắt**
* Vector từ là một vector được sử dụng để biểu diễn một từ. Kỹ thuật ánh xạ các từ sang vector số thực còn được gọi là kỹ thuật embedding từ.
* Word2vec bao gồm cả mô hình túi từ liên tục (CBOW) và mô hình skip-gam. Mô hình skip-gam giả định rằng các từ ngữ cảnh được sinh ra dựa trên từ đích trung tâm. Mô hình CBOW giả định rằng từ đích trung tâm được sinh ra dựa trên các từ ngữ cảnh.

# CHƯƠNG 4: KHẢO SÁT HỆ THỐNG

## 4.1. Mô hình hoạt động

**4.1.1. Chuẩn bị dữ liệu cho mô hình (Data Preparation)**

Bước này thực hiện các công việc tiền xử lý dữ liệu theo yêu cầu của mô hình như trích chọn thuộc tính, rời rạc hóa dữ liệu và cuối cùng là chia dữ liệu nguồn (Data Source) thành 3 tập dữ liệu dùng để huấn luyện mô hình (Training Data, chiếm 71,5% khối lượng dữ liệu ban đầu, Validation Data chiếm 20% khối dữ liệu ban đầu) và kiểm tra mô hình (Testing data, chiếm 8,5% khối lượng dữ liệu ban đầu).

Trong đề tài, khi chuẩn bị dữ liệu cho kho dữ liệu tổng quát xây dựng các thuộc tính chính là những bình luận thể hiện cảm xúc của người dùng. Vì thời gian làm đề tài và tham khảo còn hạn chế, nên hiện tại hệ thống chỉ phân tích được bình luận cảm xúc được phân hóa 2 vào chủ để. Cụ thể, hệ thống sẽ tập trung phân tích và dự đoán cảm xúc thuộc vào “yêu đời” hoặc “thất tình”.

Vì vậy, dựa vào luồng phân tích của đề tài, xây dựng kho dữ liệu cho hệ thống (tên kho dữ liệu: DATASET.xlsx)

**4.1.2 Xây dựng mô hình (Data Modeling)**

Bước này sử dụng Training Data vừa tạo ra ở mục 4.1.1 để xây dựng mô hình. Trong đề tài sẽ sử dụng thuật toán CNN để phân tích và dự đoán cảm xúc

**4.1.3 Kiểm định mô hình (Validation model)**

Sau khi sử dụng Training Data để xây dựng mô hình, bây giờ ta sử dụng Testing Data để kiểm tra xem mô hình có đủ tốt để sử dụng hay không? (Nếu chưa đủ tốt thì phải sử dụng Training Data khác để huấn luyện lại).

Kiểm định đánh giá mô hình dựa vào đánh giá trực quan

**4.1.4 Sử dụng mô hình (Model Usage)**

Sau khi mô hình được kiểm tra (Testing) nếu độ chính xác đáp ứng yêu cầu thì có thể sử dụng model đã xây dựng vào dự đoán các dữ liệu chưa biết.

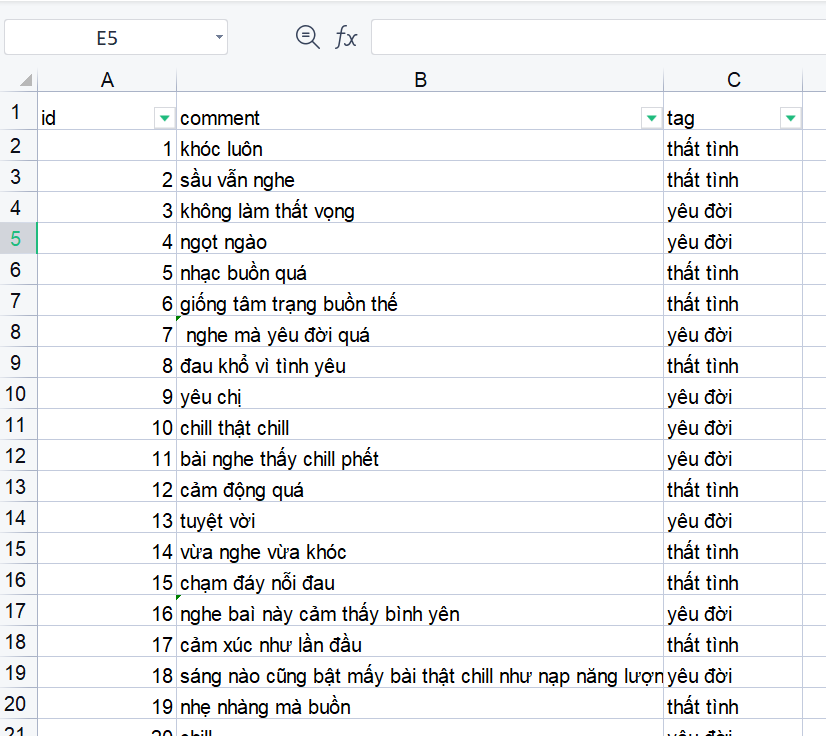
**4.2. Yêu cầu khi cài đặt thuật toán**

Hiện nay, trong lĩnh vực khai phá dữ liệu đã có rất nhiều công cụ (cả bản mã nguồn mở và bản thương mại), các công cụ phổ biến như Weka, R, Business Intelligence Development đều đã có các thư viện về các thuật toán khai phá dữ liệu rất phong phú và chính xác.

Sử dụng ngôn ngữ lập trình Python với thư viện mã nguồn mở Keras để triển khai thuật toán, tiếp đó là sử dụng ReactJs để thiết kế giao diện sử dụng.Như vậy, việc thể hiện thuật toán bằng ngôn ngữ lập trình sẽ được rút ngắn, đề tài sẽ đi sâu vào việc xây dựng mô hình khai phá dữ liệu.

**4.3. Phân tích – Xây dựng kho dữ liệu cho hệ thống**

Một kho dữ liệu cho hệ thống là dữ liệu đầu vào cho mô hình CNN. Mỗi dòng là 1 câu bình luận thể hiện cảm xúc của người dùng.



**Hình 12 Mô tả kho dữ liệu**

Tập dữ liệu tổng quát này bao gồm 1607 dòng tương ứng với 1607 ví dụ. Mỗi ví dụ biểu hiện cho bình luận cảm xúc của người dùng về bài hát. Thuộc tính “tag” có hai giá trị (yêu đời, thất tình) còn được gọi là thuộc tính đích (target attribute).

Bộ dữ liệu tổng hợp từ các Comment và đã được đánh giá phân tích cảm xúc. Mỗi tin nhắn được gắn nhãn phân loại theo các mục “Yêu Đời” , “Thất tình” . Với dữ liệu này giúp chúng ta huấn luyện cho mạng neural của máy. Mô tả file DATASET.xlsx dữ liệu gồm:

|  |  |  |
| --- | --- | --- |
|  | **Bảng 4.1: Mô tả dữ liệu thực nghiệm** | |
|  |  |  |
| **STT** | **Tên cột** | **Ghi chú** |
| 1 | Id | Mã của Comment |
| 2 | Comment | Bình luận |
|  |  | Phân loại của tin nhắn: Giá trị sẽ là |
| 3 | Tag | (Yêu đời),  (Thất tình) |
|  |  |  |

* 1. **Khảo sát hệ thống** **website nghe và tìm nhạc phẩm theo cảm xúc dựa trên phương pháp học sâu**

Một hệ website nghe và tìm nhạc phẩm theo cảm xúc dựa trên phương pháp học sâu được tổ chức như sau :

### Thông tin về các khách hàng:

Mỗi khách hàng (KH) sẽ có 1 mã khách hàng duy nhất (do hệ thống phần mềm cấp) để phân biệt với các khách hàng khác, họ, tên, ngày sinh, nơi sinh, phái, địa chỉ, email, điện thoại.

### Thông tin về các bài hát:

Mỗi bài hát sẽ có 1 mã bài hát, thứ sẽ đại diện cho việc lưu trữ chính trong cơ sở dữ liệu, thông tin về tên bài hát, đường dẫn , tên nghệ sĩ, tác giả, lyrics, thumbnail, và duration.

**Thông tin playlist**

Một playlist sẽ có duy nhất một Mã số playlist, name, author, Createtime, privacy, image.

Một Playlist sẽ có nhiều Chi Tiết Playlist-Song.

Một chi tiết đơn đặt hàng sẽ có thông tin về song Id, Playlist Id và position

Một chi tiết Playlist-Song sẽ chỉ thuộc duy nhất về 1 Playlist.

## Các chức năng của hệ thống.

### Yêu cầu chức năng của hệ thống:

## Hệ Thống Học Sâu:

## Từ bộ dữ liệu đầu vào có thể thể hiện ra 1 mô hình có tính logic rành mạch. Phù hợp với mọi logic gán nhãn cho tập người dùng nhất định.

## Phân tích được cảm xúc người dùng từ dữ liệu đầu vào là 1 đoạn văn bản, từ đó trả kết quả là nhãn cảm xúc cho hệ thống web xử lí.

Công việc User:

* + - * Đăng ký tài khoản / Đăng nhập
      * Xem thông tin Bài Hát
      * Nghe bài Hát
      * **Tìm kiếm theo cảm xúc**
      * **Tìm kiếm theo tên bài hát**
      * Xem thông tin cá nhân đổi mật khẩu
      * Xem báo cáo thống kê cá nhân
      * Tạo Playlist

Đối tượng tham gia hệ thống: Tất cả user

### Yêu cầu phi chức năng:

* Hệ thống đáng tin cậy, chính xác, giao diện thân thiện, dễ sử dụng, truy cập dữ liệu nhanh chóng.
* Đảm bảo tính bảo mật cho người điều hành hệ thống. Phải có tính linh hoạt cao.

# CHƯƠNG 5: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Mô hình chức năng nghiệp vụ (BFD – Business Function Diagram)

Chức năng là công việc mà tổ chức cần làm và được theo nhiều mức từ tổng hợp đến chi tiết.

Mỗi chức năng có một tên duy nhất, các chức năng khác nhau phải có tên khác nhau. Để xác định tên cho các chức năng một cách chính xác và cụ thể, có thể bàn luận và nhất trí với người sử dụng.

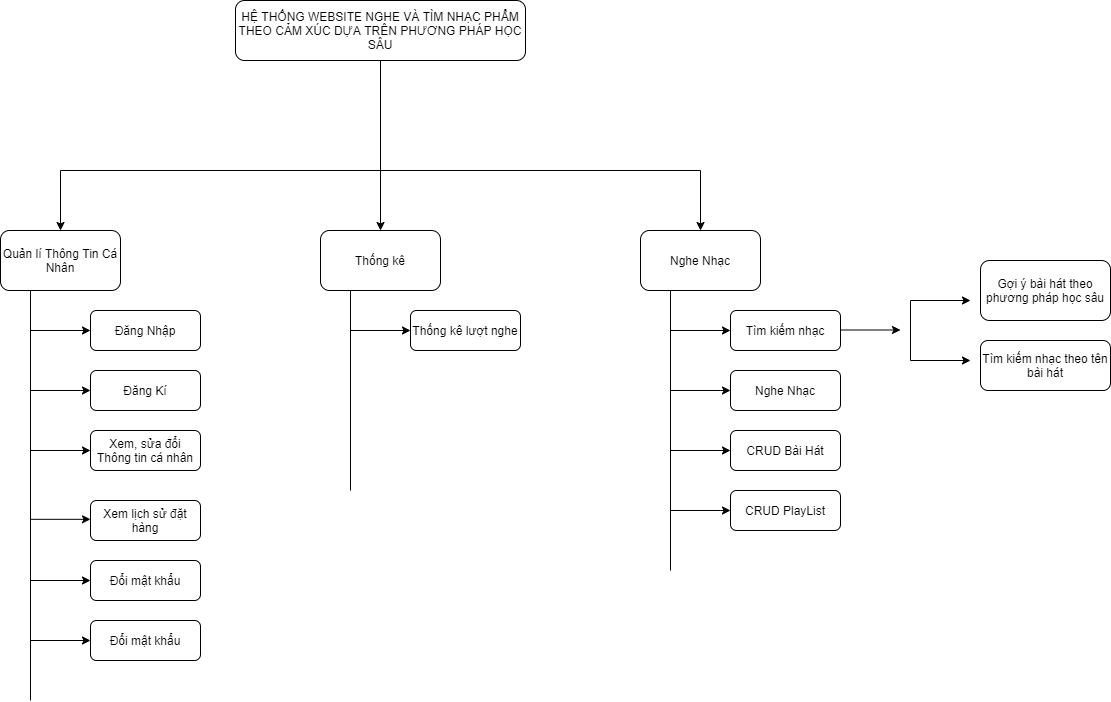
Hình thức biểu diễn: Hình chữ nhật:

Tên chức năng

Mỗi chức năng được phân rã thành các chức năng con. Các chức năng con có quan hệ phân cấp với chức năng cha.

Biểu diễn mối quan hệ phân cấp chức năng như sau:

Mô hình:



**Hình 13: . Mô hình phân rã chức năng được biểu diễn thành hình cây phân cấp**

* 1. **Mô hình luồng dữ liệu (*DFD – Data Flow Diagram*)**

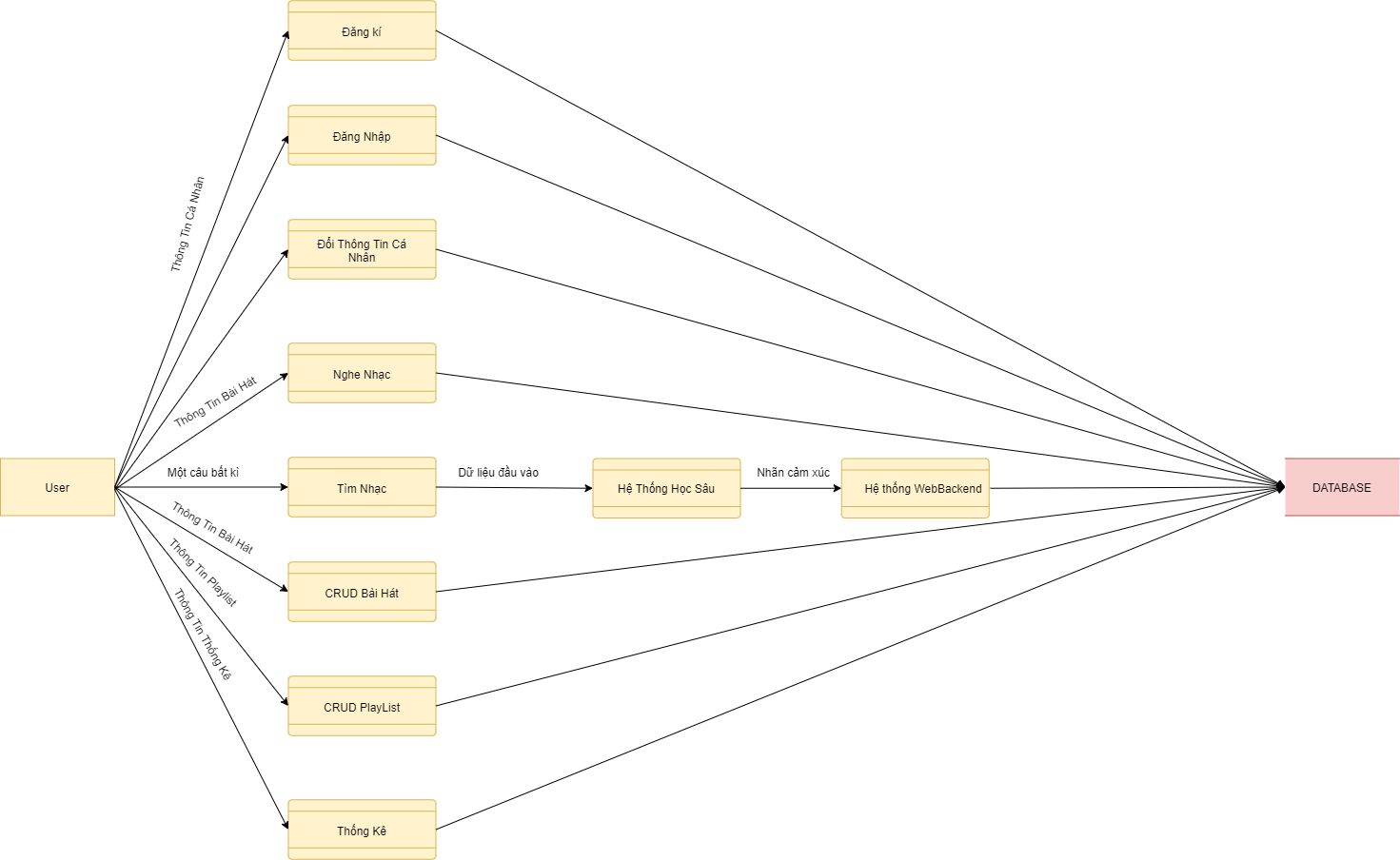
Mô hình luồng dữ liệu là một công cụ mô tả quan hệ thông tin giữa các công việc



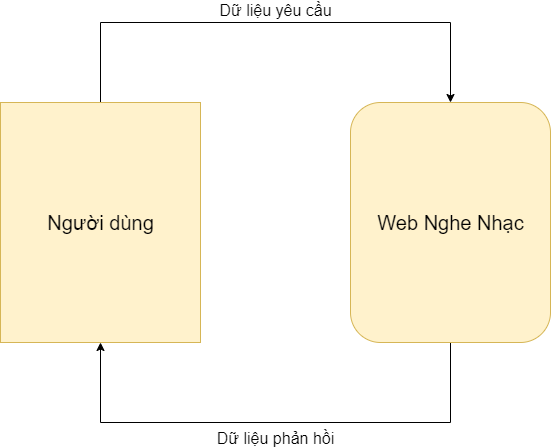
Biểu đồ luồng dữ liệu gồm có 5 yếu tố chính:

**Hình 14: Các đối tượng cơ bản sử dụng để thiết kế biểu đồ luồng dữ liệu**

Mô hình:



**Hình 15: Mô hình Data Flow Diagram tổng quát**



**Hình 16: Mô hình Data Flow Diagram mức 0**

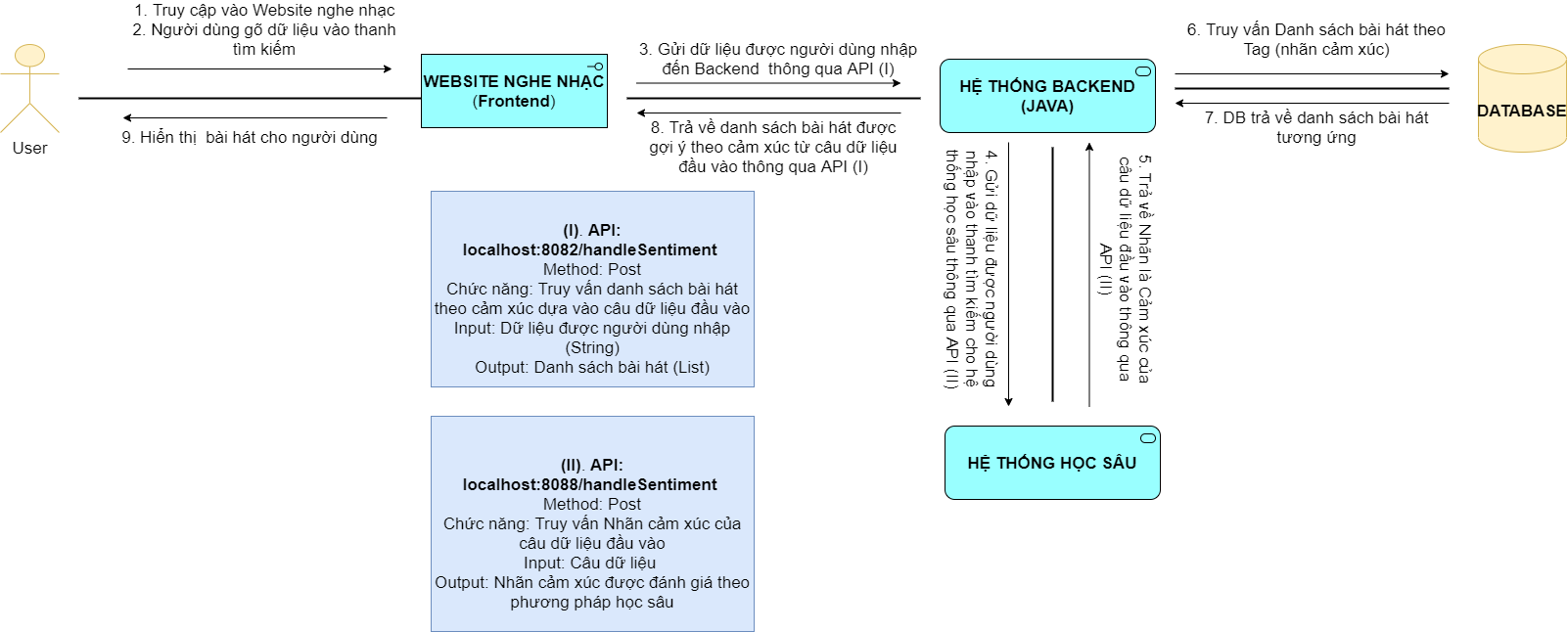
## Mô hình cộng tác (Collaboration Diagram)

Biểu đồ cộng tác là một minh họa về các mối quan hệ và tương tác giữa các đối tượng phần mềm trong Ngôn ngữ Mô hình Thống nhất

Cấu trúc của biểu đồ cộng tác:

|  |  |
| --- | --- |
| Actors - Là những thể hiện gọi ra tương tác trong biểu đồ |  |
| Messages - Những thông điệp này là thông tin liên lạc giữa các đối tượng truyền tải thông tin về hoạt động và có thể bao gồm số thứ tự |  |
| Objects - Đối tượng được hiển thị dưới dạng hình chữ nhật với nhãn đặt tên bên trong |  |
| Links - Liên kết kết nối các đối tượng với các tác nhân và được mô tả bằng cách sử dụng một đường liền nét giữa hai phần tử. Mỗi liên kết là một thể hiện mà các thông điệp có thể được gửi đi |  |

Mô hình:

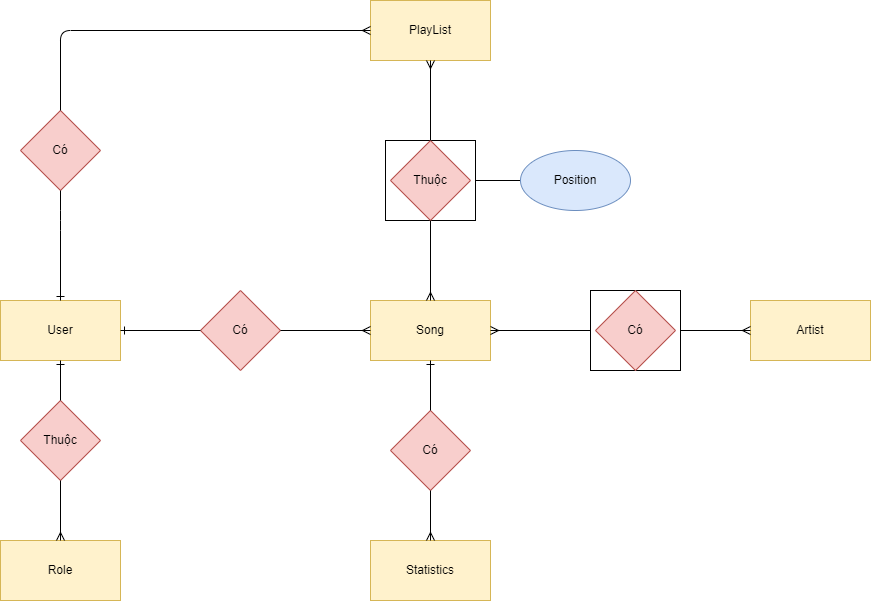


**Hình 17: Mô hình cộng tác**

## Thiết lập mô hình thực thể ERD (ERD - Entity Relationship Diagram)

### Xác định các thực thể

* + - * **User** (username, password, name, email, birthday, address, phone, role)
      * **Song** (id, path, name,author, lyrics,big thumbnail, thumnail, duration, tag)
      * **PlayList** (id, name, author, createTime, privacy, image)
      * **SongPlayList** (songId, PlayListId, position)
      * **Statistics** ( Date, songid, viewCount)
      * **Artist**(Id, name)
      * **Song\_Artist** (songId, artistId)
    1. **Mô hình thực thể kết hợp MS (ERD - Entity Relationship Diagram**

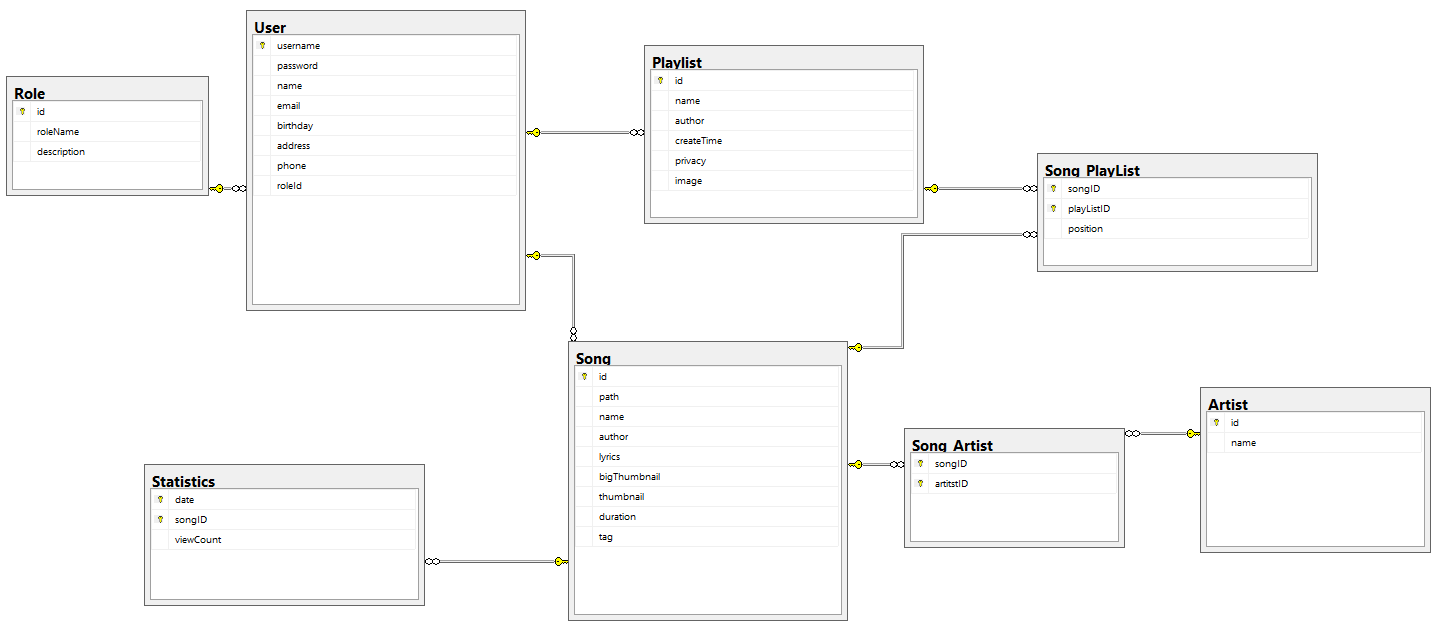
****

**Hình 18: Mô hình ERD**

# 

## Thiết kế cơ sở dữ liệu

### Mô hình Diagram



**Hình 19: Database Diagram**

### Từ điển dữ liệu

* Bảng: **User**

**User** (Account, passord, name, email, birthday, address, phone, role)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Diễn giải | Khóa duy  nhất | Khóa chính | Khóa ngoại | Not Null |
| 1 | username | nchar(20) | Tài Khoản đăng nhập |  | X |  | X |
| 2 | password | nchar(20) | Mật khẩu đăng nhập |  |  |  | X |
| 3 | name | nvarchar(100) | Họ Tên |  |  |  | X |
| 4 | email | nchar(20) | Địa chỉ email |  |  |  |  |
| 5 | Birthday | Date | Ngày sinh |  |  |  |  |
| 6 | Address | Nvarchar(200) | Địa chỉ |  |  |  |  |
| 7 | Phone | nchar(20) | Số điện thoại |  |  |  |  |
| 8 | RoleId | int | Chức vụ |  |  |  | X |

* Bảng: **Role**

**Role**(id, roleName, description)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Diễn giải | Khóa duy  nhất | Khóa chính | Khóa ngoại | Not Null |
| 1 | id | int | id |  | X |  | X |
| 2 | roleName | nchar(20) | Tên chức vụ |  |  |  | X |
| 3 | description | nvarchar(100) | Mô tả |  |  |  |  |

* Bảng: **Song**

**Song** (id, path, name, author, lyrics, bigThumbnail, Thumbnail, duration, tag)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Diễn giải | Khóa duy  nhất | Khóa chính | Khóa ngoại | Not Null |
| 1 | id | int | Khóa Chính |  | X |  | X |
| 2 | path | Nvarchar(200) | Đường Dẫn |  |  |  | X |
| 3 | name | nvarchar(100) | Tên Bài Hát |  |  |  | X |
| 4 | author | int | Tác Giả |  |  | X | X |
| 5 | Lyrics | Nvarchar(200) | Lời Bài Hát |  |  |  |  |
| 6 | bigThumbnail | nchar(20) | Hình Đại diện lớn |  |  |  |  |
| 7 | Thumbnail | nchar(20) | Hình đại diện |  |  |  |  |
| 8 | Duration | int | Thời gian |  |  |  |  |
| 9 | Tag | Nvarchar(20) | Phân loại bài hát |  |  |  |  |

* Bảng: **PlayList**

**PlayList** (id, name, author, privacy, createTime, image)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Diễn giải | Khóa duy  nhất | Khóa chính | Khóa ngoại | Not Null |
| 1 | id | int | Khóa Chính |  | X |  | X |
| 2 | name | Nvarchar(200) | Tên Playlist |  |  |  | X |
| 3 | author | int | Tác Giả |  |  | X | X |
| 4 | privacy | nvarchar(100) | Quyền riêng tư |  |  |  |  |
| 5 | createTime | DateTime | Ngày giờ tạo |  |  |  | X |
| 6 | image | Nvarchar(200) | Hình ảnh đại diện |  |  |  |  |

* Bảng: **SongPlayList**

**SongPlayList** (Songid, playlistId, position)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Diễn giải | Khóa duy  nhất | Khóa chính | Khóa ngoại | Not Null |
| 1 | songId | int | Khóa Chính |  | X | X | X |
| 2 | playListId | Nvarchar(200) | Tên Playlist |  | X | X | X |
| 3 | position | int | Tác Giả |  |  |  |  |

* Bảng: **Statistics**

**Statistics** (date, viewCount, SongId)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Diễn giải | Khóa duy  nhất | Khóa chính | Khóa ngoại | Not Null |
| 1 | date | Nvarchar(200) | Tên Playlist |  | X | X | X |
| 2 | viewCount | int | Tác Giả |  |  |  |  |
| 3 | songId | int | Song Id |  | X | X | X |

* Bảng: **Song\_Artist**

**Song\_Artist**(songId, artistId)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Diễn giải | Khóa duy  nhất | Khóa chính | Khóa ngoại | Not Null |
| 1 | songId | int | Song Id |  | X | X | X |
| 2 | artistId | int | Artist Id |  | X | X | X |

# CHƯƠNG 6: CHƯƠNG TRÌNH

## Giới Thiệu Chương Trình

Chương trình Hệ thống gồm 3 phần chính: .

1. Ứng dụng Web dành cho người dùng
2. Hệ Thống xử lí giao tiếp với DB (WebService) (API)
3. Hệ thống xử lí học sâu

### Mục Tiêu:

Mục tiêu của chương trình giúp cho việc:

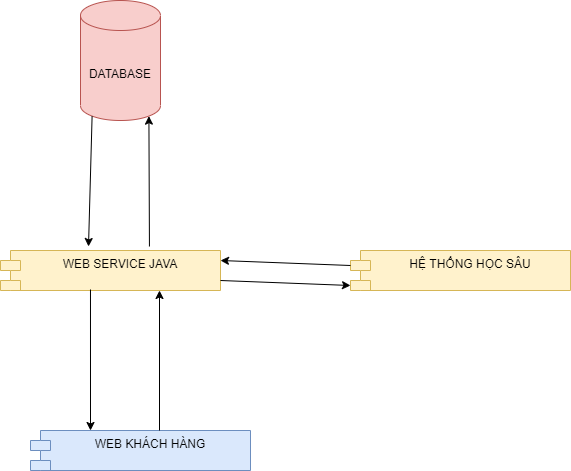
* Xây dựng được một hệ thống dựa trên giải thuật học sâu để hiểu được đặc trưng của các bình luận về cảm xúc mà người dùng có được khi nghe bài hát
* Để thuận tiện cho việc nghe nhạc được tiện lợi, website gợi ý bài hát theo cảm xúc của người dùng
* Có thể phát triển mở rộng dễ dàng khi hệ thống lớn hơn

## Thiết kế Web Service

### Giao tiếp Web Service

Trong chương trình của chúng ta Web Service truy cập đến Database.

Dữ liệu hệ thống sẽ được gửi thông qua API. WebService sẽ cung cấp các API để Chương trình hệ thống Service và Hệ Thống Học sâu có thể giao tiếp để trao đổi dữ liệu. WebService sẽ giao tiếp trực tiếp với Database.

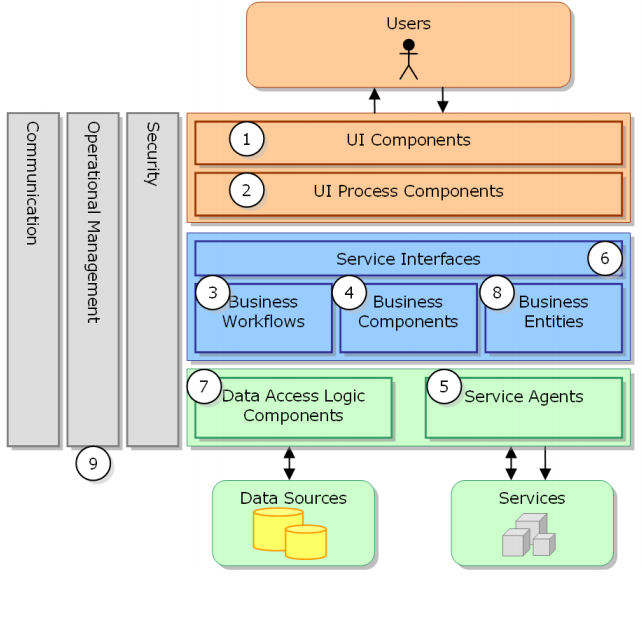


**Hình 20: Giao tiếp Toàn Hệ Thống**

### Kiến trúc của Web service

**Sử dụng Mô Hình Three-Tier Architecture:**

* + - * Lớp Interface trên cùng chịu trách nhiệm giao tiếp và xác thực các yêu cầu từ người dùng.
      * Lớp Business chịu trách nhiệm xác định và điều phối các quy trình nghiệp vụ gồm nhiều bước và kéo dài. Những quy trình này phải được sắp xếp và thực hiện theo một thứ tự chính xác. Kiểm tra các quy tắc nghiệp vụ, ràng buộc logic và thực hiện các công việc.
      * Lớp Data Access Layer chịu trách nhiệm chính lưu trữ và truy xuất dữ liệu từ các nguồn dữ liệu [(Data Sources](https://topdev.vn/blog/dung-gi-de-luu-tru-data-thay-the-local-storage/)). Hơn nữa còn tạo thuận lợi cho việc dễ cấu hình và bảo trì.

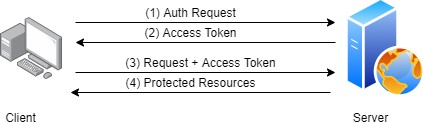


**Hình 21: Mô Hình Three-Tier Architecture**

### Authorize và Authentication

Người sử dụng giao tiếp với Web Service thông qua API. Vậy làm sao để xác định được thông tin người sử dụng và những tài nguyên nào mà người đó được phép truy cập.

Để giải quyết vấn đề này, chúng ta sẽ sử dụng một phương pháp bảo mật khá phổ biến đó là hệ thống sẽ xác định người sử dụng thông một đoạn mã được gọi là Token. Token được gắn vào Header mỗi khi gửi Request. Đoạn mã này chứa các thông tin đã được mã hóa gồm các thông tin cơ bản của người dùng và thời gian hết hạn. Khi mã hết hạn. Người dùng bắt buộc phải đăng nhập lại để lấy mã mới.



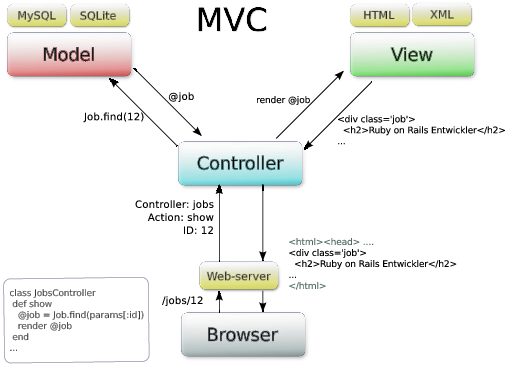
**Hình 22: Authorize và Authentication**

## Thiết kế Web cho khách hàng

### Mô hình sử dụng:

Sử dụng mô hình MVC : Mô hình MVC được chia làm 3 lớp xử lý gồm Model – View – Controller

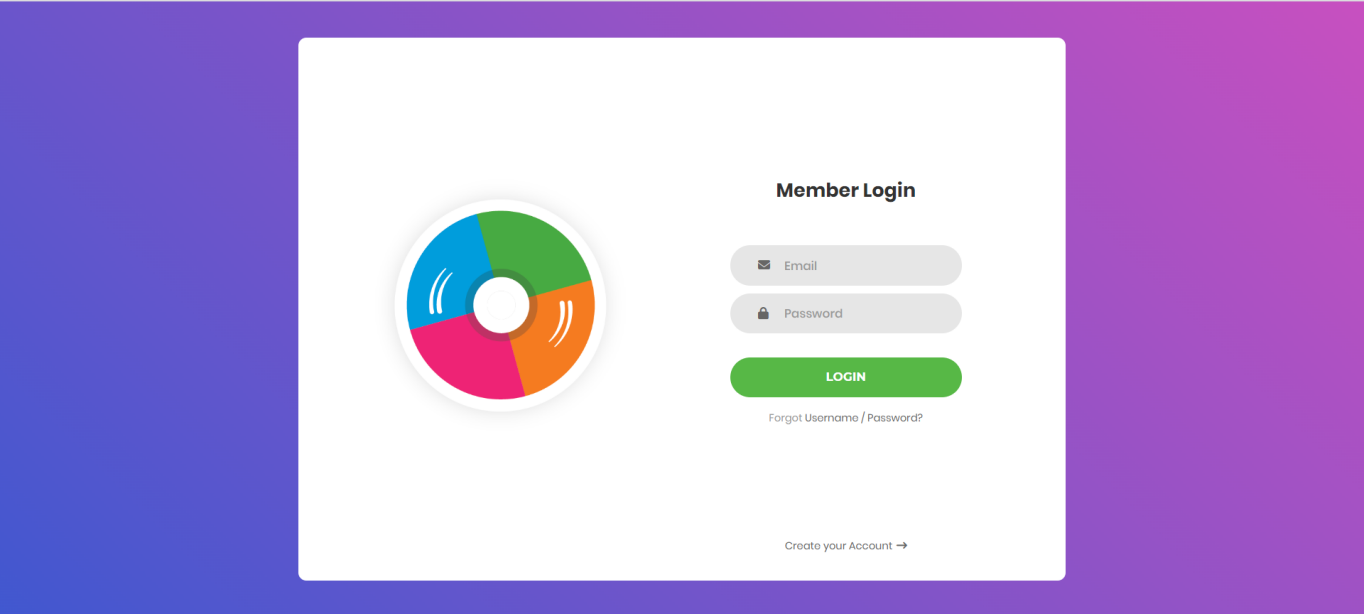
* Model : là nơi chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu (mysql, mssql… ); nó sẽ bao gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm – xóa – sửa dữ liệu…
* View : là nới chứa những giao diện như một nút bấm, khung nhập, menu, hình ảnh… nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.
* Controller : là nới tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View.



**Hình 23: Mô Hình MVC**

## Thiết kế giao diện chức năng:

* **Giao diện Đăng nhập**

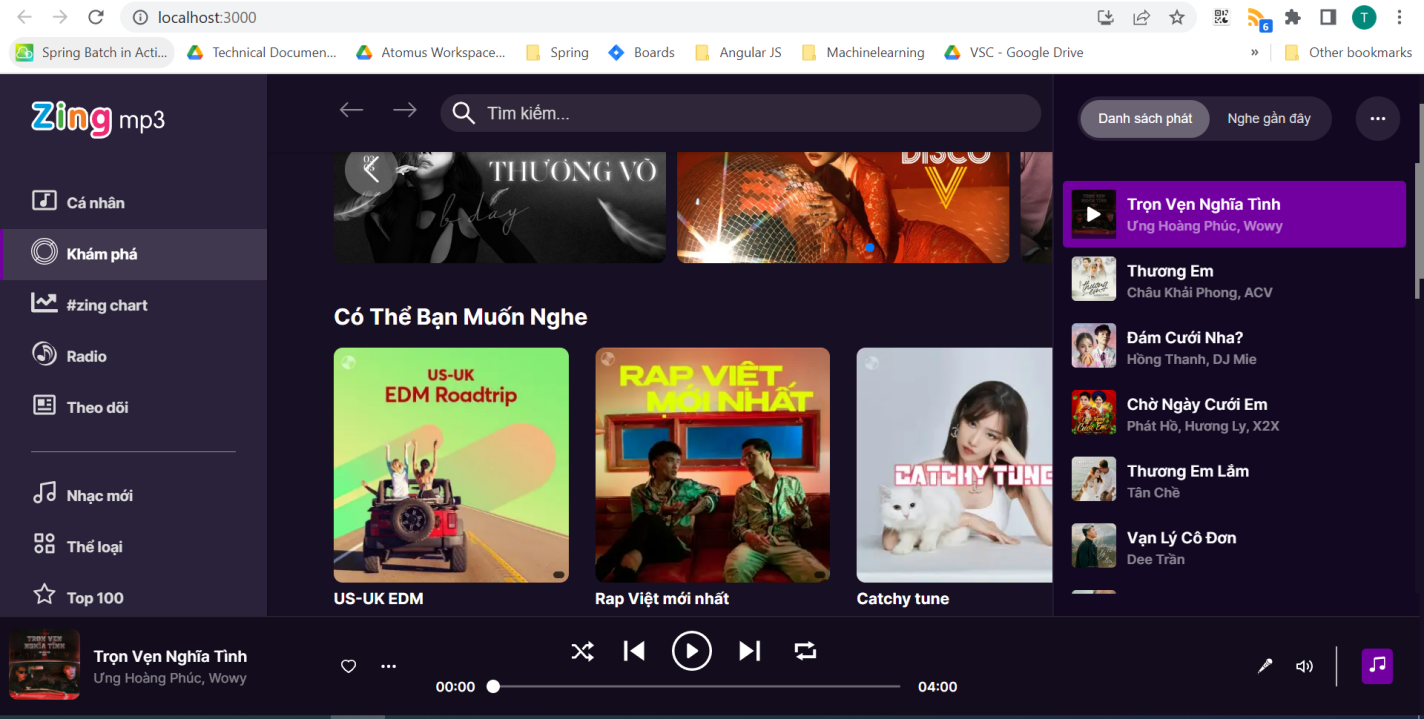


**Hình 24: Giao diện đăng nhập**

Đây là form đăng nhập thuận tiện cho người dùng điền tên đăng nhập và mật khẩu để truy cập vào website.

Nếu người dùng chưa có tài khoản thì có thể ấn “Create your account “ để truy cập sang trang đăng kí tài khoản

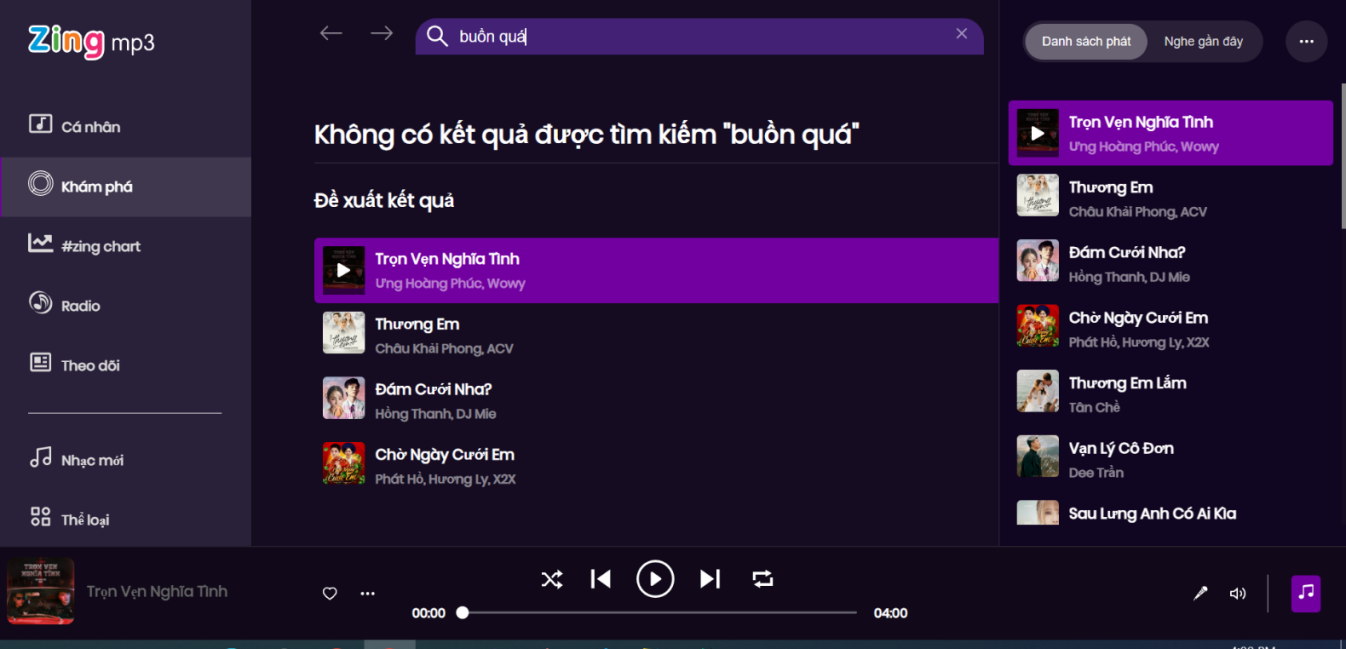
* **Giao diện Menu**



**Hình 25: Giao diện Menu**

Đây là menu chính của website nghe nhạc, bên trái là thanh menu giữa các trang, bên phải là thanh danh sách phát các bài nhạc, người dùng có thể lựa chọn bài hát và nhấn play để nghe.

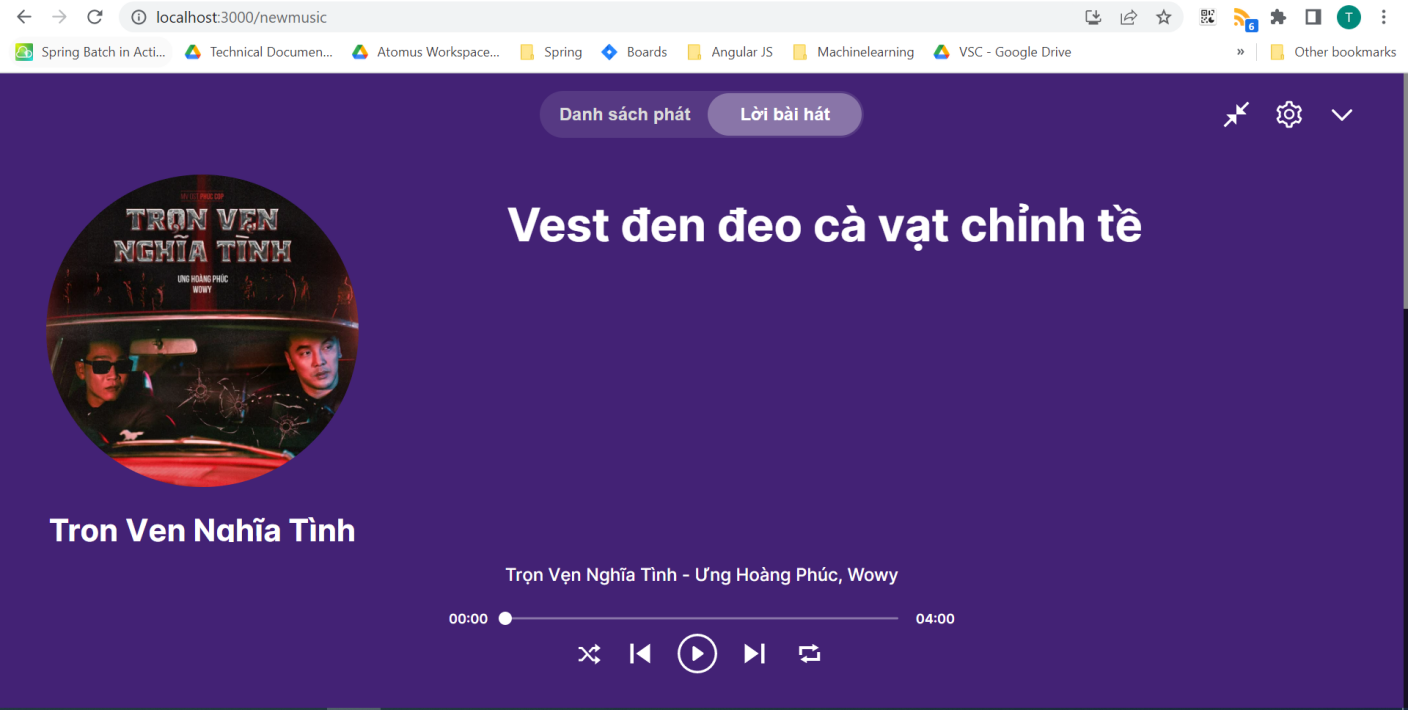
### Giao diện tìm nhạc theo cảm xúc



**Hình 26: Giao diện Tìm kiếm và kết quả xử lí câu**

Đây là menu chính của phần tìm kiếm. Người dùng có thể nhập bất cứ câu gì vào phần tìm kiếm, hệ thống sẽ thực hiện tìm kiếm theo tên bài hát, đồng thời cũng thực hiện sử dụng hệ thống học sâu để gợi ý một số bài hát dựa theo đánh giá cảm xúc của input mà người dùng nhập vào

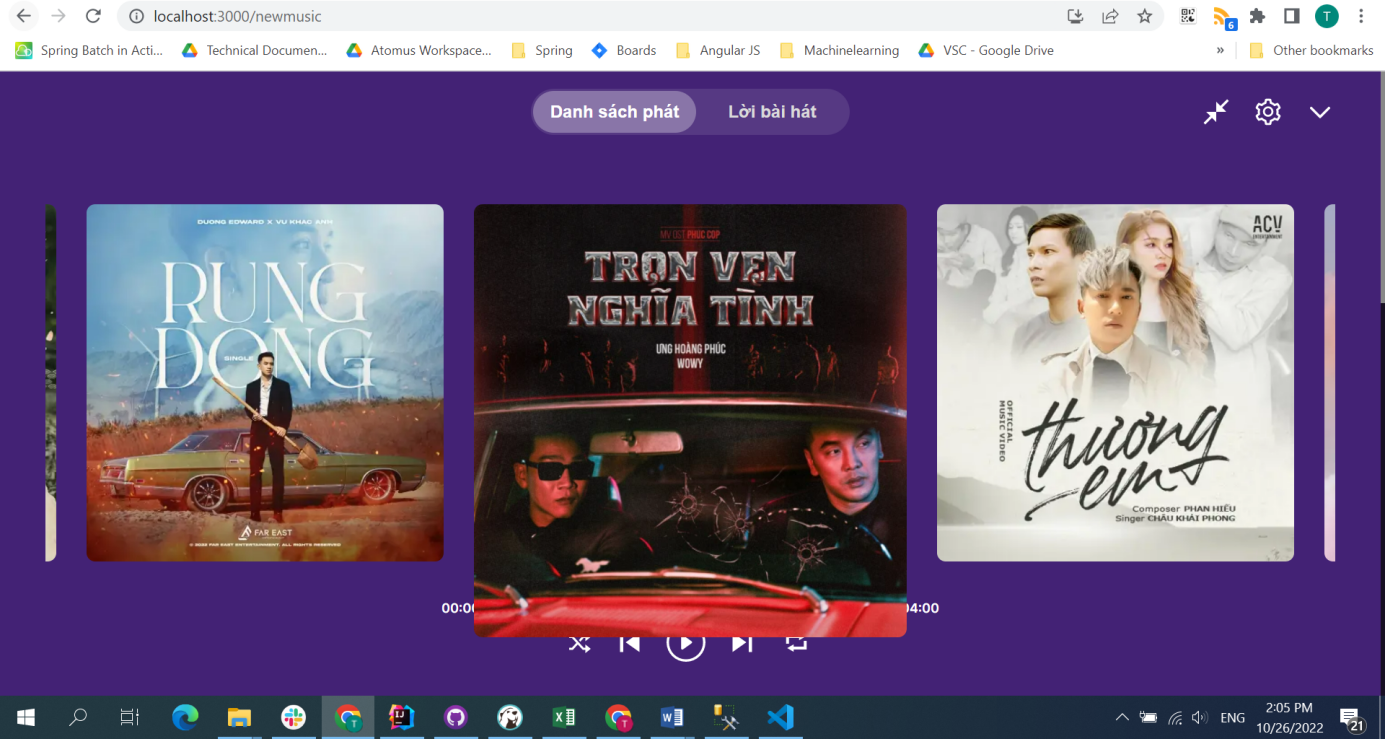
### Giao diện nghe bài hát



**Hình 27: Giao diện Nghe bài hát**

Đây là giao diện chính của phần nghe bài hát, người dùng có thể xem lời bài hát ở chính giữa màn hình.

### Giao diện bài hát



**Hình 28: Giao diện bài hát**

Ngoài ra còn có những giao diện khác tùy ý người dùng khám phá và truy cập.

# CHƯƠNG 7: KẾT LUẬN

## Kết quả đạt được và hạn chế

* + - **Kết quả đạt được:**
  + Xây dựng được website cho người dùng nghe nhạc
  + Xây dựng được một hệ thống dựa trên giải thuật học sâu để hiểu được đặc trưng của các bình luận về cảm xúc mà người dùng có được khi nghe bài hát
  + Tìm hiểu và nghiên cứu giải thuật cnn.
  + Làm việc với các công nghệ và công cụ mới hiện nay như Web Service, các ngôn ngữ như Java, ReactJs, Python.

## Hạn chế:

* + - * Giao diện thao tác người dùng chưa tối ưu.
      * Chức năng chương trình khá đơn giản.
      * Tập dữ liệu chỉ có 2 mức độ cảm xúc, trên thực tế thì nhiều hơn.
      * Tập dữ liệu dùng để huấn luyện vẫn còn rất ít, thực tế để hệ thống chính xác cần phải có thêm nhiều dữ liệu hơn nữa.

## Hướng phát triển

Trên cơ sở kế thừa những gì đã đạt được, tiếp tục sửa chữa và khắc phục hạn chế mà đồ án đang hiện có.

Hoàn thiện chương trình một cách tối ưu nhất để đem vào sử dụng thực tế và cho người dùng sử dụng một cách thuận tiện nhất.

# DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt :

[1]. “Giáo trình Kho dữ liệu và khai phá dữ liệu”, ThS Nguyễn Ngọc Duy

Tiếng Anh :

[3]. Kasra Madadipouya, “A new decision tree method for Data Mining in

Medicine” , Advanced Computational Intelligence: An International Journal

(ACII), Vol.2, No.3, July 2015,

Danh mục các Website tham khảo :

[4]. https://en.wikipedia.org/wiki

[5]. https://en.wikipedia.org/wiki/Convolutional\_neural\_network

[6]. www.delltechnologies.com/

[7]. http://stackoverflow.com/

[8]. https://www.oktot.com

[9]. <https://www.w3schools.com>

[10]. Tìm hiểu về Convolutional Neural Networks (CNN) (<https://viblo.asia/>)

[11]. Spring Framework : <https://mvnrepository.com/artifact/org.springframework>

[12]. Java EE documentation : <https://docs.oracle.com/javaee/7/index.html>

[13]. SpringBoot Document: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

[14]. Machine learning : <https://www.tensorflow.org/resources/learn-ml/basics-ofmachinelearning>

[15]. SQL Server: <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>