TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO BÀI TẬP LỚN**

**MÔN: NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**< NHÓM G04 >**

**ĐỀ TÀI: PHÁT TRIỂN PHẦN MỀM MẠNG XÃ HỘI CHIA SẺ ÂM NHẠC SOUNDFRIENDS CHO TRUNG TÂM GIẢI TRÍ MTP ENTERTAINMENT THEO CÔNG NGHỆ PHẦN MỀM**

| ***Giảng viên hướng dẫn:*** | ***ThS. Lê Hữu Dũng*** |
| --- | --- |
| ***Sinh viên thực hiện:*** | *Nguyễn Huy Hùng -1910A02*  *Trịnh Quang Toàn - 2010A01*  *Trần Văn Quang - 1910A05*  *Nguyễn Thanh Bình - 1810A01*  *Đỗ Quốc Huy - 2010A01*  *Trần Trung Kiên - 1910A04*  *Chử Văn Nam - 1910A01*  *Nguyễn Quang Thắng - 1910A01*  *Lê Thùy Trang - 1910A02*  *Lê Mạnh Cường - 2010A05* |

Hà Nội - 2023

**MỤC LỤC**

[**DANH MỤC TỪ VIẾT TẮT 4**](#_heading=h.tyjcwt)

[**DANH MỤC HÌNH ẢNH 5**](#_heading=h.6j81ngj99y2k)

[**CHƯƠNG 1. GIỚI THIỆU 7**](#_heading=h.3dy6vkm)

[1.1. Tổng quan về đề tài 7](#_heading=h.d1dga4jo14kn)

[1.2. Giới thiệu bài toán 7](#_heading=h.dd5rquvprm3l)

[1.3. Mục tiêu của đề tài 8](#_heading=h.pxuuxuhmumlw)

[1.4. Các đề tài/hệ thống liên quan 9](#_heading=h.83n0mb7yw7zk)

[1.5. Phương pháp phát triển 10](#_heading=h.al945vhcf403)

[1.6. Vai trò của thành viên dự án 11](#_heading=h.eebki9so4jjl)

[1.7. Phân công nhiệm vụ 12](#_heading=h.kgxstieudnzi)

[1.8. Kế hoạch triển khai chi tiết 17](#_heading=h.d2fcmzi85qg3)

[**CHƯƠNG 2. SẢN PHẨM CHUYỂN GIAO GIỮA CÁC PHA VÀ TIÊU CHUẨN ĐÁNH GIÁ 22**](#_heading=h.7x54zcuk5hk7)

[2.1. Danh mục tài liệu, sản phẩm 22](#_heading=h.3j2qqm3)

[2.2. Quy tắc đặt tên (naming convention) 28](#_heading=h.4d34og8)

[**CHƯƠNG 3. XÁC ĐỊNH YÊU CẦU 32**](#_heading=h.5of5ubnsut7v)

[3.1. Vai trò của việc Xác định yêu cầu đối với dự án phần mềm 32](#_heading=h.o7w01zdi2h4j)

[3.2. Phương pháp đánh giá chất lượng của yêu cầu 33](#_heading=h.kxj39md6ez9)

[3.3. Danh sách các yêu cầu của hệ thống: 34](#_heading=h.bzpmnj7okqhd)

[3.4. Kết quả đánh giá chất lượng các yêu cầu 51](#_heading=h.7id1sbezttoc)

[**CHƯƠNG 4. KIẾN TRÚC PHẦN MỀM 52**](#_heading=h.5ing20cpsofk)

[4.1. Vai trò của kiến trúc phần mềm với quá trình phát triển dự án phần mềm 52](#_heading=h.ix5eqadqvtzd)

[4.2. Một số kiến trúc phần mềm phổ biến 52](#_heading=h.hatm6ju0hcw5)

[4.3. Kiến trúc tổng thể của hệ thống 55](#_heading=h.6k3vrr5vcx05)

[4.4. Lựa chọn công nghệ phát triển cho hệ thống 55](#_heading=h.l4a0qgm5xqpv)

[4.5. Kiến trúc phần mềm của hệ thống 55](#_heading=h.gup7cp2r0eo)

[4.6. Cơ sở dữ liệu mức vật lý của hệ thống 57](#_heading=h.uk75h37d33jx)

[4.7. Một số thiết kế chi tiết theo kiến trúc được chọn 57](#_heading=h.ezqj6hfvn85n)

[**CHƯƠNG 5. XÂY DỰNG VÀ KIỂM THỬ HỆ THỐNG 80**](#_heading=h.17dp8vu)

[5.1. Kết quả xây dựng 01 module/nhóm chức năng 80](#_heading=h.3rdcrjn)

[5.2. Kiểm thử hệ thống 86](#_heading=h.35nkun2)

[**CHƯƠNG 6. KẾT LUẬN 109**](#_heading=h.nn27et8hlbhl)

[6.1. Đánh giá chất lượng phần mềm 109](#_heading=h.i7yns63ke41q)

[6.2. Ưu điểm & Nhược điểm của đề tài 111](#_heading=h.qe8xm76vjx8)

[6.3. Hướng phát triển 113](#_heading=h.4b93uj2aoubf)

[**TÀI LIỆU THAM KHẢO 114**](#_heading=h.cdjo4yi6dhif)

[**PHỤ LỤC A: QUY TẮC ĐẶT TÊN 114**](#_heading=h.2jxsxqh)

[**PHỤ LỤC B: CÁC BIỂU MẪU, TÀI LIỆU THU THẬP ĐƯỢC 115**](#_heading=h.fis1b7cxmsm4)

[**PHỤ LỤC C: CÁC BIÊN BẢN TIẾN ĐỘ 115**](#_heading=h.z337ya)

[**PHỤ LỤC D: CÁC BIÊN BẢN RÀ SOÁT KỸ THUẬT 115**](#_heading=h.5hce04ocqoee)

[**PHỤ LỤC E: HỒ SƠ TOÀN BỘ DỰ ÁN 115**](#_heading=h.ax15ze9mbzhi)

**DANH MỤC TỪ VIẾT TẮT**

| Từ viết tắt | Tên tiếng Anh | Tên tiếng Việt |
| --- | --- | --- |
| UBND |  | Ủy ban nhân dân |
| AWS | Amazon Web Services | Amazon Web Services |
| rAI | Requirement, Analysis, Implement | Yêu cầu, Phân tích, Lập trình |
| DT | Design, Testing | Thiết kế, kiểm thử |
| UI | User Interface | Giao diện người dùng |
| UML | Unified Modeling Language | Ngôn ngữ mô hình hóa thống nhất |
| CSDL |  | Cơ sở dữ liệu |
| 1NF, 2NF, 3NF | First / Second / Third Normal Form | Dạng chuẩn 1, 2, 3 |
| ID | Identity | Mã số định danh |
| URL | Uniform Resource Locator | Hệ thống định vị tài nguyên thống nhất / Địa chỉ Web |
| MVC | Model Control View |  |
| ISO 12207 | Standard 12207 of International Organization for Standardization | Tiêu chuẩn 12207 của tổ chức tiêu chuẩn hóa quốc tế |
| UAT | User Acceptance Testing | Kiểm thử người dùng chấp nhận |
| SDLC | Software Development Life Cycle | Vòng đời phát triển hệ thống |

**DANH MỤC HÌNH ẢNH**

[Ảnh 1.1. Đề tài liên quan 9](#_heading=h.p28gdggfj5cm)

[Ảnh 3.1. Sơ đồ phân rã chức năng 45](#_heading=h.tx3e6mws8anv)

[Ảnh 4.1. Cơ sở dữ liệu mức vật lý của hệ thống 57](#_heading=h.kzk6u1d4l4o6)

[Ảnh 4.2. UML chức năng đăng ký tài khoản 58](#_heading=h.vwurbfh9sc51)

[Ảnh 4.3. Xử lý chi tiết view Đăng ký: doCreate 60](#_heading=h.s6znh982ukpl)

[Ảnh 4.4. Xử lý chi tiết view Đăng ký: validateTaiKhoan 61](#_heading=h.lpno8ix67f3r)

[Ảnh 4.5. Xử lý chi tiết controller Đăng ký: objDangKy 61](#_heading=h.ru0nhume1a6c)

[Ảnh 4.6. Xử lý chi tiết model Đăng ký: checkDangKy 62](#_heading=h.21u1q4l4y5uc)

[Ảnh 4.7. Xử lý chi tiết Model Đăng ký: DangKyTK 63](#_heading=h.tt9zn15v2uvj)

[Ảnh 4.8. UML chức năng Đăng nhập 63](#_heading=h.cnb9g48vq8ik)

[Ảnh 4.9. Xử lý chi tiết view Đăng nhập: doCreate 66](#_heading=h.p46771cat1cx)

[Ảnh 4.10. Xử lý chi tiết view Đăng nhập: validate 66](#_heading=h.u83fxpp46x07)

[Ảnh 4.11. Xử lý chi tiết controller Đăng nhập: objDangNhap 67](#_heading=h.mdmnascx60v4)

[Ảnh 4.12. Xử lý chi tiết model Đăng nhập: checkTaiKhoan 68](#_heading=h.ajzfcmu2fy6d)

[Ảnh 4.13. Xử lý chi tiết model Đăng nhập: dangNhap 68](#_heading=h.yd8stqdokq7c)

[Ảnh 4.14. UML chức năng tìm kiếm 69](#_heading=h.eo3nrm3ua7r9)

[Ảnh 4.15. Xử lý chi tiết view Tìm kiếm: getTimKiem 71](#_heading=h.dw6bvvyek5a0)

[Ảnh 4.16. Xử lý chi tiết controller Tìm kiếm: objTimKiem 71](#_heading=h.2ln8l68eiqyq)

[Ảnh 4.17. Xử lý chi tiết model tìm kiếm: CheckTimKiem 72](#_heading=h.a2hvml1wkvt6)

[Ảnh 4.18. Xử lý chi tiết model Tìm kiếm: timKiem 72](#_heading=h.dzttl1kz3n5v)

[Ảnh 4.19. UML chức năng tạo danh sách nhạc 73](#_heading=h.nj8qmbykvyu8)

[Ảnh 4.20. Xử lý chi tiết model tạo danh sách phát: mTaoDanhSachPhat 73](#_heading=h.j6j6zfsl1nme)

[Ảnh 4.21. UML chức năng thêm bài hát vào danh sách 74](#_heading=h.y51hshz1d4r3)

[Ảnh 4.22. Xử lý chi tiết Thêm bài hát vào danh sách phát: objBaiHat, objDanhSachPhat 75](#_heading=h.guvyqpk0qljl)

[Ảnh 4.23. UML chức năng tải xuống nhạc 75](#_heading=h.avvyqn4i2nu)

[Ảnh 4.24. Xử lý chi tiết Tải xuống nhạc: objBaihat 76](#_heading=h.1q2x6hhwteqi)

[Ảnh 4.25. UML chức năng phát nhạc 76](#_heading=h.7iylk178svdi)

[Ảnh 4.26. Xử lý chi tiết Phát nhạc: objBaiHat 77](#_heading=h.k36hj863ri0g)

[Ảnh 4.27: UML chức năng chia sẻ danh sách phát 78](#_heading=h.6b1x85xvs1vr)

[Ảnh 4.28. Xử lý chi tiết view chia sẻ danh sách phát: viewChiaSeDS 78](#_heading=h.41gq1sv6gl50)

[Ảnh 4.29. Xử lý chi tiết controlerChiaSeDS:ValidateThongTin 79](#_heading=h.z3iwk365cj8n)

[Ảnh 4.30. Xử lý chi tiết modelChiaSeDS: NhapIDvaoDB 79](#_heading=h.1s8ujhs13xez)

[Ảnh 5.1. Kết quả xây dựng chức năng đăng nhập 80](#_heading=h.6hnr7tr88deg)

[Ảnh 5.2. Kết quả xây dựng chức năng đăng ký 81](#_heading=h.ae60szj3ex60)

[Ảnh 5.3. Kết quả xây dựng chức năng tải lên nhạc 82](#_heading=h.nnqck53hxblv)

[Ảnh 5.4. Kết quả xây dựng chức năng tìm kiếm 83](#_heading=h.n5ix1iil47gh)

[Ảnh 5.5. Kết quả xây dựng chức năng phát nhạc 85](#_heading=h.mggz57v5nzke)

[Ảnh 5.6. Kết quả xây dựng chức năng phát nhạc 85](#_heading=h.jmsmw5673k7o)

[Ảnh 5.7. Mô hình kiểm thử V-model 88](#_heading=h.bxldzli7flzg)

[Ảnh 5.8. Kết quả mức kiểm thử đơn vị 105](#_heading=h.cn1im3m186lq)

[Ảnh 5.9. Kết quả mức kiểm thử tích hợp 106](#_heading=h.1dnhzlkk6lq1)

[Ảnh 5.10. Kết quả mức kiểm thử hệ thống - đăng nhập 106](#_heading=h.lqpw2jur7qcy)

[Ảnh 5.11. Kết quả mức kiểm thử hệ thống - quản lý nhạc 107](#_heading=h.f8sjmo23oscn)

[Ảnh 5.12. Kết quả mức kiểm thử chấp nhận - đăng nhập 108](#_heading=h.eglmtemm2k1y)

**CHƯƠNG 1. GIỚI THIỆU**

* 1. **Tổng quan về đề tài**

Đề tài phát triển phần mềm mạng xã hội chia sẻ âm nhạc là một dự án phát triển phần mềm có mục tiêu xây dựng một mạng xã hội dành cho những người yêu âm nhạc tại trung tâm giải trí MTP Entertainment tại ngõ 234 Hoàng Quốc Việt, phường Cổ Nhuế, quận Bắc Từ Liêm, Hà Nội. Phần mềm sẽ cung cấp cho người dùng một nền tảng để tìm kiếm, nghe và chia sẻ âm nhạc, đồng thời tạo ra một cộng đồng chia sẻ âm nhạc trực tuyến.

* 1. **Giới thiệu bài toán**

Phần mềm này sẽ tích hợp nhiều tính năng hấp dẫn, bao gồm:

* Hệ thống tìm kiếm thông minh: Cho phép người dùng tìm kiếm âm nhạc theo tên bài hát, tên nghệ sĩ, thể loại, và các tiêu chí khác để tìm kiếm nhanh chóng và chính xác nhất.
* Nghe nhạc trực tuyến: Người dùng có thể nghe nhạc trực tuyến miễn phí, đồng thời có thể lưu trữ nhạc yêu thích của mình trên mạng xã hội.
* Chia sẻ âm nhạc: Người dùng có thể chia sẻ âm nhạc của mình với cộng đồng mạng xã hội.
* Tạo playlist: Người dùng có thể tạo ra các playlist theo chủ đề, thể loại, nghệ sĩ, album và chia sẻ với cộng đồng mạng xã hội.
* Gợi ý âm nhạc: Phần mềm sẽ cung cấp các gợi ý âm nhạc cho người dùng dựa trên sở thích âm nhạc của họ.
* Đăng ký: Người dùng có thể tạo ra tài khoản của bản thân mình
* Đăng nhập, đăng xuất: Cho phép người dùng xác minh tài khoản để thực hiện các chức năng như đăng nhạc, quản lý nhạc của bản thân đã đăng.
* Quản lý nhạc của tôi: Cho phép người dùng có thể đăng nhạc lên, xóa, chỉnh sửa lại các thông tin về nhạc của mình đã đăng.
  1. **Mục tiêu của đề tài**

Mục tiêu chính của đề tài phát triển phần mềm mạng xã hội chia sẻ âm nhạc là tạo ra một nền tảng cho người dùng chia sẻ và khám phá âm nhạc dễ dàng và tiện lợi. Các mục tiêu cụ thể của đề tài này có thể bao gồm:

Tạo ra một giao diện người dùng đơn giản và dễ sử dụng để cho phép người dùng tải lên và chia sẻ bài hát, tạo playlist, tìm kiếm và khám phá các bài hát mới.

Cung cấp cho người dùng các công cụ tìm kiếm nâng cao để tìm kiếm các bài hát theo thể loại, nghệ sĩ hoặc lời bài hát.

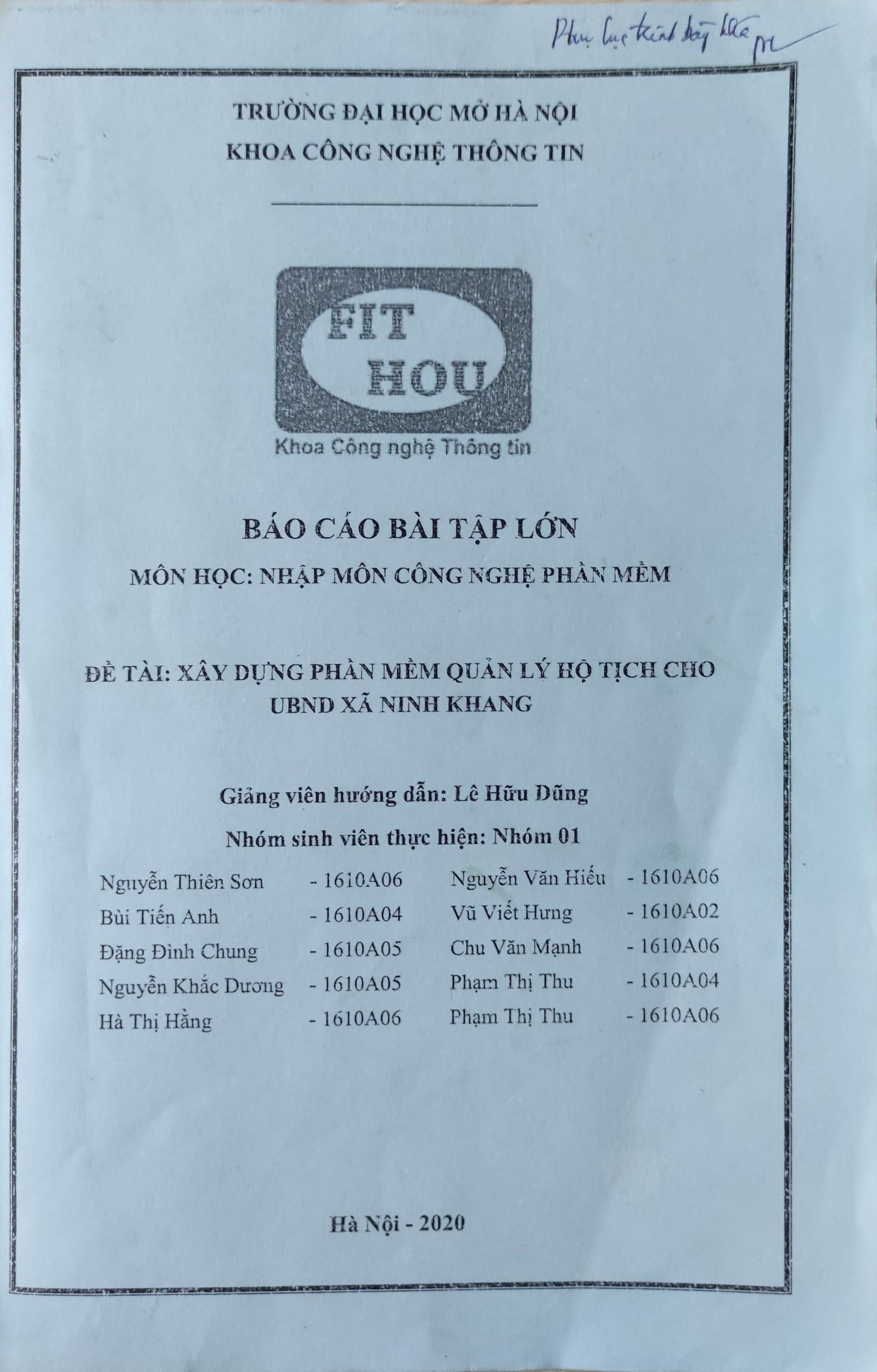
Tạo ra các tính năng giúp người dùng tương tác với nhau, bao gồm theo dõi và chia sẻ các bài hát yêu thích của mình trên các nền tảng mạng xã hội khác.

Tích hợp các tính năng bảo mật để đảm bảo rằng thông tin người dùng được bảo mật và an toàn.

Phát triển các tính năng bổ sung để cải thiện trải nghiệm người dùng, bao gồm lời bài hát, thông tin về nghệ sĩ và ban nhạc, tạo playlist tự động và các tính năng khác.

Tối ưu hóa hiệu suất và độ tin cậy của phần mềm để đảm bảo rằng nó có thể xử lý lượng dữ liệu lớn và đáp ứng được yêu cầu của người dùng.

* 1. **Các đề tài/hệ thống liên quan**



Ảnh 1.1. Đề tài liên quan

* Ưu điểm
  + Chi phí thấp
  + Dễ dàng thao tác, dễ dàng sử dụng, giao diện thân thiện với người dùng Quản lý mọi hoạt động của phần mềm 1 cách thuận tiện
  + Dễ dàng theo dõi quá trình làm việc của cán bộ Quan sát quá trình hoạt động của UBND một cách tổng quan nhất để hướng tới mục tiêu phục vụ nhân dân và phát triển UBND một cách tốt nhất. Đẩy mạnh kết quả cũng như chất lượng của công việc
* Nhược điểm
  + Các chức năng cần được hoàn thiện nhiều hơn để đảm bảo bao quát đầy đủ nhu cầu của khách hàng
  + Việc in biểu mẫu hộ tịch vẫn chưa được tích hợp trong phần mềm
  1. **Phương pháp phát triển**

Phương pháp phát triển Waterfall Model có thể tuân thủ quy trình phát triển với các bước chính như

* Thu thập yêu cầu: là giai đoạn đầu tiên của quy trình trong đó nhóm phát triển phải tìm hiểu về yêu cầu của khách hàng và người dùng cuối cùng. Cần phải tìm hiểu nhu cầu của người dùng, xác định các tính năng và chức năng quan trọng của hệ thống chia sẻ âm nhạc
* Thiết kế kiến trúc: Sau khi xác định yêu cầu của khách hàng, có thể tiến hành thiết kế kiến trúc của hệ thống, đặc biệt là cơ sở dữ liệu và giao diện người dùng.
* Lập trình: Sau khi thiết kế kiến trúc, có thể bắt đầu lập trình hệ thống mạng xã hội chia sẻ âm nhạc. Họ sẽ sử dụng các công nghệ phần mềm như Java, Python, Ruby on Rails, Angular, React, v.v.
* Kiểm thử: Kiểm thử là một phần quan trọng trong quy trình phát triển phần mềm. Trong giai đoạn này, các nhà phát triển sẽ tiến hành kiểm tra và đảm bảo rằng hệ thống hoạt động chính xác và không có lỗi.
* Triển khai và triển khai hệ thống: Sau khi các nhà phát triển đã hoàn thành việc lập trình và kiểm thử hệ thống, có thể triển khai và triển khai hệ thống. Hệ thống sẽ được cài đặt trên máy chủ của mình hoặc sử dụng các dịch vụ đám mây như AWS, Google Cloud Platform hoặc Microsoft Azure.
* Hỗ trợ và bảo trì: Sau khi triển khai hệ thống, các nhà phát triển sẽ tiếp tục hỗ trợ và bảo trì hệ thống mạng xã hội chia sẻ âm nhạc để đảm bảo rằng hệ thống hoạt động hiệu quả và đáp ứng yêu cầu của khách hàng và người dùng.
  1. **Vai trò của thành viên dự án**

| STT | Họ và tên | rAI | DT |
| --- | --- | --- | --- |
| 1 | Nguyễn Huy Hùng |  | x |
| 2 | Trần Văn Quang |  | x |
| 3 | Trịnh Quang Toàn |  | x |
| 4 | Lê Thùy Trang |  | x |
| 5 | Lê Mạnh Cường | x |  |
| 6 | Nguyễn Thanh Bình | x |  |
| 7 | Đỗ Quốc Huy | x |  |
| 8 | Trần Trung Kiên | x |  |
| 9 | Chử Văn Nam | x |  |
| 10 | Nguyễn Quang Thắng | x |  |

* 1. **Phân công nhiệm vụ**

1.7.1. Pha Requirement

Input: yêu cầu khách hàng, thông tin khảo sát

Output: chức năng cần thực hiện, mục tiêu của ứng dụng

| NHIỆM VỤ | Lê Mạnh Cường | Nguyễn Thanh Bình | Đỗ Quốc Huy | Trần Trung Kiên | Chử Văn Nam | Nguyễn Quang Thắng |
| --- | --- | --- | --- | --- | --- | --- |
| Danh mục tài liệu sản phẩm | S |  | S | S |  |  |
| Quy tắc đặt tên |  | S |  | S | P |  |
| Xác định mục tiêu chất lượng dự án | S |  | S |  | S | P |

1.7.2. Pha Analysis

Input: các chức năng cần thực hiện, mục tiêu

Output: thiết kế hệ thống , lựa chọn công nghệ

| NHIỆM VỤ | Lê Mạnh Cường | Nguyễn Thanh Bình | Đỗ Quốc Huy | Trần Trung Kiên | Chử Văn Nam | Nguyễn Quang Thắng |
| --- | --- | --- | --- | --- | --- | --- |
| Vai trò của việc xác định yêu cầu đối với dự án phần mềm | P |  | S | S | S | P |
| Phương pháp đánh giá chất lượng yêu cầu | S | P | S |  | S | S |
| Danh sách các yêu cầu hệ thống |  | S | P | S | P | S |
| Kết quả đánh giá chất lượng các yêu cầu | S |  | S | S | S | S |

1.7.3 Pha Design

Input: thiết kế hệ thống

Output: UI, UML, Database

| NHIỆM VỤ | Nguyễn Huy Hùng | Trần Văn Quang | Trịnh Quang Toàn | Lê Thuỳ Trang |
| --- | --- | --- | --- | --- |
| Kiến trúc tổng thể hệ thống | S | S | P | S |
| Thiết kế giao diện | S | S | P | P |
| Thiết kế CSDL | s | P | S | S |
| Thiết kế chức năng | P | S | S | S |
| Thiết kế kiểm soát | S | P | S | S |

1.7.4. Pha Implementation

Input: UI, UML, Database

Output: Application

| NHIỆM VỤ | Lê Mạnh Cường | Nguyễn Thanh Bình | Đỗ Quốc Huy | Trần Trung Kiên | Chử Văn Nam | Nguyễn Quang Thắng |
| --- | --- | --- | --- | --- | --- | --- |
| Đăng nhập, đăng xuất |  | S |  |  | S | P |
| Đăng ký | S |  |  |  | P | S |
| Tìm kiếm |  | P |  |  | S |  |
| Nghe nhạc trực tuyến | P | S |  |  | S |  |
| Tải nhạc | P | S |  | S | S |  |
| Tạo playlist | S |  |  | P | S | S |
| Chia sẻ | S | S |  |  | S | P |
| Quản lý nhạc của tôi | S |  |  |  | P | S |

1.7.5 Pha Testing

Input: Application

Output: Testing, bug?

| NHIỆM VỤ | Nguyễn Huy Hùng | Trần Văn Quang | Trịnh Quang Toàn | Lê Thuỳ Trang |
| --- | --- | --- | --- | --- |
| Tổng quan kiểm thử | P | S | S | S |
| Mô hình kiểm thử trong đề tài áp dụng | S | S | P | S |
| Kế hoạch kiểm thử | P | S | S | S |
| Kết quả kiểm thử | S | S | P | S |

* 1. **Kế hoạch triển khai chi tiết**

| **STT** | **Pha** | **Công việc** | **Người thực hiện** | **Thời gian bắt đầu** | **Thời gian kết thúc** |
| --- | --- | --- | --- | --- | --- |
| 1 | Thu thập và phân tích yêu cầu chức năng từ khách hàng và đối tác. | - Hệ thống tìm kiếm thông minh.  - Nghe nhạc trực tuyến.  - Chia sẻ âm nhạc.  - Tạo playlist.  - Gợi ý âm nhạc.  -Thông tin về nghệ sĩ và album | Lê Mạnh Cường  Trần Văn Quang  Chử Văn Nam  Nguyễn Quang Thắng | 28/2/23 | 04/03/23 |
| 2 | Thiết kế cấu trúc của phần mềm. | - Bao gồm xác định cấu trúc cơ sở dữ liệu.( Cơ sở dữ liệu sẽ lưu trữ thông tin về nhạc, file, …)  - Định nghĩa chức năng và tính năng.  - Thiết kế giao diện người dùng, định nghĩa các luồng công việc và sơ đồ tương tác giữa các thành phần trong hệ thống. | Lê Mạnh Cường  Trần Văn Quang  Trịnh Quang Toàn  Lê Thùy Trang  Nguyễn Thanh Bình | 05/03/23 | 11/03/23 |
| 3 | Thiết kế giao diện người dùng | - Thiết kế tổng thể hệ thống  - Thiết kế chi tiết:  + Thiết kế CSDL  + Thiết kế giao diện  + Thiết kế chức năng  + Thiết kế kiểm soát | Nguyễn Huy Hùng  Trần Văn Quang  Trịnh Quang Toàn  Lê Thùy Trang | 12/03/23 | 18/03/23 |
| 4 | Xây dựng phần mềm | - Bằng cách lập trình các tính năng và chức năng.  - Tạo bảng dữ liệu.  - Đảm bảo tính bảo mật  - Thực hiện kiểm thử và sửa lỗi.  - Tối ưu hóa hiệu suất và đảm bảo chất lượng. | Lê Mạnh Cường  Nguyễn Thanh Bình  Đỗ Quốc Huy  Trần Trung Kiên  Chử Văn Nam  Nguyễn Quang Thắng | 19/03/23 | 18/04/23 |
| 5 | Kiểm thử phần mềm và sửa lỗi | - Để đảm bảo rằng phần mềm hoạt động ổn định và đáp ứng được các yêu cầu chức năng.  - Test yêu cầu chức năng, phi chức năng  - Test giao diện | Nguyễn Huy Hùng  Trịnh Quang Toàn  Lê Thùy Trang | 19/04/23 | 20/04/23 |
| 6 | Trải nghiệm phần mềm với khách hàng và đối tác | - Đánh giá hiệu quả và đáp ứng các yêu cầu. | Nguyễn Huy Hùng | 21/04/23 | 23/04/23 |
| 7 | Cập nhật và bảo trì phần mềm | - Để đảm bảo tính mở rộng, tối ưu hóa hiệu suất và sửa lỗi khi cần thiết. | Chử Văn Nam  Nguyễn Quang Thắng | 24/04/23 | 24/10/23 |

* Khảo sát yêu cầu thực tế
* Phân tích yêu cầu

**Yêu cầu chức năng**

Đăng nhập và đăng ký tài khoản người dùng.

Tải lên nhạc bao gồm tên bài hát, nghệ sĩ, thể loại và mô tả.

Tìm kiếm nội dung bằng từ khóa.

Phát nhạc

Tải nhạc xuống

Danh sách nhạc

Tạo Playlist

* Phân tích thiết kế hệ thống
* Thiết kế giao diện
* Triển khai phát triển mã nguồn ứng dụng
* Kiểm thử sản phẩm

**CHƯƠNG 2. SẢN PHẨM CHUYỂN GIAO GIỮA CÁC PHA VÀ TIÊU CHUẨN ĐÁNH GIÁ**

**2.1. Danh mục tài liệu, sản phẩm**

| **STT** | | **Loại tài liệu/Sản phẩm** | | **Biểu mẫu** | | **Tiêu chuẩn đánh giá** | | **Pha**  **Requirement** | | **Pha Analysis** | | **Pha Design** | | **Pha Implementation** | | **Pha Testing** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Kết quả khảo sát thực tế về quy trình nghiệp vụ. | | [Mẫu không có tiêu đề (Câu trả lời)](https://docs.google.com/spreadsheets/d/1bKEpBqUM2Bt8lL-R4Yy5TM0yxOf1Cj0NsuigXNJ3_a0/edit#gid=1475661192) | | Đạt được các yêu cầu từ khách hàng. | | Out | | In | |  | |  | |  | |
| 2 | Xác định cơ cấu tổ chức | |  | | Người dùng và hệ thống | | Out | | In | |  | |  | |  | |
| 3 | Xác định các bên liên quan và các yêu cầu của họ | |  | | Stakeholder Request | | Out | | In | |  | |  | |  | |
| 4 | Xác định các yêu cầu chức năng và phi chức năng | | [Yêu cầu chức năng và phi chức năng](https://www.bacs.vn/vi/blog/kien-thuc/yeu-cau-chuc-nang-va-phi-chuc-nang-3910.html) | | Yêu cầu chức năng, phi chức năng phải đầy đủ, rõ ràng, sát với thực tế. | | Out | | In | |  | |  | | In | |
| 5 | Tiến trình nghiệp vụ: sơ đồ tiến trình nghiệp vụ của từng nghiệp vụ | |  | | Đặc tả chi tiết từng bước trong quy trình nghiệp vụ | | Out | | In | |  | |  | |  | |
| 6 | Sơ đồ phân rã chức năng | | [Mô hình phân rã chức năng (BFD) - LyTuong.net](https://lytuong.net/mo-hinh-phan-ra-chuc-nang-bfd/) | | Mô hình hóa được cấu trúc phần mềm. | |  | | Out | | In | | In | |  | |
| 7 | Đặc tả các chức năng | | [Cách viết đặc tả yêu cầu phần mềm đơn giản nhất](https://testerpro.vn/cach-viet-dac-ta-yeu-cau-phan-mem/) | | Đặc tả yêu cầu chức năng phải chỉ rõ được phạm vi của sản phẩm, các chức năng cần có, đối tượng người sử dụng và các ràng buộc khi vận hành sản phẩm. | |  | | Out | |  | | In | | In | |
| 8 | Phân tích hệ thống về dữ liệu | |  | |  | |  | | Out | |  | | In | |  | |
| 9 | Kiến trúc vật lý | | [Physical Architecture - SEBoK](https://www.sebokwiki.org/wiki/Physical_Architecture_Model_Development) | | Các yếu tố vật lý cụ thể phải được xác định có thể hỗ trợ các tính năng chức năng, hành vi và thời gian cũng như các thuộc tính mong đợi của hệ thống được suy ra từ các yêu cầu hệ thống phi chức năng. | |  | | Out | |  | | In | |  | |
| 10 | Kiến trúc logic phần mềm | |  | |  | |  | | Out | | In | | In | |  | |
| 11 | Mô hình CSDL | | <https://codegym.vn/blog/2020/05/11/3-buoc-chuan-hoa-nf-co-so-du-lieu/> | | Cơ sở dữ liệu phải đạt chuẩn 1NF, 2NF, 3NF. | |  | | Out | |  | | Out | | In | |
| 12 | Giao diện hệ thống | | <https://blog.webico.vn/nhu-the-nao-la-mot-giao-dien-website-than-thien-voi-nguoi-dung/> | | Bản thiết kế giao diện hệ thống (Mockup) phải dễ nhìn, thân thiện với người dùng và đầy đủ chức năng. | |  | |  | | Out | | In | | In | |
| 13 | Thiết kế kiểm soát | | <http://fit.mta.edu.vn/files/FileMonHoc/Tuan14_ThietkeHT_P2_SE15.pdf> | | Phải kiểm soát được thông tin nhập xuất, khả năng gián đoạn chương trình và khả năng bị xâm hại bởi con người.  <http://fit.mta.edu.vn/files/FileMonHoc/Tuan14_ThietkeHT_P2_SE15.pdf> | |  | |  | | Out | | In | |  | |
| 14 | Thiết kế xử lý chức năng | |  | | Đặc tả chi tiết cho từng unit của mỗi chức năng của hệ thống: input, output, process | |  | | Out | |  | | In | |  | |
| 15 | Công cụ, ngôn ngữ, môi trường lập trình | |  | |  | |  | |  | |  | | Out | |  | |
| 16 | Source Code của từng chức năng hệ thống | |  | |  | |  | |  | |  | | Out | | In | |
| 17 | Test Case | |  | |  | |  | |  | |  | |  | | Out | |
| 18 | Test Report | |  | |  | |  | |  | |  | |  | | Out | |

**2.2. Quy tắc đặt tên (naming convention)**

* Tên nên liên quan đến chức năng của phần mềm và đặc điểm của mạng xã hội âm nhạc
* Tên nên dễ nhớ và dễ phát âm để người dùng có thể tìm kiếm và chia sẻ thông tin một cách dễ dàng.
* Nếu có thể, hãy sử dụng từ ngữ tích cực và mang tính sáng tạo để tạo sự thu hút cho người dùng,
* Đặt tên nên phù hợp với thương hiệu và giá trị của Trung tâm giải trí MTP Entertainment, tạo nên sự liên kết giữa phần mềm và thương hiệu của công ty.

Các quy tắc đặt tên khi sử dụng với git:

* Khi đẩy lên git phải đặt tên commit sao cho mô tả rõ ràng những thay đổi đã được thực hiện
* Đặt tên ngắn gọn, tối đa khoảng 50 ký tự.
* Tránh sử dụng các ký tự đặc biệt hoặc emoji trong tên commit, vì nó có thể gây ra lỗi hoặc khó đọc cho người khác.
* Tránh sử dụng các ký tự đặc biệt hoặc emoji trong tên commit, vì nó có thể gây ra lỗi hoặc khó đọc cho người khác.
* Sử dụng các ký tự chữ thường và dấu gạch dưới để phân tách các từ trong tên nhánh.
* Sử dụng tiền tố để phân biệt các loại nhánh khác nhau. Ví dụ: feature/ cho nhánh phát triển tính năng, hotfix/ cho nhánh sửa lỗi.
* Nên đặt tên ngắn gọn, tối đa khoảng 20-30 ký tự.
* Đặt tên nhánh sao cho dễ nhớ và dễ hiểu, để tránh gây nhầm lẫn và khó quản lý trong tương lai.

2.3. Xác định : Mục tiêu chất lượng của dự án

| **STT** | **Tên độ đo** | **Ý nghĩa** | **Công thức/cách đo** |
| --- | --- | --- | --- |
| 1 | Độ tin cậy | Đảm bảo rằng phần mềm hoạt động một cách ổn định, không bị lỗi hoặc gián đoạn trong quá trình sử dụng | Kiểm tra, đánh giá lỗi |
| 2 | Hiệu suất | Đảm bảo rằng phần mềm hoạt động một cách nhanh chóng và đáp ứng được số lượng người dùng đồng thời lớn | Thời gian phản hồi, thời gian tải |
| 3 | Dễ sử dụng | Đảm bảo rằng phần mềm có giao diện dễ sử dụng và thân thiện với người dùng, giúp họ tìm kiếm, chia sẻ và tải nhạc một cách thuận tiện | Số lần sử dụng, kick chức năng |
| 4 | An toàn | Đảm bảo rằng phần mềm được bảo vệ chống lại các cuộc tấn công mạng và đảm bảo tính bảo mật cho thông tin người dùng | Kiểm tra lỗ hổng, bảo mật hệ thống |
| 5 | Khả năng mở rộng | Đảm bảo rằng phần mềm có thể mở rộng và mở rộng để đáp ứng nhu cầu của người dùng và công ty | Số lượng người dùng tối đa mà hệ thống có thể hỗ trợ |
| 6 | Khả năng tương thích | Đảm bảo rằng phần mềm có thể tương thích với các thiết bị và hệ điều hành khác nhau | Số lượng hệ điều hành, thiết bị hỗ trợ |
| 7 | Thẩm mỹ | Đảm bảo rằng phần mềm có giao diện đẹp mắt, tạo cảm giác thú vị cho người dùng khi sử dụng | Số lần tải lại và thoát khỏi trang do không hài lòng giao diện |
| 8 | Tính linh hoạt | Đảm bảo rằng phần mềm có thể điều chỉnh, thích ứng với thay đổi trong nhu cầu của người dùng và công ty | Số lượng tính năng được thêm vào |

**CHƯƠNG 3. XÁC ĐỊNH YÊU CẦU**

1. **Vai trò của việc Xác định yêu cầu đối với dự án phần mềm**
2. Xác định yêu cầu giúp định hình kế hoạch phát triển phần mềm:

Việc xác định yêu cầu giúp định hình kế hoạch phát triển phần mềm bằng cách xác định các tài nguyên, thời gian và ngân sách cần thiết để phát triển phần mềm.

1. Xác định yêu cầu giúp tăng tính khả thi của dự án phần mềm:

Việc xác định yêu cầu giúp đảm bảo rằng dự án phần mềm có tính khả thi cao bằng cách đảm bảo rằng các yêu cầu là khả thi và có thể đáp ứng.

1. Xác định yêu cầu giúp đảm bảo tính chính xác và đầy đủ của sản phẩm phần mềm:

Việc xác định yêu cầu giúp đảm bảo rằng sản phẩm phần mềm cuối cùng đáp ứng được các yêu cầu của khách hàng và người dùng và đạt được tính chính xác và đầy đủ.

1. Xác định yêu cầu giúp đảm bảo tính linh hoạt của sản phẩm phần mềm:

Việc xác định yêu cầu giúp đảm bảo tính linh hoạt của sản phẩm phần mềm bằng cách cho phép thay đổi yêu cầu trong quá trình phát triển phần mềm.

1. Xác định yêu cầu giúp tăng tính cạnh tranh của sản phẩm phần mềm:

Việc xác định yêu cầu giúp tăng tính cạnh tranh của sản phầm phần mềm bằng cách đảm bảo sản phẩm đáp ứng được các yêu cầu của khách hàng và người dùng.

1. **Phương pháp đánh giá chất lượng của yêu cầu**

* Phân tích yêu cầu: phân tích từng yêu cầu của phần mềm để đảm bảo chúng đầy đủ, rõ ràng, logic và đảm bảo tính khả thi trong thực tế.
* Kiểm tra độ phức tạp: Đánh giá mức độ phức tạp của yêu cầu. Nếu yêu cầu quá phức tạp thì sẽ gây khó khăn trong quá trình phát triển phần mềm.
* Kiểm tra tính chính xác: Yêu cầu phải được xác định chính xác để đảm bảo phát triển phần mềm đúng chức năng.
* Kiểm tra tính khả thi: Yêu cầu phải được đảm bảo về tính khả thi về mặt kỹ thuật, tài chính và thời gian.
* Kiểm tra tính nhất quán: Đảm bảo các yêu cầu không mâu thuẫn với nhau hoặc với các yêu cầu khác của phần mềm.
* Sử dụng phương pháp kiểm thử: Kiểm tra phần mềm thông qua các ca kiểm thử để đánh giá khả năng hoạt động và đáp ứng của phần mềm với các yêu cầu.
* Đánh giá hiệu suất: Để đánh giá hiệu suất của web, chúng ta cần đo lường thời gian tải trang, tốc độ xử lý các yêu cầu, và độ ổn định của nó
* Đánh giá khả năng mở rộng: Web cần hỗ trợ việc mở rộng tính năng để tiếp tục phát triển sau này. Chúng ta cũng cần xem xét khả năng mở rộng của các tính năng, như tính năng chia sẻ, tính năng bảo mật và các tính năng khác.
* Đánh giá tính bảo mật: Để đảm bảo an toàn của người dùng của web, chúng ta cần đảm bảo rằng các yêu cầu bảo mật được thực hiện một cách chính xác. Điều này bao gồm việc xác thực người dùng, bảo vệ dữ liệu và đảm bảo an toàn trong quá trình chia sẻ.

1. **Danh sách các yêu cầu của hệ thống:**

3.3.1 Yêu cầu chức năng:

* Đăng ký tài khoản.
* Đăng nhập tài khoản.
* Đăng xuất tài khoản.
* Tìm kiếm bài hát.
* Nghe nhạc trực tuyến.
* Tải xuống nhạc.
* Tạo danh sách phát cá nhân.
* Chia sẻ danh sách phát.
* Tải nhạc lên web.
* Sửa thông tin nhạc đã tải lên.
* Xóa nhạc đã tải lên.

3.3.2 Đặc tả chức năng:

* Đăng nhập và đăng ký tài khoản người dùng: Người dùng có thể đăng nhập vào hệ thống bằng tài khoản của mình hoặc đăng ký tài khoản mới để truy cập vào các tính năng của hệ thống.
* Đăng nhập:
* Input: username, password
* Output: kết quả đăng nhập và chuyển trang tương ứng
* Các bước xử lý:

> Người dùng nhập các thông tin tài khoản được yêu cầu

> Đảm bảo nội dung trong ô input khi nhập phải tuân thủ theo định dạng quy định của hệ thống:

Không ô nào được để trống

> Sau khi ấn button “Đăng nhập”, username và password sẽ được tiến hành xử lý trước khi gửi đi để đăng nhập vào hệ thống

Nếu chưa đúng định dạng thì sẽ hiện thông báo “Dữ liệu nhập chưa đầy đủ hoặc không hợp lệ”

Ngược lại sẽ gửi đi để tiến hành xử lý tài khoản có tồn tại trong hệ thống hay không và gửi thông báo tương ứng cho người dùng

* *Chi tiết xử lý chức năng đăng nhập:*

- Trang đăng nhập:

* + Hiển thị form đăng nhập với các trường thông tin cần nhập (tên đăng nhập, mật khẩu).
  + Kiểm tra tính hợp lệ của các thông tin nhập vào từ người dùng (độ dài, định dạng, v.v.).

- Xử lý đăng nhập:

* + Kiểm tra tính hợp lệ của thông tin đăng nhập (tên đăng nhập và mật khẩu) nhập từ người dùng. .
  + Nếu thông tin đăng nhập chính xác, hệ thống sẽ tạo phiên làm việc (session) và lưu thông tin phiên làm việc vào cơ sở dữ liệu để sử dụng trong quá trình sử dụng các tính năng của hệ thống.
  + Nếu thông tin đăng nhập không chính xác, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.

-Xử lý lỗi:

* + Hiển thị thông báo lỗi nếu tên đăng nhập hoặc mật khẩu không hợp lệ.
  + Cung cấp các gợi ý để giúp người dùng khắc phục lỗi.

-Đăng xuất:

* + Cung cấp tính năng cho phép người dùng đăng xuất khỏi hệ thống.
  + Xóa thông tin phiên làm việc khỏi cơ sở dữ liệu và hủy phiên làm việc.
* Đăng ký:
  + Input: username, password, re-enter password
  + Output: kết quả đăng ký và chuyển trang tương ứng
  + Các bước xử lý:

> Người dùng nhập các thông tin tài khoản được yêu cầu

> Đảm bảo nội dung trong ô input khi nhập phải tuân thủ theo định dạng quy định của hệ thống:

Không ô nào được để trống

Các kí tự trong username phải lớn hơn hoặc bằng 3 ký tự

Các kí tự trong password phải lớn hơn hoặc bằng 6 ký tự

Các kí tự trong password phải giống với các kí tự trong re-enter password

> Sau khi ấn button “Đăng ký”, username và password sẽ được tiến hành xử lý trước khi gửi đi để đăng nhập vào hệ thống

Nếu chưa đúng định dạng thì sẽ hiện thông báo “Dữ liệu nhập chưa đầy đủ hoặc không hợp lệ”

Ngược lại sẽ gửi đi để tiến hành xử lý tài khoản có tồn tại trong hệ thống hay không và gửi thông báo tương ứng cho người dùng

* *Chi tiết xử lý chức năng đăng ký*:

- Trang đăng ký:

* + Hiển thị form đăng ký với các trường thông tin cần nhập (tên đăng nhập, mật khẩu, email, số điện thoại, địa chỉ, v.v.).
  + Kiểm tra tính hợp lệ của các thông tin nhập vào từ người dùng (độ dài, định dạng, v.v.).
  + Thực hiện kiểm tra xác thực thông tin (số điện thoại, email) nếu có yêu cầu. .

- Xử lý đăng ký:

Lưu thông tin người dùng vào cơ sở dữ liệu.

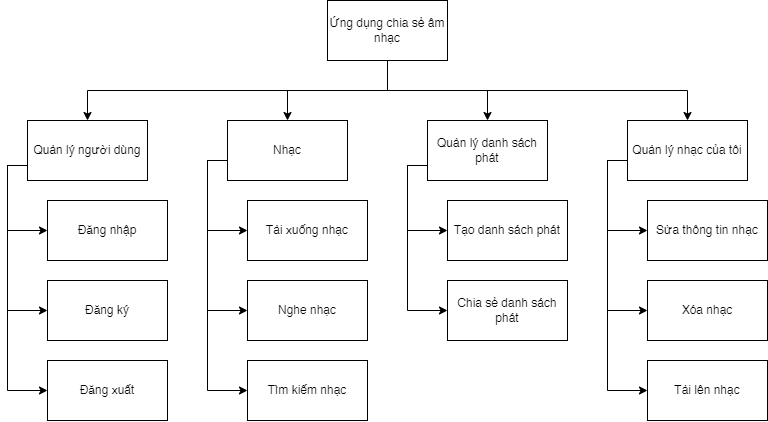
* + Tạo thông tin đăng nhập cho người dùng và lưu trữ trong cơ sở dữ liệu.
  + Gửi email xác nhận đăng ký đến người dùng (nếu cần thiết). .

-Xác nhận đăng ký:

* + Hiển thị trang xác nhận đăng ký với thông báo cho người dùng biết rằng đăng ký đã thành công
  + Cung cấp thông tin về cách sử dụng hệ thống, hướng dẫn người dùng đăng nhập.

-Xử lý lỗi:

* + Kiểm tra và thông báo lỗi đối với các trường thông tin không hợp lệ.
  + Hiển thị thông báo lỗi nếu đã có tài khoản sử dụng địa chỉ email hoặc tên đăng nhập đã đăng ký trước đó.
  + Cung cấp các gợi ý để giúp người dùng khắc phục lỗi.
* Tìm kiếm bài hát: Người dùng có thể tìm kiếm các bài hát, nghệ sĩ và album mà họ quan tâm trên thanh công cụ để nghe hoặc thêm vào danh sách phát của mình.
  + Input: các kí tự chứa tên bài hát, nghệ sĩ, hoặc album
  + Output: bài hát, nghệ sĩ hoặc album
  + Các bước xử lý:
    - Người dùng nhập thông tin bài hát, ca sĩ hoặc album muốn tìm kiếm
    - Khi người dùng nhập xong, thanh tìm kiếm sẽ tự động hiện thông tin bài hát, ca sĩ hoặc album dựa trên từ khóa tương ứng
* *Chi tiết xử lý chức năng tìm kiếm:*
  + Tìm kiếm từ khóa: Người dùng nhập từ khóa để tìm kiếm bài hát. Hệ thống sẽ tìm kiếm trong cơ sở dữ liệu của ứng dụng hoặc trang web để tìm các kết quả liên quan.
  + Phân loại kết quả: Sau khi tìm kiếm được các kết quả liên quan đến từ khóa, hệ thống sẽ phân loại các kết quả này thành các danh mục như bài hát, nghệ sĩ, album hoặc playlist.
  + Hiển thị kết quả: Hệ thống hiển thị kết quả tìm kiếm cho người dùng với các thông tin cơ bản như tên bài hát, tên nghệ sĩ, tên album, thời lượng và hình ảnh. Người dùng có thể xem các thông tin này để quyết định liệu họ có muốn nghe bài hát, nghệ sĩ hoặc album này hay không.
  + Lựa chọn kết quả: Người dùng chọn bài hát mà họ muốn nghe bằng cách nhấp vào tên bài hát.
* Nghe nhạc trực tuyến và tải nhạc: Người dùng có thể nghe nhạc trực tuyến trên hệ thống và cũng có thể tải nhạc về máy tính hoặc thiết bị di động để nghe offline.
  + Nghe nhạc trực tuyến:
    - Input: Thông tin bài hát
    - Output: âm thanh
    - Các bước xử lý:
      * Người dùng ấn vào một bài hát để bắt đầu nghe bài hát đó. Trên thanh phát nhạc, thông tin bài hát đang phát bao gồm tên bài hát, ca sĩ, hình ảnh, độ dài sẽ được hiển thị, button “Play” sẽ chuyển thành button “Pause”, thanh tiến trình bắt đầu di chuyển, thời gian phát bài hát bắt đầu chạy.
      * Người dùng có thể nhấn button “→” để chuyển bài hát tiếp theo, “←” để chuyển bài hát trước đó, ấn button “↝” để bật chế độ phát ngẫu nhiên, ấn button “↻” để bật chế độ phát lặp lại một bài hát. Để tắt chế độ phát, người dùng chỉ cần ấn 1 lần nữa vào button tương ứng
      * Người dùng có thể kéo thanh âm lượng để tùy chỉnh âm lượng của bài hát dựa trên giá trị của thanh tiến trình âm lượng.
      * Khi người dùng ấn button “Pause”, bài hát hiện tại sẽ được tạm dừng, thời gian đang phát của bài hát dừng lại.
      * Nếu bài hát không thể phát sẽ hiện lỗi lên trên màn hình
  + Tải xuống nhạc:
    - Input: thông tin bài hát
    - Output: tệp tin âm thanh định dạng .mp3
    - Các bước xử lý:
      * Khi người dùng ấn vào biểu tượng tải xuống (↓) hệ thống sẽ lấy URL của file âm thanh rồi tiến hành tải xuống
      * Nếu file tải lỗi sẽ thông báo cho người dùng
* Tạo danh sách phát cá nhân: Người dùng có thể tạo ra các danh sách phát cá nhân của riêng mình để sắp xếp các bài hát theo thứ tự mà họ muốn.
  + Input: tên danh sách phát, danh sách các bài hát
  + Output: danh sách phát bao gồm id, tên danh sách và danh sách các bài hát
  + Các bước xử lý:
    - Người dùng ấn vào button “Tạo danh sách phát” trên thanh Header
    - Người dùng nhập tên danh sách phát, sau đó tìm kiếm thông tin bài hát và button thêm vào danh sách
    - Danh sách phát sẽ tự động được lưu và xuất hiện trên thanh Header
    - Dữ liệu bao gồm ID danh sách phát, tên danh sách, danh sách bài hát sẽ được gửi lên hệ thống để lưu trữ
* Chia sẻ danh sách phát: Người dùng có thể chia sẻ danh sách phát của mình với người dùng khác để thể hiện sở thích âm nhạc và tạo mối quan hệ xã hội.
  + Input: ID danh sách phát
  + Output: URL của danh sách phát
  + Các bước xử lý:
    - Người dùng chọn danh sách phát muốn chia sẻ và nhấn button “ < ”
    - ID của danh sách phát sẽ được gửi lên hệ thống để trả về URL của danh sách phát
    - URL danh sách phát sẽ tự động được lưu vào bộ nhớ tạm thời và thông báo cho người dùng
* Quản lý danh sách nhạc đã tải lên:
* Tải nhạc lên web.
  + Input: Tên nhạc, mô tả, file nhạc
  + Output: Bài nhạc tải lên web
  + Các bước xử lý:
    - Người dùng muốn tải nhạc trước tiên phải đăng nhập.
    - Khi người dùng đã đăng nhập thì mới hiện ra một menu chức năng, người dùng chọn chức năng “đăng tải nhạc” Web sẽ chuyển sang page đăng nhạc.
    - Ở đây người dùng phải điền đủ thông tin như: tên nhạc và mô tả.
    - Trước khi đăng nhạc người dùng phải chọn file nhạc và nhấn nút tải nó nên.
    - Khi đã có file nhạc và các thông tin của nhạc đầy đủ thì lúc này nút đăng nhạc mới sáng lên và có thể sử dụng.
    - Nhấn vào đăng nhạc để đăng tải nhạc.
    - Khi nhạc đăng thành công sẽ chuyển qua trang nhạc của tôi để quản lý nhạc.
* Danh sách nhạc đã tải lên.
  + Input: yêu cầu xem danh sách, tài khoản đăng nhập.
  + Output: Danh sách nhạc đã đăng tải của tài khoản đang đăng nhập.
  + Các bước xử lý:
    - Người dùng nhấn vào nút nhạc của tôi trên menu, web sẽ chuyển qua trang nhạc của tôi.
    - Ở đây sẽ yêu cầu hệ thống lấy ra danh sách nhạc đã đăng tải của user đang đăng nhập.
* Sửa thông tin nhạc đã tải lên.
  + Input: yêu cầu sửa, tên nhạc, mô tả.
  + Output: Nhạc đã được sửa thông tin.
  + Các bước xử lý:
    - Trong trang nhạc của tôi có 1 button sửa, người dùng nhấn vào button sửa thì web sẽ được chuyển qua trang sửa nhạc.
    - Ở trang này thì chỉ hiển thị tên nhạc và mô tả để sửa, người dùng nhập đầy đủ thông tin cần sửa.
    - Xác nhận các thông tin cần sửa không bị trống thì thì nút cập nhật hoạt động, người dùng nhấn vào nút cập nhật để cập nhật thông tin nhạc.
* Xóa nhạc đã tải lên:
  + Input: yêu cầu xóa, id nhạc phải xóa
  + Output: xóa bài nhạc
  + Các bước xử lý:
    - Người dùng nhấn vào nút xóa nhạc để yêu cầu xóa nhạc.
    - Hệ thống hiển thị thông báo xác nhận.
    - Người dùng nhấn nút đóng thì sẽ hủy yêu cầu, người dùng nhấn nút đồng ý thì yêu cầu xóa sẽ được gửi đi
    - Server tiếp nhận yêu cầu đó và xóa nhạc có id gửi lên.
* Sơ đồ phân rã chức năng:



Ảnh 3.1. Sơ đồ phân rã chức năng

3.3.3 Yêu cầu phi chức năng:

**Yêu cầu phi chức năng**

Hiệu suất: Phần mềm phải có khả năng xử lý đồng thời nhiều yêu cầu từ nhiều người dùng.

Độ tin cậy: Phần mềm phải đảm bảo tính bảo mật và tránh các lỗi xảy ra trong quá trình sử dụng.

Khả năng mở rộng: Phần mềm phải có khả năng mở rộng để đáp ứng nhu cầu của người dùng trong tương lai.

Khả năng bảo trì: Phần mềm phải dễ dàng bảo trì và cập nhật khi có các yêu cầu mới.

**Yêu cầu kỹ thuật**

Phần mềm được phát triển bằng ngôn ngữ lập trình JavaScript.

Phần mềm phải được phát triển dựa trên mô hình kiến trúc MVC.

Phần mềm phải có khả năng hoạt động trên nhiều nền tảng, bao gồm Windows, MacOS và các thiết bị di động iOS và Android.

Các dữ liệu và tài liệu phải được lưu trữ trên máy chủ với cơ sở dữ liệu MySQL hoặc MongoDB.

Giao diện người dùng phải được thiết kế đơn giản, dễ sử dụng và tương thích với các trình duyệt web phổ biến như Google Chrome, Mozilla Firefox, và Safari.

Phần mềm phải có khả năng phân quyền người dùng để giới hạn truy cập đến các tính năng cần thiết.

Phần mềm phải được thiết kế với khả năng mở rộng và có thể tích hợp các tính năng mới sau này.

**Yêu cầu thực hiện**

Xây dựng mô hình kiến trúc của phần mềm hiện đại, thuận tiện cho việc nâng cấp.

Thực hiện thiết kế giao diện người dùng và tính năng phân quyền người dùng.

Lập trình các tính năng của phần mềm bằng ngôn ngữ oop.

Thiết kế cơ sở dữ liệu và cài đặt hệ quản trị cơ sở dữ liệu.

Kiểm thử và sửa lỗi phần mềm trước khi triển khai sản phẩm.

Triển khai phần mềm trên máy chủ và các thiết bị người dùng.

Cung cấp hướng dẫn sử dụng và hỗ trợ khách hàng trong quá trình sử dụng phần mềm.

**Yêu cầu đảm bảo chất lượng**

Đảm bảo tính bảo mật và độ tin cậy của phần mềm trong quá trình sử dụng.

Đảm bảo tính tương thích và hiệu suất của phần mềm trên các nền tảng khác nhau.

Đảm bảo tính ổn định và độ tin cậy của phần mềm trong quá trình triển khai và sử dụng.

Đảm bảo tính mở rộng và tính năng tương thích của phần mềm để đáp ứng nhu cầu người dùng trong tương lai.

Đảm bảo tính dễ bảo trì và cập nhật của phần mềm khi có các yêu cầu mới.

Đảm bảo tính tuân thủ các tiêu chuẩn và quy định về phát triển phần mềm như ISO 12207 và CMMI.

Đảm bảo tính đáp ứng yêu cầu của khách hàng và các bên liên quan đến sản phẩm.

Đảm bảo tính đúng hạn và hiệu quả của quá trình phát triển và triển khai sản phẩm.

Đảm bảo tính hiệu quả và hiệu suất của phần mềm, đặc biệt là khi có nhiều người dùng truy cập cùng lúc.

Đảm bảo tính dễ sử dụng và tương tác với các hệ thống khác, đặc biệt là các ứng dụng và dịch vụ liên quan đến âm nhạc.

Đảm bảo tính tương thích với các thiết bị và công nghệ mới, đặc biệt là các thiết bị và công nghệ âm thanh.

Đảm bảo tính bảo mật và riêng tư của thông tin người dùng và các dữ liệu liên quan đến âm nhạc.

**Yêu cầu triển khai**

Triển khai phần mềm trên một máy chủ đáp ứng yêu cầu về hiệu suất và tài nguyên của hệ thống.

Cài đặt và cấu hình cơ sở dữ liệu để đảm bảo tính bảo mật và hiệu suất của phần mềm.

Đảm bảo tính tương thích của phần mềm với các thiết bị và hệ điều hành khác nhau.

Cung cấp hướng dẫn cho quản trị viên hệ thống để triển khai, cài đặt và bảo trì phần mềm.

Cung cấp hỗ trợ kỹ thuật và giải đáp thắc mắc cho khách hàng và người dùng trong quá trình triển khai và sử dụng phần mềm.

**Yêu cầu bảo trì**

Đảm bảo tính liên tục và ổn định của phần mềm trong quá trình sử dụng.

Kiểm tra và sửa lỗi phần mềm trong quá trình triển khai và sử dụng sản phẩm.

Cập nhật phần mềm để đáp ứng các yêu cầu mới của khách hàng và các bên liên quan.

Hỗ trợ khách hàng và người dùng trong quá trình sử dụng phần mềm.

Bảo trì và nâng cấp hệ thống để đảm bảo tính tương thích và hiệu suất của phần mềm.

Bảo mật và bảo vệ thông tin người dùng và dữ liệu liên quan đến âm nhạc.

Đảm bảo tính linh hoạt và mở rộng của phần mềm để đáp ứng nhu cầu phát triển trong tương lai.

Đảm bảo tính tương thích của phần mềm với các phiên bản mới của hệ điều hành và các ứng dụng liên quan.

Cung cấp hỗ trợ kỹ thuật và giải đáp thắc mắc cho khách hàng và người dùng trong quá trình sử dụng và bảo trì phần mềm.

Thực hiện kiểm tra, đánh giá và nâng cấp tính năng và hiệu suất của phần mềm để đáp ứng các yêu cầu mới của khách hàng và người dùng.

Thực hiện sao lưu và phục hồi dữ liệu đảm bảo tính toàn vẹn và khả năng phục hồi dữ liệu trong trường hợp xảy ra sự cố hoặc thiết bị bị hỏng.

Đảm bảo tính tương thích và hỗ trợ các cập nhật và bảo mật mới cho phần mềm.

=> Kết luận:

* Giao diện người dùng đơn giản, dễ làm quen, nắm bắt và sử dụng với mọi lứa tuổi sử dụng : Để tránh gây rắc rối cho người dùng mới và không gây ảnh hưởng tới trải nghiệm của họ
* Hiển thị đầy đủ thông tin về bài hát và đánh giá bài hát : Người dùng có thể dựa vào các thông tin đó để tìm ra thể loại nhạc yêu thích của mình
* Tích hợp với các dịch vụ âm nhạc khác: Hệ thống cần tích hợp với các dịch vụ âm nhạc khác để đáp ứng nhu cầu của người dùng và tăng tính linh hoạt.
* Tính năng tương thích với các thiết bị di động: Hệ thống cần hỗ trợ tương thích đa nền tảng, hoạt động cả trên các thiết bị khác

1. **Kết quả đánh giá chất lượng các yêu cầu**

* **Tính chính xác:** Hệ thống đã đạt được những đặc điểm kỹ thuật và chức năng cơ bản cần có của hệ thống quản lý quán café qua việc tham khảo từ những nguồn thông tin chính xác và đáng tin cậy kết hợp với việc tìm hiểu thực tế.
* **Tính rõ ràng:** Tài liệu đã đạt được tính rõ ràng khi đã giải thích được các định nghĩa sử dụng của hệ thống khi đã nêu được chi tiết những từng nội dung đặc tả yêu cầu của hệ thống và ở việc giải thích các thuật ngữ có trong tài liệu.
* **Tính hoàn thiện:** Tài liệu đã đạt được tính hoàn thiện qua việc miêu tả được phần lớn nội dung từ phần tổng quan cho đến các chức năng của hệ thống về các yêu cầu cơ bản của một mạng xã hội âm nhạc với các yêu cầu phi chức năng và các yêu cầu về thiết kế của hệ thống cũng đã được tài liệu nhắc tới đầy đủ. Tuy nhiên phần phân bổ các yêu cầu ở trong phần tổng quan của tài liệu thì lại chưa được hoàn thiện.
* **Tính nhất quán:** Tài liệu đã đạt được tính nhất quán qua việc đối với mỗi một mục đặc tả yêu cầu chức năng hệ thống thì đều nêu những nội dung chung (sử dụng chung 1 format) bao gồm các phần như mô tả, điều kiện tiên quyết, dữ liệu đầu vào, hoạt động, dữ liệu đầu ra và các yêu cầu khác.
* **Độ quan trọng và độ ổn định:** Tài liệu vẫn chưa nêu được về xếp hạng được độ quan trọng và độ ổn định của hệ thống
* **Có thể kiểm chứng:** Tài liệu cơ bản đã đạt được về mức độ có thể kiểm chứng nội dung có trong tài liệu qua việc người đọc có thể đọc những nguồn tài liệu được tham khảo bởi SRS được liệt kê ở phần tài liệu tham khảo để có thể đối chiếu nội dung có trong SRS và xác định được tính đúng sai.
* **Có thể sửa đổi:** Tài liệu đã đạt được khả năng có thể sửa đổi.
* **Có thể truy xuất nguồn gốc:** Tài liệu đã đạt được khả năng có thể truy xuất nguồn gốc.

**CHƯƠNG 4. KIẾN TRÚC PHẦN MỀM**

* 1. **Vai trò của kiến trúc phần mềm với quá trình phát triển dự án phần mềm**
* Các kiến trúc không phải là phần mềm hoạt động. Thay vào đó, nó là một đại diện cho phép một kỹ sư phần mềm:

1. Phân tích hiệu quả của thiết kế trong việc đáp ứng các yêu cầu đề ra,

2. Xem xét lựa chọn thay thế kiến trúc khi thay đổi thiết kế vẫn tương đối dễ dàng,

3. Giảm thiểu rủi ro gắn liền với việc xây dựng các phần mềm.

* Đại diện của kiến trúc phần mềm là một tạo khả năng cho truyền thông giữa tất cả các bên (các bên liên quan) quan tâm đến sự phát triển của một hệ thống dựa trên máy tính.
* Những kiến trúc làm nổi bật thiết kế quyết định ban đầu mà sẽ có một tác động sâu sắc trên tất cả các công việc kỹ thuật phần mềm sau và, quan trọng hơn, vào sự thành công cuối cùng của hệ thống như là một thực thể hoạt động.
* Kiến trúc "tạo thành một mode minh bạch tương đối nhỏ về cách hệ thống được cấu trúc và cách các thành phần của nó làm việc cùng nhau”
  1. **Một số kiến trúc phần mềm phổ biến**

4.2.1. Kiến trúc phân lớp (Layered Architecture)

* Cấu trúc cơ bản của kiến trúc phân lớp được minh họa

trong hình bên.

* Nhiều lớp khác nhau được xác định, mỗi hoạt động hoàn thành dần dần trở nên gần gũi hơn với tập lệnh của máy.
* Ở lớp bên trong, thành phần dịch vụ giao diện người dùng hoạt động.
* Ở lớp bên ngoài, các thành phần thực hiện giao tiếp hệ điều hành.
* Tầng trung gian cung cấp các dịch vụ tiện ích và chức năng phần mềm ứng dụng.

4.2.2. Client - Server

* Mô hình Client-Server là một mô hình nổi tiếng trong mạng máy tính, được áp dụng rất rộng rãi và là mô hình của mọi trang web hiện có.
* Ý tưởng của mô hình này là máy con (đóng vai trò là máy khách) gửi một yêu cầu (request) để máy chủ (đóng vai trò server cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách.
* Trong mô hình này, chương trình ứng dụng được chia làm 2 phần:
* Client (máy khách): bao gồm máy tính và các thiết bị điện tử nói chung
* Server (máy chủ): là nơi cài đặt các chương trình dịch vụ và lưu trữ tài nguyên

4.2.3. Model - View - Controller (MVC)

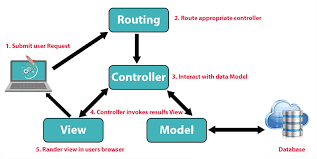
* Kiến trúc MVC là một trong số các mô hình cơ sở hạ tầng WebApp được đề xuất tách giao diện người dùng khỏi chức năng WebApp và nội dung thông tin.
* Mô hình chứa tất cả nội dung dành riêng cho ứng dụng và logic xử lý, bao gồm tất cả các đối tượng nội dung, quyền truy cập vào các nguồn thông tin/dữ liệu bên ngoài và tất cả chức năng xử lý dành riêng cho ứng dụng.
* Khung nhìn chứa tất cả các chức năng cụ thể của giao diện và cho phép trình bày nội dung và logic xử lý, bao gồm tất cả các đối tượng nội dung, truy cập vào các nguồn thông tin/dữ liệu bên ngoài và tất cả chức năng xử lý theo yêu cầu của người dùng cuối.
* Bộ điều khiển quản lý truy cập đến mô hình và khung nhìn và điều phối luồng dữ liệu giữa chúng.
* Trong một WebApp, “chế độ xem được cập nhật bởi bộ điều khiển với dữ liệu từ mô hình dựa trên đầu vào của người dùng” [WMT02].

4.2.4. Kiến trúc lấy dữ liệu làm trung tâm

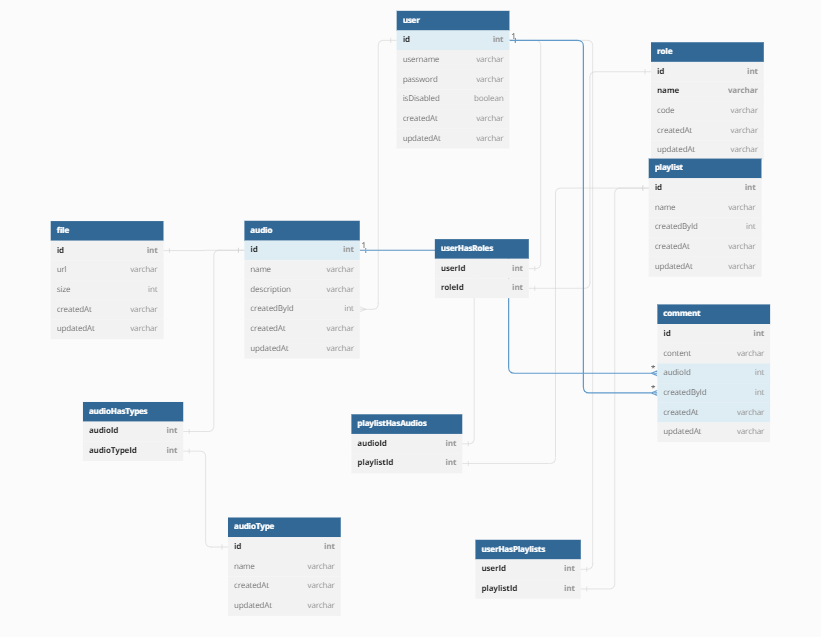
* Là một kho lưu trữ dữ liệu (ví dụ: tệp hoặc cơ sở dữ liệu) cư trú tại trung tâm của kiến trúc này và được truy cập thường xuyên bởi các thành phần khác mà cập nhật, thêm, xóa hoặc sửa đổi dữ liệu trong kho lưu trữ.
* Phần mềm máy khách truy cập kho lưu trữ trung tâm.
* Trong một số trường hợp, kho lưu trữ dữ liệu bị động. Đó là, phần mềm máy khách truy cập vào dữ liệu độc lập với bất kỳ thay đổi nào đối với dữ liệu hoặc hành động của phần mềm máy khách khác.

4.2.5. Kiến trúc hướng sự kiện

* Các thành phần của một hệ thống đóng gói dữ liệu và các hoạt động phải được áp dụng để thao tác dữ liệu.
* Giao tiếp và sự phối hợp giữa các thành phần được thực hiện thông qua việc truyền tin nhắn.
  1. **Kiến trúc tổng thể của hệ thống**

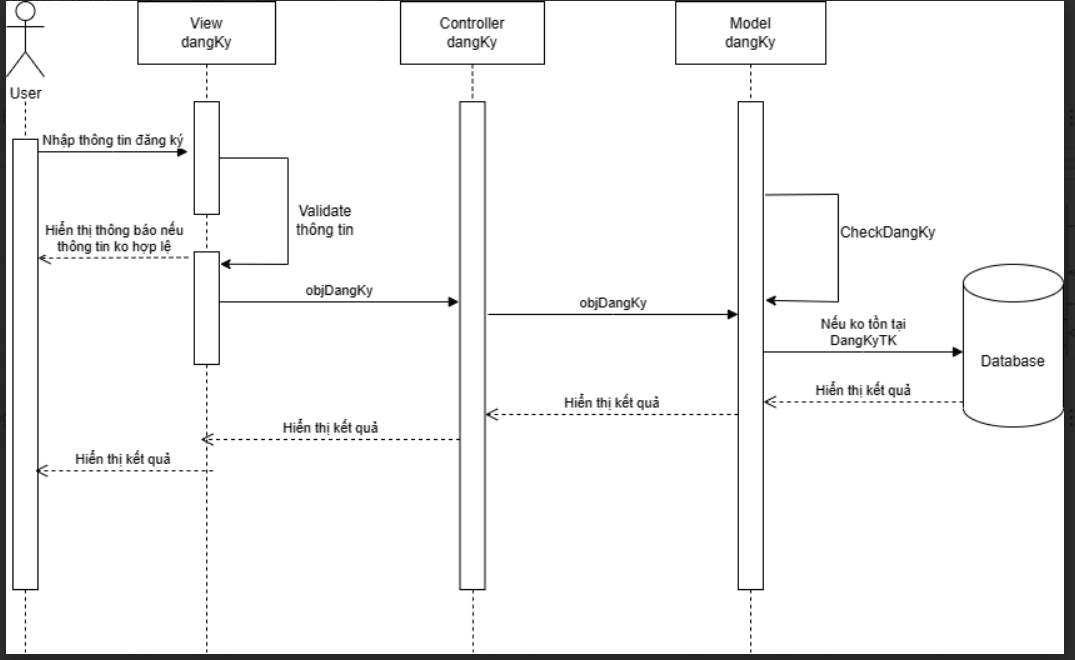


* 1. **Lựa chọn công nghệ phát triển cho hệ thống**
* Theo mức độ hiện đại của công nghệ, theo nhóm người hưởng lợi mục tiêu, theo nguồn lực đầu vào và theo sự hòa hợp; dự án đã lựa chọn các công nghệ sau để phát triển cho hệ thống:
* Node js
* React js
* My SQL
  1. **Kiến trúc phần mềm của hệ thống**
* Dự án lựa chọn mô hình kiến trúc MVC
* Lí do:
* Quy trình phát triển nhanh hơn: MVC hỗ trợ việc phát triển nhanh chóng và song song. Nếu một mô hình MVC được dùng để phát triển bất kỳ ứng dụng web cụ thể nào, một lập trình viên có thể làm việc trên **View** và một lập trình viên khác có thể làm việc với **Controller** để tạo logic nghiệp vụ cho ứng dụng đó. Do đó, việc sử dụng mô hình kiến trúc MVC có thể hoàn thanh nhanh hơn nhiều so với việc sử dụng các mô hình khác.
* Đối với bất kỳ ứng dụng nào, người dùng sẽ có xu hướng thay đổi thường xuyên. Với mô hình kiến trúc MVC, việc thêm tính năng mới vào ứng dụng có thể được thực hiện dễ dàng bằng cách thêm các thành phần mới mà không ảnh hưởng đến các thành phần khác trong ứng dụng.
* Mô hình MVC giúp phân tách các thành phần của ứng dụng, tạo điều kiện cho việc phát triển độc lập và kiểm thử. Nó cũng giúp tăng tính bảo mật và dễ dàng bảo trì ứng dụng.
* Khả năng tương thích: Mô hình MVC có thể được triển khai trên nhiều nền tảng khác nhau, giúp cho ứng dụng có thể hoạt động trên nhiều thiết bị và hệ điều hành khác nhau.
* MVC hỗ trợ phát triển các trang web thân thiện với SEO. Bằng nền tảng này, ta có thể dễ dàng phát triển các URL thân thiện với SEO để tạo ra nhiều lượt truy cập vào trang web của chúng ta hơn.
* Khả năng cung cấp nhiều chế độ view: Trong mô hình MVC, ta có thể tạo nhiều view cho một mô hình. Ngày nay, nhu cầu có thêm nhiều cách để truy cập ứng dụng đang càng ngày tăng.Do đó, việc sử dụng mô hình MVC để phát triển là một giải pháp tốt.
* MVC Model trả về dữ liệu mà không cần định dạng: MVC **pattern** có thể trả về dữ liệu mà không cần áp dụng bất kỳ định dạng nào. Do đó, các thành phần giống nhau có thể được sử dụng với bất kỳ giao diện nào.
  1. **Cơ sở dữ liệu mức vật lý của hệ thống**



Ảnh 4.1. Cơ sở dữ liệu mức vật lý của hệ thống

* 1. **Một số thiết kế chi tiết theo kiến trúc được chọn**
* **Đăng ký:**



Ảnh 4.2. UML chức năng đăng ký tài khoản

| STT | Class | Method | INPUT | OUTPUT |
| --- | --- | --- | --- | --- |
| 1 | VDangKy | doCreate | Gmail,mật khẩu | Đăng ký tài khoản thành công/không thành công |
| 2 | vDangKy | Validate | taiKhoan,matKhau | Invalid data/valid |
| 3 | ControllerDangKy | objDangKy | Username,matKhau | Đăng ký tài khoản thành công/không thành công |
| 4 | ModelDangKy | checkDangKy | Username,matKhau | Xem tài khoản đã tồn tại hay chưa |
| 5 | ModelDangKy | DangKyTK | Username,MatKhau | Thêm tk vào database |

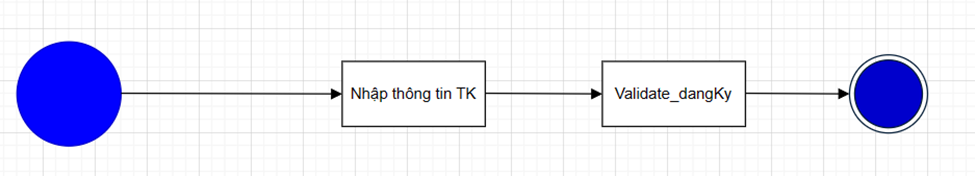
vDangKy:doCreate

o Mục đích :Nhập thông tin đăng ký và gửi yêu cầu đăng ký

o Input: username,mật khẩu

o Output: Cấp tài khoản thành công(không thành công)

o Process:



Ảnh 4.3. Xử lý chi tiết view Đăng ký: doCreate

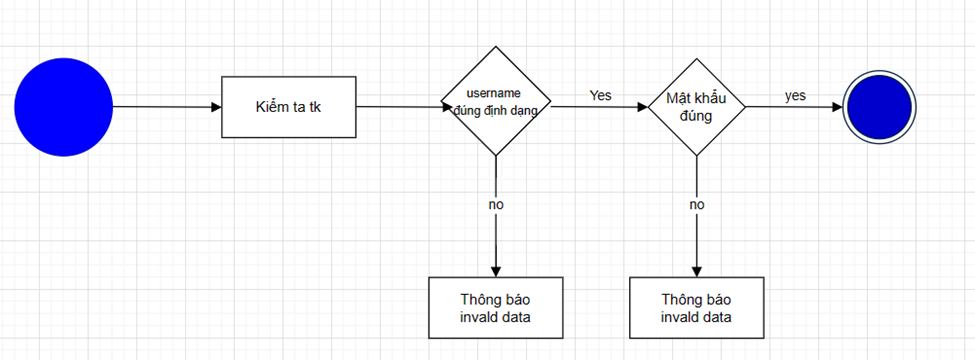
vDangKy:ValidateTaiKhoan

o Mục đích:Kiểm tra tính đúng đắn của dữ liệu.

o Input: username,mật khẩu

o Output:Valid/invalid

o Process:



Ảnh 4.4. Xử lý chi tiết view Đăng ký: validateTaiKhoan

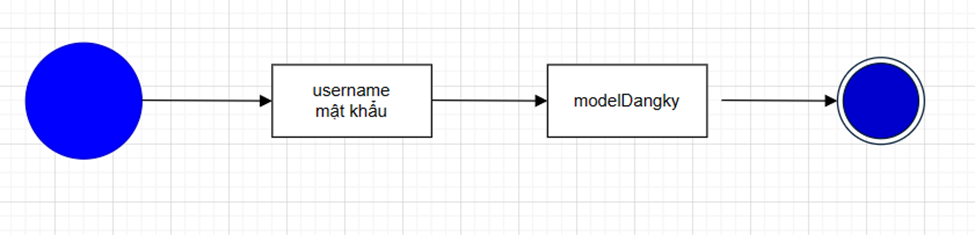
· ControlerdangKy:objDangKy

o Mục đích:chuyển dữ liệu vào modelDangKy

o Input: username,mật khẩu

o Output:Đăng ký thành công(không thành công)

o Process:



Ảnh 4.5. Xử lý chi tiết controller Đăng ký: objDangKy

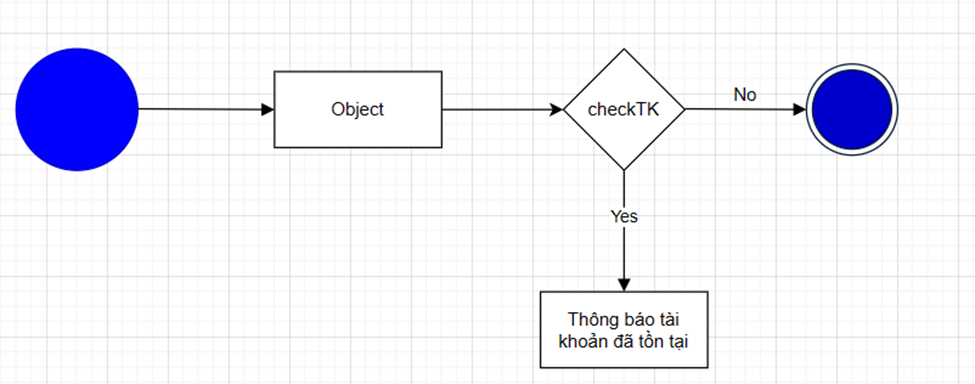
· ModelDangKy:checkDangKy

o Mục đích:kiểm tra xem tài khoản có tồn tại hay không

o Input:Username

o Output:Kiểm tra xem tài khoản có tồn tại không.

o Process:



Ảnh 4.6. Xử lý chi tiết model Đăng ký: checkDangKy

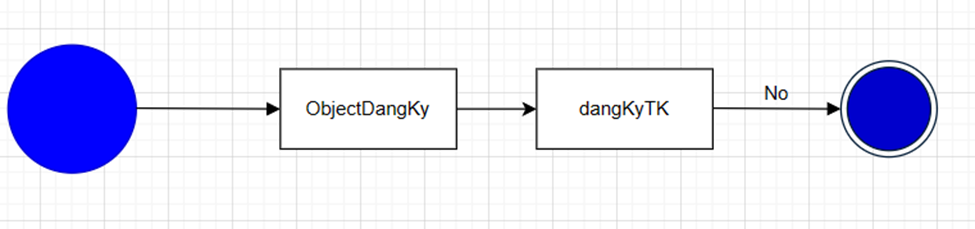
· ModelDangKy:DangKyTK

o Mục đích:Thêm tài khoản vào database

o Input:username,mật khẩu

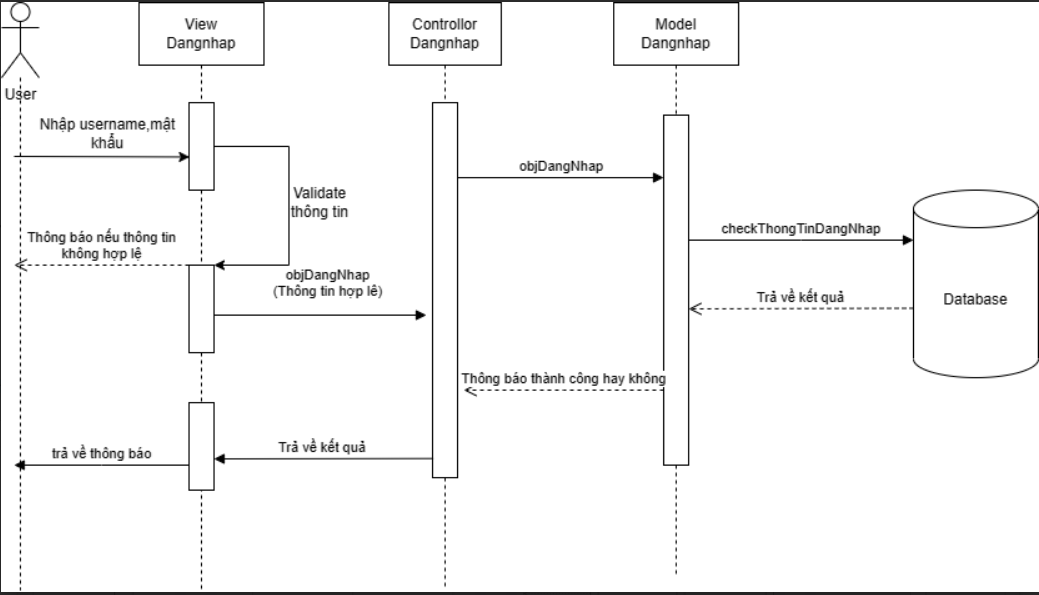
o Output:Thêm tài khoản vào database

o Process:



Ảnh 4.7. Xử lý chi tiết Model Đăng ký: DangKyTK

* **Đăng nhập:**



Ảnh 4.8. UML chức năng Đăng nhập

| Stt | Class | Method | Input | Output |
| --- | --- | --- | --- | --- |
| 1 | viewDangNhap | doCreate | Username,password | Đăng nhập thành công/không thành công |
| 2 | viewDangNhap | Validate | Username,password | Invalid data/ valid |
| 3 | controllerDangnhap | objDangNhap | Username,password | Đăng nhập thành công/không thành công |
| 4 | modelDangnhap | checkDangNhap | Username,password | taiKhoan có(không) trong database |
| 5 | modelDangNhap | dangNhap | Username,password |  |

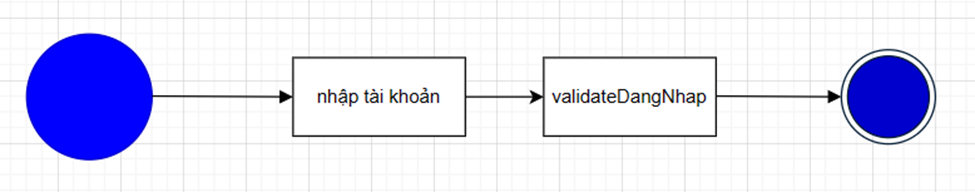
· viewDangNhap:doCreate:

o Mục đích:nhập tài khoản mật khẩu.

o Input:username,password

o Output:Thông báo đăng nhập (không) thành công

o Process



Ảnh 4.9. Xử lý chi tiết view Đăng nhập: doCreate

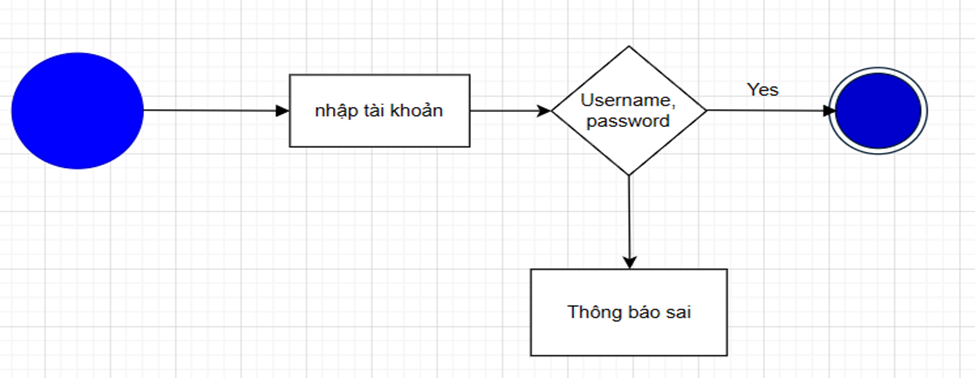
· viewDangNhap: Validate

o Mục đích:Kiểm tra tính đúng đắn của dữ liệu

o Input:username,password

o Output:Thông báo Invalid data/ valid

o Process



Ảnh 4.10. Xử lý chi tiết view Đăng nhập: validate

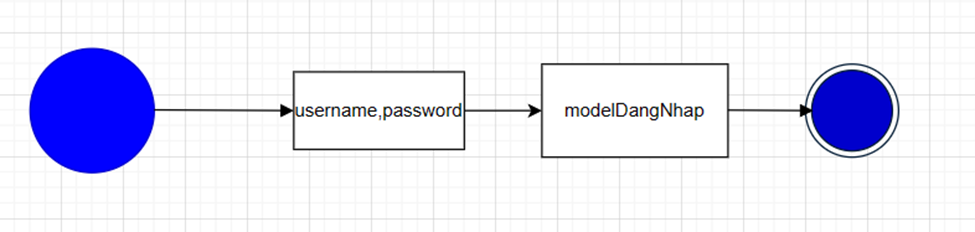
· ControllerDangNhap:objDangNhap

Mục đích:chuyển dữ liệu vào

Input:Username,password

Output:Thông báo đăng nhập(không )thành công

Process:



Ảnh 4.11. Xử lý chi tiết controller Đăng nhập: objDangNhap

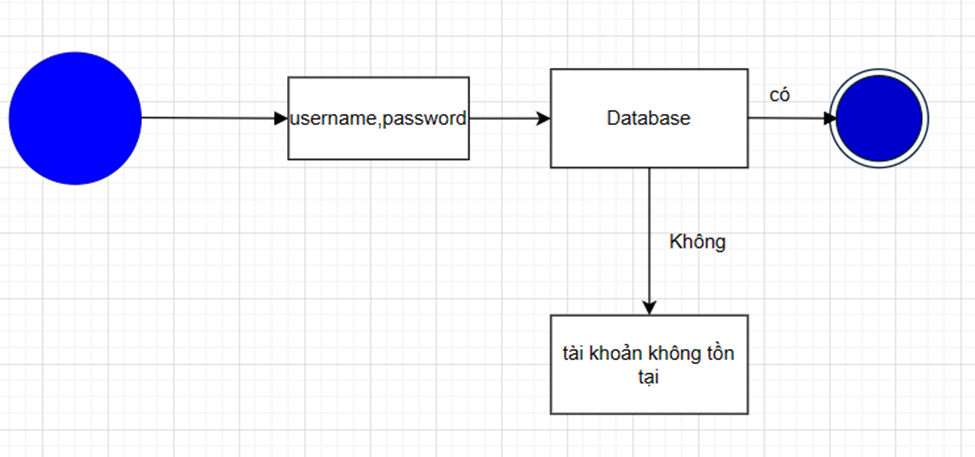
· ModelDangnhap:checkTaiKhoan

o Mục đích:Kiểm tra xem tài khoản có tồn tại trong database không:

o Input:username,password

o Output:Tài khoản (không)tồn tại

o Process:



Ảnh 4.12. Xử lý chi tiết model Đăng nhập: checkTaiKhoan

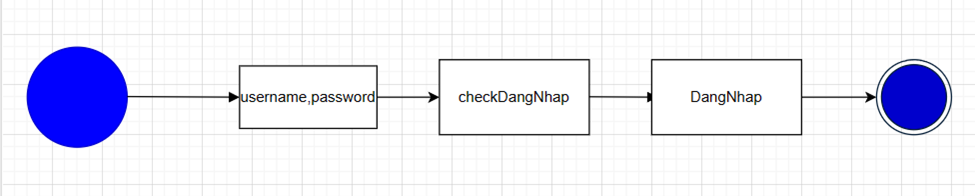
· ModelDangNhap:dangNhap

o Mục đích:đăng nhập tài khoản vào hệ thống

o Input:Username,password

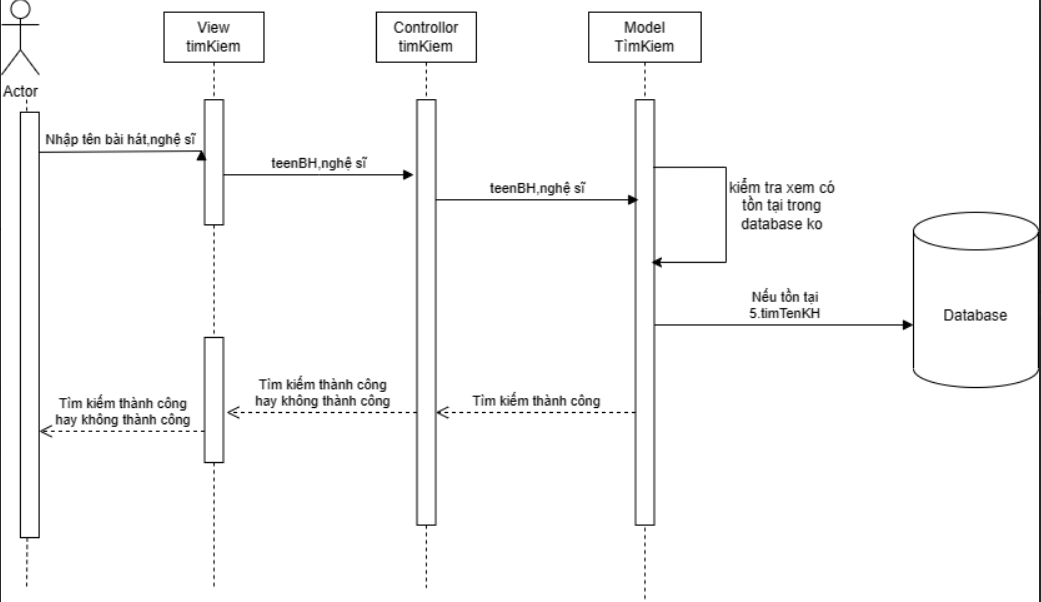
o OutPut:Đăng nhập thành công,không thành công

o Process:



Ảnh 4.13. Xử lý chi tiết model Đăng nhập: dangNhap

* **Tìm kiếm:**



Ảnh 4.14. UML chức năng tìm kiếm

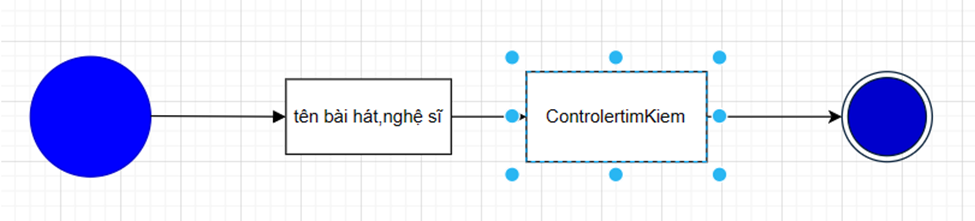
| STT | Class | Method | Input | Output |
| --- | --- | --- | --- | --- |
| 1 | viewTimkiem | getTimKiem | Tên bài hát,nghệ sĩ | Chuyển sang controller |
| 2 | ControlerTimKiem | objTimKiem | Tên bài hát,nghệ sĩ | Chuyển input sang model |
| 3 | ModelTimKiem | checkTimKiem | Tên bài hát,nghệ sĩ | Dữ liệu có tồn tại hay không |
| 4 | ModelTimKiem | timTen | Tên bài hát,nghệ sĩ | Hiện bài hát,nghệ sĩ cần tìm |

viewTimKiem:getTimKiem

· Input:Tên bài hát,nghệ sĩ cần tìm kiếm.

· OutPut: Chuyển sang controller

· Process:



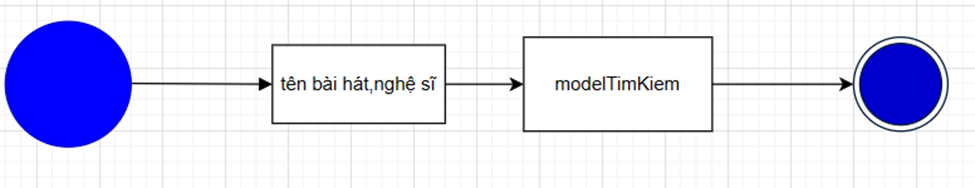
Ảnh 4.15. Xử lý chi tiết view Tìm kiếm: getTimKiem

ControllerTimKiem;objTimKiem

· Input:Tên bài hát,nghệ sĩ cần tìm kiếm

· Output:Chuyển sang Model

· Process:



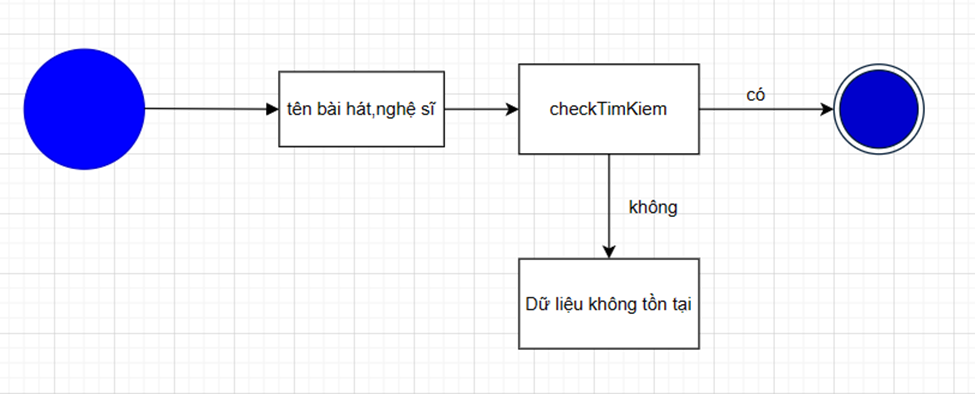
Ảnh 4.16. Xử lý chi tiết controller Tìm kiếm: objTimKiem

modelTimKiem:CheckTimKiem

· Input:Tên bài hát,nghệ sĩ

· Output:Dữ liệu có tồn tại hay không

· Process:



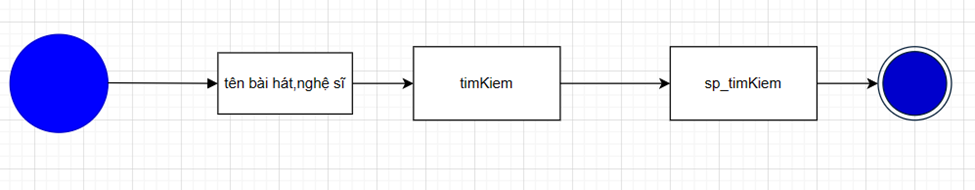
Ảnh 4.17. Xử lý chi tiết model tìm kiếm: CheckTimKiem

modelTimKiem:timKiem

· Input:tên bài hát,nghệ sĩ

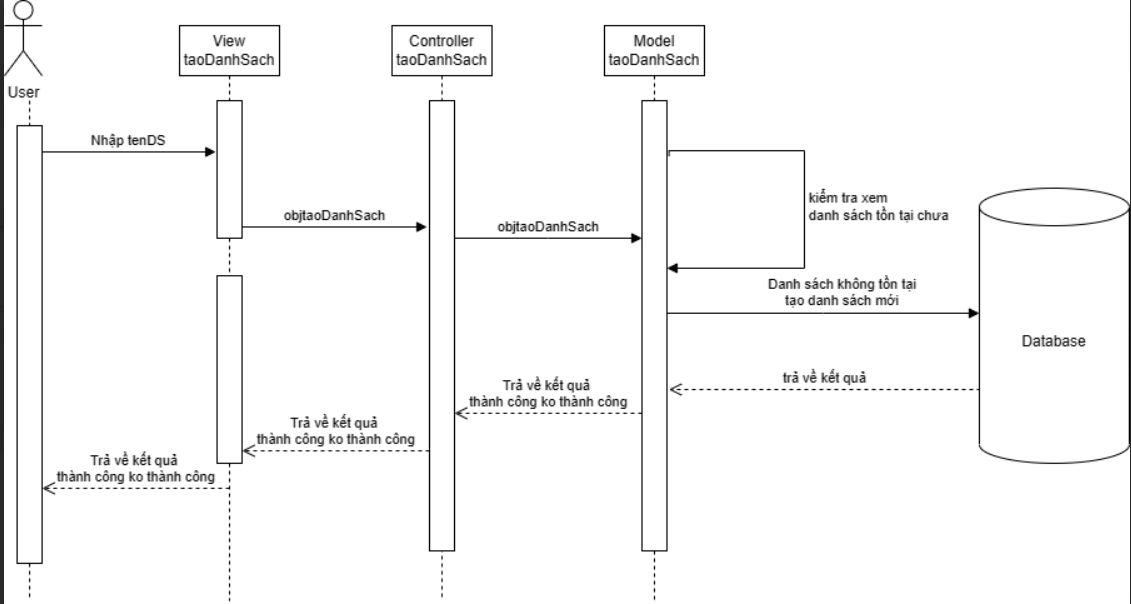
· Output:Hiện tên bài hát,nghệ sĩ cần tìm/

· Process:



Ảnh 4.18. Xử lý chi tiết model Tìm kiếm: timKiem

* **Tạo danh sách nhạc:**



Ảnh 4.19. UML chức năng tạo danh sách nhạc

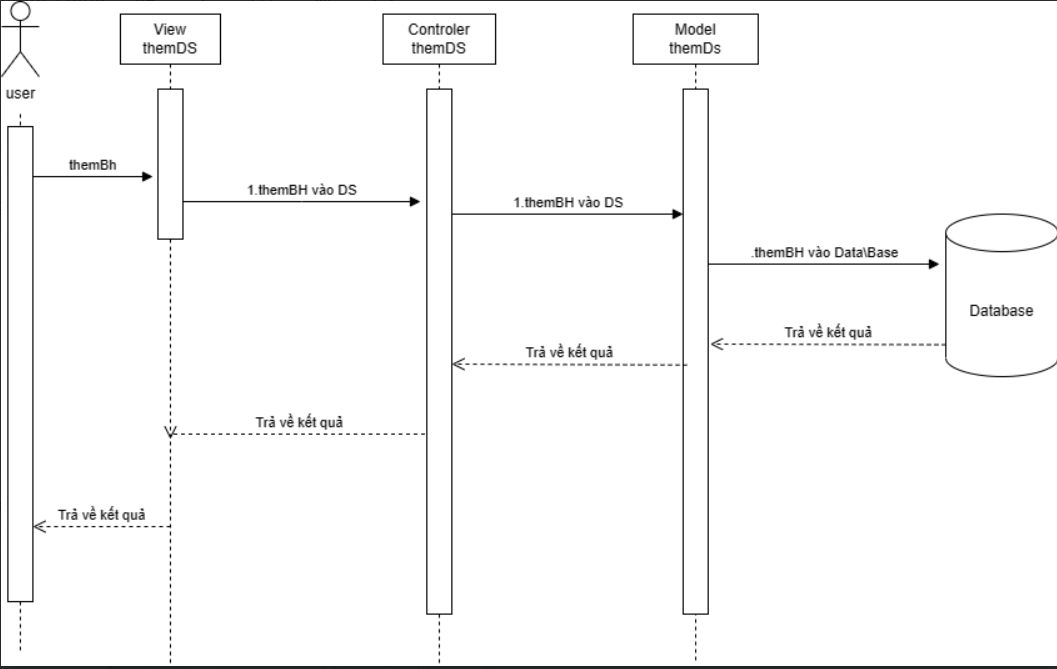
* Mục đích: Thêm danh sách phát
* Input: Thực thi mTaoDanhSachPhat
* Output: msg tạo danh sách thành công
* Quá trình xử lý:

Diagram, schematic

Description automatically generated

Ảnh 4.20. Xử lý chi tiết model tạo danh sách phát: mTaoDanhSachPhat

* **Thêm bài hát vào danh sách:**



Ảnh 4.21. UML chức năng thêm bài hát vào danh sách

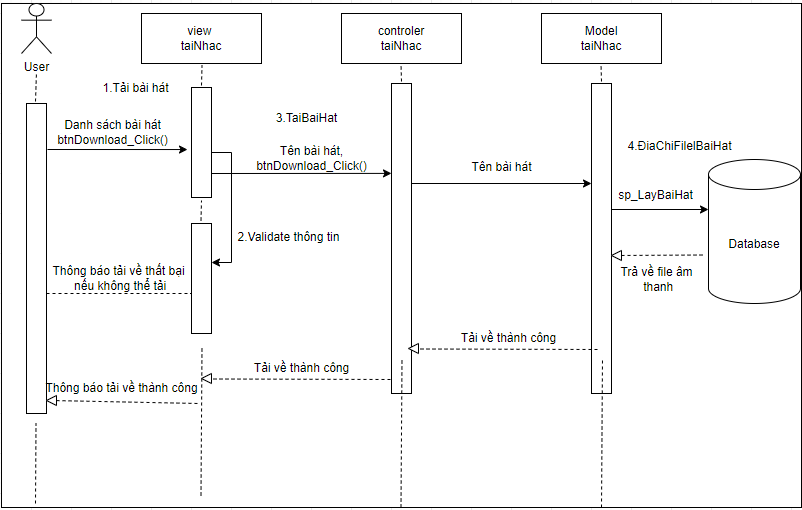
* Mục đích: Thêm bài hát vào danh sách phát
* Input: objBaiHat, objDanhSachPhat
* Output: msg thêm danh sách phát thành công (đã tồn tại trong danh sách phát)
* Quá trình xử lý:

Diagram

Description automatically generated

Ảnh 4.22. Xử lý chi tiết Thêm bài hát vào danh sách phát: objBaiHat, objDanhSachPhat

* **Tải nhạc:**

****

Ảnh 4.23. UML chức năng tải xuống nhạc

Từ sơ đồ trên, ta thấy được các unit cần có để tải nhạc:

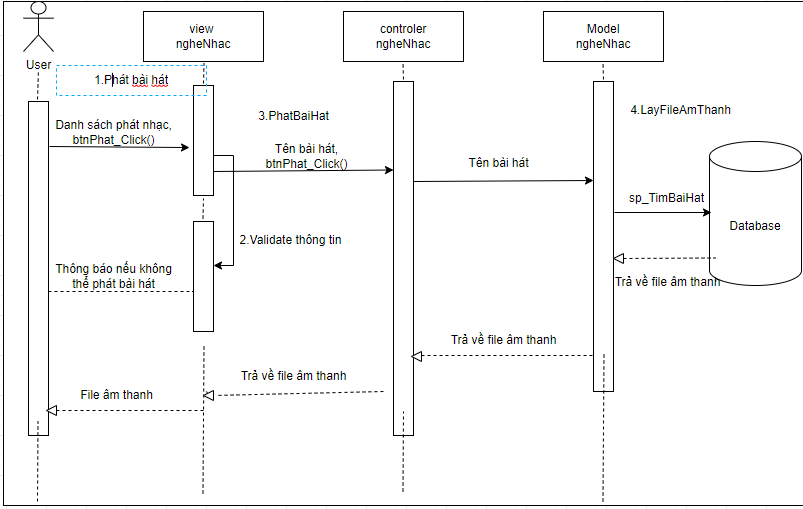
* Mục đích: Tải xuống bài hát được chọn
* Input: objBaihat
* Output: msg do db trả về
* Quá trình xử lý:

Diagram

Description automatically generated

Ảnh 4.24. Xử lý chi tiết Tải xuống nhạc: objBaihat

* **Phát nhạc:**

****

Ảnh 4.25. UML chức năng phát nhạc

Từ sơ đồ trên, ta thấy được các unit cần có để nghe nhạc:

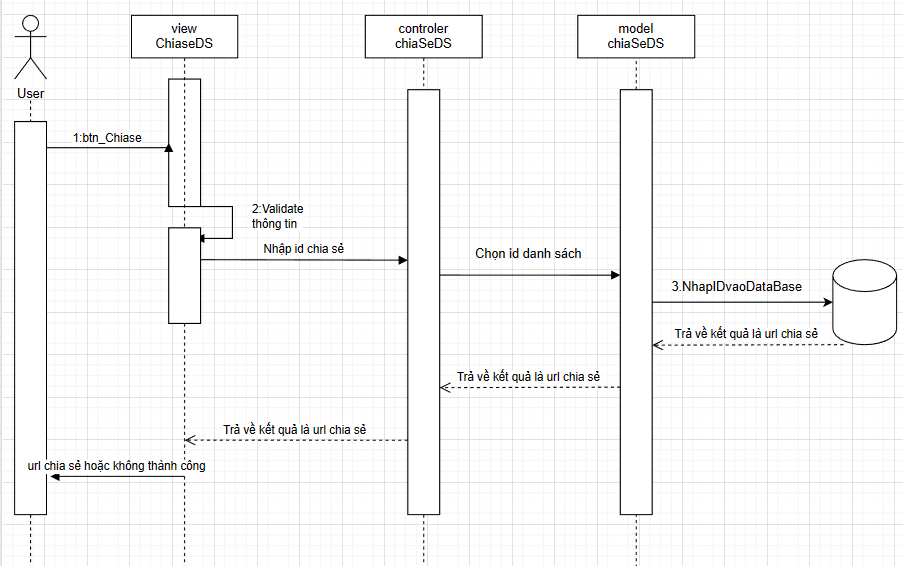
* Mục đích: Phát bài hát được chọn
* Input: objBaiHat
* Output: File âm thanh
* Quá trình xử lý:

Diagram

Description automatically generated

Ảnh 4.26. Xử lý chi tiết Phát nhạc: objBaiHat

* **Chia sẻ danh sách phát:**



Ảnh 4.27: UML chức năng chia sẻ danh sách phát

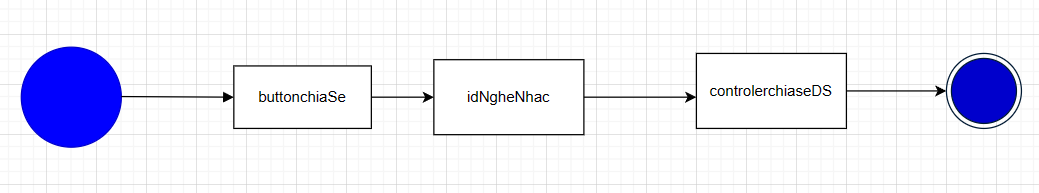
| STT | Class | Method | Input | OutPut |
| --- | --- | --- | --- | --- |
| 1 | viewChiaSeDS | Btn\_chiaSe | Danh sách phát | ID\_Danhsach |
| 2 | controlerChiaSeDS | Validate\_thongtin | Id\_danhsach | Quyền chia sẻ có hay không |
| 3 | modelChiaSeDS | NhapIDvaodDB | Id\_Danhsach | url chia sẻ |

viewChiaSeDS:btn\_chiaSe

Input: Danh sách phát

Output:ID\_Dnhsach

Process:



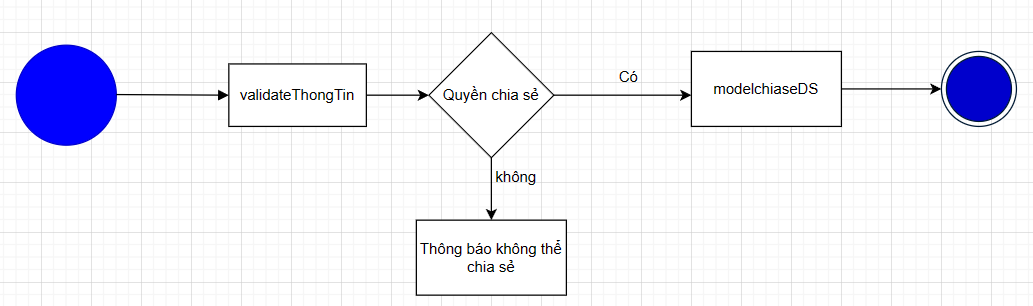
Ảnh 4.28. Xử lý chi tiết view chia sẻ danh sách phát: viewChiaSeDS

controlerChiaSeDS:ValidateThongTin

Input:Id\_DanhSach

Output:có quyền chia sẻ không

Process:



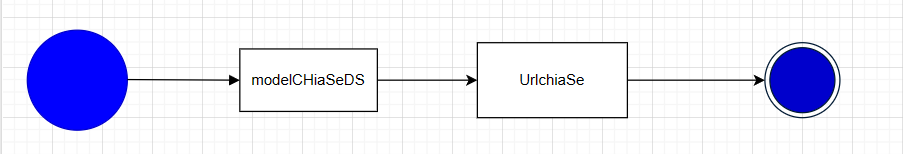
Ảnh 4.29. Xử lý chi tiết controlerChiaSeDS:ValidateThongTin

ModelChiaSeDS:NhapIDvaoDB

Input:ID\_danhsach

Output:urldanhsachPhat

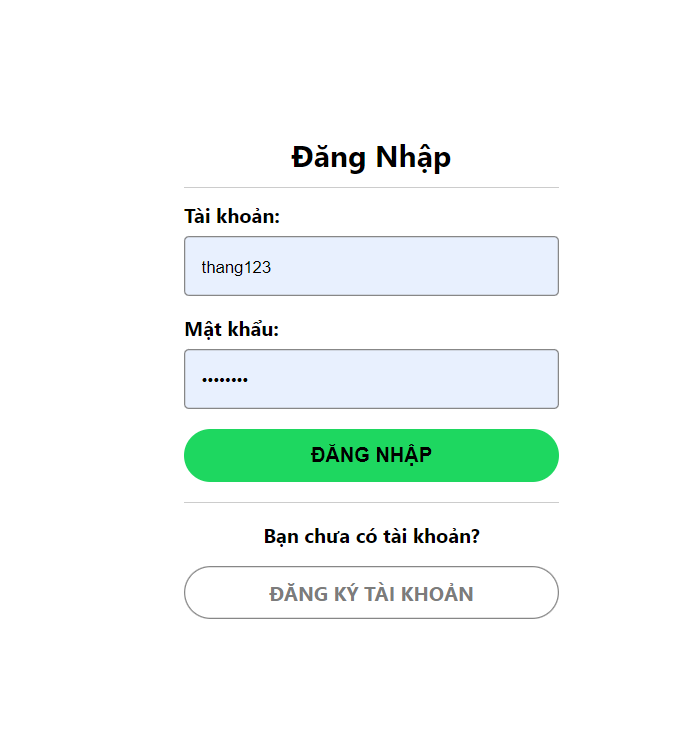
Process:



Ảnh 4.30. Xử lý chi tiết modelChiaSeDS: NhapIDvaoDB

**CHƯƠNG 5. XÂY DỰNG VÀ KIỂM THỬ HỆ THỐNG**

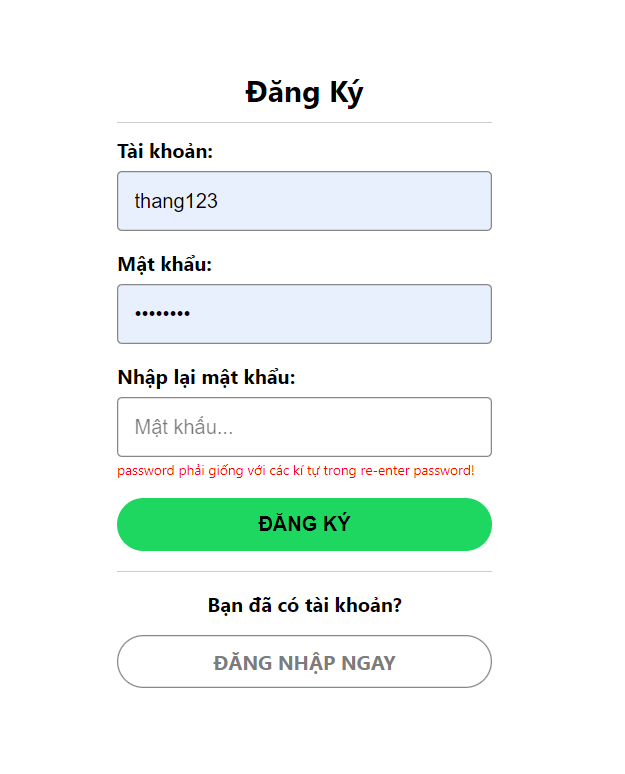
* 1. **Kết quả xây dựng 01 module/nhóm chức năng**
     1. Đăng nhập, đăng ký



Ảnh 5.1. Kết quả xây dựng chức năng đăng nhập

Input: Tài khoản, password mà người dùng đã đăng ký từ hệ thống

Output: Sẽ có quyền truy cập để tải nhạc, đăng nhạc và lưu nhạc

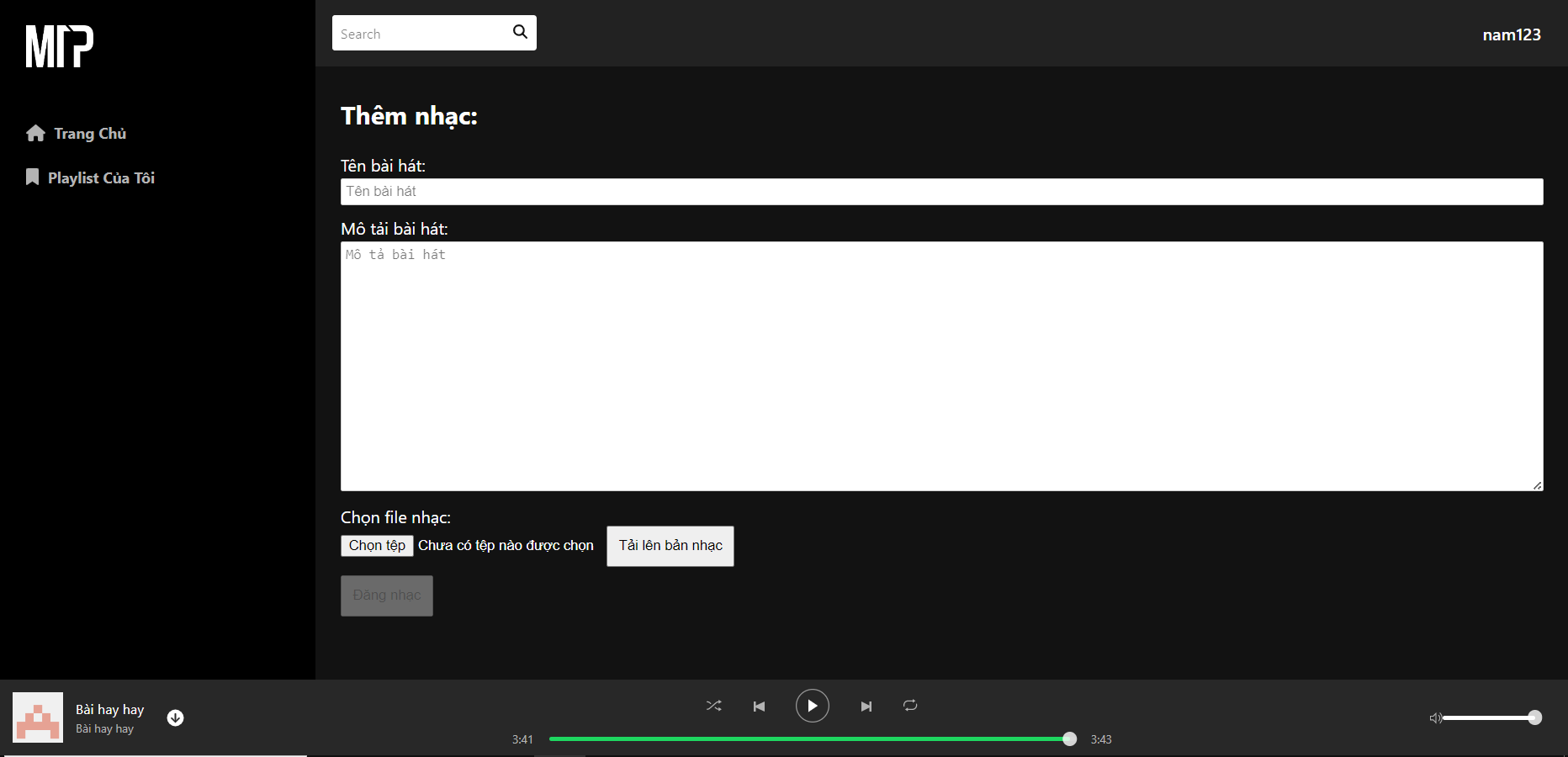


Ảnh 5.2. Kết quả xây dựng chức năng đăng ký

Input: Tài khoản, password mà người dùng muốn đăng nhập vào hệ thống

Output: Dùng làm tài khoản đăng nhập vào hệ thống sử dụng các chức năng

* + 1. Tải lên nhạc



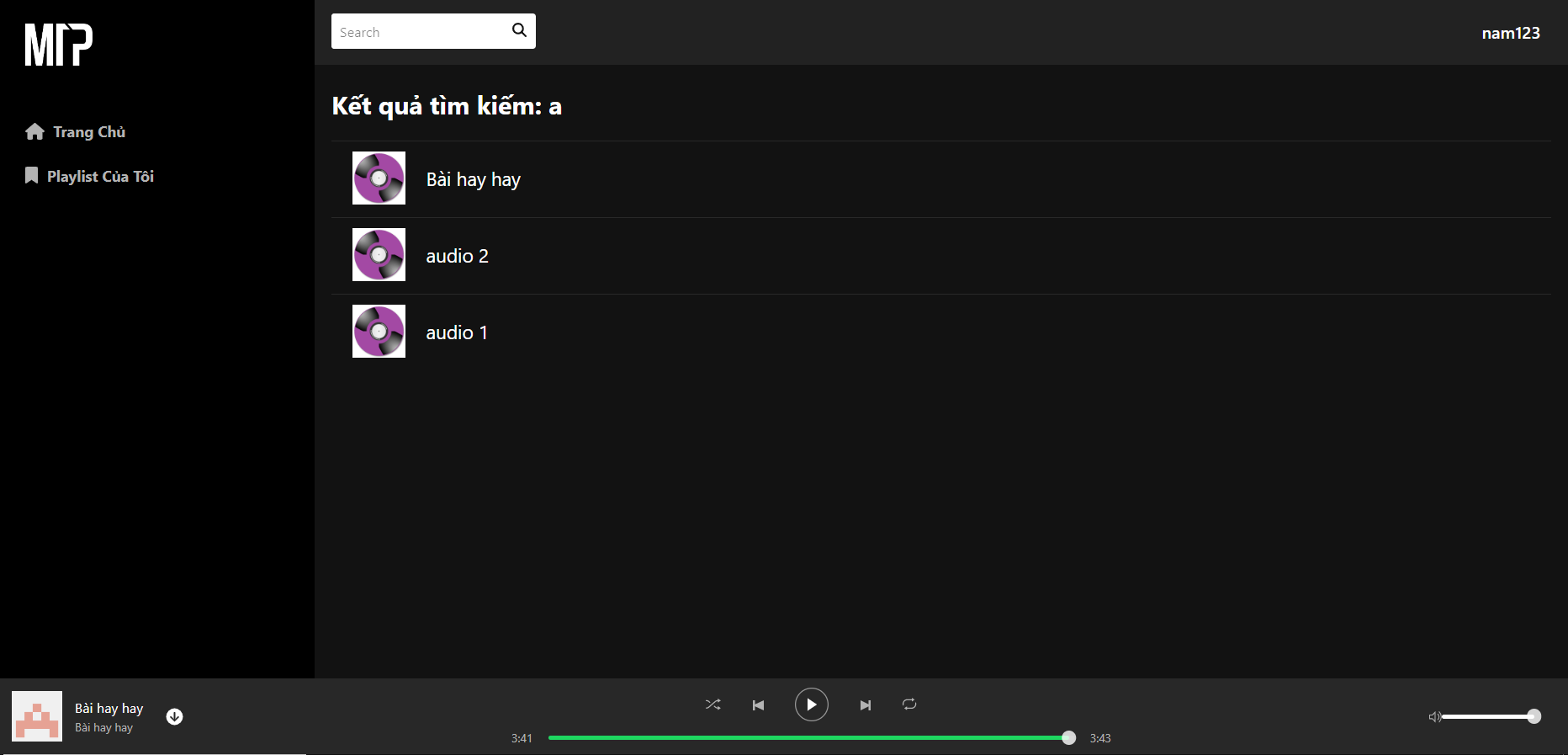
Ảnh 5.3. Kết quả xây dựng chức năng tải lên nhạc

Input:

* Tệp âm thanh: Đây là tệp nhạc mà người dùng muốn tải lên. Định dạng tệp âm thanh có thể là mp3, wav, flac, ogg, và nhiều định dạng âm thanh khác.
* Thông tin liên quan đến tệp âm thanh: Đây có thể là tiêu đề bài hát, tên ca sĩ, thể loại, hình ảnh đại diện, mô tả, và các thông tin khác liên quan đến tệp âm thanh.

Output:

* Thông báo hoàn tất tải lên: Chức năng tải lên nhạc sẽ xử lý tệp âm thanh được tải lên từ người dùng và lưu trữ tệp đó trên máy chủ hoặc nơi lưu trữ tương ứng.
* Liên kết/URL: Chức năng tải lên nhạc sẽ tạo ra một liên kết/URL đến tệp âm thanh đã được tải lên. Liên kết này có thể được chia sẻ cho người dùng hoặc được sử dụng để nhúng vào trang web hoặc ứng dụng khác để phát lại bài hát.
* Thông tin về tệp âm thanh: Chức năng tải lên nhạc cũng có thể cung cấp thông tin về tệp âm thanh đã được tải lên, chẳng hạn như tên tệp, định dạng, kích thước, thời lượng, và các thông tin metadata khác liên quan đến tệp âm thanh.
  + 1. Tìm kiếm nội dung từ từ khóa



Ảnh 5.4. Kết quả xây dựng chức năng tìm kiếm

Input:

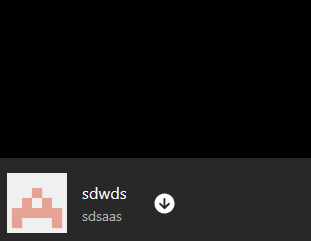
* Từ khóa tìm kiếm: Đây là từ hoặc cụm từ mà người dùng muốn sử dụng để tìm kiếm trong tập dữ liệu hoặc nguồn thông tin cụ thể. Từ khóa có thể là một từ đơn, một cụm từ, hoặc một danh sách các từ khóa liên quan.

Output:

* Kết quả tìm kiếm: Chức năng tìm kiếm từ khóa sẽ trả về các kết quả là một danh sách bài hát liên quan đến từ khóa đã được nhập, dựa trên nguồn dữ liệu được cung cấp.
  + 1. Tải nhạc xuống và phát nhạc

Input: Người dùng tải nhạc thông qua phần mềm mạng xã hội chia sẻ âm nhạc SOUNDFRIENDS.

* *Tải nhạc xuống*



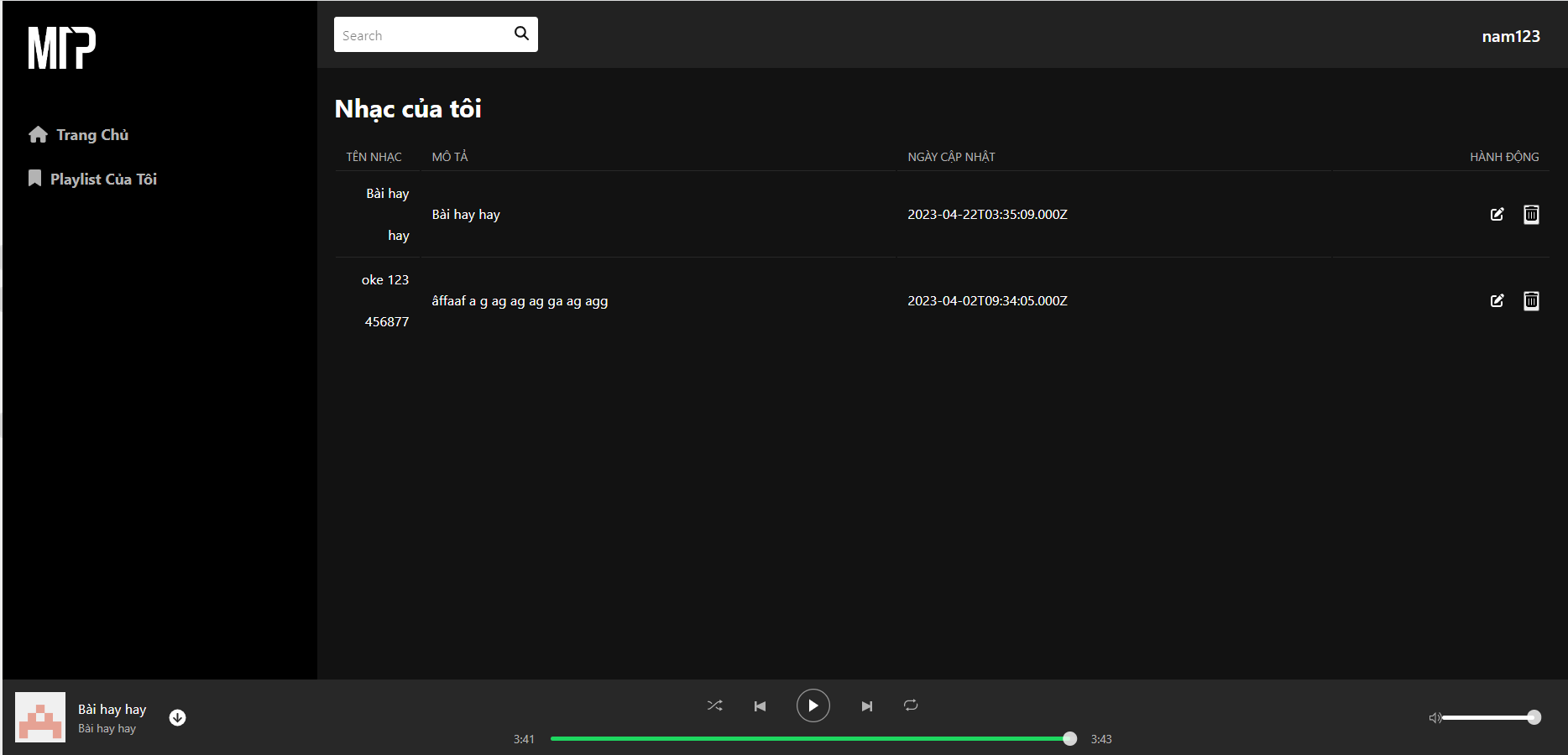
Ảnh 5.5. Kết quả xây dựng chức năng tải xuống nhạc

1. Mở ứng dụng và đăng nhập vào tài khoản của mình.
2. Tìm kiếm bài hát muốn tải xuống.
3. Bấm phát nhạc và bấm vào mũi tên chỉ xuống
4. Chờ đợi cho việc tải xuống hoàn tất.

* *Phát nhạc*

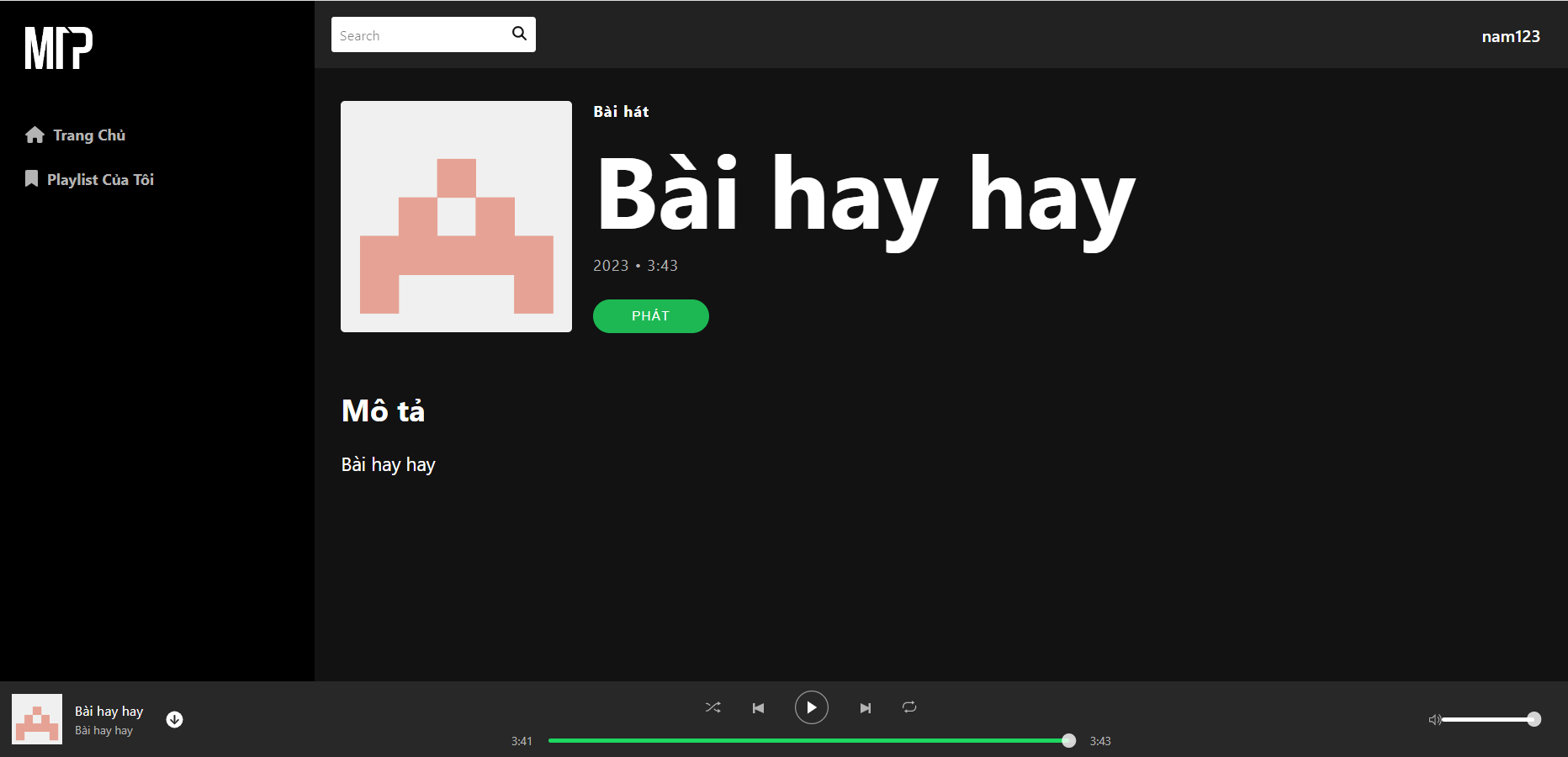
Output: File nhạc được tải xuống thành công và lưu vào file Nhạc của tôi

1. Sau khi việc Tải xuống được hoàn tất, vào phần Nhạc của tôi.



Ảnh 5.5. Kết quả xây dựng chức năng phát nhạc

1. Chọn bài hát vừa tải xuống để phát nhạc.



Ảnh 5.6. Kết quả xây dựng chức năng phát nhạc

* 1. **Kiểm thử hệ thống**
     1. **Tổng quan về kiểm thử phần mềm**

**5.2.1.1. Khái niệm kiểm thử**

* **Kiểm thử phần mềm:** là quá trình phân tích một sản phẩm phần mềm để phát hiện sự khác biệt giữa các điều kiện hiện có và yêu cầu (tức là các lỗi) và đánh giá các tính năng của sản phẩm phần mềm.***[1]***
* **Kế hoạch kiểm thử (Test plan):** là một tài liệu mô tả chi tiết các hoạt động kiểm thử được thực hiện trong quá trình kiểm thử phần mềm. Theo Pressman, Test plan là một tài liệu dựng lên từ kế hoạch kiểm thử, miêu tả cách thức kiểm thử, tiêu chí kiểm thử, phạm vi kiểm thử, các thành phần cần thiết cho kiểm thử, lịch trình kiểm thử, người thực hiện kiểm thử và cách đánh giá kết quả kiểm thử.***[1]***

Test plan phải được xác định trước khi bắt đầu thực hiện kiểm thử phần mềm và phải được thực hiện theo kế hoạch được phê duyệt. Nó cũng đóng vai trò rất quan trọng trong việc đảm bảo chất lượng của sản phẩm phần mềm, đồng thời giúp cho việc kiểm soát chi phí và thời gian cho các hoạt động kiểm thử.***[1]***

**5.2.1.2. Mục đích kiểm thử**

Quá trình kiểm thử nhằm mục đích:

* Phát hiện và sửa lỗi, khiếm khuyết của phần mềm để đảm bảo tính đúng đắn, chất lượng và tin cậy của phần mềm trong quá trình hoạt động
* Giảm rủi ro và đáp ứng yêu cầu của người dùng/tổ chức sử dụng phần mềm

**5.2.1.3. Các mức độ kiểm thử**

* **Kiểm thử đơn vị (Unit test):** là quá trình kiểm thử các đơn vị phần mềm nhỏ nhất như hàm, phương thức hay lớp để đảm bảo chức năng của chúng đáp ứng được các yêu cầu đã đặt ra và đảm bảo tính nhất quán trong các phương thức và lớp.***[1]***
* **Kiểm thử tích hợp (Integration test):** là một kỹ thuật có hệ thống để xây dựng cấu trúc chương trình đồng thời tiến hành kiểm tra để phát hiện ra các lỗi liên quan đến giao diện.

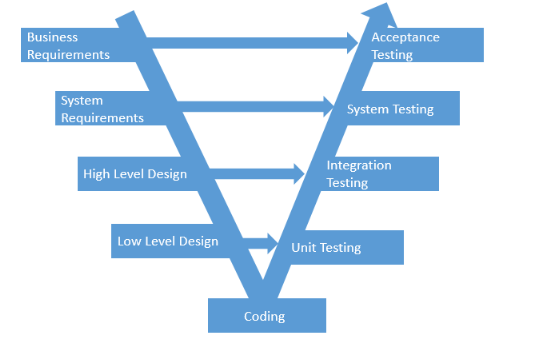
Mục tiêu là lấy các thành phần được kiểm tra đơn vị và xây dựng cấu trúc chương trình đã được quyết định bởi thiết kế.***[1]***

* **Kiểm thử hệ thống (System test):** là quá trình kiểm thử toàn bộ hệ thống phần mềm để đảm bảo chức năng và tính đúng đắn của hệ thống trước khi triển khai nó vào môi trường thực tế. Kiểm thử hệ thống sử dụng các kỹ thuật kiểm thử phần mềm đa dạng như kiểm thử hộp đen, kiểm thử hộp trắng và kiểm thử hiệu năng để đánh giá hệ thống trong nhiều khía cạnh.***[1]***
* **Kiểm thử chấp nhận người dùng (User acceptance test):** là quá trình kiểm thử cuối cùng, được thực hiện bởi khách hàng hoặc người sử dụng cuối để đảm bảo rằng phần mềm đáp ứng đầy đủ các yêu cầu và kỳ vọng của khách hàng.

Thường được thực hiện trong một môi trường giống thực tế và sử dụng các kịch bản thử nghiệm để kiểm tra các chức năng và tính năng của phần mềm.***[1]***

* + 1. **Mô hình kiểm thử được áp dụng trong đề tài**

Đề tài sử dụng mô hình kiểm thử chữ V (V - model) [3](I.E. Dorothy Graham, Founđatión of software, 2007).



Ảnh 5.7. Mô hình kiểm thử V-model

* **Các hoạt động phát triển nằm bên tay trái:**

-**Phân tích yêu cầu:** Phân tích để hiểu được mong đợi của khách hàng về sản phẩm bằng cách giao tiếp, trao đổi.

- **Yêu cầu hệ thống:** Xác định các yêu cầu và mong đợi của khách hàng đối với sản phẩm, hệ thống thiết kế chi tiết phải được phát triển để phát triển sản phẩm.

- **Thiết kế kiến trúc:** Dựa trên thiết kế mức cao, kiến trúc phần mềm được tạo ra. Các mô-đun, mối quan hệ và sự phụ thuộc của chúng, sơ đồ kiến trúc, bảng cơ sở dữ liệu, chi tiết về công nghệ đều được hoàn tất trong giai đoạn này.

- **Thiết kế chi tiết:** Trong giai đoạn này mỗi mô-đun hoặc các thành phần phần mềm đều được thiết kế riêng. Các method, class, giao diện, các kiểu dữ liệu,… đều được hoàn tất trong giai đoạn này.

- **Code:** Trong giai đoạn này, các lập trình viên thực hiện hoạt động code.

* Từ các hoạt động phát triển phần mềm trên, các tester áp dụng kiểm thử theo mô hình chữ V theo các giai đoạn:
* **Unit test (Kiểm thử đơn vị):** Kiểm thử từng phần riêng lẻ (tức là hàm, method) của chương trình để đảm bảo chúng hoạt động đúng.
* **Integration test (Kiểm thử tích hợp):** Kiểm thử sự tương tác giữa các phần khác nhau của chương trình để đảm bảo chúng hoạt động đúng khi kết hợp với nhau.
* **System test (Kiểm thử hệ thống):** Kiểm thử toàn bộ hệ thống, bao gồm cả các chức năng, yêu cầu phi chức năng và kiểm tra xem hệ thống có đáp ứng được yêu cầu của người dùng hay không.
* **UA test (Kiểm thử chấp nhận người dùng):** Được thực hiện bởi khách hàng hoặc người sử dụng cuối để đảm bảo rằng phần mềm đáp ứng đầy đủ các yêu cầu và kỳ vọng của khách hàng.
  + 1. **Kế hoạch kiểm thử**

**5.2.3.1. Tài nguyên đầu vào và đầu ra để lập plan**

|  | **Unit Test** | **Integration Test** | **System Test** | **Acceptance Test** |
| --- | --- | --- | --- | --- |
| Đầu vào | - Tài liệu thiết kế hệ thống (SDD).  - Tài liệu thiết kế chi tiết (DDD).  - Tài liệu về các giao diện của hệ thống (Interface Documents).  - Tài liệu về các yêu cầu chức năng và phi chức năng của hệ thống (Functional and Non-functional Requirements Documents).  - Test plan | - Các kịch bản thử nghiệm (Test Scenarios, Cases, and Scripts) và các bộ dữ liệu kiểm thử (Test Data).  - Các tài liệu thiết kế hệ thống (System Design Documents).  - Các tài liệu thiết kế chi tiết (Detailed Design Documents).  - Các tài liệu về các giao diện của hệ thống (Interface Documents).  - Test plan | - Tài liệu thiết kế hệ thống  - Tài liệu SRS (đặc tả yêu cầu chức năng, đặc tả yêu cầu phi chức năng):[Srs-test.docx](https://docs.google.com/document/d/1AYl1OYdy82PHeAGMMOfeqjq6BCTg0KC1/edit)  - Test plan  ( Recovery Testing  , Security Testing  Stress Testing,  Performance Testing): [Test\_Case\_NM\_CNPM.xlsx](https://docs.google.com/spreadsheets/d/198ZxGD6BG16FEyg4-jUbV0vGHq1mZR9t/edit#gid=108689351)  - Test data | - Các yêu cầu của khách hàng về tính năng, giao diện, đa nền tảng,...  - Tài liệu thiết kế hệ thống  - Test plan:  [Test\_Case\_NM\_CNPM.xlsx](https://docs.google.com/spreadsheets/d/198ZxGD6BG16FEyg4-jUbV0vGHq1mZR9t/edit#gid=108689351)  - Test data |
| Tiêu chuẩn chất lượng đầu vào | - Yêu cầu nghiệp vụ sẵn sàng  - Có sẵn các yêu cầu chức năng và đặc tả kỹ thuật.  - Tài liệu thiết kế cấp đã có.  - Giai đoạn lập kế hoạch đã hoàn thành. | - Test plan đã được chấp nhận  - Test case (Trường hợp kiểm thử)  - Test data (Dữ liệu thử nghiệm)  - Các module đã được kiểm thử đơn vị (Unit testing)  - Tất cả các lỗi quan trọng và có mức độ ưu tiên cao trước đó đều được fix xong và closed.  - Môi trường thử nghiệm được thiết lập. | -Integration Test đã được thực hiện thành công và đáp ứng các tiêu chí đầu ra của kiểm thử tích hợp.  -Tất cả các lỗi đã được sửa và đóng.  -Môi trường kiểm tra hệ thống đã được thiết lập.  -Test cases/scripts đã sẵn sàng. | - Đảm bảo các unit/module, function chính được hoạt động trơn tru và có performance tốt.  - Các khâu kiểm thử Unit test, integration test, system test đã được hoàn thành  -Không còn lỗi quan trọng tồn tại trong hệ thống  -Đảm bảo UI/UX trước UAT  -System Testing phải được hoàn thành, không có lỗi lớn để chắc chắn hệ thống đã sẵn sàng thực hiện UAT  - Môi trường production phải được chuẩn bị sẵn sàng, hoạt động đa nền tảng và đa thiết bị |
| Đầu ra | - Xác nhận chức năng unit  - Các lỗi còn tồn đọng trong từng unit  - Báo cáo thống kê kết quả test.  [Test\_Case\_NM\_CNPM.xlsx](https://docs.google.com/spreadsheets/d/198ZxGD6BG16FEyg4-jUbV0vGHq1mZR9t/edit#gid=1176202161) | - Xác định các lỗi khi tích hợp unit/ module  - Đảm bảo rằng toàn bộ ứng dụng hoạt động như thiết kế  - Kiểm tra tính tương thích của các unit/module  - Báo cáo thống kê kết quả test. | * Báo cáo về các lỗi, vấn đề.   [Test\_Case\_NM\_CNPM.xlsx](https://docs.google.com/spreadsheets/d/198ZxGD6BG16FEyg4-jUbV0vGHq1mZR9t/edit#gid=1964816818) | -Những vấn đề ở Unit hoặc Integration Test đã để lọt.  - Tìm hiểu và xác định được các yêu cầu của người dùng bằng cách kiểm chứng trực tiếp.  - Có cái nhìn tổng quan nhất về kết quả hệ thống đạt được.  - Báo cáo thống kê kết quả test. |
| Tiêu chuẩn chất lượng đầu ra | - Đảm bảo unit theo đúng tài liệu  - Chức năng của unit đảm bảo ổn định  - Performance riêng lẻ của từng unit ổn định | - Đảm bảo sự tương thích của từng unit/ module khi tích hợp  - Performance của các unit/module liên quan ổn định | - Đảm bảo các yêu cầu chức năng và phi chức năng của hệ thống phần mềm được đáp ứng đầy đủ, các lỗi và vấn đề đã được phát hiện và báo cáo một cách chi tiết và chính xác, và độ tin cậy của hệ thống đã được kiểm tra và đảm bảo.  - Tất cả test case System Testing đã được thực hiện.  - Lỗi ưu tiên đã được sửa và đóng.  - Hệ thống đang đáp ứng tất cả các yêu cầu nghiệp vụ và chức năng.  - Hệ thống tương thích trên đa nền tảng. | - Hoàn thành và phê duyệt bởi Test Management team.  - Quy trình nghiệp vụ thiết yếu đã được hoàn thành và đầy đủ tính đúng đắn  - Không có các lỗi nghiêm trọng ảnh hưởng lớn đến dự án |

**5.2.3.2. Phạm vi kiểm thử**

* **Các tính năng cần được kiểm thử:**
* Tạo tài khoản:
* Kiểm tra chức năng cho phép người dùng đăng ký tạo một tài khoản mới
* Kiểm tra chức năng đăng nhập vào tài khoản vừa tạo
* Tìm kiếm: Tên bài hát, thể loại, ca sĩ, người đăng, album
* Phát nhạc, tạo danh sách phát, quản lý nhạc: Chức năng mở một Bài hát / Playlist từ trang chủ hoặc từ trang tìm kiếm; Chức năng cho phép người dùng sao chép, chia sẻ Bài hát / Playlist
* **Các tính năng không được kiểm thử:** Tải bài hát lên, Thông tin về số lượt nghe bài hát

**5.2.3.3. Các mức độ kiểm thử**

|  | **Unit test** | **Integration test** | **System test** | **UA test** |
| --- | --- | --- | --- | --- |
| **Mục tiêu kiểm thử** | - Đảm bảo mỗi thành phần phần mềm trong hệ thống được kiểm tra đầy đủ và kịp thời để đảm bảo tính năng hoạt động của từng module.  - Đảm bảo hệ thống phần mềm đáp ứng được các yêu cầu chức năng được đề xuất, bao gồm các tính năng chia sẻ âm nhạc, quản lý tài khoản người dùng, tìm kiếm, gợi ý…  - Đảm bảo hệ thống phần mềm được kiểm tra và phát hiện các lỗi tiềm ẩn, các điểm yếu trong hệ thống  - Đảm bảo hệ thống được bảo vệ khỏi các mối đe dọa bảo mật | - Đảm bảo các giao diện giữa các module hoạt động đúng cách, đồng bộ.  - Đảm bảo các thành phần phần mềm trong hệ thống đồng bộ hoạt động với nhau, bao gồm các chức năng quản lý tài khoản người dùng, chia sẻ âm nhạc, tìm kiếm,...  - Đảm bảo hệ thống phần mềm đáp ứng được các yêu cầu hiệu suất khi tích hợp các thành phần với nhau, đảm bảo bảo mật.  - Đảm bảo hệ thống phần mềm được kiểm tra, phát hiện các lỗi tiềm ẩn, các điểm yếu trong quá trình tích hợp. | - Đảm bảo toàn bộ hệ thống phần mềm được kiểm tra và phát hiện các lỗi tiềm ẩn, lỗi chức năng và các điểm yếu có thể ảnh hưởng đến tính đầy đủ của hệ thống.  - Đảm bảo hệ thống phần mềm đáp ứng được các yêu cầu hiệu suất, bảo mật.  - Đảm bảo hệ thống phần mềm được kiểm tra và phát hiện các lỗi tiềm ẩn và các điểm yếu trong quá trình hoạt động để tăng tính tin cậy của hệ thống, bao gồm việc phục hồi dữ liệu và khôi phục hệ thống khi xảy ra sự cố. | - Thiết kế tài liệu kiểm thử một cách có hệ thống thỏa mãn yêu cầu của khách hàng và đảm bảo sản phẩm phù hợp, được khách hàng chấp nhận.  - Tìm ra các lỗi không mong muốn có thể xảy ra, đảm bảo các chức năng đăng ký, đăng nhập, các chức năng của trang chủ và quản lý nhạc hoạt động như mong đợi của người dùng.  - Xác nhận giao diện UI của trang web có hoạt động như mong đợi, đúng yêu cầu của khách hàng và được khách hàng chấp nhận. |
| **Tiêu chí kiểm thử** | **- Tiêu chí đình chỉ:**  + Bên dev chưa hoàn thành hết các module cần thiết  **- Tiêu chí thoát:**  **+** Hoàn thành phạm vi test  + Tất cả các lỗi được xác định phải được ghi nhận vào 1 giải pháp đã thỏa thuận. Không còn lỗi nào ở mức độ Medium trở lên. | **- Tiêu chí đình chỉ:**  **+** Phát hiện ra các lỗi quan trọng còn tồn tại, chưa được khắc phục  **- Tiêu chí thoát:**  + Hoàn thành phạm vi test  + Không còn lỗi nghiêm trọng  + Đảm bảo được độ phủ kiểm thử của hệ thống là cao nhất có thể ( từ 70% trở lên).  + Tất cả các thành phần của hệ thống tương thích và tích hợp tốt với nhau | **- Tiêu chí đình chỉ:**  **+** Phát hiện ra các lỗi quan trọng còn tồn tại, chưa được khắc phục  **- Tiêu chí thoát:**  **+** Tất cả các chức năng và tính năng của hệ thống đã được kiểm tra và đáp ứng đầy đủ các yêu cầu chức năng và phi chức năng.  + Không còn lỗi nghiêm trọng  + Đảm bảo được độ phủ kiểm thử của hệ thống là cao nhất có thể ( từ 85% trở lên). | **- Tiêu chí đình chỉ:** Phát hiện ra các lỗi quan trọng trong hệ thống còn tồn tại, chưa được khắc phục tại 3 khâu kiểm thử trướcđó  **Tiêu chí thoát:**  **+** Tỉ lệ giữa số trường hợp đã kiểm thử trên tổng số trường hợp kiểm thử trên kế hoạch phải đạt 100%  + Tỉ lệ giữa các trường hợp kiểm thử không đạt trên số trường hợp được thực hiện kiểm thử phải dưới 20% |
| **Cách kiểm thử** | Kiểm thử thủ công, tuần tự theo các bước được định nghĩa trong testcase | Kiểm thử thủ công, tuần tự theo các bước được định nghĩa trong testcase | Kiểm thử thủ công, tuần tự theo các bước được định nghĩa trong testcase | Kiểm thử thủ công, tuần tự theo các bước được định nghĩa trong testcase |

5.2.3.4. Lịch trình công việc

| **Các mốc quan trọng** | **Tài liệu liên quan** | **Thời gian hoàn thành dự kiến** | **Thời gian hoàn thành thực tế** |
| --- | --- | --- | --- |
| **Lập kế hoạch kiểm thử** | Tài liệu Test plan | 16/03/2023 - 18/03/2023 | 20/03/2023 - 22/03/2023 |
| **Xem xét kế hoạch kiểm thử** | Tài liệu Test plan | 19/03/2023 - 23/03/2023 | 22/03/2023 - 25/03/2023 |
| **Thiết kế các test case** | Tài liệu Test case | 26/03/2023 - 01/04/2023 | 28/03/2023 - 02/04/2023 |
| **Thực thi các test case** | Tài liệu Test case | 02/04/2023 - 09/04/2023 | 13/04/2023 - 20/04/2023 |
| **Ghi nhận và đánh giá kết quả kiểm thử** | Tài liệu Test case | 10/04/2023 - 12/04/2023 | 21/04/2023 - 22/04/2023 |

5.2.3.5. Phân công công việc

| **Thành viên** | **Vai trò** |
| --- | --- |
| **Lê Thùy Trang** | - Design/Test: Lập kế hoạch kiểm thử, quản lý tiến độ hoạt động kiểm thử, thiết kế các testcase bổ sung và thực thi các testcase bổ sung  - Ghi nhận và đánh giá kết quả kiểm thử |
| **Nguyễn Huy Hùng** | Thiết kế và viết các testcase, thực thi các testcase ở mức độ kiểm thử đơn vị và kiểm thử chấp nhận người dùng |
| **Trần Văn Quang** | Thiết kế và viết các testcase, thực thi các testcase ở mức độ kiểm thử hệ thống |
| **Trịnh Quang Toàn** | - Thiết kế và viết các testcase, thực thi các testcase ở mức độ kiểm thử tích hợp và kiểm thử chấp nhận người dùng  - Ghi nhận và đánh giá kết quả kiểm thử |

5.2.3.6. Môi trường thực hiện kiểm thử

* **Công cụ thực hiện kiểm thử:**
* Microsoft Office Excel phiên bản 2016: Quản lý Testcase, Tạo báo cáo Test
* Visual Studio Code + XAMPP (MySQL & Apache): Thực hiện test
* **Phần cứng:**
* Máy tính cá nhân có kết nối Internet
* Số lượng: 4 cái
* **Phần mềm:**

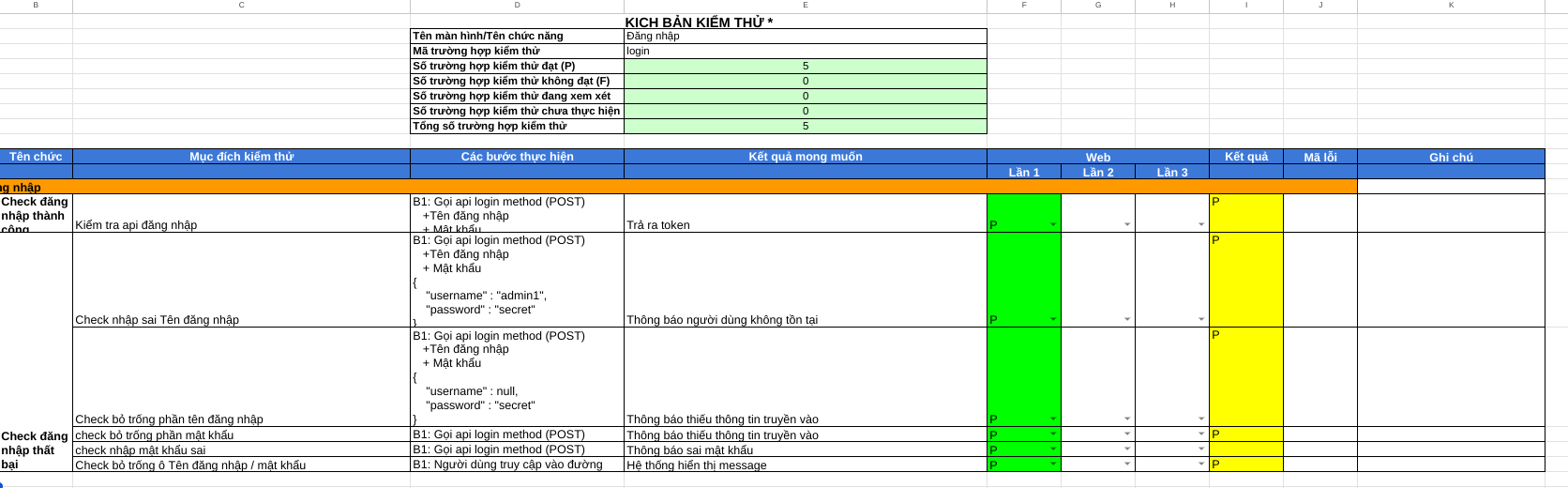
| **Tên phần mềm** | **Phiên bản** | **Loại** |
| --- | --- | --- |
| **Microsoft Windows 10** | 10 | Hệ điều hành |
| **XAMPP:**  **MySQL**  **Apache** | 8.0.28 | Hệ quản trị cơ sở dữ liệu quan hệ |
| **Google Chrome** | 19.0.1084.56 | Trình duyệt web |

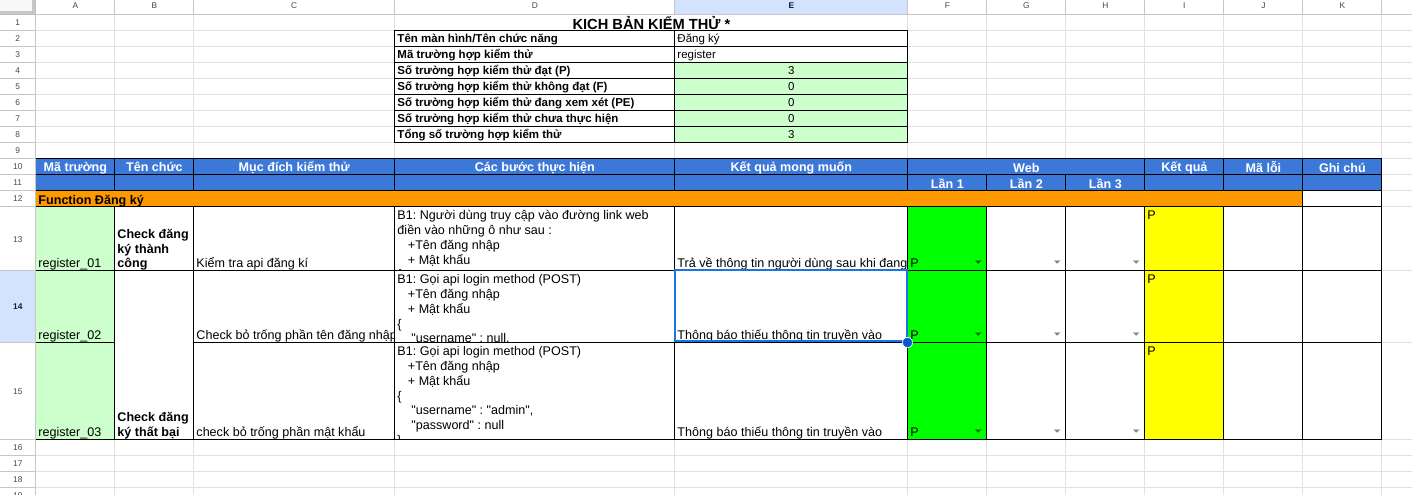
5.2.3.7. Rủi ro

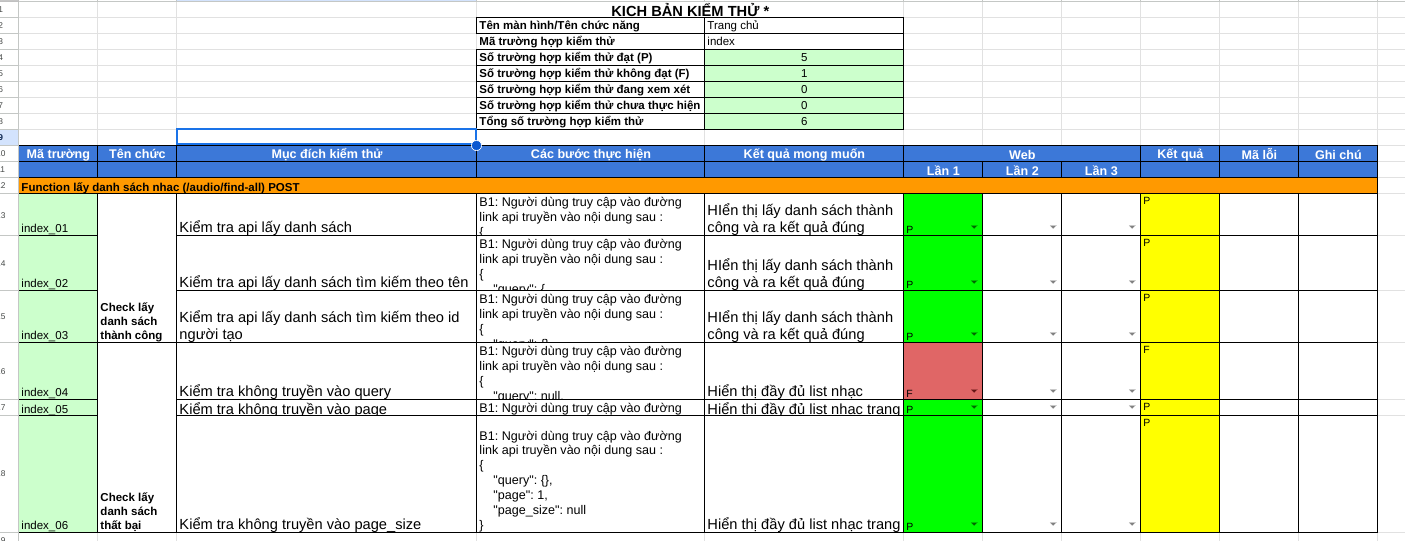
| **Rủi ro** | **Giải pháp giảm, tránh rủi ro** |
| --- | --- |
| Không đủ nhân lực cho kiểm thử | Tăng thời gian thực hiện kiểm thử |
| Các thành viên trong nhóm thiếu kỹ năng về kiểm thử | Tìm hiểu thêm các tài liệu chính thống hướng dẫn kiểm thử, hỗ trợ và thảo luận trong quá trình kiểm thử |
| Thành viên trong nhóm nghỉ hoặc có việc đột xuất | Phân công lại nhiệm vụ công việc cho các thành viên còn lại để đảm bảo tiến độ dự án |
| Code chậm so với dự án, phần mềm còn nhiều lỗi hoạt động | Thông báo các lỗi cho bên lập trình để khắc phục kịp thời trước khi đến tay khách hàng |

* + 1. Kết quả kiểm thử

**5.2.4.1. Kết quả kiểm thử mức đơn vị**

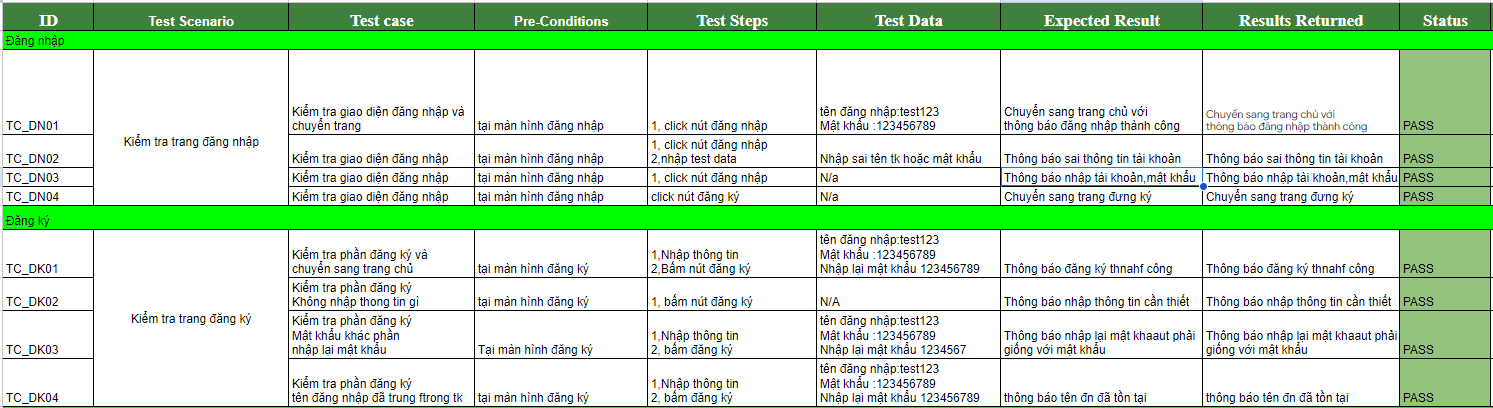
****

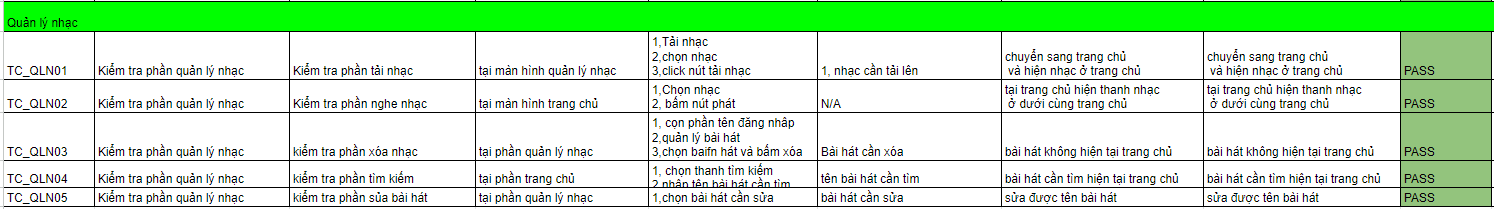
****

****

Ảnh 5.8. Kết quả mức kiểm thử đơn vị

**5.2.4.2. Kết quả kiểm thử mức tích hợp**

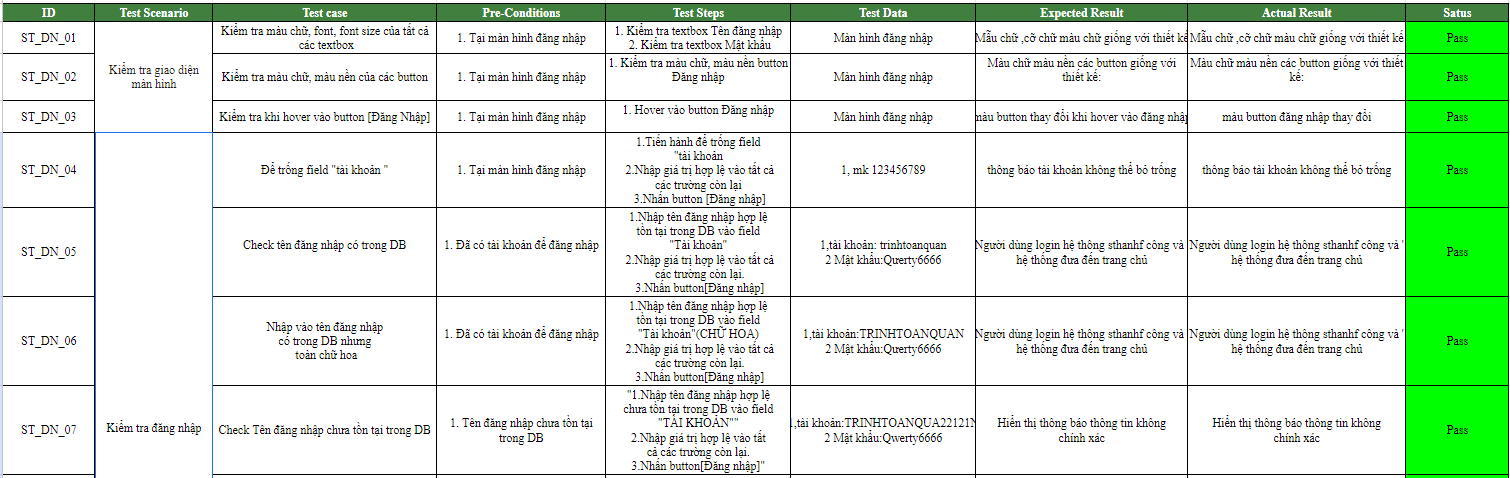
****

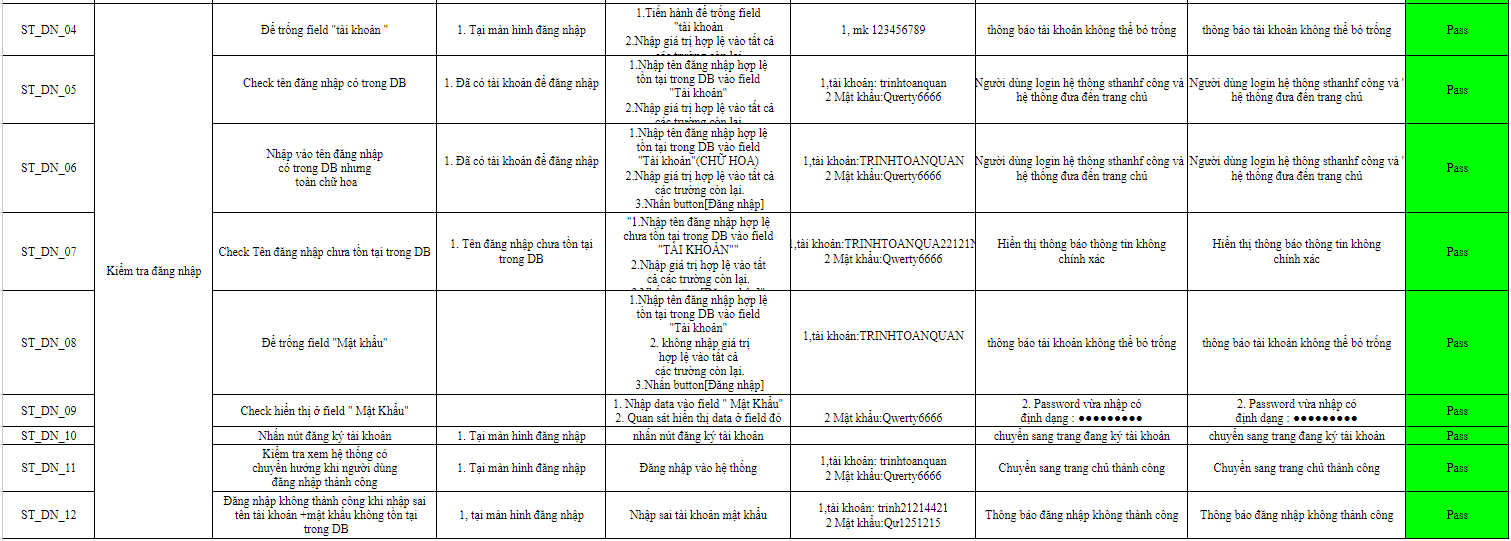
****

Ảnh 5.9. Kết quả mức kiểm thử tích hợp

**5.2.4.3. Kết quả kiểm thử mức hệ thống**

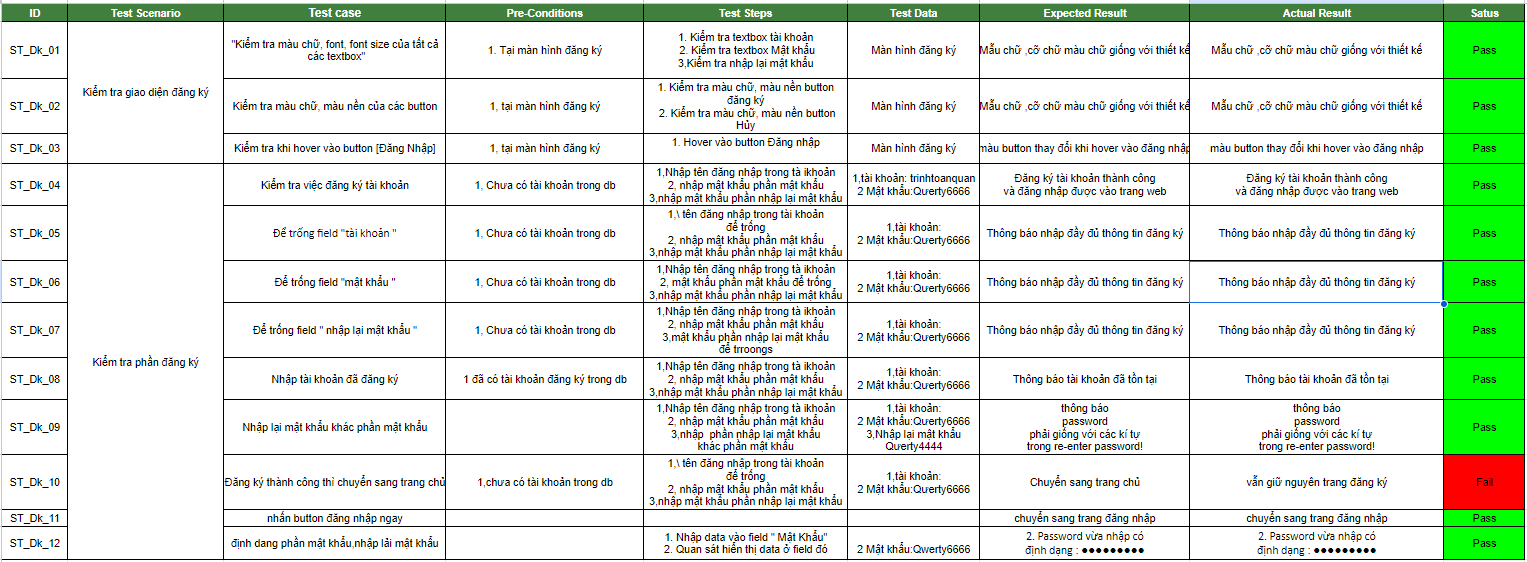
* **Chức năng đăng nhập:**

****

****

Ảnh 5.10. Kết quả mức kiểm thử hệ thống - đăng nhập

* **Chức năng đăng ký:**

****

Ảnh 5.11. Kết quả mức kiểm thử hệ thống - đăng ký

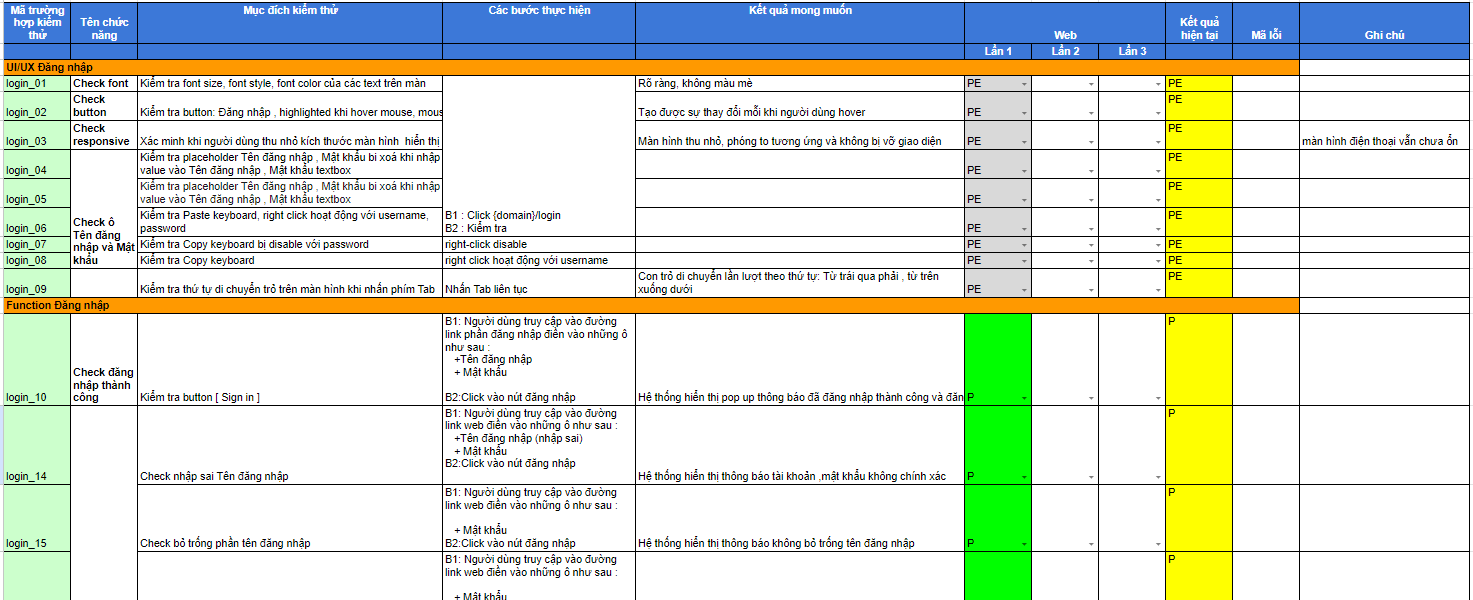
* **Chức năng quản lý nhạc:**

****

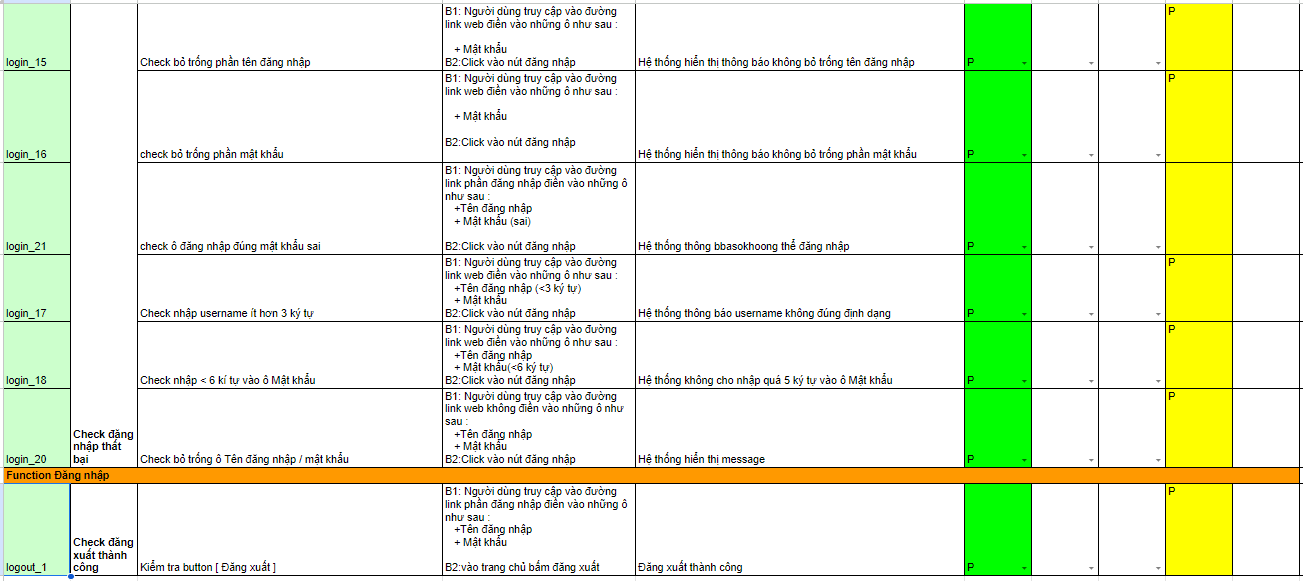
Ảnh 5.11. Kết quả mức kiểm thử hệ thống - quản lý nhạc

**5.2.4.4. Kết quả kiểm thử mức kiểm thử chấp nhận người dùng**

* **Chức năng đăng nhập:**

****

Ảnh 5.12. Kết quả mức kiểm thử chấp nhận - đăng nhập

****

* **Chức năng đăng ký:**

**CHƯƠNG 6. KẾT LUẬN**

1. **Đánh giá chất lượng phần mềm**

Đánh giá chất lượng phần mềm Mạng xã hội chia sẻ âm nhạc SoundFriends cho trung tâm giải trí MTP Entertainment:

* **Theo các yếu tố chất lượng:**
* Tính đúng đắn: Phần mềm không gây ra các lỗi lớn và sai sót trong quá trình hoạt động, các lỗi đã được phát hiện trong quá trình kiểm thử đã được khắc phục trước khi đến tay khách hàng.
* Tính tin cậy: Phần mềm không phát sinh các lỗi và hoạt động ổn định trong 1 thời gian dài tiến hành dùng thử.
* Tính khả dụng: Phần mềm mạng xã hội chia sẻ âm nhạc SoundFriends được tạo ra có giao diện, hình ảnh, màu sắc hài hòa bắt mắt, thân thiện với người dùng, có khả năng thích ứng với các thiết bị và hệ điều hành khác nhau.
* Tính toàn vẹn: Phần mềm có các tính năng bảo mật ở mức cơ bản cho thông tin và dữ liệu của người dùng
* Tính hiệu quả: Phần mềm chưa được chạy thử và hoạt động trong 1 môi trường thực tế, chưa đảm bảo được khả năng có thể xảy ra lỗi hoặc sự cố hay không khi hoạt động trên môi trường thực tế trong 1 khoảng thời gian dài.
* Tính liên tác: Các thành phần của phần mềm được phân tách thành các module độc lập nhau: Quản lý người dùng (Đăng ký, đăng nhập), Quản lý bài hát, Tìm kiếm, Phát nhạc, Chia sẻ bài hát.
* Tính khả chuyển: Phần mềm hoạt động đa nền tảng, đa thiết bị
* Tính tái sử dụng: Có thể sử dụng mã nguồn của phần mềm mạng xã hội chia sẻ âm nhạc SoundFriends cho trung tâm giải trí MTP Entertainment để phát triển các dự án mạng xã hội chia sẻ âm nhạc và dự án phần mềm nghe nhạc đơn giản khác sau này.
* Tính bảo trì: Phần mềm có khả năng được bảo trì và nâng cấp 1 cách dễ dàng mà không gây ảnh hưởng đến các thành phần khác trong hệ thống nhờ lựa chọn mô hình kiến trúc MVC.
* Tính linh hoạt: Phần mềm có thể thích ứng với các môi trường khác nhau, hoạt động được trên nhiều hệ điều hành và các thiết bị khác nhau; phần mềm đa người dùng theo đúng yêu cầu bài toán.
* Tính kiểm thử được: Phần mềm có thể được kiểm thử 1 cách dễ dàng nhờ các thành phần được phân tách thành các module độc lập và các yêu cầu rõ ràng từ khách hàng về phần mềm.
* **Theo các pha SDLC:**
* Phân tích yêu cầu: Đã thu thập được các thông tin sơ khai về phần mềm mạng xã hội chia sẻ âm nhạc SoundFriends, phân tích đúng yêu cầu của 1 phần mềm mạng xã hội chia sẻ âm nhạc và đúng theo yêu cầu của người dùng.
* Thiết kế: Hệ thống có giao diện đơn giản, dễ nhìn, quen thuộc thuận tiện cho người dùng dễ sử dụng
* Lập trình: Các sinh viên thực hiện hoạt động phát triển phần mềm đã tiến hành lập trình đầy đủ các chức năng cơ bản cần có cho 1 phần mềm mạng xã hội chia sẻ âm nhạc: đăng ký, đăng nhập, quản lý bài hát, nghe nhạc, chia sẻ âm nhạc.
* Kiểm thử: Đã phát hiện và loại bỏ được hết các lỗi nghiêm trọng, độ bao phủ của các test case cao, chưa tiến hành kiểm thử hết được tất cả các chức năng của phần mềm
* Quá trình phát triển tiếp tục và cải tiến sản phẩm: Các tester liên tục thực hiện lại các kiểm thử đã được thực hiện trước đó để khắc phục các lỗi phát sinh cũng như phát triển thêm các tính năng mới cho hệ thống mạng xã hội chia sẻ âm nhạc.

1. **Ưu điểm & Nhược điểm của đề tài**

* Ưu điểm
* Tính chuyên nghiệp: Đề tài này đang được phát triển cho một trung tâm giải trí nổi tiếng, do đó đòi hỏi tính chuyên nghiệp cao trong việc thiết kế và triển khai phần mềm, từ đó đáp ứng được yêu cầu của khách hàng.
* Tính độc đáo: Mạng xã hội chia sẻ âm nhạc SoundFriends là một đề tài độc đáo, với mục đích cụ thể là chia sẻ âm nhạc trong ngành giải trí, tạo điều kiện cho người dùng có thể giao lưu, kết nối và chia sẻ đam mê âm nhạc.
* Tính tiện ích: Phần mềm SoundFriends có tiềm năng tạo ra một cộng đồng trực tuyến về âm nhạc, đồng thời cung cấp cho người dùng nhiều tính năng tiện ích như tìm kiếm, nghe và chia sẻ âm nhạc, theo dõi hoạt động của các nghệ sĩ, tương tác với bạn bè, vv.
* Tính khả chuyển: Mã nguồn của phần mềm được thiết kế theo mô hình kiến trúc MVC, từ đó giúp đơn giản hóa quá trình bảo trì, nâng cấp và mở rộng tính năng trong tương lai.
* Nhược điểm
* Chưa được kiểm thử thực tế: Phần mềm SoundFriends chưa được chạy thử và hoạt động trong môi trường thực tế trong một khoảng thời gian dài, do đó chưa đủ dữ liệu để đánh giá độ hiệu quả và đáp ứng của phần mềm trong thực tế.
* Đòi hỏi tính bảo mật cao: Phần mềm Mạng xã hội chia sẻ âm nhạc SoundFriends chứa nhiều dữ liệu người dùng, bao gồm thông tin cá nhân và âm nhạc, do đó đòi hỏi tính bảo mật cao để đảm bảo an toàn thông tin của người dùng.
* Trong thị trường mạng xã hội chia sẻ âm nhạc khá cạnh tranh với sự hiện diện của nhiều nền tảng âm nhạc lớn như Spotify, SoundCloud, YouTube, vv. Do đó, SoundFriends cần phải đối mặt với sự cạnh tranh từ những đối thủ lớn này trong việc thu hút và giữ chân người dùng.
* Quản lý nội dung: Mạng xã hội chia sẻ âm nhạc cần phải đảm bảo tính hợp pháp của âm nhạc được chia sẻ trên nền tảng, từ việc kiểm duyệt nội dung, phát hiện và xử lý các vi phạm bản quyền, đến việc đảm bảo tính an toàn, không vi phạm đạo đức và luật pháp trong việc sử dụng và chia sẻ âm nhạc trên nền tảng.
* Tính hấp dẫn của nội dung: Sự thành công của mạng xã hội chia sẻ âm nhạc SoundFriends sẽ phụ thuộc vào tính hấp dẫn của nội dung âm nhạc được chia sẻ trên nền tảng. Nếu không có đủ nội dung hấp dẫn và thu hút người dùng, SoundFriends có thể gặp khó khăn trong việc thu hút và giữ chân người dùng.
* Chi phí và nguồn lực: Phát triển và duy trì một mạng xã hội chia sẻ âm nhạc đòi hỏi đầu tư về công nghệ, nhân lực, hệ thống máy chủ, bảo mật, và quản lý dữ liệu. Do đó, SoundFriends cần có nguồn lực đủ lớn để đáp ứng các yêu cầu này, đồng thời đối mặt với rủi ro tài chính và quản lý nguồn lực trong quá trình phát triển và hoạt động.

1. **Hướng phát triển**

* *Xét theo các yếu tố chất lượng.*
* Tốc độ xử lý: Phần mềm chia sẻ nhạc cần có tốc độ xử lý nhanh để giúp người dùng tìm kiếm và phát nhạc một cách thuận tiện và nhanh chóng. Điều này có thể đạt được thông qua việc tối ưu hóa mã nguồn và sử dụng các công nghệ hiện đại để cải thiện hiệu suất của phần mềm.
* Tối ưu hóa tốc độ tìm kiếm: giúp cho phần mềm chia sẻ nhạc được tối ưu hóa để hiển thị cao trên các trang kết quả tìm kiếm của các công cụ tìm kiếm, giúp tăng lượng truy cập trang web, tăng doanh số và nâng cao thương hiệu.
* *Xét theo các pha SDLC*
* Nên lập kế hoạch và triển khai: Sau khi có thiết kế phần mềm, kế hoạch triển khai phần mềm cũng cần được đưa ra để đảm bảo rằng dự án được triển khai một cách hiệu quả.

**TÀI LIỆU THAM KHẢO**

[1] 7th edition - Software Engineering- A Practitioner's Approach By Roger S.Pressman

[2] Software Testing and Quality Assurance - Theory and Practice - Kshirasagar Naik - Priyadarshi Tripathy

[3] Foundation of software By I.E. Dorothy Graham, 2007

**PHỤ LỤC A: QUY TẮC ĐẶT TÊN**

1. Quy tắc đặt tên biến:

* Mọi cái tên đều phải bằng tiếng anh đúng chính tả và có nghĩa.
* Tên bao gồm: tên class, tên biến, tên hằng số, tên hàm, tên file.
* Tên phải gắn liền với đặc điểm, chức năng của đối tượng tương ứng.
* Tên hàm biến dùng trong database phải được biểu diễn ở dạng snake\_case

1. Quy tắc đặt mã hiệu test:

**PHỤ LỤC B: CÁC BIỂU MẪU, TÀI LIỆU THU THẬP ĐƯỢC**

**PHỤ LỤC C: CÁC BIÊN BẢN TIẾN ĐỘ**

**PHỤ LỤC D: CÁC BIÊN BẢN RÀ SOÁT KỸ THUẬT**

**PHỤ LỤC E: HỒ SƠ TOÀN BỘ DỰ ÁN**

1. Hồ sơ khảo sát

<https://docs.google.com/spreadsheets/d/1bKEpBqUM2Bt8lL-R4Yy5TM0yxOf1Cj0NsuigXNJ3_a0/edit#gid=1475661192>

1. Hồ sơ phân tích

https://drive.google.com/drive/folders/1\_iQo03M51SS\_nVsT1DH47xJmdOvAytx-?usp=sharing

1. Hồ sơ thiết kế

https://drive.google.com/drive/folders/1QGUe8bd1DgyCQC-JqhLaZrqoaHRkACJi?usp=sharing

1. Mã nguồn

<https://drive.google.com/drive/folders/1TIOXau1QImNB81hwSz9Xc_LBE86fiVjd?usp=sharing>

<https://gitlab.com/quangtrn8821/sound-friend-fe>

<https://gitlab.com/quangtrn8821/sound-friend>

1. Hồ sơ kiểm thử

<https://docs.google.com/spreadsheets/d/198ZxGD6BG16FEyg4-jUbV0vGHq1mZR9t/edit#gid=1964816818>