**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**⸎⸎⸎⸎⸎**

A logo of a university

AI-generated content may be incorrect.

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN KĨ THUẬT PHẦN MỀM**

**Đề tài : Xây dựng website thương mại sheet nhạc**

**cho game Sky: Children of the Light**

Thành viên : Lê Ngọc Khánh – 23010546

Nguyễn Khắc Long – 23010418

Nguyễn Anh Tài – 23010584

Giáo viên hướng dẫn :

Lớp tín chỉ :

**HÀ NỘI, THÁNG 10/2025**

**BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Mã SV** | **Họ tên** | **Công việc** | **Tỉ lệ đóng góp** |
| 1 | 23010546 | Lê Ngọc Khánh | * Lên kế hoạch, phân công công việc, kiểm tra tiến độ, tổng hợp kết quả. * Thiết kế giao diện nhóm chức năng… * Thiết kế backend nhóm chức năng … * Làm slide thuyết trình … * Viết báo cáo chương …. * Thuyết trình nội dung ….. |  |
| 2 | 23010584 | Nguyễn Anh Tài |  |  |
| 3 | 23010418 | Nguyễn Khắc Long |  |  |

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

..............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

# Lời mở đầu

Trong bối cảnh bùng nổ của kỷ nguyên số, âm nhạc đã vượt ra khỏi ranh giới đời thực để trở thành một yếu tố không thể thiếu trong không gian ảo, đặc biệt là các trò chơi điện tử mang đậm tính nghệ thuật như Sky: Children of the Light. Tựa game này không chỉ là một thế giới để khám phá mà còn là một sân khấu để người chơi thể hiện cảm xúc, giao tiếp và kết nối thông qua âm nhạc. Một trong những điểm đặc sắc nhất của game là hệ thống nhạc cụ ảo, cho phép người chơi trình diễn những bản nhạc độc đáo, từ đó hình thành một cộng đồng sáng tạo và chia sẻ sheet nhạc vô cùng sôi nổi.

Nhận thấy nhu cầu thực tế từ cộng đồng, nhóm chúng em đã quyết định thực hiện đề tài "Xây dựng website thương mại sheet nhạc cho game Sky: Children of the Light". Mục tiêu của dự án là phát triển một nền tảng chuyên biệt cho việc chia sẻ và mua bán các bản nhạc được chuyển soạn riêng cho hệ thống nhạc cụ trong game. Điểm khác biệt của sheet nhạc Sky so với sheet nhạc truyền thống nằm ở định dạng đặc thù, được thiết kế để tương thích với giao diện và cơ chế chơi nhạc trong game. Điều này đặt ra những thách thức riêng về cách trình bày, lưu trữ và phân phối.

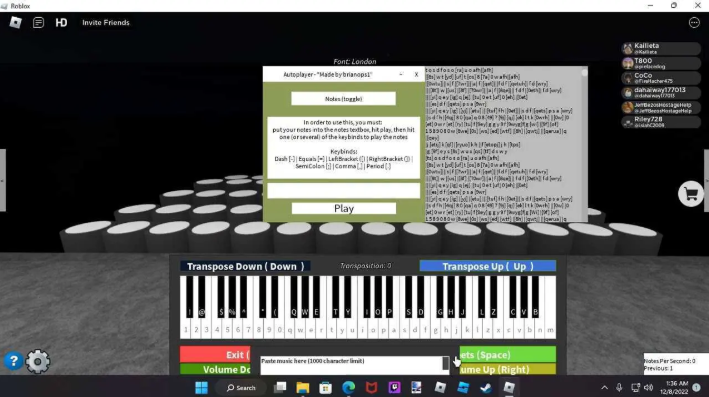
Trong khuôn khổ môn học Phân tích Thiết kế Phần mềm, dự án này là cơ hội để nhóm áp dụng các kiến thức đã học vào một quy trình phát triển hoàn chỉnh, từ khâu phân tích yêu cầu, thiết kế hệ thống, đến triển khai và kiểm thử sản phẩm. Báo cáo sẽ trình bày chi tiết toàn bộ quá trình thực hiện, những thách thức kỹ thuật đã đối mặt, giải pháp của nhóm, và định hướng phát triển trong tương lai. Chúng em kỳ vọng sản phẩm không chỉ giải quyết được bài toán thực tế mà còn góp phần thúc đẩy một cộng đồng sáng tạo bền vững trong thế giới Sky.

**Chương 1: Giới thiệu và Lập kế hoạch**

**1.1. Giới thiệu đề tài**

Trong thế giới game hiện đại, âm nhạc không chỉ còn là yếu tố phụ trợ mà đã trở thành một phần cốt lõi của trải nghiệm, một phương tiện để người chơi tương tác và thể hiện bản thân. Nhiều tựa game đã tích hợp cơ chế chơi nhạc cụ ảo hoặc các yếu tố nhịp điệu, cho phép người chơi biến không gian game thành sân khấu của riêng mình. Có thể kể đến các ví dụ như:

* Roblox: Là một nền tảng khổng lồ với vô số trò chơi do người dùng tạo ra, trong đó có nhiều game chuyên về âm nhạc như các trình giả lập piano ảo hoặc game nhịp điệu như "Funky Friday".



Hình 1.1

* Mini World: Một tựa game sandbox sáng tạo, cung cấp các công cụ để người chơi xây dựng thế giới riêng, bao gồm cả các khối nốt nhạc đơn giản để tạo ra giai điệu.

A screenshot of a video game

AI-generated content may be incorrect.

Hình 1.2

* Genshin Impact: Người chơi có thể sử dụng các nhạc cụ như "Phong Cổ Cầm" để trình diễn các bản nhạc. Cộng đồng game này cũng có những công cụ và website riêng để chia sẻ các bản nhạc được soạn cho game.



Hình 1.3

Mặc dù các tựa game trên đều có những yếu tố âm nhạc và nhịp điệu thú vị, nhóm quyết định chọn **Sky: Children of the Light** làm trọng tâm cho dự án vì những lý do khác biệt và tiềm năng sau:

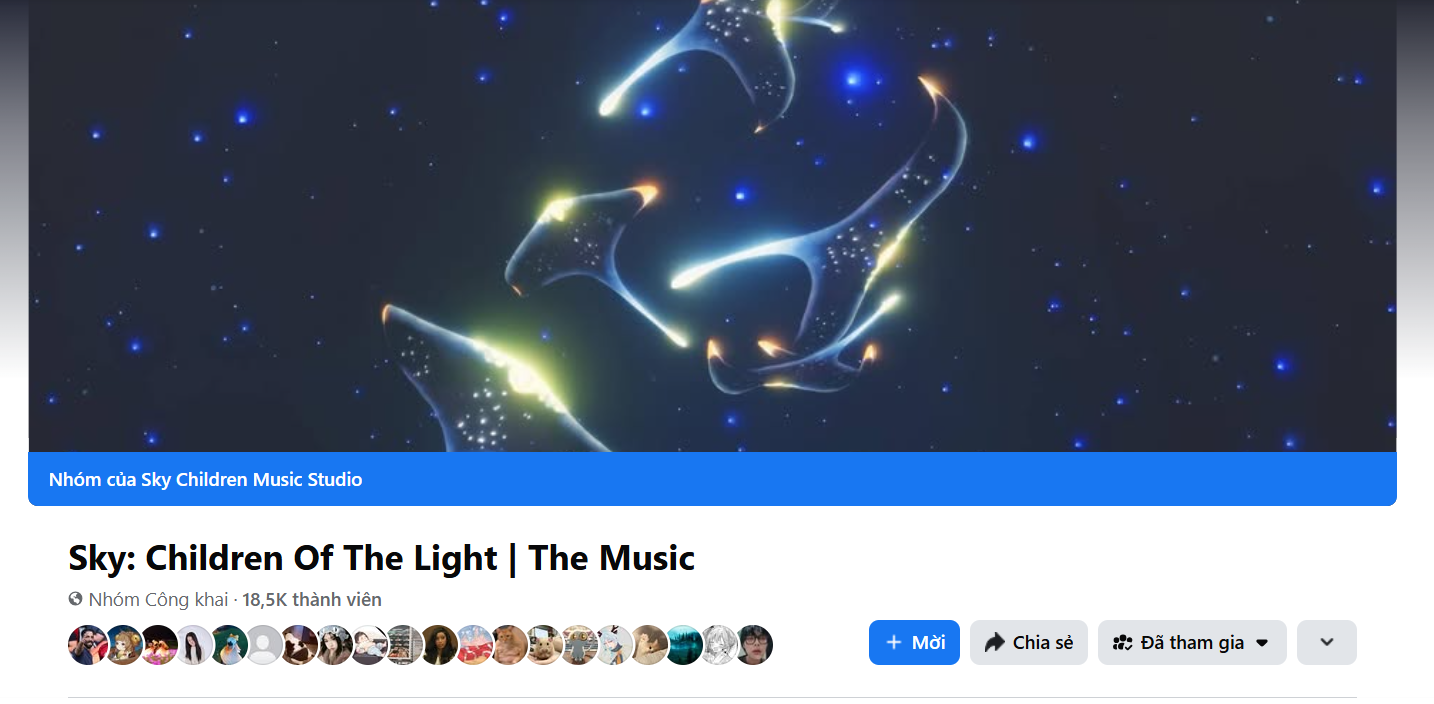
1. Âm nhạc là ngôn ngữ giao tiếp cốt lõi: Sky: Children of the Light không phải là game đầu tiên có chức năng chơi đàn ảo. Nhưng, Sky đã rất thành công trong việc đưa tính năng này trở thành một phương tiện giao tiếp và kết nối cộng đồng một cách trực quan, dễ tiếp cận và đầy cảm xúc. Không giống các game khác nơi âm nhạc là một tính năng phụ, trong Sky, âm nhạc là phương tiện giao tiếp chính. Người chơi dùng âm nhạc để kết bạn, giải đố và thể hiện cảm xúc. Điều này tạo ra một nhu cầu tự nhiên và mạnh mẽ hơn về việc học và chia sẻ các bản nhạc.

A screenshot of a video game

AI-generated content may be incorrect.

Hình 1.4

1. Cộng đồng sáng tạo khổng lồ và gắn kết: Sky sở hữu một trong những cộng đồng người chơi sáng tạo và thân thiện nhất. Họ không chỉ chơi game mà còn liên tục sáng tác, chuyển soạn và chia sẻ hàng ngàn bản sheet nhạc trên các nền tảng mạng xã hội. Tuy nhiên, việc chia sẻ này hiện còn manh mún, thiếu một nền tảng tập trung, có tổ chức để tìm kiếm, lưu trữ và thương mại hóa các tác phẩm.



Hình 1.5

1. Hệ thống nhạc cụ độc đáo và dễ tiếp cận: Hệ thống nhạc cụ của Sky được thiết kế với giao diện đơn giản hóa, trực quan, có nhiều loại nhạc cụ đa dạng và không đòi hỏi kiến thức nhạc lý phức tạp. Điều này giúp một lượng lớn người chơi, kể cả những người không chuyên, có thể dễ dàng tham gia vào việc chơi nhạc và sáng tạo. Chính sự đơn giản này đã tạo ra một "chuẩn" ký hiệu sheet nhạc riêng cho game, không giống với bất kỳ hệ thống nào khác.

A screenshot of a cell phone

AI-generated content may be incorrect.

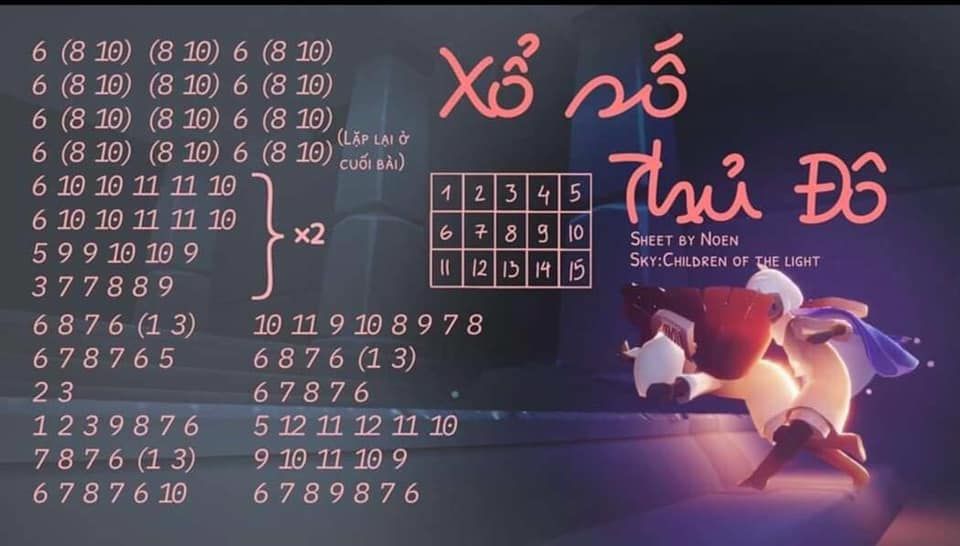
Hình 1.6

Từ việc nhận thấy khoảng trống và nhu cầu thực tế đó, đề tài **"Xây dựng website thương mại sheet nhạc cho game Sky: Children of the Light"** được hình thành. Dự án không chỉ nhằm mục đích áp dụng kiến thức kỹ thuật phần mềm mà còn mong muốn giải quyết một bài toán có thật, tạo ra một sản phẩm ý nghĩa và đóng góp cho sự phát triển bền vững của một cộng đồng game đầy sáng tạo.

**1.2. Phân tích bài toán & Giải pháp đề xuất**

Để làm rõ hơn tính cấp thiết của dự án, trước hết cần phải phân biệt rõ sự khác biệt nền tảng giữa sheet nhạc thông thường và sheet nhạc được sáng tạo riêng cho game. Sheet nhạc truyền thống sử dụng hệ thống ký âm nhạc lý chuẩn hóa toàn cầu (nốt, khuông nhạc, khóa son/fa), đòi hỏi người đọc phải có kiến thức nhạc lý nhất định. Ngược lại, sheet nhạc trong game như Sky được thiết kế để bất kỳ ai cũng có thể chơi được, sử dụng một hệ thống ký hiệu riêng biệt, trực quan và tương ứng trực tiếp với giao diện của nhạc cụ trong game. Trong bối cảnh đó, có hai định dạng chính để chia sẻ các bản nhạc:

Đầu tiên là **dạng Sheet Số**. Đây là một định dạng trực quan và dễ tiếp cận, thường được dùng trong các video hướng dẫn hoặc khi người chơi muốn chia sẻ nhanh một đoạn nhạc. Các nốt nhạc được thể hiện bằng một chuỗi số từ 1 đến 15 (hoặc ký tự đơn giản khác), tương ứng với vị trí 15 phím đàn trên giao diện 3x5 trong game.



Hình 1.7

Tuy nhiên, nhược điểm lớn nhất của dạng này là không thể hiện rõ trường độ hay nhịp điệu, khiến người chơi phải tự cảm nhận dựa trên giai điệu gốc. Vì vậy, một định dạng khác đã ra đời và trở thành phương pháp phổ biến và mạnh mẽ nhất trong cộng đồng người chơi sử dụng ứng dụng soạn nhạc, đó là **dạng Sheet Dữ liệu**.

A close up of a screen

AI-generated content may be incorrect.

Hình 1.8

Đây là một file dữ liệu có cấu trúc (tương tự JSON) không dành cho người đọc trực tiếp, mà để máy tính và các công cụ xử lý. Nó có cấu trúc chứa thông tin chi tiết như tên bản nhạc, tác giả, nhịp độ (bpm). Quan trọng nhất, mỗi nốt nhạc được ghi lại với hai thuộc tính: "key" (phím đàn) và "time" (thời điểm chơi chính xác tính bằng mili giây). Nhờ đó, nó ghi lại hoàn hảo nhịp điệu của cả bản nhạc, khắc phục hoàn toàn nhược điểm của sheet số.

Sự tồn tại song song của hai định dạng này cho thấy một nhu cầu đa dạng trong cộng đồng: một bên cần sự đơn giản, trực quan để tự luyện tập, một bên cần sự chính xác tuyệt đối để chia sẻ các bản nhạc phức tạp hoặc sử dụng cho các công cụ auto-play. Nhận thấy tiềm năng và tính ưu việt của sheet dạng dữ liệu, dự án sẽ tập trung hoàn toàn vào việc xây dựng một nền tảng chuyên biệt cho định dạng này. Nền tảng của quyết định này đến từ việc sheet dữ liệu là một "nguồn" gốc chứa đựng mọi thông tin cần thiết. Từ đó, việc tự động chuyển đổi sang dạng sheet số trực quan để người chơi dễ dàng theo dõi là hoàn toàn khả thi và có thể được tích hợp như một tính năng của website. Cách tiếp cận này đảm bảo tính chính xác và đồng bộ của dữ liệu, đồng thời tạo ra một nền tảng vững chắc, cho phép kết nối những nhà soạn nhạc tài năng với cộng đồng người chơi một cách hiệu quả nhất.

**1.3. Mục tiêu dự án**

Dự án "Xây dựng website thương mại sheet nhạc cho game Sky: Children of the Light" được thực hiện nhằm hướng đến các mục tiêu cụ thể về cả sản phẩm và kỹ năng.

**Mục tiêu chính - Xây dựng sản phẩm thực tế**

Mục tiêu cốt lõi của dự án là xây dựng thành công một website hoạt động ổn định, đáp ứng nhu cầu chia sẻ và thương mại hóa sheet nhạc trong cộng đồng game Sky. Sản phẩm cuối cùng phải đảm bảo các chức năng cơ bản sau:

* Cho phép người dùng đăng ký, đăng nhập và quản lý tài khoản cá nhân.
* Cung cấp tính năng tải lên (upload) sheet nhạc cho người bán và tìm kiếm, xem trước cho người mua.
* Hoàn thiện quy trình mua bán cơ bản, cho phép người dùng thêm sản phẩm vào giỏ hàng và thực hiện giao dịch.

**Mục tiêu phụ: Phát triển kỹ năng chuyên môn**

Bên cạnh việc tạo ra sản phẩm, dự án còn là cơ hội để các thành viên trong nhóm đạt được các mục tiêu học thuật và kỹ năng mềm, bao gồm:

* Áp dụng kiến thức môn học: Vận dụng các lý thuyết và phương pháp của môn Phân tích Thiết kế Phần mềm vào một bài toán thực tế, từ khâu phân tích yêu cầu, vẽ biểu đồ UML, thiết kế kiến trúc cho đến kiểm thử.
* Rèn luyện kỹ năng làm việc nhóm: Nâng cao khả năng phối hợp, trao đổi thông tin, và giải quyết xung đột khi làm việc chung.
* Thực hành quản lý dự án: Tập làm quen với việc lập kế hoạch, phân chia công việc và quản lý tiến độ bằng các công cụ hỗ trợ Git/GitHub.

**1.4. Phạm vi dự án**

Để đảm bảo dự án được hoàn thành đúng tiến độ và tập trung vào các giá trị cốt lõi, nhóm đã xác định rõ ràng các chức năng sẽ được xây dựng và những tính năng sẽ tạm thời lược bỏ trong khuôn khổ bài tập lớn này.

**Các chức năng trong phạm vi phát triển**

Hệ thống sẽ được xây dựng để phục vụ bốn nhóm đối tượng chính: Người dùng vãng lai (Guest), là những người chưa có tài khoản; Thành viên (Member), là những người dùng đã đăng nhập; Người bán (Seller), là thành viên có thêm chức năng bán hàng; và Quản trị viên (Admin).

**Chức năng trong phạm vi phát triển**

Hệ thống sẽ được xây dựng để phục vụ bốn nhóm đối tượng chính: Người dùng vãng lai (Guest), là những người chưa có tài khoản; Thành viên (Member), là những người dùng đã đăng nhập; Người bán (Seller), là thành viên có thêm chức năng bán hàng; và Quản trị viên (Admin).

1. Đối với Guest:
   * Thực hiện tìm kiếm và xem thông tin chi tiết của các sheet nhạc, bao gồm giá, người đăng, và xem trước một phần nội dung.
   * Xem các bài viết công khai trên trang Cộng đồng.
   * Thực hiện đăng ký để tạo tài khoản mới hoặc đăng nhập vào tài khoản đã có.
2. Đối với Member:
   * Thực hiện các chức năng của Guest, đồng thời có thể quản lý thông tin cá nhân và đăng xuất khỏi tài khoản.
   * Thực hiện quy trình mua hàng, bao gồm việc thêm sheet nhạc vào giỏ hàng và sở hữu sản phẩm sau khi hoàn tất.
   * Đăng và quản lý (chỉnh sửa, gỡ bỏ) các bài viết của chính mình trên trang Cộng đồng.
3. Đối với Seller:
   * Kế thừa tất cả các quyền của một Member.
   * Tải lên các file sheet nhạc, điền thông tin mô tả, đặt giá, và quản lý các sản phẩm đã đăng bán trên trang cá nhân của mình.
4. Đối với Admin:
   * Truy cập bảng điều khiển (Dashboard) để xem các thống kê tổng quan về hoạt động của website.
   * Quản lý tài khoản của tất cả người dùng, bao gồm xem danh sách và thực hiện các thao tác như khóa/mở khóa tài khoản.
   * Kiểm duyệt và quản lý toàn bộ nội dung trên website, bao gồm sheet nhạc, đơn hàng và bài viết của thành viên.

**Các chức năng ngoài phạm vi phát triển**

Để tránh việc dự án trở nên quá tải và không thể hoàn thành, nhóm quyết định sẽ không thực hiện các chức năng phức tạp sau trong phiên bản này:

* Hệ thống đánh giá và bình luận sản phẩm: Chức năng cho phép người mua để lại review, xếp hạng sao cho sheet nhạc.
* Live chat hỗ trợ: Công cụ chat trực tiếp giữa người mua, người bán hoặc với quản trị viên.
* Tích hợp cổng thanh toán thực tế: Quy trình thanh toán sẽ được giả lập để hoàn thành luồng chức năng, không tích hợp với các dịch vụ thanh toán bên thứ ba như MoMo, ZaloPay,...
* Hệ thống gợi ý sản phẩm: Tự động đề xuất các sheet nhạc tương tự dựa trên lịch sử xem hoặc mua hàng của người dùng.
* Chức năng xã hội: Các tính năng như theo dõi (follow) người bán, tạo danh sách yêu thích,...

**1.5. Lựa chọn mô hình quy trình phát triển**

Sau khi xem xét các mô hình phát triển phần mềm phổ biến, nhóm chúng em đã quyết định lựa chọn và áp dụng Mô hình phát triển linh hoạt (Agile), cụ thể là khung làm việc Scrum.

Scrum là một khung làm việc (framework) cho phép giải quyết các bài toán phức tạp nhưng vẫn đảm bảo bàn giao sản phẩm có giá trị cao nhất một cách hiệu quả và sáng tạo. Dự án sẽ được chia thành các giai đoạn lặp lại ngắn, gọi là Sprint (thường kéo dài 1-2 tuần), với mục tiêu là tạo ra một phần của sản phẩm có thể hoạt động được sau mỗi Sprint.

**Lý do lựa chọn mô hình Scrum**

Mô hình này được lựa chọn dựa trên sự phù hợp với đặc thù của dự án và quy mô của nhóm, cụ thể như sau:

1. Phù hợp với quy mô nhóm nhỏ (3 thành viên): Scrum được thiết kế để phát huy hiệu quả tối đa với các nhóm nhỏ (từ 3 đến 9 thành viên). Với 3 thành viên, việc giao tiếp và trao đổi thông tin diễn ra nhanh chóng, giúp các cuộc họp hằng ngày (Daily Scrum) trở nên hiệu quả, đảm bảo mọi người luôn nắm được tiến độ và khó khăn của nhau.
2. Linh hoạt và dễ dàng thích ứng với thay đổi: Đối với một dự án bài tập lớn, các yêu cầu ban đầu có thể chưa hoàn toàn chi tiết hoặc có thể thay đổi sau khi nhận được phản hồi từ giảng viên hướng dẫn. Scrum cho phép nhóm linh hoạt điều chỉnh các ưu tiên và chức năng sau mỗi Sprint, giảm thiểu rủi ro đi sai hướng và đảm bảo sản phẩm cuối cùng đáp ứng tốt nhất mục tiêu đề ra.
3. Phân chia công việc rõ ràng và quản lý tiến độ hiệu quả: Toàn bộ các chức năng của website sẽ được đưa vào một danh sách chung (Product Backlog). Đầu mỗi Sprint, nhóm sẽ cùng nhau chọn ra các công việc quan trọng nhất để thực hiện (Sprint Backlog). Cách làm này giúp 3 thành viên dễ dàng phân chia công việc, tự quản lý nhiệm vụ của mình và theo dõi tiến độ chung của toàn đội một cách trực quan.
4. Khuyến khích sự hợp tác và bàn giao sản phẩm liên tục: Thay vì chờ đến cuối kỳ mới có sản phẩm, Scrum hướng nhóm đến việc tạo ra các phần nhỏ của website sau mỗi Sprint. Điều này không chỉ giúp nhóm nhận được phản hồi sớm để cải thiện mà còn tạo động lực khi thấy được kết quả rõ ràng sau từng giai đoạn ngắn, đồng thời đảm bảo sản phẩm luôn trong trạng thái có thể demo được.

Với những ưu điểm trên, nhóm tin rằng việc áp dụng Scrum sẽ là phương pháp tối ưu để quản lý dự án một cách hiệu quả, đảm bảo sự hợp tác chặt chẽ giữa các thành viên và bàn giao sản phẩm cuối kỳ đạt chất lượng tốt nhất.

**Chương 2: Phân tích yêu cầu**

**2.1. Yêu cầu chức năng**

Dựa trên việc phân tích các đối tượng sẽ tương tác với hệ thống, các yêu cầu chức năng được xác định và phân nhóm theo 4 tác nhân chính: Người dùng vãng lai (chưa đăng nhập), Thành viên (đã đăng nhập), Người bán (Seller), và Quản trị viên (Admin).

**2.1.1. Nhóm chức năng dành cho người dùng vãng lai (Guest)**

Đây là các chức năng mà bất kỳ ai truy cập vào website đều có thể sử dụng mà không cần tài khoản.

|  |  |  |
| --- | --- | --- |
| **ID** | **Tên chức năng** | **Mô tả chi tiết** |
| **F01** | Xem Trang chủ | Hiển thị trang chủ với các nội dung chính, sản phẩm nổi bật. |
| **F02** | Xem trang Shop | Hiển thị danh sách các sản phẩm đang bán. Người dùng có thể xem hình ảnh và giá cả. |
| **F03** | Tìm kiếm/Lọc sản phẩm | Cho phép người dùng tìm kiếm sản phẩm theo tên hoặc sử dụng các bộ lọc (ví dụ: theo giá, theo danh mục). |
| **F04** | Xem chi tiết sản phẩm | Khi bấm vào một sản phẩm, người dùng có thể xem trang chi tiết với mô tả đầy đủ, hình ảnh và thông số kỹ thuật. |
| **F05** | Xem trang Cộng đồng | Hiển thị danh sách các bài viết, thảo luận công khai từ cộng đồng người dùng. |
| **F06** | Xem chi tiết bài viết | Cho phép người dùng đọc nội dung của một bài viết cụ thể trong mục Cộng đồng. |
| **F07** | Truy cập trang hỗ trợ | Cho phép người dùng xem các trang thông tin tĩnh như Câu hỏi thường gặp (FAQ), Hướng dẫn sử dụng, Chính sách. |
| **F08** | Đăng ký tài khoản | Cung cấp form để người dùng tạo một tài khoản mới. |
| **F09** | Đăng nhập | Cung cấp form để người dùng đăng nhập vào tài khoản đã có. |
| **F9.1** | Quên mật khẩu | Cung cấp form để người dùng reset lại mật khẩu mới |

**Bảng 1.1**

**2.1.2. Nhóm chức năng dành cho thành viên đã đăng nhập**

Sau khi đăng nhập, thành viên sẽ có thêm các quyền hạn và chức năng sau, áp dụng cho cả vai trò người mua và người bán.

|  |  |  |
| --- | --- | --- |
| **ID** | **Tên chức năng** | **Mô tả chi tiết** |
| **F10** | Đăng xuất | Cho phép thành viên đăng xuất khỏi tài khoản hiện tại. |
| **F11** | Quản lý thông tin cá nhân | Cho phép thành viên xem và chỉnh sửa các thông tin cá nhân (ví dụ: ảnh đại diện, mật khẩu). |
| **F12** | Thêm vào giỏ hàng | Cho phép thành viên thêm một hoặc nhiều sheet nhạc vào giỏ hàng để chuẩn bị mua. |
| **F13** | Quản lý giỏ hàng | Cung cấp trang xem danh sách các sản phẩm trong giỏ hàng, có thể thay đổi số lượng hoặc xóa sản phẩm. |
| **F14** | Mua sheet nhạc | Thực hiện quy trình mua các sheet nhạc có trong giỏ hàng. Sau khi mua thành công, thành viên có quyền truy cập vào file sheet nhạc. |
| **F15** | Đăng bài viết trong trang Cộng đồng | Cung cấp giao diện để thành viên đăng những bài viết liên quan |
| **F16** | Quản lý bài viết đã đăng | Cung cấp trang quản lý cá nhân để xem lại các bài viết đã đăng, có thể chỉnh sửa thông tin hoặc gỡ bài. |
| **F28** | Nạp Xu vào tài khoản | Cung cấp giao diện để thành viên có thể nạp "Xu" (đơn vị tiền tệ ảo) vào tài khoản của mình thông qua một quy trình giả lập. |
| **F29** | Xem số dư và lịch sử giao dịch | Cho phép thành viên xem số dư "Xu" hiện tại và lịch sử các giao dịch (nạp xu, mua sheet nhạc) của mình. |

**Bảng 1.2**

**2.1.3. Nhóm chức năng dành cho Người bán (Seller)**

Đây là các chức năng đặc thù mà chỉ những thành viên có vai trò là Người bán mới có thể thực hiện.

|  |  |  |
| --- | --- | --- |
| **ID** | **Tên chức năng** | **Mô tả chi tiết** |
| **F17** | Tải lên sheet nhạc | Cung cấp giao diện để người bán tải lên file sheet nhạc, điền thông tin mô tả, đặt giá và đăng bán sản phẩm. |
| **F18** | Quản lý sheet nhạc đã đăng | Cung cấp trang quản lý cá nhân để xem danh sách các sheet nhạc đã đăng, có thể chỉnh sửa thông tin hoặc gỡ bán. |
| **F30** | Yêu cầu rút tiền | Cho phép người bán tạo yêu cầu rút tiền từ doanh thu đã tích lũy trong tài khoản chờ rút của họ. |

**Bảng 1.3**

**2.1.3. Nhóm chức năng dành cho Quản trị viên (Admin)**

Đây là các chức năng đặc quyền dành cho người quản trị hệ thống để đảm bảo website vận hành ổn định và đúng quy định.

|  |  |  |
| --- | --- | --- |
| **ID** | **Tên chức năng** | **Mô tả chi tiết** |
| **F19** | Đăng nhập & Đăng xuất | Admin có thể đăng nhập vào trang quản trị riêng và đăng xuất khỏi hệ thống. |
| **F20** | Xem Bảng điều khiển (Dashboard) | Hiển thị một trang tổng quan với các số liệu thống kê chính, bao gồm: Tổng doanh thu, tổng số đơn hàng, tổng số người dùng, và tổng số sheet nhạc. Trang này cũng có biểu đồ doanh thu và danh sách các sheet nhạc bán chạy nhất. |
| **F21** | Quản lý Sheet Nhạc | Cho phép Admin truy cập vào mục "Quản lý sheet nhạc" để xem, thêm, sửa, hoặc xóa các sheet nhạc trên toàn hệ thống. |
| **F22** | Quản lý đơn hàng | Cho phép Admin truy cập vào mục "Đơn hàng" để xem danh sách và chi tiết các giao dịch đã được thực hiện trên website. |
| **F23** | Quản lý người nùng | Cho phép Admin truy cập vào mục "Người dùng" để xem danh sách người dùng, thông tin chi tiết và thực hiện các tác vụ quản lý như khóa/mở khóa tài khoản. |
| **F24** | Xem thống kê | Cho phép Admin truy cập vào mục "Thống Kê" để xem các báo cáo và phân tích chi tiết hơn về hoạt động của website (ví dụ: doanh thu theo thời gian, xu hướng người dùng). |
| **F25** | Cài đặt Hệ thống | Cho phép Admin truy cập vào mục "Cài Đặt" để cấu hình các thiết lập chung cho trang web. |
| **F26** | Thêm thông báo | Admin có thể xem các thông báo quan trọng của hệ thống thông qua mục "Thông báo". |
| **F27** | Quản lý bài viết | Quản lý tất cả bài viết đã đăng (thông báo từ admin và bài đăng của thành viên) |
| **F28** | Nạp xu vào tài khoản | Cung cấp giao diện để thành viên có thể nạp "Xu" (đơn vị tiền tệ ảo) vào tài khoản của mình thông qua một quy trình giả lập. |
| **F29** | Xem số dư và lịch sử giao dịch | Cho phép thành viên xem số dư "Xu" hiện tại và lịch sử các giao dịch (nạp xu, mua sheet nhạc) của mình. |
| **F30** | Yêu cầu rút tiền | Cho phép người bán tạo yêu cầu rút tiền từ doanh thu đã tích lũy trong tài khoản chờ rút của họ. |
| **F31** | Quản lý yêu cầu rút tiền | Cho phép Admin xem, duyệt, hoặc từ chối các yêu cầu rút tiền từ Người bán. |

**Bảng 1.4**

**2.2. Yêu cầu phi chức năng**

Bên cạnh các yêu cầu về chức năng, hệ thống cũng cần phải đáp ứng các yêu cầu về chất lượng, hiệu suất và trải nghiệm người dùng. Các yêu cầu phi chức năng chính của dự án bao gồm:

**Hiệu năng (Performance)**

* Tốc độ tải trang: Thời gian tải các trang quan trọng (như trang chủ, trang chi tiết sản phẩm) phải dưới 3 giây trong điều kiện mạng thông thường để đảm bảo trải nghiệm người dùng mượt mà và tránh việc người dùng rời bỏ trang.
* Khả năng đáp ứng của máy chủ: Hệ thống phải có khả năng xử lý đồng thời một lượng truy cập nhất định mà không gây ra tình trạng quá tải hay treo hệ thống.

**Bảo mật (Security)**

* Mã hóa mật khẩu: Mật khẩu của tất cả người dùng phải được mã hóa bằng các thuật toán băm (hashing) hiện đại trước khi lưu trữ vào cơ sở dữ liệu để ngăn chặn việc lộ thông tin ngay cả khi cơ sở dữ liệu bị xâm nhập.
* Phân quyền truy cập: Hệ thống phải có cơ chế phân quyền truy cập rõ ràng. Người dùng thường (người mua) không thể truy cập các chức năng của người bán (ví dụ: tải lên sheet nhạc). Chỉ có Quản trị viên mới có quyền truy cập vào các chức năng quản trị hệ thống.

**Khả năng sử dụng (Usability)**

* Giao diện thân thiện: Giao diện người dùng (UI) cần được thiết kế sạch sẽ, trực quan và nhất quán trên toàn bộ trang web để người dùng có thể dễ dàng tìm thấy thông tin và thực hiện các thao tác mong muốn mà không cần hướng dẫn phức tạp.
* Thiết kế đáp ứng (Responsive Design): Website phải tương thích và hiển thị tốt trên nhiều loại thiết bị khác nhau, từ máy tính để bàn, máy tính bảng cho đến điện thoại di động, đảm bảo trải nghiệm người dùng không bị gián đoạn.

**Khả năng bảo trì (Maintainability)**

* Mã nguồn có cấu trúc: Mã nguồn của dự án cần được tổ chức theo các module rõ ràng, tuân thủ các quy chuẩn lập trình và có các chú thích cần thiết để các thành viên trong nhóm và các nhà phát triển trong tương lai có thể dễ dàng đọc hiểu, sửa lỗi và phát triển thêm các tính năng mới.

**2.3. Mô hình hóa yêu cầu**

**2.3.1. Biểu đồ use case**

Để thể hiện một cách tổng quan các tương tác chính giữa người dùng và hệ thống, nhóm đã xây dựng biểu đồ use case. Biểu đồ này được thiết kế dựa trên các yêu cầu chức năng đã được xác định trong mục 2.1.

**Xác định các use case chính**

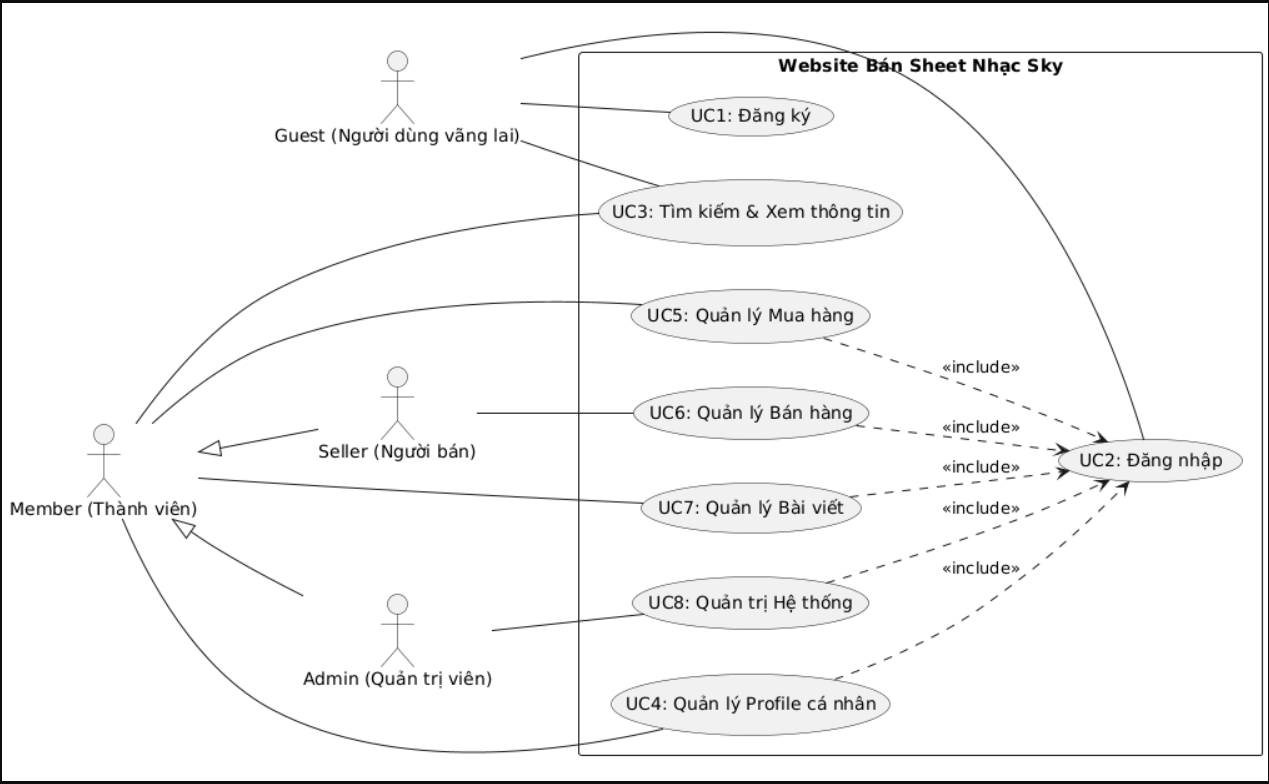
Từ danh sách các yêu cầu chức năng (F01-F27), các use case chính được gom nhóm như sau:

* **UC1:** Đăng ký
* **UC2:** Đăng nhập
* **UC3:** Tìm kiếm & Xem thông tin
* **UC4:** Quản lý Profile cá nhân
* **UC5:** Quản lý Mua hàng
* **UC6:** Quản lý Bán hàng
* **UC7:** Quản lý Bài viết
* **UC8:** Quản trị Hệ thống (gồm quản lý người dùng, sản phẩm, đơn hàng, bài viết)

|  |  |  |
| --- | --- | --- |
| **Use Case (UC)** | **Mô tả** | **Yêu cầu chức năng (F)** |
| UC1: Đăng ký | Cho phép người dùng vãng lai tạo một tài khoản mới để trở thành thành viên | F08: Đăng ký tài khoản |
| UC2: Đăng nhập | Cho phép người dùng đã có tài khoản xác thực và truy cập vào hệ thống | F09: Đăng nhập  F9.1: Quên mật khẩu  F19: Đăng nhập & Đăng xuất (Admin) |
| UC3: Tìm kiếm & Xem thông tin | Bao gồm tất cả các hoạt động xem và tìm kiếm thông tin công khai trên website | F01: Xem trang chủ  F02: Xem trang shop  F03: Tìm kiếm/lọc sản phẩm  F04: Xem chi tiết sản phẩm  F05: Xem trang cộng đồng  F06: Xem chi tiết bài viết  F07: Truy cập trang hỗ trợ |
| UC4: Quản lý Profile cá nhân | Cho phép thành viên quản lý thông tin cá nhân và phiên đăng nhập của mình | F10: Đăng xuất  F11: Quản lý thông tin cá nhân  F28: Nạp Xu vào tài khoản  F29: Xem số dư và lịch sử giao dịch |
| UC5: Quản lý Mua hàng | Bao gồm toàn bộ luồng nghiệp vụ mua một sheet nhạc của thành viên | F12: Thêm vào giỏ hàng  F13: Quản lý giỏ hàng  F14: Mua sheet nhạc |
| UC6: Quản lý Bán hàng | Các chức năng dành riêng cho Seller để quản lý sản phẩm của họ | F17: Tải lên sheet nhạc  F18: Quản lý sheet nhạc đã đăng  F30: Yêu cầu rút tiền |
| UC7: Quản lý Bài viết | Cho phép thành viên tạo và quản lý nội dung của riêng họ trên trang Cộng đồng | F15: Đăng bài viết  F16: Quản lý bài viết đã đăng |
| UC8: Quản trị Hệ thống | Các chức năng quản trị cấp cao của Admin để vận hành và kiểm soát website | F20: Xem Dashboard  F21: Quản lý Sheet Nhạc  F22: Quản lý đơn hàng  F23: Quản lý người dùng  F24: Xem thống kê  F25: Cài đặt Hệ thống  F26: Thêm thông báo  F27: Quản lý bài viết  F31: Quản lý yêu cầu rút tiền |

**Bảng 1.5**

**Biểu đồ Use Case (Use Case Diagram):**

****

**Hình 2.1**

* + 1. **Đặc tả Use Case**

**UC2: Đăng nhập**

Mô tả: Use Case này cho phép người dùng (Guest, Member, Seller, Admin) đã có tài khoản đăng nhập vào hệ thống để truy cập các chức năng cá nhân hóa và các chức năng tương ứng với vai trò của họ.

**Luồng sự kiện:**

* + **Luồng cơ bản:**
    1. Use Case bắt đầu khi người dùng muốn đăng nhập vào hệ thống.
    2. Người dùng truy cập giao diện trang "Đăng nhập".
    3. Hệ thống hiển thị form đăng nhập, yêu cầu người dùng nhập thông tin (Tên đăng nhập/Email, Mật khẩu).
    4. Người dùng nhập đầy đủ và chính xác thông tin, sau đó nhấn nút "Đăng nhập".
    5. Hệ thống kiểm tra thông tin đăng nhập trong cơ sở dữ liệu.
    6. Hệ thống xác thực thông tin hợp lệ, tạo phiên làm việc (session) và chuyển hướng người dùng vào hệ thống với vai trò tương ứng (trang chủ cho Member/Seller, trang Dashboard cho Admin). Use Case kết thúc thành công.
  + **Luồng phụ:**
    1. **S-1: Quên mật khẩu**
       1. Tại bước 4 của Luồng cơ bản, người dùng chọn chức năng "Quên mật khẩu" trên giao diện đăng nhập.
       2. Hệ thống chuyển hướng sang trang "Khôi phục mật khẩu", yêu cầu người dùng nhập địa chỉ email đã đăng ký.
       3. Người dùng nhập email và nhấn nút "Gửi yêu cầu".
       4. Hệ thống xác thực email có tồn tại trong cơ sở dữ liệu.
       5. Hệ thống gửi một email chứa đường link đặt lại mật khẩu đến cho người dùng.
       6. Hệ thống hiển thị thông báo "Yêu cầu đã được gửi, vui lòng kiểm tra email của bạn". Use Case con này kết thúc.
  + **Luồng thay thế:**
    1. **A-1: Thông tin đăng nhập không hợp lệ**
       1. Tại bước 5 của Luồng cơ bản, nếu hệ thống xác thực thông tin không hợp lệ (sai tên đăng nhập hoặc mật khẩu).
       2. Hệ thống sẽ hiển thị thông báo lỗi: "Tên đăng nhập hoặc mật khẩu không chính xác."
       3. Người dùng ở lại trang Đăng nhập và luồng sự kiện quay lại bước 4.

**Tiền điều kiện:**

* + Người dùng phải có một tài khoản đã được đăng ký và kích hoạt trong hệ thống.
  + Hệ thống đang hoạt động và sẵn sàng xử lý yêu cầu.

**Hậu điều kiện:**

* + Thành công: Người dùng đăng nhập thành công và được chuyển hướng đến giao diện phù hợp với vai trò của họ. Một phiên làm việc được tạo ra.
  + Thất bại: Người dùng không đăng nhập được, ở lại trang đăng nhập và nhận được thông báo lỗi. Trạng thái hệ thống không thay đổi.

A screenshot of a diagram

AI-generated content may be incorrect.

Hình 2.2

**UC5: Quản lý Mua hàng**

Mô tả: Use Case này mô tả quá trình một thành viên đã đăng nhập sử dụng "Xu" trong tài khoản để mua các sheet nhạc có trong giỏ hàng của họ.

**Luồng sự kiện:**

* + **Luồng cơ bản:**
    1. Use Case bắt đầu khi Member muốn thanh toán các sản phẩm trong giỏ hàng.
    2. Member truy cập vào trang "Giỏ hàng".
    3. Hệ thống hiển thị danh sách các sheet nhạc, tổng số "Xu" phải thanh toán và số dư "Xu" hiện tại của Member.
    4. Member nhấn nút "Thanh toán".
    5. Hệ thống kiểm tra lại một lần nữa và xác nhận số dư của Member đủ để thực hiện giao dịch.
    6. Hệ thống trừ số "Xu" tương ứng từ tài khoản của Member, đồng thời cộng doanh thu (sau khi trừ phí) vào tài khoản chờ rút của (các) Seller.
    7. Hệ thống tạo một bản ghi Đơn hàng mới và cấp quyền truy cập file sheet nhạc cho Member.
    8. Hệ thống hiển thị thông báo "Thanh toán thành công!" và dọn sạch giỏ hàng. Use Case kết thúc thành công.
  + **Luồng phụ:**
    1. S-1: Nạp thêm Xu
       1. Tại bước 3 của Luồng cơ bản, nếu Member thấy số dư không đủ, họ có thể nhấn vào nút "Nạp Xu".
       2. Hệ thống chuyển hướng Member đến trang Nạp Xu (chức năng F28).
       3. Sau khi Member nạp Xu thành công, Hệ thống đưa họ trở lại trang Giỏ hàng. Luồng sự kiện quay lại bước 3 của Luồng cơ bản.
  + **Luồng thay thế:**
    1. A-1: Số dư không đủ khi thanh toán
       1. Tại bước 5 của Luồng cơ bản, nếu số dư của Member không đủ (ví dụ: do có một giao dịch khác diễn ra song song), Hệ thống sẽ hiển thị thông báo lỗi "Số dư không đủ, vui lòng nạp thêm!".
       2. Luồng sự kiện quay lại bước 3.
    2. A-2: Lỗi xử lý giao dịch
       1. Tại bước 6 của Luồng cơ bản, nếu có lỗi xảy ra (ví dụ: lỗi cơ sở dữ liệu), Hệ thống sẽ hủy bỏ mọi thay đổi (rollback).
       2. Hệ thống hiển thị thông báo "Đã có lỗi xảy ra, vui lòng thử lại sau." và Use Case kết thúc thất bại.

**Tiền điều kiện:**

* + Member phải đăng nhập vào hệ thống (<<include>> UC2).
  + Giỏ hàng của Member phải có ít nhất một sản phẩm.

**Hậu điều kiện:**

* + **Thành công:** Một Đơn hàng mới được tạo. Số dư của Member và Seller được cập nhật. Member có quyền truy cập sản phẩm đã mua.
  + **Thất bại:** Trạng thái hệ thống (số dư, đơn hàng) không thay đổi. Người dùng nhận được thông báo lỗi tương ứng.

A diagram with text and words

AI-generated content may be incorrect.

**Chương 3: Thiết kế Hệ thống (40% điểm)**

Chương này trình bày chi tiết về quá trình thiết kế hệ thống cho website thương mại sheet nhạc. Quá trình thiết kế được chia thành hai phần chính: thiết kế kiến trúc ở mức độ tổng quan để xác định cấu trúc hệ thống, và thiết kế chi tiết sử dụng các biểu đồ UML để mô hình hóa các thành phần và tương tác bên trong.

**3.1. Thiết kế kiến trúc**

* *Lựa chọn mô hình kiến trúc cho website (ví dụ: 3-Tier, Microservices).*
* *Giải thích lý do lựa chọn và vẽ sơ đồ tổng quan của kiến trúc đó.*

Thiết kế kiến trúc là bước nền tảng, xác định cấu trúc cấp cao của hệ thống, cách các thành phần chính được tổ chức và tương tác với nhau. Lựa chọn một kiến trúc phù hợp là yếu tố then chốt, ảnh hưởng trực tiếp đến các yêu cầu phi chức năng như khả năng bảo trì, mở rộng, hiệu năng và bảo mật của sản phẩm.

**3.1.1. Lựa chọn Mô hình Kiến trúc**

Dựa trên quy mô và các yêu cầu đã xác định của dự án, các mô hình kiến trúc phổ biến như Monolithic (kiến trúc nguyên khối), Microservices, và Three-Tier (kiến trúc 3 tầng) đã được xem xét để đưa ra lựa chọn tối ưu.

* **Kiến trúc Monolithic:** Toàn bộ ứng dụng được xây dựng như một khối duy nhất. Mô hình này có ưu điểm là đơn giản trong việc phát triển và triển khai ở giai đoạn đầu. Tuy nhiên, khi hệ thống phát triển phức tạp hơn, việc bảo trì, sửa lỗi và cập nhật một thành phần nhỏ có thể ảnh hưởng đến toàn bộ hệ thống, đi ngược lại yêu cầu về "Khả năng bảo trì" (Maintainability) đã đề ra.
* **Kiến trúc Microservices:** Hệ thống được chia thành nhiều dịch vụ nhỏ, độc lập. Mô hình này mang lại sự linh hoạt và khả năng mở rộng vượt trội. Tuy nhiên, việc triển khai và quản lý một hệ thống Microservices đòi hỏi sự phức tạp cao về mặt hạ tầng và vận hành, không phù hợp với một dự án có quy mô vừa và nhỏ với đội ngũ phát triển gồm 3 thành viên.
* **Kiến trúc Three-Tier (3-Lớp):** Mô hình này phân tách ứng dụng thành ba tầng logic riêng biệt: Trình bày, Nghiệp vụ và Dữ liệu. Nó cung cấp một sự cân bằng lý tưởng giữa tính đơn giản và khả năng tổ chức, phù hợp với các ứng dụng web truyền thống và đáp ứng tốt các yêu cầu của dự án.

Sau khi phân tích, mô hình kiến trúc **Three-Tier (3-Lớp)** được lựa chọn làm kiến trúc mục tiêu cho hệ thống.

**Lý do lựa chọn:**

* **Phân tách rõ ràng (Clear Separation):** Kiến trúc 3-Lớp phân chia hệ thống thành ba tầng logic độc lập: Tầng Trình bày (Presentation), Tầng Nghiệp vụ (Business Logic), và Tầng Dữ liệu (Data). Sự phân tách này hoàn toàn phù hợp với yêu cầu về "Mã nguồn có cấu trúc" và "Khả năng bảo trì". Nó cho phép các thành viên trong nhóm có thể phát triển song song trên các tầng khác nhau (ví dụ: một thành viên tập trung vào giao diện người dùng ở Tầng Trình bày, trong khi các thành viên khác xây dựng logic ở Tầng Nghiệp vụ) mà ít gây ảnh hưởng chéo, từ đó tối ưu hóa hiệu suất làm việc nhóm.
* **Tăng cường bảo mật:** Việc cô lập Tầng Dữ liệu khỏi sự truy cập trực tiếp từ client (Tầng Trình bày) là một nguyên tắc bảo mật cơ bản. Mọi yêu cầu từ người dùng phải đi qua Tầng Nghiệp vụ để được xác thực và kiểm tra quyền hạn trước khi có thể tương tác với cơ sở dữ liệu. Điều này trực tiếp đáp ứng yêu cầu "Phân quyền truy cập" đã được xác định, ngăn chặn các truy cập trái phép vào dữ liệu nhạy cảm.
* **Phù hợp với công nghệ hiện đại:** Kiến trúc này tương thích hoàn hảo với các framework phát triển web hiện đại. Tầng Trình bày có thể được xây dựng bằng ReactJS, trong khi Tầng Nghiệp vụ có thể sử dụng các framework mạnh mẽ như Node.js hoặc Laravel, vốn là những công nghệ được đề xuất trong kế hoạch phát triển của dự án.
* **Khả năng mở rộng hợp lý:** Mặc dù không linh hoạt bằng Microservices, kiến trúc 3-Lớp vẫn cho phép mở rộng từng tầng một cách độc lập khi cần thiết. Ví dụ, khi lưu lượng truy cập tăng, hệ thống có thể bổ sung thêm máy chủ web ở Tầng Trình bày hoặc nhân bản máy chủ ứng dụng ở Tầng Nghiệp vụ mà không ảnh hưởng đến các tầng khác.

**3.1.2. Sơ đồ Kiến trúc Tổng quan và Mô tả các Tầng**

Sơ đồ dưới đây minh họa kiến trúc 3-Lớp của hệ thống và luồng tương tác giữa các thành phần, từ người dùng cuối đến cơ sở dữ liệu.

*(Lưu ý: Hình ảnh sơ đồ kiến trúc sẽ được trình bày ở đây, minh họa luồng dữ liệu từ Client (Browser/Mobile) -> Presentation Tier (Web Server) -> Business Logic Tier (Application Server) -> Data Tier (Database/File Storage).)*

**Mô tả chi tiết các tầng:**

* **Tầng Trình bày (Presentation Tier)**
  + **Trách nhiệm:** Đây là tầng giao diện người dùng (UI), chịu trách nhiệm chính trong việc hiển thị dữ liệu cho người dùng và thu thập các thông tin đầu vào từ họ. Tầng này được thiết kế để không chứa bất kỳ logic nghiệp vụ phức tạp nào, đảm bảo sự tách biệt rõ ràng.
  + **Thành phần:** Bao gồm các thành phần giao diện được xây dựng bằng HTML, CSS, và JavaScript, sử dụng một framework hiện đại như ReactJS để tạo ra trải nghiệm người dùng tương tác và linh hoạt. Tầng này sẽ render các trang chính của website như Trang chủ (F01), Trang Shop (F02), Trang chi tiết sản phẩm (F04), Giỏ hàng (F13), cũng như các biểu mẫu chức năng như Đăng ký (F08) và Đăng nhập (F09).
  + **Tương tác:** Tầng Trình bày giao tiếp với Tầng Nghiệp vụ thông qua các lời gọi API (Application Programming Interface), điển hình là RESTful API qua giao thức HTTP/HTTPS. Ví dụ, khi người dùng thực hiện tìm kiếm sản phẩm, Tầng Trình bày sẽ gửi một yêu cầu HTTP GET đến một endpoint API cụ thể trên Tầng Nghiệp vụ để lấy về danh sách sản phẩm phù hợp.
* **Tầng Nghiệp vụ (Business Logic Tier / Application Tier)**
  + **Trách nhiệm:** Đây là "bộ não" của hệ thống, nơi xử lý tất cả các quy tắc nghiệp vụ, logic xử lý dữ liệu, và điều phối luồng công việc của ứng dụng. Mọi yêu cầu từ Tầng Trình bày đều được xử lý tại đây trước khi tương tác với Tầng Dữ liệu.
  + **Thành phần:** Bao gồm các máy chủ ứng dụng (Application Servers) chạy mã nguồn backend (ví dụ: Node.js, PHP/Laravel). Tầng này được tổ chức thành các module logic để xử lý từng nhóm chức năng cụ thể:
    - *Module Quản lý Xác thực và Phân quyền:* Xử lý các nghiệp vụ đăng ký (UC1), đăng nhập (UC2), và thực thi cơ chế phân quyền, đảm bảo người dùng chỉ có thể truy cập các chức năng phù hợp với vai trò của mình (Member, Seller, Admin).
    - *Module Xử lý Nghiệp vụ Thương mại điện tử:* Quản lý giỏ hàng, tạo và xử lý đơn hàng (UC5), và thực hiện các giao dịch thanh toán bằng đơn vị tiền tệ ảo "Xu".
    - *Module Quản lý Sheet nhạc:* Xử lý việc tải lên (UC6), kiểm duyệt nội dung, và cung cấp dữ liệu sản phẩm cho Tầng Trình bày.
    - *Dịch vụ Giao dịch "Xu":* Do hệ thống sử dụng đơn vị tiền tệ ảo "Xu" cho mọi giao dịch , một module dịch vụ giao dịch chuyên biệt sẽ được thiết kế tại tầng này. Module này không chỉ đơn thuần cập nhật số dư của người dùng, mà còn chịu trách nhiệm ghi lại mọi giao dịch (nạp "Xu", mua hàng, nhận doanh thu) vào một sổ cái giao dịch. Cách tiếp cận này đảm bảo tính toàn vẹn, minh bạch và khả năng truy vết của dữ liệu tài chính trong hệ thống.
    - *Dịch vụ Chuyển đổi Sheet nhạc:* Để hiện thực hóa tính năng xem trước sheet nhạc một cách trực quan, một dịch vụ chuyên biệt sẽ được xây dựng để tự động chuyển đổi từ định dạng "Sheet Dữ liệu" (dạng file cấu trúc) sang định dạng "Sheet Số" (dạng ký hiệu đơn giản). Dịch vụ này sẽ phân tích cú pháp file gốc và tạo ra một phiên bản dữ liệu xem trước để Tầng Trình bày có thể hiển thị cho người dùng một cách dễ hiểu.
  + **Tương tác:** Nhận yêu cầu từ Tầng Trình bày, thực thi logic nghiệp vụ, sau đó giao tiếp với Tầng Dữ liệu để thực hiện các thao tác đọc/ghi thông tin cần thiết.
* **Tầng Dữ liệu (Data Tier)**
  + **Trách nhiệm:** Chịu trách nhiệm lưu trữ và quản lý toàn bộ dữ liệu của ứng dụng một cách bền vững và an toàn (persistent storage).
  + **Thành phần:** Bao gồm hệ quản trị cơ sở dữ liệu quan hệ (ví dụ: MySQL, PostgreSQL) để lưu trữ các dữ liệu có cấu trúc như thông tin người dùng, sản phẩm, đơn hàng. Ngoài ra, tầng này còn có thể bao gồm các giải pháp lưu trữ file (File Storage) như một thư mục trên máy chủ hoặc các dịch vụ lưu trữ đám mây (ví dụ: Amazon S3) để lưu trữ các file sheet nhạc do người dùng tải lên.
  + **Tương tác:** Tầng Dữ liệu được thiết kế để chỉ giao tiếp trực tiếp với Tầng Nghiệp vụ. Việc ngăn chặn mọi kết nối trực tiếp từ Tầng Trình bày đến Tầng Dữ liệu là một biện pháp bảo mật quan trọng, đảm bảo dữ liệu luôn được xử lý thông qua các lớp logic đã được kiểm duyệt.

**3.2. Thiết kế chi tiết (UML)**

Sau khi xác định kiến trúc tổng thể, bước tiếp theo là đi sâu vào thiết kế chi tiết các thành phần bên trong hệ thống bằng Ngôn ngữ Mô hình hóa Thống nhất (UML - Unified Modeling Language). Các biểu đồ UML giúp trực quan hóa cấu trúc và hành vi của hệ thống, làm cơ sở cho việc triển khai mã nguồn.

**3.2.1. Biểu đồ Lớp (Class Diagram)**

Biểu đồ Lớp mô tả cấu trúc tĩnh của hệ thống, bao gồm các lớp đối tượng chính, các thuộc tính (attributes), phương thức (methods) của chúng, và các mối quan hệ (kế thừa, kết hợp, tập hợp) giữa các lớp. Đây là bản thiết kế chi tiết cho cấu trúc cơ sở dữ liệu và các lớp mã nguồn của ứng dụng.

*(Lưu ý: Hình ảnh chi tiết của biểu đồ lớp sẽ được trình bày ở đây.)*

**Mô tả các Lớp chính:**

* **Cấu trúc Kế thừa Vai trò Người dùng:** Hệ thống có nhiều vai trò người dùng với các quyền hạn khác nhau (Member, Seller, Admin). Thay vì sử dụng một trường

role đơn giản, một cấu trúc kế thừa được áp dụng để mô hình hóa mối quan hệ này một cách tự nhiên và dễ mở rộng.

* + **User (Lớp cơ sở):** Chứa các thông tin và hành vi chung cho tất cả người dùng có tài khoản.
    - *Thuộc tính:* userID, username, hashedPassword, email, createdAt.
    - *Phương thức:* register(), login(), logout(), resetPassword().
  + **Member (Kế thừa từ User):** Đại diện cho một thành viên đã đăng nhập, có các chức năng mua hàng.
    - *Thuộc tính:* avatar, balance (số dư "Xu").
    - *Phương thức:* updateProfile(), viewTransactionHistory(), addToCart(), checkout().
  + **Seller (Kế thừa từ Member):** Là một dạng Member đặc biệt, có thêm các chức năng bán hàng.
    - *Thuộc tính:* withdrawableBalance (số dư có thể rút), storeName.
    - *Phương thức:* uploadSheetMusic(), manageSheetMusic(), requestWithdrawal().
  + **Admin (Kế thừa từ User):** Có các quyền hạn quản trị cao nhất.
    - *Phương thức:* manageUser(), moderateContent(), viewDashboard(), processWithdrawal().
* **Các lớp liên quan đến Sản phẩm và Đơn hàng:**
  + **SheetMusic:** Đại diện cho một sản phẩm sheet nhạc được đăng bán.
    - *Thuộc tính:* sheetID, title, description, price (tính bằng "Xu"), filePath, previewData. Một thuộc tính quan trọng là status ('pending', 'approved', 'rejected'). Việc bổ sung thuộc tính này là hệ quả trực tiếp từ yêu cầu kiểm duyệt nội dung của Admin (F27). Một sheet nhạc mới tải lên sẽ ở trạng thái 'pending' và chỉ được hiển thị công khai sau khi được Admin duyệt thành 'approved', đảm bảo chất lượng và an toàn cho nội dung trên website.
    - *Quan hệ:* Một Seller có thể đăng bán nhiều SheetMusic (quan hệ 1-N).
  + **Cart và CartItem:** Mô hình hóa giỏ hàng. Một đối tượng Cart thuộc về một Member và chứa một danh sách các CartItem. Mỗi CartItem liên kết đến một SheetMusic cụ thể.
  + **Order và OrderItem:** Ghi lại lịch sử giao dịch mua bán. Một Order được tạo khi thanh toán thành công, chứa thông tin về người mua và tổng giá trị. Mỗi Order bao gồm nhiều OrderItem, mỗi OrderItem là một bản ghi của một SheetMusic đã được mua tại một mức giá cụ thể vào thời điểm đó.
    - *Quan hệ:* Một Member có thể có nhiều Order (quan hệ 1-N).
* **Các lớp Quản lý Giao dịch:** Để quản lý tài chính của hệ thống "Xu" một cách đáng tin cậy và có thể kiểm toán, việc chỉ cập nhật một trường balance trong lớp User là không đủ và tiềm ẩn rủi ro. Do đó, các lớp chuyên biệt được thiết kế để ghi lại mọi giao dịch.
  + **Transaction:** Hoạt động như một sổ cái kế toán. Mỗi lần nạp tiền, mua hàng, hoặc nhận doanh thu đều tạo ra một bản ghi Transaction mới.
    - *Thuộc tính:* transactionID, userID (người liên quan), type ('deposit', 'purchase', 'sale\_revenue', 'withdrawal'), amount, timestamp, relatedOrderID.
  + **WithdrawalRequest:** Quản lý các yêu cầu rút tiền của người bán, tương ứng với chức năng F30 và F31.
    - *Thuộc tính:* requestID, sellerID, amount, status ('pending', 'approved', 'rejected'), requestDate, processedDate.
    - *Quan hệ:* Một Seller có thể có nhiều WithdrawalRequest (quan hệ 1-N).
* **Lớp Post:**
  + **Mô tả:** Đại diện cho một bài viết trong trang Cộng đồng, cho phép người dùng chia sẻ và thảo luận.
  + **Thuộc tính:** postID, authorID (tham chiếu đến Member), title, content, createdAt, status ('pending', 'approved').
  + **Quan hệ:** Một Member có thể đăng nhiều Post (quan hệ 1-N).

**3.2.2. Biểu đồ Trình tự (Sequence Diagram)**

Biểu đồ Trình tự mô tả tương tác động giữa các đối tượng trong một kịch bản hoặc use case cụ thể theo dòng thời gian. Chúng rất hữu ích để làm rõ các luồng logic phức tạp. Hai chức năng được chọn để mô hình hóa là "Thanh toán đơn hàng" (UC5) và "Tải lên sheet nhạc" (UC6).

**Biểu đồ Trình tự cho Chức năng Thanh toán (UC5: Quản lý Mua hàng)**

Kịch bản này bắt đầu khi một Member đã đăng nhập và có sản phẩm trong giỏ hàng, quyết định nhấn nút "Thanh toán".

*(Lưu ý: Hình ảnh biểu đồ trình tự cho chức năng Thanh toán sẽ được trình bày ở đây.)*

**Luồng sự kiện chính:**

1. Người dùng (:Member) nhấn nút thanh toán trên giao diện (:CartPage).
2. Giao diện gửi yêu cầu createOrder(cartData) đến lớp điều khiển (:OrderController).
3. :OrderController gọi phương thức xử lý nghiệp vụ processPayment(userID, cart) của lớp dịch vụ (:OrderService).
4. :OrderService gọi :UserService để lấy thông tin số dư "Xu" hiện tại của người dùng.
5. :OrderService tính toán tổng giá trị đơn hàng và kiểm tra xem số dư có đủ không.
6. **[Luồng thay thế]** Nếu số dư không đủ, :OrderService trả về một thông báo lỗi và luồng kết thúc.
7. Nếu số dư đủ, :OrderService bắt đầu một giao dịch cơ sở dữ liệu (begin transaction) để đảm bảo tính toàn vẹn dữ liệu.
8. :OrderService gọi :TransactionService để tạo các bản ghi giao dịch: một bản ghi trừ tiền (type='purchase') cho người mua và các bản ghi cộng tiền (type='sale\_revenue') cho (các) người bán tương ứng.
9. :OrderService tạo một bản ghi :Order và các :OrderItem liên quan trong cơ sở dữ liệu (:Database).
10. Hệ thống cấp quyền truy cập file sheet nhạc cho người mua.
11. :OrderService xóa các sản phẩm đã mua khỏi giỏ hàng của người dùng.
12. Nếu tất cả các bước trên thành công, giao dịch cơ sở dữ liệu được xác nhận (commit transaction).
13. :OrderController trả về thông báo thành công cho :CartPage để hiển thị cho người dùng.

**Biểu đồ Trình tự cho Chức năng Tải lên Sheet nhạc (UC6: Quản lý Bán hàng)**

Kịch bản này bắt đầu khi một Seller điền đầy đủ thông tin sản phẩm và nhấn nút "Đăng bán".

*(Lưu ý: Hình ảnh biểu đồ trình tự cho chức năng Tải lên Sheet nhạc sẽ được trình bày ở đây.)*

**Luồng sự kiện chính:**

1. Người dùng (:Seller) nhấn nút gửi trên biểu mẫu (:UploadForm) với dữ liệu và file nhạc.
2. Giao diện gửi yêu cầu uploadSheet(formData, file) đến lớp điều khiển (:SheetMusicController).
3. :SheetMusicController gọi phương thức xử lý nghiệp vụ createNewSheet(sellerID, sheetData, file) của lớp dịch vụ (:SheetMusicService).
4. :SheetMusicService gọi đến dịch vụ chuyên biệt :SheetMusicConverter để xác thực định dạng file và phân tích cú pháp, tạo dữ liệu xem trước. Đây là bước quan trọng để đảm bảo chất lượng dữ liệu đầu vào.
5. Nếu file hợp lệ, :SheetMusicService gọi dịch vụ lưu trữ (:FileStorage) để lưu file gốc và nhận lại đường dẫn (filePath).
6. :SheetMusicService tạo một đối tượng SheetMusic mới trong bộ nhớ với đầy đủ thông tin (tiêu đề, giá, filePath, previewData).
7. Quan trọng nhất, thuộc tính status của đối tượng này được thiết lập là **'pending'** để chờ quản trị viên duyệt.
8. :SheetMusicService lưu đối tượng SheetMusic mới này vào cơ sở dữ liệu (:Database).
9. :SheetMusicController trả về thông báo thành công cho :UploadForm.
10. Giao diện hiển thị thông báo "Tải lên thành công, sản phẩm đang chờ duyệt" cho người bán.

**3.2. Thiết kế chi tiết (UML)**

* **Biểu đồ Lớp (Class Diagram):** *Thiết kế các lớp chính của hệ thống (User, SheetMusic, Order,...) bao gồm thuộc tính, phương thức và các mối quan hệ (association, inheritance,...).*
* **Biểu đồ Trình tự (Sequence Diagram):** *Vẽ biểu đồ cho 2 chức năng phức tạp nhất (ví dụ: "Thanh toán đơn hàng", "Quản lý bản quyền sheet nhạc") để mô tả luồng tương tác giữa các đối tượng theo thời gian.*

**3.3. Thiết kế cơ sở dữ liệu**

* *Trình bày sơ đồ quan hệ thực thể (ERD) hoặc mô hình logic của cơ sở dữ liệu.*
* *Mô tả chi tiết các bảng, các trường và mối quan hệ giữa chúng.*

**3.4. Thiết kế giao diện (UI/UX)**

* *Chèn một vài bản thiết kế (mockup/prototype) cho các trang chính như Trang chủ, Trang chi tiết sản phẩm, Giỏ hàng.*

**Chương 4: Triển khai và Kiểm thử (20% điểm)**

**4.1. Công nghệ và Môi trường phát triển**

* **Ngôn ngữ lập trình:** *JavaScript, PHP,...*
* **Framework:** *ReactJS (Frontend), Node.js/Laravel (Backend),...*
* **Cơ sở dữ liệu:** *MySQL, MongoDB,...*
* **Công cụ khác:** *Git/GitHub, VS Code, Trello,...*

**4.2. Kế hoạch kiểm thử (Test Cases)**

* *Xây dựng các kịch bản kiểm thử hộp đen (Black-box testing) cho các chức năng chính (ví dụ: Test case cho chức năng đăng nhập, tìm kiếm,...).*

**4.3. Kết quả kiểm thử**

* *Trình bày kết quả thực hiện các test case đã nêu: Passed/Failed và hình ảnh minh họa nếu cần.*

**Chương 5: Tổng kết và Hướng phát triển (10% điểm)**

**5.1. Kết quả đạt được**

* *Tổng kết các chức năng đã hoàn thành và so sánh với mục tiêu ban đầu.*

**5.2. Hạn chế và Bài học kinh nghiệm**

* *Nêu những khó khăn gặp phải trong quá trình làm (kỹ thuật, làm việc nhóm,...) và những điểm còn hạn chế của sản phẩm.*
* *Bài học kinh nghiệm rút ra được sau khi thực hiện dự án.*

**5.3. Hướng phát triển trong tương lai**

* *Đề xuất các tính năng có thể bổ sung để hoàn thiện sản phẩm (ví dụ: Hệ thống rating/review, gợi ý sheet nhạc, tích hợp diễn đàn,...).*

**Lời kết**

**Tài liệu tham khảo**

<https://sky.hhgame.vn/tin-tuc/hanh-trinh-am-nhac-buoi-hoa-nhac-mua-aurora>

<https://sky-children-of-the-light.fandom.com/wiki/Instruments>