ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 4 HND Diploma in Computing**		
Unit number and title	Unit 7: Software Development Life Cycle		
Submission date	26/11/2023	**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Nguyen Huu Lam	**Student ID**	BH00572
Class	IT0603	**Assessor name**	Dinh Van Dong
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that a false declaration is a form of malpractice.			
		Student's signature	lâm

Grading grid

P5	P6	P7	M3	M4	M5	M6	D3	D4

❒ **Summative Feedback:** ❒ **Resubmission Feedback:**

Grade:	**Assessor Signature:**	**Date:**

Internal Verifier's Comments:

Signature & Date:

Table of Contents

Table of Figures

I. Introduction

This topic focuses on the software development process and its management process, called "Software Development Life Cycle" (SDLC). SDLC defines a sequence of activities from the beginning of defining requirements until implementing and maintaining the software. In this report, we will learn about the important stages and tools in SDLC.

We'll start by identifying the stakeholders and their roles in a specific software project. Next, we will look at requirements analysis using structural and behavioral modeling techniques. We will then discuss how to meet user requirements and software requirements during the design phase. Finally, we will conclude the report with an overview of the content presented.

II. *(P5)* Identify the stakeholders, their roles and interests in the case study.

1. Identify Business Needs

a. Primary request

The outlined project for Tune Source, as described in Assignment 1, underscores a pivotal need for the business—to facilitate the sale of digital music through both online platforms and in-store kiosks. The primary objective is clear: the incorporation of the option to purchase digital music is anticipated to have a substantial impact on overall sales.

Main Taget: By introducing the capability to buy digital music online and via in-store kiosks, Tune Source aims to tap into new revenue streams and enhance the overall customer experience. The specified functions for this project are integral to achieving this goal. The ability to search Tune Source's extensive music library online, listen to music samples, establish customer registration accounts, and facilitate the downloading of specific songs are identified as fundamental features for the envisioned online music platform.

Specific functions: Therefore, fundamental features like searching Tune Source's music library, listening to sample music, creating customer registration accounts, and downloading particular songs are required for the online music listening platform.

This calculated action responds to the increasing need for digital music accessibility while also being in line with consumer tastes and market trends. By adding these particular features, Tune Source not only meets the needs of tech-savvy customers who want easy and convenient ways to access and buy their favorite music but also positions itself competitively in the market. As the project develops, it has the potential to increase revenues right away and establish a viable model for long-term expansion and devoted clientele in the ever-changing world of digital music.

b. Expected business value

Offering digital music for purchase is a great way to satisfy the demands of current clients while also drawing in new ones and boosting sales. Customers will have more options from Tune Source's unique assortment, which will increase sales growth potential and reach a wider audience.

Subscription fees from customers are a significant source of funding for the project. Providing a monthly download service not only generates a consistent revenue stream but also fosters enduring customer relationships that lead to a more lucrative business model. This will improve Tune Source's customer interactions in addition to generating additional revenue.

Enhanced cross-selling represents a positive component of the initiative. After downloading one or two tracks, customers can improve their experience by buying the complete CD from Tune Source's website or store. In addition to increasing sales, this raises the average transaction value and opens doors for successful cross-selling. Offering gift cards for music downloads is a unique approach to drawing clients and

creating a new source of income. Not only can gift cards promote music downloads, but they also make thoughtful presents for music lovers. This not only generates income but also raises the perceived value and utility of Tune Source's music download service among consumers.

c. Priorities and Risks:

Risks and priorities are crucial factors to consider when assessing project implementation strategies.

Importance from a strategic standpoint: The marketing division views this system as a key tactic in its quest to stay competitive in the fiercely competitive music industry. This is a critical move to keep Tune Source at the top of customers' minds, in addition to being a chance to increase revenue.

Customer pressure: There is no denying the substantial demand from present clients. These important clients may stop doing business with you if you don't reply right away. This presents a problem for long-term relationships with devoted clients as well as for sales.

Competition: The music download industry is very competitive, necessitating flexibility and agility. It's critical to implement the system as soon as possible to avoid becoming bogged down in the variety of options accessible to customers. Postponements may lead to missed chances, lower competitive performance, and damage Tune Source's standing in the quickly changing digital music industry.

2. The Stakeholders

a. Stakeholder Identification, Roles, and Interests in the Tune Source Case Study

Founders (John Margolis, Phil Cooper, Megan Taylor):

Roles: Business Owners and Founders. As entrepreneurs and business owners, the founders, including John Margolis, Phil Cooper, and Megan Taylor, are pivotal in setting the vision and direction for Tune Source. Their entrepreneurial spirit drives the company's initiatives and strategic decisions.

Interests: Financial success, customer base expansion, and optimal use of Tune Source's unique music archive. The founders are deeply invested in the financial success of Tune Source. Their interest in expanding the customer base reflects a commitment to reaching new audiences. The emphasis on maximizing the utilization of the unique music archive underscores their dedication to offering a distinctive value proposition in the market.

Carly Edwards:

Role: Project Sponsor. Carly Edwards has a crucial position as the project sponsor, which affects the project's overall success. Her responsibility is to guarantee that the project is in line with the goals of the business by offering resources, support, and direction.

Interests : Increasing sales, strategic market positioning, meeting customer demands, and leveraging the music archive for marketing. Carly's primary interests lie in increasing sales, strategically positioning Tune Source in the market, and meeting customer demands. Her focus on leveraging the unique music archive for marketing indicates a keen awareness of the platform's competitive advantage.

Customers:

Role : End-users. Customers, both current and potential, are the end-users shaping the success of Tune Source. Their preferences and behaviors directly impact the platform's design, functionality, and overall appeal.
Interests: Access to diverse digital music, user-friendly experience, subscription options, and specific track purchases. Customers seek access to a diverse range of digital music, emphasizing the importance of content variety. Their interest in a user-friendly experience reflects a desire for seamless navigation, and the demand for subscription options and specific track purchases underscores the need for personalized and flexible offerings.

Marketing Department:
Role: Strategic planning for market competitiveness. The marketing department plays a crucial role in shaping Tune Source's market competitiveness. Through strategic planning, they contribute to defining the platform's positioning and differentiation strategies.
Interests : Utilizing digital music downloads as a strategic asset, meeting customer demands, and enhancing Tune Source's market position. The marketing team is keen on utilizing digital music downloads as a strategic asset, recognizing their potential to attract and retain customers. Their interest in meeting customer demands aligns with ensuring that Tune Source remains relevant and appealing in a competitive landscape.

IT Department:
Role: Maintaining and developing the technology infrastructure. The IT department is responsible for maintaining and developing the technology infrastructure that underpins Tune Source. Their role is essential in ensuring the platform's technical viability and functionality.
Interests : Successful implementation of the digital music download platform, ensuring technical feasibility, and seamless integration with existing systems. The IT team is deeply interested in the successful implementation of the digital music download platform. Their focus on technical feasibility and seamless integration reflects a commitment to delivering a robust and efficient technological solution for Tune Source, aligning technology with business objectives.

b. Review the Requirements Definition for the Tune Source
Search and purchase function:
Stakeholder(s): Customers (Current and Potential).
Requirements: The ability to search for and purchase digital music downloads, including features such as listening to music samples, individual downloads, subscription accounts, and the option to purchase gift cards. This requirement primarily stems from the customers' desire for a seamless and comprehensive music shopping experience.

Strategic System and Market Competitiveness:
Stakeholder(s): Marketing Department.
Requirements: Viewing the system as a strategic asset, emphasizing the critical need for digital music downloads to remain competitive, and maximizing the utilization of the

unique music archive for marketing purposes. These requirements are driven by the marketing department's focus on enhancing Tune Source's market position and utilizing its distinctive offerings.

Sales Growth and Revenue Streams:
Stakeholder(s): Founders and Marketing Department.
Requirements: Increasing sales through digital music downloads, gaining new revenue from customer subscriptions, exploring cross-selling opportunities, and generating revenue from the sale of music download gift cards. The founders are keen on financial success, while the marketing department emphasizes revenue generation and strategic growth.

Quick Implementation and Customer Retention:
Stakeholder(s): Founders.
Requirements: Urgent implementation due to customer demand and the potential loss of current customers if the system is not promptly introduced. These requirements reflect the founders' concern for customer satisfaction and the urgency to meet market expectations.

Technical Feasibility and System Maintenance:
Stakeholder(s): IT Department.
Requirements: Ensuring technical feasibility, swift system implementation, and ongoing maintenance and support for the digital music download platform. These requirements highlight the IT department's responsibility for the seamless integration of technology and the sustained functionality of the platform.

c. General discussion.

The Tune Source project requirements definition is a meticulously planned symphony of many stakeholder inputs, each bringing distinct goals and points of view to the table. End users, or customers, have a big say in what the platform offers and how it works. Intuitive search and purchasing procedures, as well as the availability of gift cards and subscription choices, highlight their demand for a seamless and all-encompassing music discovery experience.

As the forward-thinking leader, the marketing division incorporates strategic components into the design. Its focus on using Tune Source's distinctive music archive for marketing and seeing the system as a strategic asset highlights its dedication to market competitiveness and strategic positioning. Including digital music downloads in your marketing plan shows that you are aware of the preferences of today's consumers and the possibilities for revenue growth.

The project's creators instilled an atmosphere of financial desire through their entrepreneurial spirit. Their commitment to client happiness and retention is demonstrated by the urgency of the quick implementation, which is motivated by customer demand and the fear of losing current customers. The founders of Tune Source also saw it as a successful business endeavor with substantial financial success, in addition to being a music platform.

Similar to behind-the-scenes coordinators, IT departments guarantee the seamless integration and technical viability of the digital music download platform. Their emphasis is on upholding and expanding the technology infrastructure in accordance with the project's long-term sustainability and adaptability, highlighting the significance of well-balanced technology and objectives. company.

Essentially, the smooth cooperation of these parties is what makes Tune Source successful. A strong technological foundation, financial objectives, customer-centric features, and strategic marketing considerations all work together to produce a harmonious whole that not only fulfills but goes beyond client expectations. The world of digital music is competitive and ever-changing. The degree to which these disparate needs are integrated into a coherent and meaningful experience for stakeholders and customers will determine the project's overall success.

3. Requirements in Tune Source

a. Requirements

In the context of software and project management, "requirements" are detailed descriptions of the features, functions, and constraints that a system must meet to meet customer requirements. or user. Requirements can be divided into two main categories: functional requirements and non-functional requirements.

Functional requirements describe specific behaviors that the system must have, while non-functional requirements focus on factors not directly related to the system's functionality such as performance, security, and compatibility. prefer.[1]

Source : Anon, (n.d.). Available at: https://vironit.com/wp-content/uploads/bfi_thumb/2020/07/image8-7ceb58o4k3hfo5z369dagner0zhoxbkny1u0h2y41ao.png.webp.

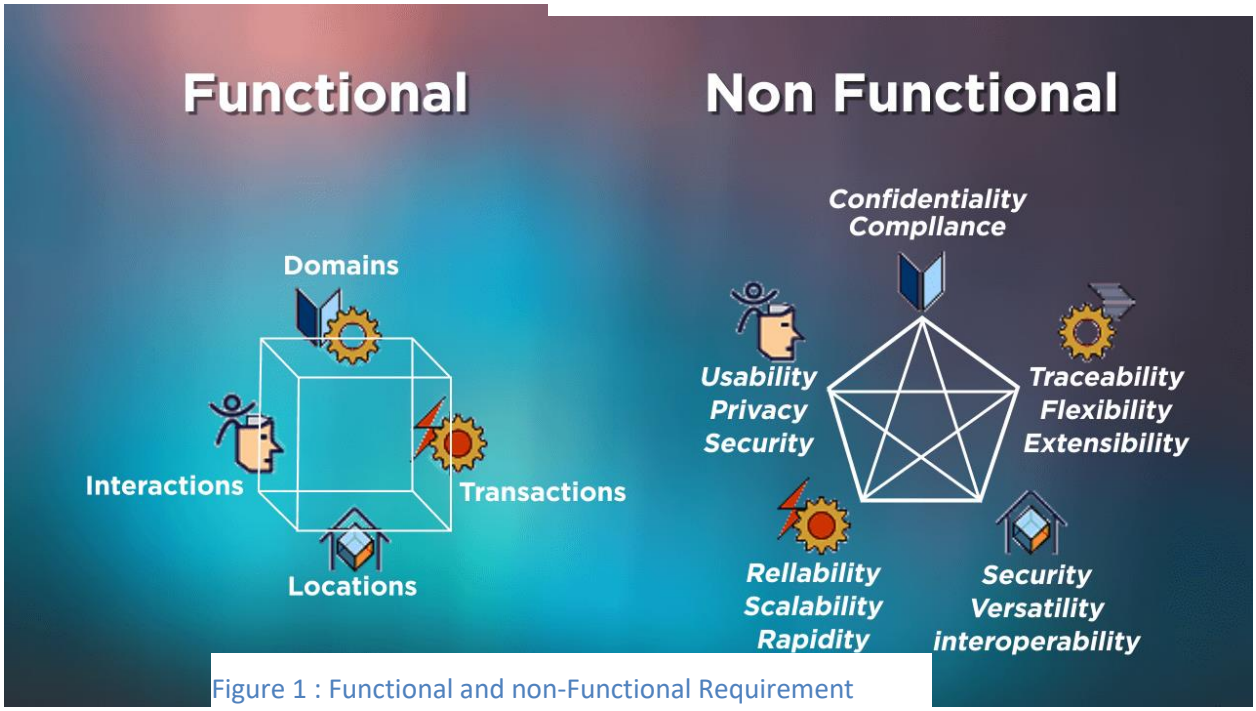

Figure 1 : Functional and non-Functional Requirement

b. Funcional Requirement

Concept:

Functional requirements are an important part of requirements description during software development. These are requirements that accurately describe the specific functions, services, and behaviors that the system needs to perform. These requirements focus on "what" the system needs to do and provide a detailed picture of the features and capabilities that end users will experience.[2]

Functional Requirement in Tune Source:

Tune Source, in its project, sets out a number of functional requirements to build a diverse and attractive music download platform for users. First, the system needed to provide flexible search capabilities within Tune Source's exceptional music collection, making it easy for users to identify and shop for desired tracks. Next, the feature of listening to music samples before buying is important to enhance the user experience, helping them make accurate shopping decisions. The system also needs to support the specific music purchasing process, allowing users to select and purchase their favorite music at a fixed price. As for subscription capabilities, the system will provide the feature of setting up a subscription account, allowing users to enjoy unlimited music downloads for a monthly fee. Finally, the integrated gift card purchase feature will increase the value and convenience for users when using the music download service.

Specifically about the role of functional requirements in Tune Source:

In the Tune Source project, functional requirements are the specific features and behaviors that the system must perform to meet business needs and user expectations. A number of specific functions can be listed. in Tune Source :

Search in Music Collection:

Flexible and precise search capabilities across Tune Source's music collection.

Listen to Music Samples Before Buying:

Provide a preview feature so users can evaluate the quality of the music before deciding to buy.

Buy Specific Tracks:

Allows users to purchase specific tracks at a fixed price per download.

Setting up a Registered Account:

Provides account registration feature so users can enjoy unlimited music downloads for a monthly fee.

Gift Card Purchase Integration:

Integrate the ability to buy gift cards to use for music download services, as gifts for others.

Functional requirements for the project Tune Source is primarily responsible for building a diverse and attractive music download platform for users. First, they play an important role in defining the specific scope of functionality of the system, including features such

as searching the collection, listening before buying, and managing user accounts.Not only do they help build a positive user experience, but they also add value to the service by offering the ability to purchase gift cards, purchase specific tracks at fixed prices, and more. These requirements not only meet customer expectations but also support business strategy, generate subscription revenue, and drive sales through flexible account management.More importantly, the functionality requirements in Tune Source are shaped to meet user needs and desires, from finding special music to the ability to listen before deciding to buy. They help create flexibility for users, allowing them to enjoy and manage their music entertainment experience in a personal and comfortable way. Therefore, functional requirements are not just a technical description but also a decisive factor in building a unique and complete music download platform.

Functional requirements were essential in developing a distinctive and adaptable music download platform during project implementation. Pre-purchase music samples and adaptable search features enhance the user experience and help users make precise and well-informed purchases. Users may easily and conveniently manage their music entertainment demands with the flexibility and simplicity of purchasing individual tracks and creating a subscription account. The ability to purchase integrated gift cards adds value to the service while also simplifying and enhancing the present-giving experience.

c. Non-Function Requirement

Concept: Non-functional requirements are requirements that do not focus on the specific functionality of the system, but instead, they focus on the comprehensive properties of the system. These requirements include aspects such as performance, security, scalability, and compatibility to ensure that the system not only works properly but also meets quality and performance criteria.[2]

Non-Functional Requirement in Tune Source:

To guarantee the system runs efficiently and safely, Tune Source also establishes a number of non-functional requirements in addition to functional ones. In order to maintain customer satisfaction, the system must first ensure that it can efficiently handle the anticipated volume of users and have a fast response time. The system must strictly protect consumers' personal information and transactions, as security is of utmost importance. A crucial prerequisite for handling future growth without compromising performance is scalability. Finally, to guarantee that customers can effortlessly access the service from anywhere, compatibility with a variety of devices and browsers is crucial.

Specifically about the role of non-functional requirements in Tune Source:

While functional requirements focus on specific features, non-functional requirements focus on aspects not directly related to the functionality of the system such as performance, security, and compatibility. We have We can list a number of specific non-functions such as:

Efficiency:
The system must operate efficiently with the expected number of users and ensure quick response times.
Security:
Ensuring the security of customers' personal information and transactions, including login management and data protection.
Ability of extension:
The system needs to be scalable to cope with future growth without sacrificing performance.
Compatible:
Ensure compatibility with many different devices and browsers to optimize user convenience.

In the Tune Source project, non-functional requirements are crucial to guaranteeing that the system satisfies quality and safety standards and functions well. The primary objective of performance criteria is to guarantee that the system can efficiently manage the anticipated volume of users and react promptly, thereby enhancing the user experience.
One of the most crucial prerequisites is security, which guarantees the protection of consumers' private data and online transactions. For consumers to feel trusted and to reduce the danger of data theft, security measures must be firmly integrated.
Scalability is a crucial technical need that makes the system more adaptable to growth and enables it to react to user requests more rapidly. By guaranteeing that the system is scalable to address future difficulties, it also supports corporate strategy.
Another crucial prerequisite is cross-platform compatibility, which guarantees that the Tune Source system can function on a range of gadgets and browsers. This broadens the service's user base while simultaneously improving user convenience.
To put it succinctly, Tune Source's non-functional requirements are crucial standards that guarantee the system not only functions as intended but also satisfies demanding performance requirements. efficiency, security, and adaptability.

Non-functional needs are crucial to maintaining the Tune Source system's general functionality and security. The finest user experience is necessary in the music entertainment industry, where smooth performance is essential to customer happiness. In order to reduce the possibility of information theft and foster user trust, security is of utmost importance. While cross-platform compatibility is essential to guaranteeing user convenience across all platforms and devices, scalability sets the system up for growth. The development of a music download platform that not only satisfies needs but also guarantees quality and safety is supported by all of these non-functional requirements.

d. Relationship between functional requirements and non-functional requirements in Tune Source

The Tune Source project's interaction between functional and non-functional objectives is essential to creating a distinctive music download platform that satisfies user expectations while guaranteeing performance and security. elevated. The system's functional requirements are centered on what it must be able to accomplish, such as browse collections, listen to music before buying it, and buy individual tracks. Through the purchase of gift cards and the establishment of subscription accounts, these requirements not only specify the extent of capability but also influence the user experience and raise the value of the service.

Non-functional needs, however, require significant changes in order to reach perfection. Performance is crucial for maintaining a good user experience, since it guarantees that the system can manage the anticipated volume of users with ease and react fast. In order to reduce the possibility of information theft, foster user confidence, and safeguard Tune Source's reputation, security comes first. Scalability guarantees that system growth can be accommodated without compromising performance and quality. Compatibility across platforms is a crucial component in guaranteeing consumer comfort when utilizing various gadgets.

A careful balance between the user experience and significant technical aspects like performance and security can be seen in the interplay between functional and non-functional criteria in Tune Source. Ensuring that the system functions smoothly, safely, and flexibly in all respects is just as important as offering appealing functionalities. This partnership helped to design the initiative and guaranteed that Tune Source would offer a whole range of music-related entertainment options rather than just a place to download songs.

4. Discuss the technique(s) you would use to obtain the requirements.

a. Joint Application Development (JAD)

Concept :

Joint Application Development (JAD) is a method of organizing group sessions involving stakeholders to pose, analyze, and define requirements for a project. The team typically includes decision makers and project influencers.[3]

MANAGEMENT

Source: Newline.tech. (2023)

Figure 2 :Jad

Discuss :

The JAD (Joint Application Development) method offers many benefits and has specific disadvantages. One of JAD's notable strengths is its ability to form focus groups, where it directs the attention of stakeholders to a specific aspect of the project, helping to quickly reach consensus. Sharing information directly during group work sessions also plays an important role in minimizing misunderstandings and increasing transparency.

However, JAD also has its disadvantages. One of the major challenges is the requirement of significant time from the parties involved, especially when thorough preparation is required. Additionally, JAD performance depends largely on the instructor's hosting ability, which can create fluctuations in meeting outcomes.

JAD is often applied in situations where there are multiple stakeholders with diverse perspectives on the project and when consensus and commitment from all stakeholders are desired. This method is commonly used in software development, business process improvement, and information technology projects where focus and alignment from stakeholders are important.

b. Interview

Concept :

Interviewing is the process of meeting and talking with stakeholders to better understand their needs, desires, and perspectives on the project.

Source: Newline.tech. (2023)

Figure 3 : Interviews

Discuss :

In addition to its many advantages, the interview method has many drawbacks. One of the main advantages of interviews is their capacity to yield comprehensive and in-depth information, enabling the formulation of thorough questions and the acquisition of perceptive responses from relevant parties. It fosters a more intimate, conversational atmosphere that promotes openness and honesty in the sharing of information.

But it's also important to take this method's drawbacks into account. Interviews can take a lot of time, particularly when speaking with several stakeholders. Another difficulty is the possibility of miscommunication, which arises particularly when inquiries are not made correctly or when the person involved is not entirely informed about the topic under discussion.

When it's important to know each stakeholder's individual perspective about the project, conducting interviews is frequently a good alternative. This is especially true when you want to test opinions using assumptions and optional questions. In situations where face-to-face communication and thorough comprehension from all stakeholders are crucial, such as consulting projects, market research, and product development processes, this approach is frequently employed.

c. Observation

Concept : Observation is the process of tracking and recording user behavior, interactions, and activities in a real-life environment.

Source: Timviec365.vn. (2017).

Figure 4 : Observation

Discuss:

The observational method has many drawbacks in addition to its many benefits. Observation's capacity to comprehend real-world behavior—in which users' interactions with the system or existing environment are directly observed—is one of its main advantages. This technique is especially helpful for identifying issues about which people might not be able or ready to communicate orally.

Nevertheless, there are drawbacks to observation as well. It is not universally applicable to all situations and is restricted to particular surroundings. Observation also necessitates technique and analytical skills in order to comprehend the data gathered.

When faced with issues that consumers might not be aware of or when trying to learn how people behave naturally in real-world settings, observation becomes crucial. In the domains of UX (user experience) research, product testing, and project management,

where an accurate evaluation of user behavior is crucial, observation is frequently utilized.

d. Business Analytics (BPA)

Concept : Business Process Analysis (BPA) is the process of evaluating, understanding, and optimizing business processes to ensure that they operate efficiently and meet business needs.[4]

Source: SlimCRM Resources (2022).

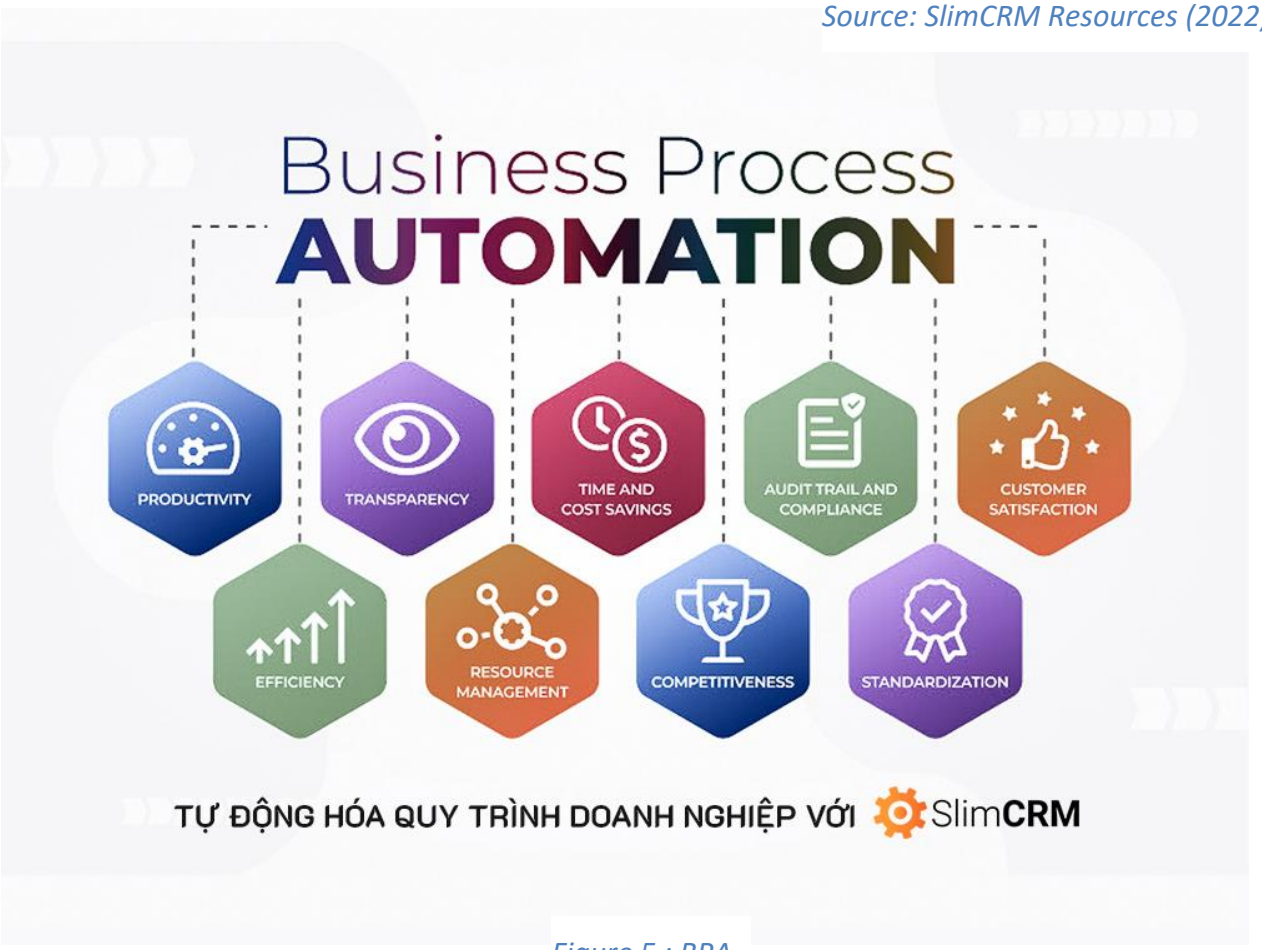

Figure 5 : BPA

Discuss:

Business Process Analysis (BPA) has certain drawbacks in addition to its many benefits. One of BPA's main advantages is its capacity to maximize business process performance, which aids in locating chances to improve an organization's performance. This method offers a thorough understanding of organizational processes by offering a view of how business activities interact with systems and with each other.

BPA does, however, have several drawbacks. demands a thorough understanding of technological platforms and business processes, which can be difficult for companies

lacking the necessary resources or expertise. Additionally, the BPA procedure can be time-consuming, particularly for large-scale companies.

When business processes need to be optimized for efficiency and quick reaction to changes in the business environment, business process architecture (BPA) is frequently used. It's also a good option if you want to learn more about how various company components interact with one another. BPA is frequently utilized in situations where a thorough understanding of the process is crucial, such as business process improvement, cost optimization, and the deployment of new systems.

e. Business Process Analysis (BPI)

Concept : Business Process Improvement (BPI) is the process of changing and improving business processes to optimize performance and better respond to business

Source : Shutterstock. (n.d.).

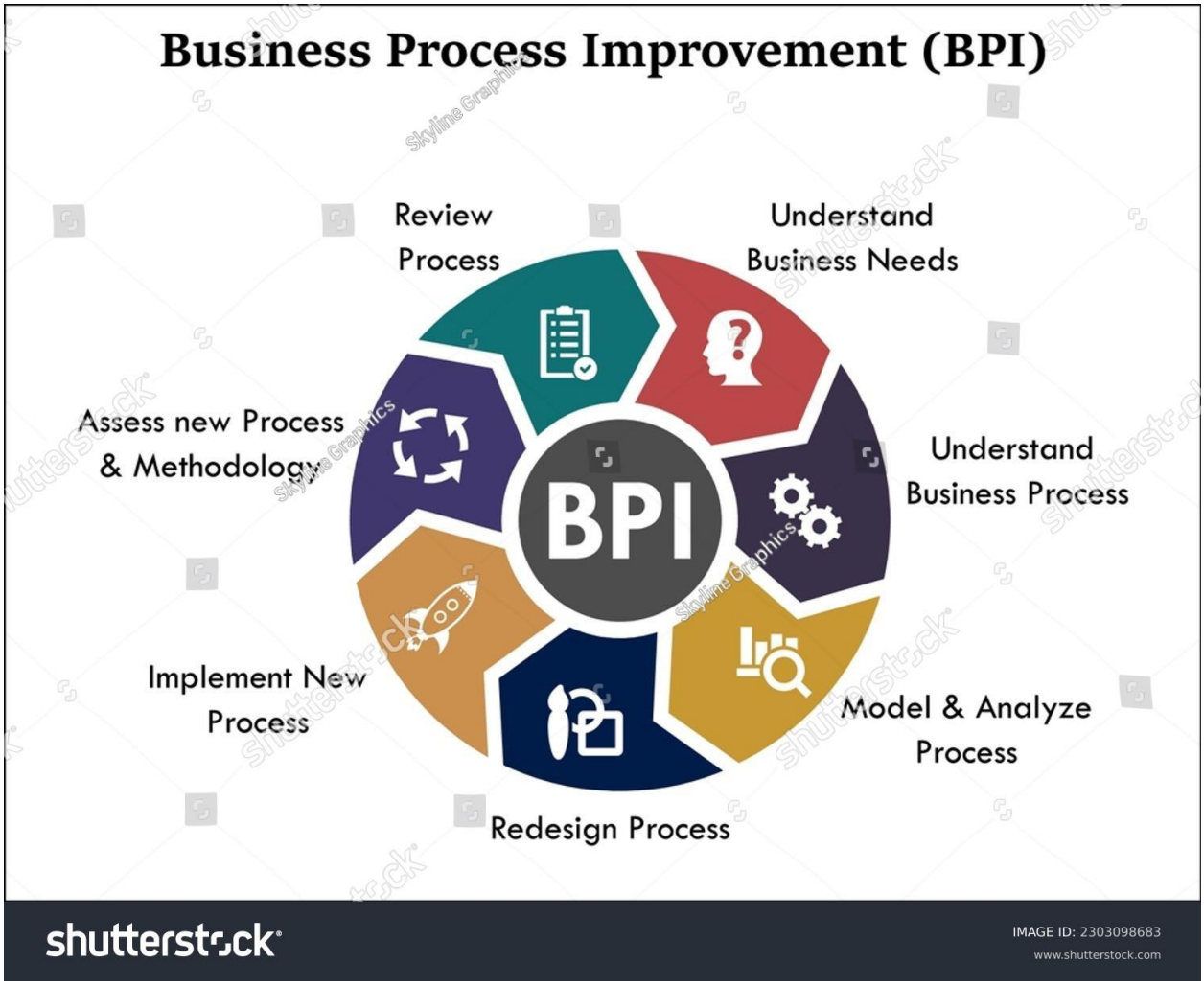

Figure 6 : BPI

requirements.[5]

Discuss :

There are certain drawbacks to the Business Process Improvement (BPI) methodology in addition to its many benefits. Enhancing performance by emphasizing change to improve performance and increase business responsiveness is one of BPI's main advantages. Additionally, it offers flexibility, enabling the company to quickly adjust to changes and maintain its dynamic nature while being open to new prospects.

BPI does, however, have several drawbacks. needs staff and stakeholders' complete support in order to succeed. BPI can also improve the job that employees do, particularly when they need to adjust to new work methods.

BPI is frequently used when significant adjustments to a company's operational procedures are required. If you want to make sure that modifications are applied smoothly and successfully, this is the best option. When an emphasis on continuous improvement and quick adaptation to a changing business environment is required, BPI is frequently employed in process improvement, business reorganization, and digital transformation initiatives.

f. Business Process Reengineering (BPR)

Concept : Business Process Reengineering (BPR) is a strategic approach to improving business performance and efficiency through redesigning and reorganizing business processes.

Discuss :

The business restructuring (BPR) approach has many drawbacks in addition to its many benefits. One of BPR's key advantages is its capacity to modify organizational structures. This is frequently done in conjunction with an organization-wide redesign, which serves to improve processes and organizational structure alike. Additionally, it is made to undergo significant adjustments swiftly, enabling businesses to promptly adjust to a shifting business environment.

BPR does, however, come with some drawbacks. Employees run the danger of having more job pressure and can find it harder to adjust to changes. demands a large investment of time and money from stakeholders in order to be successful.

BPR is frequently used when a company needs thorough innovation in both organizational structure and business operations. When a significant change must be made in reaction to a corporate environment that is changing quickly, this is the best option. BPR is frequently used in initiatives involving extensive reorganization, digital transformation, and strategic change, where quick and thorough change is crucial.

g. Demonstrate how to gather requirements based on the chosen technique.

Joint Application Development (JAD):

Reasons to choose : For Tune Source's project, JAD is the ideal choice. With the need to rapidly deploy new systems and the diversity of stakeholders such as founders, marketing, and IT, JAD helps focus attention and buy-in in a collaborative environment. group work school.

How to get requests in Tune Source :
Host a JAD session with the founder, Carly Edwards, and representatives from the IT department.
During the JAD session, the focus was on setting and defining requirements for implementing a new digital music download system.
Use group techniques to analyze functional and non-functional requirements, setting performance, security, and system integration criteria.
Organize discussion sessions to address special aspects of the project such as brand promotion through a music download system.

Observation::
Reasons to choose : With Tune Source, observation is the logical choice. By monitoring how users interact with the current website, we can identify problems they may be experiencing and recommend specific improvements to the new system.
How to get requests in Tune Source : Observe different aspects of the current website, from the search process to the checkout process.
Take note of positive aspects and areas for improvement.
Conduct a meeting to discuss with the development team and set specific requirements based on these observations.

Business Analytics (BPA) :
Reasons to choose : With Tune Source, applying BPA is necessary to ensure that business processes related to music sales and downloads occur efficiently. This helps optimize the customer experience and at the same time respond quickly to changing market needs.
How to Collect Requirements:
Conduct a detailed analysis of current processes related to music sales and downloads.
Identify the strengths and weaknesses in these processes.
Focus on improvements that can be made to optimize performance and respond quickly to new market demands.
Hold brainstorming sessions with stakeholders to ensure that technical requirements are in sync with proposed business improvements.
Business Process Analysis (BPI) :
Reasons to Choose:
For Tune Source's project, BPI is vital to making improvements in business processes and ensuring that the business can respond flexibly to changing market demands.

How to Collect Requirements:
Conduct a detailed review of current business processes and identify areas for improvement.

Discuss with stakeholders to identify specific expectations and requirements for process improvements.
Develop change scenarios and ensure buy-in from stakeholders.
Organize brainstorming sessions and work sessions to ensure that the technical requirements reflect the proposed business improvements.

Business Process Reengineering (BPR):
Reasons to Choose:
For Tune Source, BPR is important to ensure a significant transformation in the way businesses conduct business processes and organizational structures, responding flexibly to the changes taking place in the music market .
How to Collect Requirements:
Conduct a detailed assessment of the entire business infrastructure and identify current process dead spots.
Identify opportunities and challenges during process redesign and reorganization.
Organize workgroups and brainstorming sessions to ensure that the technical requirements properly reflect the proposed innovation.
Dialogue with key stakeholders to ensure commitment and proper understanding of proposed changes.

III. (P6) Analyze requirements using a combination of structural and behavioral modeling techniques

1. Use Case Diagram for the whole system

Use case diagrams illustrate the interaction between users and the system. It provides a high-level view of the system's functions from an external perspective.
In the context of Tune Source, Use Case Diagrams capture the key actors (entities that interact with the system) and use cases (the functionality the system provides).

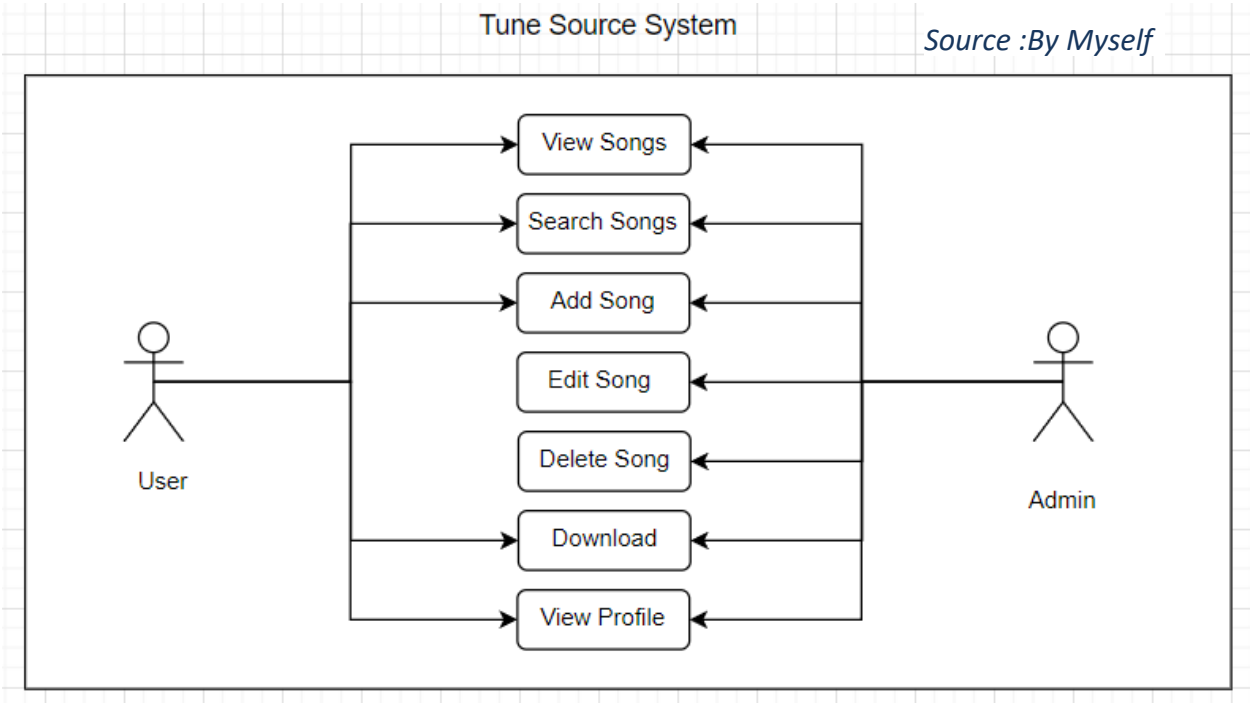

Figure 7 : Use case diagram

a. Actors anh use case

Actors :

Customer: Represents individuals who interact with the system to search, listen, and purchase music.

Admin: Represents Tune Source administrators who manage the digital music archive and the overall system.

Use Case :

Music Search: This use case involves customers searching for specific digital music tracks in the Tune Source archive.

Listen to samples: Customers can listen to sample music before making a purchasing decision.

Music Purchase: The process by which customers purchase digital music tracks.

Subscription management: For customers who choose a subscription model, this use case allows them to manage their subscription settings.

Redeem gift cards: Customers can redeem gift cards to access digital music content.

Music creation: For special customers, they can upload their own music as a collaborator. For admins, they can create new music based on their tastes.

Update Music: For Admin, they can update the information of a song

Delete Music: For songs with few listeners or that do not meet expectations or errors during operation, the admin can delete them.

The Music Search application for Tune Source offers a comprehensive set of use cases catering to a seamless and enjoyable user experience. The primary use case involves customers searching for specific digital music tracks within the Tune Source archive. This feature ensures that users can easily locate and access the music they desire, enhancing overall satisfaction.

Additionally, the "Listen to samples" use case provides customers with the opportunity to preview music before making a purchasing decision. This feature adds value by allowing users to gauge the suitability of a track to their preferences, ultimately contributing to informed purchasing choices.

The "Music Purchase" use case is central to the platform's functionality, facilitating the smooth process by which customers can buy digital music tracks. This streamlined approach ensures a user-friendly and efficient transaction process.

For customers opting for a subscription model, the "Subscription Management" use case is crucial. It enables users to have control over their subscription settings, offering flexibility and customization according to their preferences.

The "Redeem Gift Cards" use case introduces a promotional element, allowing customers to redeem gift cards for access to digital music content. This feature not only promotes customer loyalty but also serves as an effective marketing tool.

The "Music Creation" use case stands out as a unique feature, providing a platform for special customers to upload their own music as collaborators. Simultaneously, administrators have the ability to create new music based on their tastes, fostering a collaborative and creative community within Tune Source.

Administrators also possess the authority to "Update Music" by modifying song information, ensuring accurate and up-to-date content on the platform. Additionally, the "Delete Music" use case allows administrators to remove songs with few listeners, those not meeting expectations, or those encountering operational errors. This feature maintains the quality and relevance of the Tune Source music library.

b. Relationships, boundaries and goals

To understand how to interact with the system, divide it into objects (actors) and particular use cases in the use case diagram. In this scenario, all use cases—such as finding music online, purchasing music, managing subscriptions, and redeeming gift cards—are applicable to the "Customer" object. This illustrates how the user interacts with the system in real time. The "Administrator" object, on the other hand, is mostly focused on use cases pertaining to general system functions and managing digital music stores.

The framework that envelops all objects and use cases serves as a representation of the system's boundary, demarcating what is inside the system and what is outside of it.

Use case icons are meant to give a broad overview of the ways in which objects engage with the system and the roles they play. It aids in identifying important tasks and offers a framework for creating more particular system-specific needs and functions.

To understand how "Customers" use Tune Source's capabilities, it's vital to grasp the relationship between use cases and "Customers." The fact that this interaction is so common could mean that users can benefit from a variety of features, such as music search and subscription management. On the other hand, it is crucial that the "Administrator" object stays out of the way of other facets of customer communication and concentrates mainly on content management and system functionality.

This model is a foundational tool, a step towards more specialized tools for requirements analysis and identification. It aids in precisely defining the functions and duties of every element in the system, laying the groundwork for the development of programs and features that satisfy user requirements and Tune Source's corporate objectives.

c. Use case specification

Use case specification for "Music Search":

Use Case Name: Search for Music

Actors: Customer,Admin

Description: This use case allows customers to search for specific digital music tracks within the Tune Source archive.

Preconditions:

The customer is logged into the Tune Source system.

The customer has access to the search functionality.

Basic Flow:

The customer accesses the "Search for Music" feature from the main menu.

The system presents a search interface allowing the customer to input search criteria.

The customer enters keywords, genres, or artist names in the search bar.

The system performs a real-time search within the music archive based on the entered criteria.

The system displays a list of matching music tracks.

The customer selects a specific track for more details.

Alternative Flow:

If no matching tracks are found, the system provides a clear message indicating the absence of results.

Postconditions:

The customer can view details of the searched music track.

The customer can proceed to listen to a sample or purchase the selected music.

Use Case Priority: High

Use Case Specification for "Purchase Music(Download Funciton)":

Use Case Name: Purchase Music

Actors: Customer ,Admin

Description: This use case enables customers to buy specific digital music tracks from the Tune Source archive.

Preconditions:

The customer is logged into the Tune Source system.

The customer has successfully searched for and selected a music track.

Basic Flow:

The customer selects the "Purchase" option for a chosen music track.

The system displays the price and prompts the customer to confirm the purchase.

The customer confirms the purchase.

The system deducts the appropriate amount from the customer's account.

The system provides a download link for the purchased music track.

Alternative Flow:

If the purchase confirmation is not successful, the system provides an error message.

Postconditions:

The purchased music track is added to the customer's library for download.

The customer receives a confirmation email with the purchase details.

Use Case Priority: High

Use Case Specification for "View Song":

Use Case Name: View Song

Actors: Customer, Admin

Description: This use case allows users, both customers, and administrators, to view details of a specific music track.

Preconditions:

The user is logged into the Tune Source system.

The user has successfully searched for and selected a music track.

Basic Flow:

The user selects the "View Song" option for a chosen music track.

The system displays detailed information about the selected music track, including artist, genre, and release date.

Alternative Flow:

If the detailed information is not available, the system provides a message indicating the absence of data.

Postconditions:

The user can review the details of the selected music track.

Use Case Priority: High

Use Case Specification for "Add Song":

Use Case Name: Add Song

Actors: Customer,Admin

Description: This use case enables administrators to add a new music track to the Tune Source archive.

Preconditions:

Administrator or Customer has logged in to the Tune Source system.

Admin or Customer has the necessary permissions to add songs.

Basic Flow:

The admin or Customer selects the "Add Song" option from the admin dashboard.

The system prompts the admin to enter details of the new music track, including title, artist, genre, and release date.

The admin or Customer submits the information.

The system adds the new music track to the Tune Source archive.

Alternative Flow:

If there is an error in submitting the information, the system provides an error message.

Postconditions:

The new music track is successfully added to the Tune Source archive.

Use Case Priority: Medium

Use Case Specification for "Delete Song":

Use Case Name: Delete Song

Actors: Admin

Description: This use case allows administrators to remove a music track from the Tune Source archive.

Preconditions:

The admin is logged into the Tune Source system.

The admin has the necessary permissions to delete songs.

Basic Flow:

The admin selects the "Delete Song" option from the admin dashboard.

The system presents a list of existing music tracks for the admin to choose from.

The admin selects the specific music track to be deleted.

The system prompts the admin to confirm the deletion.

The admin confirms the deletion.

Alternative Flow:

If the admin decides not to delete the song, the process is aborted.

Postconditions:

The selected music track is removed from the Tune Source archive.

Use Case Priority: High

Use Case Specification for "Update Song":

Use Case Name: Update Song

Actors: Admin

Description: This use case enables administrators to modify details of an existing music track in the Tune Source archive.

Preconditions:

The admin is logged into the Tune Source system.

The admin has the necessary permissions to update songs.

Basic Flow:

The admin selects the "Update Song" option from the admin dashboard.

The system presents a list of existing music tracks for the admin to choose from.

The admin selects the specific music track to be updated.

The system displays the current details of the selected music track.

The admin modifies the necessary information (e.g., title, artist, genre).

The admin submits the updated information.

Alternative Flow:

If there is an error in submitting the updated information, the system provides an error message.

Postconditions:

The selected music track is successfully updated in the Tune Source archive.

Use Case Priority: High

Use Case Specification for "View Profile":

Use Case Name: View Profile

Actors: Customer, Admin

Description: This use case allows users, both customers and administrators, to view their user profile.

Preconditions:

The user is logged into the Tune Source system.

Basic Flow:

The user selects the "View Profile" option from the user dashboard.

The system displays detailed information about the user's profile, including username, email, and subscription details.

Alternative Flow:

If the detailed information is not available, the system provides a message indicating the absence of data.

Postconditions:

The user can review the details of their user profile.

Use Case Priority: Low (for Admin), Medium (for Customer)

The Use Case specification above provides a detailed look at how specific functions in the Tune Source system are implemented and interacted with users. Overall, these Use Cases are designed to optimize the user experience. user and administration experience, while ensuring flexibility and efficiency in managing Tune Source's unique music content. This combination helps the system fully meet the diverse needs of both customers and administrators.

2. Context Diagram

Source :By Myself

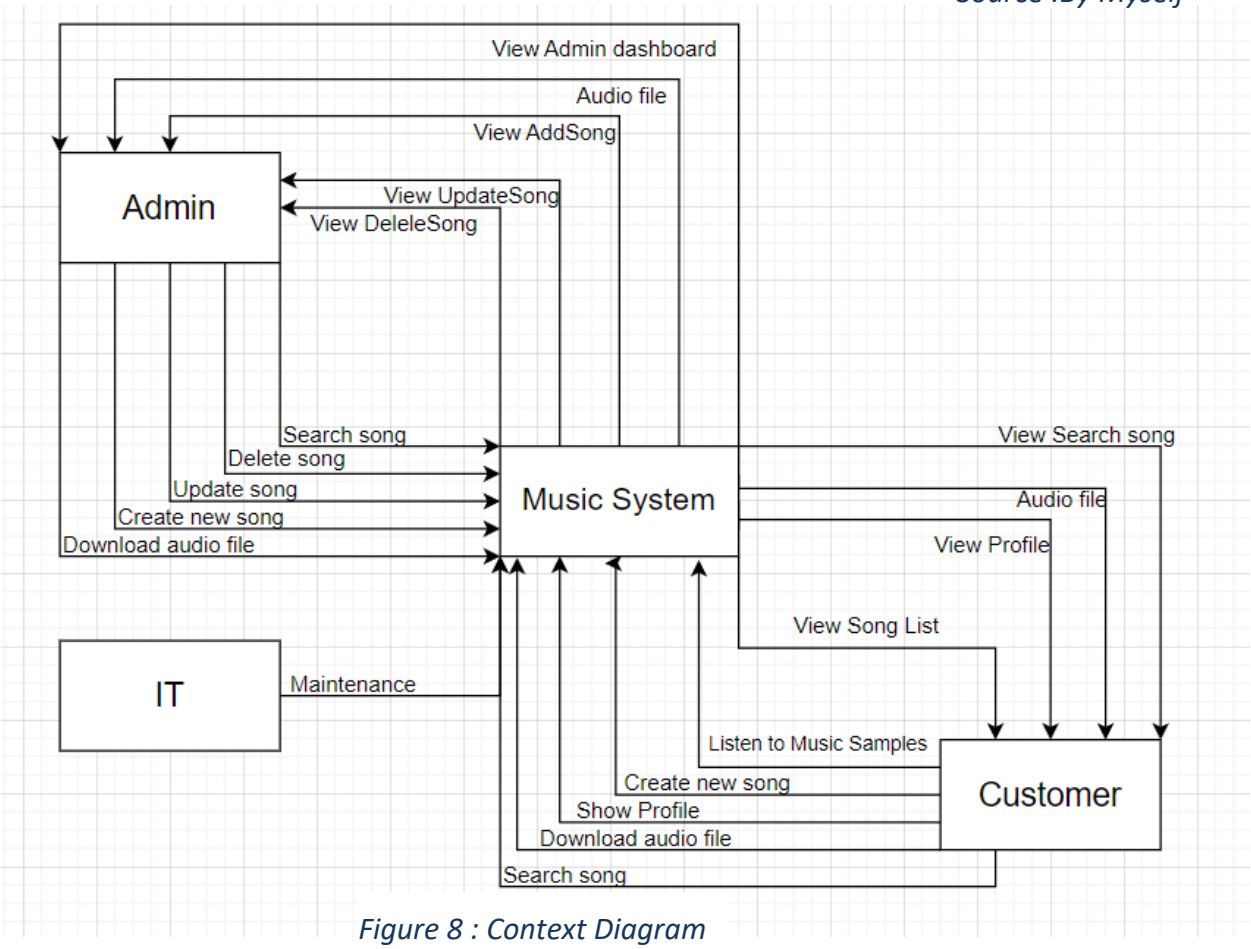

Figure 8 : Context Diagram

The Tune Source system Context Diagram is designed to show the relationship between the system and the main (external) entities involved.

Description of Categories:

Web Users: Represents all end users, including customers and administrators, participating in the system to take advantage of music downloading and content management services.

Website System: Tune Source's website interface, where users can access and use functions to download music, view information and perform shopping transactions.

Unified Kiosk System: The interface represents kiosks located at Tune Source stores, where customers can experience and shop for music.

Relationship:

Users and Website System: Users can access the information system via a web browser, taking advantage of online functions such as searching, testing and shopping.

Users and dominant Kiosk system: Customers can interact with the information system via in-store kiosk, providing a practical and convenient shopping experience.

Context Diagrams provide an overview of how the system interacts with the user and its surroundings. It is a useful tool to help better understand key touchpoints between systems and users, including online and in-store shopping experiences. This helps Tune Source optimize the user experience and fully meet the different needs of users.

3. **Data Flow Diagram – Level 0 for the whole system.**

Source :By Myself

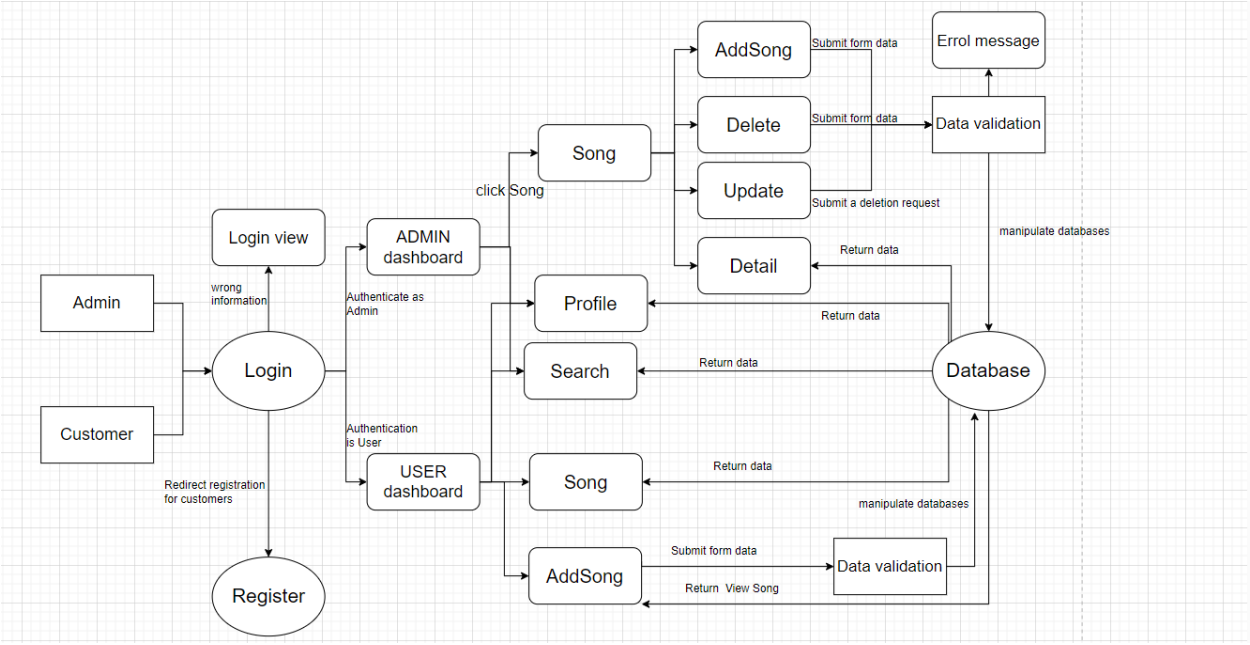

Figure 9 :Data Flow Diagram

The level 0 Data Flow Diagram (DFD) for the Tune Source system describes the main processes of the system and how data is transformed and delivered to various components.

Process (Main Process):

The main process in the system plays an important role, representing core activities such as searching for information, processing shopping transactions and managing content

information. It is the backbone of the system, ensuring smoothness and efficiency in every interaction.

External Entities:

End User:

Users, including customers and administrators, play a decisive role in interacting with the system. They are the ones who ultimately make purchasing decisions and manage information, creating diversity and importance for the system.

Website System:

The website system is where all online users can connect and interact with the system. It is not only a visual interface, but also the main portal where people can access information, make transactions, and enjoy the services the system provides.

Kiosk System:

Kiosk systems represent points in stores where customers can use the same services as on the website. These kiosks provide convenience and a hands-on interactive experience when customers want to take advantage of the service the system offers without going online.

In this infrastructure, core processes combine with peripheral entities to create a comprehensive system that is flexible and easy to use for everyone, from customers to administrators, both online and offline. routes and at kiosk locations.

Main Data Stream:

In the Tune Source system, there are three main data streams that play an important role in providing a comprehensive user experience.

Search and View Information:

Users have the ability to search and view information about music through an online interface or at physical stores. Data from this process, including search information and music details, is passed to main processing to determine the results. This optimizes search capabilities and provides detailed information about available tracks.

Shopping Transactions:

Users can make shopping transactions online or in stores. Data related to order and payment information is passed to main processing for validation and database updates. This process ensures that every shopping transaction goes smoothly and satisfies the user properly.

Content Information Management:

Administrators and users can manage music information, including adding, deleting, and updating details. Content information management data is passed to main processing to update the database. This process ensures that the database always contains accurate and up-to-date information about music content.

The level 0 DFD diagram is not only a process description tool but also a basis for understanding the key interactions and relationships between components in a Tune Source system. This not only supports the development process but also helps manage the system more effectively. This is a result of the flexibility and integration of key data streams, creating a system that is responsive and easy to manage, while providing a unique user experience.

4. ERD for the whole system

Source :By Myself

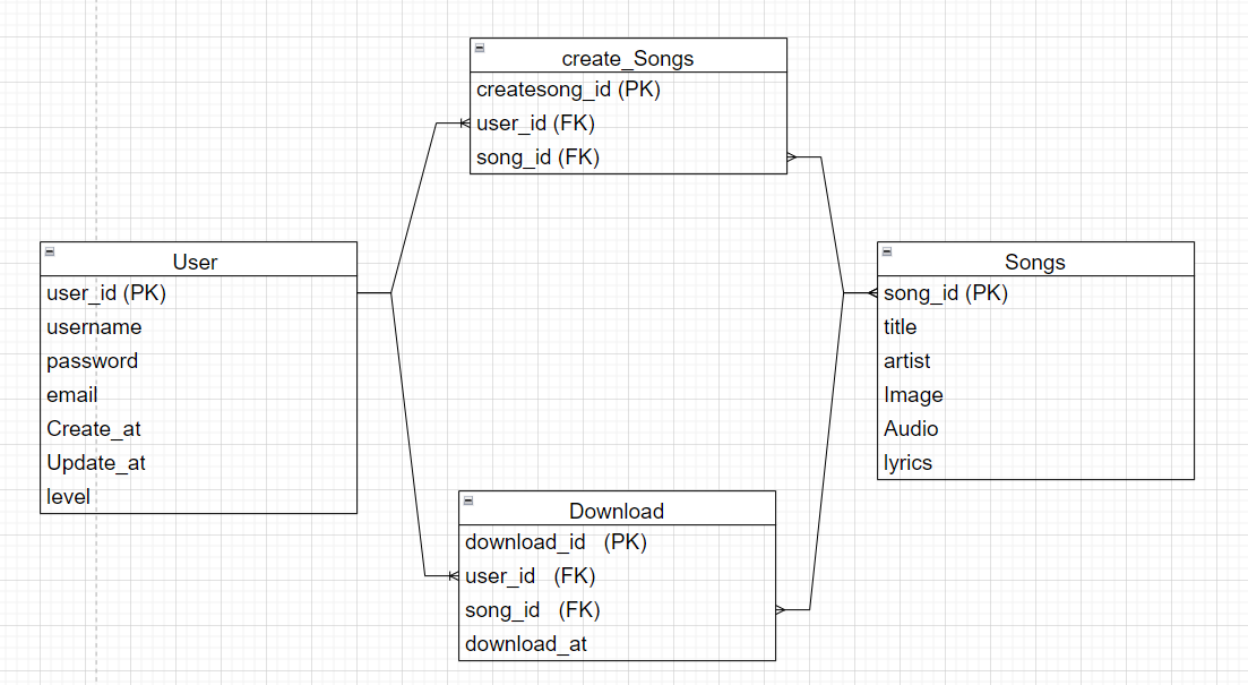

Figure 10 : ERD

The Entity-Relationship Diagram (ERD) diagram for the Tune Source system is designed to depict key entities and the relationships between them. This is a data modeling tool used to express data structures and relationships between entities in an information system or database.

Main Entity:

End User: Represents all end users, including customers and administrators.

Music (Song): Represents music tracks in the Tune Source archive, including information such as name, artist, genre, and year of release.

Download History : Represents information about shopping transactions, including details such as order number, order date, and total value.

User Account: Represents the user's login information and personal details, including login name, password, and contact information.
Create History: Represents information about products created by users.

Users and Downloads:
A one-to-many relationship with each user can have many orders, but each order belongs to only one user.
Users and User Accounts:
A one-to-one relationship with each user having a unique user account, and each user account being associated with only one user.
Downloads and Music:
Many-to-many relationship, because each order can contain many tracks and each track can appear in multiple orders.
CreateSong and Song :
Many-to-many relationship, because each CreateSong can contain multiple tracks and each track can appear in multiple CreateSongs.
CreateSong and users :
One-to-many relationships with each user can have many CreeteSongs, but each CreateSong belongs to only one user.

The ERD diagram helps understand how key entities in the Tune Source system interact with each other. The relationships between users, orders, user accounts and tracks are clearly specified, helping to build an effective database. It provides an overview of how data is organized and linked in the system, serving as a basis for deploying and managing data in an organized manner.

IV. (P7) Discuss how the user and software requirements are addressed in the design phase.

1. Web design

a. Mock-up and Wireframe

Wireframe: This is a simple icon, without color graphics, typically created to depict the structure and layout of a website or application. Wireframes help focus on showing the placement and distribution of elements, such as squares, buttons, and text areas, without including detailed images or design colors. It focuses on structure and arrangement.

Mock-up (Virtual Model): An original version of the final look of the application or website, with colors, images and detailed design. Mockups are typically created to show how a user interacts with an interface, with the primary goal being the final display experience.

How to Apply to Tune Source Project:

Wireframe in Tune Source Project:

Creating a wireframe will help define the structure of your Tune Source website or application from a layout and arrangement perspective.
Determine the position and size of important elements such as navigation bars, search boxes, playlists, and shopping buttons.
Pay attention to the flow of data, for example, from the search page to the product detail page.
Mock-up in Project Tune Source:
Use mock-ups to show how the interface will look once fully designed with colors and images.
Shows the user experience, including how they will interact with elements like shopping buttons, navigation bars, and search boxes.
Integrate input from stakeholders, including end users and administrators, to ensure mock-ups reflect requirements and expectations.
Using both wireframes and mock-ups in the Tune Source project helps build a solid base for interface design, while also helping to ensure that the structure and user experience are laid out in a rigorous and consistent manner. effective.

b. Design

Register :

Source :By Myself

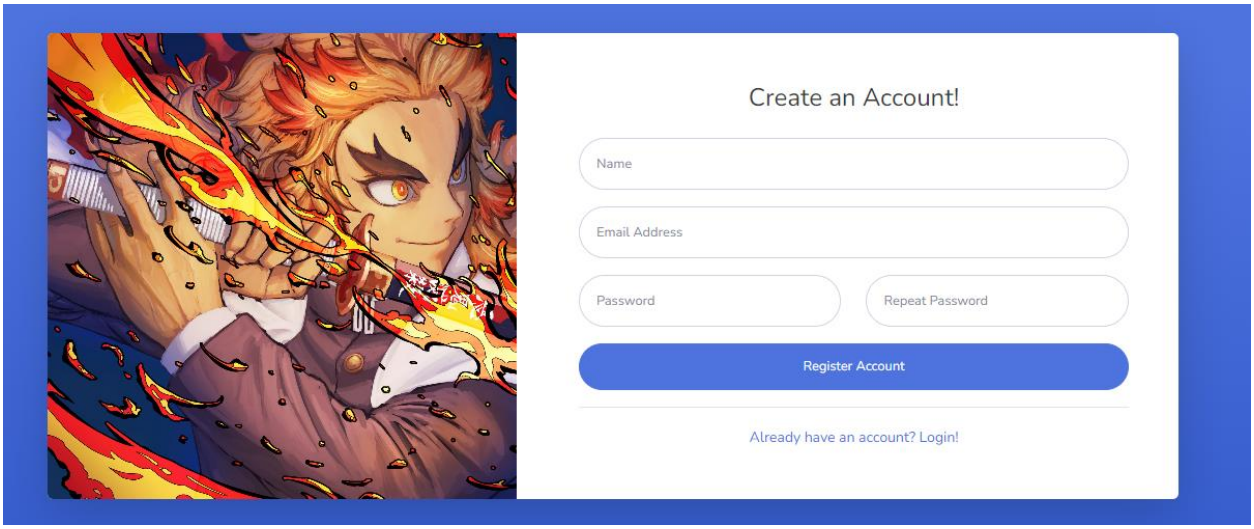

Figure 11 :Register interface

The registration interface in the Tune Source project plays an important role in creating a simple, convenient and secure user experience. The following short discussion will focus on important aspects of this interface.
The registration interface is designed with a focus on user convenience. The registration process is organized logically, from entering basic personal information to confirming

information and creating an account. The information fields are placed clearly and easily to understand, making it easy for users to fill in information without difficulty.
The interface provides icons and instructions to assist users in understanding the meaning of each information field and how to perform the registration steps. This reduces the potential for confusion and enhances the user experience, especially for first-time subscribers.
Information security is a top priority. The registration interface uses security measures such as encryption to ensure that users' personal information is kept safe. Additionally, there are clear notices about the management of personal information and how users' data will be used.
Finally, the registration interface is designed to be cross-platform and device compatible, helping to ensure that users can experience the registration process smoothly from any device, from desktop to phone mobile.
In short, the registration interface is not only an important step in the application usage process but also plays an important role in creating a positive impression on users, affirming professionalism and commitment to the application. safety and convenience.

Login :

Source :By Myself

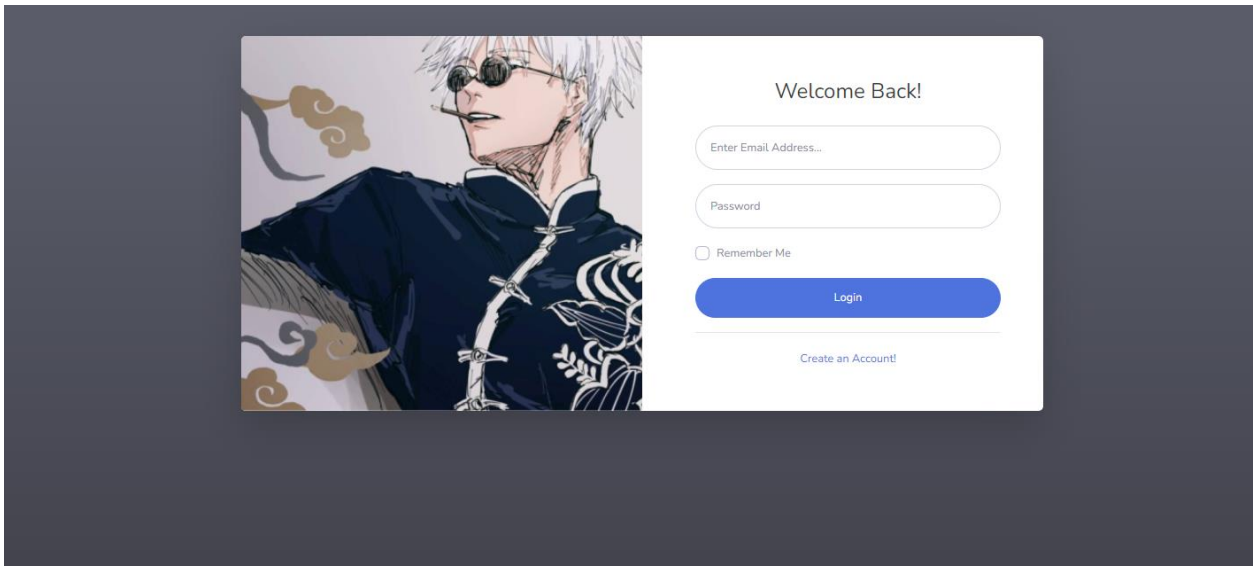

Figure 12 :Login interface

The Tune Source project's login interface is a key part of the system, providing a gateway to a personalized and interactive experience with the digital music platform. Below is a brief discussion of the important aspects of this login interface.

The login interface is designed to be simple yet powerful. Users easily enter their login information, with fields such as username and password placed clearly and conveniently. Error messages are also displayed clearly, helping users understand and fix problems easily.

Information security is a priority. The login interface uses security measures such as password encryption to ensure the safety of user accounts. The system can also provide additional protections such as two-factor verification for added security.

The interface supports forgotten password and account recovery, making it easy for users to fix problems when necessary without interrupting their experience. Additionally, the login interface can integrate quick login options, such as logging in with a Google or Facebook account for added convenience.

The login interface is optimized for cross-platform, ensuring compatibility from computer to mobile phone. This helps users access the system anywhere and anytime conveniently.

In short, the login interface is not only the gateway to the Tune Source experience but also the key point for users to easily access and enjoy the world of digital music conveniently and securely.

Song Admin Interface :

Source :By Myself

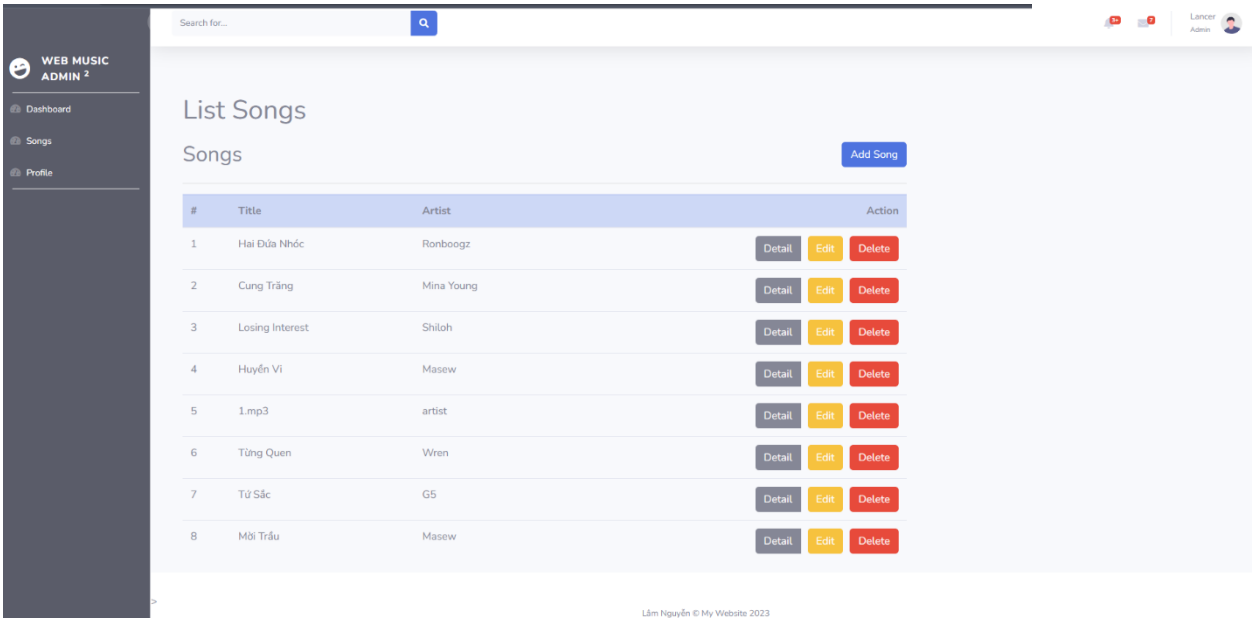

Figure 13 :Song interface

The song interface in the Tune Source project plays an important role in creating a simple, convenient user experience. Specifically for admins, this interface provides

Delete functionality and navigation buttons to the interfaces. Add, edit and view music details.

Detail Interface :

Source :By Myself

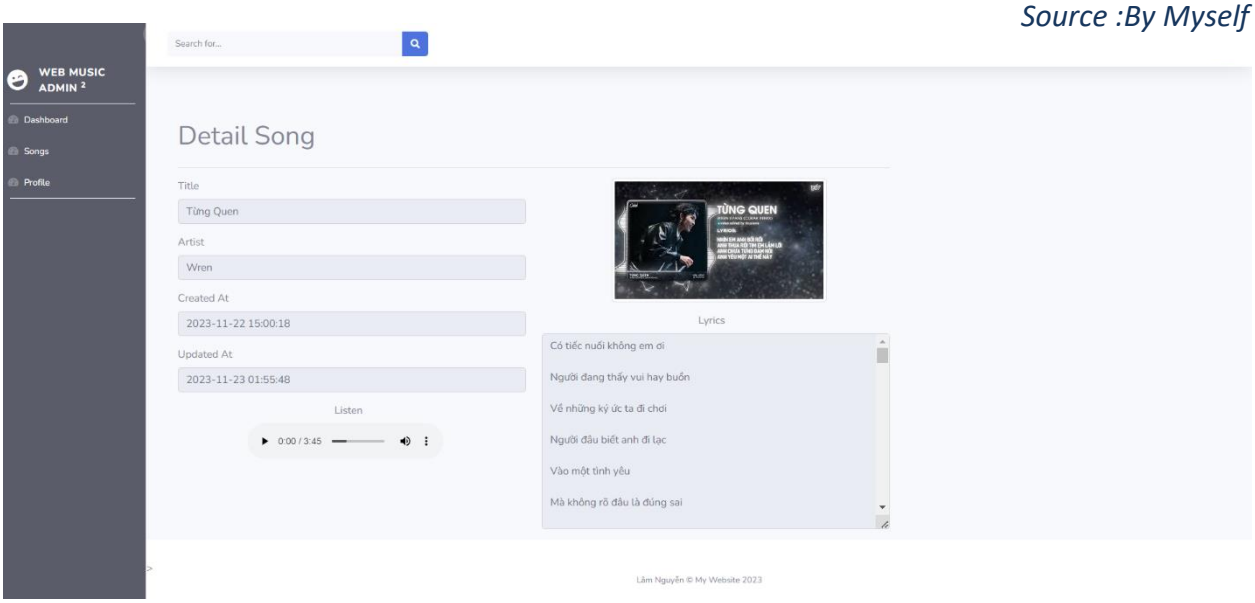

Figure 14 : Detail interface

The song detail interface of the Tune Source application is an important part, giving users a detailed and in-depth experience about each track.

The detailed interface is designed to display complete information about each song, including title, artist, and other details about release year, genre, and lyrics. Users can easily read and understand this information, giving them an overview of the music.

An important feature is the ability to listen to samples before deciding to buy. The detailed interface provides a small button or player so the user can listen to a short snippet from the song. This helps them evaluate music before making a purchasing decision, increasing engagement and excitement.

Edit Song :

Source :By Myself

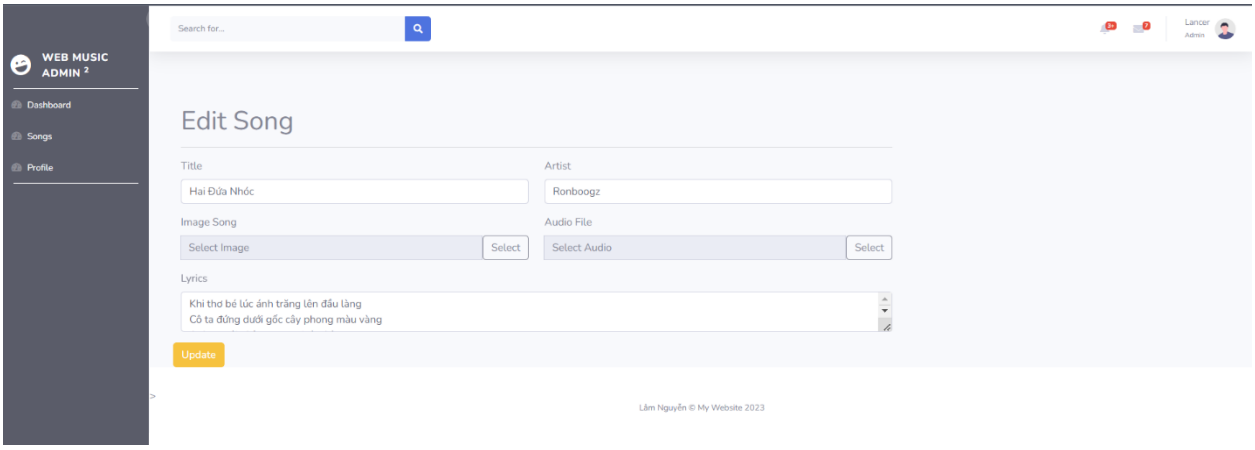

Figure 15 : Edit Song

The song information editing interface in the Tune Source app plays an important role in providing the ability to customize and update information about each track.
The editing interface is designed to be simple and easy to use, making it easy for users to make changes to song information. Information fields such as title, artist, release year, genre and lyrics can all be conveniently edited.

An important feature is the ability to upload or update images that represent songs. Users can take advantage of this function to personalize and beautify their tracks, creating a unique listening experience.

This interface can integrate error checking and validation to ensure that information is updated without formatting issues or empty fields. This helps users avoid unexpected errors and maintain data accuracy.

Additionally, the editing interface can display edit history, allowing users to track previous changes and restore information if necessary. This increases the flexibility and safety of the editing process.

The song information editing interface not only offers customization, but also creates a flexible and convenient editing experience for users, ensuring that they have control and personalization of music content to their liking. the way they want.

Search interface :

Source :By Myself

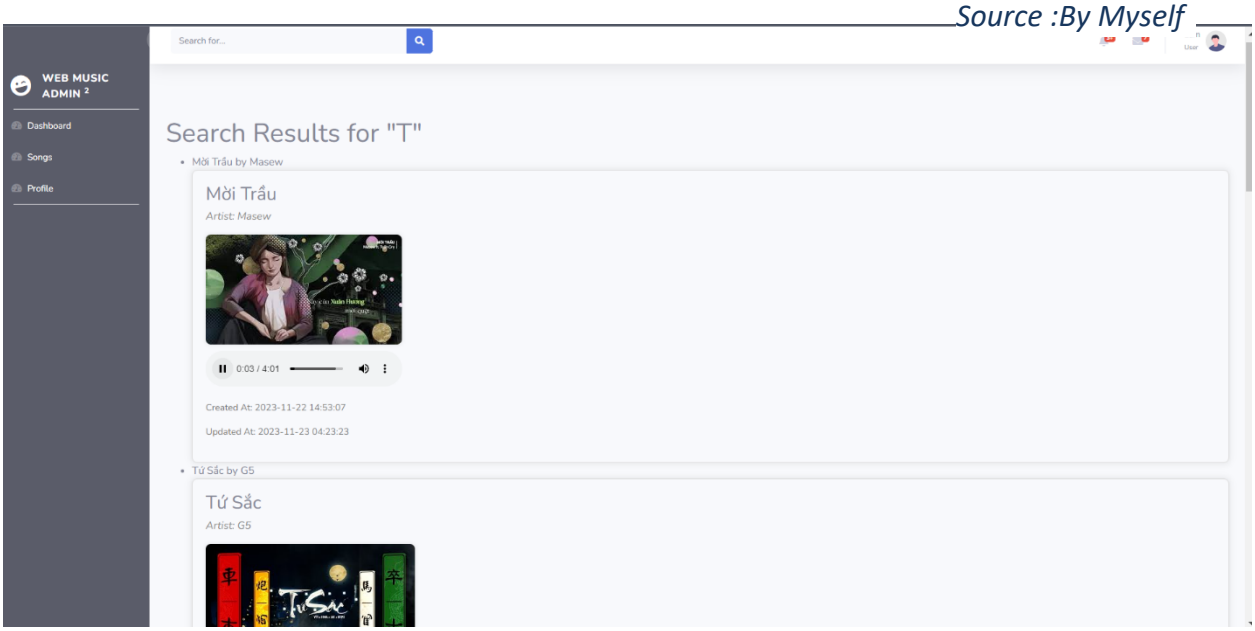

Figure 16 : Search Interface

The song information editing interface in the Tune Source app plays an important role in providing the ability to customize and update information about each track.
The editing interface is designed to be simple and easy to use, making it easy for users to make changes to song information. Information fields such as title, artist, release year, genre and lyrics can all be conveniently edited.

An important feature is the ability to upload or update images that represent songs. Users can take advantage of this function to personalize and beautify their tracks, creating a unique listening experience.

This interface can integrate error checking and validation to ensure that information is updated without formatting issues or empty fields. This helps users avoid unexpected errors and maintain data accuracy.

Additionally, the editing interface can display edit history, allowing users to track previous changes and restore information if necessary. This increases the flexibility and safety of the editing process.

In short, the song information editing interface not only offers customization, but also creates a flexible and convenient editing experience for users, ensuring that they have control and personalization of the content. music the way they want.

Profile Interface :

Source :By Myself

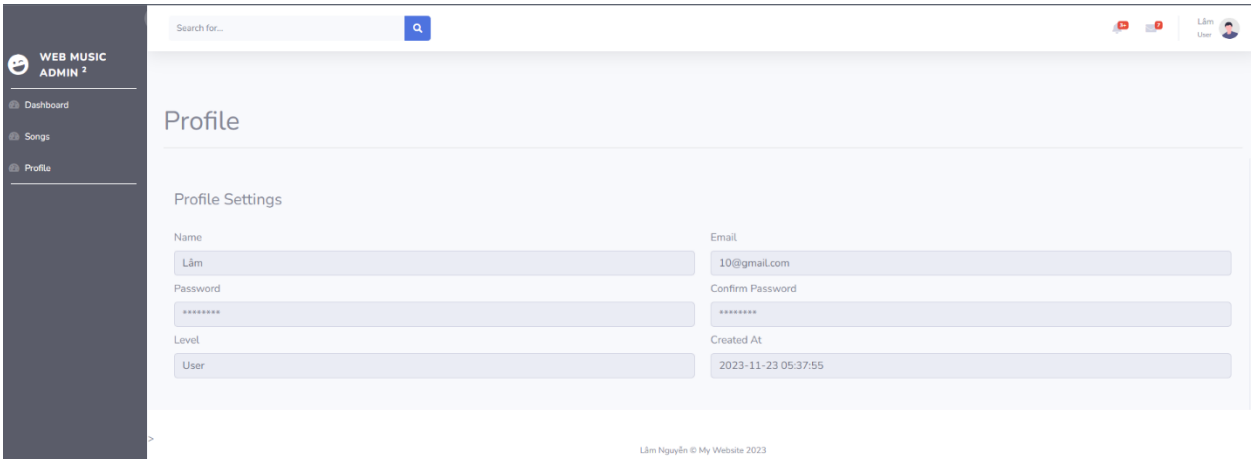

Figure 17 : Profile Interface

The User Profile interface in the Tune Source application is an important part, providing a detailed view of the user's personal information and music preferences.
The User Profile interface is designed to be clear and easy to see, while helping users easily track and manage their personal information. Information fields such as name, email address, avatar and personal description can be displayed intuitively.

In addition, the interface may also include information about the user's music listening history, favorite playlists, and other statistics about the user's musical activity. This creates a personalized space, allowing users to express their musical preferences and explore content flexibly.

The Profile interface may incorporate editing options so users can easily update their personal information. These updates may include changes to your profile picture, personal description, or other information.

Security and privacy features can also be integrated to ensure that users' personal information is protected and shared only according to their wishes.

In short, the User Profile interface is not only a place to view personal information but also a personalized and interactive space, helping users enjoy and manage their music experience flexibly.
.

2. **The architecture is suitable for the project**
   a. **Client-Server**

Characteristic:

A server provides services to many clients.

Users interact directly with the server to receive data or perform functions.

Advantage:

Easy to manage and maintain.

Clear division between interface and logic processing.

Defect:

Scalability may be limited if there are many users.

Complexity increases when there are more functions that need to be processed on the server.

Select:

Choose when the project is small to medium in scale and requires simplicity in management.

   b. **N-tier**

Characteristic:

Divide the application into different layers such as presentation, logic, and data.

Classes can run on different servers.

Advantage:

Easy to manage and expand.

Clearly divide responsibilities between classes.

Defect:

Sometimes there is a delay when data has to go through multiple layers.

Complexity in deployment and maintenance.

Select:

Choose when clear and scalable separation is needed.

   c. **Microservices**

Characteristic:

Divide the application into independent microservices.

Each service can be deployed and updated independently.

Advantage:

Flexible and well scalable.

Easy to deploy and maintain independent services.

Defect:

Requires cloud management and monitoring to ensure continuity.

Complexity in managing relationships between services.

Select:

Choose when you need independence and flexibility in deployment and updates.

d. The architecture is suitable for the Tune Source project

Choice: Microservices Architecture

The Microservices architecture was chosen for the Tune Source project because of the many benefits it offers in the specific context of this music system.

Function Separation:

Benefits: Microservices architecture allows the application to be broken down into independent microservices, each responsible for a specific function.

Applications in Tune Source: There may be separate services that manage searches, shopping transactions, and content information management. This increases scalability and ease of management.

Flexible Expansion:

Benefits: Microservices allow the expansion of only necessary services, without the need to scale the entire system.

Applications in Tune Source: If a function like search requires more resources due to growth, simply extend the search service without affecting other parts of the system.

Easy to Manage and Deploy:

Benefit: Each service can be deployed and updated independently, minimizing impact on the entire system.

Applications in Tune Source: When shopping functionality needs to be updated, simply redeploy the transaction management service without affecting other parts.

Microservices architecture is a suitable choice for Tune Source due to the flexibility and independence it offers. In the context of the music industry, where functions such as search, shopping, and content information management can evolve unevenly, Microservices allow applications to scale flexibly and efficiently. .

Furthermore, the scalability of Microservices helps Tune Source respond flexibly to uneven growth in functionality. If a particular function requires more resources, we can scale up just that service without affecting the entire system. This helps optimize resource usage and minimize costs.

Finally, with Microservices' independent deployment and management capabilities, updates and maintenance become easier. Development teams can work independently on the services they are responsible for, without affecting the work of other teams. This enhances the independence and efficiency of system development and maintenance.

3. Manual document

a. Non-technical

Login ,Logout and Register (Admin and user):

Login :

Step 1: Visit the Login Page

Open a web browser and navigate to the app's login page.
Step 2: Enter Login Information
Enter your username or email address in the "Account" field.
Enter your password in the "Password" field.
Step 3: Click Login
Click the "Log In" button to access your account.
If the account already exists and is correct you will be redirected to the DashBoard interface.
Step 4: Forgot Password
If you forget your password, use the "Forgot Password" option to reset your password.
Step 5: Don't have an account yet
If you do not have an account, click "Create an account" to be navigated to the registration interface.

Register :
Step 1: Visit the Registration Page
If you don't have an account, visit the app's registration page.
Step 2: Enter Registration Information
Fill in personal information such as name, email address, and password in the corresponding fields.
Step 3: Confirm Password
Re-enter the password to ensure accuracy.
Step 4: Accept Terms
Read and accept the terms of use if any.
Step 5: Click Register
Click the "Register" button to complete the registration process.

Logout :
Step 1: Once you have successfully logged in, click on the user icon in the right corner of the screen.
Step 2: select logout in the options panel, then you will be returned to the login page

For listening and downloading music (Admin and user):
Step 1: Navigate to the songs page by clicking songs on the toolbar.
Step 2: Click the play button on the music player to listen
Step 3: click the options button (3 dots) on the music player to customize the music playback speed and download the music as an mp3 file.

Search (Admin and user):
Step 1: Access the Search engine
Access the search bar anywhere.
Step 2: Enter Search Keywords
In the search box, enter the keyword or phrase you want to search for.
Step 3: Click the Search Button
Press the "Search" button or similar button to perform a search.
Step 4: Browse Results
Browse the search results and check if any match your requirements.
Step 5: Option to Filter Results
If needed, use filtering or sorting options to narrow results by specific criteria.

Create song (admin and User):
Step 1: Visit the music creation page
Navigate to the music creation page by clicking "Create new song" on the navigator on the left side of the screen (for Users) or clicking the Addsong button on the navbar on the right corner of the screen (for admins).
Step 2: Enter information
Enter the correct information format, the image item must be in the file format of the image, the audio item must be in mp3 format.
Step 3: Submit form
Click Submit to send the request form to process adding music
Step 4 :Result
If "Song Create Succesfully" appears, you have successfully added music.
If an error warning is returned, please review the file format or see if you have filled in all the information.

Delete song (only admin):
Step 1: Access song
Click Song on the left navigator to go to the Song page
Step 2: Delete
Find the song you want to delete and click "Delete" in the Action column
Step 3: Click "Ok" on the deletion confirmation message sent to the browser number.

Update song (only admin) :
Step 1: Access song
Click Song on the left navigator to go to the Song page
Step 2: Access the Edit page
Click Edit in the Edit column to navigate to the Edit page
Step 3: Enter all the information and make sure it matches the input format.

Step 4: Update
Click Update to perform the update
Step 5 :Result
If the notification is successful, the action ends
If you receive an error message, please review the file format.

b. Teacnical

System or Product Deployment:

Primary Objective: Teach readers how to properly install and deploy a system or product.

Target Audience: System administrators, developers, or project managers.

Implementation process :

System installation
Data Preparation
Product Deployment
Performance Testing

Operating manual :

Backup and Restore Procedure:
Use backup services and have backup servers
Maintenance and Updates:
Periodically check and update the database, upgrade code and develop the system

Performance Optimization:

To optimize the performance of the Tune Source music listening website, you can take the following measures:

Image and Multimedia Optimization:

Use lower resolution images to reduce page size and speed up page loading.
Compress and optimize audio files to reduce download size.

Database Optimization:

Use efficient SQL queries to ensure quick and efficient data retrieval.
Caching frequently used query results to reduce database access time.

Using Content Delivery Network (CDN):

Use CDN to distribute resources such as images, sounds and CSS files to servers far away from users, reducing page load latency.

Minify and Merge CSS/JavaScript Files:

Use minification tools to reduce CSS and JavaScript file sizes.
Combine CSS and JavaScript files into one to reduce the number of HTTP requests.

Caching Optimization:
Use cache-control headers to set how long resources are cached on the user's browser.
Use server-side cache to store database query results and reduce response time.

Optimize JavaScript Code:
Test and optimize JavaScript code to reduce execution time in user browsers.

Using Lazy Loading Technique:
Apply lazy loading to images and rich media content to download only when the user scrolls to that part of the page.

Browser Compatibility Guaranteed:
Test and ensure that the website works smoothly on various browsers.

Control the Number of HTTP Requests:
Reduce the number of HTTP requests using file pooling techniques and minimize resource segregation.

Using Static IP Address:
Use a static IP address for your website to reduce DNS lookup time and increase page loading speed.

Safety and Security:
To ensure the safety and security of the Tune Source music listening website, you can take the following measures:
Use a Secure Connection (SSL/TLS):
Enable SSL/TLS to create a secure connection between the server and browser, protecting data transmission.

Two-Factor Authentication (2FA):
Provides two-factor authentication to enhance user account security.

Account Management and Authorization:
Implement strict account management and set permissions carefully to limit access to the minimum principle.

User Data Protection:
Encrypt user data in the database to protect personal information.

Managing Security Risks:
Perform periodic security testing to detect and fix security vulnerabilities that may exist.

Continuous System Monitoring:
Set up continuous monitoring to monitor system activity and detect unusual behavior early.

Information Security Training Organization:
Train employees on information security and practice security measures.

Protection From Types of Attacks:
Use firewalls, antivirus software, and network security solutions to protect against cyberattacks.

System Log Management:
Store and monitor system logs to analyze and track important events.

Compliance with Security Standards:
Comply with security standards such as ISO 27001 to ensure the system meets international information security requirements.

Data Integrity Security:
Use digital signatures and data integrity security to ensure data is not illegally changed.

Implementing Security at the Application Level:
Tests and protects against application-level vulnerabilities, including OWASP Top 10 attacks.

Fisico Resource Security:
Protect your website's physical resources such as servers and data centers from unauthorized access.

Error Correction and Testing :
Possible error cases for this website mainly come from functions that handle user requests, so we will pay a lot of attention to AuthController, SearchController and SongController.

```php
<?php

namespace App\Http\Controllers;
```

```php
use Illuminate\Http\Request;
use App\Models\User;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use Illuminate\Validation\ValidationException;

class AuthController extends Controller
{
    public function register()
    {
        return view('auth/register');
    }

    public function registerSave(Request $request)
    {
        Validator::make($request->all(), [
            'name' => 'required',
            'email' => 'required|email',
            'password' => 'required|confirmed'
        ])->validate();

        User::create([
            'name' => $request->name,
            'email' => $request->email,
            'password' => Hash::make($request->password),
            'level' => 'User'
        ]);

        return redirect()->route('login');
    }

    public function login()
    {
        return view('auth/login');
    }

    public function loginAction(Request $request)
    {
        Validator::make($request->all(), [
            'email' => 'required|email',
            'password' => 'required'
        ])->validate();
```

```
            if   (!Auth::attempt($request->only('email',   'password'),
$request->boolean('remember'))) {
            throw ValidationException::withMessages([
                'email' => trans('auth.failed')
            ]);
        }
        $user = Auth::user();

        if ($user) {
            if ($user->level === 'Admin') {
                return redirect()->route('dashboard');
            } elseif ($user->level === 'User') {
                return view('userdashboard');
            }
        }


        $request->session()->regenerate();
    }

    public function logout(Request $request)
    {
        Auth::guard('web')->logout();

        $request->session()->invalidate();

        return redirect('/');
    }

    public function profile()
    {
        return view('profile');
    }
    public function userprofile()
    {
        return view('userprofile');
    }
    public function update(Request $request)
    {
        $user = auth()->user();

        // Chỉ cho phép cập nhật các trường cụ thể
        $user->name = $request->input('name');
        $user->phone = $request->input('phone');
        $user->address = $request->input('address');
```

```
        $user->save();

            return  redirect()->route('profile.edit')->with('success',
'Profile updated successfully.');
    }

}
```

This function is the code of AuthController. It is responsible for requests like registration, login, Logout and navigating users to different pages based on their permissions.

```
<?php

namespace App\Http\Controllers;
use App\Http\Requests\StoreRequest;
use Illuminate\Http\Request;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Storage;
use App\Models\Song;
use Illuminate\Http\Response;
class SongController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $songs = Song::orderBy('created_at', 'DESC')->get();

        return view('songs.index', compact('songs'));
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        return view('songs.create');
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(StoreRequest $request)
```

```
    {
        try {
                $imageName = Str::random(32) . '.' . $request->image->getClientOriginalExtension();
            $audioName = Str::random(32) . '.' . $request->audio_file->getClientOriginalExtension();

            // Create Song
            Song::create([
                'title' => $request->title,
                'artist' => $request->artist,
                'image' => $imageName,
                'audio_file' => $audioName,
                'lyrics' => $request->lyrics,
            ]);

            // Save Image in Storage folder
                 Storage::disk('public')->put('images/' . $imageName, file_get_contents($request->image));

            // Save Audio in Storage folder
               Storage::disk('public')->put('audio_file/' . $audioName, file_get_contents($request->audio_file));

            // Set success message in session
            session()->flash('success', 'Song successfully created.');

            // Redirect back to the previous page
            return redirect()->back();
        } catch (\Exception $e) {
            // Log the error
            \Log::error('Error in store method: ' . $e->getMessage());

            // Set error message in session
            session()->flash('error', 'Something went really wrong!');

            // Redirect back to the previous page
            return redirect()->back();
        }
    }



    /**
     * Display the specified resource.
```

```
     */
    public function show(string $id)
    {
        $songs = Song::findOrFail($id);

        return view('songs.show', compact('songs'));
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        $song = Song::findOrFail($id);

        return view('songs.edit', compact('song'));
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(StoreRequest $request, string $id)
    {
        try {
            $song = Song::findOrFail($id);

            // Update title and artist
            $song->update([
                'title' => $request->input('title'),
                'artist' => $request->input('artist'),
                'lyrics' =>$request->input('lyrics'),
            ]);

            // Update image
            if ($request->hasFile('image')) {
                // Public storage
                $storage = Storage::disk('public');

                // Old image delete
                if ($storage->exists('images/' . $song->image)) {
                    $storage->delete('images/' . $song->image);
                }

                // Image name
```

```
                    $imageName = Str::random(32) . "." . $request->file('image')->getClientOriginalExtension();
                $song->update(['image' => $imageName]);

                // Image save in public folder
                            $storage->put('images/' . $imageName, file_get_contents($request->file('image')));
            }

            // Update audio path
            if ($request->hasFile('audio_file')) {
                // Public storage
                $storage = Storage::disk('public');

                // Old audio path delete
                if ($storage->exists('audio_file/' . $song->audio_file)) {
                    $storage->delete('audio_file/' . $song->audio_file);
                }

                // Audio name
                    $audioName = Str::random(32) . "." . $request->file('audio_file')->getClientOriginalExtension();
                $song->update(['audio_file' => $audioName]);

                // Audio save in public folder
                            $storage->put('audio_file/' . $audioName, file_get_contents($request->file('audio_file')));
            }
            $song->save();
            // Return a success message
            session()->flash('success', 'Song successfully created.');

            // Redirect back to the previous page
            return redirect()->back();
        } catch (\Exception $e) {
            // Log the exception for debugging purposes
            \Log::error('Error updating song: ' . $e->getMessage());

            // Return a more informative response
            session()->flash('error', 'Something went really wrong!');

            // Redirect back to the previous page
            return redirect()->back();
        }
```

```
    }


    /**
     * Remove the specified resource from storage.
     */
    public function destroy(string $id)
    {
        $song = Song::findOrFail($id);
        $storage = Storage::disk('public');
        if ($storage->exists('images/' . $song->image)) {
            $storage->delete('images/' . $song->image);
        }
        if ($storage->exists('audio_file/' . $song->audio_file)) {
            $storage->delete('audio_file/' . $song->audio_file);
        }
        $song->delete();

        session()->flash('success', 'Song successfully created.');

            // Redirect back to the previous page
            return redirect()->back();
    }
}
```

This function is the code of SongController. It is responsible for requests to perform basic functions like CRUD.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Song;
class SearchController extends Controller
{
    public function index(Request $request)
    {
        // Nhận dữ liệu từ thanh tìm kiếm
        $search = $request->input('search');

        // Thực hiện tìm kiếm
        $results = Song::where('title', 'like', '%' . $search . '%')
                        ->orWhere('artist', 'like', '%' . $search . '%')
```

```
                    ->get();

        // Trả về view với kết quả tìm kiếm
        return view('search.index', ['results' => $results, 'search' => $search]);
    }
}
```

This is the Controller that performs the search function for the entire system. This function has a high frequency of use, so it is necessary to maintain it regularly.

1 Number of specific error cases:

Source :By Myself

Case_No	Case_Name	Reason	Treatment
1	Login Failed	1. Wrong Information 2. Error source code	1. Correct your information and log in again 2. Check the interface file and route path there Check the function responsible for Login at AuthController
2	Register Failed	1. The data type filled in for registration is not in the correct format 2. Error source code	1. Please enter the correct required format 2. Check the registration processing function and the registration interface form
3	Add Failted	1. The data type filled in for add is not in the correct format 2. Error source code	1. Please enter the correct required format 2. Check the Create processing function and the Create interface form
4	Upfate Failed	1. The data type filled in for update is not in the correct format 2. Error source code	1. Please enter the correct required format 2. Check the Update processing function and the Update interface form
5	Delete	1. The Show data function has a problem and shows data that does not exist or has been deleted. 2. The delete processing function has an error 3. The interface is broken and the route is lost 4. The connection to the data store is broken	1.Check and fix the show function if there is an error 2.Check and fix the Delete function if there is an error 3. Check and fix if the interface and route are broken 4. Reconnect to the data store
6	Search failed	1. The handler function is corrupted 2. interface and route are broken	1. Edit the search handler function 2. Fix broken interfaces and routes

Figure 18 : Error cases

V. Conclusion

In this report, we learned about the software development process and its management, the SDLC. We have identified the stakeholders and their roles in the specific software project, to ensure participation and support from all interested parties.

We analyzed requirements using structural and behavioral modeling techniques, including Use Case diagrams, data flow diagrams, and entity-relationship diagrams (ERD). Thereby, we have an overview of the system's functions and requirements.

Next, we discussed how to meet user requirements and software requirements during the design phase. We learned about web interface design, the right architecture for the project, and creating user documentation.

To summarize, understanding and applying SDLC plays an important role in effective and reliable software development. Through the process of researching and implementing the SDLC stages, we can build software products that meet customer requirements and deliver business value.

VI. References

1. insights.magestore.com. (n.d.). Làm sao khám phá yêu cầu (Requirement). [online] Available at: https://insights.magestore.com/posts/business-analyst-requirement#:~:text=Requirement%20l%C3%A0%20m%E1%BB%99t%20%C4%91%C3%B2i%20h%E1%BB%8Fi [Accessed 24 Nov. 2023].
2. BAC TRAINING & CONSULTANCY VN. (n.d.). Yêu cầu chức năng và phi chức năng. [online] Available at: https://www.bacs.vn/vi/blog/kien-thuc/yeu-cau-chuc-nang-va-phi-chuc-nang-3910.html. [Accessed 24 Nov. 2023].
3. Newline.tech. (2023). Available at: https://newline.tech/wp-content/uploads/2018/02/Illustration_for_nlt_blog_joint_application_development-compressor.webp [Accessed 25 Nov. 2023].
4. Asana (n.d.). Another efficiency hack? This one is worth doing. [online] Asana. Available at: https://asana.com/resources/business-process-analysis. [Accessed 25 Nov. 2023].
5. SearchCIO. (n.d.). What is business process improvement (BPI)? - Definition from WhatIs.com. [online] Available at: https://www.techtarget.com/searchcio/definition/business-process-improvement-BPI. [Accessed 25 Nov. 2023].
6. Dương, N.T. (2020). Wireframe trong thiết kế web: Cách thức, công cụ, ví dụ tốt nhất năm 2020. [online] Vũ Thành Lâm. Available at: https://lamvt.vn/wireframe-thiet-ke-web-cach-thuc-cong-cu-vi-du-tot-nhat-nam-2020/ [Accessed 26 Nov. 2023].
7. LancerLam2k4 (2023). SDLC-ASM. [online] GitHub. Available at: https://github.com/LancerLam2k4/SDLC-ASM.git [Accessed 26 Nov. 2023].