ASM2 SDLC - ASM

Information Technology (Trường Đại học FPT)

ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 9: Software Development Life Cycle		
Submission date	14/8/2022	**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Vũ Ngô Tuấn Kiệt	**Student ID**	BH00284
Class	PBIT17103	**Assessor name**	Ngô Thị Mai Loan
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	

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

I. Introduction

- In this ASM I will briefly present the basic knowledge of the system requirements, categorize it with diagrams like ERD, DFD, Usecase,... and apply all of those things practically to my web post.

II. Undertake a software investigation to meet a business need

1. The requirements in the software system

1.1 What is the Requirement in the system software?

- An individual physical or functional requirement is something that product development and process improvement attempt to address through a particular design, product, or procedure. It is frequently utilised in formal engineering design processes like systems engineering, software engineering, or enterprise engineering. It is a broad concept that can refer to any necessary (or occasionally desired) feature, ability, quality, or characteristic of a system in order for it to be valuable and practical for a client, business, internal user, or other stakeholders.

1.2 Classify

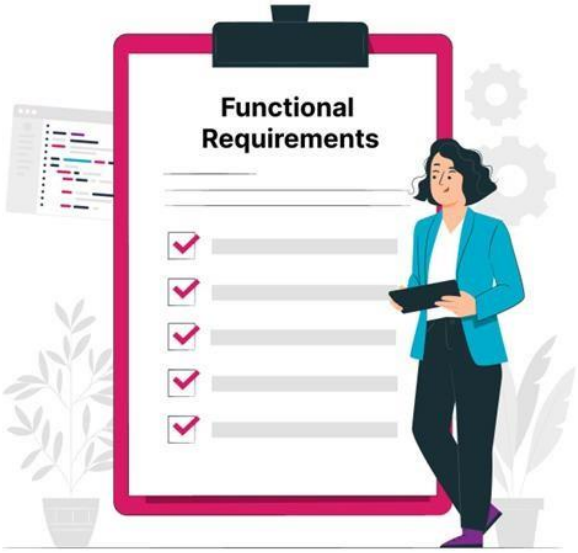
- Broadly software requirements should be categorized in two categories:

*Functional Requirements : Requirements, which are related to functional aspect of software fall into this category. They define functions and functionality within and from the software system. It might be a computation, data manipulation, business procedure, user interaction, or any other particular capability that specifies what purpose a system is likely to serve.

Examples -

- Search option given to user to search from various invoices.

- User should be able to mail any report to management.
- Users can be divided into groups and groups can be given separate rights.
- Should comply business rules and administrative functions.
- Software is developed keeping downward compatibility intact.



* Non-Functional Requirement: Describes a software system's quality aspect. They assess the software system according to non-functional criteria such as responsiveness, usability, security, portability, and other criteria essential to the software system's success.

Non-functional requirements include -

- Security
- Logging
- Storage
- Configuration
- Performance
- Cost
- Interoperability
- Flexibility
- Disaster recovery
- Accessibility

Requirements are categorized logically as

- **Must Have** : Software cannot be said operational without them.
- **Should have** : Enhancing the functionality of software.

- **Could have** : Software can still properly function with these requirements.
- **Wish list** : These requirements do not map to any objectives of software.

2. Analyze functional and non-functional requirements\

- The functional requirements and non-functional lay the foundation for a software development project's success. Functional requirements mentioned specific features to promote the development of software. Non-functional requirements are directly linked to software features.

Parameters	Functional Requirement	Non-Functional Requirement
What is it?	Verb	Attributes
Requirement	It is mandatory	It is non-mandatory

Capturing type	It is captured in use case.	It is captured as a quality attribute.
End-result	Product feature	Product properties
Capturing	Easy to capture	Hard to capture
Objective	It helps you verify the functionality of the software.	Helps you to verify the Performance of the software.
Area of focus	Focus on user requirement	Concentrates on the user's Expectations.
Documentation	Describe what the product does	Describes how the product works
Type of Testing	Functional Testing like System, Integration, End to End, API testing, etc.	Non-Functional Testing like Performance, Stress, Usability, Security testing, etc.
Test Execution	Test Execution is done before non-functional testing	After the functional testing
Product Info	Product Features	Product Properties

-Business Requirements: Using the Web or in-store kiosks, customers will be able to search for and purchase digital music downloads. The specific functionality that the system should have includes the following:

- Search for music in our digital music archive.
- Listen to music samples.
- Purchase individual downloads at a fixed fee per download.
- Establish a customer subscription account permitting unlimited downloads for a monthly fee.
- Purchase music download gift cards.

3. List the required functions and the system of non-functions

FUNCTIONAL REQUIREMENTS	DESCRIPTION	EXAMPLE
Download	**Allow customers to download the music they have purchased**	**Customers can download their favourite songs and play them without internet. Also, you can create an album to add it on**
Listen	**Allow consumers to play music online without downloading by connecting to the internet.**	**Customers click the song to listen. When the song ends, the music will next to a new one**
Search	**Allow users to search for songs by entering song-related keywords such as song title, artist name, album name, or a word in the song**	**When a consumer enters a song title or lyric of the song, the system will display the name song**
Register	**Allows new users to sign up for a new account to use the**	**Customers can connect with their registered**

	Tune Source service.	account and download the copyrighted from Tune Source
Login to website	Allow customers and admins to login into the Tune Source system and access all other features by using available accounts	When logged, the customer can use complete services of the system, as well as a purchase gift card
Purchase gift card/ song	Allows customers to purchase gift cards and song	Customers can buy the song to enjoy themselves and gift cards to their friends, like trail 15 days to trail service
Manage account/information	Admin can manage and modify account information of customers	Admin can change customers' information such as name, address, bill, date of birth,...

FUNCTIONAL REQUIREMENTS	DESCRIPTION

Security	**Security on a user's database should include a firewall to prevent unauthorised access**
Capacity	**Make sure that the capacity of the system is enough to accommodate when many people visit at the same time; users can adjust and save settings based on their preferences**
Compatibility	**The system works efficiently in a variety of environments. Users can access the website on many devices such as phones, tablets, laptops...**
Reliability	**Technology that is highly reliable functions with the same or similar efficacy after extensive use**

4. Requirements gathering techniques and appropriate technique selection

- Requirements gathering techniques include:

- Observation
- Prototyping
- Focus group
- Survey/Questionnaire
- Document Analysis/Review
- Use Cases and Scenarios
- Interface Analysis
- Interview

- In this Tune Source project, a survey/questionnaire is used to collect needs since it is simple to collect data from a broad audience, takes less time for users to reply, and provides more reliable information than other strategies.

- Assumption about the project justifying the techniques that you have chosen:

The Questionnaires approach enables team projects to get critical information and data. Furthermore, if the team project is to create an excellent research report, the project must have exceptional facts. Furthermore, if the project requires high-quality data, a well-designed questionnaire or research would be required. Questionnaire procedures are the most popular and easiest way to gather data in research. As a result, a team project may create a survey, disseminate it to a large number of individuals, and then analyze and assess the responses.

III. Use appropriate software analysis tool/techniques to carry out a software investigation and create supporting documentation (P6)

1. Use case diagram

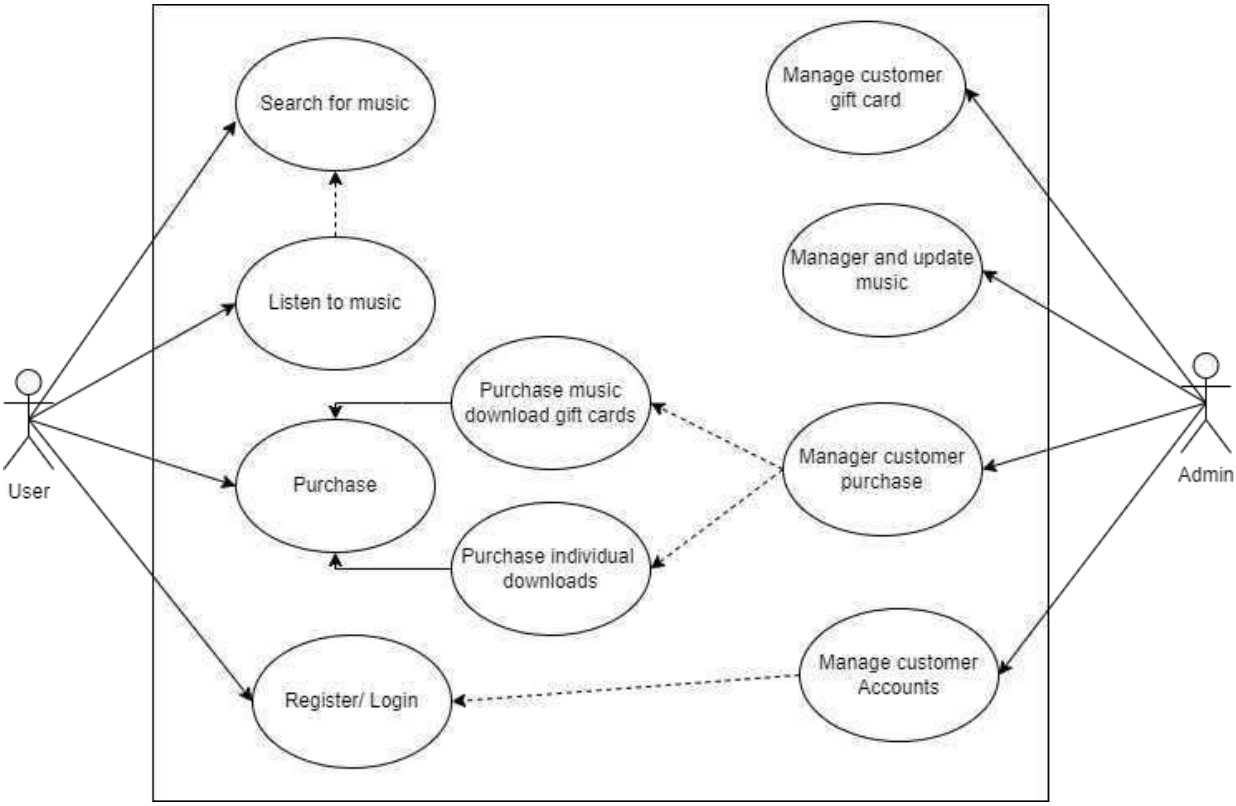
1.1 Definition

- A UML use case diagram serves as the main representation of the system/software requirements for a software application that has not yet been developed. Use cases describe the desired behavior (what), not the method for accomplishing it (how). Use cases can be expressed textually and visually after being defined (i.e., use case diagram). Use case modeling's contribution to system design from the perspective of the end user is a key idea. By defining each externally visible system action, it's possible to explain system behavior to users in their own words.

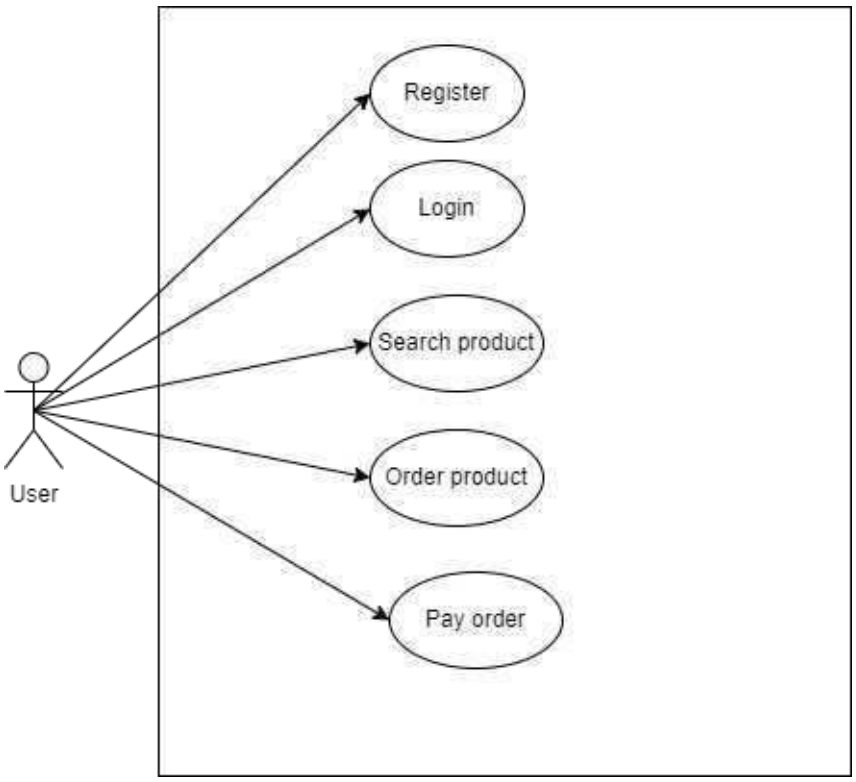
- Purpose of Use Case Diagram:

- Specify the context of a system
- Capture the requirements of a system
- Validate a systems architecture
- Drive implementation and generate test cases • Developed by analysts together with domain experts

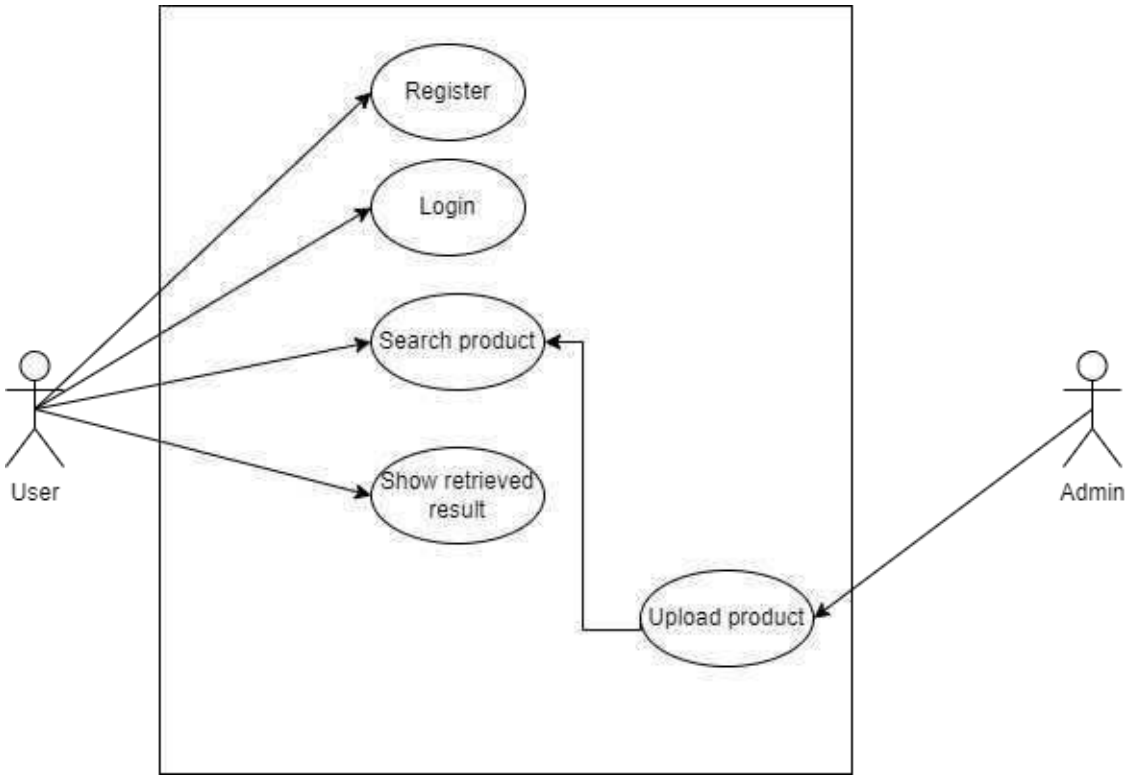
1.2 Apply



Major use case



Buy Function

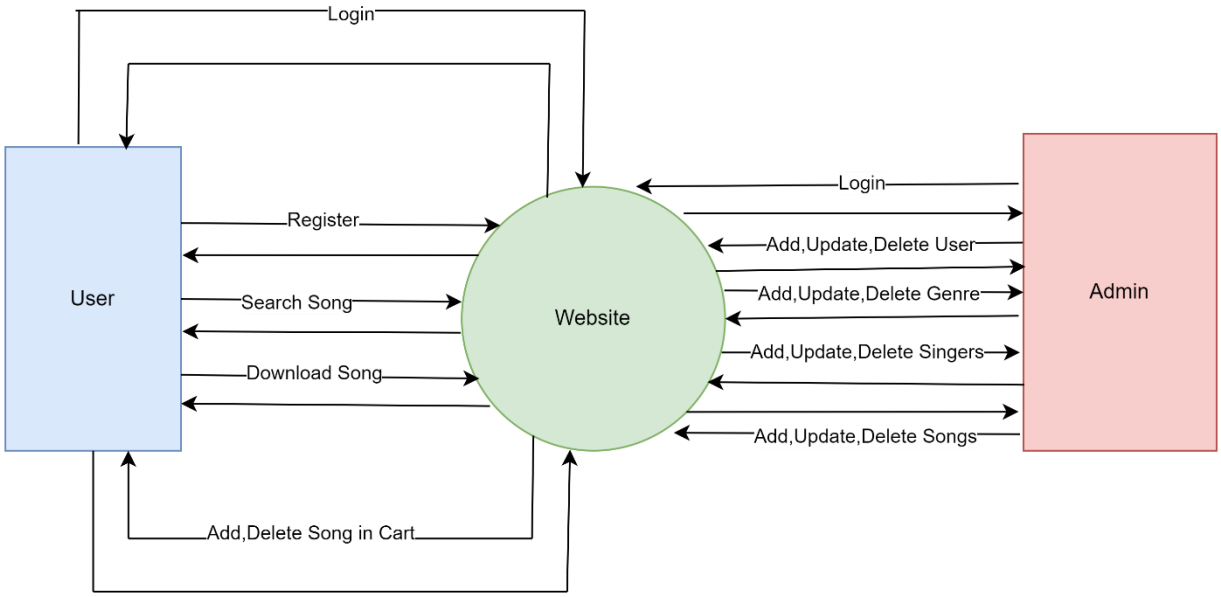


Search Function

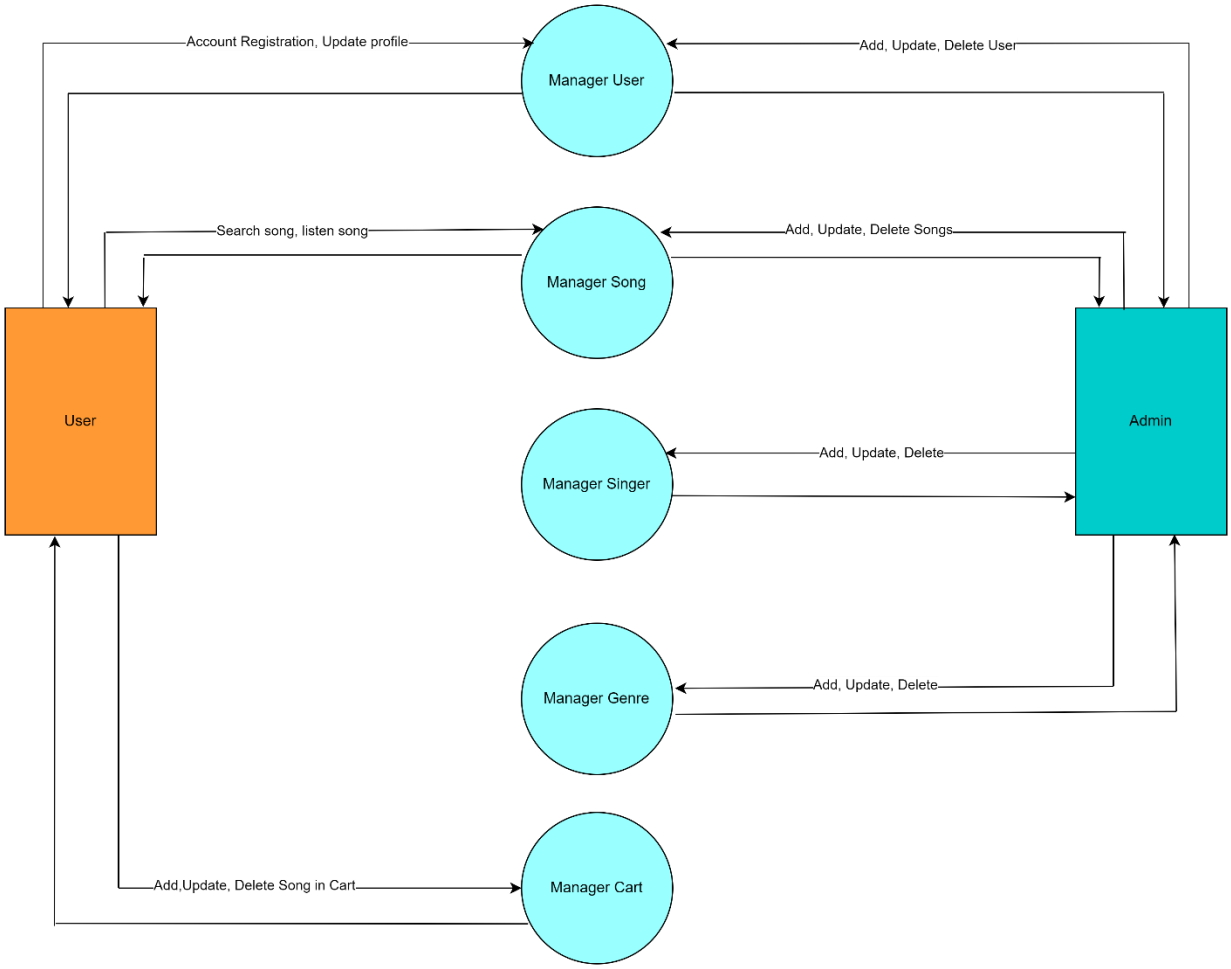
2.Data Flow Diagram

- A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

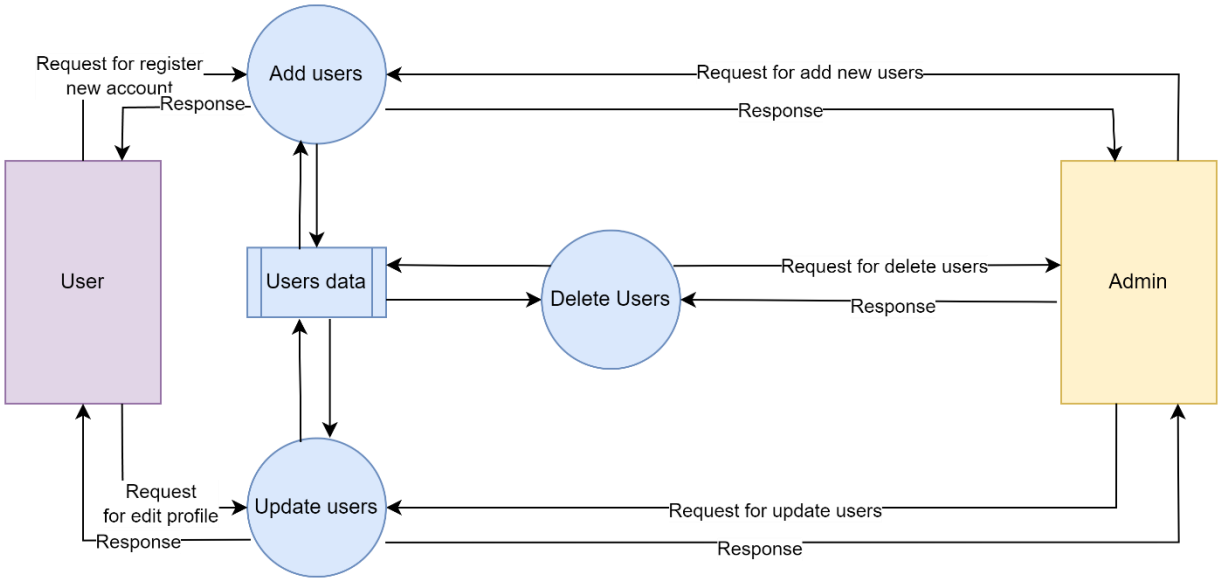
*DFD Lv0



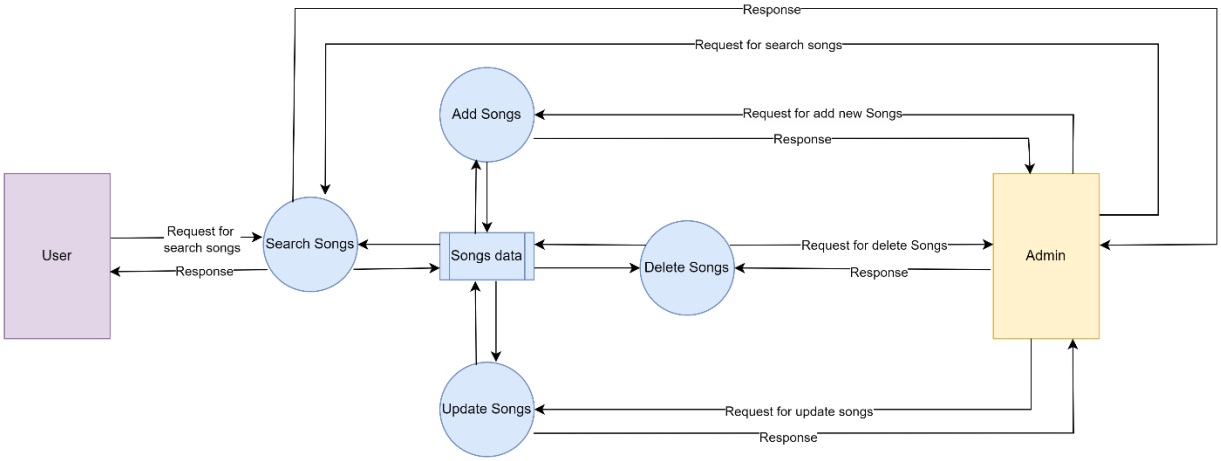
*DFD Lv1



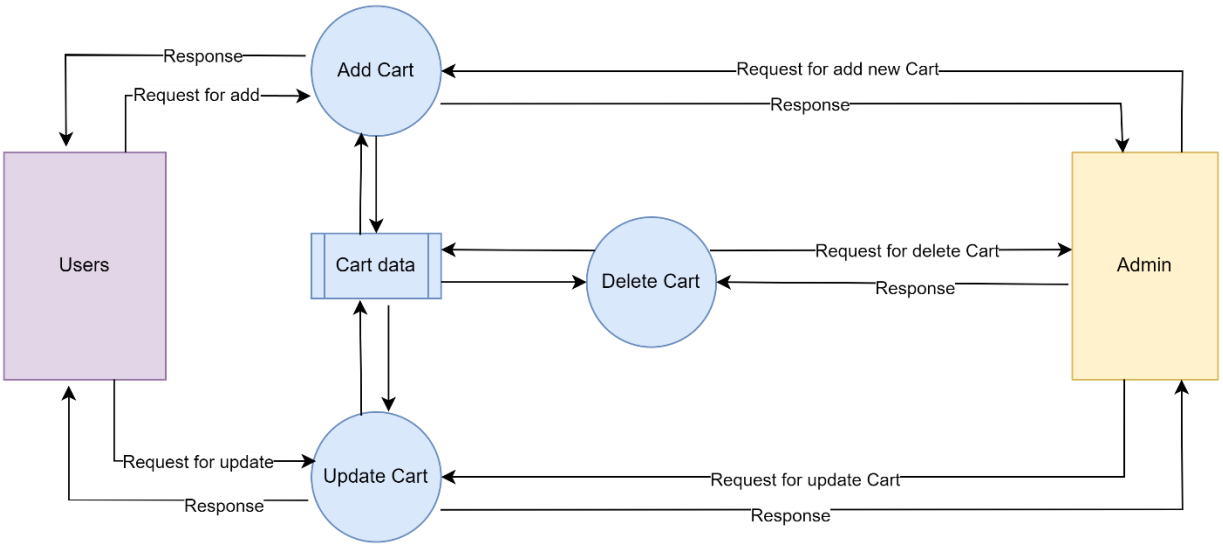
*DFD Lv2



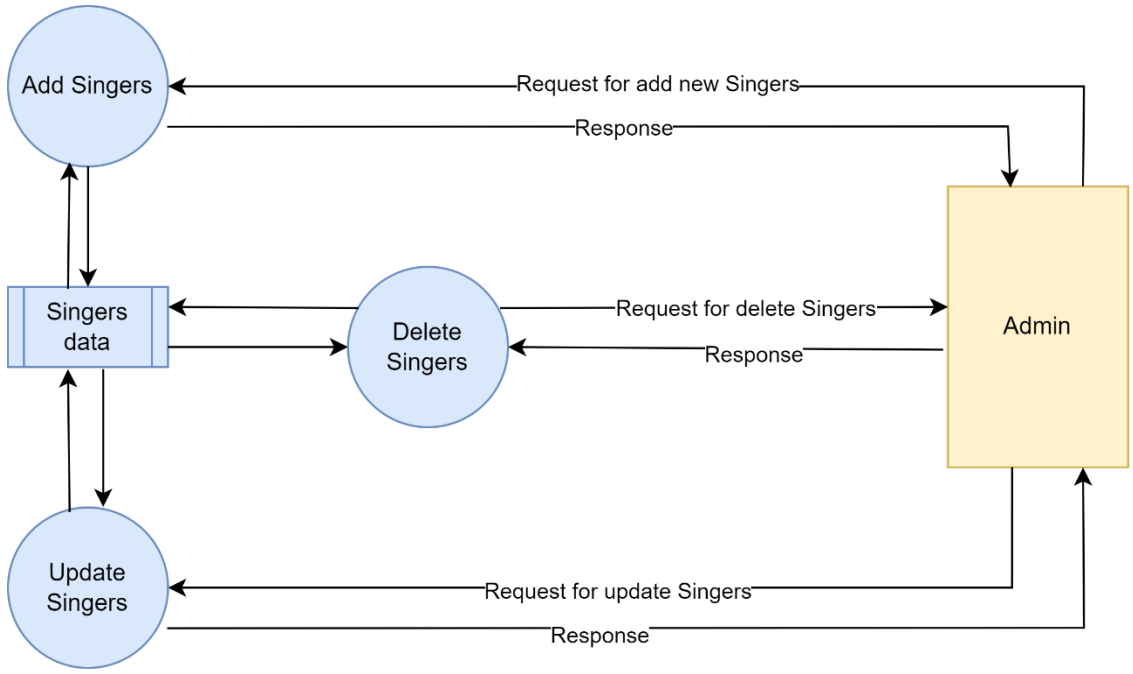
Manage users



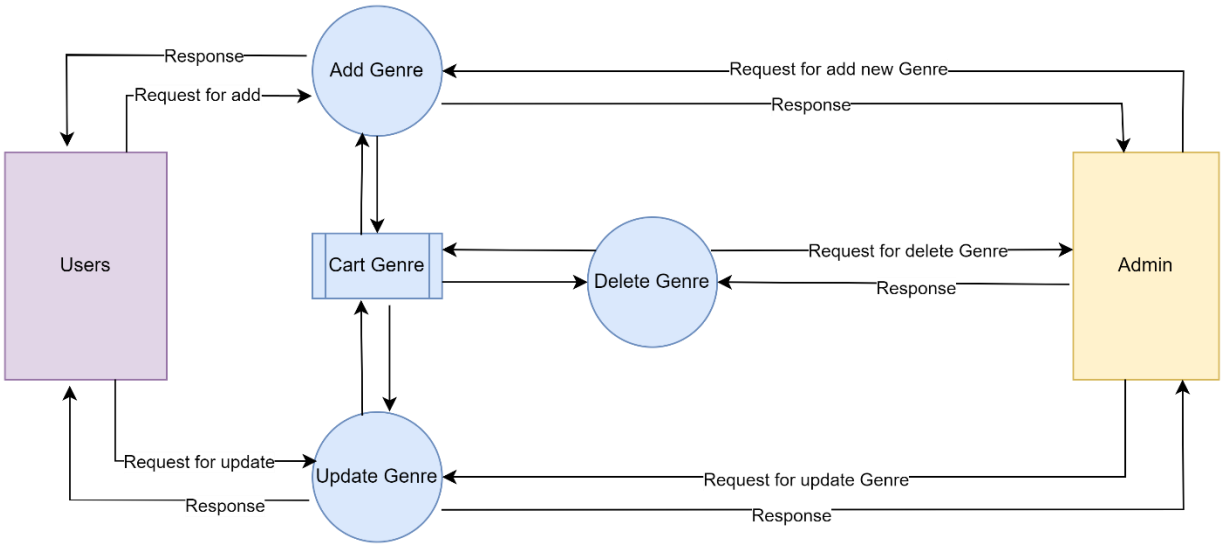
Manage Songs



Manage Cart



Manage singer



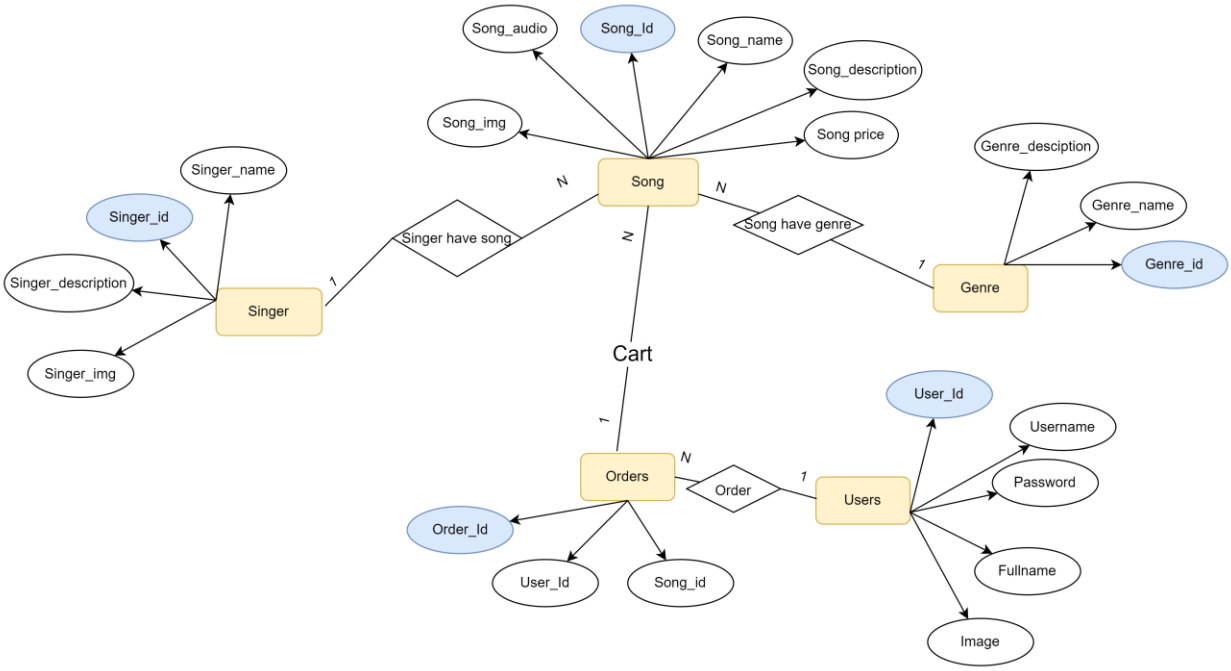
Manage Genre

3. Entity Relation Diagram

- An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

Entity	Description of Entity	Attribute	Description of Attribute
Song	All song in website	SongID	Primary key. It is a unique identifier for each song in the website
		SongName	Name of the song
		GenreID	Genre of the song: POP, lyrical, ...
		SingerID	It is a unique identifier for each singer in the website

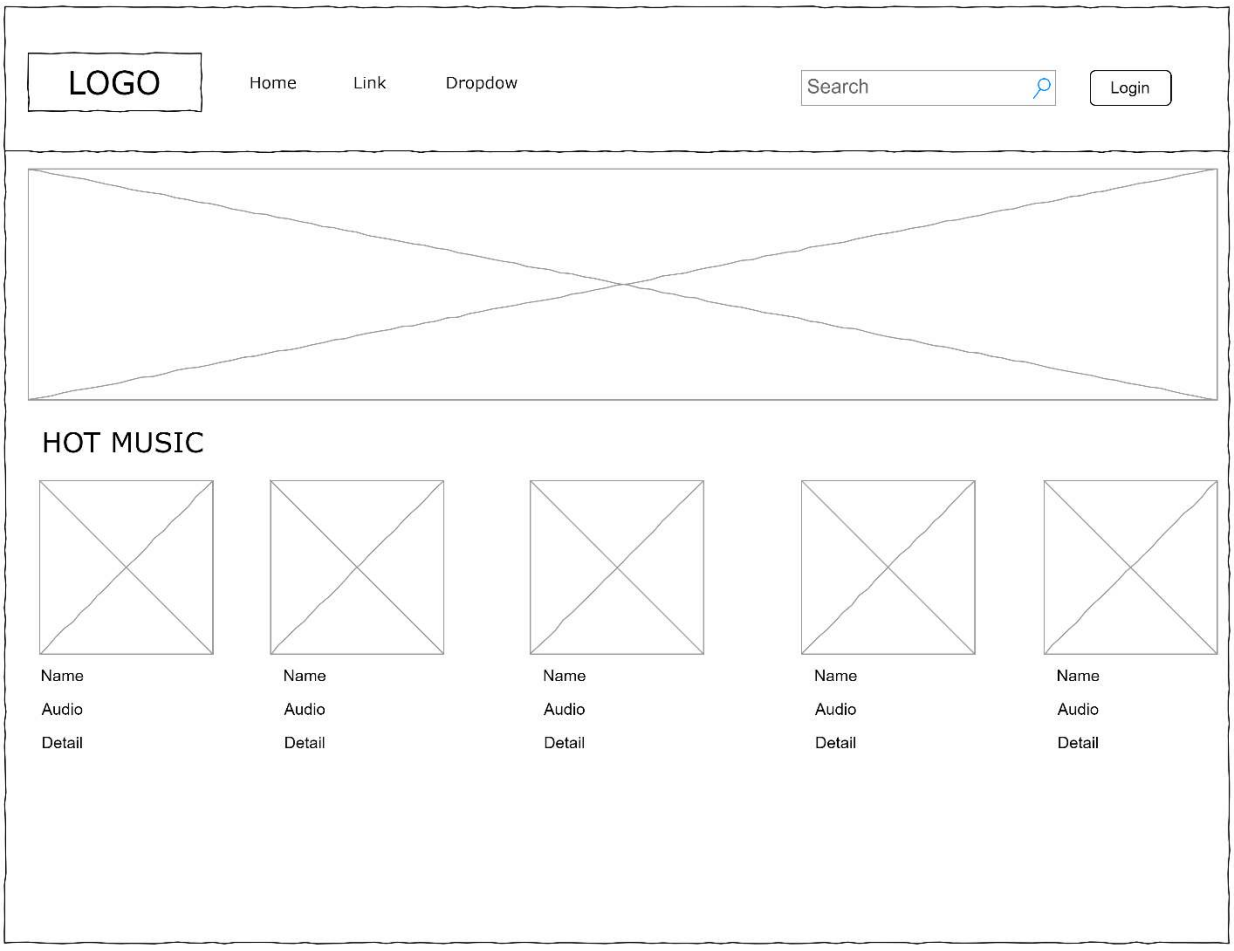
		Song_img	Show song image
		Song Audio	Make the song playable on the web
		Song Price	To help users can see the price of the song to buy and listen to the full song
		Song Description	Use to description song
User	All users on the site	UserID	Primary key. It is a unique identifier for each user in the website
		UserName	Name of user
		Password	Help users log in to the system
		Fullname	User's full name
		Email	Email of user
Genre	Genre of all songs on the web	GenreID	Primary key. It is a unique identifier for each song genre in the website
		GenreName	The title of the song's genre
		GenreDescription	Details of the song genre
Singer	All singers on the website	SingerID	Primary key. It is a unique identifier for each singer in the website
		SingerName	Show singer name
		SingerDescription	Song details
		SingerImage	Displaying song images on the website
Slide	Upload photos on the website	SlideID	Primary key. It is a unique identifier for each slide in the website
		SlideImage	Display images on the website



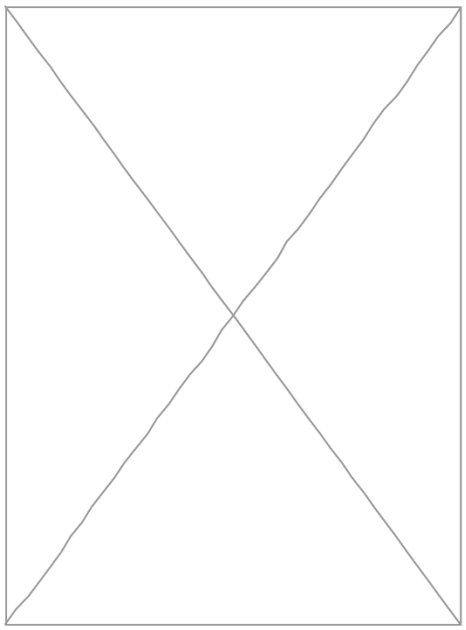
ERD diagrarm

IV. Project Execution (P7)

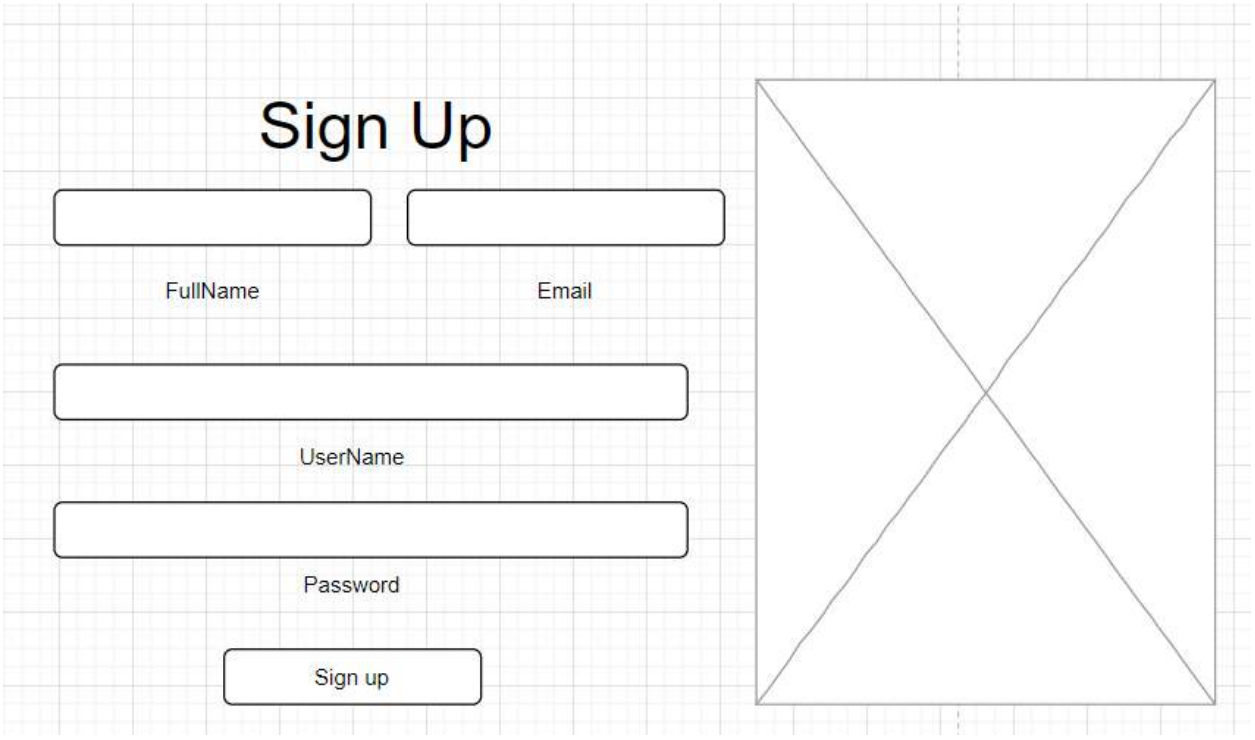
1. Website design interface (using wireframe)



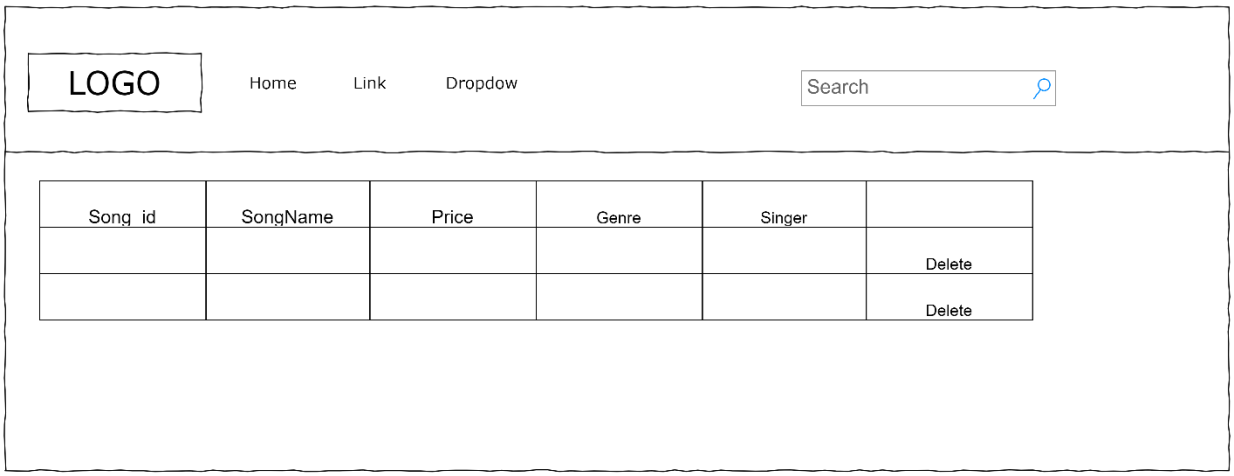
HomePage



Login



Register



Manager Song

Addsong

2. Architectural design

- I choose client server because:

- Client server has the ability to prevent network overload

- Client server ensures data integrity when something goes wrong
- Easy network expansion
- Just need to share the same communication format without the same platform to be able to work
- Client server allows integration of modern techniques such as GIS, object-oriented design model, ...
- With the Client server model, users can remotely access data, perform simple operations to send and receive files or search for information.

3. Database design

Bảng	Hành động	Hàng	Kiểu	Bảng mã đối chiếu	Kích thước	Tổng chi ph
genre	Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa	1	InnoDB	utf8mb4_general_ci	16,0 KiB	
orders	Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa	1	InnoDB	utf8mb4_general_ci	48,0 KiB	
singer	Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa	2	InnoDB	utf8mb4_general_ci	16,0 KiB	
slide	Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa	0	InnoDB	utf8mb4_general_ci	16,0 KiB	
song	Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa	4	InnoDB	utf8mb4_general_ci	48,0 KiB	
users	Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa	3	InnoDB	utf8mb4_general_ci	16,0 KiB	
6 bảng	Tổng	11	InnoDB	utf8mb4_general_ci	160,0 KiB	0

All table

	genre_id	genre_name	genre_description
Sửa Chép Xóa bỏ	1	Pop	NULL

Genre table

	order_id	user_id	song_id
Sửa Chép Xóa bỏ	1	2	6

Order table

	singer_id	singer_name	singer_description	singer_image
Sửa Chép Xóa bỏ	1	AMEE	NULL	NULL
Sửa Chép Xóa bỏ	2	Sơn Tùng-MTP	NULL	

Singer table

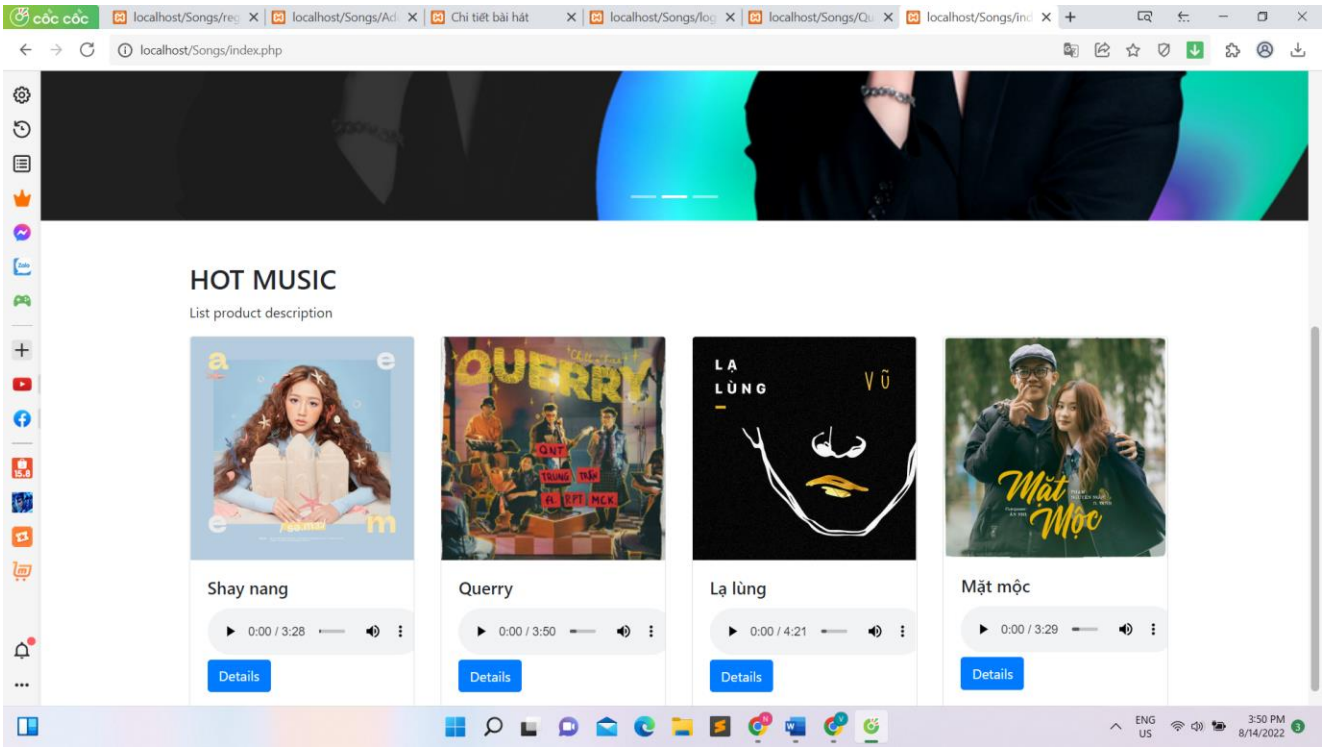
	song_id	song_name	song_description	song_price	song_audio	song_img	genre_id	singer_id
Sửa Chép Xóa bỏ	6	Shay nang		1000	Shay nắng.m4a	summer.jpg	1	1
Sửa Chép Xóa bỏ	7	Querry		2000	Querry.mp3	Querry.jpg	1	1
Sửa Chép Xóa bỏ	8	Lạ lùng		2000	La Lùng.mp3	1516930244148_640.jpg	1	1
Sửa Chép Xóa bỏ	9	Mặt mộc		1000	Mat-Moc-Pham-Nguyen-Ngoc-x-VAnh-x-An-Nhi.mp3	Matmoc.png	1	1

Song table

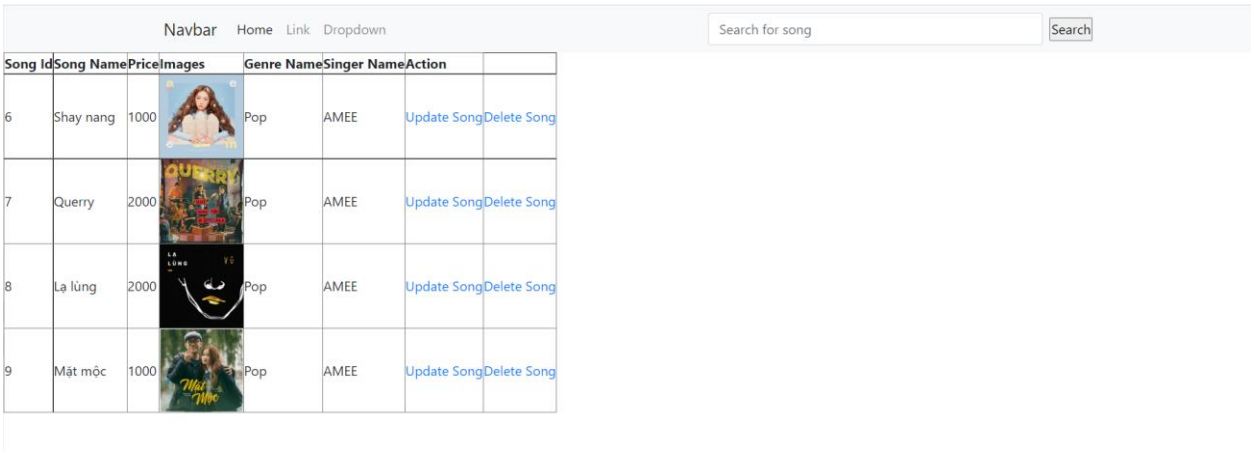
	user_id	username	password	fullname	email
Sửa Chép Xóa bỏ	1	Admin	Kiet17092003	Vũ Ngô Tuấn Kiệt	
Sửa Chép Xóa bỏ	2	Trường	123456	Hồ Trường	
Sửa Chép Xóa bỏ	3	Hna	123456	Ngọc Ánh	vntkwhite17092003@gmail.com

User table

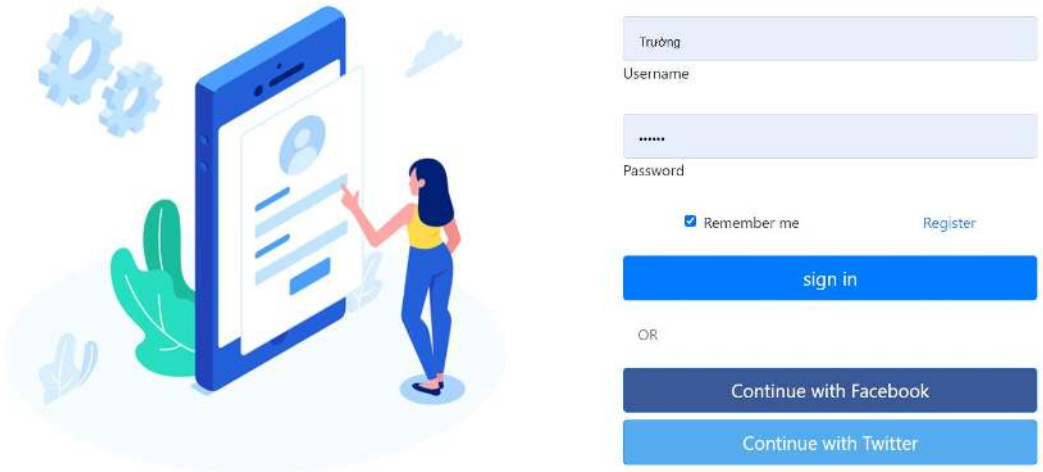
4. Description of the software done



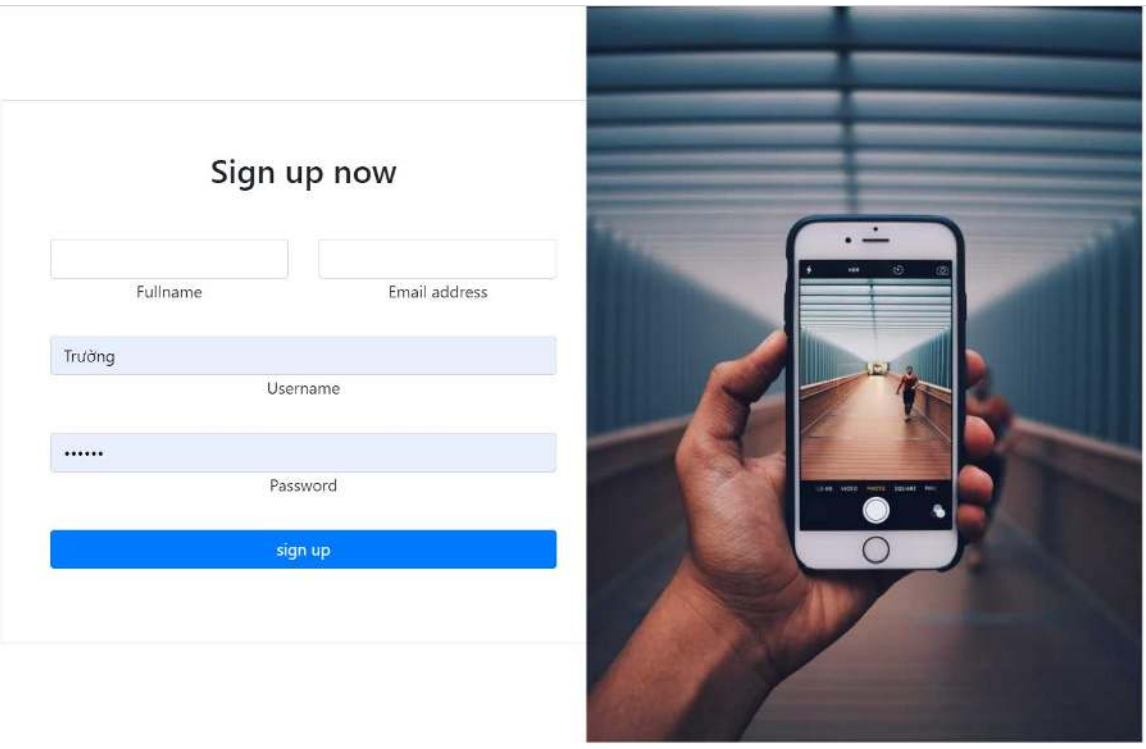
HomePage



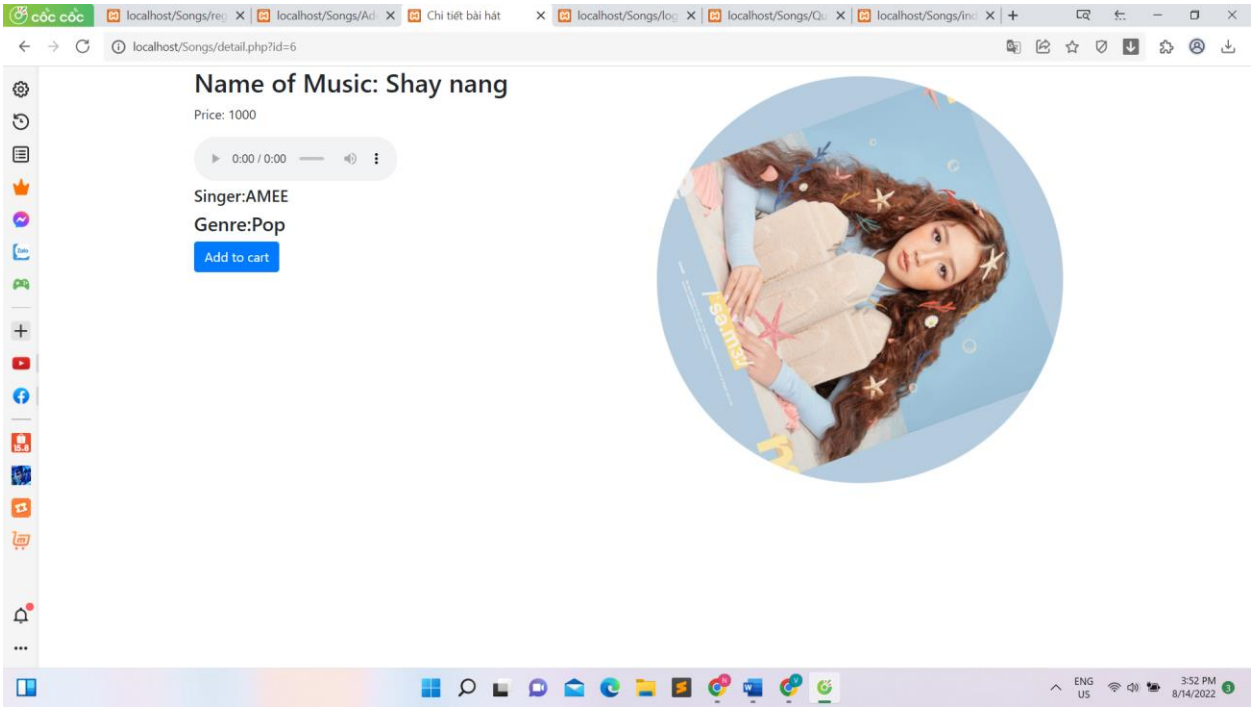
Manager



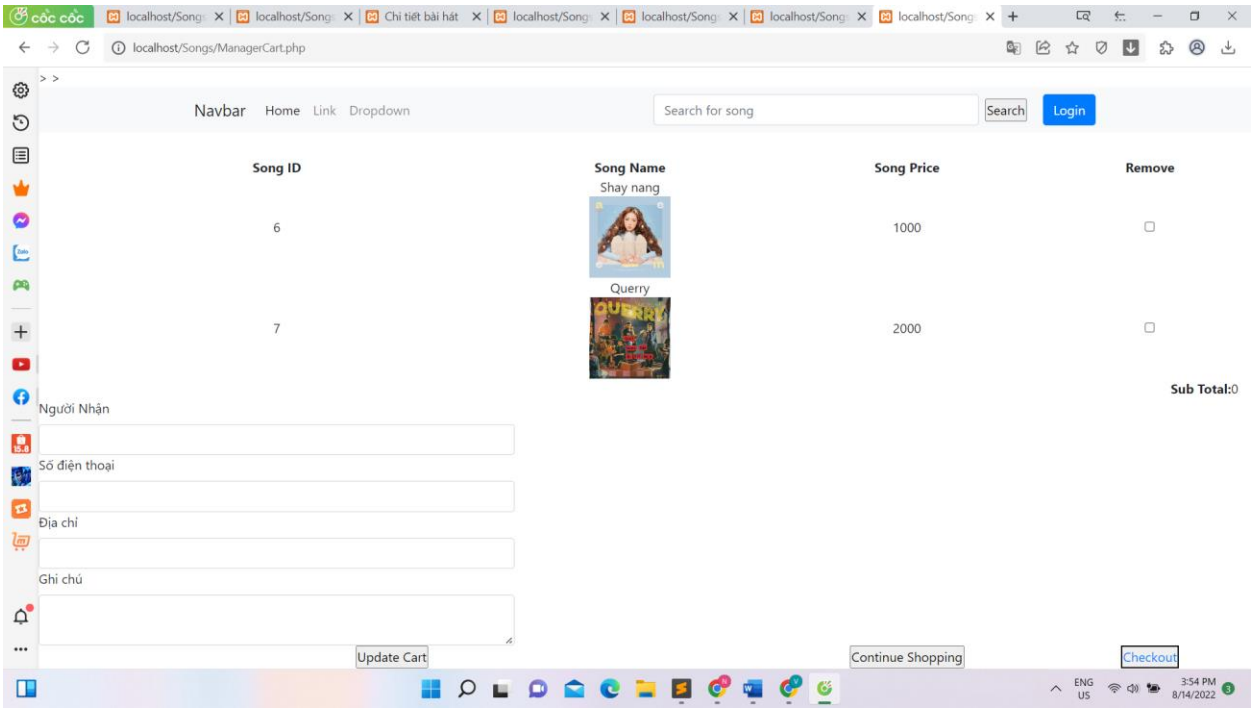
Login



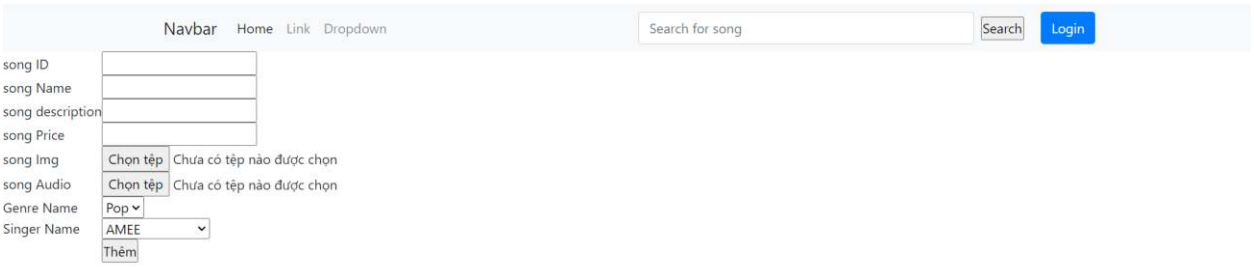
Register



Detail



Add and Manager Cart



AddSong

V.Conclution

- Through the report, I showed everyone what the requirements in the system are, classified the difference between functional and non-functional requirements and applied it to my web article. Besides that, I gave basic and applied concepts of diagrams

like Usecase, DFD, ERD, etc. Finally I made my project by wire frame, architecture design and database. material and finally the result I get, which is my website.

VI. Reference

Matthew Martin (2022). What is a Functional Requirement? Specification, Types, EXAMPLES. [online] Guru99.com. Available at: https://www.guru99.com/functional-requirement-specification-example.html [Accessed 26th Jun 2022].

BKNS.VN. (2020). *BKNS.VN*. [online] Available at: https://www.bkns.vn/client-server-la-gi.html.

Smartdraw (2019). *Entity Relationship Diagram (ERD) - What is an ER Diagram?* [online] Smartdraw.com. Available at: https://www.smartdraw.com/entity-relationship-diagram/.

Lucidchart (2017). *What is a Data Flow Diagram | Lucidchart*. [online] Lucidchart.com. Available at: https://www.lucidchart.com/pages/data-flow-diagram.