Student1279565



**FPT ACADEMY INTERNATIONAL**

**FPT – APTECH COMPUTER EDUCATION**

**Music Mp3**

|  |  |  |
| --- | --- | --- |
| **Center Name:** | FPT – Aptech – HCM2 | |
|  | Address:590 Cach Mang Thang Tam,    District 3, Ho Chi Minh City | |
| **Instructor:** | Mr. Hoang Duc Quang | |
| **Class:** | T1.2009.A0 | |
| **Group:** | 0 | |
| **Group members** | Name  Luu Thanh Sang | Student ID  Student1278914 | |
|  | Nguyen Duc Nam | Student1279119 | |
|  | Luu Huy Hoang | Student1279605 | |

Nguyen Duc Dung

**July-2022**

**This is to certify That**

**Mr./Ms.:**

**Luu Thanh Sang**

**Nguyen Duc Nam**

**Luu Huy Hoang**

**Nguyen Duc Dung**

Has successfully designed & developed

Project: Music-MP3

**Submitted by:**

Luu Thanh Sang

**Date of issue:**

**Authorized Signature:**

## ***Contents***

[**REVIEW I** 4](#_Toc110860272)

[1. ACKNOWLEDGE 5](#_Toc110860273)

[2. INTRODUCTION 5](#_Toc110860274)

[3. PROBLEM DEFINITION 6](#_Toc110860275)

[4. CUSTOMER'S REQUIREMENTS SPECIFICATIONS 6](#_Toc110860276)

[5. SCOPE OF WORK 8](#_Toc110860277)

[6. SYSTEM REQUIREMENTS SPECIFICATIONS 8](#_Toc110860278)

[7. TASK SHEET REVIEW I 10](#_Toc110860279)

[**REVIEW II** 11](#_Toc110860280)

[1. USE CASES 12](#_Toc110860281)

[2. USE CASE ADMIN 32](#_Toc110860282)

[3. DATA FLOW DIAGRAM 56](#_Toc110860283)

[4. LOGICAL ENTITY RELATIONSHIP DIAGRAM 58](#_Toc110860284)

[5. ENTITY PROPERTIES 59](#_Toc110860285)

[6. DATABASE DIAGRAM 62](#_Toc110860286)

[7. TABLE DESIGN 63](#_Toc110860287)

[8. TASK SHEET REVIEW II 68](#_Toc110860288)

[**REVIEW III** 69](#_Toc110860289)

[1. DESIGN – FRONT-END: 70](#_Toc110860290)

[2. TASK SHEET REVIEW III 73](#_Toc110860291)

## **REVIEW I**

### 1. ACKNOWLEDGE

Our group would like to devote this chapter to express our sincere appreciation to all those who offered us valuable assistance during in carrying out this project.

First,we would like to express our sincere thanks to our dedicated guide Mr.Hoang Duc Quang who gave us advice and supported us to overcome not only technological challenges but also psychological stresses.Without him,our eProject would not have been fully complete and excellent

Thirdly, we would also like to express my gratitude to all the staff members of my center for not only providing me with the opportunity to work with them on this project, but also for their support and encouragement throughout the process.

Finally yet importantly, we wish to say a personal thank you to all our dear classmates and team members who are always willing to support us openly and enthusiastically.

### 2. INTRODUCTION

We are in area where information technology makes listening music and buying music products easier than ever. People can listen everything they want with just one telephone and everything is completed electronically and interactively in real- time. Therefore, we develop a website called Music Mp3 to bring good times and interesting experience to more customers .

Music Mp3 is an online music site where customers can access and listen to the new song , new album,...,buy music products easily without taking too much time.Moreover, It is a place that provide quality songs,products that make users satisfied when listen . And It will provide its users the following features.

For Music Mp3 customer:

* Customer can register account.
* Customer can login and manage their account.
* Customer can search any songs about pop,jazz....
* Customer can search song by name.
* Customer can see songs’ introduction before listen.
* Customer can see MV.
* Customer can see products’ introduction before buying.
* Customer can buy products after adding products to cart.
* Customer can add product to wish list.
* Customer can view cart history.
* Customer can check out with PayPal checkout.
* Customer can view news.

### 3. PROBLEM DEFINITION

After analyzing the project specifications of Music Mp3, to fulfil customers’s requirements properly, we came to conclusion that in order to achieve the best result, we should deal with following queries:

*From the view point of CUSTOMERS*

* A homepage will show the main song, suitable logo, display the logo on all the pages in the site.
* A menu that will include all the functions implemented in the website.
* User can listen songs, view song detail, view MV,buy music products, view cart history and check out after login their account.

*From the view point of the Administrator*

* A login page will grant access Music Mp3 admin.
* Admin can view song.
* Admin can add song.
* Admin can update song.
* Admin can delete song.
* Admin can find song.
* Admin can view products.
* Admin can add products.
* Admin can update products.
* Admin can delete products.
* Admin can find products.
* Admin can view customer account.
* Admin can add customer account.
* Admin can update customer account.
* Admin can delete customer account.
* Admin can find customer account.
* Admin can view news.
* Admin can add news.
* Admin can update news.
* Admin can delete news.
* Admin can find news.

### 4. CUSTOMER'S REQUIREMENTS SPECIFICATIONS

CLIENT: MUSIC MP3

BUSINESS/PROJECT OBJECTIVE:

The project is to design an online website for listening mucic and buying music products

***Input to the system:***

**Music Mp3 administrators need to provide:**

* Information required in order creating account to access to the admin system.
* Songs and products music are going to be shown on the website.

**Music Mp3 customers need to provide:**

* Personal information to buy products.

***Output from the system:***

**Music Mp3 administrators will be able to:**

* Login/Logout to the admin system.
* Manage view song.
* Manage add song.
* Manage update song.
* Manage delete song.
* Manage find song.
* Manage view products.
* Manage add products.
* Manage update products.
* Manage delete products.
* Manage find customer products.
* Manage view customer account.
* Manage add customer account.
* Manage update customer account.
* Manage delete customer account.
* Manage find customer account.
* Manage view order.
* Manage update order.
* Manage view news.
* Manage add news.
* Manage update news.
* Manage delete news.
* Manage find news.

**Music Mp3 guide will be able to:**

* Register customer account.
* Buy products after add information.
* View products that they have chosen.

**Music Mp3 customer will be able to:**

* Register account.
* Login/Logout to the account.
* Access account (change profile, email, password…)
* Search song they want to listen by name, album, genre...
* Add product to wish list.
* Add product to cart
* Checkout.
* View purchase history.
* View News

### 5. SCOPE OF WORK

The application should perform the following functions:

Login**:**

There are 2 types of account in the application: administrator, Customers accounts.

* Administrator will login as “admin”.
* Customers will login as “user”.

**Buying products:**

* When customer choose a product and the selected product will be added to the cart.
* Customer can buy products at shopping cart after pressing the “Checkout” button and fill in the information.

**Database:**

* The information about song, customers, news will be stored in the database.
* The information from the contact us page.

### 6. SYSTEM REQUIREMENTS SPECIFICATIONS

HARDWARE/ SOFTWARE REQUIREMENTS:

**Hardware**

*Web Server*

|  |  |
| --- | --- |
| **Processor** | Intel Core I5 or higher. |
| **Memory** | 4 GB RAM or greater. |
| **Modem/ADSL** | Internet access is required. |

*Client*

|  |  |
| --- | --- |
| **Processor** | Intel Core I3 or higher. |
| **Memory** | 2 GB RAM or greater. |
| **Monitor** | Super VGA (1024x768) or higher resolution. |
| **Modem/ADSL** | Internet access is required. |

**Software**

*Web Server*

|  |  |
| --- | --- |
| **Operation System** | Window 7 or higher |
| **Browser** | Google Chrome version 35 or Microsoft Edge |
| **Database** | SQL Server. |
| **Software** | Internet Information Service |
| Java – Spring Boot framework |

*Client*

|  |  |
| --- | --- |
| **Operation System** | Window 7 or higher. |
| **Browser** | Google Chrome version or Microsoft Edge |

**Development software**

* Microsoft Windows 10.
* SQL Server.
* NetBeans.
* Google Chrome.

**Technology**

* Java - Spring Boot framework.
* JQuery.
* Bootstrap.
* ThymeLeaf

HTML/CSS/JS

VERSION CONTROL SYSTEMS: GIT

### 7. TASK SHEET REVIEW I

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Project Ref.  No: 3 | | Project Name | Activity Plan Prepared By | Date of Preparation of Activity Plan: | | | | |
| Sr./No. | Task | Actual Start Date | Actual Days | Team member name | Status |
| 1 | Acknowledgment | Music Mp3 | All Team | 28 July 2022 | 10 | Hoang | Finished |
| 2 | Introduction | 28 July 2022 | 10 | Hoang |
| 3 | Problem Definition | 28 July 2022 | 10 | Hoang |
| 4 | Customer Requirement Specification | 28 July 2022 | 10 | Hoang |
| 5 | Scope of Work | 28 July 2022 | 10 | Hoang |
| 6 | System Requirement Specification | 28 July 2022 | 10 | Hoang |
| 7 | Task Sheet | 28 July 2022 | 10 | Hoang |

|  |  |
| --- | --- |
| Date: 28 July 2022 | |
| Signature of Instructor:  **Mr. Hoang Duc Quang** | Signature of Team Leader:  **Luu Thanh Sang** |

## **REVIEW II**

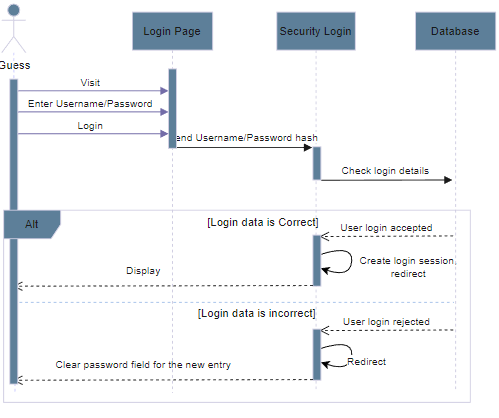
### 1. USE CASES

#### HOME PAGE

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.1 | |
| **Use Case Name** | Home Page | |
| **Author** | Nam | |
| **Actors** | Guest - User | |
| **Description** | Guest - User view home page | |
| **Trigger** | Guest - User access on homepage | |
| **Pre-conditions** | The guest’s device is already connected to the internet | |
| **Post-conditions** | Success: Show homepage | |
| Fail: No page is shown | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to homepage | 2. System shows the homepage |

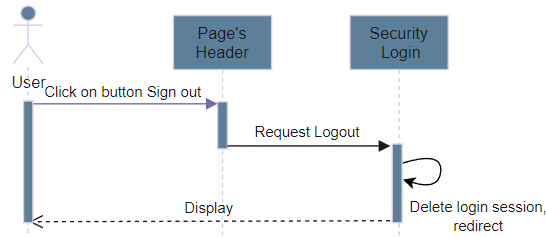
#### SIGN UP

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.2 | |
| **Use Case Name** | Sign up | |
| **Author** | Sang | |
| **Actors** | User | |
| **Description** | Guess create an account to login Muzik page | |
| **Trigger** | Guess want to create an account | |
| **Pre-conditions** | The guess’s device is already connected to the internet | |
| **Post-conditions** | Success: A account is created successfully | |
| Fail: Email exist. | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Home page and click on Sign up button  3. Actor fill the information and click on “Sign up” button | 2. System shows the Sign up form with following controls:  - “First Name” text field  - “Last Name” text field  - “Email” text field  - “Password” text field  - “Sign up” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “First Name” text field blank | First Name cannot be blank |
| Actor left “Last Name” text field | Last Name cannot be blank |
| Actor left “Email” text field | Email cannot be blank |
| Actor left “Password” text field blank | Password cannot be blank |
| Actor input an existed Email | Email already registered !!! |



#### LOGIN

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.3 | |
| **Use Case Name** | Login | |
| **Author** | Sang | |
| **Actors** | Admin - User | |
| **Description** | Admin login to admin system. User login to user system | |
| **Trigger** | Admin want to login to admin system. User login to user system | |
| **Pre-conditions** | The device of admin or user is already connected to the internet | |
| **Post-conditions** | Success: Admin-User login successfully | |
| Fail: Admin- User login fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to admin system  3. Actor fill the information and click on “Login” button | 2. System shows the login form with following controls:  - “Username” text field  - “Password” text field  - “Login” button  -“Sign up” link  -“[Forgot your password?](http://localhost:8080/resetPassword)” link  4. System validates the information  5. System redirect to admin system |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Username” text field blank | Username cannot be blank |
| Actor left “Password” text field blank | Password cannot be blank |
| Actor input an incorrect login details | Username or Password is incorrect |
| Account is not active | Account is not active |

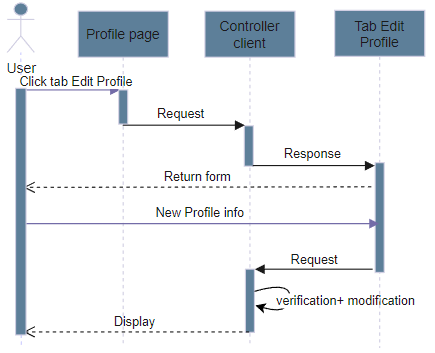


#### LOG OUT

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.4 | |
| **Use Case Name** | Log out | |
| **Author** | Sang | |
| **Actors** | Admin - User | |
| **Description** | Admin log out of admin system. User log out of admin system | |
| **Trigger** | Admin want to log out of admin system. User want to log out of admin system | |
| **Pre-conditions** | The admin or User’s device is already connected to the internet  Admin has already log in to admin system | |
| **Post-conditions** | Success: Log out successfully | |
| Fail: Log out fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Log out link | 2. System redirect to log out page of system. |

#### PROFILE

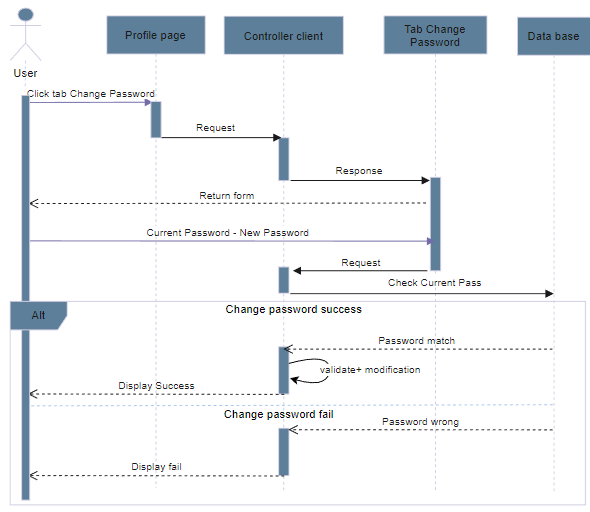
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.5 | |
| **Use Case Name** | Profile | |
| **Author** | Sang | |
| **Actors** | User | |
| **Description** | Show 3 tab profile, update profile, change password | |
| **Trigger** | User want to go profile | |
| **Pre-conditions** | The user’s device is already connected to the internet  User has already log in to user system | |
| **Post-conditions** | Success: User access profile page successfully | |
| Fail: User can’t access profile page | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Muzik’s web app after login and click on User’s image then click My Profile. | 2. System shows the active tab profiles  with actor’s profile infomation. |



#### UPDATE PROFILE

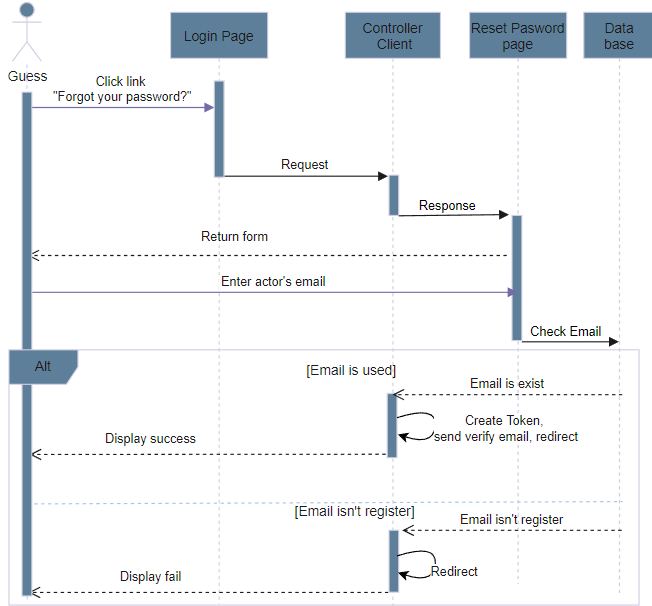
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.6 | |
| **Use Case Name** | Update Account | |
| **Author** | Sang | |
| **Actors** | User | |
| **Description** | Show User’s profile and update profile | |
| **Trigger** | User want to update profile | |
| **Pre-conditions** | The user’s device is already connected to the internet  User has already log in to user system | |
| **Post-conditions** | Success: User’s account is update successfully | |
| Fail: User’s account is update fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Muzik’s web app after login and click on User’s image then click My Profile, and click tab Edit Profile  3. Actor fill the information and click on “Submit” button | 2. System shows the update profiles form with following controls:  - “Image” file field  - “First Name” text field  - “Last Name” text field  - “Username” text field read only  - “Submit” button  - “Cancel” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “First Name” text field blank | Full Name cannot be blank |
| Actor left “Last Name” text field | Email cannot be blank |

#### CHANGE PASSWORD



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.7 | |
| **Use Case Name** | Change Account | |
| **Author** | Sang | |
| **Actors** | User | |
| **Description** | Show User’s profile and change password | |
| **Trigger** | User want to change password | |
| **Pre-conditions** | The user’s device is already connected to the internet  User has already log in to user system | |
| **Post-conditions** | Success: User’s account is change password successfully | |
| Fail: User’s account is change password fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Muzik’s web app after login and click on User’s image then click My Profile, and click tab Change Password  3. Actor fill the information and click on “Submit” button | 2. System shows the profile/change password form with following controls:  - “Current Password” text field  - “New Password” text field  - “Submit” button  - “Cancel” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Current Password” text field blank | Current Password cannot be blank |
| Actor left “New Password” text field | New Password cannot be blank |

#### RESET PASSWORD



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.8 | |
| **Use Case Name** | Reset password | |
| **Author** | Sang | |
| **Actors** | User | |
| **Description** | User forgot password and reset password | |
| **Trigger** | User forgot password and want to login | |
| **Pre-conditions** | The user’s device is already connected to the internet  User has already log in to user system | |
| **Post-conditions** | Success: User’s account receive an verify email from system | |
| Fail: User’s account is reset password fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to login page and click on “Forgot your password” link  3. Actor fill the information and click on “Reset password” button | 2. System shows the reset page with following controls:  - “Email adress” text field  - “Reset password” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Email adress” text field blank | Email adress cannot be blank |
| Actor input an Email doesn’t Exist | Email address has not been used yet |

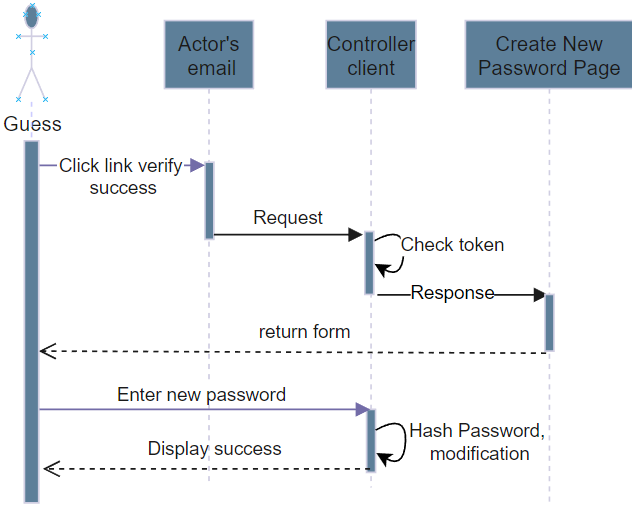
#### VERIFY EMAIL

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.9 | |
| **Use Case Name** | Verify Email | |
| **Author** | Sang | |
| **Actors** | Guess | |
| **Description** | Actor verify email to create new account or reset password | |
| **Trigger** | Actor go to email verify email to create new account or reset password | |
| **Pre-conditions** | The guess’s device is already connected to the internet | |
| **Post-conditions** | Success: Actor’s account is actived or Actor redirect to page create new password | |
| Fail: Actor’s verify email expired | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to his/her email and click link verify email  3. Actor fill the information and click on “Verify” link | 2. Actor’s email with following controls:  - “Verify” link  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor let verify email expired | Go to page message error |

#### RESEND

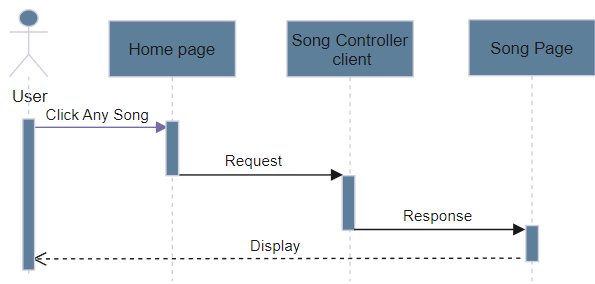
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.10 | |
| **Use Case Name** | Resend | |
| **Author** | Sang | |
| **Actors** | Guess | |
| **Description** | Actor verify email but the link expired | |
| **Trigger** | Actor want to verify email and too late | |
| **Pre-conditions** | The Actor’s device is already connected to the internet | |
| **Post-conditions** | Success: New verify emaid is sent | |
|
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to the Verify link in email is sent from system and the link expired. | 2. System shows page message with following controls:  - “Resend Verify Email” Link |

#### CREATE NEW PASSWORD



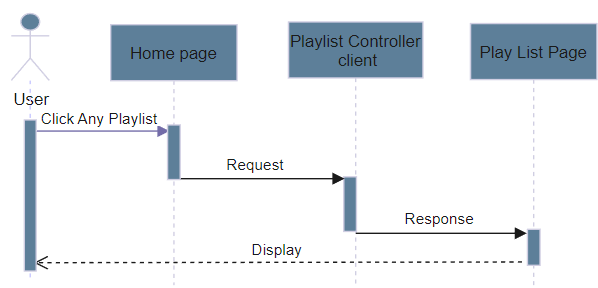
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.10 | |
| **Use Case Name** | Create new password | |
| **Author** | Sang | |
| **Actors** | Guess | |
| **Description** | Actor forgot password and create new password | |
| **Trigger** | Actor want to reset password | |
| **Pre-conditions** | The Actor’s device is already connected to the internet | |
| **Post-conditions** | Success: Actor’s account create new password successfully | |
| Fail: Actor’s account create new password fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to the Verify link in email is sent from system  3. Actor fill the information and click on “Submit” button | 2. System shows the actor’s email and create new password form with following controls:  - “Email” text field read only  - “New Password” text field  - “Submit” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “New Password” text field | New Password cannot be blank |
|

#### SONG



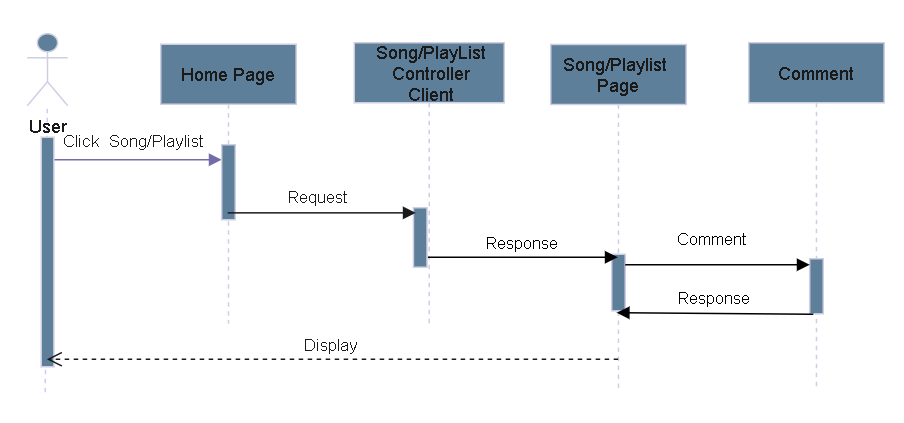
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.11 | |
| **Use Case Name** | Song | |
| **Author** | Nam | |
| **Actors** | Guess , User | |
| **Description** | Guess and User listen to song | |
| **Trigger** | Guess and User listen to song online on web | |
| **Pre-conditions** | The guess’s device is already connected to the internet | |
| **Post-conditions** | Success: Song Player Interface Uploaded | |
| Fail: Request Not Found | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to song player page when click on any song | 2. System will return the requested song on the song playback page |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| The user requested a song that no longer exists | Go to page message error |

#### PLAYLIST



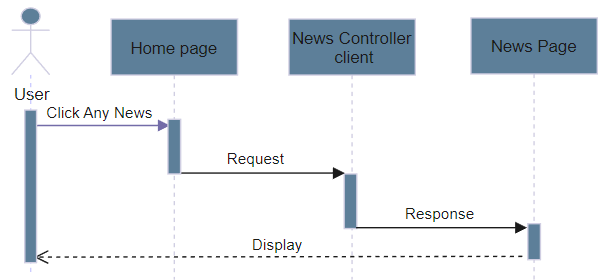
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.12 | |
| **Use Case Name** | Playlist | |
| **Author** | Nam | |
| **Actors** | User | |
| **Description** | Users Manage Song Playlists | |
| **Trigger** | Users Want To Manage Personal Playlists | |
| **Pre-conditions** | The guess’s device is already connected to the internet | |
| **Post-conditions** | Success: List of personal song playlist page loaded | |
| Fail: Request Not Found | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor click Add Song To Play List button ,select specific playlist name and submit form . | 2. System will get ID of playlist requested and specific song ,finally add data to database |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| The Actor requested a song and playlist that no longer exists | Display message error |

#### COMMENT



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.13 | |
| **Use Case Name** | Comment | |
| **Author** | Nam | |
| **Actors** | User | |
| **Description** | Users can comment | |
| **Trigger** | Users can comment below the song they are interested in | |
| **Pre-conditions** | The guess’s device is already connected to the internet | |
| **Post-conditions** | Success: List of personal song playlist page loaded | |
| Fail: Request Not Found | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor leave comments below the song and submit | 2. System will get data requested ,add to database and refresh result list comments to actor |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor leave empty comments  Actor leave comments on songs that no longer exist | System does not process this case  Go to page message error |

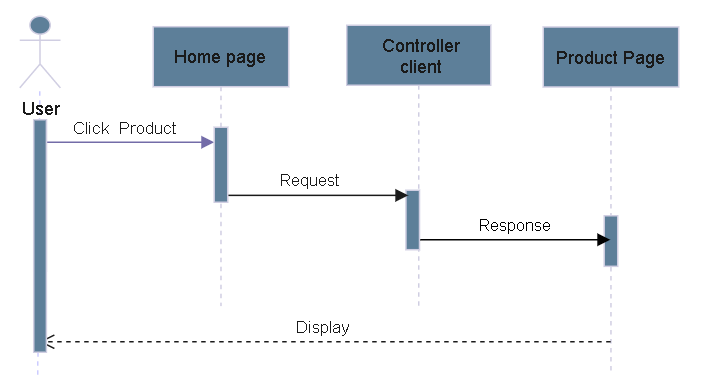
#### NEWS



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.14 | |
| **Use Case Name** | News | |
| **Author** | Dung | |
| **Actors** | User | |
| **Description** | Users Read The News | |
| **Trigger** | Users Can read the news and details | |
| **Pre-conditions** | The guess’s device is already connected to the internet | |
| **Post-conditions** | Success: List of news page loaded | |
| Fail: Request Not Found | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor click News To show the list of news ,select article title to view details . | 2. System will transfer from home to news list, then system will get ID of the news and show details of this news. |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| The actor requested a news that no longer exists | Display message error |

#### 

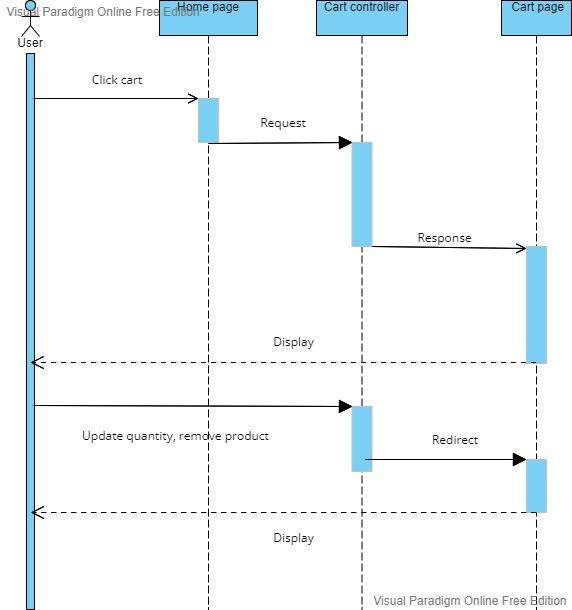
#### PRODUCT



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.12 | |
| **Use Case Name** | Product | |
| **Author** | Hoang | |
| **Actors** | User | |
| **Description** | Users View Products | |
| **Trigger** | Users Can View Products and Product Details | |
| **Pre-conditions** | The guess’s device is already connected to the internet | |
| **Post-conditions** | Success: List item of product page loaded | |
| Fail: Request Not Found | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actorcan click Detail Icon button | 2. System will get ID of product requested and display product detail page |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| The Actor requested a product that no longer exists | Display message error |

#### 

#### CART



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.15 | |
| **Use Case Name** | Cart | |
| **Author** | Dung | |
| **Actors** | User | |
| **Description** | Users View the cart | |
| **Trigger** | Users can view the cart after adding products to cart | |
| **Pre-conditions** | The guess’s device is already connected to the internet | |
| **Post-conditions** | Success: Cart page loaded | |
| Fail: Request Not Found | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor click Cart  2. Actor choose product to update quantity, delete  3. Actor click checkout. | 2. System will transfer from home to cart page, if cart doesn’t have product, system will show message “cart is empty”, else system will show list of product in the cart.  3. System will update subtotal, total price and remove product from cart.  4. System will transfer to paypal page to payment |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| The actor requested a cart that doesn’t have product | Display message “cart is empty” |

#### PURCHASE



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.16 | |
| **Use Case Name** | Purchase | |
| **Author** | Dung | |
| **Actors** | User | |
| **Description** | Users click Purchase | |
| **Trigger** | Users click Purchase after checking a cart | |
| **Pre-conditions** | The guess’s device is already connected to the internet | |
| **Post-conditions** | Success: Show message payment success | |
| Fail: Request Not Found | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor click buy to purchase and redirect to paypal. | 2. System will transfer from cart to purchase page. |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| The Actor requested a payment that no longer exists | Display message error |

#### PAYPAL

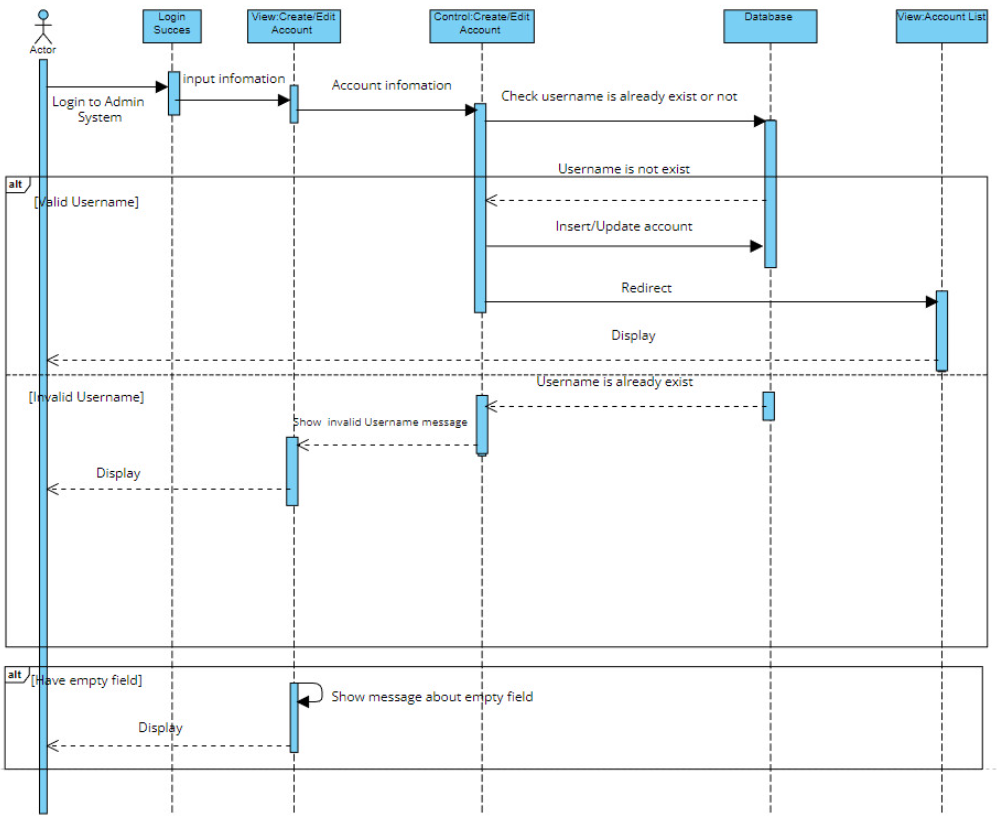
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1.17 | |
| **Use Case Name** | Paypal | |
| **Author** | Sang | |
| **Actors** | User | |
| **Description** | User use Paypal to pay money | |
| **Trigger** | User want to use Paypal to pay money | |
| **Pre-conditions** | The Actor’s device is already connected to the internet  Actor must be login in system | |
| **Post-conditions** | Success: Purchase succ | |
| Fail: Request Not Found | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor login paypal and make payment | 2. Paypal show login paypal. |

#### 

### 2. USE CASE ADMIN



#### CREATE/EDIT ACCOUNT



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.0, UC-2.1 | |
| **Use Case Name** | Create/Edit Account | |
| **Author** | Sang | |
| **Actors** | Mod/Admin | |
| **Description** | Mod/Admin create or edit an account subject | |
| **Trigger** | Mod/Admin want to create/edit an account | |
| **Pre-conditions** | The actor’s device is already connected to the internet  actor has already log in to admin system | |
| **Post-conditions** | Success: A account is created/edited successfully | |
| Fail: Account is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to account management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Full Name” text field  - “Username” text field  - “Password” text field  - “Role” select box  - “Submit” and “Reset” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Full Name” text field blank | Full Name cannot be blank |
| Actor left “Username” text field | Username cannot be blank |
|
| Actor left “Password” text field blank | Password cannot be blank |
| Actor left “Role” select box | Role cannot be blank |
| Actor input an existed username | Username is existed. Please choose another one |
|

#### **LOGIN**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.2 | |
| **Use Case Name** | Login | |
| **Author** | Sang | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Actors login to admin system | |
| **Trigger** | Actors want to login to admin system | |
| **Pre-conditions** | The Actors’s device is already connected to the internet | |
| **Post-conditions** | Success: Actors login successfully | |
| Fail: Actors login fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to admin system  3. Actor fill the information and click on “Login” button | 2. System shows the login form with following controls:  - “Username” text field  - “Password” text field  - “Login” button  4. System validates the information  5. System redirect to admin system |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Username” text field blank | Username cannot be blank |
| Actor left “Password” text field blank | Password cannot be blank |
| Actor input an incorrect login details | Username or Password is incorrect |
| Account is not active | Account is not active |

#### **LOG OUT**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.3 | |
| **Use Case Name** | Log out | |
| **Author** | Sang | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Actors log out of admin system | |
| **Trigger** | Actors want to log out of admin system | |
| **Pre-conditions** | The actors’s device is already connected to the internet  Actors has already log in to admin system | |
| **Post-conditions** | Success: Log out successfully | |
| Fail: Log out fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Log out link | 2. System redirect to login page of admin system. |

#### **PROFILE/UPDATE PROFILE**

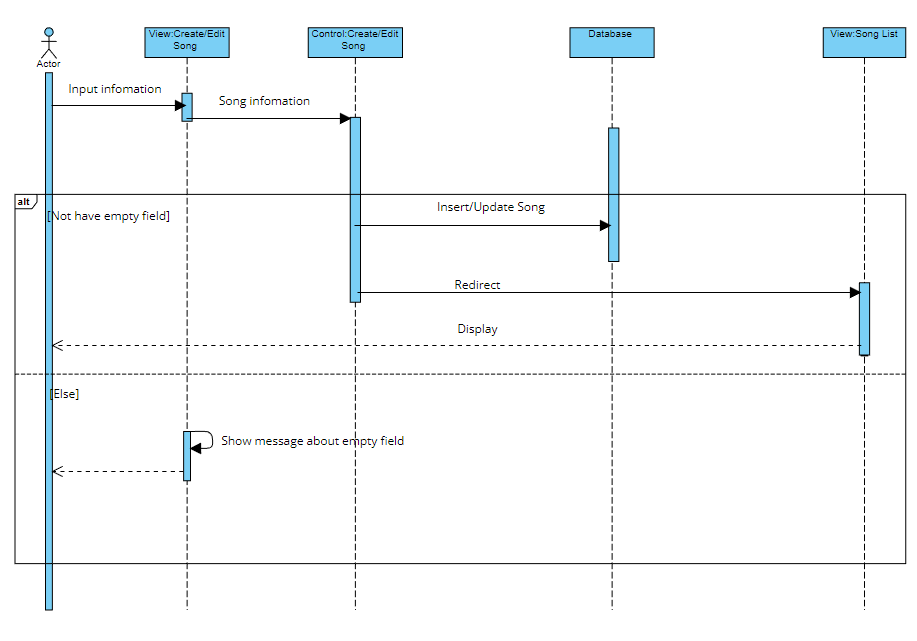
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.4 | |
| **Use Case Name** | Profile/Update Account | |
| **Author** | Sang | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Show Actors’s profile and update profile | |
| **Trigger** | Actors want to update profile | |
| **Pre-conditions** | The Actors’s device is already connected to the internet  Actors has already log in to user system | |
| **Post-conditions** | Success: Actors’s account is update successfully | |
| Fail: Actors’s account is update fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Muzik’s web app after login and click on User’s image then click My Profile.  3. Actor fill the information and click on “Submit” button | 2. System shows the update profiles form with following controls:  - “Full Name” text field  - “Password” text field  - “Username” text field read only  - “Role” text field read only  - “Submit” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “First Name” text field blank | Full Name cannot be blank |
|

#### **CREATE/EDIT NEWS**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.5 | |
| **Use Case Name** | Create/Edit News | |
| **Author** | Dung | |
| **Actors** | Mod/Admin | |
| **Description** | Mod/Admin create or edit a news subject | |
| **Trigger** | Mod/Admin want to create/edit a news | |
| **Pre-conditions** | The actor’s device is already connected to the internet | |
| **Post-conditions** | Success: A news is created/edited successfully | |
| Fail: A news is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to news management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Title” text field  - “Content” text field  - “Description” text field  - “Image” choose file  - “Img\_title” text field  - “View” text field  - “Created\_at” choose date  - “Submit” and “Reset” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor left “Title” text field blank | Title cannot be blank |
| Actor left “Img\_title” text field | Img\_title cannot be blank |
|
| Actor left “Image” choose file | Image must be chosen |

#### **DELETE NEWS**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.6 | |
| **Use Case Name** | Delete news | |
| **Author** | Dung | |
| **Actors** | Admin | |
| **Description** | Admin delete a news | |
| **Trigger** | Admin wants to delete a news | |
| **Pre-conditions** | The admin’s device is already connected to the internet | |
| **Post-conditions** | Success: A news is deleted successfully | |
| Fail: A news is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to news management and click on Delete button | 2. System delete the correspond record in database |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor requested a news that no longer exists | Display error message |



#### **CREATE/EDIT SONG**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.7 | |
| **Use Case Name** | Create/Edit Song | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor create or edit a song | |
| **Trigger** | Mod/Admin/Editor want to create/edit a song | |
| **Pre-conditions** | The actor’s device is already connected to the internet | |
| **Post-conditions** | Success: A song is created/edited successfully | |
| Fail: A song is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to song management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Name” text field  - “Audio” choose file  - “Video” choose file  - “Artist” select field  - “Album” select field  - “Lyric” text field  - “Submit” and “Reset” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor leave empty information | All field cannot be blank |

#### **DELETE SONG**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.8 | |
| **Use Case Name** | Delete song | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor delete song | |
| **Trigger** | Mod/Admin/Editor wants to delete a song | |
| **Pre-conditions** | The author device is already connected to the internet | |
| **Post-conditions** | Success: A song is deleted successfully | |
| Fail: A song is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to song management and click on Delete button | 2. System delete the correspond record in database |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor requested a song that no longer exists | Display error message |

#### **CREATE/EDIT ALBUM**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2.9 | |
| **Use Case Name** | Create/Edit Album | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor create or edit a album | |
| **Trigger** | Mod/Admin/Editor want to create/edit a album | |
| **Pre-conditions** | The actor’s device is already connected to the internet | |
| **Post-conditions** | Success: A album is created/edited successfully | |
| Fail: A album is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to album management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Name” text field  - “Image” choose file  - “Release Date” choose file  - “Artist” select field  - “Submit” and “Reset” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor leave empty information | All field cannot be blank |

#### **DELETE ALBUM**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3.0 | |
| **Use Case Name** | Delete album | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor delete album | |
| **Trigger** | Mod/Admin/Editor wants to delete a album | |
| **Pre-conditions** | The author device is already connected to the internet | |
| **Post-conditions** | Success: A album is deleted successfully | |
| Fail: A album is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to album management and click on Delete button | 2. System delete the correspond record in database |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor requested a album that no longer exists | Display error message |

#### **CREATE/EDIT ARTIST**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3.1 | |
| **Use Case Name** | Create/Edit Artist | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor create or edit a artist | |
| **Trigger** | Mod/Admin/Editor want to create/edit a artist | |
| **Pre-conditions** | The actor’s device is already connected to the internet | |
| **Post-conditions** | Success: A artist is created/edited successfully | |
| Fail: A artist is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to artist management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Name” text field  - “Image” choose file  - “Description” text field  - “Submit” and “Reset” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor leave empty information | All field cannot be blank |

#### **DELETE ARTIST**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3.1 | |
| **Use Case Name** | Delete artist | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor delete artist | |
| **Trigger** | Mod/Admin/Editor wants to delete a artist | |
| **Pre-conditions** | The author device is already connected to the internet | |
| **Post-conditions** | Success: A artist is deleted successfully | |
| Fail: A artist is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to artist management and click on Delete button | 2. System delete the correspond record in database |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor requested a artist that no longer exists | Display error message |

#### **CREATE/EDIT GENRE**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3.2 | |
| **Use Case Name** | Create/Edit Genre | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor create or edit a genre | |
| **Trigger** | Mod/Admin/Editor want to create/edit a genre | |
| **Pre-conditions** | The actor’s device is already connected to the internet | |
| **Post-conditions** | Success: A genre is created/edited successfully | |
| Fail: A genre is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to genre management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Name” text field  - “Submit” and “Reset” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor leave empty information | All field cannot be blank |

#### **DELETE GENRE**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3.3 | |
| **Use Case Name** | Delete artist | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor delete genre | |
| **Trigger** | Mod/Admin/Editor wants to delete a genre | |
| **Pre-conditions** | The author device is already connected to the internet | |
| **Post-conditions** | Success: A genre is deleted successfully | |
| Fail: A genre is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to genre management and click on Delete button | 2. System delete the correspond record in database |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor requested a genre that no longer exists | Display error message |

#### **CREATE/EDIT SUBTITLE**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3.4 | |
| **Use Case Name** | Create/Edit Subtitle | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor create or edit a subtitle | |
| **Trigger** | Mod/Admin/Editor want to create/edit a subtitle | |
| **Pre-conditions** | The actor’s device is already connected to the internet | |
| **Post-conditions** | Success: A subtitle is created/edited successfully | |
| Fail: A subtitle is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to subtitle management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit facilities form with following controls:  - “Name” text field  - “VTT” choose file  - “Submit” and “Reset” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor leave empty information | All field cannot be blank |

#### **DELETE SUBTITLE**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3.5 | |
| **Use Case Name** | Delete subtitle | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor delete subtitle | |
| **Trigger** | Mod/Admin/Editor wants to delete a subtitle | |
| **Pre-conditions** | The author device is already connected to the internet | |
| **Post-conditions** | Success: A subtile is deleted successfully | |
| Fail: A genre is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to genre management and click on Delete button | 2. System delete the correspond record in database |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor requested a subtitle that no longer exists | Display error message |

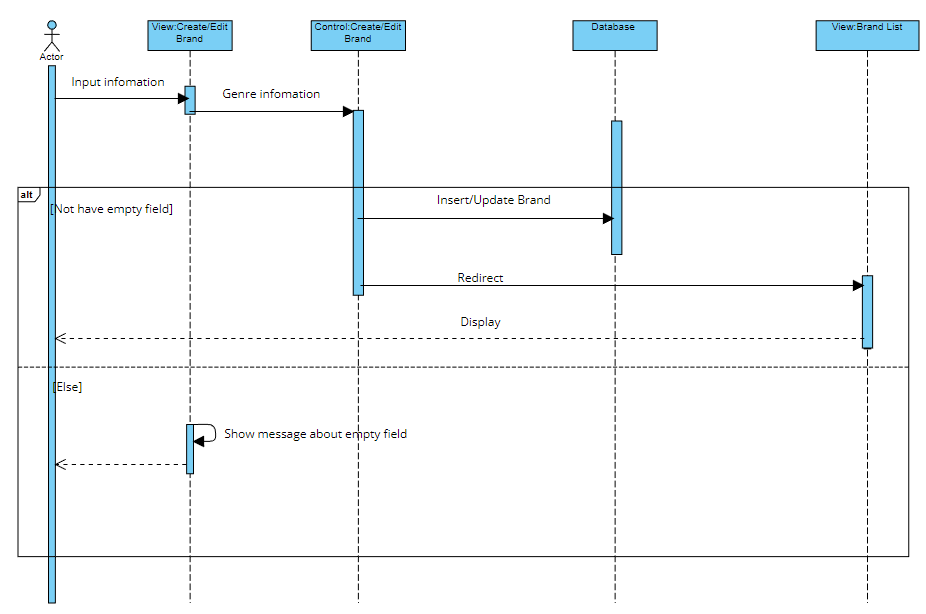
#### **CREATE/EDIT CATEGORY**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3.6 | |
| **Use Case Name** | Create/Edit Category | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor create or edit a category | |
| **Trigger** | Mod/Admin/Editor want to create/edit a category | |
| **Pre-conditions** | The actor’s device is already connected to the internet | |
| **Post-conditions** | Success: A category is created/edited successfully | |
| Fail: A category is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to category management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit category form with following controls:  - “Name” text field  - “Submit” and “Reset” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor leave empty information | All field cannot be blank |

#### **DELETE CATEGORY**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3.7 | |
| **Use Case Name** | Delete Category | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor delete category | |
| **Trigger** | Mod/Admin/Editor wants to delete a category | |
| **Pre-conditions** | The author device is already connected to the internet | |
| **Post-conditions** | Success: A category is deleted successfully | |
| Fail: A category is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to category management and click on Delete button | 2. System delete the correspond record in database |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor requested a category that no longer exists | Display error message |

#### **CREATE/EDIT BRAND**



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3.8 | |
| **Use Case Name** | Create/Edit Brand | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor create or edit a brand | |
| **Trigger** | Mod/Admin/Editor want to create/edit a brand | |
| **Pre-conditions** | The actor’s device is already connected to the internet | |
| **Post-conditions** | Success: A brand is created/edited successfully | |
| Fail: A brand is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to brand management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit brand form with following controls:  - “Name” text field  - “Submit” and “Reset” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor leave empty information | All field cannot be blank |

#### **DELETE BRAND**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3.9 | |
| **Use Case Name** | Delete Brand | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor delete brand | |
| **Trigger** | Mod/Admin/Editor wants to delete a brand | |
| **Pre-conditions** | The author device is already connected to the internet | |
| **Post-conditions** | Success: A brand is deleted successfully | |
| Fail: A brand is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to brand management and click on Delete button | 2. System delete the correspond record in database |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor requested a brand that no longer exists | Display error message |

#### **CREATE/EDIT PRODUCT**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-4.0 | |
| **Use Case Name** | Create/Edit Product | |
| **Author** | Hoang | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor create or edit a product | |
| **Trigger** | Mod/Admin/Editor want to create/edit a product | |
| **Pre-conditions** | The actor’s device is already connected to the internet | |
| **Post-conditions** | Success: A subtitle is created/edited successfully | |
| Fail: A subtitle is created/edited fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to Product management and click on Create Field link or Edit button  3. Actor fill the information and click on “Create” / “Save” button | 2. System shows the create/edit product form with following controls:  - “Name” text field  - “VTT” choose file  - “Price” text field  - “Description” text field  - “Brand” select field  - “Category” select field  - “Submit” and “Reset” button  4. System validates the information  5. System save the information to database and show the success message |
| **Exception** | **Actor’s actions** | **System’s responses**  System displays message to actor |
| Actor leave empty information | All field cannot be blank |

#### **DELETE PRODUCT**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-4.1 | |
| **Use Case Name** | Delete product | |
| **Author** | Nam | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor delete product | |
| **Trigger** | Mod/Admin/Editor wants to delete a product | |
| **Pre-conditions** | The author device is already connected to the internet | |
| **Post-conditions** | Success: A product is deleted successfully | |
| Fail: A product is deleted fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to product management and click on Delete button | 2. System delete the correspond record in database |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor requested a product that no longer exists | Display error message |

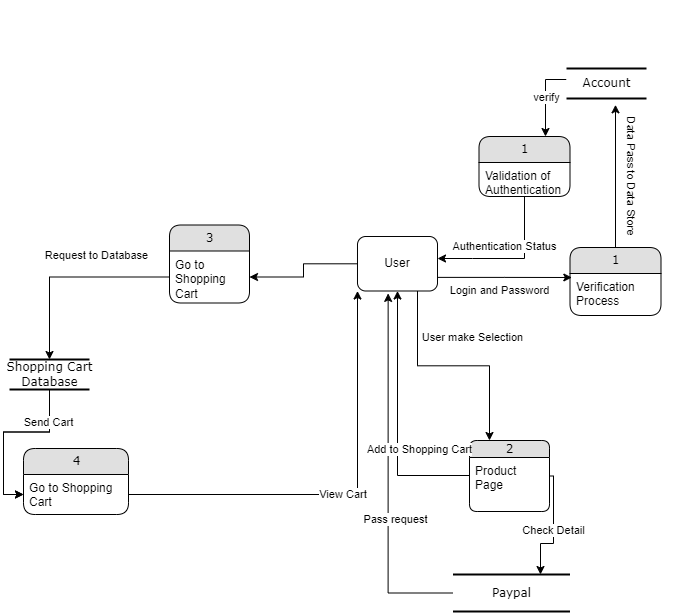
#### **MANAGE ORDER**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-4.2 | |
| **Use Case Name** | Manage order | |
| **Author** | Dung | |
| **Actors** | Mod/Admin/Editor | |
| **Description** | Mod/Admin/Editor manage order | |
| **Trigger** | Mod/Admin/Editor wants to manage an order | |
| **Pre-conditions** | The author device is already connected to the internet | |
| **Post-conditions** | Success: A subtile is managed successfully | |
| Fail: A genre is managed fail | |
| **Basic flow** | **Actor’s actions** | **System’s responses** |
| 1. Actor access to genre management and click on details button to view orderdetail | 2. System will transfer from order to orderdetail |
| **Exception** | **Actor’s actions** | **System’s responses** |
| Actor requested an order that no longer exists | Display error message |

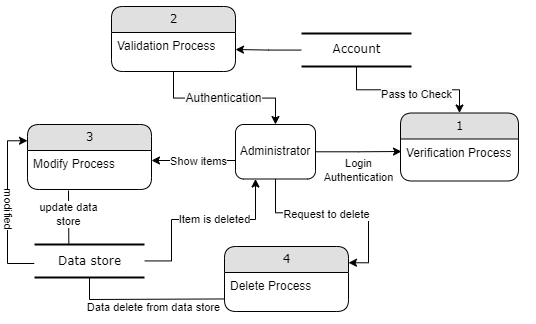
### 3. DATA FLOW DIAGRAM

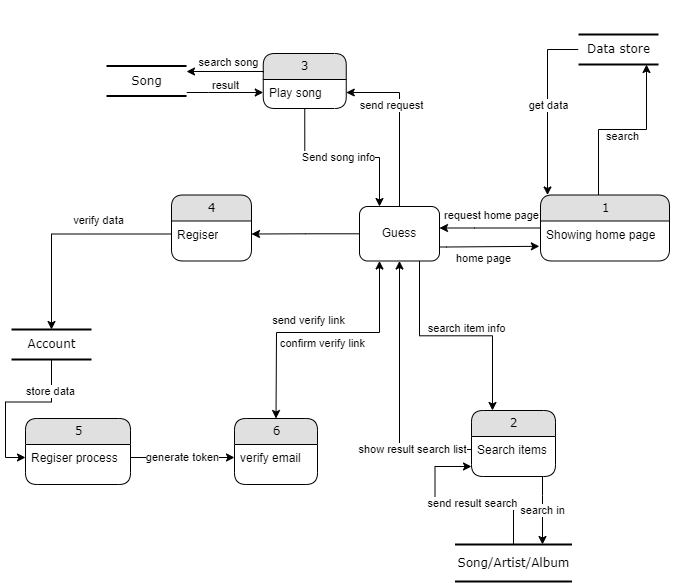
#### **CONTEXT DIAGRAM OR LEVEL 0**

#### **LEVEL 1**

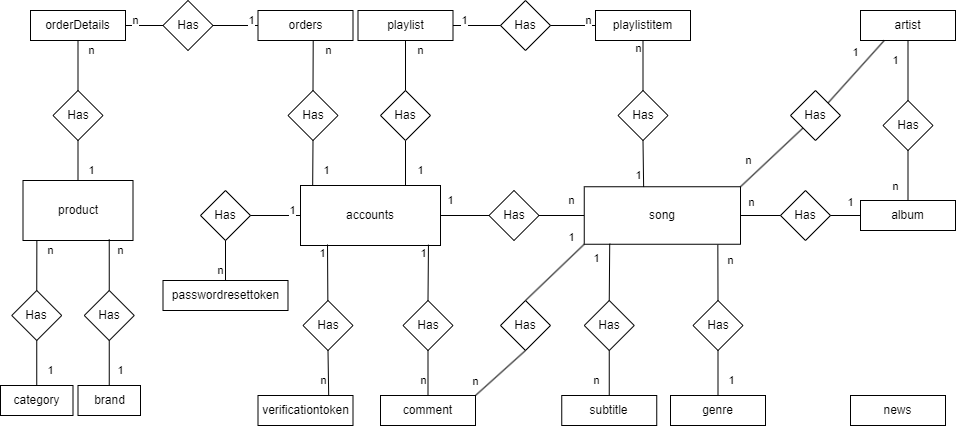


#### **LEVEL 2**



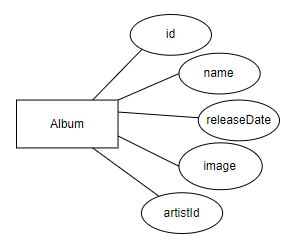


### 4. LOGICAL ENTITY RELATIONSHIP DIAGRAM

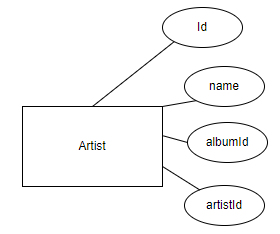


### 5. ENTITY PROPERTIES

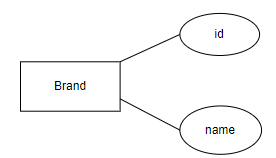
#### **ACCOUNT**



#### **ALBUM**

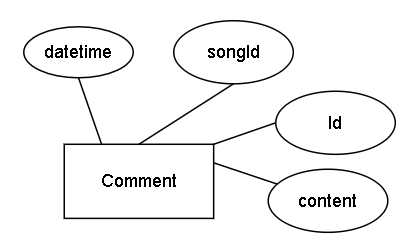


#### **ARTIST**



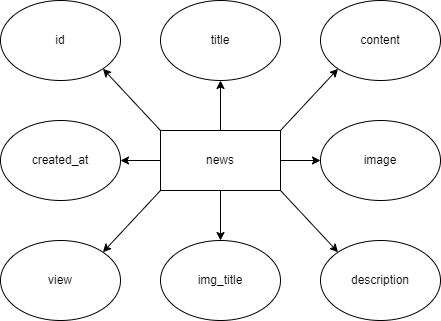
#### **BRAND**

#### **CATEGORY**

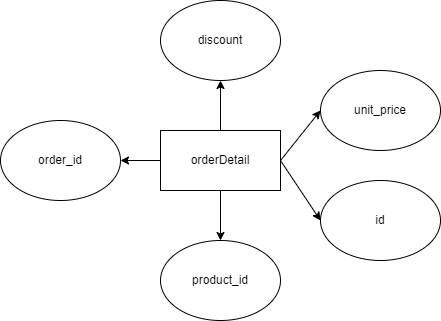


#### **COMMENT**

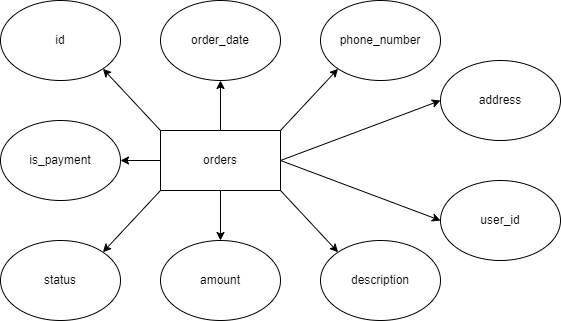
#### **GENRE**



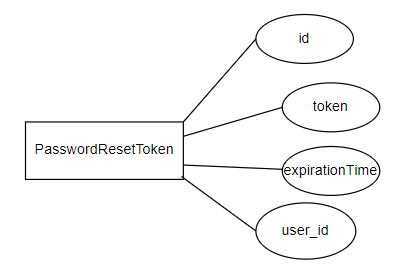
#### **NEWS**



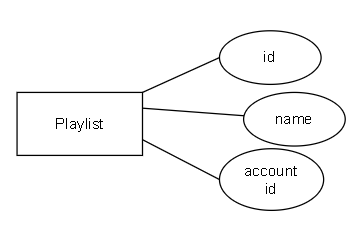
#### **ORDERDETAIL**



#### **ORDERS**

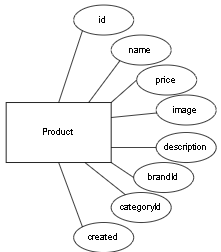


#### **PASSWORDRESETTOKEN**

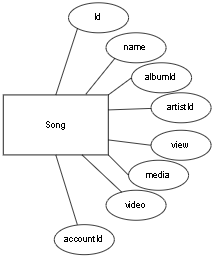


#### **PLAYLIST**

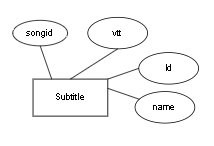
#### **PLAYLISTITEM**



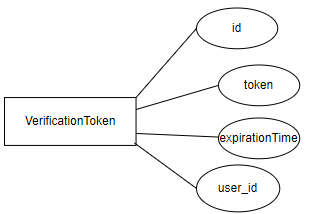
#### **PRODUCT**



#### **SONG**

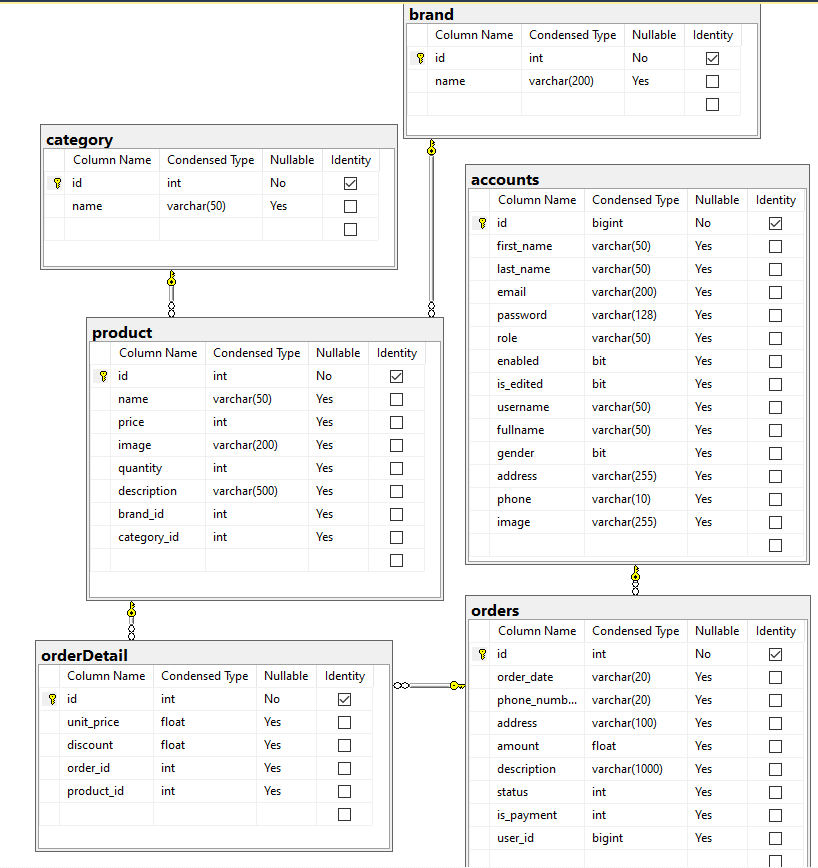


#### **SUBTITLE**



#### **VERIFICATIONTOKEN**

### 6. DATABASE DIAGRAM



##### 

### 7. TABLE DESIGN

#### **ACCOUNT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | bigint | no | 8 | Primary Key |
| 2 | first\_name | varchar | yes | 50 | Account's first name |
| 3 | last\_name | varchar | yes | 50 | Account's last name |
| 4 | email | varchar | yes | 200 | Account's email |
| 5 | password | varchar | yes | 128 | Account's password |
| 6 | role | varchar | yes | 50 | Account's role |
| 7 | enabled | bit | yes | 1 | Check account active or not |
| 8 | is\_edited | bit | yes | 1 | Check upgrade or not |
| 9 | username | varchar | yes | 50 | Username use to login |
| 10 | fullname | varchar | yes | 50 | Account's full name |
| 11 | gender | bit | yes | 1 | Account's gender |
| 12 | address | varchar | yes | 255 | Account's address |
| 13 | phone | varchar | yes | 10 | Account's phone |
| 14 | image | varchar | yes | 255 | Account's image info |

#### **ALBUM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | name | varchar | yes | 50 | Album name |
| 3 | artist\_id | int | yes | 4 | Foreign key of artist id |
| 4 | release\_date | date | yes | 3 | Album release date |
| 5 | image | varchar | yes | 200 | Album image |

#### **ARTIST**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | name | varchar | yes | 50 | Artist name |
| 3 | description | varchar | yes | 50 | Artist description |
| 4 | image | varchar | yes | 200 | Artist image |

#### **BRAND**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | name | varchar | yes | 200 | Brand’s name |

#### **CATEGORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | name | varchar | yes | 50 | Category’s name |

#### **COMMENT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | content | nvarchar | yes | 8000 | Comment content |
| 3 | account\_id | bigint | yes | 8 | Foreign key of account id |
| 4 | song\_id | int | yes | 4 | Foreign key of song id |
| 5 | datetime | datetime | yes | 8 | Comment datetime |

#### **GENRE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | name | varchar | yes | 50 | Genre name |

#### **NEWS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | title | nvarchar | yes | 1000 | Title of news |
| 3 | content | nvarchar | yes | 1000 | Content of news |
| 4 | description | nvarchar | yes | 1000 | Description of news |
| 5 | image | nvarchar | yes | 1000 | Image og news |
| 6 | img\_title | nvarchar | yes | 1000 | Image title of news |
| 7 | view | int | yes | 4 | views of news |
| 8 | created\_at | nvarchar | yes | 1000 | Release date of news |

#### **ORDERS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | order\_date | nvarchar | yes | 20 | Order date of order |
| 3 | phone\_number | nvarchar | yes | 20 | phone number of user |
| 4 | address | nvarchar | yes | 100 | Address of user |
| 5 | amount | float | yes | 10 | Amount of order |
| 6 | description | nvarchar | yes | 1000 | Description of order |
| 7 | status | int | yes | 4 | Status of order |
| 8 | is\_payment | int | yes | 4 | Is payment of order |
| 9 | user\_id | int | yes | 4 | Foreign key of user id |

#### **ORDERDETAIL**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | order\_id | int | yes | 4 | Foreign key of order id |
| 3 | product\_id | int | yes | 4 | Foreign key of product id |
| 4 | unit\_price | float | yes | 10 | Unit price of order details |
| 5 | discount | float | yes | 15 | Total price of order details |

#### **PASSWORDRESETTOKEN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | bigint | no | 8 | Primary Key |
| 2 | token | varchar | yes | 255 | Generate token when reset password |
| 3 | expiration\_time | datetime | yes | 8 | Timing expired |
| 4 | user\_id | bigint | yes | 8 | Foreign key of account's id |

#### **PLAYLIST**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | name | varchar | yes | 50 | Playlist name |
| 3 | account\_id | bigint | yes | 8 | Foreign key of account's id |

#### **PLAYLISTITEM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | song\_id | int | yes | 4 | Foreign key of song id |
| 3 | playlist\_id | int | yes | 4 | Foreign key of playlist id |

#### **PRODUCT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | name | varchar | yes | 50 | Product’s name |
| 3 | price | int | yes | 4 | Product’s price |
| 4 | image | varchar | yes | 200 | Product’s image |
| 5 | quantity | int | yes | 4 | Product’s quantity |
| 6 | description | varchar | yes | 500 | Product’s description |
| 7 | brand\_id | int | yes | 4 | Foreign key of brand's id |
| 8 | category\_id | int | yes | 4 | Foreign key of category's id |

#### **SONG**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | name | varchar | yes | 50 | Song name |
| 3 | media | varchar | yes | 500 | Song audio |
| 4 | album\_id | int | yes | 4 | Foreign key of album id |
| 5 | artist\_id | int | yes | 4 | Foreign key of artist id |
| 6 | lyric | ntext | yes | 16 | Song lyric |
| 7 | genre\_id | int | yes | 4 | Foreign key of genre id |
| 8 | view | int | yes | 4 | Views of the song |
| 9 | video | varchar | yes | 500 | MV of the song |
| 10 | account\_id | bigint | yes | 8 | Foreign key account id |

#### **SUBTITLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | int | no | 4 | Primary Key |
| 2 | name | nvarchar | yes | 100 | Subtitle name |
| 3 | song\_id | int | yes | 4 | Foreign key of song id |
| 4 | vtt | varchar | yes | 500 | File .vtt name |

#### **VERIFYCATIONTOKEN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Column Name** | **Data Type** | **Nullable** | **Length** | **Description** |
| 1 | id | bigint | no | 8 | Primary Key |
| 2 | token | varchar | yes | 255 | Generate token when register |
| 3 | expiration\_time | datetime | yes | 8 | Timing expired |
| 4 | user\_id | bigint | yes | 8 | Foreign key of account's id |

### 8. TASK SHEET REVIEW II

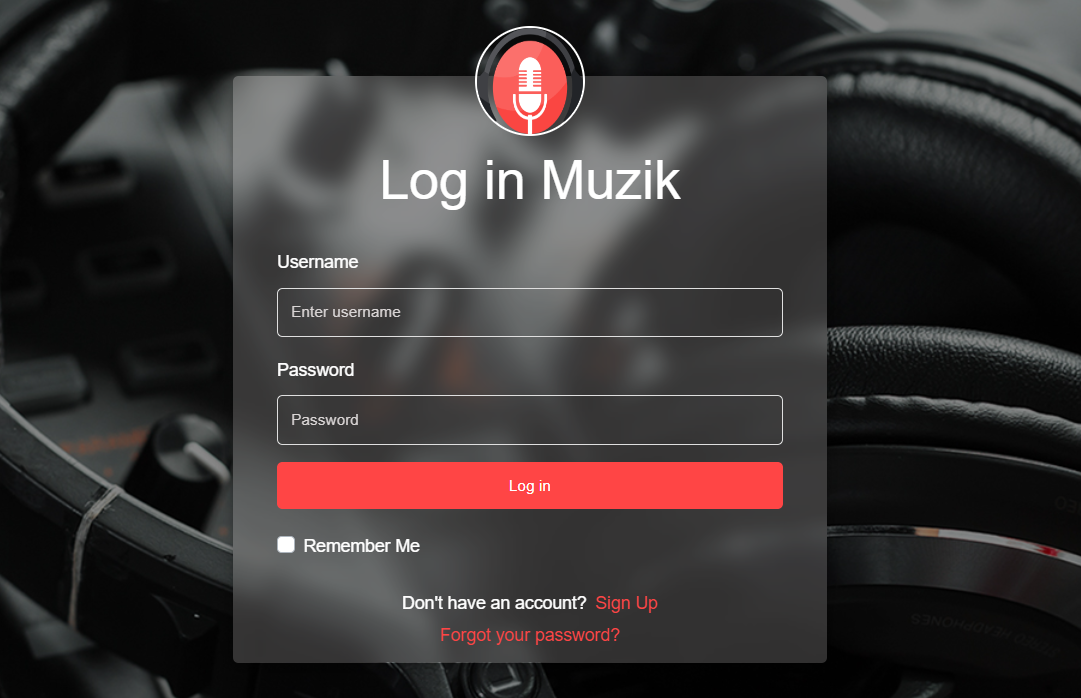
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Ref.  No: 3 | Project Title:  Muzik’web page application | Date of Preparation of Activity Plan | | | |
| No. | Task | Actual Start Date | Actual Days | Team Member Names | Status |
| 01 | Use Cases User | 28 July , 2022 | 10 | All Members | Completed |
| 02 | Use Cases Admin | All Members | Completed |
| 03 | Data Flow Diagrams | All Members | Completed |
| 04 | Logical ERD | Sang | Completed |
| 05 | Entity Properties | All Members | Completed |
| 06 | Database Diagram | Sang | Completed |
| 07 | Table Design | All Members | Completed |
| 08 | Task Sheet Review II |  |  | Sang | Completed |

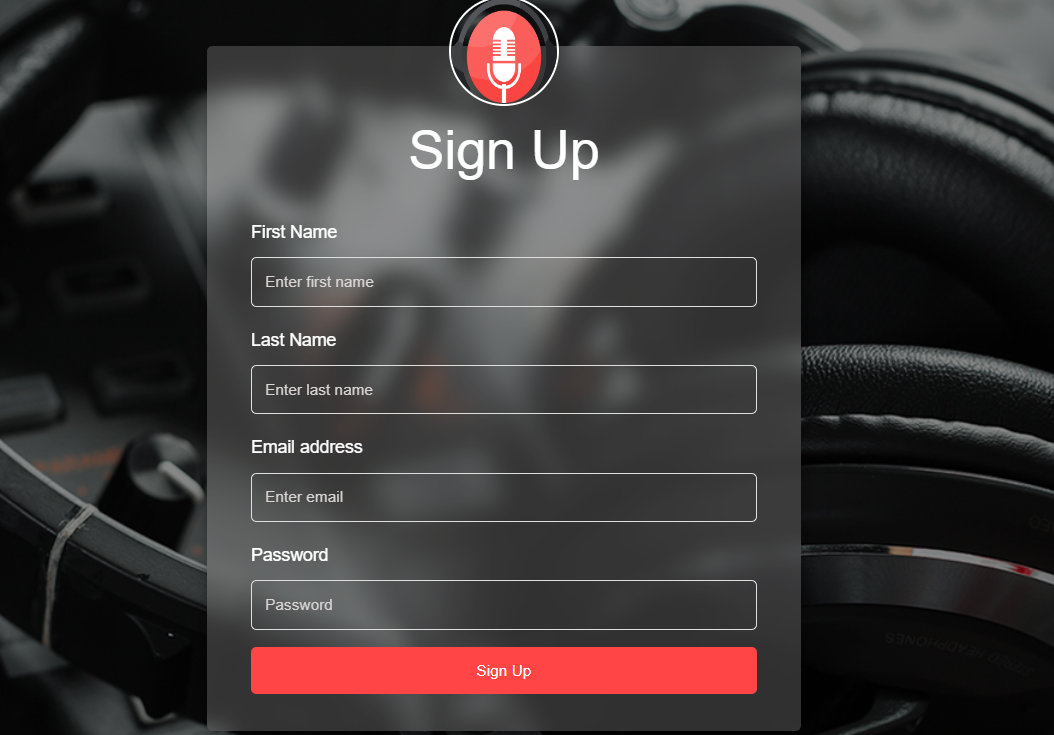
|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 2 | Approved By: Faculty |
| Date: 8 Augus , 2022 | Team Leader  Luu Thanh Sang | Mr. **Hoang Duc Quang** |

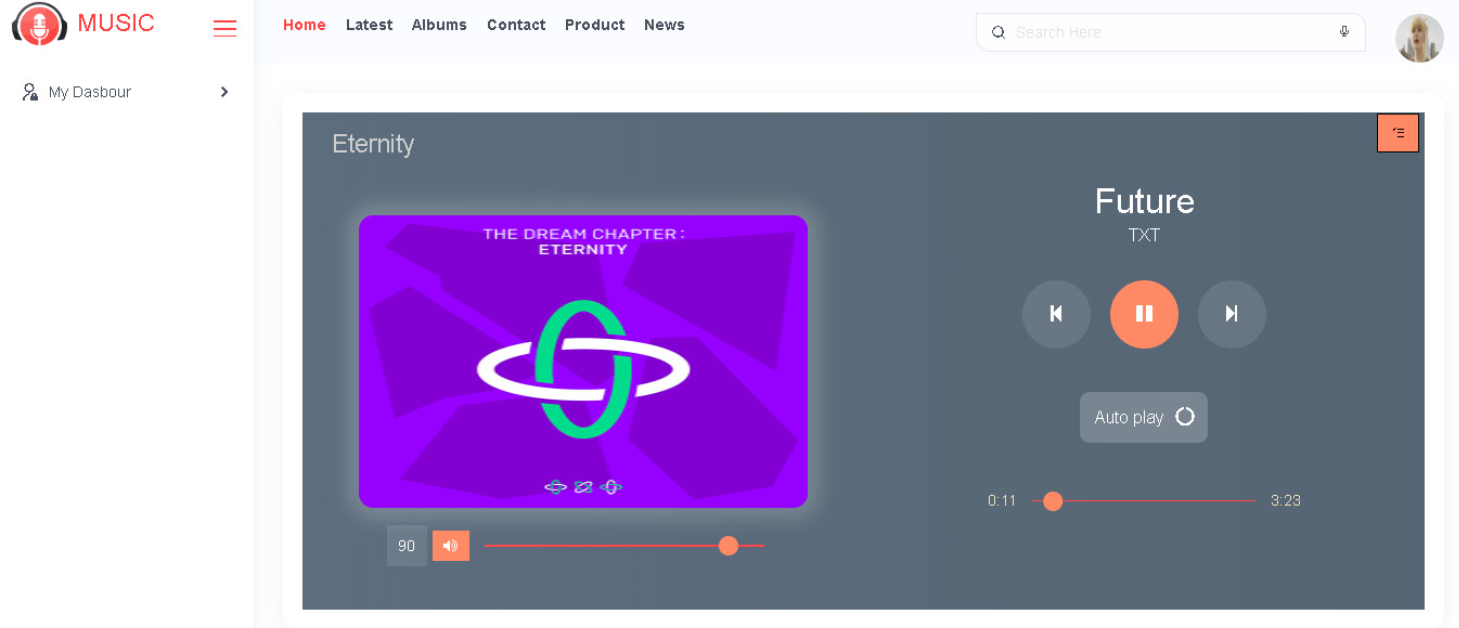
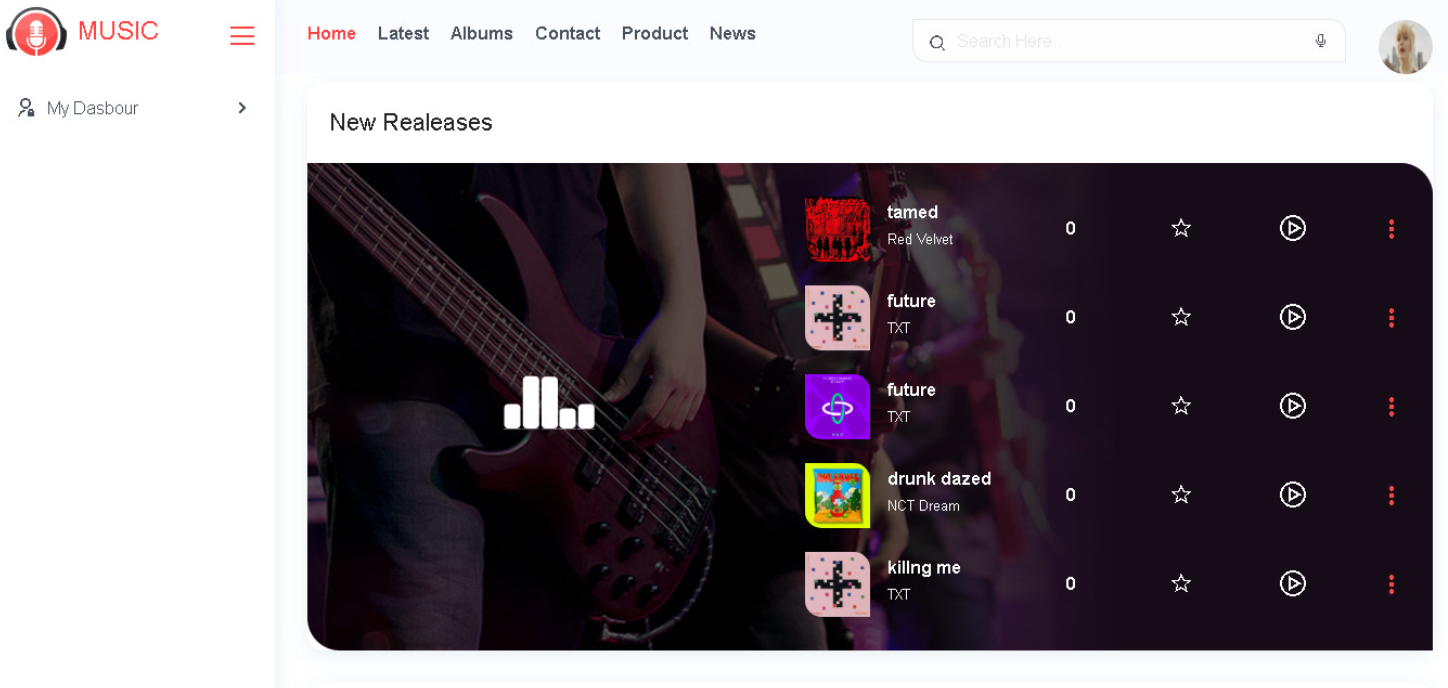
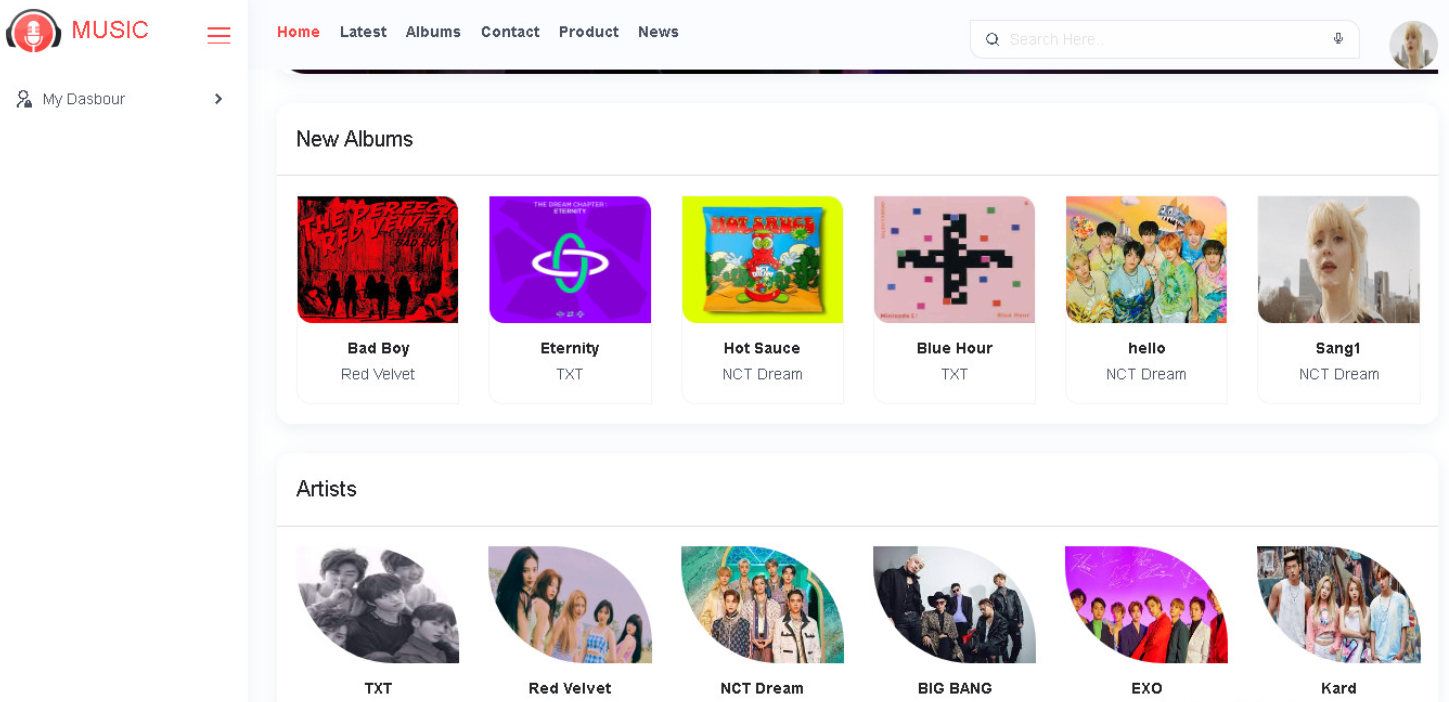
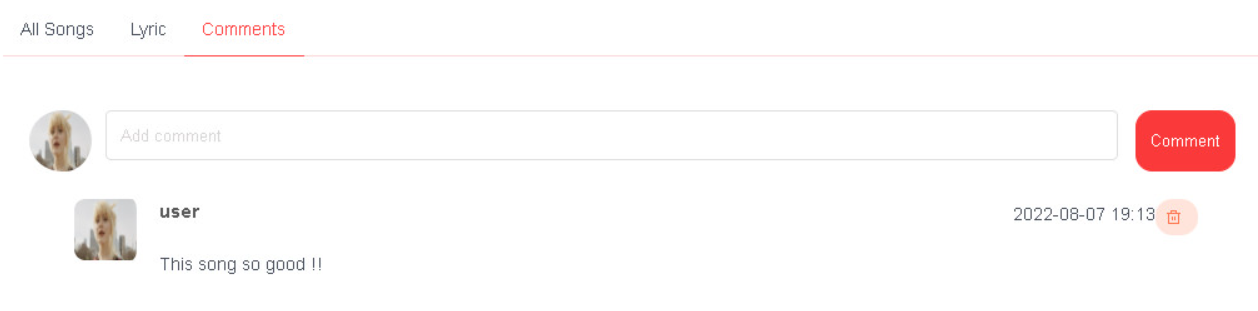
## **REVIEW III**

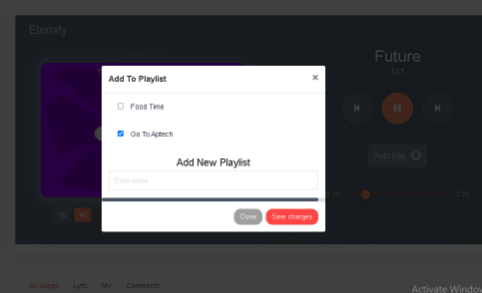
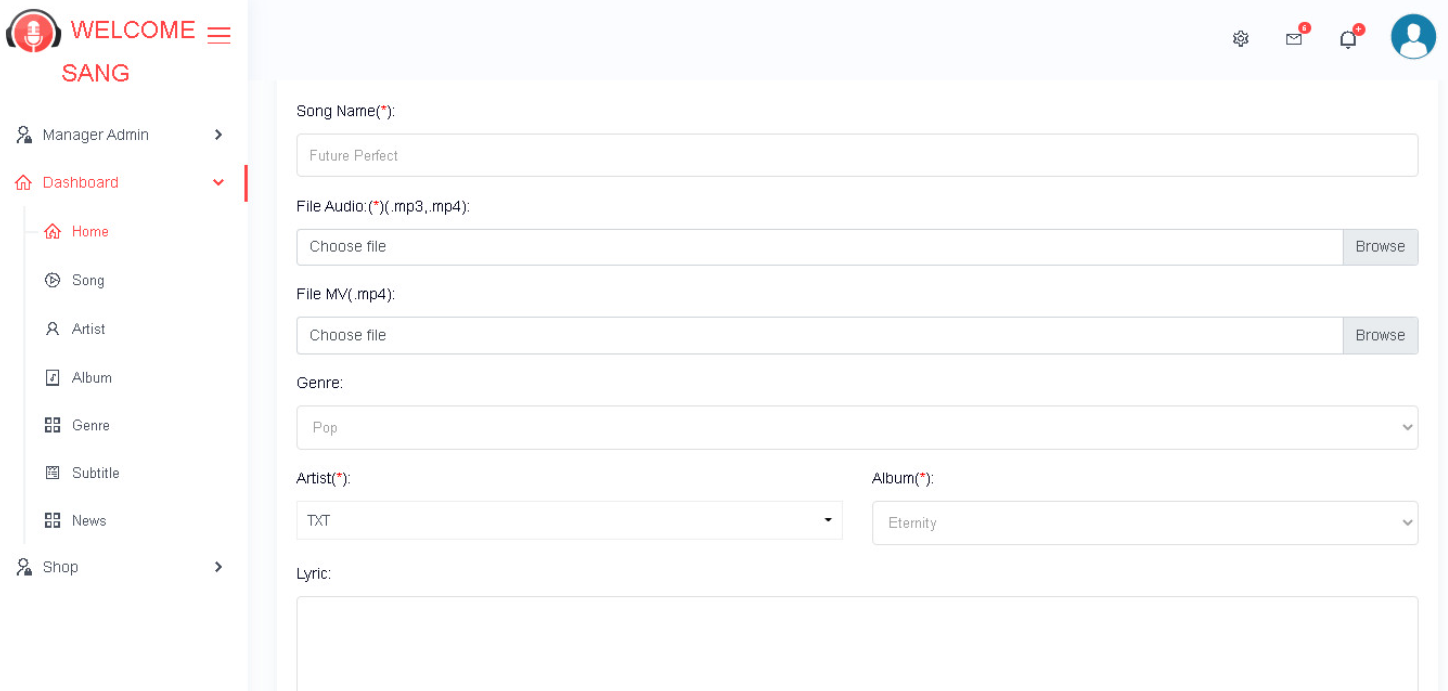
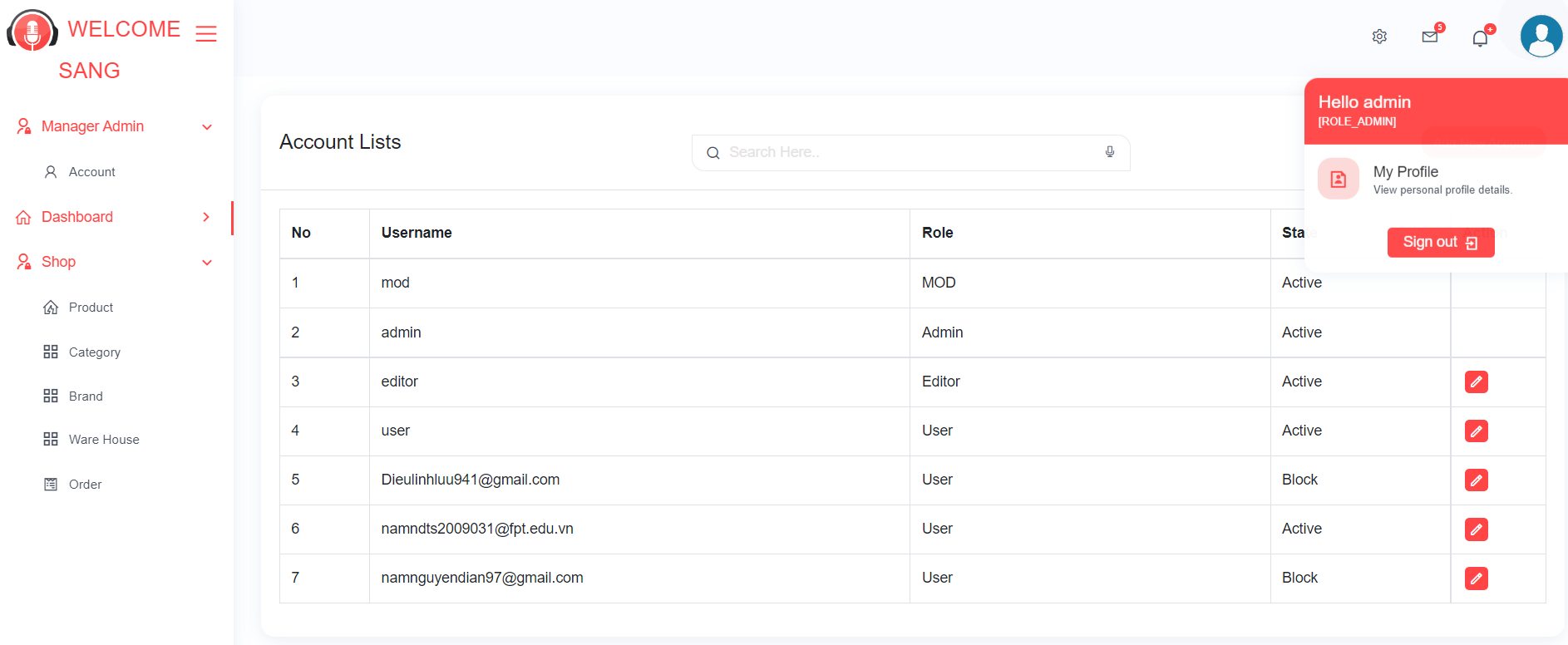
### 

### DESIGN – FRONT-END:









### TASK SHEET REVIEW III

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Ref.  No: 3 | Project Title:  Muzik’web page application | Date of Preparation of Activity Plan | | | |
| No. | Task | Actual Start Date | Actual Days | Team Member Names | Status |
| 01 | Design, Front-end | 28 July , 2022 | 10 | All Members | Completed |
| 02 | Task Sheet Review III | All Members | Completed |

|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 2 | Approved By: Faculty |
| Date: 8 Augus , 2022 | Team Leader  Luu Thanh Sang | Mr. **Hoang Duc Quang** |