

ASSIGNMENT 2 FRONT SHEET

Qualification	BTEC Level 5 HND Diploma in Computing		
Unit number and title	Unit 9: Software Development Life Cycle		
Submission date	February 3, 2021	Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	
Student Name	NGUYEN CHI HAI	Student ID	GCC18033
Class	GCC0701	Assessor name	NGUYEN THAI NGHE
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	<i>Nguyen Chi Hai</i>

Grading grid

P5	P6	P7	M3	M4	M5	M6	D3	D4

☐ **Summative Feedback:**
☐ **Resubmission Feedback:**
Grade:
Assessor Signature:
Date:
IV Signature:

ASSIGNMENT 2 BRIEF

Qualification	BTEC Level 5 HND Diploma in Computing		
Unit number	Unit 9: Software Development Life Cycle		
Assignment title	Undertake a software development lifecycle		
Academic Year	2019 – 2020		
Unit Tutor	LE Minh Duc		
Issue date		Submission date	
Name and date			

Submission Format:

<i>Format:</i>	<p>The submission is in the form of 1 document.</p> <p>You must use the <i>Times font</i> with <i>12pt size</i>, turn on <i>page numbering</i>; set <i>line spacing</i> to <i>1.3</i> and <i>margins</i> to be as follows: left = 1.25cm, right = 1cm, top = 1cm, bottom = 1cm. Citation and references must follow the Harvard referencing style.</p> <p>Word limit: 3000 words (excluding figures and references). Submissions that exceed this limit will be rejected.</p>
<i>Submission:</i>	<p>You must submit the assignment by the due date and follow the submission method specified by the Tutor. The submission form is soft copy, which is to be uploaded to the following URL: http://cms.greenwich.edu.vn.</p>
<i>Note:</i>	<p>Your assignment <i>must</i> be your own work, and not copied by or from another student or from other sources, such as book etc. If you use ideas, quotes or data (such as diagrams) from books, journals or other sources, you must reference the sources, using the Harvard style. Make sure that you know how to reference properly and that you understand the plagiarism guidelines.</p> <p>Plagiarism is a very serious offence, which will result in a failing grade.</p>

Unit Learning Outcomes:

LO3 Undertake a software development lifecycle.

LO4 Discuss the suitability of software behavioural design techniques.

Assignment Brief and Guidance:

Tasks

At this stage, you have convinced Tune Source to select your project for development. Complete the following tasks to analyse and design the software.

Task 1 – Analysis (1)

1. (P5.a) Identify the stakeholders, their roles and interests in the case study.

Review the requirement definition of the project. Clearly indicate which stakeholder(s) provide what requirements.

Word limit: 150 - 200

Identify FRs and NFRs of TuneSource Project

Discuss the relationships between the FRs and NFRs.

Word limit: 300 – 400 words

2. (P5.b) Discuss the technique(s) you would use to obtain the requirements.

If needed, you may state suitable additional assumptions about the project in order to justify the technique(s) that you choose.

Techniques: JAD, Interview, Observation, etc ...

Demonstrate how to collect requirements based on chosen technique

Word limit: 700 - 1000

3. (M3) Discuss how you would trace these requirements throughout the project.

Word limit: 400 – 500 words

Task 2 – Analysis (2)

(P6) Analyse the requirements that you identified in Task 1 using a combination of structural and behavioural modelling techniques that you have learnt.

Scope: you only need to construct following items for the system. You will have to include

- Use Case Diagram for the whole system
- Use Case specification for 2 Use cases
- Context Diagram for the whole system
- Data Flow Diagram – Level 0 for the whole system
- ERD for the whole system

Word limit: 1000 – 1200 words

Task 3 – Design

Based on the analysis result, discuss how you would conduct the design phase:

1. (P7) Discuss how the user and software requirements are addressed in the design phase.
 - You will explain how Mockup and Wireframe are used in the project. You should include some of the mockup or wireframe (at least 5) design of the TuneSource project to justify that it matches users' requirements
 - You will explain which architecture (client – server, n-tier, microservices, etc.) is suitable for the project with clear illustrations and why
 - Then you will address which solution stack could be suitable to implement the project with clear explanations
2. (M5) Discuss how activity diagram and pseudocode are used to specify the software behaviour.

3. (M6) Discuss how UML state machine can be used to specify the software behaviour. Differentiate between FSM And extended FSM using the case study.
4. (D4) Discuss how the data-driven approach improves the reliability and effectiveness of software.

Word limit: 400 - 1500

Task 4 – Software quality management

1. (M4.a) Discuss two software quality attributes that are applicable to the project.
2. (M4.b) Discuss two quality assurance techniques that can help improve the software quality in the project.
3. (D3) Discuss how the design techniques and approaches that you have used can help improve the software quality.

Word limit: 400 - 1500

Learning Outcomes and Assessment Criteria		
Pass	Merit	Distinction
LO3 Undertake a software development lifecycle		D3 Critically evaluate how the use of the function design paradigm in the software development lifecycle can improve software quality.
<p>P5 Undertake a software investigation to meet a business need.</p> <p>P6 Use appropriate software analysis tools/techniques to carry out a software investigation and create supporting documentation.</p>	<p>M3 Analyse how software requirements can be traced throughout the software lifecycle.</p> <p>M4 Discuss two approaches to improving software quality.</p>	
LO4 Discuss the suitability of software behavioural design techniques		D4 Present justifications of how data driven software can improve the reliability and effectiveness of software.
<p>P7 Explain how user and software requirements have been addressed.</p>	<p>M5 Suggest two software behavioural specification methods and illustrate their use with an example.</p> <p>M6 Differentiate between a finite state machine (FSM) and an extended- FSM, providing an application for both.</p>	

Contents

1. Tune Source Project	11
1.1. The role of stakeholders	11
1.2. Functional requirement.....	11
1.3. Non-Functional requirement.....	11
1.4. Security.....	12
1.4. Observation.....	12
2. Use Case Diagram for the whole system.....	13
2.1. UC - Administrator.....	13
2.2. UC – Customer	14
2.3. Use case:	14
2.3.1. UC - Login	14
2.3.2. UC – Listen to Music	15
2.3.3. UC - Log out	15
2.3.4. UC - Account	16
2.3.5. UC - Music.....	17
2.3.6. UC - Category	17
2.3.7. UC – Order Details.....	18
2.3.8. UC – Order	19
2.3.9. UC – Register	20
2.4. DFD (Data Flow Diagram).....	21
2.5 ERD (Entity Relationship Diagram).....	24
3. Software requirements.....	25
3.1. Login	25
3.2. Register page.....	26
3.3. Home page of admin account.....	27

3.4. Management Account	27
3.5. Management Music	28
3.6. Management Category	28
3.7. Management Order Details	29
3.8. Home page of User account	29
3.9. Re-View Category of Music.....	30
3.10. Order	30
3.11. Listening to Music basic	31
Conclusion.....	31
References.....	32

List of Figure

Figure 1: Administrator.....	13
Figure 2: Customer	14
Figure 3: DFD – level 0	21
Figure 4: DFD - Level 1 of admin	22
Figure 5: DFD - Level 1 of User.....	23
Figure 6: ERD.....	24
Figure 7: Database Diagram	24
Figure 8: Login page.....	25
Figure 9: Register page	26
Figure 10: Home page of admin account	27
Figure 11: Management account.....	27
Figure 12:Management Music	28
Figure 13: Management category	28
Figure 14: Management Order Details.....	29
Figure 15: Home page of user account	29
Figure 16: Re-view category of music.....	30
Figure 17: Order Music.....	30
Figure 18: Listening to Music basic.....	31

List of Table

Table 1: Statistical tables	12
Table 2: UC – 01	15
Table 3: UC – 02.....	15
Table 4: UC – 03.....	16
Table 5: UC – 04.....	17
Table 6: UC – 05.....	17
Table 7: UC – 06.....	18
Table 8: UC - 07	19
Table 9: UC – 08.....	19
Table 10: UC – 09.....	20

1. Tune Source Project

We are carrying out this project according to the requirements that our stakeholders need. Stakeholders include an on-demand music supplier and a partner with whom we create online music systems. We enforce security and functionality requirements in the process.

1.1. The role of stakeholders

The stakeholders include:

- An on-demand music supplier: Providing music genres at the request of customers.
- Customer: Provide functional requirements, useful features on the system.

1.2. Functional requirement

- Register
- Log in / Log out
- Listen to Music
- Order
- Manage Account
- Manage Music
- Manage Category
- Manage Order Details

Note: All of the management function has included CRUD.

1.3. Non-Functional requirement

System features such as security, reliability, performance, maintainability, scalability, and usability are specified by non-functional requirements (NFRs). On various backlog profiles, they serve as limitations or restrictions to the device design. We ensure the usability and performance of the entire system. Failure to meet any of these requirements which result in the system failing to meet the internal needs of the company, the customer or the market, or the regulatory authorities or customers failing to comply with the mandatory requirements. Standards Imposed. Failure to comply can cause severe legal issues in some cases (privacy, security, safety, some issues).

Our systems are operated and hosted on Cloud Computing. We recommend for partners to use cloud systems of Microsoft Azure. It ensures the security and operability of the system.

1.4. Security

For the systems we designed, we introduced the security specifications.

Safe SQL injection: SQL injection is a common formula of web attack focused on some websites of manipulation, the reason being that the content is often not officially coded and hacking tools take advantage of operating vulnerabilities. destructive. The most prevalent SQL injection prevention consists of two elements. The first is to upgrade and patch all servers, services and applications on a regular basis, then create and use the source code well and test the source code of the website in order not to allow SQL commands to exist. Signs that are irregular (Media, n.d.).

XSS secure website: XSS attack or malicious JavaScript attack on your website, which then runs in the browser of the user and can change the content of the website or steal the details to send the user attack back. Material Security Policy is another important weapon in the XSS Defender toolbox (CSP). CSP is an attribute that I can return to the browser from our server to restrict how JavaScript is executed on the web page, which makes it harder to deal with the writer's scripts. I do, even though we should include it on our website (Media, n.d.).

1.4. Observation

Several outlets from trending music blogs, including Zing MP3 and Spotify, were consulted. In Table 1, we will briefly review the key material.

Product	Cost	Quality (Song)	Playlist	Music preview	Management	User Experience
Zing mp3	Medium	Low	Yes	Yes	No	Normal
Spotify	High	High	Yes	Yes	No	Good
Our Product	Medium	High	Yes	Yes	YES	Good

Table 1: Statistical tables

We received this feedback from real-world humans. These comparative reviews just tell us what needs to be changed and made better by our method.

2. Use Case Diagram for the whole system.

2.1. UC - Administrator

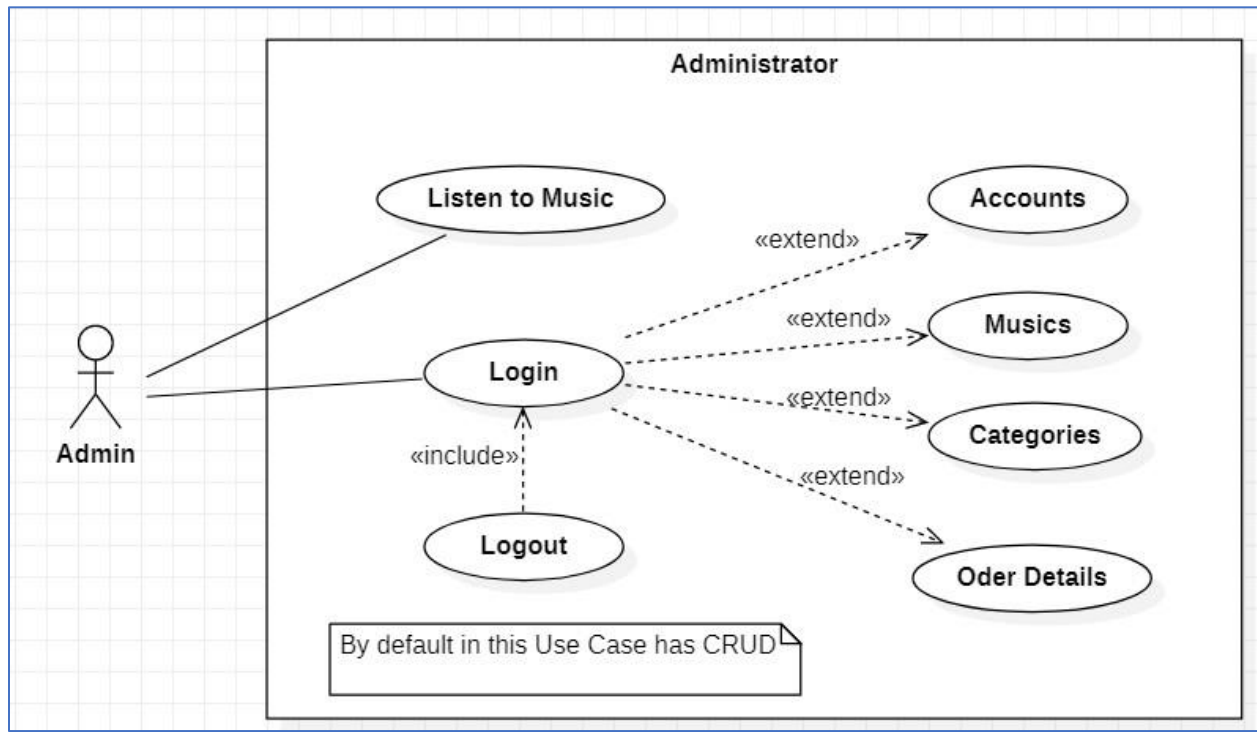


Figure 1: Administrator

2.2. UC – Customer

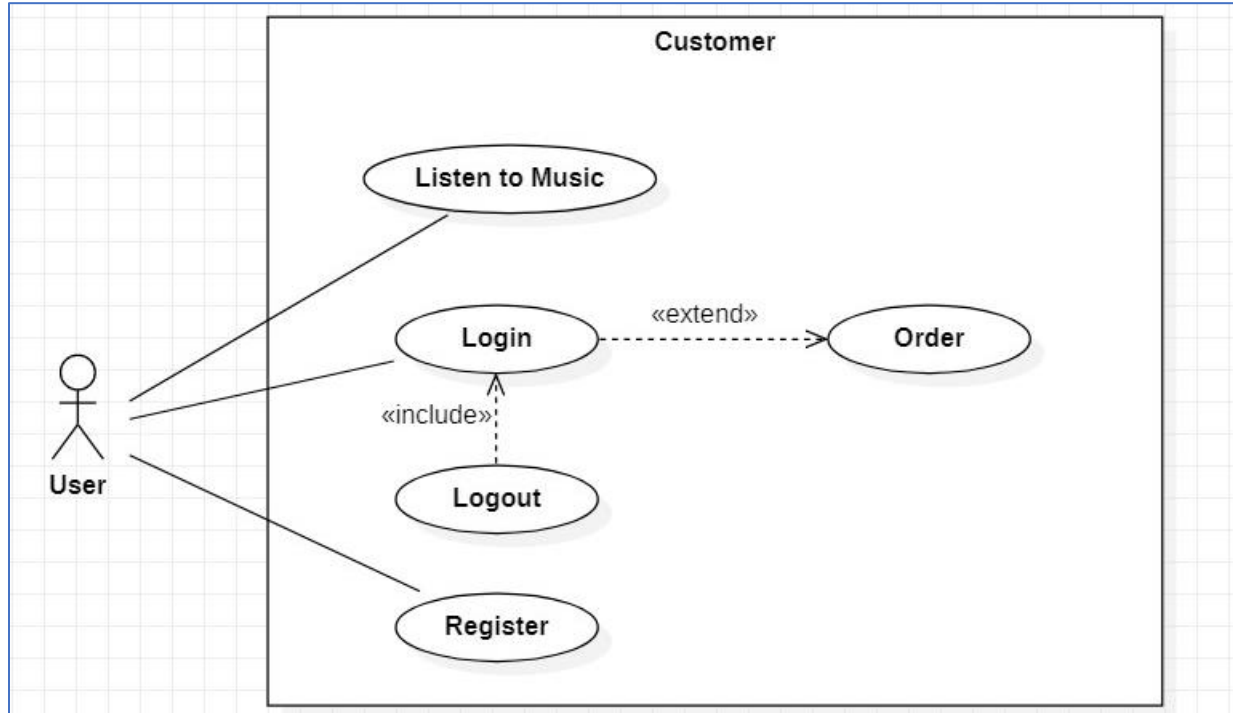


Figure 2: Customer

2.3. Use case:

2.3.1. UC - Login

Use case name: Login	ID: UC - 01	Priority: High
Actors: Admin		
Description: An administrator get log in through log in page.		
Trigger: Admin use their information to get in to their Tune Source account.		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions: required exist data account.		
Normal Course: <ol style="list-style-type: none"> 1. Log in at Tune Source page. 2. Input data account (Username and Password). 		Information for steps:

3. Sending next page, if: <ul style="list-style-type: none"> - Login successfully: Send to a Home page. - Login failed: Re-load the log in page and input data account (Username and Password). Otherwise, use case end!	
Postconditions: <ol style="list-style-type: none"> 1. Account online. 2. Admin can get other functions when they are logged in. 	

Table 2: UC – 01

2.3.2. UC – Listen to Music

Use case name: Listen to Music	ID: UC - 02	Priority: Medium
Actors: Admin, User.		
Description: User and Admin can listen to music on the Home page.		
Trigger: User and Admin can listen to music on the Home page.		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions: Does not require logging into the system.		
Normal Course: - Access the Home page and listen to music. Otherwise, use case end!	Information for steps:	
Postconditions:		

Table 3: UC – 02

2.3.3. UC - Log out

Use case name: Logout	ID: UC - 03	Priority: High
Actors: Admin, User.		

Description: Admin and User can get log out when they were logged in before.	
Trigger: Admin and User can get log out when they were logged in before.	
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal	
Preconditions: Required logging into the system.	
Normal Course: <ol style="list-style-type: none"> 1. Access the Log in page and input data account (Username and Password). 2. Click button Log out. <p>Otherwise, use case end!</p>	Information for steps:
Postconditions:	

Table 4: UC – 03

2.3.4. UC - Account

Use case name: Account	ID: UC - 04	Priority: High
Actors: Admin.		
Description: Administrators that have previously logged into the system will be able to manage UC - Account.		
Trigger: The UC-Account includes the following functions: Edit user account information, delete user accounts if necessary.		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions: Required logging into the system.		
Normal Course: <ol style="list-style-type: none"> 1. Logged in to the system with an administrator account. 2. Access the Account page. <p>Otherwise, use case end!</p>	Information for steps:	

Postconditions:

1. Account online.
2. Admin can get other functions when they are logged in

Table 5: UC – 04

2.3.5. UC - Music

Use case name: Music	ID: UC - 05	Priority: High
Actors: Admin.		
Description: Administrators that have previously logged into the system will be able to manage UC - Music.		
Trigger: The UC-Music includes the following functions: Add new music song, edit a music song information or delete a music song if necessary.		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions: Required logging into the system.		
Normal Course: <ol style="list-style-type: none"> 1. Logged in to the system with an administrator account. 2. Access the Music page. Otherwise, use case end!		Information for steps:
Postconditions: <ol style="list-style-type: none"> 1. Account online. 2. Admin can get other functions when they are logged in 		

Table 6: UC – 05

2.3.6. UC - Category

Use case name: Category	ID: UC - 06	Priority: High
Actors: Admin.		

Description: Administrators that have previously logged into the system will be able to manage UC - Category.	
Trigger: The UC-Category includes the following functions: Add new music category, edit a music category information or delete a music category if necessary.	
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal	
Preconditions: Required logging into the system.	
Normal Course: <ol style="list-style-type: none"> 1. Logged in to the system with an administrator account. 2. Access the Category page. <p>Otherwise, use case end!</p>	Information for steps:
Postconditions: <ol style="list-style-type: none"> 1. Account online. 2. Admin can get other functions when they are logged in 	

Table 7: UC – 06

2.3.7. UC – Order Details

Use case name: Order Details	ID: UC - 07	Priority: High
Actors: Admin.		
Description: Administrators that have previously logged into the system will be able to manage UC - Order Details.		
Trigger: The UC- Order Details includes the following functions: Edit an order details information or delete an order details if necessary.		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions: Required logging into the system.		
Normal Course:	Information for steps:	

<ol style="list-style-type: none"> 1. Logged in to the system with an administrator account. 2. Access the Order Details page. <p>Otherwise, use case end!</p>	
Postconditions: <ol style="list-style-type: none"> 1. Account online. 2. Admin can get other functions when they are logged in 	

Table 8: UC - 07

2.3.8. UC – Order

Use case name: Order	ID: UC - 08	Priority: Medium
Actors: User.		
Description: User that have previously logged into the system will be able to buy music song.		
Trigger: Users can buy any song they like in the Order.		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions: Required logging into the system.		
Normal Course: 1. Logged in to the system with a user account. 2. Access the Order page. Otherwise, use case end!	Information for steps:	
Postconditions: 1. Account online. 2. Admin can get other functions when they are logged in		

Table 9: UC – 08

2.3.9. UC – Register

Use case name: Register		ID: UC - 09	Priority: High
Actors: User.			
Description: Users can register membership accounts through the page "Register" on the website.			
Trigger: Upon successful registration, the user can log in through the login page and can purchase music.			
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal			
Preconditions: Does not require logging into the system.			
Normal Course: 1. Access to the Register page on the website. 2. Input data account need to register. 3. Click button to submit. Otherwise, use case end!		Information for steps:	
Postconditions:			

Table 10: UC – 09

2.4. DFD (Data Flow Diagram)

Level 0 DFD: This DFD show how admin and user relation to Tune Source.

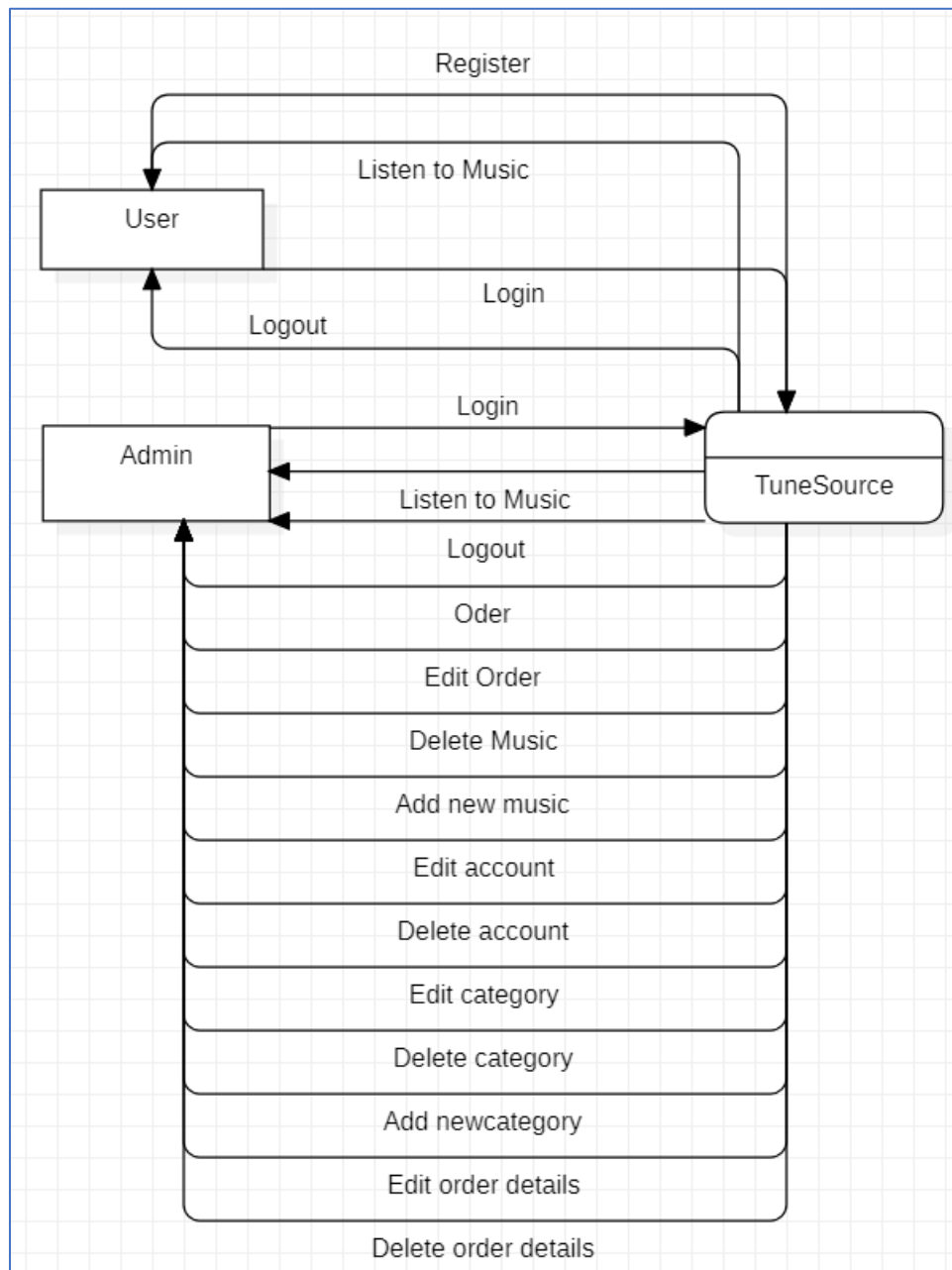


Figure 3: DFD – level 0

Level 1 DFD: More detail DFD, Admin interact with each process, and system interact with the database.

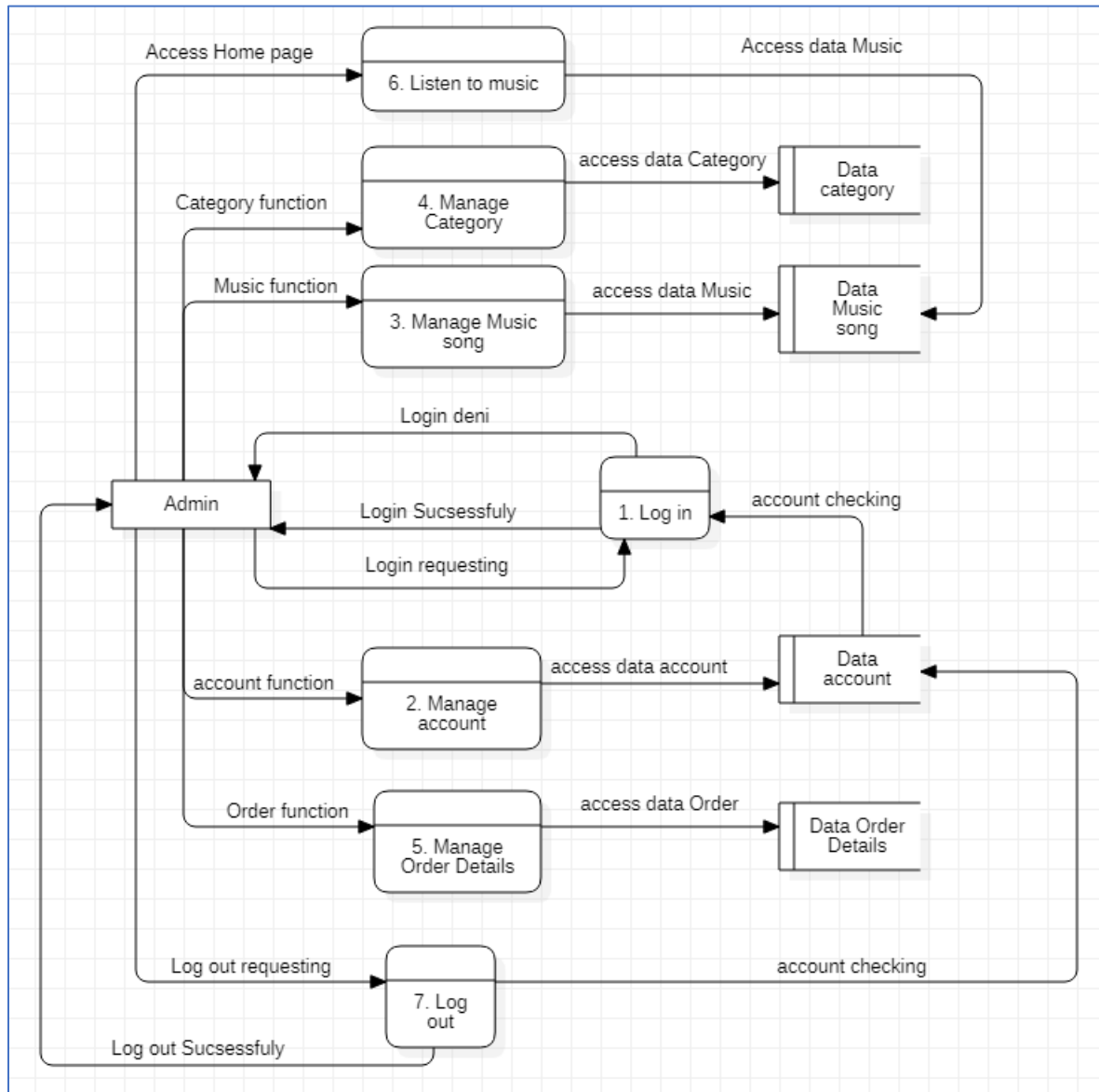


Figure 4: DFD - Level 1 of admin

Level 1 DFD: More detail DFD, User interact with each process, and system interact with the database.

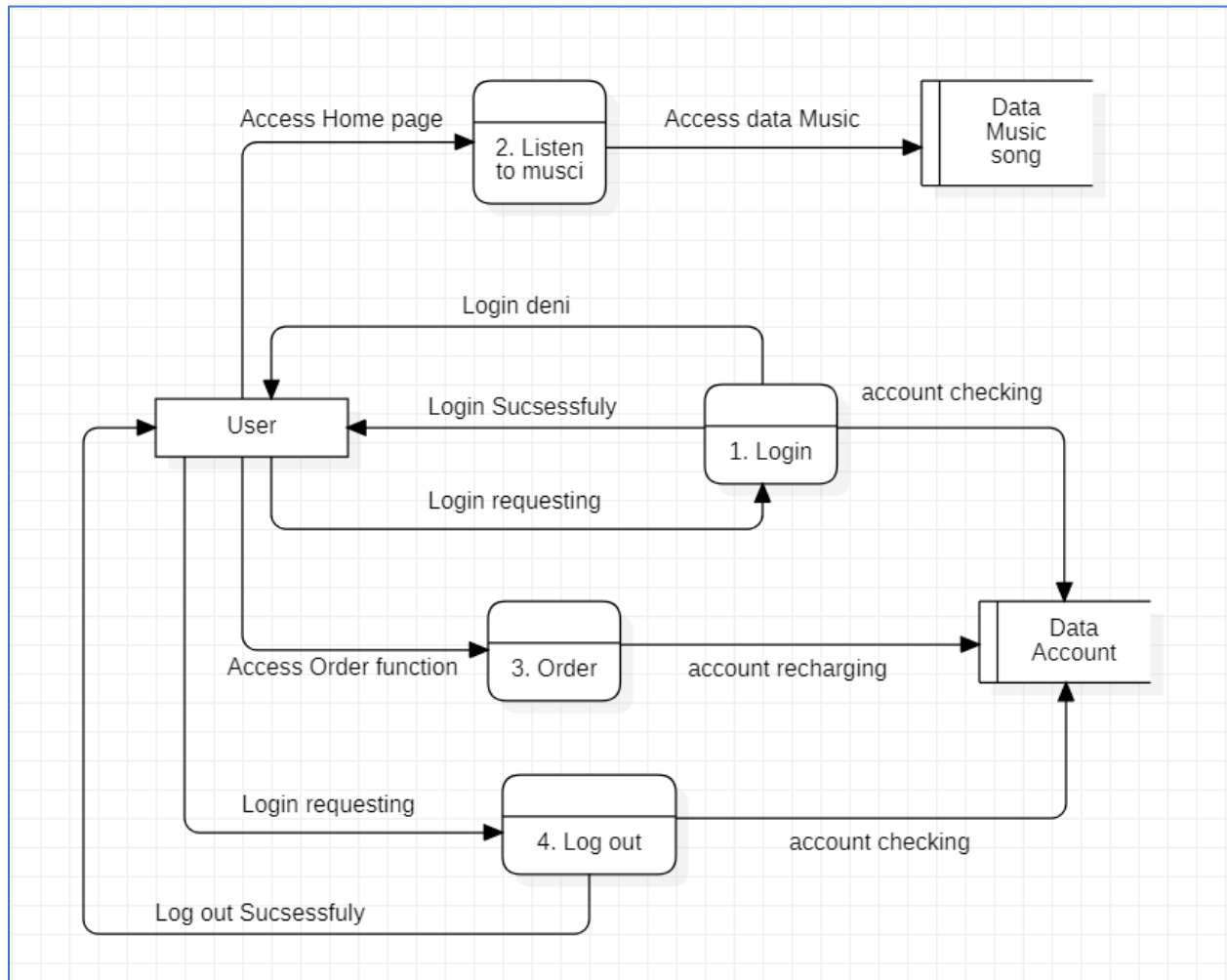


Figure 5: DFD - Level 1 of User

2.5 ERD (Entity Relationship Diagram)

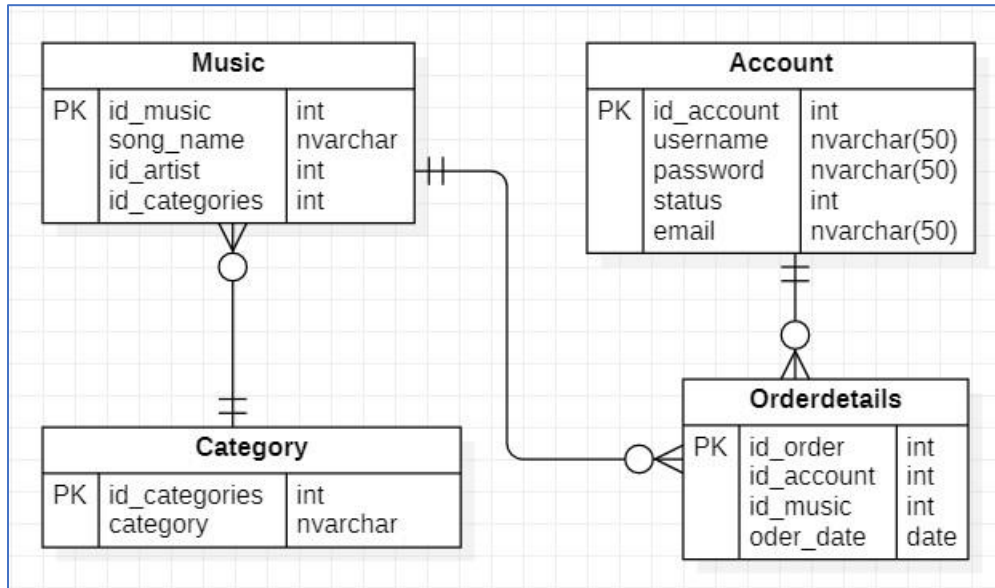


Figure 6: ERD

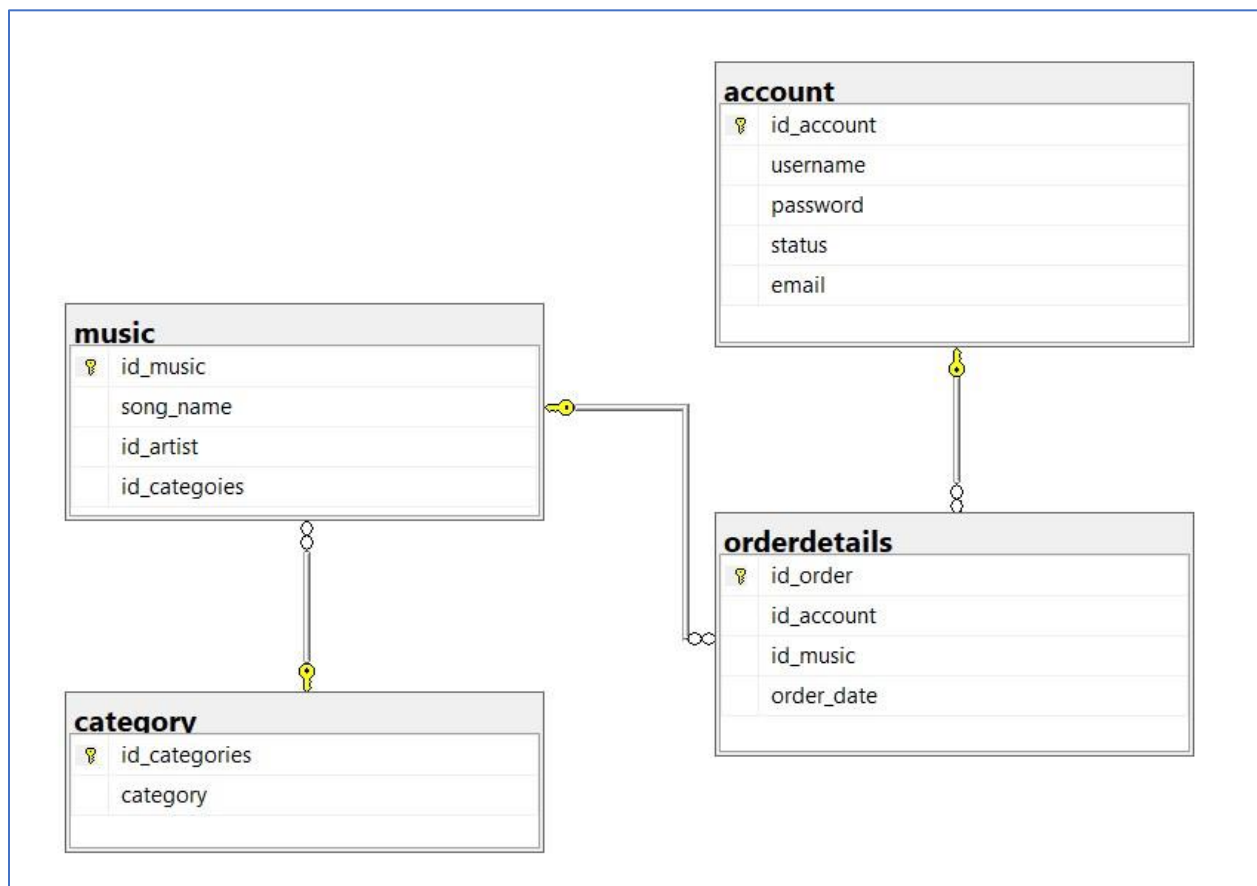


Figure 7: Database Diagram

3. Software requirements

At this point, to comply with the above requirements, We will elaborate. I choose the functions of Tune Source for the diagram, as – Register, Login / Log out, Music listening, Order, Account management, Music management, Category management, Order information management.

Browser Login:

- There are 2 text boxes in the login window to define login data.
- This is the link to the registration window and the theme of Tune Source
- Link to SQL database for testing login details
- Set char to * for password

3.1. Login

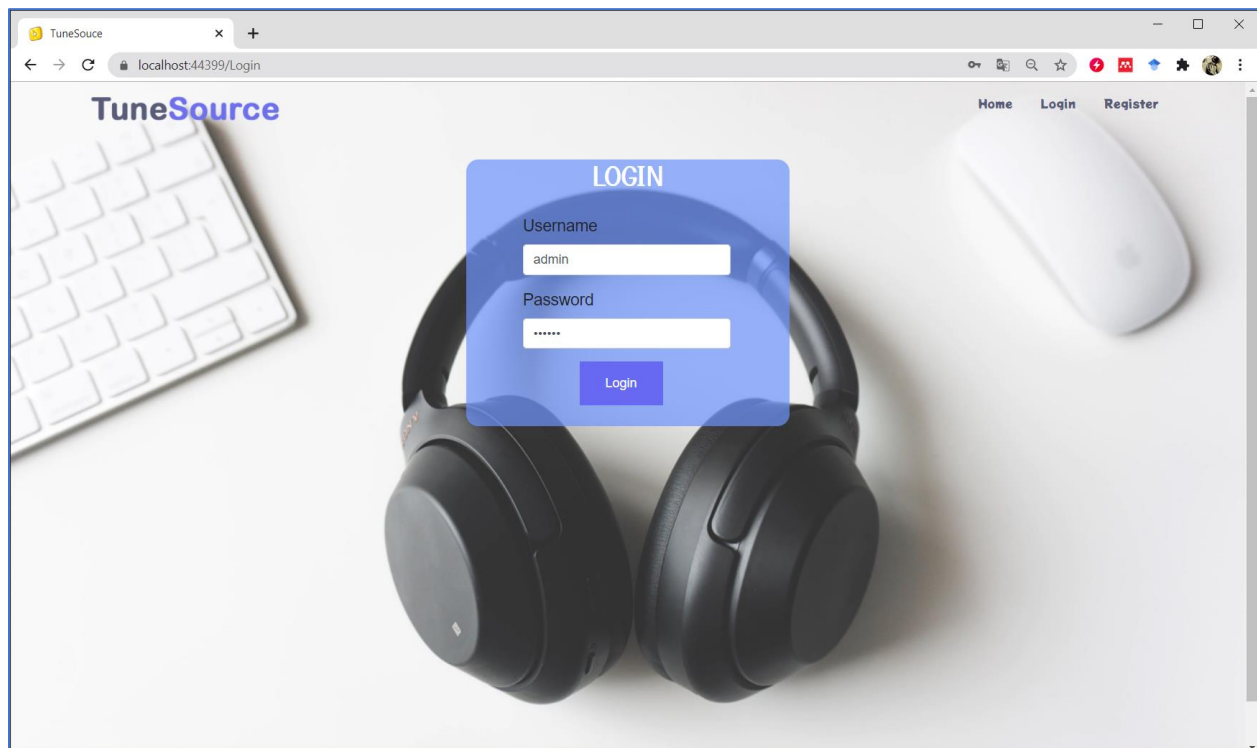
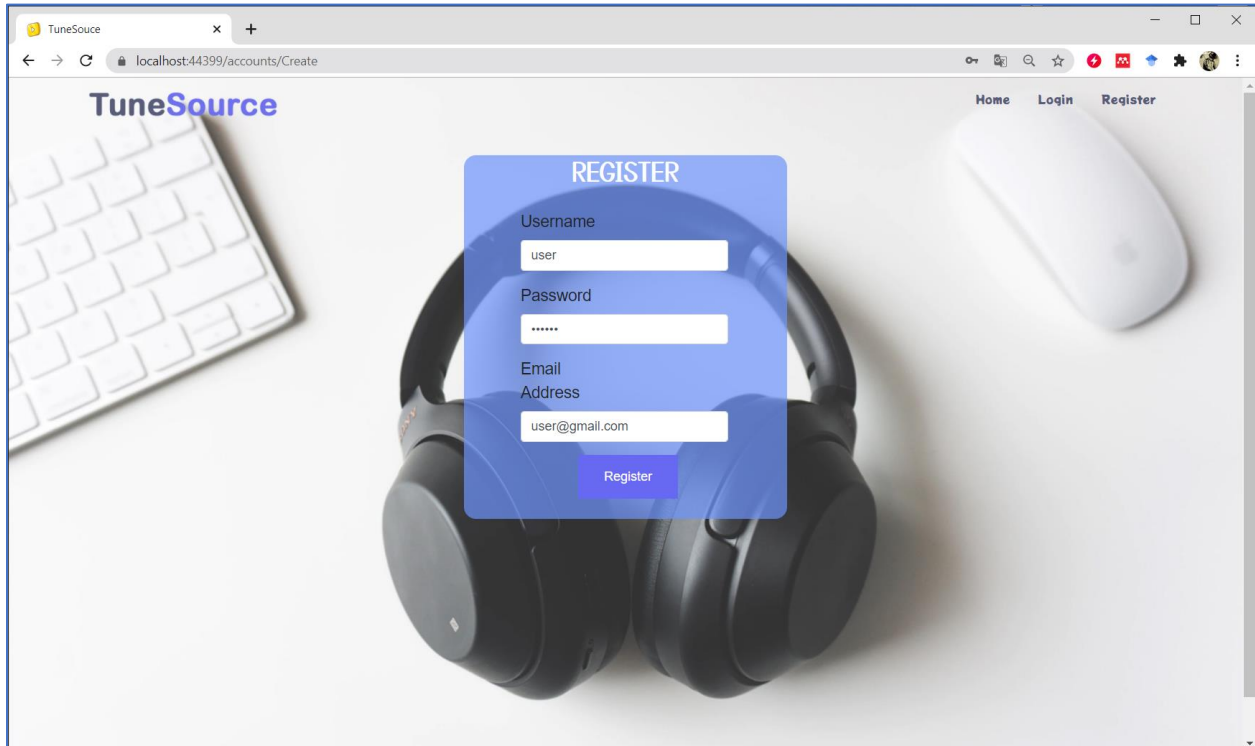


Figure 8: Login page

3.2. Register page



The screenshot shows a web browser window with the URL `localhost:44399/accounts/Create`. The page features the TuneSource logo in the top left and navigation links for Home, Login, and Register in the top right. A blue semi-transparent modal box is centered on the page, titled "REGISTER". It contains four input fields: "Username" with the value "user", "Password" with masked characters "*****", "Email Address" with the value "user@gmail.com", and a "Register" button at the bottom.

Figure 9: Register page

Note: This Register page for the only user.

3.3. Home page of admin account

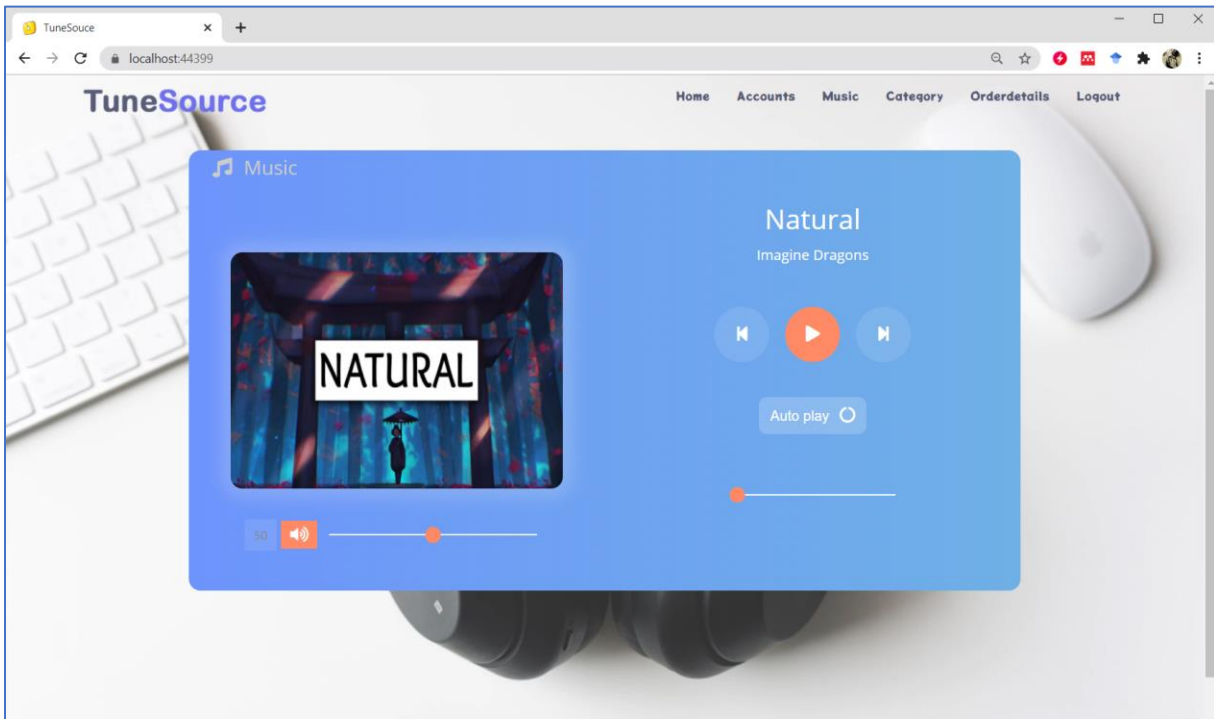


Figure 10: Home page of admin account

3.4. Management Account

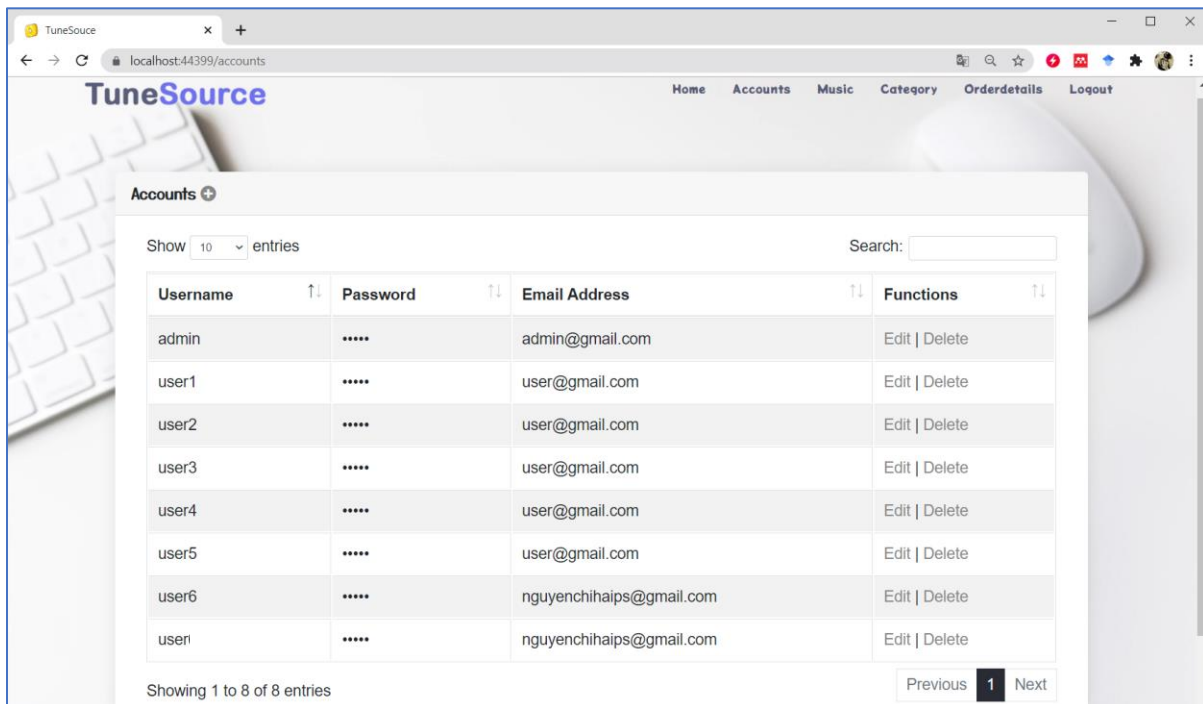


Figure 11: Management account

3.5. Management Music

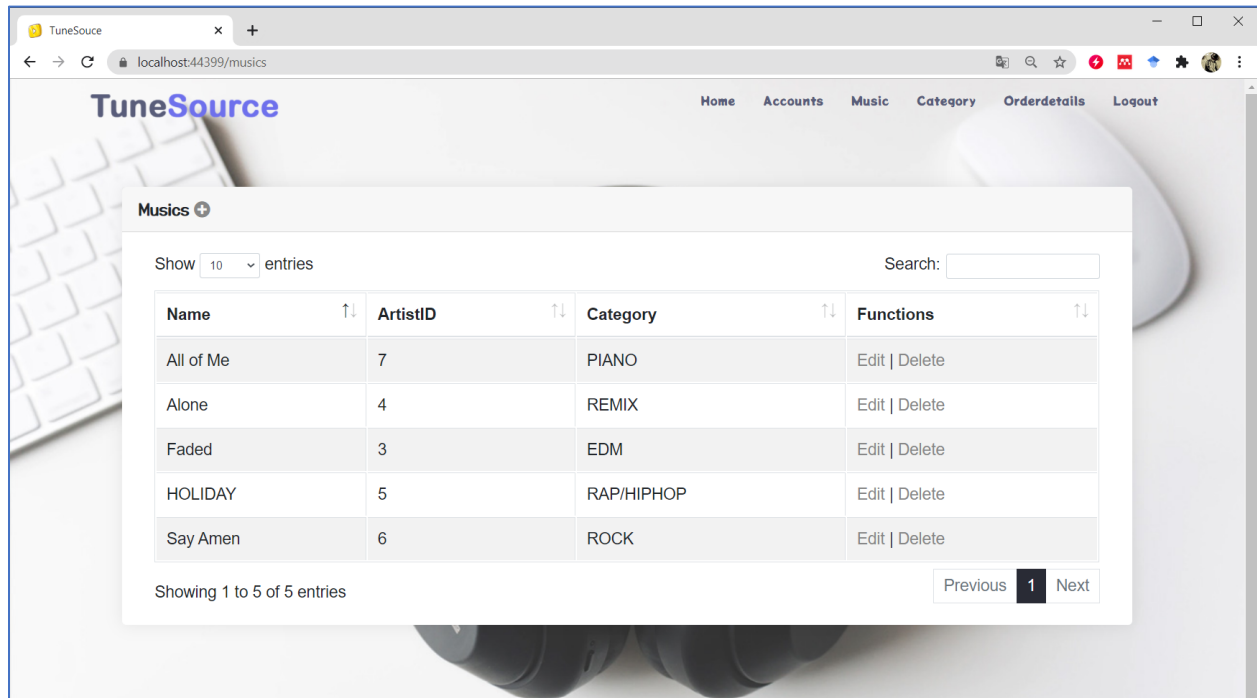


Figure 12: Management Music

3.6. Management Category

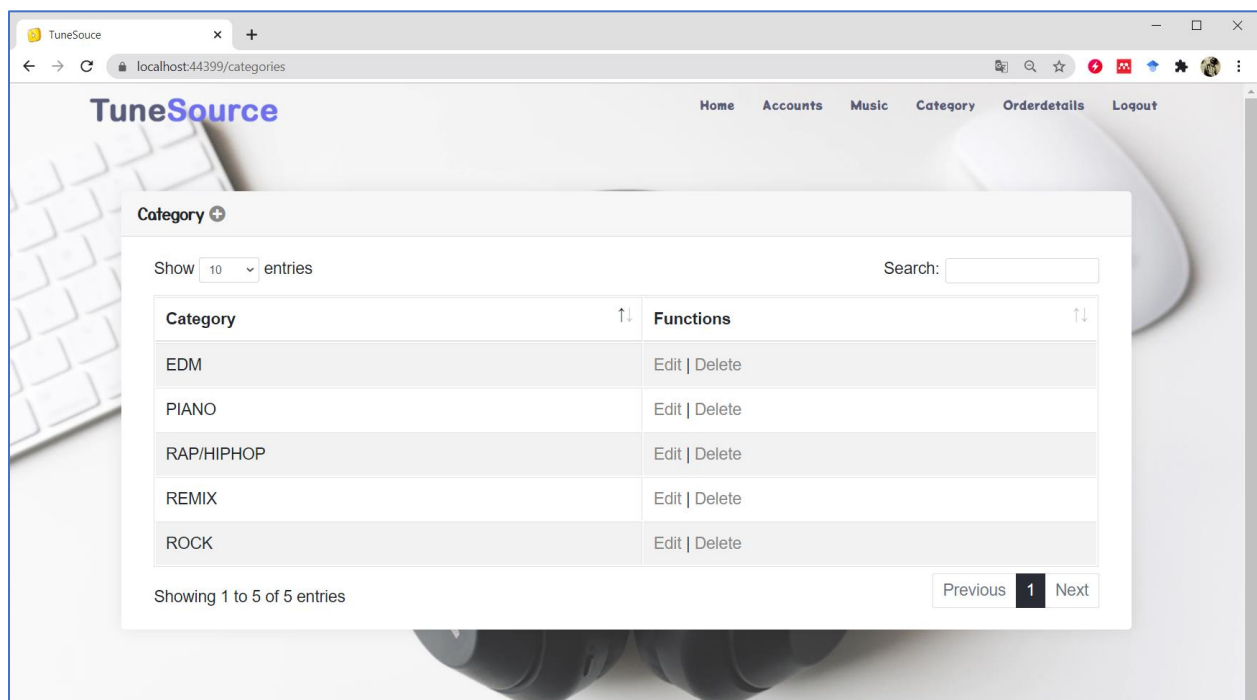


Figure 13: Management category

3.7. Management Order Details

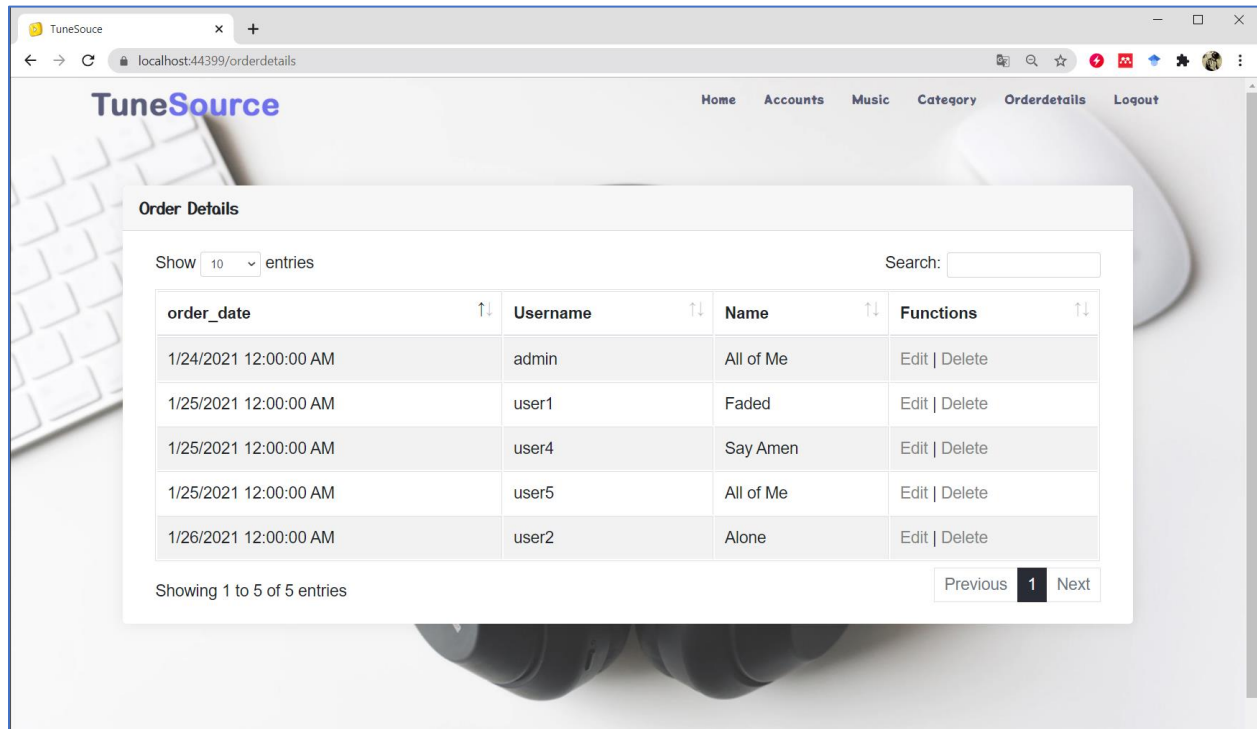


Figure 14: Management Order Details

3.8. Home page of User account

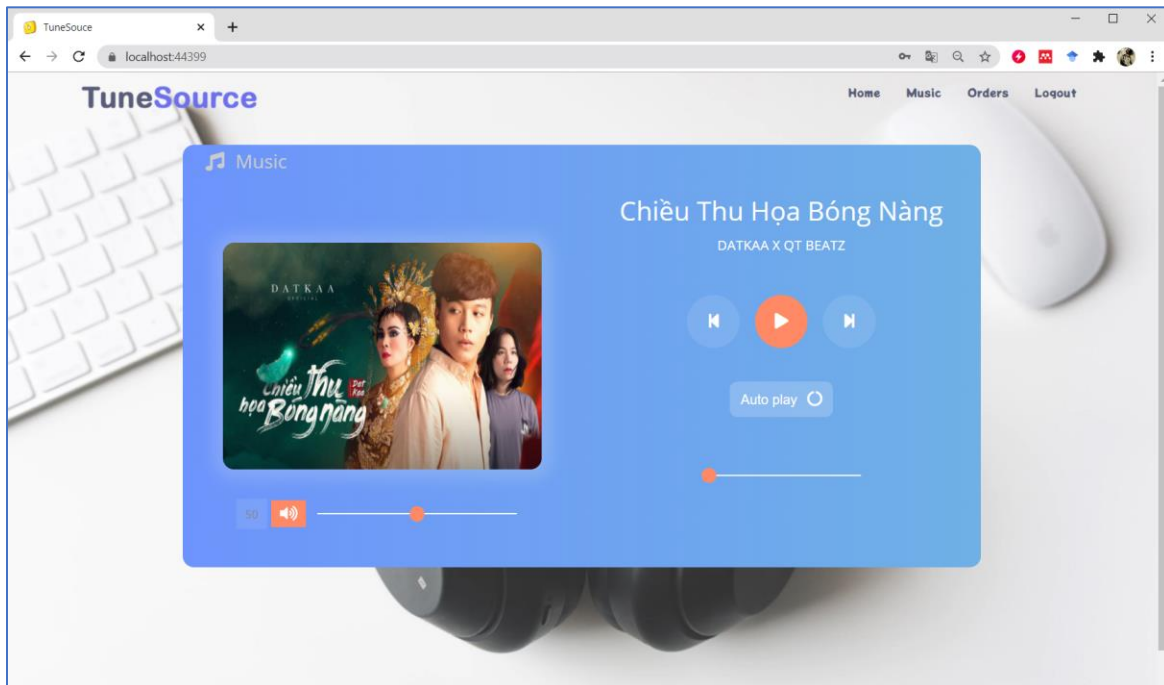


Figure 15: Home page of user account

3.9. Re-View Category of Music

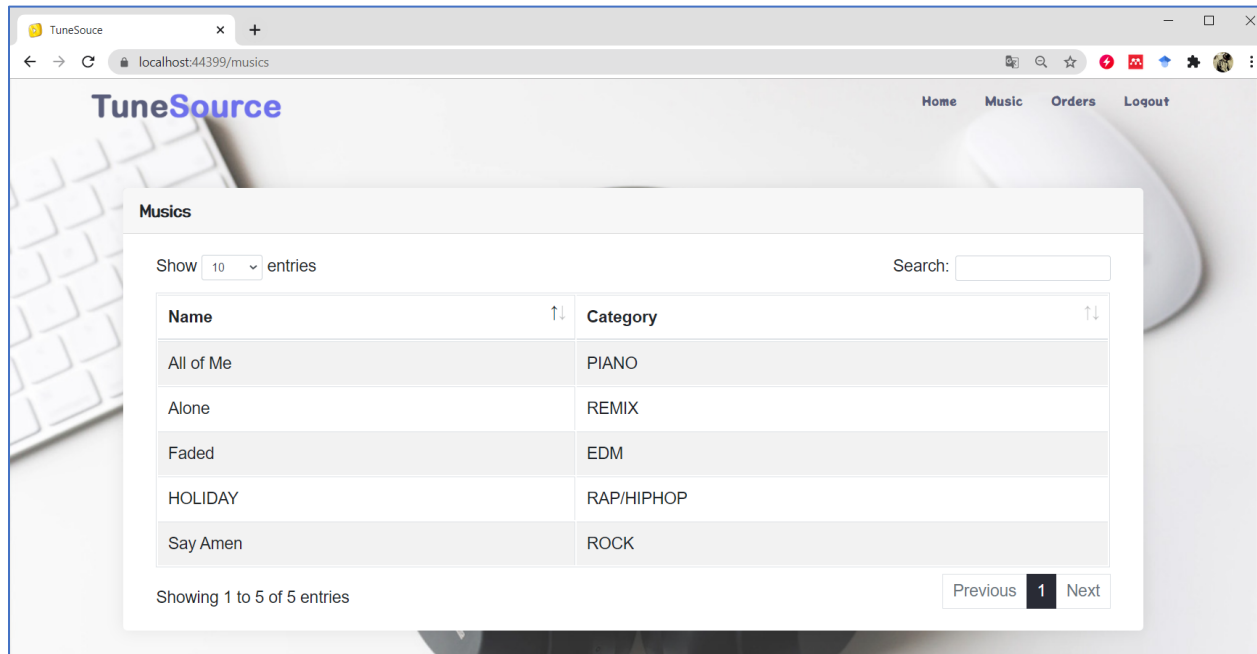


Figure 16: Re-view category of music

3.10. Order

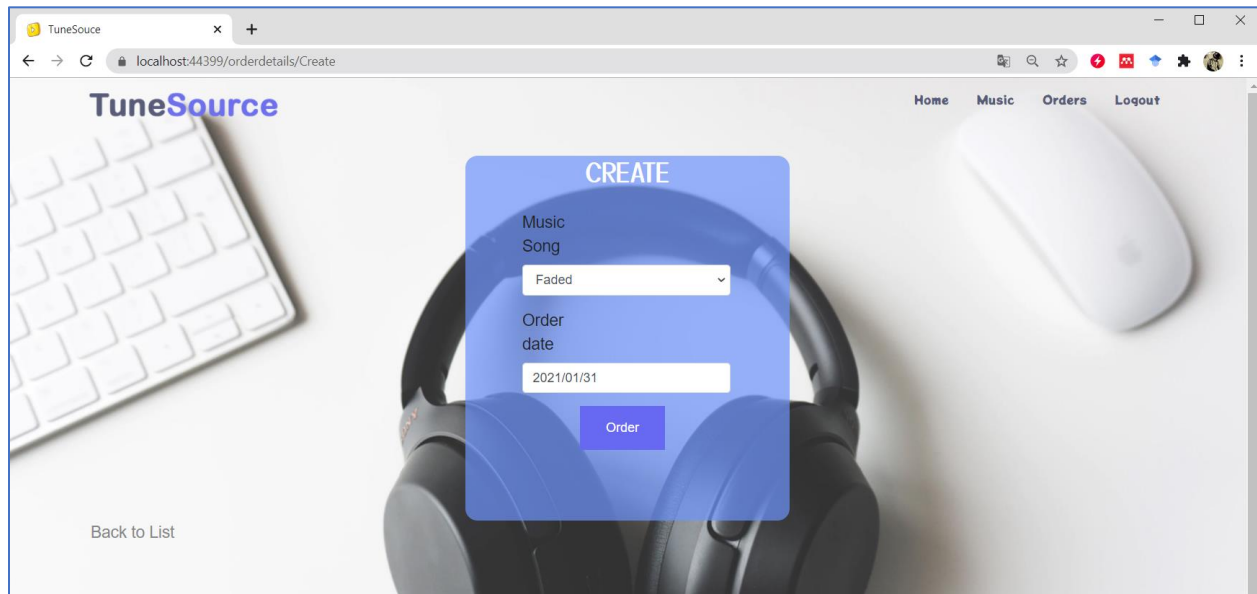


Figure 17: Order Music

3.11. Listening to Music basic

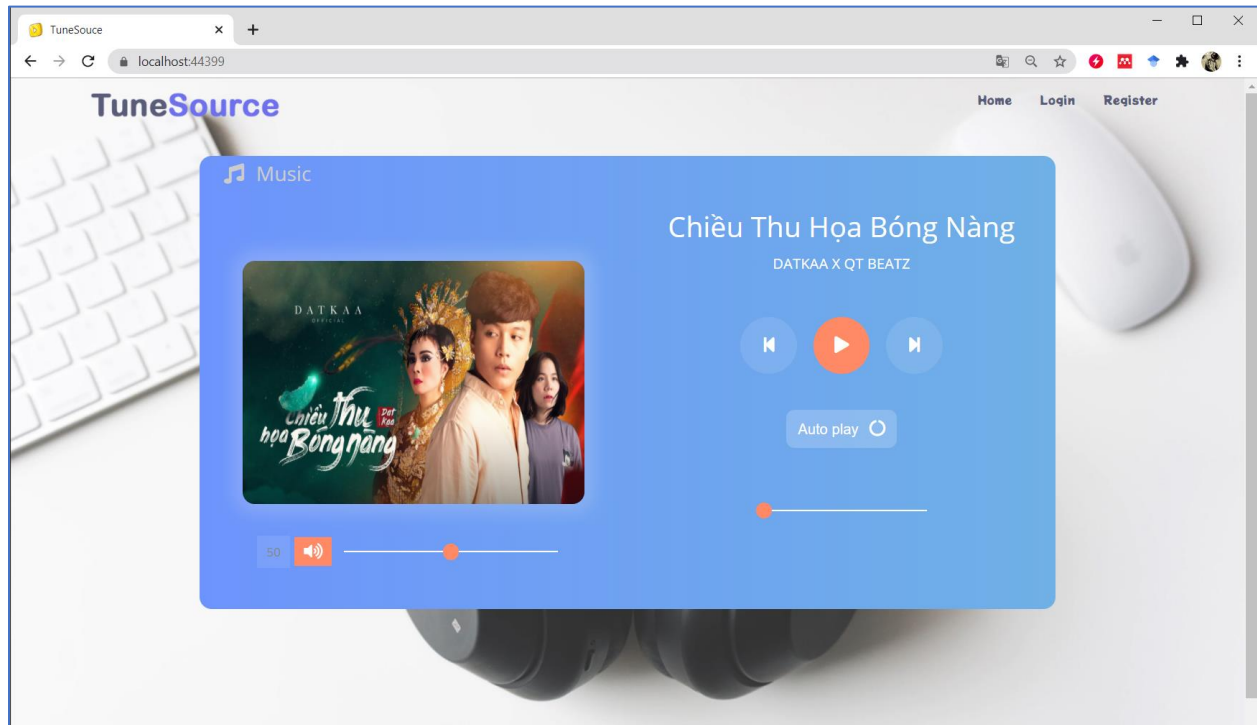


Figure 18: Listening to Music basic

Note: Without having to log in, individuals can listen to simple home page system music results.

Conclusion

We are based on function requirement and have fulfilled these roles based on the functional criteria and suggested by partners. We have consulted other websites for listening and have developed a user-friendly GUI. For further compatibility, we'll upgrade the management GUI.

References

Media, M., n.d. *Mona Media*. [Online]
Available at: <https://mona.media/bao-mat-website-va-nhung-dieu-can-biet-de-toi-uu/>
[Accessed 30 01 2021].