

ASSIGNMENT 02 FRONT SHEET

Qualification	BTEC Level 5 HND Diploma in Computing		
Unit number and title	Unit 09: Software Development Life Cycle		
Submission date	Wednesday, 7 September 2022	Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	
Student Name	Truong Van Tuan Kiet	Student ID	GCC200203
Class	GCC0903	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	

Grading grid

P5	P6	P7	M3	M4	M5	M6	D3	D4

☐ **Summative Feedback:**

☐ **Resubmission Feedback:**

Grade:

Assessor Signature:

Date:

Internal Verifier's Comments:

Signature & Date:

Assignment Brief 02 (RQF)

Higher National Certificate/Diploma in Business

Student Name/ID Number:	
Unit Number and Title:	Unit 09: Software Development Life Cycle
Academic Year:	
Unit Assessor:	
Assignment Title:	Undertake a software development life cycle
Issue Date:	07/12/2020
Submission Date:	
Internal Verifier Name:	
Date:	

Submission Format:
<p><i>Format:</i></p> <ul style="list-style-type: none"> • The submission is in the form of 1 document. • You must use the Times font with 12pt size, turn on page numbering; set line spacing to 1.3 and margins to be as follows: left = 1.25cm, right = 1cm, top = 1cm, bottom = 1cm. Citation and references must follow the Harvard referencing style. <p><i>Submission:</i></p> <ul style="list-style-type: none"> • Students are compulsory to submit the assignment in due date and in a way requested by the Tutor. • The form of submission will be a soft copy posted on http://cms.greenwich.edu.vn/. • Remember to convert the word file into PDF file before the submission on CMS. <p><i>Note:</i></p> <ul style="list-style-type: none"> • The individual Assignment must be your own work, and not copied by or from another student. • If you use ideas, quotes or data (such as diagrams) from books, journals or other sources, you must reference your sources, using the Harvard style. • Make sure that you understand and follow the guidelines to avoid plagiarism. Failure to comply this requirement will result in a failed assignment.

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. 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 Tune Source Project.

Discuss the relationships between the FRs and NFRs.

Word limit: 300 – 400 words.

2. 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. Discuss how you would trace these requirements throughout the project by using Requirement Traceability matrix. You will have to provide real usage of it.

Word limit: 400 – 500 words.

Task 2 – Analysis (2)

Analyze the requirements that you identified in Task 1 using a combination of structural and behavioral 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.

For each diagram, you will have to explain properly.

Word limit: 1000 – 1200 words.

Task 3 – Design

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

1. Discuss how the user and software requirements are addressed in the design phase.
 - You will explain how Mock-up, and Wireframe are used in the project. You should include some of the mockup or wireframe (at least 5) design of the Tune Source 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 technical solution stack could be suitable to implement the project with clear explanations.
2. Discuss how activity diagram and pseudocode are used to specify the software behaviour.
3. Discuss how UML state machine can be used to specify the software behaviour. Differentiate between FSM and extended FSM using the case study.
4. Discuss how the data-driven approach improves the reliability and effectiveness of software.

Word limit: 800 – 1500.

Task 4 – Software quality management

1. Discuss two software quality attributes that are applicable to the project.
2. Discuss two quality assurance techniques that can help improve the software quality in the project.
3. 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 (Assignment 02):			
Learning Outcome	Pass	Merit	Distinction
LO3 Undertake a software development lifecycle	<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>	<p>D3 Critically evaluate how the use of the function design paradigm in the software development lifecycle can improve software quality.</p>
LO4 Discuss the suitability of software behavioural design techniques	<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>	<p>D4 Present justifications of how data driven software can improve the reliability and effectiveness of software.</p>

Table of Contents

Assignment Brief 02 (RQF)	1
Higher National Certificate/Diploma in Business	1
Task 1 – Analysis (1).....	6
I. Identify the stakeholders, their roles and interests in the case study.....	6
1. Review the requirement definition of the project. Clearly indicate which stakeholder(s) provide what requirements.	6
2. Identify FRs and NFRs of Tune Source Project.	6
3. Discuss the relationships between the FRs and NFRs.....	7
II. Discuss the techniques that would use to obtain the requirements	8
Task 2 – Analysis (2).....	11
1. Use Case Diagram for the whole system.....	11
2. Use Case specification for 2 Use cases.	12
3. Context Diagram for the whole system.	14
4. Data Flow Diagram – Level 0 for the whole system	15
5. ERD for the whole system.....	16
Task 3 – Design	17
I. Mock-up of the project.....	17
1. Sign up function.	17
2. Login function	18
3. Homepage.....	18
4. Listen to music.	19
5. Pick Premium	19
II. Explanation architecture in the project	20
III. Address which technical solution stack could be suitable to implement the project with clear explanations.....	20
References	22

Assignment 2

Task 1 – Analysis (1)

I. Identify the stakeholders, their roles and interests in the case study.

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

The user is the main object of the Tune Source project. Users can view trending music lists, listen to music, search for music by song name and artist name and register for an account. After successful account registration, users can pay fees to download music, give gift music, and receive gifts gift music from other users.

A producer or singer is a person who creates music products for users. Producers can upload new music, update music when there is an error or they want to improve the quality of the product and delete the music product when there is an unexpected error or problem. They also extend all the functionality of the User.

Admin has the role of managing user information, verifying the accuracy of user information. Verifying community standards and legality of music products uploaded and updated on the website by producers or singers.

2. Identify FRs and NFRs of Tune Source Project.

According to (Geeks For Geeks, 2020), the success of a system or software project may be evaluated thanks to the extremely important procedure known as requirements analysis. Functional and non-functional requirements are the two main categories of requirements.

The success of a system or software project may be evaluated thanks to the extremely important procedure known as requirements analysis. Functional and non-functional requirements are the two main categories of requirements.

The success of a system or software project may be evaluated thanks to the extremely important procedure known as requirements analysis. Functional and non-functional requirements are the two main categories of requirements.

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

Following are the differences between Functional and Non-

Functional Requirements	Non-Functional Requirements
A functional requirement defines a system or its component.	A non-functional requirement defines the quality attribute of a software system.
It specifies “What should the software system do?”	It places constraints on “How should the software system fulfill the functional requirements?”
Functional requirement is specified by User.	Non-functional requirement is specified by technical peoples e.g. Architect, Technical leaders and software developers. It is not mandatory.
It is mandatory.	It is captured as a quality attribute.
It is captured in use case.	Applied to a system as a whole.
Defined at a component level.	Helps you to verify the performance of the software.
Functional Testing like System, Integration, End to End, API testing, etc are done.	Non-Functional Testing like Performance, Stress, Usability, Security testing, etc are done.
Usually easy to define.	Usually more difficult to define.
Example 1) Authentication of user whenever he/she logs into the system. 2) System shutdown in case of a cyber attack. 3) A Verification email is sent to user whenever he/she registers for the first time on some software system.	Example 1) Emails should be sent with a latency of no greater than 12 hours from such an activity. 2) The processing of each request should be done within 10 seconds 3) The site should load in 3 seconds when the number of simultaneous users are > 10000

3. Discuss the relationships between the FRs and NFRs.

FRs and NFRS requirements lay the foundation for a software development project's success. To stimulate software development, specific qualities were mentioned in functional requirements. Software features are directly linked to non-functional requirements. Non-functional requirements are product criteria that define how a project should be implemented to produce an end-user experience from the user's perspective. When the required product features are determined and the requirements for each of those features are established, the software development process is considerably accelerated. The project will suffer greatly if the functional and nonfunctional requirements are not specified.

NFRs explain a product's or application's entire experience, such as security or performance, whereas FRs describe a specific function.

User stories, use cases, and functional scores can all be used to collect FRS. NFRs are frequently found throughout a product, particularly in the user experience and user interface (UI/UX). On the same project, both NFR and FR must exist side by side.

II. Discuss the techniques that would use to obtain the requirements

Taking customer requirements is very important, it determines the direction and quality of the entire Tune source project. There are some techniques to obtain the requirement:

Interviews

According to (Jama Software, 2022), the process of eliciting needs may be started off well with interviews. They are useful for compiling background data on organizational requirements, user and customer issues, and support personnel and other key stakeholders' worries. Interviews can also be used as a follow-up to collect information in greater depth.

A varied and representative cross-section of the system's stakeholders should be covered in the interviews. The whole spectrum of user and customer profiles should be included. To ensure that your system requirements aren't biased in favor of one group, it is important to acquire a proper perspective on conflicting needs.

Asking open-ended questions is crucial while conducting interviews. A simple "yes" or "no" response cannot be given in response to an open-ended question. They extract certain data. They ask the interviewee to clarify their ideas and give reasoning, which gives the context of the requirements and can be evaluated and validated.

During the interview, you should also ask a ton of follow-up questions. Good follow-up inquiries either pull up to acquire a broad picture of the background or delve down for further specifics. Some people have a propensity to discuss details and exceptions. You'll have to pull up with them. Some people will discuss background without ever going into detail. You'll have to dig a little further with such people.

User Observation

One of the best ways to understand what users truly need is to observe them performing their daily tasks.

User observation can be either passive or active. Active observation—asking questions of users while observing them—is the best approach for gaining an understanding of an existing process. Passive observation is more effective when gathering user feedback on a design prototype (see technique #11).

When observing users, record the actions and activities that take place. What already works well? What causes users difficulty? Note the obstacles users must routinely overcome.

By observing end users in the real context in which they perform their tasks, you'll gain a true understanding of what they are up against and what improvements they need so they can perform better. You'll then be better able to specify a system that successfully reinvents users' processes and grants them far greater productivity and usability, rather than simply providing them an incremental improvement.

JAD (Joint Application Development)

According to (TechTarget, 2022), Through a series of interactive workshops called JAD sessions, the client or end user is involved in the design and development of an application using the JAD methodology. JAD was created by IBM employees Chuck Morris and Tony Crawford in the late 1970s, and it was first taught through workshops in 1980.

Because the client is involved at every stage of the development process, the JAD technique is said to result in quicker development timelines and more client satisfaction than the more conventional practice. In contrast, the conventional method of systems development entails the developer researching the needs of the system and creating an application with customer input coming through a series of interviews.

Rapid application development (RAD), a version on JAD, produces an application more quickly by employing techniques including using fewer formal procedures and reusing software components.

The tune source project is a relatively small project that is suitable for using the waterfall model. It is extremely important to get customer information, the information needs to be clear and detailed to avoid changes later. Clear information helps the project to be completed quickly and with high quality. So, choose the interview technique to get requirements from customers that will be used for this project.

The interview method includes 2 processes, the Formal Interview Process, and the Informal Interview Process. These are the steps to implement.

- Formal Interview Process Steps:

1. Identify stakeholders to be interviewed
2. Obtain a general understanding of the customers business
3. Develop interview questions using open-ended questions
4. Set meeting time and location for the interview
5. Provide a set of questions to interviewees prior to the interview (if they will need to prepare for the interview)
6. Use one or more Recorders to accurately preserve results of the interview
7. Provide results to interviewees for confirmation of content

- Informal Interview Process Steps:

1. Identify stakeholders to be interviewed
2. Obtain a general understanding of the customers business
3. Develop interview questions (for interviewer's use only) to make sure certain questions are answered during the session

4. Set up a casual meeting or telephone conversation time for the interview.
5. Takes handwritten notes during the interview; avoid using electronic data capture.
6. Provide results to interviewee for confirmation of content

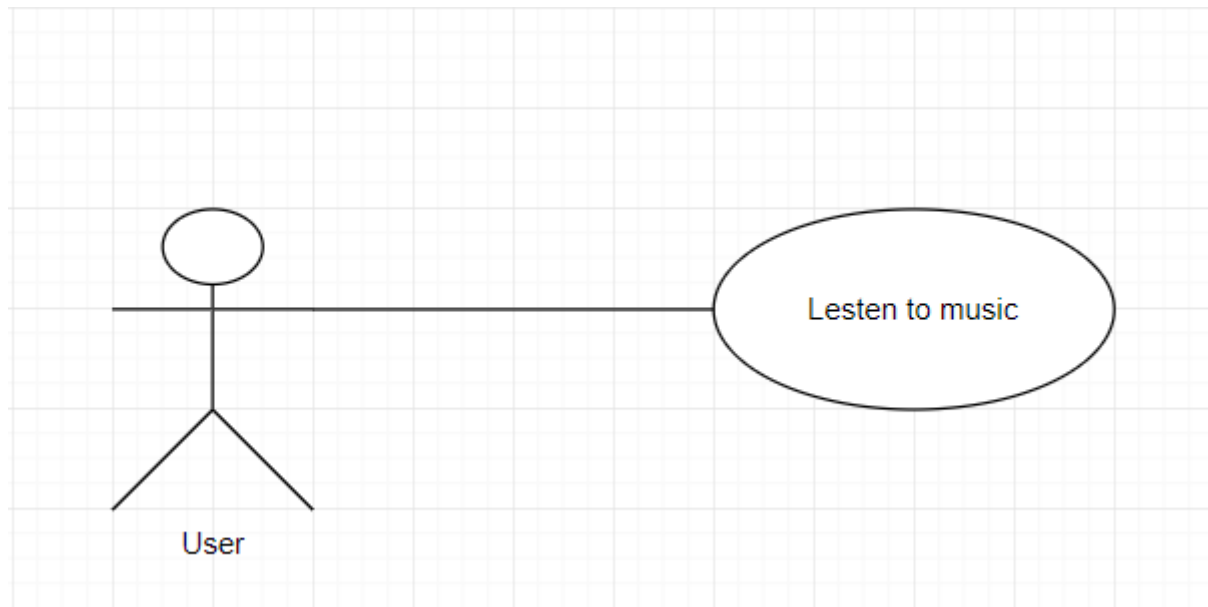
Task 2 – Analysis (2)

1. Use Case Diagram for the whole system.



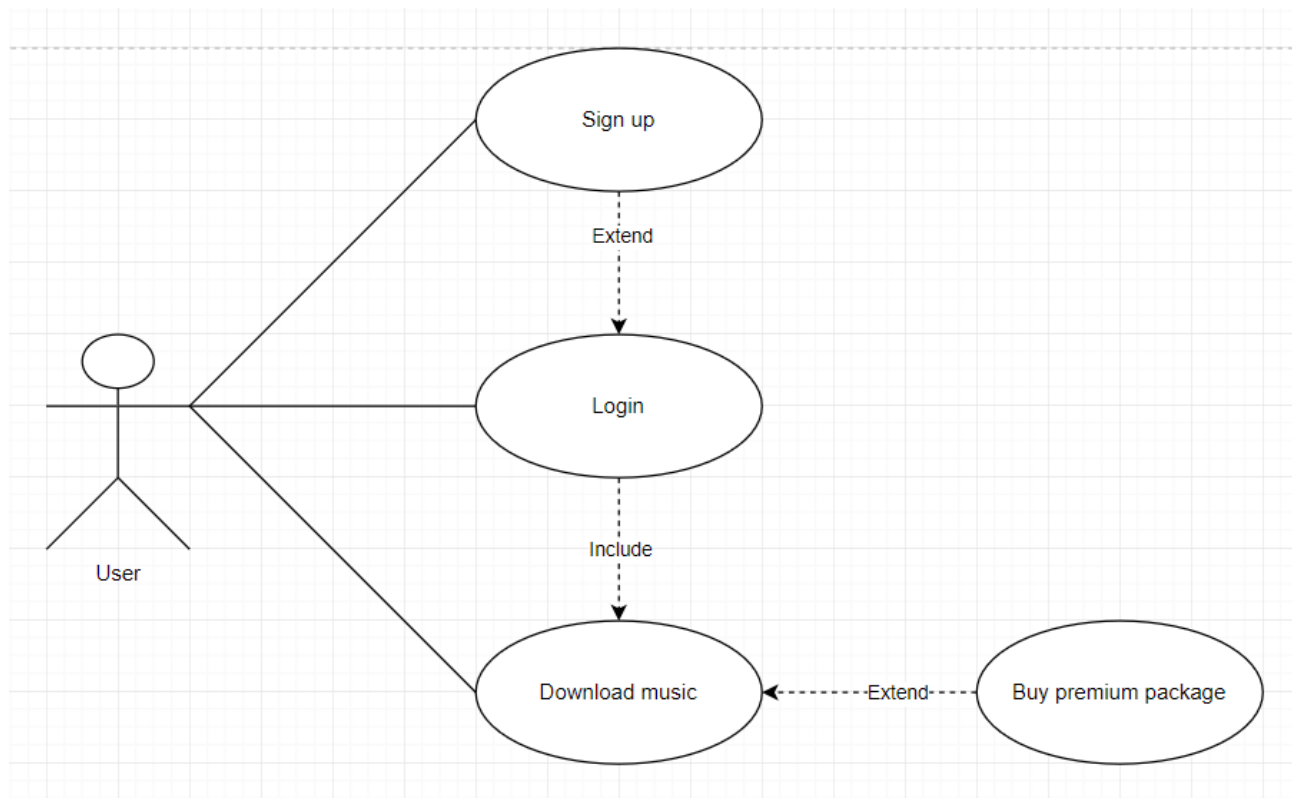
2. Use Case specification for 2 Use cases.

-Listen to music.



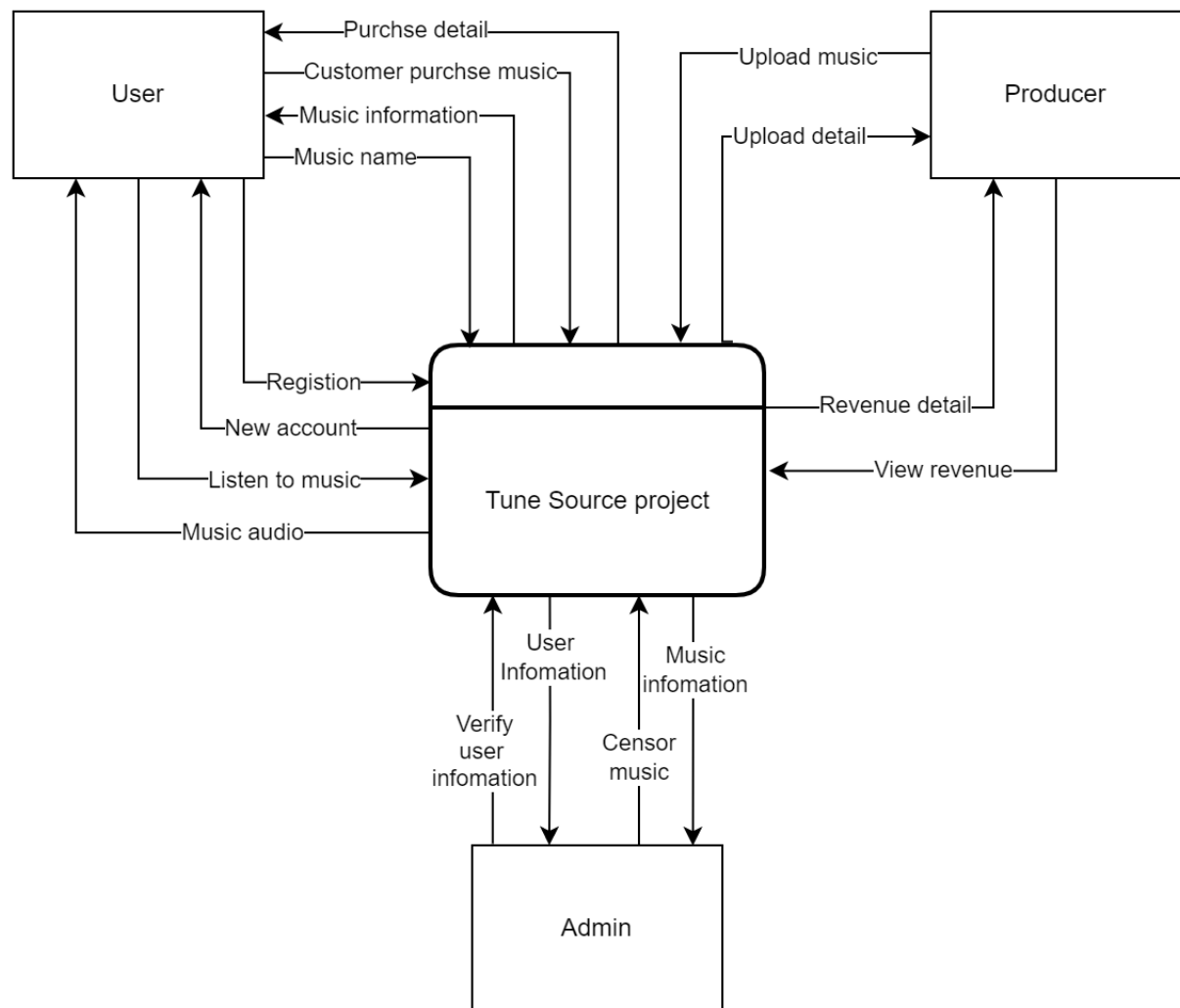
Use Case Name: Listen to music		ID: F03	Priority: high
Actor: User			
Description: This is the main function of the project. When the user clicks on a song, the system will select it in the database after displaying the song on the interface and playing that song.			
Trigger:			
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal			
Preconditions:			
<div><div>1.</div>User must click select the song and, press the play button</div> <div><div>2.</div>System is always online</div> <div><div>3.</div>Song or singer must be in the database</div>			
Normal Course:		Information for Step:	
<div><div>1.</div>The system takes the key provided by the user when the user clicks to select a song, then selects it in the database.</div> <div><div>2.</div>Display the song on the interface and play it.</div>		Key ID of the song	
		Display and play the song	
Alternative Course:			
Postconditions: none			
Exceptions:			
Summary Inputs	Source	Outputs	Destination
<div>- Music ID</div> <div>- Press play button</div>	<div>User</div> <div>Database system of Tune</div> <div>Source</div>	<div>- Display on interface</div> <div>- Play music</div>	<div>Interface to display and</div> <div>play music</div>

- Download music.

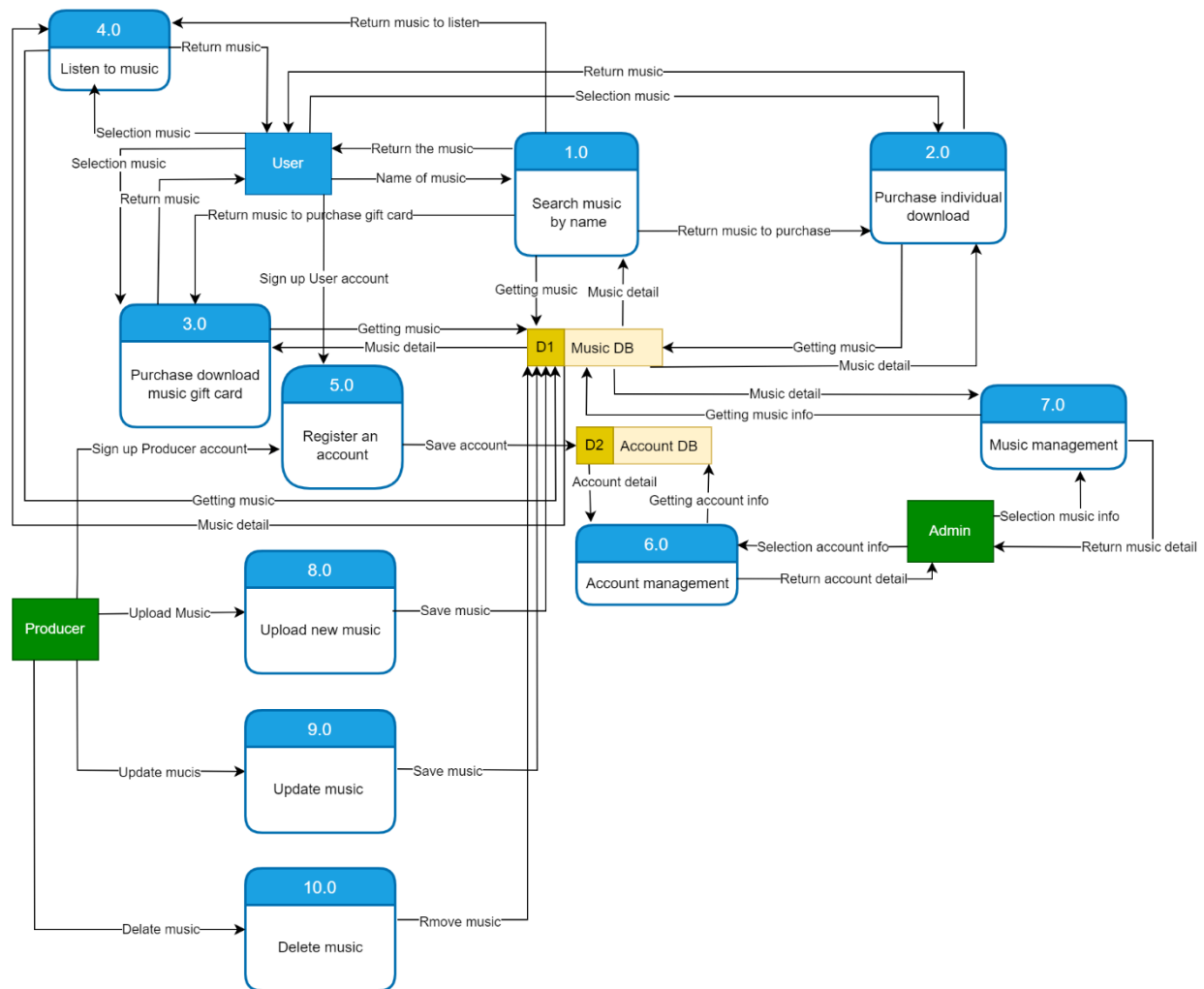


Use Case Name: Download music		ID: F07	Priority: high
Actor: User			
Description: When user clicks Download button on music that user choose to download, system will select the music in DB and send it to user device.			
Trigger: Download button			
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal			
Preconditions: <ol style="list-style-type: none"> 1. User must click select the song and, press the download button 2. System is always online 3. Song or singer must be in the database 			
Normal Course: <ol style="list-style-type: none"> 1. The system takes the key provided by the user when the user clicks to select a song, then selects it in the database. 2. Select the music in DB and send it to user device. 		Information for Step: Key ID of the song Send the music to user device	
Summary Inputs	Source	Outputs	Destination
Music ID Download button	User Database	Send music to user device	Download detail

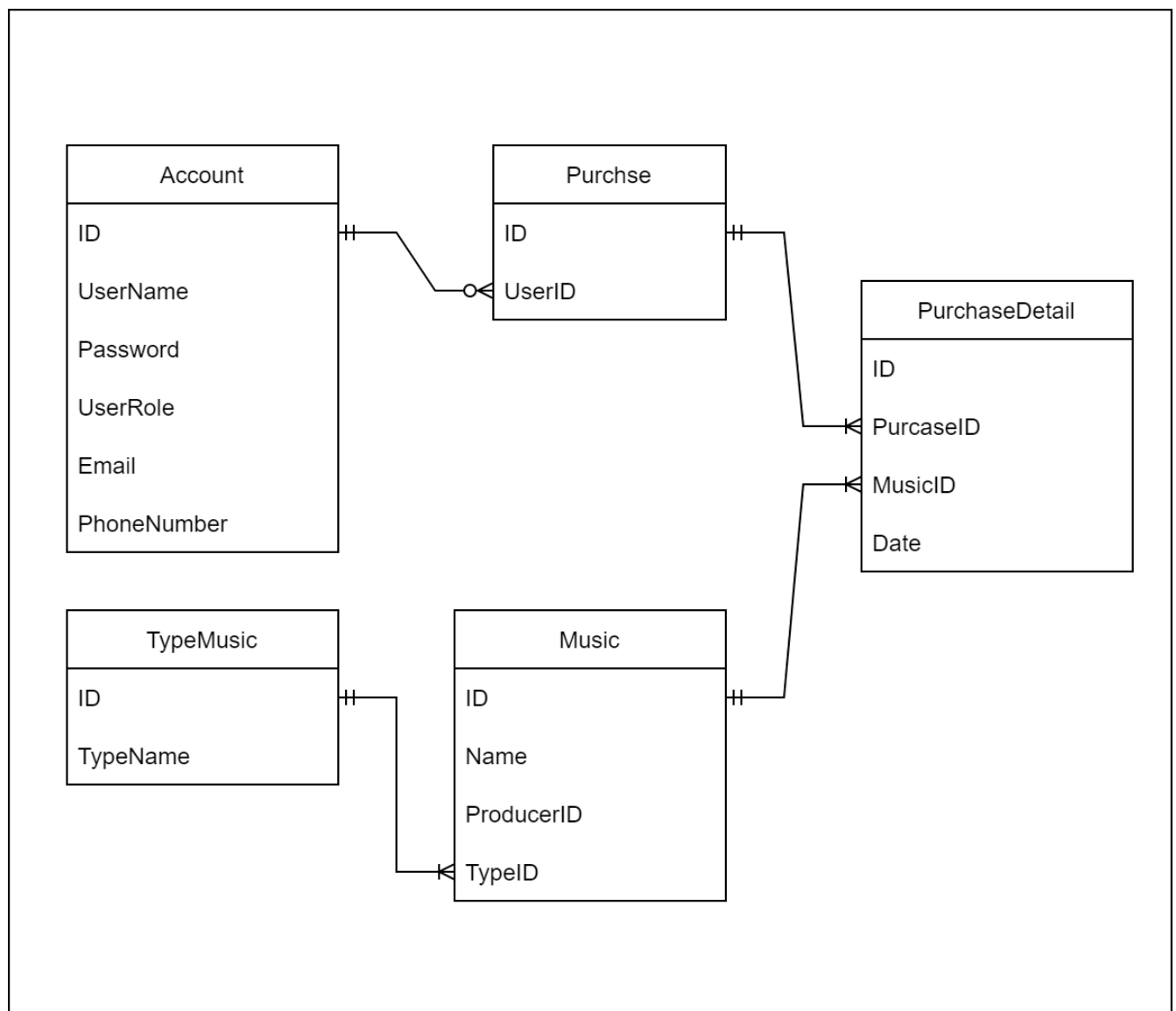
3. Context Diagram for the whole system.



4. Data Flow Diagram – Level 0 for the whole system



5. ERD for the whole system




Task 3 – Design


I. Mock-up of the project

1. Sign up function.

User sign up a new account by Facebook, Gmail, Email.

Sign up for free to start listening.

 Sign up with Facebook

 Sign up with Google


or

What's your email?

[Use phone number instead.](#)

Confirm your email

Create a password

☐ I'm not a robot  reCAPTCHA
[Privacy](#) - [Terms](#)


Sign up


Have an account? [Log in](#).


2. Login function

User can login with Facebook, Apple ID, Gmail, account that them registered.

To continue, log in to Tune Source.

 CONTINUE WITH FACEBOOK

 CONTINUE WITH APPLE

 CONTINUE WITH GOOGLE

CONTINUE WITH PHONE NUMBER

OR

Email address or username

Email address or username

Password

Password

Forgot your password?

☒ Remember me

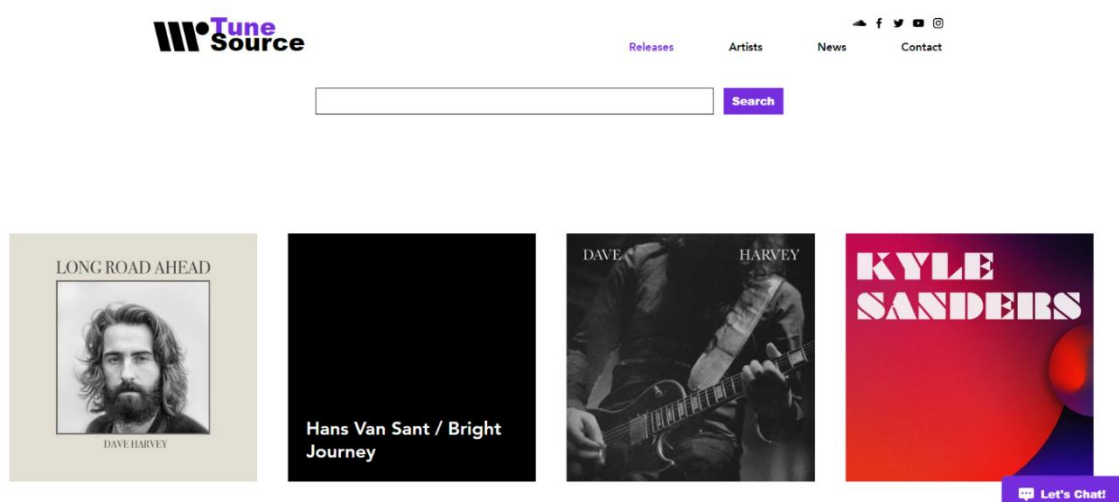
LOG IN

Don't have an account?

SIGN UP FOR SPOTIFY

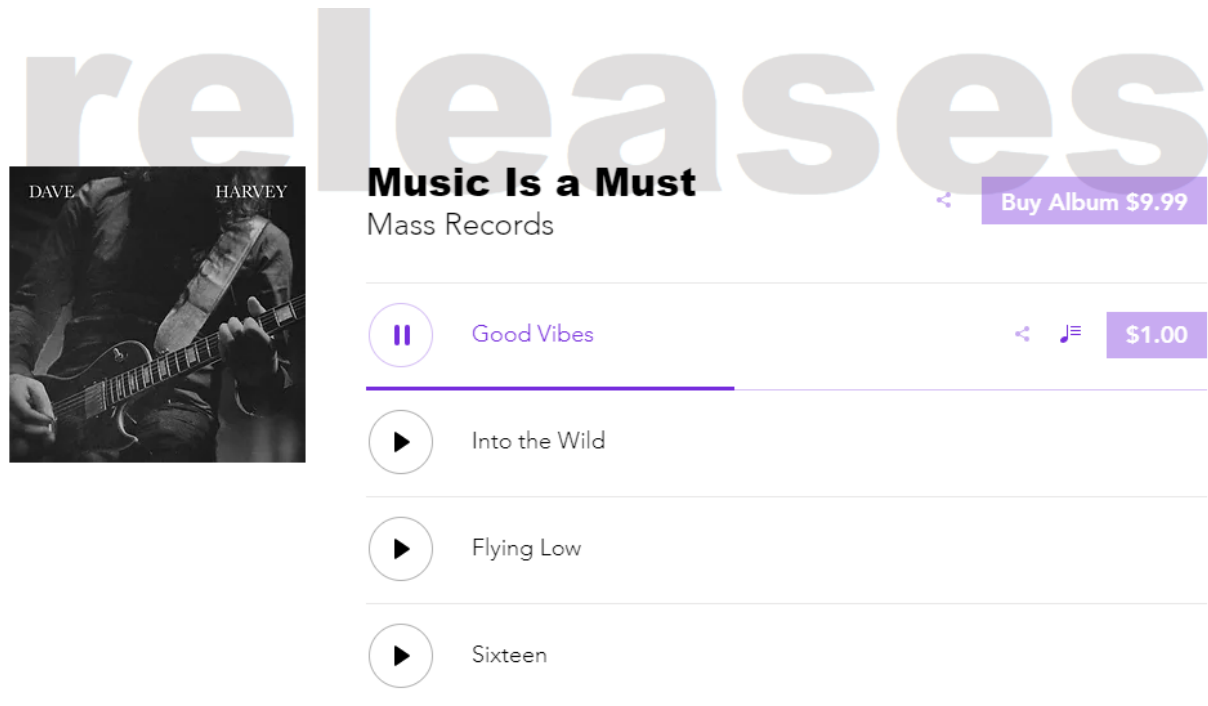
3. Homepage

This is the homepage, where the system displays trendy albums, and user can choose and listen.



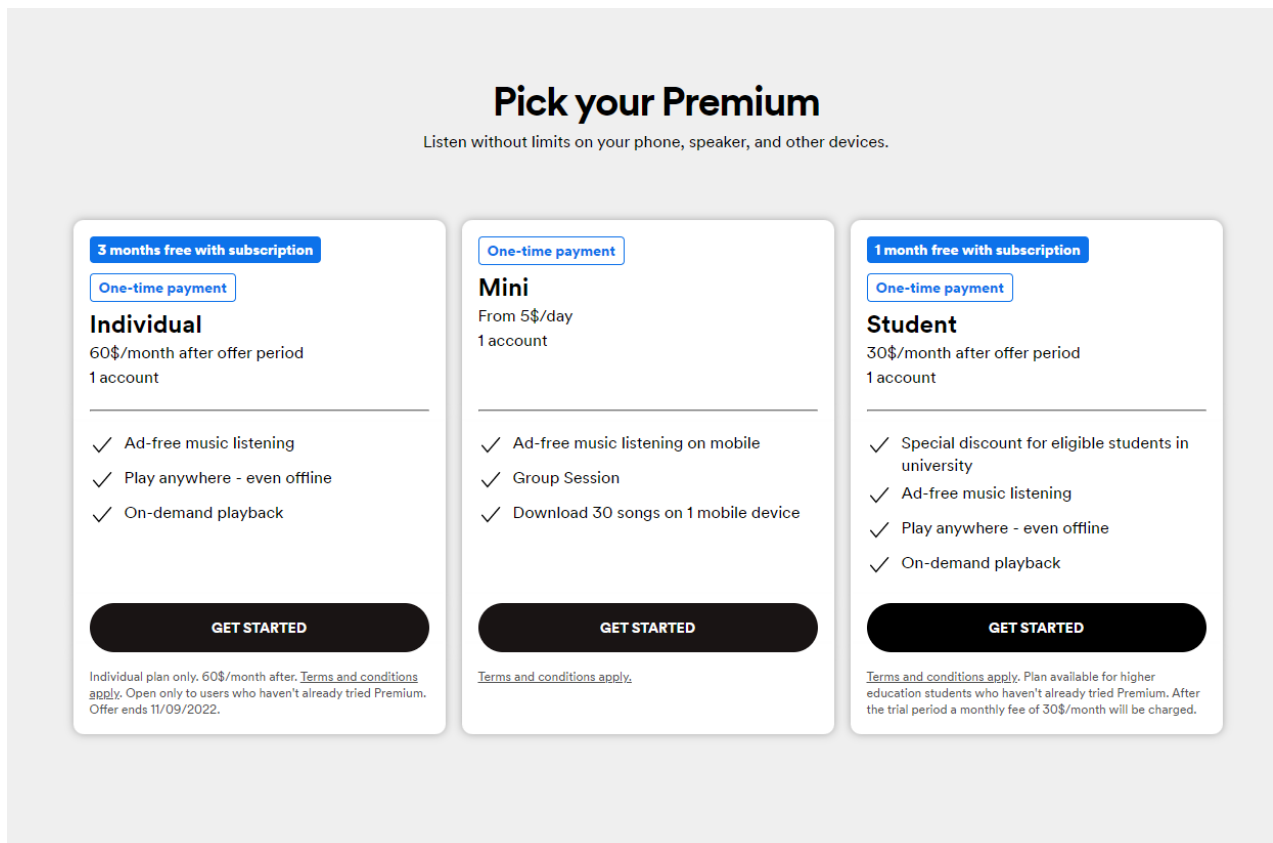
4. Listen to music.

User can listen to music for free and pay to download.



5. Pick Premium

Users can choose and purchase premium matching packages.



II. Explanation architecture in the project

I choose the client-server model for the tune source project because of the benefits it brings:

Concentrate

The first advantage of the Client-Server model of a client-server network is the built-in centralized control. Under this model, all necessary information will be placed in a single location. This is an extremely useful advantage loved by network administrators because they have full control over everything.

This feature helps all problems in the network to be solved in one unified place. At the same time, updating the resource base and data will also be a lot easier.

Security

In the Network Client Server, all data will be protected with maximum protection thanks to the central system of the network. Thereby, it will help users check access so that only those who are granted access can perform the necessary operations.

To do so, we need to apply login information as well as Username or Password. Besides, if our data is lost, the files will be restored extremely easily from a single backup only.

Expanding possibilities

The model networked client server is extremely scalable. Only users need to use it at any time, they can also increase their number of resources. For example, the Client or Server number. Thanks to that we can increase the size of the Server easily without much interruption.

Access Capability

There is absolutely no distinction between positions or backgrounds. All clients have the ability to log on to the system network computer. This thing will help all people can to access the information of the company is an easy way but not need to use a terminal mode or other processors.

III. Address which technical solution stack could be suitable to implement the project with clear explanations.

In my Tune Source project, I use the Symfony framework for programming. Symfony has advantages:

- Easy extension operation, capable of integrating other libraries.
- Easy to install, configure compatible with most operating systems, and ensure to work well on standard *nix (Linux and Unix) and Windows operating systems.
- Independent of the database.
- Wide application in all cases including complex cases.
- Preconfigured operation.
- The source code is easy to read, uses phpDocumentor style annotations, and maintenance is quick and easy.
- Suitable for business environment with information technology policies and knowledge.
- Stable for long-term projects.
- Follow best practices and design patterns.

In addition, I also use the administrative system's my SQL database to store data. Sldl my sql system manager is 1 the management csdl source code open, common variable and used by many actors. I choose the priority points in my SQL:

- Ease of use: MySQL is a stable and high-speed database, the tool is easy to use and works on a wide range of operating systems providing great functional utilities.
- High security: MySQL is suitable for applications with databases accessing the internet because it possesses many high-security features, even advanced security.
- Multi-feature: MySQL can support a wide range of SQL functions from both direct and indirect database management systems.
- Especially the structure of the Symfony framework project follows the MVC model. MVC makes it easy to test and detect errors. MVC separates the Model, Controller, and View parts from each other. Applications can be easily maintained because they are isolated from each other.

References

Geeks For Geeks, 2020. *Geeksforgeeks.org*. [Online]

Available at: <https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/>

Jama Software, 2022. *www.jamasoftware.com*. [Online]

Available at: <https://www.jamasoftware.com/requirements-management-guide/requirements-gathering-and-management-processes/11-requirements-gathering-techniques-for-agile-product-teams>

TechTarget, 2022. [Online]

Available at: <https://www.techtarget.com/searchsoftwarequality/definition/JAD>