

ASSIGNMENT 02 FRONT SHEET

Qualification	BTEC Level 5 HND Diploma in Computing		
Unit number and title	Unit 09: Software Development Life Cycle		
Submission date	27/08/2023	Date Received 1st submission	
Re-submission Date	04/09/2023	Date Received 2nd submission	
Student Name	Dinh Dinh Khoi	Student ID	GCC210345
Class	GCC1003	Assessor name	Nguyen Kim Khanh
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
I. Analysis (1).	6
1. Identify the stakeholders, their roles and interests in the case study (P5).	6
a. Review the requirement definition of the project. Clearly indicate which stakeholder(s) provide what requirements.	6
b. Identify FRs (Functional requirements) and NFRs (Non-Functional requirements) of Tune Source Project.....	6
c. Discuss the relationships between the FRs and NFRs.....	7
2. Discuss the technique(s) you would use to obtain the requirements (P5).	8
II. Analysis (2).	9
1. Use Case Diagram for the whole system.....	9
2. Use Case specification for 2 Use cases.	9
3. Context Diagram for the whole system.	12
4. Data Flow Diagram – Level 0 for the whole system.	13
5. ERD for the whole system.....	13
III. Design.....	14
1. Discuss how the user and software requirements are addressed in the design phase (P7).	14
a. Wireframe.....	14
b. Explain architecture in project.....	16
c. Address which technical solution stack could be suitable to implement the project with clear explanations.....	17
2. Discuss how activity diagram and pseudocode are used to specify the software behavior (P7).	17
a. Activity diagram.....	17
b. Pseudocode.....	18
IV. References.	19

I. Analysis (1).

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

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

Tune Source is a music company known for selling Jazz, Rock, Country, and Folk records in California. The main audience of this company is Users. Users can register for an account, search for songs by name or by artist name. After completing account registration, Users can download songs for free or purchase songs for a fee.

A producer is a person who creates music products for commercial purposes. Producers can upload new songs, update song information, and delete songs when there is an unexpected problem.

Admin has the role of managing user's information and censorship of songs.

b. Identify FRs (Functional requirements) and NFRs (Non-Functional requirements) of Tune Source Project.

Functional Requirements: These are the standards that the system must meet in order to provide the fundamental services that the end user has requested. As specified in the contract, each of these features must be included in the system. These are shown or articulated in terms of the input to be provided to the system, the action taken, and the results anticipated. They are essentially the user-stated criteria that, in contrast to non-functional requirements, are visible immediately in the finished product.

- Functional Requirement of Tune Source:

- Searching songs by name or artist name: Users can search for songs by song title or by artist name. The system will classify and give a list of songs to the user.
- Download songs: Users can download free songs on the Tune Source system to their devices.
- Purchase songs: Besides the free songs that users can download, there are also paid songs. So, users must pay to buy those songs. After payment, the system will authenticate and notify the user.
- Upload songs: After creating a new product, the Producer will upload the new product to the system.

- Update song information: Producer can modify song information and then update new information on the system.
- Delete songs: When the recording of the song is damaged or the song has copyright issues, etc. The producer will remove the song from the system.
- Manage user's information: After receiving user's information, Admin will check and perform user's information management tasks.
- Censorship of songs: Admin performs the job of censoring songs. If there is a problem, Admin will immediately notify the system.

Non-functional requirements: Essentially, these are the standards of quality that the system must meet in order to comply with the project contract. The importance or degree to which these criteria are applied varies from project to project.

- Non-functional Requirements of Tune Source:
 - Websites can run on many popular browsers today.
 - The users can review payment history.
 - The system must be always ready.
 - Automatically log out the account when the user is not in use for a certain time.
 - The system can work automatically.
 - The system can be operated and maintained periodically by the company's management and technical team.

c. Discuss the relationships between the FRs and NFRs.

An essential component of software engineering is the interaction between Functional Requirements (FRs) and Non-Functional Requirements (NFRs). While NFRs define the "quality attribute" of a system, such as performance, reliability, usability, and security, FRs are a requirement for the "functionality" of a software system.

NFRs are frequently implicit and subjective, making them more challenging to deduce and define than FRs. NFRs must be addressed in software development, nevertheless, as they can significantly affect the system's overall quality.

One method for merging FR and NFR is to use a goal-oriented, case-by-case approach. This method offers a framework for describing and fusing NFR and FR in a use case model at four connection points: subject (system boundary), actor, use case, and communication link. Based

on suggested NFR propagation principles, superfluous NFR specifications may be removed by implicitly linking NFR to other pertinent use case model pieces.

Another approach is to use a use case and goal-driven approach. Utilizing a goal-driven and use case-driven approach is an additional strategy. Through the use of use cases and a goal-oriented analysis, this method examines user needs. Cross-cutting quality criteria are suggested for requirements engineering.

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

Capturing customer requirements is the decisive factor in determining the direction and quality of the entire project. Here are some techniques needed for the project:

✖ *Interviews.*

To get background information on organizational needs, user and customer difficulties, support staff issues, and other important stakeholder concerns, interviews are helpful. Interviews are a further step that may be used to acquire more data.

The variety of stakeholders and system stakeholders, including consumer information, should be mentioned in interviews. To prevent system requirements from favoring one side, a fair assessment of competing demands must be conducted.

During the interview, it's crucial to ask open-ended questions. You shouldn't only ask "yes" or "no" questions. Give the interviewee the context of the request and ask them to explain their perspective and provide justifications.

✖ *User Observation.*

Observing users as they go about their regular routines is one of the finest methods to learn their demands. Passive and active observation are both types of observation. The best method for understanding the current process is active observation, which involves asking the user questions as you watch. It is more efficient to employ passive observation while gathering user input on design prototypes.

While monitoring clients, keep track of their actions. What was effective? What challenges? Reminder to visit loyal clients. Observing users in their natural environment can help us better understand their experiences and the changes that are required to address the issue.

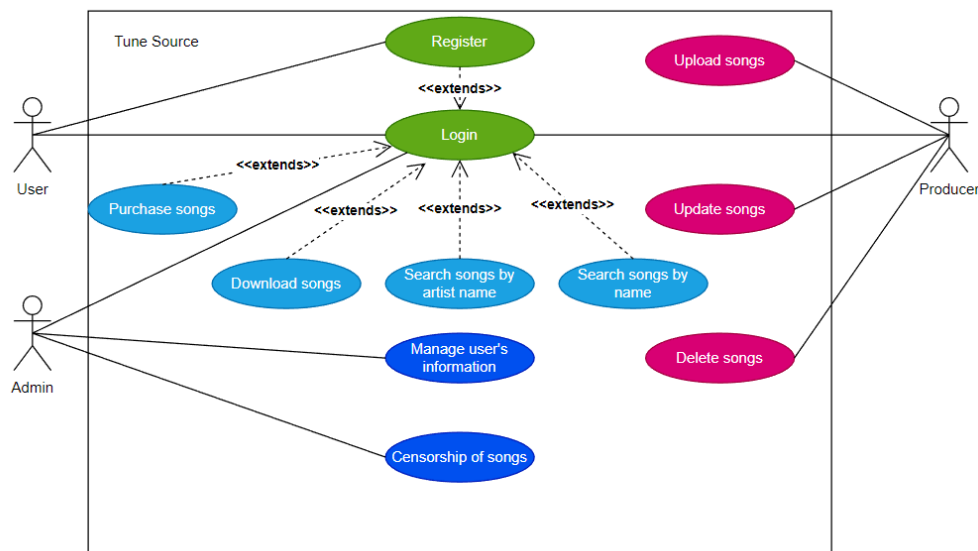
✖ *JAD (Joint Application Development).*

JAD is a distinctive method for developing software and systems because it engages the user and the customer at every stage of the design and development process. Both a consensus-based problem-solving methodology and a team-oriented development strategy are emphasized.

The JAD technique is helpful for a variety of initiatives, including the development of new goods and improvements to existing products. Projects involving user groups or users with cross-functional or cross-departmental responsibilities are the greatest candidates for JAD. Projects and initiatives that are for the first time seen as crucial to the organization's success are excellent JAD candidates. JAD may be used to build new goods or to improve existing products for companies that are addressing problematic histories or interactions between their products and end consumers.

II. Analysis (2).

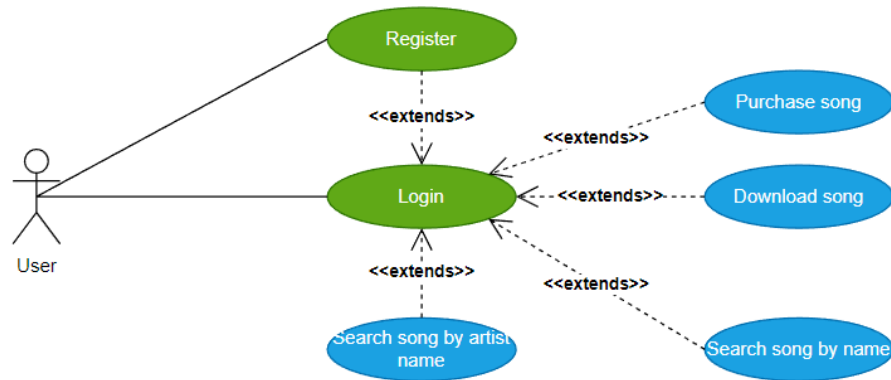
1. Use Case Diagram for the whole system.



The project's use case diagram focuses on three audiences: User, Admin, and Producer. User who want to log in to the system must register/log in to a previously registered account to be able to use the User's own functions such as: download songs, purchase songs, searching for songs by name or artist name. Admin and Producer have separate accounts to log into the system. Administrators can manage user's information and censorship of songs. Producers upload songs, update song information and delete songs.

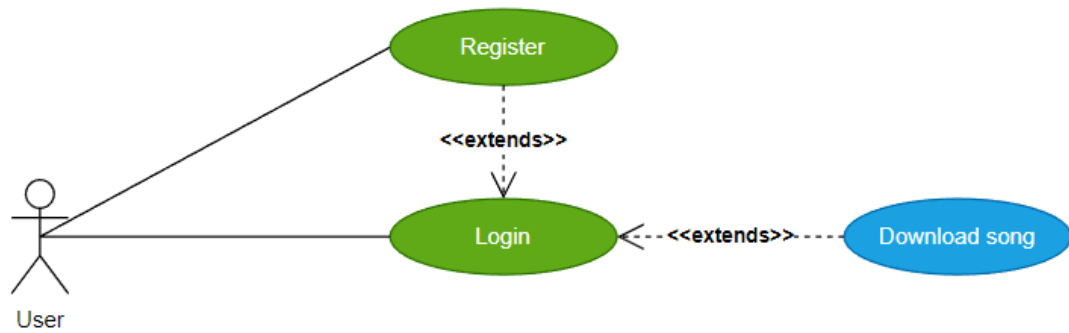
2. Use Case specification for 2 Use cases.

- **Login:**



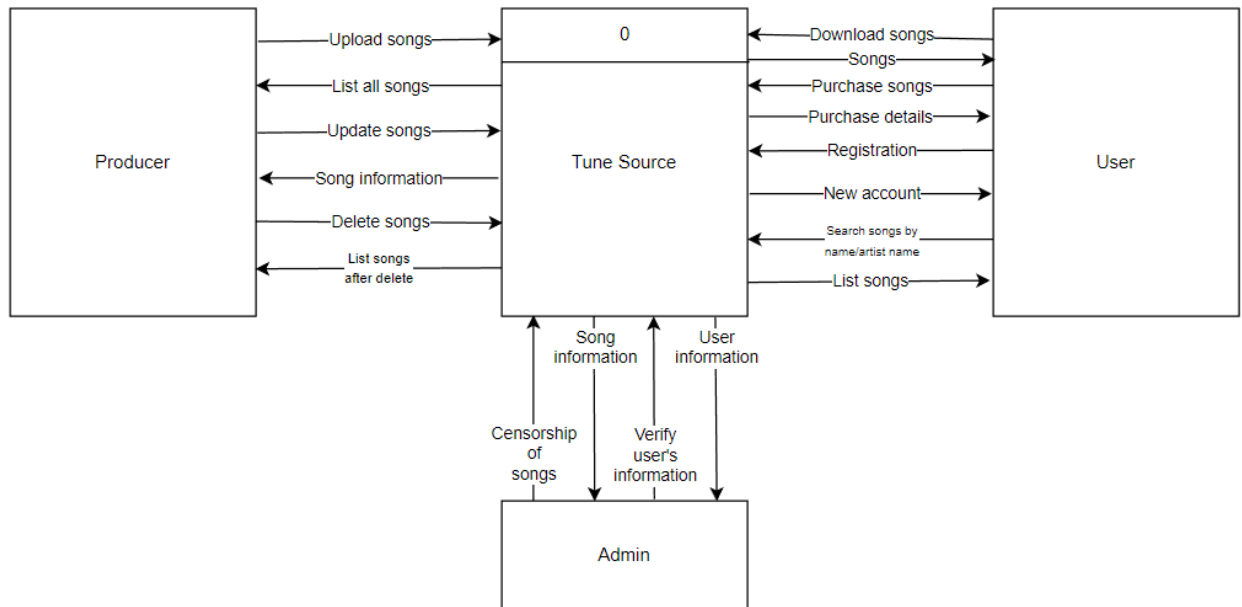
Use Case Name: Login		ID: D01	Priority: high
Actor: User			
Description: Allow users to log into the system if they already have an account and ask users to register for an account if they don't have one.			
Trigger: Login button			
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal			
Preconditions: <ul style="list-style-type: none">1. The user must enter username and password and press login button.2. The system is always online.3. Account must be registered first.			
Normal Course: <ul style="list-style-type: none">1. Check if the username and password exist in the system.2. Allow users to log into the system.		Information for Step: Check username and password Allow to login	
Alternative Course: <ul style="list-style-type: none">- Display the message "Incorrect username or password".- Display the message "Please enter an account or password".		Enter invalid username or password	
Postconditions: The system will allow users to successfully login to the system if the login information is correct. The system will be displayed error message if the user is logged in to the incorrect account.			
Exceptions:			
Summary Inputs	Source	Outputs	Destination
<ul style="list-style-type: none">- Username- Password- Login button	<ul style="list-style-type: none">- User- Database	<ul style="list-style-type: none">- Login successful	<ul style="list-style-type: none">- Login to the system

- **Download song:**



Use Case Name: Download song		ID: D02	Priority: high
Actor: User			
Description: When a user wants to download a certain song, they select the song they want and press the download button. The system will send the song to the user device after receiving the request.			
Trigger: Download button			
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal			
Preconditions: <div><div>1. The user must select the song and press download button.</div><div>2. The system is always online.</div><div>3. Song and singer must be in database.</div></div>			
Normal Course: <div><div>1. The system receives the song ID when the user selects the song, and then receives the download request when the user presses download.</div><div>2. Select the song with the selected id and send it to the user's device.</div></div>		Information for Step: <div>ID of the song</div> <div>Send the song to user’s device</div>	
Alternative Course:			
Postconditions: None			
Exceptions:			
Summary Inputs	Source	Outputs	Destination
<div><div>- Song ID</div><div>- Download button</div></div>	<div><div>- User</div><div>- Database</div></div>	<div><div>- Send the song to user’s device</div></div>	<div><div>- Download detail</div></div>

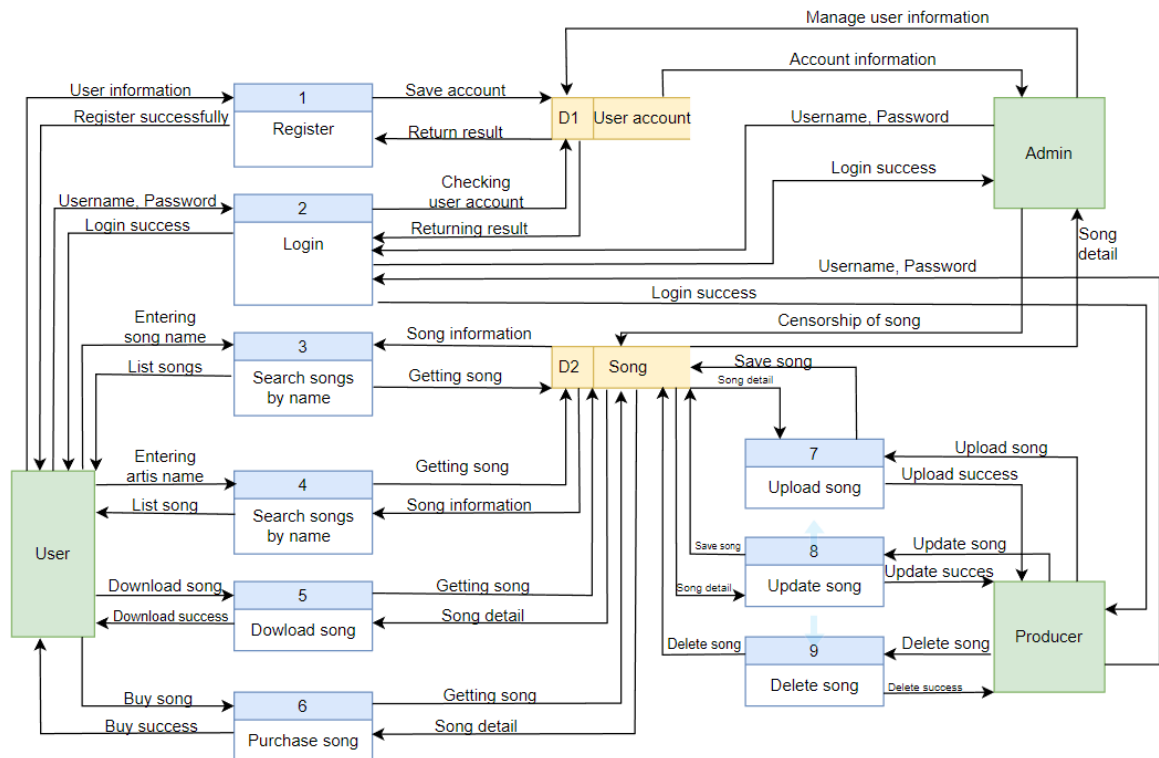
3. Context Diagram for the whole system.



There are three entities surrounding the Tune Source system: User, Admin, and Producer. System entry and exit arrows are data flowing back and forth between the system and objects.

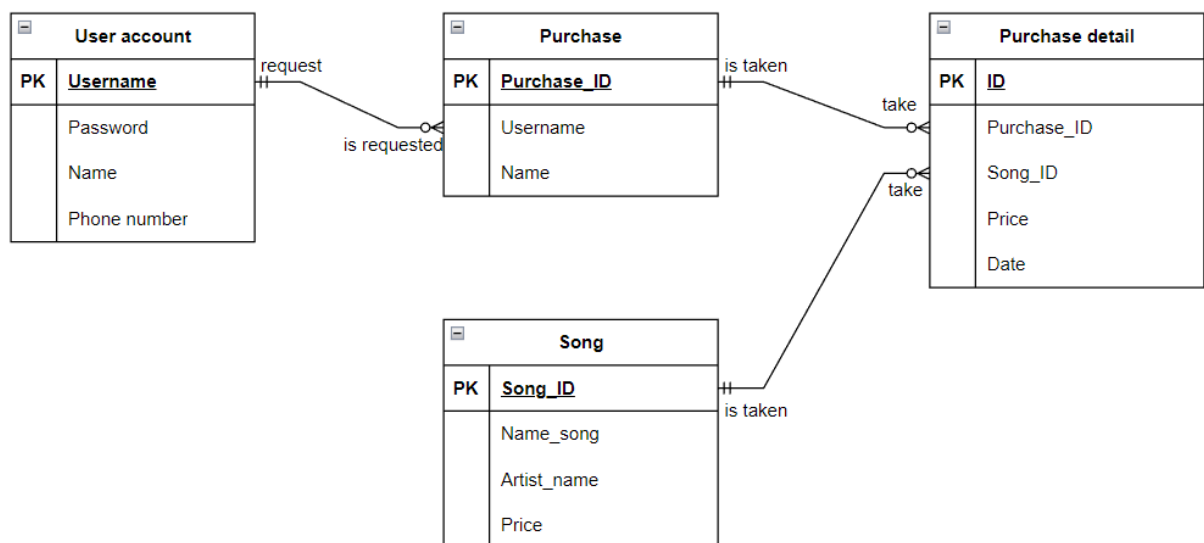
- User:
 - When the user downloads a song, the system returns the downloaded song.
 - When the user purchases a song, the system returns the purchased details.
 - When the user registers an account, the system returns the new account.
 - When the user search songs by name/artist name, the system returns list of related songs.
- Admin:
 - The system sends user's information when registering an account to the administrator, the administrator will verify the information and send a notification to the system.
 - The system sends song information to the admin, the admin takes steps to censor the songs and then sends a notification to the system.
- Producer:
 - The producer uploads new songs to the system, the system sends the list of songs to the producer.
 - The producer updates song information, the system sends the new songs detail to the producer.
 - The producer deletes a song, the system sends the new list of songs after deleted.

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



Data flow diagram illustrating the relationship between the external entities User, Admin, Producer, and the Tune Source system. We can observe the relationship between the system and external entities because Tune Source queries input data and then returns the queries. To utilize the download, purchase, and search songs by name or artist name, User must first create an account and log in to the system. To manage user's information and censorship of songs, the Admin must log in. To upload, update, and delete songs, Producer must be logged in.

5. ERD for the whole system.



An Entity Relationship Diagram (ERD) is a type of structural diagram used in database design. It includes a variety of symbols and connectors that represent two key pieces of data:

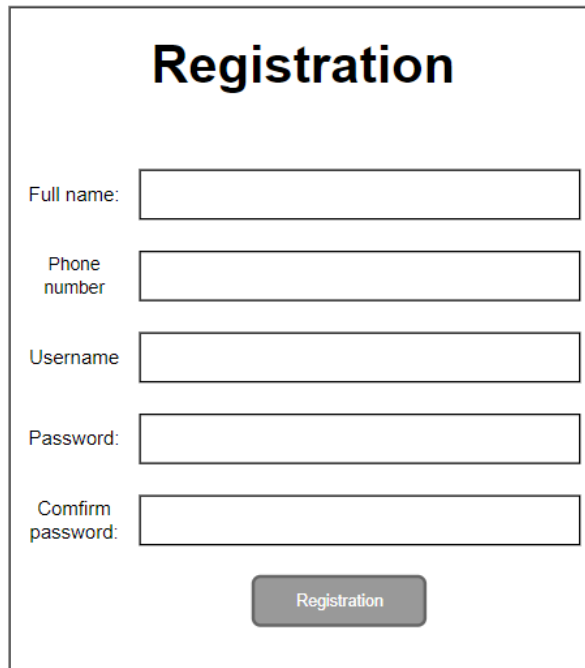
the key entities inside the system's scope and their interconnections. In the disciplines of software engineering, corporate information systems, education, and research, relational databases are most frequently designed or debugged using ERDs. Before making modifications to the database, ERDs are helpful for visualizing the database structure and spotting errors and design issues. By giving a complete view of the whole database structure, they can also aid in the debugging of database problems.

III. Design.

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

a. Wireframe.

- Register interface:



The wireframe shows a registration form with the title "Registration" at the top. Below the title are five input fields, each with a label to its left: "Full name:", "Phone number", "Username", "Password:", and "Confirm password:". At the bottom of the form is a button labeled "Registration".

This is the interface for users to register for an account to log in to the system. Here, the customer fills in the necessary information such as Name, phone number, username, password, etc. Then click register to complete the procedure.

- Login interface:

Login

Username:

Password:

Login


Cancel

The login interface requires the user to enter a username and password to log in to the system.

- Home page interface:

Logo

Tune Source

Searching.....

Jazz

Rock

Folk

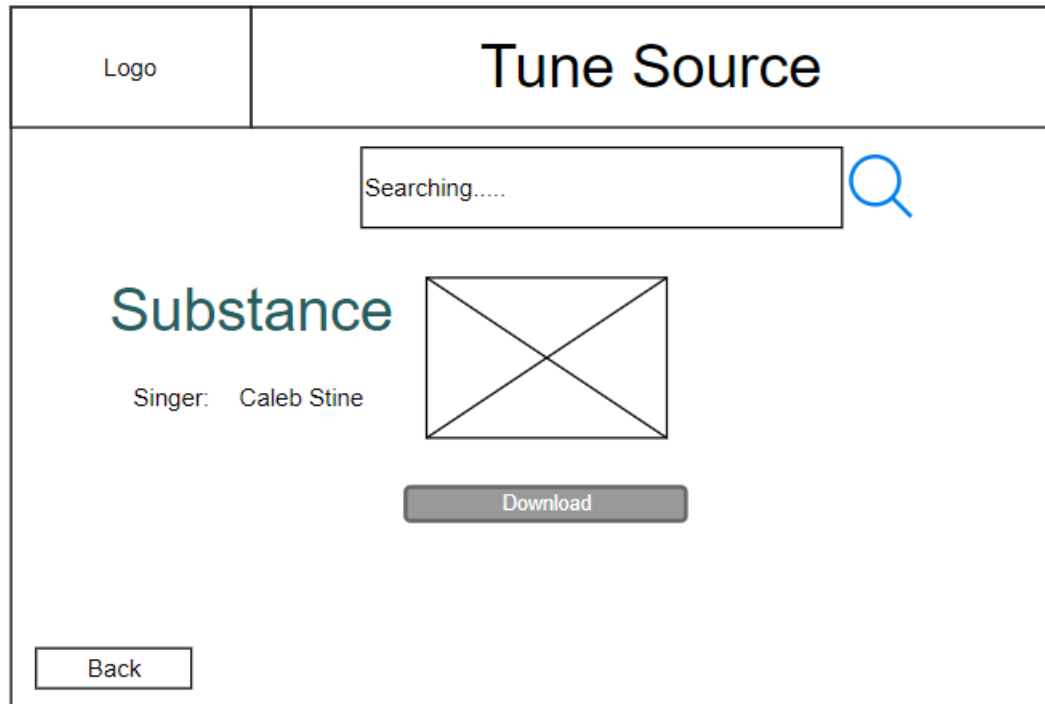
Country

Login

Register

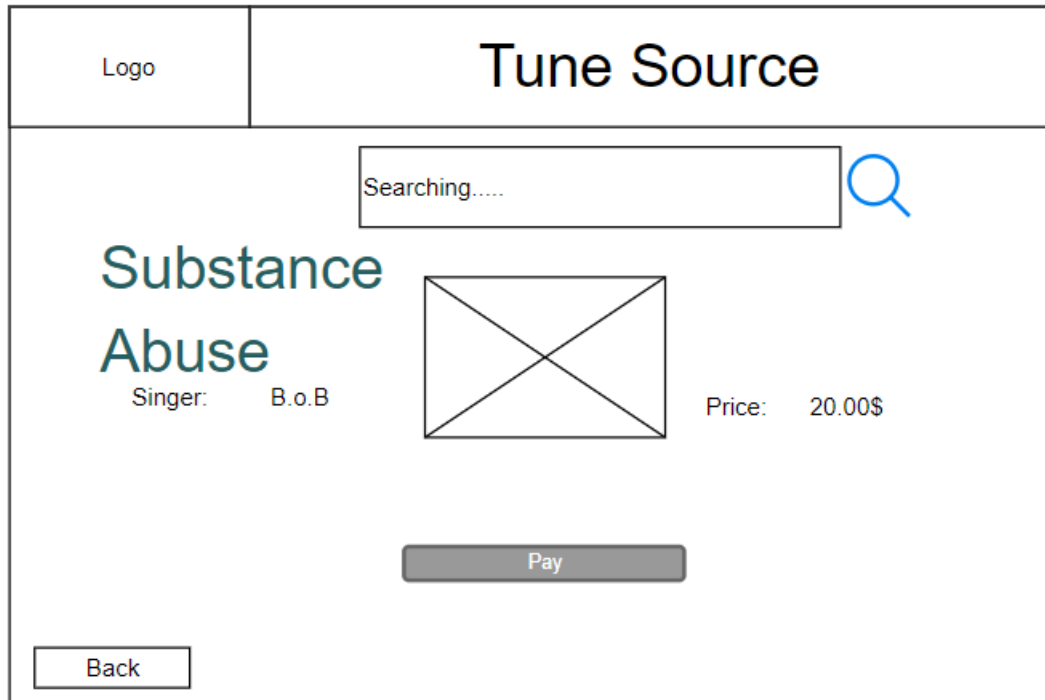
This is the homepage interface, the left side of the homepage includes the logo, music classification. Above is the Company Name, in the middle is the search bar and the list of new works.

- Download interface:



Here users can click download to download the song to the device.

- Purchase interface:



Here users can pay to buy songs.

b. Explain architecture in project.

In this project, I decided to use Client-server architecture for one of the following reasons:

- **Centralization:** The centralized control it relates to is the client server network's key benefit. One area has all the information that is required. Since they have complete control over management and administration, this is especially advantageous for the network administrator. Any issue that arises throughout the whole network may be resolved in a single location. Additionally, because of this, upgrading resources and data has become much simpler.
- **Security:** Its centralized architecture ensures that data is secure. Access restrictions can make sure that only authorized users are allowed access. Using credentials like a username and password as a requirement is one such technique. Additionally, if data loss occurs, files can be quickly restored from a single backup.
- **Scalability:** The user can expand the amount of resources, such as clients and servers, as needed. growing the server's size without too many disruptions. Since the server is centralized, even if the size grows, there is no uncertainty over authorization to network resources. As a result, the arrangements require a relatively small number of staff members.
- **Management:** File management is rather simple because every file is kept on the same server. The finest management for tracking and finding records of necessary files is available in client-server networks.
- **Accessibility:** Every client has the chance to log into the system, regardless of location or platform. This will allow all employees to access their business data without using a terminal mode or processor.

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

In this project, I used the Symfony framework, which is quite famous nowadays. In many projects, programmers often use this framework because it is very easy to use. In addition, this framework also integrates a lot of libraries to make coding easier. This framework is great for projects like Tune Source. The data of this project will be saved to MySQL because it is easy to use, reliable and secure.

2. Discuss how activity diagram and pseudocode are used to specify the software behavior (P7).

a. Activity diagram.

Similar to flowcharts, an activity diagram graphically depicts the progression of events or operations inside a system. It is frequently used in business process modeling to present a clear portrayal of activity flows for system analysis and design. It represents concurrent actions, choices, and circumstances.

Activity diagrams may be used to represent how a system acts when actions are taken to finish a task or process. Even though they are often arranged in a sequential sequence, concurrent or simultaneous activities can also be shown in activity diagrams.

b. Pseudocode.

In order to explain a computer program's behavior at a high level, pseudocode combines normal language with programming language components. It is used to define the behavior of the program in terms that are simple enough for humans to comprehend while still being detailed enough to be converted into actual code.

One such method entails specifying events or noteworthy occurrences in a software system, followed by whatever restrictions the designer might want to put on the simultaneity or ordering of those events. Using pseudocode to specify input-output relationships between predicate calculus expressions, which can act as specifications for a program's intended behavior, is an additional method.

Pseudocode offers a way for defining the expected behavior of a suggested software system design long before it is ever implemented, making it a useful tool for building complicated, concurrent software systems. It enables designers to define the behavior of the system under development as a guideline for the final program code, which is particularly important and challenging when that behavior incorporates theoretically or practically concurrent operation.

IV. References.

FLM, n.d. [Online]

Available at: <https://flm.greenwich.edu.vn/gui/role/student/SyllabusDetails?syllID=2538>