

## 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			
Unit Tutor			
Issue date		Submission date	01/07/2022
Name and date	Tran Dang Khoa / GCS190407		

Submission Format:	
Format:	<p>The submission is in the form of 1 document.</p> <p>You <b>must</b> use the <i>Times font</i> with <i>12pt size</i>, turn on <i>page numbering</i>; set <i>line spacing to 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><b>Word limit:</b> 3000 words (excluding figures and references). Submissions that exceed this limit will be rejected.</p>
Submission:	<p>You <b>must</b> submit the assignment <b>by the due date</b> and follow the submission method specified by the Tutor. The submission form is <b>soft copy</b>, which is to be uploaded to the following URL: <a href="http://cms.greenwich.edu.vn">http://cms.greenwich.edu.vn</a>.</p>
Note:	<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. <b>Plagiarism is a very serious offence</b>, which will result in a failing grade.</p>
Unit Learning Outcomes:	
<p><b>LO3</b> Undertake a software development lifecycle.</p> <p><b>LO4</b> Discuss the suitability of software behavioural design techniques.</p>	
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.

Discuss the relationships between the FRs and NFRs.

*[MV]*

- *Intro purpose of this section*
- *Introduction of requirement definition, FR, NFR and more details*
- *Present the discussion about relationships between the FRs and NFRs*
- *List out stakeholders who can provide requirements, their roles and their interests in Tune Source project*
- *Present the Requirement Definition (with explaining its need/ purpose) of TS project. Clearly indicate which stakeholder(s) provide what requirements. Propose some NFRs for TS project and explain why.*

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.

*[MV]*

- *Intro purpose of this section*
- *Describe 5 requirement gathering techniques: Interviews, Join Application Development, Questionnaires, Document Analysis, Observation (description, steps, pros & cons, ..)*
- *Apply to TS: select ones that suitable for Tune Source and explain why*
- *Add at least 2 Interview notes for TS*

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

### 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 the activity and class diagrams for the use case of one top-level function. The activity diagram needs only cover 2 sub-functions.

4. You can choose to use either

*object oriented* analysis and design techniques (use case diagram, class diagram, activity diagram)  
OR

*structured* analysis and design techniques (DFD, ERD, flowchart).

5. Do not mix these two sets of techniques. For example, do NOT use ERD with activity diagram!

*[MV]*

- *Intro purpose of this section*
- *Introduce about Requirements Modellings*
- *Apply to TS: choose 1 of below options:*
  - *At least 1 ERD, 2 DFD, 2 flowchart, 1 pseudocode*
  - *At least 1 class diagram, 2 activity diagrams, 2 sequence diagrams, 1 pseudocode*

### **Task 3 – Design**

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

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

*[MV]*

- *Intro purpose of this section*
  - *Intro details of use case diagram*
  - *Apply to TS:*
    - *1 major use case diagram*
    - *At least 4 elaborating UC digrams*
7. (M5) Discuss how activity diagram and pseudocode are used to specify the software behaviour.
8. (M6) Discuss how UML state machine can be used to specify the software behaviour. Differentiate between FSM And extended FSM using the case study.
9. (D4) Discuss how the data-driven approach improves the reliability and effectiveness of software.

### **Task 4 – Software quality management**

10. (M4.a) Discuss two software quality attributes that are applicable to the project.

11. (M4.b) Discuss two quality assurance techniques that can help improve the software quality in the project.
12. (D3) Discuss how the design techniques and approaches that you have used can help improve the software quality.

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

## Contents

<b>P5. Undertake a software investigation to meet a business need.</b>	7
1. Introduce.	7
2. Introduce functional requirement and non- functional requirement.	7
2.1. Functional requirements.	7
2.2 Non- functional requirement.	9
3. The relationships between the FRs and NFRs	10
4. List out stakeholders in Tune Source project.	10
5. Present the Requirement Definition of TS project.	11
<b>P5.b Discuss the technique(s) you would use to obtain the requirements.</b>	12
1. Intro purpose of this section.	12
2. Five requirement gathering techniques.	12
3. Apply to TS.	16
4. Interview notes for TS.	16
<b>P6 Analyse the requirements that you identified in Task 1 using a combination of structural and behavioural modelling techniques that you have learnt.</b>	17
1. Intro purpose of this section.	17
2. Introduce about Requirements Modellings.	17
3. Apply to TS.	18
<b>P7. Discuss how the user and software requirements are addressed in the design phase.</b>	22
1. Intro purpose of this section.	22
2. Intro details of use case diagram.	22
3. Apply to TS.	24
<b>REFERENCES.</b>	27

**P5. Undertake a software investigation to meet a business need.**

**1. Introduce.**

As my group did in Assignment 1, we continued to develop software for tuning under the following assignment. This report will explain the necessary steps, procedures, and most importantly, techniques, so that we may better ourselves by following the instructions.

**2. Introduce functional requirement and non- functional requirement.**

**2.1. Functional requirements.**

**Define:**

The service that the program must offer is described in a functional requirement (FR). It provides information on a software system or its parts. A function is nothing more than an input that affects the behavior and outputs of a software system. Any specific function that specifies what the system is capable of performing might be computing, data manipulation, business process, user interaction, or another specialized function. In software engineering, functional specifications are another name for functional requirements.

### **Types of functional requirement:**

- Transaction Handling
- Business Rules
- Certification Requirements
- Reporting Requirements
- Administrative functions
- Authorization levels
- Audit Tracking
- External Interfaces
- Historical Data management
- Legal and Regulatory Requirements

### **Functional Requirements Best Practices:**

Do not combine two requests into one. Keep detailed requirements.

- You should make each request as complete and accurate as possible.
- The document should draft all technical requirements.
- Map all requirements with goals and principles that contribute to successful software delivery
- Make inquiries using interviews, seminars, and casual communication.
- If there are any known, verified constraints that seriously affect a requirement, it is a significant status that should be documented.
- You need to document all assumptions in the document.

### **Error while creating functional requirements:**

- Giving unwarranted additional information can confuse developers
- Full details are not included in the request document.



- You add rules or examples, scope statements or goals anything but the request itself.
- Omit an important part of information that is absolutely necessary to fully, accurately and completely state the request.
- Some professionals began to defend the claims they recorded when the claim was modified, instead of finding out the exact facts.
- The requirements are not related to goals or principles.

## **2.2 Non- functional requirement.**

### **Define**

Non-functional requirements (NFRs) specify the quality attribute of a software system. They evaluate the software system against Responsiveness, Usability, Security, Portability, and other non-functional standards that are critical to the success of the software system.

### **Types of non-functional requirement**

- Usability requirements
- Serviceability requirements
- Management requirements
- Resilience Requirements
- Security requirements
- Data integrity requirements
- Capability Requirements
- Availability requirements
- Scalability Requirements
- Interoperability requirements
- Reliability requirements
- Maintainability requirements
- Mandatory provisions
- Environmental requirements

### **Advantages of non-functional requirements**

- Non-functional requirements ensure that the software system follows legal and compliance rules.

- They ensure the reliability, availability and performance of the software system
- They ensure a good user experience and easy-to-operate software.
- They help build the security policy of the software system.

### **Disadvantages of Non-functional requirement.**

- None functional requirement may affect the various high-level software subsystem
- They require special consideration during the software architecture/high-level design phase which increases costs.
- Their implementation does not usually map to the specific software sub-system,
- It is tough to modify non-functional once you pass the architecture phase.

### **3. The relationships between the FRs and NFRs**

	<b>Functional Requierments</b>	<b>Non-Functional Requierments</b>
<b>Key Question</b>	What ought to the system do?	How ought to the framework perform the assignments?
<b>Result</b>	Item highlights	Item properties.
<b>Focus</b>	Client necessities	Client desire and experiences
<b>Type of Testing</b>	Useful Testing like System Integration, Conclusion to End, API testing, etc	Non-Functional Testing like Performance, Push, Usability, Security testing, etc.
<b>Text Execution</b>	Test Execution is done before non-functional testing	After the utilitarian testing.
<b>Product Info</b>	Item highlights	Item properties

### **4. List out stakeholders in Tune Source project.**

- Stakeholders in the project include 4 main components: project sponsors, executives, IT department and customers.

	<b>Role</b>	<b>Benefits</b>
<b>Project sponsors</b>	they will describe the project and its duration, determine the requirements and risks of the project, and they will follow the project closely to evaluate and report on the	provide goals and benefits in the project and see who benefits in the project, view the project and offer good solutions.

	project and help complete the project The sentence is more correct than requested.	
<b>Executives</b>	consists of people who have a broad vision, have the ability to run about the management and the organization, set current goals and can predict what will happen in the future.	they can export their products to the market and earn extra income from it
<b>IT department</b>	carry out projects that are launched, solved, processed, and repaired	benefit from the projects done.
<b>Customers</b>	is a person who uses products from businesses, requests to make and buy products from businesses, and experiences and gives feedback on products from businesses.	experience products according to their ideas.

## 5. Present the Requirement Definition of TS project.

### The purpose

The goal of this project is to provide users with the essential features they need, such as the ability to search for their favorite music, listen to it online, download it for offline listening at no additional cost, register an account to view your activity history, and finally purchase a gift card to download music.

### Stakeholders in the project

#### **IT Division.**

- All of the employees in this department of ours have a great deal of project experience since they are the ones who will correct any mistakes in the project right away.

### Managing director.

- They have the management skills, the market judgment skills, and the capacity to determine what our system's next move should be in this field.

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

#### 1. Intro purpose of this section.

- The purpose I present this section is to outline the techniques I will use to accomplish my goal

#### 2. Five requirement gathering techniques.

##### Interview:

- Every stage of the hiring process—from the job description to the interview to the job offer—can influence whether you make the proper choice. The interview is particularly crucial in determining if a candidate is a good match for your business and whether you are a good fit for them. However, given all the choices, there is a wide range in the benefits and drawbacks of interviews.

Pros	Cons
The most common and successful factors are these interviews since they can be scheduled quickly. This is also the manner that tours are most adaptable because each person's issues might cause them to be formal or casual, ask certain questions, or utilize their own particular style.	It is simple for applicants to get ready for interviews. Additionally, the interviewer must have sufficient expertise to know which questions to avoid asking and how to frame them. Comparing candidates on these problems is equally challenging. The questions and methods may not be consistent from one interview to the next if there are multiple instances of a particular problem. If this happens, the interview process may become tedious.

##### Step interview.

1. Establish position specifications and create detailed job descriptions.
3. Pre-employment tests should be used to evaluate all candidates.
2. Prepare questions tailored to the position in advance.
4. Before doing face-to-face interviews, conduct one-way phone or video screen interviews.
5. Give the interview adequate time.
6. Communication with applicants beforehand
7. Before the interview, research the applicant.

- Join Application Development.

In the sphere of business analysis, Application General Development (JAD) is a widely used approach. It uses a systematic approach based on discussion groups to bring together system developers and users in an effective and innovative setting with the aim of obtaining needs and specifications.

**Pros:**

JAD allows you to solve problems faster and develop better, bug-free software.

The combined involvement of the company and the customer reduces all hazards.

JAD saves project development costs and time.

System quality is improved by well-defined criteria.

Progress is being made at a faster rate due to close communication.

JAD motivates the team to encourage each other to work faster and deliver on time.

**Cons:**

Different perspectives within the team make it difficult to align goals and stay focused.

JAD can require a large time commitment depending on the size of the project.

**Step:**

**Step 1:** Define project goals and constraints.

**Step 2:** Identify critical success factors.

**Step 3:** Identify the project deliverables.

**Step 4:** determine the schedule of workshop activities.

**Step 5:** choose participants.

**Step 6:** Prepare in advance for the workshop.

**Step 7:** Organize activities and exercises of the workshop.

**Step 8:** Prepare, Inform and Educate Participants.

**Step 9:** Coordinate the logistics of the workshop.

**Questionnaires:**

A questionnaire is a research tool consisting of a series of questions used to collect useful information from respondents. These tools include written or oral questions and include an interview format. The questionnaire can be qualitative or quantitative and can be

administered online, by phone, on paper or in person, and the questionnaires do not need to be administered with a researcher present.

### Pros:

**Practicality:** While gathering substantial volumes of data on any subject, questionnaires enable researchers to proactively choose their target audience, questions, and format.

**Cost-effective:** You may send survey questions to respondents via email or by posting them on your website for a charge instead of using surveyors. free or inexpensive.

**Speed:** Using mobile tools, you may swiftly and simply collect survey answers, getting feedback and insights in as little as 24 hours.

**Comparability:** To get important insights and minimize translation problems, researchers might utilize the same questionnaire year after year and compare and evaluate study findings.

**Scalability:** Since questionnaires are very scalable, researchers may send them to every demography in the world.

**Normalize:** You may include as many questions as you'd like on any topic in your panel's standard question.

**Respondent comfort:** Respondents are fully anonymous while answering a panel question, and there are no time constraints, which makes them more at ease and motivates them to speak up. honest.

**Easy analysis:** Table questions often have built-in tools to automate analysis, making it quick and easy for you to work out your results.

### Cons:

**Dishonest Answer:** Respondents might not always be totally truthful with their responses; some could provide program facts while others would give an alternative response. which is the most socially acceptable.

**Skip question:** Make sure to ask for answers for all you refer to the question. Otherwise, you run the risk of the respondent answering the unanswered question.

**Difficulty interpreting:** Respondents may struggle to answer a question accurately if it is too complex. Because of this, it's crucial to ask a question that is succinct and clear and includes any essential justifications.

**Lack of relevance:** Respondents who don't read your question thoroughly or attentively risk providing inaccurate responses that might skew the results. By making your queries as brief and straightforward as possible, you may lower this danger.

### **Types of questionnaires in research:**

**Postal:** Participants in postal questionnaires receive printed surveys in the mail. Once respondents have finished the survey, they will submit it back to the sending company.

**Internal:** In this type of questionnaire, researchers visit respondents at their home or workplace and conduct the survey in person.

**Phone:** When conducting a telephone survey, researchers call participants and ask them questions.

**Electronic:** Probably the most common type of questionnaire, electronic surveys are presented via email or through another online medium.

### **Document Analysis:**

- Document analysis is a helpful method for analyzing data, frequently used in conjunction with parsing projects and interviews.
- Theory-based examinations can also make use of this method. You may better grasp pre-warnings and problem triggers by examining significant papers and reports pertinent to your project and the challenges it is tackling. By checking to see if your formats are covered in the documentation, you may also assess your broadcasts and evaluate them.

### **Pros:**

- 1 Focusing on interview questions and knowing what to watch for while monitoring participants are made easier using document analysis.
- 2 It is particularly helpful when you want to narrow in and concentrate on a particular case study, whether it be a particular patient, student, community, subdivision, or workshop.
- 3 It is quite quick and affordable if data is accessible (e.g. over the internet).
- 4 The documents offer precise and consistent data that are unaffected by the researchers' presence (in contrast to participant observations, project participants may behave differently when present). they are conscious of seeing them).

### **Cons:**

1. You will need to adopt a critical perspective and not assume that the information included in them is factual or unbiased because the materials may be incomplete or written objectively.
2. There might be too much information due to the volume of papers involved. Which section of the document most closely matches your query? Is the focus primarily on the development of your project and the variables that aided or impeded it?

3. It might be necessary to anonymize some documents before comparing them to others. Do all the documents you used say the same thing, or does the one you used say something substantially different? What causes this, if so? Is the literature different in terms of its aim, design, or source?

**Observation:**

- A technique for gathering data that involves observing individuals, events, or capturing physical characteristics in their natural habitat. The subject of the observation may be aware that they are being monitored in public or secret (not aware they are being watched).

**Participants' observations:**

- The researcher immerses themselves in the setting or culture being studied.
- For a study to be effective, the researcher has to be accepted as a member of the culture being studied.

**Direct observation:**

- In order to avoid influencing the observations, the researcher strives to maintain a more detached demeanor.
- Technology has its uses (i.e. video, audio recording).

**Indirect observation:**

- The outcome of an encounter, procedure, or observable behavior (for example, counting the number of plates left behind by students in the cafeteria of a school to see if a.

**3. Apply to TS.**

I choose the interview as my application method for the TS because it involves several phases, from preparation to conclusion, and offers both direct and indirect interviewing options, however I will prioritize direct interviewing. directly because in-person interviews allow us to observe the interviewee's looks, demeanor, communication style, and situational handling to determine whether or not they are a good fit.

**4. Interview notes for TS.**

Interview Note
<p><b>Person Interview:</b> J Homen</p> <p><b>Interview:</b> Tran Dang Khoa</p> <p><b>Purpose of Interview:</b> - Learn the problem in TS - How to make the system more and more developed?</p> <p><b>Summary of Interview:</b> J.Homen's problem summary</p>



<ul style="list-style-type: none"> <li>• State the two biggest problems of the system</li> <li>- Number of people working together in the system</li> <li>- System capacity</li> </ul> <p><b>Open Items:</b> - Get processing of the number of visitors</p> <p>- Continue to meet the next interviewer</p> <p><b>Detailed Notes:</b> See attached transcript</p>
--

<p><b>Interview Note</b></p> <p><b>Person Interview:</b> L Wonem</p> <p><b>Interview:</b> Tran Khoa</p> <p><b>Purpose of Interview:</b> How to make the system more and more developed ?</p> <p><b>Summary of Inter view:</b> L Wonem's problem summary</p> <ul style="list-style-type: none"> <li>• State the two biggest problems of the system</li> <li>- Manage the level of the VIP package</li> <li>- Set many favors for high VIP package users</li> </ul> <p><b>Open Items:</b> - VIP system management</p> <p><b>Detailed Notes:</b> See attached transcript</p>
---

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

**1. Intro purpose of this section.**

- In this part, I will introduce the analysis and requirements that I have stated earlier, and I will combine the models and structures that I have mentioned before.

**2. Introduce about Requirements Modellings.**

• **Define:**

In software development projects, requirements and solutions are continually altered via coordinated efforts and cooperation. This process is known as requirements modeling. You may make sure that your team fully satisfies the requirements of your stakeholders by employing this self-organizing and cross-functional team method.

• **Purpose:**

support the objective of developing software. The following objectives are also included:

- There are always possibilities to enhance the overall modeling strategy.
- Identify and define the best practices required to produce an effective model.
- Outline the ways you propose to put said techniques into action.

- **Objectives:**

**Scenario-based modeling:** A script must go through a series of phases in order to depict an imagined system interacting with it. As part of the requirements analysis process, scenarios are used to define certain features of the suggested solution. They preserve the system's integrity by photographing it from the outside. The user is allowed to utilize certain samples, or specific sorts of code.

**Data modeling:** Data modeling is the technique of utilizing words and symbols to describe the data and how it flows to create a streamlined picture of a software system and the data pieces it includes. For creating a new database or reengineering an old application, data models offer a blueprint.

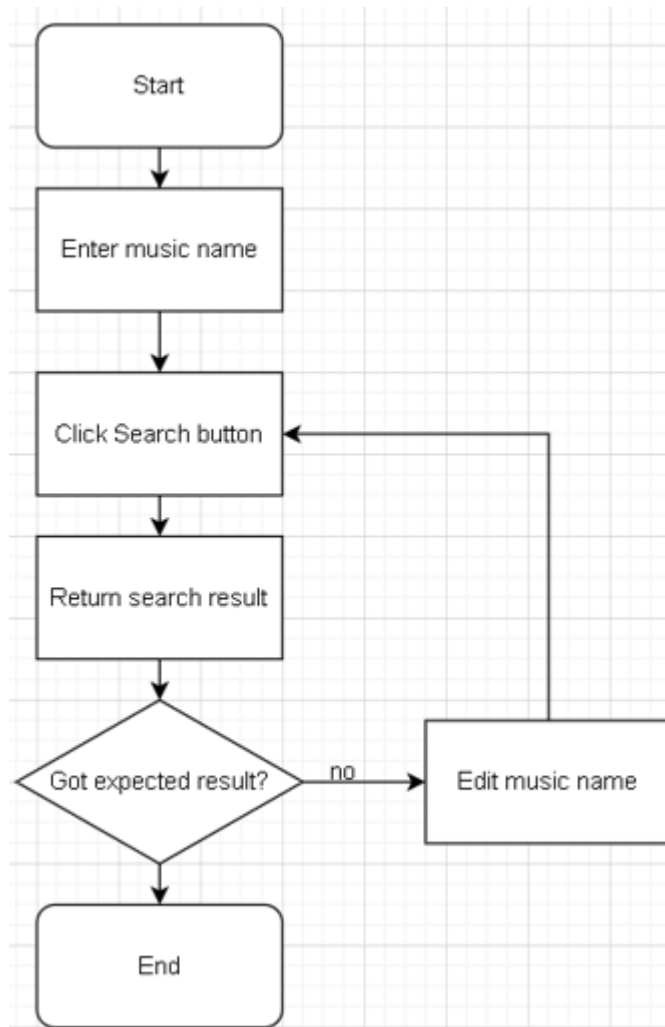
**Flow-oriented modeling:** This type of modeling illustrates how data items change as they pass through a system. Data flow diagrams, a model notation that illustrates how inputs are turned into outputs as data items travel through the system, are used in flow modeling, which is derived from structured analysis. A process definition or narrative is used to explain each data transformation software function. This model element also describes control flow in addition to data flow.

**Class-based modeling:** A stage of requirements modeling is class-based modeling. Examining the specifications that a proposed software application or system must fulfill in order to be successful is a necessary part of modeling in the context of software engineering. The first step in requirements modeling is often scenario-based modeling, which involves creating a use case that will be useful for the next phases, such as data and class-based modeling. The classes, attributes, and actions that the application will utilize are extracted from a use case via the class-based model. Class-based modeling often ends with a diagram or set of diagrams, most frequently produced using UML, or rather, Language, as is the case with other modeling stages. common modeling language

**Behavioral modeling:** This strategy is utilized by businesses to comprehend and forecast consumer behavior. Consumer and company spending data are now available, and behavioral modeling is used to predict future behavior under particular conditions. Financial organizations use behavioral modeling to determine the risk of lending money to a person or business, while marketing firms use it to focus their advertising.

### 3. Apply to TS.

**flowchart of Listen to music:**



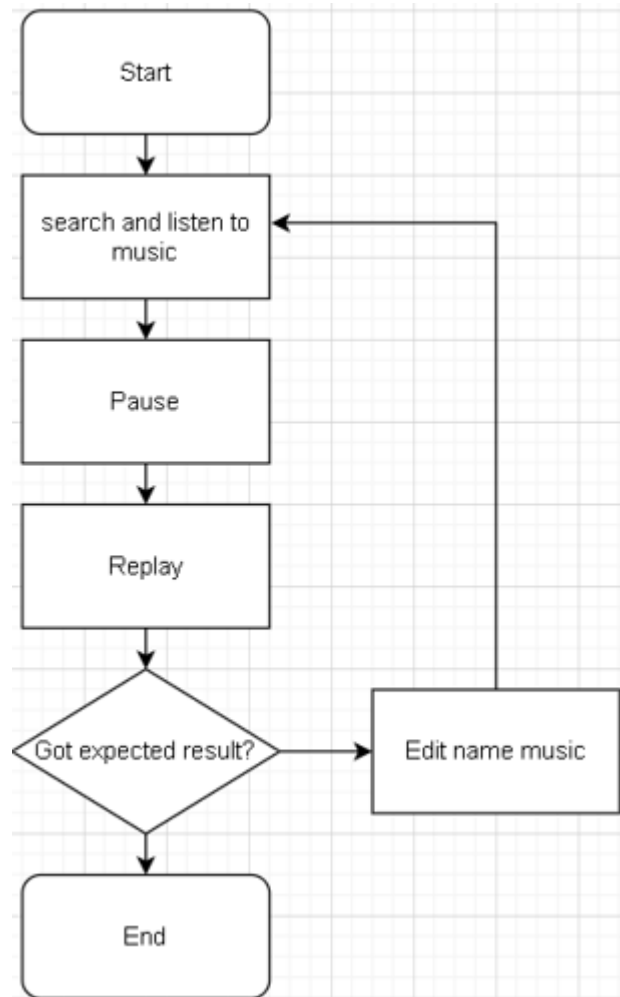
**Step 1:** Type the song's name into the search box.

**Step 2:** is to select the "music search" button.

**Step 3:** Show the table of search results.

**Step 4:** Finish if the results table is accurate; otherwise, proceed to the step where you update the song's title and click the search button.

**flowchart of Pause and Replay music :**



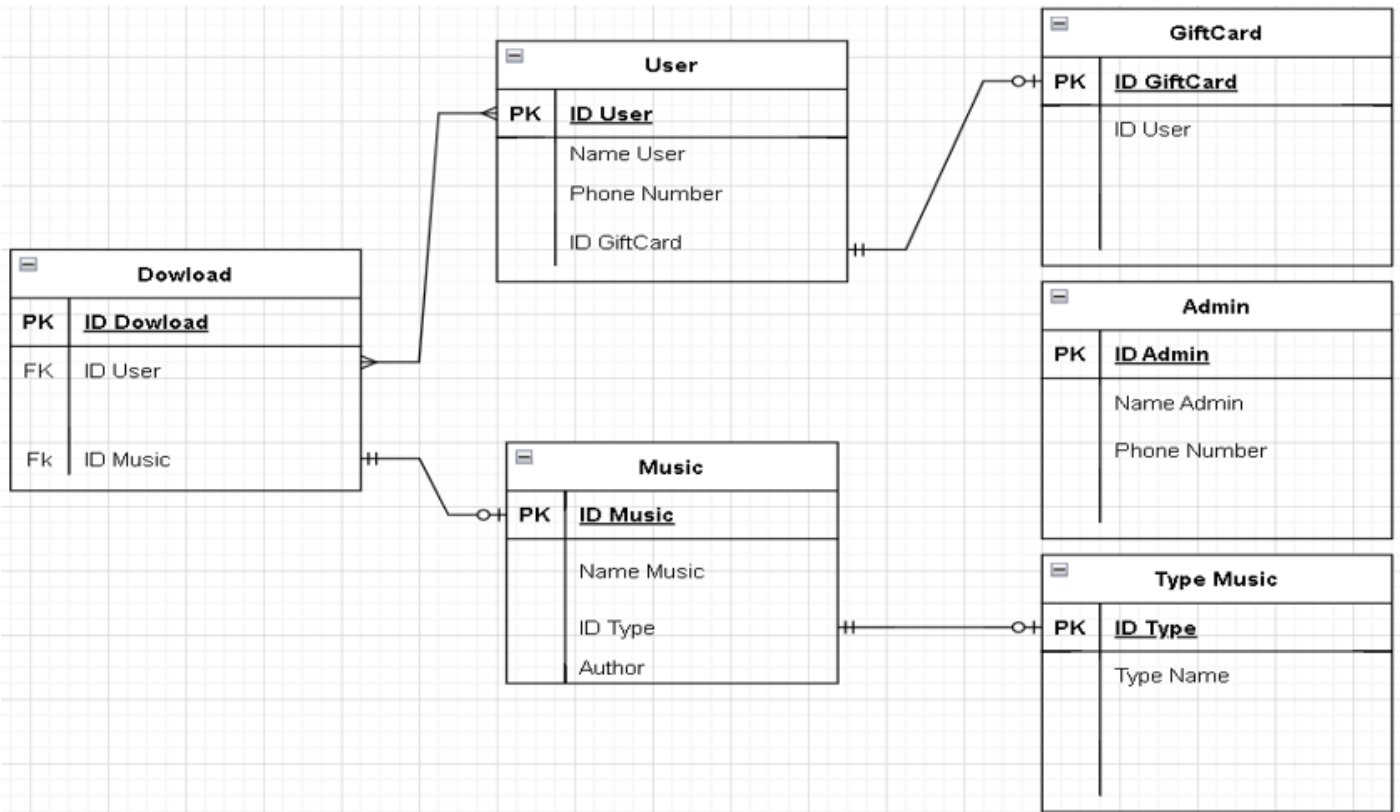
**Step 1:** is to look up and play a song.

**Step 2:** click stop.

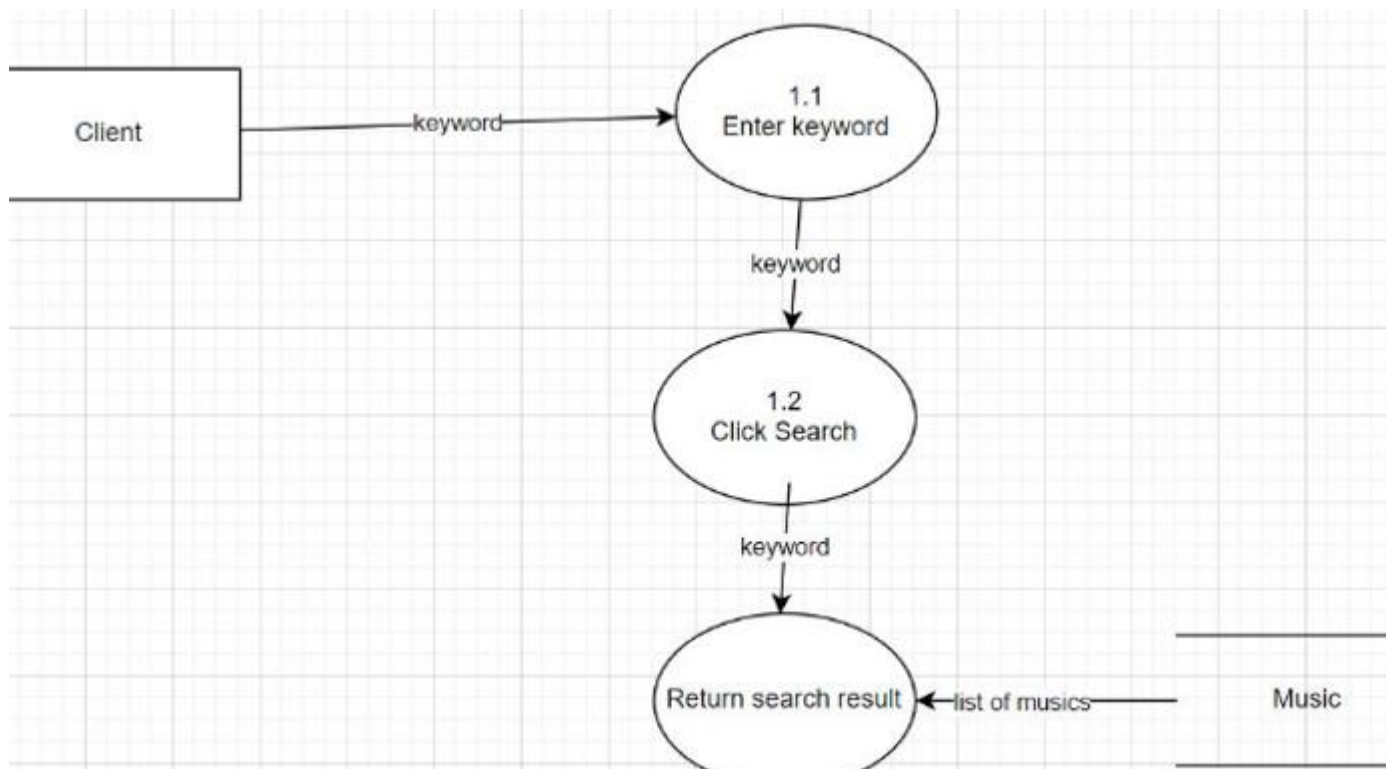
**Step 3:** choose "play music again".

**Step 4:** If this music is appropriate, continue; if not, alter the song title and go back to step 1.

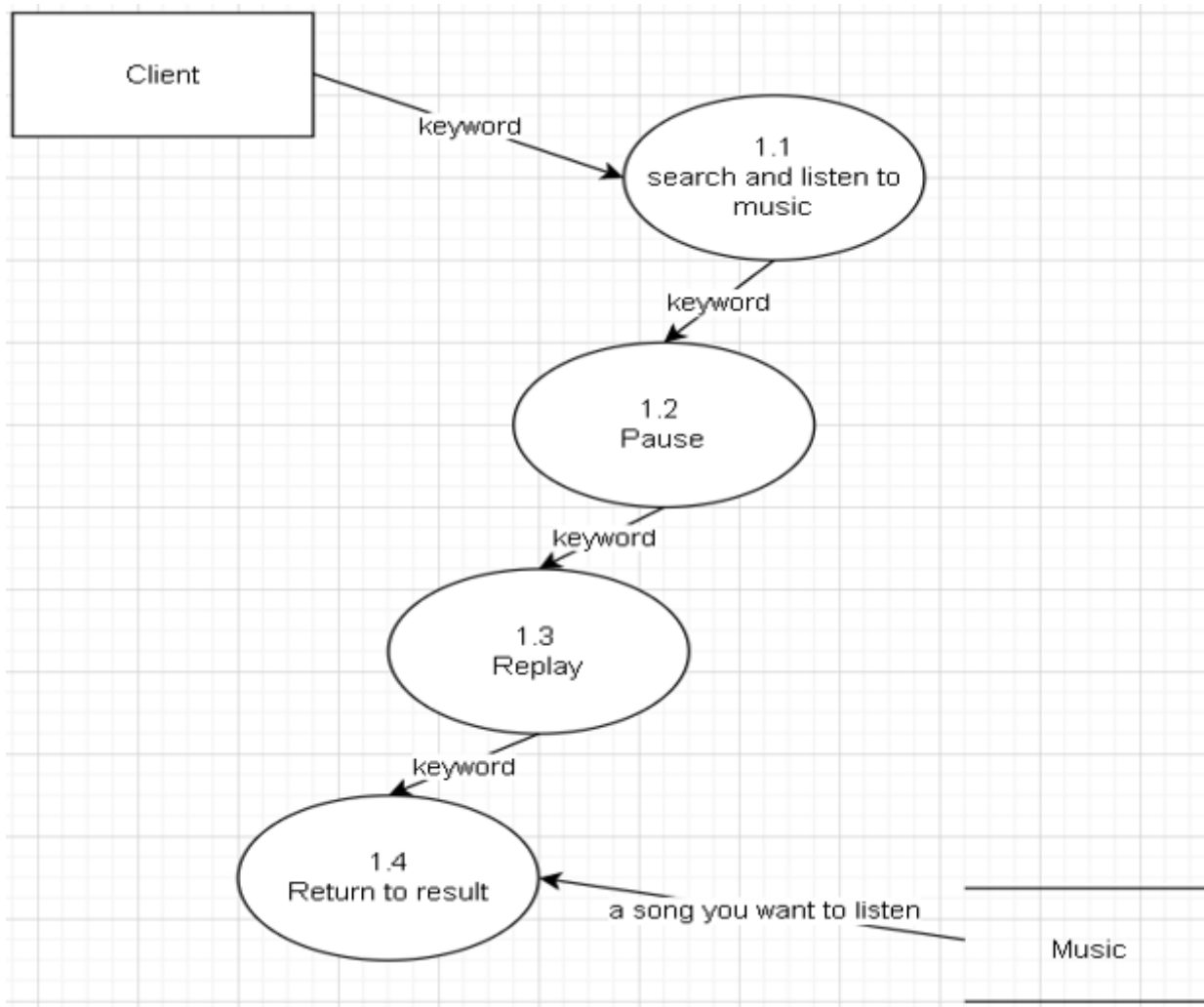
**ERD model:**



**DFD of listen to music:**



**DFD of Pause and Replay music:**



**Pseudocode:**

```

GET LoginID
GET Password
IF (LoginID == EnteredUsername && Password == EnteredPassword) THEN
Login Successful
ELSE
Login Failed.
ENDIF
  
```

**P7. Discuss how the user and software requirements are addressed in the design phase.**

**1. Intro purpose of this section.**

- I will address all user-related complaints in this section.

**2. Intro details of use case diagram.**

**Definion**

Use case diagrams are a useful tool for condensing in-depth information about a system and its users. It is often displayed as a graphic that shows how various system components interact with one another. While a use case diagram will detail system events and how they occur, it does not go into detail on how those events are implemented.

### **Advantages of use case diagrams**

Tools like use case diagrams are useful and adaptable. They may assist your team in communicating and analyzing the breadth of your system, any circumstances in which it interacts with a client, business, or system outside of it, as well as the objective or issue your app helps your customers resolve. Any time you need to describe the context and conditions necessary for the system to function or represent the sequence of events in a specific use case, create a use case diagram.

### **Steps to use**

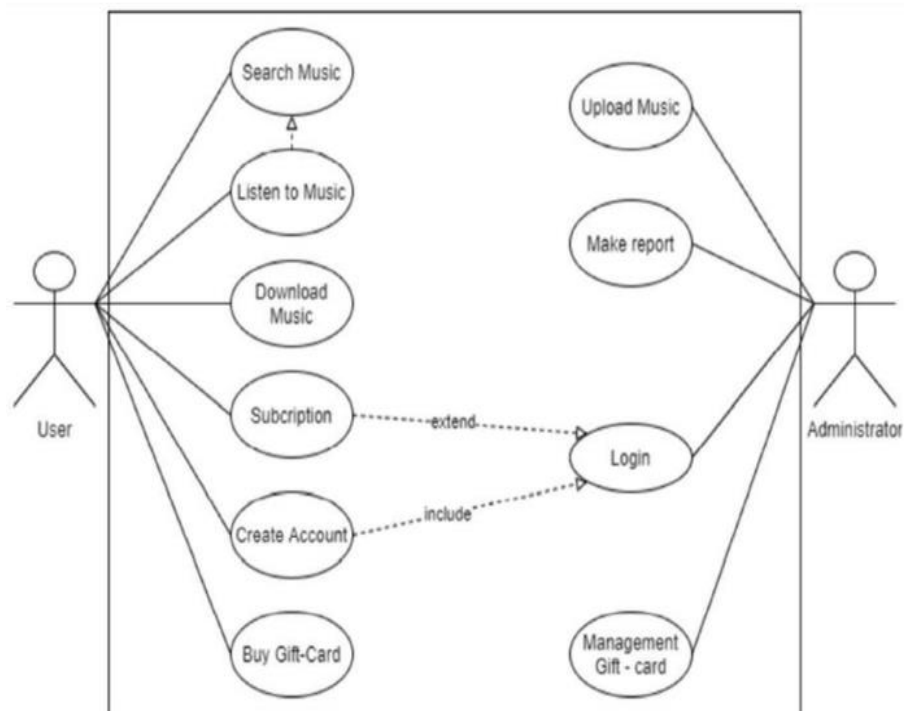
**Step 1:** is to choose your actors. Anyone who engages with your system is an actor. Customers, users, people, systems, or organizations can all be actors.

**Step 2:** Identify use cases in. As a general guideline, ask actors what they need from the system before anything else. Actors, for instance, must borrow and return books from the library, obtain library cards, make reservations, and utilize the internet. All of them are usage cases.

**Step 3:** Determine which common functions may be applied throughout the whole system. If a function is shared by two or more of your use cases, you can add that function to a different use case.

**Step 4:** Recognize the contours. Exist any actors engaged in comparable use cases that result in special circumstances for them? Then you may extrapolate to that specific agent. The use case for “make payment” in a payment system, which may be generalized to “pay by debit card”, “pay with cash”, “pay with by credit card”, etc... one that is frequently used as an example.

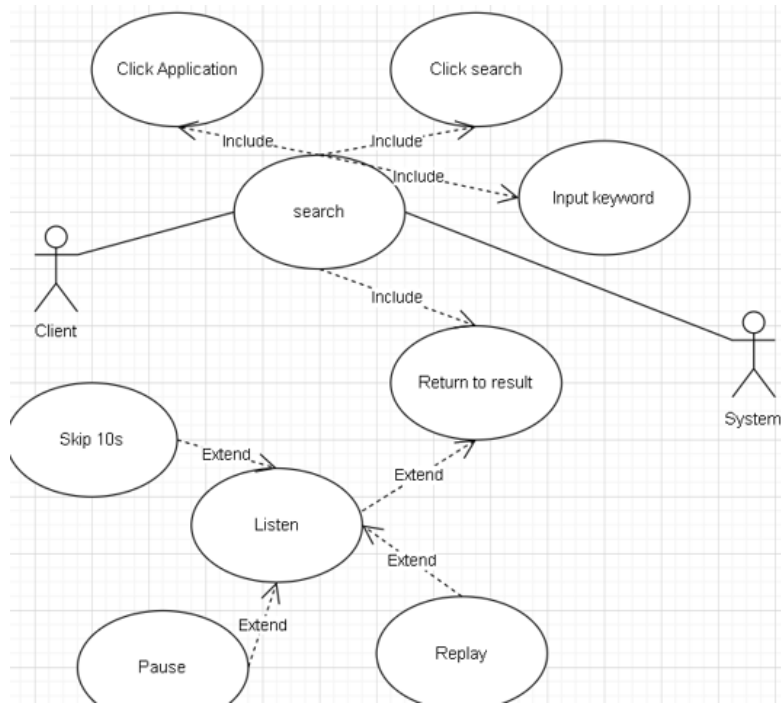
### 3. Apply to TS.



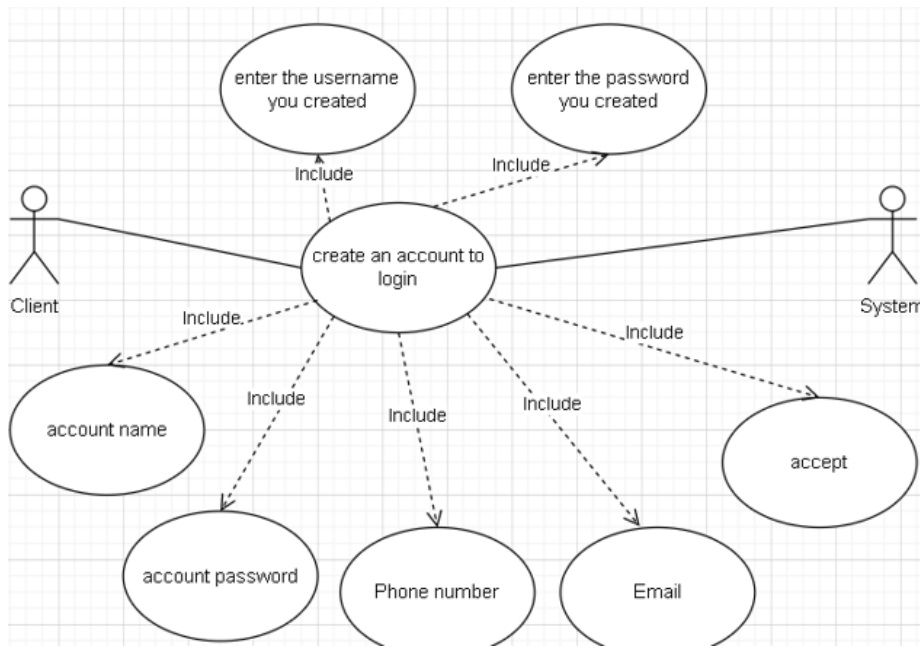
#### **Explain:**

- Above we can see 2 people representing the user and admin, the user can use the features created by the admin such as: search for music, listen to music, download music, register and create an account. buy card.admin can manage such as: upload music, report, login, manage tags.
- 4 elaborating UC diagrams.
- Diagram Pause and Replay music.

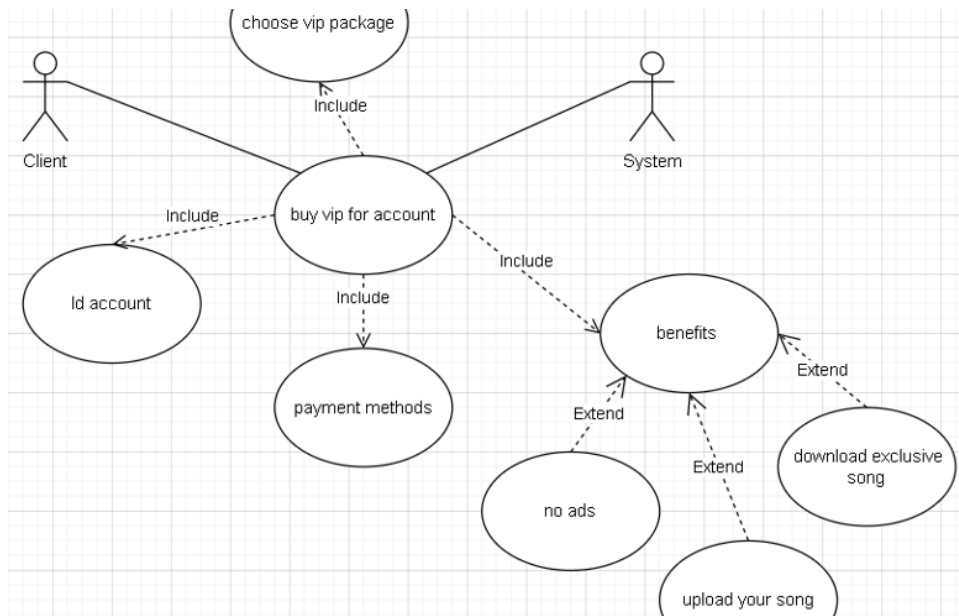




**Diagram create an account to login.**



**Diagram buy VIP for account.**



## REFERENCES.

1. Thetoolkit.me – Document Analysis [online] – Available at: <https://thetoolkit.me/123-method/theory-based-evaluation/step-3-data-checking/document-analysis/> [Accessed July 1, 2022].
2. Guru99.com – non-functional requirement[online] – Available at: <https://www.guru99.com/non-functional-requirement-type-example.html> [Accessed July 1, 2022].
3. Guru99.com – functional requirement[online] – Available at: [https://www.guru99.com/functional-requirement-specification-example.html#:~:text=A%20Functional%20Requirement%20\(FR\)%20is,%2C%20its%20behavior%2C%20and%20outputs](https://www.guru99.com/functional-requirement-specification-example.html#:~:text=A%20Functional%20Requirement%20(FR)%20is,%2C%20its%20behavior%2C%20and%20outputs). [Accessed July 1, 2022]
4. Luc.id – What is a questionnaire [online] – Available at: <https://luc.id/knowledgehub/what-is-a-questionnaire/> [Accessed July 1, 2022].
5. Tectarget.com- what is definition use case diagram[online] – Available at: <https://www.techtarget.com/whatis/definition/use-case-diagram> [Accessed July 1, 2022]
6. studentprojectguide.com- Pseudocode[online] – Available at: <https://studentprojectguide.com/project-report/pseudo-code-for-login-page/> [Accessed July 1, 2022]

