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	2020 – 2021		
Unit Tutor	Vo Ngoc Mai		
Issue date		**Submission date**	22/12/2020
IV name and date	Nguyen Vuong Thai Bao		

Submission Format:
Format: The submission is in the form of 1 document You must use font *Calibri size 12, set number of the pages and use multiple line spacing at 1.3. Margins must be: left: 1.25 cm; right: 1 cm; top: 1 cm and bottom: 1 cm.* The reference follows Harvard referencing system. *Submission* Students are compulsory to submit the assignment in due date and in a way requested by the Tutors. The form of submission will be a soft copy posted on http://cms.greenwich.edu.vn/ *Note:* The Assignment *must* be your own work, and not copied by or from another student or from books etc. 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 know how to reference properly, and that understand the guidelines on plagiarism. *If you do not, you definitely get failed*
Unit Learning Outcomes:
LO3 Undertake a software development lifecycle **LO4** Discuss the suitability of software behavioural design techniques
Assignment Brief and Guidance:
Task1

Now your team had been accepted to create the Software to Tune Source. As a member of a development team, your task now is to produce the requirements for Tune Source. You also need to specify the technique(s) or processes you used in order to get these requirements.

Task 2

Based on the requirements which established in Task1 provide the following diagrams: Use Case, ERD, DFD.. which can help to identify more clearly about the system you are going to implement.

Task 3

Based on your understanding about the Tune Source's requirements in Task1 and Task 2, show how the requirement can be addressed. Your method could include software behavioural specification methods and reliability and effectiveness of software.

Task 4

Your client want to improve the software quality. Create a report which shows how software quality could be improved from tracing requirements and programme design.

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.
P5 Undertake a software investigation to meet a business need. **P6** Use appropriate software analysis tools/techniques to carry out a software investigation and create supporting documentation.	**M3** Analyse how software requirements can be traced throughout the software lifecycle. **M4** Discuss two approaches to improving software quality.	
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.
P7 Explain how user and software requirements have been addressed.	**M5** Suggest two software behavioural specification methods and illustrate their use with an example. **M6** Differentiate between a finite state machine (FSM) and an extended-FSM, providing an application for both.	

Table of Contents

Answer Assgignment 2

P5 Undertake a software investigation to meet a business need.

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

1. Introduce :

- **Takeholder requirements.** The needs of discrete stakeholder groups are also specified to define what they expect from a particular solution.
- **Solution requirements.** Solution requirements describe the characteristics that a product must have to meet the needs of the stakeholders and the business itself.
  - **Nonfunctional** requirements describe the general characteristics of a system. They are also known as quality attributes.
  - **Functional** requirements describe how a product must behave, what its features and functions.

2. Identify the stakeholders, theirs roles and interests

2.1. What is Steakholder:

- A stakeholder is a party that has an interest in a company and can either affect or be affected by the business. The primary stakeholders in a typical corporation are its investors, employees, customers, and suppliers. However, with the increasing attention on corporate social responsibility, the concept has been extended to include communities, governments, and trade associations.

2.2. Types of Stakeholders

- Internal Stakeholders: They are a part of the management of the company and have voting powers. They are the major investors in the company and a part of the board of directors. Therefore they have all the powers that other higher-level management have and can change the direction of the company.
- External Stakeholders: Unlike internal stakeholders, their major role is to invest or disinvest in the company. They hardly can bring any change in the company's direction. They do not take part in any internal operations or decision making of the company.

2.3. Benefit of Steaholdeer:

- Education
  - Communicating directly with a stakeholder allows you to learn not only their perspective, but can provide new insights on a product or issue to help you gain a competitive advantage. Be sure to include diverse perspectives in your outreach to ensure you are hearing from all angles. You never know what you may learn!
- Effective Decision Making

- The education gleaned from the fresh perspectives described above may change your mind on an issue and allow you to make a more informed decision. Or better yet, hearing from your stakeholders may reinforce a decision you've already made. It never hurts to consider a problem from a different approach, and an informed decision should always be the goal.

Trust

- When you reach out to stakeholders, you are letting them know you value their perspective. This collaborative approach helps build trust and goodwill toward you and your organization. This can be especially useful if you're working in the wake of a crisis; rebuilding trust can be a long, arduous process, but it starts with making sure all of your stakeholders feel like they have a seat at the table.

Cost Savings

- Engaging with stakeholders can ultimately save time and money. Data shows that companies who engage stakeholders improve their chances of finishing a project on time and on budget. That savings can come from the elimination of roadblocks, and the mitigation of surprises that can slow your organization's process.

Risk Management

- Groups and individuals may help you identify potential risks before they become threats to your project or organization. Preventing these threats also eliminates the harm (budgetary and otherwise!) They can bring.

Accountability

- In the end, engaging with groups and individuals is key to improving accountability within your own organization as well as with external audiences. Transparency is important – be clear about the outcomes you are hoping to achieve and the steps you are taking along on the way. Don't forget to follow up with your stakeholders to let them know how you are doing!

2.4. Roles of Stakeholders in Tune Source project

Direct the Management: The stakeholders are Carly Edwards, Assistant Vice President, Marketing . They take over certain departments like service, human resources or research and development and manage them for ensuring success.

They Bring in Money: Stakeholders are the large investors of the company and they can anytime bring in or take out money from the company. Their decision shall depend upon the company's financial performance. Therefore they can pressurize the management for financial reports and

change tactics if necessary. Some stakeholders can even increase or decrease the investment to change the share price in the market and thus make the conditions favorable for them.

- **Help in Decision Making**: Major stakeholders are part of the board of directors. Therefore they also take decisions along with other board members. They have the power to disrupt the decisions as well. They and bring n more ideas a threaten the management to obey them. The stakeholders also have all the powers to appoint senior-level management. Therefore, they are there in all the major decision-making areas. They also take decisions regarding liquidations and also acquisitions.
- **Corporate Conscience**: Large stakeholders are the major stakeholders of the company and have monitored over all the major activities of the company. They can make the company abide by human rights and environmental laws. They also monitor the outsourcing activities and may vote against any business decision if it harms the long term goals of the company.
- **Other Responsibilities**: Apart from the above four major roles they also have some other roles to play in the company. They can identify new areas for market penetration and increased sales. They can bring in more marketing ideas. They also attract other investors like honeybees in the company. They can be a part of a selection board or a representative for the company. Moreover, they can take all the major social and environmental decisions.

❖ ***The expectations of the users and group that make up the project:***

- Our team has high hopes on this project for its viability. Because we have calculated and compared the success and risk of the "Tune source" project. And we are confident that we will complete this project well
- On the part of the user, it is expected that this project will be able to develop in the best way, minimize risks and be implemented well.

3. Requirement definition of the project (FRs and NFRs)

3.1. Functional requirement:

3.1.1. What is functional requirement:

- Functional requirements define the basic system behaviour. Essentially, they are what the system does or must not do, and can be thought of in terms of how the system responds to inputs. Functional requirements usually define if/then behaviours and include calculations, data input, and business processes.

- Functional requirements are features that allow the system to function as it was intended. Put another way, if the functional requirements are not met, the system will not work. Functional requirements are product features and focus on user requiremen

3.1.2. What should be included in the Functional Requirements Document?

Functional Requirements should include the following things:

- Details of operations conducted in every screen
- Data handling logic should be entered into the system
- It should have descriptions of system reports or other outputs
- Complete information about the workflows performed by the system
- It should clearly define who will be allowed to create/modify/delete the data in the system
- How the system will fulfill applicable regulatory and compliance needs should be captured in the functional document

3.1.3. Benefits of Functional Requirement

- Here, are the pros/advantages of creating a typical functional requirement document-

- Helps you to check whether the application is providing all the functionalities that were mentioned in the functional requirement of that application
- A functional requirement document helps you to define the functionality of a system or one of its subsystems.
- Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.
- Errors caught in the Functional requirement gathering stage are the cheapest to fix.
- Support user goals, tasks, or activities

3.1.4. Types of Functional Requirements

Here, are the most common functional requirement types

- Transaction Handling
- Business Rules
- Certification Requirements
- Reporting Requirements

- Administrative functions
- Authorization levels
- Audit Tracking
- External Interfaces
- Historical Data management
- Legal and Regulatory Requirements

3.1.5. FRs of Tune Source project:

- Authenticate user when logging in to the system to personalize and download music.
- The user can listen to the sample music
- Users can find the names of the songs they need in the database
- Shut down the system in case of a network attack.
- Verification emails or phone number are sent to users whenever they register for the first time on some software systems.
- User can use the subscribe function to prioritize their favorite songs.
- In addition, user can buy gift card to download free music for a certain period of time

3.2. Nonfunctional requirements

3.2.1. What is Nonfunctional requirements ?

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, "how fast does the website load?" Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

3.2.2. Types of Non-functional Requirement

- Usability requirement
- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement

- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

3.2.3. Advantages of Non-Functional Requirement

- **Benefits/pros of Non-functional testing are:**
  - The nonfunctional requirements ensure the software system follow legal and compliance rules.
  - They ensure the reliability, availability, and performance of the software system
  - They ensure good user experience and ease of operating the software.
  - They help in formulating security policy of the software system.

3.2.4. Disadvantages of Non-functional requirement

- Cons/drawbacks of Non-function requirement are:
  - 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.

3.2.5. NFRs of Tune Source project

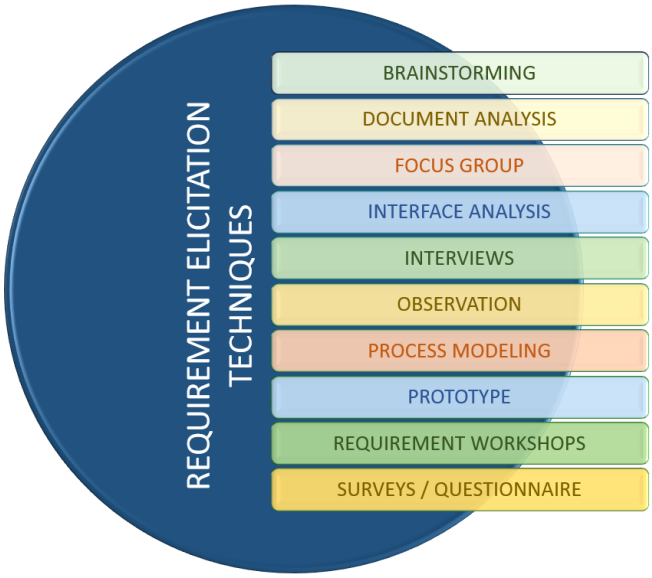
- Database of Tune source system can store an unlimited number of songs
- Confirmation email is sent with a delay of no more than 2 minutes.
- Each request will be processed within 10 seconds.
- Website will load in 3 seconds when the number of concurrent users> 10000

4. Relationships between the FRs and NFRs:

	Function Requirement	Non- Function Requirement
What is it?	**Verb**	**Attributes**
Requirement	It is mandatory	It is non-mandatory
Capturing type	It is captured in use case.	It is captured as a quality attribute.
End-result	Product feature	Product properties
Capturing	Easy to capture	Hard to capture

Objective	Helps you verify the functionality of the software.	Helps you to verify the performance of the software.
Area of focus	Focus on user requirement	Concentrates on the user's expectation.
Documentation	Describe what the product does	Describes how the product works
Type of Testing	Functional Testing like System, Integration, End to End, API testing, etc.	Non-Functional Testing like Performance, Stress, Usability, Security testing, etc.
Test Execution	Test Execution is done before non-functional testing.	After the functional testing
Product Info	Product Features	Product Properties

II. Discuss the technique(s) you did use to obtain the requirements.



1. Introduction:

- Though there are a plethora of requirements elicitation methods, but there are five key requirements elicitation techniques. On the other hand, all of them are needed at some time or the other, but the most famous ones are the needed the most. Any business professional cannot do without these famous elicitation techniques. So, which one to pick when to get the most from it yielding oodles of benefits? Deciding on the most suitable one depends on a lot of factors, which must be contemplated prior to the selection to get the desired outcomes. The firm's structural set-up, the strengths of the team members for the project, the environment of your project, the firm's requirements and likings, all these factors must be taken into account before zeroing on any elicitation technique. Based on these factors decide on the method that works best for your firm and the project.

- Cases in point, let us discuss the most famous requirements elicitation techniques that specialists need them at any point of the project development, which in turn will provide a pronounced profit on the venture.

2. Five requirement gathering techniques

- **1 - Brainstorming**

Brainstorming is an exceptional method of eliciting a lot of innovative notions for a zone of curiosity. Organized brainstorming yields several innovative notions around any particular vital request or area. When the brain is used to storm an imaginative difficulty by audaciously confronting the similar objective, it is brainstorming. It is a method, which supports diversion sort of thoughtfulness. Here, in this case, diversion talks about those team happenings that creates a wide or assorted set of opportunities. With the help of brainstorming, a lot of questions are solved. They are as follows:

- How many choices are obtainable to get an answer to the existing problem just around the corner?
  • What aspects are compelling the group from moving forward with a method or choice?
  • Which factor is delaying the activities involved?
- In what way can the problem be solved by a particular group?

This exercise of Brainstorming emphasizes on an issue or problem, and then offering a lot of fundamental way outs to it. This practice is best useful in a crowd as it appeals on the knowledge and imagination of all participants in the crowd. This really helps in sparking novel ideas to get a perfect solution to the existing problem.

- **2 - Prototyping**

Prototyping is a comparatively current technique for collecting requirements. In this line of attack, you collect introductory requirements for creating a primary type of the explanation — known as a prototype. After the creation of the primary type, it is shown to the customer to get additional requirements. Based on the client's inputs the application is altered accordingly and then this process is repeated again till the application meets client requirements, business needs and brand essence.

- **3 - Requirements Workshops**

Requirements workshops includes collecting an earlier recognized investors in a planned situation for a definite quantity of time so as to elicit, improve, and/or revise necessities. To get fruitful results, requirements workshops must contain a recorder to record contributors' effort. As contributors' may also brainstorm together, they can offer instant reaction to recognized business requirements, which can confirm a firm, effective elicitation of requirements.

- **4 - Observations**

© Can Stock Photo

Observation helps the experts in gauging and picking up the useful info that is already present. This plays a key role as all the business analysts or experts is able to document what he or she perceives through many kinds of diagramming and business procedure prototypes along with the use cases. This way an expert is able to make out the exact needs of the business allowing them to use the most comfortable techniques and methods. It is definitely a decent exercise for an analyst to offer summaries from her observations. Also, it allows the analyst to offer an oral narrative of her understanding of the work making sure that there were no confusions of the procedure.

- **5 - Interviews**

In gathering requirements, interviews play a major role. From one to one to group interviews, they help in gathering information. Ahead of time, you can discuss the business stakeholder's thoughts, which allows getting his or her viewpoint on the business requirement and the viability of probable explanations. As a result, they are one of the most effective gathering data techniques.

3. Conclusion

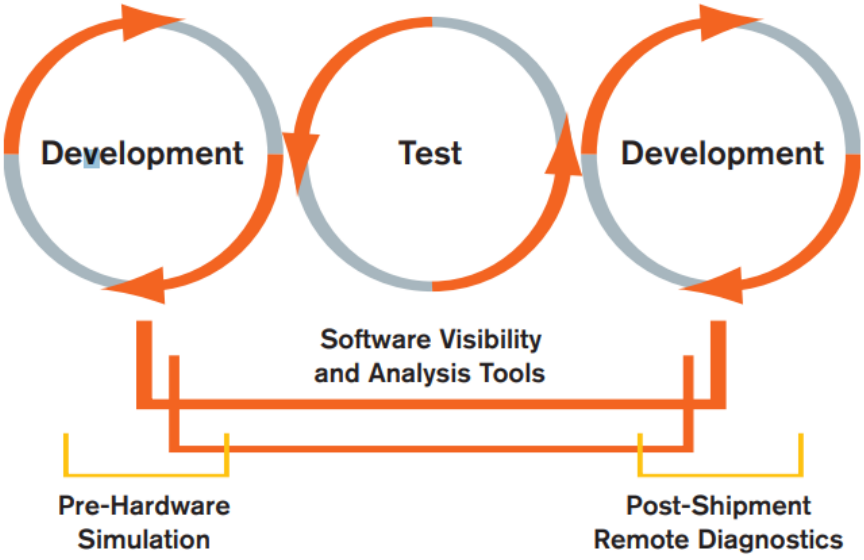
- In a nutshell, we have understood that for doing the job effectively at requirements elicitation; brace elicitation abilities to avoid getting our actions off track. We have used all five techniques outlined above to obtain the requirements in tune source project.

- We make sure that everyone from the team is involved in the conversation to outline the business requirement before designing the answer for an effective outcome. And we make sure that there is a common idea of the necessities so that the necessities can be documented in a combined tactic with the all the Tune Source project team members.
- And then the documented elicitation will be shared with suitable shareholders to authorize their arrangement with the results. This step is vital to make sure the analyst will have perfectly grabbed, and stakeholders have precisely conversed, the venture's requirements.

P6 Use appropriate software analysis tools/techniques to carry out a software investigation and create supporting documentation.

1. Introduction

When selecting which software analysis tools to use, it is important to consider the entire software development process. From emulation and simulation in the pre-hardware phase to remote diagnostics after the product has shipped, thorough data streams are crucial. Software analysis tools can provide this data at every stage of the cycle.



1.1. UML (Unified Modeling Language):

UML is an acronym that stands for **Unified Modeling Language**. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on **diagrammatic representations** of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

- **What is the use of UML?**
  - Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficacy.
- **Sketch**

- UML diagrams, in this case, are used to communicate different aspects and characteristics of a system. However, this is only a top-level view of the system and will most probably not include all the necessary details to execute the project until the very end.

  - **Forward Design** – The design of the sketch is done before coding the application. This is done to get a better view of the system or workflow that you are trying to create. Many design issues or flaws can be revealed, thus improving the overall project health and well-being.
  - **Backward Design** – After writing the code, the UML diagrams are drawn as a form of documentation for the different activities, roles, actors, and workflows.

Blueprint

- In such a case, the UML diagram serves as a complete design that requires solely the actual implementation of the system or software. Often, this is done by using CASE tools (Computer Aided Software Engineering Tools). The main drawback of using CASE tools is that they require a certain level of expertise, user training as well as management and staff commitment.

Pseudo Programming Language

- UML is not a stand-alone programming language like Java, C++ or Python, however, with the right tools, it can turn into a pseudo programming language. In order to achieve this, the whole system needs to be documented in different UML diagrams and, by using the right software, the diagrams can be directly translated into code. This method can only be beneficial if the time it takes to draw the diagrams would take less time than writing the actual code.

2. Structured analysis and design techniques

2.1. DFD

Introduction: *A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.*

❖ ***Below is the data-flow diagram of the "Tune Source" project.***

2.1.1. DFD Lv 0 1 of Tune Source project:

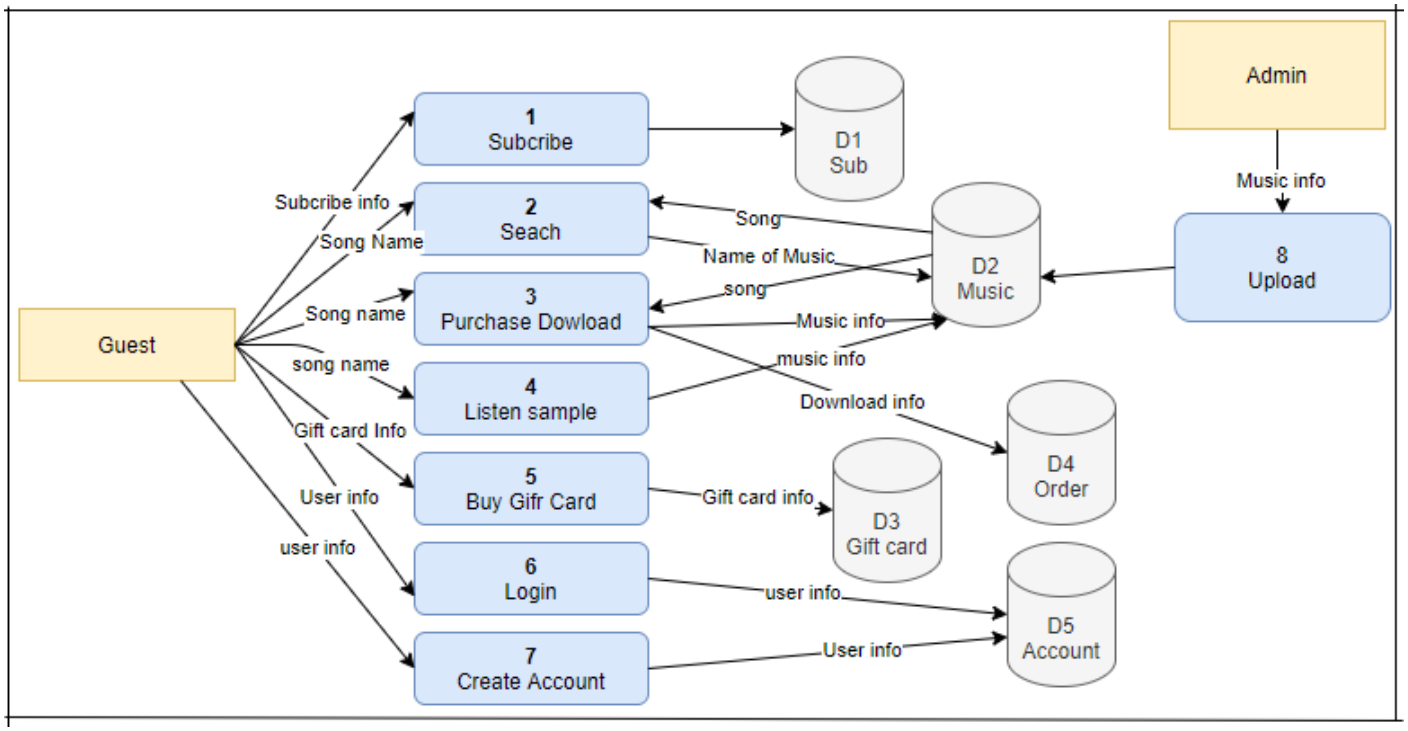


Figure 1: This DFD level 0 diagram shows an overview of the user and administrator's interaction with the "Tune source" system from the moment data was entered until data was accessed.

2.1.2. DFD Lv 1 of Tune Source project

- **Create Account**

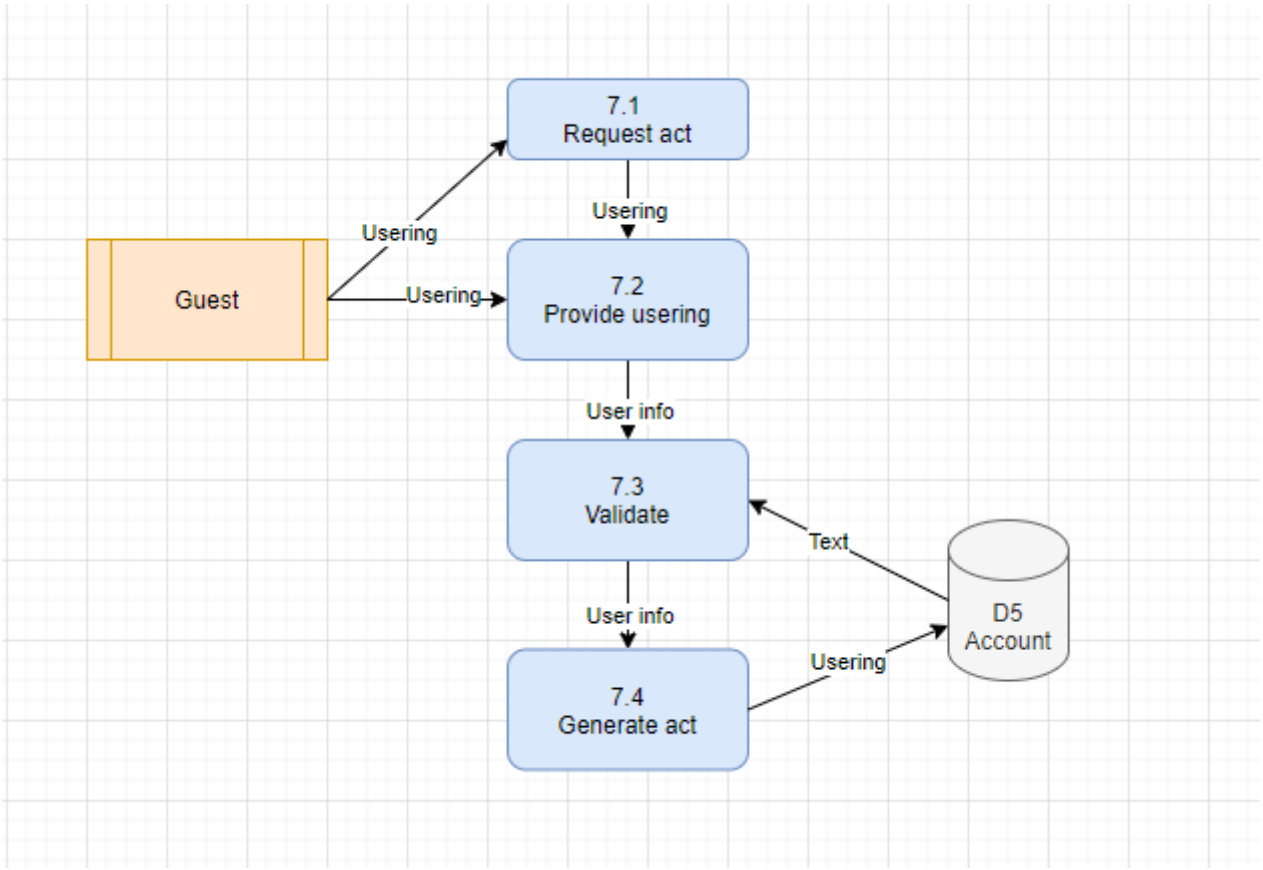


Figure 2 User Create Account: This DFD level 1 diagram shows the user's process when creating a personal account, the User requests to create an account (7.1), then the user will provide their information (7.2), System will confirm that the account matches or not (duplicate check, data type) (7.3), if the account is valid it will be created (7.4) and saved to the database (account data store). Users can use the account they have created.

- **Listen sample music:**

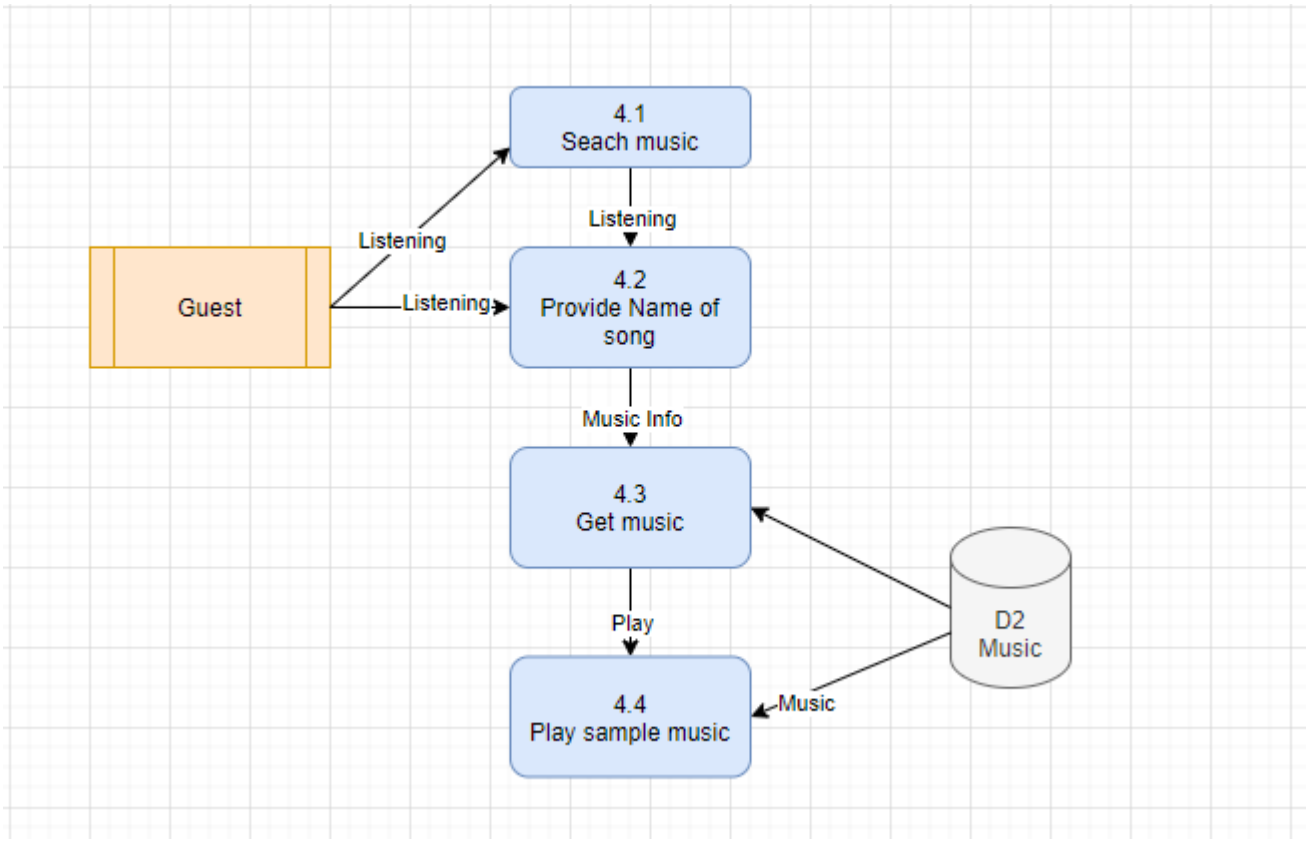
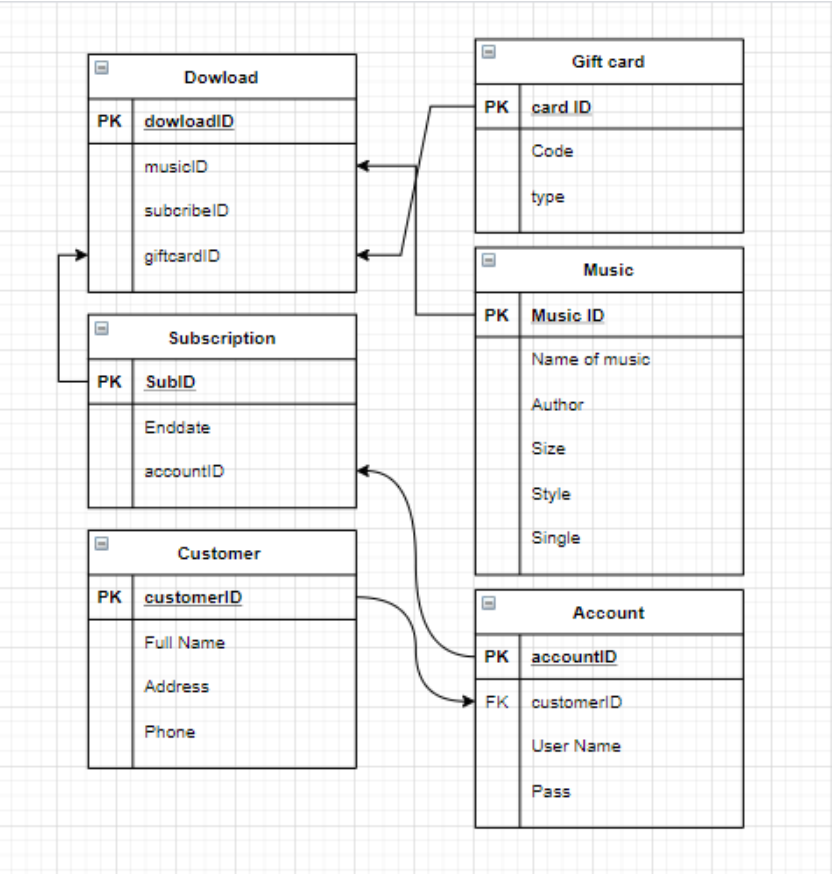


Figure 3 User Listen Sample Music: This level 1 DFD diagram shows the user's progress when finding a song and listening to a sample of the song they are looking for. The user will search for the song they need (4.1), Enter the song information (4.2), after entering the song name to find the song, the system will search and give results against the song information that the user has entered, the user chooses a song (4.3), user performed sample music action (4.4)

2.2. ERD of Tune Source project:

Introduct ERD: *An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.*

- ***Below is entity relationship diagram (ERD) of the "Tune Source" project***



2.3. Flowchart of Tune Source project

- **Introduct Flowchart:**
  - *A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.*
  - *The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.*

❖ ***Below is entity relationship diagram (ERD) of the "Tune Source" project***

- **Admin Upload music into Tune Source**

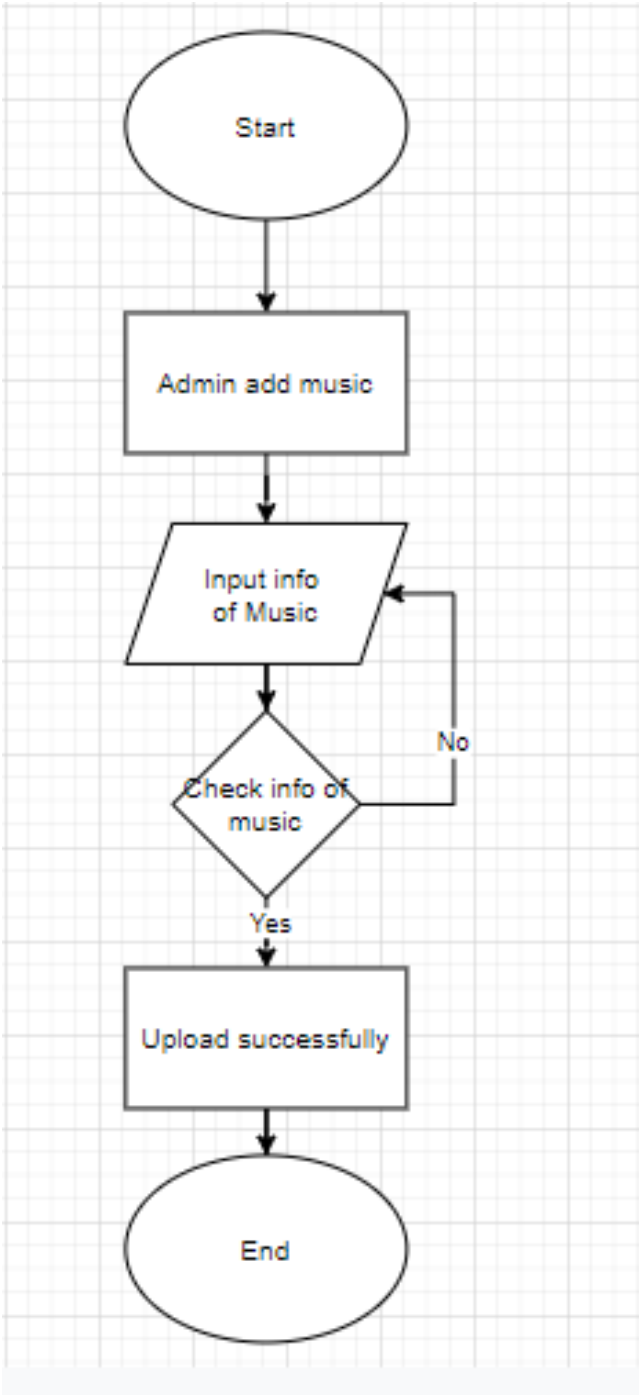


Figure 4 ADMIN up load music

- **User login Tune source:**

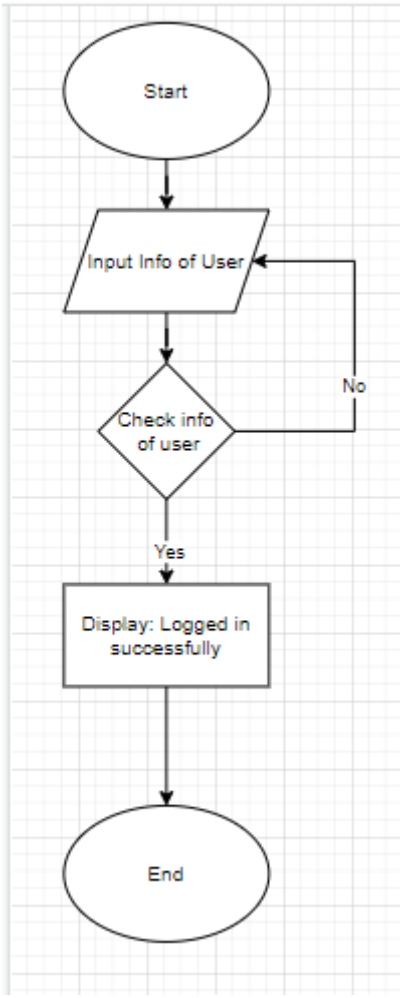


Figure 5 User login

2.4. Pseudocode:

❖ **Pseudo code for Login**

- Step 1: Start
- Step 2: Get User Name
- Step 4: Validate
- Step 5: If

(User name = Stored User name && Password = Stored Password) => Login Succeed

Else

Login failed

- Step 6: End

❖ **Pseudo code for Upload Music**

- Step 1: Start
- Step 2: Choose Music from store on Device
- Step 2: Input Music info
- Step 3: Check size

If Size > 4Mp Display (This file over size)

Else

Display (Upload Succeed)

- Step 4: Save folder upload into Music store
- Step 5: Setting Access
- Step 6: End

(P7) Explain how user and software requirements have been addressed.

1. Introduction

Introduct: *A use case is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.*

- Elements of a Use Case
  - Depending on how in depth and complex you want or need to get, use cases describe a combination of the following elements:
    - **Actor** – anyone or anything that performs a behavior (who is using the system)
    - **Stakeholder** – someone or something with vested interests in the behavior of the system under discussion (SUD)
    - **Primary Actor** – stakeholder who initiates an interaction with the system to achieve a goal
    - **Preconditions** – what must be true or happen before and after the use case runs.
    - **Triggers** – this is the event that causes the use case to be initiated.
    - **Main success scenarios** [Basic Flow] – use case in which nothing goes wrong.

- **Alternative paths [**Alternative Flow] – these paths are a variation on the main theme. These exceptions are what happen when things go wrong at the system level.

2. Identifying the Major Use Cases :

Identifying Use Cases

Before we can produce a use case diagram we must first identify the groups of related scenarios - the use cases. In addition we need to identify the initiators of the use cases - the actors. Recall from the previous sections, actors reside outside of the system and interact with it; use cases describe the functionality that helps actors achieve their goals. There are many approaches to identifying actors and use cases. In this section we present a method for doing this.

To identify use cases we will take the following steps:

- **Step 1**: Identify candidate system actors.
- **Step 2**: Identify the goals of the actors.
- **Step 3**: Identify the candidate use cases.
- **Step 4**: Identify the start point for each use case.
- **Step 5**: Identify the end point for each use case.
- **Step 6**: Refine and scope units of interaction.

Below is Use case diagram of the "Tune Source" project

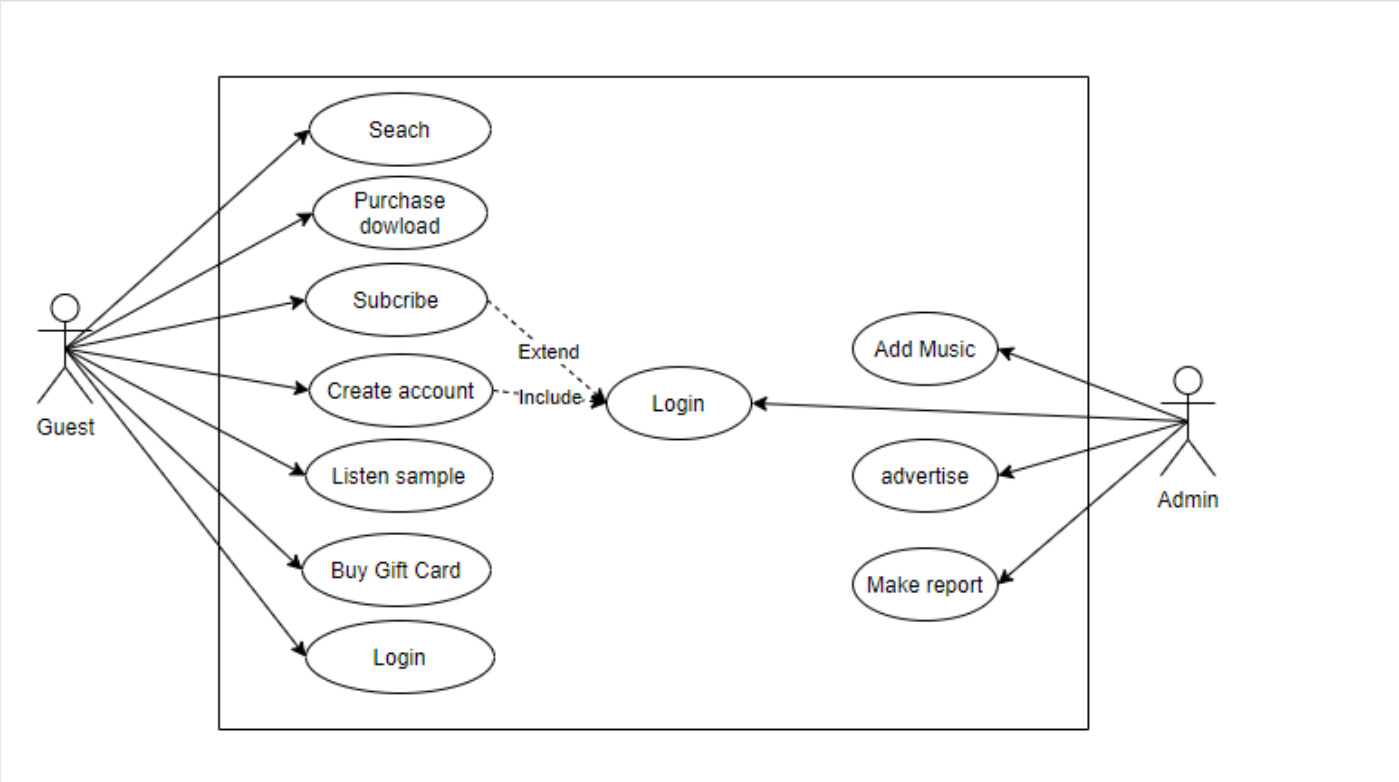


Figure 6 Use case diagram of Tune Source project

- **Actor:**
  - **Guest**: On behalf of the client, who uses and experiences "Tune Source"
  - **Admin**: On behalf of the administrator, who may grant access to, and control over, the "Tune Source" system and client information
- **Use case:**
  - **Seach**: This use case to looking for music
  - **Purchase Dowload**: This use case to dowload music
  - **Subcribe**: This use case to follow Music, Story and News the music they like
  - **Create account**: This use case to register an new account
  - **Buy Gift Card**: To buy Sale code for Download music.
  - **Login**: for personal access to the System and music can be purchased
  - **Listen sample**: To listen to the short music before downloading

3. Elaborating on the Use Cases

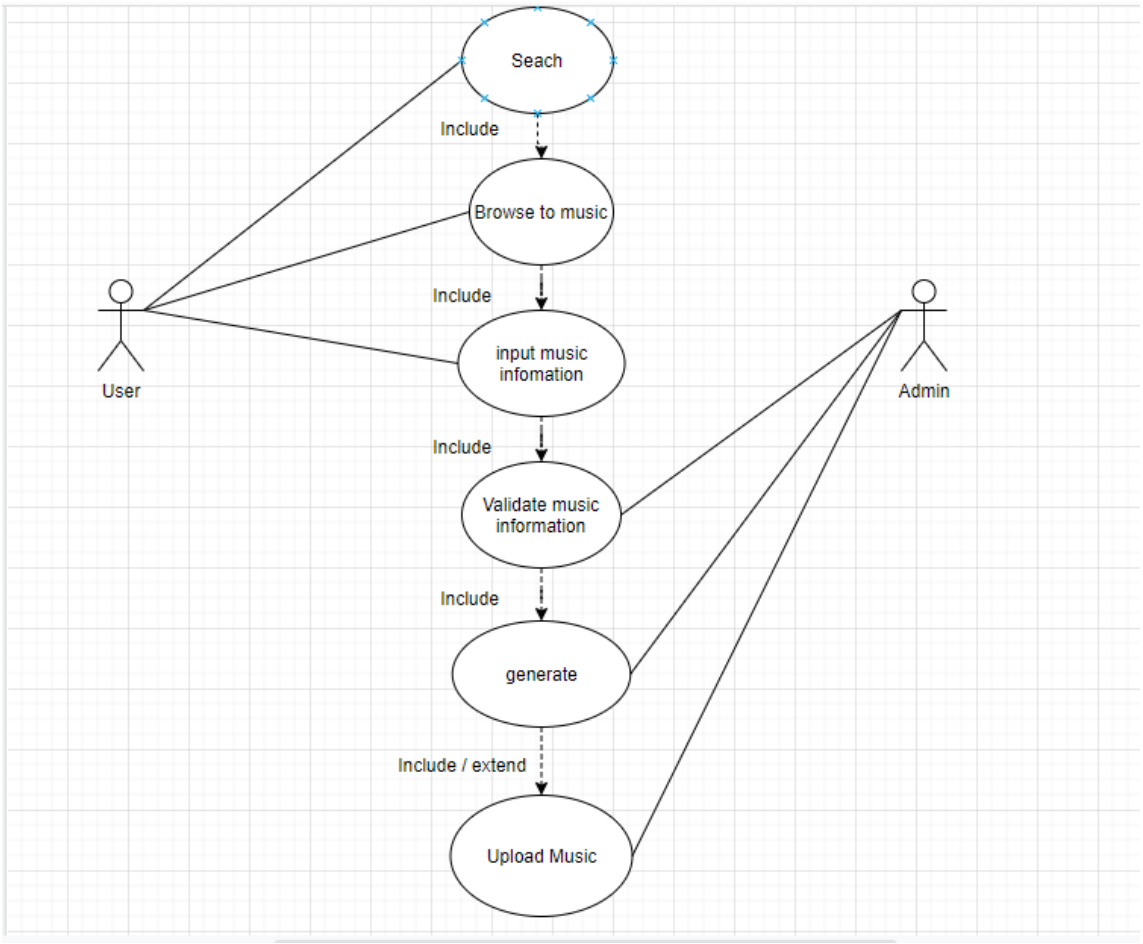


Figure 7 User seach music

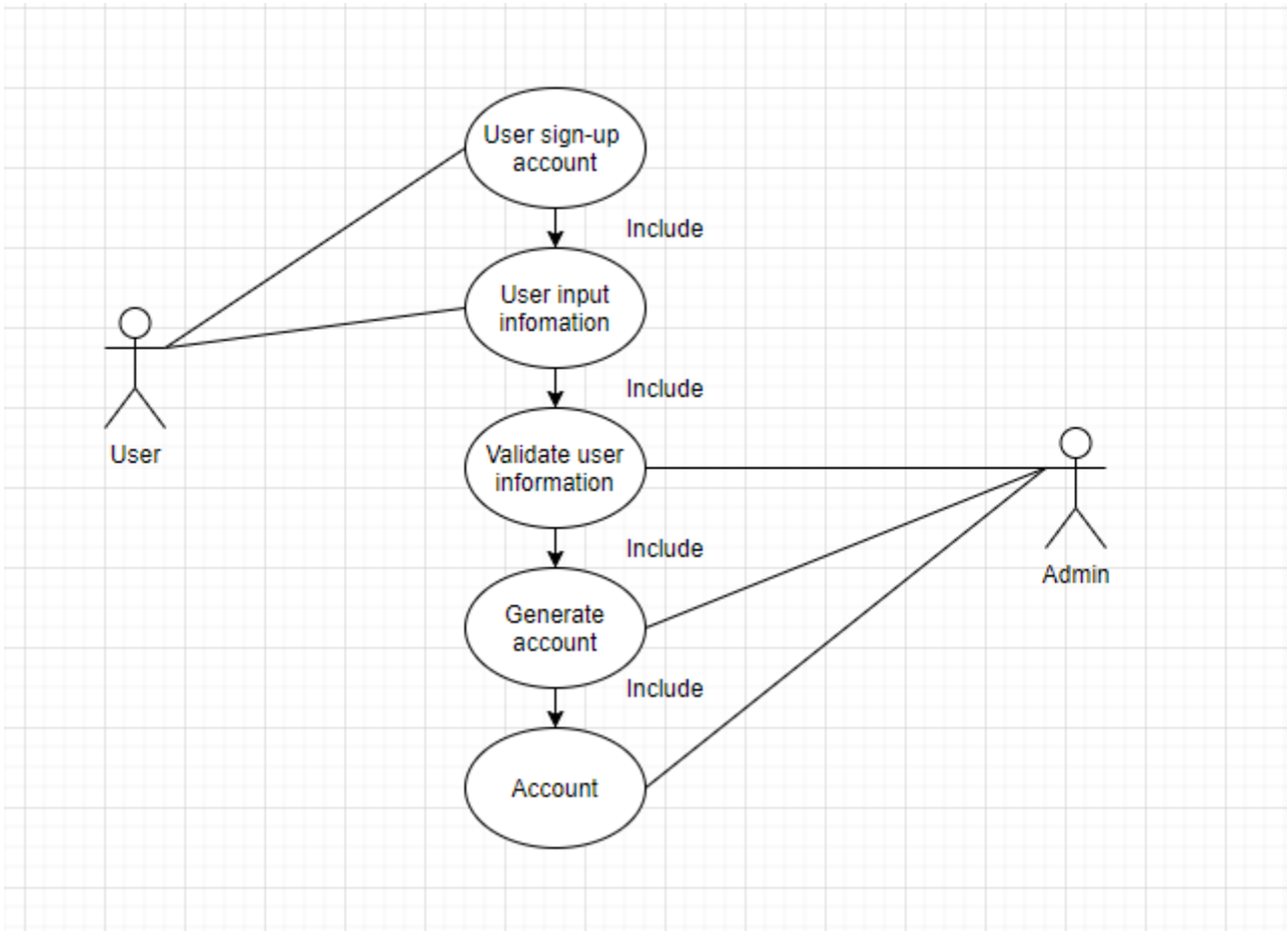


Figure 8 User create account

Reference

- Stakeholder [Online] **INVESTOPEDIA**. Available at: https://www.investopedia.com/terms/s/stakeholder.asp [Accessed 10 December.2020]
- Six Benefits of Stakeholder Engagement [Online] **NPSTRAT**. Available at : https://www.npstrat.com/highlights/six-benefits-of-stakeholder-engagement/ [Accessed 10 December.2020]
- Role of Stakeholders in Business Organization [Online] **COMPLIANCEPRIME**. Available at: https://www.complianceprime .com/blog/2019/10/17/role-of-stakeholders-in-business-organization/#:~:text=A%20stakeholder%20is%20a%20person,suppliers%2C%20vendors%20or%20any%20partner.&text=Stakeholders%20can%20also%20be%20an,the%20outcome%20of%20the%20company [Accessed 10 December.2020]
- What is Non-Functional Requirement? Types and Examples [Online] **GURU99**. Available atL https://www.guru99.com/non-functional-requirement-type-example.html [Accessed 10 December.2020]
- What is a Functional Requirement? Specification, Types, EXAMPLES [Online] **GURU99.** Available at: https://www.guru99.com/functional-requirement-specification-example.html
- Nonfunctional Requirements [Online] **SCALEDAGILEFRAMWORK**. Available at: https://www.scaledagileframework.com/nonfunctional-requirements/ [Accessed 10 December.2020]
- Why is the difference between functional and Non-functional requirements important? [Online] **REQTEST**. Available at: https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/ [Accessed 10 December.2020]
- TOP 5 REQUIREMENTS ELICITATION TECHNIQUES [Online] **Anarsolutions**. Available at: https://anarsolutions.com/top-5-requirements-elicitation-techniques/ [Accessed 10 December.2020]
- All You Need to Know About UML Diagrams: Types and 5+ Examples [Online] **Tallyfy**. Available at: https://tallyfy.com/uml-diagram/ [Accessed 10 December.2020]
- Object-Orientated Design and Programming Introduction (Online) **Cems.** Available at: http://www.cems.uwe.ac.uk/~jsa/UMLJavaShortCourse09/CGOutput/Unit1/unit1(0809)/page_24.htm [Accessed 10 December.2020]