

ASSIGNMENT 02 FRONT SHEET

Qualification	BTEC Level 5 HND Diploma in Computing		
Unit number and title	Unit 09: Software Development Life Cycle		
Submission date	26/08/2022	Date Received 1st submission	
Re-submission Date	31/08/2022	Date Received 2nd submission	31/08/2022
Student Name	Nguyen Van Tuan	Student ID	GCH200546
Class	GCH1001	Assessor name	Dinh Duc Manh
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	tuan

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:

Contents

INTRODUCTION	4
TASK 1 – ANALYSIS	5
A. UNDERTAKE A SOFTWARE INVESTIGATION TO MEET A BUSINESS NEED(P5).....	5
I. Requirement definition of the project:	5
II. Identify FRs and NFRs of the Tune Source Project.....	7
III. Discuss the relationship between the FRs and NFRs.....	9
TASK 2 – ANALYSIS	14
B. USE APPROPRIATE SOFTWARE ANALYSIS TOOLS/TECHNIQUES TO CARRY OUT A SOFTWARE INVESTIGATION AND CREATE SUPPORTING DOCUMENTATION	14
I. Use case Diagram for the whole system.....	14
II. Use Case specification for 2 Use Cases	17
III. Context Diagram for the whole system.....	23
IV. Data flow Diagram-Level 0 for the whole system	23
V. ERD for the whole system.....	25
VI. Pseudo Code For One Module Of The Program	27
TASK 3 – DESIGN	29
C. Explain how user and software requirements have been addressed.....	29

I. Discuss how the user and software requirements are addressed in the design phase	29
TASK 4 – SOFTWARE QUALITY MANAGEMENT	41
I. Discuss two software quality attributes that are applicable to the project	41
II. Discuss two quality assurance techniques that can help improve the software quality in the project	42
1. Plan and record your testing process	42
2. Test-driven development	43
3. Discuss how the design techniques and approaches that you have to used can help improve the software quality	43
CONCLUSION	43
References	44

INTRODUCTION

The Software Development Life Cycle (SDLC) can be thought of as a well defined process for creating high-quality, inexpensive software as quickly as possible. The goal of the SDLC is to develop great software that exceeds all requirements and expectations from customers. A complete plan with stages, or phases, that each have their own methodology and deliverables is specified and explained using the sequential development life cycle (SDLC). The software development lifecycle will be thoroughly covered in this paper, along with the applicability of software behavioral design methodologies for the TuneSource Project.

TASK 1 – ANALYSIS

A. UNDERTAKE A SOFTWARE INVESTIGATION TO MEET A BUSINESS NEED(P5)

I. Requirement definition of the project:

1. Identify the stakeholders, roles, and interests in the case study.

Identify Stakeholders is the first process in the Knowledge Area for Project Communications Management and it belongs to the Initiating process group. This process comprises identifying and documenting all project stakeholders, as well as their motivations, effects, and any detrimental effects on the undertaking. Identification of stakeholders should start as soon as it is practical and continue throughout the project's life cycle. (Johnston, 2020)

In a project, there are two categories of stakeholders: internal stakeholders and external stakeholders.

a. Internal Stakeholders:

Internal stakeholders are those who are a part of the system or business that is being assessed. If the system is a component of a hospital, internal stakeholders can include doctors, nurses, pharmacists, therapists, administrative staff, porters, and managers. Founders John Margolis, Megan Taylor, Phil Cooper, Carly Edwards, Assistant Vice President, Marketing Department, IT Department are internal stakeholders in TuneSource.

b. External Stakeholders:

External stakeholders have an impact on the system being examined even though they are not a part of it. If the system is a hospital, for instance, external partners might be GPs and community health workers, social care commissioners, professional organizations, and regulators. Customers are internal stakeholders in TuneSource.

Stakeholder	Objectives	Requirement Provided
<ul style="list-style-type: none"> Founders: John Margolis, Megan Taylor, Phil Cooper 	<ul style="list-style-type: none"> Business Growth 	<ul style="list-style-type: none"> Attracting new clients who are interested in our distinctive collection of rare and difficult-to-find music.
<ul style="list-style-type: none"> Project Sponsor: Carly Edwards, Assistant Vice President, Marketing 	<ul style="list-style-type: none"> Profit, Financial returns 	<ul style="list-style-type: none"> Create the capacity to sell digital music downloads to boost sales.

		<ul style="list-style-type: none"> Get a new source of income through gift cards and consumer subscriptions.
<ul style="list-style-type: none"> Marketing department 	<ul style="list-style-type: none"> Market share 	<ul style="list-style-type: none"> Being able to download songs digitally
<ul style="list-style-type: none"> IT department 	<ul style="list-style-type: none"> Regular payment 	<ul style="list-style-type: none"> Short-term development, specific functionality the system should have.
<ul style="list-style-type: none"> Customers 	<ul style="list-style-type: none"> Product/service quality and value 	<ul style="list-style-type: none"> Nicely built, requesting this capability.

Table 1: Stakeholders in Tune Source project clearly indicating which stakeholder(s) provide what requirements.

2. Stakeholder role with an interest

Stakeholder	Role	Interest	Note
<ul style="list-style-type: none"> Founders: John Margolis, Megan Taylor, Phil Cooper Project Sponsor: Carly Edwards, Assistant Vice President, Marketing 	<ul style="list-style-type: none"> Voting and Decision-making Investing 	<ul style="list-style-type: none"> Costs – how much money is required to run this project. Gains and higher sales Support extending the effect of TuneSource has on the community 	We need to launch this system as quickly as feasible because users have a variety of other options for downloading music. Many of our current, devoted clients have been asking for this feature, and we must now offer it or risk losing their company.
<ul style="list-style-type: none"> Marketing department IT department 	<ul style="list-style-type: none"> Environmental and social responsibilities Management 	<ul style="list-style-type: none"> There is little to no risk of injury Their financial stability Productive results 	In order to compete in our market niche, having digital music download options is essential.
<ul style="list-style-type: none"> Customers 	<ul style="list-style-type: none"> Evaluate and give feedback on value and quality 	<ul style="list-style-type: none"> Social impact, effectiveness, and software quality. 	Nothing

		<ul style="list-style-type: none"> Unique archive of rare and hard-to-find music 	
--	--	---	--

Table 2: Stakeholder role with an interest

II. Identify FRs and NFRs of the Tune Source Project.

1. Functional Requirements

An explanation of the service that the software must provide is contained in a functional requirement (FR). It describes a piece of software or a software system. A function is nothing more than the inputs, behavior, and outputs of the software system. A system's likely function can be determined by a computation, data manipulation, business process, user interaction, or any other specialized feature. In software engineering, functional specifications are another name for functional requirements. (Martin, 2022)

Functional Requirements	Reason
As managing director, I want customers to establish accounts for shopping so that I may market to them using their information.	It aids in customer management and tracking, tracking numbers, and customer system product advertising. Additionally, customer data collection may result in data sales to other parties. That's a significant rise in profits.
As an IT department, I would like to have sort and search functionality in the system to control and manage it better	The function request includes the request for data manipulation. The management of the IT department is made easier by this feature. They are therefore more proficient.
As a customer, I want to be able to buy songs in many forms such as electronic payment.	It had to do with corporate operations. This function ought to be included in the system; it is crucial for helping our system manage time efficiently and handle transactions in the quickest and most practical manner. Additionally, it can distinguish between an old and new system.
As an IT department, I wanted a hierarchy between privileged accounts and regular user accounts.	The function request category includes requests for data processing. A crucial necessity because managers are required to act within their means. They might also possess other abilities like adding

As a customer, I want to be able to customize my shopping cart	It relates to business processes and is essential to the system since it can make clients feel at ease and satisfied with the company model.
--	--

Table 3: Functional requirements for the Tune source projects

2. Non-Functional Requirements

The quality attribute of a software system is provided by the non-functional requirement (NFR). The responsiveness, usability, security, portability, and other non-functional factors that are essential to the software system's success are taken into account when evaluating it.

Non-Functional Requirements	Reason
As a Managing Director, I want the user interface to be attractive when looking at it to show that we have flair.	This is a prerequisite for a website that publishes music. Even if they haven't utilized the service, they will have a look at the user interface. That will draw in a lot of customers.
As a Managing Director, I want the app should be error free.	The system's overall efficiency and accessibility. The level of consumer satisfaction with the service is increased thanks to this criteria. This increases the amount of customers who need the service over time and aids in retaining long-term clients.
As a customer, I want to post my reviews directly on the system's homepage so that everyone can see and review my reviews.	Non-function requests are those that involve user engagement. This aids in the system's continual improvement. Upon understanding and addressing all user comments. It is a flawless method.
As an IT, I want mobile devices to be able to trigger notifications when new songs are available.	It has to do with system requirements, and this function is excellent since it can assist users receive information whenever they need it and from any location. Users won't have to worry about missing the start of the music show, and in general, a system truly needs to have this capability. Advance notice increases consumer security, improves product quality, and facilitates close cooperation in business.

As a customer, I want a feedback system where people could make suggestions for improvement.	It had to do with commercial requirements. It is particularly important since it enables organizations to track the model's development and make adjustments to advance the model. Additionally, it makes the system's use more interactive.
--	--

Table 4: Non-functional requirements for tune source projects

III. Discuss the relationship between the FRs and NFRs

The connections between functional requirements and non-functional requirements are listed below.

Functional Requirement	Non-Functional Requirement
A specification that describes a system or one of its components is known as a functional requirement.	A non-functional software system's quality attribute is determined by its
It queries, "What should the software system do?"	What should the software system do to fulfill the functional specifications? is restricted.
The functional specifications are provided by the user	Non-functional requirements are specified by technical people like architects, technical leaders, and software developers.
Functional testing includes system, integration, end-to-end, API, and other forms.	Testing that is not functional is done, including performance, stress, usability, and security testing.
Easy to capture	Hard to capture
Prior to conducting non-functional testing, tests are executed.	After the functional testing
Product Features	Product Properties

Table 5: Different between FRs & NFRs

1. Discuss the approach/technique (es) you'd take to obtain the requirements

Here are a few methods for fulfilling the requirements:

a. Joint Application Development (JAD)

JAD is a method of information collecting that enables our project team, users, and management to work together to identify system needs.

A facilitator skilled in JAD techniques leads a group of 10 to 20 users through a systematic process called JAD, which reduces scope creep by 50%.

Pros	Cons
<ul style="list-style-type: none"> The creation of the documentation only takes a few hours, and it is promptly given to the participants for review. It is feasible to certify requirements immediately. successfully gathered needs from a large group in a short period of time. Consensus can be formed when issues and queries are brought up in front of all interested parties. 	<ul style="list-style-type: none"> Accessibility of stakeholders could put the meeting at danger. The success rate depends on the facilitator's level of expertise. The objective will be impossible to accomplish if there are too many participants in the workshop.

b. Interview

The most common method for gathering requirements is the interview. Interview techniques should be used to create solid relationships between business analysts and stakeholders. In this approach, stakeholders are interviewed in order to acquire information. An interview with just one person is the most common approach.

An interview that follows a predetermined format is referred to be structured. When the interviewer does not follow a predetermined framework or ask specific questions, the interview is unstructured.

Pros	Cons
<ul style="list-style-type: none"> Stakeholders engage in a discussion that is interactive. 	<ul style="list-style-type: none"> Interview preparation and execution take time. Each participant is required to commit.

- Make careful to check in as soon as you can to see if the interviewer has understood.
- By building a rapport with the stakeholder, you may promote participation and create collaborations.

- To conduct effective interviews, training may occasionally be required.

c. Observation

Learning about other people's activities, tasks, tools, and events is the main objective of the observation session.

According to the observation plan, all stakeholders are aware of the observation session's objective, they concur on the anticipated results, and the session meets their expectations. The participants must understand that their performance will not be evaluated.

The observer must record all activities taken throughout the session as well as how long it takes each person to do a task in order to reproduce it afterwards. After the session, the BA will review the findings and get in touch with the participants.

Both active and passive forms of observation are possible.

Pros	Cons
<ul style="list-style-type: none"> • You might be able to obtain information in a single session as opposed to conducting one-on-one interviews. • A lively exchange of ideas between the participants fosters a healthy environment. • It is possible to learn from other people's experiences. 	<ul style="list-style-type: none"> • Participant agitation is possible. • If participants alter their working methods during the observation, observers might not get a clear picture. • Actions based on knowledge are imperceptible.

d. Document Analysis

By assessing and looking into the elements that exist and characterize the business environment, this approach is utilized to gather business data. This study helps to clarify the business requirement and validate the application of current solutions.

Document analysis includes, among other things, reviewing business strategies, technical papers, problem reports, and existing requirement documents. This is quite helpful when it comes to modernizing an existing system.

This method is beneficial in migration initiatives.

Comparing the AS-IS and TO-BE processes is one method for using this methodology to uncover system faults. This analysis can be useful if the person who created the current documentation is no longer employed by the system.

Pros	Cons
<ul style="list-style-type: none"> Existing documentation can be used to compare current and upcoming procedures. The use of existing documents as a jumping off point for additional study is possible. 	<ul style="list-style-type: none"> It's possible that already published papers won't get updated. Existing records can be completely out of date. The people who worked on the earlier items might not be available to provide information. This process requires lots of time.

e. Questionnaires

In order to quantify their ideas, stakeholders are given a set of questions to respond to. After gathering responses from the stakeholders, the data is assessed to ascertain the areas of interest of the stakeholders.

Questions should be based on high priority threats. It is preferable to ask straightforward inquiries. Once the survey is finished, notify the participants and remind them to take part.

Pros	Cons
<ul style="list-style-type: none"> Data collection from a sizable population is easy. Less time will be given to the participants to respond. You can get more accurate information compared to interviews. 	<ul style="list-style-type: none"> Not all stakeholders may participate in the surveys, it is conceivable. It's probable that not all of the participants will be able to grasp all of the questions. Deeper research is required in response to open-ended questions.

- It might be required to conduct follow-up surveys in light of the participants' responses.

The method of choice for gathering requirements for this project is through interviewing.

Techniques	Yes	No
JAD		×
Interview	✓	
Observation		×
Document analysis		×
Questionnaires		×

Table 6: Techniques used for requirement gathering requirement combination of two or more techniques can be allowed

2. Demonstrate how to collect requirements based on the chosen technique

Utilizing the selected Technique, needs were gathered

Step 1: Determine the essential information that we must have.

Step 2: We must identify the recipients of the survey.

Step 3: How to gather the data (phone, email, and postal mail)?

Step 4: What kind of questions will be asked

Step 5: Check that the questions only expose potential respondents and make any necessary adjustments.

Step 6: Make the survey accessible to the general public

Step 7: Determine the answer

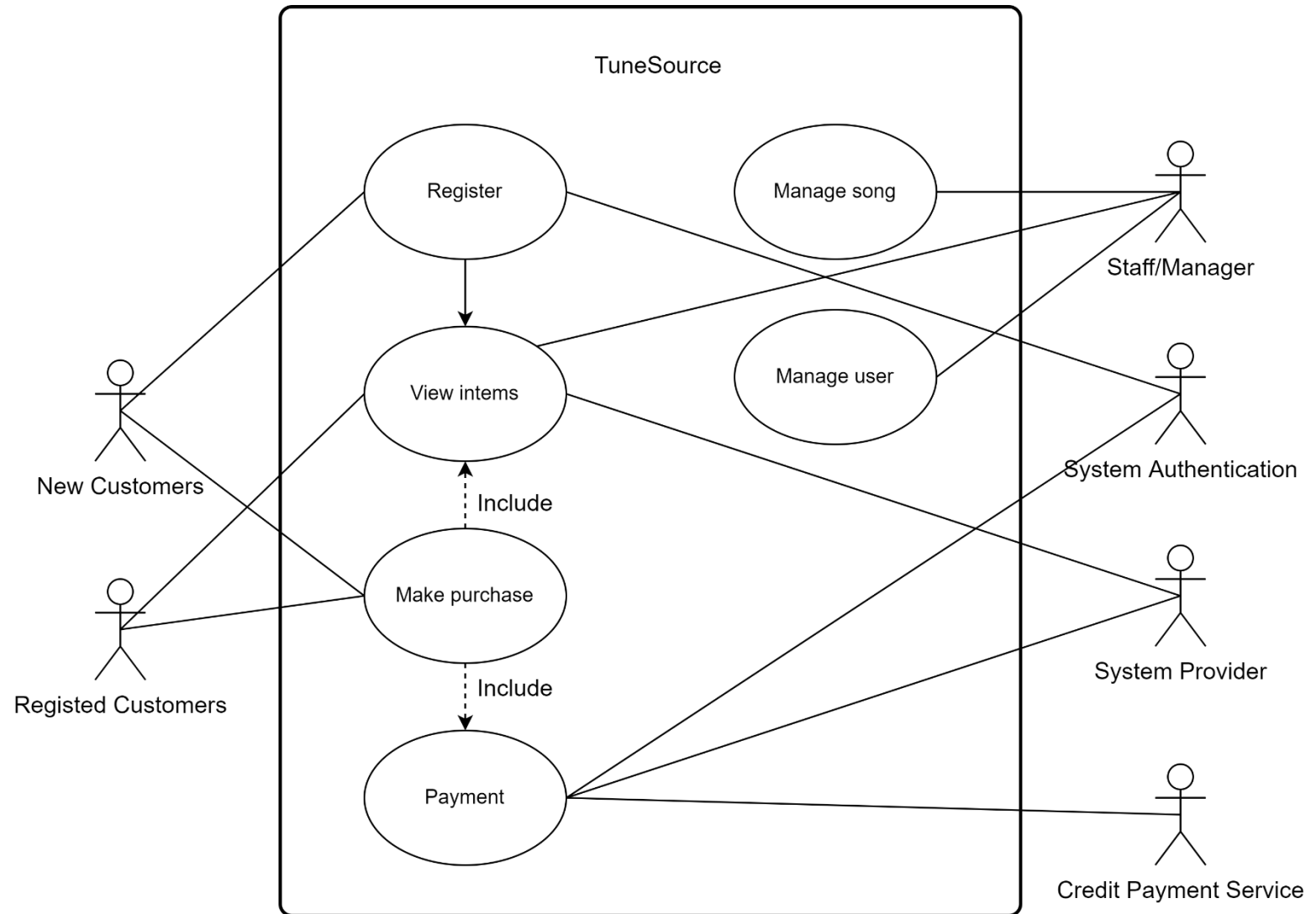
Step 8: Test

Step 9: Apply the outcome

TASK 2 – ANALYSIS

B. USE APPROPRIATE SOFTWARE ANALYSIS TOOLS/TECHNIQUES TO CARRY OUT A SOFTWARE INVESTIGATION AND CREATE SUPPORTING DOCUMENTATION

I. Use case Diagram for the whole system.



Description	The general operation of the TuneSource software system is described in this use case.
Actors	<ul style="list-style-type: none"> • Customers who want to purchase digital music or find rare audio recording • Personnel who oversee this software's data
Supporting Actors	<ul style="list-style-type: none"> • System that handles authorization and authentication • A system that is functional • System that facilitates payments
Pre-conditions	<ul style="list-style-type: none"> • The user had installed the TuneSource Software after downloading it. • For employees, the private power account issued by the developers is required.
Post conditions	When users utilize the software, the system responds to their demands.
Normal Course	<ol style="list-style-type: none"> 1) User starts the program. 2) User logs in or registers for this program. 3) The system manages these accounts' authentication and authorization. 4) Users must continue step 2 until their accounts are approved by the system. 5) The system displays the software's primary view or the user's chosen view. 6) User can choose whether to interact with the software. 7) User finds, buys, or uploads music; algorithm is implemented by system; system calculates. 8) The user's queries are handled by the system, which then answers them and stores them. 9) Time step is updated by the system. Repeat steps 5 through 9 using the system. 10) The statistic and data of this software are managed by staff or a management. 11) Users cease utilizing all functions. 12) The use case is ended by the system.
Alternatives Course	<p>If the system is still retaining the function that consumers desire to utilize at step 7 of the Normal Course</p> <ol style="list-style-type: none"> 1) System freezes the request of the user in a hold status and notifies the user. 2) Updates and an expected time period for this maintenance are provided by the system. 3) System returns to the original view, and step 6 of the normal course continues with the use case.
Exception	<ol style="list-style-type: none"> a) The use of the software is no longer permitted (step 3).

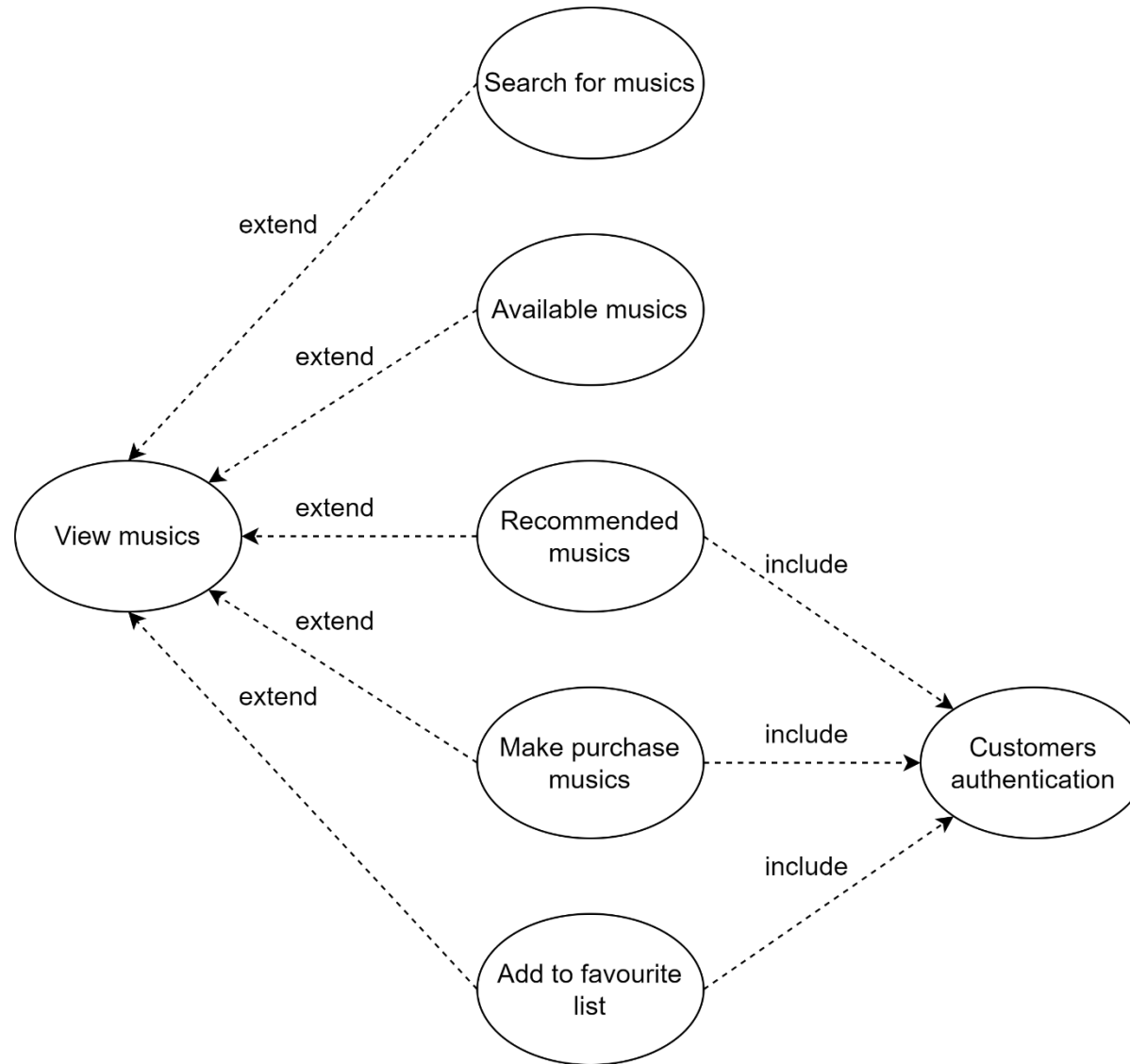
	<ul style="list-style-type: none"> b) The System provides no results. c) Users are not responded to by the system, or the system reacts to unexpected results. d) Crashing.
Requirements	<p>The user-friendliness of the system is impacted by the GUI for receiving requests.</p> <p>Prior to submitting any music purchases, balance availability must be confirmed.</p>

Explanation Use Case:

The Customer actor makes purchases of digital music using this program in this use case. View Items, Make Purchase, and Customer Register are the three most important use cases. If a consumer merely wants to search for and view some music, they may use the View Items use case as their top-level use case (record-album-song). The Make Purchase use case might also incorporate this use case. The customer Register use case enables the customer to sign up for this software, for instance, to buy a single song or a gift card for download music. Payment is a use case that is incorporated with making a music purchase and is not available separately. Only TuneSource employees and managers use the Manage Song and Manage User use case.

II. Use Case specification for 2 Use Cases

1. View music Use case



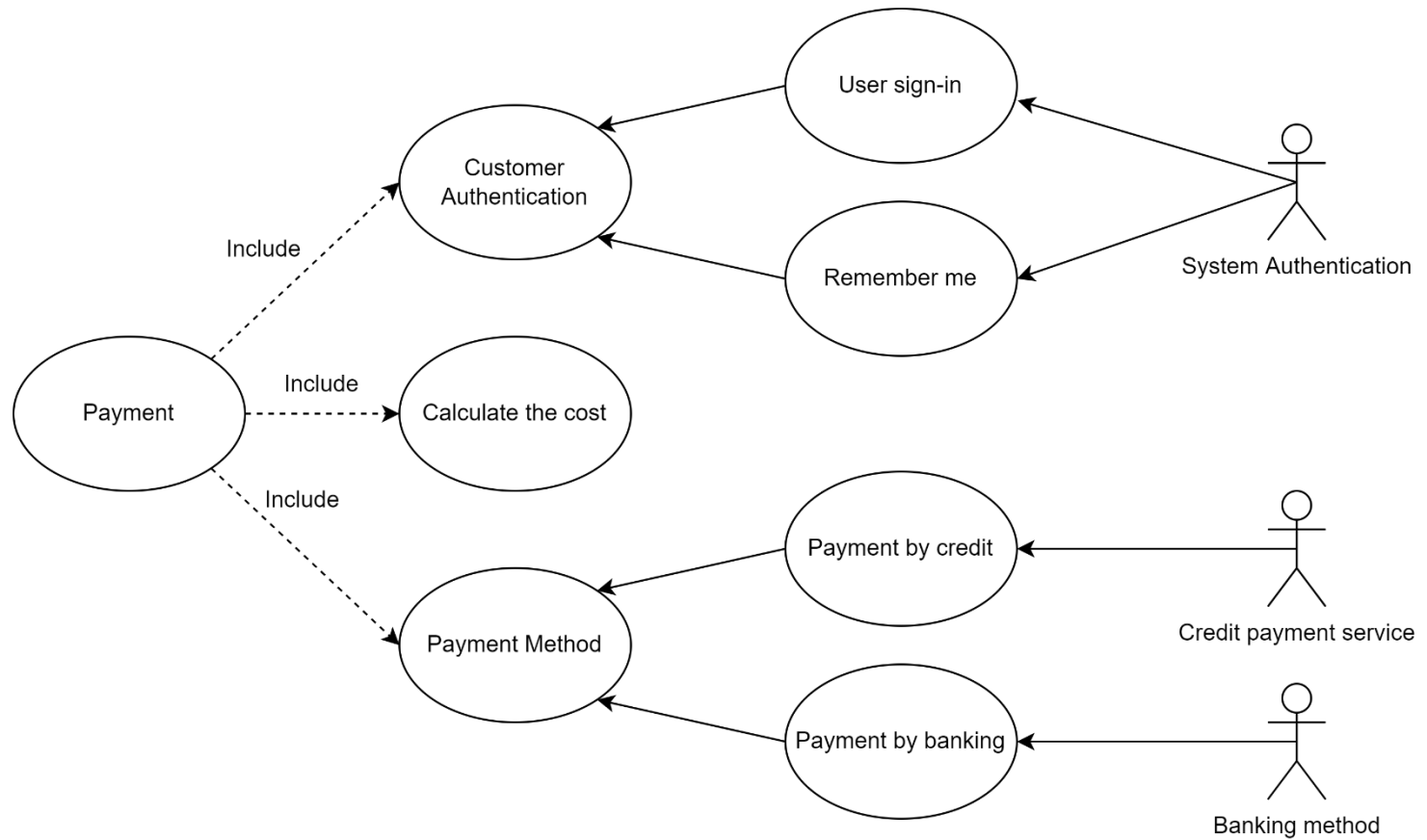
Description	The activities in the software interface are described in this use case.
Actors	
Pre-conditions	<ul style="list-style-type: none"> • The user had downloaded and set up TuneSource Software. • The user had used a legitimate account to access the software.
Post Conditions	<p>When users engage with the software, the system responds to their demands.</p> <p>The system sends and saves the information that users can interact with.</p>
Normal Course	<ol style="list-style-type: none"> 1) Customers access the software's main interface. 2) The system displays music based on the accessible music list and the recommended music list, which is generated by AI and tailored to the user's account after verification. 3) Customers use the system's primary functionalities, such as searching for music, buying music albums, or adding songs to their favorite list. 4) The system displays, processes, and responds appropriately to the customer's view. 5) The system uses an algorithm and an AI program to alter and suggest music that the user will enjoy. 6) The program keeps data. 7) Time step is updated by the system. Repeat steps 5 through 7 using the system. 8) Users cease utilizing all functions. 9) The use case is ended by the system.
Alternative Course	<p>If clients look up an unavailable music CD in stage 3 of the Normal Course</p> <ol style="list-style-type: none"> 1) The system alerts the user and places the user's request on hold. 2) System resets to the original view, and step 2 of the normal course is where the use case proceeds.
Exception	<ol style="list-style-type: none"> a) The software is no longer authorized for usage (step 1 occurs). b) The System provides no results. c) Users are not responded to by the system, or the system reacts to unexpected results. d) Crashing.
Requirements	The user-friendliness of the system is impacted by the GUI for receiving requests.

Explanation Use Case:

In the Examine Items use case, customers can do music searches, browse the catalog, view music that has been recommended to them, buy music, and add songs to their favorite list. All of these use cases are expanding use cases since they give buyers more tools to find the music.

The Customer Authentication use case is addressed because the customer must be approved in order to View Recommended Music and Add to Favorite List. Additionally, with simultaneous user authentication, music could be added to the purchasing list.

2. Payment Use Case



Description	The Tunesource software's payment processes are described in this use case.
Actors	<ul style="list-style-type: none"> • A system that handles permission and authentication. • Service for credit payments. • Banking strategy.

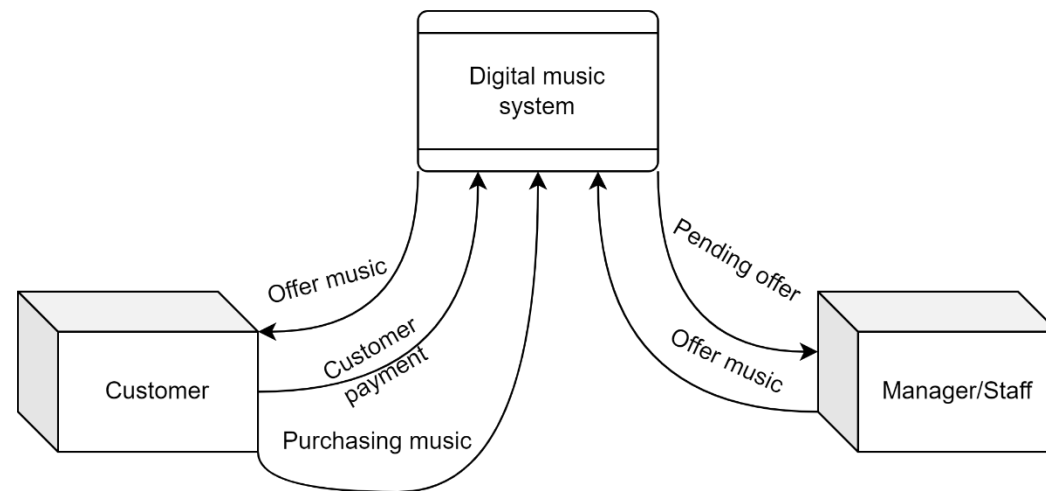
Pre-conditions	<ul style="list-style-type: none"> • The user had used a legitimate account to access the software. • The customer requests certain payment actions (such as recharging or making an unbalanced purchase).
Post conditions	<ul style="list-style-type: none"> • When users use the program to make payments, the system handles their requests for cash. • The system guarantees each transaction session's integrity, correctness, and safety. • The system transmits and retains the payment information.
Normal Course	<ol style="list-style-type: none"> 1) Clients ask for a payment activity. 2) The system automatically verifies the customer's account's authentication. 3) The system offers various payment options, such as credit or banking. 4) After the customer completes payment, the system determines the transaction's fee and tax. 5) The system keeps track of the transaction's data. 6) Time step is updated by the system. Repeat steps 1 through 5 using the system. 7) Users cease utilizing all functions. 8) The use case is ended by the system.
Alternative Course	<p>If any of these two approaches is momentarily unavailable in step 3 of the Normal Course.</p> <ol style="list-style-type: none"> 1) The system notifies the user and places the user's request on hold. 2) System returns to the payment display and prompts users to select a different payment method. 3) If clients concur, the use case moves on to stage 2 of the Normal Course.
Exception	<ol style="list-style-type: none"> a) The software is no longer authorized for usage (step 1 occurs). b) The system offers no means of payment. c) The transaction is delayed by the system. d) Users are not responded to by the system, or it does so in an unanticipated way. e) Crashing.
Requirements	The system's and the customers' accuracy.

Explanation Use Case:

The payment use case includes a number of necessary use cases. Customer authentication is necessary. It may be done using a user login page or a cookie that requests user authentication ("Remember me"). The software authentication service is used in each of these use cases.

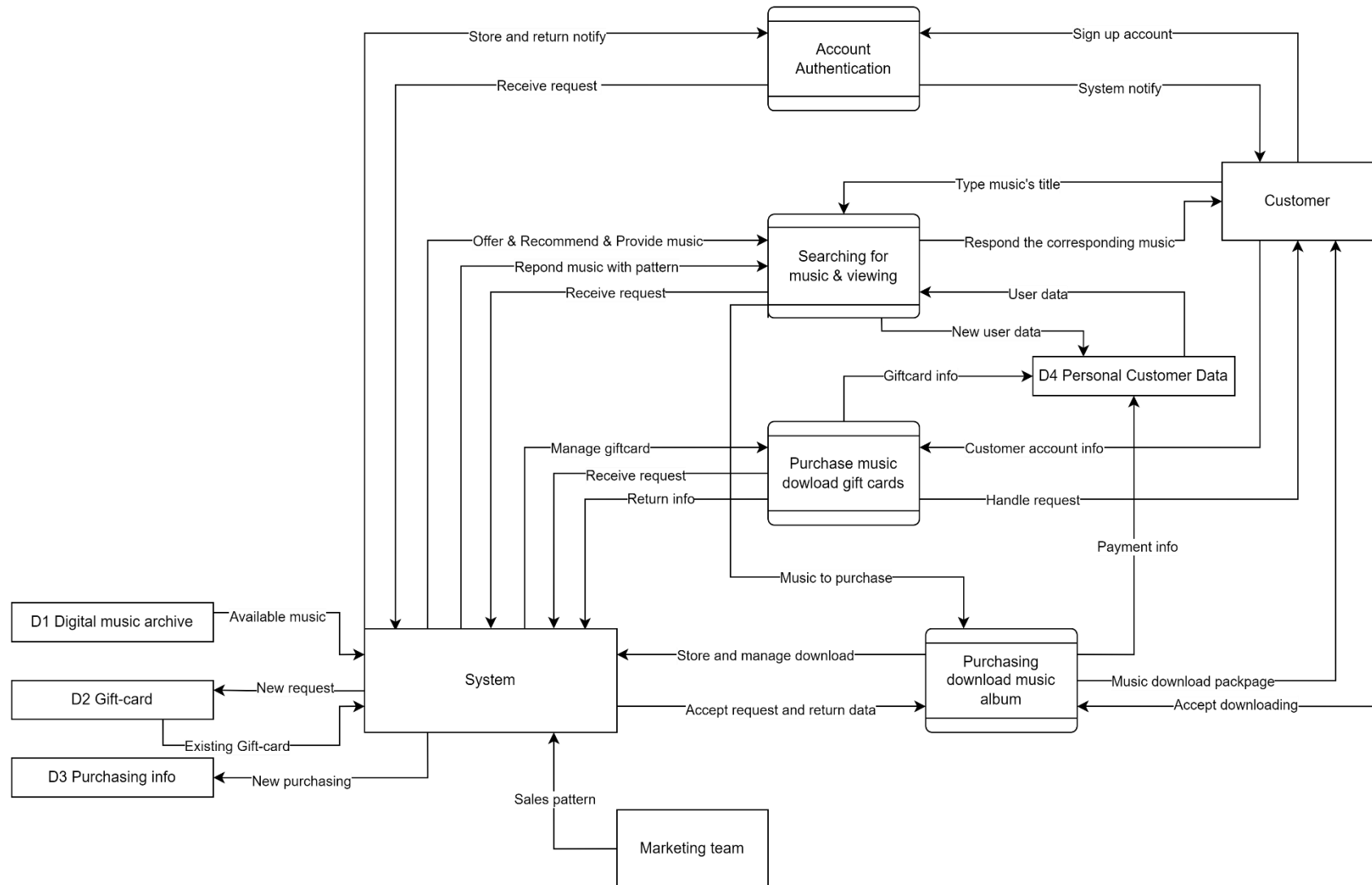
Another sub-use case for payment is payment method, which can be accomplished either through banking or with the use of a credit card and an external credit payment provider.

III. Context Diagram for the whole system



Explanation of Context Diagram: The consumer will receive the music data that the administrator (manager or staff) provides via a digital music system. Thus, the consumer decides to buy music, which turns into a request that is sent to the digital music system together with payment. Some pending offers will be sent to Admin (manager/staff) during the transaction and operation procedure from the digital music system.

IV. Data flow Diagram-Level 0 for the whole system

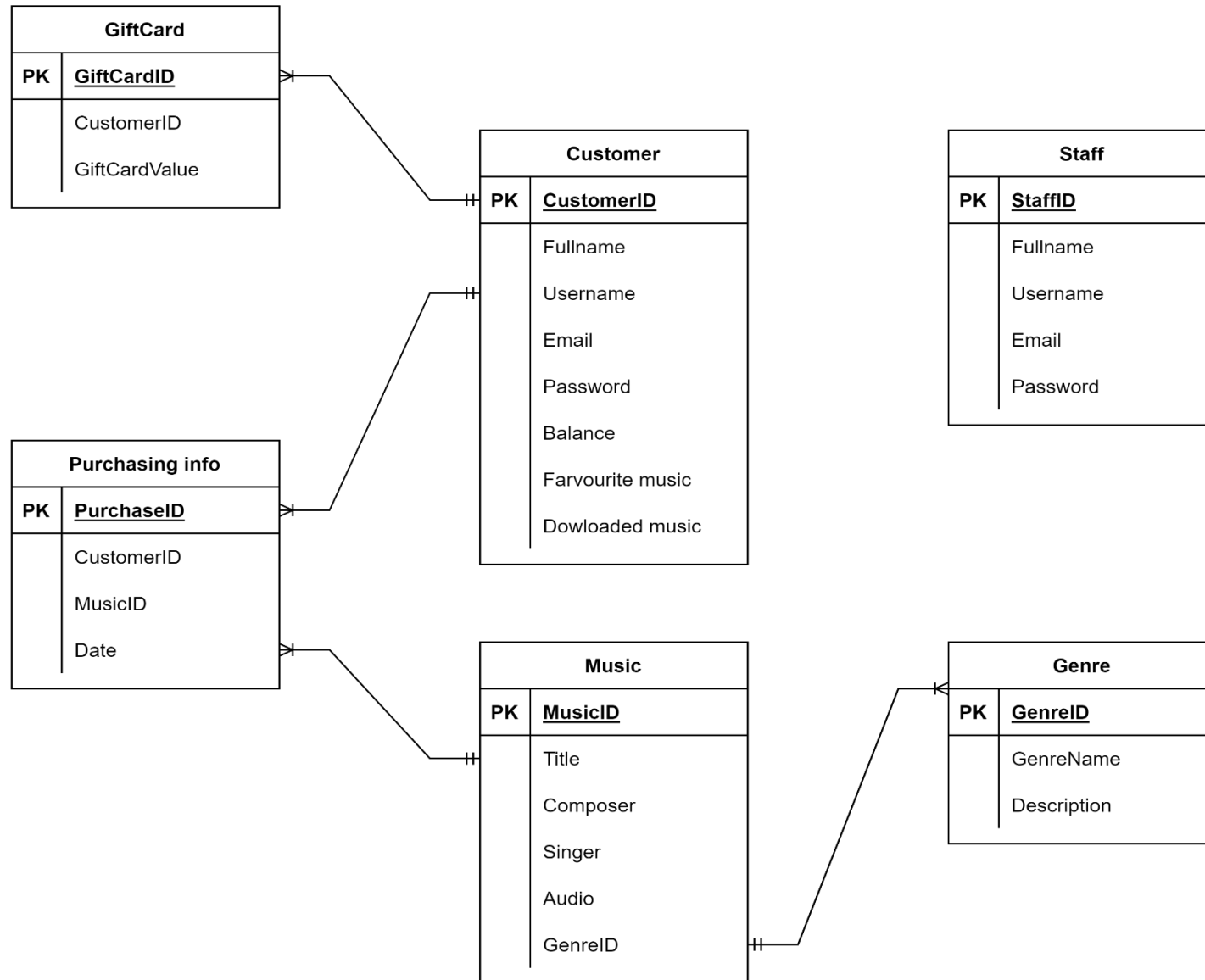


Explanation of Data flow Diagram level 0:

Entire system The DFD (Data Flow Diagram) depicts the general flow of data inside the program. The essential input and output data for TuneSource software's transmission process are stored in 4 data stores with ids ranging from D1 to D4. D1 is a digital music archive that stores and provides music that is available (songs, records, albums); D2 is a gift card that stores gift cards; D3 is purchasing information that stores all information about every software purchase; and D4 is personal customer data that will grow based on the number of customers and is responsible for storing and providing personal information about customers (favorite music, purchasing history, ...).

All of the processes in this DFD are top-level processes that correspond to the software's primary functionalities. They include behaviors related to gift cards, purchase, searching and viewing, and authentication. System and Customers are two top-level external entities that regularly communicate with the high-level process mentioned above. Data flows in DFD level 0 and higher also work around the external entities of the System and the Customer. In section 1, the use-case for the entire system includes a declaration and detailed step-by-step description of the data flows of this DFD. Data flows are explicitly documented with distinct thread arrows in the DFD shown above. The marketing team, which uses sales pattern data in its data flow, is another high-level external entity. This indicates that the sales data is sent to the program by the marketing department, and that the data flow will pass through the "system" external entity to be transmitted to the "finding & viewing music" process.

V. ERD for the whole system



Description: This ERD includes 6 primary entities

- GiftCard Entity and its fields: ID, CustomerID, GCValue
- Genre Entity and its fields: ID, GenreName, Description.
- Customer Entity and its fields: ID, FullName, User name, Password, Email, Balance, Farvourite music, downloaded music.
- Music Entity and its fields: ID, Title, Composer, Singer, Audio, GenreID.
- Staff Entity and its fields: ID, FullName, User name, Password, Email.
- PurchasingInfo Entity and its fields: ID, CustomerID, MusicID, Date.

Explanation:

- A customer may possess zero or several gift cards.
- One client may make numerous purchases, but only one consumer may be present on any given purchase.
- One music can only be purchased once, however one music can be bought multiple times.
- One song can comprise multiple genres, but only one genre per music.

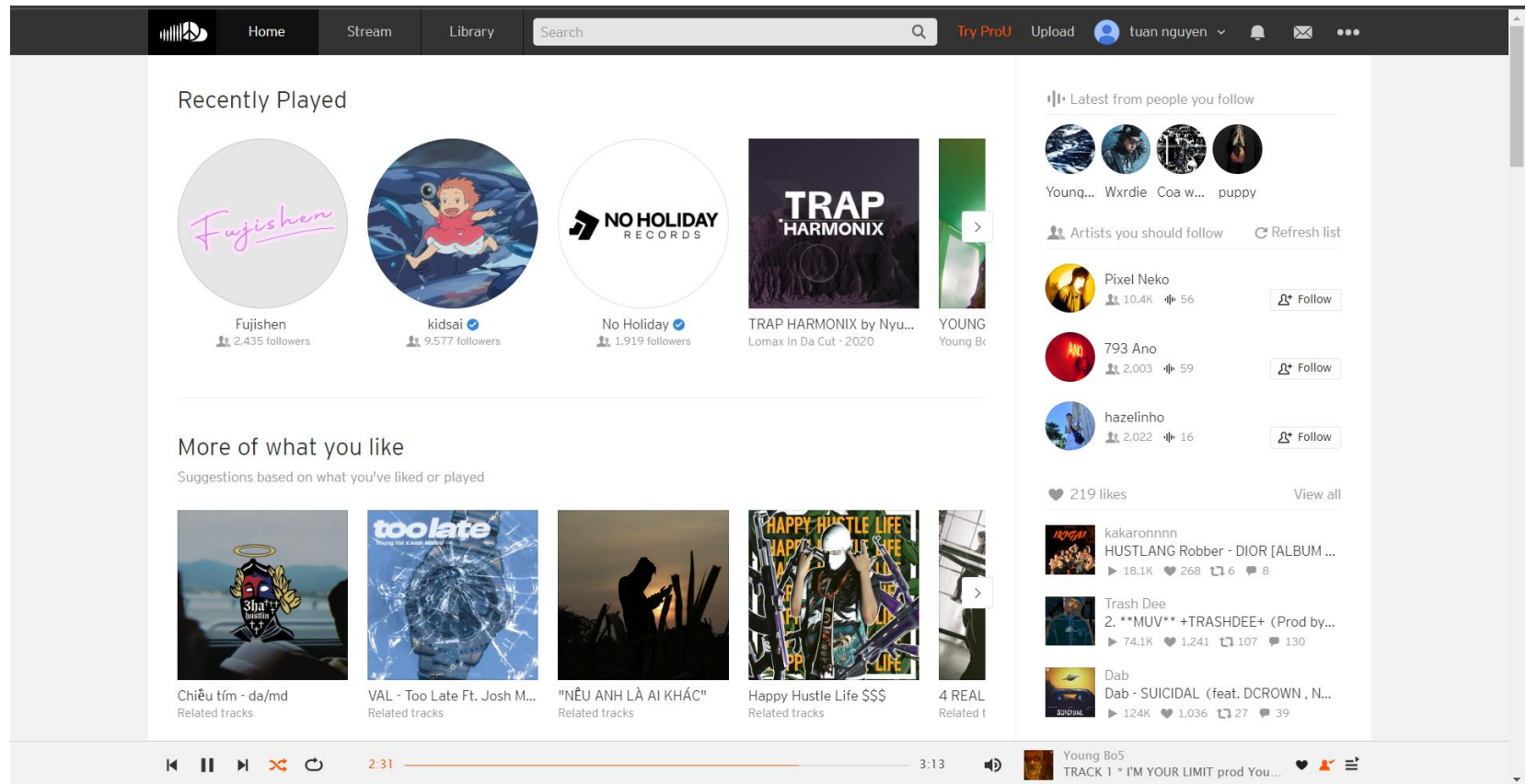
VI. Pseudo Code For One Module Of The Program

```
//consider the following pseudocode fragment:
begin
invalid input_password := true
invalid_input_email := true
logged_status := false
while (logged_status = false) do
begin
print("Username: ") //input Username
scan(inputted_username)
print("Password; *") //input Password
scan(inputted_password)
//begin matching process if inputted_username is in Account_database then
//Match inputted login details with each existing valid set of Login details from the Login_database
valid_input_username := true
if
inputted_password is in Account_database then
valid_input_password := true //Username and Password are both matched
login_status := true
print("Logins have been validated")
else
valid_input_password := false // If username was matched but password was not matched
logged_status := false
print( "Error")
end if else
valid_input_username := false // If username wasn't matched.
logged_status := false
print("Error")
end if
end while
```

TASK 3 – DESIGN

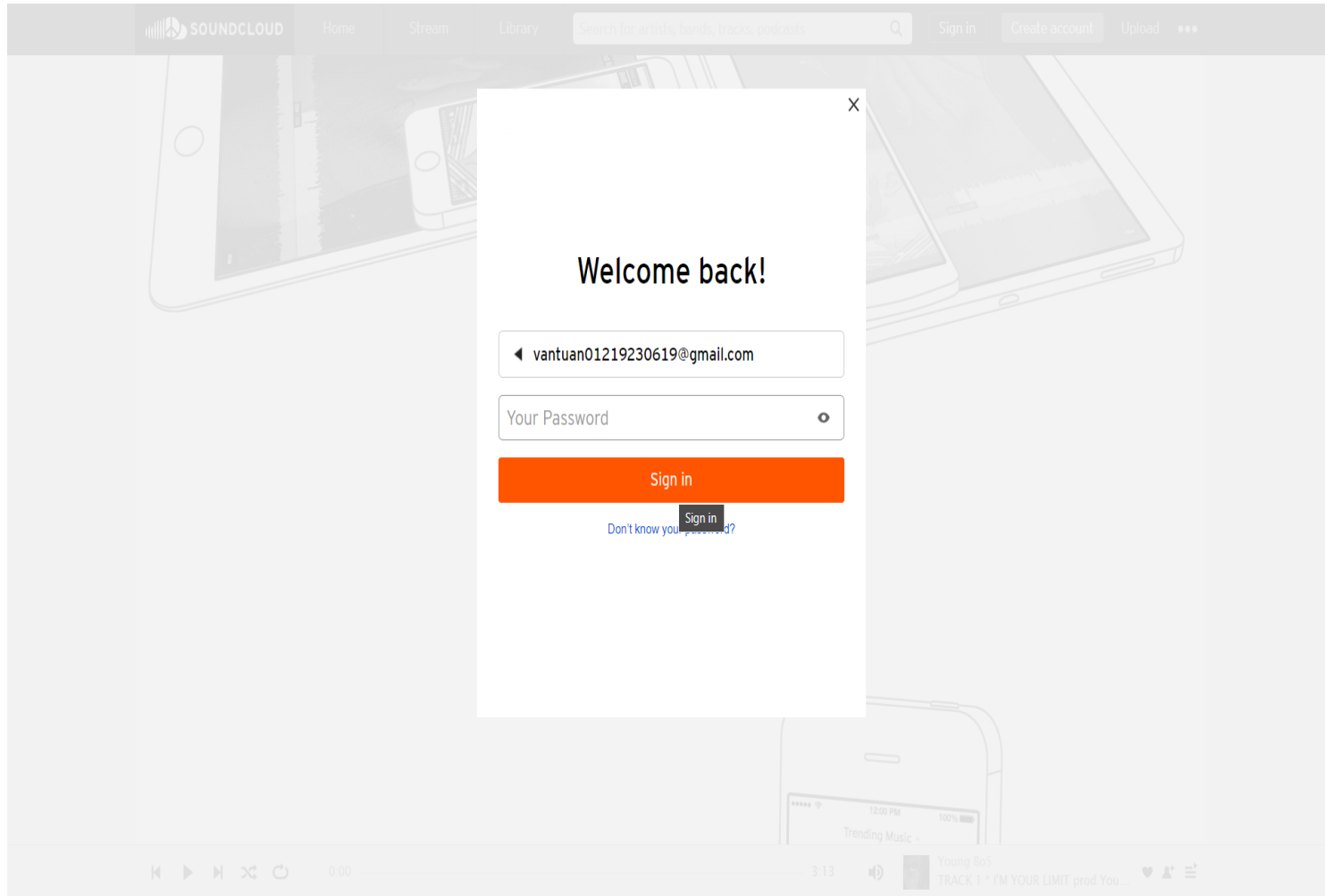
C. Explain how user and software requirements have been addressed.

- I. Discuss how the user and software requirements are addressed in the design phase**
 - 1. Explain how Mock-up, and Wireframe are used in the project**
 - a. Maintain Interface Wireframe**



Some of the fundamental capabilities that satisfy the standards are included in the primary interface, such as playing a music sample, buying music, downloading albums, and searching for music or albums.

b. Login Wireframe



The image shows a SoundCloud login modal window. At the top, it says "Welcome back!". Below this, there is an email input field containing "vantuan01219230619@gmail.com". Underneath the email field is a password input field labeled "Your Password" with a toggle icon. A prominent orange "Sign in" button is positioned below the password field. At the bottom of the modal, there is a link that says "Don't know your password?" with a "Sign in" button overlaid on it. The background of the page shows the SoundCloud navigation bar with links for Home, Stream, and Library, along with a search bar and options to Sign in, Create account, and Upload. The bottom of the page features a music player interface with playback controls, a progress bar at 0:00, and a track titled "Young Bo's TRACK 1 * I'M YOUR LIMIT prod You...".

The Email and Password fields in this frame are sufficient to ensure the Login authentication process. In case users forget their password, it also features a "forgot" feature.

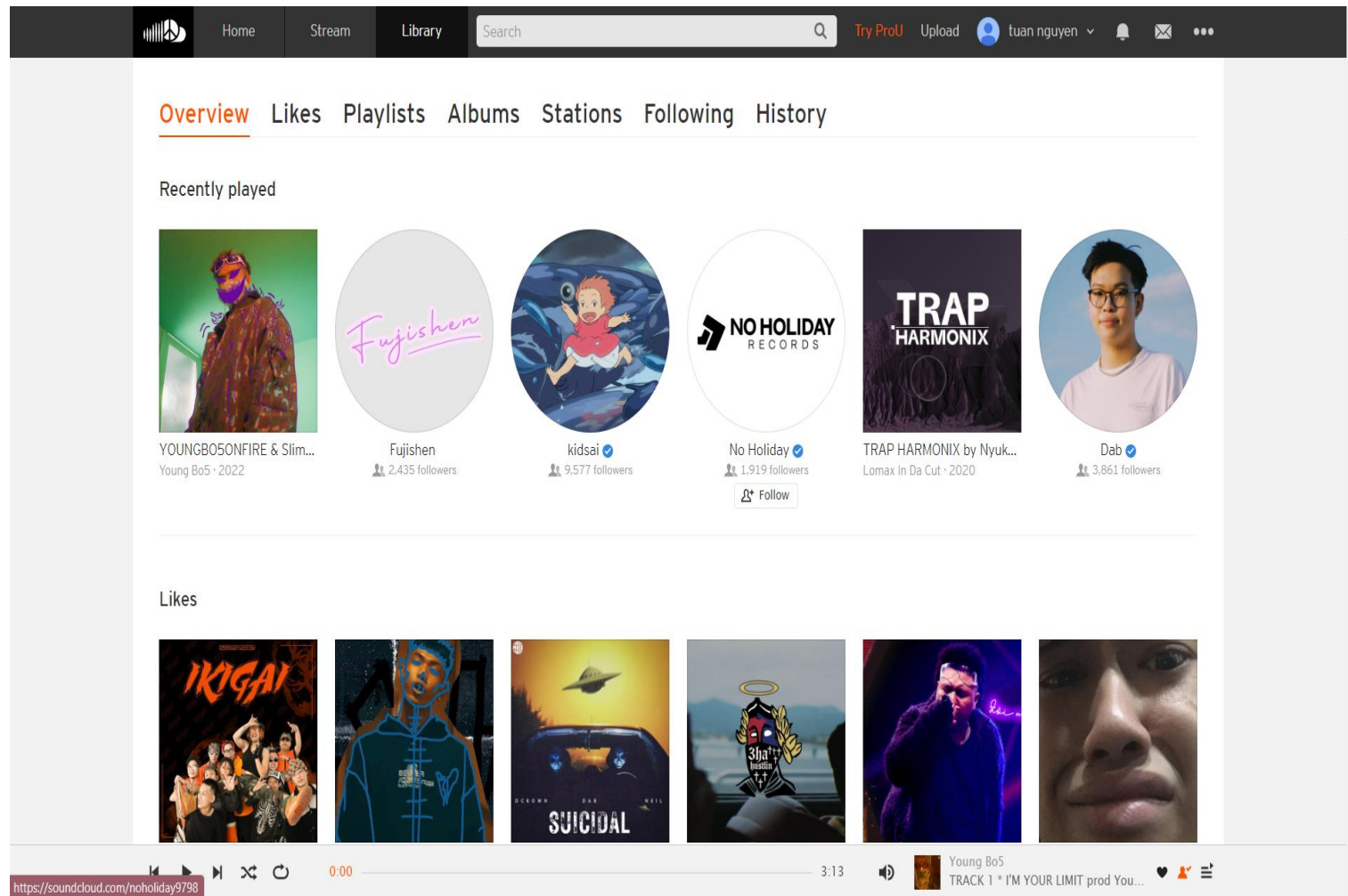
c. Monthly fee Wireframe

The wireframe displays three subscription plans for SoundCloud, each in a separate card. The cards are arranged horizontally. The first card is for 'SoundCloud Basic', which is free. The second card is for 'SoundCloud Pro Unlimited', priced at £137,500/month when billed yearly. The third card is for 'Repost by SoundCloud', priced at \$2.50/month when billed yearly. Each card includes a list of features and a call-to-action button. The 'Basic' card has a 'You're on Basic' button, while the 'Pro Unlimited' and 'Repost' cards have 'Get Pro Unlimited' and 'Get Repost' buttons respectively. The 'Pro Unlimited' card also includes a note about being billed yearly.

Plan	Price	Features	Call to Action
SoundCloud Basic	Free	<ul style="list-style-type: none">✓ Upload up to 3 hours of your work✓ Get lossless HD storage✓ Access basic insights	You're on Basic
SoundCloud Pro Unlimited	£137,500/month (when billed yearly)	<ul style="list-style-type: none">✓ Everything in Basic & Repost, plus:✓ Unlock unlimited upload time**✓ Get paid fairly for your plays✓ Access advanced audience insights✓ Replace your track without losing its stats✓ Pin your favorite tracks	Get Pro Unlimited
Repost by SoundCloud	\$2.50/month (billed yearly)	<ul style="list-style-type: none">✓ Distribute tracks to all major music services✓ Extend your reach with promotional tools✓ Get premium services to help you get paid✓ Split payments with your collaborators	Get Repost

This frame satisfies the functional requirement that the program must be able to create a customer subscription account that allows for limitless downloads in exchange for a monthly price. There are three premium plans with the anticipated charge so that clients can choose or act reasonably. Each premium plan is also created to accommodate various customer objects.

d. User Wireframe



User wireframe where music are recorded and kept here that users interact with or modify. Both downloaded tracks and favorite songs are included. The application that determines the statistics data on the music genres that customers frequently prefer is also available here. Additionally, artist will show up in the archive after being followed by a client.

e. Payment Wireframe



Get SoundCloud Pro Unlimited

1. Select billing cycle



Yearly billing

₫1,650,000, that's ₫137,500/month

25% YEARLY DISCOUNT



Monthly billing

₫183,500/month

2. Add payment details



Card



PayPal



3. Review your purchase



SoundCloud Pro Unlimited

Total

₫1,650,000

Billing cycle

Yearly

Subscription will automatically renew at ₫1,650,000 every year, starting Aug 20, 2023, unless you cancel before the day of your next renewal in your subscription settings.

All prices in VND

Buy subscription

By submitting your payment information and clicking Buy subscription you agree to the [SoundCloud Pro Terms of Use](#)

Signed in as tuan nguyen. [Sign out](#)

[Legal](#) - [Privacy](#) - [Cookies](#) - [Consent Manager](#) - [Imprint](#) - [Help Center](#)

Language: English (US) ▾

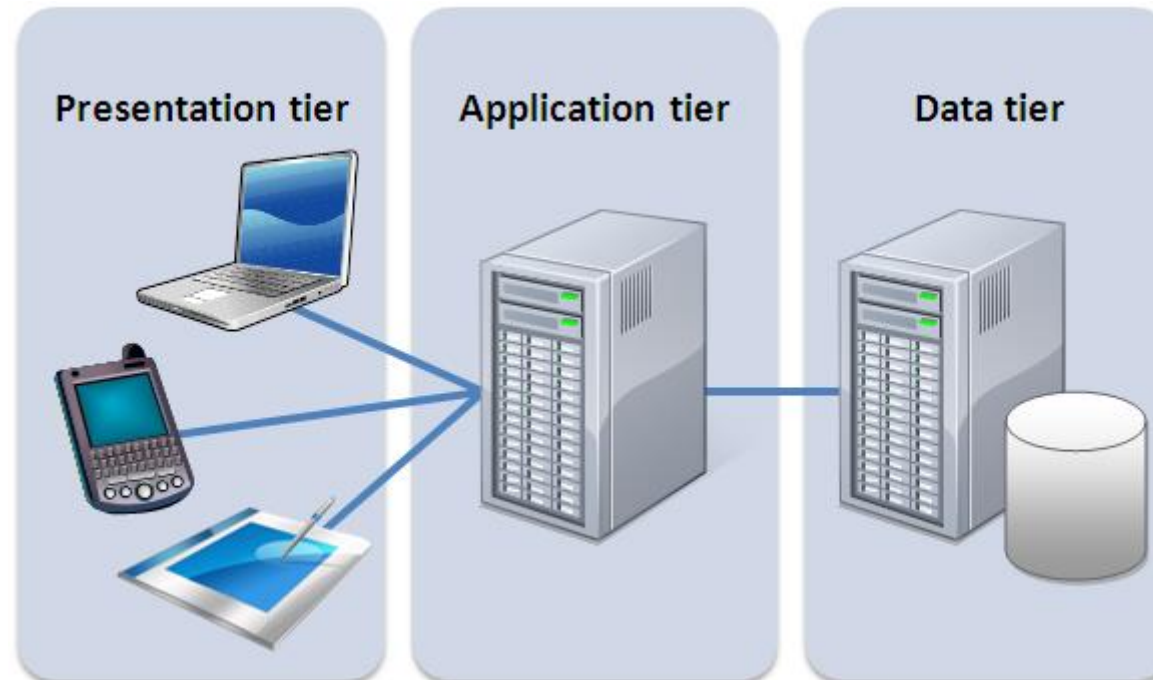
Payment is absolutely necessary in order to achieve this software's primary objectives of selling music and subscription accounts. As a result, the payment structure presented above includes two payment options: credit/debit card and banking method.

2. Explain which architecture (client-server, n-tier, microservices, etc.) is suitable for the project with clear illustrations and why.

Architectures such client-server, n-tier, microservices, etc. must be taken into account during the software development process in order to design this software.

Client-server	N-tier	Microservices	Three-tiered Architecture
✗	✗	✗	✓

Three-tiered architecture was found during the evaluation process to be the best choice for this project.

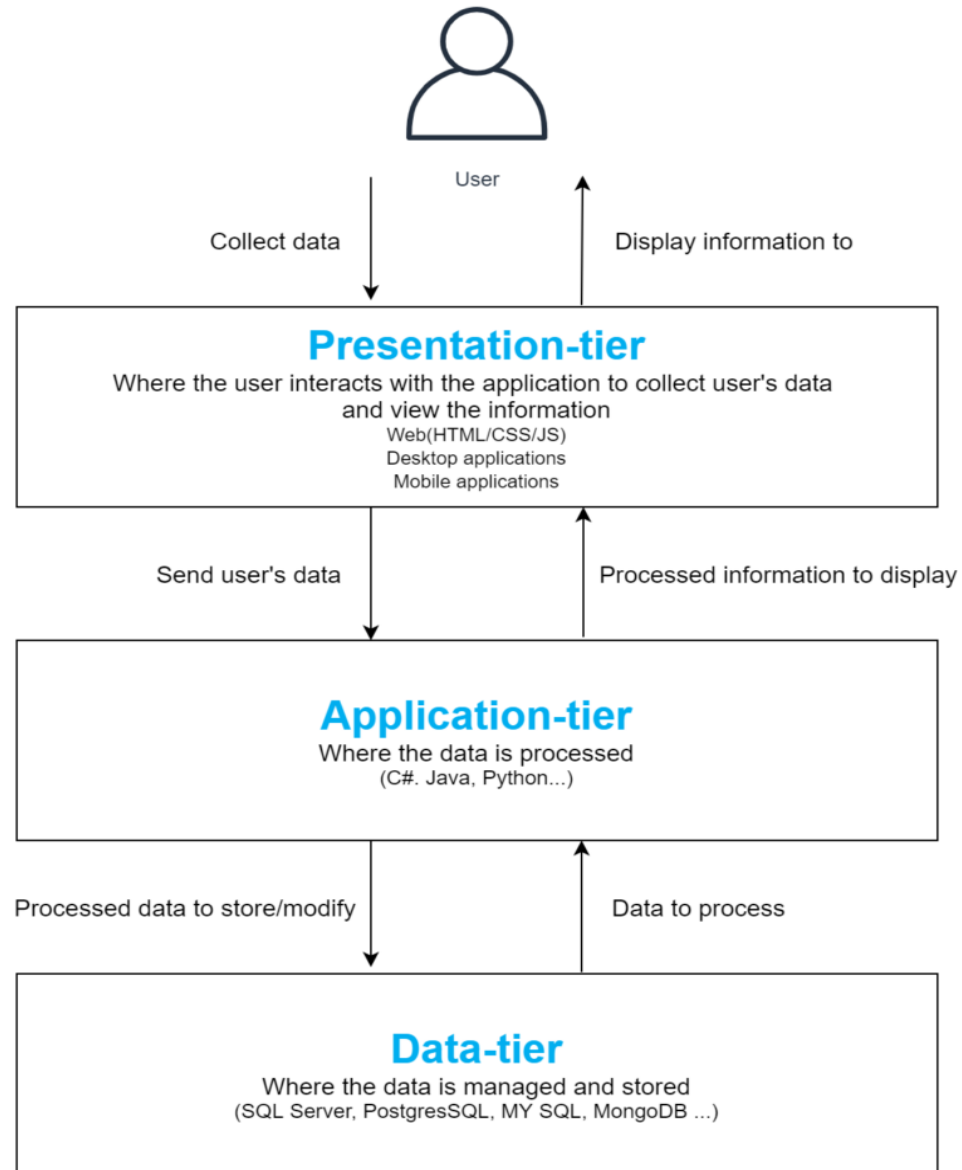


The presentation tier, or user interface, the application tier, where data is processed, and the data tier, where the application's associated data is stored and managed, are the three logical and physical computer tiers that make up the well-known three-tier architecture. (Education, 2020)

For more detail here are descriptions of three-tier:

- **Presentation Tier:** The user interface and communication layer of the application, or presentation tier, is where end users engage with the software. Information display to and data collection from the user are its primary goals. This top-level tier can function, for instance, on a desktop application, a web browser, or a graphical user interface (GUI). Tiers of web presentation are often created with HTML, CSS, and JavaScript. Depending on the platform, a variety of languages can be used to create desktop apps. (Education, 2020)
- **Application Tier:** The core of the application is the application tier, sometimes referred to as the logic tier or middle tier. In this layer, data gathered in the presentation tier is processed using business logic, or a particular set of business rules, sometimes in comparison to data gathered in the data tier. Data in the data tier may also be added, removed, or modified by the application tier. The application layer often uses Python, Java, Perl, PHP, or Ruby for development and uses API calls to talk to the data tier. (Education, 2020)
- **Data tier:** The information processed by the application is stored and handled in the data tier, which is sometimes referred to as the back-end, database tier, or data access tier. This could be a NoSQL database server like Cassandra, CouchDB, or MongoDB, or a relational database management system like PostgreSQL, MySQL, MariaDB, Oracle, DB2, Informix, or Microsoft SQL Server. All communication in a three-tier application passes through the application tier. Direct communication between the presentation tier and the data tier is not possible. (Education, 2020)

How to Achieve the true Three-tier architecture



3. Explanation of why the chosen three-tiered is suitable for this project

Based on this specific problem, it is clear that the program needs to be built quickly: "Because clients have a variety of music download options available to them elsewhere, we need to bring this system to the market as quickly as possible." Fortunately, a three-tier architecture allows for simultaneous development of each layer by several teams, allowing a business to release software more quickly and enabling programmers to use the most modern languages and tools for each tier.

Scalability can be improved using a three-tier architecture, which is also one of the project's non-functional requirements. Each layer can be scaled independently thanks to the separation of the layers based on the demand at any given moment. For instance, the software server scaling is achievable without affecting the program's servers if the number of requests is high at our end but few of them have an impact on the application layer.

Additionally, the three-tier design guarantees the independence and dependability of the underlying servers or services. The three-tier architecture also satisfies the project's primary goal, which is to boost digital music sales revenue. It facilitates simple code base maintenance, business logic administration, and individual code display.

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

	Standard client	Standard Webserver	Standard App Server	Standard DB Server
Operating system	Cross-platform	Linux	Linux	Linux
Special Software	Realtek(R) Audio	Made by ExpressJS	Nodejs	MongoDB
Hardware	250-500 GB disk drive Intel-core i5- 7400 professor 24inch IPS monitor	1 TB disk drive Intel Xeon E5- 2680 12 core	250-500 GB disk drive Intel Xeon W-1250p 6 core	250-500 GB disk drive
Network	4G/Wi-fi	Dual 100 Mbps Internet	Dual 100 Mbps Ethernet	Dual 100 Mbps Ethernet

Explanation:

The client's operating system will be cross-platform to condense time and effort savings. It is feasible to deploy a single script rather than creating unique scripts for each platform. Because of the shorter time to market and shorter development time, everyone benefits, including the development team.

Linux is more suited than the others for developing web servers, application servers, and database servers. This is due to Linux OS's high level of dependability, security, and software development efficiency. This OS is also so light that my company's programmers can use it to run scripting languages.

The MEAN (MongoDB, ExpressJS, Angular, and Node.js) technical stack is responsible for this project's web-app-DB server-specific software. The costs associated with this project are reasonable, and the MEAN technology stack is quite cost-efficient. The MEAN stack simply needs a fundamental knowledge of JavaScript to function. Among other technologies, expertise of PHP, JavaScript, and MySQL is required by other stacks. MEAN requires fewer developers to work on this project than other frameworks do. The price of creating TuneSource software will be much reduced as a result. Furthermore, it is impossible to overestimate the benefits that this feature alone offers. As a result, my development team's working environment is enhanced, ensuring that we generate software that appeals to customers by satisfying both their functional and non-functional needs. A single programming language also guarantees that the backend response unit will be able to quickly and efficiently handle client requests as the application grows. Additionally, utilizing Node.js speeds up project completion.

At the moment, hardware is specified in accordance with technique and devices.

The webserver's network will use dual 100 Mbps Internet connections for a stable connection since they offer greater communication between multiple physical locations rather than just speed and security. The app server and DB server will, however, use an Ethernet network. This is because we need the most rapid upload and download speeds, the lowest feasible latency, and the highest level of security to guarantee the protection of the data.

TASK 4 – SOFTWARE QUALITY MANAGEMENT

I. Discuss two software quality attributes that are applicable to the project

There are two software quality attributes in this project that are applicable to it.

Maintainability

The several product versions should be easy to manage. It should be easy to add code to an existing system and upgrade it for new features and technologies as they become available for development.

It should be cheap and easy to maintain. It is easy to maintain and modify the system by fixing bugs or updating the software.

Usability

The utility of this can be evaluated. It should be easy to use the program. It ought to be straightforward to comprehend. It need to be simple to navigate. The following should apply to the Tune Source project:

- Simple to operate, to prepare input, and to understand output.
- Make sure that the standards and practices for our user interfaces are compatible with those of our other commonly used systems.
- It is easy for new or infrequent users to pick up using the system.

II. Discuss two quality assurance techniques that can help improve the software quality in the project

1. Plan and record your testing process

Writing testing documentation should be the first step in the quality control process. This documentation frequently includes a list of the things that need testing. The document also includes a list of testing equipment, settings, and acceptance criteria that determine whether the finished product meets the customer's original objectives. (Clockwise, 2022)

The QA engineer can get ready for the testing procedure with proper planning. When certain testing equipment is discovered to be missing at the beginning of a test, the test is considered to have failed. In that case, you'll have to wait till the necessary gadget is delivered (which can take several days). Meanwhile, the quantity of experimental features and bugs increases. (Clockwise, 2022)

The QA engineer can afterwards review the tests he's completed and keep control of the procedure. For new team members, testing documentation is also highly beneficial. They can enter the quality control process right away. (Clockwise, 2022)

2. Test-driven development

One of the core tenets of extreme programming is test-driven development, a form of agile software development process. Extreme programming prioritizes frequent releases with short development cycles, thorough code review (typically accomplished through pair programming), and unit tests that span the whole code base in order to provide the highest quality output. (Clockwise, 2022)

A test that covers the required change must first be written before functionality can be added or changed. Code is then written to pass the first test after that. The act of a developer writing tests for his own code is regarded as excellent practice. By doing that, he may comprehend the specifications for a new functionality and create it so that it will pass the test. (Clockwise, 2022)

3. Discuss how the design techniques and approaches that you have to used can help improve the software quality

Hiring a website hosting

The place where website content is kept online is called hosting. Every time a consumer types the website's domain name into their web browser, the website's server will fetch the material so that customers may quickly utilize and access the website. The quantity of website visitors determines the hosting capacity. The ideal option for maintaining the Tune Source project website, which serves a large number of clients, is dedicated hosting because it is a private server for the business and does not need to be shared. However, renting this server rather than buying it will save money.

CONCLUSION

The components of the software development life cycle have been covered in detail above, with a particular emphasis on carrying out the lifecycle and discussing whether software behavioral design methodologies are appropriate for the Tune Source Project. It is clear from the software's document analysis and design how the software development lifecycle method encourages the creation of high-quality, secure software throughout the whole development process. Additionally, it gave me the information and abilities I required to comprehend software development lifecycles and build one with the right approach.

References

Clockwise, 2022. *Clockwise software*. [Online]

Available at: <https://clockwise.software/blog/quality-assurance-methods-and-best-practices-qa-definition/>

[Accessed 22 08 2022].

Education, I. C., 2020. *IBM*. [Online]

Available at: <https://www.ibm.com/cloud/learn/three-tier-architecture#toc-the-three--iwvUcK4b>

[Accessed 20 08 2022].

Johnston, K., 2020. *CHRON*. [Online]

Available at: <https://smallbusiness.chron.com/stakeholders-objectives-organization-64385.html>

[Accessed 15 08 2022].

Martin, M., 2022. *Guru99*. [Online]

Available at: <https://www.guru99.com/functional-requirement-specification-example.html>

[Accessed 15 08 2022].

