ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 9: Software Development Life Cycle		
Submission date	9/3/2020	**Date Received 1st submission**	
Re-submission Date	17/3/2020	**Date Received 2nd submission**	
Student Name	LE ANH HIEU	**Student ID**	GCH18106
Class	GCH0716	**Assessor name**	LAI MINH DUC
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	LE ANH HIEU

Grading grid

P5	P6	P7	M3	M4	M5	D3	D4
✓	✓	✓					

❒ Summative Feedback:	☑ Resubmission Feedback:
2.1	

Grade:	Assessor Signature:	Date:

Internal Verifier's Comments:

Signature & Date:

Table of Contents

I. Introduction:

Today, the design of a product or project cannot be without the Software Development Life Cycle (SDLC). Like other manufacturing industries, SDLC is one of the first and extremely important factors that bring success to software developers, it helps every project member from the old to the new, inside or outside the company can simultaneously handle the work corresponding to their position through the common way of the company. It can be said that the software development process is crucial to creating a product with low cost and high productivity.

Figure: SDLC Model

II. Analysis (1):

1: Requirement definition

1.1: Stakeholder:

According Katharine, et al. (2005, p.8), *"Stakeholder: Any group or individual who can affect, or is affected by, an organisation or its activities. Also, any individual or group that can help define value propositions for the organisation."*

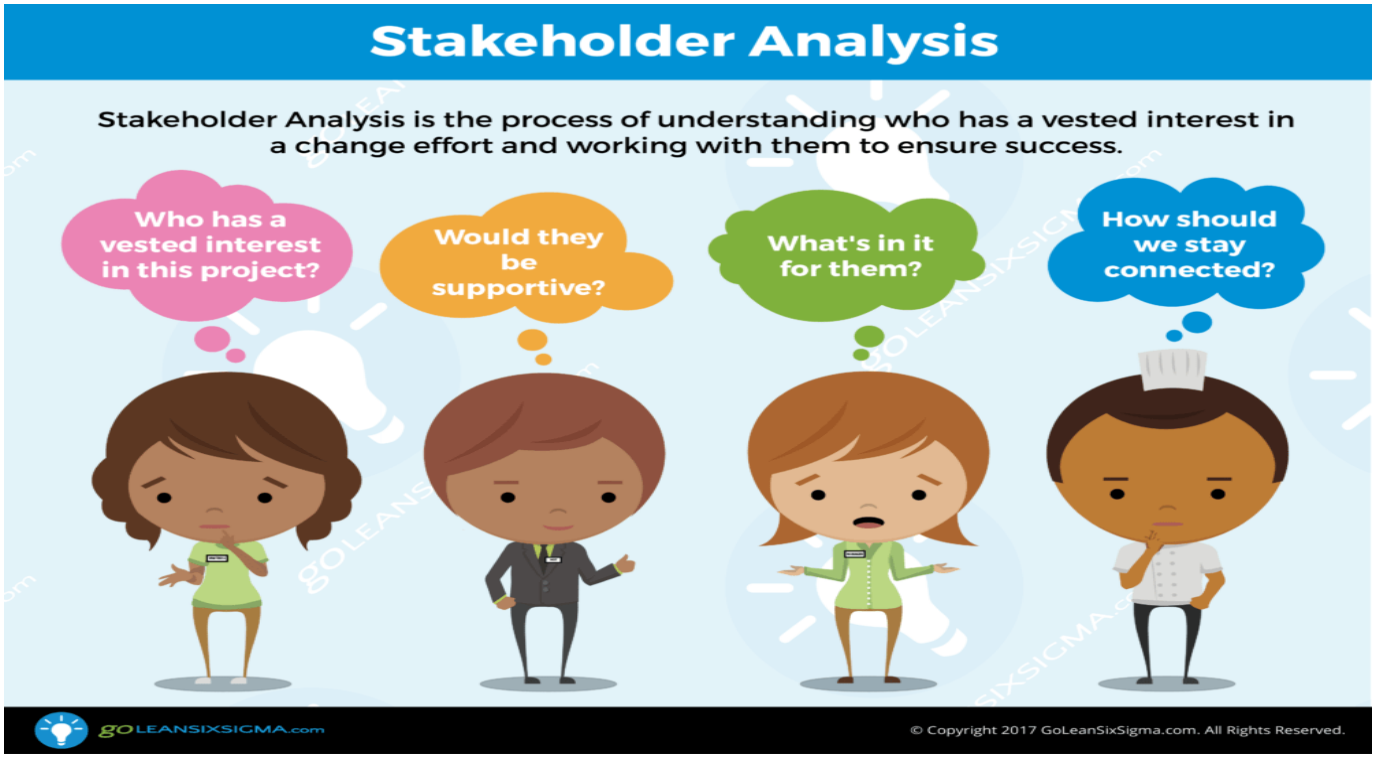


Figure: Stakeholder

In a very understandable way, "Stakeholders" are those who are more or less, having direct or indirect effects, affecting or affecting projects, programs and software.

For example:

- Key stakeholders (champion, management, users)
- Others (IT staff, external entities,...)

The role of the Stakeholders:

- Create project charter and declare project scope

- Develop a project management plan
- Approve project changes
- Identify constraints and assumptions
- Determine the requirement
- Risk management

Stakeholders have different and often inconsistent needs. Inconsistencies may not be obvious at first, they only emerge after deeper analysis.

In Tune Source's project, the stakeholders include:

- Project champion: Carly Edwards, Assistant Vice President, Marketing, high-level executive, provides support (time and resources).
- Supplier: ABC company. Providing software, software packages, personnel implementing the project.
- Organisational management: Tune Source and ABC employees and members, contribute, support and promote, encourage people to use the software.
- System users: Customers of Tune Source, should involve early and throughout the development.

1.2: Requirement:

(Harwell et al. 1993) defining requirements is "*If it mandates that something must be accomplished, transformed, produced, or provided, it is a requirement.*"

Figure: Requirements

In other words, requirement is a statement of:

- What the system must do or.
- What characteristics it needs to have.

Main types of requirement: each takes a different viewpoint:

- **Business requirement**: Stated in the system request. For example: Tune Source wants to develop a music download website project to increase digital music sales and CD sales online.
- **User requirement**: What user needs to do (tasks). For example: Tune Source customers expect Tune Source software to be able to download digital music.
- **System requirements**: What the software does for the user's needs. For example: The company's Tune Source website allows users to download music.

In particular, the system requirement has 2 types:

Functional requirement (FR): Statements of services (or functions) that the system should provide (to its users).

Expressed at different levels of detail:

- High-level (lacks specific details on input and output)
- Detailed (has specific details on input and output).

There are two types:

- Process oriented: describes processes (functions)
- Information oriented: describes information and data related to input and output.

Two important qualities:

- Complete: covers all the user needs.
- Consistent: contains no contradiction.

Non-functional requirement (NFR): Constraints on the services (or functions) offered by the system.

Types of NFR:

- Product requirement: usability, performance, security, dependability,...
- Organisational requirement: budget, policy,...
- External requirement: safety or privacy regulations,...

Are often critical to the success of the software as a whole: usually affect more than one functions.

NFR should be expressed in a 'testable' form: constraints should be as specific (quantifiable) as possible.

1.3: Requirement definition of the project:

Functional Requirements:

1. Search and Browse

    1.1 The system will allow customers to browse music choices by predefi ned categories.

    1.2 The system will allow customers to search for music choices by title, artist, and genre.

    1.3 The system will allow customers to listen to a short sample of a music selection.
    1.4 The system will enable the customer to add music selections to a “favorites” list.
2. Purchase
    2.1 The system will enable the customer to create a customer account (if desired) that will store customer data and payment information.
    2.2 The system will enable the customer to specify the music choice for download.
    2.3 The system will collect and verify payment information. Once payment is verifi ed, the music selection download process will begin.
3. Play music:
    3.1 The system will allow users to play and create playlists of purchased tracks.
    3.2 The system will allow users to rewind music, display lyrics.
4. Account:

    4.1 The system will allow storage of personal information of users when they register for an account.

    4.2 The system will have a system of accumulating points when buying music for discounts or redeeming gifts.

5. Promote

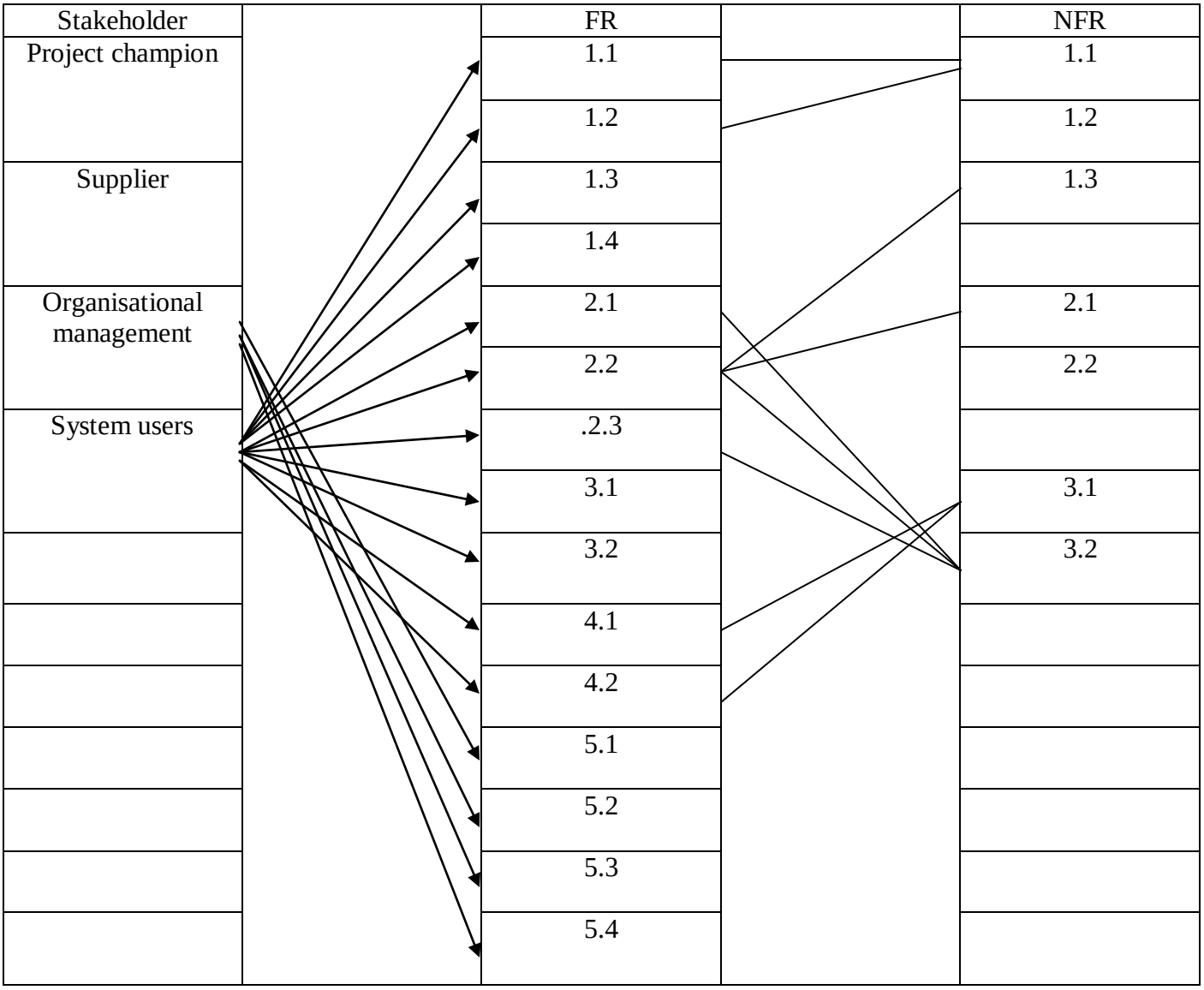
    5.1 The system will keep track of the customer’s interests on the basis of samples selected for listening and will use this information to promote music selections during future visits to the web site.
    5.2 Marketing department can create promotions and specials on the web site.
    5.3 Based on customer’s previous purchases, music choices can be targeted to the customer on future visits to the web site.
    5.4 On the basis of customer interests, customers can be notifi ed of special offers on CDs that can be purchased at the regular Tune Source web site or in a Tune Source store.

Nonfunctional Requirements:

1. Operational
    1.1 The digital music database will be constructed to facilitate searches by title, artist, and genre.
    1.2 The system will run on any Web browser and on in-store kiosks.
    1.3 In the event of a failure during a download, the customer will be able to restart the download.

2. Performance
   2.1 Download speeds will be monitored and kept at an acceptable level.
   2.2 Website performance will work well with both computers and phones.
3. Security
   3.1 Customer information will be secured.
   3.2 Payment information will be encrypted and secured.
4. Cultural and political: No special cultural and political requirements are expected.

1.4: Relationship between FRs and NFRs:



2. Requirement elicitation techniques:

According Ivan et al, (2018), *"Requirements elicitation is a critical activity that forms part of the Requirements Engineering process because it has to discover what the software must do through a solid understanding of the wishes and needs of the various stakeholders and to transform them into software requirements."*

Figure: Requirement Elicitation

Common techniques:

- Interview
- JAD
- Questionnaire
- Document analysis
- Observation

2.1: Interview:

A natural (and most common) technique. Usually one-on-one, but can be with many interviewees.

Steps:

- Select interviewees: who, where and when.
- Design questions: close-ended, open-ended, probing questions.
- Prepare for interview: a general interview plan (lists the questions to ask in suitable order).
- Conduct the interview: carefully record information.
- Follow up: prepare interview report.

Advantages:

- Improving, adding more new information.
- Depth of information is high.
- User involvement is high.

Disadvantages:

- Breadth of information is low.
- Integration of information is low.
- High cost.

2.2: Joint Application Development (JAD):

A team-based, information gathering technique to identify requirements.

Format:

- 10–20 users (system users + management)
- Facilitator: sets the meeting agenda, guides the discussion (does not join, remains neutral)
- 1-2 scribes: assist by recording notes, making copies,...
- Work together until necessary information has been collected
- CASE tools are often used to record information

Advantages:

- Improving, adding more new information.
- Depth of information is high.
- Integration of information is high.
- Low-medium cost.
- User involvement is high.

Disadvantages:

- Breadth of information is medium.

2.3: Questionnaire:

Questionnaire: a set of written questions for obtaining information from individuals. Today, most questionnaires are being distributed in electronic form: e-mail or WWW (e.g. Google's forms).

Suitable when:

- There is a large number of stakeholders from which to get Information.
- Stakeholders outside of the organisation (e.g., by customers or vendors).

Advantages:

- Improving, adding more new information.
- Depth of information is medium.
- Breadth of information is high.
- Low cost.

Disadvantages:

- Integration of information is low.
- User involvement is low.

2.4: Document analysis:

Document analysis is used to understand the current (as-is) system. For example: Review the system documents written by the previous project team.

Two types of document:

- Formal: forms, reports, policy manuals, organisation charts.
- Informal: created by users to add additional information.

Signs of the need for system improvement:

- Large quantity of informal documents are produced.
- Users need to access multiple documents to perform a task.

Advantages:

- Breadth of information is high.
- Low cost.
- Depth of information is medium.

Disadvantages:

- No additional information.
- Integration of information is low.
- User involvement is low.

2.5: Observation:

Act of watching processes being performed to gain insight into the current system.

Also used to check the validity of information gathered from other sources, e.g. interviews and questionnaires.

Careful to keep a low profile: people tend to behave well when being watched!

Advantages:

- Low-medium cost.

Disadvantages:

- Breadth of information is low.
- No additional information.
- Integration of information is low.
- Depth of information is low.
- User involvement is low.

2.6: The most suitable technique for the project:

Compare between the techniques:

	Interviews	JAD	Questionnaire	Document analysis	Observation
Type of information	As-is, improvements, to-be	As-is, improvements, to-be	As-is, improvements	As-is	As-is
Depth of information	High	High	Medium	Low	Low
Breadth of information	Low	Medium	High	High	Low
Integration of information	Low	High	Low	Low	Low
User involvement	Medium	High	Low	Low	Low
Cost	Medium	Low-Medium	Low	Low	Low-Medium

Compare each technique applied to the Tune Source project:

Techniques	Project of Tune Source
Interviews	- Very beneficial by adding the information of the stakeholders. - Information has depth. - However, the cost is high and it is difficult to interview each person. - Each one has an idea, so it is difficult to Integration the information.
JAD	- Very beneficial by adding the information of the stakeholders. - Information has depth. - Integration the information is high. - Users can participate to improve the product. Help Tune Source compete well with other competitors in the music download market. - Cost is moderate.
Questionnaire	- Very beneficial by adding the information of the stakeholders. - Breadth of information is high. - Low cost,but each one has an idea, so it is difficult to Integration the information.
Document analysis	- No additional information from stakeholders. - Depth of information is low. - Tune Source website needs more information from users to be able to improve, compete with competitors in the market.
Observation	- No additional information from stakeholders. - Depth of information is low. - Tune Source website needs more information from users to be able to improve, compete with competitors in the market.

Through the table comparing the techniques with the project, ABC Company selected JAD as the most suitable technique for the project of Tune Source.

III. Analysis (2):

1. Use case modelling:

Use case diagram for Tune Source's music download system:

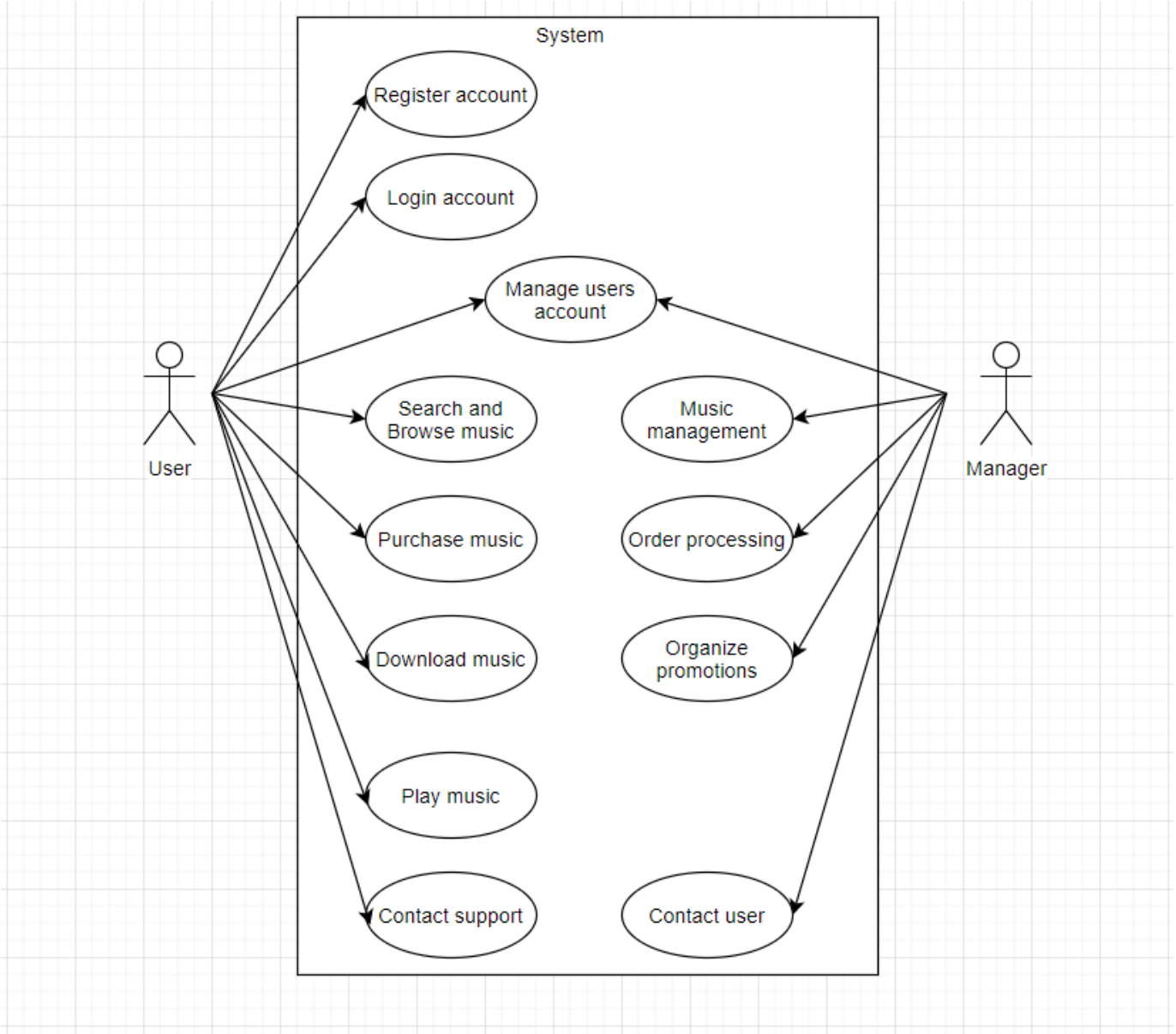


Figure: Use case diagram

Use case description:

	Actor	Description	Data	Stimulus	Response	Comments
Register account	User	Require users to enter an account, password and related information to register for an account.	Account name, password, username, email, address, form of payment, phone number,...	User press the register button.	The system notifies successful registration and and save user data to the database.	Unable to register if the account already exists. User information is saved to the database and must be kept confidential.
Login account	User	Log into the software with the user's account.	Account name, password.	User press the login button.	The system notifies successful login and creates SESSION for the user.	Unable to log in if wrong account or password.
Manage users account	User, manager.	Manage, update, view user information.	Account name, password (for user only), username, email, address, phone	User or manager press the manage user account button.	The system shows user information. Allows editing if viewers have permission.	Editing information is not allowed if user's SESSION does not have permission.
Search and Browse music	User	Allows users to search and order music.	Name of song, singer, songwriter, album.	User find the song.	The system displays song(s) for user.	Error message when the song cannot be found.

Purchase music	User	User purchase the song.	Song information, user information.	Users press the purchase button.	Allow users to download the full version of the song if the payment is successful.	Cannot buy if user does not have enough money.
Download music	User	Allow users to download digital music	The song	Users press the download button.	Allows users to download if user has already purchased the song.	
Play music	User	Allow users to listen to purchased music or play previews if not purchased.	The song	Users press the play button.	Allow users to listen to purchased music or play previews if not purchased.	
Contact support	User	Users contact the support of Tune Source	User information	Users press the contact button.	Send user information to Tune Source support staff to contact the user.	

2. Activity modelling:

In this report, ABC Company selected the Play music function in the Tune Source project to represent the Activity diagram:

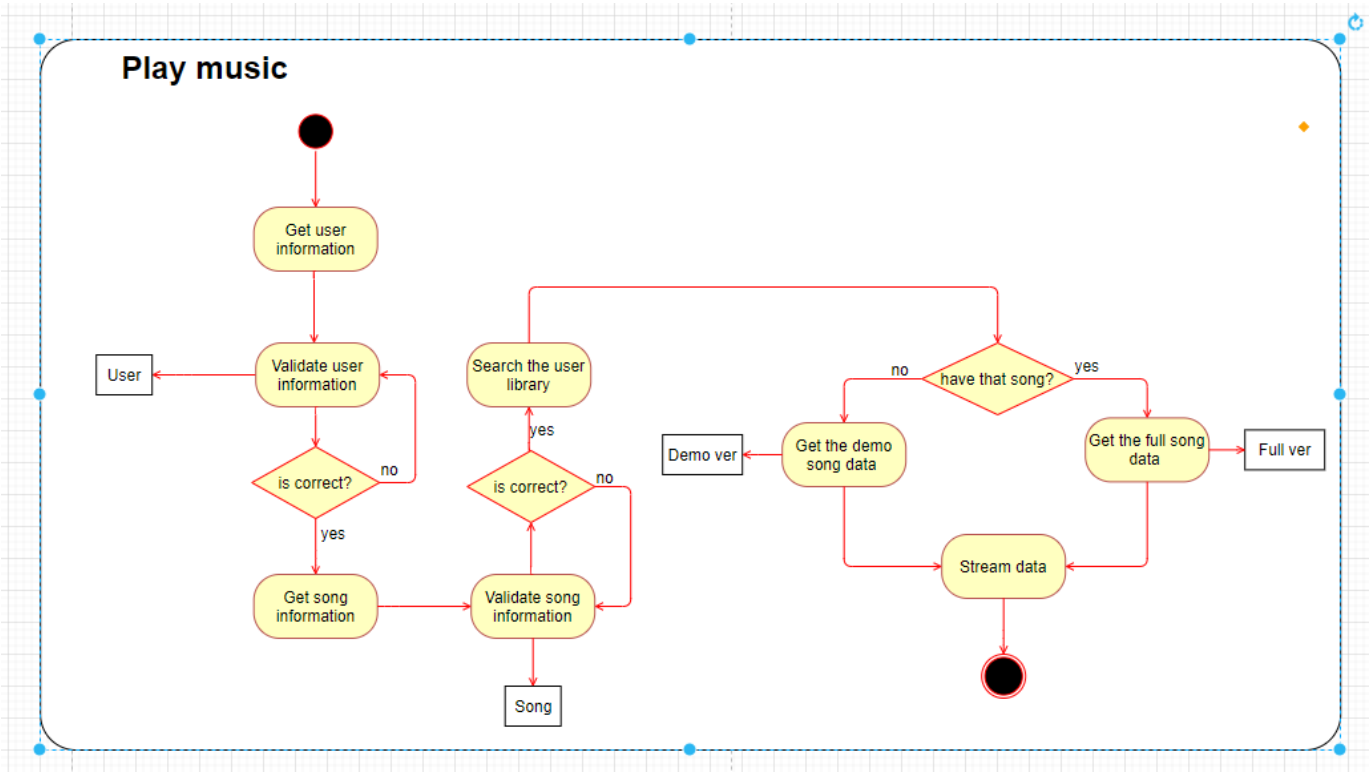


Figure: Activity Play music diagram

Explain:

- When the user presses the Play button on a certain song, the Play music function begins.
- Retrieve user information.
- Authentication with user information in database.
- Get the song information.
- Authentication with song information in the database.
- Check if the user's Library already owns that song.
- If the user already owns the song, get full song data information.
- If the user does not yet own the song, get the demo song data information.
- Stream the song data to the user's device.

3. Structural modelling with class diagram:

In the function "Play music" there are four Classes:

- User: user information (id, name, library).
- Song: song information (id, name, demo version file path, full version file path).
- Demo version: data stream of demo version song (binary).
- Demo version: data stream of full version song (binary).

User's relationship with Song is 1 to Many.

Song's relationship with Demo version and Full version is 1 to 1.

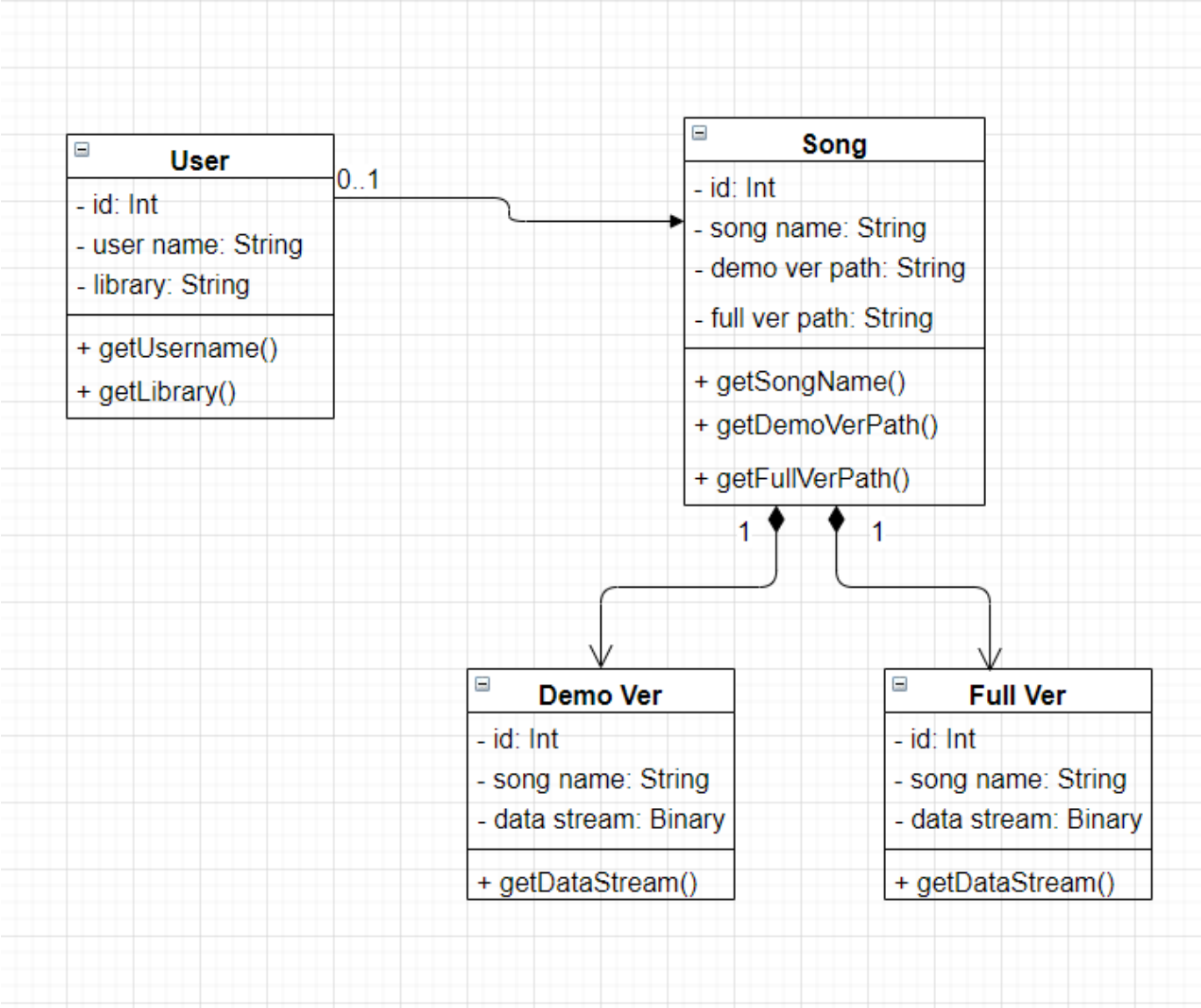


Figure: Class diagram

IV: Design:
1. Refinement procedure:

Design phase:

Determine how the software will operate:

- Refine the concept developed in analysis.
- To specify a design blueprint.

Main steps:

- Determine design strategy.
- Design software architecture.
- Construct design models: components, user interface, database...

Design models:

Software design is concerned with:

- What components will be created.
- The overall design model, explaining how components fit together.
- The specification of each component: explaining its behaviour (what it does), for programmer to write code (later in implementation phase).

Creating design model by transforming the analysis models: Transformation: to add technical details to the model (e.g. programming language, underlying system software, ...), also means refinement.

Employ **object oriented modelling** techniques to construct the design models: useful if analysis phase was also conducted using these techniques!

Techniques to use:

- Behavioural model:
  - Activity diagram (to be refined)
  - Interaction diagrams: sequence diagram, communication diagram
  - Pseudocode
- Structural model: class diagram (to be refined).

Design usages of activity diagram:

Activity diagram can describe processes at different levels of abstraction:

- High-level business processes.
- Low-level, programming methods.

Two design usages of activity diagram:

- Refinement: to describe the low-level detail of a process.

- Alternative to pseudocode: to describe the algorithm of an object operation.

Activity diagram for design refinement:

Check each action in an activity: if it is still decompsable then decompose it by creating a nested activity.

- Stop refinement when all actions are 'truly' nondecomposable
- Update the structural models with new classes (if any)

2. Illustrate the procedure:

In this report, ABC Company will use the Activity Diagram "Play Music" designed in the previous section to illustrate the refinement procedure.

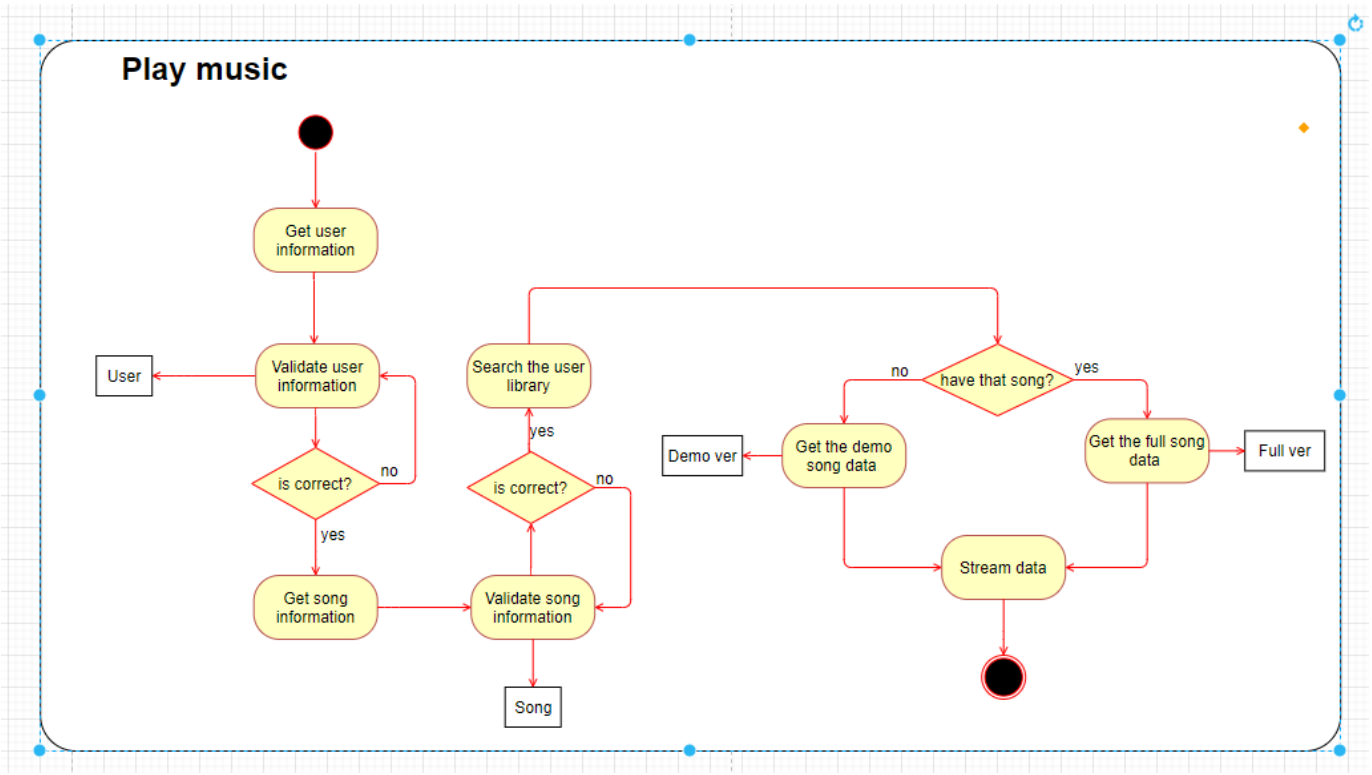


Figure: Activity Play music diagram

Refine the function Get user info and Get song info:

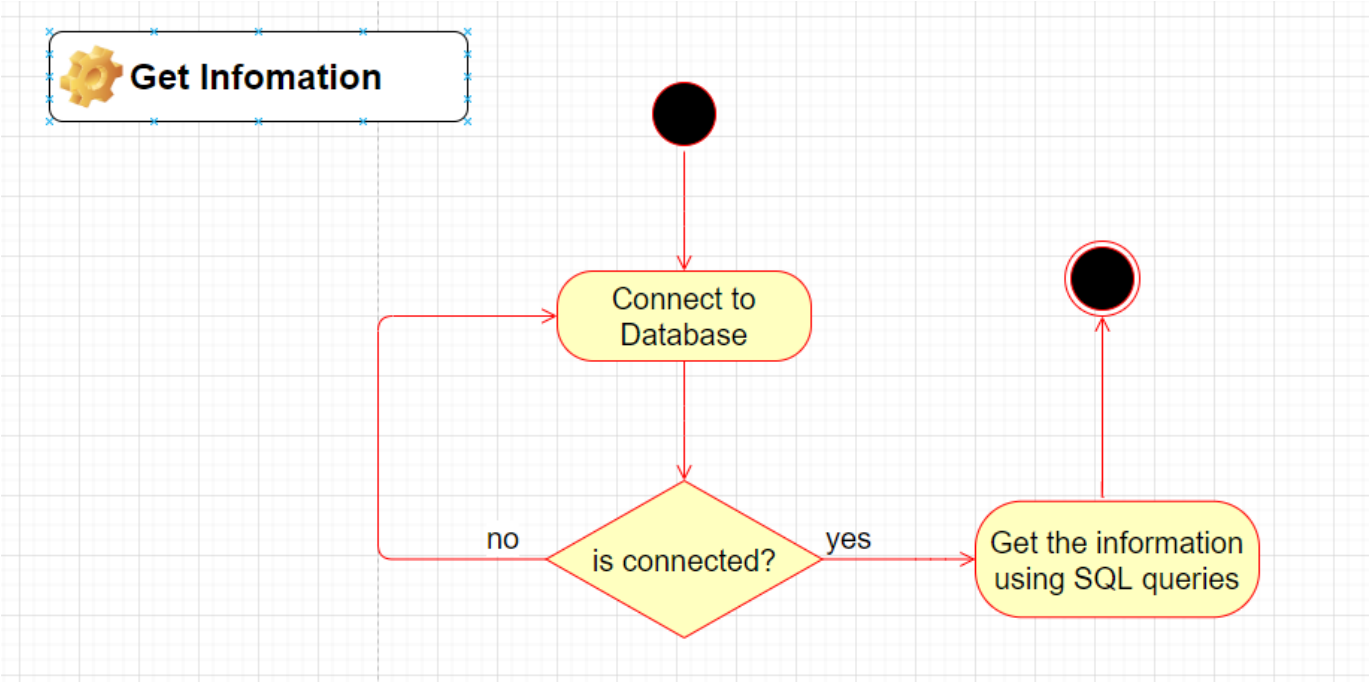


Figure:Get infor function

Refine Search function:

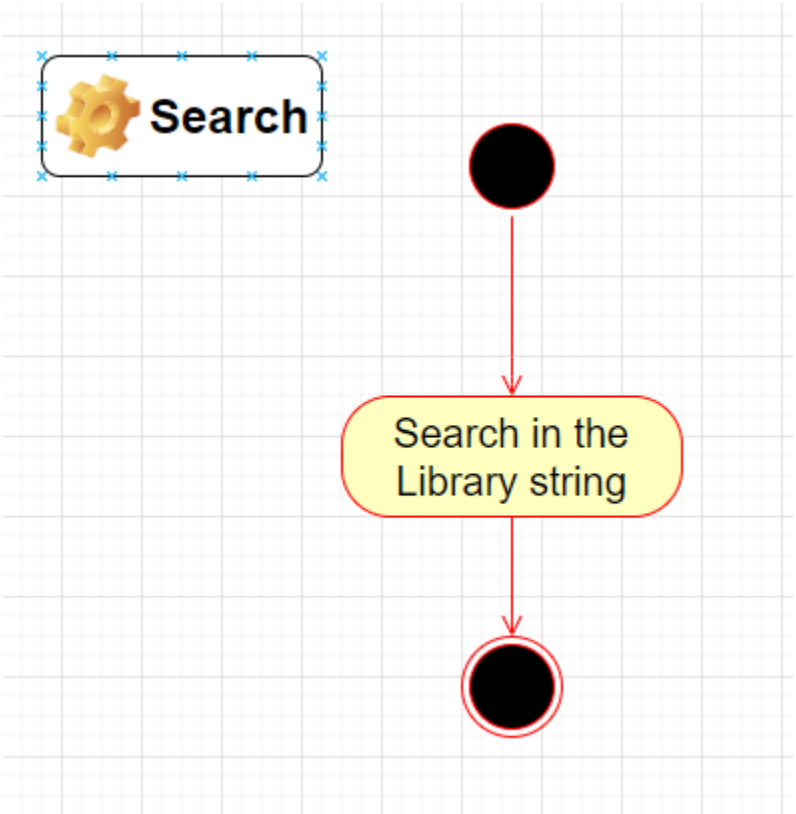


Figure: Search function

Refine Stream data function:

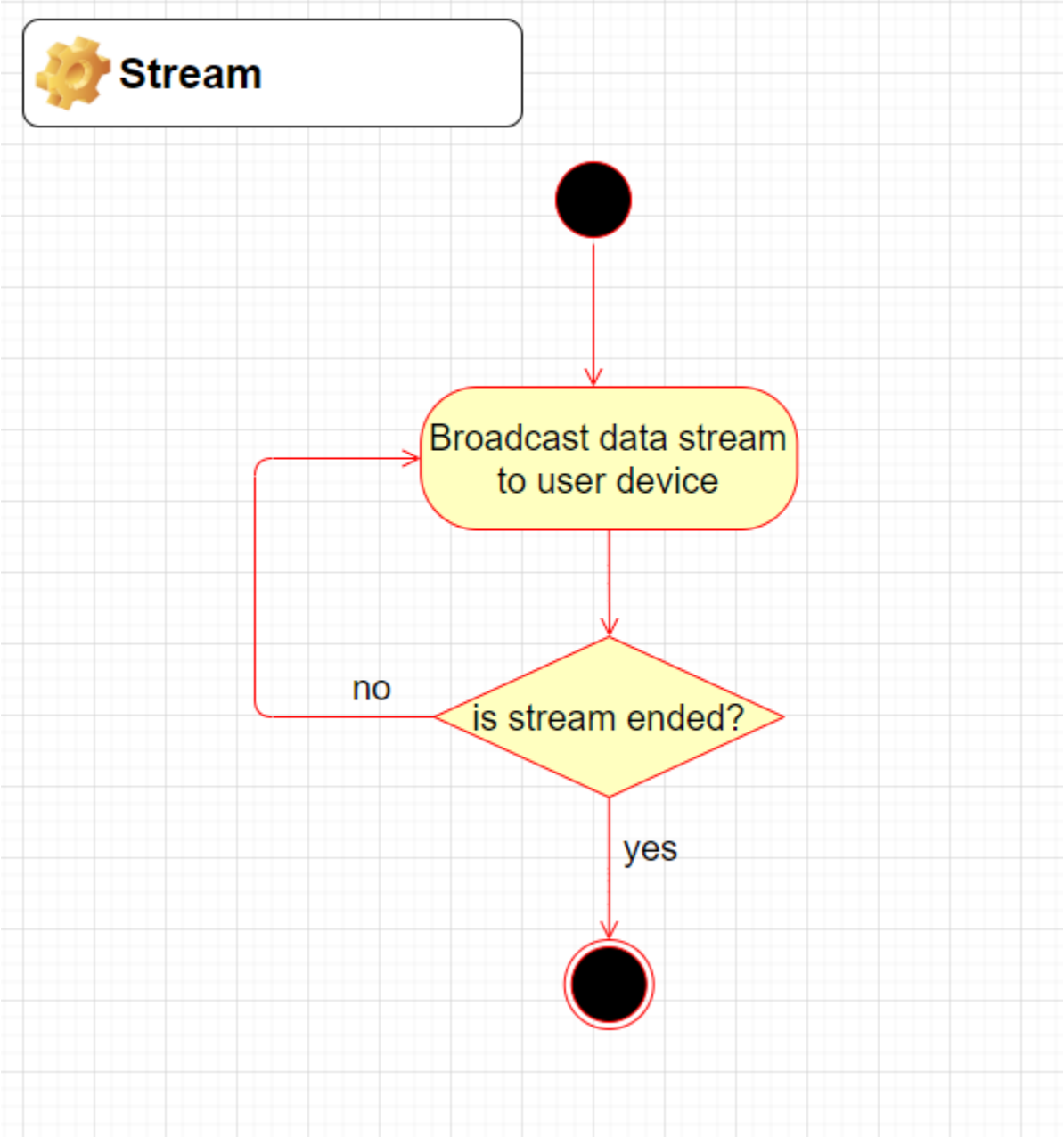


Figure: Stream data function

Link Draw.io:
https://drive.google.com/file/d/1JdVA86wmdnnR0igbE3W3ZqeLt2UJOBSu/view?usp=sharing

V. Conclusion:

Today, the Software Development Life Cycle (SDLC) has become an indispensable task for every project before it starts. Feasibility study, choosing the right software development model will help reduce costs, time, manage risks and bring success to the project and product.

VI. Reference:

Katharine , P., Charles, J., David, W., Asaf, Z. (2005). “The Stakeholder Engagement Manual”. Stakeholder Research Associates Canada Inc., Ontario, pp.8.

Harwell, R. et al. (1993). “What is a Requirement?”. Proceedings of NCOSE 1993, Washington, DC.

Ivan, G., Pacheco, C., Reyes, M. (2018). “Requirements elicitation techniques: A systematic literature review based on the maturity of the techniques”. IET Software. 12. 365-378.

Index of comments

2.1 An average report, in which you demonstrated an acceptable level of improvement from the first attempt.

The following improvements should be considered to make the report more compelling:

Analysis (2):
Activity diagram:
Sect 2:
- Full and Demo ver: should become input of Stream Data

Class diagram:
- Demo and Full ver: should be subclasses of Song

Design:
Fig 7:
- Customer: should have more attributes
Fig 9:
- Object: should have more specific name