**ASSIGNMENT 02 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | Unit 09: Software Development Life Cycle | | |
| **Submission date** | 07/09/2022 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Luong Gia Luan | **Student ID** | GCC200292 |
| **Class** | GCC0903 | **Assessor name** | Nguyen Thai Nghe |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** |  |

**Grading grid**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P5 | P6 | P7 | M3 | M4 | M5 | M6 | D3 | D4 |
|  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Internal Verifier’s Comments:** | | |
| **Signature & Date:** | | |

# Assignment Brief 02 (RQF)

## Higher National Certificate/Diploma in Business

|  |  |
| --- | --- |
| **Student Name/ID Number:** |  |
| **Unit Number and Title:** | **Unit 09: Software Development Life Cycle** |
| **Academic Year:** |  |
| **Unit Assessor:** |  |
| **Assignment Title:** | **Undertake a software development life cycle** |
| **Issue Date:** | **07/12/2020** |
| **Submission Date:** |  |
| **Internal Verifier Name:** |  |
| **Date:** |  |

|  |
| --- |
| **Submission Format:** |
| *Format:*   * The submission is in the form of 1 document. * You must use the Times font with 12pt size, turn on page numbering; set line spacing to 1.3 and margins to be as follows: left = 1.25cm, right = 1cm, top = 1cm, bottom = 1cm. Citation and references must follow the Harvard referencing style.   *Submission:*   * Students are compulsory to submit the assignment in due date and in a way requested by the Tutor. * The form of submission will be a soft copy posted on <http://cms.greenwich.edu.vn/>. * Remember to convert the word file into **PDF** file before the submission on CMS.   *Note:*   * The individual Assignment must be your own work, and not copied by or from another student. * If you use ideas, quotes or data (such as diagrams) from books, journals or other sources, you must reference your sources, using the Harvard style. * Make sure that you understand and follow the guidelines to avoid plagiarism. Failure to comply this requirement will result in a failed assignment. |
| **Unit Learning Outcomes:** |
| **LO3** Undertake a software development lifecycle.  **LO4** Discuss the suitability of software behavioural design techniques. |
| **Assignment Brief and Guidance:** |
| **Tasks**  At this stage, you have convinced Tune Source to select your project for development. Complete the following tasks to analyse and design the software.  **Task 1 – Analysis (1)**   1. Identify the stakeholders, their roles and interests in the case study.   Review the requirement definition of the project. Clearly indicate which stakeholder(s) provide what requirements.  *Word limit: 150 – 200.*  Identify FRs and NFRs of Tune Source Project.  Discuss the relationships between the FRs and NFRs.  *Word limit: 300 – 400 words.*   1. Discuss the technique(s) you would use to obtain the requirements.   If needed, you may state suitable additional assumptions about the project in order to justify the technique(s) that you choose.  *Techniques: JAD, Interview, Observation, etc.*  *Demonstrate how to collect requirements based on chosen technique.*  *Word limit: 700 – 1000.*   1. Discuss how you would trace these requirements throughout the project by using Requirement Traceability matrix. You will have to provide real usage of it.   *Word limit: 400 – 500 words.*  **Task 2 – Analysis (2)**  Analyze the requirements that you identified in Task 1 using a combination of structural and behavioral modelling techniques that you have learnt.  *Scope:* You only need to construct following items for the system. You will have to include:   * + Use Case Diagram for the whole system.   + Use Case specification for 2 Use cases.   + Context Diagram for the whole system.   + Data Flow Diagram – Level 0 for the whole system.   + ERD for the whole system.   For each diagram, you will have to explain properly.  *Word limit: 1000 – 1200 words.*  **Task 3** **– Design**  Based on the analysis result, discuss how you would conduct the design phase:   1. Discuss how the user and software requirements are addressed in the design phase.    * You will explain how Mock-up, and Wireframe are used in the project. You should include some of the mockup or wireframe (at least 5) design of the Tune Source project to justify that it matches users’ requirements.    * You will explain which architecture (client – server, n-tier, microservices, etc.) is suitable for the project with clear illustrations and why.    * Then you will address which technical solution stack could be suitable to implement the project with clear explanations. 2. Discuss how activity diagram and pseudocode are used to specify the software behaviour. 3. Discuss how UML state machine can be used to specify the software behaviour. Differentiate between FSM and extended FSM using the case study. 4. Discuss how the data-driven approach improves the reliability and effectiveness of software.   *Word limit: 800 – 1500.*  **Task 4 – Software quality management**   1. Discuss two software quality attributes that are applicable to the project. 2. Discuss two quality assurance techniques that can help improve the software quality in the project. 3. Discuss how the design techniques and approaches that you have used can help improve the software quality.   *Word limit: 400 – 1500.* |

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning Outcomes and Assessment Criteria (Assignment 02):** | | | |
| Learning Outcome | Pass | Merit | Distinction |
| **LO3** Undertake a software development lifecycle | **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. | **D3** Critically evaluate how the use of the function design paradigm in the software development lifecycle can improve software quality. |
| **LO4** Discuss the suitability of software behavioural design techniques | **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. | **D4** Present justifications of how data driven software can improve the reliability and effectiveness of software. |

Table of Contents

[Assignment Brief 02 (RQF) 1](#_Toc112864211)

[Higher National Certificate/Diploma in Business 1](#_Toc112864212)

[**1.** **Undertake a software investigation to meet a business need.** 6](#_Toc112864213)

[**1.1.** **Identify the stakeholders, their roles and interests in the case study.** 6](#_Toc112864214)

[**1.1.1.** **Review the requirement definition of the project. Clearly indicate which stakeholder(s) provide what requirements.** 7](#_Toc112864215)

[**1.1.2.** **Identify FRs and NFRs of Tune Source Project.** 7](#_Toc112864216)

[**1.1.3.** **Discuss the relationships between the FRs and NFRs.** 8](#_Toc112864217)

[**1.2.** **Discuss the technique(s) you would use to obtain the requirements.** 8](#_Toc112864218)

[**2.** **Use appropriate software analysis tools/techniques to carry out a software investigation and create supporting documentation.** 9](#_Toc112864219)

[**2.1.** **Use Case Diagram for the whole system.** 9](#_Toc112864220)

[**2.2.** **Use Case specification for 2 Use cases.** 9](#_Toc112864221)

[**2.3.** **Context Diagram for the whole system.** 10](#_Toc112864222)

[**2.4.** **Data Flow Diagram – Level 0 for the whole system.** 10](#_Toc112864223)

[**2.5.** **ERD for the whole system.** 10](#_Toc112864224)

[**3.** **Explain how user and software requirements have been addressed.** 10](#_Toc112864225)

[**4.** **Analyse how software requirements can be traced throughout the software lifecycle.** 10](#_Toc112864226)

[**5.** **Discuss two approaches to improving software quality.** 10](#_Toc112864227)

[**6.** **Suggest two software behavioural specification methods and illustrate their use with an example.** 10](#_Toc112864228)

[**7.** **Differentiate between a finite state machine (FSM) and an extended-FSM, providing an application for both.** 11](#_Toc112864229)

[**8.** **Critically evaluate how the use of the function design paradigm in the software development lifecycle can improve software quality.** 11](#_Toc112864230)

[**9.** **Present justifications of how data driven software can improve the reliability and effectiveness of software.** 11](#_Toc112864231)

1. **Undertake a software investigation to meet a business need.**
   1. **Identify the stakeholders, their roles and interests in the case study.**

According to <https://www.exposit.com/blog/primary-secondary-stakeholders-software-project/> , a software project's stakeholders are people, groups, or organizations that are actively participating in it, have the power to influence it because of their position, and whose interests could be impacted by the project's success or failure. Throughout the software development lifecycle, stakeholders may have varying levels of authority and responsibility. Competing expectations of various stakeholders can occasionally spark a contentious discussion. Engaging stakeholders, however, is one of the best ways to reach your main business objectives and generate excellent project results.

The table below will display the stakeholders, their roles and interests in the case study of Tune Source project:

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Role** | **Interest** |
| **Project Manager** | Manage all aspects of the project creation process | Their main goal is to satisfy clients by delivering a quality product on schedule and within budget. They manage a group of developers and keep an eye on the procedures involved in project execution, making improvements as needed. |
| **Development Team** | Implements and develops the project | The developers can offer stakeholders advise on how to implement business ideas and provide quality estimation needed to comprehend the scope of work and resources needed based on their experience and innovation trends. Additionally, this group consists of QA engineers who create defects to satisfy criteria and avoid problematic user scenarios. |
| **Designers** | Make the software interface | They are aware that users visit the website or utilize the application to address a certain issue. Therefore, designers ensure that clients can easily and swiftly receive what they desire. |
| **Customer** | Define the project's core requirements and scope | They collaborate often with the team, give their approval, or add additional implementation details to the strategy. |
| **Supplier** | Supplies essential products for the project. | Suppliers are individuals or organizations that provide goods to the company and depend on it to generate income from the selling of those commodities. |

* + 1. **Review the requirement definition of the project. Clearly indicate which stakeholder(s) provide what requirements.**

According to the scenario of the Tune Source project, by enabling the sale of digital music downloads to customers through kiosks in our stores and over the Internet utilizing our website, this project has been started to improve sales. Customers will be able to look for and purchase digital music downloads using the Internet or in-store kiosks. The following are some examples of the precise functionality that the system has to have:

* Search for music in our digital music archive.
* Listen to music samples.
* Purchase individual downloads at a fixed fee per download.
* Establish a customer subscription account permitting unlimited downloads for a monthly fee.
* Purchase music download gift cards.

Beside the requirement definition of the project, there are also some requirements from the stakeholder of the Tune Source project:

**Customer requirement**: the system must satisfy the following 5 functions that customer can search a music by its name or music’s singer, listen to music, purchase individual downloads at a fixed fee per download, register an account to purchase music, and purchase music download gift cards. Moreover, the system must ensure the security of each user's account. In addition, the system interface must be simple and user-friendly, and the project implementation time must be fast to meet the needs of users promptly.

**Supplier requirement**: The information of the songs provided by the supplier must ensure the accuracy of the song title to the singer's name. For copyrighted songs, the parties involved need to agree on a contract before posting them on the system. Refrain from publishing songs related to politics or opposing the state.

* + 1. **Identify FRs and NFRs of Tune Source Project.**

According to <https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/> , the success of a system or software project may be evaluated thanks to the very important procedure known as requirements analysis. Functional and non-functional requirements are the two main categories of requirements.

* **Functional Requirements**: These are the features that the system must provide that are considered essential by the end user. As a requirement of the contract, all of these functionalities must be built into the system. These are shown or described as the input to be provided to the system, the operation carried out, and the intended outcome. In contrast to non-functional needs, they are essentially the user-stated criteria that are visible in the finished product.
* **Non-functional requirements**: These are essentially the quality requirements specified in the project contract that the system must meet. Depending on the project, these criteria may be prioritized differently or applied to a different degree. Additionally known as non-behavioral requirements.

In the Tune Source, funtional and non-functional requirements are:

**Functional requirements**:

* Search for music in our digital music archive.
* Listen to music samples.
* Purchase individual downloads at a fixed fee per download.
* Establish a customer subscription account permitting unlimited downloads for a monthly fee.
* Purchase music download gift cards.

**Non-functional requirements**:

* The system can run on PC, laptop, and smartphone.
* The system can access from at least browsers Chrome, Opera, Microsoft Edge, Firefox, and so on.
* The screens would have a response time of less than 3 seconds.
* The number of transactions in the system can handle at a time at least 5000 people.
* Improve system security to avoid common security vulnerabilities: SQL injection, cross-site scripting, broken authentication, and so on.
* Authenticator between user and admin that admin can manage information of music and account user but not the important information such as password, surplus, and so on.
* The usability functions in the system must be less than or equal to 3 clicks.
* The currency used for payment in the system is dollars or Vietnamese dong.
  + 1. **Discuss the relationships between the FRs and NFRs.**

Functional requirements explain how the system must work, while non-functional requirements explain how the system must work. These two requirements are completely different but they resonate and complement each other to implement and execute a complete project. The project will ensure the main requirements in addition to developing a number of other functions to improve the efficiency and quality of a system in a particular project. In addition, if there are functional requirements without non-functional requirements, the system, although able to meet the requirements of the customer in terms of main functions, will create many problems. Therefore, non-functional requirements must be added, as a form of risk control for the system and to improve the efficiency and availability of the system.

* 1. **Discuss the technique(s) you would use to obtain the requirements.**

Before starting the Tune Source project, obtaining the requirements from customers is important that guide the entire direction of the project. According to <https://www.jamasoftware.com/requirements-management-guide/requirements-gathering-and-management-processes/11-requirements-gathering-techniques-for-agile-product-teams> , there are some techniques to obtain the requirement:

* **Interviews**

The process of eliciting requirements can be started off well with interviews. They are invaluable for compiling background data on organizational requirements, user and customer issues, and support personnel and other key stakeholders' worries. Interviews can also be used as a follow-up to collect information in greater detail.

A varied and representative cross-section of the system's stakeholders should be covered in the interviews. The whole spectrum of user and customer profiles should be included. To ensure that your system requirements aren't biased in favor of one group, it is important to acquire a proper perspective on conflicting needs.

Asking open-ended questions is crucial while conducting interviews. A simple "yes" or "no" response cannot be given in response to an open-ended question. They extract particular data. They ask the interviewee to clarify their ideas and give reasoning, which gives the requirements context and can be evaluated and validated.

* **User Observation**

Numerous difficulties arise during one-on-one interviews. They might be time-consuming for the interviewers to conduct and difficult to schedule. Additionally, the requirements you compile might just scrape the surface because not all interviewers are adept at following up in the moment.

Surveys and questionnaires are effective alternatives. They enable simultaneous follow-up with numerous parties.

An effective method for identifying those underlying requirements that stakeholders may not be entirely aware of but which are crucial to a successful design is a well-thought-out questionnaire that poses probing questions.

* **JAD (Joint Application Development)**

An approach known as JAD (Joint Application Development) engages the client or end user in the design and development of an application through a series of cooperative workshops known as JAD sessions. Because the client is involved at every stage of the development process, the JAD technique is said to result in quicker development timelines and higher client satisfaction than the more conventional practice. In contrast, the traditional method of systems development entails the developer researching the needs of the system and creating an application with client input coming from a series of interviews.

<https://www.techtarget.com/searchsoftwarequality/definition/JAD>

Based on the project's scenario, this is a small project, suitable for the waterfall model or v-model. Therefore, taking requirements from customers must be clear and detailed to avoid the case of adjusting or changing requirements for the project. This affects the project execution model leading to a delay in the project implementation process, affecting the efficiency and quality of the project. So, choose the interview technique to get requirements from customers that will be used for this project.

According to <https://www.directutor.com/content/interview-technique-requirements-gathering-techniques> , there are some steps to collect requirements from customer with interview technique:

1. Identify stakeholders to be interviewed

2. Obtain a general understanding of the customers business

3. Develop interview questions using open-ended questions

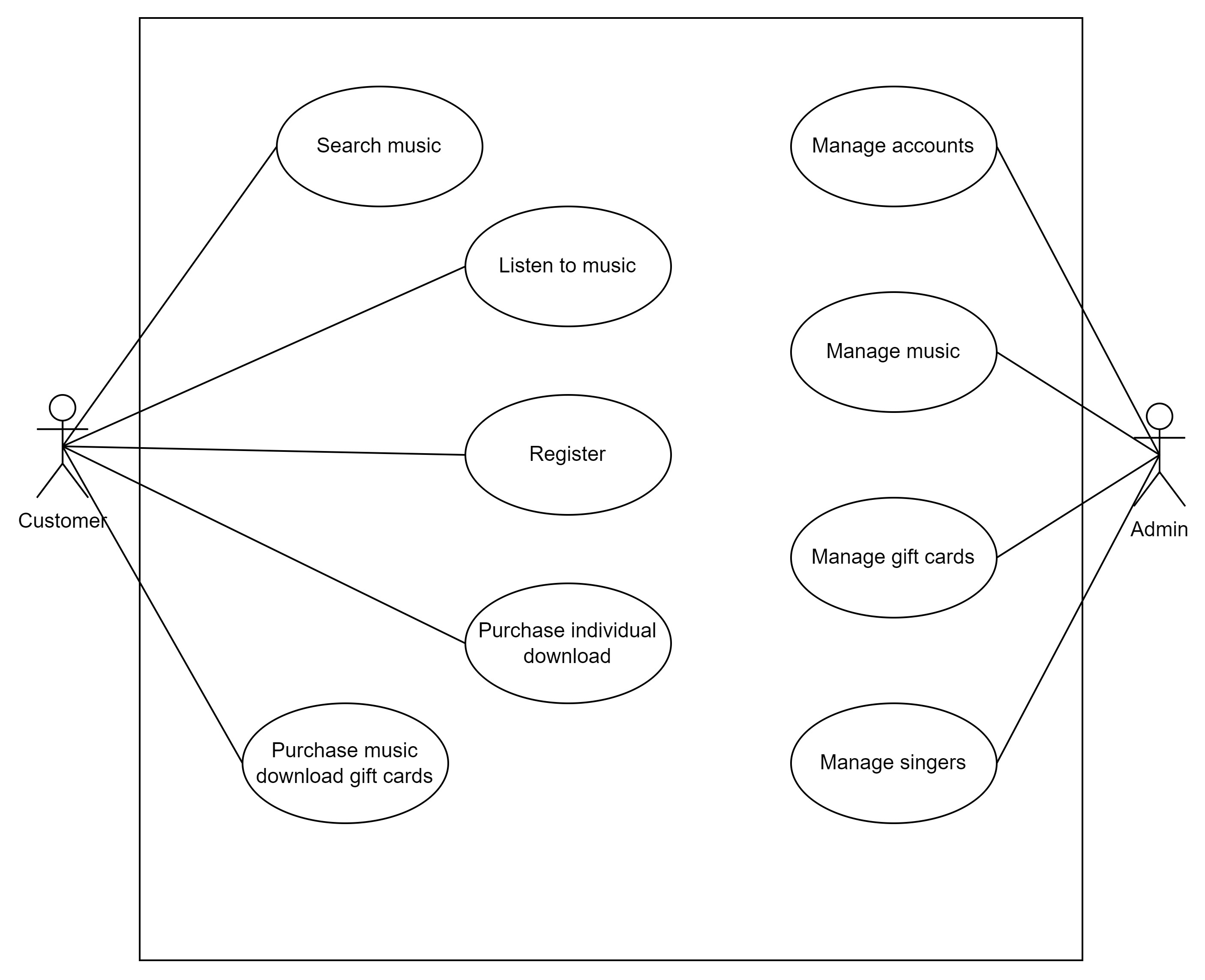
4. Set meeting time and location for the interview

5. Provide a set of questions to interviewees prior to the interview (if they will need to prepare for the interview)

6. Use one or more Recorders to accurately preserve results of the interview

7. Provide results to interviewees for confirmation of content

1. **Use appropriate software analysis tools/techniques to carry out a software investigation and create supporting documentation.**
   1. **Use Case Diagram for the whole system.**

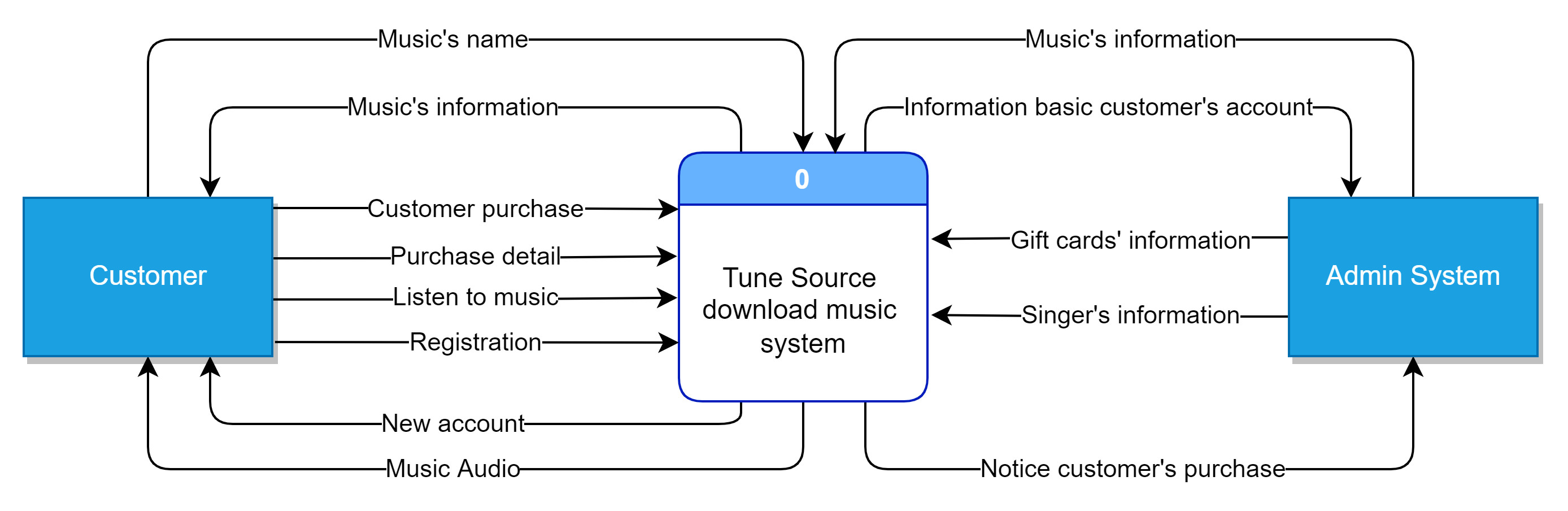


* 1. **Use Case specification for 2 Use cases.**

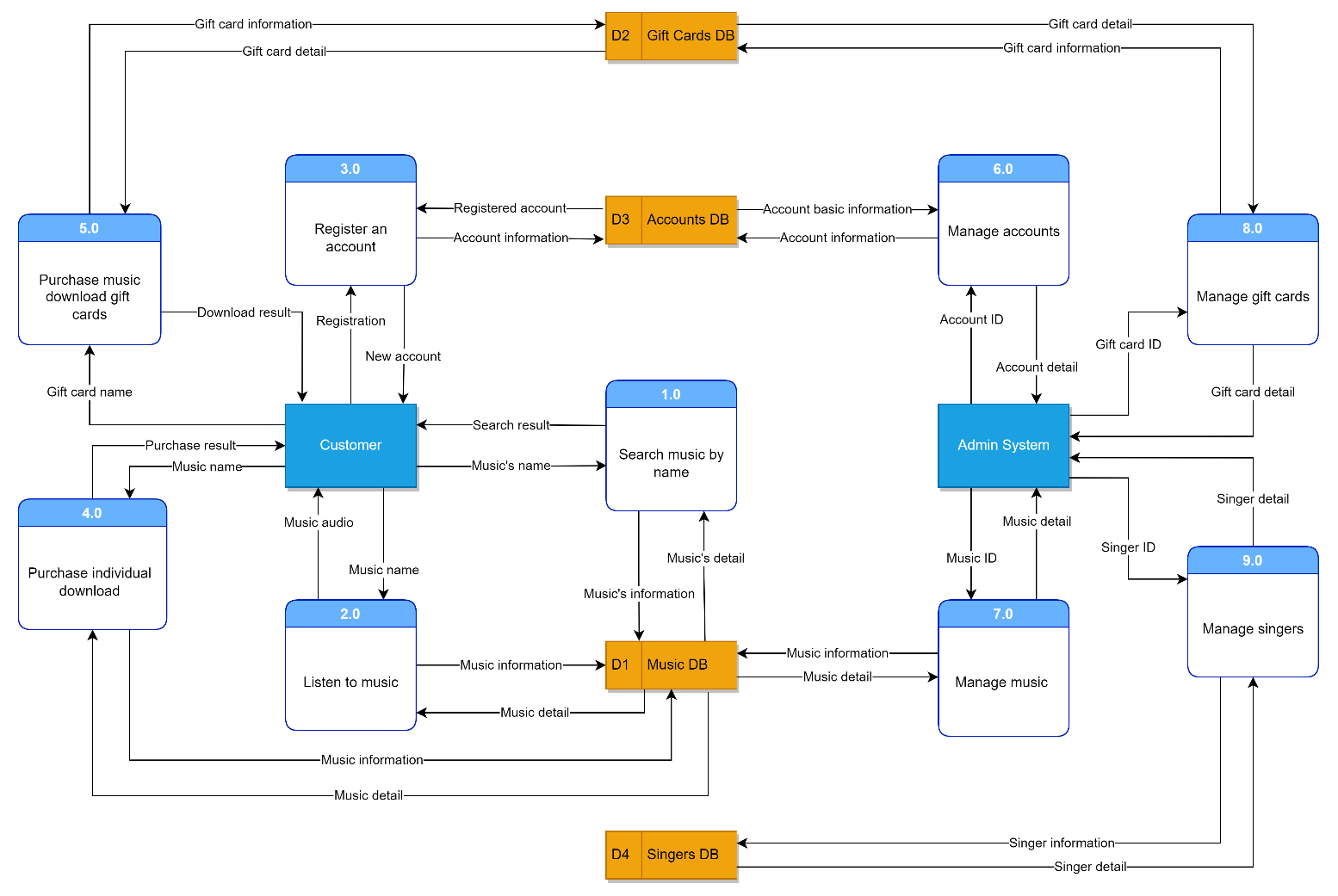
|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Search Music | | ID: UC-1 | Priority: High |
| Actor: User | | | |
| Description: The search music function will help the user search for the music song that they want by entering the music’s name or music’s singer into the search bar on the header of the website. | | | |
| Trigger: The user clicks button search or press Enter  Type:  External  Temporal | | | |
| Preconditions:  1. The user must already have access to the website.  2. The user must enter music’s name or music’s singer into the search bar and click button search or enter to perform search function.  3. Music information must be in database. | | | |
| Normal Course:   1. Requires the user fill out music’s name or music’s singer    1. The system receives information    2. The system gets and check informations   of music in database   * 1. Return the information of music in   homepage | | **Information for Steps:**  Music’s name or music’s singer  Clicks button search or press Enter  Show list of music like search result | |
| Alternative Courses: | |  | |
| Postconditions: None | | | |
| Exceptions:  1. The system display a message “No result” when there is no music song like keyword searched | | | |
| Summary Inputs | **Source** | **Outputs** | **Destinations** |
| Music’s name  Music’s singer  Click button search  Press Enter | User  Database music  in the system | List of music | Interface to  display list  of music |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case Name: Purchase individual download | | ID: UC-4 | | Priority: High |
| Actor: User | | | | |
| Description: The function will allow the user to purchase for music song they buy in the system but the user must be logined before perform the function and already have money in their account. | | | | |
| Trigger: The user purchase for music song  Type:  External  Temporal | | | | |
| Preconditions:  1. The user must be having an account  2. The user must be login into the account  3. The balance of user’s account must be greater than 20$ before perfomr purchase | | | | |
| Normal Course:  1.0 Purchase individual download music   1. User login into account 2. Click music song 3. Click icon download 4. The system checks the balance in the account 5. The system confirms the account and music 6. The system reduces the balance in the account and gives the music into the account | | | **Information for Steps:**  Username and Password  Display music detail  Balance in the account  AccountID and MusicID  Display a message to notify purchase  successfully | |
| Alternative Courses:  1.1 Not enough money in the account to purchase   1. User login into account 2. Click music song 3. Click icon download 4. The system checks the balance in the account 5. The balance in the account is less than the music’s price 6. The system cancels the purchase 7. The system recommends recharging more money into the account | | | Username and Password  Display music detail  Balance in the account  Balance and music’s price  Display an error message  Display interface account user | |
| Postconditions:  1. The balance in the account must be reduces by the same price of music song the user purchase  2. The quantity of music must be reduced after the user purchase  3. The system recode the purchase as history into database | | | | |
| Exceptions:  1. The system will display an error message “The balance in the account not enough to purchase for music song”  2. The system will redirect the user to login page when the user download music without login first. | | | | |
| Summary Inputs | **Source** | **Outputs** | | **Destinations** |
| Username and Password  AccountID and MusicID | User  Database account and music in the system | Music information | | Redirect to homepage |

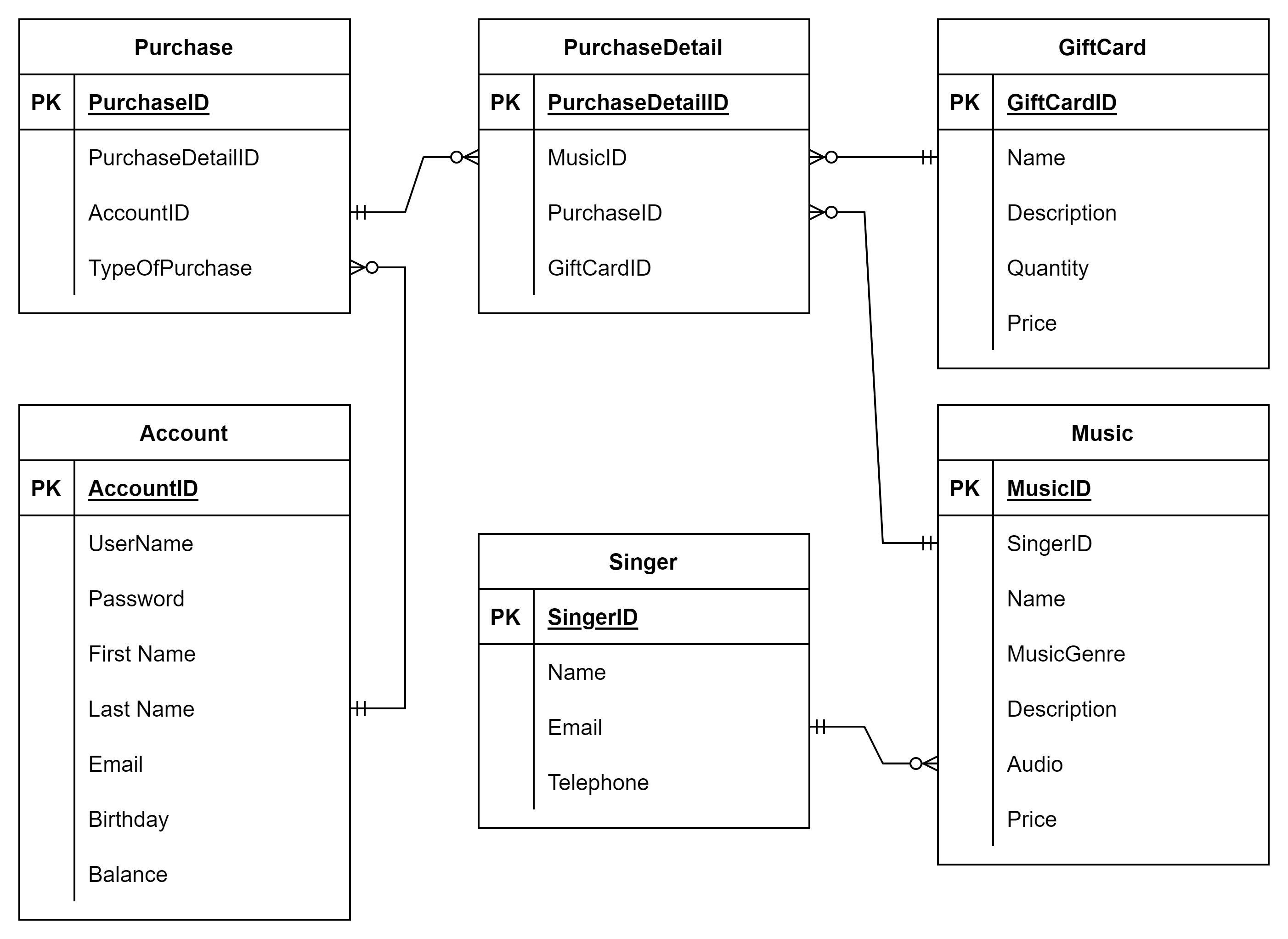
* 1. **Context Diagram for the whole system.**



* 1. **Data Flow Diagram – Level 0 for the whole system.**



* 1. **ERD for the whole system.**



1. **Explain how user and software requirements have been addressed.**
2. **Analyse how software requirements can be traced throughout the software lifecycle.**
3. **Discuss two approaches to improving software quality.**
4. **Suggest two software behavioural specification methods and illustrate their use with an example.**
5. **Differentiate between a finite state machine (FSM) and an extended-FSM, providing an application for both.**
6. **Critically evaluate how the use of the function design paradigm in the software development lifecycle can improve software quality.**
7. **Present justifications of how data driven software can improve the reliability and effectiveness of software.**