**ASSIGNMENT 02 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | Unit 09: Software Development Life Cycle | | |
| **Submission date** | 7/9/2022 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Do Huu Duy | **Student ID** | GCC200018 |
| **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** | huuduy |

**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 behavioral 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 analyze 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 behavior. 3. Discuss how UML state machine can be used to specify the software behavior. 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** Analyze 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 behavioral design techniques | **P7** Explain how user and software requirements have been addressed. | **M5** Suggest two software behavioral 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. |

I. Undertake a software investigation to meet a business need

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

1.1. What is stakeholder

A stakeholder is a party that has an interest in the company for a project and can influence or be affected by the project's business. The main stakeholders of a typical company are investors, employees, customers, and suppliers (FERNANDO, 2022)

1.2. their roles and interests in the case study

For the Tune Source project, the stakeholder includes users and singers. For the users, they will use the access to the website of Tune Source to register an account and they can listen to music online on there. In addition, when they register an account success, they can log in and search the music and download the music that they like but they must pay fees. Besides, they can buy some gift card music to send to their relatives. However, to perform the above things they have to register an account and have money in their account. For singers, they will be the person that supplies the songs for the Tune Source project. Besides, when the Tune Source project gets any music, they must be allowed by the owner of that music. In addition, they can buy the copyright of that music. To sum up, the users they can access the Tune Source website to register an account and log in, they can listen to music online, download music, buy gift card music when their account have money. On the other hand, the singers will the person that supplies songs for the Tune Source and the Tune Source may buy the copyright of them.

1.3. Identify FRs and NFRs of Tune Source Project

1.3.1 Define FR and NFR

**FR:** A Functional Requirement (FR) is a description of the services that the software must provide. Describes a software system or its components. Functions are nothing more than inputs to a software system, its actions, and outputs. This could be a calculation, data manipulation, business process, user interaction, or some other specific function that defines what the system is expected to do. Functional requirements in software engineering are also called functional specifications (Martin, 2022)

**NFR:** Non-Functional Requirements (NFRs) are a set of specifications that describe the operational capabilities and limitations of a system and seek to enforce its functionality. These are basically requirements that describe good performance, including speed, security, reliability, data integrity, and so on (altexsoft, 2022)

1.3.2 FRs and NFRs of Tune Source Project

**FR:** For the Tune Source project, the functional requirement (FR) is to describe a software system and its components. The software of the Tune Source project is performing the listen to music online website with some functions such as the user can register an account and log in with that account. In addition, users can download music, listen to music online, and share gift card music with the constraints that users must have an account and that account must have money. This project will use the Symfony framework comes with PHP languages programming to develop the project.

**NFR:** For the Tune Source project, the non-functional requirement is to perform checking about performance, including speed, security, reliability, and data integrity in the project. Checking the specifications such as the server of the system while performing the project. Besides, identify about operational capabilities and limitations of a system, and check whether the specifications used in the project are suitable or not.

FR and NFR have a relationship intimately with each other. For a project that can be well performed, the FR and NFR in the project cannot lack. Before performing a project, the FR is the first work that we need to perform, we need to collect requirements from the users to know in the project what need we to do, so FR in a project is important. Besides, NFR cannot lack in a project, when we collect the FR from the users, the next work that we need to do is to check the system and specifications whether it is suitable for the project or not. When we perform a project but we ignore one of two of these steps, the project will get lots of risks. For example, if we don't perform functional requirements (FR) from users, the project may be wrong about function and how we know the project will perform which is the function, whether the function that we performing is suitable for the project or user's requirements. On the other hand, non-functional requirements (NFR) too. If we don't perform this, how do we know whether the system and specifications are suitable for the project and the risk is given when we use this technology for the project? To perform a project, we need to get requirements from users, and from that, we will consider the specifications to apply for the project. To sum up, these two functions have a relationship intimately and are supported in a project.

2. Discuss the technique(s) that would use to obtain the requirements

There are some techniques to collect the functional requirements of users such as interview, observation, and JAD

* **Interview technique:** Interviews are the initial requirements-gathering technique. This can be done through a meeting, phone call, or email. In this requirement-gathering technique, project managers interview stakeholders to obtain requirements. Checklists and surveys can be prepared, and project managers can freely express their expectations of the project to stakeholders. Project managers record and store requirements received from project stakeholders (masterofproject, 2022)
* **Observation technique:** A requirements gathering technique called observation involves observing potential users of the product to identify their requirements. For example, consumers can be observed to determine the user experience and most used features of an e-commerce shopping website. You can identify or prioritize project needs based on the steps consumers take (masterofproject, 2022)
* **Joint Application Development (JAD):** Collaborative application development is primarily used for gathering, designing, and developing software requirements. This method consists of holding a series of joint meetings, known as JAD sessions, which can last from several hours to several weeks. The picture 1 is the stages of JAD (Zehra, 2022)

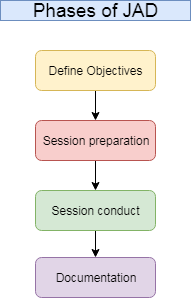
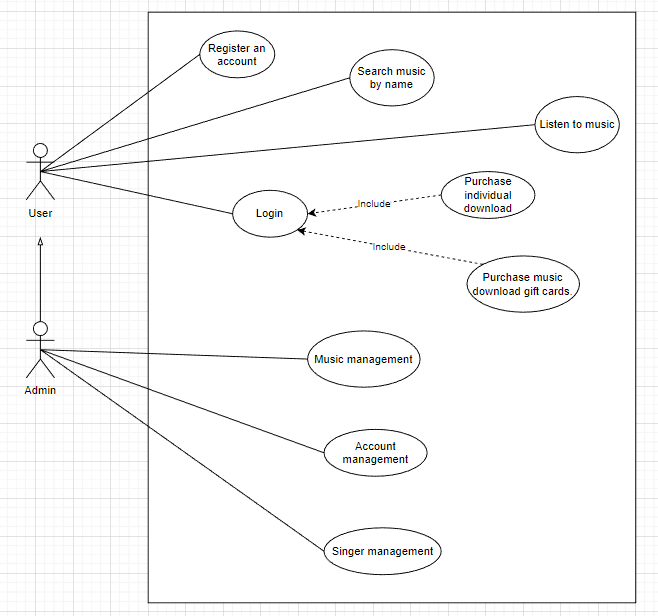


Image 1. the stages of JAD (study, 2021)

For the Tune Source project, I will choose the interview technique to perform collection requirements from users. I will perform some meet to talk about the project and get ideas from stakeholders. In addition, I will send emails and call the phone users to perform the surveys and collect their expectations of the project stakeholders. To sum up, in the Tune Source project, I will choose the interview technique to perform surveys and collect requirements from users and stakeholders.

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

1. Use case diagram



2. Use Case specification for 2 Use cases

2.1. Account management use case

2.2.

3. Context Diagram for the whole system

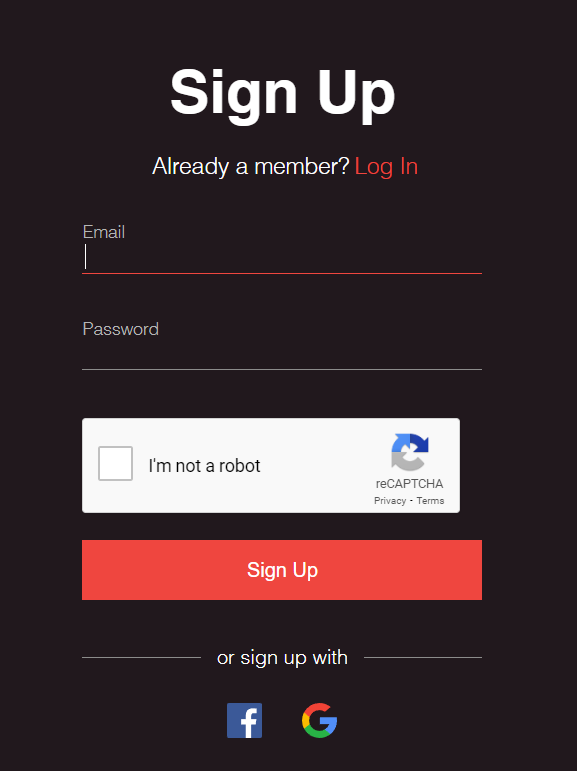
4. Data Flow Diagram – Level 0 for the whole system

5. ERD for the whole system

III. Explain how user and software requirements have been addressed

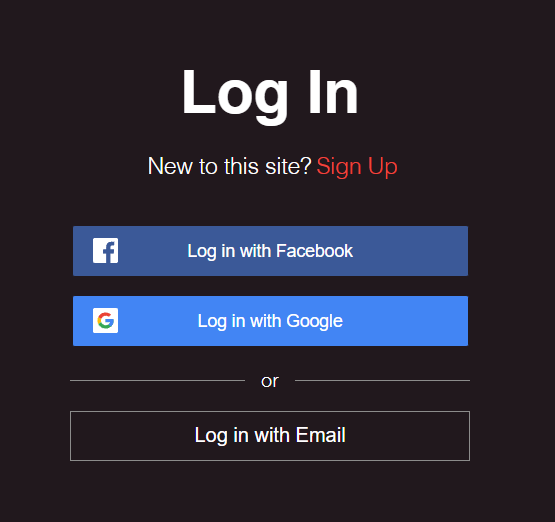
1. Register function

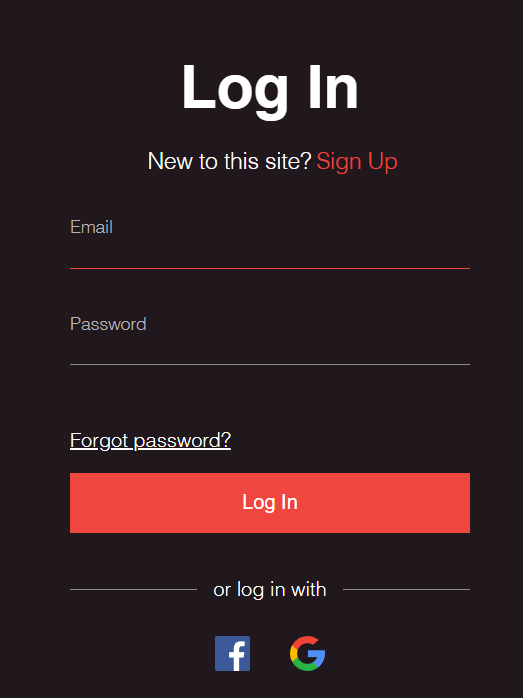
User can register an account by Email and Facebook. When the user registers an account means that they subscribed.



2. Login function

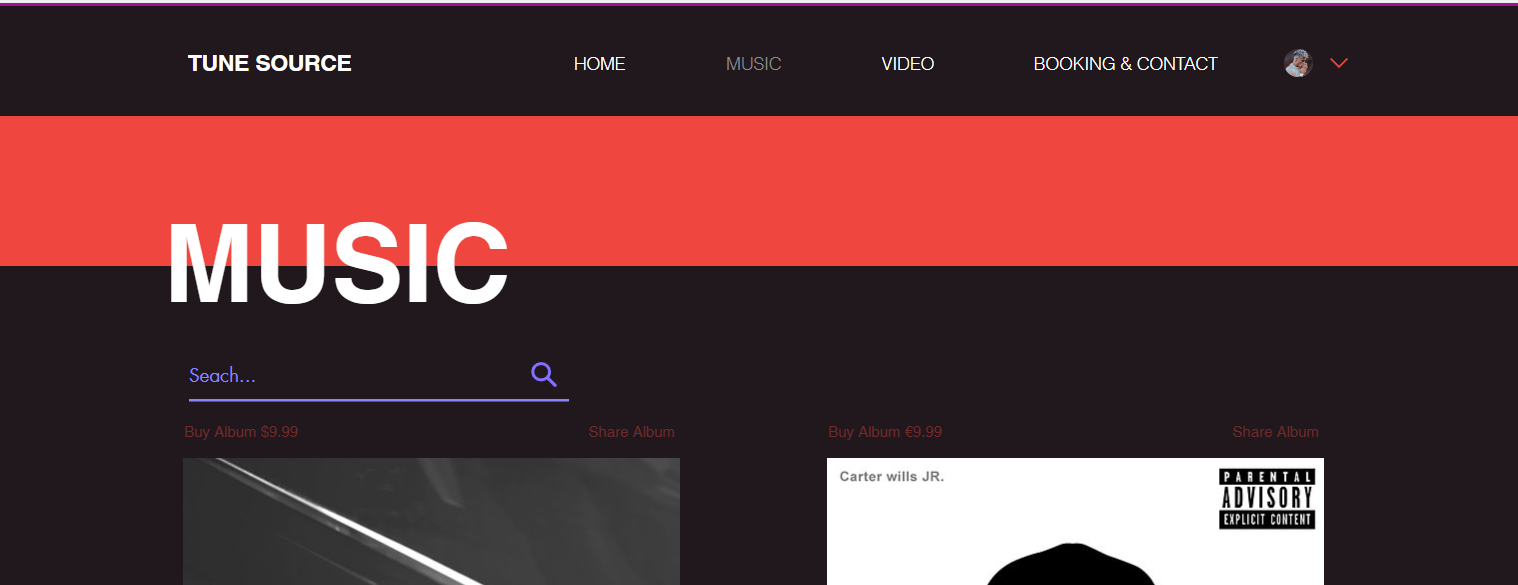
User can login with the account that they registered.





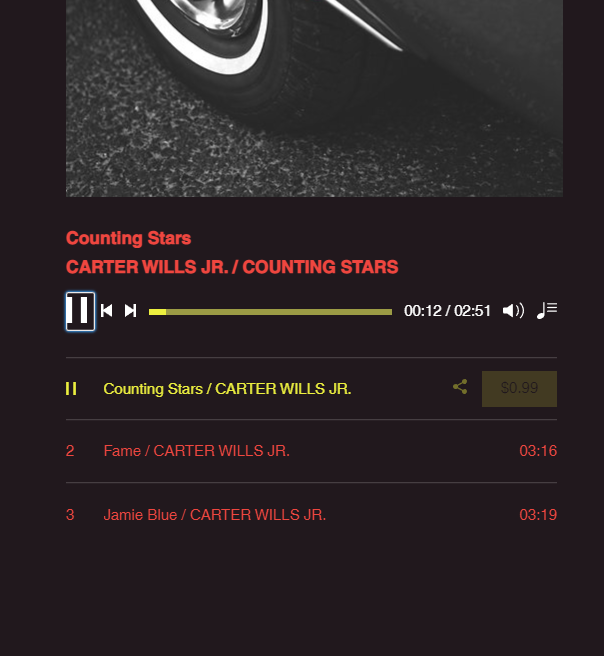
3. Search music

User can text the name of music in search text and click on search button to find the music that they want to search.



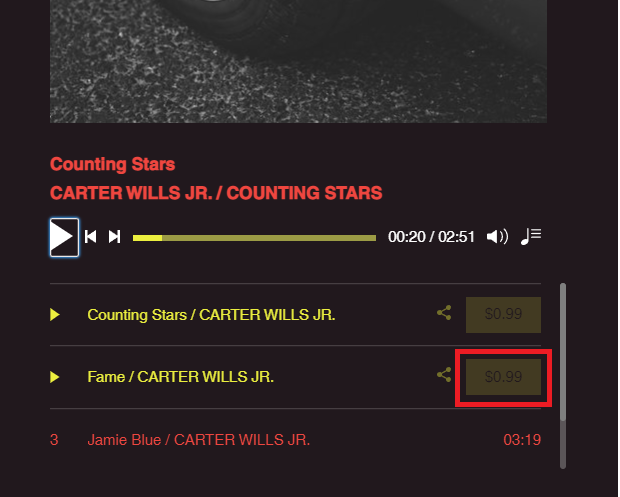
4. Listen to music

There are the list of music and user can listen to music online by click on the music that they want to listen. The picture is a description about listening to music online.



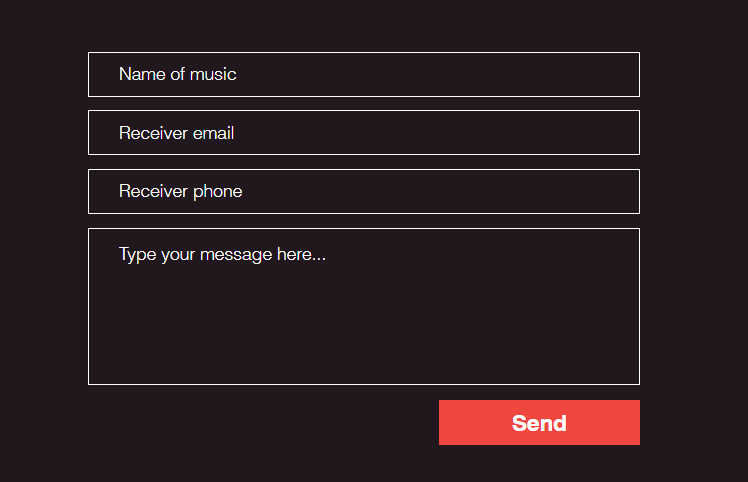
5. Purchase individual downloads

There is a price of music in each music and the user can purchase individual downloads by clicking on the price of each music in the list of music. The picture is a description about the purchase individual download function.



6. Purchase music download gift cards

User can share gift card music by fill in the form with fields such as “Name of music”, “Receiver email”, “Receiver phone”, and “Type your message here”, then they must click on “Send” button to send the gift card music that they want to share to the person that they want to share. The picture is a description about share gift card music function.



IV. Analyze how software requirements can be traced throughout the software lifecycle

V. Discuss two approaches to improving software quality

VI. Suggest two software behavioral specification methods and illustrate their use with an example

VII. M6 Differentiate between a finite state machine (FSM) and an extended-FSM, providing an application for both

VIII. Critically evaluate how the use of the function design paradigm in the software development lifecycle can improve software quality

IX. Present justifications of how data driven software can improve the reliability and effectiveness of software

X. Conclusion