UNIVERSITY of GREENWICH

Alliance with FPT Education

ĐẠI HỌC GREENWICH (VIỆT NAM)

Liên kết với Tổ chức Giáo dục FPT

# ASSIGNMENT

# SOFTWARE DEVELOPMENT LIFE CYCLE REPORT 2

**Student performance:** TRAN QUANG HUY

**ID:** GCD18457

**Class:** GCD0822

**Teacher:** NGUYEN DUY NGHIEM

# ASSIGNMENT 1 FRONT SHEET

| | | | |
|---|---|---|---|
| **Qualification** | **BTEC Levels 4 and 5 Higher Nationals in Computing** | | |
| **Unit number and title** | **Unit 9: Software Development Lifecycles** | | |
| **Assignment due** | | **Assignment submitted** | 2 |
| **Learner's name** | Tran Quang Huy | **Assessor name** | Nguyen Duy Nghiem |

**Learner declaration:**

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

| | | | |
|---|---|---|---|
| **Learner signature** | Huy | **Date** | |

**Grading grid**

| P | M | D |
|---|---|---|
| | | |

| | |
|---|---|
| **Assignment title** | **Assignment 2: Undertake a Software Development Lifecycle** |

| Learning Outcomes and Assessment Criteria | | |
|---|---|---|
| **Pass** | **Merit** | **Distinction** |
| **LO3** Undertake a software development lifecycle | | **D3** Critically evaluate how **the** use of the function design paradigm in the software development lifecycle can improve software quality. |
| **P5** Undertake a software investigation to meet a business need.<br><br>**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.<br><br>**M4** Discuss two approaches to improving software quality. | |
| LO4 Discuss the suitability of software behavioural design techniques | | **D4** Present justifications of how data driven software can improve the |
| **P7** Explain how user and software requirements | **M5** Suggest two software behavioural specification | |

| | | |
|---|---|---|
| have been addressed. | methods and illustrate their use with an example.<br><br>**M6** Differentiate between a finite state machine (FSM) and an extended-FSM, providing an application for both. | reliability and effectiveness of software. |

TRƯỜNG ĐẠI HỌC FPT

| Summative feedback: 5.1 | | | |
|---|---|---|---|
| **Assessor's signature** | | **Date** | |

# Table of Contents

**INTRODUCTION** ........ **1**

**PART 1: UNDERTAKE A SOFTWARE DEVELOPMENT LIFECYCLE** ........ **2**

**1. Undertake software investigation to meet a business need.** ........ 2

**2. Use appropriate software analysis tools/techniques to carry out a software investigation and create supporting documentation** ........ 6

**3. Analyse how software requirements can be traced throughout the software lifecycle.** ........ 10

**4. Discuss two approaches to improving software quality.** ........ 13

**5. Critically evaluate how the use of the function design paradigm in the software development lifecycle can improve software quality.** ........ 16

**PART 2: Discuss the suitability of software behavioural design techniques** ........ **22**

**1. Explain how user and software requirements have been addressed.** ........ 22

**2. Suggest two software behavioural specification methods and illustrate their use with an example.** ........ 31

**3. Finite state machine (FSM) and providing an application.** ........ 35

**4. Present justifications of how data driven software can improve the reliability and effectiveness of software.** ........ 38

**CONCLUSION** ........ **41**

**References** ........ **42**

# TABLE OF FIGURE

Figure 1 - Tune Source's Survey Questionnaire .....4
Figure 2 - Use case software investigation Tune Source .....6
Figure 3 - Flowchart software investigation Tune Source .....7
Figure 4 - Documentation types (altexsoft, 2017) .....8
Figure 5 - Project Documentation (altexsoft, 2017) .....8
Figure 6 - An alternative view of Software Requirements Traceability (wikipedia, n.d.) .....10
Figure 7 - A level 0 data flow diagram .....17
Figure 8 - A level 1 data flow diagram .....18
Figure 9 - A level 2 data flow diagram (Base on: (lucidchart, n.d.)) .....19
Figure 10 - Structure decomposition (Context Design) of Tune Source .....20
Figure 11 - Sequence diagram login in Tune Source .....21
Figure 12 - Tune Source website main menu .....23
Figure 13 - Tune Source login .....24
Figure 14 - Tune Source subscription account .....25
Figure 15 - Tune Source listening to music and download .....26
Figure 16 - Tune Source main login (Application) .....27
Figure 17 - Tune Source main menu (Application) .....28
Figure 18 - Music store and songs/albums recommended (Application) .....29
Figure 19 - Listening to music and purchase music (Application) .....30
Figure 20 - Flowchart function download music .....31
Figure 21 - Flowchart function subscriptions .....33
Figure 22 - Example finite state machine (wikipedia, n.d.) .....35
Figure 23 - FSM for subscribed customer .....36
Figure 24 - FSM for unsubscribed customer .....37

# TABLE OF TABLES

Table 1 - Tune Source website functional ........................................................................................3
Table 2 - Requirement of Tune Source project .....................................................................................9
Table 3 - Example Requirements Traceability Matrix (guru99, n.d.).........................................................11
Table 4 - Requirement Traceability Matrix in Tune Source ......................................................................12
Table 5 - Effectives Data Drive to Tune Source ...........................................................................39

# INTRODUCTION

In building an information system, there should be a plan to manage and control the overall project, from the beginning to the end, to achieve the estimated goals and objectives. In order to ease the management of the project, including the analysis, design, and other development activities, we need a guidance or framework that can help in coordinating the work flow of the project. The framework that is needed is the one that identifies all activities required to build, launch, and maintain an information system. The framework also should include the part of analysis, design, programming, testing, and maintaining the system until we can launch and implement the system. The framework is called System Development Life Cycle (SDLC).

Although SDLC requires many various activities or processes, there are several core processes that are required in order to build and develop the project's output, which is the system. The core processes in developing a new system are as the following:

- Identify the problems or needs and obtain approval to proceed
- Plan and monitor the project about what to do, how, and who does it
- Discover and understand the details of the problems or the needs
- Design the system components to solve the problems
- Build, test, and integrate system components
- Complete system tests and deploy the solution

# PART 1: UNDERTAKE A SOFTWARE DEVELOPMENT LIFECYCLE

## 1. Undertake software investigation to meet a business need.

### 1.1. Tune Source's website.

#### 1.1.1. Overview Tune Source website.

The goal is to build the website of Tune Source towards selling music products to customers. The use of this website will gradually replace traditional stores. In addition, through the economic analysis of Tune Source, the main revenue is the sale of products through the website channel. So to build a website is a top requirement for developing the company as well as increasing profits.

Currently the website is being built on a PC-based platform and a basic application, but in the future Tune Source will apply.

Progressive Web App (PWA). With this technology, users can easily use the website as an application on the phone without having to install anything.

Based on the situation of Tune Source, functional requirements and website design must meet the following these requirements:

- Allows users to register for an account
- Allow users to login with personal account
- Customer can get subscription to listening and download digital music & gift card.
- The Tune Source website also sell CD online.
- News, events related to music, singers.
- Ranking the most popular songs from time to time

### 1.1.2. Tune Source's website functional.

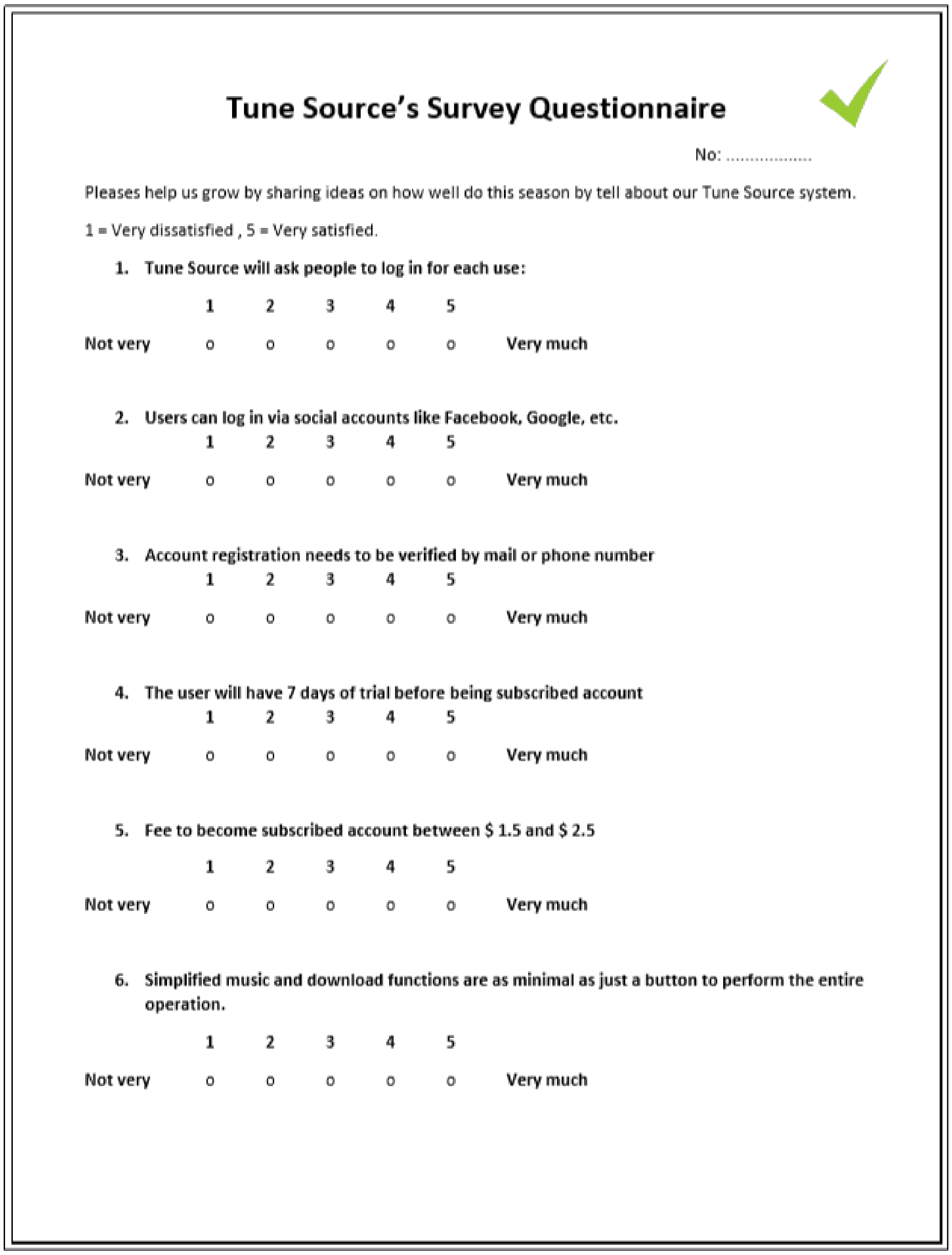
| Business need | Description | Functional |
|---|---|---|
| Selling digital music | Selling digital music through the website when the user is subscribed account. | Sell songs module, electronic wallet module. |
| Music download | Subscribed account can download every music of Tune Source that they want. | Check subscribed account module, music download module. |
| Listening to music | Subscribed account or new account (7 days trial) can listening to music. | Play music online module, check subscribed account module, trial account module. |
| Searching music | User can search every music at Tune Source without listening and downloading expect subscribed account and trial account. | Searching music module, check subscribed account module, trial account module. |

**Table 1 - Tune Source website functional**

- Sell songs module: This module link with electronic wallet module will let Tune Source sell digital music.
- Electronic wallet module: This module is used to show how customer use their payment to buy digital music
- Check subscribed account module: This module will make sure that user is using Tune Source as listening and downloading is subscribed account.
- Trial account module: This module will allow new customer can listen to music 7 days, after 7 days, the IP will be locked.
- Music download module: This module will allow user download music as *.mp3, *.pcm, *.wav, *.aac, *.ogg, *.flac . Before download music, this module will make sure that Check subscribed account or trial account module confirm that account first.
- Searching music module: This module link with database system, allow customer can search songs by name, artic, bans, etc.

## 1.2. Software Investigation to meet a business need.

In order to meet the requirements of Tune Source, the survey tools were applied, through interviewing and monitoring customer behavior, the questionnaire was used as the best option to meet the business requirements. Below is a sample question:



**Tune Source's Survey Questionnaire**

No: .................

Pleases help us grow by sharing ideas on how well do this season by tell about our Tune Source system.

1 = Very dissatisfied , 5 = Very satisfied.

1. **Tune Source will ask people to log in for each use:**

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Not very | o | o | o | o | o | Very much |

2. **Users can log in via social accounts like Facebook, Google, etc.**

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Not very | o | o | o | o | o | Very much |

3. **Account registration needs to be verified by mail or phone number**

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Not very | o | o | o | o | o | Very much |

4. **The user will have 7 days of trial before being subscribed account**

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Not very | o | o | o | o | o | Very much |

5. **Fee to become subscribed account between $ 1.5 and $ 2.5**

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Not very | o | o | o | o | o | Very much |

6. **Simplified music and download functions are as minimal as just a button to perform the entire operation.**

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Not very | o | o | o | o | o | Very much |

Figure 1 - Tune Source's Survey Questionnaire

Through the questionnaire sent to 55 users who experienced the Tune Source system, there were 52 users who answered the questionnaire and had 50 questionnaires valid. General statistics on the purpose achieved after using the question system are as follows:

This software is a comprehensive system that includes retail and live streaming for music. It is important to identify key stakeholders:

- Music store: is the person that the question will target. This includes online and on-site stores. They will be asked primarily about access to the system and to limit their access.
- Customer: Online and off site. For customers, it is important to provide a smooth music buying / playing experience. This requires strong database and connectivity.
- Artist / Manufacturer: Will receive commission based on the number of copies sold. Contracts are completed before their products are available on the system, they talk less about how the software will work.

From reviewing the needs of stakeholders, the questions therefore aim to attract stores. Comments on system security, frequency of access and the degree of authority that stores need to access. system.

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

### 2.1. Analysis function of Tune Source

There are some function that Tune Source need have and should improve in the future:

- Download music
- Listening to music
- Searching music
- Registration
- Login
- Download gift card
- Subscriptions
- Buy CD

### 2.2. Tool/Techniques to carry out a software investigation.

#### 2.2.1. Use case:

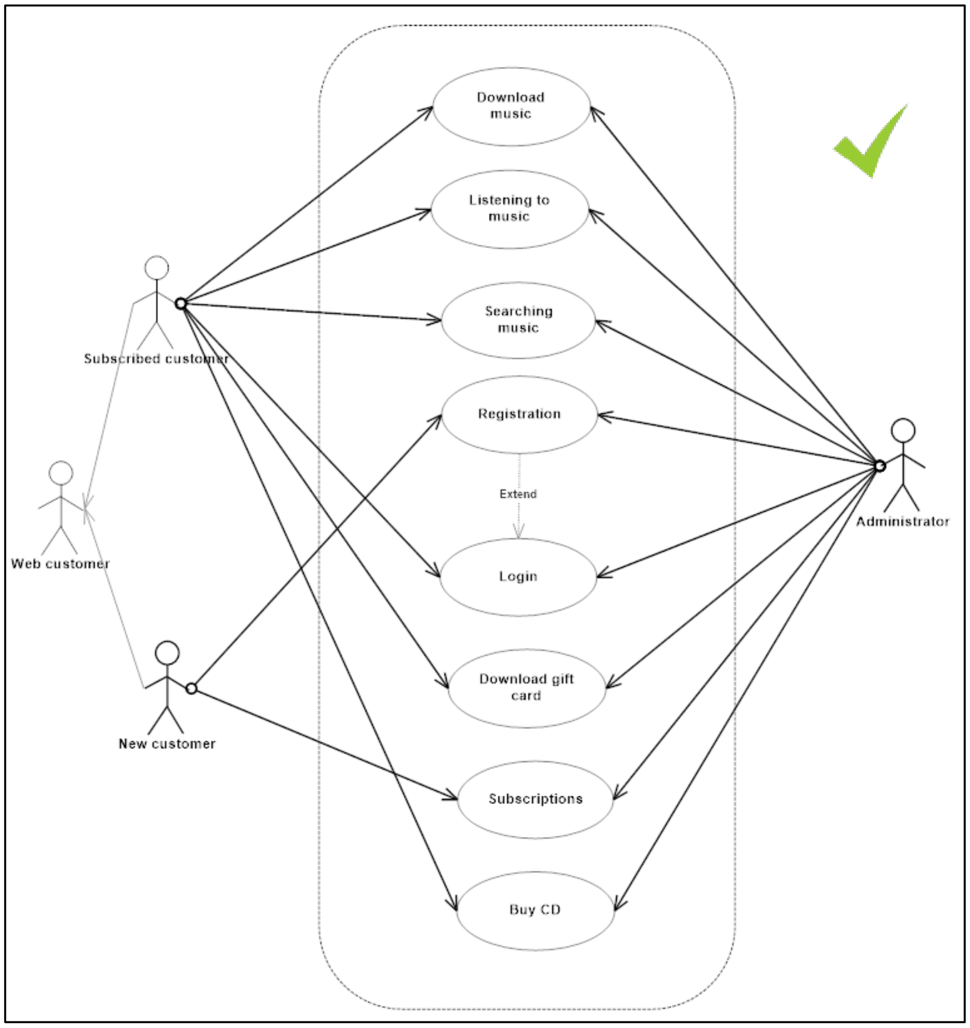


Figure 2 - Use case software investigation Tune Source

### 2.2.2. Flowchart:

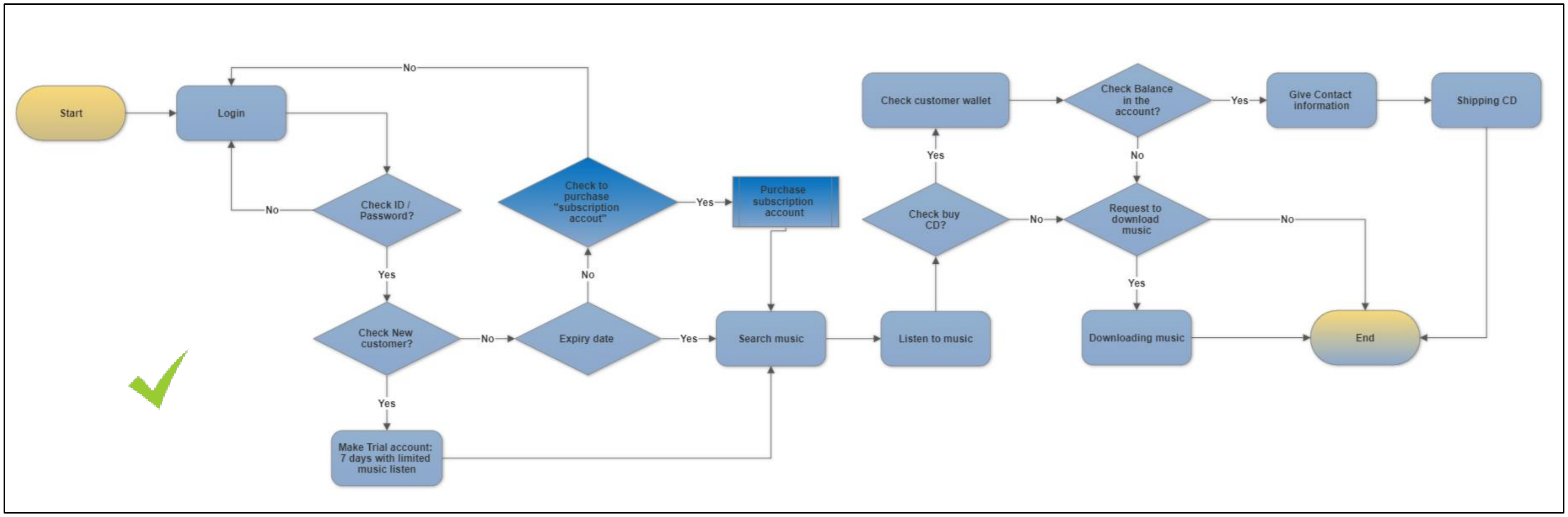


Figure 3 - Flowchart software investigation Tune Source

### 2.3. Supporting documentation

Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it and may mean different things to people in different roles. (wikipedia, n.d.)

Base on Tune Source that will have support documentation include: Requirements, Architecture/Design,

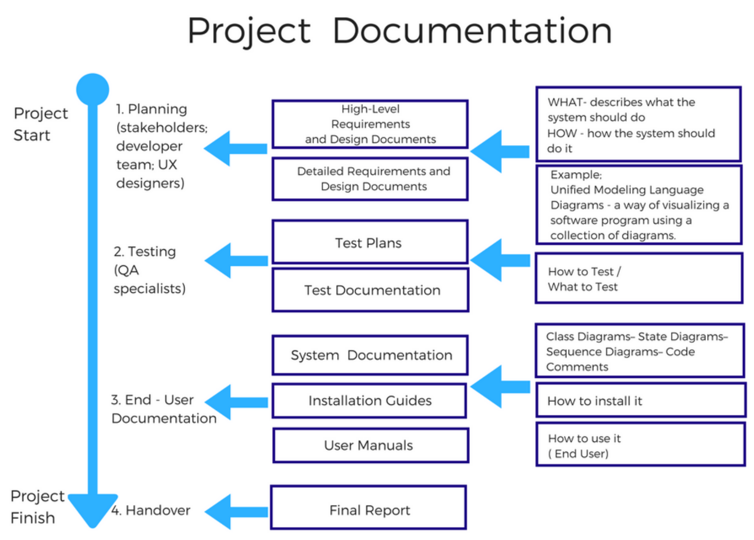

Project Documentation
Project Start
1. Planning (stakeholders; developer team; UX designers)
High-Level Requirements and Design Documents
Detailed Requirements and Design Documents
WHAT- describes what the system should do
HOW - how the system should do it
Example; Unified Modeling Language Diagrams - a way of visualizing a software program using a collection of diagrams.
2. Testing (QA specialists)
Test Plans
Test Documentation
How to Test / What to Test
System Documentation
Class Diagrams- State Diagrams- Sequence Diagrams- Code Comments
3. End - User Documentation
Installation Guides
How to install it
User Manuals
How to use it ( End User)
Project Finish
4. Handover
Final Report

Figure 5 - Project Documentation (altexsoft, 2017)

Technical, End user and marketing.

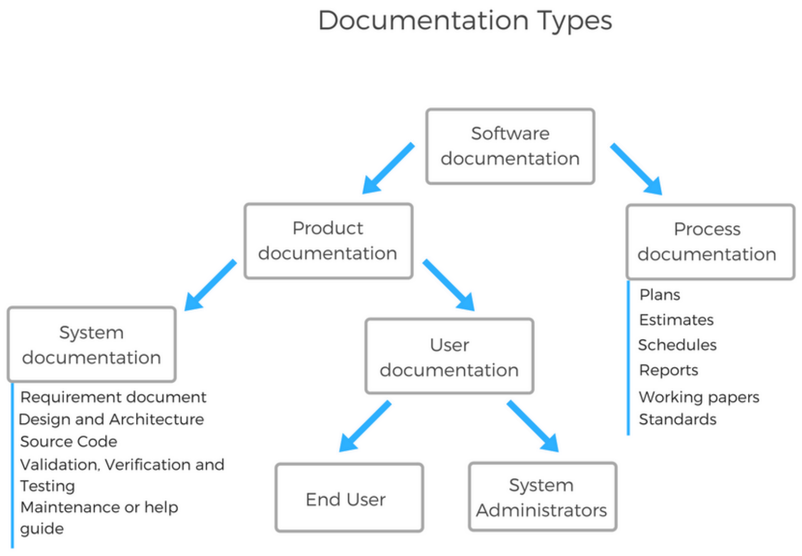


Figure 4 - Documentation types (altexsoft, 2017)

✓ - Requirement of Tune Source project:

| **Tune Source company** | **Content** |
|---|---|
| Business value | $757,500 in deals with individual music downloads<br>$950,000 in deals from client memberships<br>$205,000 in extra in-store or site CD deals<br>$153,000 in deals with music download gift vouchers |
| Business requirement | • Look for music in our advanced music file.<br>• Tune in to music tests.<br>• Buy individual downloads at a settled charge for each download.<br>• Build up a client membership account allowing boundless downloads for a month to month charge.<br>• Buy music download gift vouchers. |
| Project’s cost | Approximately 30% of total value: $700,000. |

**Table 2 - Requirement of Tune Source project**

## 3. Analyse how software requirements can be traced throughout the software lifecycle.

### 3.1. Overview software requirements can be traced throughout the development of Tune Source

Base on software development life cycle (SDLC), traceability primarily refers to the traceability of requirements throughout the application development process. This helps ensure that distributed software meets all requirements and thus helps prevent errors. Traceability is a major regulation of effective software requirements management. (Sehlhorst, n.d.)

Traceability in the context of the software development life cycle means the ability to ensure that:

- All requirements are determined
- Accepted requirements are divided into development and test tasks, with references to each other
- Source code is reviewed based on acceptance criteria (Baseline) during development
- Changes at any time during the development life cycle are monitored and collaboration is guaranteed (through Change management).
- Tests made and released are ready to deploy on time.

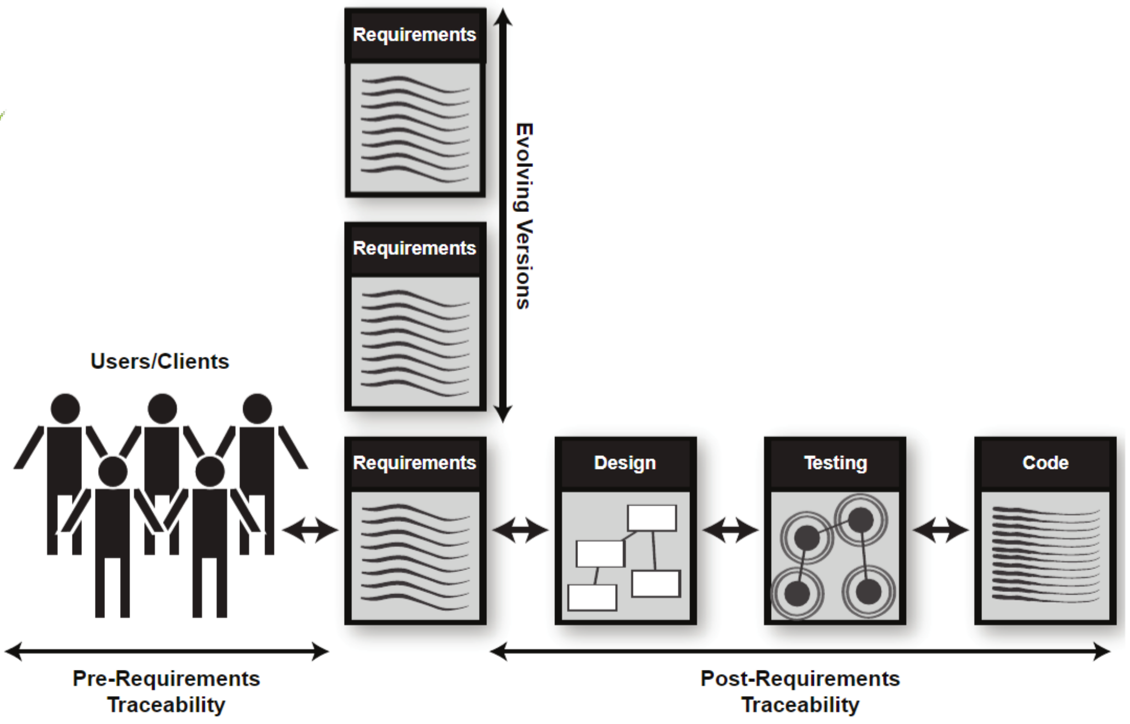
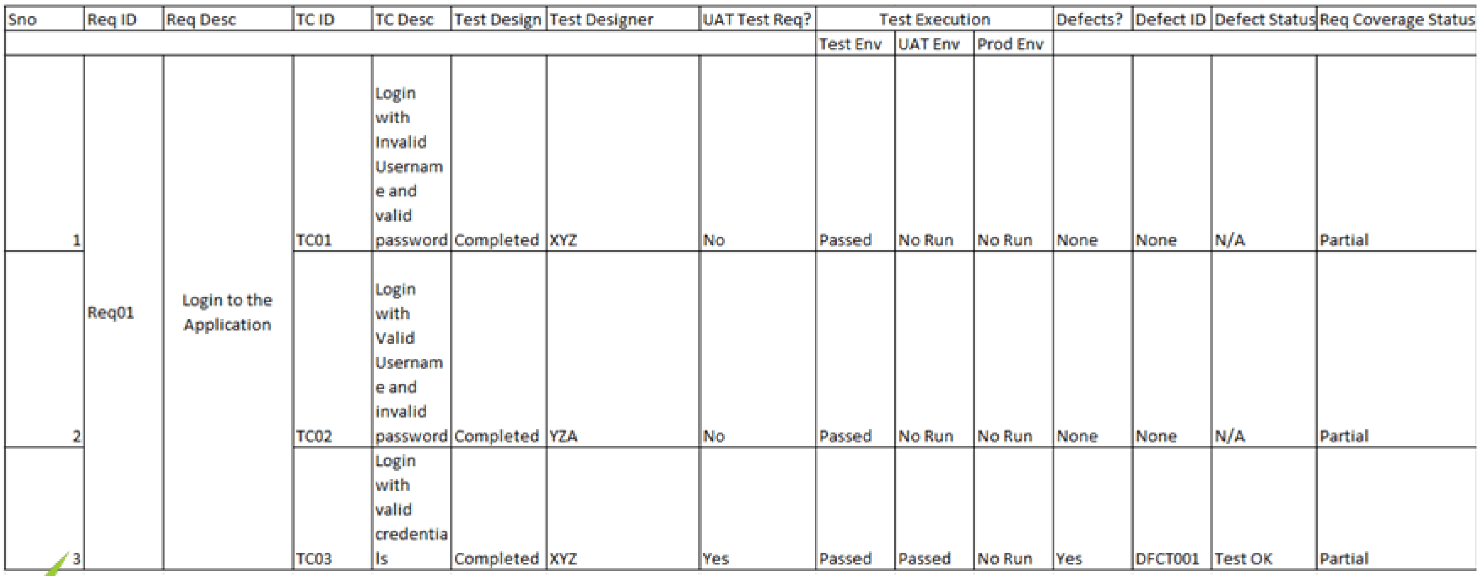


Figure 6 - An alternative view of Software Requirements Traceability (wikipedia, n.d.)

**Requirement Traceability Matrix** or **RTM** captures all requirements proposed by the client or software development team and their traceability in a single document delivered at the conclusion of the life-cycle.

In other words, it is a document that maps and traces user requirement with test cases. The main purpose of Requirement Traceability Matrix is to see that all test cases are covered so that no functionality should miss while doing Software testing.

**RTM** will include various parameters: Requirement ID, Requirement Type and Description, Test cases with descriptions, Test design status, Test designer:



| Sno | Req ID | Req Desc | TC ID | TC Desc | Test Design | Test Designer | UAT Test Req? | Test Execution | | | Defects? | Defect ID | Defect Status | Req Coverage Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Test Env | UAT Env | Prod Env | | | | |
| 1 | Req01 | Login to the Application | TC01 | Login with Invalid Username and valid password | Completed | XYZ | No | Passed | No Run | No Run | None | None | N/A | Partial |
| 2 | | | TC02 | Login with Valid Username and invalid password | Completed | YZA | No | Passed | No Run | No Run | None | None | N/A | Partial |
| 3 | | | TC03 | Login with valid credentials | Completed | XYZ | Yes | Passed | Passed | No Run | Yes | DFCT001 | Test OK | Partial |

Table 3 - Example Requirements Traceability Matrix (guru99, n.d.)

The required traceability matrix can:

- Display the required range in the number of test cases
- Design status as well as execution status for specific test cases
- f there is any Acceptable user test performed by the user, the UAT status can also be recorded in the same matrix.
- Related defects and current state can also be mentioned in the same matrix.

**Baseline:** is a reference point in the software development life cycle marked by the formal completion and approval of a pre-defined work set. The goal of the baseline is to reduce the vulnerability of the project to uncontrolled change by modifying and changing the official control of the various major products (configuration items) at the Important points in the development life cycle. (hashe, n.d.)

A baseline will provide:

- Progress is measurable in the system development life cycle
- Basis for controlling changes in the next project stages
- A stable reference for future work
- Intermediate and final points to assess the suitability for the purpose of project work products.

**Change management:** The change management process in systems engineering is the process of requesting, determining attainability, planning, implementing, and evaluating of changes to a system. Its main goals are to support the processing and traceability of changes to an interconnected set of factors. The process identifies the changes' origin, defines critical project decision points, and establishes project roles and responsibilities. Change management is not an isolated process. The project team must be clear on what, when, how, and why to carry it out. (Crnkovic, 2003)

**Tools:** Today there are many tools dedicated to providing the most integrated method of managing software requirements, including the disadvantages of traditional methods:

- Allow requests to be managed in the database
- Automate traceability functions: create electronic links between parent and child requests or between test cases and requests
- Electronic base document creation, version control and change management
- Allows creation of specification documents by exporting required data into a standard document application

The management tools required advantages for organizations in terms of increasing business value, reducing budget problems. In today's developed world, it is essential to automatically monitor changes in requirements at every step. Auditing requirements are made more efficient with the participation of tools.

### 3.2. Requirement Traceability Matrix in Tune Source.

| Code | Description | User | Admin | Manager |
|---|---|---|---|---|
| Req.01 | Login, logout and registration | Add | Add, edit | Add, edit, remove |
| Req.02 | Search, download music | Use | Use, edit | Use, edit, add, remove |
| Req.03 | Customer information | Add, edit, remove | Add, edit, remove | Add, edit, remove, view |
| Req.04 | Manage system | - | View | Add, edit, remove, view |

Table 4 - Requirement Traceability Matrix in Tune Source

## 4. Discuss two approaches to improving software quality.

### 4.1. Total Quality Management (TQM)

✓

#### 4.1.1. Definition of TQM

TQM is a management approach aimed at satisfying all the customer requirements, needs and expectations using a continuous improvement approach The TQM principles can be grouped into the following practical and common sense concepts (Wang, 1996)

- Customer focus (internal and external customers)
- Leadership (management role changes to active leadership)
- Teamwork (multidisciplinary teams, include involvement of customers and suppliers)
- Continuous improvement process
- Measurement (the improvement process is based on quantitative and qualitative metrics)
- Benchmarking as a driver to improvement in a competitive environment.

TQM focuses on continually improving the ability to provide high quality products and services to customers. It suggests that any improvements made in the enterprise, be it a better design of a component or a better process of a system, will help improve the total quality of the organization. and the quality of the final product (Base on Li, (2000)).

#### 4.1.2. Improving software quality of Tune Source with TQM.

Applying TQM to regulate site development Tune Source, a process that can control software quality and productivity and choose an appropriate tool that can enhance the ability of software quality, correction, prevention, error correction and feedback.

Some TQM development activities related to Business System Planning (BSP), Quality Function Deployment (QFD) and key success factors (CSF):

- QFD has a cross function so that all departments work together to achieve the common goal of satisfying customer needs.
- CSF will develop systems for planning and control
- BSC is a method of turning strategy into action and has been successfully implemented in all types of companies worldwide.

With the integration of TQM activities, Tune Source developers can create a quality assurance plan and successfully implement all related tasks.

## 4.2. Test Driven Development (TDD)

### 4.2.1. Definition of TDD.

Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle. First, the developer writes an (initially failing) automated test case that defines a desired improvement or new function, then produces the minimum amount of code to pass that test, and finally refactors the new code to acceptable standards. In Test Driven Development (TDD) automated unit tests are written before the code is actually written. Running these tests give you fast confirmation of whether your code behaves as it should or not. (Apiumhub, n.d.)

- Add a test:
    - In TDD, each feature begins by writing a test. This test determines the function or improvement of the function based on use cases to include exceptions and requirements.
    - This is a distinct feature of TDD compared to unit testing written after the code is written, which makes the developer focus on the requirements before writing the code.
- Run all tests and recognize the problem:
    - This validates that the test harness is working correctly. The new test should fail for the expected reason.
- Write code:
    - The next step is to write some code that makes the test pass. At this point, the sole purpose of the code is to pass the test, the programmer cannot write the code beyond the function that the test checks.
- Run tests:
    - If all the test cases are now over, the programmer can be confident that the new code meets the testing requirements and does not break or reduce any existing features. If they don't, the new code must be adjusted until they do.
- Refactor code
    - The developing code base must be cleaned regularly in TDD. The new code can be moved from a convenient place to pass a test to a more logical place. Copying must be removed.
    - By continually rerunning test cases at each refactoring stage, the developer can be confident that the process does not change any existing functionality. (Beck, Kent (1999))

Advantages of TTD development:

- Writing the tests first requires you to really consider what do you want from the code
- Fast feedback
- Creates a detailed specification

- Reduces time spent on rework
- Less time spent in the debugger
- Identify the error/problem quickly
- Tells you whether your last change (or refactoring) has broken previously working codes
- Allows the design to evolve and adapt to your changing understanding of the problem
- Forces radical simplification of the code, you will only write codes in response to the requirements of the tests
- Forces you to write small classes focused on one thing
- Creates SOLID code
- Cleans Interface
- Maintainable, Flexible & Easily Extensible
- The resulting Unit Tests are simple and act as documentation for the code. Since Test Driven Development use-cases are written as tests, other programmers can view the tests as usage examples of how the code is intended to work
- Shortens the development Time to Market
- Increases the programmer's productivity
- Cuts development costs
- Improves quality
- Reduces bugs
- Test Driven Development gives programmers the confidence to change the larger architecture of an application when adding new functionality. Without the flexibility of TDD, developers frequently add new functionality by virtually bolting it to the existing application without true integration – clearly, this can cause problems down the road

### 4.2.2. Improving software quality of Tune Source with TDD.

By implementing **TDD**, Tune Source's development will get:

- Code access through test cases instead of through mathematical assertions or prejudices
- Allow the developer team to focus on the task at hand because the first goal is to take the test. This gives developers and users the next level of code reliability.
- Limit the number of errors in the code to a large number of tests, preventing them from becoming endemic and expensive issues, avoiding long-term and tedious debugging in the project.
- Having more scalable, flexible and scalable code because systems considered under small units can be written and tested independently and integrated later.

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

**Function Oriented Design** is a top-down approach to software design in which the design of the system is decomposed into a set of interactive units, in which each unit has a well-defined function. obviously. These functional units can share information with each other by transmitting information and using information available globally.

When a program calls a function, this function will change the status of the program, which is sometimes not accepted by other modules. **Function Oriented Design** works well when the system status is not important and the program / function operates on the input instead of the status.

Due to the requirement that Tune Source system is fully functional and directed at customer input: searching for music, listening to music samples, buying music or gift cards, requesting account registration, Design implementation Functional orientation will help increase the development process of the timer:

- Data flow diagram will be responsible for presenting the basic concepts of how the entire system works. With DFD, all stakeholders will have a clear, broad vision of how the new system will look.
- Develop the system by focusing on its functions, instead of its entities or characteristics. It will reduce flexibility and effectiveness of the project.

### 5.1. Data flow design:

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

- A level 0 data flow diagram (DFD), also known as a context diagram, shows a data system as a whole and emphasizes the way it interacts with external entities. This DFD level 0 example shows how such a system might function within a typical retail business. (lucidchart, n.d.)

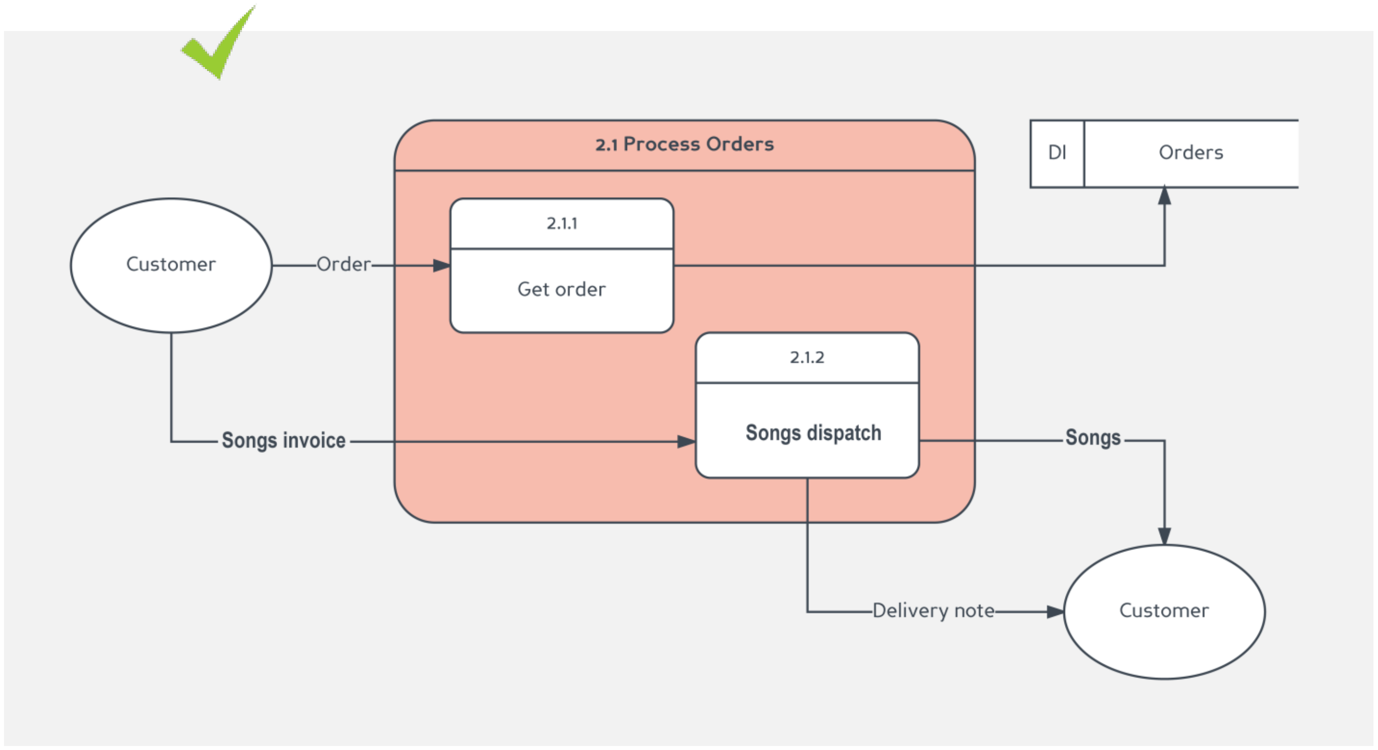


Figure 7 - A level 0 data flow diagram

A level 1 data flow diagram (DFD) is more detailed than a level 0 DFD but not as detailed as a level 2 DFD. It breaks down the main processes into subprocesses that can then be analyzed and

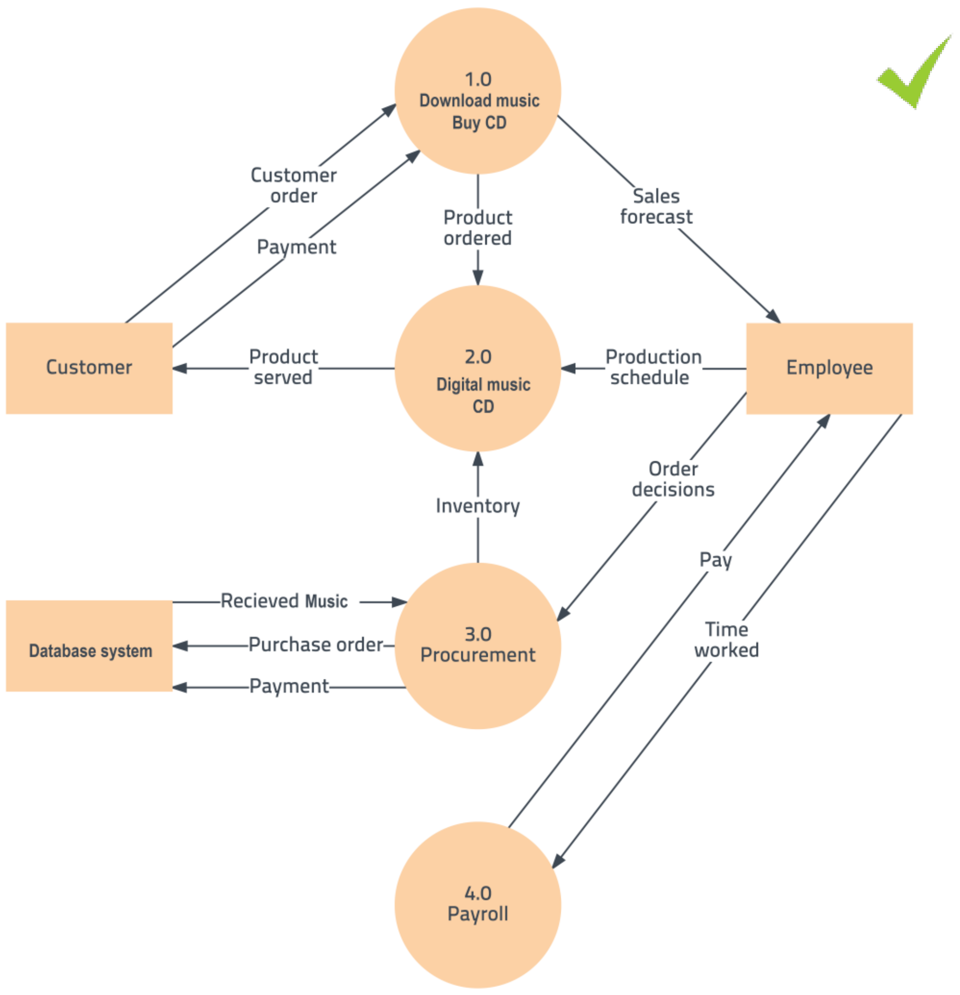


Figure 8 - A level 1 data flow diagram

improved on a more intimate level. (lucidchart, n.d.)

- A level 2 data flow diagram (DFD) offers a more detailed look at the processes that make up an information system than a level 1 DFD does. It can be used to plan or record the specific makeup of a system. (lucidchart, n.d.)

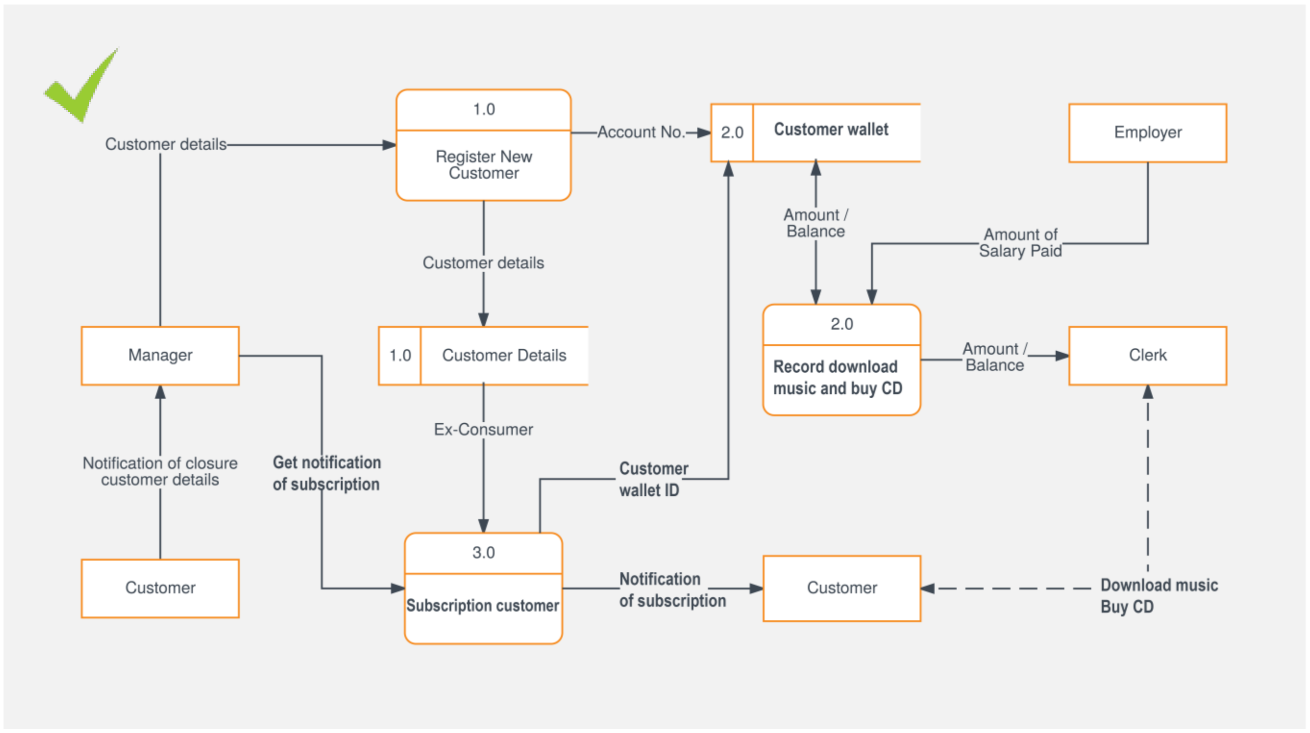


Figure 9 - A level 2 data flow diagram (Base on: (lucidchart, n.d.))

### 5.2. Structure decomposition:

Decomposition in computer science, also known as factoring, is breaking a complex problem or system into parts that are easier to conceive, understand, program, and maintain.

There are different types of decomposition defined in computer sciences:

- In structured programming, algorithmic decomposition breaks a process down into well-defined steps.
- Structured analysis breaks down a software system from the system context level to system functions and data entities
- Object-oriented decomposition, on the other hand, breaks a large system down into progressively smaller classes or objects that are responsible for some part of the problem domain.

A decomposition paradigm in computer programming is a strategy for organizing a program as a number of parts, and it usually implies a specific way to organize a program text. Usually the aim of using a decomposition paradigm is to optimize some metric related to program complexity, for example the modularity of the program or its maintainability.

Most decomposition paradigms suggest breaking down a program into parts so as to minimize the static dependencies among those parts, and to maximize the cohesiveness of each part. Some popular decomposition paradigms are the procedural, modules, abstract data type and object oriented ones. (wikipedia, n.d.)

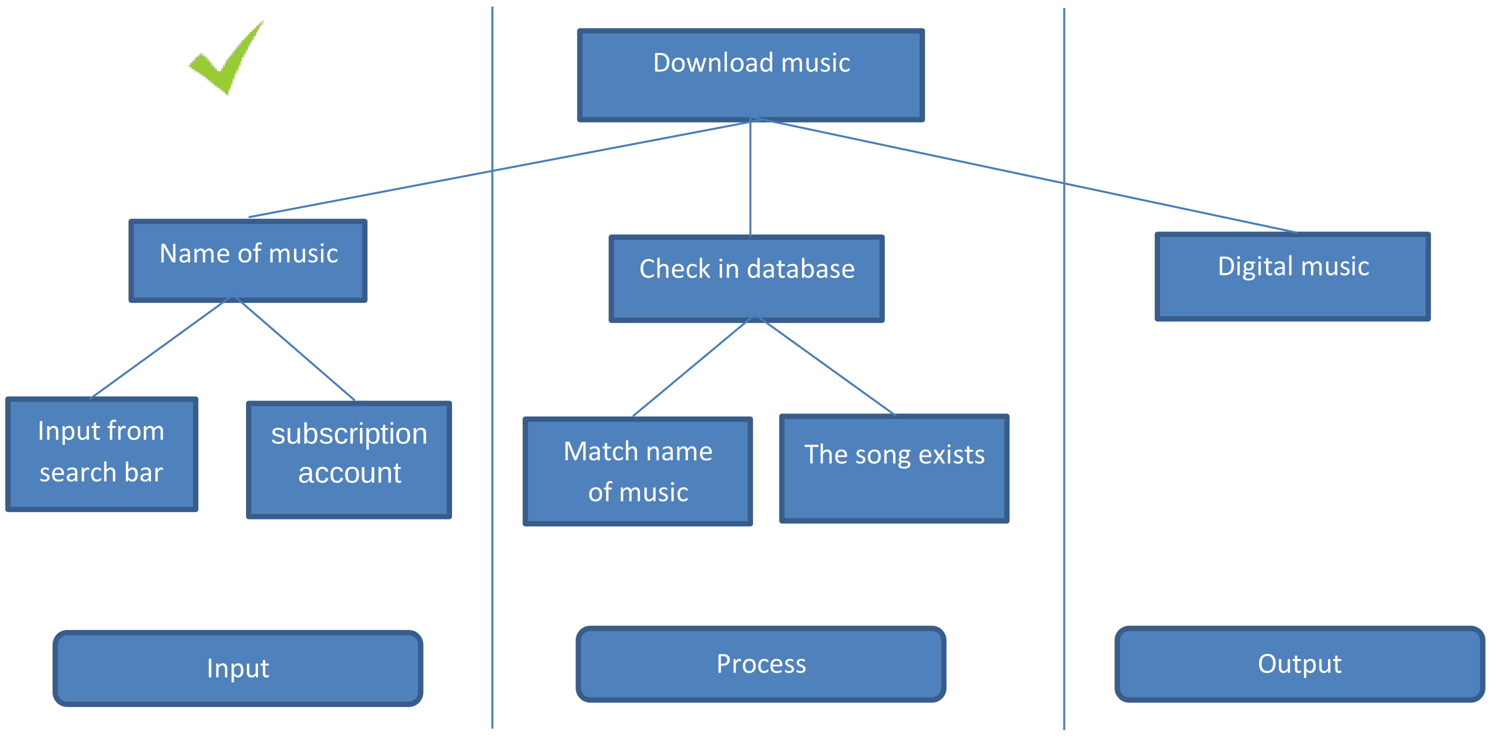


Figure 10 - Structure decomposition (Context Design) of Tune Source

### 5.3. Detailed design:

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. (wikipedia, n.d.)

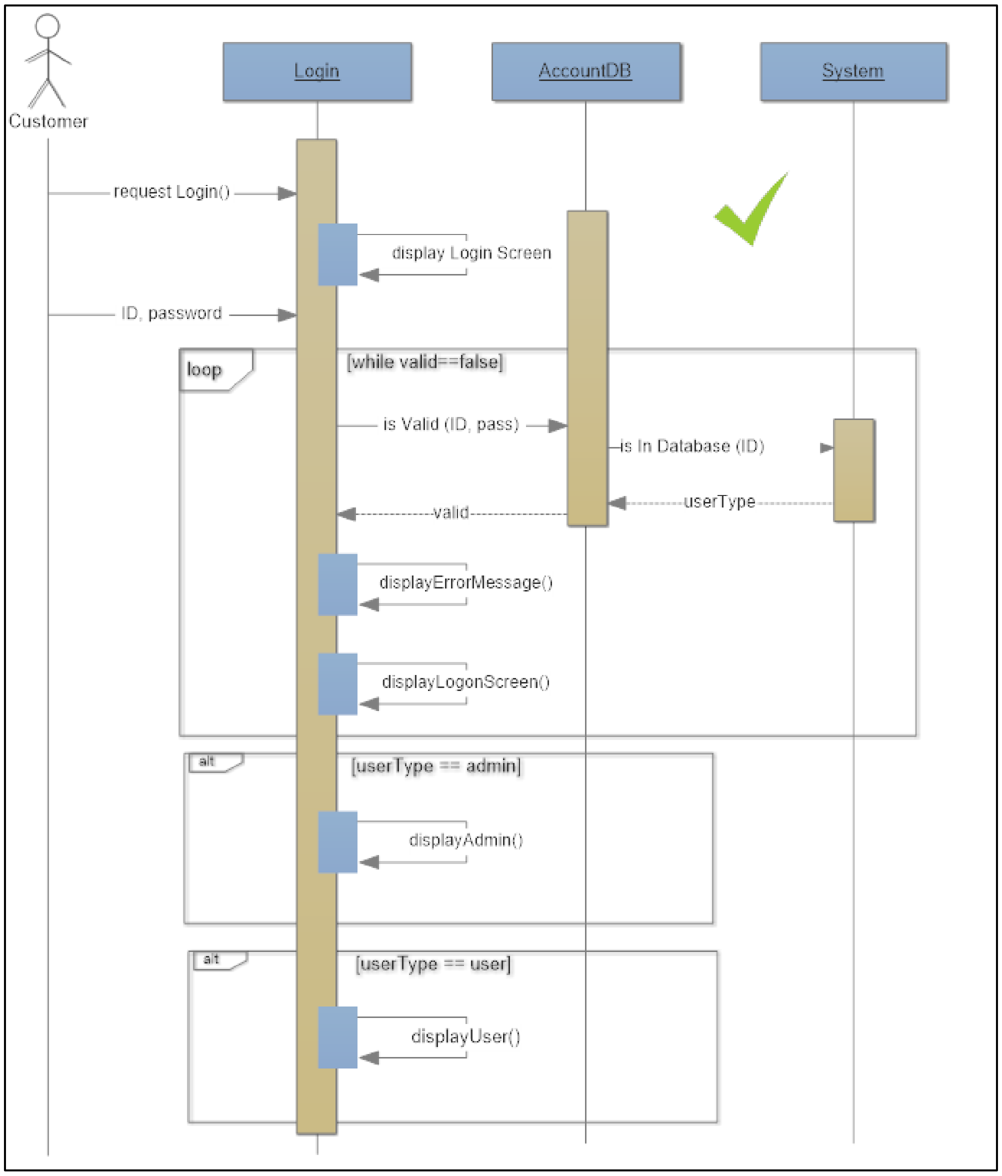


Figure 11 - Sequence diagram login in Tune Source

# PART 2: Discuss the suitability of software behavioural design techniques

## 1. Explain how user and software requirements have been addressed.

### 1.1. Overview Tune Source function's requirement.

To accomplish all of Tune Source's user requirements should be implemented:

- Main menu: To get information about Tune Source, news, login, sign up, etc.
- Login: Allow customer log in the system with account is defined, sign up, etc.
- Subscription site: If customer want to buy CD or listening to music, they must be subscribed account. This will let customer know about the amount that customers pay for become subscribed account.
- Listening and download music site: This is a main requirement which allow customer can listen to music and download it if they want.

## 1.2. Tone Source’s prototype.

### 1.2.1. Tune Source website prototype:

- **Tune Source main website**: In the main menu, customer can sign up, log In, get information about Tune Source. (The picture belong to (netz, n.d.))

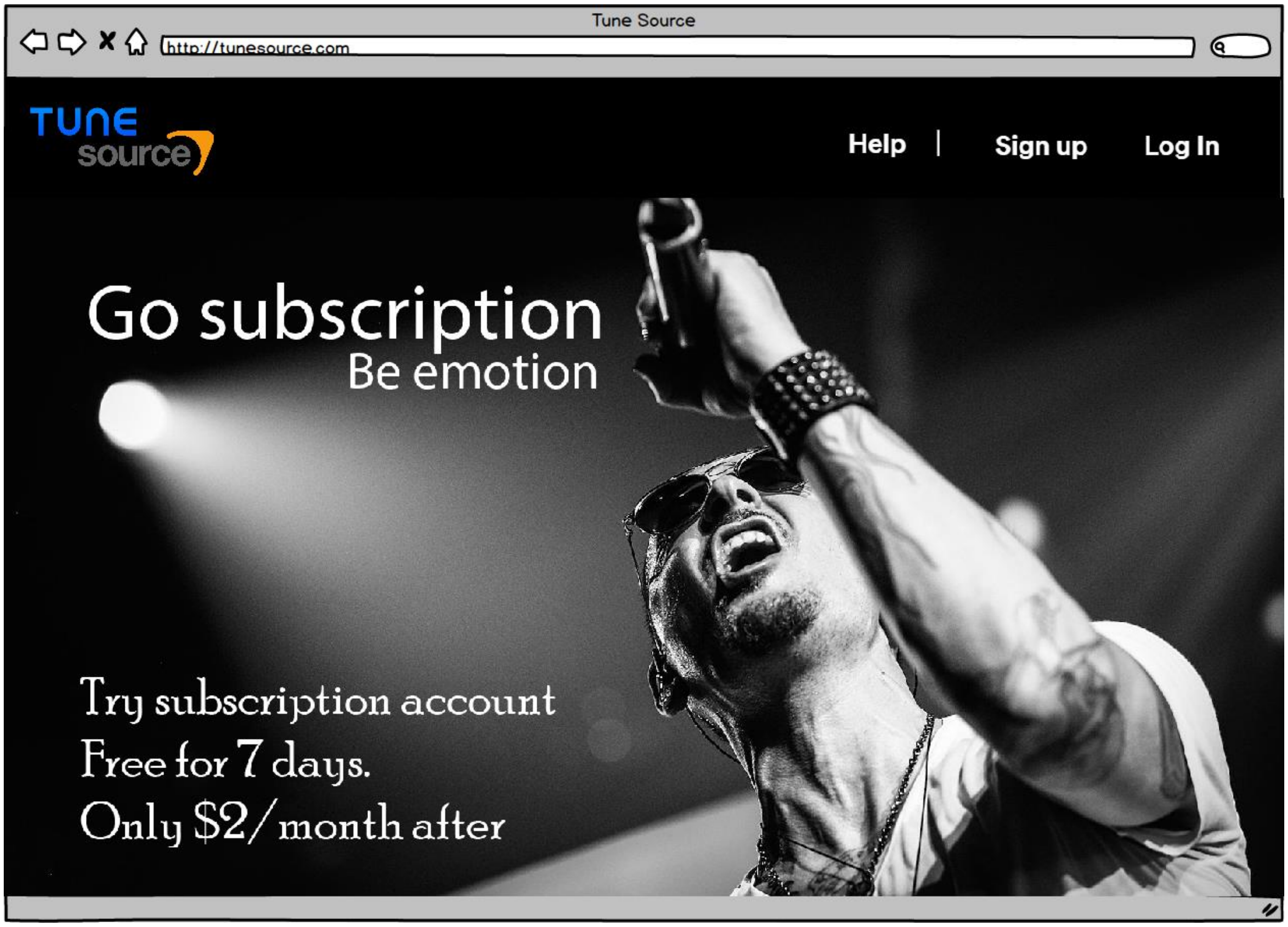
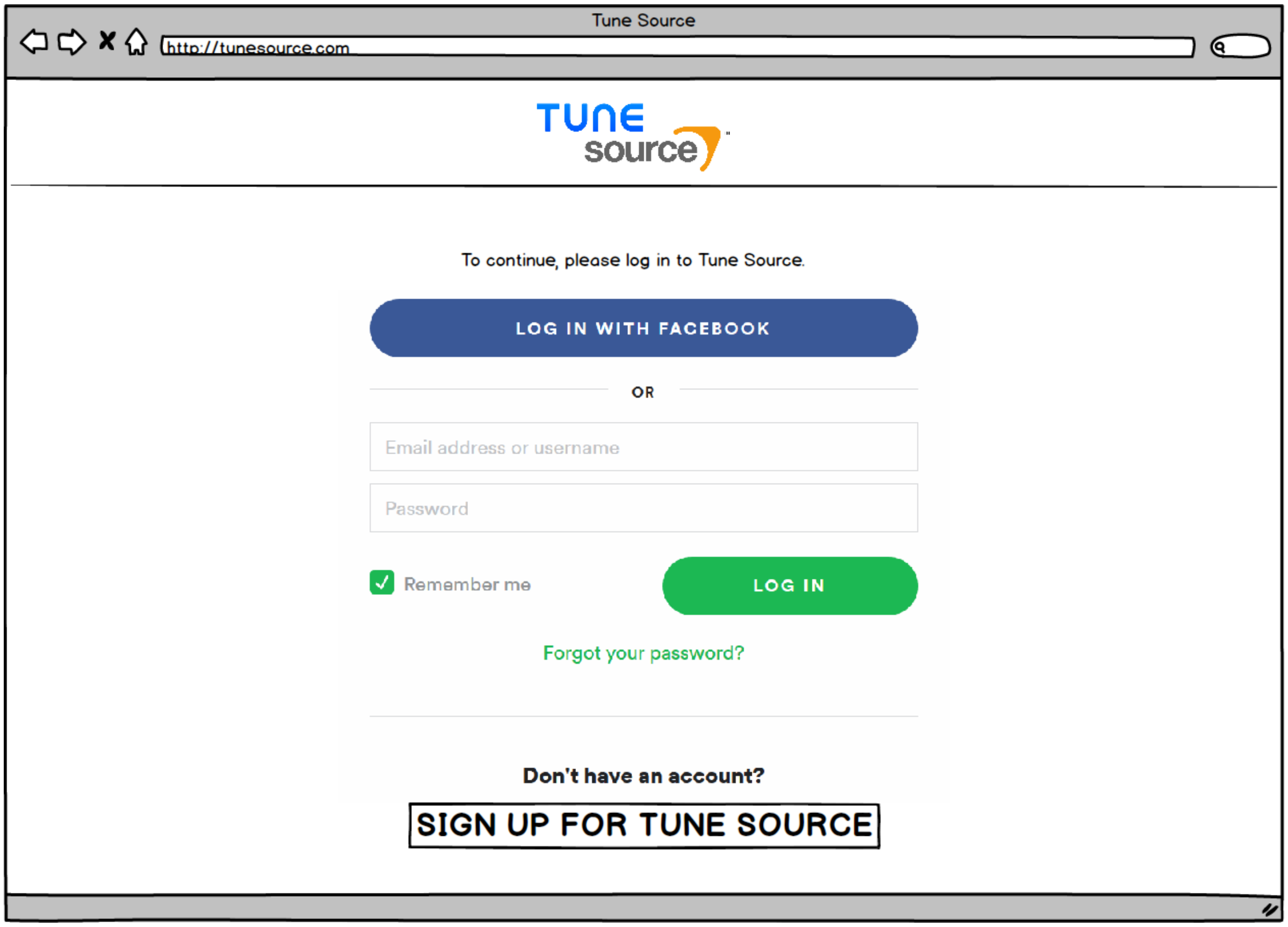


Figure 12 - Tune Source website main menu

- **Tune Source log in**: Customer login in this site and maybe also login with facebook account or sign up for new account. (This interface base on (spotify, n.d.))

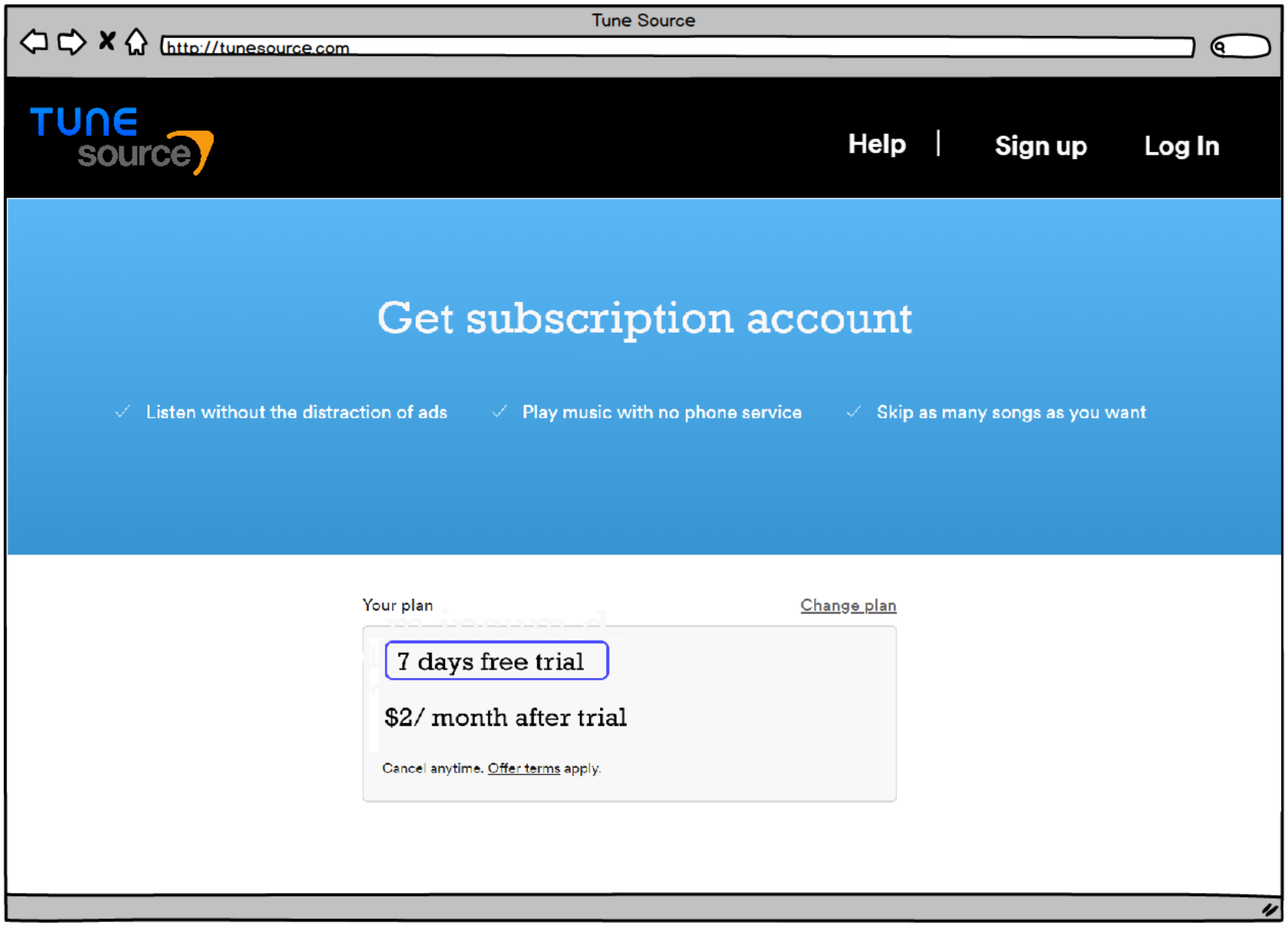


To continue, please log in to Tune Source.

Don't have an account?

Figure 13 - Tune Source login

- **Tune Source subscription account**: Users can purchase various product packages when using Tune Source



Get subscription account

✓ Listen without the distraction of ads ✓ Play music with no phone service ✓ Skip as many songs as you want

Your plan Change plan

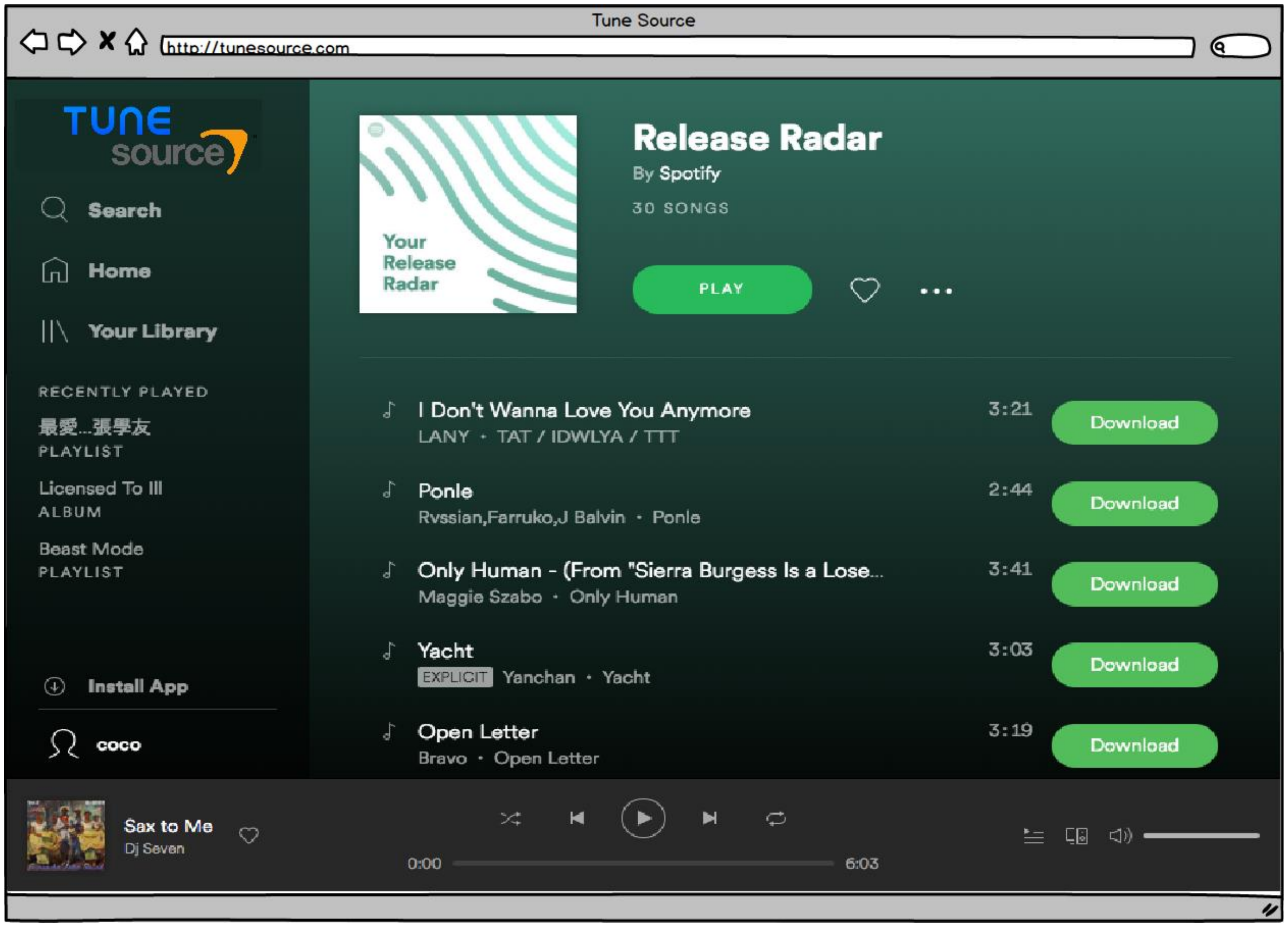
7 days free trial

$2/ month after trial

Cancel anytime. Offer terms apply.

Figure 14 - Tune Source subscription account

- **Tune Source Listening to music and download**: This site allow customer can listen to music and download music if the user is subscribed account. This interface base on (spotify, n.d.).



| | | |
|---|---|---|
| I Don't Wanna Love You Anymore<br>LANY · TAT / IDWLYA / TTT | 3:21 | Download |
| Ponle<br>Rvssian,Farruko,J Balvin · Ponle | 2:44 | Download |
| Only Human - (From "Sierra Burgess Is a Lose...<br>Maggie Szabo · Only Human | 3:41 | Download |
| Yacht<br>EXPLICIT Yanchan · Yacht | 3:03 | Download |
| Open Letter<br>Bravo · Open Letter | 3:19 | Download |

Figure 15 - Tune Source listening to music and download

### 1.2.2. Tune Source application prototype:

- **Tune Source main login:** Here customers are required to log into tune source, if new customers can register or log in via facebook, google or twitter ID, when the customer successfully login, the system will notify.

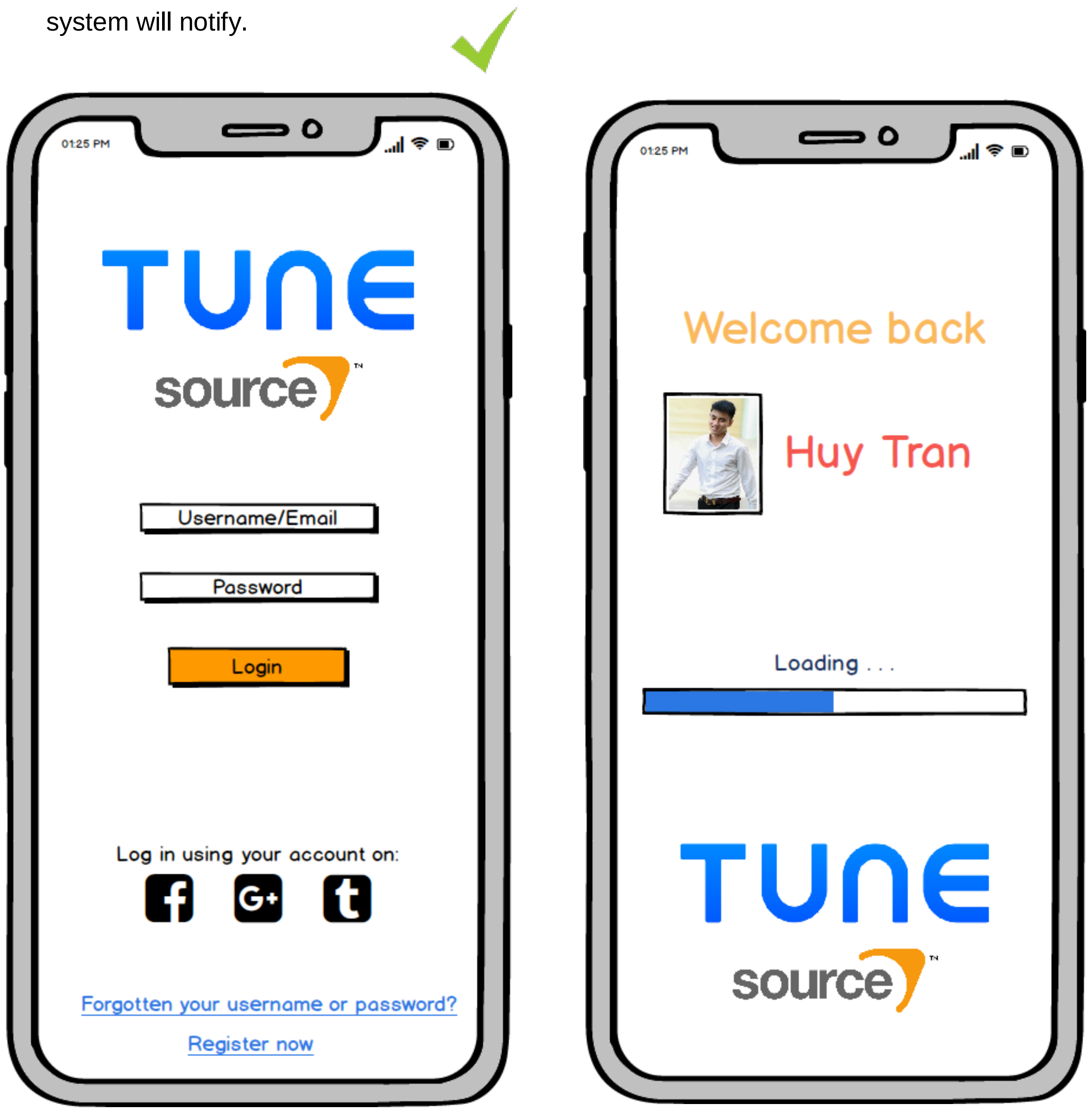


Figure 16 - Tune Source main login (Application)

- **Tune Source main menu:** In the Main menu, customers can choose music options as well as profile adjustments, check account balances or transaction history. (Base on: (Katrina, 2018))

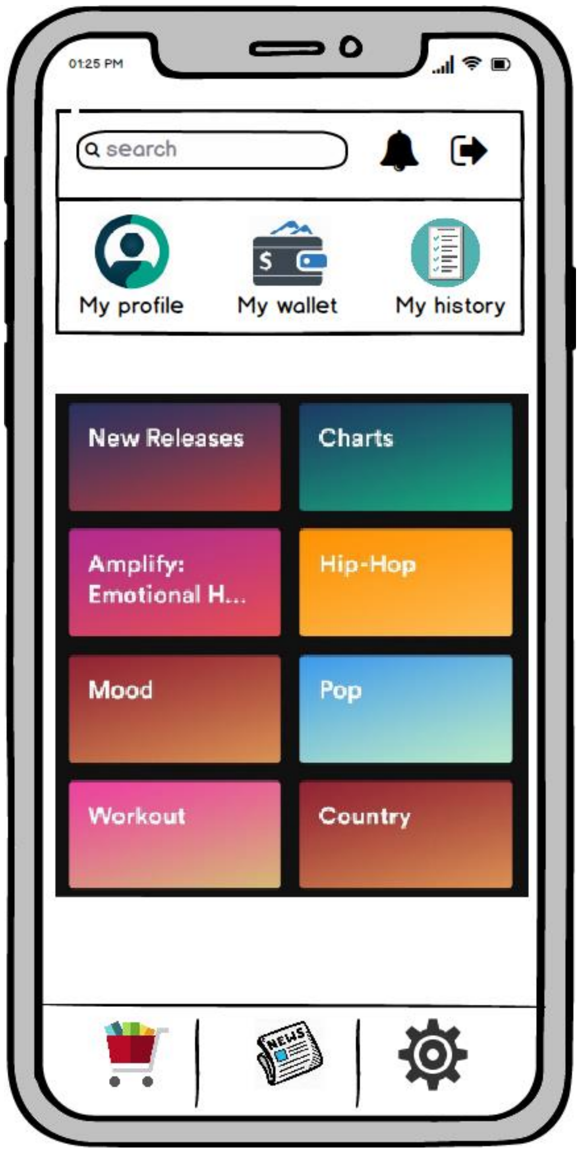


Figure 17 - Tune Source main menu (Application)

- **Music store and songs/albums recommended:** Customers can easily search their favorite songs on the system and Tune Source will also provide customers with music suggestions. (Base on: (Katrina, 2018))

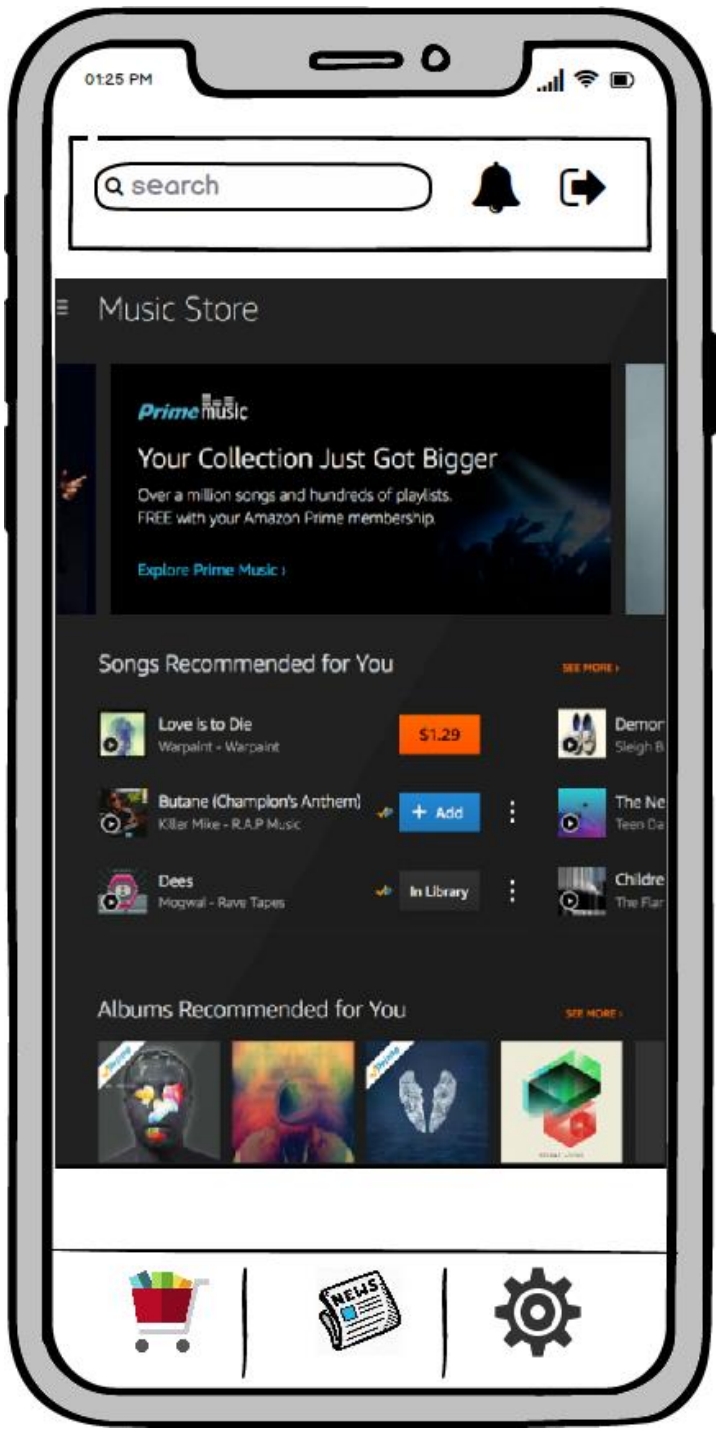

01:25 PM
search
Music Store
Prime music
Your Collection Just Got Bigger
Over a million songs and hundreds of playlists.
FREE with your Amazon Prime membership.
Explore Prime Music ›
Songs Recommended for You
SEE MORE ›
Love is to Die
Warpaint - Warpaint
$1.29
Butane (Champion's Anthem)
Killer Mike - R.A.P. Music
+ Add
Dees
Mogwai - Rave Tapes
In Library
Albums Recommended for You
SEE MORE ›
NEWS

Figure 18 - Music store and songs/albums recommended (Application)

- **Listening to music and purchase music:** This is the user music interface, if the user likes the currently playing music or album, it can be downloaded if it has purchased the music. (Base on: (Katrina, 2018))

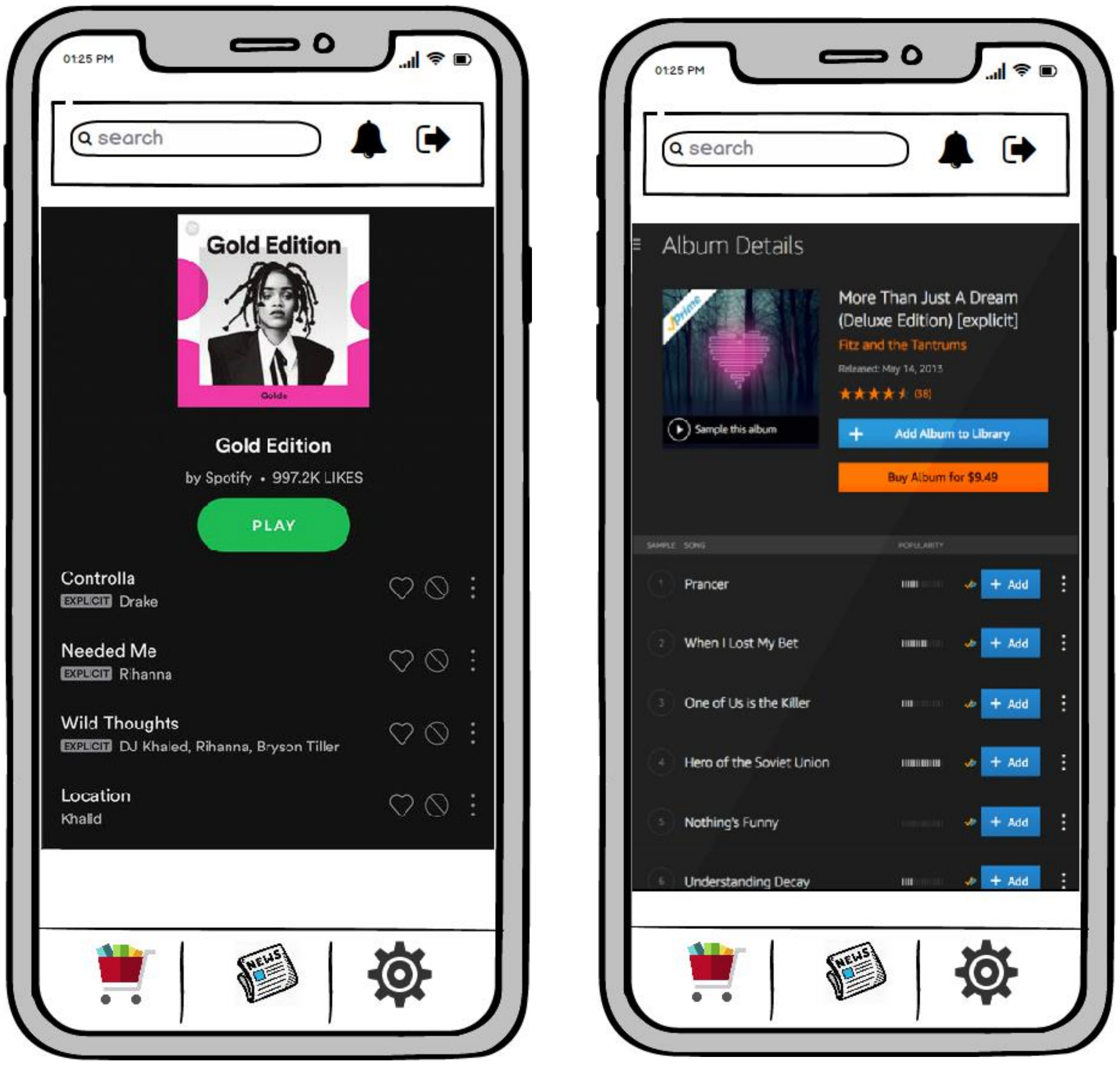


Figure 19 - Listening to music and purchase music (Application)

## 2. Suggest two software behavioural specification methods and illustrate their use with an example.

### 2.1. Function download music in Tune Source:

#### 2.1.1. Flowchart:

In this situation, the user has successfully logged in to the system as a customer subscripted, the flowchart only describes the download music, but does not describe the cross-selling of CD products as the overall model of the system Tune Source.

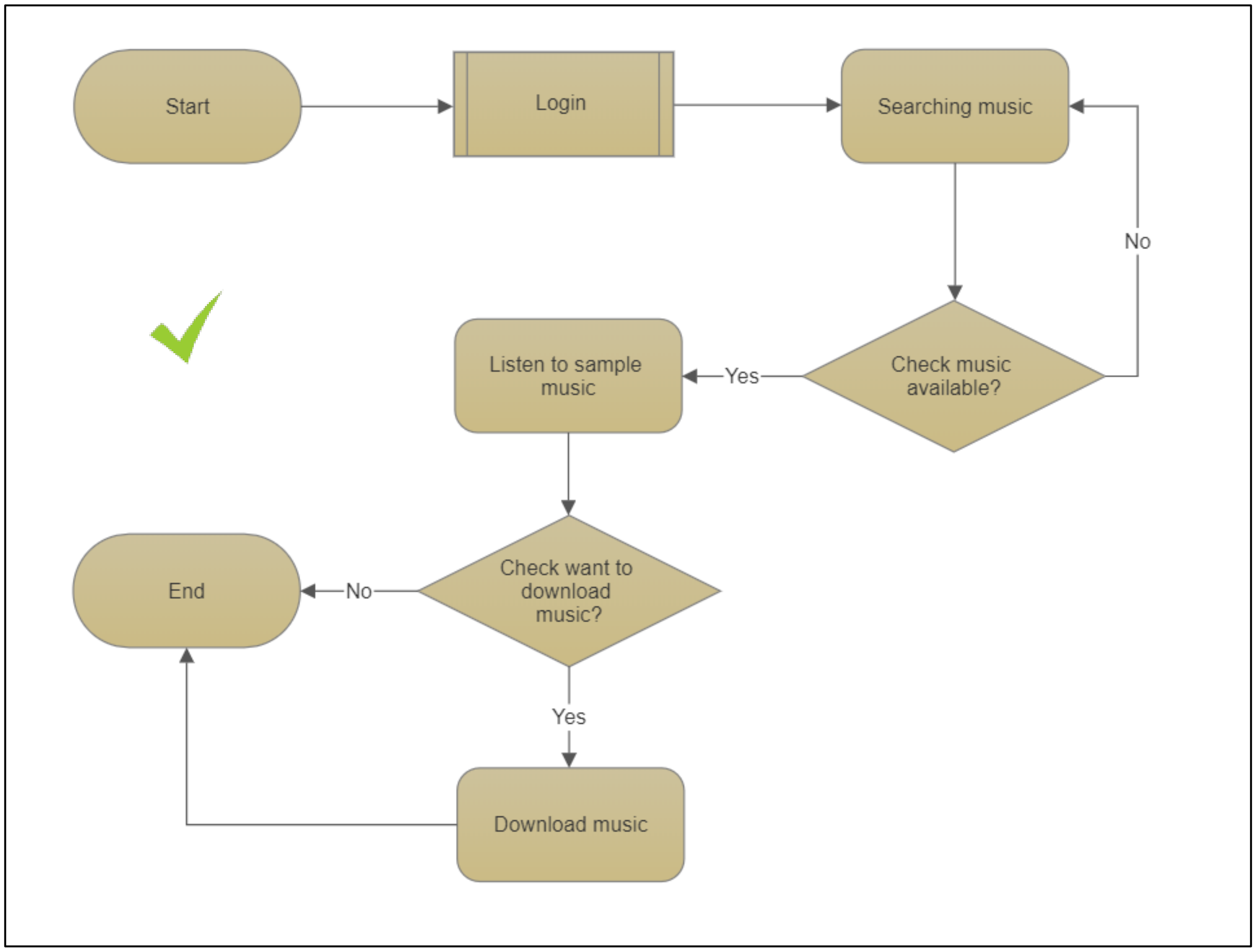


Figure 20 - Flowchart function download music

### 2.1.2. Pseudo code:

Function download music in Tune Source implement by C# programming language.

```
namespace SDLC_Download_Music
{
    class Program
    {
        static void Main(string[] args)
        {
            char name_of_music = Convert.ToChar(Console.ReadLine());
            bool request_download;
            bool music_available;
            bool account_subscriptions;

            if (account_subscriptions=true)
            {
                name_of_music = true;
            }
            else
            {
                name_of_music = false;
            }


            if (name_of_music == DB.name_of_music)
            {
                request_download = true;
                show.message("Please wait a second, the song will be downloading...");
            }
            else
            {
                request_download = false;
                show.message("We can not find a song you want, please try again!");
            }
        }
    }
}
```

## 2.2. Function subscriptions In Tune Source:

### 2.2.1. Flowchart:

With the In Tune Source subscriptions function, the user must first register the account at Tune Source then when logging into the system, the system will verify whether the user is already a subscription account or not.

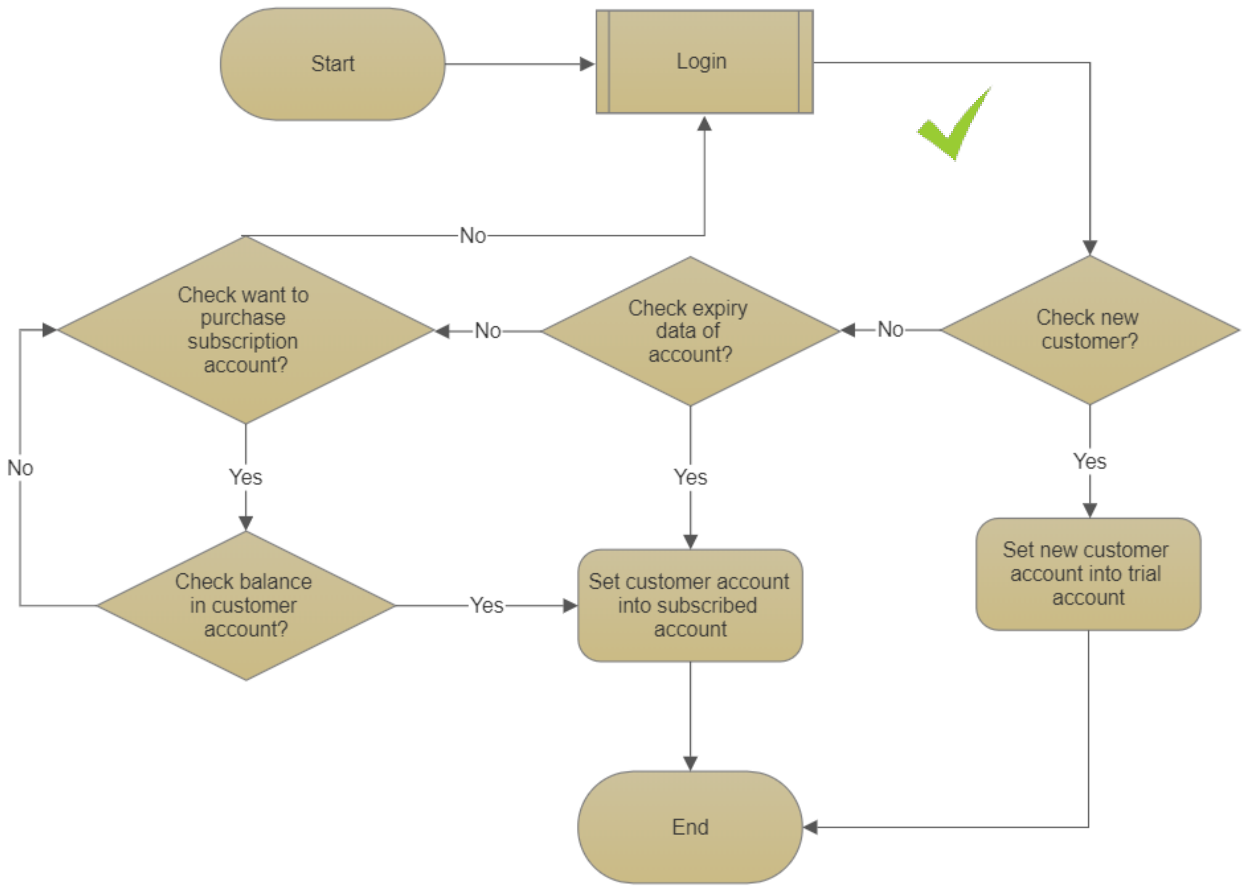

Start
Login
No
Check want to purchase subscription account?
No
Check expiry data of account?
No
Check new customer?
No
Yes
Yes
Yes
Check balance in customer account?
Yes
Set customer account into subscribed account
Set new customer account into trial account
End

Figure 21 - Flowchart function subscriptions

### 2.2.2. Pseudo code:

Function subscriptions in Tune Source implement by C# programming language.

```
namespace SDLC_subscriptions_account
{
    class Program
    {
        static void Main(string[] args)
        {
            char customer_id = Convert.ToChar(Console.ReadLine());
            char customer_password = Convert.ToChar(Console.ReadLine());
            bool login_status;
            bool new_customer_status;
            float account_balance;
            char sub_date;
            bool expiry_date;
            bool subscriptions_status;
            bool purchase_sub;

            if ((customer_id == DB.customer_id) & (customer_password ==
DB.customer_password))
            {
                show.message("Welcome to Tune Source");
                login_status = true;
            }
            else
            {
                login_status = false;
                show.message(" Your ID or password is incorrect");
            }

            if ((login_status = true) & (new_customer_status = true))
            {
                subscriptions_status = true;

            }
            else if (expiry_date = true)
            {
                subscriptions_status = true;
            }
            else if (purchase_sub = false)
            {
                show.message("Please subscribe to listen and download music");
            }
            else if ( account_balance > DB.account_balance)
            {
                account_balance = account_balance - DB.account_balance;
                subscriptions_status = true
            }
            else
            {
                show.message("Please Add money to your account");
            }
        }
    }
```

## 3. Finite state machine (FSM) and providing an application.

### 3.1. What is finite state machine (FSM):

A state machine is any device that stores the status of something at a given time and can operate on input to change the status and/or cause to take place for any given change.

- It consists of states, input and outputs.
- Suitable for control process that react to local conditions.
- The machine is only one state at a time – the current state
- It can change from one state to another when initiated by a triggering event or condition: the transition. (Landau, 2013)

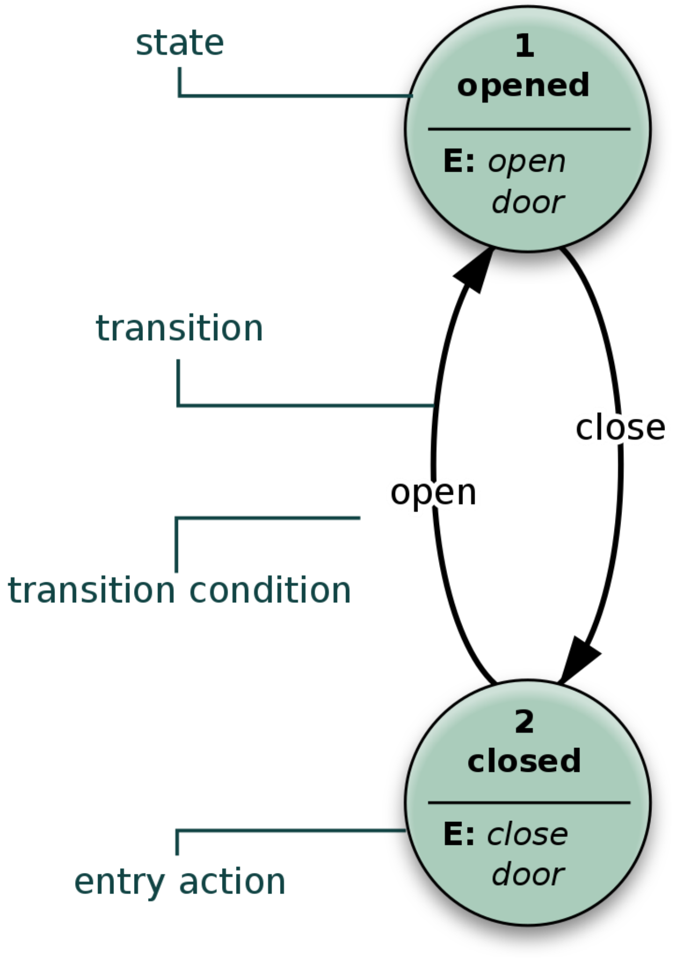


Figure 22 - Example finite state machine (wikipedia, n.d.)

## 3.2. Finite state machine (FSM) in Tune Source project.

### 3.2.1. FSM for subscribed customer:

- The FSM model below applies to registered / accountable customers and subscribed account of Tune Source.

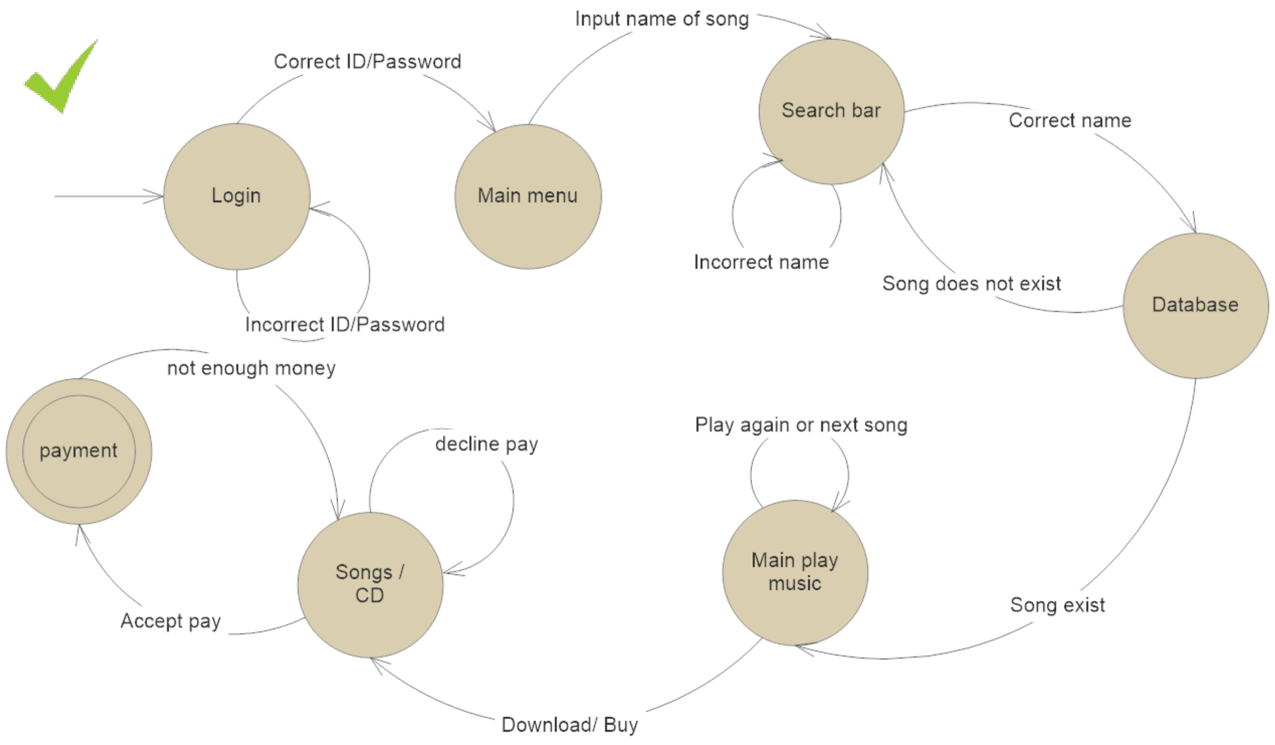


Figure 23 - FSM for subscribed customer

- This means that the following inputs are valid:
    - Correct ID/ Password - Input name of song – correct name – song exist- Download/ Buy – Accept pay.
    - Correct ID/ Password - Input name of song – correct name – song exist- Download/ Buy – decline pay.
    - Correct ID/ Password - Input name of song – correct name – song exist- Download/ Buy – decline pay - Accept pay.
    - Correct ID/ Password - Input name of song – correct name – song exist- Download/ Buy – decline pay - Accept pay – not enough money – accept pay.
    - Incorrect ID/Password - Correct ID/ Password - Input name of song – incorrect name - correct name – song exist- Download/ Buy – decline pay.
    - More and more inputs are valid.

### 3.2.2. FSM for unsubscribed customer:

- The FSM model below applies to registered / accountable customers and unsubscribed account of Tune Source:

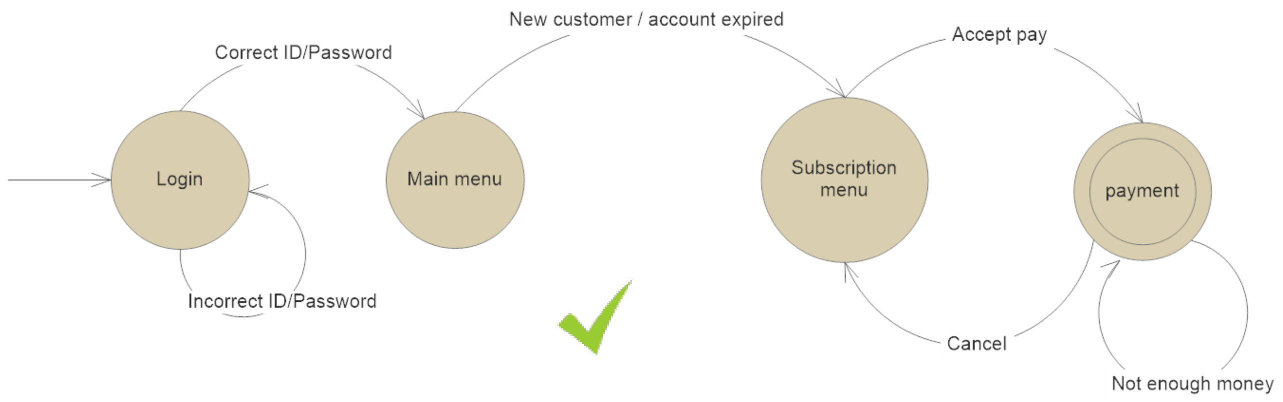


Figure 24 - FSM for unsubscribed customer

- This means that the following inputs are valid:
    - Correct ID/Password – New customer / account expired – Accept pay
    - Correct ID/Password – New customer / account expired – Accept pay – Not enough money
    - Correct ID/Password – New customer / account expired – Accept pay – Not enough money – cancel – accept pay
    - Incorrect ID/Password - Correct ID/Password – New customer / account expired – Accept pay
    - More and more inputs are valid.

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

### 4.1. What is data driven?

In most programming languages, data structures are easier to manipulate than code. As a result, it is often useful to design applications so that as much of their behavior as possible is represented as data rather than in the form of methods. Programs that work this way are said to be data driven.

In software-driven programming, data-driven programming is a programming model in which the source code of the program does not contain very little business logic but only general logic, which allows processing other data sets. together. Data-driven programs can be considered only tools used to read and process data and then produce results corresponding to data. Instead of logic processing statements written rigidly in program source code, the data itself is statements, the data contains information about the control flow, sequence that needs to be executed. In other words, data is business.

Data-driven program is designed to be able to change the logic of the program by changing the data without changing the source code of the program. The program now acts as a pipeline handling the commands based on the input data and producing the corresponding results. (Tuan, 2018)

Data-driven programming is often applied to structured data streams, for filtering, converting, and aggregating information. Input streams include log files, split data or email ...

A few popular Data-driven programming languages:

- AWK
- Sed
- Perl
- Lua
- Oz
- Clojure

In a data-driven system, the actual program (which is called a driver) is usually very small. Such driver programs operate in two phases:

- Read data from a file into a suitable internal data structure.
- Use the data structure to control the flow of the program.

### 4.2. Evaluate how this approach would help improving the reliability.

The primary focus idea of data-driven programs in Tune Source are programs in which the flow of execution is controlled by the data structures. Data-driven programs are usually shorter, more flexible, and easier to maintain than programs that incorporate the same information directly into the program design.

The Tune Source database was designed to control and save all system-wide data, including music data, purchase transaction data and, most importantly, customer data. With the implementation of Data Driven method, Tune Source database will become the most dynamic criterion to create the success of the whole system.

Content is the core to attract and keep customers. Using advanced techniques such as text analysis and predictive analysis, the system developer of Tune Source can provide truly compelling multi-channel experience, omnichannel experience and truly compelling contents (such as Songs, Albums suggestion, based on the type of music, singers that customers often find), the content will be responsible for optimizing the overall experience of the customer. Furthermore, with these features, Tune Source can improve Search Engine Optimization (SEO) for digital content, quickly categorize content and exploit text for words, phrases and topics for visitors. line.

| Criteria | Requirement | Effective |
|---|---|---|
| 1. Quality ✓<br>- Song quality<br><br>- File Format<br><br>- Kind of music | <br>- Over 320 kbps, 20 Khz<br><br>- *.mp3, *.pcm, *.wav, *.aac, *.ogg, *.flac.<br>- Lossness | <br>- Songs ensure high quality for users<br>- Manu formats will be used for various software and devices<br>- Quality sound is similar CD |
| 2. License<br>- Co-owner<br><br>- Authorized by publisher | <br>- Acknowledge their rights to their trademark<br>- Got trusted from publisher | <br>- Ensure absolute copyright.<br><br>- Customers feel more secure about the quality of the songs |
| 3. Easy to use<br>- Interface<br><br>- Function<br><br>- Management & report | <br>- Less than 3 clicks to use<br><br>- 1 click for 1 function<br><br>- Auto create report charts and business performance statistics | <br>- Simplify all operations on the system<br>- Simple operations to perform function<br>- Check and report more easily. |

Table 5 - Effectives Data Drive to Tune Source

Moreover, through advanced data analysis, Tune Source system can:

- Determine the different levels of customer segments that are able to respond to specific content, campaigns or marketing actions by implementing detailed analysis, detailed analysis of both registered customers and not yet registered.
- Reaching the target population that Vials can very well respond to content, campaigns and other marketing activities. With the predictive model, Tune Source can understand and predict the behavior of each targeted group.
- Improve economic results by using optimization to make the most of individual customer communication. Consider limitations on resources and budgets, contact policies, customer feedback and more.
- With a unique view of each customer and a detailed understanding of what to show and promote for each customer, the Tune Source website will have a key to successful marketing, thus achieving branding. Maximum results of the system. The Data-based approach combined with diverse data analysis will provide the Tune Source system with the necessary knowledge to nurture loyal customers based on quality music content, thus increasing value of customers and increase profits at the same time

# CONCLUSION

Through the evaluation results from Tune Source, we realized that the system of Tune Source is increasingly developing, applying new forms to bring very high efficiency. The feasibility of technology, economic is very high. In addition, the interface of Tune Source has met most of the requirements from users as well as the requirements of the company's business. The functional models of Tune Source have been completed with complete code. And finally, to let Tune Source grow, it is not possible to rely on the current results to assess but to have long-term plans, practical economic requirements for users' needs and use of capital need to be effective.

# References

*altexsoft*. (2017). Retrieved from altexsoft: https://www.altexsoft.com/blog/business/software-documentation-types-and-best-practices/

*Apiumhub*. (n.d.). Retrieved from Apiumhub: https://apiumhub.com/tech-blog-barcelona/advantages-of-test-driven-development/

Crnkovic, A. &.-D. (2003).

*guru99*. (n.d.). Retrieved from guru99: https://www.guru99.com/traceability-matrix.html

*hashe*. (n.d.). Retrieved from hashe: https://www.hashe.com/software-development-life-cycle-sdlc/

Landau, D. ( 2013).

*lucidchart*. (n.d.). Retrieved from lucidchart: https://www.lucidchart.com/pages/templates/data-flow-diagram/dfd-level-0-template

*lucidchart*. (n.d.). Retrieved from lucidchart: https://www.lucidchart.com/pages/templates/data-flow-diagram/dfd-level-1-template

*lucidchart*. (n.d.). Retrieved from lucidchart: https://www.lucidchart.com/pages/templates/data-flow-diagram/dfd-level-2-template

*netz*. (n.d.). Retrieved from netz: https://netz.id/news/2017/07/21/00816/1003210717/ini-5-fakta-menyedihkan-di-balik-kematian-chester-bennington

Sehlhorst, S. (n.d.). Retrieved from https://searchsoftwarequality.techtarget.com/answer/How-traceability-benefits-the-software-development-lifecycle

*spotify*. (n.d.). Retrieved from spotify: www.spotify.com/

Tuan, P. M. (2018, 06). Retrieved from https://www.cppdeveloper.com/tutorial/lap-trinh-huong-du-lieu-data-driven-programming-la-gi/

Wang. (1996). Retrieved from https://pdfs.semanticscholar.org/4994/388a165316cdb54bf7e502342dd576ccd253.pdf

*wikipedia*. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Software_documentation

*wikipedia*. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Finite-state_machine

*wikipedia*. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Data-flow_diagram

*wikipedia*. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Decomposition_(computer_science)

*wikipedia*. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Sequence_diagram

# Index of comments

5.1 1. Strength points
The content that shows is very clear
The quality of investing is very good
Almost charts are clear and detailed.
2. Weak points

N/A
3. How to do better in future:

Attention more on the flowchart end sequence more detail
The format of the document should more clear.
Increase the quality of design by ERD (Entity relationship diagram)

40.1 Too detail, just to apply some steps