**Assignment Submission Form**

This form must be filled in and completed by the student(s) submitting an assignment

|  |
| --- |
| Name(s): Killian Byrne |
| Programme: CA |
| Module Code: CA214 |
| Assignment Title: SSADM PROJECT |
| Submission Date: 10/12/2017 |
| Module Coordinator: Renaat Verbruggen |

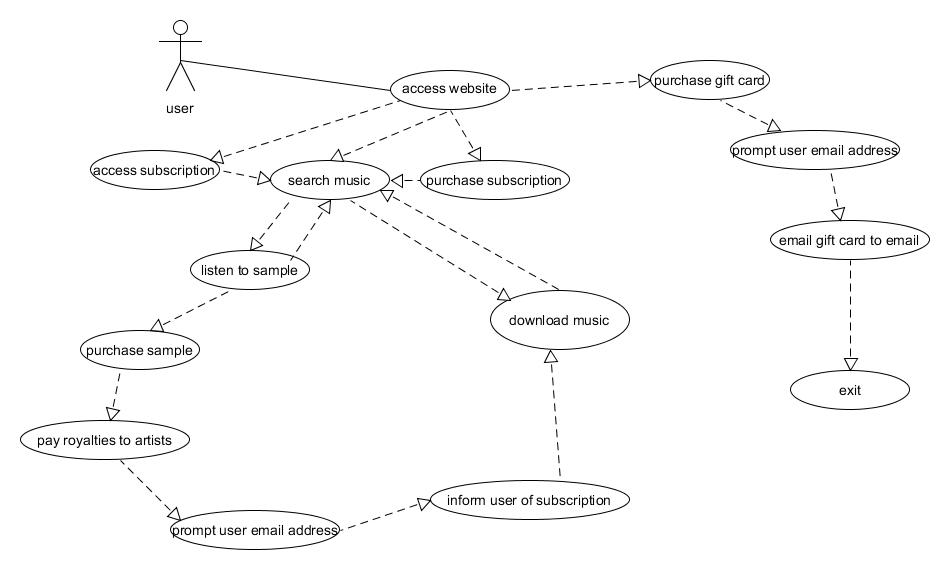
I/We declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my/our work. I/We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I/We have read and understood the Assignment Regulations. I/We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me/us or any other person for assessment on this or any other course of study.

I/We have read and understood the referencing guidelines found at

<http://www.dcu.ie/info/regulations/plagiarism.shtml>, <https://www4.dcu.ie/students/az/plagiarism>

and/or recommended in the assignment guidelines.

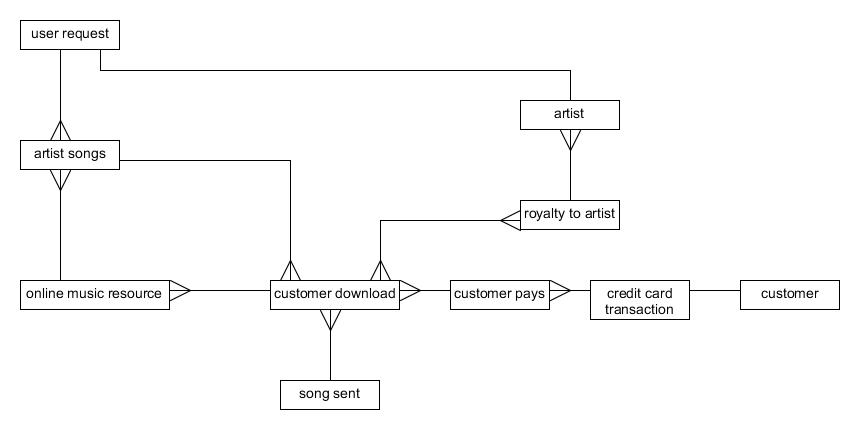
Name(s): Killian Byrne Date: 09/12/2017

BUSINESS ACTIVITY MODEL: BAM DIAGRAM (FIG1.1)

**BAM TASKS:**

1. Access website
2. Purchase gift card
3. Prompt user email address
4. Email gift card to email address
5. Access subscription
6. Purchase subscription
7. Search music
8. Listen to sample
9. Purchase sample
10. Pay royalties to artists
11. Prompt user email address
12. Inform user of subscription
13. Download music

Logical Data Model – LDM (FIG 2.1)



**IDENTIFY CANDIDATE ENTITIES**

1. User request
2. Artist songs
3. Online music resource
4. Artist
5. Royalty to artist
6. Customer download
7. Song sent
8. Customer purchase
9. Credit card transaction
10. Customer

**IDENTIFY RELATIONSHIPS**

* Relationships will be marked with an ‘X’ in table 1.

**Table :** *Relationships present* (FIG 2.2)

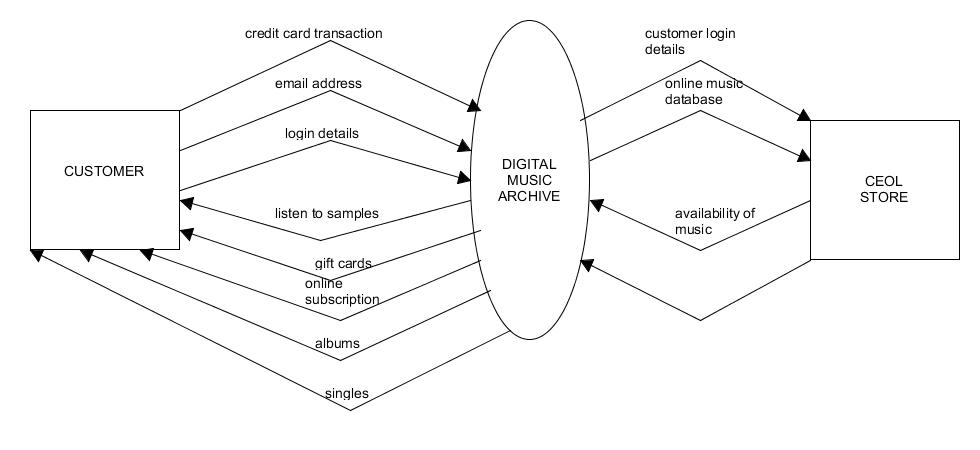
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Customer | Card  Transaction | Customer  purchase | Customer  Download | Song  Sent | Royalty | Artist | Music  Resource | A.  Songs | User  request |
| User request |  |  |  |  |  |  | X |  | X |  |
| Artist songs |  |  |  | X |  |  |  |  |  |  |
| Music resource |  |  |  |  |  |  |  |  | X |  |
| Artist |  |  |  |  |  |  |  |  |  | X |
| Royalty |  |  |  | X |  |  | X |  |  |  |
| Song sent |  |  |  | X |  |  |  |  |  |  |
| Customer download |  |  |  |  |  |  |  | X |  |  |
| Customer purchase |  |  |  | X |  |  |  |  |  |  |
| Card transaction |  |  | X |  |  |  |  |  |  |  |
| Customer |  | X |  |  |  |  |  |  |  |  |

* A user may only request one artist at a time, and an artist in the database cannot summon many users, therefore it is a 1:1 relationship.
* With one purchase, many downloads can occur, so it is a 1:m relationship.
* One royalty can go to many artists, so it is a 1:m relationship also.

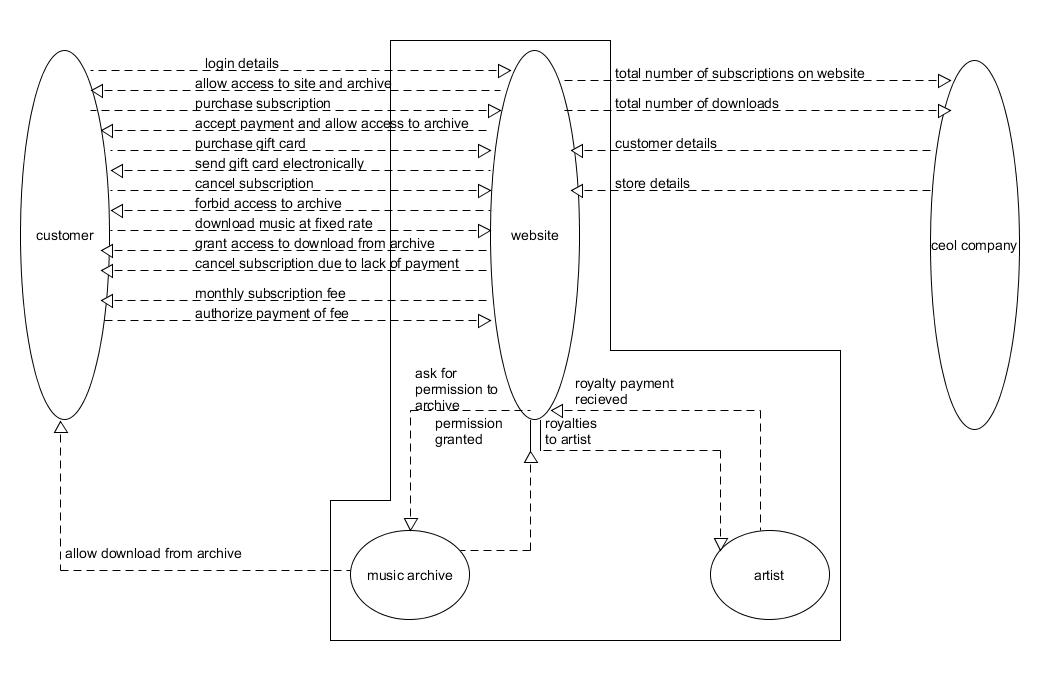
**DEGREES OF RELATIONSHIP:**

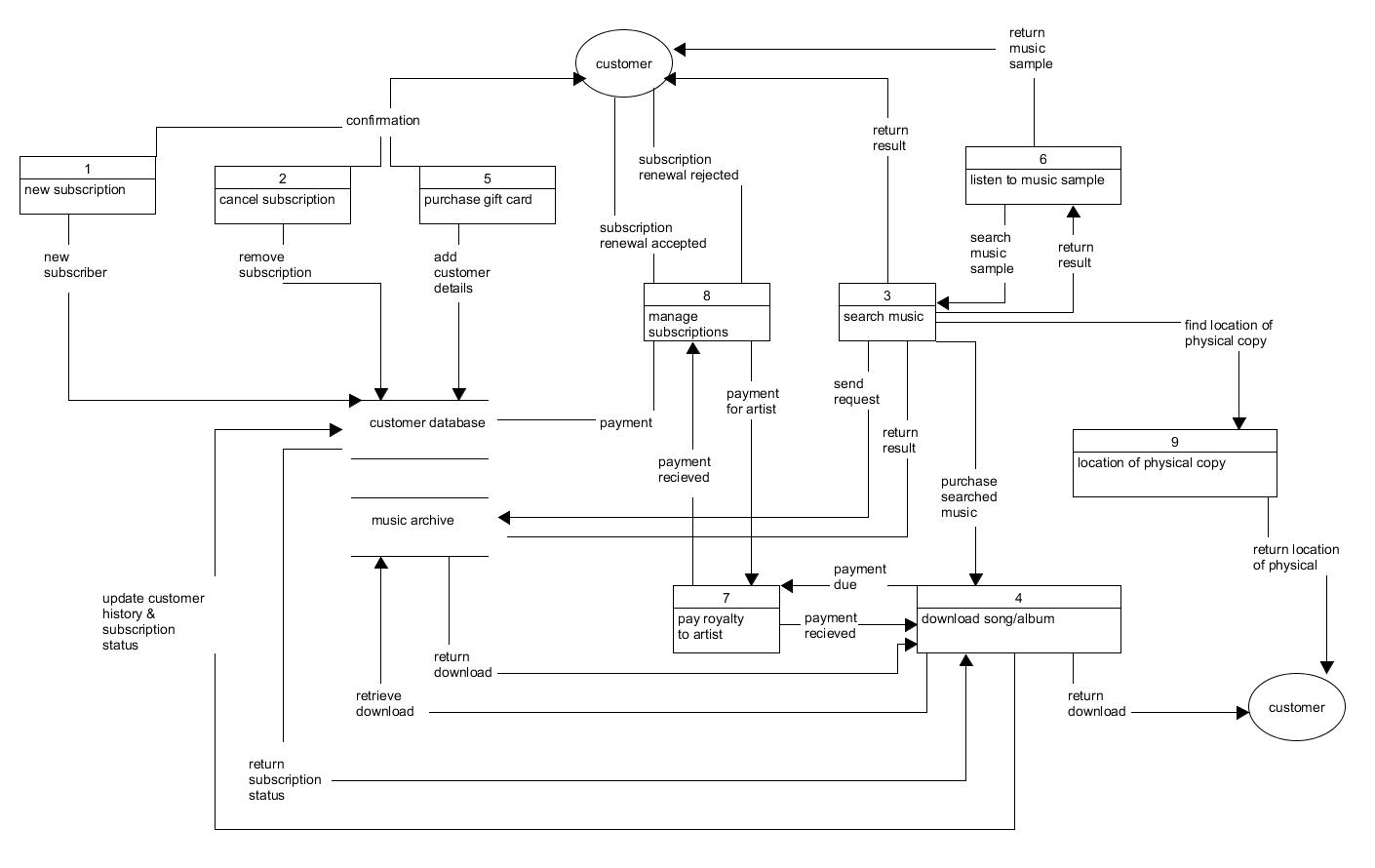
* Credit card transaction : customer = 1:1
* Artist’s songs : online music resource = m:1
* Customer download : royalty to artist = m:n
* User request : artist songs = 1:m
* Customer purchase: credit card transaction = m:1
* Customer download : customer purchase = m: 1

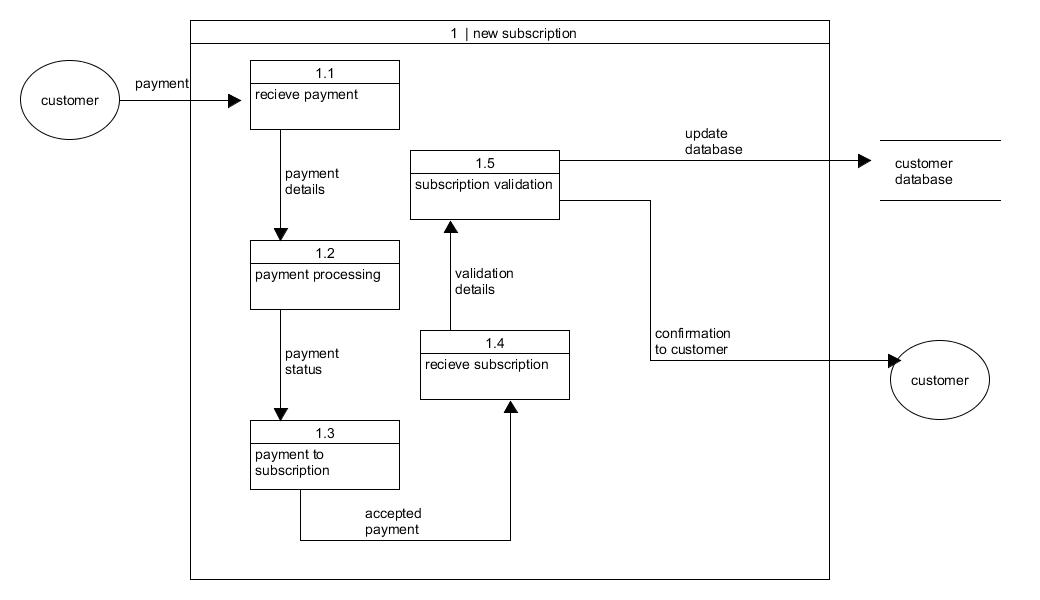
TOP LEVEL CONTEXT DIAGRAM (FIG 3.1)

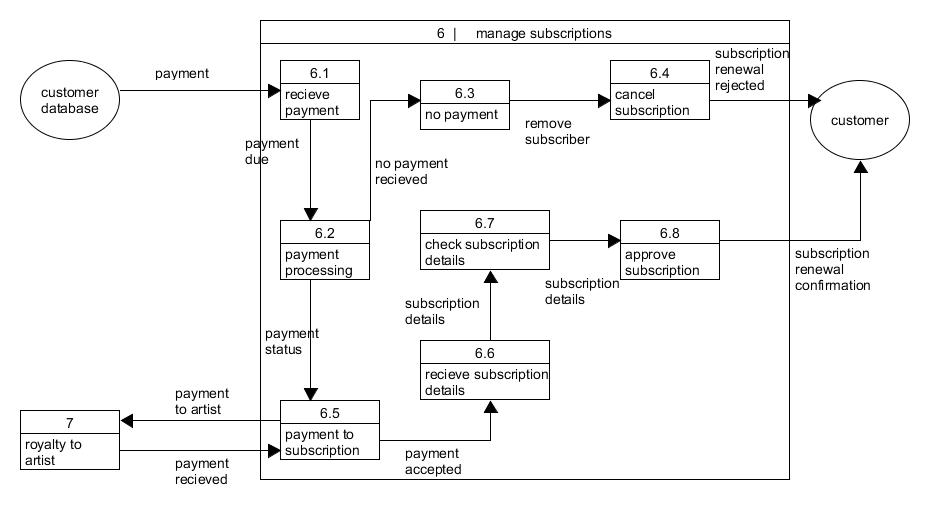


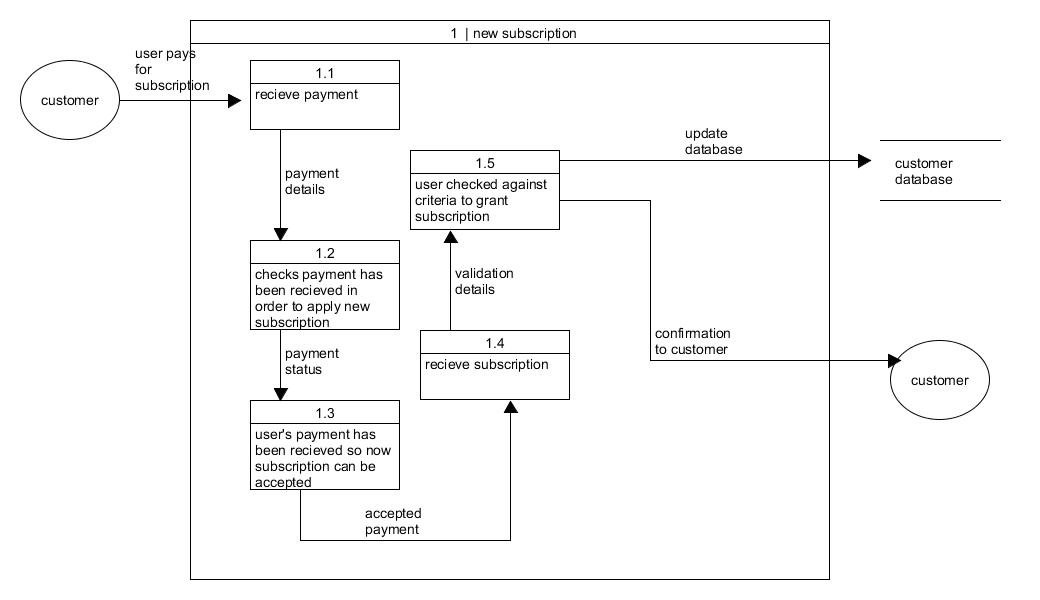
DOCUMENT FLOW DIAGRAM (FIG 4.1)

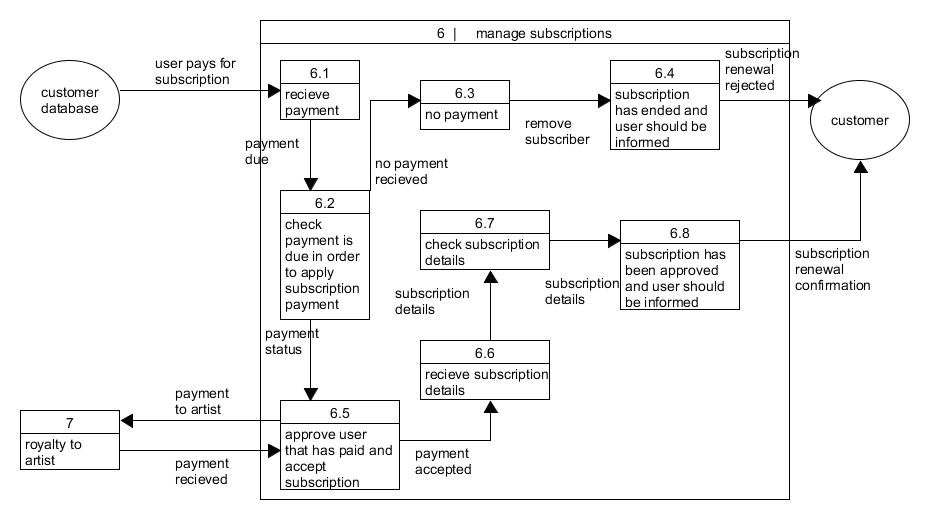


PHYSICAL LEVEL ONE DFD (FIG 5.1)

PHYSICAL LEVEL 2 DFD – NUMBER ONE (FIG 6.1)

PHYSICAL LEVEL 2 DFD – NUMBER TWO (FIGURE 6.2)

LOGICAL DFD – NUMBER ONE (FIG 7.1)

LOGICAL DFD – NUMBER TWO (FIG 7.2)

**FUNCTION 1 - PAYMENT PROCESSING(DFD 1):**

*START*

Payment processing function

*GET* subscription holder authorisation

*REPEAT UNTIL* authorisation is approved

*IF* authorisation is approved

*DISPLAY* “login approved” *AND ALLOW* subscription holder to access system

*DISPLAY* subscription holder’s current subscription details

*PROMPT* user to pay subscription fee

*REPEAT UNTIL* payment is successful

*IF* payment successful

*Renew* subscription

*ELSE*

*DISPLAY “*payment unsuccessful” AND don’t renew subscription

*ELSE*

*DISPLAY “*payment unsuccessful” AND don’t allow access

*END REPEAT*

*END*

**FUNCTION 2 – Payment to subscription (DFD 2):**

*START*

Payment to subscription function

*GET* subscription payment

*REPEAT UNTIL* payment has been received

*IF* payment is received

*GET* calculation of royalties owed to artists

*GET* calculation of subscription fee minus royalty payment

*SEND* payment to royalty account

*GET* confirmation payment has been sent

*SEND* rest of payment to account department

*APPROVE* user’s subscription payment

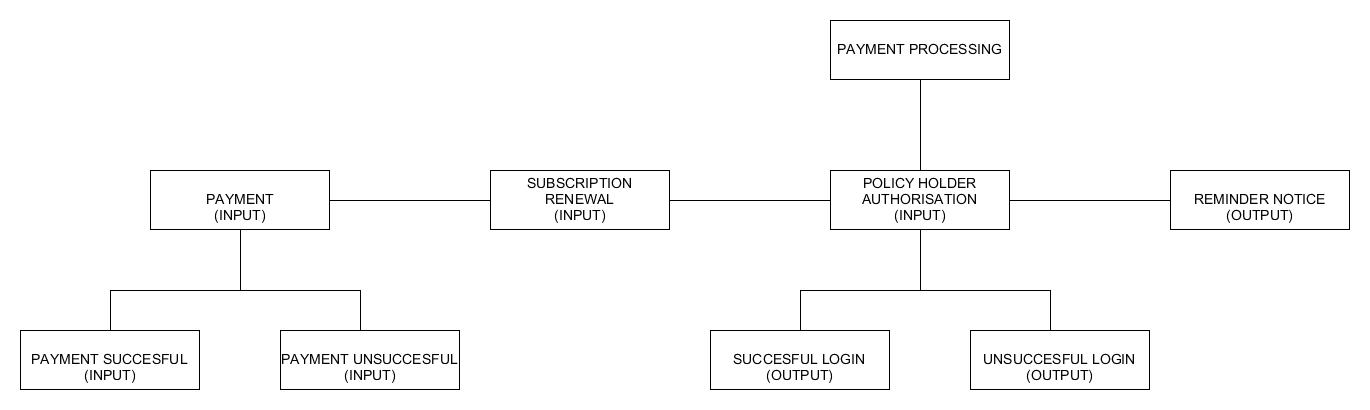
*SEND* confirmation to database of user’s approval

*ELSE* return “payment unsuccessful” AND deny access

END REPEAT

END

**INPUT/ OUTPUT DIAGRAM - PAYMENT PROCESSING DFD (9.1):**



**PROCESS DESCRIPTION (1) – MANAGE SUBSCRIPTIONS**

*START*

*GET* payment details from subscription holder

*SEND* payment details to accounts department and royalty account

*IF* payment is valid

*APPLY* subscription

*SEND* confirmation to customer

*SEND* update to database

*ELSE*

*IF* subscription is yet to be paid and subscription will run out in a week

*SEND* reminder to customer

*IF* subscription is yet to be paid and subscription will run out in a day

*SEND* reminder to customer

*IF* subscription is cancelled or payment is not received

*SEND* update to database

*SEND* confirmation to customer

*EXIT*

**PROCESS DESCRIPTION (2) – MANAGE NEW SUBSCRIPTIONS**

*START*

*GET* payment from new subscription holder

*SEND* payment to accounts department and royalty account

*IF* payment is valid

*APPLY* subscription

*ELSE*

*EXIT*

*GET* subscription details

*LINK* subscription details to new user

*GENERATE* completed policy

*SEND* completed policy to database

*SEND* confirmation to customer

*END*

**EXTERNAL ENTITY DESCRIPTION (FIG.12.1)**

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **DESCRIPTION** |
| a | Customer | The person who applies for a product of ceol or desires to use the online music application. |

**DATA FLOW (FIG 13.1)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FROM** | **TO** | **DATA FLOW NAME** | **DATA CONTENT** | **COMMENTS** |
| a | 5.1 | Purchase gift card | Name  Date of Birth  Address  Desired price for gift card  Customer e-mail  Recipient e-mail  Payment details | User should specify a date for the recipient to receive the gift card eg a date for a birthday |
| 5.1 | b | Add customer details | Name  Date of Birth  Address  Desired price for gift card  Customer e-mail  Recipient e-mail  Payment details | Simply updates the database with the data content supplied by the customer |

**ENTITY DESCRIPTION (FIG 14.1)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name:** Ceol online gift card | | | | | | | |
| **Description:** A voucher worth money that can be spent within Ceol | | | | | | | |
| **ATTRIBUTE** | | **PRIMARY KEY** | | **FOREIGN KEY** | | **MANDATORY/OPTIONAL** | |
| Gift card number  Date of purchase  Date of desired online delivery  Customer e-mail  Recipient email | |  | |  | | **M**  **M**  **O**  **M**  **O** | |
| *MUST/MAY BE* | *EITHER/OR* | | *LINK PHRASE* | | *ONE & ONLY ONE / ONE OR MORE* | | *ENTITY NAME* |
| Must be  Must |  | | Placed with  Result in | | Only one  Only one | | Customer  Payment |
| Entity Volumes: Maximum = 1 | Minimum = 1 | Average = 1 | | | | | | | |
| USER: | | | | ACCESS: | | | |
| Customer  Ceol Website | | | | Read  Read, create, modify, delete | | | |
| **Archiving:** Each gift card purchase should be stored away and linked to the customer’s details | | | | | | | |

**DATA STORE (FIG 15.1)**

|  |
| --- |
| *DATA STORE (FILE DESCRIPTION)* |
| *DATA STORE ID:* 5.1 |
| *DATA STORE NAME:* Create customer gift card |
| **Description:**   * Customer gift card details are received from customers. * The details are read to determine the price the gift card should appropriately be worth, who should receive the gift card, the recipient’s email and when it should be received. * Once all the variables have been determined, the gift card will be created and be put on standby. * If the customer desires a particular date for the gift card to be received, then the gift card will remain on standby until this date, upon when it is sent to the recipient. * If the user does not desire a particular date, then the gift card will b |