Requirements Engineering

Music Store Ordering System

Submitted By: Jack Carroll (T00194823)

Computing with Games Development

Date Submitted: 12/12/2017

**Table of Contents**

[1. Introduction/overview 4](#_Toc500777684)

[2. Functional Components 5](#_Toc500777685)

[3. User Requirements 6](#_Toc500777686)

[3.1. MusicStoreSYS will perform Supplier administration. 6](#_Toc500777687)

[3.2. MusicStoreSYS will perform Stock administration. 6](#_Toc500777688)

[3.3. MusicStoreSYS will process Orders. 6](#_Toc500777689)

[3.4. MusicStoreSYS will perform Administrative reporting. 6](#_Toc500777690)

[4. System Requirements 7](#_Toc500777691)

[4.1. System Level Use Case Diagram 8](#_Toc500777692)

[4.2. Manage Suppliers 10](#_Toc500777693)

[**4.2.1.** **Register Supplier** 10](#_Toc500777694)

[**4.2.2.** **Update Supplier Details** 13](#_Toc500777695)

[**4.2.3.** **Deregister Supplier** 15](#_Toc500777696)

[**4.2.4.** **Generate Supplier List** 17](#_Toc500777697)

[4.3. Manage Stock 19](#_Toc500777698)

[**4.3.1.** **New CD** 19](#_Toc500777699)

[**4.3.2.** **Update CD** 22](#_Toc500777700)

[**4.3.3.** **Discontinue CDs** 25](#_Toc500777701)

[**4.3.4.** **Query CDS** 27](#_Toc500777702)

[4.4. Process Orders 29](#_Toc500777703)

[**4.4.1.** **Place CD Pre-Order** 29](#_Toc500777704)

[**4.4.2.** **Receive CD Order** 31](#_Toc500777705)

[**4.4.3.** **Record Supplier Payment** 33](#_Toc500777706)

[4.5. Reporting Administrative Analysis 36](#_Toc500777707)

[**4.5.1.** **Analyse CD Orders** 36](#_Toc500777708)

[**4.5.2.** **Analyse CD Stock** 38](#_Toc500777709)

[5. System Model 42](#_Toc500777710)

[Level-0 DFD 43](#_Toc500777711)

[5.1. Level-1 DFD 44](#_Toc500777712)

[5.2. Level-2 DFD Manage Suppliers 45](#_Toc500777713)

[5.3. Level-2 DFD Manage Stock 46](#_Toc500777714)

[5.4. Level-2 DFD Process Orders 47](#_Toc500777715)

[5.5. Level-2 DFD Administration Analysis 48](#_Toc500777716)

[6. Data Model (Class Diagram) 49](#_Toc500777717)

[6.1. Class Diagram 49](#_Toc500777718)

[6.2. Relational Schema 50](#_Toc500777719)

[6.3. Database Schema 51](#_Toc500777720)

[7. Conclusion 55](#_Toc500777721)

[8. Appendices 56](#_Toc500777722)

[8.1. Appendix A – Title 56](#_Toc500777723)

[8.2. Appendix B – Title 56](#_Toc500777724)

# Introduction/overview

This Requirements Engineering Specification document focuses on the MusicStoreSYS System that will be created by the author of this document. The MusicStoreSYS handles orders of CDs to Suppliers which the Manager will carry out as the details of the Suppliers and CDs will be entered by the Manager before a CD is pre-ordered and delivered by the Supplier which results in the Supplier paying the Supplier and the system saves the payment made to the Supplier.

A hierarchy chart in this document gives a sneak peek into what functions the system will execute as the system is being planned and designed and it teases the functionality of the MusicStoreSYS to give an easier understanding of the actions that will be implemented by the system.

The user requirements of the MusicStoreSYS will give a glimpse of what is expected from the system when the user carries out the actions. It gives a brief explanation of what the system will perform within the system once the system is developed.

This document will detail the functional requirements using the use case narratives for each step or function of the MusicStoreSYS to make functionality easier for those willing to use it and the system requirements will appear later in this document to offer the reader of this document a detailed description of what the system will do once certain functions have been performed.

As well as that, a system model will be featured to show the data flow of particular modules of the system when the Manager performs a certain action, e.g, managing the Suppliers by entering and updating the Supplier details as well as de-registering a chosen Supplier and generating a Supplier list. The data flow diagrams (DFDs) will provide more details of the data flows as the levels of DFDs head down to a level 2 DFD for the modules.

Finally, the document will present a data model where the UML class diagram of the data storage files for the MusicStoreSYS is produced. Afterwards, the relational schema and database schema for the MusicStoreSYS is developed and will project detailed information surrounding the data that will be eventually placed into a database for the numerous tables formed from the UML Class Diagram after normalization and identifying the primary and foreign keys for each file of data.

# Functional Components

(Hierarchy Chart)

# User Requirements

## MusicStoreSYS will perform Supplier administration.

* + 1. MusicStoreSYS will register a new supplier.
    2. MusicStoreSYS will update the details of a supplier.
    3. MusicStoreSYS will deregister a supplier.
    4. MusicStoreSYS will form a supplier list.

## MusicStoreSYS will perform Stock administration.

* + 1. MusicStoreSYS will register new CD stock items.
    2. MusicStoreSYS will update a CD stock item’s details.
    3. MusicStoreSYS will mark a CD stock item as discontinued.
    4. MusicStoreSYS will execute a CD stock enquiry.

## MusicStoreSYS will process Orders.

* + 1. MusicStoreSYS will arrange an order to the supplier.
    2. MusicStoreSYS will process the CD stock received from a supplier.
    3. MusicStoreSYS will record payments to the supplier.

## MusicStoreSYS will perform Administrative reporting.

* + 1. MusicStoreSYS will deliver a CD orders analysis report.
    2. MusicStoreSYS will generate a CD stock-take report.

# System Requirements

First of all, when a supplier registers with the Music Store business, the supplier’s details must be captured and updated through the user interface from a completed application form. The data that is entered is validated. The system will choose the next SupplierID to be assigned. The SupplierID and validated supplier details are stored in the Supplier File data store. The Supplier’s details can be edited on the system and when the Supplier is already registered to the system. The Supplier can also be de-registered on the system where orders are not allowed to be sent to that Supplier. The system displays the list of Suppliers in an order selected by the Manager.

Secondly, a CD’s details can be obtained and placed into the CD File when a Manager introduces a new CD to the System. It is validated and the system will choose the next CD\_ID to be assigned. The CD can also be amended where its updated details are placed into the CD File for example when a CD is re-released. The system can also discontinue a CD and remove it from the CD File. The list of CDs are displayed by the system in an order chosen by the Manager.

Thirdly, a CD can be pre-ordered by a Manager to retrieve a CD from the CD File in order to prepare for the eventual delivery of the CD to the Supplier. The system must confirm the order before carrying on with the process of delivering the CD. A confirmation message can be sent to the Supplier once the Manager accepts the chosen CD that the Supplier is ordering. Afterwards, the system records the Supplier payment made for the ordered CD where the money that the Supplier pays will be recorded by the system. Then, the system can print out a confirmation statement that the Supplier has paid for the CD.

Finally, the system can analyse the CD orders that are generated from the purchasing and pre-ordering of CDs where the money made from orders are displayed. The orders analysis is displayed in an order selected by the administrator and the data is obtained from the Orders File, OrderItems File and the CD File. The system can also provide the administrator with analysis on the CD Stock based on data in the CD File and information from the OrderItems File in a specific order chosen by the administrator.

## System Level Use Case Diagram

The Manager manages the suppliers to perform registration of the suppliers, amend the supplier’s details, de-register the supplier if required and listing the suppliers registered onto the system.

The Manager manages the CD Stock featured in the system to insert CDs into the Music Store System, amending the CDs details if a remastered edition comes out, to discontinue CD stock if needed and display the list of CDs currently in the system.

The Manager will process the order that is made by the Supplier where the Manager will place the pre-order once the Supplier reveals the CD that he/she wants to be ordered. The Manager will also enable the CD to be delivered to the Supplier by confirming the order for delivery preparation. The Manager can also the Supplier to be paid money for the chosen CD by printing out a Supplier Payment statement.

The Administrator carries out administrative analysis of the Music Store System where the CD Orders and CD Stock are analysed by the administrator for assessing the progress of the Music Store System.

MusicStoreSYS

Supplier

Manager

Administrator

## Manage Suppliers

This module provides functions to perform supplier registration, amendment, de-registration and listing.

### **Register Supplier**

This function registers a supplier’s details on the system.

Manager

Supplier

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Register Supplier** | |
| **Use Case ID** | 1.1.1 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function registers a Supplier on the Music Store system. | |
| **Preconditions** | Supplier must complete and sign a Supplier registration form. | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1**: The manager invokes the Register Supplier function.  **Step 4:** Manager enters the Supplier details:   * Supp\_Name * Phone No * Email * Street * Town/Village * County | **Step 2:** Determine the next Supp\_ID  **Step 3:** The system displays the UI.  **Step 5:** System will validate Supplier details/data entered:   * All fields need to be entered * Email address must have a valid format. * Phone number must have a valid format.   **Step 6:** Set Acc\_Bal to zero.  **Step 7**: Set status to active (“A”).  **Step 8:** Save Supplier details in Supplier File:   * Supp\_ID * Supp\_Name * Phone No * Email * Street * Town * County * Acc\_Bal * Status   **Step 9:** Display confirmation message.  **Step 10:** Clear UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Missing Name** |  | **Step 4:** Missing name detected.  **Step 5:** Display message “No Supplier Name written”.  **Step 6:** Position cursor in the Supp\_Name field and return to step 3**.** |
| **Invalid email address** |  | **Step 4:** Invalid email format entered.  **Step 5:** Display message “Email format invalid”.  **Step 6**: Position cursor in the Email field and return to step 3**.** |
| **Invalid Phone Number** |  | **Step 4:** Invalid phone number format entered.  **Step 5:** Display message “Phone Number format invalid”.  **Step 6:** Position cursor in the Phone Number field and return to step 3**.** |
| **Missing Town/Village** |  | **Step 4:** Missing town/village detected.  **Step 5:** Display message “Town/Village not detected”.  **Step 6:** Position cursor in the Town/Village field and return to step 3**.** |
| **Conclusions** | Supplier is registered on the system. | |
| **Post conditions** | Orders can now be sent to this supplier. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### **Update Supplier Details**

This function amends a supplier’s details on the system.

Supplier

Manager

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update Supplier Details** | |
| **Use Case ID** | 1.1.2 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function edits the details of a Supplier on the Music Store System. | |
| **Preconditions** | Supplier must already be registered to the system. | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1**: The manager invokes the Update Supplier Details function.  **Step 4:** The manager selects the Supplier to update.  **Step 6:** Manager updates the Supplier details:   * Supp\_Name * Phone No * Email * Street * Town/Village * County * AccBal | **Step 2:** Retrieve summary details of active Suppliers from Supplier File and load on form control.  **Step 3:** The system displays the UI.  **Step 5:** Retrieve all details of selected Supplier from Supplier File and display on UI for updating.  **Step 7:** System will validate Supplier details:   * All fields need to be entered * Email address must have a valid format. * Phone number must have a valid format.   **Step 8:** Update Supplier details in Supplier File:   * Supp\_Name * Phone No * Email * Street * Town/Village * County * AccBal   **Step 9:** Display confirmation message.  **Step 10:** Clear UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 7:** Blank field detected  **Step 8:** Display message “This field needs to be entered”.  **Step 9:** Position cursor in the offending field and return to step 7**.** |
| **Missing Name (No Supplier chosen)** |  | **Step 4:** Missing Supplier detected.  **Step 5:** Display message “No Supplier selected”. |
| **Missing Town/Village** |  | **Step 4:** Missing town/village detected.  **Step 5:** Display message “Town/Village not detected”.  **Step 6:** Position cursor in the Town/Village field and return to step 3**.** |
| **Invalid email address** |  | **Step 7:** Invalid email format entered.  **Step 8:** Display message “Email format invalid”.  **Step 9**: Position cursor in the Email field and return to step 7**.** |
| **Invalid Phone Number** |  | **Step 7:** Invalid phone number format entered.  **Step 8:** Display message “Phone Number format invalid”.  **Step 9:** Position cursor in the Phone Number field and return to step 7**.** |
| **Conclusions** | The Supplier’s details are updated on the system | |
| **Post conditions** |  | |
| **Business Rules** | Only ‘Active’ Suppliers can be updated. | |
| **Implementation Constraints** |  | |

### **Deregister Supplier**

This function sets a supplier’s status to be deregistered on the system.

Supplier

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Deregister Supplier** | |
| **Use Case ID** | 1.1.3 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function deregisters a Supplier from the Music Store system. | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1**: The Manager invokes the Deregister Supplier function.  **Step 4:** The manager selects the Supplier to de-register.  **Step 7:** Manager confirms Supplier is to be de-registered. | **Step 2:** Retrieve summary details of active Suppliers from Supplier File.  **Step 3:** The system displays the UI.  **Step 5:** Retrieve all details of selected Supplier from Supplier File and display on UI for deregistration.  **Step 6:** System asks user “Are you sure you want to de-register this Supplier?”  **Step 8**: Set status to inactive (‘I’).  in Supplier File  **Step 9:** Display confirmation message.  **Step 10:** Clear UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Conclusions** | Supplier is deregistered on the system. | |
| **Post conditions** | Orders cannot be sent to this Supplier. | |
| **Business Rules** | Only active suppliers may be de-registered. | |
| **Implementation Constraints** |  | |

### **Generate Supplier List**

This function will form a Supplier List based on the Suppliers on the System.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Generate Supplier List** | |
| **Use Case ID** | 1.1.4 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function generates a Supplier list in order of Supplier Id or Supplier name | |
| **Preconditions** | Suppliers must be registered onto the system. | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1**: The manager invokes the Generate Supplier List function. | **Step 2:** Retrieve Supplier details from Supplier File in order of Supplier ID and load on form control  **Step 3:** The system displays the UI.  **Step 7:** Display list on screen. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Order Selected** | **Step 4:** Manager selects order for listing:   * SuppID * Supp Name | **Step 5:** Retrieve Supplier details from Supplier File in order selected and loads on form control  **Step 6:** Display list on screen. |
| **Conclusions** | A list of Suppliers is generated by the system. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

**Example of Supplier List in order of Supp\_ID (Original before edited headings in Visual Studio):**

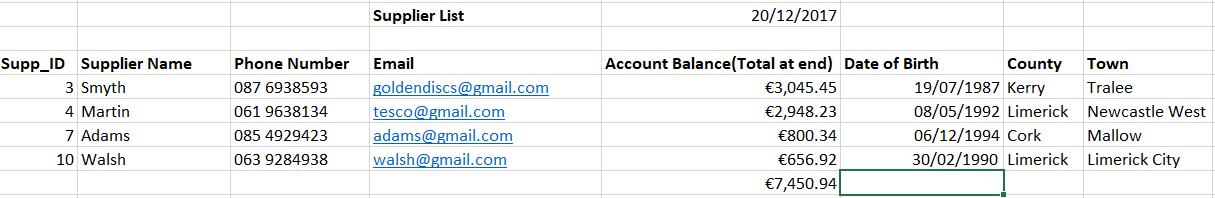


Figure 1: Supplier Listing Excel File (Example)

## Manage Stock

This module provides functions for inserting, amending (updating), stop the insertion of CDs and displaying a list of CDs available in the CD Stock File.

### **New CD**

This function allows new CDs to be placed into a CD Stock File.

Supplier

Manager

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **New CD** | |
| **Use Case ID** | 1.2.1 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function inserts the first set of new CDs to a CD Stock File | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1**: The Manager invokes the New CD function.  **Step 4:** Manager enters the CD details:   * Album\_Name * Artist * Year\_Released * Artist\_Label * Cost Price * Qty | **Step 2:** The System determines the next CD\_ID  **Step 3:** The system displays the UI.  **Step 5:** System will validate Supplier details:   * All fields need to be entered * Cost must be numeric and > 0 * Qty must be numeric and > 0   **Step 6:** Set availability to available (“a”)  **Step 8:** Save CD details in CD File:   * CD\_ID * Album\_Name * Artist * Year\_Released * Artist\_Label * Cost Price * Qty * Availability   **Step 9**: Display confirmation message.  **Step 10:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Artist Name Missing** |  | **Step 5:** No artist name filled in text field.  **Step 6:** Display message “This artist field needs to be entered”.  **Step 7**: Position cursor in the Artist field and return to step 3**.** |
| **Missing Album Name** |  | **Step 5:** No album name in text field.  **Step 6:** Display message “This album field must be entered”.  **Step 7**: Position cursor in the Album field and return to step 3**.** |
| **Low Quality Stock** |  | **Step 5:** Quality of stock is negative or zero.  **Step 6:** Display message “This quality field needs to be re-entered”.  **Step 7**: Position cursor in the Qty field and return to step 3 |
| **Cost too low / negative** |  | **Step 5:** Cost is negative or zero.  **Step 6:** Display message “This Cost field needs to be re-entered”.  **Step 7**: Position cursor in the Cost field and return to step 3 |
| **Conclusions** | The new CD is added to the Stock File. | |
| **Post conditions** | This CDS can be ordered from the supplier | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### **Update CD**

This function enables new CDs to be amended.

Manager

Supplier

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update CD** | |
| **Use Case ID** | 1.2.2 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function updates the details of a CD. | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1**: The Manager invokes the Update New CDs function.  **Step 4:** The manager selects the CD to update.  **Step 6:** Manager updates the CD details:   * CD\_ID (Optional) * Album\_Name * Artist * Year\_Released * Artist\_Label * Cost Price * Qty | **Step 2:** Retrieve CD details from CD File and load on form control  **Step 3:** The system displays the UI.  **Step 5:** Retrieve all details of selected CD from CD File and display on UI for update.  **Step 7:** System will validate CD details:   * All fields need to be entered * Cost must be numeric and > 0 * Qty must be numeric and > 0   **Step 8:** Update CD details in CD File:   * CD\_ID * Album\_Name * Artist * Year\_Released * Artist\_Label * Cost Price * Qty * Availability     **Step 9**: Display confirmation message.  **Step 10:** Clear UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 7:** Blank field detected  **Step 8:** Display message “This field needs to be entered”.  **Step 9:** Position cursor in the offending field and return to step 3**.** |
| **Missing Album Name** |  | **Step 7:** No album name in text field.  **Step 8:** Display message “This album field must be entered”.  **Step 9**: Position cursor in the Email field and return to step 3. |
| **Artist Name Missing** |  | **Step 7:** No artist name filled in text field.  **Step 8:** Display message “This artist field needs to be entered”.  **Step 9**: Position cursor in the offending field and return to step 3**.** |
| **Cost too low / negative** |  | **Step 7:** Cost is negative or zero.  **Step 8:** Display message “This Cost field needs to be re-entered”.  **Step 9**: Position cursor in the Cost field and return to step 3. |
| **Low Quality Stock** |  | **Step 7:** Quality of stock is negative or zero.  **Step 8:** Display message “This quality field needs to be re-entered”.  **Step 9**: Position cursor in the Qty field and return to step 3. |
| **Conclusions** | The CD details are updated | |
| **Post conditions** |  | |
| **Business Rules** | Only available CDS can be updated | |
| **Implementation Constraints** |  | |

### **Discontinue CDs**

This function terminates various CDs from the CD Stock and out of the Music Store system.

Manager

Supplier

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Discontinue CDS** | |
| **Use Case ID** | 1.2.3 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function discontinues the CD that is in stock. | |
| **Preconditions** | A bigger number of CDs than 2 required to discontinue some CDs. | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1**: The Manager invokes the Discontinue CDs function.  **Step 4:** Manager selects the CDs to discontinue.  **Step 7**: Manager confirms whether or not to discontinue the CDs. | **Step 2:** Retrieve CD details from CD File  **Step 3:** The system displays the UI.  **Step 5:** Retrieve all details of selected CDs from CD Stock File and display on UI for discontinuation.  **Step 6:** System asks user “Are you sure you want to remove these CDs from this System?”  **Step 8:** Set availability to unavailable (‘u’) in CD File.  **Step 9:** Display confirmation message.  **Step 10:** Clear UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Keeping CDs available in Stock** |  | **Step 8:** Keep availability to available (‘a’)  **Step 9:** Display confirmation message of the CDs remaining in Stock. |
| **Conclusions** | The CDs are discontinued. | |
| **Post conditions** | This CD cannot be ordered. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### **Query CDS**

This function enables a list of CDs to be displayed according to the question asked.

Supplier

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Query CDS** | |
| **Use Case ID** | 1.2.4 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function allows the System to query for the list of CDs. | |
| **Preconditions** | The CDs must be included in the CD File and inserted into the System. | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1**: The Manager invokes the Query CDs function.  **Step 5:** Enter a “Title” and find all CDs. Manager chooses an order for listing to answer the question:   * CD\_ID * Artist\_Label | **Step 2:** Retrieve CD details from CD File in order of CD ID and load on form control.  **Step 3:** The system displays the UI.  **Step 4:** System asks user how you want the list of CDs to be displayed  **Step 6:** Retrieve CD details from CD File in order selected and loads on form control.  **Step 7:** Display list on screen.  **Step 8:** Clear UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | A list of CDs is displayed in accordance with a certain query by the system. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

**Example of CD List in order of CD\_ID (Original before edited headings in Visual Studio):**



Figure 2: CD List Excel File (Example)

## Process Orders

This module provides functions for placing pre-orders on CDs, receiving the CD order and paying the Supplier.

### **Place CD Pre-Order**

This function allows a CD stored within the CD Stock in the system to be pre-ordered following the Supplier’s permission to order the CD.

Supplier

Manager

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Pre-order CD** | |
| **Use Case ID** | 1.3.1 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function enables the System to place a pre-order for a CD. | |
| **Preconditions** | Requires a CD to be already placed into the system. | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1**: The Manager invokes the Pre-order CD function.  **Step 5:** Manager selects the Supplier whose order is being sent to.  **Step 7:** Manager chooses the CD to be ordered by the Supplier. If there are more CDs, go to this step.  **Step 9:** Manager decides either to proceed with the order (“y”) or to cancel the order (“n”). | **Step 2:** Determine the next Ord\_ID.  **Step 3:** Retrieve all Suppliers from Supplier File.  **Step 4:** The system displays the Supplierson UI.  **Step 6:** Retrieve all details of selected Supplier from Supplier File and display on UI. Retrieve all the CD’s supplied by that Supplier.  **Step 8:** Retrieve all details of selected CD from CD File and display on UI for pre-order choice.  **Step 10**: Save order details in Order File.   * OrderID * SuppID * OrdDate * OrdValue * Status = “Placed”   **Step 11**: Save order details in OrderItems File.   * OrdID * CD\_ID * Qty   **Step 12**: Display confirmation message that the CD has been pre-ordered from the Supplier.  **Step 13**: Email the order to the Supplier.  **Step 14:** Clear UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Refusing the Pre-order** |  | **Step 12**: Display confirmation message that the order will not be placed  **Step 13:** Clear Message and return to Step 4. |
| **Conclusions** | A CD is pre-ordered. | |
| **Post conditions** | The Pre-Order is sent to the System, so it can prepare for the order to be delivered to the Supplier and prepare the payment for the Supplier. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### **Receive CD Order**

This function allows the CD to be delivered from the System to the Supplier.

Supplier

Manager

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Receive CD Order** | |
| **Use Case ID** | 1.3.2 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function will enable the Supplier to receive the CD order on the day of the CD release. | |
| **Preconditions** | Requires the CDs to be pre-ordered before it can be sent and given to the Supplier. | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1**: The Manager invokes the Receive CD Order function.  **Step 3:** Enter the Supplier who is about to receive the order of CDs.  **Step 6**: Manager selects the order that was received.  **Step 9:** Manager chooses either yes (“Y”) or no (“No”) for confirming delivery of order.  **Step 11:** Set Order Status to “Received” in Order File. | **Step 2:** Display the UI when retrieving the CD details from the CD file and load on form control.  **Step 4:** Retrieve Order details from Order File and OrderItems for this Supplier and load on form control.  **Step 5:** The system displays the UI.  **Step 7:** Retrieve details and display on UI for viewing.  **Step 8:** The System asks the user to confirm if the CDs that the Supplier chose are the ones that are going to be delivered**.**  **Step 10:** Increase Qty in CD File.  **Step 12**: Clear UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Refusing the delivery order.** |  | **Step 9**: Display confirmation message that the order will not be delivered.  **Step 10:** Clear Message and return to Step 2. |
| **Conclusions** | The Supplier receives the confirmation that their chosen CD will be delivered in a matter of hours or days. | |
| **Post conditions** | The payment is sent to the Supplier. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### **Record Supplier Payment**

This function enables the System to confirm that we pay the Supplier for the CD after the CD is pre-ordered.

Manager

Supplier

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Record Supplier Payment** | |
| **Use Case ID** | 1.3.3 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function will enable the system to pay the Supplier for the CDs ordered. | |
| **Preconditions** | The CDs must already be pre-ordered and must be confirmed for delivery to the Supplier. | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System** |
|  | **Step 1**: The Manager invokes the Pay Supplier function.  **Step 5:** Manager selects the Supplier who supplied an order and enters  -Amount  -Pay Date | **Step 2:** Determine the next PaymentID.  **Step 3:** The system displays the UI.  **Step 4:** Retrieve all Suppliers and display on UI from Supplier/Order Files  **Step 6**: TheSystem (Payment File) records a certain amount of money, e.g €12.00 that the Supplier pays for the CD and saves it in the Payment File.   * PaymentID * SuppID * Order\_ID * Amount * PaymentDate   **Step 7:** Display confirmation message.  **Step 8:** The System prints out a statement confirming that the Supplier has paid for the CD.  **Step 9:** Clear UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The Payment statement is given to the Supplier which was stored in the Payment File. | |
| **Post conditions** | The Supplier receives the ordered CD and a payment statement. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Reporting Administrative Analysis

This module carries out functions for inspecting the CD Sales and analysing the CD Stock available in the system.

### **Analyse CD Orders**

This function allows an administrator to analyse the orders generated by the CDs when ordered and purchased.

Administrator

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Analyse CD Orders** | |
| **Use Case ID** | 1.4.1 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Manager | |
| **Description** | This function analyses the CD Orders and the money it generated from these orders. | |
| **Preconditions** | The orders must already be calculated for each CD and processed before analysis. | |
| **Trigger** | None | |
| **Expected Scenario** | **Administrator** | **System** |
|  | **Step 1**: The administrator invokes the Analyse CD Orders function.  **Step 5:** Administrator chooses an order for analysing the results.   * CD\_ID * Album\_Name * Highest\_Sale * Sale\_Price | **Step 2:** Retrieve CD details from CD File in addition to the Orders made from the CD and load on form control.  **Step 3:** The system displays the UI.  **Step 4:** System asks user in what order they want CD Orders to be displayed.  **Step 6:** Retrieve CD details from CD File along with the Orders made from the CD from the Orders File in the order selected along with the OrderItems to display the items ordered in accordance with the OrdID and loads on form control.  **Step 7:**  Display analysis of CD Orders with the details of the CDs in the order chosen.  **Step 8**: Clear UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The CD orders are analysed. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### **Analyse CD Stock**

This function allows the Administrator to analyse the CD Stock File within the system.

Manager

Administrator

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Analyse CD Stock** | |
| **Use Case ID** | 4.5.2 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Manager | |
| **Description** | This function will analyse the CD stock. | |
| **Preconditions** | The CD Stock must be present with more than one CD contained before the analysis can be carried out. | |
| **Trigger** | None | |
| **Expected Scenario** | **Administrator** | **System** |
|  | **Step 1**: The administrator invokes the Analyse CD Stock function  **Step 5:** Administrator chooses an order for analysing the results   * CD\_ID * Artist * No\_Sold (Qty) | **Step 2:** Retrieve CD details from CD File and Order Items’ details from the OrderItems File and load on form control  **Step 3:** The system displays the UI.  **Step 4:** System asks user in what order they want CD Stock to be displayed.  **Step 6:** Retrieve CD details from CD File and Order Items’ details from the OrderItems File in the order selected and loads on form control.  **Step 7:** Display analysis of CD Stock with the details of the CDs and stock in the order chosen.  **Step 8**: Clear UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The CD Stock is analysed. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

# System Model

**Data Stores:**

**D1:** Supplier File

**D2:** CD File

**D3:** Payment File

**D4:** Order File

**D5:** OrderItems File

**External Entity:**

Supplier

**Processes:**

**P1: Manage Suppliers**

**P 1.1** Register Supplier

**P 1.2** Update Supplier Details

**P 1.3** De**-**register Supplier

**P 1.4** Generate Supplier List

**P2: Manage Stock**

**P 2.1** Insert CDs

**P 2.2** Update CDs

**P 2.3** Discontinue CDs

**P 2.4** Query CDs

**P3: Process Orders**

**P 3.1** Place Pre-order

**P 3.2** Receive Order

**P 3.3** Pay Supplier

**P4: Administration Analysis**

**P 4.1** Analyse Sales

**P 4.2** Analyse Stock

## Level-0 DFD

Registration

Supplier

MusicStoreSYS

CD Order

## Level-1 DFD

D1

Supplier File

Order Details

Supplier Details

P4

Report Analysis

P1

Manage Suppliers

Supplier Details

CD Details

D5

OrderItems File

D4

Order File

Order Details

Order Details

D2

CD File

CD Details

P3

Process Orders

Selected Supplier Order

P2

Manage Stock

Supplier Details

CD Details

Supplier

Order

Payment Details

Payment

D3

Payment File

Supplier

## Level-2 DFD Manage Suppliers

P1.4

Generate Supplier List

Supplier Details

Supplier Details

P1.1

Register Supplier

Supplier Details

D1

Supplier File

Supplier Details

Supplier

Supplier

Updated Supplier Details

Updated Supplier Details

Supplier Details

P1.3

Deregister Supplier

Supplier Status

Supplier Details

P1.2

Update Supplier Details

## Level-2 DFD Manage Stock

P2.4

Query CDS

P2.1

Insert New CDS

CD Details

CD Details

D2

CD File

CD Details

Updated CD Details

CD Status

P2.3

Discontinue CDS

P2.2

Update New CDS

CD Details

## Level-2 DFD Process Orders

Supplier

D4

Order File

Selected Supplier Details

P3.1

Place Pre-Order

Order Details

Order Details

D5

OrderItems File

Order Details

D4

Order File

P3.2

Receive Order

Status “Received”

CD Details

CD Details

Order

Order Details

D2

CD File

D5

OrderItems File

Supplier Details

Supplier Details

Selected CD Details

D1

Supplier File

D2

CD File

P3.3

Record Supplier Payment

Payment

Supplier

Payment Statement

Payment Details

Selected Supplier Details

Supplier Details

D3

Payment File

D1

Supplier File

## Level-2 DFD Administration Analysis

Order Details

Order Details

D5

OrderItems File

CD Details

D2

CD File

CD Details

P4.1

Analyse CD Orders

P4.2

Analyse CD Stock

# Data Model (Class Diagram)

This is the data model that will be established to represent the data in the files for the Supplier, CD, Payment, Order and OrderItems that will be stored in a database.

## Class Diagram

## Relational Schema

***Suppliers (***Supp\_ID, Supp\_Name, PhoneNo, Email, Street, Town/Village, County\_Code, Acc\_Bal, Status***)***

***CDs (***CD\_ID, Album\_Name, Artist, Year\_Released, Artist\_Label, Cost\_Price, Qty, Availability***)***

***Orders (***OrderID, OrdDate, OrdValue, Status, SuppID***)***

***OrderItems (***OrdID, CD\_ID, Qty, Unit\_Cost, OrderID***)***

***Payments (***PaymentID, OrderID, Amount, PaymentDate, Supp\_ID***)***

***Counties (***County\_Code,Name***)***

## Database Schema

**Schema:** MusicStoreSYS.sql

**Relation: Suppliers**

Attributes:

Supp\_Id numeric (3) NOT NULL

Supp\_Name char (20) NOT NULL

PhoneNo numeric (15)

Email char (30)

Street char (25) NOT NULL

Town/Village char (20) NOT NULL

County\_Code char (20) NOT NULL

Acc\_Bal numeric (5, 2)

Status char (1)

**Primary Key:** Supp\_ID

[**Foreign Key:**] County\_Code

**Relation: CDS**

Attributes:

CD\_ID numeric (5) NOT NULL

Album\_Name char (30) NOT NULL

Artist char (30) NOT NULL

Year\_Released numeric (4)

Artist\_Label char (15) NOT NULL

Cost\_Price numeric (5, 2) NOT NULL

Qty numeric (3)

Availability char (1)

SuppID char (3) NOT NULL

**Primary Key:** CD\_ID

[**Foreign Key:**] Supp\_Id References Suppliers

**Relation: Orders**

Attributes:

OrderID numeric (4) NOT NULL

OrdDate Date NOT NULL

OrdValue numeric (5, 2)

Status char (1)

SuppID char (3) NOT NULL

**Primary Key:** OrderID

[**Foreign Key:**] Supp\_Id References Suppliers

**Relation: OrderItems**

Attributes:

OrdID numeric (5) NOT NULL

CD\_ID numeric (5) NOT NULL

Qty numeric (5)

Unit\_Cost numeric (5, 2)

OrderID numeric (4) NOT NULL

**Primary Key:** OrdID,CD\_ID

[**Foreign Key:**] OrderID References Orders

[**Foreign Key:**] CD\_ID References CDs

**Relation: Payments**

Attributes:

PaymentID char (6) NOT NULL

OrderID numeric (4) NOT NULL

Amount numeric (5, 2)

PaymentDate Date

Supp\_ID numeric (3) NOT NULL

**Primary Key:** PaymentID,OrderID

[**Foreign Key:**] Supp\_Id References Suppliers

**Relation: Payments**

Attributes:

County\_Code char (2)

Name char (15)

**Primary Key:** County\_Code

# Conclusion

Firstly, one change that would have been made to the system during the prototype section was to form another analysis module for the administrator to analyse the suppliers’ payments made for numerous CDs where suppliers frequently pre-order CDs at multiple stages of the year and spend an enormous amount of money on various CDs available in the market. This process did not make the cut as it would have made the MusicStoreSYS more complicated and complex to use throughout the entire program.

In conclusion, a music store system will be created from this master document which provides a manager with the opportunity to pre-order CDs for the chosen Suppliers. The MusicStoreSYS System contains plenty of functionality for the Manager as the design of the system is uniquely designed to form an original system inspired by other pre-ordering systems. This system can eventually be modified once the basic functionality is implemented into the system which would allow the ordering of CDs and registration of Suppliers to become much easier for better access to the system. It would also allow the details of the orders of CDs by Suppliers and the stock of CDs available to be analysed in a much simpler way so these details can be listed in a database.

# Appendices

## Appendix A – Title

(CD Image – Edited outskirts of image in Paint)

Proximo, Dr., 2017. CD. [photograph] Available at <https://i1.wp.com/www.geeksandbeats.com/wp-content/uploads/2017/10/cd.jpg?w=693&ssl=1> [Accessed 09 December 2017]

## Appendix B – Title