Requirements Engineering

DVDSYS

Submitted By: Kenneth Malon (T00195873)

Computing with Games Development/Software Development/Multi Media

Date Submitted: 04/05/2018

**Table of Contents**

[1. Introduction/overview 4](#_Toc505257932)

[2. Functional Components 5](#_Toc505257933)

[3. User Requirements 6](#_Toc505257934)

[1. DVDSYS will perform DVD administration. 6](#_Toc505257935)

[1.1. DVDSYS will set the rate for DVD rentals. 6](#_Toc505257936)

[1.2. DVDSYS will register a new DVD rental item. 6](#_Toc505257937)

[1.3. DYDSYS will allow a DVD item’s details to be updated. 6](#_Toc505257938)

[1.4. DVDSYS will record a DVD item as discontinued. 6](#_Toc505257939)

[4. System Requirements 7](#_Toc505257940)

[4.1. System Level Use Case Diagram 7](#_Toc505257941)

[4.2. Manage DVD 9](#_Toc505257942)

[4.2.1. Set Rate 9](#_Toc505257943)

[4.2.2. Register DVD 11](#_Toc505257944)

[4.2.3. Update DVD 13](#_Toc505257945)

[4.2.4. Discontinue DVD 16](#_Toc505257946)

[4.3. Manage Members 18](#_Toc505257947)

[4.3.1. Register Members 18](#_Toc505257948)

[4.3.2. Update Members 20](#_Toc505257949)

[4.3.3. De-Register Member 24](#_Toc505257950)

[4.4. Process Rental 25](#_Toc505257951)

[4.4.1. Execute Rentals 25](#_Toc505257952)

[4.4.2. Process Return 27](#_Toc505257953)

[4.5. Perform Admin 29](#_Toc505257954)

[4.5.1. Rental Analysis 29](#_Toc505257955)

[5. System Model 31](#_Toc505257956)

[5.1. Level-0 DFD 32](#_Toc505257957)

[5.2. Level-1 DFD 33](#_Toc505257958)

[5.3. Level-2 DFD (Process P1: Manage DVD) 34](#_Toc505257959)

[5.4. Level-2 DFD (Process P2: Manage Members) 35](#_Toc505257960)

[5.5. Level-2 DFD (Process P3: Process Rentals) 36](#_Toc505257961)

[6. Data Model (Class Diagram) 37](#_Toc505257962)

[6.1. Class Diagram 37](#_Toc505257963)

[6.2. Relational Schema 38](#_Toc505257964)

[6.3. Database Schema 38](#_Toc505257965)

[7. Conclusion 42](#_Toc505257966)

[8. Appendices 43](#_Toc505257967)

[8.1. Appendix A – Title 43](#_Toc505257968)

[8.2. Appendix B – Title 43](#_Toc505257969)

# Introduction/overview

**DVDSYS** is a rental system that involves the execution of functions such as Managing DVD’s, Members, Processing Rentals and Generating Revenues.

The primary functions of the system are adding, setting, updating and removing DVD and member details. It also carries out processing rentals where the system executes rentals and process returns. The admin function regenerates revenues in the system.

# Functional Components

# User Requirements

# DVDSYS will perform DVD administration.

# DVDSYS will set the rate for DVD rentals.

# DVDSYS will register a new DVD rental item.

# DYDSYS will allow a DVD item’s details to be updated.

# DVDSYS will record a DVD item as discontinued.

1. DVDSYS will perform Member administrator.
   1. DVDSYS will register a new member.
   2. DVDSYS will update members.
   3. DVDSYS will de-register members.
2. DVDSYS will process Rentals.
   1. DVDSYS will execute Rentals.
   2. DVDSYS will process return Rentals.
3. DVDSYS will perform administrative reporting.
   1. DVDSYS will generate a rental revenue report.
   2. DVDSYS will generate member count report.

# System Requirements

|  |  |  |
| --- | --- | --- |
| **Functional Requirements** | **Non-Functional Requirements** | **Domain Requirements** |
| * To allow for a DVD and members to be added. * To allow DVD and members to be updated its details e.g. setting rates, updating, and discontinuing * To allow the system processing the rentals and returns. * To allow the system to generate revenues of the rentals. | * The system must be easy and quick to use. Each function happens on one form. * The admin is the only user that can access the system. He has he’s own unique username and password that can access the system. | * The admin will have access to the system in his office and the system will not be able to be accessed online. |

## System Level Use Case Diagram

DVDSYS will allow the admin to carry out functions for management of the DVDs. These functions include adding, setting rate, updating and discontinuing DVD.

DVDSYS will allow the admin to carry out the functions for managing costumer. These function includes registering, updating and de-registering members.

DVDSYS will allow the admin to carry out the function for processing rentals. These includes executing rentals and processing returns.

DVDSYS will allow the admin to carry out the function that allows the system to generate revenues from rentals and generate member count.

DVDSYS

Admin

Members

## Manage DVD

Outline the functions to be delivered in this component.

### Set Rate

This function will set the rates for DVD items in the system.

Admin

<<Extends>>

<<Includes>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Set Rate** | |
| **Use Case Id** | 4.2.4 | |
| **Priority** | High | |
| **Source** | Administrator | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | None | |
| **Description** | This function will set the rates for DVD. | |
| **Preconditions** | DVD details must be active in the system. | |
| **Trigger** | None | |
| **Expected Scenario** | **Administrator** | **System Response** |
|  | **Step 1:** Administrator starts Set Rate functions  **Step 3:** Administrator Enters required data:   * Rate\_Code * Description * Rate | **Step 2:** Display UI  **Step 4:** System validates data:   * All fields must be entered * Rate code must be alphabetic and length 2 * Rate must be numeric and greater than 0 * Rate code must not already be registered   **Step 5:** Save Rate details in Rates File:   * Rate\_Code * Description * Rate   **Step 6:** Display confirmation message  **Step 7:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 4:** A blank field detected  **Step 5:** Display message “This field must be entered”  **Step 6:** Position cursor in the offending field and return to step 3 |
| **Rate Invalid** |  | **Step 4:** Rate is alphabetic or less than 0  **Step 5:** Display message “Rate must be numeric and greater than 0”  **Step 6:** Position cursor in the offending field and return to step 3 |
| **Rate code already registered** |  | **Step 4:** Rate code has already registered.  **Step 5:** Display message “Rate code must not be already registered”  **Step 6:** Position cursor in the offending field and return to step 3 |
| **Conclusions** | Rate code has been registered. | |
| **Post conditions** | DVD can now be assigned this rate code. | |
| **Business Rules** | The rate code must be unique. | |
| **Implementation Constraints** |  | |

### Register DVD

This function will register a new DVD item in the system.

Admin

<<Extends>>

<<Includes>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Register DVD** | |
| **Use Case Id** | 4.2.1 | |
| **Priority** | High | |
| **Source** | Administrator | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | None | |
| **Description** | This function will register DVD details in the system. | |
| **Preconditions** | None | |
| **Trigger** | None | |
| **Expected Scenario** | **Administrator** | **System Response** |
|  | **Step 1:** Administrator starts Register DVD function  **Step 3:** Administrator Enters DVD details:   * Title * Description * Release\_Year * Category | **Step 2:** Display UI  **Step 4:** System validates DVD details:   * All fields must be entered * Field must not contain numeric * Fields must contain the correct values   **Step 5:** DVD\_Id is automatically assigned and can’t be changed  **Step 6:** Register  **Step 7:** Save DVD details in DVD File:   * DVD\_Id * Title * Description * Release\_Year * Category   **Step 8:** Display confirmation message  **Step 9:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 4:** A blank field detected  **Step 5:** Display message “This field must be entered”  **Step 6:** Position cursor in the offending field and return to step 3 |
| **Numeric data detected** |  | **Step 4:** A numeric value detected in the text field  **Step 5:** Display message “This field must not be numeric”  **Step 6:** Position cursor in the offending field and return to step 3 |
| **Invalid Release Year** |  | **Step 4:** An invalid format entered for release year  **Step 5:** Display message “This field must be numeric”  **Step 6:** Position cursor in the offending field and return to step 3 |
| **Conclusions** | DVD is registered in the system. | |
| **Post conditions** | DVD is now active in the system. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Update DVD

This function will update DVD item in the system.

Admin

<<Extends>>

<<Includes>>

**Administrator System**

Start function

Display UI

Enter DVD title

Retrieve a summary of DVD title

Display UI

Select DVD to update

Retrieve all details for selected DVD

Display UI for Updating

Administrator updates required field

System Validates Data

Valid?

N

Error Message

Y

System updates DVD File

Confirm Dialog

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update DVD** | |
| **Use Case Id** | 4.2.2 | |
| **Priority** | High | |
| **Source** | Administrator | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | None | |
| **Description** | This function will update DVD details that are currently in the system. | |
| **Preconditions** | DVD must be already registered in the system. | |
| **Trigger** | None | |
| **Expected Scenario** | **Administrator** | **System Response** |
|  | **Step 1:** Administrator starts Update DVD functions  **Step 3:** Enter DVD title  **Step 5:** Select DVD to Update  **Step 6:** Administrator Updates DVD details:   * Category\_Id * DVD\_Rate | **Step 2:** Display UI  **Step 4:** Retrieves a summary of DVD title including DVD\_Id  **Step 7:** System validates DVD details:   * All fields must be entered * Field must not contain numeric * Fields must contain the correct values   **Step 8:** System Updates DVD details from DVD File:   * DVD\_Id * Category\_Id * Title * DVD\_Rate   **Step 9:** Set Status(“A” or “U”)  Available or Unavailable  **Step 10:** Display confirmation message  **Step 11:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 4:** A blank field detected  **Step 5:** Display message “This field must be entered”  **Step 6:** Position cursor in the offending field and return to step 7 |
| **Numeric data detected** |  | **Step 4:** A numeric value detected in the text field  **Step 5:** Display message “This field must not be numeric”  **Step 6:** Position cursor in the offending field and return to step 7 |
| **Invalid DVD rate** |  | **Step 4:** An invalid format entered for DVD rate  **Step 5:** Display message “This field must be numeric”  **Step 6:** Position cursor in the offending field and return to step 7 |
| **Conclusions** | DVD details is updated in the system. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Discontinue DVD

This function will discontinue DVD item in the system.

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Discontinue DVD** | |
| **Use Case Id** | 4.2.3 | |
| **Priority** | High | |
| **Source** | Administrator | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | None | |
| **Description** | This function will remove DVD details in the system. | |
| **Preconditions** | DVD details must be in the system. | |
| **Trigger** | None | |
| **Expected Scenario** | **Administrator** | **System Response** |
|  | **Step 1:** Administrator starts Discontinue DVD functions  **Step 3:** Enter DVD title  **Select 5:** Select DVD to discontinue | **Step 2:** Display UI  **Step 4:** Retrieves summary of DVD titles and DVD ID  **Step 6:** Remove DVD details in DVD File:   * DVD\_Id * Category\_Id * Title * Description * Release\_Year * Status   **Step 7:** Display confirmation message  **Step 8:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | DVD details is removed in the system. | |
| **Post conditions** | DVD is not available for renting. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Manage Members

This component includes function for management of members. These function includes registration, updating members and de-registering members.

### Register Members

This function will register new members in the system.

Costumer

Staff

<<Includes>>

<<Extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Register Members** | |
| **Use Case Id** | 4.3.1 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Customer | |
| **Description** | This function will register new members in the system | |
| **Preconditions** | Costumer must complete and sign registration form | |
| **Trigger** | None | |
| **Expected Scenario** | **Manager** | **System Response** |
|  | **Step 1:** Staff starts Register Members function  **Step 3:** Staff enters  Costumer details:   * Name * Email * Address * Contact No. | **Step 2:** Display UI  **Step 4:** System validates DVD details:   * All fields must be entered * Field must not contain numeric values only * Email address must have valid format * Address must have valid format   **Step 5**: Member\_Id is automatically assigned and can’t be changed  **Step 6:** Register Customer  **Step 7:** Save Costumer details in the Member File:   * Member\_Id * Name * Email * Address * Contact No.   **Step 8:** Display confirmation message  **Step 9:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 4:** A black field detected  **Step 5:** Display message “This field must be entered”  **Step 6:** Position cursor in the offending field and return to step 3 |
| **Numeric data detected** |  | **Step 4:** A numeric value detected in the text field  **Step 5:** Display message “This field must be numeric”  **Step 6:** Position cursor in the offending field and return to step 3 |
| **Invalid Email address** |  | **Step 4:** An invalid format entered for email  **Step 5:** Display message “Email format invalid”  **Step 6:** Position cursor in Email  Field and return to step 3 |
| **Invalid Address** |  | **Step 4:** An invalid format entered for Eir\_Code  **Step 5:** Display message “Address format invalid”  **Step 6:** Position cursor in Address Field and return to step 3 |
| **Conclusions** | Costumer is registered as a member. | |
| **Post conditions** | Costumer may now be able to rent DVD. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Update Members

This function will update current member details in the system.

Member

Administrator

<<Includes>>

<<Extends>>

**Administrator System**

Start function

Display UI

Enter Member

Retrieve a summary member names

Display UI

Select member to update

Retrieve all details for selected member

Display UI for Updating

Administrator updates required field

System Validates Data

Valid?

N

Error Message

Y

System updates Member File

Confirm Dialog

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update Member** | |
| **Use Case Id** | 4.3.2 | |
| **Priority** | High | |
| **Source** | Administrator | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Member | |
| **Description** | This function will update members details in the system | |
| **Preconditions** | Costumer must give notice. | |
| **Trigger** | None | |
| **Expected Scenario** | **Administrator** | **System Response** |
|  | **Step 1:** Administrator starts Update Member detail function  **Step 3:** Enter member name  **Step 5:** Select member to update  **Step 7:** Administrator updates  Member details:   * Name * Email * Address * Contact No. | **Step 2:** Display UI  **Step 4:** Retrieve a summary of members with name  **Step 6:** Retrieve member details from Member File:  **Step 8:** System validates DVD details:   * All fields must be entered * Field must not contain numeric values only * Email address must have valid format * Address must have valid format   **Step 9:** System Updates member details from member File:   * Name * Email * Address * Contact No.   **Step 10:** Set Status to “A” or “D”  Active or Deactivate  **Step 11:** Display confirmation message  **Step 12:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 4:** A black field detected  **Step 5:** Display message “This field must be entered”  **Step 6:** Position cursor in the offending field and return to step 3 |
| **Numeric data detected** |  | **Step 4:** A numeric value detected in the text field  **Step 5:** Display message “This field must be numeric”  **Step 6:** Position cursor in the offending field and return to step 3 |
| **Invalid Email address** |  | **Step 4:** An invalid format entered for email  **Step 5:** Display message “Email format invalid”  **Step 6:** Position cursor in Email Field and return to step 7 |
| **Invalid Address** |  | **Step 4:** An invalid format entered for Address  **Step 5:** Display message “Eir\_Code format invalid”  **Step 6:** Position cursor in Address field and return step 7 |
| **Conclusions** | Member details has been updated | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### De-Register Member

This function will de-register members in the system.

Member

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **De-Register Member** | |
| **Use Case Id** | 4.3.3 | |
| **Priority** | High | |
| **Source** | Administrator | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | Member | |
| **Description** | This function will remove deregister current members in the system | |
| **Preconditions** | Costumer details must be in the system. | |
| **Trigger** | None | |
| **Expected Scenario** | **Administrator** | **System Response** |
|  | **Step 1:** Administrator starts Update Member detail function  **Step 3:** Administrator enters  Member details:   * Name   **Step 5:** Select Member to de-register | **Step 2:** Display UI  **Step 4:** Retrieves summary of members including Member\_Id from Member File.  **Step 6:** Display confirmation message  **Step 7:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | Member details has been de-registered from the system | |
| **Post conditions** |  | |
| **Business Rules** | What business rules, if any, must be factored into the software | |
| **Implementation Constraints** | Specify any non-functional requirements that may impact the realisation of the use case | |

## Process Rental

This component includes function for management of DVD rentals. These functions are mainly for processing rentals items.

### Execute Rentals

This function will process rental DVD item in the system.

Staff

Member

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Execute Rental** | |
| **Use Case Id** | 4.4.1 | |
| **Priority** | High | |
| **Source** | Staff | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** | Member | |
| **Description** | This function will process DVD rentals. | |
| **Preconditions** | None | |
| **Trigger** | None | |
| **Expected Scenario** | **Staff** | **System Response** |
|  | **Step 1:** Staff starts Process Rentals function  **Step 3:** Staff Enters Member Name  **Step 5:** Staff Selects a Member for rentals  **Step 6:** Staff Enters DVD Title  **Step 8:** Staff Selects a DVD for rentals  **Step 9:** Staff enters the Rental\_Date and Return\_Date | **Step 2:** Display UI  **Step 4:** Retrieves a summary for Members names this includes member ID.  **Step 7:** Retrieves a summary for DVD names  **Step 10:** System validates Rental details and Member details:   * All fields must be entered * Fields must contain the correct values   **Step 11:** Save DVD details in Rental File:   * DVD\_Id * Member\_Id * Name * Title * Description * Release\_Year * Status * Rental\_Date * Return\_Date   **Step 12:** Save Details in rental file.  **Step 13:** Display confirmation message  **Step 14:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 4:** A black field detected  **Step 5:** Display message “This field must be entered”  **Step 6:** Position cursor in the offending field and return to step 7 |
| **Invalid dates format** |  | **Step 4:** An invalid format entered for dates  **Step 5:** Display message “This field must be in the right format”  **Step 6:** Position cursor in the offending field and return to step 7 |
| **Conclusions** | DVD is being rented | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Process Return

This function will process return for DVD item in the system.

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Process Return** | |
| **Use Case Id** | 4.4.2 | |
| **Priority** | High | |
| **Source** | Administrator | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | None | |
| **Description** | This function will process returns for DVD. | |
| **Preconditions** | None | |
| **Trigger** | None | |
| **Expected Scenario** | **Staff** | **System Response** |
|  | **Step 1:** Staff starts Return Process function  **Step 3:** Administrator Enters DVD title  **Step 5:** Select DVD title for return. | **Step 2:** Display UI  **Step 4:**  Retrieves summary DVD titles with DVD\_Id details  **Step 6:** System checks Return\_Date  **Step 7:** Save DVD Details to rental file  **Step 8:** Set Status back to active (“A”)  **Step 9:** Display confirmation message  **Step 10:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | DVD is returned in the system | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Perform Admin

This component includes function for management of DVD analysis. These functions includes storing data for analysis.

Admin

### Rental Analysis

This function will analyse rentals in the system.

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Rental Analysis** | |
| **Use Case Id** | 4.5.1 | |
| **Priority** | High | |
| **Source** | Administrator | |
| **Primary Business Actor** | Administrator | |
| **Other Participating Actors** | None | |
| **Description** | This function will analyse rentals | |
| **Preconditions** | None | |
| **Trigger** | None | |
| **Expected Scenario** | **Administrator** | **System Response** |
|  | **Step 1:** Administrator starts Rental Analysis function  **Step 3:** Administrator Selects Revenue for year or months | **Step 2:** Display UI  **Step 4:** System retrieves the revenue   * Sales Revenue   **Step 5:** Save rentals analysis in the Analysis File  **Step 6:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | Rental Analysis generated and is saved. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

# System Model

The following dataflow diagrams have been produced for the system:

External Entities:

Member

Data Stores:

* D1 DVD File
* D2 Member File
* D3 Rental File
* D4 Analysis File

Processes:

P1 Process DVD

P1.1 Set Rate

P1.2 Add DVD

P1.3 Update DVD

P1.4 Discontinue DVD

P2 Process Member

P2.1 Add Member

P2.2 Update Member

P2.3 Deregister Member

P3 Process Rental

P3.1 Execute Rentals

P3.2 Process Rentals

P3.3 Rental Analysis

## Level-0 DFD

Rent DVD

Member

DVD System

DVD

## Level-1 DFD

DVD Details

P1

Manage DVDs

D1 DVD File

Rental Details DVD Details

P3

Process Rentals

DVD Details

D3 Rental File

Member Details

Rental Details

D2 Member File

Receipt

D4 Analysis File

Member Details

P2

Manage Members

Payment

Member

Member Details

## Level-2 DFD (Process P1: Manage DVD)

P1.2

Add DVD

P1.1

Set Rate

Rate Details

DVD Details

D1 DVD File

Current DVD Details DVD Details

DVD Status

New DVD Details

P1.3

Update DVD

P1.4

Discontinue DVD

## Level-2 DFD (Process P2: Manage Members)

P2.1

Add Member

Member Details

Member Details

Member

Member Details

New Member Details

D2 Member File

P2.2

Update Member

Member Details

Current Member Details

P2.3

Deregister Member

Member Status

## Level-2 DFD (Process P3: Process Rentals)

P3.1

Execute Rental

D3 Rental File

P3.2

Process Return

P3.3

Rental Analysis

Member

D4 Analysis File

Member

Member Details

Rental Details Member Details

Rental Details Rental Details

Rental Details

Rental Details

Member Details

# Data Model (Class Diagram)

My data model has 6 classes these includes Members, DVD’s, Rentals, Payments, Category and membership.

## Class Diagram



## Relational Schema

Members (Member\_Id, Name, Contact, Email, Address, Eir\_Code, Town, County, Acc\_Status)

DVDs (DVD\_Id, Title, Description, Release\_Year, DVD\_Out, Member\_Id)

Rentals (Rental\_Id, Rental\_Date, Return\_Date, Due\_Date, Fine, DVD\_Id, Member\_Id)

Payments (Payment\_Id, Date, Cost, Member\_Id, Rental\_Id)

Category’s (Category\_Id, DVD\_Rate, DVD\_Id)

Membership (Acc\_status, Member\_Id)

## Database Schema

**Schema:** DVDSYS

**Relation:** Members

Attributes:

Member\_Id numeric (10) NOT NULL

Name char (30) NOT NULL

Contact numeric (15) NOT NULL

Email char (30) NOT NULL

Address char (30) NOT NULL

Eir\_Code char (8) NOT NULL

Town char (10) NOT NULL

County char (10) NOT NULL

Acc\_Status char (8) NOT NULL

**Primary Key:** Owner\_Id

**Foreign Key:** Acc\_Status REFERENCES Membership Relation

**Relation:** DVDs

Attributes:

DVD\_Id numeric (10) NOT NULL

Title char (30) NOT NULL

Description char (200) NOT NULL

Release\_Year Date NOT NULL

DVD\_Out numeric (10) NOT NULL

Member\_Id numeric (10) NOT NULL

**Primary Key:** DVD\_Id

**Foreign Key:** Member\_Id REFERENCES Members Relation

**Relation:** Rentals

Attributes:

Rental\_Id numeric (10) NOT NULL

Rental\_Date Date NOT NULL

Return\_Date Date NOT NULL

Due\_Date Date NOT NULL

Fine decimal (10, 0)

DVD\_Id numeric (10) NOT NULL

Member\_Id numeric (10) NOT NULL

**Primary Key:** Rental\_Id

**Foreign Key:** DVD\_Id REFERENCES DVDs Relation

**Foreign Key:** Member\_Id REFERENCES Members Relation

**Relation:** Payments

Attributes:

Payment\_Id numeric (10) NOT NULL

Date Date NOT NULL

Cost decimal (10, 0) NOT NULL

Member\_Id numeric (10) NOT NULL

Rental\_Id numeric (10) NOT NULL

**Primary Key:** Payment\_Id

**Foreign Key:** Member\_Id REFERENCES Members Relation

**Foreign Key:** Rental\_Id REFERENCES Rentals Relation

**Relation:** Category’s

Attributes:

Category\_Id numeric (10) NOT NULL

DVD\_Rate decimal (10, 0) NOT NULL

DVD\_Id numeric (10) NOT NULL

**Primary Key:** Category\_Id

**Foreign Key:** DVD\_Id REFERENCES DVDs Relation

**Relation:** Membership

Attributes:

Acc\_Status char (5) NOT NULL

Member\_Id numeric (10) NOT NULL

**Primary Key:** Acc\_Status

**Foreign Key:** Member\_Id REFERENCES Members Relation

# Conclusion

The aim for my project was to make a DVD Rental System that can be used by the admin to manage DVD’s, members, processes rentals and analyses rentals. It contains specific user and system requirements which describes in detail the functions that the system must perform. Models were used to show the interaction between each process and databases and how the databases relates to one another.

I made a simple prototype for my system using visual studio to show the functions and what actions were taking place behind the user interface. An important factor in building the prototype was making it user friendly, easy to use and to understand. The prototype has simple validations where if the user doesn’t input anything in the field the system will ask the user to fill in the blank fields.

# Appendices

## Appendix A – Title

## Appendix B – Title

Might include:

* **Lookup / Reference tables**
* **Sample reports / Listings**