**PROJECT SCENARIO (Business Case) for Project #2**

Pre-Read: Netflix history and business model.

* <http://www.cass.city.ac.uk/__data/assets/pdf_file/0017/220517/Netflix.pdf>
* <http://en.wikipedia.org/wiki/Netflix>
* <http://www.Netflix.com>

Netflix. Netflix is the old Blockbuster brick and mortar business model of the 21st century. They offer consumers on-demand streaming video with unlimited movie watching at a fraction of the old Blockbuster rental fee model in the comfort of their homes on any electronic device.

Due to the overwhelming success of Netflix’s on-demand video streaming service for consumers, the Meramec bookstore would like to offer a similar but scale down service to the college students. Their service would offer on-line DVD rentals and viewing from on-line devices connected to the college local area network.

Besides the on-line rental of movies to the students, the new system will need to track new and existing memberships, media rentals, vendor movie purchases, tracking inventory control, rentals account receivables and various reports.

The bookstore would like to have the ability to market certain demographics with movie specials during the holiday and non-holiday season. Besides promoting these movie specials, management thought it would be a great idea if the bookstore could make their movies available on-line to registered members via the college’s Intranet web system. This would give the students the ability to browse their inventory on-line and see the various movie specials the bookstore has to offer in the coming months.

Your team has been hired as an outside software consultant to convert Meramec's manual media rental system to a computerized multi-user client/server on-line (kiosk) system.

The team has been approved to start gathering formal requirements by created UML use cases of the proposed systems and providing a prototype of the new system in 7 weeks.

On the next several pages, you will find an overview of the project specification business, design requirements and a preliminary logical database model that was put together by the team’s business analyst.

As the software design consultant, your job is to validate the business requirements, design the system flow using UML use cases, do screen mock-ups, create the physical database/tables/stored procedures/etc., create the required business objects (classes) and develop the application UI using Microsoft Visual Studio C# and OO design concepts and building blocks.

**Class Project #2 Specifications**

The purpose of this project is to test/validate your understanding of client/server 3-tier development, object oriented class design in C#, database programming with Microsoft SQL Server, using SQL and creating a Windows C# application.

**Client/Server Class Project: 240 points (\*40 pts from project #6)**

**Due Date: End of semester (Final Exam Day)**

**Points breakout: Documentation deliverables………………...0pts.**

**Business Objects – 10 @ 10 pts…………100 pts**

**Login Screen with full functionality …......10 pts.**

**Member Screen w/full functionality ..…....10 pts.**

**Movie Screen w/full functionality………….10 pts.**

**Media Rental Screen w/full functionality…35 pts.**

**Mystery Screen w/full functionality……….10 pts.**

**Reports (min 2 rpts.)…………………………10 pts.**

**Class Diagram of Business objects……......5 pts.**

**Use Case Diagram…………………….……...10 pts.**

**\*Add Project #6 points (Challenge).……….. 40 pts.  
 \*(Add functionality to show the video trailer)**

The individual or group will be responsible for the following:

**(DBA and Software Developer duties)**

1. Creating the database and populating the tables in the Microsoft SQL Server machine using SQL Server Studio Management (SSMS) Tool based
2. Designing/creating all business classes (components) including all CRUD business components using OO design techniques/concepts
3. Creating the Windows user interface including all data entry screens and menu items based on the UML Use Case and business requirements
4. Creating reports using Microsoft Reports based on the business requirements
5. Creating and validating all required DML statements using ANSI 92 SQL
6. Creating ADO.Net database code with using custom business classes to manipulate the data in the tables using ADO.Net objects and ANSI 89/92 SQL commands and stored procedures.
7. Create at least one UML Use Case Diagram
8. Create the ERD model
9. Creating at least one stored procedure and trigger to enforce business rules or logic on the server.
10. Unit testing and debugging your application before the final system is presented to the end user group (Instructor) for evaluation.

**PROJECT GUIDELINES**

1. You have totally freedom of designing the look and fool of the application GUI including the data forms and reports. However, the UI should be designed with the best user design as possible following the use cases and application specs.
2. You must use OO design techniques in designing your business classes according to the design specs of what properties and methods that each class should have.
3. All business classes should have a default constructor
4. All database tables must conform to the 1st -3rd form of database normalization
5. All database CRUD tasks will be performed using the ADO.Net main three classes.
6. No Table adapter wizards should be used in this project.

**PROJECT DELIVERABLES**

Your group project will consist of the following deliverables. **A sample deliverable mock-up binder will be given to each student as a guide.**

* Business requirements document
* Functional Design Specification Document
* UML Use-Cases
* Logical data model of the new system
* Entity Relationship Design (ERD)
* System test plans
* Printouts of the physical database model including indexes, triggers, business rules, constraints, stored procedures, triggers and user permissions
* Screen printouts of data entry screens and reports

***\*\* Note: All deliverables must be computer generated. No hand-written deliverables will be accepted. \*\****

**Project Grade**

Your project grade will be based upon meeting the business requirements/specifications, business objects, screen design, database design, system functionality according to the specs, reporting and the professional quality of the deliverables.

**Functional and Technical Design Guidelines** **MOVIE RENTAL LOGICAL Database SYSTEM DESIGN**

Table Names and Fields

**MOVIE**

Primary Key Field: movie\_number

Index: Create clustered index on movie\_number

The movie table contains information about the type of movies the store has for renting.

**FIELDNAME DESCRIPTION DATABASE DEFINITION**

|  |  |  |
| --- | --- | --- |
| movie\_number | movie unique identifier | int not null |
| movie\_title | movie title | varchar(30) not null |
| Description | movie descriptions | varchar(255) not null |
| movie\_year\_made | year the movie was made | smallint not null |
| genre\_id | Type of movie genre | smallint not null |
| movie\_rating | rating of the movie | char(5) not null |
| media\_type | What type of medium is the media. DVD, Blue-Ray | varchar(10) not null |
| movie\_retail\_cost | retail cost of the movie | float not null |
| tape\_rental\_cost | rental cost of the movie to the renter | float not null |
| copies\_on\_hand | number of copies the stores has | smallint null |
| image | Movie image filename | varchar(30) nul |
| trailer | Filename of the video trailer | varchar(30 null |

**MEMBER**

Primary Key Field: number

Index: Create clustered index on number

The member table contains membership information.

**FIELDNAME DESCRIPTION DATABASE DEFINITION**

|  |  |  |
| --- | --- | --- |
| number | customer or member unique number | int not null |
| joindate | date in which the member join club | datetime not null |
| firstname | first name of the member | char(15) not null |
| lastname | last name of the member | char(25) not null |
| address | address of the member | char(30) not null |
| city | city where the member resides | char(20) not null |
| state | state where the member resides | char(2) not null |
| zipcode | zipcode of the member | char(5) not null |
| phone | daytime phone number of the member | char(10) not null |
| member\_status | active or inactive member status | char(1) not null |
| password | member login password | Varchar(20) not null |

**RENTAL**

Primary Key Field: movie\_number and member\_number

Index: Create clustered index on movie\_number, member\_number

The taperental table contains information pertaining to members who have tapes that are check or rented out.

**FIELDNAME DESCRIPTION DATABASE DEFINITION**

|  |  |  |
| --- | --- | --- |
| movie\_number | movie unique identifier | int not null |
| member\_number | member unique number | int not null |
| media\_checkout\_date | date the media (dvd) was checked out | datetime not null |
| media\_return\_date | date the media (dvd) was returned to store | datetime null |
| media\_exp\_return\_date | expected return date. i.e. 3 days after checkout date –Derived or calculated | Datetime not null |
| media\_rental\_cost | rental tape amount | Small money |

**GENRE**

Primary Key Field: id

Index: Create clustered index on id

The genre table holds the various movie genres. i.e. Comedy, Sci-Fic, Drama, etc.

**FIELDNAME DESCRIPTION DATABASE DEFINITION**

|  |  |  |
| --- | --- | --- |
| id | Unique genre id | int not null |
| name | Movie genre description | Varchar(30) not null |

## **Mystery Table Defintion ?????**

**One mystery table needs adding. You will have to read the business case scenario to uncover the mystery table. Also, a change will be needed to one of the above tables to accommodate the mystery table’s primary key relationship.**

Primary Key Field:

Index: Create clustered index on

Purpose of table (TBD)

**FIELDNAME DESCRIPTION DATABASE DEFINITION**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*\* Note: The system should utilize at least one stored procedure and trigger to maintain database referential integrity. It is up to each group to come up with a scenario for each. \*\***

**Meramec Media Rental Reports**

1. Report that prints a membership listing

2. Report that prints the dvd inventory and grand total for the retail cost of the dvds.

3. Account receivable report broken down and subtotal by month and a year to date grand total.

4. Report that prints membership badges that consist of the membership number, first name and last name of the customer.

5. Report that prints what dvds are checked out and by whom on any given date.

6. Report that prints the overdue tapes on a daily basis along with the person's full name, address and phone number.

1. Print a report from another table

**C# Business Objects**

Create the following business objects to represent each database table domain. You will have two C# classes for each table that you will used in your application design.

1. One class that represents a single table object value
2. One static class that facilitates the database access (CRUD) calls. This class should be using proper Try..Catch..Finally Exception Handling and Using statements for all resources used.

* Your individual class should have the following:

***Properties (Add properties to represent each field in the table)***

***Methods***

None

* ***CRUD Static class <TableName>DB represents each table in the database that will be used for your C****reate****R****ead****U****pdate****D****elete**Operations*

Your database access class should have the following:

**Add the following namespaces in this class**

Using System.Data;

Using System.Data.Sqlclient;

***Properties***

None

***Methods Names Purpose***

|  |  |
| --- | --- |
| Public static List<ObjectType> Get<*TableNames>()* | Returns a list of generic type objects from the table |
| Public static Get<TableName) (accepts a parameter to return a specific record) | Returns a single record from the table whose parameter matches a table field condition |
| Public static bool Add<TableName) (accepts an object of that type as a parameter)  True – Record was added successfully  False – Record was not added successfully | Adds a record to the table with a Boolean returned status of True or False. |
| Public static bool Update<TableName) (accepts an object of that type as a parameter, possible other parameters that you need to use in the update)  True – Record was added successfully  False – Record was not added successfully | Updates a record in the table with a Boolean returned status of True or False |
| Public static bool Delete<TableName) (accepts an object of that type as a parameter  True – Record was added successfully  False – Record was not added successfully | Deletes a record from the database with a Boolean returned status of True or False |

\*\*\*Important \*\*\*\*

Use the sample C# Business Objects (classes) that are provided by the instructor to model your own business classes for this project.

Refer to the ADO.Net C# Demo Solution

**Optional - Other C# Business Objects that you can create if time permits.   
Not Required**

* **Login Class** – This class will be used to facilitate the checking of the user logging into the application to see if the user is valid or not.

**Add the following namespaces in this class**

Using System.Data;

Using System.Data.Sqlclient;

Your individual class should have the following:

***Properties***

Credentials - string

Password - string

***Methods***

Public Boolean IsValid() – This method should determine if the user logging into the system is valid or not. The method should return a Boolean (True or False). **True represents** the credentials are valid and you should enable the Menu Items.

**False represents** the credentials are invalid and should not enabled the Menu Items.

* **SQL Class** – This class will be used to store all your SQL for each table that is used in the system. **You will define this class as a static class and all of its methods**. The Table CRUD classes will use this internally to get the SQL to use in its own class to interface with the database.

|  |  |
| --- | --- |
| Method Name | Purpose |
| TBD |  |

* **AccessDataSQLServer** –This class will be used to return a database connection. You can make this a super class to be inherited by all the sub-classes. This would replace the private function in each of the CRUD static classes.

# Need and Define Your Use Cases



| Use Case 1.0: XXXX |
| --- |
| **ID: UC\_X.X.X** |
| **Description:** xxxx |
| **Actors: xxx** |
| **Preconditions:**   * xxx |
| **Main Event Flow:**   * xx * xx |
| **Alternate Event Flow:** |
| **Postconditions:**   * None |
| **Comments:** |

Problem Domain Object Class Model Diagram from Visual Studio

Database Entity Relationship Diagram (ERD)  
(Add your Database Entities and how are they related. 1:1, 1:Many, Many:Many)

**Helpful resources and other links**

**Movie Industry links**

Movie Genres - <http://www.imdb.com/genre/>

Film ratings - <http://www.mpaa.org/film-ratings/>

**Database Related links**

Database Design – TBD

Entity Relationship Diagram – TBD

**Object Oriented Design links**

UML (Use Cases) -

**C# links**

<http://csharp.net-tutorials.com/>

Understanding the ‘using’ statement in C#

<http://www.codeproject.com/Articles/6564/Understanding-the-using-statement-in-C>

Creating Objects  
<http://www.homeandlearn.co.uk/csharp/csharp_s10p1.html>  
<http://msdn.microsoft.com/en-us/library/ms173109.aspx>  
<http://csharp.net-tutorials.com/classes/method-overloading/>

Creating Collections  
<http://channel9.msdn.com/Series/C-Fundamentals-for-Absolute-Beginners/22>

How to bind data to a datagrid  
<http://www.c-sharpcorner.com/UploadFile/deveshomar/ways-to-bind-datagridview-in-window-forms-C-Sharp/>

**SQL Server links**

SQL Server Management Studio Tutorial - <https://technet.microsoft.com/en-us/library/bb934498(v=sql.105).aspx>

SQL - <http://www.w3schools.com/sql/>

ADO.NET Code Samples and ADO.Net connection strings

<https://msdn.microsoft.com/en-us/library/dw70f090(v=vs.110).aspx>

<https://msdn.microsoft.com/en-us/library/ms254500(v=vs.110).aspx>

SQL Server Connection String for Visual Studio (Select the .Net Library Option)

<https://www.connectionstrings.com/sql-server/>