Abstract

This project aims to discuss MVC patterns alongside Using Data Mining which represent the Entity Framework and LINQ to manipulate and retrieve data from a Database to convert it to useful information.

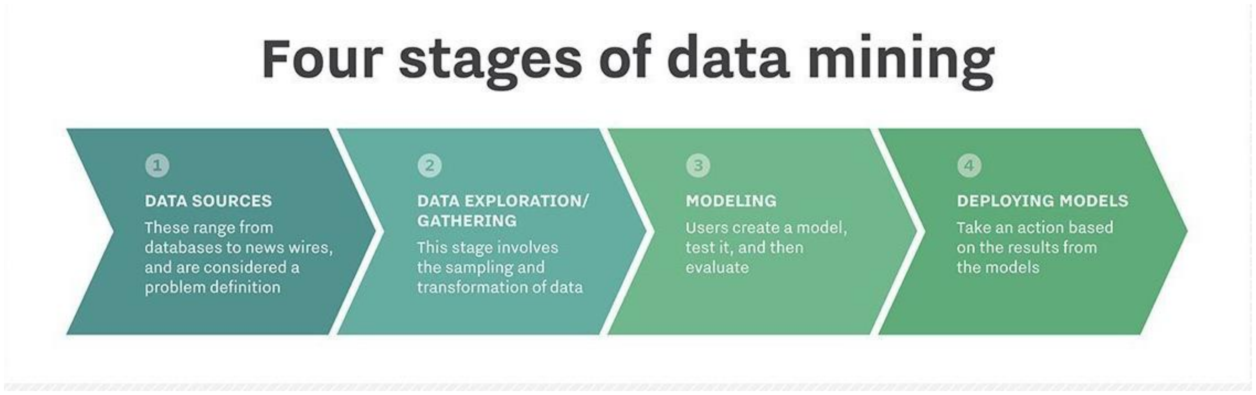

Four stages of data mining
1
DATA SOURCES
These range from databases to news wires, and are considered a problem definition
2
DATA EXPLORATION/ GATHERING
This stage involves the sampling and transformation of data
3
MODELING
Users create a model, test it, and then evaluate
4
DEPLOYING MODELS
Take an action based on the results from the models

Table of Contents

1 The meta behind the project

In this project (CRM Music Store) the admin can add, delete and edit albums to show them to the customer and then he can buy, view by his genre interest the albums he needed. To apply this WEB-Based application MVC pattern will be used (Model-View-Controller) and SQL server to save this data and use them at any time. Entity framework will be also used to control this data and create some useful information.

This will create a simple app using data mining.

1.1 What is Data Mining?

It is the way of having solid data and try to get some info out of it. And it goes through some steps:

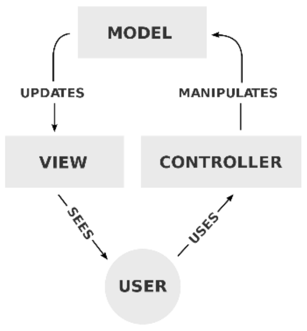
- Data Cleaning: using the albums class as the way to retrieve the important data.
- Data integration: Combine the assets classes to create multiple datasets.
- Data selection: In this case we will use the **LINQ** to select which data we will show.
- Data transformation: where data relevant to the analysis task are retrieved from the database.
- Knowledge presentation: this when the web app shines to show the wanted data.

1.2 Rational database

A relational database is a collection of tables, each of which is assigned a unique name. Each table consists of a set of attributes (columns or fields) and usually stores a large set of tuples (records or rows). Each tuple in a relational table represents an object identified by a unique key and described by a set of attribute values. A semantic data model, such as an entity-relationship (ER) data model, is often constructed for relational databases. An ER data model represents the database as a set of entities and their relationships.

1.3 A Glance

Creating models: Album, Artist, Genre, Entities -to save the database-. To create the backbone of the project. Then creating controllers: Store, Store manager and Shopping to handle the data passing from the models-database- to the view. And creating Views to show the result info for each controller while each controller handles methods and each one will have its view template to view its corresponding function in the project of:



- Editing and creating Albums and save the to the data base.
- View the tables from the manager and the guest side of view.
- Selecting albums with genre interest.
- Buying albums.

2 Building the project

In the next few page there will be the discussion of creating the project and the data base.

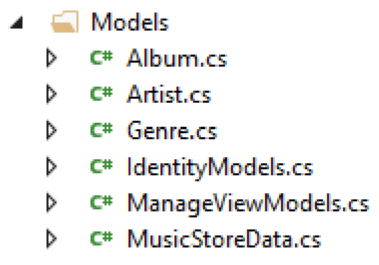
2.1 Styling the project

Using CSS and Bootstrap to create a look to the project. They can be found at http://www.free-css.com/template-categories/music and editing the CSS to go along with the project.

https://bootswatch.com/ offers great bootstraps to design the tables, buttons, etc…

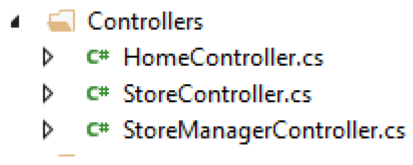
2.2 Creating the Models

As classes are the form of the projects we will create these classes alongside Cart to get the customer info and albums wanted

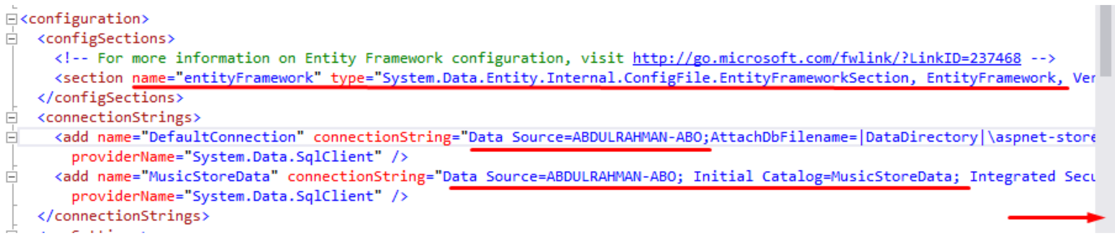


2.3 Creating the controllers

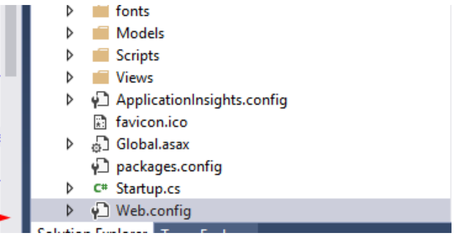
Controllers will manage to handle the requests from the users and respond we will use Entity framework for StoreManager. and LINQ to just show the data for the customer in Store. Alongside with cartcontroller to control the wanted albums and so on. And accountcontroller to secure the data.



2.4 Creating the DataBase

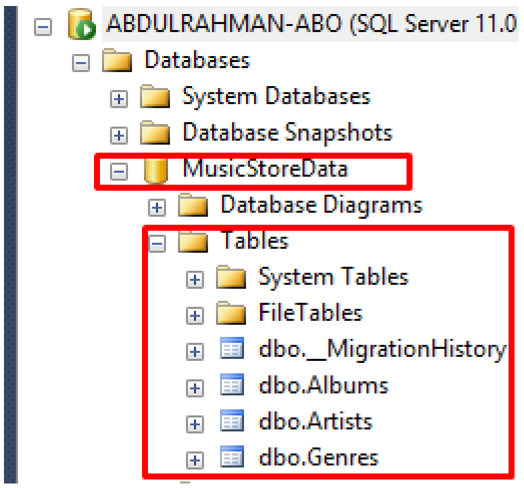


```
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Ver
  </configSections>
  <connectionStrings>
    <add name="DefaultConnection" connectionString="Data Source=ABDULRAHMAN-ABO;AttachDbFilename=|DataDirectory|\aspnet-store
      providerName="System.Data.SqlClient" />
    <add name="MusicStoreData" connectionString="Data Source=ABDULRAHMAN-ABO; Initial Catalog=MusicStoreData; Integrated Secu
      providerName="System.Data.SqlClient" />
  </connectionStrings>
```



This will create the database in SQL as Entity framework creates all the databases in a local which are hard to manipulate.

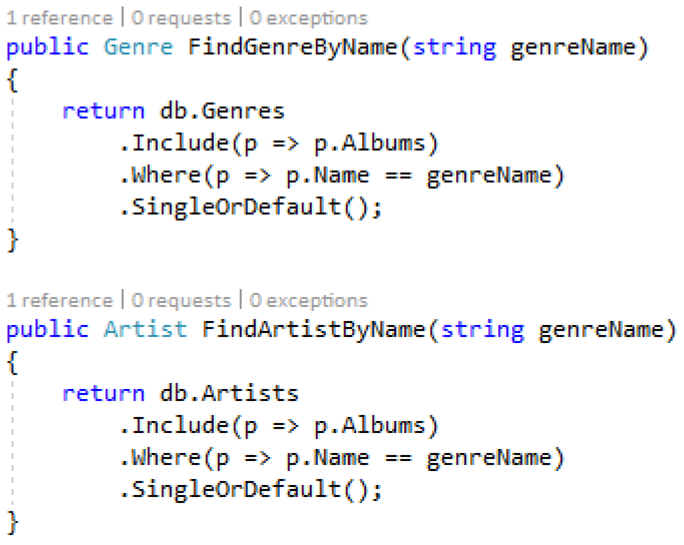
Another way is to Create a seeder to seed the database with some Info but with SQL is easier and more sophisticated so it will be used in the project.



2.5 Editing the layout

In views/shared/-layout the overall layout of the view project is handled, we will use it to view permanent tabs and Partial Views “Genre View” to make it easier to navigate through the app.

2.6 LINQ

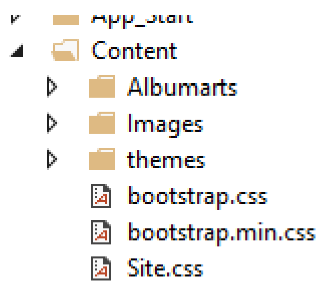


```
1 reference | 0 requests | 0 exceptions
public Genre FindGenreByName(string genreName)
{
    return db.Genres
        .Include(p => p.Albums)
        .Where(p => p.Name == genreName)
        .SingleOrDefault();
}

1 reference | 0 requests | 0 exceptions
public Artist FindArtistByName(string genreName)
{
    return db.Artists
        .Include(p => p.Albums)
        .Where(p => p.Name == genreName)
        .SingleOrDefault();
}
```

2.7 Updating the album arts

In Content/albumarts we will upload the pics of the albums and reference them in the database.



2.8 Securing using accounting

We don’t need the customer to edit or mess with the data so we will create account and authorization to enter as only the admin can edit his data.

Using Authorized attribute on the Storemanager.