Computer Programming 2 – BuyDVDS Program Report

By Toby Gates

ID - 000540476

Table of Contents

Statement on the level of functionality achieved

The level of functionality I have achieved for my program is Level 5 functionality.

I have completed USE CASE 1 in my program with the use of JTextFields allowing the user to input and JLabels indicating where the user should input each value. I also made use of Array Lists for USE CASE 1 which allowed me to add to, edit parts and delete from without having a database or any other kind of external storage method implemented.

I have completed USE CASE 2 by connecting the program up to an external storage database which the user can add records to, delete records from and edit records much like the Array Lists I used in USE CASE 1. I have also completed USE CASE 3 as I am able to search for DVDs by their description/notes field, I am also able list all DVD entries in order of alphabetical name and for the final point of USE CASE 3 I have restricted both of these so that they only display DVDs with the types “Film” or “Music” any other DVD types are still in the database but they are just not shown when sorted and searched for.

I have also completed USE CASE 4 as I have managed to get the program to scroll through the DVD records with the use of Next and Previous buttons. I have also managed to get the program to display the record image for the DVD of the user’s choice as well as being able to display a thumbnail of the record image if the user wishes it. I have also completed USE CASE 5 as my program is able to select a DVD from the list of records and then create a webpage which displays all of the details plus the image assigned to it, if the URL link for the image is correct. However, I have not managed to complete the sophisticated functionality for USE CASE 5.

List of Bugs/Weaknesses in the Program

I feel that one of the weaknesses in my program is concerned with both the Next and Previous Record buttons. Both buttons do what they are supposed to do and allow the user to scroll to the next or previous record; however they both require the user to input the number of the current record they are on each time they are clicked on.

One other weakness involved the validation of JOptionPane message boxes, the validation messages worked and appeared normally when an input that wasn’t supposed to be entered was entered, however when I clicked cancel to close it the validation message would also appear on screen there too even if no value had been entered. I later managed to fix this problem with the JOptionPane validation by making use of a pattern and a matcher which I found out about from the link below: http://www.roseindia.net/tutorial/java/swing/joptionpaneValidation.html

Another weakness in my program is the lack of a message indicating to the user that there have been no results found after a search of records which produced no results that match the description entered by the user.

Special Strengths of the Application that should be taken into account

One of the strengths of the application that should be taken into account when marking is my Instructions button on the main menu of the program, when clicked it allows the user to view the readme.txt file as most users would probably tend to overlook or not bother reading it when using the program.

The validation added to the JOptionPane should be taken into account as it doesn't show the validation when the other buttons (such as the cancel button) are clicked (which is what happens when it is written out in a catch statement). My validation makes use of a pattern and a matcher to check the input and see if it's a valid input or to see if it's an input that is invalid and requires an message to show up on screen so that the user will realise their input was invalid. This then allowed me to use the try and catch statements for other information messages for the user.

Another strength of my program is the ability to edit each separate part of an already saved record in the database instead of always having to edit them as one whole record. My program provides multiple buttons that can be clicked on in order to edit a single section of the record such as; the DVD Name, the DVD Type, the DVD Description and so on, which is easier if one part of the record is missing as this allows you to fill in that one part without changing anything else.

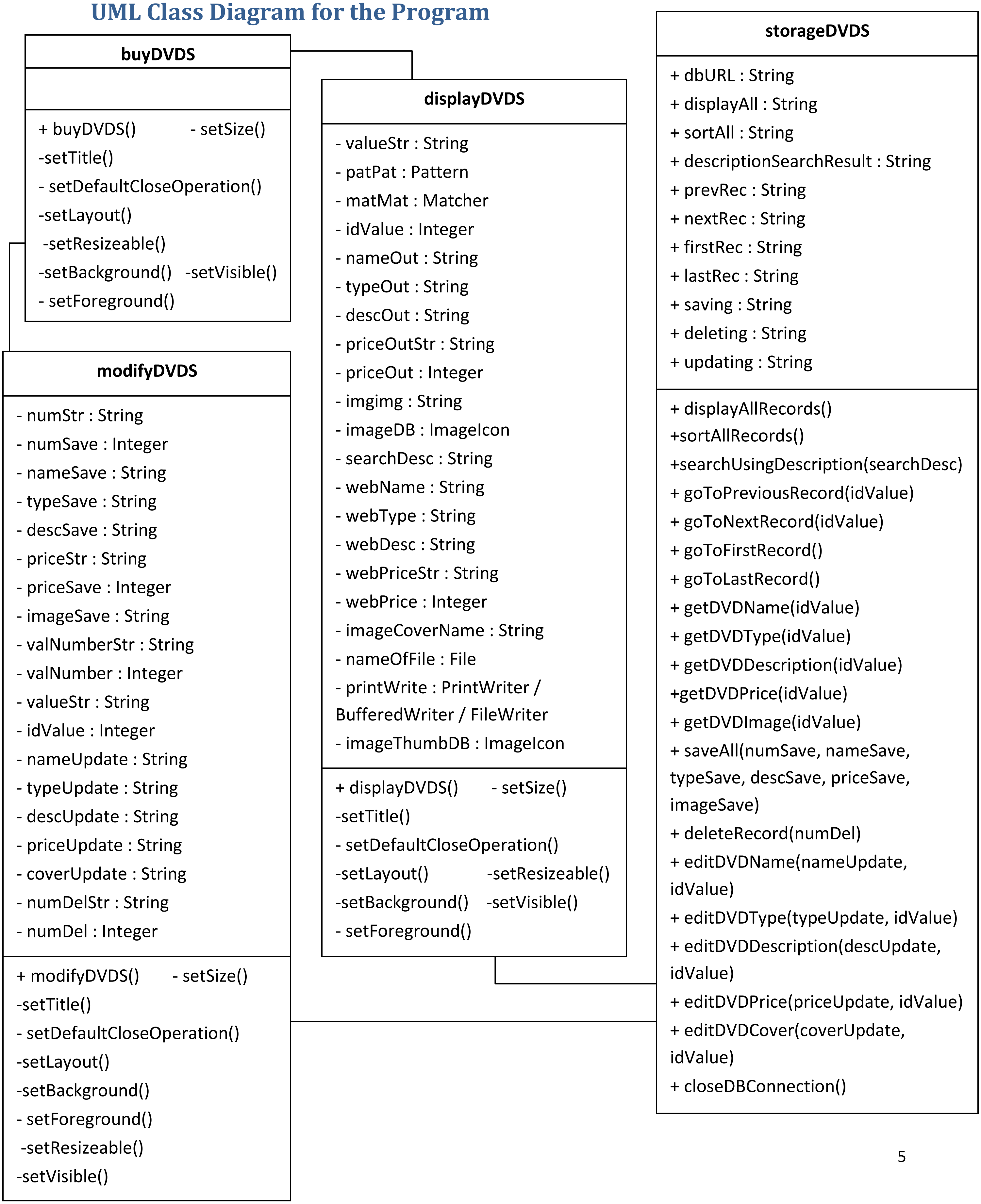
Brief Description of any Particular Design Features in the Program

The reason for my design feature of splitting the program up into 4 classes was to split up the different functions of the program; this allowed me to create a Main Menu and make it the source of navigation between all of the classes of the program. The modifyDVDS class has a JFrame which allows the adding of records as well as the editing of records and certain parts of records as well as the deleting of records. The displayDVDS class has also has a JFrame which is used to display the records of the program, whether it is to sort them by DVD Name or display some search results and so on.

I felt that keeping the whole input and editing aspect of the program away from the record display aspect of the program made it simpler for the user to find what they wanted and to also prevent any human error such as deleting a record by accident instead of initiating a search for records, for example. I also created a 4th class without a JFrame called storageDVDS which separately handles all the database operations such as adding, editing, deleting values and also getting values from the database to display on the program which also includes getting image files from the database too. I found this helpful as I would know which class to look in if I needed to see what the method did.

My UML Class Diagram is found on the next page.

UML Class Diagram for the Program



buyDVDS

+ buyDVDS() - setSize()
-setTitle()
- setDefaultCloseOperation()
-setLayout()
-setResizeable()
-setBackground() -setVisible()
- setForeground()

modifyDVDS

- numStr : String
- numSave : Integer
- nameSave : String
- typeSave : String
- descSave : String
- priceStr : String
- priceSave : Integer
- imageSave : String
- valNumberStr : String
- valNumber : Integer
- valueStr : String
- idValue : Integer
- nameUpdate : String
- typeUpdate : String
- descUpdate : String
- priceUpdate : String
- coverUpdate : String
- numDelStr : String
- numDel : Integer

+ modifyDVDS() - setSize()
-setTitle()
- setDefaultCloseOperation()
-setLayout()
-setBackground()
- setForeground()
-setResizeable()
-setVisible()

displayDVDS

- valueStr : String
- patPat : Pattern
- matMat : Matcher
- idValue : Integer
- nameOut : String
- typeOut : String
- descOut : String
- priceOutStr : String
- priceOut : Integer
- imgimg : String
- imageDB : ImageIcon
- searchDesc : String
- webName : String
- webType : String
- webDesc : String
- webPriceStr : String
- webPrice : Integer
- imageCoverName : String
- nameOfFile : File
- printWrite : PrintWriter / BufferedWriter / FileWriter
- imageThumbDB : ImageIcon

+ displayDVDS() - setSize()
-setTitle()
- setDefaultCloseOperation()
-setLayout() -setResizeable()
-setBackground() -setVisible()
- setForeground()

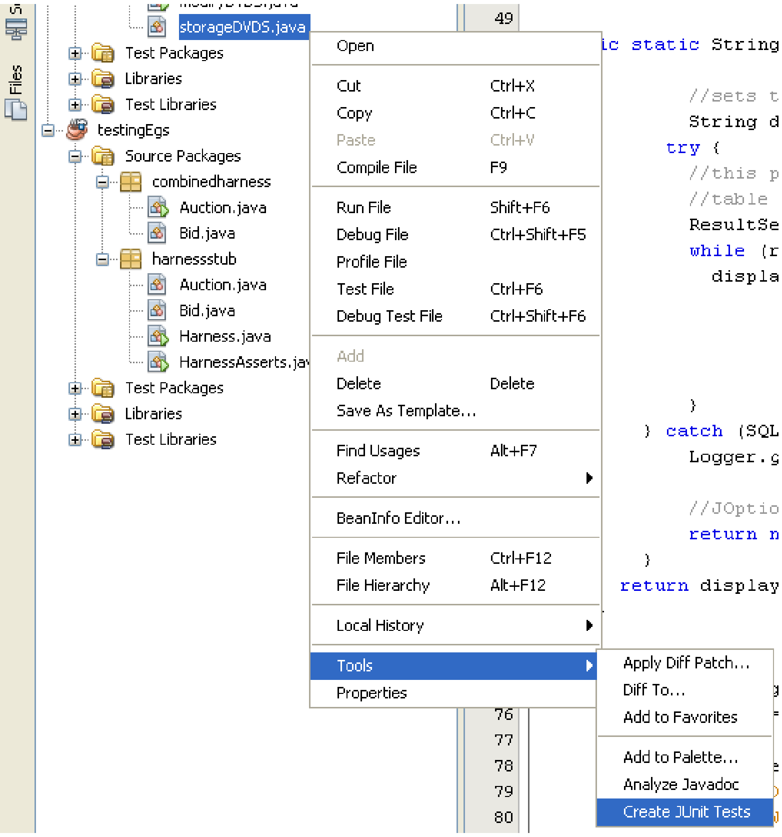
storageDVDS

+ dbURL : String
+ displayAll : String
+ sortAll : String
+ descriptionSearchResult : String
+ prevRec : String
+ nextRec : String
+ firstRec : String
+ lastRec : String
+ saving : String
+ deleting : String
+ updating : String

+ displayAllRecords()
+sortAllRecords()
+searchUsingDescription(searchDesc)
+ goToPreviousRecord(idValue)
+ goToNextRecord(idValue)
+ goToFirstRecord()
+ goToLastRecord()
+ getDVDName(idValue)
+ getDVDType(idValue)
+ getDVDDescription(idValue)
+getDVDPrice(idValue)
+ getDVDImage(idValue)
+ saveAll(numSave, nameSave, typeSave, descSave, priceSave, imageSave)
+ deleteRecord(numDel)
+ editDVDName(nameUpdate, idValue)
+ editDVDType(typeUpdate, idValue)
+ editDVDDescription(descUpdate, idValue)
+ editDVDPrice(priceUpdate, idValue)
+ editDVDCover(coverUpdate, idValue)
+ closeDBConnection()

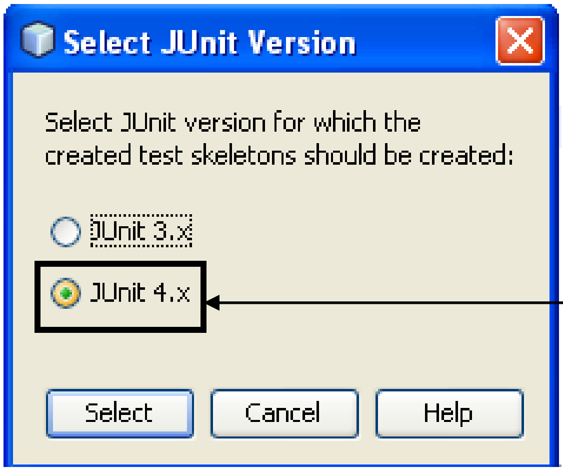
How Automated Unit Testing was carried out on my program

In order to test my program I made use of the JUnit used for Unit Testing that was built into the NetBeans IDE. In order to use this I first had to right click one my classes (I chose one that had multiple methods in) then I selected Tools, and then Create JUnit Tests which would take me to a wizard that would help me create a test template for the selected class.



```
ic static String
        //sets t
        String d
      try {
        //this p
        //table
        ResultSe
        while (r
          displa

        }
      } catch (SQL
        Logger.g

        //JOptio
        return n
      }
   return display
```

The image to the left shows the method of creating JUnit Tests in a Test Skeleton class for Unit Testing.



Select JUnit version for which the created test skeletons should be created:

○ JUnit 3.x

● JUnit 4.x

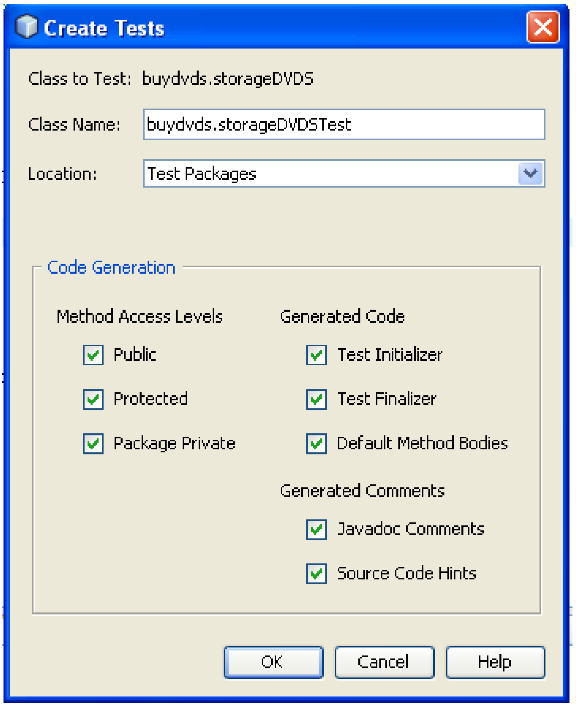
After I had clicked on Create JUnit Tests, the window to the left came up on screen, so I selected to use JUnit 4.x to do the testing of my program with.

Once I had chosen the version of JUnit I would use to create test skeletons in order for me to test with, I had another window pop up on screen which allows me to create the test skeletons, as I was advised to use the default settings I just simply clicked the OK button with nothing changed.

The image to the right shows the Create Tests window that concerns the Test Skeletons.

It shows which code it will generate in the test skeleton with the various check boxes available to tick or untick. It also allows you to name the class and specify the location of where this test skeleton would be saved.

The image to the right however shows the different settings I used to create my test skeleton for Unit Testing my program.



Create Tests

Class to Test: buydvds.storageDVDS

Class Name: buydvds.storageDVDSTest

Location: Test Packages

Code Generation

Method Access Levels

☑ Public

☑ Protected

☑ Package Private

Generated Code

☑ Test Initializer

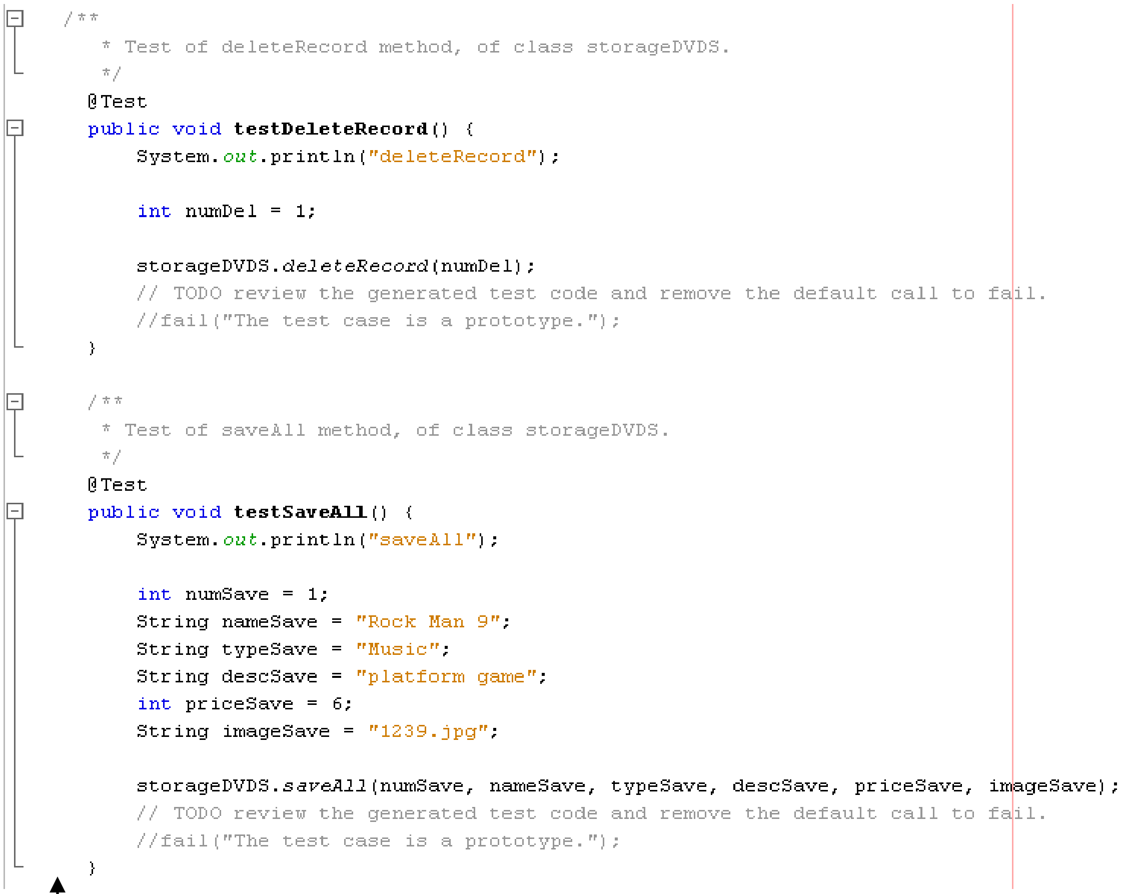
☑ Test Finalizer

☑ Default Method Bodies

Generated Comments

☑ Javadoc Comments

☑ Source Code Hints

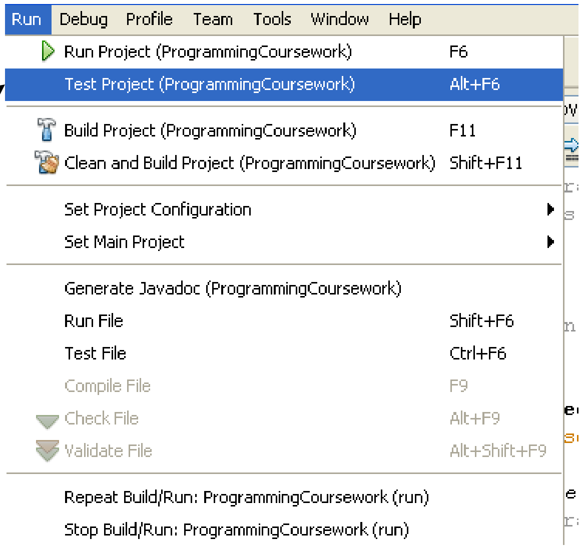


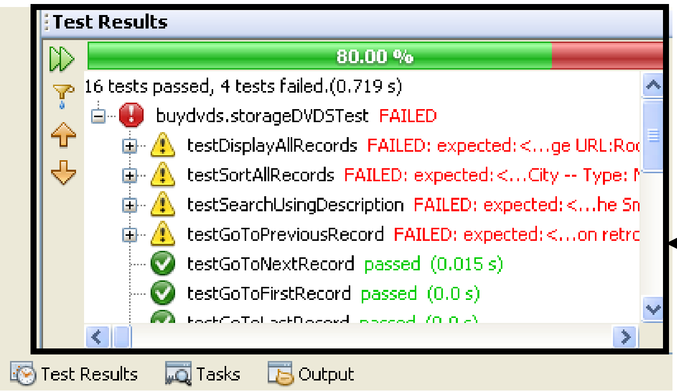
```
  /**
     * Test of deleteRecord method, of class storageDVDS.
     */
    @Test
    public void testDeleteRecord() {
        System.out.println("deleteRecord");

        int numDel = 1;

        storageDVDS.deleteRecord(numDel);
        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }

    /**
     * Test of saveAll method, of class storageDVDS.
     */
    @Test
    public void testSaveAll() {
        System.out.println("saveAll");

        int numSave = 1;
        String nameSave = "Rock Man 9";
        String typeSave = "Music";
        String descSave = "platform game";
        int priceSave = 6;
        String imageSave = "1239.jpg";

        storageDVDS.saveAll(numSave, nameSave, typeSave, descSave, priceSave, imageSave);
        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }
```

The image above shows two out of twenty tests that I wrote out in order to test my program with Unit Testing. As you can see these two are just taking values and calling methods to see if they work, but there are other methods that were tested which compare the actual result with an expected result.

The image to the left shows how I tested the tests I wrote out in the Test Skeleton Class.

I first clicked Run and then clicked on the Test Project option underneath Run Project so see which ones passed and failed.

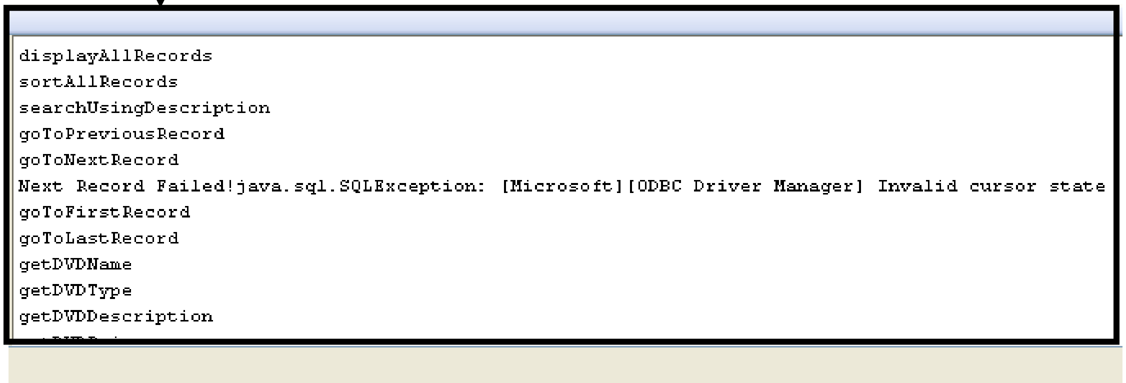

Run Debug Profile Team Tools Window Help
Run Project (ProgrammingCoursework) F6
Test Project (ProgrammingCoursework) Alt+F6
Build Project (ProgrammingCoursework) F11
Clean and Build Project (ProgrammingCoursework) Shift+F11
Set Project Configuration
Set Main Project
Generate Javadoc (ProgrammingCoursework)
Run File Shift+F6
Test File Ctrl+F6
Compile File F9
Check File Alt+F9
Validate File Alt+Shift+F9
Repeat Build/Run: ProgrammingCoursework (run)
Stop Build/Run: ProgrammingCoursework (run)



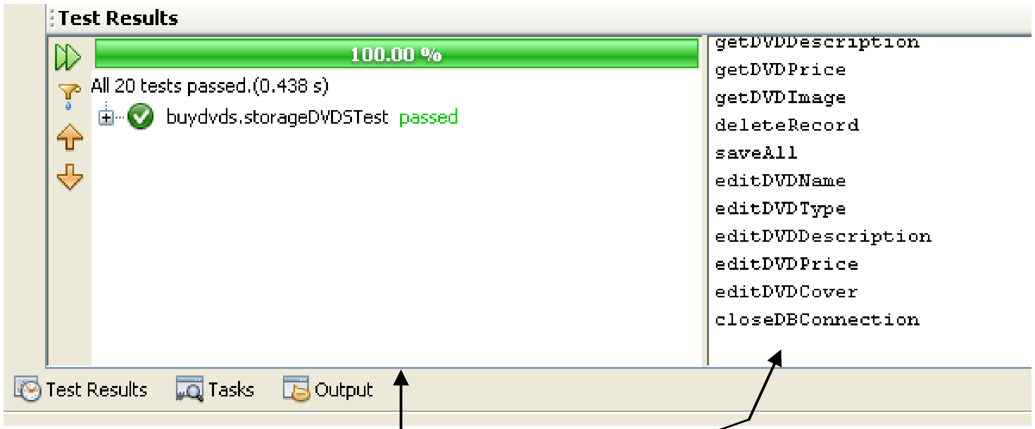
```
16 tests passed, 4 tests failed.(0.719 s)
buydvds.storageDVDSTest FAILED
    testDisplayAllRecords FAILED: expected:<...ge URL:Roc
    testSortAllRecords FAILED: expected:<...City -- Type: N
    testSearchUsingDescription FAILED: expected:<...he Sn
    testGoToPreviousRecord FAILED: expected:<...on retrc
    testGoToNextRecord passed (0.015 s)
    testGoToFirstRecord passed (0.0 s)
    testGoToLastRecord passed (0.0 s)
```

My test results for the first couple of tests ranged from about 70 to 80%, with usually these 6 texts failing due to whatever reason.

I then decided to spend some time and look over them to see if I could fix these failed tests.



```
displayAllRecords
sortAllRecords
searchUsingDescription
goToPreviousRecord
goToNextRecord
Next Record Failed!java.sql.SQLException: [Microsoft][ODBC Driver Manager] Invalid cursor state
goToFirstRecord
goToLastRecord
getDVDName
getDVDType
getDVDDescription
```



```
All 20 tests passed.(0.438 s)
buydvds.storageDVDSTest passed
```

```
getDVDDescription
getDVDPrice
getDVDImage
deleteRecord
saveAll
editDVDName
editDVDType
editDVDDescription
editDVDPrice
editDVDCover
closeDBConnection
```

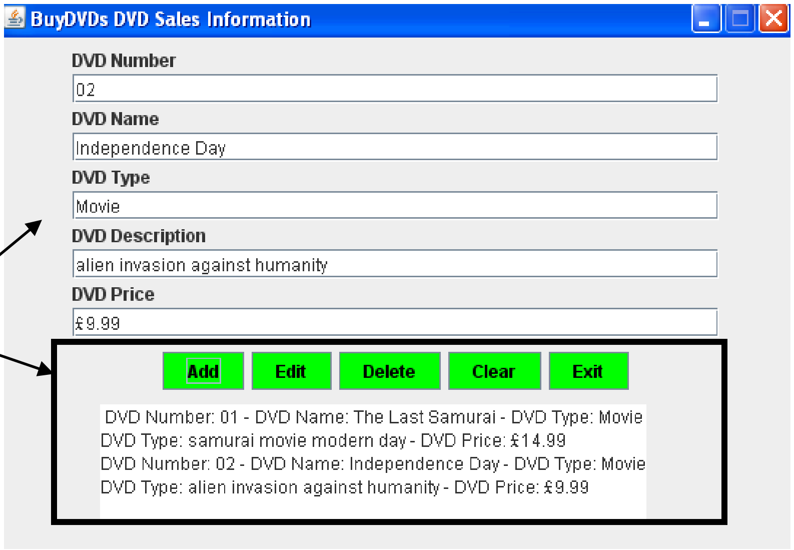
After a while, I managed to fix the testing errors as most of the outputs for those 4 to 6 methods gave the wrong errors mainly because of the spacing of the values that were returned by those methods.

All twenty of my written tests in the test skeleton class were passed eventually.

Screenshots of Program in Operation to Demonstrate Functionality Achieved

USE CASE 1

NOTE: Most of the following screenshots were taking during earlier program development and the layouts may differ from the final program itself.



DVD Number
02

DVD Name
Independence Day

DVD Type
Movie

DVD Description
alien invasion against humanity

DVD Price
£9.99

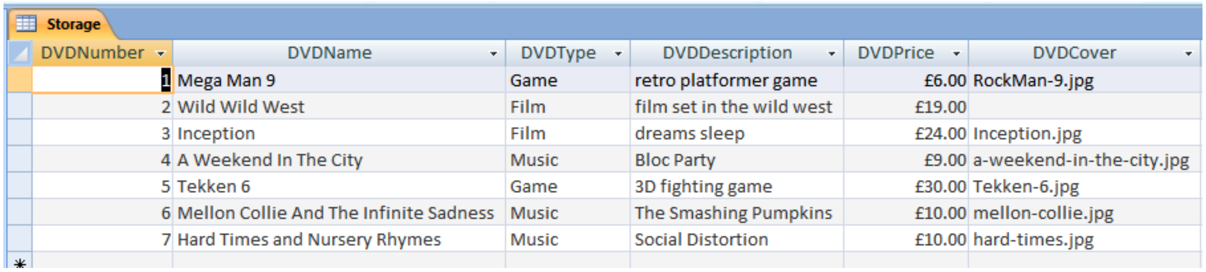
```
 DVD Number: 01 - DVD Name: The Last Samurai - DVD Type: Movie
DVD Type: samurai movie modern day - DVD Price: £14.99
DVD Number: 02 - DVD Name: Independence Day - DVD Type: Movie
DVD Type: alien invasion against humanity - DVD Price: £9.99
```

The screenshot to the left is what my program currently looks like. It has all 5 inputs arranged using a GridLayout on the top, at the bottom of the GUI is the output of the current values saved into the different ArrayLists via the buttons.

In between the visual output and the inputs there are the buttons used to interact with the program. All of the buttons present in the screenshot are working correctly.

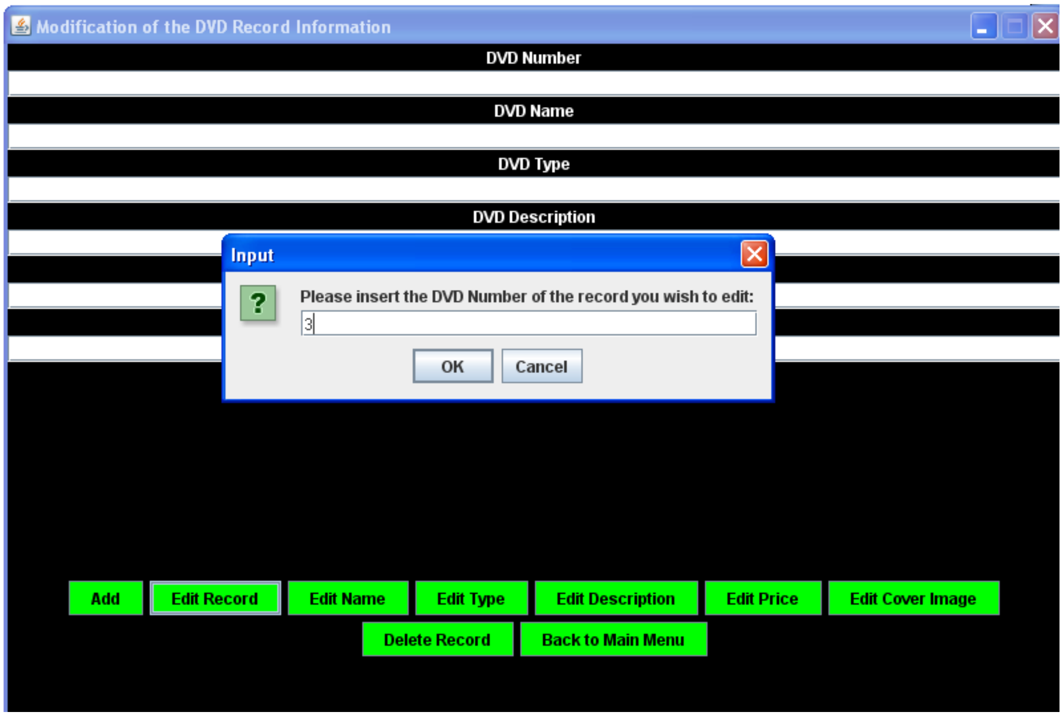
USE CASE 2

Edit Database



DVDNumber	DVDName	DVDType	DVDDescription	DVDPrice	DVDCover
1	Mega Man 9	Game	retro platformer game	£6.00	RockMan-9.jpg
2	Wild Wild West	Film	film set in the wild west	£19.00	
3	Inception	Film	dreams sleep	£24.00	Inception.jpg
4	A Weekend In The City	Music	Bloc Party	£9.00	a-weekend-in-the-city.jpg
5	Tekken 6	Game	3D fighting game	£30.00	Tekken-6.jpg
6	Mellon Collie And The Infinite Sadness	Music	The Smashing Pumpkins	£10.00	mellon-collie.jpg
7	Hard Times and Nursery Rhymes	Music	Social Distortion	£10.00	hard-times.jpg

The image above shows the current records that are stored in the database before I used my program to edit any of them. Compare the screenshot here and the other screenshots later to see how the record was edited.

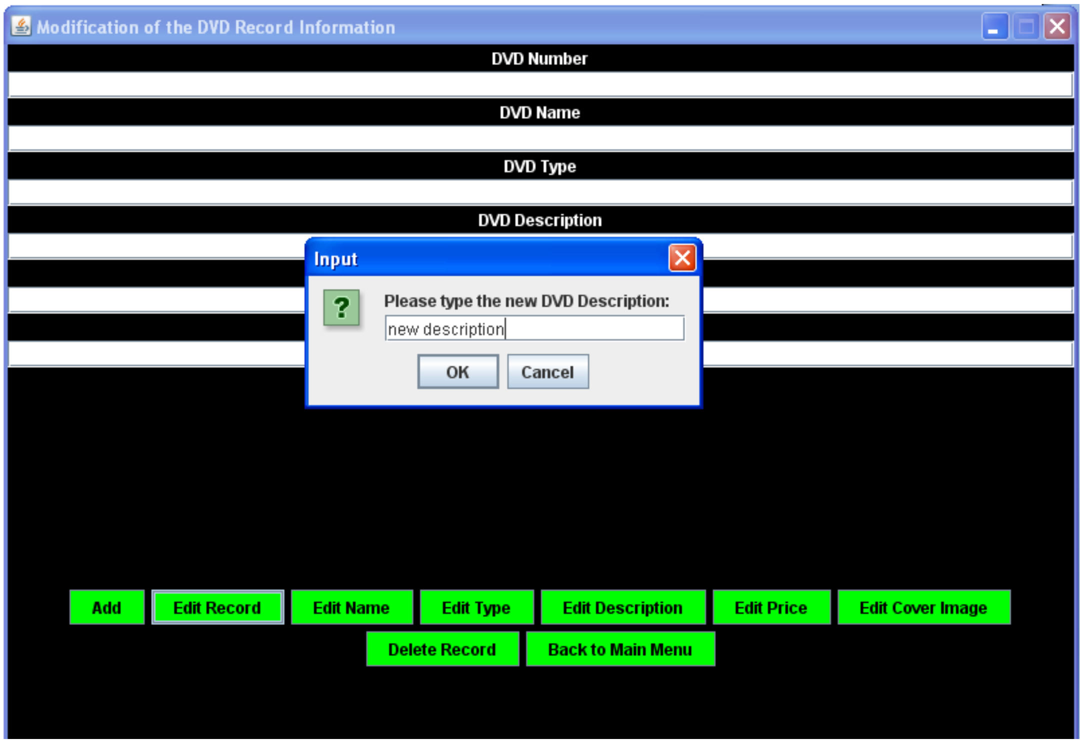


Please insert the DVD Number of the record you wish to edit:

3

When I clicked the Edit Record button on my program, a JOptionPane opened asking me to insert the number of the record that is going to be edited after clicking OK; other JOptionPane windows appeared on screen.

They then asked me to insert the new value for the DVD Name, then a new value for the DVD Type, then a new value for the DVD Description, and then a new value for the DVD Price, then finally a new value for the DVD Cover Image.

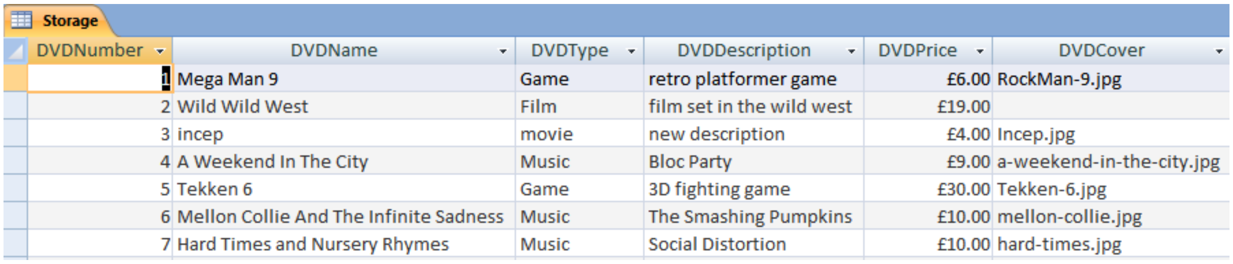


Please type the new DVD Description:

new description

As you can see the window above shows the entry of the new DVD Description, but the JOptionPane windows asking for the new DVDName value and so on are also like this. Similar windows also appear when the Edit Name, Edit Type, Edit Description, Edit Price and Edit Cover Image buttons are clicked when the user wants to change each one separately.

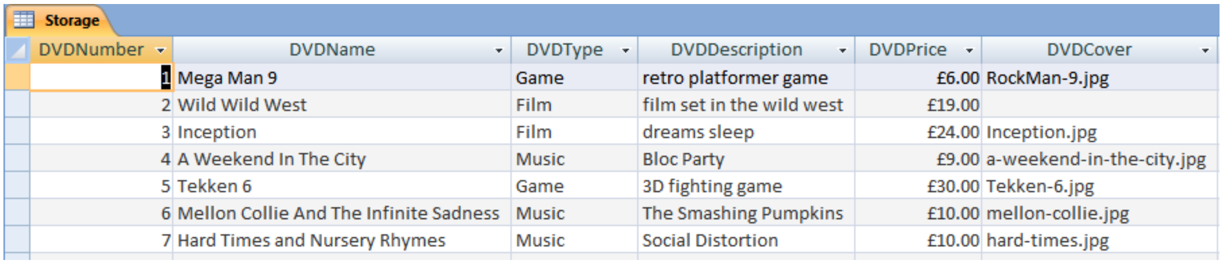
JOptionPane windows are used mainly for the separate field edits (and for editing the record) which require an entry for that one field, this can get confusing with the JTextFields as there are six showing on the program window.



DVDNumber	DVDName	DVDType	DVDDescription	DVDPrice	DVDCover
1	Mega Man 9	Game	retro platformer game	£6.00	RockMan-9.jpg
2	Wild Wild West	Film	film set in the wild west	£19.00	
3	incep	movie	new description	£4.00	Incep.jpg
4	A Weekend In The City	Music	Bloc Party	£9.00	a-weekend-in-the-city.jpg
5	Tekken 6	Game	3D fighting game	£30.00	Tekken-6.jpg
6	Mellon Collie And The Infinite Sadness	Music	The Smashing Pumpkins	£10.00	mellon-collie.jpg
7	Hard Times and Nursery Rhymes	Music	Social Distortion	£10.00	hard-times.jpg

The image above shows the edits that have been made to record number 3. The whole field was edited so it made more sense to use the Edit Record button instead of the separate edit buttons that were also available.

Delete from Database



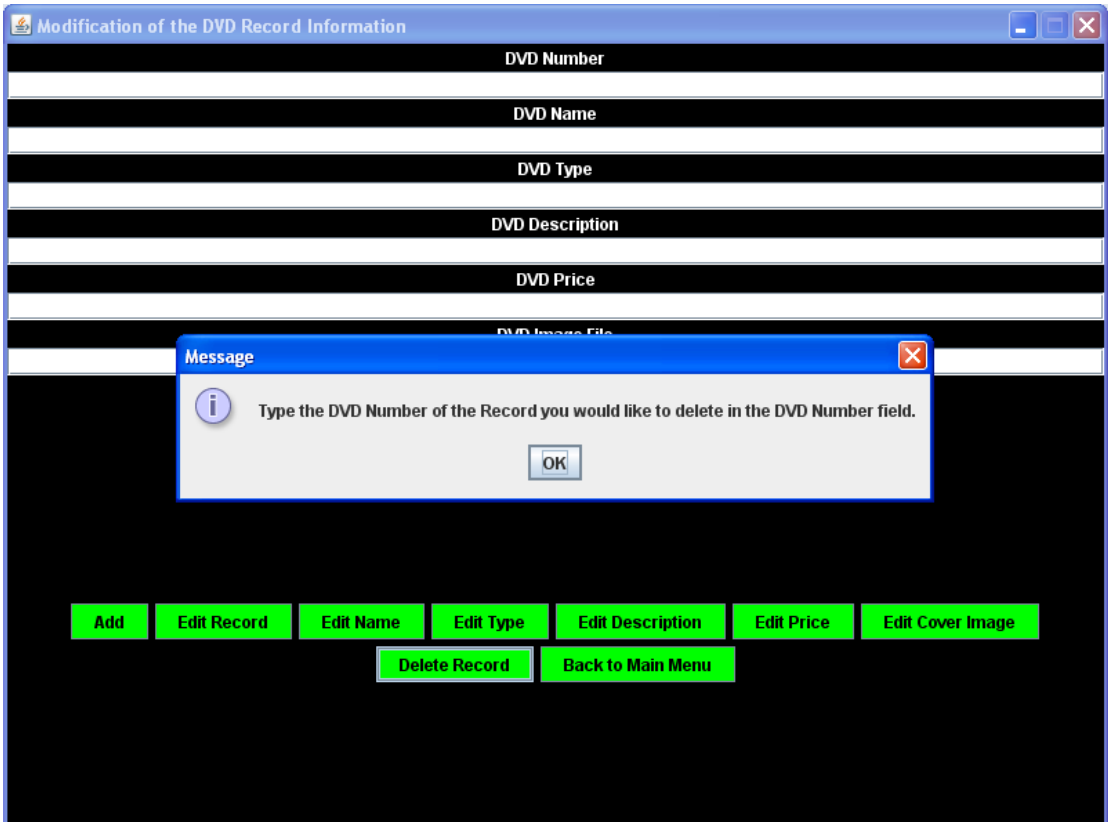
Storage

DVDNumber	DVDName	DVDType	DVDDescription	DVDPrice	DVDCover
1	Mega Man 9	Game	retro platformer game	£6.00	RockMan-9.jpg
2	Wild Wild West	Film	film set in the wild west	£19.00	
3	Inception	Film	dreams sleep	£24.00	Inception.jpg
4	A Weekend In The City	Music	Bloc Party	£9.00	a-weekend-in-the-city.jpg
5	Tekken 6	Game	3D fighting game	£30.00	Tekken-6.jpg
6	Mellon Collie And The Infinite Sadness	Music	The Smashing Pumpkins	£10.00	mellon-collie.jpg
7	Hard Times and Nursery Rhymes	Music	Social Distortion	£10.00	hard-times.jpg

The image above once again shows the records that are currently stored in the external database connected to the program before any edits are made.

The image below shows what happens when you just try to just outright click the Delete button. A message comes up saying that the DVD Number of the Record should be entered on screen for the record you wish to delete.

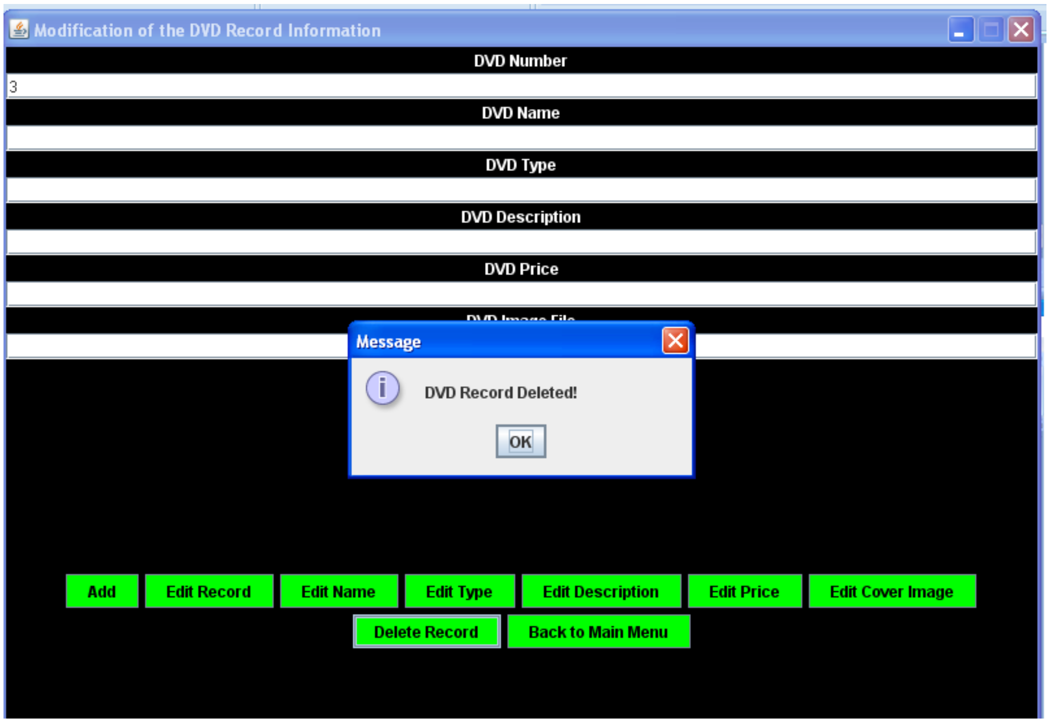
This was done as it’s easier to keep track of which record you just deleted and where you may want to add a new record to the database after deletion.



Modification of the DVD Record Information

Message

Type the DVD Number of the Record you would like to delete in the DVD Number field.



Modification of the DVD Record Information

DVD Number: 3

DVD Name:

DVD Type:

DVD Description:

DVD Price:

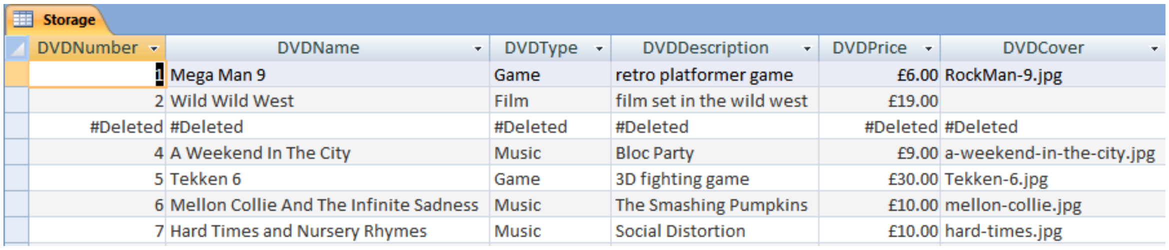
DVD Image File:

Message

DVD Record Deleted!

After typing the DVD Number ‘3’ in the DVD Number JTextField I then clicked on the Delete Record button in order to delete record number 3.

When I had clicked on the Delete Button, a message in a JOptionPane saying “DVD Record Deleted” came up on screen to ensure that the record was deleted.

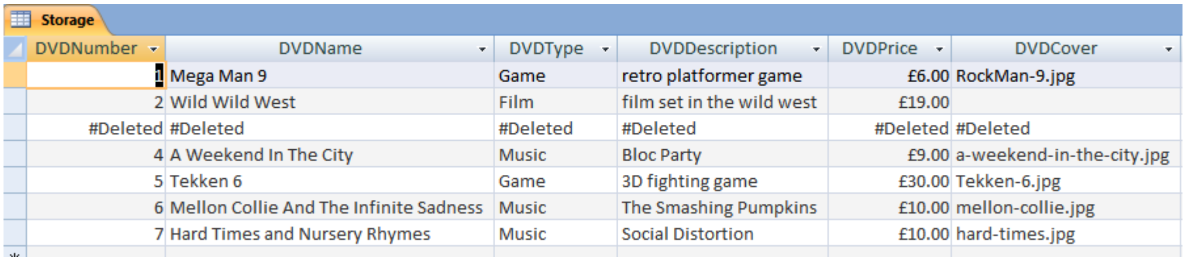


Storage

DVDNumber	DVDName	DVDType	DVDDescription	DVDPrice	DVDCover
1	Mega Man 9	Game	retro platformer game	£6.00	RockMan-9.jpg
2	Wild Wild West	Film	film set in the wild west	£19.00	
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted
4	A Weekend In The City	Music	Bloc Party	£9.00	a-weekend-in-the-city.jpg
5	Tekken 6	Game	3D fighting game	£30.00	Tekken-6.jpg
6	Mellon Collie And The Infinite Sadness	Music	The Smashing Pumpkins	£10.00	mellon-collie.jpg
7	Hard Times and Nursery Rhymes	Music	Social Distortion	£10.00	hard-times.jpg

When I went back to the database to check if it had worked, I found a whole row of the caption “#Deleted” where Record Number 3 was once stored in the database. This allows another record to be added as record number 3 in the old record’s place as well as getting rid of record number 3 from the database completely.

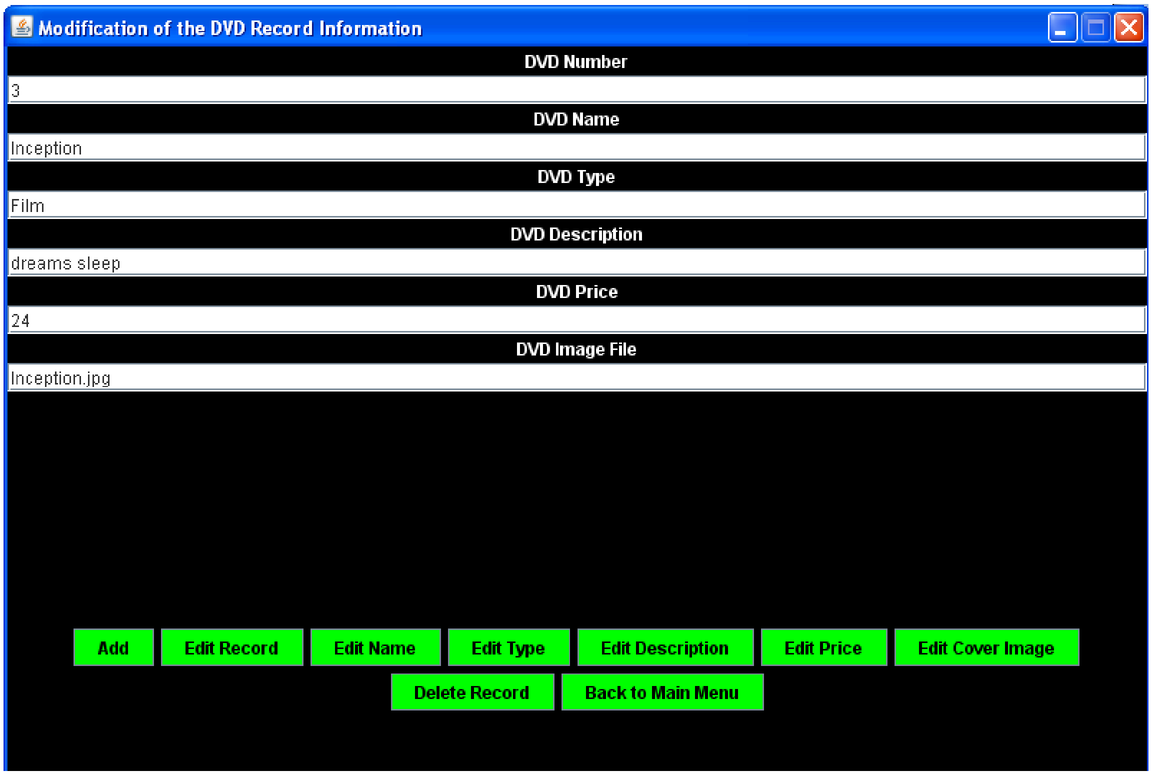
Add to Database



Storage

DVDNumber	DVDName	DVDType	DVDDescription	DVDPrice	DVDCover
1	Mega Man 9	Game	retro platformer game	£6.00	RockMan-9.jpg
2	Wild Wild West	Film	film set in the wild west	£19.00	
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted
4	A Weekend In The City	Music	Bloc Party	£9.00	a-weekend-in-the-city.jpg
5	Tekken 6	Game	3D fighting game	£30.00	Tekken-6.jpg
6	Mellon Collie And The Infinite Sadness	Music	The Smashing Pumpkins	£10.00	mellon-collie.jpg
7	Hard Times and Nursery Rhymes	Music	Social Distortion	£10.00	hard-times.jpg

The image above once again shows all the records stored in the external database that is connected to my program.



Modification of the DVD Record Information

DVD Number: 3

DVD Name: Inception

DVD Type: Film

DVD Description: dreams sleep

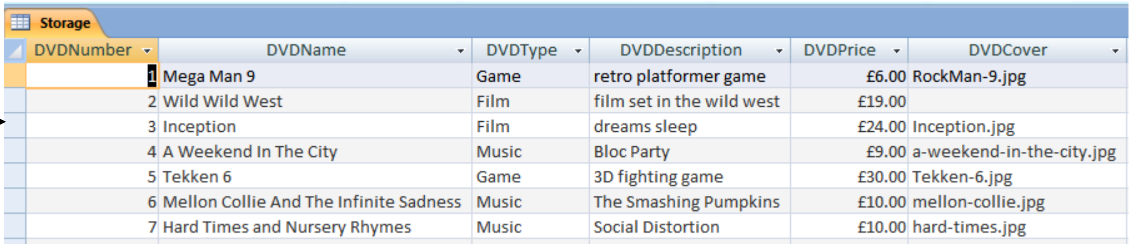
DVD Price: 24

DVD Image File: Inception.jpg

In order to add to database, I simply have to fill in the fields on screen with a DVD Number, a DVD Name, and a DVD Type, a DVD Description, a DVD Price and a DVD Image File name such as “picture.jpg” or full URL such as “my documents/folder/picture.jpg”.

Once I had filled all 6 fields in, I clicked the Add button to add it to the external database so it could be stored there.

Note: This was before I had added in a message confirming to the user that the record had been added.

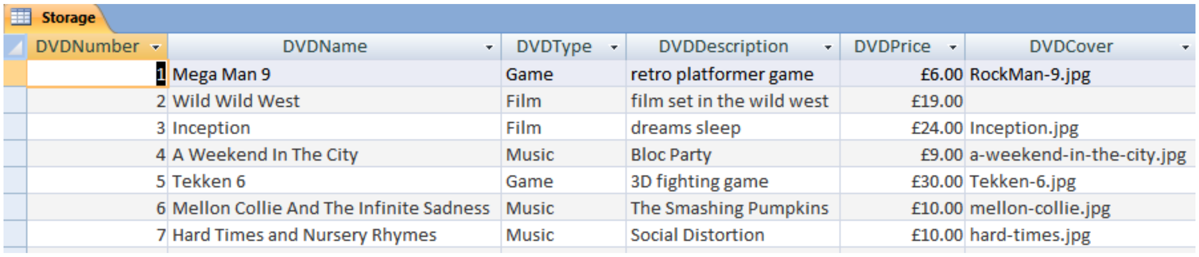


DVDNumber	DVDName	DVDType	DVDDescription	DVDPrice	DVDCover
1	Mega Man 9	Game	retro platformer game	£6.00	RockMan-9.jpg
2	Wild Wild West	Film	film set in the wild west	£19.00	
3	Inception	Film	dreams sleep	£24.00	Inception.jpg
4	A Weekend In The City	Music	Bloc Party	£9.00	a-weekend-in-the-city.jpg
5	Tekken 6	Game	3D fighting game	£30.00	Tekken-6.jpg
6	Mellon Collie And The Infinite Sadness	Music	The Smashing Pumpkins	£10.00	mellon-collie.jpg
7	Hard Times and Nursery Rhymes	Music	Social Distortion	£10.00	hard-times.jpg

This image above shows that record number 3 has been added to the database after it was shown as deleted in the other database record image.

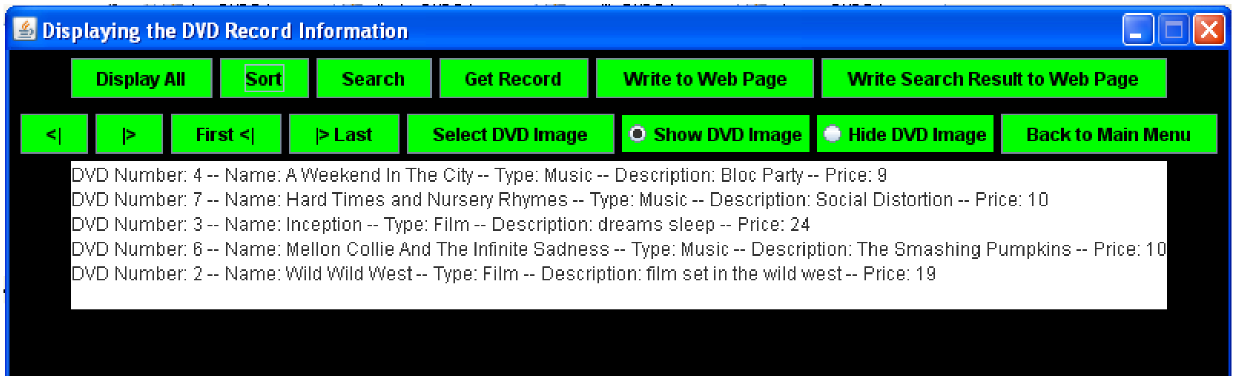
USE CASE 3

List/Sort All into Alphabetical Order



DVDNumber	DVDName	DVDType	DVDDescription	DVDPrice	DVDCover
1	Mega Man 9	Game	retro platformer game	£6.00	RockMan-9.jpg
2	Wild Wild West	Film	film set in the wild west	£19.00	
3	Inception	Film	dreams sleep	£24.00	Inception.jpg
4	A Weekend In The City	Music	Bloc Party	£9.00	a-weekend-in-the-city.jpg
5	Tekken 6	Game	3D fighting game	£30.00	Tekken-6.jpg
6	Mellon Collie And The Infinite Sadness	Music	The Smashing Pumpkins	£10.00	mellon-collie.jpg
7	Hard Times and Nursery Rhymes	Music	Social Distortion	£10.00	hard-times.jpg

This image above shows that record number 3 has been added to the database after it was shown as deleted in the other database record image.

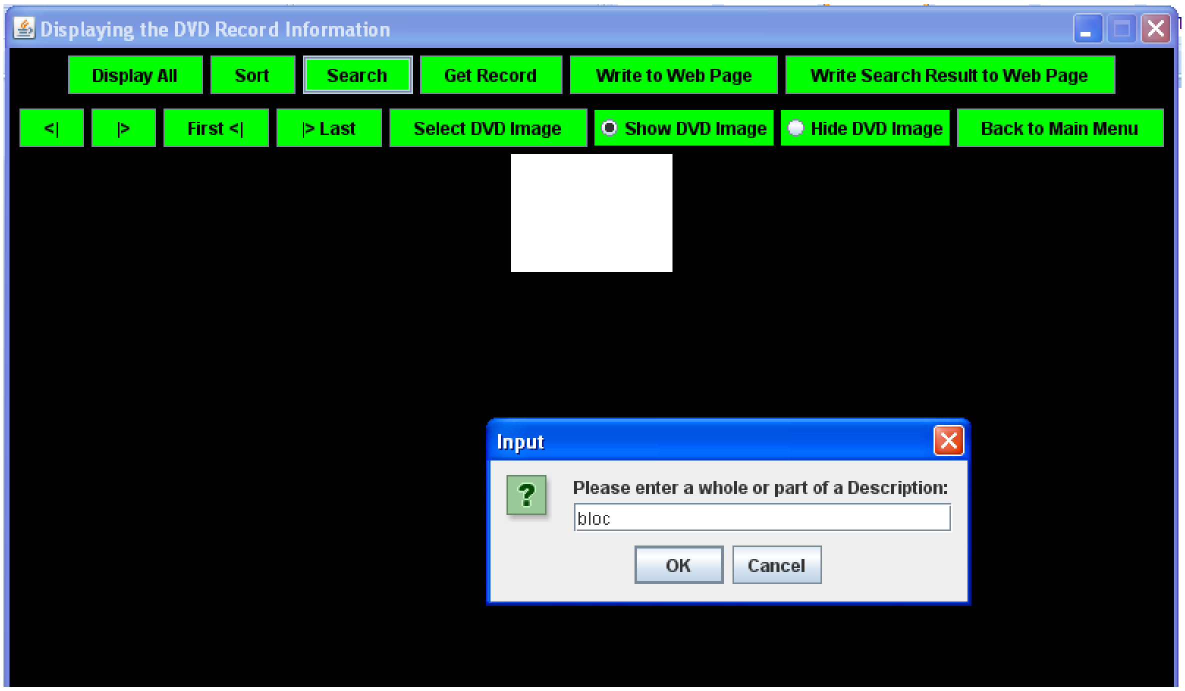


```
DVD Number: 4 -- Name: A Weekend In The City -- Type: Music -- Description: Bloc Party -- Price: 9
DVD Number: 7 -- Name: Hard Times and Nursery Rhymes -- Type: Music -- Description: Social Distortion -- Price: 10
DVD Number: 3 -- Name: Inception -- Type: Film -- Description: dreams sleep -- Price: 24
DVD Number: 6 -- Name: Mellon Collie And The Infinite Sadness -- Type: Music -- Description: The Smashing Pumpkins -- Price: 10
DVD Number: 2 -- Name: Wild Wild West -- Type: Film -- Description: film set in the wild west -- Price: 19
```

When the Sort Button is clicked all of the records in the database are sorted into alphabetical order based on their DVD Name.

Take note that it has been restricted to only show Film and Music DVDs which is the reason why none of the Game DVD's are present in the sorted list shown on the program.

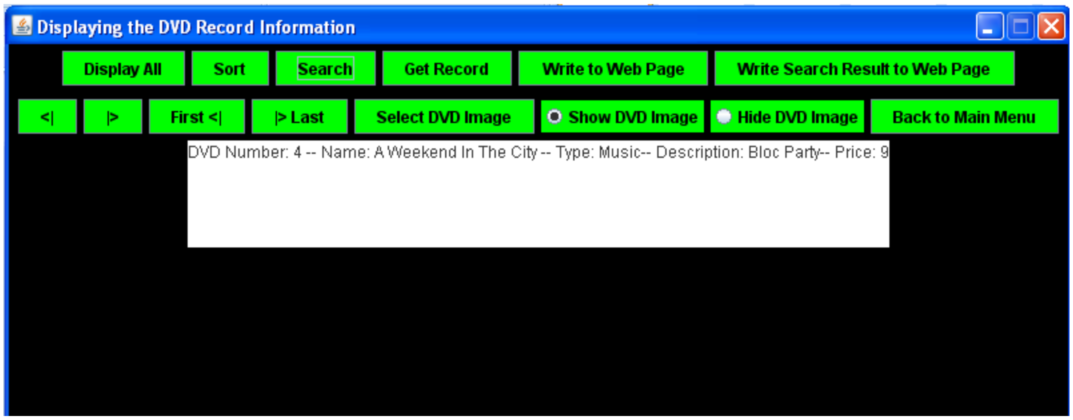
Search Records by Description



Please enter a whole or part of a Description:

bloc

In order to search for records by description the search button needs to be clicked; when it is clicked it brings up a JOptionPane asking for the entry of a whole description of part of a description (words or even letters). When that is done, OK must be clicked for the search to start.

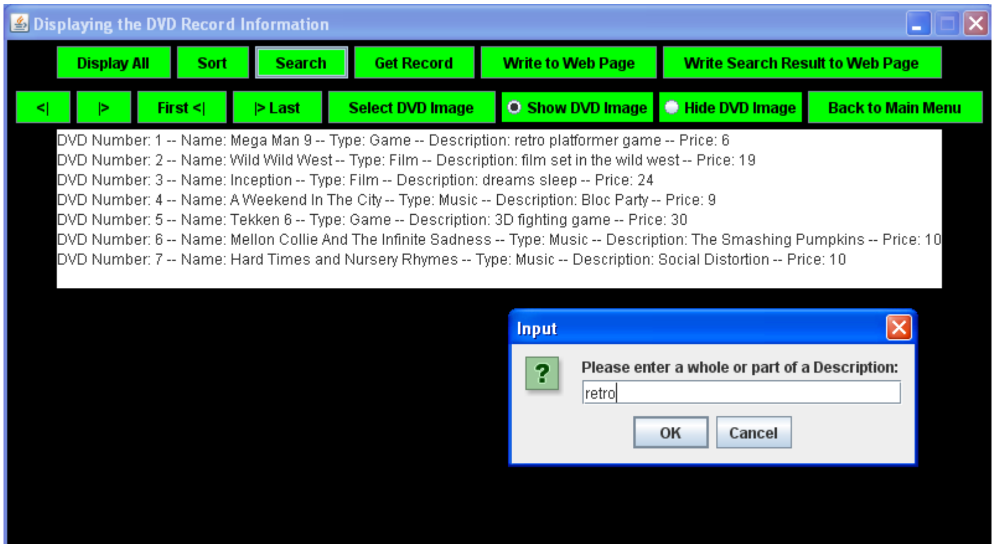


```
DVD Number: 4 -- Name: A Weekend In The City -- Type: Music-- Description: Bloc Party-- Price: 9
```

As you can see in the previous image, I searched for “bloc” and the only record that has bloc in its description is the record with DVD Number 4, so the full details (minus the image) of record number 4 is shown up on screen.

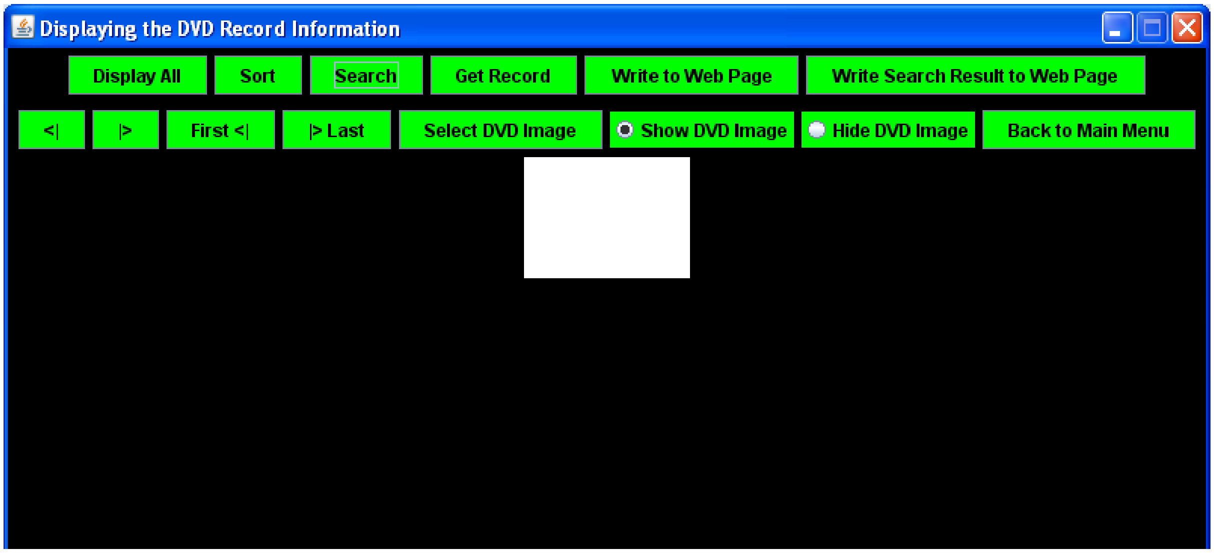
However this search by description feature has also been restricted to showing only Film and Music DVDs.

In the screenshot to the right I type in part of the description for the game DVD of Mega Man 9 (which is DVD Number/Record 1).



```
DVD Number: 1 -- Name: Mega Man 9 -- Type: Game -- Description: retro platformer game -- Price: 6
DVD Number: 2 -- Name: Wild Wild West -- Type: Film -- Description: film set in the wild west -- Price: 19
DVD Number: 3 -- Name: Inception -- Type: Film -- Description: dreams sleep -- Price: 24
DVD Number: 4 -- Name: A Weekend In The City -- Type: Music -- Description: Bloc Party -- Price: 9
DVD Number: 5 -- Name: Tekken 6 -- Type: Game -- Description: 3D fighting game -- Price: 30
DVD Number: 6 -- Name: Mellon Collie And The Infinite Sadness -- Type: Music -- Description: The Smashing Pumpkins -- Price: 10
DVD Number: 7 -- Name: Hard Times and Nursery Rhymes -- Type: Music -- Description: Social Distortion -- Price: 10
```

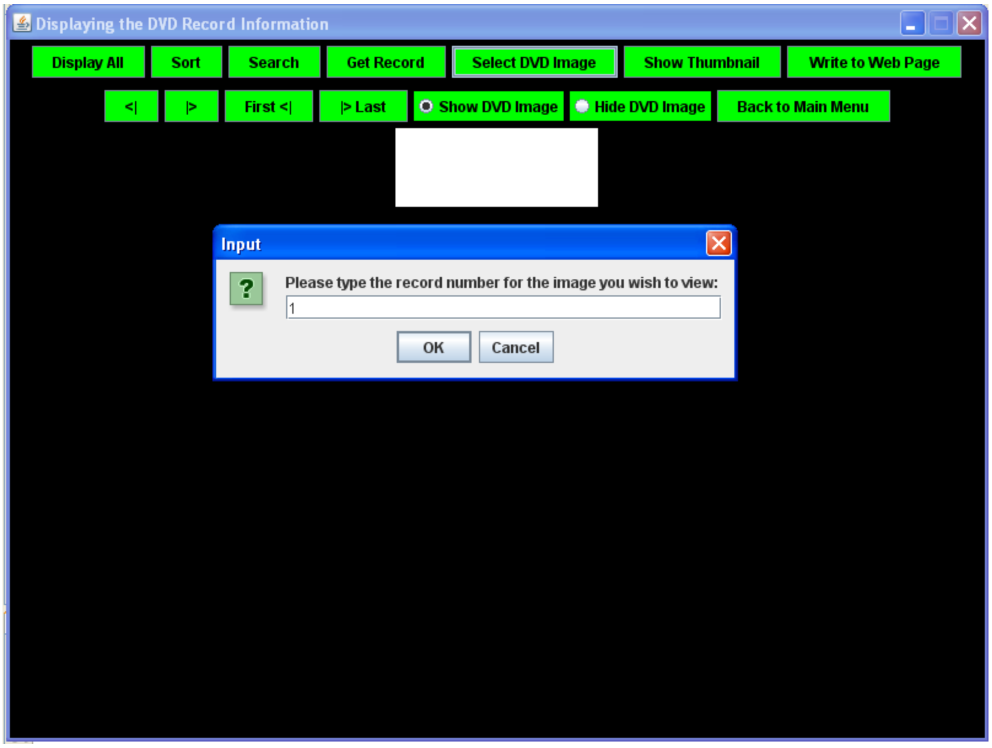
Please enter a whole or part of a Description: retro



After I click the OK button, the screen is left blank as it could not find any records to display that are Film or Music type DVD’s with the description or part of the description containing the word “retro”.

USE CASE 4

Displaying the Image for the Associated DVD

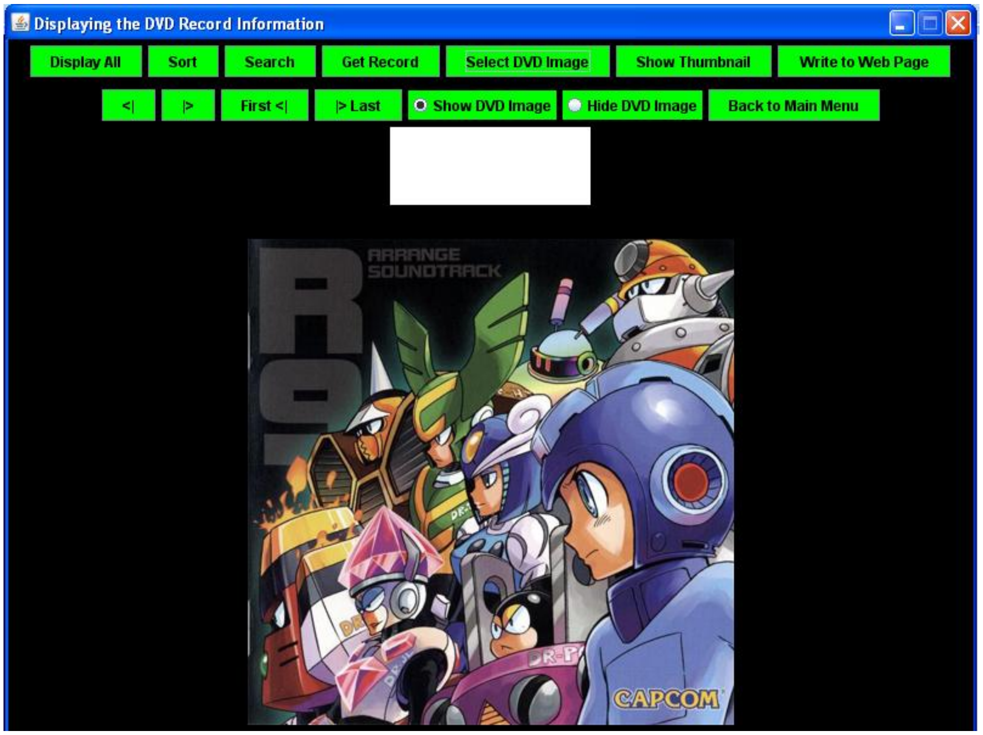


Please type the record number for the image you wish to view:

1

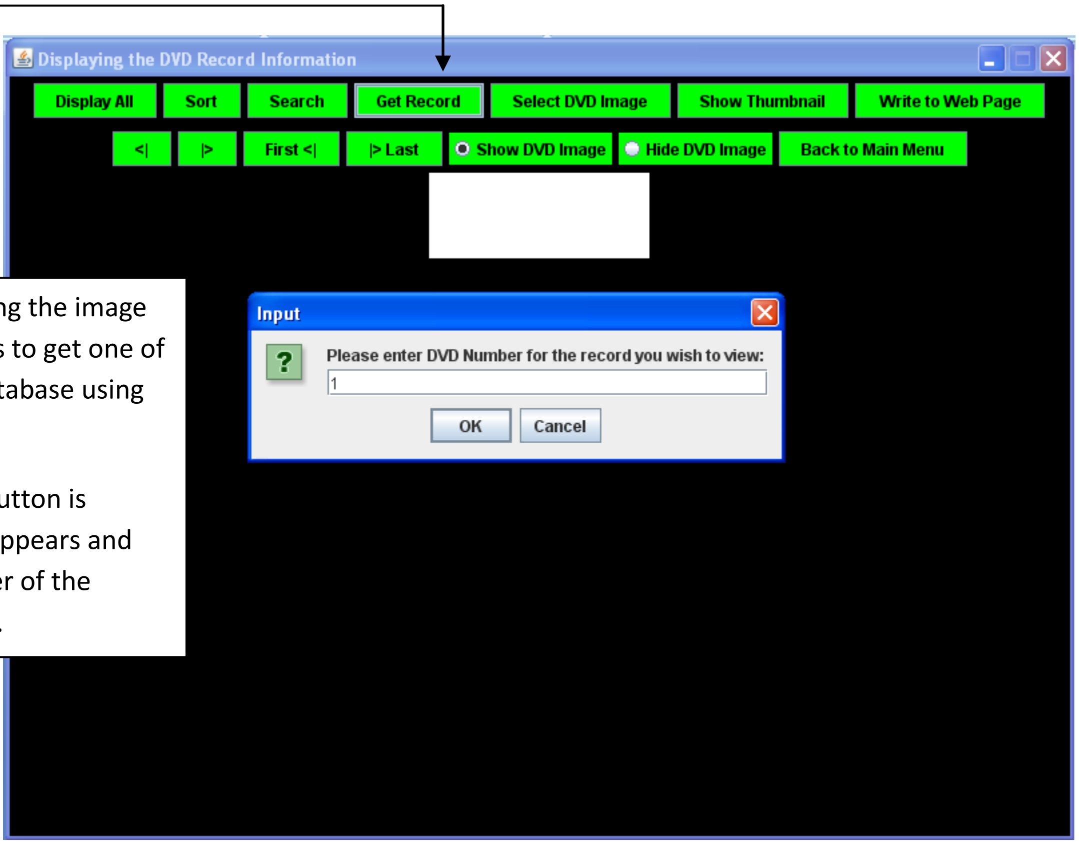
In order to display the image for the associated DVD, the user can click the Select DVD Image button.

This brings up a JOptionPane asking them to enter the record number for the image they wish to see.



Once a record number has been entered and the OK button has been pressed, the image should display at the bottom of the screen.

This is mainly used for viewing just the image by itself or viewing another image while looking at a completely different record.

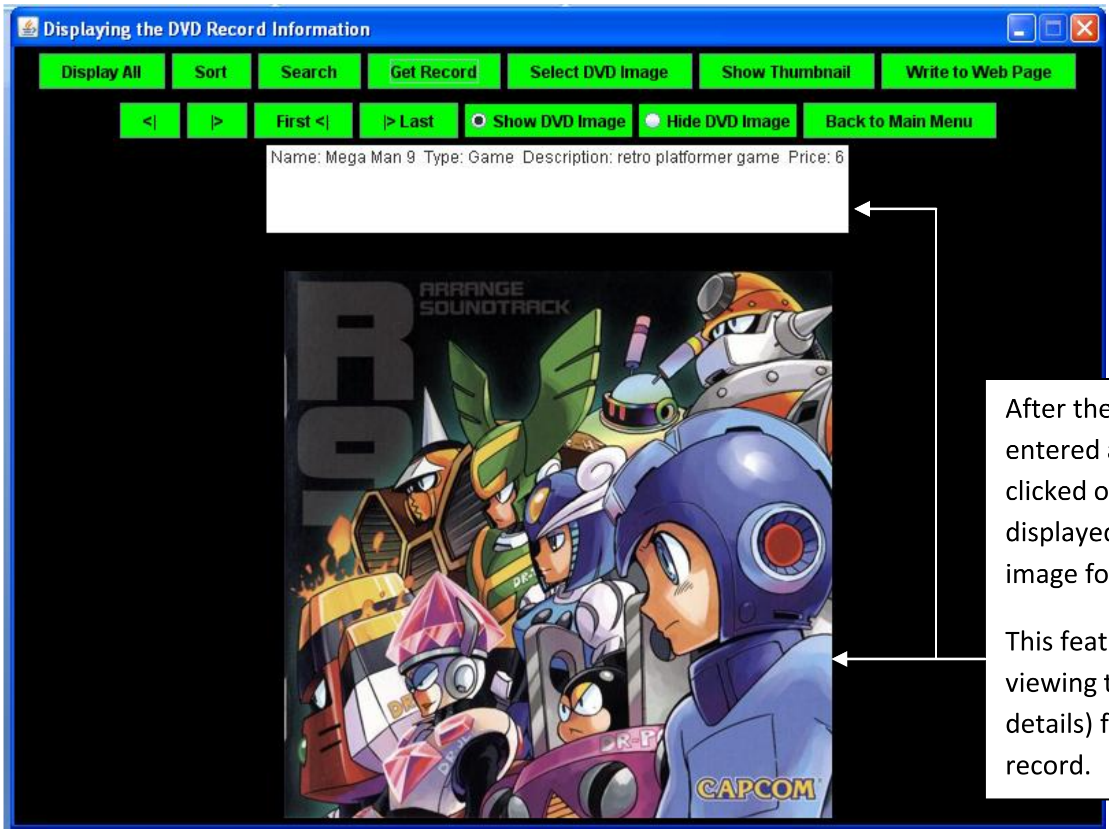


Please enter DVD Number for the record you wish to view:

1

Another way of displaying the image is when the user decides to get one of the records from the database using the Get Record button.

When this Get Record button is clicked, a JOptionPane appears and asks for the DVD Number of the record you wish to view.



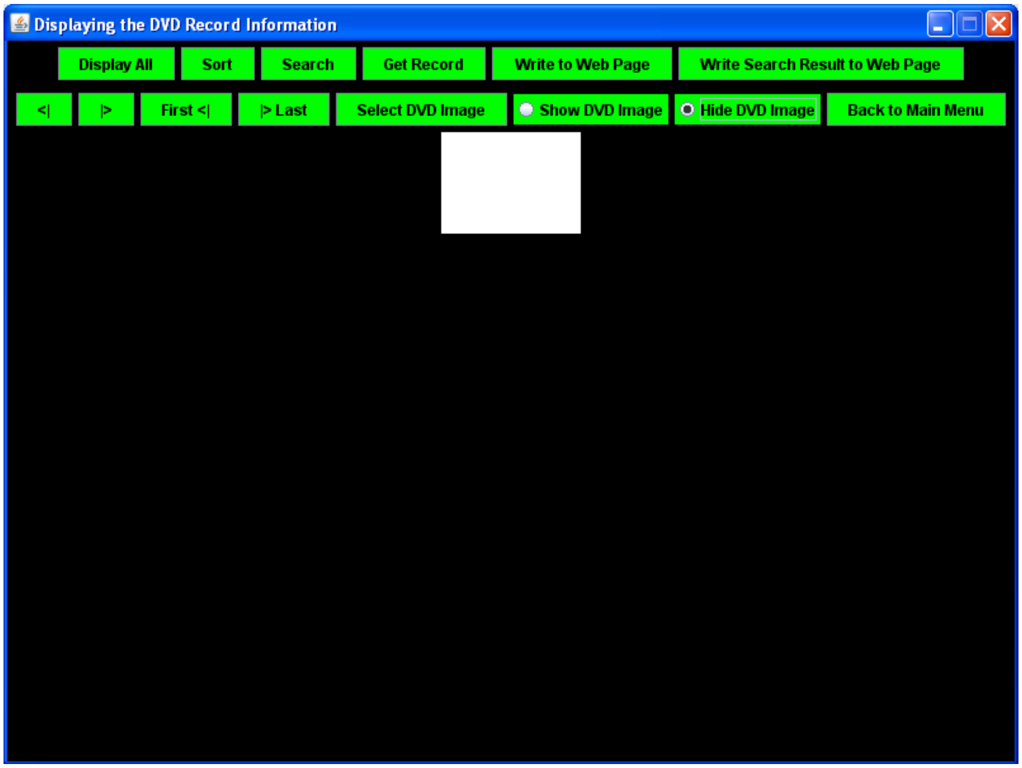
Name: Mega Man 9 Type: Game Description: retro platformer game Price: 6

After the DVD Number is entered and the OK button is clicked on, the record is displayed along with the image for that specific record.

This feature is used for viewing the image (and details) for a single specific record.

It is also possible to hide the displayed DVD Image with the use of the JRadioButton with the caption “Hide DVD Image”.

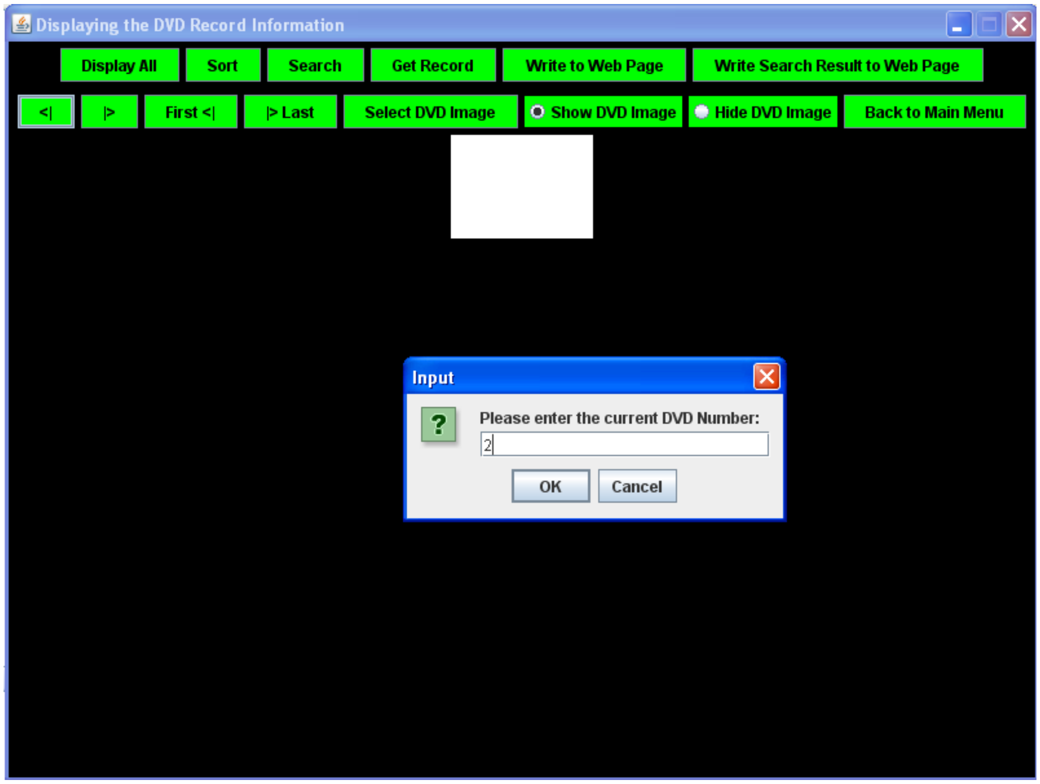
This hides the DVD Image displayed, it can also be re-displayed by clicking on the JradioButton with the caption “Show DVD Image”.



Scrolling through DVDS (Backgrounds and Forwards)

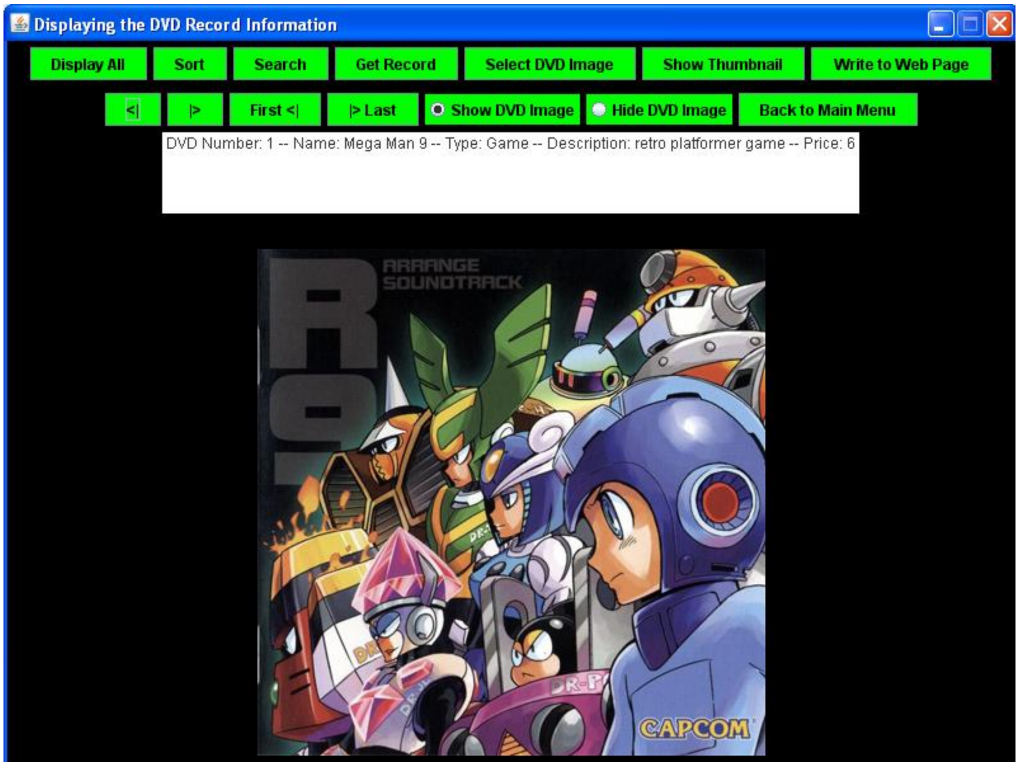
The user is able to scroll through forwards and backwards through DVDS via JButtons.

When clicking the Previous Record JButton with the caption “<|” a JoptionPane appears asking for the current DVD number.

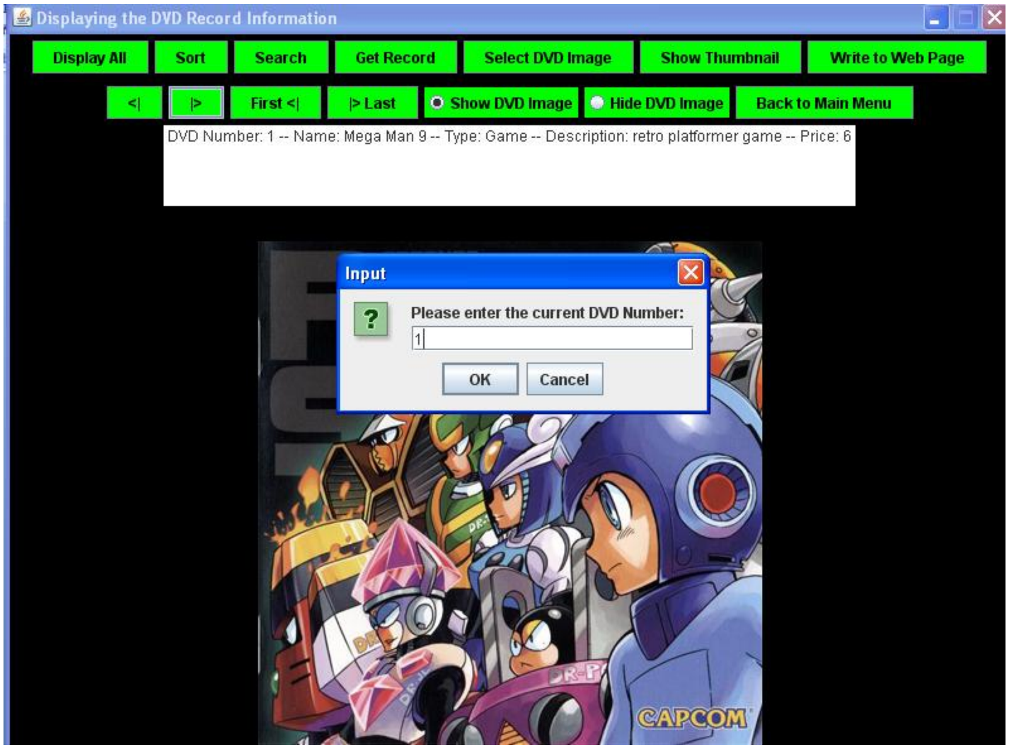


Please enter the current DVD Number:
2

When the current DVD number is entered (shown as 2 in the screenshot) and the OK button has been clicked, the previous record (record 1) is shown up on screen.



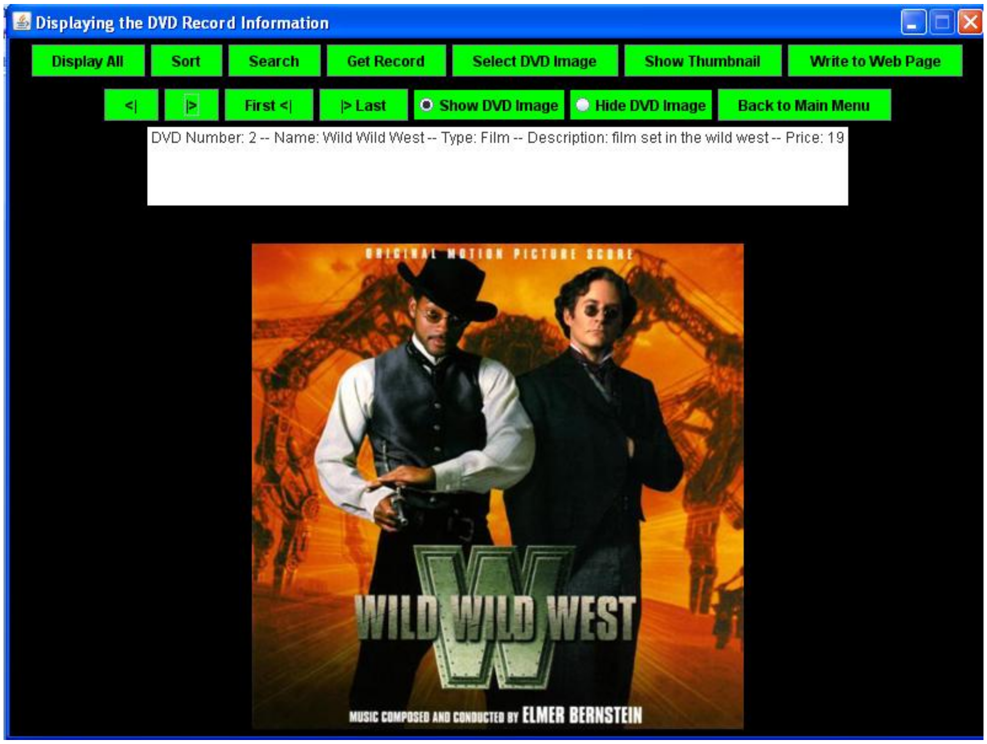
DVD Number: 1 -- Name: Mega Man 9 -- Type: Game -- Description: retro platformer game -- Price: 6



DVD Number: 1 -- Name: Mega Man 9 -- Type: Game -- Description: retro platformer game -- Price: 6

Please enter the current DVD Number: 1

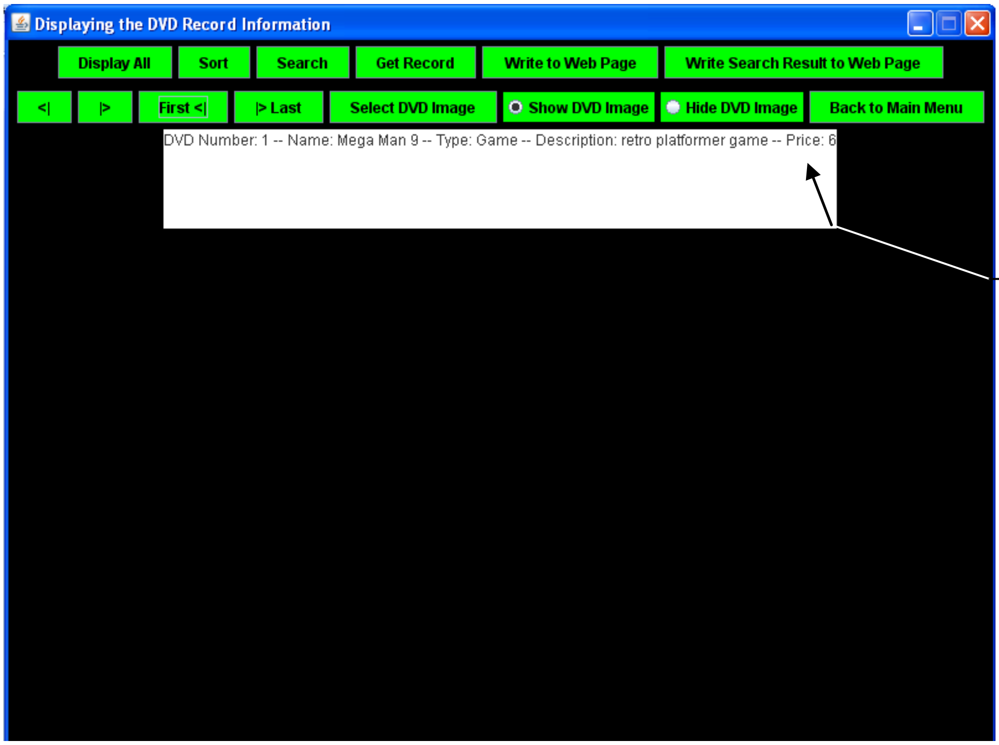
The same process applies to the Next Button as well. When the Next Button with the caption “|>” is clicked, a JoptionPane appears on screen asking for the DVD number for the current record.



DVD Number: 2 -- Name: Wild Wild West -- Type: Film -- Description: film set in the wild west -- Price: 19

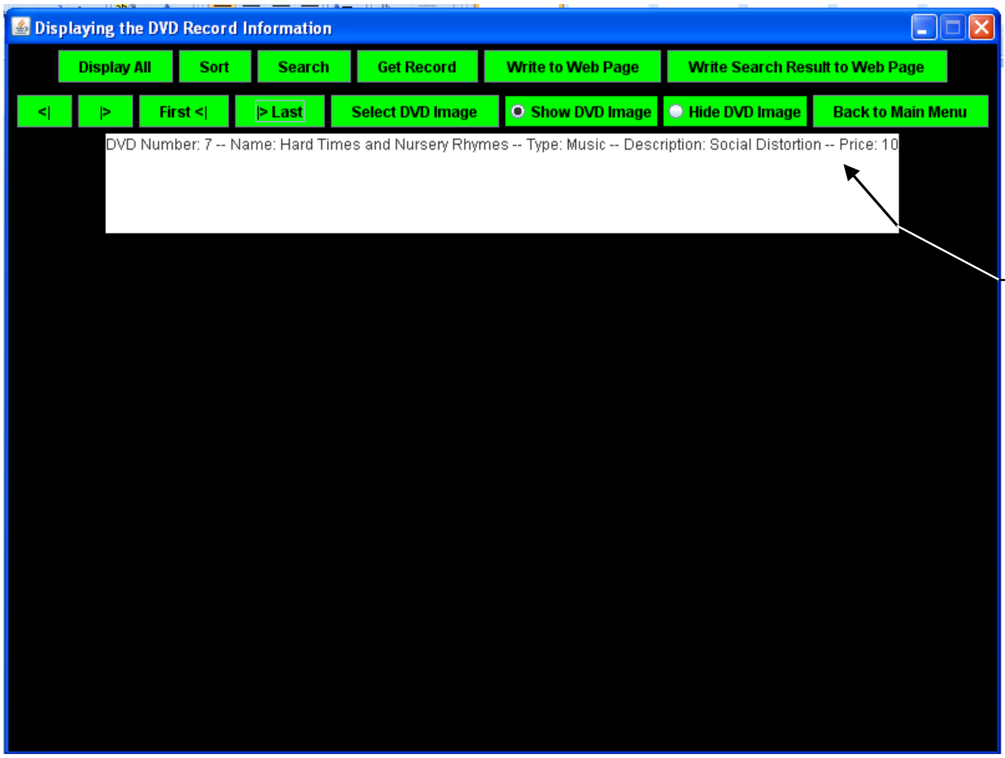
Once again when the DVD Number of the current record (shown as record 1 in the screenshot) has been added and the OK button has been clicked, the next record (record 2) is shown on screen.

Scrolling through DVDS (Skip to First and Skip to Last Records)



DVD Number: 1 -- Name: Mega Man 9 -- Type: Game -- Description: retro platformer game -- Price: 6

With a simple click of the First Record button, the display instantly shows the first record of the database (which has DVD Number 1).



DVD Number: 7 -- Name: Hard Times and Nursery Rhymes -- Type: Music -- Description: Social Distortion -- Price: 10

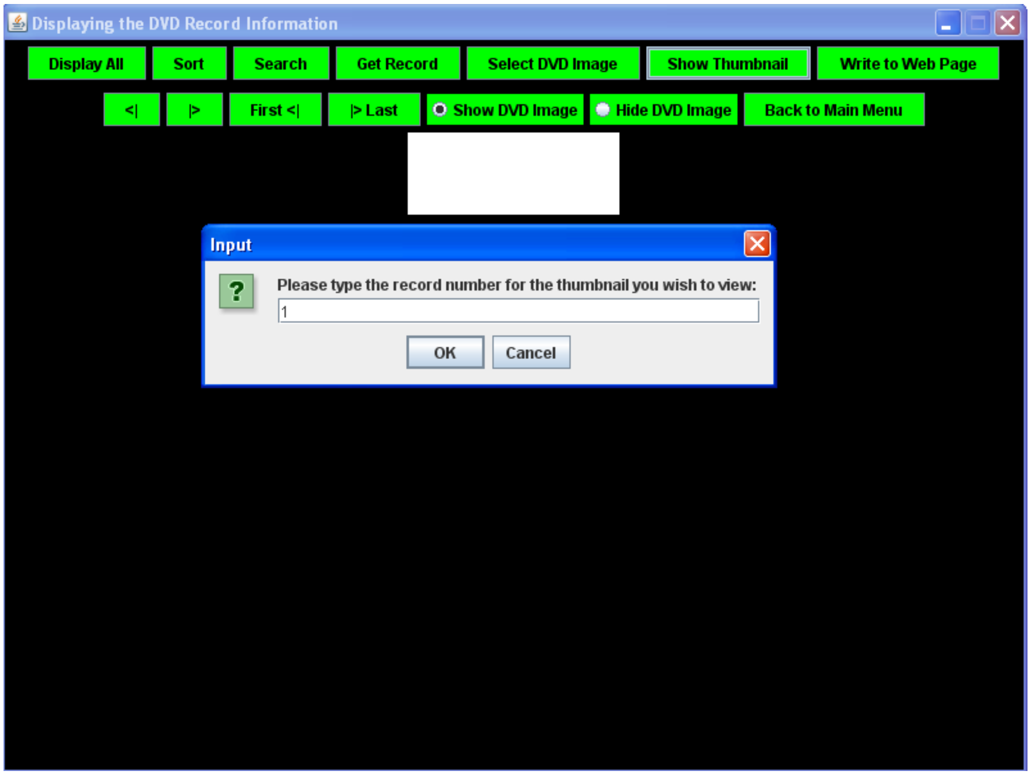
With a simple click of the Last Record button, it displays the last record in the database.

This button functions similarily to the First Record Button except in reverse.

Displaying Thumbnails of Images

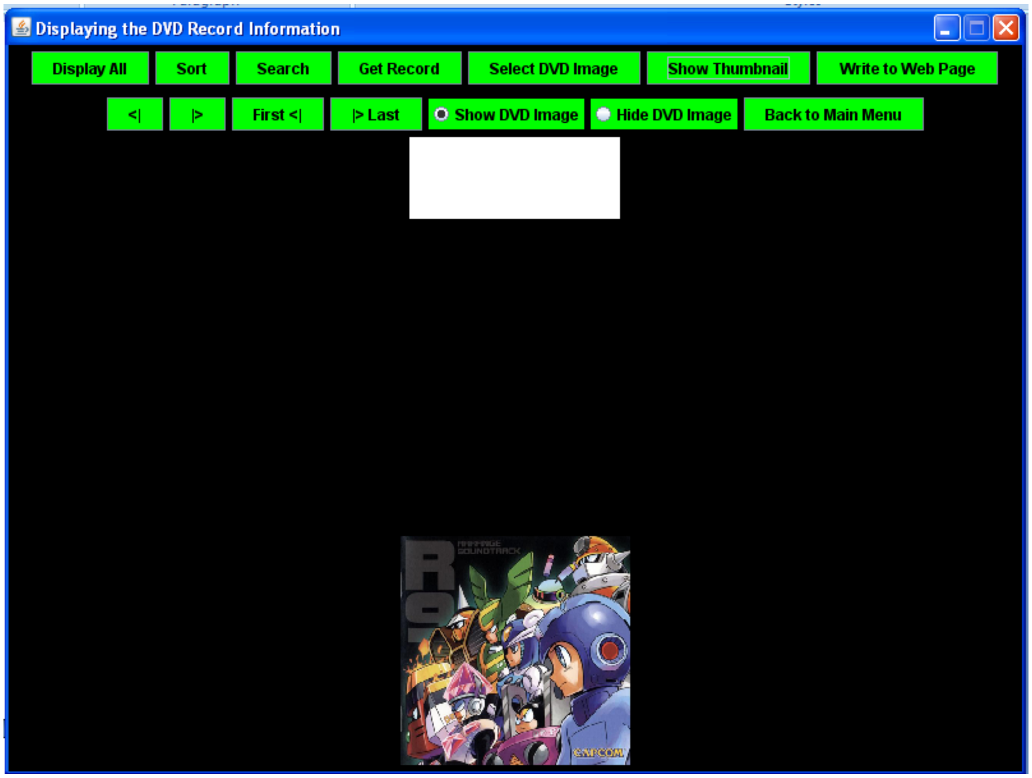
Displaying a thumbnail of an image is quite similar to displaying the image; first the "Show Thumbnail" button needs to be clicked.

Once that is clicked a JOptionPane appears on screen and requests the record number for the thumbnail the user wishes to view.



Please type the record number for the thumbnail you wish to view:

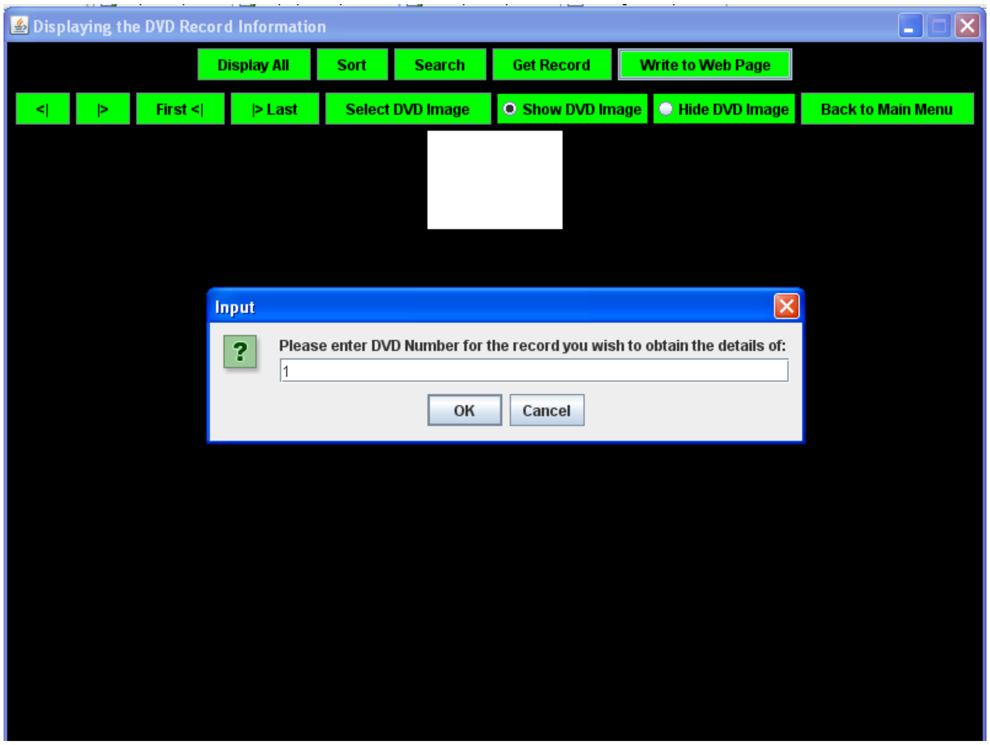
1



Once the record number is entered and the OK button is pressed, a thumbnail of the record's image will appear at the bottom end of the screen for viewing.

USE CASE 5

Creating a Web Page from a DVD Record



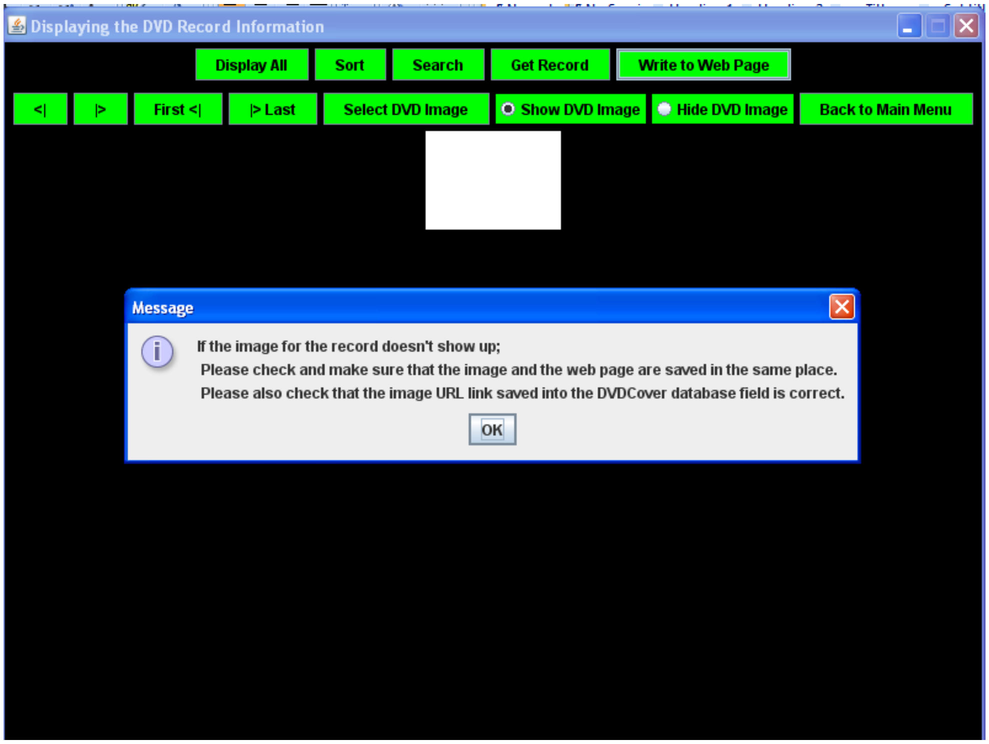
Please enter DVD Number for the record you wish to obtain the details of:

1

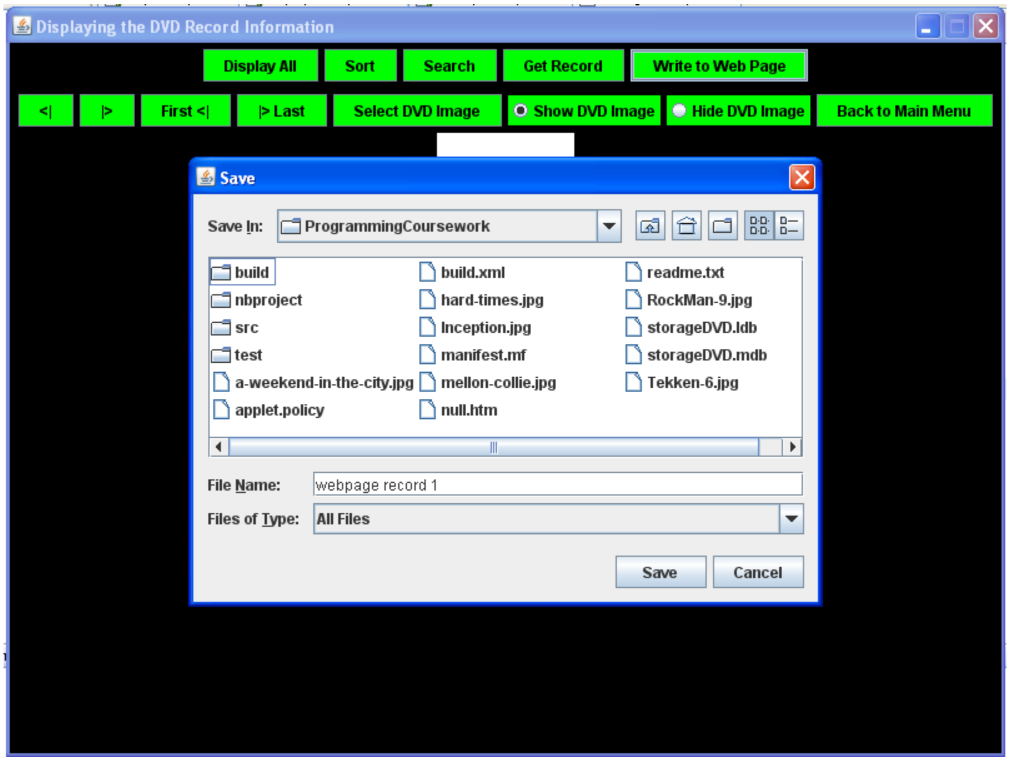
To create a web page from a DVD Record in my program, the Write to Web Page button needs to be clicked.

Once it is a JoptionPane is brought up on screen asking for the entry of the record the user wishes to obtain the details of.

After the user clicks OK, another message will appear on screen informing the user of possible reasons for why the image in the html won't show up, if it doesn't show up. When the OK button in this window is pressed ..



If the image for the record doesn't show up;
Please check and make sure that the image and the web page are saved in the same place.
Please also check that the image URL link saved into the DVDCover database field is correct.



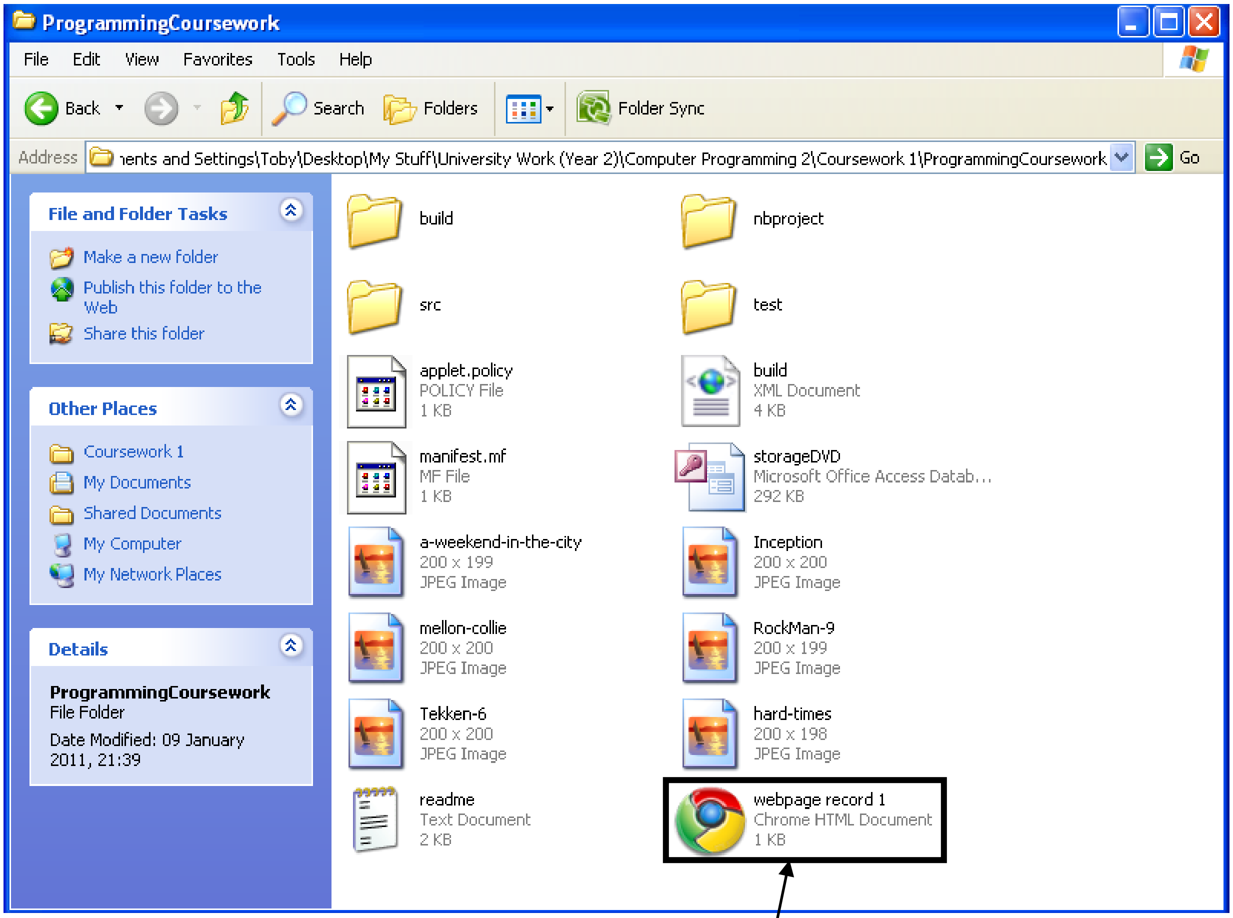
Save In: ProgrammingCoursework

File Name: webpage record 1

Files of Type: All Files

.. a save window appears allowing the user to save anywhere they want with any name they wish.

However if they take notice of the message before, the user should save it somewhere appropriate.



After the save button is clicked in the previous window, the html file will appear in the location where it was saved.

When opened, the html file will display the record details and also display the image as it is saved in the same file as the image with the image URL in the database pointing to that image for the record saved.