Common TextUI / GUI library

The project implements:

4A Text based user interface (TextUI) for the Video Rental Outlet (**VRO**) application.

4B Fail-safe improved code of the text based interface application written in 4A.

4C Graphical user interface (Windows Forms based GUI) for the Video Rental Outlet application.

My solution was to write ***identical application code*** both for the console (TextUI) and windows forms based (GUI) application by isolating business logic of the application from the presentation tier as much as possible. One of the major results of this approach was the **TextUI library**, specially written for this project, which implements basic UI components compatible with the standard .NET Windows Forms library.

The resulting code for 4A/B/C is divided into several assemblies:

Assembly	Source Folder	Description
`VideoRentalOutlet-Console.exe`	`Application`	4A/B/C application
`VideoRentalOutlet-GUI.exe`	`Application`	4A/B/C application
`TextUserInterface.dll`	`Library-TextUI`	4A Text User Interface Library
`VideoRentalOutlet.dll`	`Library-VideoRentalOutlet`	Library (from Lab 2A and 3A)
`VideoRentalOutlet-TestSuite.dll`	`Library-TestSuite`	Test suite (from Lab 3A)
`MbkCommons.dll`	`Library-MbkCommons`	Common utilities

(See the "Project Dependency Graph" section for more details.)

Implementation Notes

Every source folder has as a rule its own VS C# project (a csproject file) that builds the particular assembly. However, because TextUI and GUI assemblies share the same `Application` source folder, that particular folder contains *two* csproject files: one building TextUI application and the other building GUI application. So, the source code is divided in six folders and grouped into seven VS C# projects.

Code Metrics

The summary of the code metrics for the solution is shown on the following table:

Project	Maintability Index	Lines of Code (effective)	Lines of Code (with comments)
`Application GUI + TUI`	71	4 000	16 564
`Library TextUI`	86	3 900	15 047
`Library VideoRentalOutlet`	83	1 067	5 344
`Library TestSuite`	72	629	2 510
`Library MbkCommons`	77	514	2 136
Total	**78**	**10 110**	**41 601**

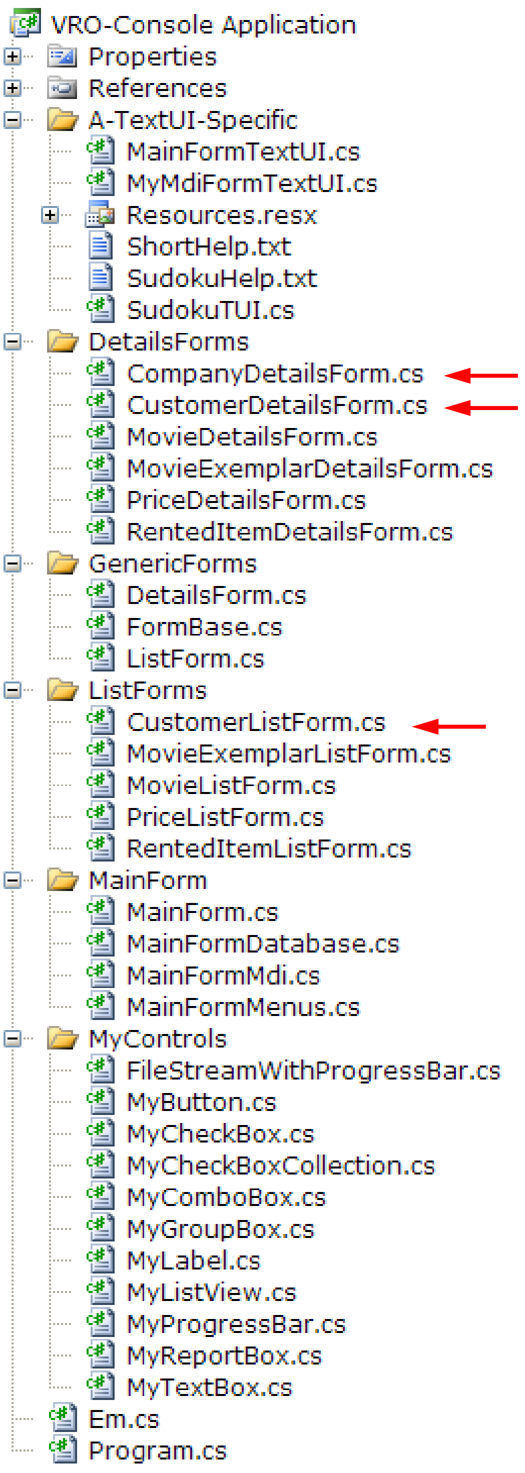
The following pages elaborate:

- source code overview (file listing) of the solution,
- project dependency graph
- essential class diagrams
- how 4A/B/C requirements are met

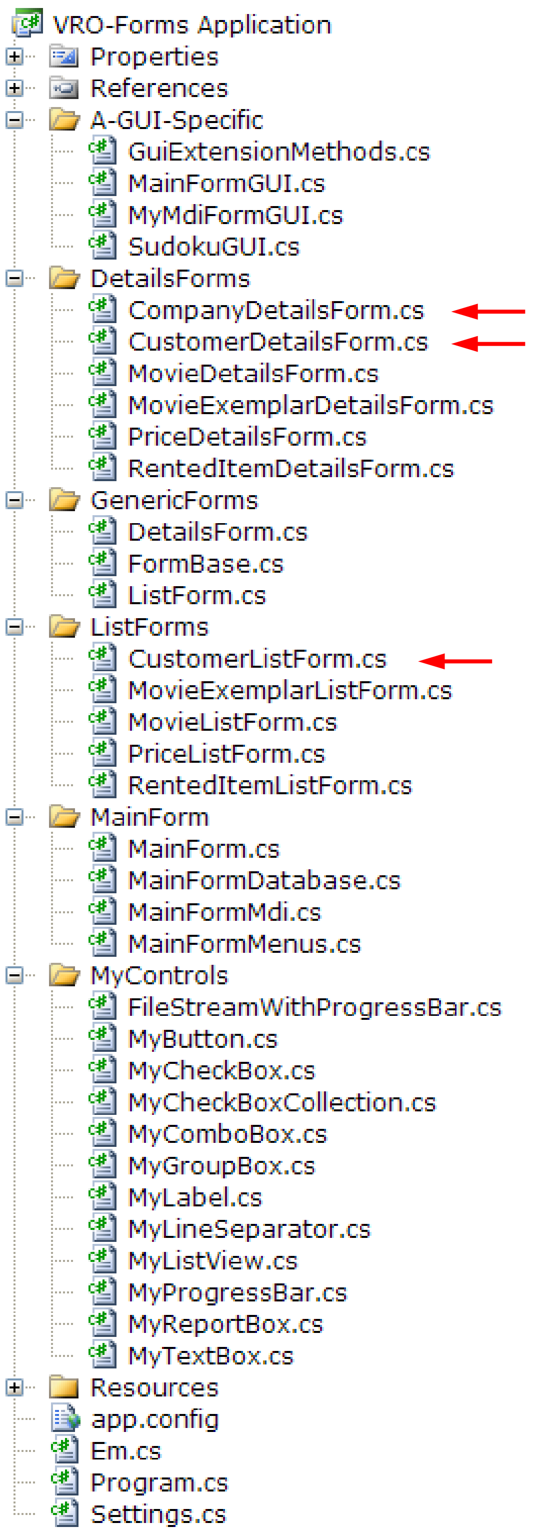
Solution Contents Overview

A good starting point for browsing the code are the following files:

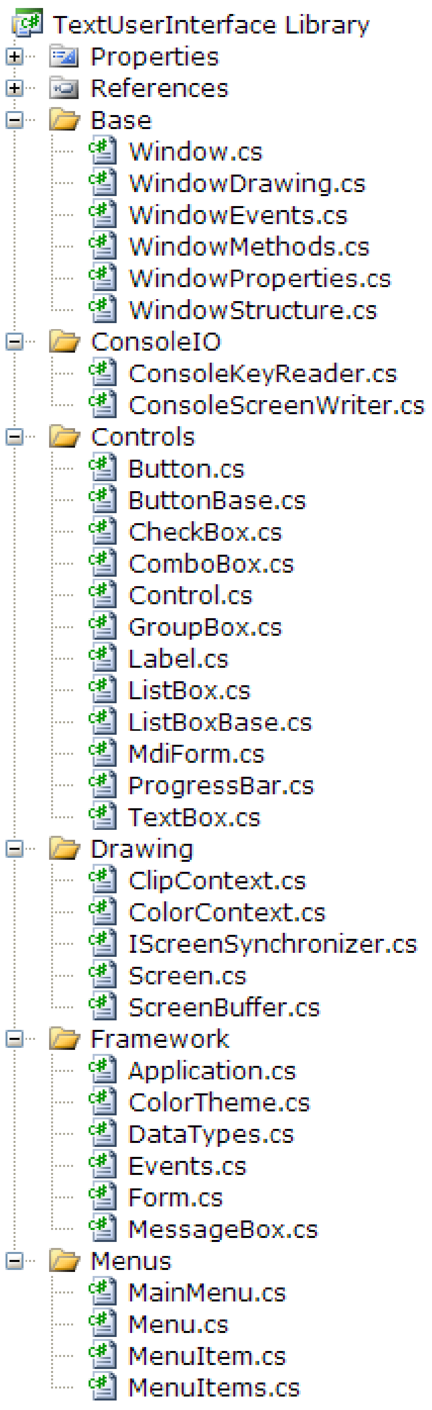
- `Application/DetailsForms/`**`CompanyDetailsForm`**`.cs`
- `Application/DetailsForms/`**`CustomerDetailsForm`**`.cs`
- `Application/ListForms/`**`CustomerListForm`**`.cs`



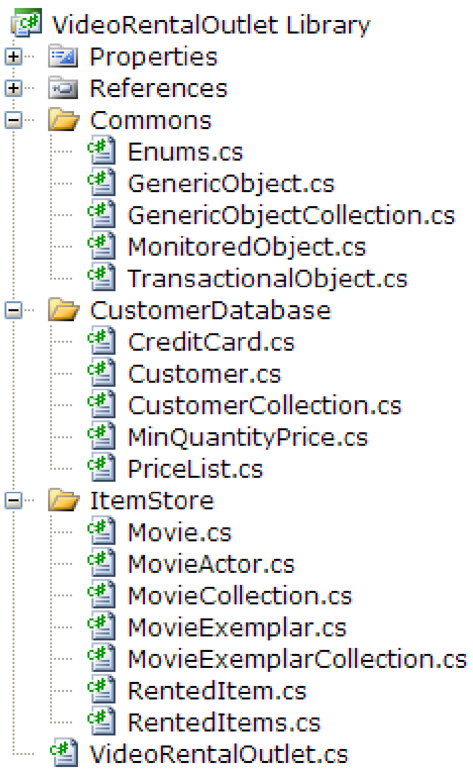
```
VRO-Console Application
+ Properties
+ References
- A-TextUI-Specific
    MainFormTextUI.cs
    MyMdiFormTextUI.cs
  + Resources.resx
    ShortHelp.txt
    SudokuHelp.txt
    SudokuTUI.cs
- DetailsForms
    CompanyDetailsForm.cs        ←
    CustomerDetailsForm.cs       ←
    MovieDetailsForm.cs
    MovieExemplarDetailsForm.cs
    PriceDetailsForm.cs
    RentedItemDetailsForm.cs
- GenericForms
    DetailsForm.cs
    FormBase.cs
    ListForm.cs
- ListForms
    CustomerListForm.cs          ←
    MovieExemplarListForm.cs
    MovieListForm.cs
    PriceListForm.cs
    RentedItemListForm.cs
- MainForm
    MainForm.cs
    MainFormDatabase.cs
    MainFormMdi.cs
    MainFormMenus.cs
- MyControls
    FileStreamWithProgressBar.cs
    MyButton.cs
    MyCheckBox.cs
    MyCheckBoxCollection.cs
    MyComboBox.cs
    MyGroupBox.cs
    MyLabel.cs
    MyListView.cs
    MyProgressBar.cs
    MyReportBox.cs
    MyTextBox.cs
  Em.cs
  Program.cs
```



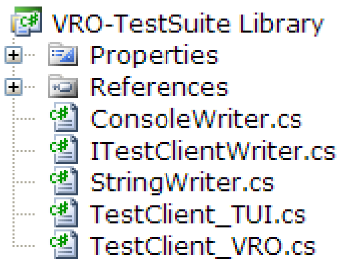
```
VRO-Forms Application
+ Properties
+ References
- A-GUI-Specific
    GuiExtensionMethods.cs
    MainFormGUI.cs
    MyMdiFormGUI.cs
    SudokuGUI.cs
- DetailsForms
    CompanyDetailsForm.cs        ←
    CustomerDetailsForm.cs       ←
    MovieDetailsForm.cs
    MovieExemplarDetailsForm.cs
    PriceDetailsForm.cs
    RentedItemDetailsForm.cs
- GenericForms
    DetailsForm.cs
    FormBase.cs
    ListForm.cs
- ListForms
    CustomerListForm.cs          ←
    MovieExemplarListForm.cs
    MovieListForm.cs
    PriceListForm.cs
    RentedItemListForm.cs
- MainForm
    MainForm.cs
    MainFormDatabase.cs
    MainFormMdi.cs
    MainFormMenus.cs
- MyControls
    FileStreamWithProgressBar.cs
    MyButton.cs
    MyCheckBox.cs
    MyCheckBoxCollection.cs
    MyComboBox.cs
    MyGroupBox.cs
    MyLabel.cs
    MyLineSeparator.cs
    MyListView.cs
    MyProgressBar.cs
    MyReportBox.cs
    MyTextBox.cs
+ Resources
  app.config
  Em.cs
  Program.cs
  Settings.cs
```



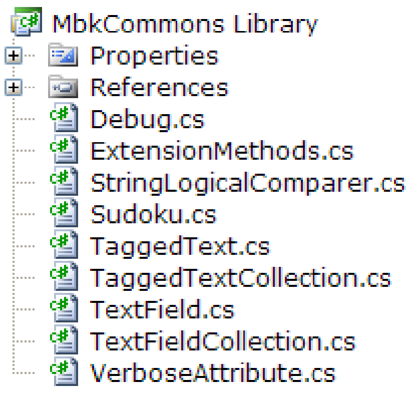
```
TextUserInterface Library
  Properties
  References
  Base
    Window.cs
    WindowDrawing.cs
    WindowEvents.cs
    WindowMethods.cs
    WindowProperties.cs
    WindowStructure.cs
  ConsoleIO
    ConsoleKeyReader.cs
    ConsoleScreenWriter.cs
  Controls
    Button.cs
    ButtonBase.cs
    CheckBox.cs
    ComboBox.cs
    Control.cs
    GroupBox.cs
    Label.cs
    ListBox.cs
    ListBoxBase.cs
    MdiForm.cs
    ProgressBar.cs
    TextBox.cs
  Drawing
    ClipContext.cs
    ColorContext.cs
    IScreenSynchronizer.cs
    Screen.cs
    ScreenBuffer.cs
  Framework
    Application.cs
    ColorTheme.cs
    DataTypes.cs
    Events.cs
    Form.cs
    MessageBox.cs
  Menus
    MainMenu.cs
    Menu.cs
    MenuItem.cs
    MenuItems.cs
```



```
VideoRentalOutlet Library
  Properties
  References
  Commons
    Enums.cs
    GenericObject.cs
    GenericObjectCollection.cs
    MonitoredObject.cs
    TransactionalObject.cs
  CustomerDatabase
    CreditCard.cs
    Customer.cs
    CustomerCollection.cs
    MinQuantityPrice.cs
    PriceList.cs
  ItemStore
    Movie.cs
    MovieActor.cs
    MovieCollection.cs
    MovieExemplar.cs
    MovieExemplarCollection.cs
    RentedItem.cs
    RentedItems.cs
  VideoRentalOutlet.cs
```



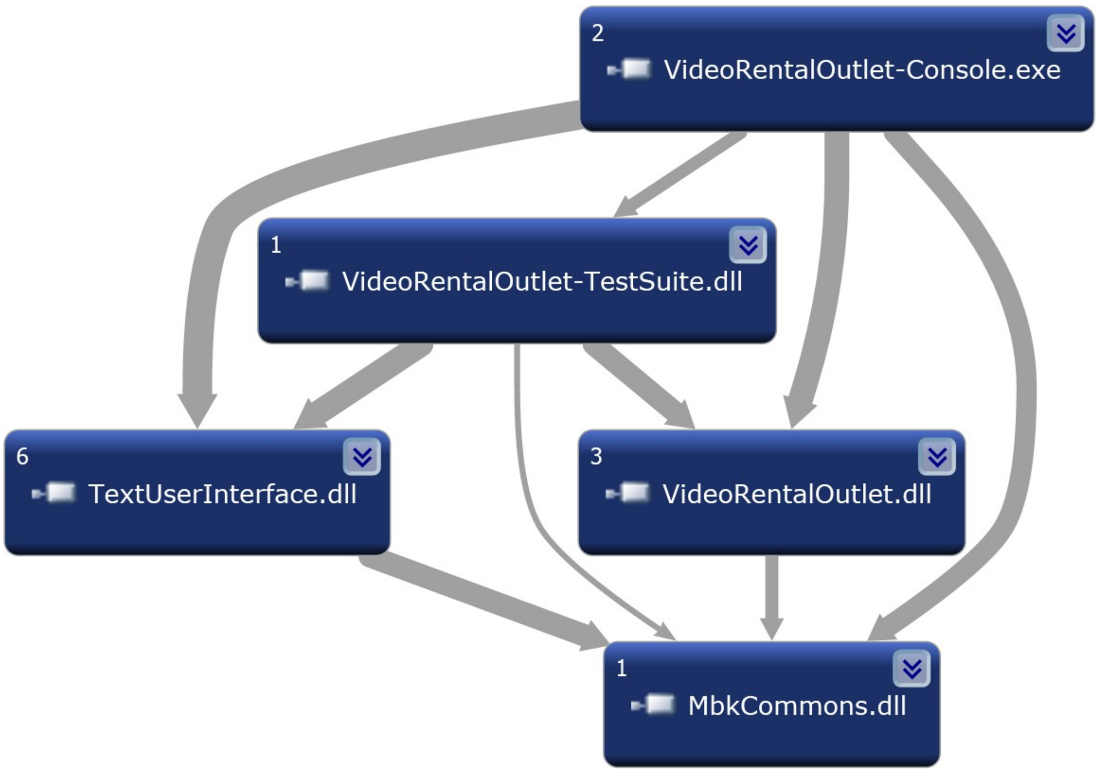
```
VRO-TestSuite Library
  Properties
  References
  ConsoleWriter.cs
  ITestClientWriter.cs
  StringWriter.cs
  TestClient_TUI.cs
  TestClient_VRO.cs
```



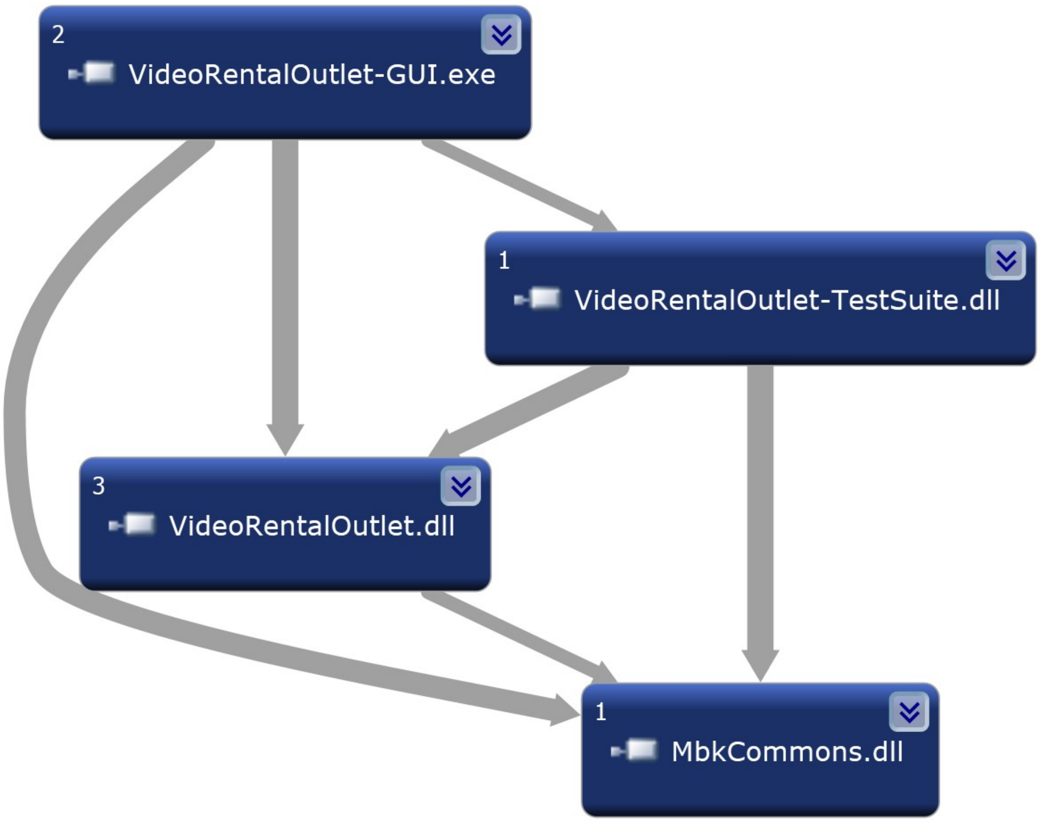
```
MbkCommons Library
  Properties
  References
  Debug.cs
  ExtensionMethods.cs
  StringLogicalComparer.cs
  Sudoku.cs
  TaggedText.cs
  TaggedTextCollection.cs
  TextField.cs
  TextFieldCollection.cs
  VerboseAttribute.cs
```

Project Dependency Graph

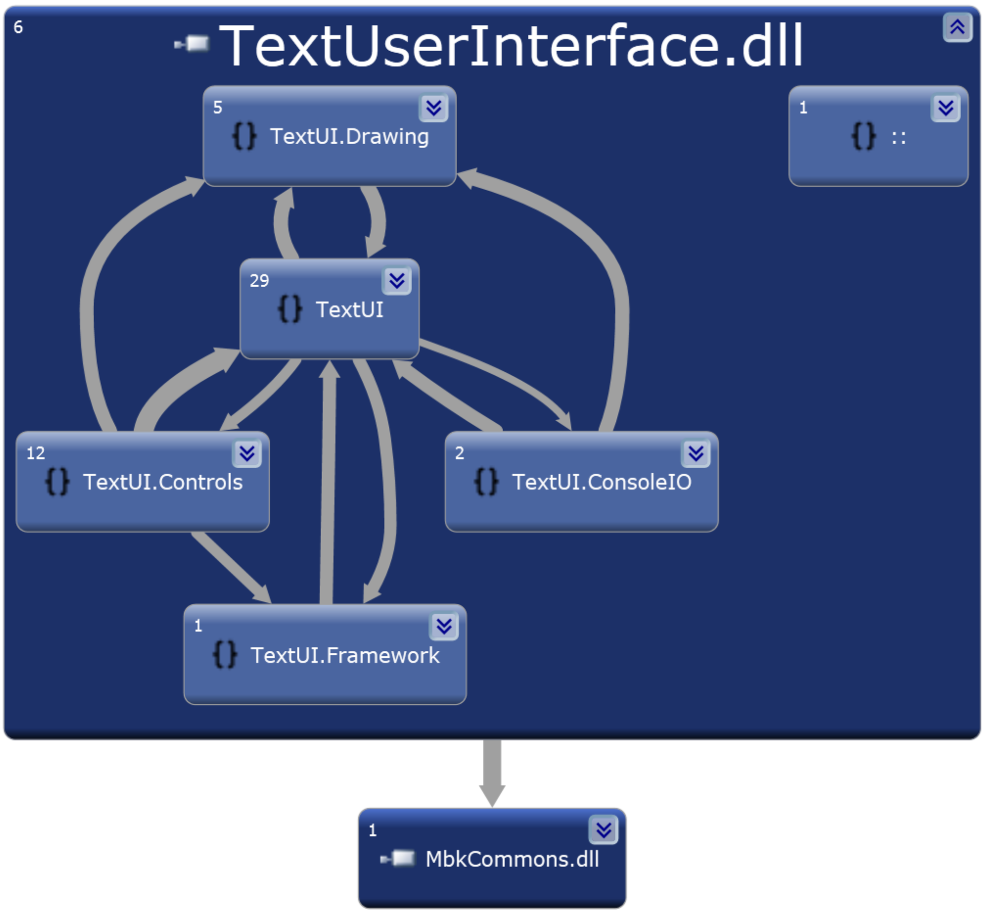
Video Rental Outlet TextUI Application



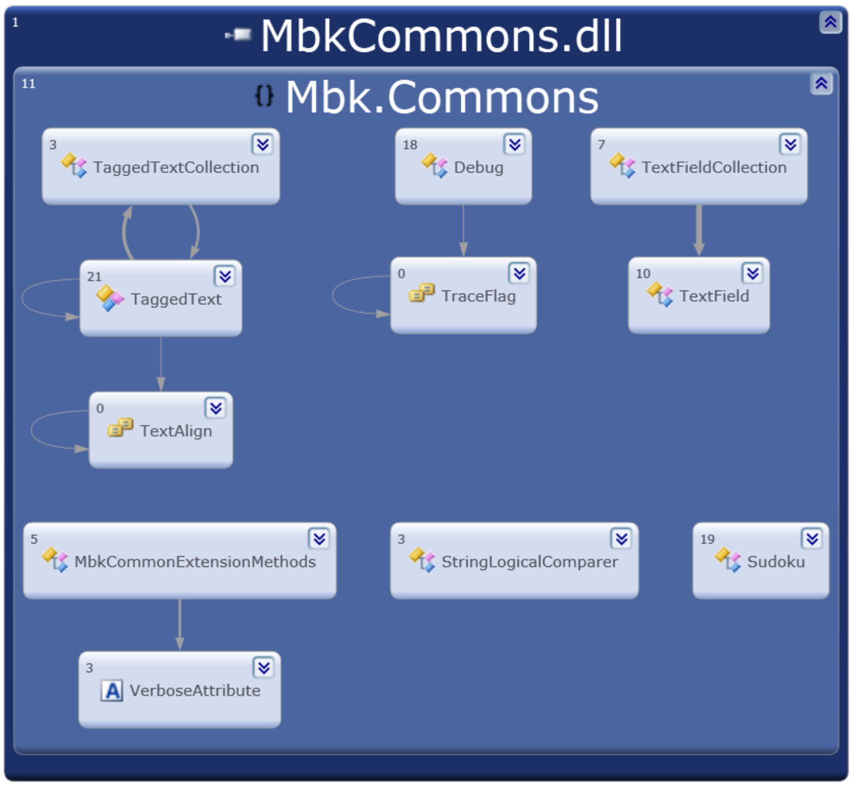
Video Rental Outlet GUI Application



TextUI Library

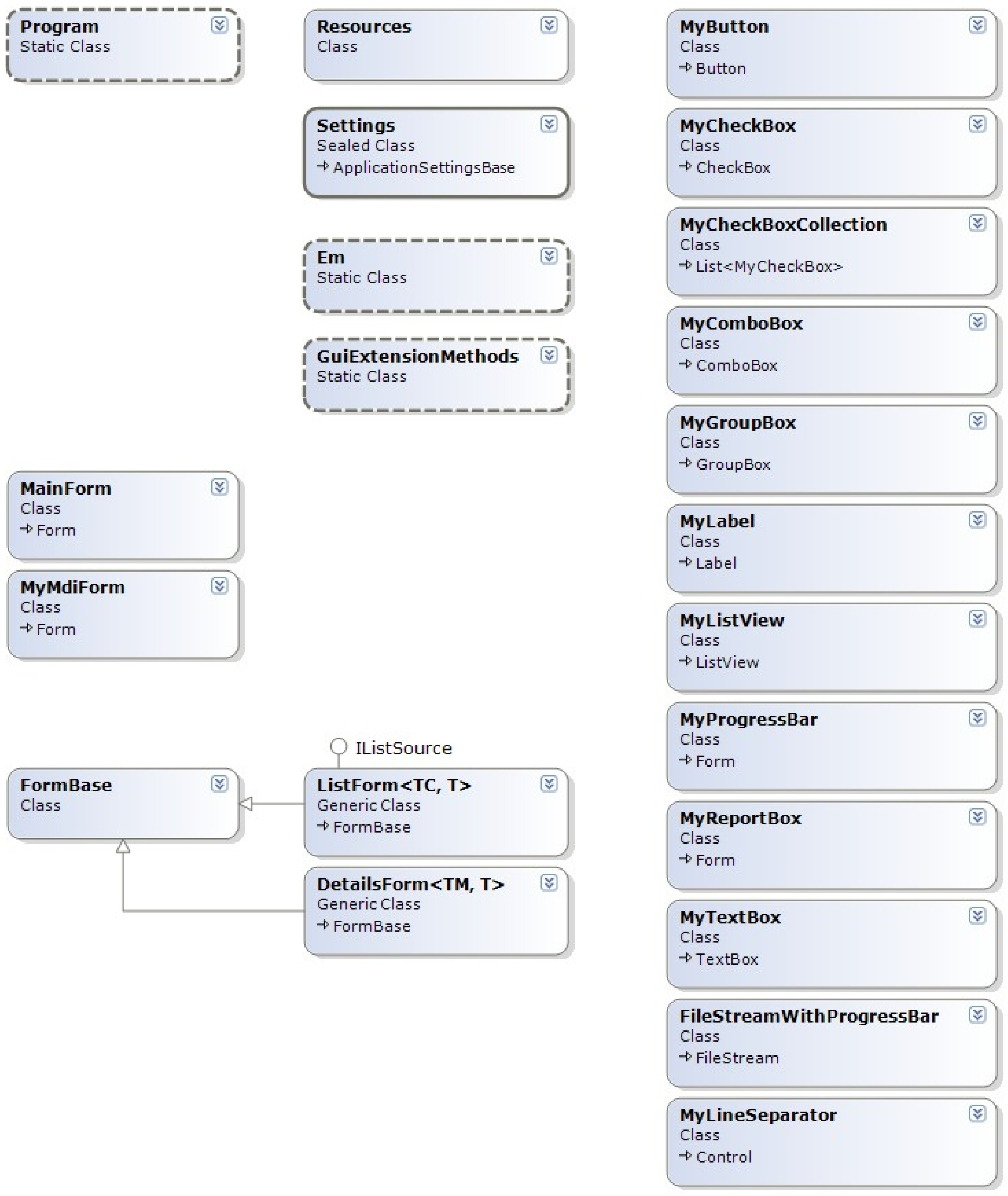


MbkCommons Library

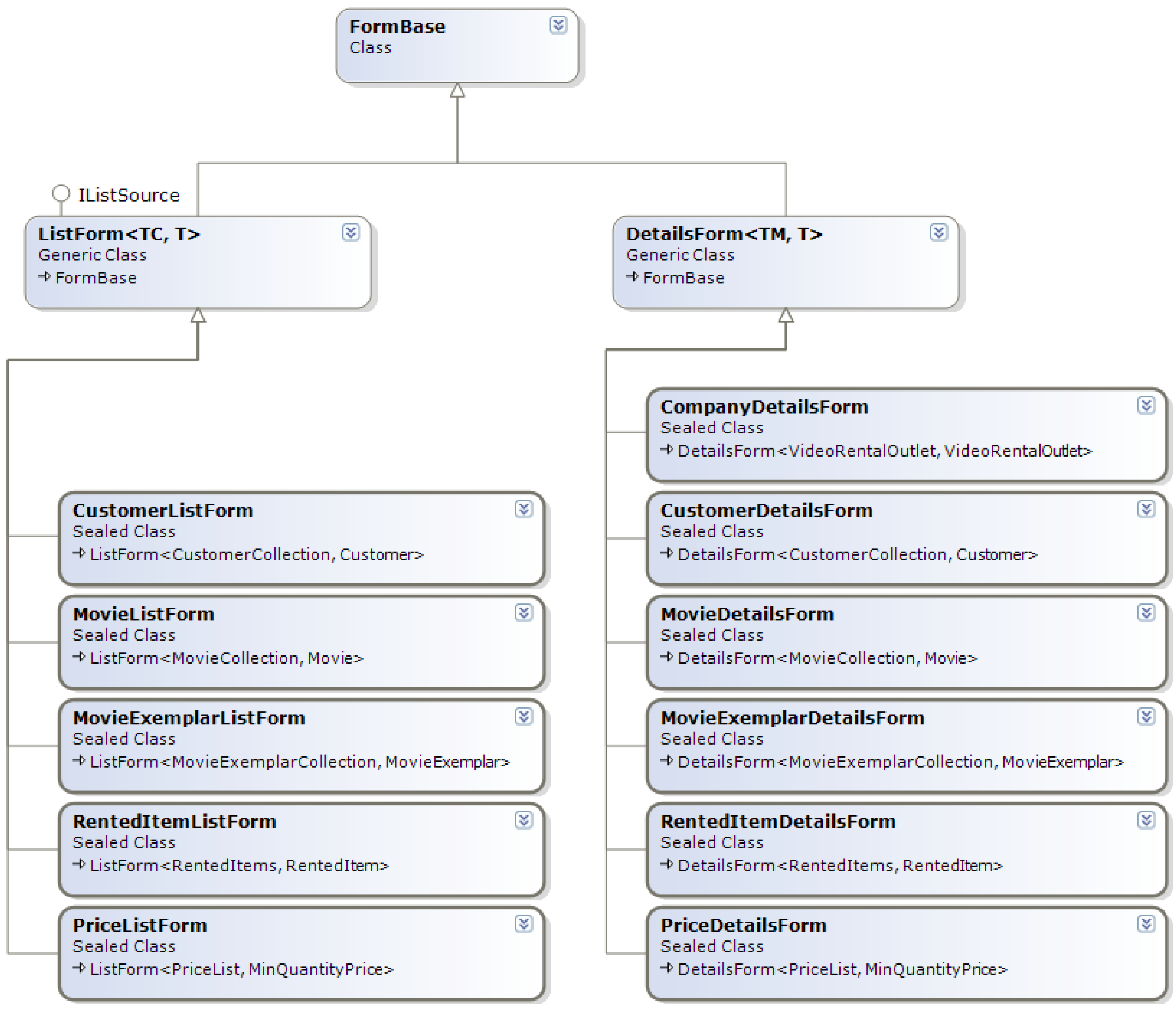


Class Diagrams

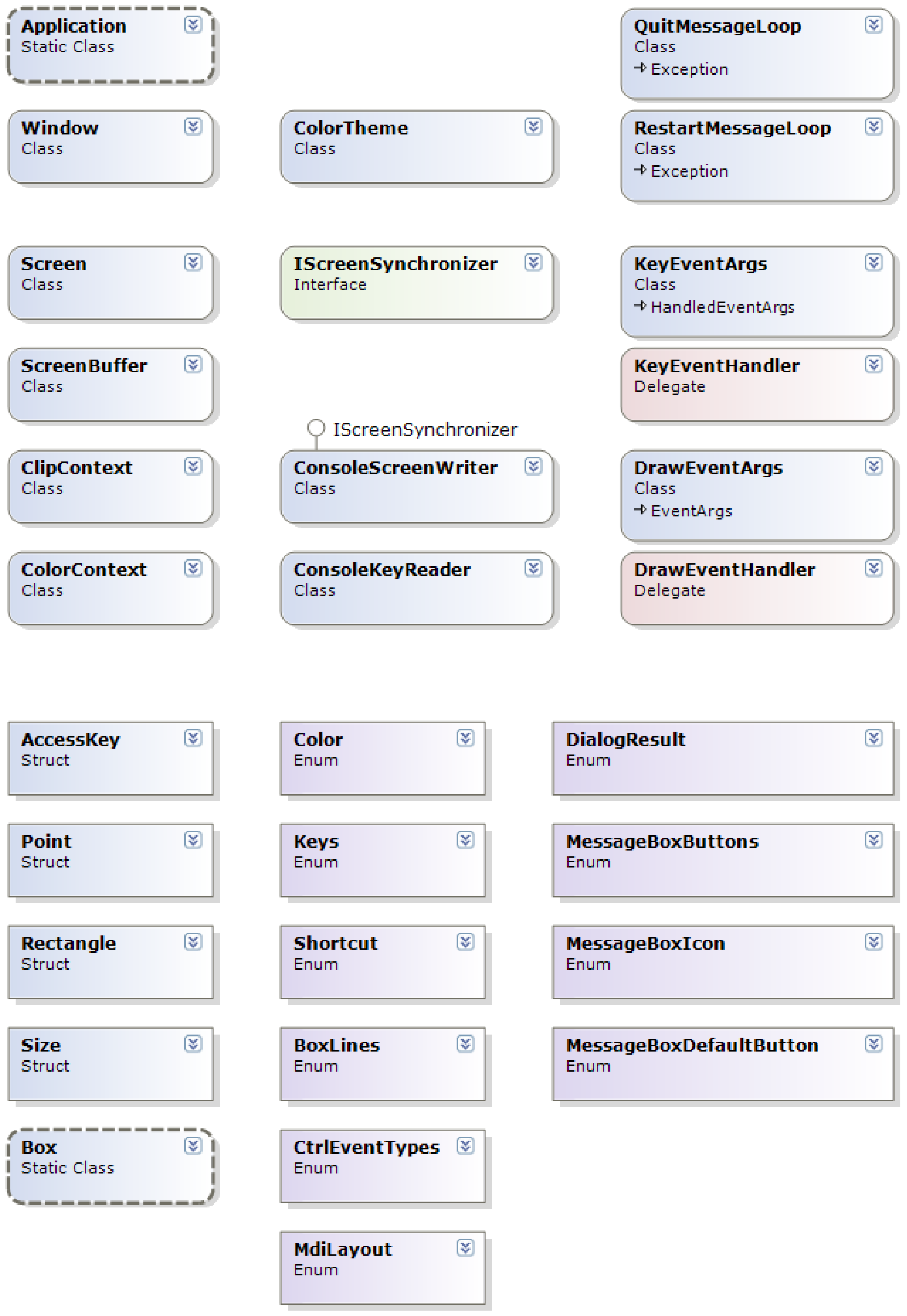
GUI/TUI Application Common Class Diagram – 1 of 2



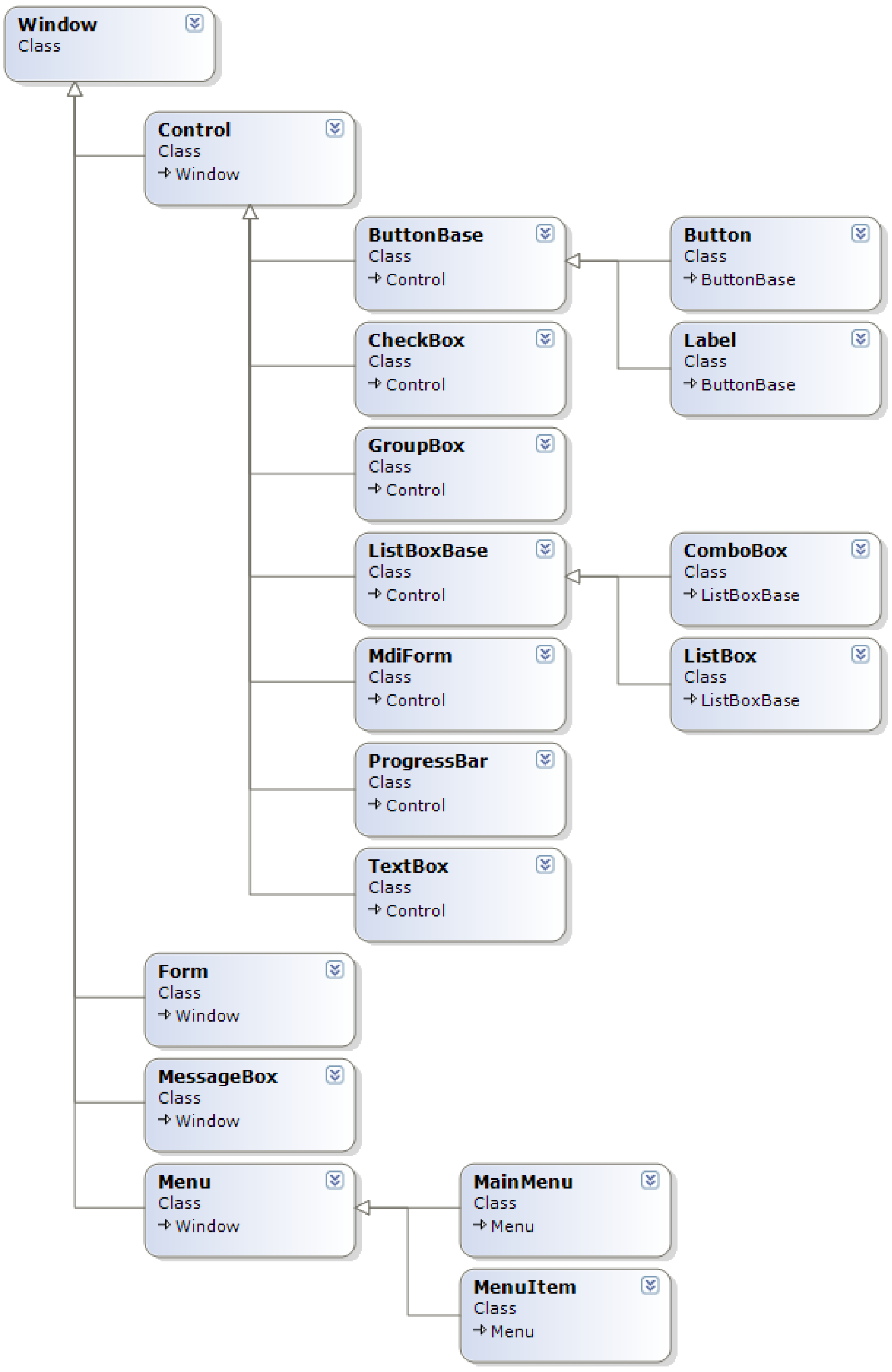
GUI/TUI Application Common Class Diagram – 2 of 2


FormBase
Class
IListSource
ListForm<TC, T>
Generic Class
FormBase
DetailsForm<TM, T>
Generic Class
FormBase
CompanyDetailsForm
Sealed Class
DetailsForm<VideoRentalOutlet, VideoRentalOutlet>
CustomerListForm
Sealed Class
ListForm<CustomerCollection, Customer>
CustomerDetailsForm
Sealed Class
DetailsForm<CustomerCollection, Customer>
MovieListForm
Sealed Class
ListForm<MovieCollection, Movie>
MovieDetailsForm
Sealed Class
DetailsForm<MovieCollection, Movie>
MovieExemplarListForm
Sealed Class
ListForm<MovieExemplarCollection, MovieExemplar>
MovieExemplarDetailsForm
Sealed Class
DetailsForm<MovieExemplarCollection, MovieExemplar>
RentedItemListForm
Sealed Class
ListForm<RentedItems, RentedItem>
RentedItemDetailsForm
Sealed Class
DetailsForm<RentedItems, RentedItem>
PriceListForm
Sealed Class
ListForm<PriceList, MinQuantityPrice>
PriceDetailsForm
Sealed Class
DetailsForm<PriceList, MinQuantityPrice>

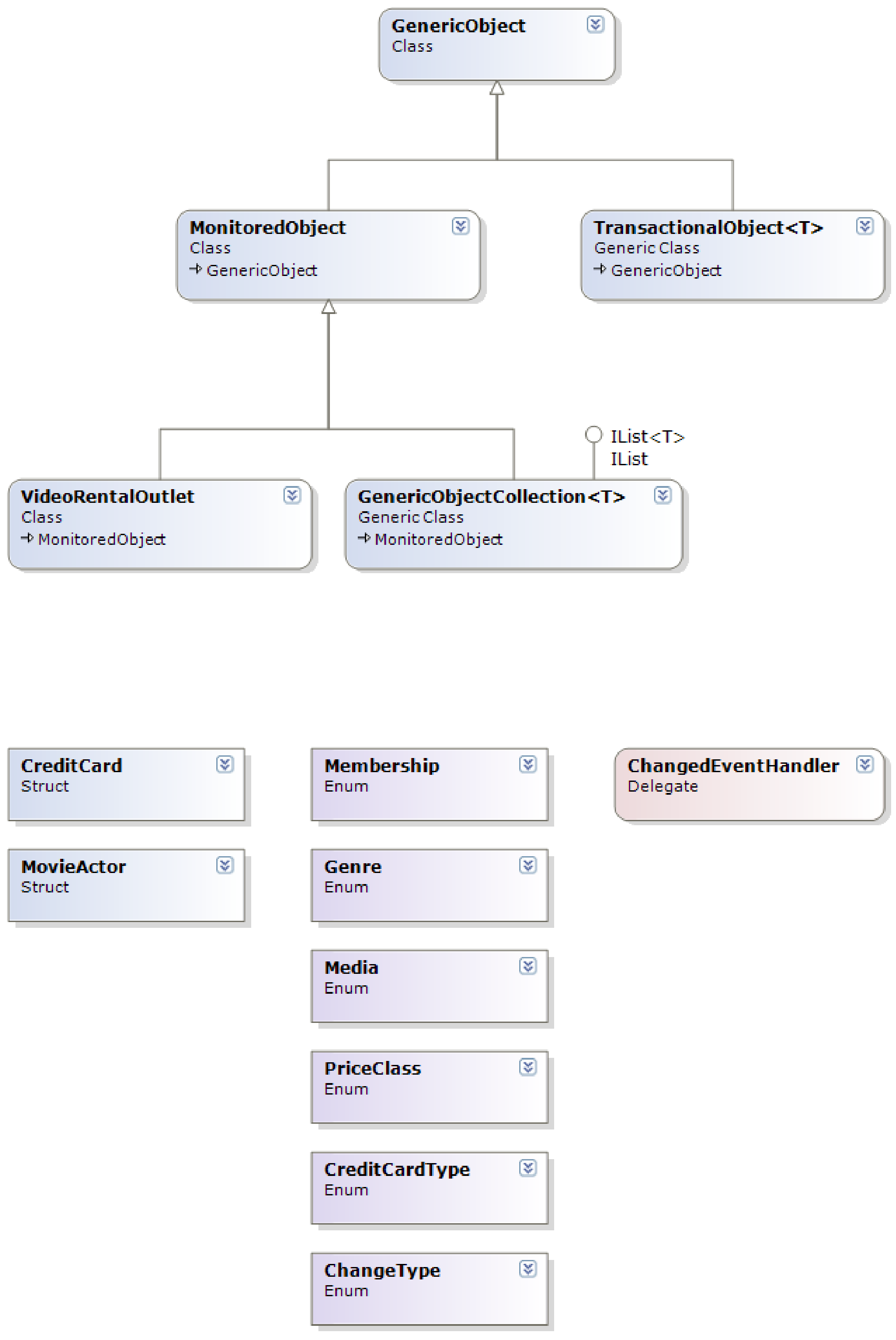
TextUI Library Class Diagram – 1 of 2



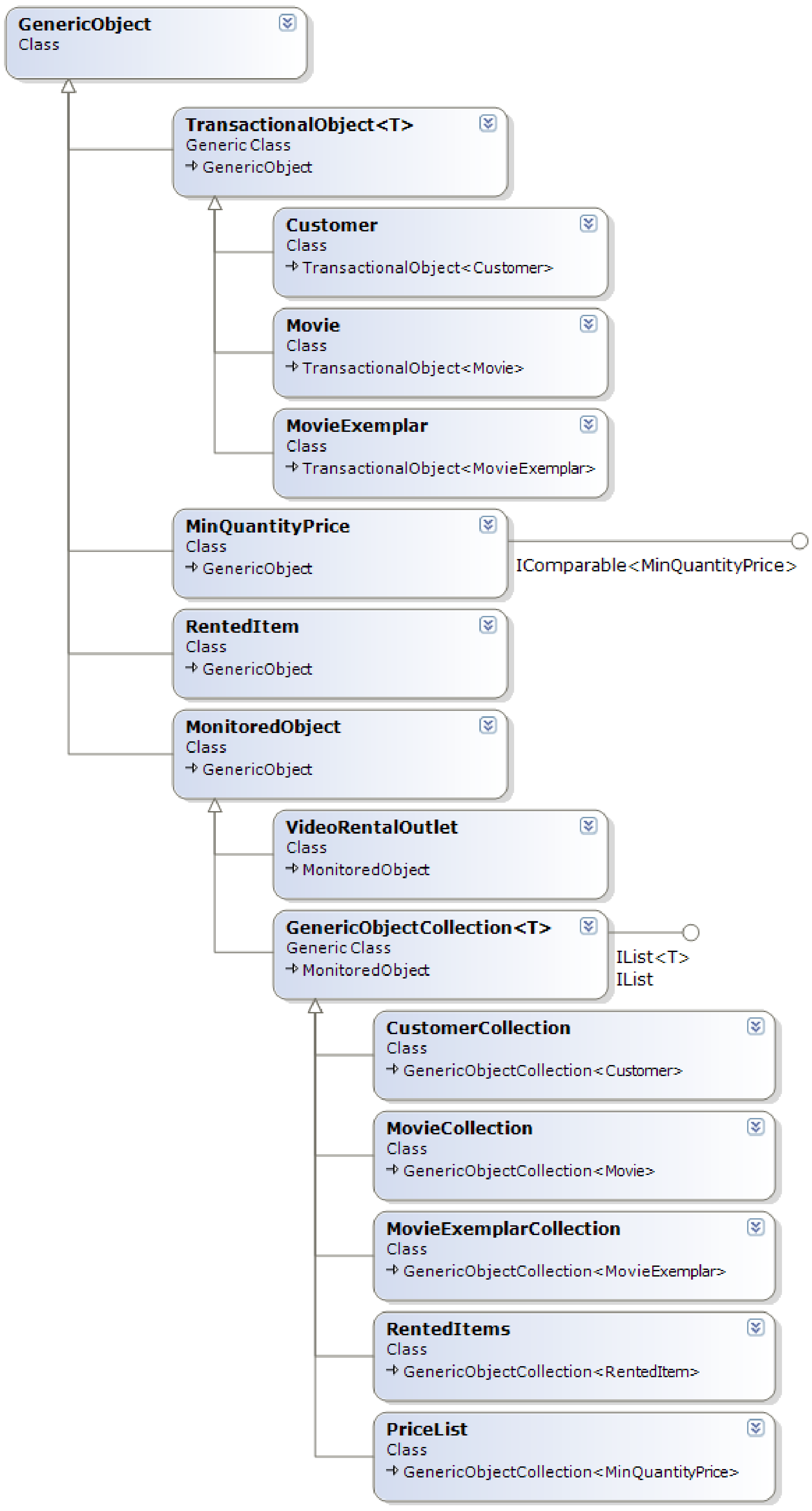
TextUI Library Class Diagram – 2 of 2



Video Rental Outlet Library Class Diagram – 1 of 2



Video Rental Outlet Library Class Diagram – 2 of 2



Meeting The Requirements…

4A I denna laboration ska du skriva ett textbaserat användargränssnitt för din videobutik.

The whole new Text User Interface (`TextUI` alt. `TUI`) library was created as the result of this requirement. The TextUI library is implemented on the top of the `Console` .NET class; however, the library can work on any media that provides `TextUI.IScreenSynchronizer` interface and the `KeyDown` event message pump.

In this implementation, the major message loop (a message pump reading keys from the `Console` and dispatching it to windows in focus) is implemented in the `ConsoleKeyReader` class, while the screen output is implemented in the `ConsoleScreenWriter` class.

The essential **`ConsoleKeyReader`** methods are ☞:

```
public void Run( Screen screen );
public DialogResult DoModal( Screen screen, Window modalWindow );
private void MessagePump( Screen screen, Window modalWindow = null );
private void ReadKey_Worker ();
```

Writing to a screen is implemented using double-buffering scheme where all screen output is written to the off-line screen buffer (represented as matrix of screen elements consisting of a character and the character attributes i.e. its foreground/background colors), which is synchronized with the Console, when needed.

The updating process in the **`ConsoleScreenWriter`** class is implemented, as a part of `IScreenSynchronizer` interface, in the methods ☞:

```
public void BeginUpdate ()
public void EndUpdate ()
public void Update ()
```

All TextUI controls are derived from the base **`Window`** class, which source code is divided into several source files ☞:

```
Window.cs
WindowDrawing.cs
WindowEvents.cs
WindowMethods.cs
WindowProperties.cs
WindowStructure.cs
```

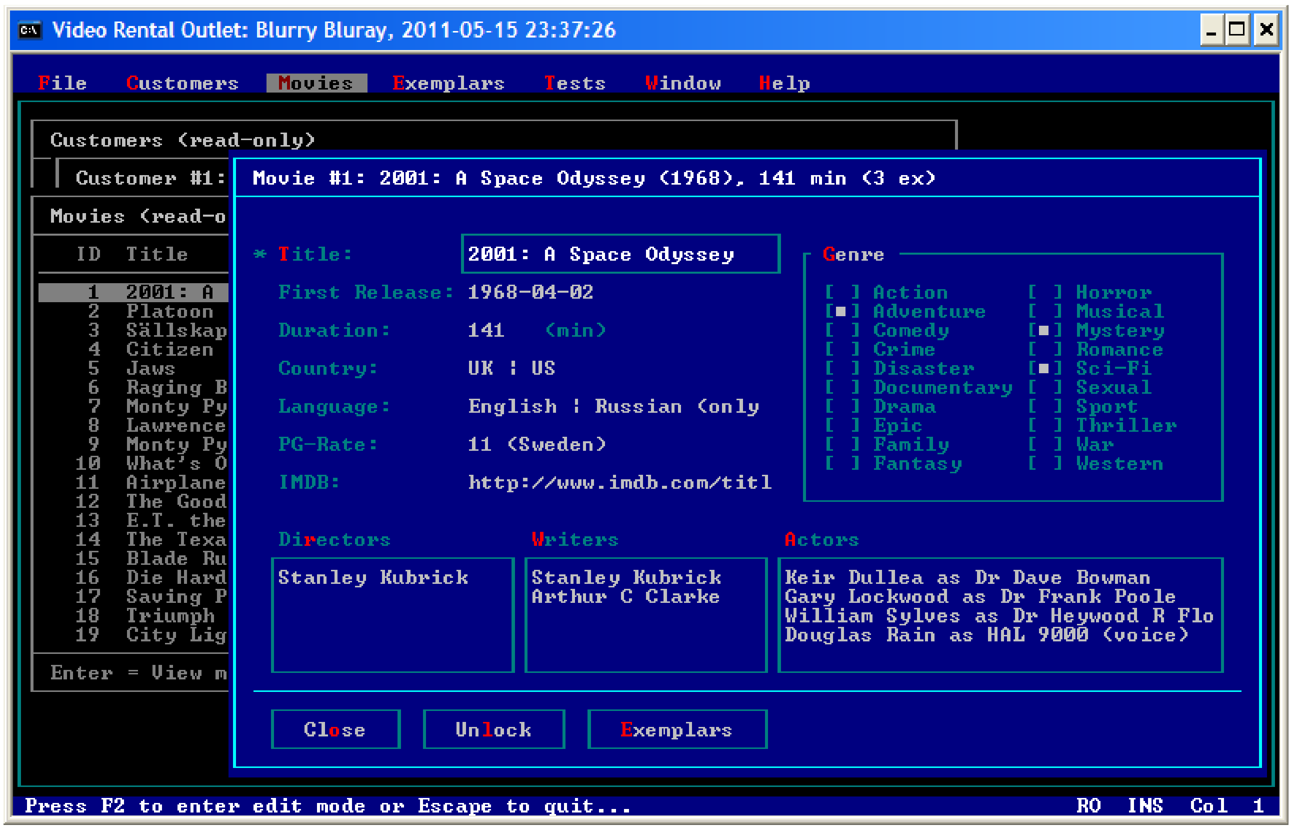
with the most complicated code is in the `WindowStructure.cs` that deals with parent/children relationship between controls and the focus navigation.

The TextUI controls used by the Video Rental Outlet console application are ☞:

```
Button                    ButtonBase                Menu
CheckBox                  ListBoxBase               MenuItem
ComboBox                                            MainMenu
GroupBox                  Form
Label                     MdiForm
ListBox
ProgressBar               MessageBox
TextBox
```

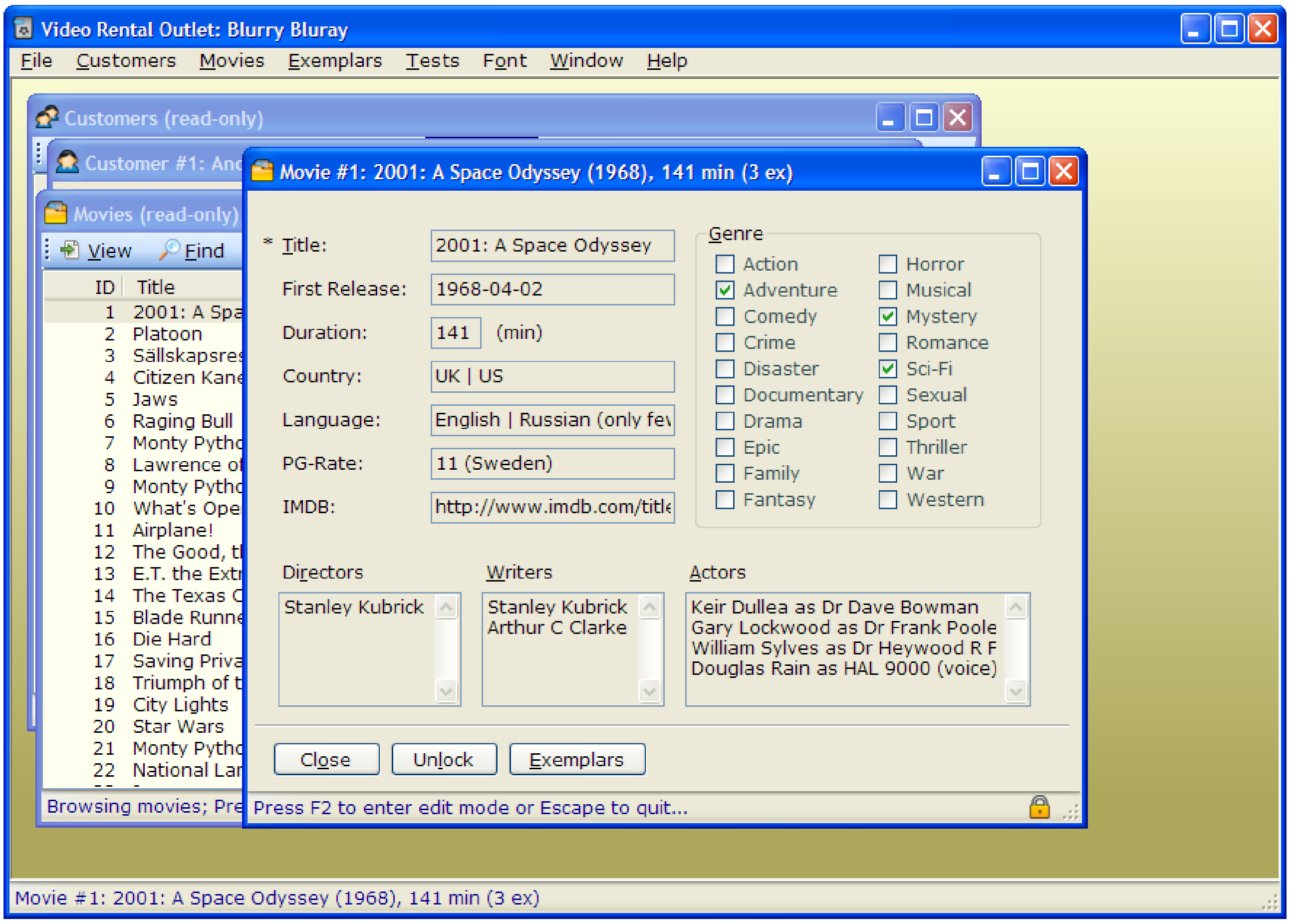
The following page shows the sample screenshot of some controls listed above, as displayed in the console application. The page also shows the windows forms application for comparison.

Movie Details Form in TextUI (console) application:



```
Video Rental Outlet: Blurry Bluray, 2011-05-15 23:37:26

 File   Customers   Movies   Exemplars   Tests   Window   Help

 Customers (read-only)
  Customer #1: Movie #1: 2001: A Space Odyssey (1968), 141 min (3 ex)
 Movies (read-o
  ID  Title      * Title:          2001: A Space Odyssey    Genre
                                                            [ ] Action        [ ] Horror
   1  2001: A    First Release:  1968-04-02                 [■] Adventure     [ ] Musical
   2  Platoon                                               [ ] Comedy        [■] Mystery
   3  Sällskap   Duration:       141   (min)                [ ] Crime         [ ] Romance
   4  Citizen                                               [ ] Disaster      [■] Sci-Fi
   5  Jaws       Country:        UK | US                    [ ] Documentary   [ ] Sexual
   6  Raging B                                              [ ] Drama         [ ] Sport
   7  Monty Py   Language:       English | Russian (only    [ ] Epic          [ ] Thriller
   8  Lawrence                                              [ ] Family        [ ] War
   9  Monty Py   PG-Rate:        11 (Sweden)                [ ] Fantasy       [ ] Western
  10  What's O
  11  Airplane   IMDB:           http://www.imdb.com/titl
  12  The Good
  13  E.T. the   Directors        Writers             Actors
  14  The Texa
  15  Blade Ru   Stanley Kubrick  Stanley Kubrick     Keir Dullea as Dr Dave Bowman
  16  Die Hard                    Arthur C Clarke     Gary Lockwood as Dr Frank Poole
  17  Saving P                                        William Sylves as Dr Heywood R Flo
  18  Triumph                                         Douglas Rain as HAL 9000 (voice)
  19  City Lig

 Enter = View m    Close    Unlock    Exemplars

Press F2 to enter edit mode or Escape to quit...                      RO  INS  Col  1
```

The same Movie Details Form in Windows Forms application:



```
* Title:          2001: A Space Odyssey        Genre
                                               [ ] Action        [ ] Horror
  First Release:  1968-04-02                   [✓] Adventure     [ ] Musical
                                               [ ] Comedy        [✓] Mystery
  Duration:       141    (min)                 [ ] Crime         [ ] Romance
                                               [ ] Disaster      [✓] Sci-Fi
  Country:        UK | US                      [ ] Documentary   [ ] Sexual
                                               [ ] Drama         [ ] Sport
  Language:       English | Russian (only fev  [ ] Epic          [ ] Thriller
                                               [ ] Family        [ ] War
  PG-Rate:        11 (Sweden)                  [ ] Fantasy       [ ] Western

  IMDB:           http://www.imdb.com/title

  Directors         Writers            Actors
  Stanley Kubrick   Stanley Kubrick    Keir Dullea as Dr Dave Bowman
                    Arthur C Clarke    Gary Lockwood as Dr Frank Poole
                                       William Sylves as Dr Heywood R F
                                       Douglas Rain as HAL 9000 (voice)
```

```
ID  Title
 1  2001: A Spa
 2  Platoon
 3  Sällskapsres
 4  Citizen Kane
 5  Jaws
 6  Raging Bull
 7  Monty Pytho
 8  Lawrence of
 9  Monty Pytho
10  What's Ope
11  Airplane!
12  The Good, t
13  E.T. the Extr
14  The Texas C
15  Blade Runne
16  Die Hard
17  Saving Priva
18  Triumph of t
19  City Lights
20  Star Wars
21  Monty Pytho
22  National Lar

Browsing movies; Pre   Press F2 to enter edit mode or Escape to quit...

Movie #1: 2001: A Space Odyssey (1968), 141 min (3 ex)
```

4A **k1** All information om kunder och om filmer ska sparas ner på fil när programmet avslutas. När programmet startas ska all information läsas in igen. Du får välja om du vill lagra information på en textfil eller en binärfil (spara ett helt objekt). Du kan själv välja om du vill spara all information i samma fil eller om du vill dela upp det på fler filer (t.ex. en fil med filmer och en med kunder).

Video Rental Outlet (VRO) database is serialized to a single binary file using the `BinaryFormatter` class. The `Serialize` and `Deserialize` method of the VideoRentalOutlet class are responsible for saving/loading a graph of VRO objects to/from file:

```
[Serializable] public class VideoRentalOutlet : MonitoredObject
{
    /// <summary>
    /// Saves (serializes) database in binary form to an instance of the file stream.
    /// </summary>
    ///
    public void Serialize( Stream fileStream )
    {
        new BinaryFormatter ().Serialize( fileStream, this );

        // Note that 'dirty' flags are not serialized so we can reset flags *after* we
        // have saved object to file (as it will appear 'clean' when reloaded).
        // If Serialize() throws an exception, the following code won't be executed,
        // so the video store would keep its dirty status unchanged.
        //
        IsDirty                = false;
        Customers.IsDirty      = false;
        PriceList.IsDirty      = false;
        Movies.IsDirty         = false;
        MovieExemplars.IsDirty = false;
    }

    /// <summary>
    /// Returns a new instance of VideoRentalOutlet class deserialized (loaded)
    /// from binary file stream.
    /// </summary>
    /// <returns>a new instance of VideoRentalOutlet database</returns>
    /// <remakrs>VideoRentalOutlet factory method</remakrs>
    ///
    public static VideoRentalOutlet Deserialize( Stream fileStream )
    {
        BinaryFormatter formatter = new BinaryFormatter ();

        // Deserialize Video Rental Outlet database from the file and
        // assign the reference to the local variable.
        //
        object deserializedObject = formatter.Deserialize( fileStream );
        VideoRentalOutlet store = deserializedObject as VideoRentalOutlet;

        if ( store != null && ! store.IsValidVersion )
        {
            string info = store.Version == null
                ? "of unkown version" : "v" + store.Version;

            throw new Exception( "Incompatible VRO database version!\r\n\r\n"
                + "File format is " + info
                + ". Application can handle only files in v"
                + VROLibVersion + " format." );
        }

        if ( store == null )
        {
            throw new Exception( "Failed to deserialize VRO database." );
        }

        return store;
    }
}
```

The user interface part, which automatically loads/saves database on application startup/shutdown, is implemented in the `MainFormDatabase.cs` file as a part of the `MainForm` class.

For more details see the `MainForm` class methods ☞:

```
    private void LoadDatabase( string filename, string caption, bool silent = false );
    private void SaveDatabase( string filename, string caption, bool silent = false );
```

The code snippet showing the sample usage of the methods listed above is:

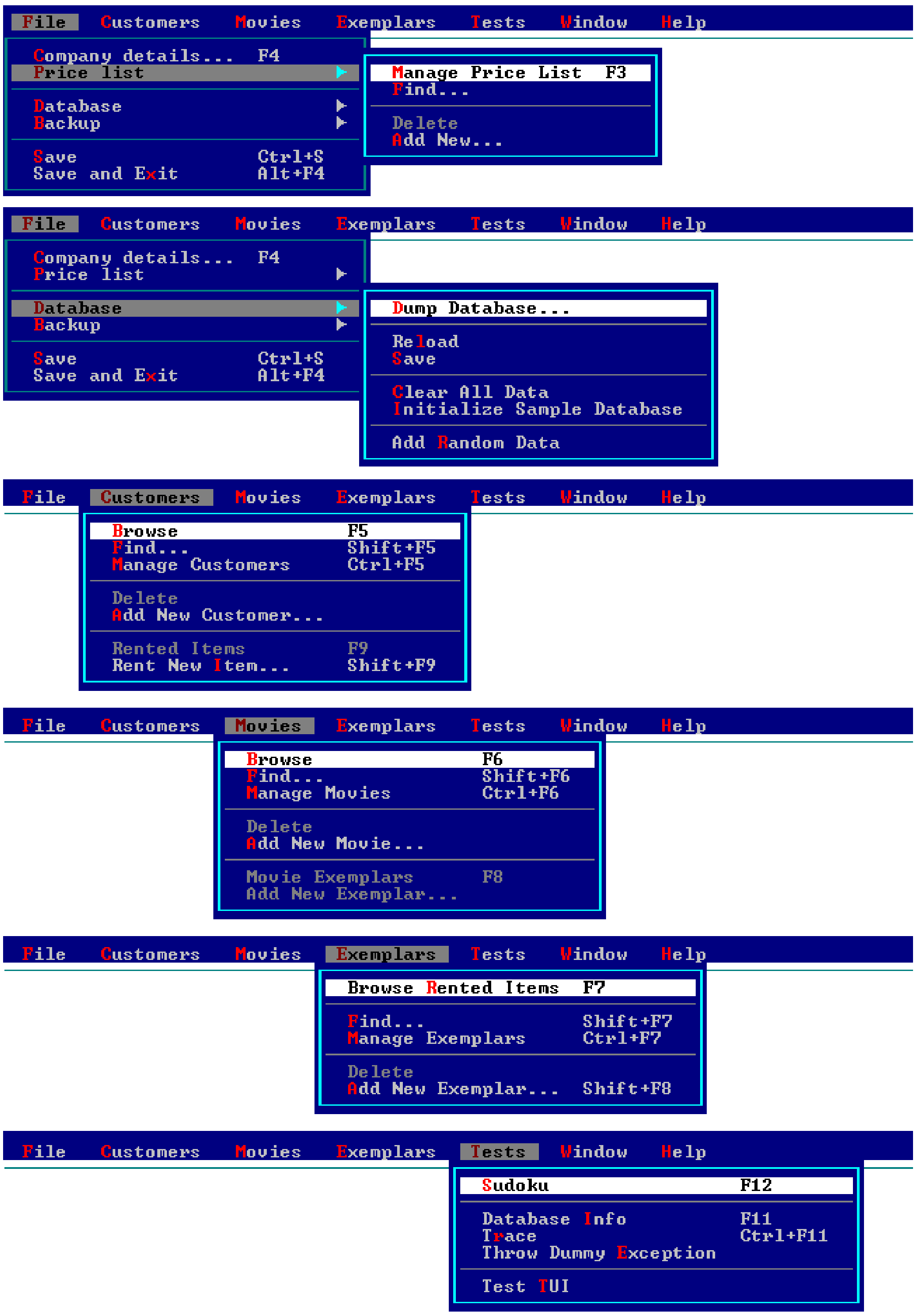
```
    /// <summary>
    /// Handles the Shown event. Loads database from file.
    /// </summary>
    ///
    private void MainForm_Shown( object sender, EventArgs e )
    {
        try
        {
            LoadDatabase( this.databaseFilename,
                "Video Rental Outlet Database", /*silent*/ true );
        }
        catch( Exception ex )
        {
            // As LoadDatabase method uses external DLL, we cannot catch
            // "could not load dll error" inside LoadDatabase the method, so we do it here.
            // Note that the LoadDatabase method handles itself its own exceptions.
            //
            MessageBox.Show( ex.Message, "Error Loading DLL",
                MessageBoxButtons.OK, MessageBoxIcon.Hand );
        }
    }

    /// <summary>
    /// Handles the FormClosing event. Asks user whether to save database, if database
    /// is dirty.
    /// </summary>
    ///
    private void MainForm_FormClosing( object sender, FormClosingEventArgs e )
    {
        if ( ! this.VideoStore.IsDirty )
        {
            return;
        }

        DialogResult rc = MessageBox.Show(
            "Database has been modified...\n\n"
                + "Do you want to save changes?",
            "Video Rental Oultet",
            MessageBoxButtons.YesNoCancel, MessageBoxIcon.Exclamation );

        if ( rc == DialogResult.Yes )
        {
            SaveDatabase( this.databaseFilename, "Saving Database", /*silent*/true );
        }
        else if ( rc == DialogResult.Cancel )
        {
            e.Cancel = true;
        }
    }
```

4A **k2**

Nedan beskrivs vilka menyval som ska vara *möjliga* i ditt användargränssnitt:
Visa alla kunder, Visa alla filmer, Lägg till ny kund, Lägg till ny film, Hyr ut en film, Lämna tillbaks en film, Avsluta

The menu system of the VRO application is coded in the file `MainFormMenus.cs` as a part of the `MainForm` class implementation.

The following screenshot displays some of the menu items in the TextUI application:



4B

Ditt program ska nu göras säkert. Om användaren skriver in något felaktigt värde, t.ex. skriver "hej" när ett menyalternativ efterfrågas, ska programmet ej krasha utan istället ge ett vettigt felmeddelande och låta användaren försöka igen. Detsamma gäller på alla ställen där användaren kan skriva in felaktiga värden.

För att uppgiften ska vara godkänd ska programmet ej krasha. Den du redovisar för kommer att försöka krasha programmet genom att skriva in konstiga saker, eller ta bort filen där data är sparat m.m.

Both GUI and TextUI applications provide exception handling and recovery by catching `AppDomain` exceptions. Besides that, all database violation exceptions (e.g. trying to insert Credit Card with an invalid credit number checksum) are catched by UI forms and displayed to the user.

The user interaction is based on two types of forms:

1) **`ListForm`** – listing a collection of items (e.g. of Customers, Movies etc)
2) **`DetailsForm`** – showing particular item in the collection (e.g. Customer, Movie ect)

(See the base class `FormBase`, and derived classes `ListForm` and `DetailsForm`.)

Validation of data entered by used is performed using `Validating` event handlers for particular fields (i.e. UI controls).

For example, in the `CustomerDetailsForm` class, the following code verifies that user has entered proper values as a part of the `InitializeComponents` method:

```
////////////////////////////////////////////////////////////////////////////////
// Field validation event handlers

this.firstName.Validating += ( sender, e ) =>
{
    MdiForm.ErrorMessage = null;

    if ( ReadOnly || ! this.firstName.ContentsChanged )
    {
        return;
    }

    ValidateNotNull( "Customer's first name", this.firstName.Text, e );
};

this.lastName.Validating += ( sender, e ) =>
{
    MdiForm.ErrorMessage = null;

    if ( ReadOnly || ! this.lastName.ContentsChanged )
    {
        return;
    }

    ValidateNotNull( "Customer's last name", this.lastName.Text, e );
};

this.personID.Validating += ( sender, e ) =>
{
    MdiForm.ErrorMessage = null;

    if ( ReadOnly || ! this.personID.ContentsChanged )
    {
        return;
    }

    ValidateNotNull( "Customer's Person-ID", this.personID.Text, e );
```

```
        if ( ! ReadOnly && ! e.Cancel )
        {
            string notValidInfo = Customer.ValidatePNR( this.personID.TrimmedText );

            if ( notValidInfo != null )
            {
                MdiForm.ErrorMessage = notValidInfo;
                MdiForm.Beep ();
                e.Cancel = true;
            }
        }
    };

    this.email.Validating += ( sender, e ) =>
    {
        MdiForm.ErrorMessage = null;

        if ( ReadOnly || ! this.email.ContentsChanged )
        {
            return;
        }

        ValidateEMailAddress( "E-Mail", this.email.Text, e );
    };
```

The helper methods `Validate(something)` are provided in the `DetailsForm` class, e.g.:

```
    /// <summary>
    /// Validates text value not to be null or empty.
    /// </summary>
    ///
    public void ValidateNotNull( string fieldName, string value, CancelEventArgs e )
    {
        if ( e.Cancel )
        {
            return; // Already cancelled
        }

        if ( ReadOnly )
        {
            return; // Assume always data valid while browsing
        }

        value = value == null ? null : value.Trim ();

        if ( string.IsNullOrEmpty( value ) )
        {
            MdiForm.ErrorMessage = fieldName + " must not be empty or null.";
            MdiForm.Beep ();
            e.Cancel = true;
        }
    }

    /// <summary>
    /// Validates text value to be a valid HTTP or HTTPS Universal Resource Identifier
    /// (URI).
    /// </summary>
    ///
    public void ValidateHttpURI( string fieldName, string value, CancelEventArgs e )
    {
        if ( e.Cancel )
        {
            return; // Already cancelled
        }

        if ( ReadOnly )
        {
            return; // Assume always data valid while browsing
        }

        value = value == null ? null : value.Trim ();
        // Accept NULL values always as valid. Not null must be validated separatelly.
        //
        if ( string.IsNullOrEmpty( value ) )
        {
            return;
        }

        // Parse URI and accept only URIs having scheme either http or https
        //
        try
        {
            string scheme = new Uri( value.Trim () ).Scheme.ToLower ();
            if ( scheme != "http" && scheme != "https" )
            {
                throw new Exception( fieldName + " must be either http or https URI." );
            }
        }
        catch( Exception ex )
        {
            MdiForm.ErrorMessage = ex.Message;
            MdiForm.Beep ();
            e.Cancel = true;
        }
    }

    ...
```

4C Du ska skapa en Windowsapplikation (grafiskt gränssnitt) för din redan befintliga videobutik. Den enda klassen du bör behöva ändra på är själva startklassen som hanterar hela videobutiken. Det grafiska gränssnittet ska istället ta över dess roll.

As mentioned earlier, TextUI and GUI has the same source base and the `MainForm` class is common both for TextUI and GUI mode.

4C **k1** Koden för det grafiska gränssnittet får **inte** vara genererad av Visual Studio .Net - du ska ha skrivit den helt själv.

The common UI is based on two types of forms:

1) `ListForm` – listing a collection of items (e.g. of Customers, Movies etc)
2) `DetailsForm` – showing particular item in the collection (e.g. Customer, Movie ect)

(See the base class `FormBase`, and derived classes `ListForm` and `DetailsForm`.)

The UI components placement is guided by units defined in the `Em` class, that specifies dimensions in pixels of the typical character in current font in GUI mode. In TextUI mode, the units specified in the `Em` class equals "one" so the forms could be layout (almost) independently from the character/graphical mode and the used font (in GUI mode).

Usage of the `New`*`SomeControl`* factory methods of the `DetailsForm` class makes a transparent creation of the UI components both for TextUI and GUI modes. The common UI components initialization, e.g. for the `CompanyDetailsForm` class (derived from the `DetailsForm` class), looks like:

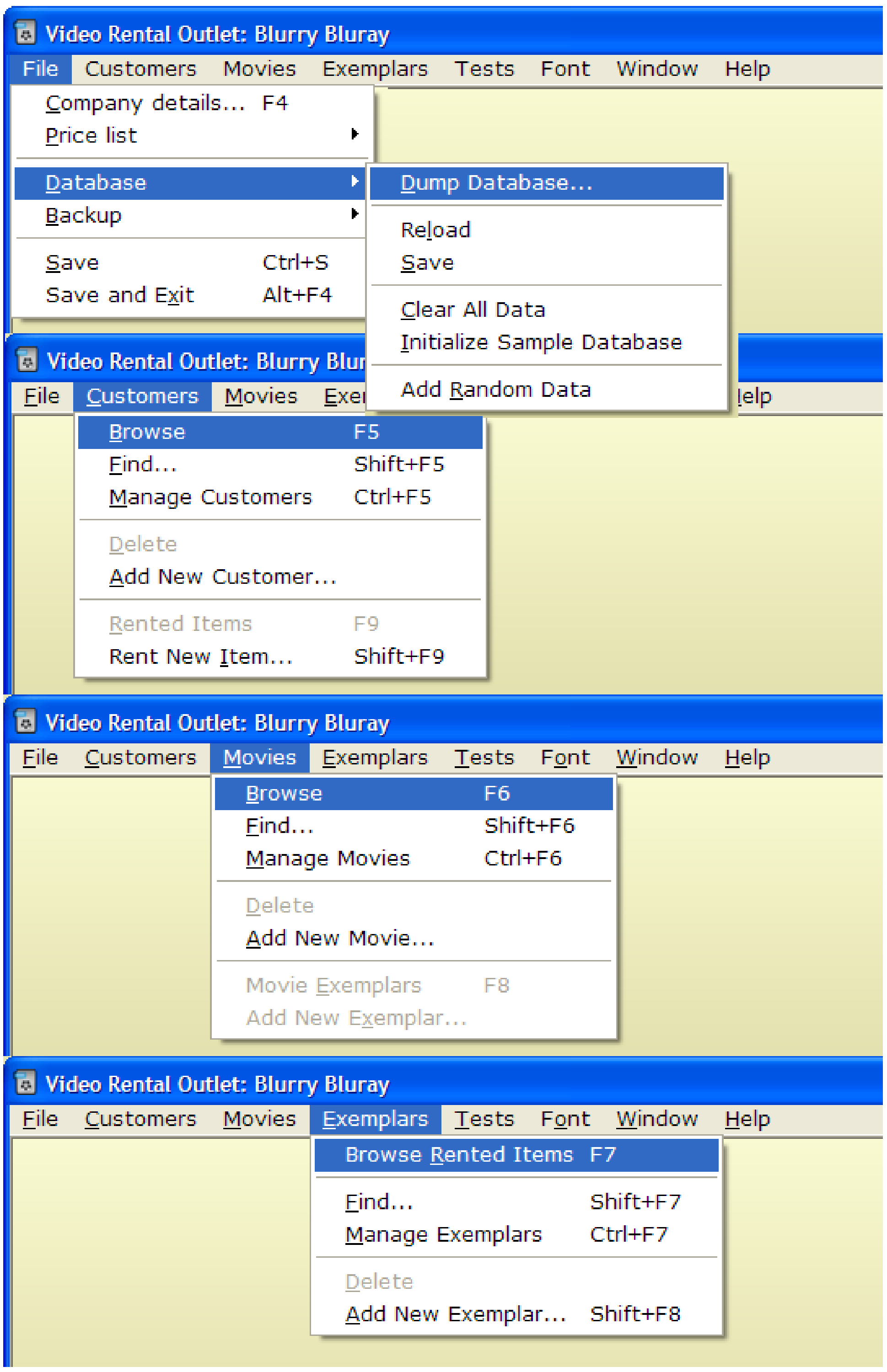
```
protected override void InitializeComponents ()
{
    ////////////////////////////////////////////////////////////////////////////////
    // Table layout grid definition, with rows and columns in Em units.

    float[] col = { 4, 22 };
    float[] row = { 2, 4, 6, 8, 10, 12, 14, 16, 18, 21, 23 };

    // Maximum text length
    float maxLen = (float)MdiForm.Width / Em.Width - col[1] - 2f;

    ////////////////////////////////////////////////////////////////////////////////
    // Static text

    int r = 0, c = 0;

    NewStaticText( col[c], row[r++], "Company Name:"    );
    NewStaticText( col[c], row[r++], "VAT Number:"      );
    NewStaticText( col[c], row[r++], "Address:"         );
    NewStaticText( col[c], row[r++], "Post Code:"       );
    NewStaticText( col[c], row[r++], "City:"            );
    NewStaticText( col[c], row[r++], "Country:"         );
    NewStaticText( col[c], row[r++], "Phone:"           );
    NewStaticText( col[c], row[r++], "Home Page:"       );
    NewStaticText( col[c], row[r++], "E-Mail Address: " );

    // Mandatory fields marker:
    NewStaticText( col[c]-2, row[0], "*" );
    NewStaticText( col[c]-2, row[1], "*" );
    NewStaticText( col[c]-2, row[2], "*" );

    ////////////////////////////////////////////////////////////////////////////////
    // TextBox fields

    r = 0; c = 1;
```

```
        this.companyName = NewTextField( col[c], row[r++], maxLen );
        this.vatNo       = NewTextField( col[c], row[r++], maxLen );
        this.address     = NewTextField( col[c], row[r++], maxLen );
        this.postCode    = NewTextField( col[c], row[r++], maxLen );
        this.city        = NewTextField( col[c], row[r++], maxLen );
        this.country     = NewTextField( col[c], row[r++], maxLen );
        this.phone       = NewTextField( col[c], row[r++], maxLen );
        this.homePage    = NewTextField( col[c], row[r++], maxLen );
        this.email       = NewTextField( col[c], row[r++], maxLen );

        ////////////////////////////////////////////////////////////////////////////////
        // Field validation event handlers

        this.companyName.Validating  += ( sender, e ) =>
        {
            MdiForm.ErrorMessage = null;

            if ( ReadOnly || ! this.companyName.ContentsChanged )
            {
                return;
            }

            ValidateNotNull( "Company Name", this.companyName.Text, e );
        };

        this.vatNo.Validating += ( sender, e ) =>
        {
            MdiForm.ErrorMessage = null;

            if ( ReadOnly || ! this.vatNo.ContentsChanged )
            {
                return;
            }

            ValidateNotNull( "Company's VAT Number", this.vatNo.Text, e );
        };

        this.homePage.Validating += ( sender, e ) =>
        {
            MdiForm.ErrorMessage = null;

            if ( ReadOnly || ! this.homePage.ContentsChanged )
            {
                return;
            }

            ValidateHttpURI( "Home Page", this.homePage.Text, e );
        };

        this.email.Validating += ( sender, e ) =>
        {
            MdiForm.ErrorMessage = null;

            if ( ReadOnly || ! this.email.ContentsChanged )
            {
                return;
            }

            ValidateEMailAddress( "E-Mail", this.email.Text, e );
        };

        ////////////////////////////////////////////////////////////////////////////////

        base.InitializeComponents ();
    }
```

4C **k2** Textmenyn du tidigare hade gav användaren möjlighet att välja en siffra. Nu ska det istället finnas knappar eller grafiska menyer för att välja de olika valen.

The application mnu system is coded in the file `MainFormMenus.cs` as a part of the `MainForm` class implementation. The menu system is identical for TextUI and GUI.

The following screenshot displays essential GUI menus, which may be compared to the screenshot shown under the requirement 4a.k2.



4C
k3

Man ska i någon ruta kunna se informationen om videofilmer/kunder. Om den inte uppdateras automatiskt när något ändras bör en knapp för uppdatering finnas.

There are five classes displaying VRO collections (derived from the `ListForm` class) and six classes displaying item details (derived from the `DetailsForm` class):

List Forms:

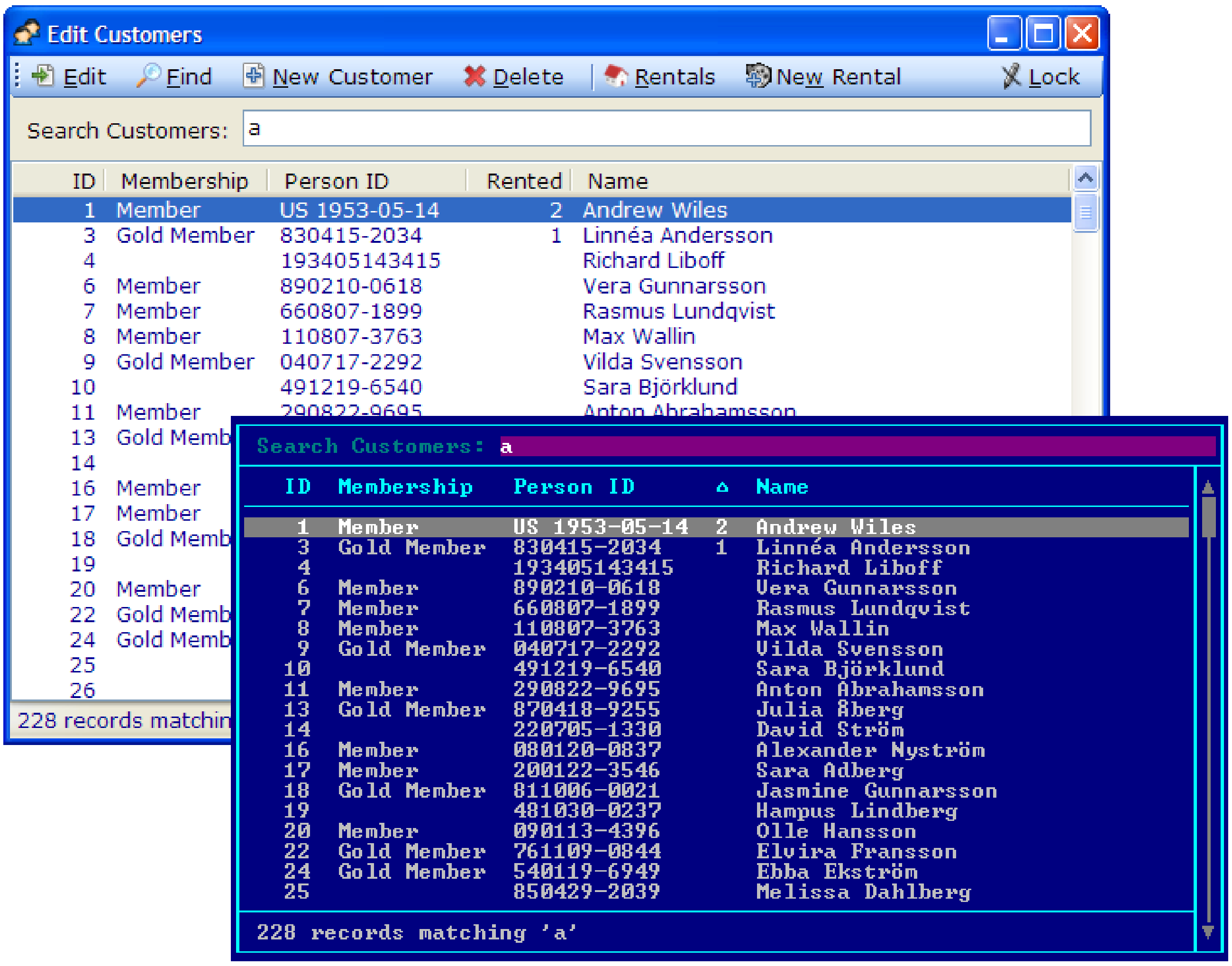
- `CustomerListForm`
- `MovieExemplarListForm`
- `MovieListForm`
- `PriceListForm`
- `RentedItemListForm`

Details Forms:

- `CompanyDetailsForm`
- `CustomerDetailsForm`
- `MovieDetailsForm`
- `MovieExemplarDetailsForm`
- `PriceDetailsForm`
- `RentedItemDetailsForm`

The List Forms interface is unified having the common toolbar, context menu and the search panel. All forms are updated automatically (changes on data in one form are immediately reflected in other forms).

The following image displays the same `CustomersListForm` class in GUI and TextUI applications:



Search Customers: a

ID	Membership	Person ID	Rented	Name
1	Member	US 1953-05-14	2	Andrew Wiles
3	Gold Member	830415-2034	1	Linnéa Andersson
4		193405143415		Richard Liboff
6	Member	890210-0618		Vera Gunnarsson
7	Member	660807-1899		Rasmus Lundqvist
8	Member	110807-3763		Max Wallin
9	Gold Member	040717-2292		Vilda Svensson
10		491219-6540		Sara Björklund
11	Member	290822-9695		Anton Abrahamsson
13	Gold Memb			
14				
16	Member			
17	Member			
18	Gold Memb			
19				
20	Member			
22	Gold Memb			
24	Gold Memb			
25				
26				

228 records matchin

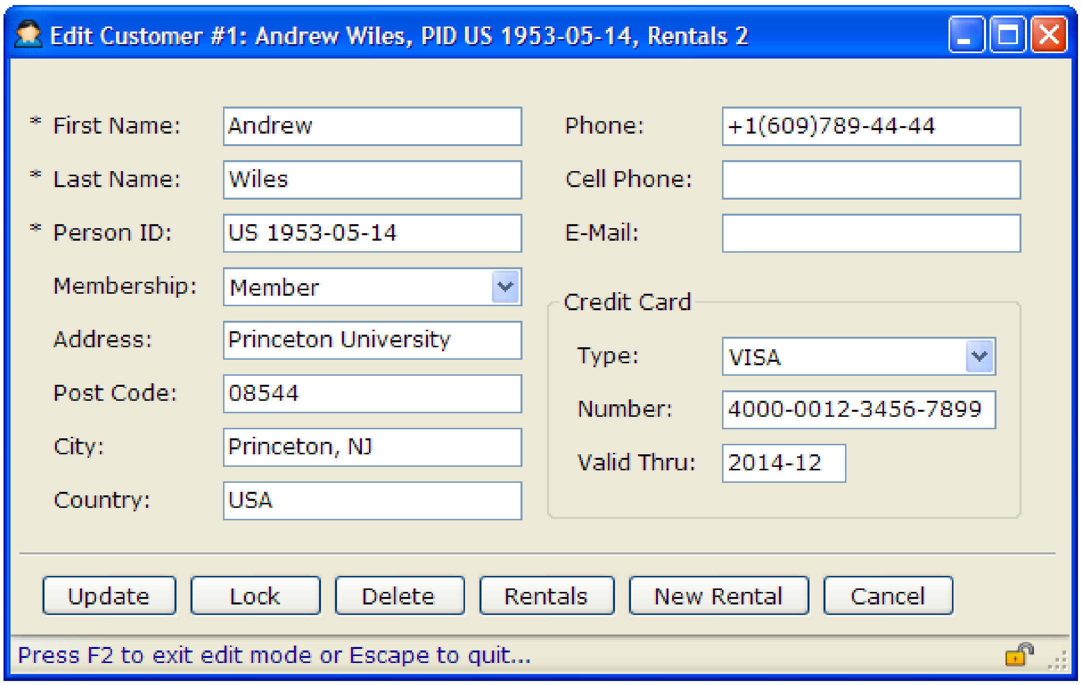
```
Search Customers: a

 ID  Membership    Person ID      ▵  Name

  1  Member        US 1953-05-14  2  Andrew Wiles
  3  Gold Member   830415-2034    1  Linnéa Andersson
  4                193405143415      Richard Liboff
  6  Member        890210-0618       Vera Gunnarsson
  7  Member        660807-1899       Rasmus Lundqvist
  8  Member        110807-3763       Max Wallin
  9  Gold Member   040717-2292       Vilda Svensson
 10                491219-6540       Sara Björklund
 11  Member        290822-9695       Anton Abrahamsson
 13  Gold Member   870418-9255       Julia Åberg
 14                220705-1330       David Ström
 16  Member        080120-0837       Alexander Nyström
 17  Member        200122-3546       Sara Adberg
 18  Gold Member   811006-0021       Jasmine Gunnarsson
 19                481030-0237       Hampus Lindberg
 20  Member        090113-4396       Olle Hansson
 22  Gold Member   761109-0844       Elvira Fransson
 24  Gold Member   540119-6949       Ebba Ekström
 25                850429-2039       Melissa Dahlberg

228 records matching 'a'
```

4C **k4** Om någon information ska skrivas in (t.ex. en ny videofilm) bör detta ske i textfält. För att skicka iväg det är det lämpligt med en knapp.

The application uses the following controls (common both to TextUI and Windows Forms):

- **`TextBox`**, **`ComboBox`** and **`CheckBox`** – for user input
- **`Label`** and **`GroupBox`** – for labeling and grouping
- **`Button`** – for user actions

Customer Details Form in the windows forms VRO application:



Edit Customer #1: Andrew Wiles, PID US 1953-05-14, Rentals 2

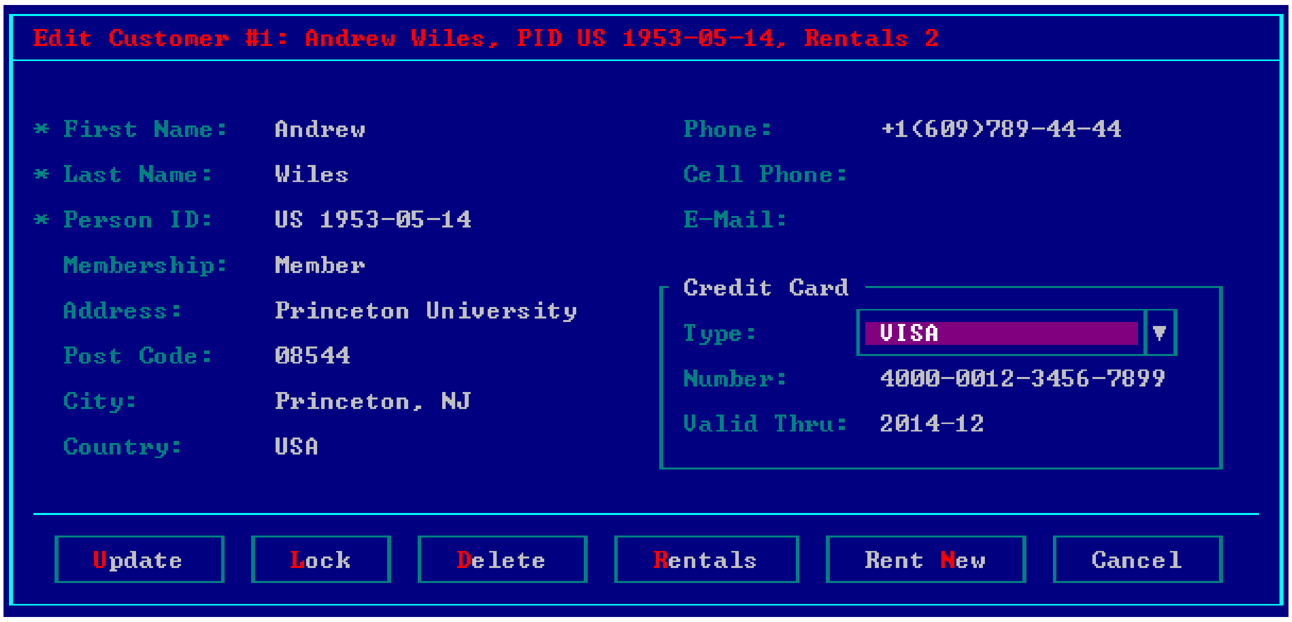
* First Name:	Andrew	Phone:	+1(609)789-44-44
* Last Name:	Wiles	Cell Phone:	
* Person ID:	US 1953-05-14	E-Mail:	
Membership:	Member		
Address:	Princeton University		
Post Code:	08544		
City:	Princeton, NJ		
Country:	USA		

Credit Card

Type:	VISA
Number:	4000-0012-3456-7899
Valid Thru:	2014-12

Press F2 to exit edit mode or Escape to quit...

The same **Customer Details Form** in the console VRO application:



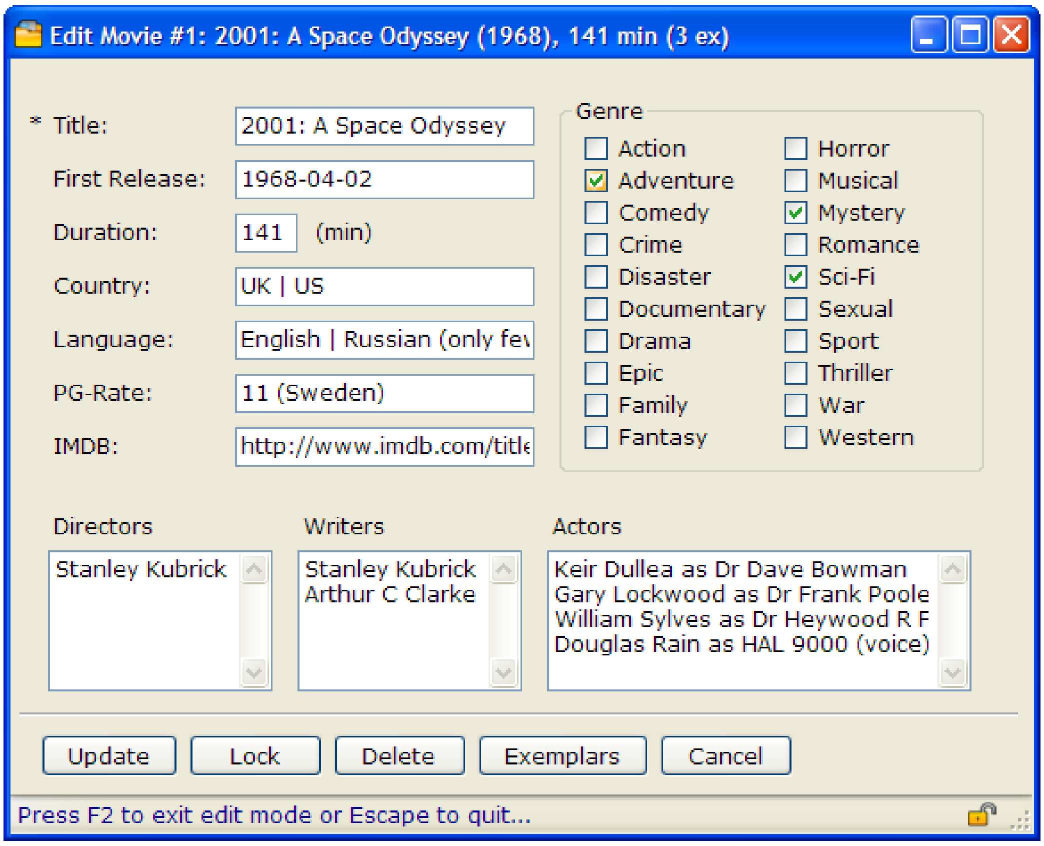
Edit Customer #1: Andrew Wiles, PID US 1953-05-14, Rentals 2

* First Name:	Andrew	Phone:	+1(609)789-44-44
* Last Name:	Wiles	Cell Phone:	
* Person ID:	US 1953-05-14	E-Mail:	
Membership:	Member		
Address:	Princeton University		
Post Code:	08544		
City:	Princeton, NJ		
Country:	USA		

Credit Card

Type:	VISA
Number:	4000-0012-3456-7899
Valid Thru:	2014-12

Movie Details Form in the windows forms VRO application:



Edit Movie #1: 2001: A Space Odyssey (1968), 141 min (3 ex)

* Title:	2001: A Space Odyssey
First Release:	1968-04-02
Duration:	141 (min)
Country:	UK | US
Language:	English | Russian (only fev
PG-Rate:	11 (Sweden)
IMDB:	http://www.imdb.com/title

Genre

☐ Action	☐ Horror
☑ Adventure	☐ Musical
☐ Comedy	☑ Mystery
☐ Crime	☐ Romance
☐ Disaster	☑ Sci-Fi
☐ Documentary	☐ Sexual
☐ Drama	☐ Sport
☐ Epic	☐ Thriller
☐ Family	☐ War
☐ Fantasy	☐ Western

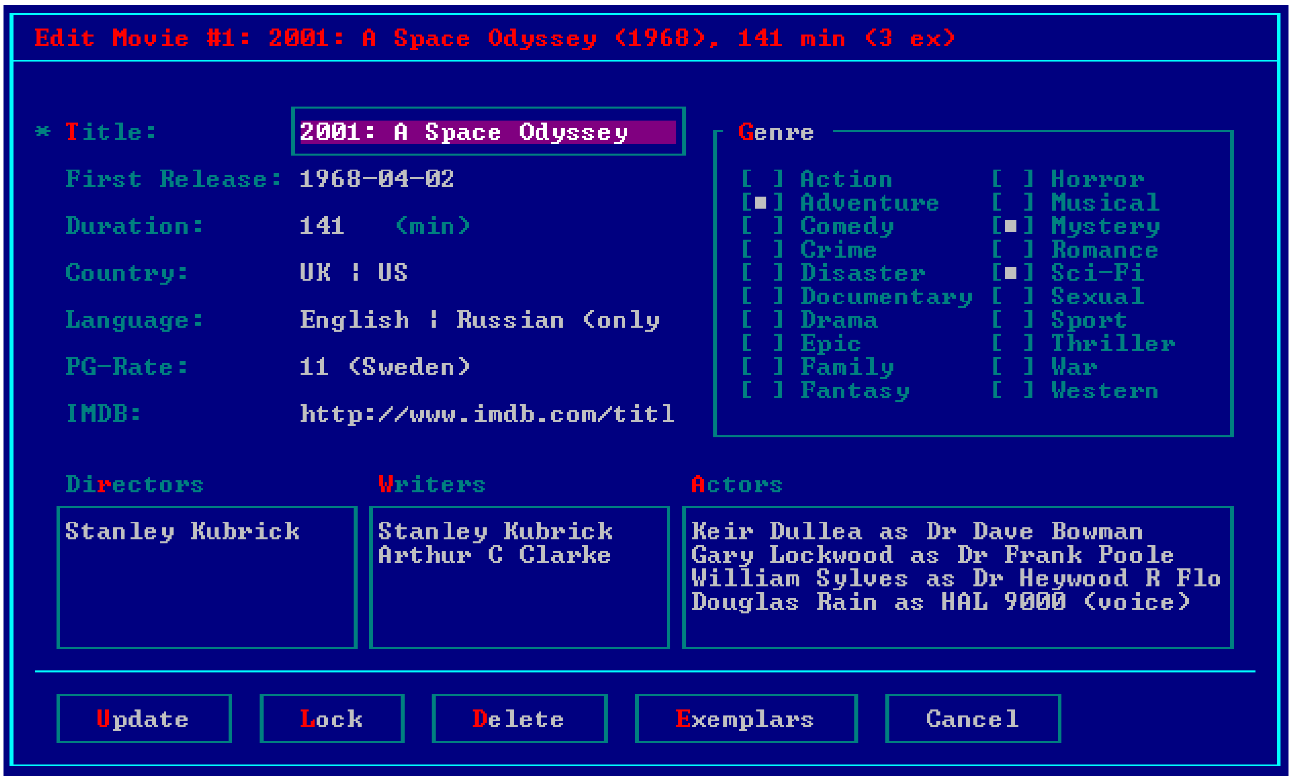
Directors
Stanley Kubrick

Writers
Stanley Kubrick
Arthur C Clarke

Actors
Keir Dullea as Dr Dave Bowman
Gary Lockwood as Dr Frank Poole
William Sylves as Dr Heywood R F
Douglas Rain as HAL 9000 (voice)

Press F2 to exit edit mode or Escape to quit...

The same **Movie Details Form** in the console VRO application:



Edit Movie #1: 2001: A Space Odyssey (1968), 141 min (3 ex)

* Title:	2001: A Space Odyssey
First Release:	1968-04-02
Duration:	141 (min)
Country:	UK ¦ US
Language:	English ¦ Russian (only
PG-Rate:	11 (Sweden)
IMDB:	http://www.imdb.com/titl

Genre

```
[ ] Action        [ ] Horror
[■] Adventure     [ ] Musical
[ ] Comedy        [■] Mystery
[ ] Crime         [ ] Romance
[ ] Disaster      [■] Sci-Fi
[ ] Documentary   [ ] Sexual
[ ] Drama         [ ] Sport
[ ] Epic          [ ] Thriller
[ ] Family        [ ] War
[ ] Fantasy       [ ] Western
```

Directors
Stanley Kubrick

Writers
Stanley Kubrick
Arthur C Clarke

Actors
Keir Dullea as Dr Dave Bowman
Gary Lockwood as Dr Frank Poole
William Sylves as Dr Heywood R Flo
Douglas Rain as HAL 9000 (voice)