

Abbildung 3: OR-Diagramm

Die Entität *Restrictions* kann benutzt werden, um bestimmte Darbietungskategorien einschränken zu können. Zum Beispiel kann eine mögliche Kategorie für Kinder so eingeschränkt, dass die Aufführungen für Kinder nur während einer bestimmten Zeitspanne vorgetragen werden können. Auch Feueraufführungen sollen nur zwischen 21 und 23 Uhr stattfinden, da zu diesem Zeitpunkt das Tageslicht nicht mehr so stark ist.

Die Entität *Artist* besitzt ein Attribut der Typ *Bool* *deleted*, der gesetzt wird, falls ein Künstler gelöscht werden soll. Damit wird der Künstler als gelöscht markiert, wird aber nicht aus der Datenbank gelöscht.

1.2.2 Datenzugriffsschicht

Um eine möglich gute Abstraktion zu erreichen, wurde für jede Domainklasse eine *Dao* Interface angelegt. Die Daten aus der Datenbank können nur über die Methoden, die im jeweilige Interface deklariert wurden, zugegriffen werden.

Um die Daten besser zwischen unterschiedlichen Schichten zu transportieren, wurde für jede Entität eine eigene *Domain*-Klasse implementiert.

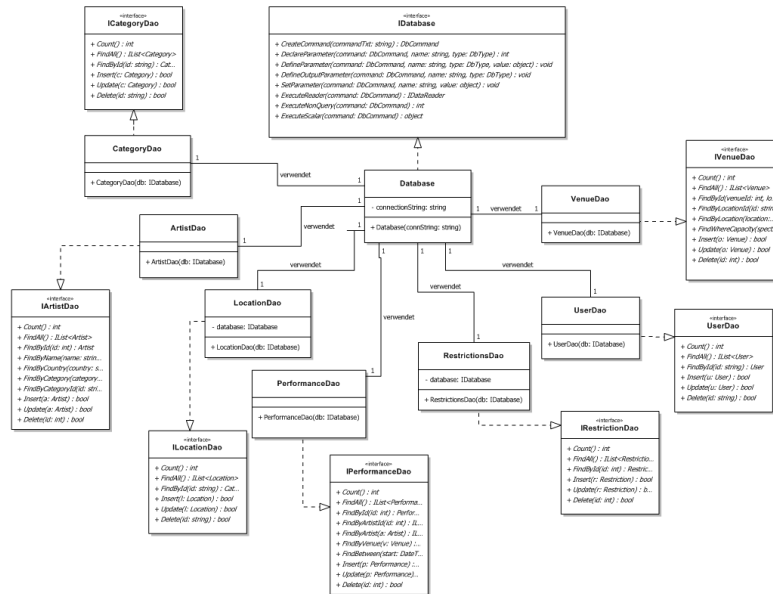


Abbildung 4: Data Access

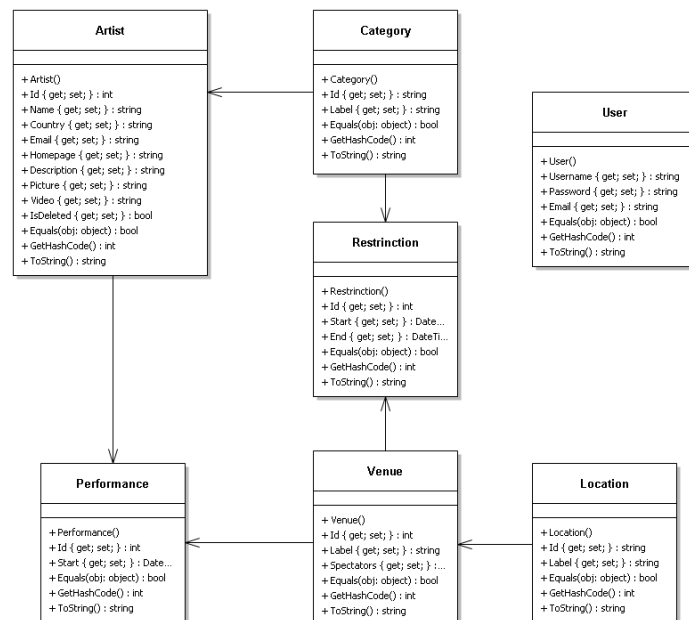


Abbildung 5: Domain Classes

1.2.3 Geschäftslogik

Die Geschäftslogik ist die einzige Komponente, die mit die Datenzugriffsschicht kommuniziert und stellt die Daten für die Präsentationsschicht zur Verfügung. Dafür werden zwei Schnittstellen implementiert:

Die erste Schnittstelle bietet alle Operationen die von UFO Commander verwendet werden. Diese Operationen können verwendet um Benutzer, Kategorien, Künstler, Spielstätten, Aufführungen zu verwalten. Es können neue Künstler oder Spielstätte angelegt, neue Kategorien definiert oder Aufführungen von Künstler editieren. Die Präsentationsschicht für den UFO Commander ist nur von diese Schnittstelle abhängig. Die Klasse `ManagerImpl` implementiert diese Schnittstelle. Um die Geschäftslogik zu verwendet wird eine Factory-Methode zur Verfügung gestellt. Diese Methode erstellt ein Business Logic-Objekt.

Die zweite Schnittstelle stellt alle Informationen zur Verfügung, die auf einem Client angezeigt werden. Das heißt, diese Schnittstelle bietet nur read-only Operationen. Ein Client kann somit nur Daten anzeigen und keine Daten bearbeiten. Er kann ein Programmübersicht anzeigen oder Informationen von Künstler, Spielstätten und Aufführungen abfragen. Die Klasse `ViewerImpl` implementiert diese Schnittstelle.

1.2.4 Web-Service

Die letzte wichtige Systemkomponente des Servers ist die Web-Service-Komponente. Diese wird mit ASP.NET als SOAP-basiertes Web-Service implementiert. Der Web-Service implementiert die Funktionalität die notwendig ist um den UFO-Client zu realisieren. Web-Service beinhaltet Operationen zum Abfragen von Aufführungen, Spielstätten oder Künstler und stellt auch Login-Operationen zur Verfügung, damit sich registrierte Organisatoren am Server anmelden können, um das Programm zu ändern.

1.3 Commander

Die Präsentationsschicht für den UFO Commander wird im WPF nach der Model-View-ViewModel-Pattern (MVVM) realisiert. Als Model werden die Domainklassen verwendet. Der ViewModel wird verwendet, um die Sicht unabhängig von die Domainklassen zu implementieren und greift auf die Schnittstelle der Geschäftslogik, um Daten von Server abfragen zu können. Dafür wird den Factory-Pattern verwendet. Der Server stellt eine Factory-Methode zur Verfügung, welche ein Geschäftslogik-Objekt returniert.

1.3.1 Benutzerhandbuch

Beim starten der Applikation wird der Benutzer aufgefordert, sich mit Benutzername und Kennword anzumelden. Im Datenbank wurde ein Benutzer mit der Name *swk5* und Kennword *swk5* angelegt. Nach der Anmeldung, wird als Startbild der Applikation die Programmübersicht des ersten Tages angezeigt. Hier kann der Benutzer das Programm editieren, indem er eine Kachel anklickt.

Die Applikation öffnet ein Editierfenster und der Benutzer kann hier aus eine Liste mit alle Künstler ein Künstler auswählen und der Auswahl mit save speichern. Falls der Künstler zu diesem Zeitpunkt, zu keinem anderen Auftritt zugeteilt ist und er zumindest eine Stunden Pause gehabt hat, wird der Künstler gespeichert, ansonsten wird dem Benutzer mitgeteilt, dass diesen Künstler nicht ausgewählt werden kann.

Location	16:00	17:00	18:00	19:00	20:00	21:00	22:00
Alte Stadt							
Klosterstraße 7	Capuzi Senzitta Argentinien K	Berri Green Italien K	San Deadly Großbritannien M	Urban Safari Niederlande K	Hoozlepp Deutschland A	The LEDies Deutschland T	Kornel Deutschland A
Landhaus	El Diablero Österreich K	Les Contes Asphalt Belgien K	Capuzi Senzitta Argentinien M	Berri Green Italien K	San Deadly Großbritannien M	Urban Safari Niederlande K	Hoozlepp Deutschland A
Altstadt 13	Hunderpfund Deutschland K	Lutrek Status Polen K	El Diablero Österreich K	Les Contes Asphalt Belgien K	Capuzi Senzitta Argentinien M	Berri Green Italien K	
Altes Markt	Kornel Deutschland A	OAKLEAF Stadtkunst Deutschland F	Hunderpfund Deutschland K	Lutrek Status Polen K	El Diablero Österreich K	Les Contes Asphalt Belgien K	Capuzi Senzitta Argentinien M
Hofgasse 13	Hoozlepp Deutschland A	The LEDies Deutschland T	Kornel Deutschland A	OAKLEAF Stadtkunst Deutschland F	Hunderpfund Deutschland K	Lutrek Status Polen K	El Diablero Österreich K
Schlossmuseum	San Deadly Großbritannien M	Urban Safari Niederlande K	Hoozlepp Deutschland A	The LEDies Deutschland T	Kornel Deutschland A	OAKLEAF Stadtkunst Deutschland F	Hunderpfund Deutschland K
Hauptplatz							
Dreifaltigkeitskirche	Sigismund Österreich A	El Otro Spanien A	Jean Philippe Kikolas Spanien K	Matheus Circus Portugal A	La Trocola Circo Spanien A	Lusa Reflejos Italien K	Los Galindos Spanien M
Mader Böden	cicciPanga Schweiz K	Frutix Con Crema Chile K	Sigismund Österreich A	El Otro Spanien A	Jean Philippe Kikolas Spanien K	Matheus Circus Portugal A	La Trocola Circo Spanien A
Hallenstele	Compagnie Antipodes Frankreich A	Passabernet Spanien K	cicciPanga Schweiz K	Frutix Con Crema Chile K	Sigismund Österreich A	El Otro Spanien A	
Altes Rathaus	Los Galindos Spanien M	Fausto Gori Italien K	Compagnie Antipodes Frankreich A	Passabernet Spanien K	cicciPanga Schweiz K	Frutix Con Crema Chile K	Sigismund Österreich A
Fa. Mammut	La Trocola Circo Spanien A	Lusa Reflejos Italien K	Los Galindos Spanien M	Fausto Gori Italien K	Compagnie Antipodes Frankreich A	Passabernet Spanien K	cicciPanga Schweiz K
Bank Austria	Jean Philippe Kikolas Spanien K	Matheus Circus Portugal A	La Trocola Circo Spanien A	Lusa Reflejos Italien K	Los Galindos Spanien M	Fausto Gori Italien K	Compagnie Antipodes Frankreich A
Landstraße							
Annagasse	Kornel Deutschland A	San Deadly Großbritannien M	OAKLEAF Stadtkunst Deutschland F	Grande Compagnie Savatini Italien K	Shag Hoozlepp Italien K	Sigismund Österreich A	Sigismund Österreich A

Abbildung 6: Startbild

Performance

Artist: The LEDies

Venue: Hofgasse 13

Starts at:

2016-07-22

17:00

Cancel
Remove
Save

Abbildung 7: Aufführung editieren.

Im Reiter *Artists* wird auf die Linke Seite des Fensters eine Liste mit alle im System vorhandene Künstler. Wird ein Künstler ausgewählt, so wird auf die Rechte Seite des Fensters aller Informationen zu dem ausgewählten Künstler angezeigt. Diese Informationen können hier auch editiert und gespeichert werden, oder neue Künstler angelegt.

Im Reiter *Venues* wird auf die Linke Seite des Fensters eine Liste mit alle im System vorhandene Spielstätten. Wird eine Spielstätte ausgewählt, so wird auf die Rechte Seite des Fensters aller Informationen zu der ausgewählte Spielstätte angezeigt. Diese Informationen können hier auch editiert und gespeichert werden, oder neue Spielstätten angelegt.

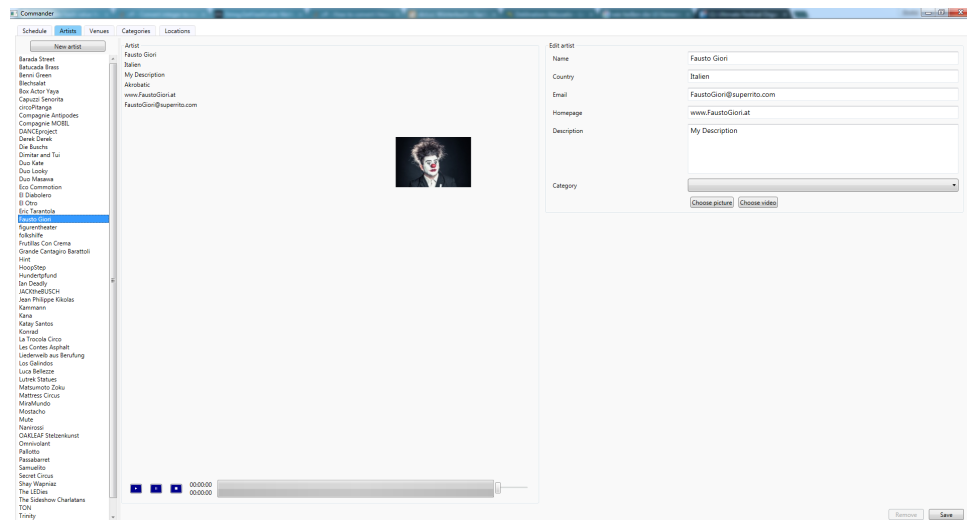


Abbildung 8: Startbild

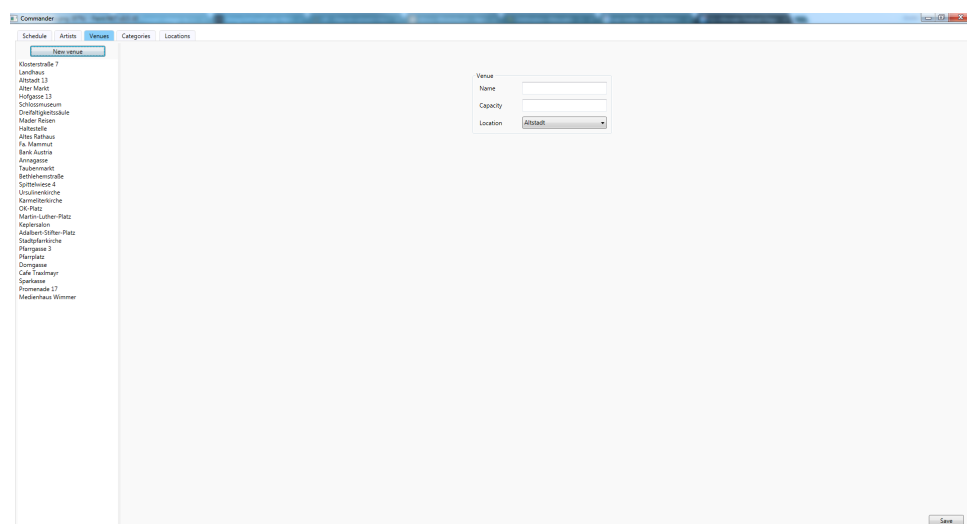


Abbildung 9: Startbild

Im Reiter *Categories* wird auf die Linke Seite des Fensters eine Liste mit alle im System vorhandene Kategorien. Wird eine Kategorie ausgewählt, so wird auf die Rechte Seite des Fensters aller Informationen zu der ausgewählte Kategorie angezeigt. Diese Informationen können hier auch editiert und gespeichert werden, oder neue Kategorien angelegt.

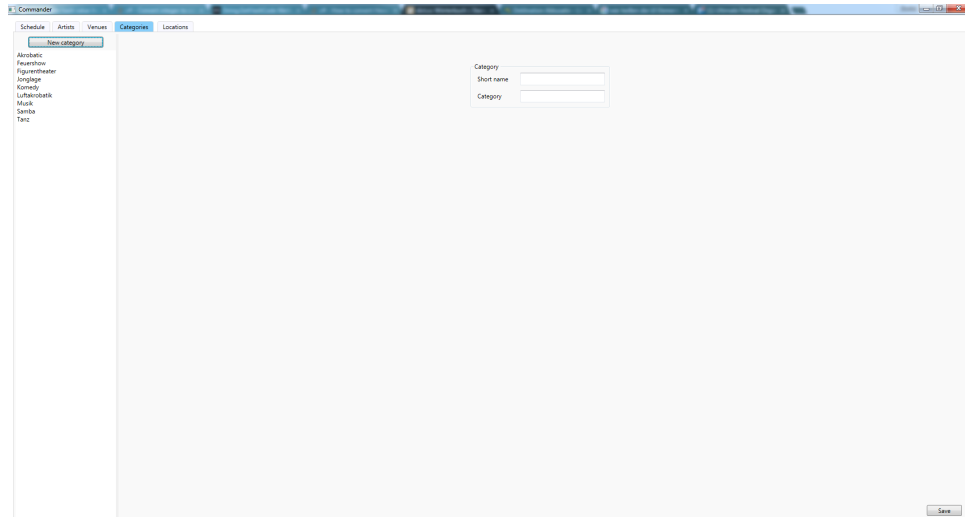


Abbildung 10: Startbild

Im Reiter *Locations* wird auf die Linke Seite des Fensters eine Liste mit alle im System vorhandene Standorte. Wird ein Standort ausgewählt, wird auf die Rechte Seite des Fensters aller Informationen zu dem ausgewählten Standort angezeigt. Diese Informationen können hier auch editiert und gespeichert werden, oder neue Standorte angelegt.

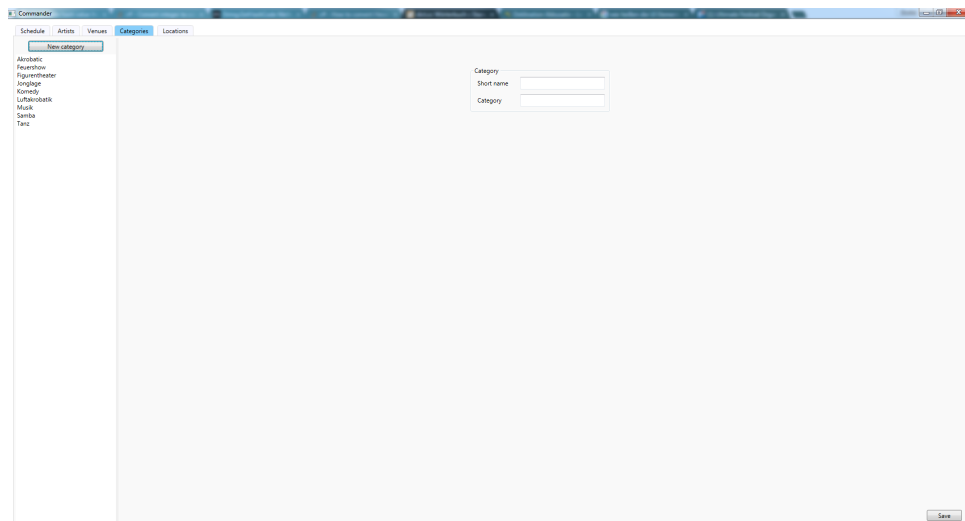


Abbildung 11: Startbild

1.4 Anwendungsfälle

1.4.1 Künstler anlegen

1.4.2 Programm abfragen

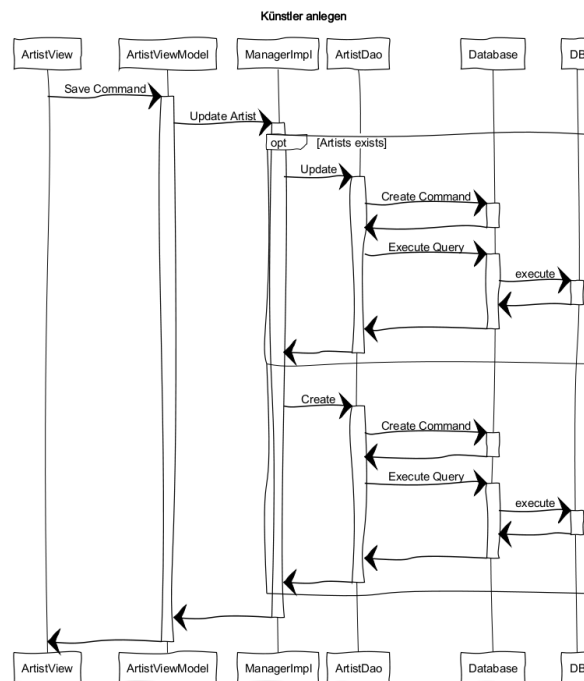


Abbildung 12: Künstler anlegen

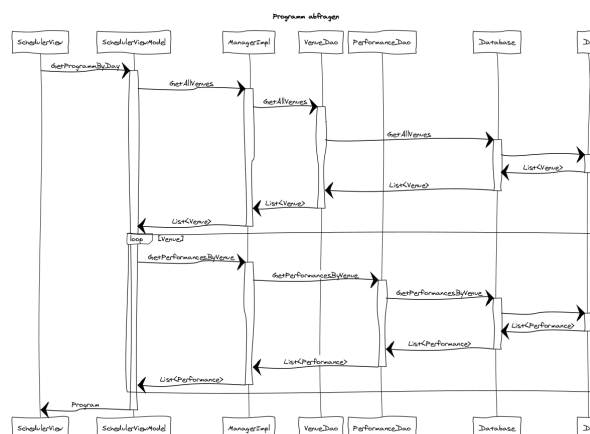


Abbildung 13: Programm abfragen

1.5 Web Client

Der Web-Client wird in Form einer JSF-Anwendung realisiert. Für die Benutzeroberfläche wird PrimeFaces Zusatzbibliothek verwendet. Der Web-Client implementiert eine SOAP-basiertes Service-Proxy, um Daten von Server über den Web-Service abfragen zu können. Der Benutzer kann über den Web-Clients Informationen über den aktuellen Programm, Künstler oder Spielstätten abfragen. Ein registrierter Organisator, kann sich über den Web-Client an Server anmelden und kann Änderungen am Programm durchführen.

Auf die Startseite wird dem Benutzer auf eine Karte der Ort wo das Pflasterspektakel stattfindet. Die JSF Anwendung bietet dem Benutzer die Möglichkeit das aktuelle Programm zu visualisieren, eine Liste mit alle Künstler oder Spielstätten zu sehen. Der Benutzer kann anhand eine Filter ein bestimmen Artist oder eine bestimmte Spielstätte suchen.

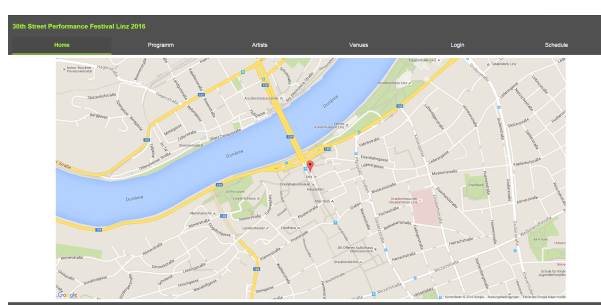


Abbildung 14: Startseite

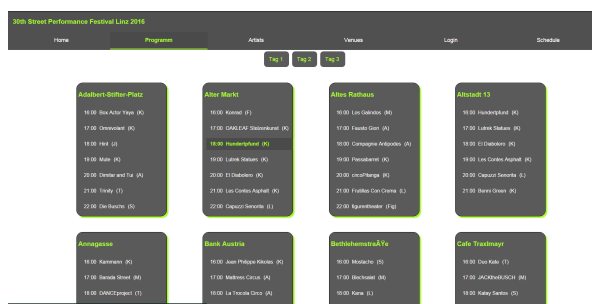


Abbildung 15: Programm

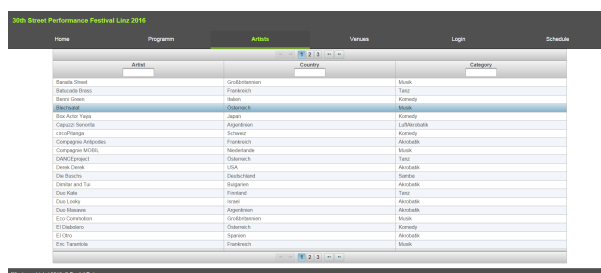


Abbildung 16: Künstler

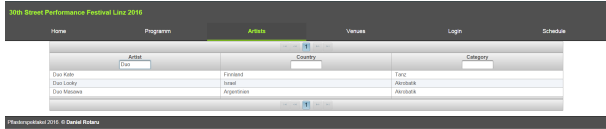


Abbildung 17: Künstler gefiltert

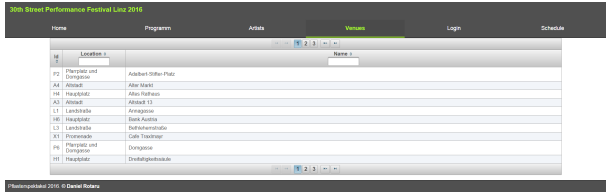


Abbildung 18: Spielstätten

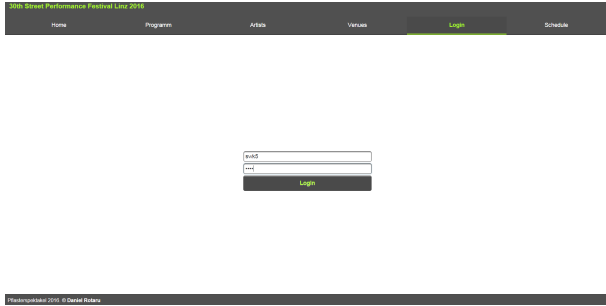


Abbildung 19: Login

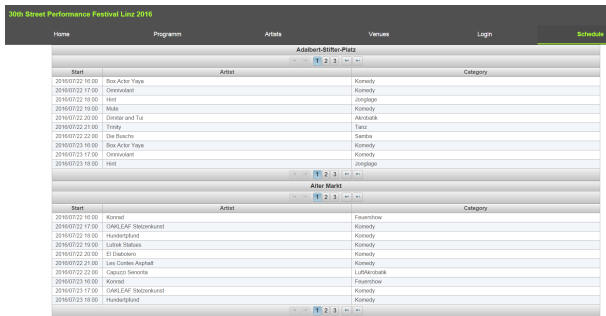


Abbildung 20: Organisatoren Ansicht

1.6 Unit-Tests

Für jede implementierte DAO-Methode wurde jeweils ein Unit-Test implementiert. Dafür wurde eine eigene Testdatenbank erstellt. Als Testumgebung wurde XUnit-Framework verwendet. Diesen *Framework* bietet ein *AutoRollback*-Attribut, der von Entwickler implementiert werden muss. Diesen Attribut verwendet Datenbanktransaktionen um die Unit-Test Daten nicht in die Datenbank zu speichern. Damit gibt es keine Abhängigkeit zwischen einzelnen Unit-Tests und jeden Unit-Test ist atomar.

```
using System;
using System.Reflection;
using System.Transactions;
using Xunit.Sdk;

namespace Ufo.DAL.Test
{
    [AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, AllowMultiple = false, Inherited =
        true)]
    class AutoRollbackAttribute : BeforeAfterTestAttribute
    {
        IsolationLevel isolationLevel = IsolationLevel.Unspecified;
        TransactionScope scope;
        TransactionScopeOption scopeOption = TransactionScopeOption.Required;
        long timeoutInMS = -1;

        /// <summary>
        /// Gets or sets the isolation level of the transaction.
        /// Default value is <see cref="IsolationLevel"/>.Unspecified.
        /// </summary>
        public IsolationLevel IsolationLevel
        {
            get { return isolationLevel; }
            set { isolationLevel = value; }
        }

        /// <summary>
        /// Gets or sets the scope option for the transaction.
        /// Default value is <see cref="TransactionScopeOption"/>.Required.
        /// </summary>
        public TransactionScopeOption ScopeOption
        {
            get { return scopeOption; }
            set { scopeOption = value; }
        }

        /// <summary>
        /// Gets or sets the timeout of the transaction, in milliseconds.
        /// By default, the transaction will not timeout.
        /// </summary>
        public long TimeoutInMS
        {
            get { return timeoutInMS; }
            set { timeoutInMS = value; }
        }

        /// <summary>
        /// Rolls back the transaction.
        /// </summary>
        public override void After(MethodInfo methodUnderTest)
        {
            scope.Dispose();
        }

        /// <summary>
        /// Creates the transaction.
        /// </summary>
        public override void Before(MethodInfo methodUnderTest)
        {
            TransactionOptions options = new TransactionOptions();
            options.IsolationLevel = isolationLevel;
            if (timeoutInMS > 0)

```

```

        options.Timeout = new TimeSpan(timeoutInMS * 10);
        scope = new TransactionScope(scopeOption, options);
    }
}
}

```

Test Category	Test Name	Execution Time
ArtistTest (5)	Ufo.DAL.Test.ArtistTest.DeleteArtistTest	1 sec
	Ufo.DAL.Test.ArtistTest.FindAllArtistsTest	22 ms
	Ufo.DAL.Test.ArtistTest.FindArtistByIdTest	25 ms
	Ufo.DAL.Test.ArtistTest.InsertArtistTest	14 ms
	Ufo.DAL.Test.ArtistTest.UpdateArtistTest	27 ms
CategoryTest (5)	Ufo.DAL.Test.CategoryTest.DeleteCategoryTest	1 sec
	Ufo.DAL.Test.CategoryTest.FindAllCategoriesTest	19 ms
	Ufo.DAL.Test.CategoryTest.FindCategoryByIdTest	43 ms
	Ufo.DAL.Test.CategoryTest.InsertCategoryTest	23 ms
	Ufo.DAL.Test.CategoryTest.UpdateCategoryTest	27 ms
LocationTest (5)	Ufo.DAL.Test.LocationTest.DeleteLocationTest	26 ms
	Ufo.DAL.Test.LocationTest.FindAllLocationsTest	32 ms
	Ufo.DAL.Test.LocationTest.FindLocationByIdTest	1 sec
	Ufo.DAL.Test.LocationTest.InsertLocationTest	37 ms
	Ufo.DAL.Test.LocationTest.UpdateLocationTest	24 ms
PerformanceTest (5)	Ufo.DAL.Test.PerformanceTest.DeletePerformanceTest	24 ms
	Ufo.DAL.Test.PerformanceTest.FindAllPerformancesTest	31 ms
	Ufo.DAL.Test.PerformanceTest.FindPerformanceByIdTest	61 ms
	Ufo.DAL.Test.PerformanceTest.InsertPerformanceTest	31 ms
	Ufo.DAL.Test.PerformanceTest.UpdatePerformanceTest	47 ms
RestrictionTest (5)	Ufo.DAL.Test.RestrictionTest.DeleteRestrictionTest	29 ms
	Ufo.DAL.Test.RestrictionTest.FindAllRestrictionsTest	27 ms
	Ufo.DAL.Test.RestrictionTest.FindRestrictionByIdTest	45 ms
	Ufo.DAL.Test.RestrictionTest.InsertRestrictionTest	79 ms
	Ufo.DAL.Test.RestrictionTest.UpdateRestrictionTest	21 ms
UserTest (5)	Ufo.DAL.Test.UserTest.DeleteUserTest	32 ms
	Ufo.DAL.Test.UserTest.FindAllUsersTest	16 ms
	Ufo.DAL.Test.UserTest.FindUserByIdTest	16 ms
	Ufo.DAL.Test.UserTest.InsertUserTest	13 ms
	Ufo.DAL.Test.UserTest.UpdateUserTest	14 ms
VenueTest (5)	Ufo.DAL.Test.VenueTest.DeleteVenueTest	23 ms
	Ufo.DAL.Test.VenueTest.FindAllVenuesTest	22 ms
	Ufo.DAL.Test.VenueTest.FindVenueByIdTest	32 ms
	Ufo.DAL.Test.VenueTest.InsertVenueTest	20 ms
	Ufo.DAL.Test.VenueTest.UpdateVenueTest	1 sec

Summary

Abbildung 21: Ergebnis Unit-Tests