

1 Ultimate Festival Organizer (UFO)

1.1 Datenmodell

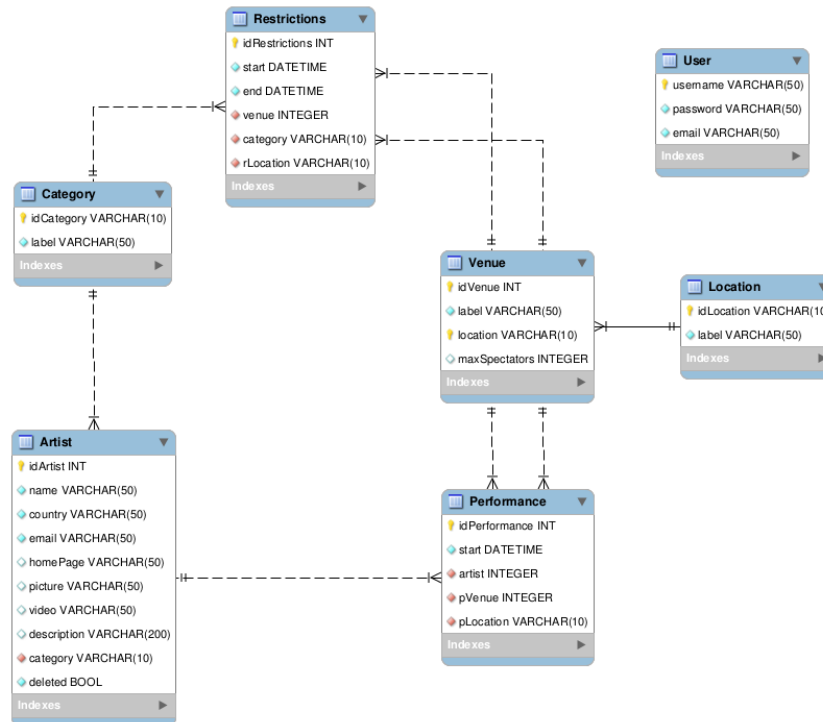


Abbildung 1: OR-Diagramm

Das Datenmodell enthält zusätzlich zu den vorgegebenen Entitäten zwei weitere Entitäten: *Location*-Entität und *Restrictions*-Entität. Die *Location*-Entität enthält als Primärschlüssel eine kurze Bezeichnung einer bestimmter Ort und der vollständige Name des Ortes, z.B. H - Hauptplatz, L - Landstraße. Der Primärschlüssel der *Location*-Entität wird als Fremdschlüssel und gleichzeitig Primärschlüssel in der Spielstätte-Entität verwendet. Die Spielstätte-Entität besitzt einen zusammengesetzten Primärschlüssel aus zwei Attributen: *idVenue* und der Fremdschlüssel *location*. Das Attribut *idVenue* wird nicht automatisch inkrementiert, sondern wird von Programm bestimmt, indem man der Anzahl an aktuell existierende Spielstätte für ein bestimmtes Ort um eins inkrementiert und als nächstes *idVenue* speichert. Dadurch schafft man, dass die *idVenue* für ein bestimmtes Ort, z.B. Hauptplatz, nicht größer wird als der Tatsächlichen Anzahl an Spielstätten für diesen Ort.

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.

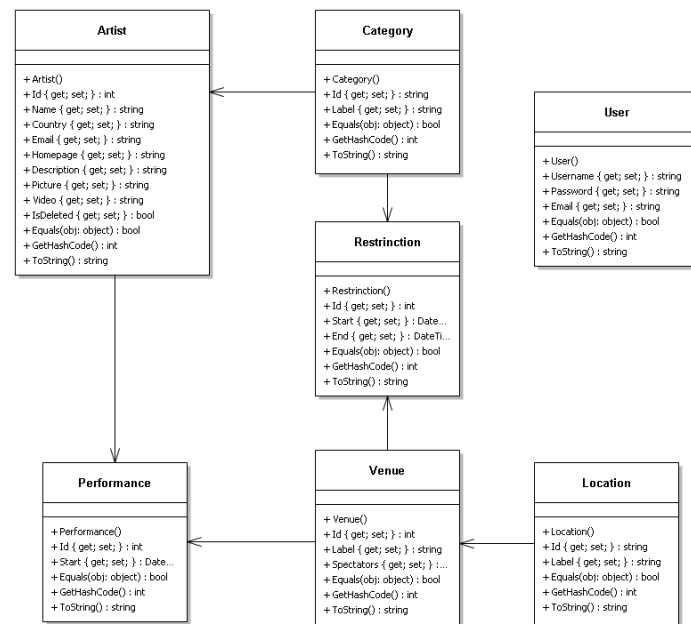


Abbildung 3: Domain Classes

1.2.1 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	Duration
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

1 Test, 1 Run, 15 Tests, 1 Run, 15 Tests, 0.00.00

Abbildung 4: Ergebnis Unit-Tests

1.3 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:

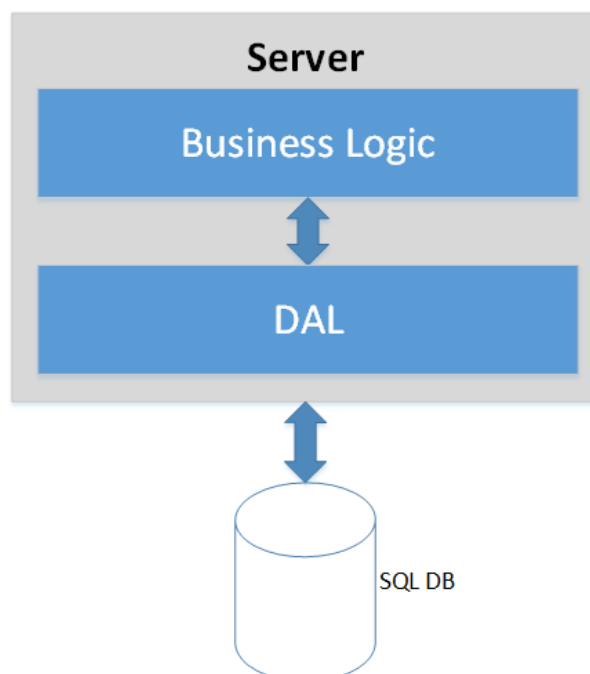


Abbildung 5: Ergebnis Unit-Tests

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.

1.3.1 Unit-Tests

Für jede Methode aus der Geschäftslogik wurde ein Test-case geschrieben. Damit wird sichergestellt, dass die Geschäftslogik immer die richtige Daten an die höhere Schicht liefert.

1.4 Präsentationsschicht

Die Präsentationsschicht für den UFO Commander wird im WPF nach der Model-View-ViewModel Pattern realisiert. Als Model werden die Domainklassen verwendet.



Abbildung 6: Schichten

Der ViewModel wird verwendet, um die Sicht unabhängig von die Domainklassen zu implementieren.

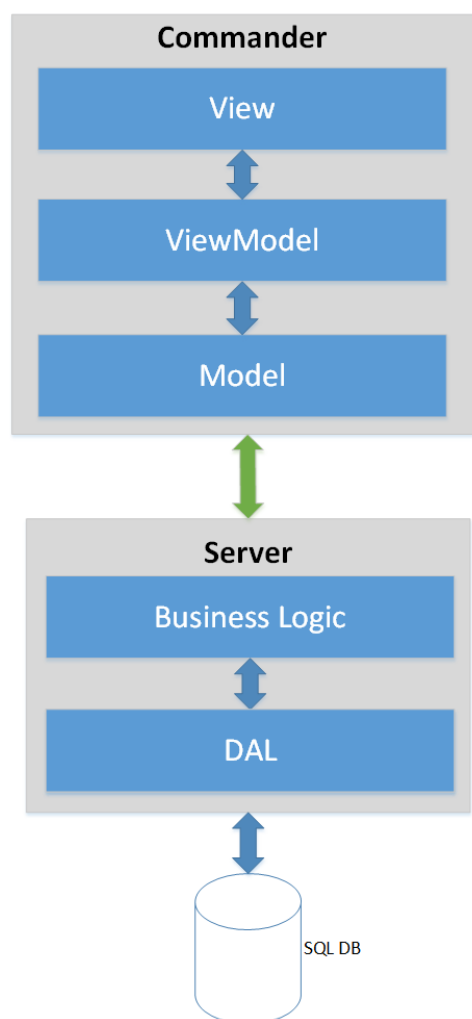


Abbildung 7: Komponentenübersicht

1.4.1 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.

Venue	16:00	17:00	18:00	19:00	20:00	21:00	22:00
Altstadt							
Klosterstraße 7	Capuzzi Seniors Argentinien K	Benni Green Italien K	Ian Deadly Großbritannien M	Urban Safari Niederlande K	HoopStep Deutschland A	The LEDies Deutschland T	Konrad Deutschland F
Landhaus	El Diablero Österreich K	Les Contes Asphalt Belgien K	Capuzzi Seniors Argentinien K	Benni Green Italien K	Ian Deadly Großbritannien M	Urban Safari Niederlande K	HoopStep Deutschland A
Altstadt 13	Hundertfund Deutschland K	Lutrek Statues Polen K	El Diablero Österreich K	Les Contes Asphalt Belgien K	Capuzzi Seniors Argentinien K	Benni Green Italien K	
Alter Markt	Konrad Deutschland F	OAKLEAF Stelzenkunst Deutschland K	Hundertfund Deutschland K	Lutrek Statues Polen K	El Diablero Österreich K	Les Contes Asphalt Belgien K	Capuzzi Seniors Argentinien L
Hofgasse 13	HoopStep Deutschland T	The LEDies Deutschland T	Konrad Deutschland F	OAKLEAF Stelzenkunst Deutschland K	Hundertfund Deutschland K	Lutrek Statues Polen K	El Diablero Österreich K
Schlossmuseum	Ian Deadly Großbritannien M	Urban Safari Niederlande K	HoopStep Deutschland A	The LEDies Deutschland T	Konrad Deutschland F	OAKLEAF Stelzenkunst Deutschland K	Hundertfund Deutschland K
Hauptplatz							
Dreifaltigkeitskirche	Figurentheater Österreich Fig	El Otro Spanien A	Jean Philippe Kikolas Portugal K	Mattress Circus Portugal A	La Trocola Circo Spanien A	Luca Bellezze Italien K	Los Galindos Spanien M
Mader Reiten	circoPitanga Schweiz K	Frutillas Con Crema Chile L	Figurentheater Österreich Fig	El Otro Spanien A	Jean Philippe Kikolas Portugal K	Mattress Circus Portugal A	La Trocola Circo Spanien A
Haltestelle	Compagnie Antipodes Frankreich A	Passabarret Spanien K	circoPitanga Schweiz K	Frutillas Con Crema Chile L	Figurentheater Österreich Fig	El Otro Spanien A	
Altes Rathaus	Los Galindos Spanien M	Fausto Giori Italien A	Compagnie Antipodes Frankreich K	Passabarret Spanien K	circoPitanga Schweiz K	Frutillas Con Crema Chile L	Figurentheater Österreich Fig
Fa. Mammut	La Trocola Circo Spanien A	Luca Bellezze Italien K	Los Galindos Spanien M	Fausto Giori Italien A	Compagnie Antipodes Frankreich K	Passabarret Spanien K	circoPitanga Schweiz K
Bank Austria	Jean Philippe Kikolas Portugal A	Mattress Circus Portugal A	La Trocola Circo Spanien A	Luca Bellezze Italien K	Los Galindos Spanien M	Fausto Giori Italien A	Compagnie Antipodes Frankreich A
Landstraße							
Amgasse	Kammann Deutschland F	Barada Street Großbritannien M	DANCEProject Österreich K	Grande Cartaggio Barattoli Italien A	Shay Wapnias Italien A	Folkshilfe Österreich M	Samuelito Schweiz F

Abbildung 8: Startbild

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.

Performance

Artist:

Venue:

Starts at:

Abbildung 9: 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.

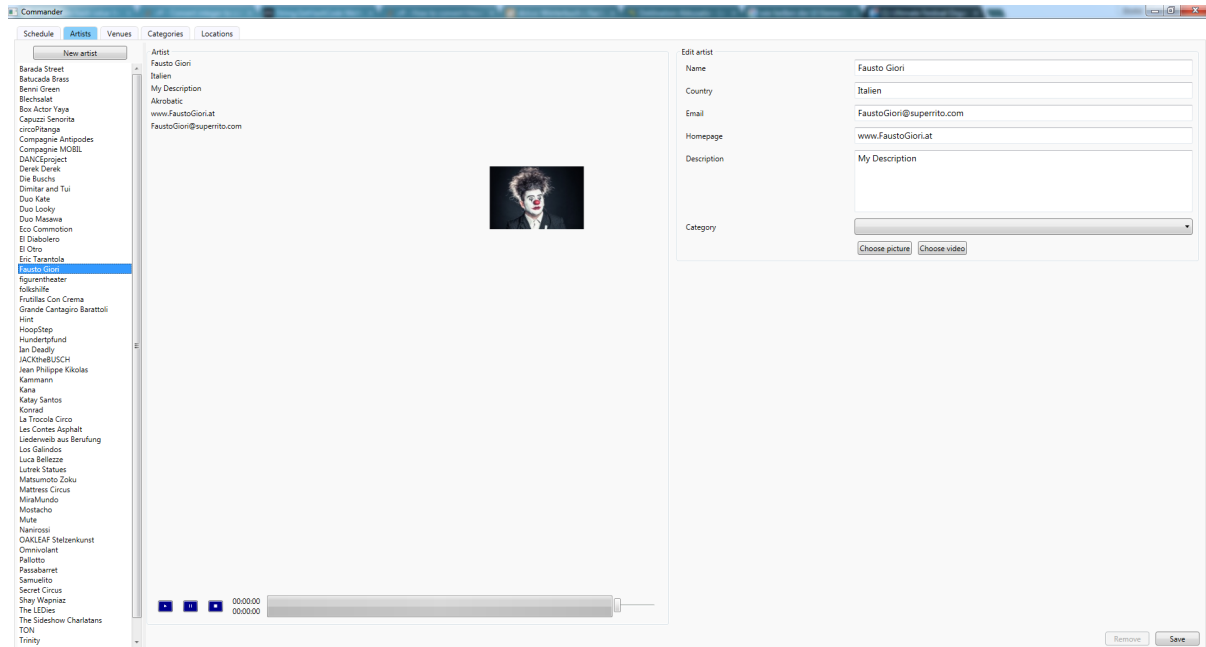


Abbildung 10: Startbild

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.

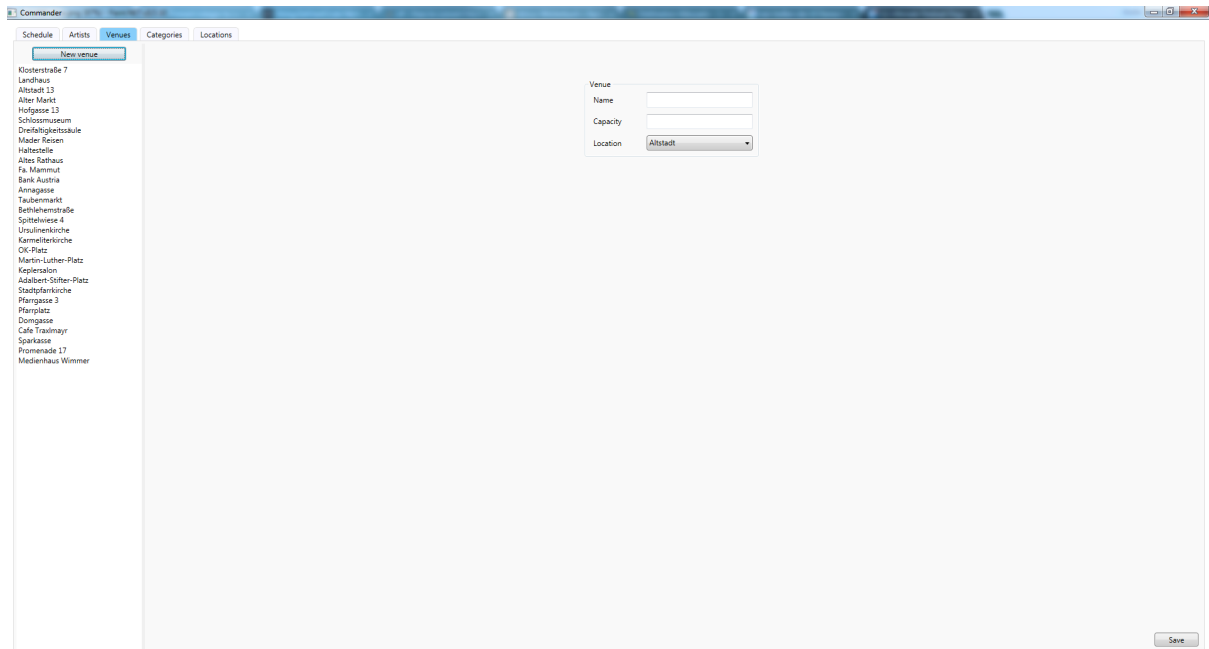


Abbildung 11: Startbild

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.

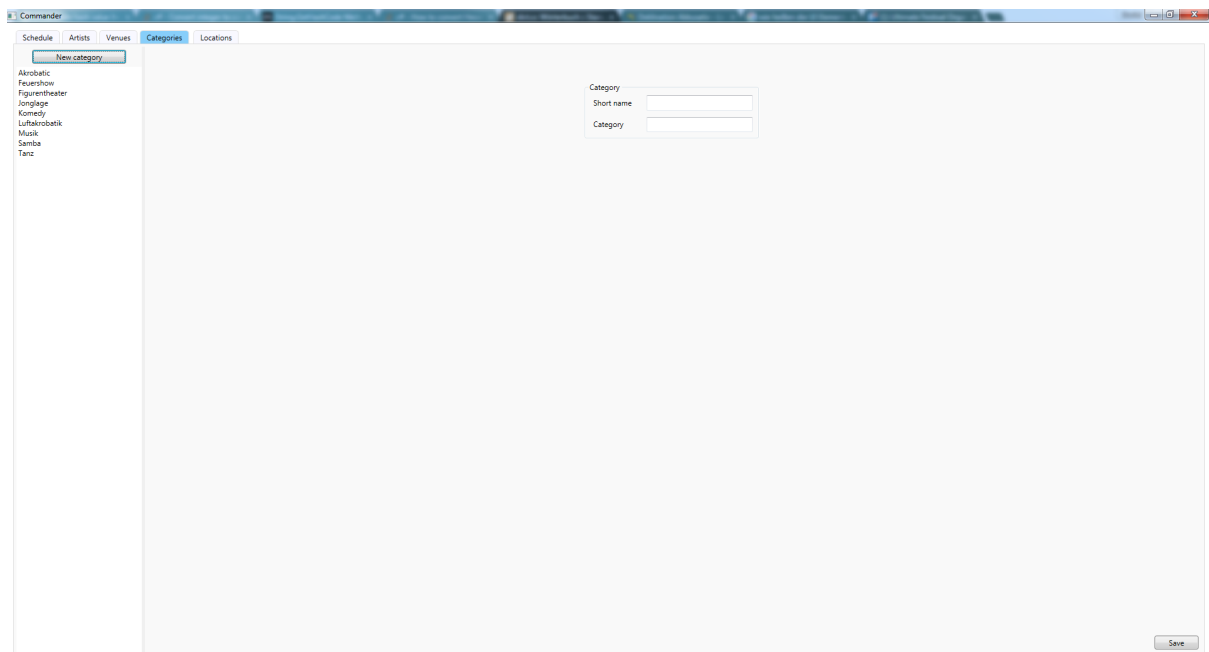


Abbildung 12: Startbild

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.

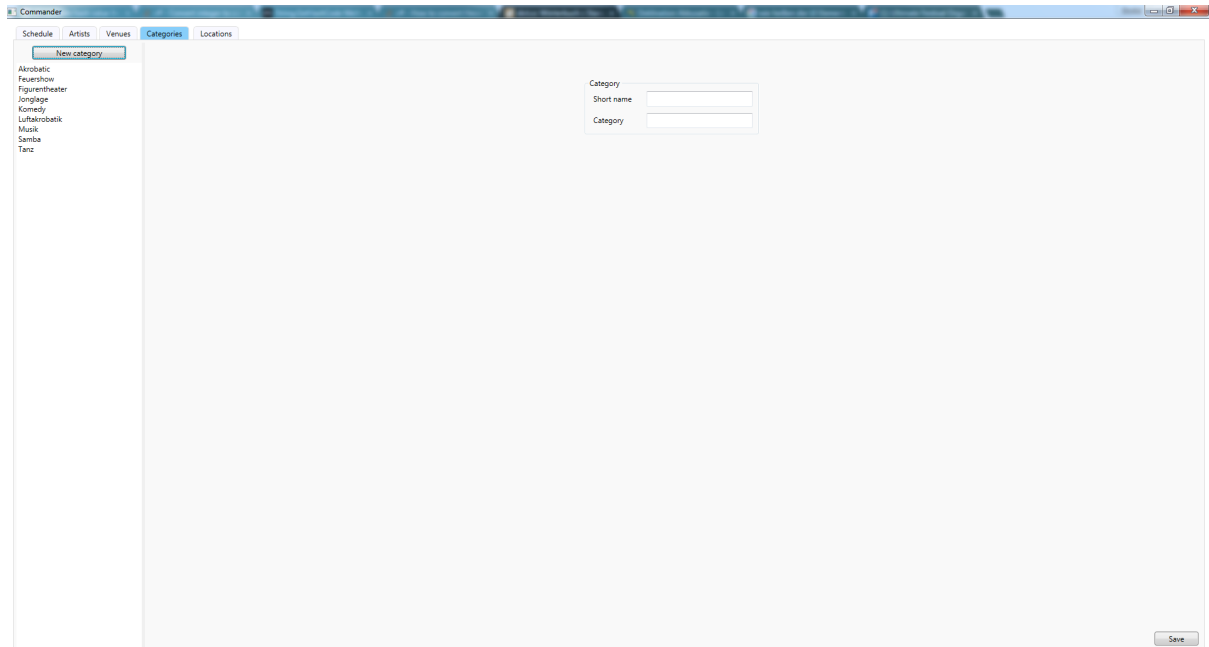


Abbildung 13: Startbild