

autoPSI::Architekturdokument

Architekturdokument

autoPSI

Gruppe 06: Feledi, Firato, Mildner, Zapotocky

Inhalt

Inhalt	2
Systemarchitektur.....	4
UserInterface.....	4
GenericDAO.....	4
Grundsätzliche Idee.....	4
Eingesetzte Techniken.....	4
JavaSpace.....	5
Die Idee hinter JavaSpaces.....	5
JavaSpaces in autoPSI.....	5
Domänenmodell.....	6
Pakete.....	7
Paket-Diagramm.....	7
Klassendiagramme.....	10
Hinweise zu den Diagrammen.....	10
Basis.....	11
gui und gui.exception.....	12
gui.component.....	13
gui.dialog.....	15
database.dao.....	16
database.sql.....	17
database.table.....	18
javaspace.....	19
Datenbankbeschreibung.....	20
ER-Diagramm.....	20
Datenbanktabellen.....	21
attachable_object.....	21
termincontainer.....	22
termin.....	23

Iva.....	24
Kontakt.....	25
Lehrmittel.....	26
Pruefung.....	27
Notiz.....	28
Universitaet.....	29
lehrmittel_kategorie.....	30
attachable_object_kategorie.....	31
Termin_Kategorie.....	32
anhaengen_termincontainer.....	33
anhaengen_termin.....	34

Systemarchitektur

UserInterface

autoPSI soll über eine übersichtliche und intuitiv bedienbare Benutzeroberfläche verfügen. Alle Funktionen sollen schnell erreichbar sein. Es sollen möglichst wenige und vor allem selbsterklärende Steuerelemente zum Einsatz kommen, denn es soll vermieden werden, dass Benutzer in einer Dokumentation nachschlagen müssen. Ähnliche Objekte sollen in ähnlichen Abläufen bearbeitbar sein, da dies im allgemeinen sehr intuitiv ist (Man probiert etwas, bemerkt, dass es funktioniert und kann den Ablauf dann an verschiedenen anderen Stellen im Programm wiederverwenden). Durch möglichst sinnvolle Gliederung der Informationen in graphischer Form soll die Übersicht auch bei vielen Terminen möglichst erhalten bleiben. Durch die Möglichkeit, mehrere Fenster zum Bearbeiten zu öffnen, wird das gleichzeitige Bearbeiten von mehreren Objekten ermöglicht (und dadurch auch Copy-And-Paste gefördert).

GenericDAO

Grundsätzliche Idee

Die grundsätzliche Idee ist es, den Datenbankzugriff so generisch wie möglich zu gestalten. Das zu erreichende Ziel ist es, dass möglichst nur ein Objekt tatsächlich mit der Datenbank kommuniziert und dass dieses eine Objekt auf beliebige Datenbanktabellen zugreifen kann und Daten auch in Form von Objekten zurückgeben kann.

Eingesetzte Techniken

Erreicht wird dies durch Verwendung von Reflection. Durch diese Technik ist es möglich, die Daten einer Datenbanktabelle in ein dem GenericDAO-Objekt unbekanntes Objekt zu schreiben.

Um dem Benutzer möglichst große Flexibilität beim Auffinden von Objekten zu geben, soll die Suche nach Objekten mithilfe eines sogenannten LookupObjects erfolgen. Dieses Objekt ist ein Template, welches der Suchfunktion übergeben wird und bei dem diejenigen Eigenschaften, die gefundene Objekte aufweisen sollen, gesetzt sind, während alle anderen Felder Nullreferenzen sind.

JavaSpace

Die Idee hinter JavaSpaces

Die Idee hinter der JavaSpaces-Technologie ist es, einen möglichst einfach verwendbaren Datenspeicher für Objekte anzubieten. Ein JavaSpace bietet vier grundsätzliche Operationen:

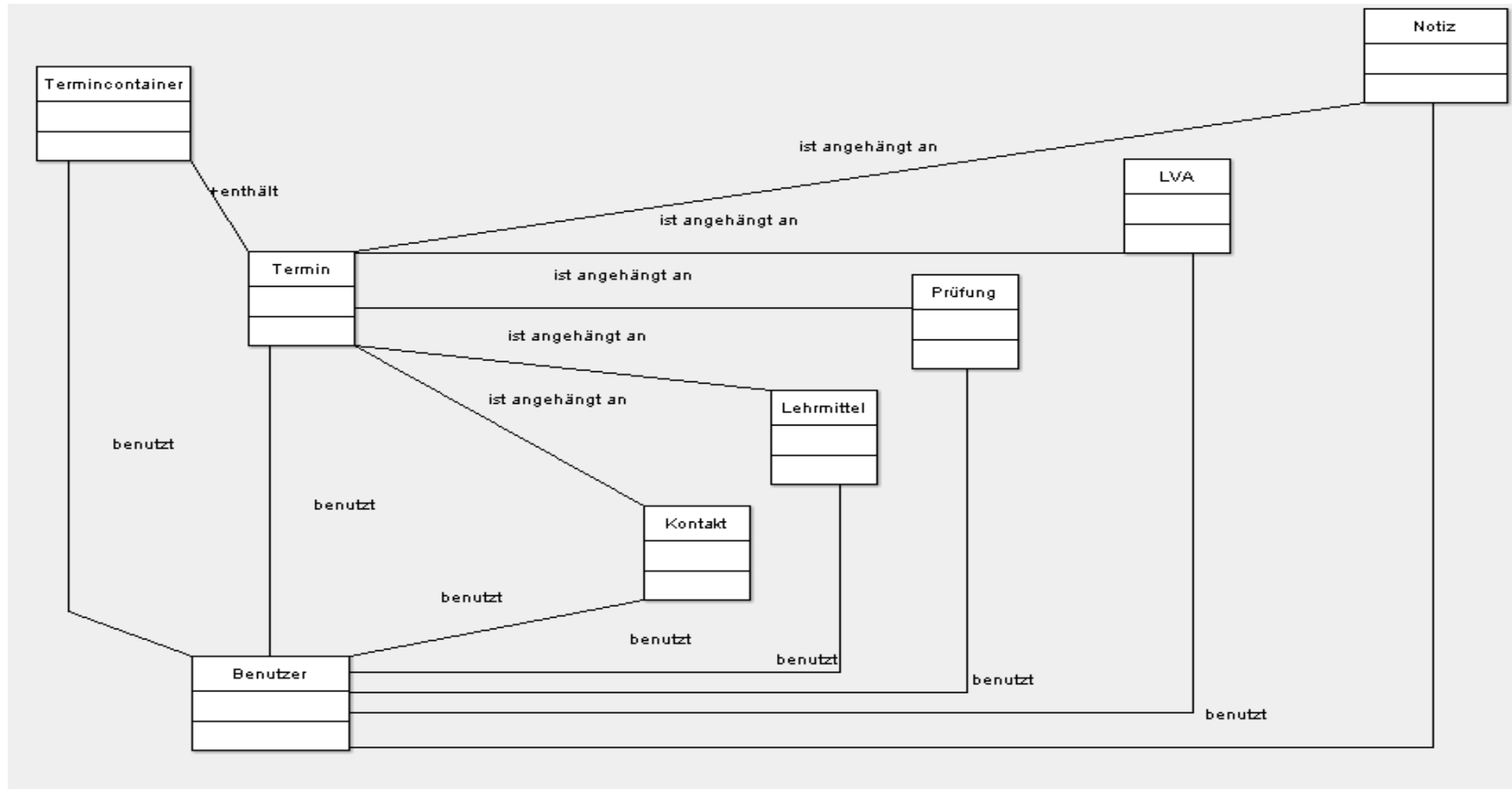
1. **Write:** Schreibt ein Objekt in den JavaSpace
2. **Read:** Liest ein Objekt aus dem JavaSpace
3. **Take:** Liest ein Objekt konsumierend aus dem JavaSpace
4. **Notify:** Informiert den Aufrufer, wenn ein Objekt in den JavaSpace eingefügt wurde, das der vom Aufrufer geforderten Beschreibung entspricht

Mit diesen Grundoperationen kann dann eine beliebige Business-Logik aufgebaut werden.

JavaSpaces in autoPSI

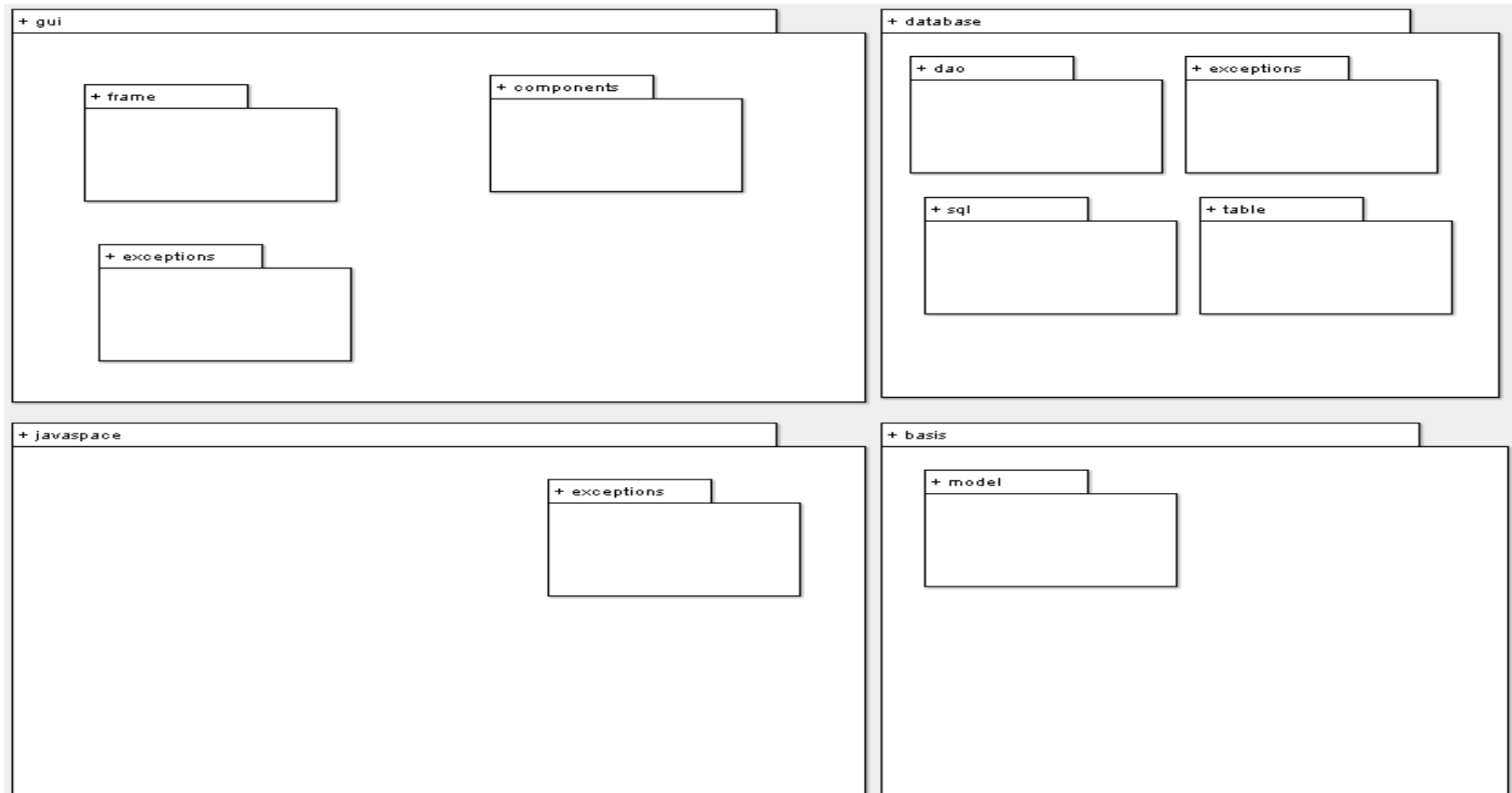
In autoPSI wird der Datenaustausch zwischen Studenten per JavaSpace unterstützt. Dadurch können Informationen einfach und effizient weitergegeben werden.

Domänenmodell



Pakete

Paket-Diagramm



Paketbeschreibung

<i>Package</i>	<i>Funktionalität</i>
basis	<u>Kurzbeschreibung:</u> Enthält alle Konfigurationsklassen <u>Beschreibung:</u> Enthält die Klasse AutopsiConfigurator, welche die benutzten Dateipfade setzt
basis.model	<u>Kurzbeschreibung:</u> Enthält alle TableModels, ListModels und ComboBoxModels des Projekts <u>Beschreibung:</u> Enthält alle Klassen, die TableModel implementieren bzw von AbstractTableModel abgeleitet sind sowie Klassen, die von ListModel bzw ComboBoxModel abgeleitet wurden.
gui	<u>Kurzbeschreibung:</u> Enthält alle Klassen, die das graphische User Interface erzeugen <u>Beschreibung:</u> Enthält die Klassen MainFrame, TerminEditFrame, TermincontainerEditFrame, SearchFrame, GenericEditFrame
gui.frame	<u>Kurzbeschreibung:</u> enthält die von JFrame abgeleiteten Fenster des UserInterfaces <u>Beschreibung:</u> Implementierung des UserInterfaces
gui.components	<u>Kurzbeschreibung:</u> enthält selbst erstellte Komponenten, die beim UserInterface zum Einsatz kommen <u>Beschreibung:</u> Dieses Package enthält z.B. die TableModels für die Monatsansicht und die Ergebnistabelle im Suchfenster
gui.exceptions	
database	<u>Kurzbeschreibung:</u> Enthält alle Klassen, die mit der Datenbank oder dem Datenbankzugriff zu tun haben <u>Beschreibung:</u>
database.dao	<u>Kurzbeschreibung:</u> Enthält die Komponenten für den generischen Datenbankzugriff und das Objekt, welches benutzt wird, um eigene Datenbankobjekte abzuleiten <u>Beschreibung:</u> Enthält IGenericDAO, GenericDAO, GenericDataObject
database.exceptions	<u>Kurzbeschreibung:</u> Exceptions, die in Datenbankkomponente auftreten können <u>Beschreibung:</u> Enthält EAttributeNotFound, EDatabase, EDatabaseConnection
database.table	<u>Kurzbeschreibung:</u> Enthält Objekte, die die Daten halten <u>Beschreibung:</u>

<i>Package</i>	<i>Funktionalität</i>
database.sql	<u>Kurzbeschreibung:</u> Die SQL-Struktur, die das GenericDAO benutzt, um mit SQL-Abfragen umzugehen <u>Beschreibung:</u>
javaspace	<u>Kurzbeschreibung:</u> Enthält die Komponenten, die mit dem JavaSpace kommunizieren <u>Beschreibung:</u>
javaspace.exceptions	<u>Kurzbeschreibung:</u> Exceptions, die im Zusammenhang mit den JavaSpace-Objekten auftreten können <u>Beschreibung:</u>

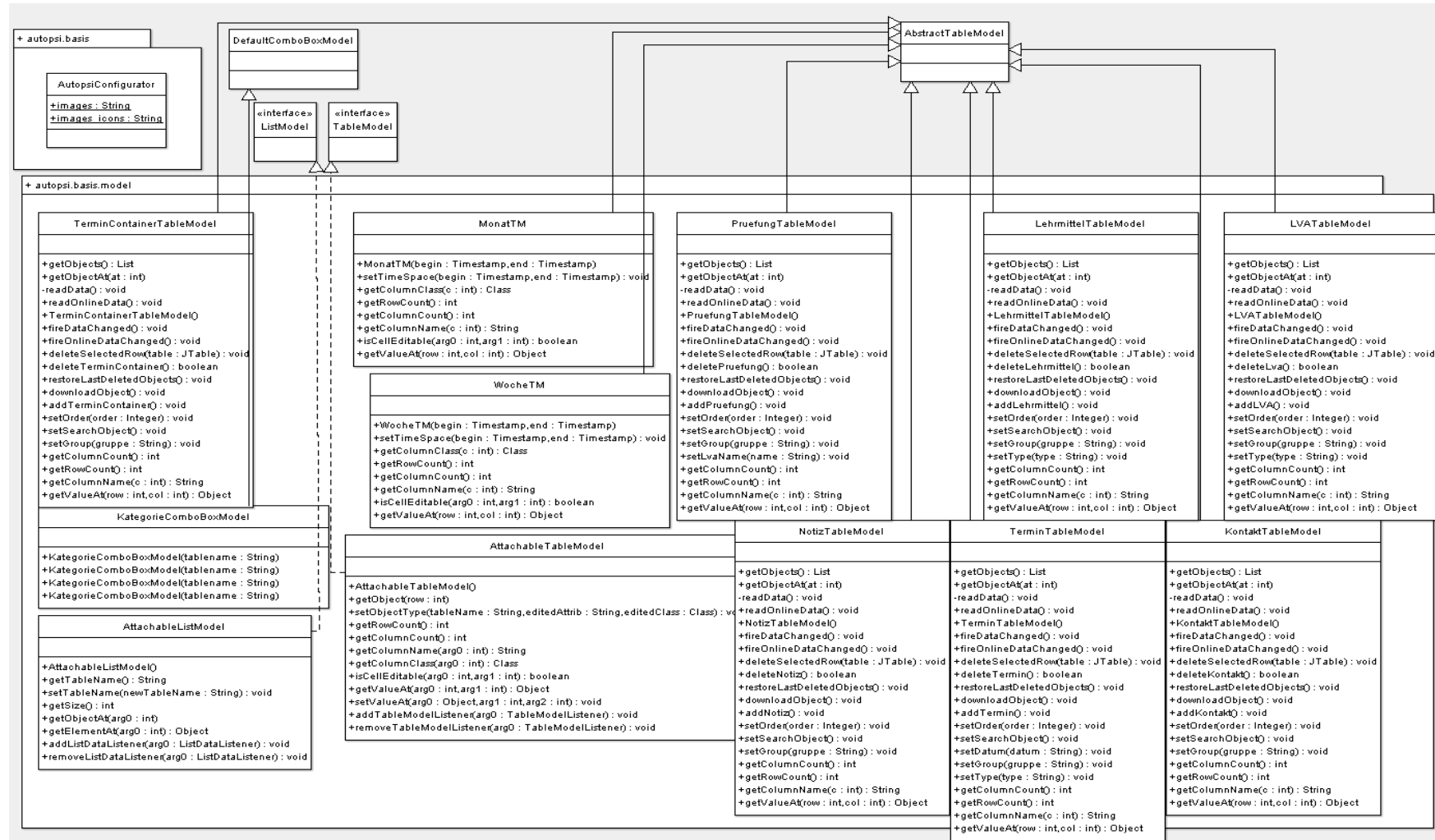
Klassendiagramme

Hinweise zu den Diagrammen

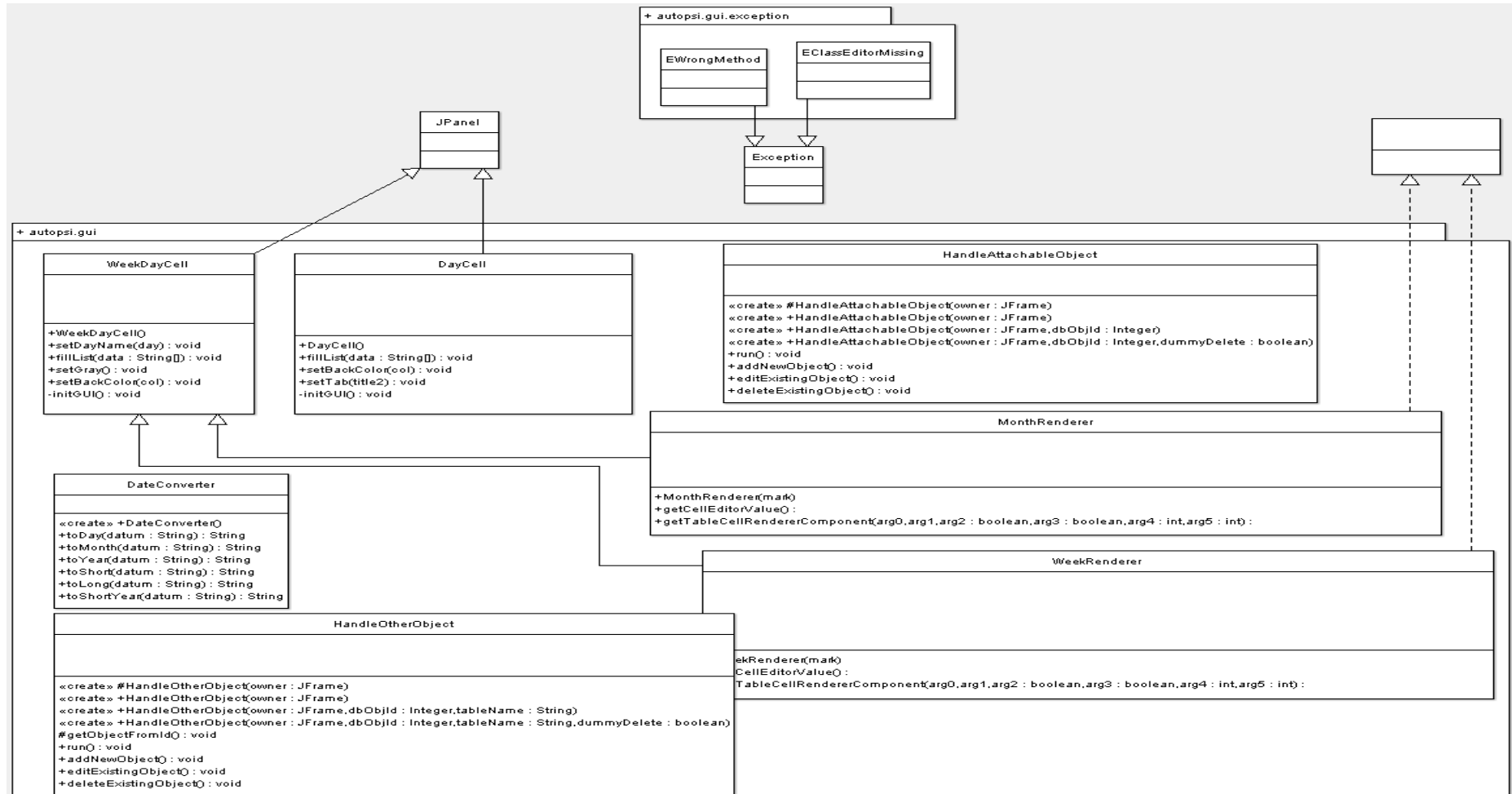
Bei den Diagrammen wurden die internen Variablen, die zur grafischen Darstellung benötigt wurden, größtenteils weggelassen, um die Übersichtlichkeit zu fördern; Der interessierte Leser sei hierbei auf die Dokumentation (autopsi/doc) verwiesen.

Im Paket autopsi.database.table wurden die Vererbungs- und Realisierungsrelationen weggelassen, wiederum mit dem Zwecke, die Übersichtlichkeit zu fördern; alle in diesem Package befindlichen Klassen haben als Oberklasse die Klasse GenericData (autopsi.gui.component) und implementieren die Interfaces Entry (net.jini.entry) und GenericDataObject (autopsi.database.gdo).

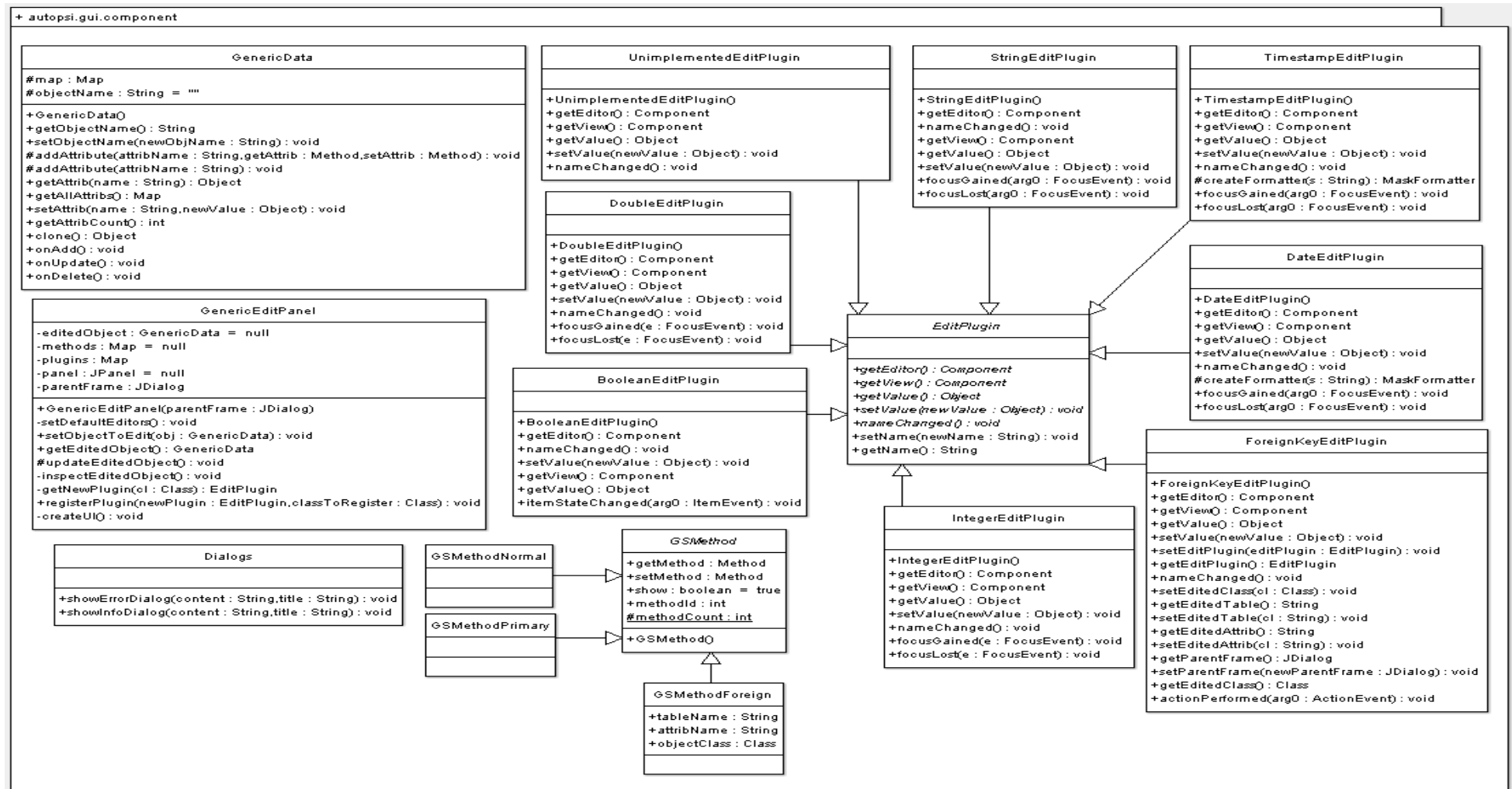
Basis



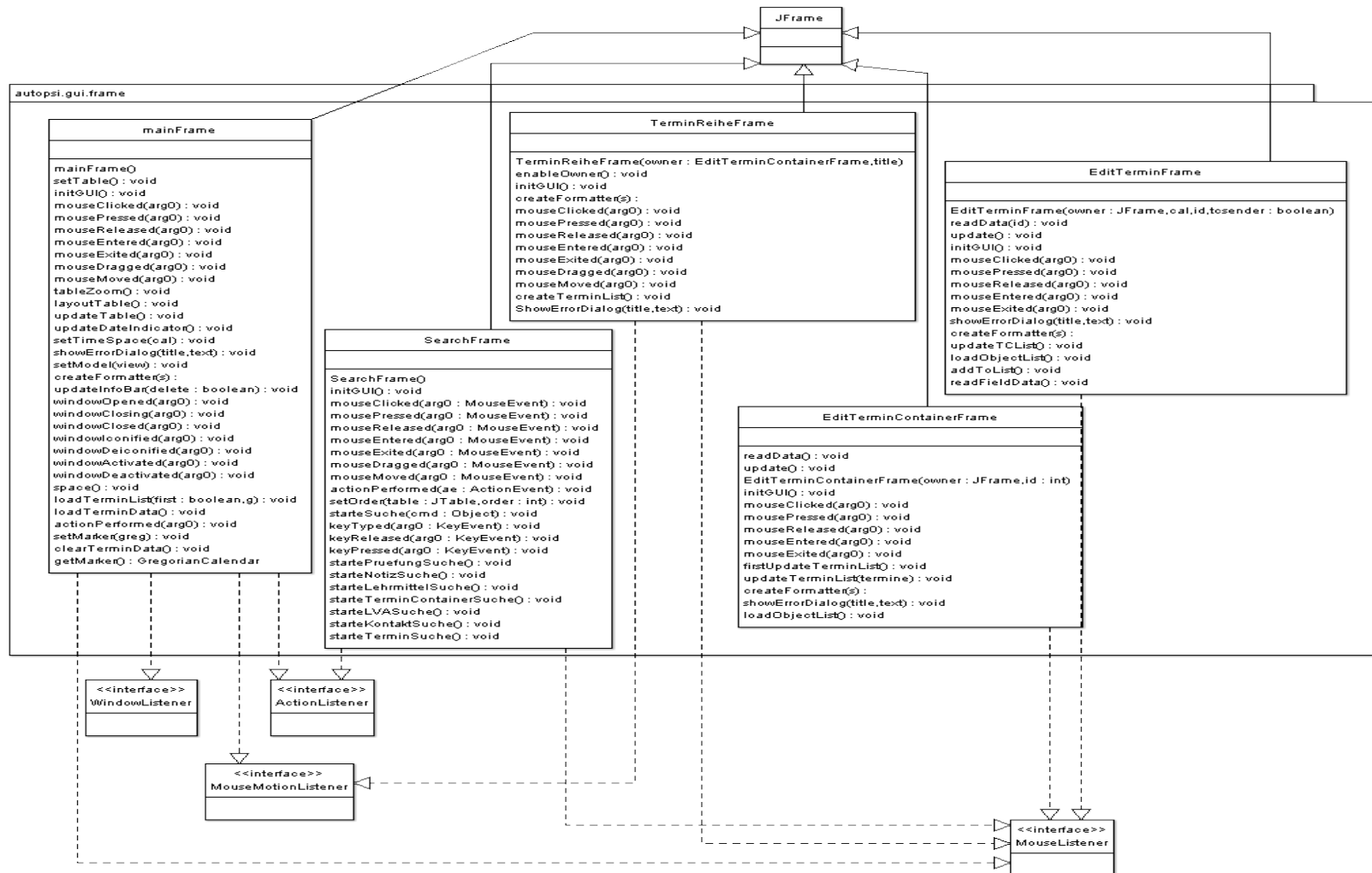
gui und gui.exception



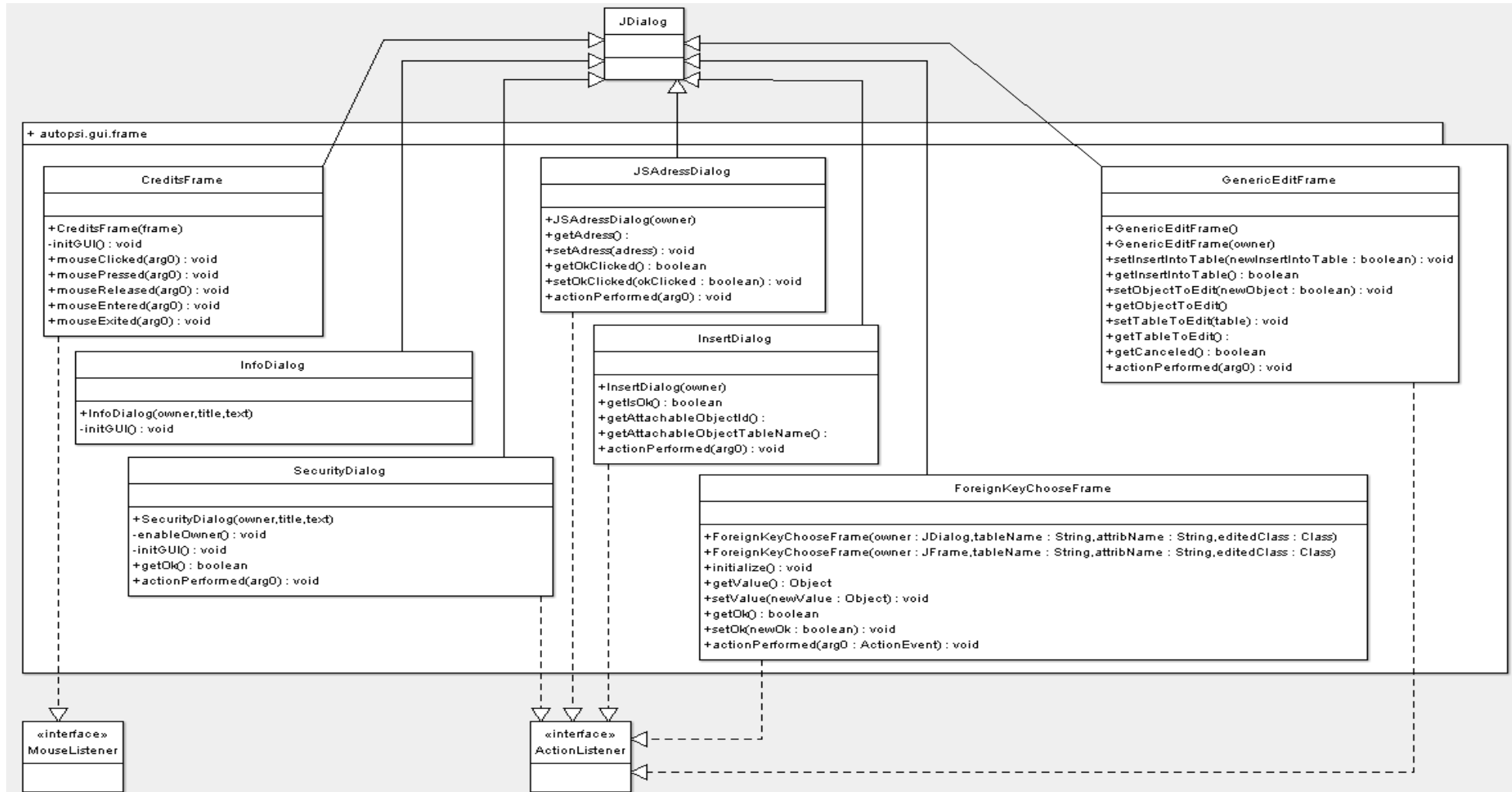
gui.component



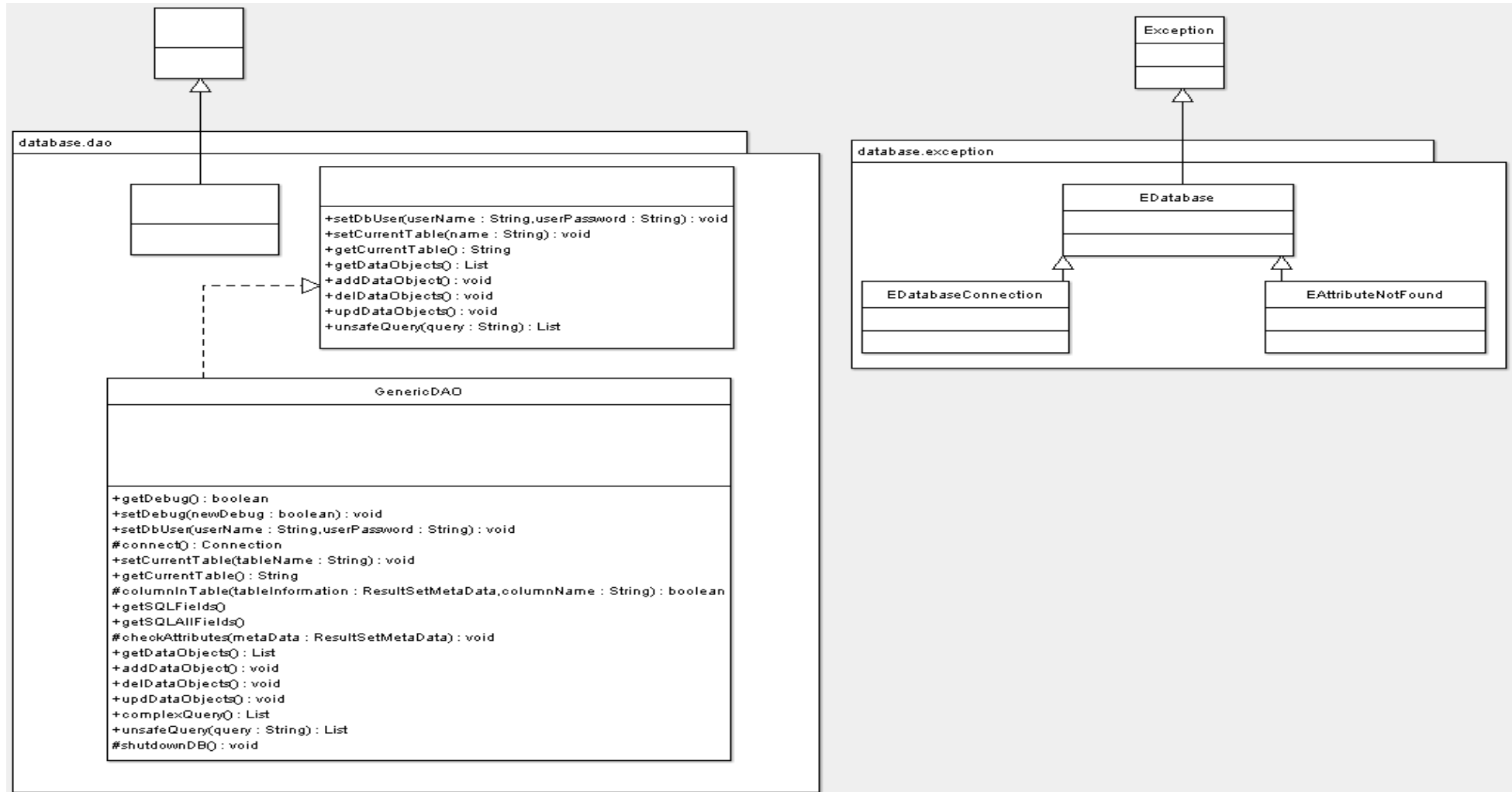
gui.frame



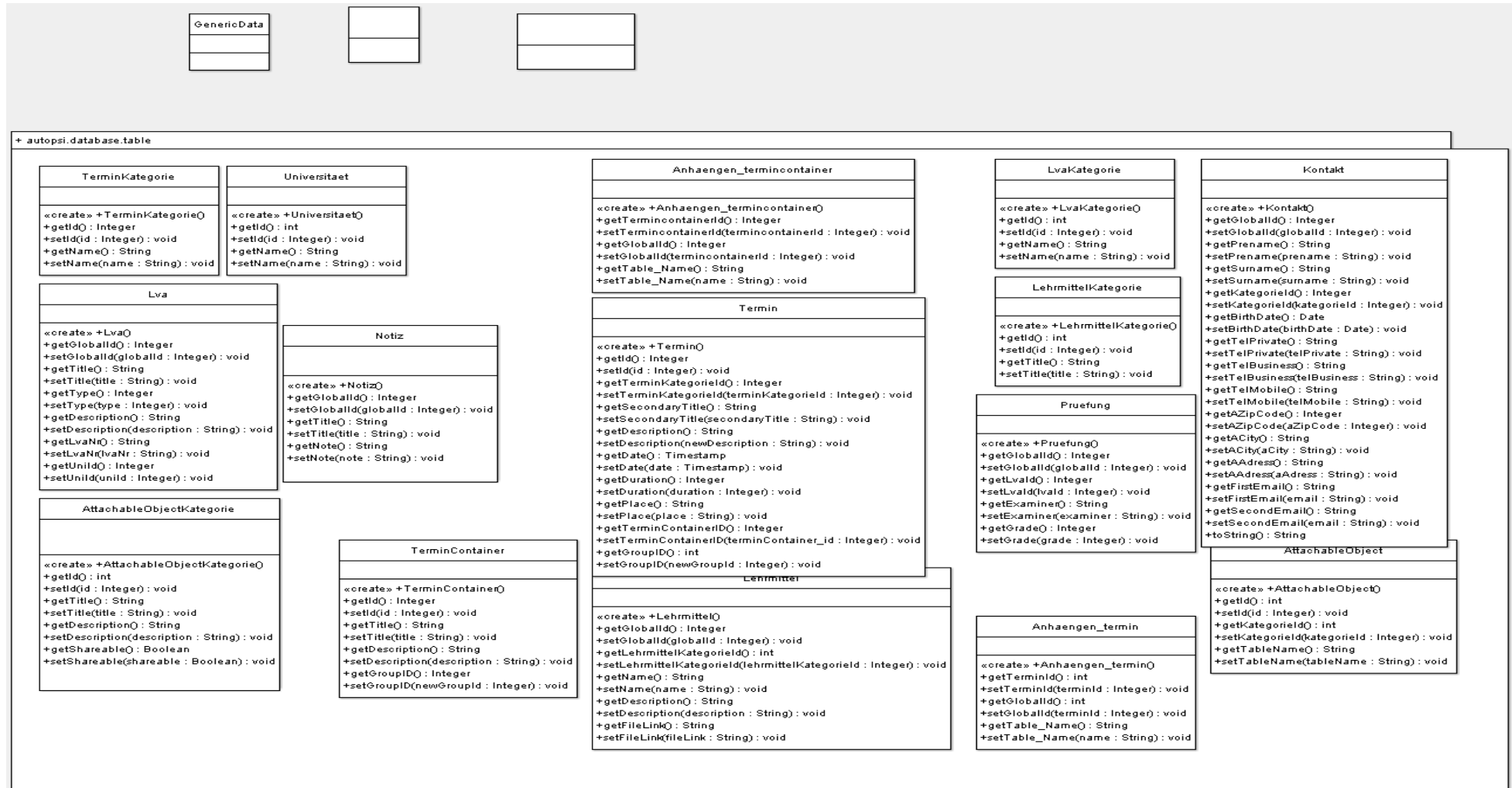
gui.dialog



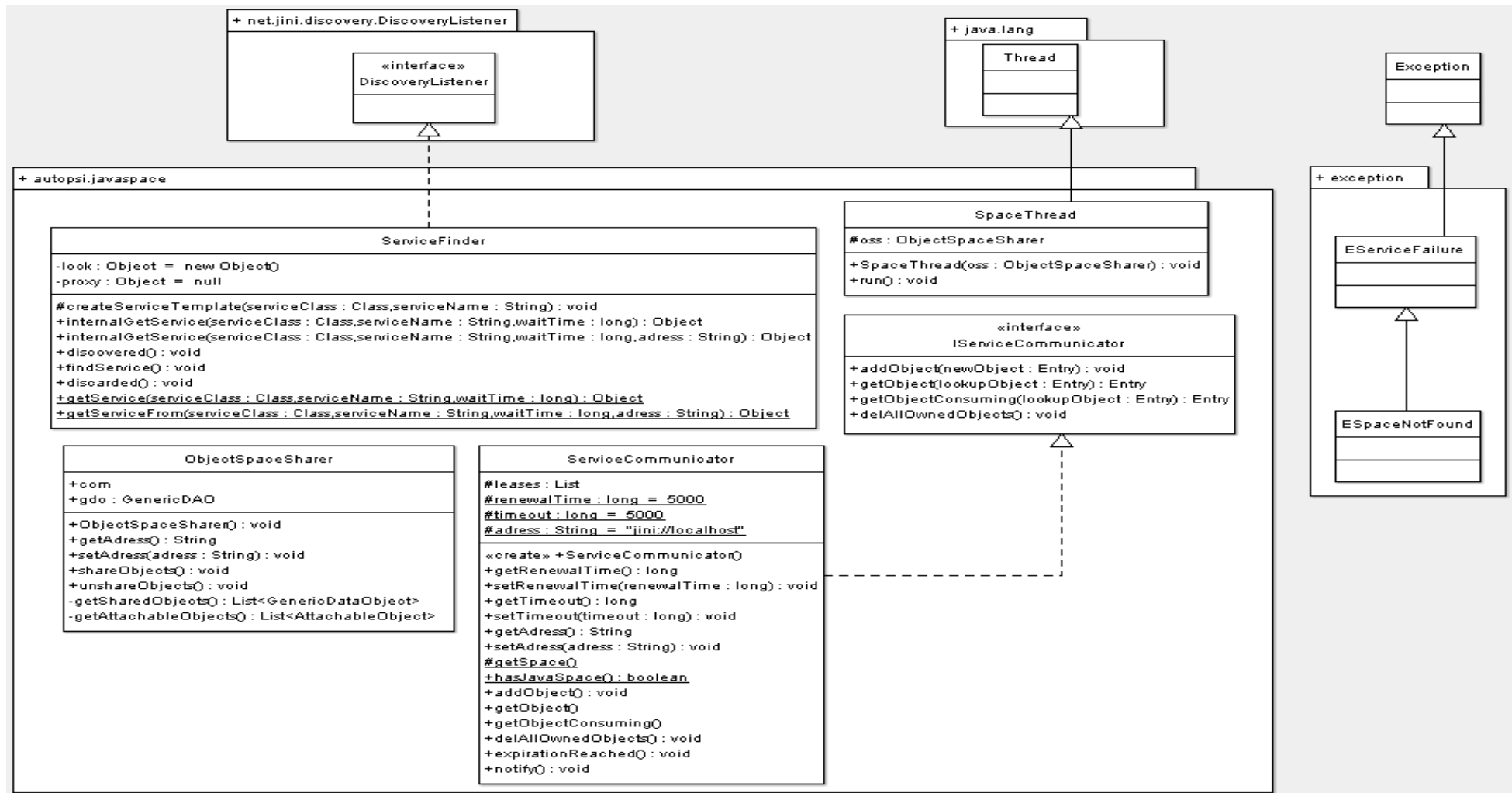
database.dao





database.table

javaspace



Datenbankbeschreibung

ER-Diagramm



Datenbanktabellen

attachable_object

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
global_id	INTEGER	!1+P	Globale ID, wird von einer Funktion in einer Helper-Klasse erzeugt und ist für alle anhängbaren Objekte eindeutig
kategorie_id	INTEGER	F	references object_kategorie->id
table_name	VARCHAR(255)		Name der Tabelle des eigentlichen Objekts

termincontainer

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
id	INTEGER	!1+P	
group_id	INTEGER	F	references ATTACHABLE_OBJECT_KATEG ORIE(ID)
title	VARCHAR(255)		
description	VARCHAR(255)		

termin

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
id	INTEGER	!1+P	
group_id	INTEGER	F	references Attachable_Object_Kategorie(id)
termin_kategorie_id	INTEGER	F	Fremdschlüssel zu (Termin_Kategorie.id)
secondary_title	VARCHAR(255)		
description	VARCHAR(255)		
date	TIMESTAMP		Anfangsdatum- und Zeit des Termins
duration	INTEGER		Termindauer in Minuten
place	VARCHAR(255)		
termincontainer_id	INTEGER		

lva

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
global_id	INTEGER	!1+PF	Globale ID, wird von einer Funktion in einer Helper-Klasse erzeugt und ist für alle anhängbaren Objekte eindeutig references attachable_object(global_id)
title	VARCHAR(255)		
type	INTEGER	F	references lva_kategorie(id)
description	VARCHAR(255)		
lva_nr	VARCHAR(255)	!1	
uni_id	INTEGER	F	references universitaet(id)

Kontakt

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
global_id	INTEGER	!1+PF	Globale ID, wird von einer Funktion in einer Helper-Klasse erzeugt und ist für alle anhängbaren Objekte eindeutig references attachable_object(global_id)
kategorie_id	INTEGER		
prename	VARCHAR(255)	!	
surname	VARCHAR(255)	!	
birth_date	DATE		
tel_private	VARCHAR(255)		
tel_business	VARCHAR(255)		
tel_mobile	VARCHAR(255)		
a_zipcode	INTEGER		
a_city	VARCHAR(255)		
a_adress	VARCHAR(255)		
first_email	VARCHAR(255)		
second_email	VARCHAR(255)		

Lehrmittel

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
global_id	INTEGER	!1+PF	Globale ID, wird von einer Funktion in einer Helper-Klasse erzeugt und ist für alle anhängbaren Objekte eindeutig references attachable_object(global_id)
lehrmittel_kategorie_id	INTEGER	F	references lehrmittel_kategorie(id)
name	VARCHAR(255)		
description	VARCHAR(255)		
file_link	VARCHAR(255)		

Pruefung

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
global_id	INTEGER	!1+PF	Globale ID, wird von einer Funktion in einer Helper-Klasse erzeugt und ist für alle anhängbaren Objekte eindeutig references attachable_object(global_id)
lva_id	INTEGER	F	references lva(global_id)
examiner	VARCHAR(255)		
grade	INTEGER		between 1 and 5

Notiz

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
global_id	INTEGER	!1+PF	Globale ID, wird von einer Funktion in einer Helper-Klasse erzeugt und ist für alle anhängbaren Objekte eindeutig references attachable_object(global_id)
title	VARCHAR(255)		
note	VARCHAR(2000)		

Universitaet

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
id	INTEGER	!1+P	
name	VARCHAR(255)		

autoPSI::Architekturdokument

lehrmittel_kategorie

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
id	INTEGER	!1+P	
title	VARCHAR(255)		

attachable_object_kategorie

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
id	INTEGER	!1+P	
title	VARCHAR(255)		
description	VARCHAR(255)		
shareable	BOOLEAN	!	

Termin_Kategorie

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
id	INTEGER	!1+P	
name	VARCHAR(255)		

anhaengen_termincontainer

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
termincontainer_id	INTEGER	F[1]	references termincontainer(id)
global_id	INTEGER	F[1]	Eindeutiger Schlüssel über alle anhängbaren Objekte references attachable_object(global_id)
table_name	VARCHAR(255)		Datenbanktabelle, in der das angehängte Objekt liegt; prinzipiell nicht notwendig, allerdings besser für Performance, da dann nur diese Datenbanktabelle durchsucht werden muss

anhaengen_termin

<i>Feldname</i>	<i>Typ</i>	<i>Optionen</i>	<i>Bemerkung</i>
termin_id	INTEGER	F[1]	references termin(id)
global_id	INTEGER	F[1]	Eindeutiger Schlüssel über alle anhängbaren Objekte
table_name	VARCHAR(255)		Datenbanktabelle, in der das angehängte Objekt liegt; prinzipiell nicht notwendig, allerdings besser für Performance, da dann nur diese Datenbanktabelle durchsucht werden muss