

Diplomarbeit
LernAss 2003

Fachgebiet der Diplomarbeit:
Multimedia und Softwareentwicklung

Diplomand: Andreas Rehm

Referent: Professor Dr. rer. nat. Manfred Merkel

Koreferent: Professor Dr.-Ing. Wolf-Rainer Novender



Bereich Friedberg

Fachbereich IEM (FB 11)

Informationstechnik – Elektrotechnik – Mechatronik

Fachrichtung TIS (Technische Informatik-Systeme)

Sommersemester 2003

Eidesstattliche Versicherung

Ich versichere an Eides statt, die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe erstellt, und aus Veröffentlichungen entnommene Daten und Texte als solche kenntlich gemacht zu haben. Es wurden ausschließlich die angegebene Literatur sowie bereits erworbenes Fachwissen verwendet.

Andreas Rehm, Friedberg 06.08.2003

Dieses Dokument wurde mit L^AT_EX₂ ϵ gesetzt.

Verwendet wurden hierzu ein Linux-System mit Debian Woody (L^AT_EX), ein SuSE 8.0-Server (L^AT_EX) und ein SuSE 8.2 mit dem Vektorgrafikprogramm Dia und XFig (Flussdiagramme, Schaubilder). Die L^AT_EX-Vorlagen, alle benötigten Shell-Skripte und Makefiles wurden von mir in Eigenarbeit erstellt.

Mein Dank gilt meiner Familie und meinen Freunden für Ihre Unterstützung.

Großer Dank gilt vor allem Herrn Prof. Dr. Manfred Merkel, der mich seit meinem zweiten Semester kennt, und mir im Studium mit Rat und Tat zur Seite stand. Das LernAss Entwicklerteam habe ich durch Herrn Merkel im Jahr 1998 kennen gelernt.

Als ich das Team kennen lernte, waren die Entwicklung und die Diplomarbeit noch nicht abgeschlossen, Herr Prof. Dr. Manfred Merkel hat mir im Zuge dessen zum VLS-Projekt¹ in der Bibliothek der Fachhochschule-Friedberg verholfen.

Dank an das Team der FH-Bibliothek Friedberg für drei schöne Jahre als studentische Hilfskraft. Insbesondere Dank an Frau Andrea Thiel, Frau Bianca Klee, Frau Helga Berger, Frau Franziska Gottwalz und Frau Maria Haines, und alle studentischen Hilfskräfte.

Dank für mein BPS, das ich in der FH-Friedberg absolvieren konnte: die Installation eines Citrix MetaFrame Applikationsserver als Multimedia Lern-Server.

Ein großer Dank auch den beiden Entwicklern von LernAss – Jürgen Sommer und Arne Vortisch – die mein Interesse an Datenbanken und Webprogrammierung geweckt haben. Insbesondere danke ich Jürgen Sommer für seine Unterstützung bei der Programmierung mit VisualStudio .NET.

Vielen Dank auch an Rainer Frädlich², dem Systemadministrator des DVSZ³ der FH-Friedberg, für die unzähligen Fachgespräche und seine Unterstützung.

¹Virtueller Lesesaal der FH-Bibliothek Friedberg (1998-2001)

²<http://www.raifra.de>

³Datenverarbeitungs- und Servicezentrum

Inhaltsverzeichnis

0	Abkürzungen und Begriffe	1
0.1	Abkürzungen	1
0.2	Begriffe	2
1	Einleitung	4
1.1	Vorwort	6
1.2	Befragung	7
1.2.1	Gründe für eine Befragung	7
1.2.2	Fazit der Befragung	8
1.3	Vor- und Nachteile von LernAss	8
1.3.1	Analyse der Vor- und Nachteile	8
1.3.2	Fazit der Vor- und Nachteile	9
1.4	Plattformunabhängigkeit	9
1.5	Datenbanken	10
1.5.1	Grundlegendes	10
1.5.2	Relationale Datenbanken	10
1.5.3	Komponenten eines relationalen Datenbanksystems	11
1.5.4	Relational vollständige Sprache	11
1.5.5	Operatoren und SQL	12
1.5.6	Datenmanagement	13
1.6	Datenmodell	14
1.6.1	Grundlegendes	14
1.6.2	Generalisation	14
1.6.3	Aggregation	15
1.6.4	Assoziationstypen	15
1.6.5	Beziehungsmengen	15
1.6.6	Sieben Regeln zur Erstellung der Datenbank	16
1.6.7	Die Normalformen des Datenbankdesigns	17

1.6.8	Anwendung des Datenmodells und Tabellenstruktur der Datenbanken	18
1.6.9	Tabellenbeziehungen	24
1.7	Das Betriebssystem für den Server	25
1.8	Die Programmiersprache	26
1.9	Die Datenbank MySQL	26
1.9.1	Das Datenbanksystem	26
1.9.2	MySQL Provider für .NET	27
2	.NET	28
2.1	Grundlegendes	29
2.2	Gründe für .NET	29
2.2.1	Vorteile von .NET	30
2.2.2	Nachteile von .NET	31
2.3	Das .NET-Framework	32
2.4	Architektur	33
2.4.1	Microsoft Intermediate Language	33
2.4.2	Just in Time Compiler	34
2.4.3	Verwalteter Code gegen nicht verwalteten Code	34
2.4.4	Sicherer gegen unsicherer Code	35
2.4.5	Assemblies und der Global Assembly Cache	35
2.4.6	Common Language Runtime	36
2.4.7	Common Type System	36
2.4.8	Common Language Specification	36
2.5	Sicherheitsmechanismen	37
2.5.1	Unterteilung der Sicherheitsmechanismen	37
2.5.2	Konfiguration des Sicherheitssystems	37
2.6	Die Sprachen	38
2.6.1	VB.NET	38
2.6.2	ASP.NET	39
2.6.3	C#	39
2.7	.NET Services	40
2.7.1	Web-Services	40
2.7.2	.NET MyServices	40
2.7.3	.NET Passport	40
3	Lernsysteme	41
3.1	Einleitung	41
3.2	Was sind Lernsysteme?	42

3.3	Der PC als Hilfestellung	42
3.4	Die verschiedenen Lernsysteme	42
3.4.1	Verwendete Begriffe	42
3.4.2	Computer unterstützter Unterricht (CUU)	43
3.4.3	Intelligente tutorielle Systeme (ITS)	44
3.4.4	Der Unterschied von CBT und WBT	44
3.5	Motivation des Benutzers	46
4	Anforderungen an LernAss	47
4.1	Einleitung	47
4.2	Modulares Design	48
4.3	Vermittlung des Wissens	48
4.4	Aufgaben der Datenbank	48
4.5	Eingabe der Daten	49
4.6	Abfragen des Wissens	49
4.6.1	Multiple Choice	49
4.6.2	Lösungsschritte	49
4.6.3	Test und Klausurmodus	49
4.7	Arbeits erleichterungen	50
5	Technisches Redesign	51
5.1	Einleitung	52
5.2	Grafikformate	52
5.3	Frontend	54
5.4	Datenbearbeitung	54
5.5	Datenbank und Tabellen	54
5.5.1	Datenbankstruktur	55
5.5.2	Tabellenstruktur von LernAss	56
5.6	Programmstruktur	56
5.6.1	Datenbankverbindung	57
5.6.2	Der Sessioncache	58
5.6.3	Lernerfolgsüberwachung	58
5.7	Suchfunktion	58
5.8	Vom Word-Dokument zur Webgrafik	59
5.9	Datenbank-Sicherheitskonzept	61
6	LernAss Programme und Komponenten	63
6.1	Einleitung	65
6.2	LernAss Konverter	65
6.2.1	Aufbau	65

6.2.2	Bedienung	67
6.3	LernAss Web	69
6.3.1	Aufbau	69
6.3.2	Bedienung	71
6.4	LernAss Admin	78
6.4.1	Aufbau	78
6.4.2	Bedienung	78
6.5	LernAss Exerciser	86
6.5.1	Aufbau	86
6.5.2	Bedienung	86
6.6	LernAss Komponenten	97
6.6.1	ADODB_LIB	97
6.6.2	LA_EMF_Clip	97
6.6.3	MySQL_DB	97
6.6.4	MySQLClient	106
7	Installation des Servers	107
7.1	Vorwort	108
7.2	Windows 2000 Server	109
7.3	.NET Framework	109
7.4	MySQL	113
7.4.1	Installation	113
7.4.2	PHP	116
7.4.3	phpMyAdmin	121
7.4.4	Die Rechtevergabe mittels GRANT	122
7.4.5	Der SQL Code für die LernAss-Zentral-Datenbank	123
7.4.6	Der SQL Code für die LernAss-Aufgaben-Datenbank	124
7.4.7	Entfernen der GRANTS durch REVOKE	127
7.4.8	Einrichtung der Webapplikation	128
A	Flussdiagramme	I
A.1	Flussdiagramme	I
A.1.1	Aufgabeneingabe	III
A.1.2	LernAss Exerciser	IV
A.1.3	LernAss Konverter	V
A.1.4	LernAss Web	VI
A.2	Funktionsabläufe	VII
A.2.1	Datenbankfunktionen	VII
A.2.2	Aufgabenfunktionen	VIII

A.2.3	Zusatzdaten	VIII
A.2.4	Verweise	IX
A.2.5	Stichworte	IX
B	Grafiken für LernAss Web	X
B.1	Einleitung	X
B.2	Erstellung der Bilder	X
B.3	Bilderablage auf dem Server	XIV
C	Internet Links	XV
C.1	Allgemein	XV
C.2	E-Mailadressen	XV
C.3	.NET	XVI
C.4	Alternative Links zu .NET	XVI
C.5	Datenbanken	XVII
C.6	PHP	XVII
D	Adressen	XVIII
D.1	Andreas Rehm	XVIII
D.2	Fachhochschule Gießen-Friedberg	XVIII
D.3	Prof. Dr. rer. nat. Manfred Merkel	XIX
D.4	Prof. Dr.-Ing. Wolf-Rainer Novender	XIX
E	To do	XX
E.1	Einleitung	XX
E.2	Was man noch verbessern kann	XX
E.3	Was noch erledigt werden muss	XXI
F	Sourcecode	XXII
	Literaturverzeichnis	XXIII

Abbildungsverzeichnis

1.1	Beispiel zur Generalisation	14
1.2	Tabellenbeziehungen der Datenbank LernAss	24
1.3	Tabellenbeziehungen der Aufgabendatenbanken	25
3.1	Computer unterstützter Unterricht	44
3.2	Aufbau von WBT Systemen	46
5.1	Datenstruktur im Stern	56
5.2	Programmkomponenten	57
5.3	Datenbank Sicherheitskonzept	62
6.1	LernAss Konverter	68
6.2	LernAss Web-Startseite	71
6.3	LernAss Web Aufgabendatenbank Ansicht	72
6.4	LernAss Web-Aufgabenübersicht	73
6.5	LernAss Web Multiple Choice-Test	74
6.6	LernAss Web Multiple Choice-Test Auswertung	75
6.7	LernAss Web-Aufgabenausdruckfunktion	76
6.8	LernAss Web-Suchfunktion	77
6.9	LernAss Admin Anmelden/Abmelden	79
6.10	LernAss Admin Datenbanken-Übersicht	80
6.11	LernAss Admin Datenbanken anlegen	81
6.12	LernAss Admin Datenbanken ändern	82
6.13	LernAss Admin Link-Übersicht	83
6.14	LernAss Admin Links anlegen	84
6.15	LernAss Admin Links ändern	85
6.16	LernAss Exerciser anmelden/abmelden	87
6.17	LernAss Exerciser Aufgaben	88
6.18	LernAss Exerciser Aufgabenbearbeitung	89
6.19	LernAss Exerciser Aufgabenverlinkung	90

6.20	LernAss Exerciser Aufgabenlevel	91
6.21	LernAss Exerciser Fachgebiete	92
6.22	LernAss Exerciser Aufgabenliste	93
6.23	LernAss Exerciser Aufgabenlisten	94
6.24	LernAss Exerciser Stichworte	95
6.25	LernAss Exerciser Links	96
7.1	.NET-Framework-Installation beginnen?	109
7.2	Durchführung der .NET-Framework-Installation	109
7.3	Windows Installer Update	110
7.4	Installation des .NET-Frameworks	110
7.5	Installation und Kompilierung des .NET-Framework	110
7.6	Ende der .NET-Framework Installation	111
7.7	Initialisierung des .NET-Framework-Updates	111
7.8	.NET-Framework-Update durchführen?	111
7.9	EULA des .NET-Framework-Updates akzeptieren	112
7.10	Beginn des .NET-Framework-Updates	112
7.11	Ende des .NET-Framework-Updates	112
7.12	Nach dem Update Rechner neu starten	112
7.13	MySQL-Installation beginnen?	113
7.14	Hinweise zur MySQL-Installation	113
7.15	Auswahl des Verzeichnisses	114
7.16	Auswahl der MySQL-Installationsvariante	114
7.17	Beginn der MySQL Installation	115
7.18	Ende der MySQL-Installation	115
7.19	MySQL „myini.ini“ in WinMySQLAdmin	116
7.20	Beginn der PHP-Installation	117
7.21	Rechtliche Hinweise zur PHP-Installation	117
7.22	Auswahl des Installationsmodus der PHP-Installation	118
7.23	Auswahl des Verzeichnisses der PHP-Installation	118
7.24	Konfiguration der PHP-E-Mail-Anbindung	119
7.25	Auswahl des Webservers für PHP	119
7.26	Beginn der PHP-Installation	120
7.27	Installationsvorgang	120
7.28	Auswahl des Webservers der PHP-Installation	121
7.29	Ende der PHP-Installation	121
7.30	Applikationseinstellung bei den Internetinformationsdiensten	128
7.31	WEB-Applikation erstellen	129

A.1	Datenfluss in LernAss	II
A.2	Aufgabeneingabe	III
A.3	LernAss Exerciser	IV
A.4	LernAss Konverter	V
A.5	LernAss Web	VI
B.1	Öffnen der Bildvorlage in Paintshop Pro 5	XI
B.2	Knopf Zeichensatz auswählen	XII
B.3	Knopf Bezeichnung eingeben	XIII
B.4	Knopf speichern	XIV

Tabellenverzeichnis

0.1	Auflösung der Abkürzungen	1
0.2	Erklärung gängiger Begriffe	2
1.1	Die Operatoren in SQL	12
1.2	Die vier Eckpfeiler des Datenmanagements	13
1.3	Die Normalformen des Datenbankdesigns	18
1.4	Die Tabelle „LernAss.Index“	19
1.5	Die Tabelle „LernAss.Links“	19
1.6	Die Tabelle „Extra“	20
1.7	Die Tabelle „Hint“	20
1.8	Die Tabelle „Index“	21
1.9	Die Tabelle „Level“	21
1.10	Die Tabelle „Link“	22
1.11	Die Tabelle „ListExercises“	22
1.12	Die Tabelle „Listname“	22
1.13	Die Tabelle „Metadata“	22
1.14	Die Tabelle „MultipleChoice“	23
1.15	Die Tabelle „Solution“	23
1.16	Die Tabelle „Source“	23
1.17	Die Tabelle „Topics“	24
1.18	Die Tabelle „View“	24
2.1	Die wichtigsten Klassenbibliotheken des .NET-Frameworks	33
2.2	.NET-Sicherheitssystem – Konfigurationsdateien	38
5.1	Die Rechte in der Datenbank	62
6.1	LernAss V2.2 Schemadateiformat	67
6.2	Die Dateien von LernAss Web und ihre Funktion	70
6.3	Funktionsliste ADODB_LIB	97
6.4	Funktionsliste LA_EMF_Clip	97

6.5	MySQL_DB Datenbankverbindungsbehandlung	98
6.6	MySQL_DB WEB-Funktionen für die Datenbanken	98
6.7	MySQL_DB WEB-Link-Funktionen	98
6.8	MySQL_DB WEB-Aufgabenfunktionen	99
6.9	MySQL_DB WEB-Aufgabensuchfunktionen	99
6.10	MySQL_DB LernAss Admin-Datenbankfunktionen	100
6.11	MySQL_DB LernAss Admin / LernAss Exerciser-Linkfunktionen .	100
6.12	MySQL_DB LernAss Exerciser-Extrafunktionen	100
6.13	MySQL_DB LernAss Exerciser-Aufgabenfunktionen	101
6.14	MySQL_DB LernAss Exerciser-Aufgabenverlinkung	101
6.15	MySQL_DB LernAss Exerciser Aufgabenlisten	102
6.16	MySQL_DB LernAss Exerciser-Lösungen	102
6.17	MySQL_DB LernAss Exerciser Multiple Choice	103
6.18	MySQL_DB LernAss Exerciser Source Up-/Download	103
6.19	MySQL_DB LernAss Exerciser-Bildfunktionen	103
6.20	MySQL_DB LernAss Exerciser-Hintfunktionen	104
6.21	MySQL_DB LernAss Exerciser-Topicfunktionen	104
6.22	MySQL_DB LernAss Exerciser Levelfunktionen	104
6.23	MySQL_DB Generelle Ausleseoperationen	105
6.24	MySQL_DB Datenbankrechteverwaltung	105
6.25	MySQL_DB Basisdatenbankbearbeitungsoperationen	105
7.1	Server-Software – Voraussetzungen	109

0 Abkürzungen und Begriffe

0.1 Abkürzungen

Tabelle 0.1: Auflösung der Abkürzungen

Abk.	Auflösung
ANSI	American National Standards Institute
API	Application Programming Interface – Programmierschnittstelle
ASP	Active Server Pages
C#	Csharp – C-Abkömmling von Microsoft
CBT	Computer Based Training
CLR	Common Language Runtime
CLS	Common Language Specification
CMI	Computer Managed Instruction – Computer gestütztes Lernen
CUU	Computer unterstützter Unterricht
FH	Fachhochschule
GAC	Global Assembly Cache
GC	Garbage Collector – Müllsammler
HTTP	Hyper Text Transfer Protocol
HTML	Hyper Text Markup Language
IIS	Internet Information Server
IP	Internet Protocol – das Netzwerkprotokoll
ISO	International Standardization Organization
ITS	Intelligente tutorielle Systeme
JIT	Just in Time compiler
LAMP	Linux Apache MySQL PHP
LAN	Local Area Network – lokales, standortgebundenes Netzwerk
Fortsetzung auf der nächsten Seite	

Abk.	Beschreibung
NF	Normalform – Normalisierungsgrad einer Datenbank
MS	Microsoft
MSIL	Microsoft Intermediate Language
NF	Normalform
PC	Personal Computer
PHP	Personal Hypertext Preprocessor
QBE	Query By Example
QUEL	Query Language
RDBMS	Relationale Datenbank Management Systeme
RTF	Rich Text Format
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
SQL	Structured Query Language – Strukturierte Abfragesprache
VB	VisualBasic
VC	VisualC
VLS	Virtueller Lesesaal
WAN	Wide Area Network – großes Standort übergreifendes Netzwerk
WBT	Web Based Training

0.2 Begriffe

Tabelle 0.2: Erklärung gängiger Begriffe

Begriff	Beschreibung
case sensitive	Groß- und Kleinschreibung wird unterschieden
Compiler	Übersetzer – vom Quellcode zum Pogramm in einem bestimmten Format für Computer
Intranet	Ein normales LAN zum Datenaustausch wird auch Intranet genannt, sofern keine direkte oder gar keine Verbindung zum Internet besteht (VPN oder NAT ist keine direkte Verbindung)
Internet	Das weltweite Netzwerk - ein großer Verbund von Rechnern zu einem WAN mit öffentlichem Zugang
Patch	Pflaster – Eine kleine Fehlerbehebung in der Software
Fortsetzung auf der nächsten Seite	

Begriff	Beschreibung
\$USERDIR\$	Ersatz für „Dokumente und Einstellungen“ bei Windows 2000 und XP oder für Windows NT4 %WINDIR%\profiles
%WINDIR%	Variable für den Pfad zum Windowsverzeichnis

1 Einleitung

Inhaltsangabe

1.1	Vorwort	6
1.2	Befragung	7
1.2.1	Gründe für eine Befragung	7
1.2.2	Fazit der Befragung	8
1.3	Vor- und Nachteile von LernAss	8
1.3.1	Analyse der Vor- und Nachteile	8
1.3.2	Fazit der Vor- und Nachteile	9
1.4	Plattformunabhängigkeit	9
1.5	Datenbanken	10
1.5.1	Grundlegendes	10
1.5.2	Relationale Datenbanken	10
1.5.3	Komponenten eines relationalen Datenbanksystems	11
1.5.4	Relational vollständige Sprache	11
1.5.5	Operatoren und SQL	12
1.5.6	Datenmanagement	13
1.6	Datenmodell	14
1.6.1	Grundlegendes	14
1.6.2	Generalisation	14
1.6.3	Aggregation	15
1.6.4	Assoziationstypen	15
1.6.5	Beziehungsmengen	15
1.6.6	Sieben Regeln zur Erstellung der Datenbank	16
1.6.7	Die Normalformen des Datenbankdesigns	17

1.6.8	Anwendung des Datenmodells und Tabellenstruktur der Datenbanken	18
1.6.9	Tabellenbeziehungen	24
1.7	Das Betriebssystem für den Server	25
1.8	Die Programmiersprache	26
1.9	Die Datenbank MySQL	26
1.9.1	Das Datenbanksystem	26
1.9.2	MySQL Provider für .NET	27

Abbildungen

1.1	Beispiel zur Generalisation	14
1.2	Tabellenbeziehungen der Datenbank LernAss	24
1.3	Tabellenbeziehungen der Aufgabendatenbanken	25

Tabellen

1.1	Die Operatoren in SQL	12
1.2	Die vier Eckpfeiler des Datenmanagements	13
1.3	Die Normalformen des Datenbankdesigns	18
1.4	Die Tabelle „LernAss.Index“	19
1.5	Die Tabelle „LernAss.Links“	19
1.6	Die Tabelle „Extra“	20
1.7	Die Tabelle „Hint“	20
1.8	Die Tabelle „Index“	21
1.9	Die Tabelle „Level“	21
1.10	Die Tabelle „Link“	22
1.11	Die Tabelle „ListExercises“	22
1.12	Die Tabelle „Listname“	22
1.13	Die Tabelle „Metadata“	22
1.14	Die Tabelle „MultipleChoice“	23
1.15	Die Tabelle „Solution“	23
1.16	Die Tabelle „Source“	23
1.17	Die Tabelle „Topics“	24
1.18	Die Tabelle „View“	24

1.1 Vorwort

Der Studiengang Elektrotechnik entstammt dem Fachbereich Elektrotechnik II an der FH Gießen-Friedberg. Mein Studien-Schwerpunkt ist „Technische Informatik-Systeme“. Mein Werdegang im Studium ist unter anderem durch die Hilfe von Herrn Prof. Dr. Manfred Merkel und Frau Dipl.-Bib. Thiel gezeichnet, durch die ich an das VLS-Projekt in der FH-Bibliothek Friedberg gekommen bin.

In der Bibliothek der FH-Friedberg habe ich in meiner dreijährigen Tätigkeit den VLS eingerichtet. Mein Wissen über die Programmierung und Anwendung von PC basierten Systemen konnte ich hierbei stetig erweitern. Ich bin seit 1999 an UNIX Systemprogrammierung sowie LAMP, Webprogrammierung und Datenbanken interessiert und engagiert.¹

Als Mitglied der FriLUG (Friedberger Linux User Group) versuche ich, anderen Studierenden bei PC Problemen und Fragen zu Linux (aber auch Windows) mit Rat und Tat zur Seite zu stehen. Teile meiner Software dienen hier dem Allgemeinwohl.

Diese Diplomarbeit beruht auf dem ursprünglichen LernAss-System [1] von Jürgen Sommer und Arne Vortisch. Als Stichwort für diese Diplomarbeit wurde Multimedia genannt. Zum Zeitpunkt der Entwicklung von LernAss waren die Ziele noch nicht endgültig definiert; daher sind Unstimmigkeiten und Redundanzen im Design entstanden, die im Zuge einer Softwareaktualisierung und dem damit verbundenen Redesign behoben werden sollen.

Das Ziel dieser Diplomarbeit ist, das CBT-System LernAss auf den aktuellen Softwarestand zu bringen und die Integration in ein CMI-System vorzubereiten.

In der heutigen Zeit ist Multimedia ein fester Bestandteil der PC-Industrie. Der Kampf um das beste Multimedia-Angebot ist im Moment zurückgestellt, da sich das Windows-Betriebssystem gegen freie Betriebssysteme durchsetzen will. Darüber streitet sich die PC-Gemeinde, andere „Schauplätze“ sind momentan unwichtig. Die Entwicklung im Betriebssystemsektor ist dadurch dauernd anwachsend, was unter anderem auch zur anstehenden Software-Aktualisierung beiträgt. Ein Software-System, das man heute einsetzt, ist morgen schon wieder veraltet. Leider trifft das zum Zeitpunkt der Diplomarbeit auch schon für VisualStudio .NET zu, das Mitte 2003 als Version „VisualStudio .NET 2003“ (Version 1.1) auf den Markt gekommen ist.

Wichtige Schlagworte für die Softwareentwicklung sind heute nicht mehr nur Multimedia, einfache Bedienung und Stabilität, sondern vielmehr auch Plattformun-

¹Siehe <http://www.andreasrehm.de>

abhängigkeit und Sicherheit. Die Plattformunabhängigkeit hängt bei der Entwicklung von speziellen Anwendungen stark von der Sprache ab. (Siehe dazu das Kapitel 1.4 auf Seite 9.)

Es ist heute ferner nicht mehr unbedingt die Frage zu stellen, ob die Studierenden Zugang zu Onlinepublikationen haben, da die Vernetzung mit dem Internet bis in die Privathaushalte Einzug gehalten hat und bei Studierenden mittlerweile zur Standardausstattung gehört.

1.2 Befragung

1.2.1 Gründe für eine Befragung

Zu Beginn der Diplomarbeit sollten eine Befragung und eine Analyse dieser Befragung stehen. Hierzu wurden die Benutzer von LernAss befragt, wie ihnen das Programm gefällt und was man verbessern kann. Die Benutzer von LernAss sind die Studierenden, die das Wissen/die Aufgaben von LernAss nutzen, um z.B. das Gelernte besser zu verstehen oder um für die bevorstehenden Klausuren zu lernen.

Die Dateneingabe wird von den Dozenten und den dafür eingestellten studentischen Hilfskräften durchgeführt. Deswegen wurden die Dozenten mit einem auf sie angepassten Bogen befragt.

Die von mir erstellten Fragebögen wurden über die Sekretariate an alle Dozenten verteilt. Den Studierenden wurden 300 Fragebögen in Vorlesungen ausgegeben. Leider war die Resonanz sehr gering.

Nur vier Fragebögen wurden abgegeben. Drei davon sind aus direktem Ansprechen von Bekannten entstanden. Aus den beantworteten Fragebögen gingen nur wenige Forderungen hervor, die durch schon vorher laut gewordene Forderungen ergänzt wurden.

Das Ergebnis dieser Befragung habe ich ausgewertet und in die Diplomarbeit einbezogen, sofern man aussagekräftige Punkte extrahieren konnte. Im Prinzip sind bereits durch Herrn Prof. Dr. Merkel und das LernAss Entwicklerteam bekannte Forderungen mit dieser Diplomarbeit wieder aufgenommen worden.

1.2.2 Fazit der Befragung

Es sind folgende Dinge aufgefallen:

- Studierende und Dozenten tun sich schwer mit Fragebögen, sofern sie denn überhaupt beantwortet werden
- Die meisten Dozenten und Studierende wollen nicht angesprochen oder befragt werden, oder wollen ihre Meinung nicht preisgeben
- Informationen sind nur schwer zu bekommen
- Das Interesse an Entwicklung und Forschung ist (trotz oft geäußerte Kritik über nicht wunschgemäße Dinge) sehr gering

Mit dem Redesign von LernAss sollten folgende Punkte erfüllt werden:

- Ausdruckmöglichkeit
- Erweiterte Aufgabensuche
- Querverweise der Aufgaben untereinander
- Verweise auf Skriptstellen
- Zusatzmaterial zu Aufgaben (Links, Audio, Video, Text).

1.3 Vor- und Nachteile von LernAss

1.3.1 Analyse der Vor- und Nachteile

Die Vorteile von LernAss:

- Zentraler Wartungspunkt für Updates des Lernstoffs und der Software
- Durch die Verwendung des Internets weltweite Verfügbarkeit
- Suchmöglichkeiten mit verschiedenen Kriterien
- Auswahllisten.

Die nicht so schwer wiegenden Nachteile:

- Die Datenbank ist zu alt (Access 2.0) und langsam
- LernAss ist nicht modular aufgebaut (Kernkomponenten sind nicht wiederverwendbar)
- Es gibt keine Querverweise zwischen den Aufgaben
- Internet-Version und Offline-Version können differieren
- Die Offline-Version hat ein anderes Design als die Online-Version (Wiedererkennungseffekt ist nicht gegeben).

1.3.2 Fazit der Vor- und Nachteile

LernAss muss eine einheitliche und strukturierte Oberfläche bekommen. Am einfachsten und effizientesten geschieht dies über eine Weboberfläche.

Die Datenbankengine unter LernAss sollte austauschbar sein → Die Programmteile von LernAss müssen einfach auf eine andere Datenbank umzurüsten sein. In Folge meiner Machbarkeitsstudie hat sich die Einführung von Namespaces in der VisualStudio .NET Entwicklungsumgebung als sinnvoll erwiesen. Hiermit ist unter anderem auch die Entscheidung für die Sprache VisualBasic .NET gefallen.

Die Aufgaben müssen zentral verwaltet und dargestellt werden. Eine Plugin-Schnittstelle für spezielle Funktionen und Erweiterungen sollten in Betracht gezogen werden.

Das Grundprinzip eines CBT ist die Speicherung der Daten in einer Datenbank und Visualisierung über eine ansprechende Benutzeroberfläche.

1.4 Plattformunabhängigkeit

Die Plattformunabhängigkeit ist bei der Konzeption eines CBT-Systems nur sehr schwer realisierbar. Die Standardisierung der Datenformate für Office-Dokumente ist noch nicht weit genug fortgeschritten, obgleich man mit XML schon sehr gute Mittel in der Hand hat und RTF ein Standard ist.

Auf der Client-Seite des LernAss-Systems wird eher Microsoft-Software verwendet; dies ist im Sinne des alten LernAss Word mit eingebetteten Objekten. Diese Daten

muss man durch eine Datenkonvertierung zu XML oder einem anderen tauglichen Grafikformat internetfähig machen.

Das eigentliche Datenbanksystem könnte auf jeder möglichen Plattform liegen, genauso wie der darstellende Webserver auf jeder möglichen Hard- und Softwarekombination aufgesetzt werden könnte. Im Wesentlichen müssen die Daten nur richtig kodiert werden, um vom System unabhängig zu sein.

1.5 Datenbanken

1.5.1 Grundlegendes

Die Theorien zum Thema Datenbanken stammen aus dem Buch [2] „Relationale Datenbanken“. In diesem Kapitel möchte ich nur auf die relationalen Datenbanken eingehen, da diese heute die grösste Verbreitung finden.

Das älteste Datenbankmodell ist das hierarchische. In diesem werden die Daten in einer Reihe von Datensätzen speichert, mit denen verschiedene Felder verknüpft sind. Die Instanzen eines bestimmten Datensatzes werden als Datensatzabbild zusammengefasst. Diese Datensatzabbilder sind vergleichbar mit den Tabellen einer relationalen Datenbank.

Verknüpfungen zwischen den Datensatzabbildern werden in hierarchischen Datenbanken als Eltern-Kind-Beziehungen (Parent-Child Relationships) realisiert, die in einer Baumstruktur abgebildet werden. Der Nachteil von hierarchischen Datenbanken ist, dass sie nur mit einem solchen Baum umgehen können. Verknüpfungen zwischen verschiedenen Bäumen oder über mehrere Ebenen innerhalb eines Baumes sind nicht möglich.

Das hierarchische Modell wird nicht mehr als Grundlage für die Entwicklung von Datenbanken verwendet.

1.5.2 Relationale Datenbanken

Bei relationalen Datenbanken speichert man Daten und Beziehungen in benannten Tabellen. Die Tabellen sind mit eindeutigen Merkmalen oder Attributen (attributes) versehen, die die Spalten benennen. Neben den Benutzertabellen existieren auch Systemtabellen, die Beschreibungsinformationen enthalten.

Die Verwaltungskomponente enthält eine Datendefinitions- und -manipulationssprache, wie z.B. SQL (die Sprache umfasst weiterhin auch Funktionen zur Gewährung der Datenintegrität, der Datensicherung und des Datenschutzes).

Daten werden in bestimmten Datenbereichen (domains) gespeichert. Schlüsselmerkmale dienen zur eindeutigen Identifizierung eines Datensatzes.

Merkmalsnamen und Tabellennamen sind eindeutig. Die Anzahl an Tabellen und Merkmalen ist unbegrenzt und die Anordnung der Spalten in den Tabellen ist beliebig. Im Allgemeinen ist bei relationalen Datenbanken die referenzielle Integrität gewährleistet. Das heißt, dass Datensätze, die von mindestens einem anderen Datensatz über einen Fremdschlüssel referenziert sind, nicht einfach gelöscht werden können. Für diesen Fall sind der Datenbank bei der Datendefinition bestimmte Operationen vorzugeben.

1.5.3 Komponenten eines relationalen Datenbanksystems

Prinzipiell werden RDBMS in zwei Teile aufgeteilt:

- Speicherkomponente
 - Daten und Datenbeziehungen in Form von Tabellen
 - Beschreibungsdaten und statistische Angaben (Systemtabellen)
- Verwaltungskomponente
 - relationale Datendefinitions- und Datenmanipulationssprache
 - zentrale Funktionen für Reorganisation, Schutz, Sicherheit etc.

1.5.4 Relational vollständige Sprache

Zitat aus [2] (Seite 68, Kapitel 3.3, vorletzter Absatz): „Eine relationale Abfragesprache heißt vollständig im Sinne der Relationenalgebra, wenn sie mindestens die mengenorientierten Operatoren Vereinigung, Subtraktion und kartesisches Produkt sowie die relationenorientierten Operatoren Projektion und Selektion unterstützt.“

Vollständige relationale Sprachen sind SQL², QUEL³ und QBE⁴. SQL ist die bekannteste und am häufigsten verwendete Sprache, sie ist stark beschreibend (an die

²In den 70er Jahren für „System R“ definiert, durch ANSI und ISO normiert

³In den 70er Jahren für das Datenbanksystem Ingres definiert.

⁴Anschauliche Auswertungssprache z.B. in ACCESS verwendet

Sprache angelehnt) und daher leicht verständlich. Außerdem ist SQL für komplexe Datenbankabfragen am besten geeignet.

1.5.5 Operatoren und SQL

Die Beschreibungssprache SQL soll hier nicht erklärt werden. Der Umfang aller möglichen Konstrukte und Befehle sowie der verschiedenen Versionen der Sprache würde den Rahmen dieser Diplomarbeit sprengen.

Tabelle 1.1: Die Operatoren in SQL

Typ	Symbol	SQL Befehlsbeispiel
Vereinigung	$T_1 \cup T_2$	SELECT * FROM T ₁ UNION SELECT * FROM T ₂
Subtraktion	$T_1 \setminus T_2$	
kartesisches Produkt	$T_1 \times T_2$	SELECT * FROM T ₁ ,T ₂
Projektion	$\pi_M(T)$	SELECT M FROM T
Selektion	$\sigma_F(T)$	SELECT * FROM T WHERE F
Nicht vorausgesetzt:		
Durchschnitt	$T_1 \cap T_2$	
Division	$T_1 \div T_2$	
Verbund	$T_1 \bowtie_P T_2$	SELECT * FROM T ₁ ,T ₂ WHERE P

Tabelle = T, Menge = M, Formel = F, Prädikat = P

1.5.6 Datenmanagement

Tabelle 1.2: Die vier Eckpfeiler des Datenmanagements

	Ziele	Werkzeuge
Datenarchitektur	Formulieren und Pflegen des unternehmensweiten Datenmodells, Unterstützen der Anwendungsentwicklung bei der Datenmodellierung	Datenanalyse und Entwurfsmethodik, Werkzeuge für die rechnergestützte Datenmodellierung
Datenadministration	Verwalten von Daten und Funktionen anhand von Standardisierungsrichtlinien und internationalen Normen, Beraten von Entwicklern und Endbenutzern	Data Dictionary-Systeme, Werkzeuge für den Verwendungsnachweis
Datentechnik	Installieren, Reorganisieren und Sicherstellen von Datenbanken, Durchführen von Datenbankrestaurierungen nach einem Fehlerfall	Datenbankverwaltungssysteme, Hilfsmittel zur Wiederherstellung von Datenbanken und zur Leistungsoptimierung
Datenbenutzung	Bereitstellen von Auswertungs- und Reportfunktionen unter Berücksichtigung des Datenschutzes respektive der Dateneignerschaft	Sprache für Datenbankabfragen und -manipulationen, Reportgeneration

Entnommen aus: „Relationale Datenbanken“ [2] (Seite 11, Abb. 1-6.)

Das Datenmanagement verursacht in der Regel hohe Kosten. Widerspruchsfreie und verständliche Datenmodelle sind schwer zu erstellen und benötigen viel Zeit. Enorm wichtig sind dabei konsistente Datenbestände und gute Sicherheitskonzepte. Firmen sollten bei der Betrachtung der Investitionen in ein Datenmodell die Wirtschaftlichkeit nicht ausser Acht lassen, wobei auch die Überlegung der Lebensdauer der Datenbestände beachtet werden sollte.

Zitat aus [2] (Seite 11, Kapitel 1.4, letzter Absatz): „Unter dem Datenmanagement fasst man alle betrieblichen und technischen Funktionen der Datenarchitektur, der Datenadministration und der Datentechnik zusammen, die der unternehmensweiten Datenhaltung, Datenpflege und Datennutzung dienen.“

1.6 Datenmodell

1.6.1 Grundlegendes

Zitat aus [2] (Seite 13, Kapitel 2.1): „Ein Datenmodell (engl. data model) beschreibt auf strukturierte und formale Art die für ein Informationssystem notwendigen Daten und Datenbeziehungen.“

Ein Datenmodell benötigt Datenklassen (Entitätsmengen) und Beziehungen der Daten zueinander.

Zur Festlegung eines Datenbanklayouts dienen drei Schritte:

- Datenanalyse
- Entwurf des Entitätenbeziehungsmodells
- Überführung in ein relationales Datenbankschema

1.6.2 Generalisation

Die Generalisation ist ein Abstraktionsvorgang, bei dem man Entitätsmengen in „neue“ untergeordnete Sub-Entitäten einteilt. Die so erstellten Sub-Entitäten können als eine Art Spezialisierung interpretiert werden.

Als Beispiel wird nachfolgend die wichtigste Tabelle des LernAss-Systems dokumentiert:

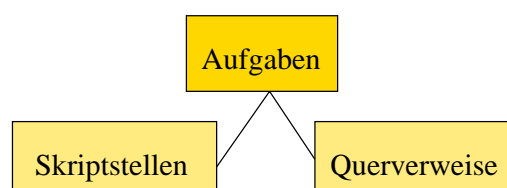


Abbildung 1.1: Beispiel zur Generalisation

1.6.3 Aggregation

Mit der Aggregation fasst man die durch die Generalisation aufgeteilten Entitätsmengen wieder zu einer übergeordneten Entität zusammen.

1.6.4 Assoziationstypen

Die Entitätsmengen können vier Assoziationstypen zugeordnet werden, gegeben seien hier zwei Entitätsmengen EM1 und EM2:

- Typ 1: genau ein
Jeder Entität aus der Entitätsmenge EM1 ist eine Entität der Entitätsmenge EM2 zugeordnet → einfache Assoziation
- Typ c: kein oder ein
Jeder Entität aus der Entitätsmenge EM1 ist eine oder keine Entität der Entitätsmenge EM2 zugeordnet → konditionelle Assoziation
- Typ m: mehrere
Jeder Entität aus der Entitätsmenge EM1 sind mehrere Entitäten der Entitätsmenge EM2 zugeordnet → komplexe Assoziation
- Typ mc: kein, ein oder mehrere
Jeder Entität aus der Entitätsmenge EM1 sind mehrere Entitäten der Entitätsmenge EM2 zugeordnet → mehrfach konditionelle Assoziation.

1.6.5 Beziehungsmengen

Die Beziehungsmengen sind die Beschreibung der Generalisation und Aggregation mit Assoziationstypen. Durch die Zusammenfassung von zwei Assoziationstypen entsteht eine Mächtigkeit.

Man unterscheidet zwischen den folgenden Beziehungsmengen:

- einfach-einfach
Beide Assoziationstypen sind entweder einfach (1) oder konditionell (c) Mächtigkeiten: (1,1), (1,c), (c,1) und (c,c)

- einfach-komplexe

Beziehung zwischen einem einfachen (1, c) und einem komplexen (m, mc) Assoziationstyp. Mächtigkeiten: (1,m), (1,mc), (c,m), (c,mc), (m,1), (m,c), (mc,1) und (mc,c)

- komplex-komplex

Beide Assoziationstypen sind komplex. Mächtigkeiten: (m,m), (m,mc), (mc,m) und (mc,mc).

1.6.6 Sieben Regeln zur Erstellung der Datenbank

1. (Entitätsmengen)

Entitätsmengen werden in eigenen Tabellen gespeichert, die mit einem eindeutigen Primärschlüssel zu versehen sind. Als Schlüsselkandidat dürfen nur eindeutige und minimale Schlüssel (geringe Redundanz) dienen.

2. (Beziehungsmengen)

Beziehungsmengen können als eigenständige Tabelle gespeichert werden. Der Identifikationsschlüssel der Entitätsmenge tritt als Fremdschlüssel⁵ auf. Als Primärschlüssel ist eine Kombination der Fremdschlüssel oder ein eigener Schlüssel denkbar.

3. (komplex-komplexe Beziehungen)

Komplex-komplexe Beziehungsmengen müssen als eigene Tabelle definiert werden. Die Identifikationsschlüssel der Entitätsmengen treten als Fremdschlüssel auf. Als Primärschlüssel ist eine Kombination der Fremdschlüssel oder ein eigener Schlüssel denkbar.

4. (einfach-komplexe Beziehungen)

Einfach-komplexe Beziehungsmengen können mit den beiden verknüpften Tabellen ohne eine eigene Beziehungsmengentabelle dargestellt werden. Dazu speichert man in der Tabelle mit der einfachen Assoziation (1 oder c) einen Fremdschlüssel auf die referenzierte Tabelle, gegebenenfalls mit weiteren Merkmalen der Beziehungsmenge.

5. (einfach-einfache Beziehungen)

Einfach-einfache Beziehungen können direkt in den zu verknüpfenden Tabellen als Fremdschlüssel gespeichert werden. Eine Beziehungstabelle ist nicht notwendig.

⁵Ein Fremdschlüssel ist ein Merkmal oder eine Merkmalskombination, die in derselben oder einer anderen Tabelle als Identifikationsschlüssel auftritt.

6. (Generalisation)

Entitätsmengen sind in eigenen Tabellen gespeichert, „teilen“ sich aber den Primärschlüssel

7. (Aggregation)

Entitäts- und Beziehungsmengen sind in eigenen Tabellen, wenn der komplex-komplexe Beziehungstyp vorliegt. Bei einer einfach-komplexen Beziehung können die Entitäts- und die Beziehungsmenge zu einer einzigen Tabelle kombiniert werden.

1.6.7 Die Normalformen des Datenbankdesigns

Datenbanken haben im Allgemeinen mehrere Fallstricke und Probleme, die mit bestimmten Definitionen namentlich genannt werden können:

Mutationsanomalien:

- Einfüge-Anomalie: Neue Daten stellen neue Informationen bereit, die noch nicht eindeutig sind
- Lösch-Anomalie: Ungewolltes Verloren gehen von Information, durch Löschen der letzten vorhandenen Information.

Änderungsanomalien:

- Änderungsanomalie: Änderung eines Datensatzes ohne Verlust der Information, aber durch mehrfaches Vorhandensein ergeben sich unterschiedliche Benennungen eines eindeutigen Datensatzes.

Mit den Normalformen sollen Anomalien und damit auch Redundanzen⁶ vermieden werden. Durch die Anwendung des Entitätenbeziehungsmodells (Kapitel 1.6.6 Seite 16) hat man in der Regel schon die ersten drei Normalformen erfüllt.

Die Normalformen (siehe Tabelle 1.3 Seite 18) sind bis zur fünften Normalform definiert; in der Praxis werden sie aber meist nur bis zur dritten Normalform angewendet, da die vierte und fünfte Normalform nur in sehr speziellen Fällen auftritt. Im Redesign von LernAss ist für die Stichworte (siehe Kapitel 1.3 Seite 18) eine Tabelle in der vierten Normalform erforderlich, daher wird diese hier auch aufgeführt.

⁶Redundanz liegt vor, wenn Daten weggelassen werden können, ohne Informationen zu verlieren.

Tabelle 1.3: Die Normalformen des Datenbankdesigns

NF	Bedingungen
erste	Alle Merkmalswerte sind atomar, es sind keine Wiederholungen zugelassen. Es dürfen keine Mengen, Aufzählungstypen oder Wiederholungsgruppen vorkommen.
zweite	Die erste Normalform muss erfüllt sein, und jedes Nicht-Schlüsselmerkmal ist von jedem Schlüssel voll funktional abhängig. Redundanz durch Aufteilen in weitere Tabellen beheben. Die Merkmale sind ohne die Schlüssel zu den ausgelagerten Daten nicht mehr mit der vollständigen Information behaftet. Abhängigkeit ein Schlüssel (S) \rightarrow Merkmal (B) $=\rightarrow B$ oder mehrere Schlüssel (S1, S2) $\rightarrow B$.
dritte	Die zweite Normalform muss erfüllt sein, und kein Nicht-Schlüsselmerkmal ist von irgendeinem Schlüssel transitiv* abhängig. Abhängigkeit Schlüssel (A) zu Schlüssel (B) und Schlüssel zu Merkmal (C) $A \rightarrow B$ und $B \rightarrow C$. Die Abhängigkeit $A \rightarrow C$ ist funktional und transitiv, wenn neben den Abhängigkeiten $A \rightarrow B$ und $B \rightarrow C$ A nicht funktional von B abhängt. Zitat aus [2] (Seite 40, Kapitel 2.4.3): „Das Merkmal C ist transitiv abhängig von A, falls B funktional abhängig von A, C funktional abhängig von B und nicht gleichzeitig A funktional abhängig von B ist.“
vierte	Mehrwertige Abhängigkeiten bei Nicht-Schlüsselmerkmalen werden in einer anderen Tabelle gespeichert. Zitat aus [2] (Seite 42, Kapitel 2.4.4): „Ein Merkmal C ist mehrwertig abhängig vom Merkmal A (ausgedrückt durch die Schreibweise $A \rightarrow \rightarrow C$), falls zu jeder Kombination eines bestimmten Wertes aus A mit einem beliebigen Wert aus B eine identische Menge von Werten aus C erscheint.“

* über Umwege funktional abhängig

1.6.8 Anwendung des Datenmodells und Tabellenstruktur der Datenbanken

Das Datenmodell wird in LernAss wie folgt angewandt:

In der Steuerdatenbank (Datenbank „LernAss“ Kapitel 1.6.8 auf Seite 19) befinden sich nicht redundante Tabellen, die keiner Optimierung bedürfen. Dies sind „LernAss.Index“ und „LernAss.Links“. In „LernAss.Index“ werden die Aufgabendatenbanken verwaltet und in „LernAss.Links“ Internet URLs, wobei beide Tabellen nicht

voneinander abhängen und auch keinerlei redundante Informationen beinhalten. Die Tabellen entsprechen der ersten Normalform.

Tabellen der Datenbank „LernAss“

Tabelle 1.4: Die Tabelle „LernAss.Index“

Feld	Typ	Beschreibung
Key_Index	int(10)	Primärschlüssel
DB_activated	tinyint(1)	Kennzeichnung ob aktiv oder nicht
DB_Info	text	Beschreibung der Datenbank
DB_User	text	Benutzername zum Zugriff auf die Datenbank
DB_Password	text	Passwort zum Zugriff auf die Datenbank
DB_Location	text	Standort der Datenbank
DB_Name	text	Der Name der Datenbank

Tabelle 1.5: Die Tabelle „LernAss.Links“

Feld	Typ	Beschreibung
Key_Link	int(10)	Primärschlüssel
Link_Name	text	Name des Links
Link_Shortinfo	text	kurze Info zum Link
Link_URL	text	URL des Links
Link_Picture_URL	text	URL des Bildes zum Link

Bei „LernAss.Index“ ist unter Umständen eine Optimierung nach der zweiten Normalform möglich, da z.B. die Benutzer-, Passwörter- und Location-Datensätze Wiederholungen unterliegen können. In der Regel ist die „Location=localhost“. Damit wäre hier eine Redundanz, wobei sich diese Redundanz nicht wirklich schwer wiegend auswirkt. Eine Optimierung an dieser Stelle lohnt sich wegen dem zu hohen Optimierungsaufwand und zu geringem Nutzen nicht.

Es werden keine speziellen Suffixe für Tabellennamen verwendet. Die Benennung der Tabellen und Felder ist nicht „case sensitive“. Dennoch sollten sicherheitshalber die hier angegebenen Namen verwendet werden, sofern man eine Datenbank im case sensitive-Modus betreibt (Beispiel MySQL unter Unix, bei Windows sind nur die Feldnamen „case sensitive“).

Die Schlüsselmerkmale der Tabellen werden mit dem Präfix „Key_“ versehen.

In den Aufgabendatenbanken werden folgende Informationen gespeichert:

Tabelle 1.6: Die Tabelle „Extra“

Feld	Typ	Beschreibung
Key_Extra	int(10)	Primärschlüssel
Key_Exercise	int(10)	Verknüpfung mit den Aufgaben
Extra_URL	text	Die URL zu den Extradaten
Extra_Name	text	Der anzuzeigende Name der Extradaten

In der Tabelle „Extra“ werden URLs gespeichert, die für die Aufgaben als sinnvolle Ergänzung dienen sollen.

Tabelle 1.7: Die Tabelle „Hint“

Feld	Typ	Beschreibung
Key_Hint	text	Primärschlüssel in vierter NF
Hint_Hints	int(10)	Zugehöriges Stichwort

In der Tabelle „Hint“ werden Stichwörter und mögliche Ersatzstichwörter gespeichert.

Tabelle 1.8: Die Tabelle „Index“

Feld	Typ	Beschreibung
Key_Exercise	int(10)	Primärschlüssel
Key_Exercise_ID	text	Vom Administrator änderbare eigene Bezeichnung einer Aufgabe
Key_Source	int(10)	Source-Verknüpfung
Key_Topic	int(10)	Topic-Verknüpfung
Key_Hint	text	Hint-Verknüpfung
Key_View	int(10)	View-Verknüpfung
Key_Level	tinyint(3)	Level-Verknüpfung
Exercise_Aktiv	tinyint(1)	Boolean Wert, ob Aufgabe angezeigt werden darf
Exercise_Info	text	Informationstext zur Aufgabe
Exercise_Show_Solution	tinyint(1)	Boolean Wert, ob die Lösungen angezeigt werden dürfen
Exercise_Show_MultipleChoice	tinyint(1)	Boolean Wert, ob die Multiple Choice Lösungen angezeigt werden dürfen
Exercise_Show_Extra	tinyint(1)	Boolean Wert, ob die Extras angezeigt werden dürfen

In der Tabelle „Index“ werden die Aufgaben mit ihren zugehörigen Daten verknüpft. Hier werden auch die Anzeigoptionen gesetzt. Gespeichert werden die Daten in den Tabellen „Source“ und „View“.

Tabelle 1.9: Die Tabelle „Level“

Feld	Typ	Beschreibung
Key_Level	int(10)	Primärschlüssel
Level_Name	text	Name des Levels

In der Tabelle „Level“ werden die Level der Aufgaben gespeichert. Die Level sind eine einfache Einstufung in Schwierigkeitsgrade.

Tabelle 1.10: Die Tabelle „Link“

Feld	Typ	Beschreibung
Key_LinkID	int(10)	Primärschlüssel
Key_Link	int(10)	Ursprungsaufgabe
Key_Next_Link	int(10)	Folgeaufgabe zu der Ursprungsaufgabe

In der Tabelle „Link“ werden die Aufgaben untereinander verknüpft. Damit kann man einer Aufgabe eine Folgeaufgabe zuweisen.

Tabelle 1.11: Die Tabelle „ListExercises“

Feld	Typ	Beschreibung
Key_ListExercise	int(10)	Primärschlüssel
Key_Listname	int(10)	Die Liste, zu der die Aufgabe gehört
Key_Exercise	int(10)	Verweis auf die Aufgabe

In der Tabelle „ListExercises“ werden Aufgaben Aufgabenlisten zugewiesen.

Tabelle 1.12: Die Tabelle „Listname“

Feld	Typ	Beschreibung
Key_ListName	int(10)	Primärschlüssel
ListName_Name	int(10)	Name der Aufgabenliste

In der Tabelle „Listname“ werden die Namen und die IDs von Aufgabenlisten gespeichert.

Tabelle 1.13: Die Tabelle „Metadata“

Feld	Typ	Beschreibung
Teacher	text	Name des Aufgabenstellers
Subject	text	Fachgebiet
Actual_List	int(10)	Aktuelle Aufgabenliste oder 0 für alle Aufgaben

In der Tabelle „Metadata“ können verschiedene Daten gespeichert werden, mindestens Teacher (Dozent) und Subject (Fach).

Tabelle 1.14: Die Tabelle „MultipleChoice“

Feld	Typ	Beschreibung
Key_MultipleChoice	int(10)	Primärschlüssel
Key_Exercise	int(10)	Exercise-Verknüpfung
Key_Source	int(10)	Source-Verknüpfung
Key_View	int(10)	View-Verknüpfung
MultipleChoice_Is_Correct	tinyint(1)	Boolean Wert, ob die Lösung richtig ist

In der Tabelle „MultipleChoice“ werden die Multiple Choice-Lösungen zu den Aufgaben referenziert. Gespeichert werden die Daten in den Tabellen „Source“ und „View“.

Tabelle 1.15: Die Tabelle „Solution“

Feld	Typ	Beschreibung
Key_Solution	int(10)	Primärschlüssel
Key_Exercise	text	Exercise-Verknüpfung
Key_Source	int(10)	Source-Verknüpfung
Key_View	int(10)	View-Verknüpfung
Solution_Step	tinyint(3)	Nummer des Lösungsschrittes

In der Tabelle „Solution“ werden die Lösungsschritte zu den Aufgaben referenziert. Gespeichert werden die Daten in den Tabellen „Source“ und „View“.

Tabelle 1.16: Die Tabelle „Source“

Feld	Typ	Beschreibung
Key_Source	int(10)	Primärschlüssel
Source_Data	mediumblob	BLOB-Daten
Source_Name	text	Name der Quelldatei
Source_Type	text	Datentyp der Quelldatei

In der Tabelle „Source“ werden die OLE-Daten, also die Word-Quellobjekte der Aufgaben, Lösungen und Lösungsschritte gespeichert. Der Dateiname und der Datentyp werden zum einfacheren Herunterladen der Quelldaten auch abgespeichert.

Eine weitere Tabelle mit MIME Types oder einer anderen Form der Datentyp-Kennzeichnung ist hier wegen dem notwendigen Pflegeaufwand nicht realisiert.

Tabelle 1.17: Die Tabelle „Topics“

Feld	Typ	Beschreibung
Key_Topic	int(10)	Primärschlüssel
Topic_Name	text	Name des Fachgebietes

In der Tabelle „Topics“ werden die Fachgebiete der Aufgaben gespeichert. Die Fachgebiete dienen der Unterteilung des Fachbereiches.

Tabelle 1.18: Die Tabelle „View“

Feld	Typ	Beschreibung
Key_View	int(10)	Primärschlüssel
View_Data	text	BLOB Daten

In der Tabelle „View“ werden die aus den OLE-Daten konvertierten Bilddaten zu den Aufgaben, Lösungen und Lösungsschritten gespeichert.

1.6.9 Tabellenbeziehungen

Die folgende Übersicht ermöglicht eine bessere Verständlichkeit hinsichtlich der Beziehungen der Tabellen untereinander.

Index	Links
Key_Index	Key_Link
DB_activated	Link_Name
DB_Info	Link_Shortinfo
DB_User	Link_URL
DB_Password	Link_Picture_URL
DB_Location	

Abbildung 1.2: Tabellenbeziehungen der Datenbank LernAss

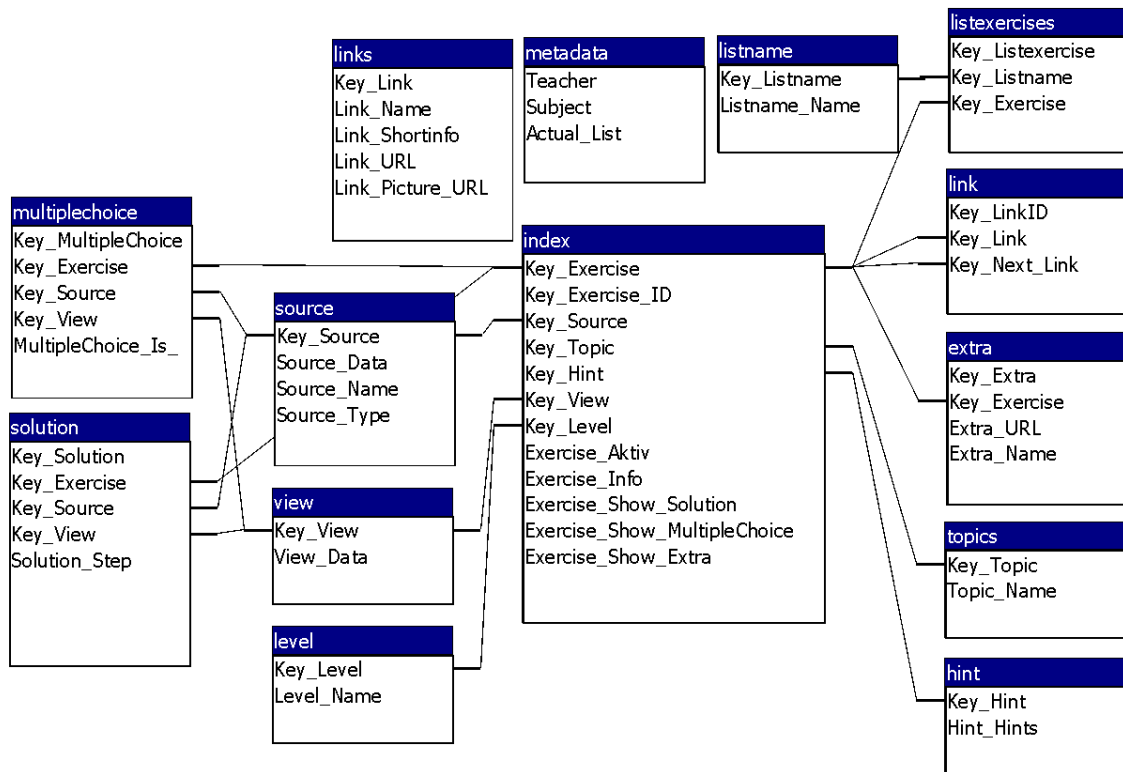


Abbildung 1.3: Tabellenbeziehungen der Aufgabendatenbanken

1.7 Das Betriebssystem für den Server



Die Wahl des Betriebssystems ist sehr schwierig. Voraussetzung ist mindestens NT4 mit SP6a – völlig veraltet und nicht in Betracht zu ziehen. Windows 2000 oder Windows 2000 Server sind die erste Wahl, WindowsXP Professional wäre eine Überlegung wert, ist jedoch noch sehr neu und als Server erst seit Mitte 2003 erhältlich. Der WindowsXP-Server nennt sich .NET Server und ist wahrscheinlich in Zukunft eine mögliche Basis für LernAss.

1.8 Die Programmiersprache

Microsoft hat über mehrere Jahre hinweg die Entwicklung von VisualStudio vorangetrieben. In der heutigen Zeit werden Programme objektorientiert und multi-threaded geschrieben, was die alten Programmierumgebungen von Microsoft noch nicht so gut beherrschten. Die komplette Architektur von .NET (siehe Kapitel 2) ist ein Ansatz der Java ähnlich sieht – man braucht einen RuntimeCompiler (siehe Kapitel 2.4.2 auf Seite 34), der den vorkompilierten Code ausführen kann.

Wichtig ist, dass die Klassen sehr stark überarbeitet wurden. Man hat viele Dinge neu angeordnet und vor allem auch wirklich geordnet und verbessert. Dadurch sind alte Codes nicht mehr verwendbar.

Leider haben die Programmierer durch diese große Herausforderung auch den einen oder anderen Fehler gemacht. So ist z.B. die Clipboardkomponente komplett neu geschrieben worden, wobei man aber die Komponenten für WMF und EMF vergessen oder nicht korrekt geschrieben hat. Das verschärfte die Problematik der Konvertierung von Word in ein Grafikformat noch weiter (siehe Kapitel 5.8 auf Seite 59).

Eine weitergehende Einführung zu .NET ist in Kapitel 2 ab Seite 28 zu finden.

1.9 Die Datenbank MySQL



1.9.1 Das Datenbanksystem

MySQL ist ein relationales Datenbanksystem, das für den privaten und nicht Geld bringenden Gebrauch offiziell ohne eine Lizenzierung verwendet werden darf. Es sind leichte Schwächen (z.B. Fehlen der referenziellen Integrität – bis Version 4 nicht vorhanden) gegenüber kommerziellen Produkten vorhanden, die aber durch Programmierung umgehbar sind.

So ist zum Beispiel in Versionen <4 noch keine referenzielle Integrität gewährleistet. Dies ist ein Manko, was zwingend durch eine programmierte Abfangroutine behoben werden muss.

In punkto Schnelligkeit und Plattformunabhängigkeit ist MySQL die erste Wahl, zumal es auch relativ geringe Anforderungen an die Hardware stellt.

1.9.2 MySQL Provider für .NET

Microsoft hat in der Entwicklung von .NET nur die für das Unternehmen gängigsten Datenbanken in die Bibliotheken aufgenommen. Dadurch existieren so genannte Dataprovider nur für den Microsoft SQL-Server, ADO, ODBC und natürlich auch Oracle. Für andere Datenbanksysteme muss man auf kostspielige oder auch auf unter OpenSource entwickelte Provider zurückgreifen.

Während meiner Diplomarbeit habe ich mehrmals den Dataprovider gewechselt. Die besten Resultate erzielte ich mit dem MySQLdotnet Provider⁷, bei dessen Entwicklung ich dann auch durch Debugging mitgeholfen habe.

Später ist der ByteFX⁸ Dataprovider in Betracht zu ziehen, da dieser plattformunabhängig ist.

⁷<http://www.sourceforge.net/projects/mysqldotnet/> Abrufdatum 04.08.2003

⁸<http://sourceforge.net/projects/mysqlnet/> Abrufdatum 04.08.2003

2 .NET

Inhaltsangabe

2.1	Grundlegendes	29
2.2	Gründe für .NET	29
2.2.1	Vorteile von .NET	30
2.2.2	Nachteile von .NET	31
2.3	Das .NET-Framework	32
2.4	Architektur	33
2.4.1	Microsoft Intermediate Language	33
2.4.2	Just in Time Compiler	34
2.4.3	Verwalteter Code gegen nicht verwalteten Code	34
2.4.4	Sicherer gegen unsicherer Code	35
2.4.5	Assemblies und der Global Assembly Cache	35
2.4.6	Common Language Runtime	36
2.4.7	Common Type System	36
2.4.8	Common Language Specification	36
2.5	Sicherheitsmechanismen	37
2.5.1	Unterteilung der Sicherheitsmechanismen	37
2.5.2	Konfiguration des Sicherheitssystems	37
2.6	Die Sprachen	38
2.6.1	VB.NET	38
2.6.2	ASP.NET	39
2.6.3	C#	39
2.7	.NET Services	40
2.7.1	Web-Services	40
2.7.2	.NET MyServices	40

2.7.3 .NET Passport	40
-------------------------------	----

Tabellen

2.1 Die wichtigsten Klassenbibliotheken des .NET-Frameworks	33
2.2 .NET-Sicherheitssystem – Konfigurationsdateien	38

2.1 Grundlegendes

Angaben aus dieser und folgender Sektionen zu .NET sind angelehnt an das Buch von Michael Kofler [3].



2.2 Gründe für .NET

Bevor Microsoft .NET (auch dotnet genannt) herausbrachte, war die Programmentwicklung unter Windows primär auf C und C++ ausgerichtet (ist sie auch noch, wenn es um Betriebssysteme und wichtige Komponenten geht). Hierfür wurden damals die MFC (Microsoft Foundation Class Library) sowie die für C++ und VB6 gedachte COM (Component Object Model) verwendet. Die Konzepte waren jeweils ein Fortschritt zu den bestehenden Technologien, jedoch entstanden mit der Zeit immer mehr Probleme.

Die größten Probleme der COM(+) und MFC:

- **DLL-Hell:** Die alten COM Bibliotheken werden in das Systemverzeichnis von Windows (%SYSTEMDIR%) kopiert. Es existierte kein Versionsschutz, wodurch bei der Installation neuer Software mit neuen (unter Umständen inkompatiblen DLLs) Probleme auftraten, die bis zur Unbrauchbarkeit von Windows führen konnten.
- **Speicherverwaltung:** Bei COM ist der Programmierer für die Freigabe nicht mehr verwendeter Objekte verantwortlich. Bei einzelnen Objekten war das kein Problem, jedoch bei aufeinander verweisenden Objekten entstand schnell ein Speicherverbrauch, der nicht rückgängig zu machen war.

- Sicherheit: Die Programme liefen meistens mit den Rechten des Benutzers, egal, von wo aus sie gestartet wurden. Dies ist einer der Gründe für rasche Virenverbreitung unter Windows. Ausnahmen waren in der Regel COM+-Komponenten, die mit der Hilfe des Microsoft Transaction Server liefen.
- Konsistenz, Objektorientierung: COM ist mittlerweile zehn Jahre alt. Die meisten Bibliotheken sind über diese zehn Jahre immer wieder erweitert worden, ohne deren Konsistenz zu wahren. Vergleichbare Operationen sind in verschiedenen Objekten auf unterschiedlichste Weisen implementiert. Die Anforderungen, die heute an objektorientierte Programmiersprachen gestellt werden, sind nicht mehr mit den alten Techniken vereinbar.

Die Probleme der alten Technik waren nicht einfach durch Updates zu lösen, da der Aufbau von Anfang an nicht den heutigen Bedürfnissen entspricht. Wartung des COM-Codes wird immer schwieriger.

Aus den genannten und aus weiteren Gründen (wie z.B. der Implementierung von Webfunktionen, Streaming/SOAP und XML) hat sich Microsoft zur Neuentwicklung der gesamten Infrastruktur entschlossen. Die Entwicklung dauerte mehrere Jahre und ist erstaunlich konsistent für ein Microsoft-Produkt.

2.2.1 Vorteile von .NET

.NET ist ein Ansatz zur Lösung der bekannten Probleme. Eine der wichtigsten Neuerungen ist die Strategie zur Vermeidung der DLL-Hell. Die Parallelinstallation mehrerer verschiedener DLLs ist möglich.

Die Weitergabe von VisualStudio .NET-Programmen ist wesentlich einfacher als bei den VisualStudio Vorgängern, da man die Komponenten nicht mehr in der Windows-Registrierdatenbank eintragen muss. Einzige Voraussetzung ist ein installiertes .NET-Framework.

Die Speicherfreigabe wird wie bei JAVA durch einen im Hintergrund laufenden „Garbage Collector“ (Müllsammler) erledigt, der nicht mehr verwendete Objekte erkennt und entfernt. Microsoft hat es wieder einmal verstanden, gute Ideen zu kopieren und zu adaptieren.

In der Neukonzeption von .NET sind einige Sicherheitsmechanismen eingebaut worden, die Sicherheitsprobleme zumindest verringern. Vom Prinzip her ist der Sicherheitsmechanismus mit dem des Internet Explorers zu vergleichen. Die Regeln basie-

ren auf der Herkunft der Software (Internet, lokale Anwendung, Entwicklungsumgebung).

Die neuen Klassenbibliotheken bieten neue Möglichkeiten, bessere Konsistenz und ein besseres Konzept. Zur Lösung von alltäglichen Problemen sind viele Funktionen vereinfacht worden. Die neuen Klassen muss man sich aber, wie früher auch, erst einmal aneignen.

Einer der größten Vorteile besteht in der Sprachunabhängigkeit. Die zur Auswahl stehenden Sprachen umfassen standardmäßig C++, VB.NET und J#. Es existieren aber auch Lösungen von Fremdanbietern, die zum Beispiel Fortran und weitere, auch sehr ausgefallene Sprachen in die .NET Umgebung integrieren.

.NET ist so strukturiert worden, dass jede Sprache dieselben Datentypen und die selben Bibliotheken verwendet. Damit entsteht kein weiterer Aufwand bei der Installation und/oder der Erstellung der Software.

In Medien wurde .NET oft als neue Internetplattform dargestellt. Richtig ist, dass .NET die Entwicklung von Netzwerkanwendungen vereinfacht, aber es ist eine vollwertige Programmierumgebung, die auch Windowsapplikationen erstellen kann.



Die Entwicklungsumgebung VisualStudio .NET ist eine Vereinfachung der Code-Entwicklung unter Microsoft-Betriebssystemen der neueren Generation (ab Windows 2000).

2.2.2 Nachteile von .NET

Microsoft versprach seit je her bei jeder neuen Technologie das Blaue vom Himmel. Bei .NET ist das durchaus genauso.

Ein kritischer Punkt ist, dass .NET noch sehr neu ist. Mehrere Millionen Zeilen Code sind geschrieben worden und die Wahrscheinlichkeit, dass dort noch etliche Fehler vorhanden sind, ist sehr groß. Die Dokumentation ist an vielen tiefgreifenden Stellen noch lückenhaft.

Die enge Bindung von .NET an Windows ist ein Nachteil – versprochen ist .NET für NetBSD und durch Opensource-Entwickler auch für Linux. Inwieweit die Projekte

allerdings wirklich zum Ziel kommen, bleibt abzuwarten. Als Gegenbeispiel sei nur der Internet Explorer für Linux genannt, der nicht fertig entwickelt wurde. Leider erschweren zur Zeit auch Softwarepatente diese Entwicklungen, und Microsoft will Patente für .NET.

Ein großes Problem stellt die Sourcecode-Inkompatibilität dar. VB6-Projekte können mit VB.NET nicht gewartet werden. Zur Kompatibilität von .NET und VB6 wurden dennoch etliche Teile in .NET übernommen (Namespace Microsoft.VisualBasic.*), die aber keine 100%ige Funktionsgarantie darstellen. Durch die Rückwärtskompatibilität werden auch viele Vorteile von .NET zunichte gemacht.

Problematisch ist die Kompatibilität des .NET-Frameworks mit den Windowsbetriebssystemen vor Windows 2000. .NET funktioniert nur auf den „besseren“ Windowsversionen Windows 2000 (Server) und Windows XP zu 100 %. Im .NET-Server ist .NET vorinstalliert. Das Framework ist 20MB groß und muss installiert sein um .NET-Programme auszuführen.

Microsoft hat bereits auf einer Entwicklerkonferenz (2002) bekundet, das C++ wieder stärker in Betracht gezogen wird, aber koexistent zu .NET. In Visual Studio 2002 und 2003 ist C++ integriert.

2.3 Das .NET-Framework

.NET-Framework ist die Bezeichnung für die Bibliotheken (siehe Tabelle 2.3 auf Seite 33) und Komponenten, die zur Ausführung von MSIL-Code notwendig sind. Die zentrale Komponente ist der JIT, der den MSIL-Code in die Maschinensprache übersetzt (wie bei JAVA). Dazu gehört auch ein Satz von Administrationstools, der unter anderem die Codesicherheit gewährleisten soll.

Das .NET-Framework SDK enthält zusätzlich zum .NET-Framework auch alle Entwicklungswerkzeuge für C# und VB.NET, viele Beispiele und eine sehr umfangreiche Dokumentation zum .NET-Framework. Die Tools sind im Gegensatz zu VisualStudio .NET frei erhältlich, aber eben ohne die VisualStudio .NET-Entwicklungsumgebung.

Tabelle 2.1: Die wichtigsten Klassenbibliotheken des .NET-Frameworks

Bibliothek	Funktion
mscorlib.dll	Basisdatentypen, grundlegende Operationen
System.dll	Basisklassen zur Nutzung verschiedener Netzwerkprotokolle
System.Data.dll	ADO.NET – Klassen zur Datenbankprogrammierung
System.Drawing.dll	GDI+ – Klassen zur Grafikprogrammierung
System.Management.dll	Zugriff auf die WMI*
System.Web.dll	ADO.NET – Klassen zur Programmierung von Internetanwendungen
System.Windows.Forms	Klassen zur Windows-Programmierung
System.Xml.dll	Klassen zur XML-Bearbeitung
Microsoft.VisualBasic	Grundkompatibilität zu VB6

* Windows Management Instrumentation

2.4 Architektur

2.4.1 Microsoft Intermediate Language

Microsoft Intermediate Language (MSIL) ist ein Zwischencode, der vom .NET-Compiler erzeugt wird. Der Code enthält auch Metadaten, die die im Programm oder der Bibliothek enthaltenen Klassen und Methoden beschreiben. Dies ist ein großer Vorteil bei Bibliotheken, da diese sich damit selbst beschreiben und der Compiler anhand der Metadaten auch die Aufrufparameter überprüfen kann.

Bei der Ausführung des MSIL-Codes wird dieser mittels des JIT in Maschinencode übersetzt. Man kann den MSIL-Code mit dem Java-Bytecode und dem P-Code von früheren VB-Versionen vergleichen.

Der Vorteil von MSIL ist die mögliche Plattformunabhängigkeit, die aber zum Zeitpunkt der Diplomarbeit noch nicht unter Beweis gestellt wurde. Bisher existiert nur das .NET-Framework für Windows. Microsoft hat anscheinend Corel mit einem Linux Port für das .NET-Framework beauftragt. Das soll aber noch bis zu drei Jahre dauern. Im Moment gibt es bereits ein freies Projekt, das bestrebt ist, das .NET-Framework nachzuprogrammieren.¹

¹Mono Projekt <http://www.go-mono.com> Abrufdatum 05.08.2003

Der MSIL-Code ist nicht besonders sicher – man kann den MSIL-Code mit dem Programm „ildasm.exe“ lesen und es gibt sogar Programme, die den MSIL-Code in fast perfekten C#-Code übersetzen.² Dadurch ist kein Schutz des Sourcecodes gegeben (eigentlich ist der nie sicher, aber C-Code ist schwieriger wiederherzustellen, da man disassemblieren muss (Assembler ist nicht mehr so gängig). In VisualStudio .NET 2003 ist ein „obfuscator“ installiert, der das Disassemblieren erschweren soll. Für die Vorgängerversion gibt es den [dotfuscator](#) – dieser ist in einer Light-Version in VisualStudio .NET 2003 integriert.

2.4.2 Just in Time Compiler

.NET Programme werden beim Start durch den JIT übersetzt, der aus dem MSIL-Code einen auf die Hardware des vorliegenden Rechners optimierten Maschinencode erstellt. Allerdings werden immer nur die Teile übersetzt, die auch gerade benötigt werden. Eine Optimierung auf den Prozessor ist hier möglich, ob, und wie weit Microsoft diese Möglichkeiten nutzt, ist jedoch fraglich.

Ein Vorteil des JIT ist, dass bei der Ausführung des Codes Sicherheitsprüfungen erfolgen können, zum Beispiel, ob der MSIL-Code korrekt ist, ob unerlaubte Speicherzugriffe erfolgen oder ob die Aufrufparameter für Bibliotheken stimmen. Hier greifen auch die .NET-Sicherheitseinstellungen. Der JIT-Compiler ist somit ein sehr wichtiger Teil des .NET-Konzeptes.

Die Bibliotheken des .NET-Frameworks werden bei der Installation schon in den Maschinencode übersetzt, damit die Leistung des Systems gesteigert wird.

2.4.3 Verwalteter Code gegen nicht verwalteten Code

Der Unterschied zwischen „managed code“ (verwalteter Code) und „unmanaged code“ (nicht verwalteter Code) besteht darin, dass der „unmanaged code“ sich nicht um die Ausführung von Code und die Verwaltung dessen kümmert.

Beim „managed code“ kümmert sich .NET um die Speicherverwaltung der Objekte. Nicht mehr benötigter Speicher wird automatisch freigegeben (garbage collection = Müllsammlung). In Bezug auf Debugging und Sicherheitsaspekte ist der „managed code“ dem „unmanaged code“ stark überlegen. Allerdings werden durch API-Calls und COM-Komponenten belegter Speicher nicht automatisch freigegeben – hier ist der code „unmanaged“.

²Anakrino <http://www.saurik.com/net/exemplar/> Abrufdatum 05.08.2003

2.4.4 Sicherer gegen unsicherer Code

Beim „safe code“ kann ein Programm nicht direkt auf den Speicher zugreifen (z.B. VB.NET). Dazu benötigt man Zeiger, die nur C# bietet. Um einen „unsafe code“ zu schreiben, muss eine Funktion in C# als „unsafe“ gekennzeichnet werden. Das ist bei bestimmten Funktionen eine Geschwindigkeitssteigerung, wird aber wegen dem abgeschalteten Speicherschutz nur in der höchsten Sicherheitsstufe von .NET ausgeführt. Ein falscher Zeiger kann schnell zu Datenverlust oder Absturz des Programms führen. Durch die Sicherheitsfunktionen von .NET reduziert sich die Anwendbarkeit von „unsafe code“.

2.4.5 Assemblies und der Global Assembly Cache

Eine Assembly ist ein Programm (*.exe) oder eine Bibliothek (*.dll), wenn man es einfach betrachtet; jedoch sieht die Realität ein wenig anders aus. Eine Assembly ist eine .NET-Ausführungseinheit. Sie enthält ausführbaren Code (MSIL), der unter Umständen auch über mehrere Dateien verteilt sein kann. Außerdem sind Beschreibungen des Inhalts in Form von Metadaten enthalten, die verfügbare Klassen, Methoden, Typen usw. enthalten. Es können aber auch Ressourcen wie z.B. Text, Bilder, Musikstücke, Videos usw. enthalten sein.

Eine Assembly bildet eine logische Einheit, die nur als Ganzes weitergegeben werden kann und die nur einen logischen Einsprungpunkt hat. Viele der Sicherheitsmechanismen gelten auf dieser Ebene. Der Datenaustausch in einer Assembly ist effizienter als über sie hinaus.

Die Assemblies sind ein Teil der .NET-Lösung gegen die DDL-Hell (siehe Kapitel [2.2](#) Seite 29). Jedes Assembly hat eine eindeutige Versionsnummer. Eine parallele Installation mehrerer verschiedener Assembly-Versionen ist möglich, da jedes .NET-Programm die Versionsnummer der verwendeten Assemblies beinhaltet.

Die Assemblies werden im GAC (Global Assembly Cache) abgelegt. Dieser befindet sich in „%WINDIR%\assembly“. Im GAC sind alle .NET-Basisbibliotheken enthalten. Beim Einfügen von Assemblies werden Sicherheitsüberprüfungen durchgeführt, die eine hohe Sicherheit gewährleisten sollen.

Microsoft empfiehlt, selbst entwickelte Bibliotheken nicht im GAC abzulegen, da der GAC nicht überfüllt werden sollte. Eine Art Müllsammelner gibt es hier nicht, Applikationen sind selbst für die Bereinigung oder Entfernung eigener nicht mehr benötigter Komponenten verantwortlich.

2.4.6 Common Language Runtime

Die CLR (Common Language Runtime) fasst die wesentlichen Teile zusammen, die benötigt werden, um .NET-Programme auszuführen. Dies sind hauptsächlich:

- der JIT-Compiler und die Tools zum Starten und Umwandeln von MSIL Code
- die .NET-Bibliotheken.

Durch die Installation des .NET-Frameworks steht die CLR auf dem Rechner zur Verfügung. Die CLR ist sprachunabhängig (common = gemeinsam) – die VB6- und VC-Runtime waren schlicht verschieden und jeweils für deren Sprachen.

2.4.7 Common Type System

Das CTS (Common Type System) ist ein allgemeines Typensystem, das eine formale Beschreibung der Datentypen darstellt, die von .NET unterstützt werden. Dazu zählen Wert- und Verweistypen, unterschiedliche Klassentypen (vererbbar, nicht vererbbar usw.), Strukturen, Felder, Aufzählungen usw. Die .NET-Sprachen müssen nicht alle Elemente des CTS beherrschen.

2.4.8 Common Language Specification

Die CLS (Common Language Specification) beschreibt die Teilmenge des CTS, die eine Programmiersprache beherrschen muss. Bibliotheken, die der CLS entsprechen, können in jeder Sprache ohne Einschränkungen verwendet werden.

Weiterhin beschreibt die CLS-Regeln, die eine Bibliothek nach der CLS-Spezifikation zertifizieren können. Das betrifft aber nur die Schnittstellen nach außen und nicht die Interna.

VB.NET verwendet zwar als Standardvariablentypen nur die CLS kompatiblen Typen, aber man kann durchaus nicht CLS konforme Bibliotheken erstellen, indem man z.B. UInt16 oder UInt32 verwendet. Schlimm daran ist nur, dass nicht jeder Compiler diesbezüglich eine Warnung oder eine Information ausgibt. Der C#-Compiler erstellt diese Warnungen bereits, und bei VB.NET soll das noch nachgerüstet werden.

2.5 Sicherheitsmechanismen

Das .NET-Framework und die Bibliotheken implementieren ein komplexes Sicherheitskonzept. Vor jeder Befehlsausführung wird geprüft, ob die benötigten Rechte des Benutzers gegeben sind. Wenn ein Benutzer nicht die benötigten Rechte hat, tritt ein Fehler auf.

2.5.1 Unterteilung der Sicherheitsmechanismen

Die Rechte sind unterteilt in:

- **Gewöhnliche Benutzerrechte:** Hier greifen die Rechte von Windows NT, 2000 und XP, die durch Gruppen- und Benutzerrichtlinien erstellt werden. Dieser Mechanismus existiert unter Multi-Usersystemen von Microsoft schon immer.
- **Herkunft des Programms:** .NET unterscheidet zwischen lokaler Festplatte (am sichersten), aus dem LAN (Intranet), aus dem Internet (am unsichersten) und explizit als vertrauenswürdig bezeichneten Orten (ähnlich dem Konzept des Internet Explorers).
- **.NET Sicherheitskonfiguration:** Das .NET-Sicherheitssystem kann auf drei Ebenen konfiguriert werden: Organisation, Rechner oder Benutzer.

Auf einer oberen Ebene gemachte Beschränkungen können nur noch erweitert, nicht aber gelockert werden. In der Standardeinstellung werden die Zugriffsrechte durch die Rechnerebene bestimmt und können auch nur auf dieser erweitert werden. Dadurch kann sich ein Benutzer nicht mehr Rechte geben als die, die der Administrator auf der Rechnerebene vorsieht.

Die Rechte werden mit dem .NET-Konfigurationsprogramm eingestellt. Es befindet sich in der „Systemsteuerung\Verwaltung\Microsoft .NET Framework Konfiguration“.

2.5.2 Konfiguration des Sicherheitssystems

Das Sicherheitssystem beruht auf drei XML-Dateien, die die einzelnen Sicherheitsebenen steuern.

Tabelle 2.2: .NET-Sicherheitssystem – Konfigurationsdateien

Unternehmen	Organisation, Enterprise – Jeder darf alles
%WINDIR%\Microsoft.NET\Framework\v1.0.3705\config\enterprisesec.config	
Rechner	Computer - Die Standardeinstellung ist zonenabhängig: Lokaler Rechner (Arbeitsplatz): jeder darf alles (Full Trust) Intranet: eingeschränkte Rechte (Local Intranet) Vertrauenswürdige Plätze: stärkere Einschränkungen (Internet) Internet: alles ist verboten (Nothing)
%WINDIR%\Microsoft.NET\Framework\v1.0.3705\config\security.config	
Benutzer	Standardeinstellung: Jeder darf alles
\$USERDIR\$\BENUTZER\Anwendungsdaten\Microsoft\CLR Security Config\v1.0.3705\security.config	

Ich möchte an dieser Stelle nicht tiefer in das Sicherheitskonzept einsteigen – es geht in dieser Diplomarbeit nicht um .NET.

2.6 Die Sprachen

2.6.1 VB.NET

VB.NET ist die konsequente Weiterentwicklung von VB6. Es besteht allerdings keine Sourcecodekompatibilität, was die Migration sehr erschwert. Dieser Schritt von Microsoft war angesichts der unüberschaubaren Klassenbibliotheken und jahrelanger Erweiterungen und Fehlerbehebungen (es gab viele Patches, aber keine wirkliche Behebung der grundlegenden Probleme) notwendig geworden.

VB.NET ist die am einfachsten zu erlernende Sprache von VisualStudio .NET. Durch das Konzept von .NET ist kein Geschwindigkeitsunterschied mehr zwischen VB, C++ oder anderen für .NET verfügbaren Sprachen auszumachen. Auch im Vergleich zu „älteren“ Programmiersprachen ist VB.NET recht schnell, obwohl man nur precompiled Code erhält.

2.6.2 ASP.NET

ASP.NET ist eine konsequente Weiterentwicklung von ASP. Die Sprache, in der die ASP.NET-Seiten verfasst werden, ist frei wählbar. Das LernAss-Webfrontend ist in VB.NET geschrieben.



In VisualStudio .NET ist die ASP.NET-Entwicklung durch IntelliSense[®] stark vereinfacht worden. Man kann eine Webseite aufbauen wie ein Windowsform, was allerdings nur mit dem InternetExplorer >5 als Browser wirklich richtig funktioniert. ASP konnte man früher nur in Editoren, nicht aber in einer Entwicklungsumgebung mit Debugger erstellen.

ASP.NET ist nicht interpretierter (wie ASP), sondern kompilierter Code, was die Geschwindigkeit stark erhöht. WebClasses aus VB6 gibt es in VB.NET nicht mehr, fertige ASP.NET-Lösungen können einfach in neue Verzeichnisse kopiert werden. In der Verwaltung der „Internet-Informationdienste“ muss man nur das Verzeichnis der Webapplikation in den Eigenschaften als Applikation konfigurieren.

ASP.NET ist in Anbetracht des Entwicklungskomforts ein sehr großer Fortschritt – auch im Vergleich zu dem sehr beliebten PHP.³ Der Nachteil ist der Preis – man braucht einen Windowsrechner mit IIS.

Eine Minderung der Kosten bei ASP.NET auf der Entwicklerseite stellt das freie Tool „ASP.NET Web Matrix“⁴ dar. Außerdem bleibt abzuwarten, was sich aus den freien Implementationen des .NET-Frameworks entwickelt.

2.6.3 C#

Auf C# soll hier nicht weiter eingegangen werden, es sollte jedoch genannt werden, da es die stärkste Sprache des .NET-Frameworks ist. Für diese Diplomarbeit wurde ein Dataprovider benötigt, der in dieser Sprache geschrieben sein muss. Im Prinzip ist es eine Weiterentwicklung von C und C++.

³<http://www.php.net> Abrufdatum 05.08.2003

⁴<http://www.asp.net/webmatrix/> Abrufdatum 05.08.2003

2.7 .NET Services

2.7.1 Web-Services

Web-Services sollen die Kommunikation von Internet-Servern über das HTTP-Protokoll standardisieren. Das verwendete Datenformat ist XML, im Gegensatz zu ASP, wo HTML als Datenformat benutzt wird. Sinnvolle Verwendung finden Web-Services in Webanwendungen, die nur Daten liefern, wie zum Beispiel Börsenkurse oder Wetterdaten.

Web-Services sind nichts wirklich Neues, [IBM](#) und [Sun](#) haben ähnliche Protokolle erarbeitet. Die Basis bilden XML und SOAP, wodurch die Web-Services zueinander kompatibel sind. Microsoft versucht hier eine Standardisierung durchzudrücken.

2.7.2 .NET MyServices

.NET MyServices⁵ ist eine der ersten Anwendungen für die Web-Services. Ziel ist (und war – früher nannte sich das Projekt Hailstorm) die zentrale Speicherung von Kundendaten, wie Name, Vorname, Adresse, E-Mail und Kontodaten. Das System soll Kooperationspartnern nur benötigte Informationen über den Kunden zukommen lassen und außerdem soll es eine sichere Transaktion gewährleisten.

Von Vorteil wäre die dadurch erleichterte Anmeldung bei den Webseiten, da man sich nur authentifizieren, nicht aber wieder neu anmelden muss. Das Konzept des Datenaustausches ist standardisiert und einfach, jedoch bestehen Zweifel bezüglich der Datensicherheit und des Datenschutzes. Durch die Bedenken haben sich für diese Technik nur wenige Firmen entschieden.⁶

2.7.3 .NET Passport

.NET Passport ist ein Dienst, der ein zentral verwaltetes Login erlaubt. Durch diesen Dienst wird .NET MyServices (siehe oben) erst ermöglicht. Voraussetzung für die Webseiten, die den Passport nutzen möchten, ist die Kooperation mit Microsoft. Dadurch gibt es auch kaum Microsoft unabhängige Webseiten, die diesen Dienst nutzen. Ein bekannter Vertreter für die .NET Passport-Anmeldung ist Ebay.⁷

⁵<http://www.microsoft.com/myservices/> Abrufdatum 05.08.2003

⁶Zu Recht – es gab eine [Sicherheitslücke in Passport](#) Abrufdatum 05.08.2003

⁷<http://www.ebay.de> Abrufdatum 05.08.2003

3 Lernsysteme

Inhaltsangabe

3.1	Einleitung	41
3.2	Was sind Lernsysteme?	42
3.3	Der PC als Hilfestellung	42
3.4	Die verschiedenen Lernsysteme	42
3.4.1	Verwendete Begriffe	42
3.4.2	Computer unterstützter Unterricht (CUU)	43
3.4.3	Intelligente tutorielle Systeme (ITS)	44
3.4.4	Der Unterschied von CBT und WBT	44
3.5	Motivation des Benutzers	46

Abbildungen

3.1	Computer unterstützter Unterricht	44
3.2	Aufbau von WBT Systemen	46

3.1 Einleitung

In diesem Kapitel werden Lernsysteme erklärt. Dieses Kapitel ist an die vorhergehende Diplomarbeit über LernAss [1] angelehnt. In dieser Diplomarbeit ist es wieder einmal das Ziel, mit LernAss ein interaktives Lernsystem zu schaffen, jedoch basierend auf der „bewährten“ alten Technik.

Es sollen Unstimmigkeiten und Designschwächen durch die Neuentwicklung ausgemerzt werden. Computer unterstütztes Lernen wird heute von vielen Firmen benötigt, jedoch gibt es noch kaum vernünftige Systeme. Manche Firmen entwickeln zur Zeit auch eigene Systeme, was jedoch nicht immer von Erfolg gekrönt ist.

3.2 Was sind Lernsysteme?

Lernsysteme sind, egal auf welcher Basis sie beruhen, alle Systeme, die einen Lernenden beim Lernen unterstützen. Mögliche Basis für ein Lernsystem sind Bücher, Video/Audio oder eben PC basierte Systeme. Der Vorteil von PC gestütztem Lernen gegenüber den anderen Systemen ist die einfache Durchsuchbarkeit sowie Skalierbarkeit und Verschachtelung.

Man kann vorhandene Informationen komfortabel bearbeiten und erweitern sowie in gewünschter Reihenfolge zusammenstellen. Wichtige Voraussetzung ist die Erkenntnis, dass jeder Mensch anders lernt.

3.3 Der PC als Hilfestellung

Digital gespeichertes Wissen kann über einfache bis hin zu sehr komplexen Suchfunktionen abgerufen werden. Man kann durch Vorauswahlen und Erweiterungen des Lernstoffs die Lerndatenbank skalieren. Durch Verlinkung auf bestimmte für den Lernstoff relevante Themen wird eine Verschachtelung erreicht.

Der PC ist programmierbar und kann folglich durch definierte Regelwerke auch Analysen über das Benutzerverhalten erstellen, über die eventuell weiteres Wissen angeboten werden können. Aber auch die Entscheidung über das Fortsetzen einer Übung, je nach Qualifikation des Probanden, kann über ein Regelwerk definiert werden.

3.4 Die verschiedenen Lernsysteme

3.4.1 Verwendete Begriffe

Für CBTs werden die folgenden Begriffe im üblichen verwendet:

- **Computer Based Training (CBT):** Computer basiertes Lernen
Der Begriff umfasst alles Computer basierte Lernen (Training); im Prinzip alles, was strukturiert zu einer bestimmten Sache Auskunft gibt, so zum Beispiel auch eine Anleitung zu einem Programm in Form von einer „Computerhilfe“.

- **Distance Learning:** Fernunterricht
Als Beispiel: Es werden über Radio und Fernsehen Lektionen ausgestrahlt.
Aber auch Videounterricht gehört dazu.
- **Desktop Training:** Arbeitsplatz-Training
Jede Art von lokal am Desktop ausgeübten Training.
- **Computer assisted Instruction:**
Jede Art des Unterrichts, der durch PCs unterstützt wird.
- **Interaktives Lernen:**
Oberbegriff über CBT und multimediales Training
- **Self paced training:** Selbst geplantes/getimetes Training
Der Proband bestimmt den Zeitpunkt und die Geschwindigkeit des Trainings.
Das ist der „neue“ Überbegriff für CBT, WBT und multimediales Training.

Man unterscheidet folgende Arten von Lernsystemen:

- Computer unterstützter Unterricht (CUU)
 - Tutorielle Systeme
 - Übungssysteme
 - Simulationssysteme
 - Lernumgebungen
- Intelligente tutorielle Systeme (ITS)

3.4.2 Computer unterstützter Unterricht (CUU)

Die heute verwendeten Lernsysteme sind vorwiegend CUU-Systeme, was unter anderem auch an der Komplexität der ITS-Systeme liegt.

Bei dem Computerunterstützten Unterricht wird der Lernstoff in kleine Einheiten zerlegt, die man Instruktions- und Testteile nennt. Der Proband bekommt die Einheiten in einer vorgegebenen Reihenfolge präsentiert. Verlinkungen zu anderen Einheiten dienen der Individualisierung des Lernstoffes.

Generell kann man den Computer unterstützten Unterricht (englisch: „computer based training“ (CBT)) noch in Arbeitsplatzübungen (Desktoptraining) und Netzübungen (Web based training (WBT)) unterteilen. Der Vorteil der WBTs liegt

in der zentralen Verwaltung und Wartung, die auch über das Netzwerk/Internet erfolgen kann, was bei den Desktoptraining-Ausführungen in einem gewaltigen Verwaltungsakt ausarten würde. Eine Änderung im WBT ist sofort und im Normalfall ohne ein Update der Clients verfügbar.

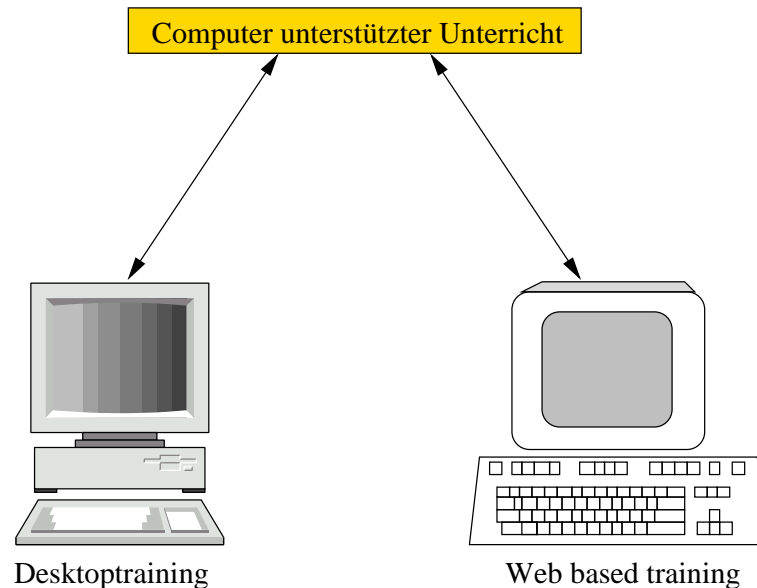


Abbildung 3.1: Computer unterstützter Unterricht

3.4.3 Intelligente tutorielle Systeme (ITS)

Das intelligente tutorielle System ist durch die Programmierung des PC gesteuert. Die Regelwerke erfassen einzeln den Lehrstoff, die Lehrstrategie und den Kenntnisstand des Probanden. Durch eine künstliche Intelligenz (KI – in Form von Regelwerken) werden die einzeln erfassten Daten zusammengeführt und mit Referenzen verglichen. Anhand des Vergleichsergebnisses werden dann Lernstrategien ermittelt.

ITS-Systeme werden vor allem in der Medizin und der Diagnostik sowie in der Forschung eingesetzt.

3.4.4 Der Unterschied von CBT und WBT

Web based training (WBT) ist alles, was als Unterricht über Internet oder Intranet angeboten wird. Die Funktionen des WEB werden im Falle des WBT weitgehend genutzt. Unter anderem dadurch sind WBTs effizienter als CBTs.

Die Vorteile von WBTs sind:

- Zugang ist unabhängig vom Ort – nur eine Verbindung zum Netz muss vorhanden sein
- Multiuserfähig
- Just in Time-Distribution der Lerninhalte – man braucht keinen lokalen Platz
- Zentrale Programmversion – aktueller Stand einfach zu gewährleisten; Updates sind schnell und zentral erledigt
- Feedback ist möglich → tutorieller Effekt
- Anpassbarkeit der Software an eigene Begebenheiten
- Bei Einhaltung der Standards ist eine Plattformunabhängigkeit bei den Clients garantiert. Es wird nur der Webbrowser beim Client benötigt, alles andere stellt der Server.

Die möglichen Nachteile von WBTs sind:

- Die Bandbreite des Servers begrenzt theoretisch die Anzahl der Benutzer
Das Verhalten der Benutzer kann unter Umständen Verzögerungen im Datenfluss verursachen, was zu Ungeduld, Ablenkung oder Missgunst führen kann
- Soziale Vereinsamung – es wird weniger menschlicher Kontakt hergestellt
- WBTs kosten Entwicklungsgeld und Zeit
- Moderiertes Lernen ist mit WBTs nur begrenzt möglich
Emotionale und psychologische Trainings benötigen immer noch menschliche Lehrer.

Generell ist bei CBT und WBT zu sagen, dass Einsparungen durch „lokal“ absolvierbare Trainings möglich sind, so zum Beispiel bei den Reisekosten und der Zeit. Den Menschen als Lehrer sollten WBTs oder CBTs niemals ersetzen, sozial und psychologisch gesehen wäre das nicht gut. WBTs und CBTs sollten „nur“ als Zusatz zu normalen Lernsystemen verwendet werden.

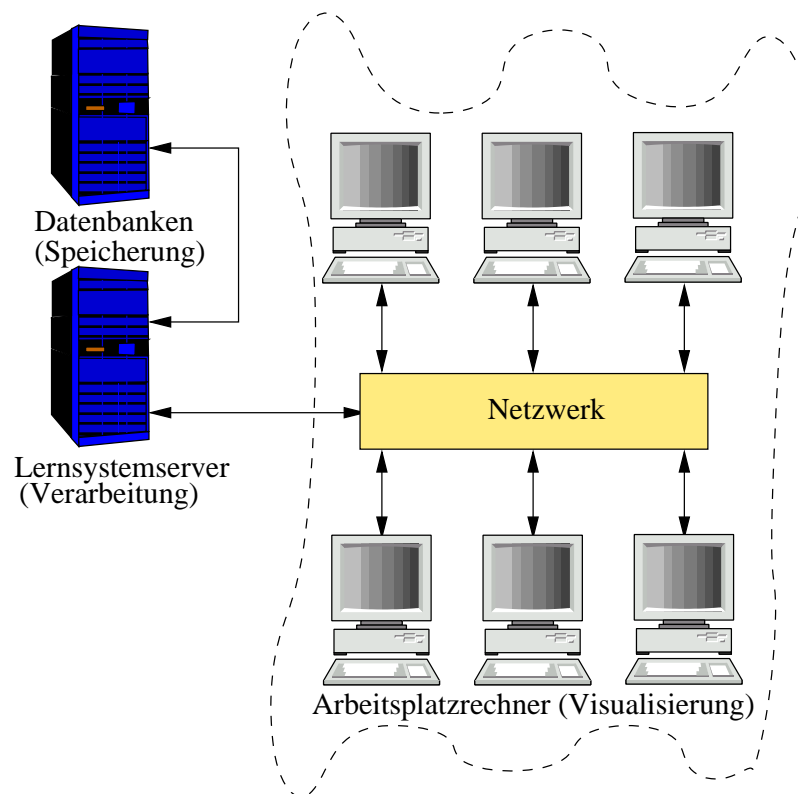


Abbildung 3.2: Aufbau von WBT Systemen

3.5 Motivation des Benutzers

Nach all der Theorie darf nicht vergessen werden, dass die Benutzer Spaß am Lernen haben sollten. Glücklich lernt es sich leichter. Ein bisschen Witz und lockere Bemerkungen sind erheiternd und sicher nicht schädlich. Ein „trockenes“ Lernsystem wird nicht viel fruchten, daher sind multimediale Elemente auch ein wichtiger Bestandteil von WBTs.

Der Lerninhalt darf einen Benutzer nicht erschlagen, er muss folglich übersichtlich und verständlich dargeboten werden. Wichtig sind auch Referenzen, wie die Angabe der Skriptstellen, weiter führende Aufgaben und/oder weiteres Material (Audio/Video/Grafik).

In der heutigen Zeit sind viele nur noch durch bunte Bilder und viele Animationen zum Lernen zu bewegen, aber es muss ein „gesundes Maß“ haben, damit die multimedialen „Nebensächlichkeiten“ nicht zu sehr von der eigentlichen Aufgabe ablenken.

4 Anforderungen an LernAss

Inhaltsangabe

4.1	Einleitung	47
4.2	Modulares Design	48
4.3	Vermittlung des Wissens	48
4.4	Aufgaben der Datenbank	48
4.5	Eingabe der Daten	49
4.6	Abfragen des Wissens	49
4.6.1	Multiple Choice	49
4.6.2	Lösungsschritte	49
4.6.3	Test und Klausurmodus	49
4.7	Arbeitserleichterungen	50

4.1 Einleitung

Dieses Kapitel beschreibt die Anforderungen, die LernAss erfüllen sollte. Allgemein gültige Forderungen sind eine weitgehende Plattformunabhängigkeit, Skalierbarkeit und Zukunftssicherheit.

Die Verfügbarkeit von LernAss sollte in der heutigen Zeit mit der Verwendung des Internets gegeben sein. Was die Offlineversion von LernAss als obsolet erscheinen lässt.

LernAss soll eine Aufgabenverwaltungssoftware sein, und ist folglich nicht nur für das Studium vorgesehen, sondern auch für Weiterbildung in jeder Form geeignet.

Anforderungen der Studierenden und Dozenten wurden nachgefragt. Die Befragung ist auf Seite 7 in Kapitel 1.2 zu finden.

4.2 Modulares Design

Ein modulares Design gewährleistet die Erweiterbarkeit des Systems.

Im Zuge meiner Versuche mit verschiedenen Entwicklungsumgebungen und der Entwicklung der Software habe ich mich für ein modulares Design entschieden (siehe Kapitel 5.6 auf Seite 56).

Der Aufbau von LernAss ist unterteilt in die Module:

- Datenspeicher (Datenbank)
- Datenvisualisierung
- Datenspeicherung und Aufbearbeitung (siehe Kapitel 5.8 auf Seite 59).

4.3 Vermittlung des Wissens

Die Vermittlung des Wissens soll durch das Anbieten von Aufgaben und Tests erfolgen. Damit die Benutzer lernwilliger sind, sollte die Oberfläche ansprechend und klar gegliedert sein. Eine Ausdruckmöglichkeit ist für Papierbegeisterte zu realisieren.

4.4 Aufgaben der Datenbank

Die Hauptaufgabe der Datenbank ist die Speicherung von:

- Aufgaben (in Bild und ursprünglicher Form)
- Lösungen zu den Aufgaben
- Zusatzmaterial (Audio/Video/Grafik/Links)
- Metadaten (Dozent, Fach, ...)
- Auswahllisten
- Stichwörtern

Datenbanken dienen in erster Linie zur Speicherung großer Datenmengen und deren sicherer und schneller Verarbeitung. Die Datenbank kann LernAss zum Beispiel

durch Hash und Volltext-Funktionen zu einer schnellen Suchfunktion verhelfen. Dazu muss für jedes zu durchsuchende Feld ein Hash oder Volltextindex erstellt werden (siehe Ausführungen auf Seite 58 in Kapitel 5.7).

Sehr wichtig ist die Konsistenz der Datenbank. Jede Funktion sollte immer die Konsistenz prüfen, auch wenn die referenzielle Integrität durch die Datenbank geprüft werden kann. Das Design der Tabellen muss gut durchdacht sein (siehe Kapitel 1.6.8 auf Seite 19).

4.5 Eingabe der Daten

Die Eingabe der Daten sollte möglichst einfach und mit Standardtools wie z.B. MS-Office erfolgen. Bevorzugt wird Word, in das man alle möglichen Formate einbinden kann, fast egal, was man verwenden möchte.

Die Verwaltung der Funktionen (Auswahllisten, Zusatzmaterial, Skriptstellen und Querverweise) sollte ergonomisch und einfach sein.

4.6 Abfragen des Wissens

Gelerntes Wissen sollte natürlich auch abgefragt werden. Dazu werden die folgenden Optionen angeboten:

4.6.1 Multiple Choice

Es sollten unbegrenzt Lösungen für einen Multiple Choice-Test angeboten werden können. Generell sollten auch mehrere Lösungen richtig sein können.

4.6.2 Lösungsschritte

Es sollen unbegrenzt viele Lösungsschritte angeboten werden können.

4.6.3 Test und Klausurmodus

Im Klausurmodus werden nur noch Aufgaben oder Lösungen angezeigt, die durch eine Vorauswahlliste erlaubt werden. Diese Funktion ist gegebenenfalls nachzurüsten.

4.7 Arbeitserleichterungen

Generell sollte es möglich sein, eine gerade angezeigte Aufgabe zu drucken. Außerdem sollte es möglich sein, mehrere Aufgaben auf einmal zu drucken. Wichtig ist, dass der Benutzer den Überblick behält. Daher sollte auf Designwechsel und zu viele dynamische Seitenveränderungen verzichtet werden. Besonders die Linkeiste sollte statisch sein, damit der Wiedererkennungseffekt immer gegeben ist.

5 Technisches Redesign

Inhaltsangabe

5.1	Einleitung	52
5.2	Grafikformate	52
5.3	Frontend	54
5.4	Datenbearbeitung	54
5.5	Datenbank und Tabellen	54
5.5.1	Datenbankstruktur	55
5.5.2	Tabellenstruktur von LernAss	56
5.6	Programmstruktur	56
5.6.1	Datenbankverbindung	57
5.6.2	Der Sessioncache	58
5.6.3	Lernerfolgsüberwachung	58
5.7	Suchfunktion	58
5.8	Vom Word-Dokument zur Webgrafik	59
5.9	Datenbank-Sicherheitskonzept	61

Abbildungen

5.1	Datenstruktur im Stern	56
5.2	Programmkomponenten	57
5.3	Datenbank Sicherheitskonzept	62

5.1 Einleitung

Die Autoren von LernAss [1] haben ein technisches Redesign bereits geplant. Die wichtigsten Vorschläge sind die Verwendung neuer Datenbanksoftware sowie eine Synchronisierung des Datenbanklayouts – die so genannte dritte Normalform des Datenbanklayouts.

Dieser Abschnitt ist an die bereits von den ursprünglichen Autoren geplanten Verbesserungsvorschläge angelehnt. Als wichtiges Designziel wurde die Änderbarkeit der Daten angemerkt. Eine einmal eingegebene Aufgabe sollte nicht nur in dem für das Internet tauglichen Format vorhanden sein, sondern auch in einem wieder bearbeitbaren Format. Deswegen werden die Daten in zwei Versionen in der Datenbank gespeichert: zum einen als Abbild der Originaldatei als Binary Large Object (BLOB) und zum anderen als Bild für die Internet-Version.

Für die Internet Version ist ein geeignetes, standardisiertes Bildformat zu wählen (siehe dazu Seite 52 in Kapitel 5.2).

5.2 Grafikformate

Die Speicherung der Aufgaben soll in Bildform erfolgen. Dazu muss ein standardisiertes, nicht kostenpflichtiges Bildformat mit geeigneten Eigenschaften gewählt werden.

Da im Grunde genommen nur Text gespeichert wird, muss die Farbanzahl mindestens 2 (schwarz/weiß) betragen. Damit auch Bilder möglich sind, sollten mindestens 16 (besser 256) Farben speicherbar sein. Eine Übertragung im Web ist durch Komprimierung schneller möglich, daher sollte das Grafikformat dies beherrschen.

Die verschiedenen Formate

GIF

- + 256 farben (wenig – aber genug)
- + komprimierbar
- – Lizenzkosten (in Deutschland nicht, da nachträgliche Patentierung nicht erlaubt ist)

JPEG

- + alle Farben
- + mittlere Bildqualität bei Text, bei Bildern gut
- + komprimierbar
- – bei hoher Komprimierung schlechte Qualität
- – generell neigt das JPEG-Verfahren zu Bildrauschen
- – der alte, abgelöste Standard fürs Web

PNG

- + alle Farben
- + sehr gute Bildqualität
- + komprimierbar
- + der neue Standard fürs Web

TIFF

- + alle Farben
- + sehr gute Bildqualität
- – gering komprimierbar
- – nicht wirklich tauglich fürs Web.

PNG ist das beste Format, daher wird es in dieser Diplomarbeit (GIF war das Format im alten LernAss) verwendet. Leider können manche Browser nicht mit der Transparenz von PNGs umgehen, daher musste in der Programmierung eine Löschung der Transparenz¹ vorgesehen werden.

Die erweiterten Funktionen von .NET, die sich hinter den „EncoderParameter“ der Klasse „Image“ verbergen, sind weder mit GIF noch mit PNG verwendbar. Daher entsteht hier auch kein Nachteil, und die Funktionen bringen auch keine wichtigen Verbesserungen.

¹Pixel für Pixel müssen die Alphakanalwerte gelöscht werden

5.3 Frontend

Als Designziel sollte ein von MS-ACCESS unabhängiges Frontend gesehen werden. Es sollten durch das Frontend keine Einschränkungen für die Module von LernAss bestehen.

5.4 Datenbearbeitung

Die Daten, die in LernAss eingegeben werden sollen, sind per Word zu erstellen. Mit einem Programm soll die Verbindung zwischen dem Word-Objekt und der Datenbank geschaffen werden. Der „Upload“ muss auch die Datenkonvertierung in das Grafikformat übernehmen.

Im Prinzip muss LernAss 2003 an dieser Stelle so ähnlich verfahren wie das alte LernAss. LernAss 2003 führt die Konvertierung aber sofort durch, wodurch auch sofort die aktuellste Version online ist.

Durch die Verwendung von .NET stellte mich dieser Punkt auf eine harte Geduldprobe. Nicht alles ist in .NET so einfach, wie es in VB6 war. Damit ist insbesondere die Verwendung des Clipboards gemeint. Während der Programmierung stellte ich fest, dass VB.NET nicht mehr dieselbe Clipboard-Klasse beinhaltet wie VB6. Dadurch konnte ich bekannte Codes nicht verwenden.

5.5 Datenbank und Tabellen

LernAss soll aus mehreren einzelnen Datenbanken bestehen, und zwar für jedes Fach eine eigene. Zur Steuerung soll eine Datenbank mit Informationen zu den untergeordneten „Aufgabendatenbanken“ verwendet werden. Diese hat die wichtigsten Daten (Datenbanknamen, Benutzer, Passwort, Informationen, Standort, Fachgebiet, Fachbereich, . . .) zu speichern. Als Identifikationsmerkmal der einzelnen Aufgabendatenbanken wird eine ID gespeichert.

Für jedes Fach muss eine eigene Datenbank angelegt werden. Dies birgt folgende Nachteile:

- Stichworte müssen in jeder Fach-Datenbank gespeichert werden (Tests mit zentraler Speicherung in der LernAss-Verwaltungsdatenbank ergaben, dass es die beste Lösung ist. In der Verwaltungsdatenbank wären die Stichworte zwar

besser aufgehoben, aber die Übersichtlichkeit und die Komplexität der Datenbankintegritätsprüfung lassen mir keine andere Wahl).

- Zusatzdaten können nur für eine Aufgabendatenbank verwendet werden (siehe oben, es ist das selbe Problem wie bei den Stichworten). Es können jedoch Links auf Webinhalte erstellt werden. Sofern man die Daten auf dem Webserver ablegt und einen Link dorthin erstellt, kann man die Zusatzdaten auch in anderen Datenbanken verwenden.
- Querverweise auf eine andere Aufgabendatenbank können nur als „Text“ erfolgen, eine Verlinkung ist nicht möglich (wieder das Problem der Integrität).

Aber durch dieses Design ergeben sich die folgenden Vorteile:

- Dozenten müssen nicht in jeder Aufgabe oder einer zusätzlichen Tabelle gespeichert werden
- Fächer und Fachgebiete müssen nicht in jeder Aufgabe oder einer zusätzlichen Tabelle gespeichert werden
- Die Aufgabendatenbanken sind in sich konsistent, und können im Falle einer Offline-Version auch einzeln weitergegeben werden. (Hierzu sollte ein spezieller Vorgang verwendet werden. Die Informationen über den Dozenten und das Fach sollten ersichtlich sein.)

5.5.1 Datenbankstruktur

Durch die Forderungen ergibt sich eine Art Sternstruktur. In der Mitte steht die zentrale Steuerdatenbank – „LernAss“ – in der die Verlinkung auf die Sternpunkte – die Aufgabendatenbanken – erfolgt (Bild [5.1](#)).

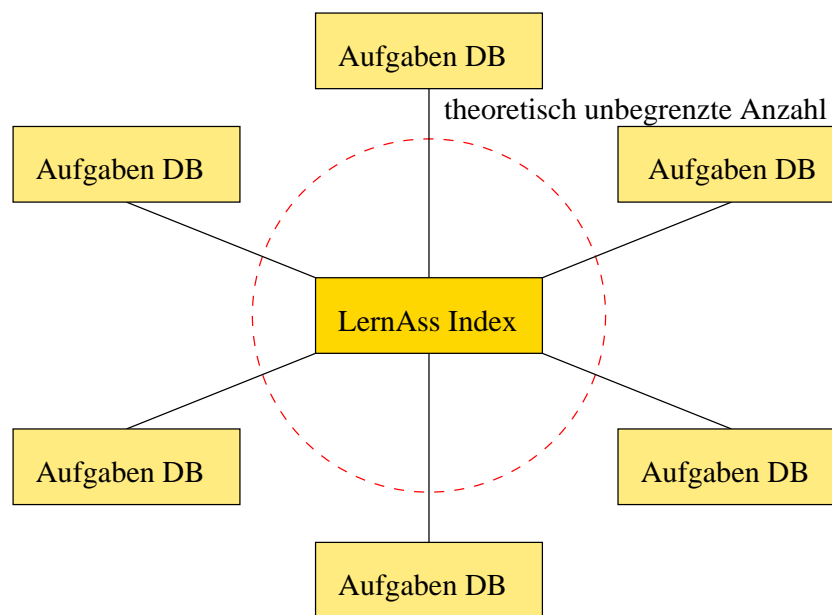


Abbildung 5.1: Datenstruktur im Stern

5.5.2 Tabellenstruktur von LernAss

Es werden keine speziellen Suffixe für Tabellennamen verwendet. Die Benennung der Tabellen und Felder ist nicht „case sensitive“ (Groß- und Kleinschreibung wird nicht unterschieden). Dennoch sollten sicherheitshalber die hier angegebenen Namen verwendet werden, sofern man eine Datenbank im „case sensitive“ Modus betreibt (Beispiel MySQL unter Unix, unter Windows nur bei Spaltennamen).

Die Schlüsselmerkmale der Tabellen werden mit dem Präfix „Key_“ versehen. Die Tabellenfelder der Tabellen haben als Präfix den Namen der Tabelle.

Die Tabellenstruktur der LernAss- und der Aufgabendatenbanken befindet sich in der Einleitung in Kapitel [1.6.8](#) auf Seite [19](#).

5.6 Programmstruktur

Die Programmierer von LernAss haben eine Trennung der Teilkomponenten von LernAss vorgeschlagen. Diese Trennung sollte in folgende Teile erfolgen:

- Visualisierung
 - Webansicht – beim User lokal (Ausgliederungsvorgang notwendig)

- Webansicht – über Internet oder Intranet

- Datenkonvertierung zur Speicherung (Userseite)
- Datenbank
- Datenbearbeitung.

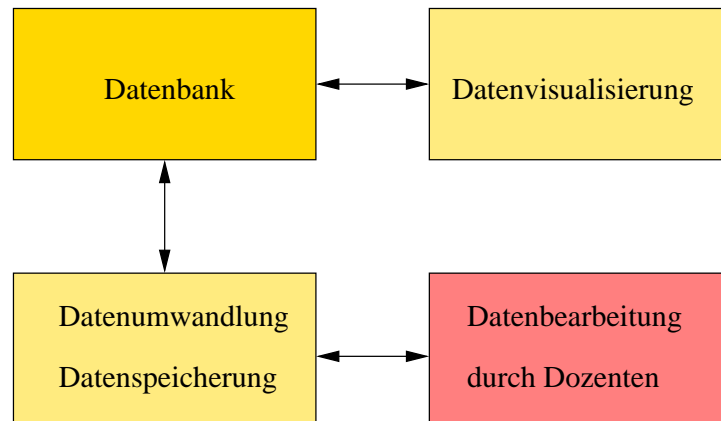


Abbildung 5.2: Programmkomponenten

5.6.1 Datenbankverbindung

Die Datenbankverbindung innerhalb der Anwendung ist entweder persistent oder dynamisch. Wichtig hierbei ist, dass bestimmte Seiten unbedingt eine persistente Verbindung zur Datenbank benötigen, da sonst nichts richtig funktioniert. Leider ist im Allgemeinen die Verwendung einer persistenten Verbindung nicht sinnvoll, da man nur eine begrenzte Anzahl an offenen Verbindungen haben darf.

Es geht an dieser Stelle um zwei Webseiten, die nicht dynamisch mit der Datenbank verbunden werden dürfen. Das sind „DB_Printout.aspx“ und „Picture.aspx“, die Ausdruckfunktion von LernAss, bei der sehr viele Datenbankverbindungen im dynamischen Modus verwendet werden könnten, was zu Performance-Problemen und außerdem zu Fehlern führen kann.

Das Konzept an dieser Stelle ist, dass „DB_Printout.aspx“ die Datenbankverbindung persistent macht, und dass im Prinzip jede andere Seite (außer „DB_Printout.aspx“ und „Picture.aspx“) die Verbindung wieder schließt, wenn das DBCon_Persistent Flag auf False gesetzt ist. Aus Performance und Sicherheitsgründen übernehmen die Seiten „DB_Exercise.aspx“ und „DB_Multiplechoice.aspx“ das Zurücksetzen der „LernAss_EDB“-Variablen im Sessioncache.

Manchmal sind die persistenten Verbindungen langsamer als die dynamischen, außerdem besteht, wie gesagt, das Problem mit zu vielen offenen Verbindungen. Durch diese Tatsachen bin ich zum Schluss gekommen, dass die Einstellung „DBCon_Persistent“ nur zu Testzwecken auf „True“ gesetzt werden darf.

5.6.2 Der Sessioncache

Der Sessioncache ist sehr wichtig für die gesamte Anwendung. Je nach vorliegendem Ablauf sind verschiedene, für das Programm wichtige Daten hier zu finden. Alles, was beim Anwendungsstart und -ende im Sessioncache passiert, wird durch die Datei „global.asax“ geregelt.

Im Sessioncache sind generell die URL von LernAss und die URL zu den Bildern von LernAss gespeichert. Für statistische Zwecke speichere ich beim Start der Session die Startuhrzeit und beim Ende die Enduhrzeit. Hier könnte man die Daten noch in einer getrennten Datenbank zur Auswertung speichern. Die Art der Datenbankverbindung und die Datenbankverbindung an sich werden hier gespeichert. Und zuallerletzt wird auch die URL, die beim Verlassen von LernAss angesprungen wird, hier eingestellt.

Wenn LernAss später einmal zu einem richtigen CMS mit Lernerfolgsüberprüfung wachsen sollte und/oder soll, so muss die Einbindung unter anderem hier im Sessioncache geschehen. Einzelne Auswertungen beruhen auf den jeweiligen Seiten.

5.6.3 Lernerfolgsüberwachung

Eine Überwachung des Lernerfolges war zum Zeitpunkt der Diplomarbeit noch nicht vorgeschrieben, ist jedoch durch ein einfaches Nachrüsten möglich.

Hierbei zu verwendende Techniken und Standards sind nicht Teil der Diplomarbeit.

5.7 Suchfunktion

Die Suchfunktion in LernAss sollte nach den Kriterien „Aufgaben Information“, „Aufgaben Level“, „Aufgaben Fachgebiet“ und nach zu definierenden Suchhilfen („Aufgaben Hinweistext“) suchen können. Als Standard sollte eine Datenbanksuche mit LEFT JOIN verwendet werden, die die Suche über „Aufgaben Information“ und „Aufgaben Hinweistext“ gleichzeitig ermöglicht.

Die Suchoptionen:

- Aufgaben Info und Hinweise
- Aufgaben Info
- Aufgaben Level
- Aufgaben Fachgebiet
- Aufgaben Hinweise.

Zur Unterstützung der Suchfunktion sollte die Datenbank Hashtabellen erstellen können (siehe dazu Seite 48 (Kapitel 4.4)). Diese sollten insbesondere über die Spalten „Aufgaben Information“ und „Aufgaben Hinweis Text“, sowie über „Aufgaben Level“ und „Aufgaben Fachgebiet“ erstellt werden.

5.8 Vom Word-Dokument zur Webgrafik

Die entscheidende Frage in LernAss ist immer noch wie man von .DOC zur Webgrafik kommt. Der Ansatz ist immer noch der gleiche. Beim alten LernAss ist der Konverter in der Bildgröße eingeschränkt, es gab feste Vorgaben. Das soll im neuen LernAss nicht mehr der Fall sein.

- Word-Dokument erstellen
- Das Dokument in der Datenbank speichern
- Inhalt des Word-Dokuments ins Clipboard kopieren
- Inhalt des Clipboards als Enhanced Metafile auslesen
- Das Enhanced Metafile in eine Grafik umwandeln
- Die Grafik in der Datenbank speichern.

Und das Ganze sollte bis auf die Erstellung der Dokumente automatisiert ablaufen. In .NET besteht wie in den Vorgängern VB1-6 das Problem mit der Konvertierung der Word-Dokumente in Grafiken. Erschwerend zu der notwendigen Typumwandlung kommt die komplette Neugestaltung der Programmiertechnik hinzu. Die aus

VB bekannten Methoden funktionieren nicht mehr, da .NET ein neues Clipboard-Objekt bekommen hat.

Im Prinzip bleibt die Kommunikation von Word mit der System-Zwischenablage gleich. In Word markierter Text wird als ASCII-Text, Unicode-Text, Richtext (RTF), HTML und Windows-Metafile in die Zwischenablage kopiert. Dann muss LernAss die Daten aus dem Clipboard-Objekt holen.

Um mit .NET auf die Daten zugreifen zu können, muss der Datentyp bekannt sein. Es gibt eine Klasse mit Beschreibungen der Datentypen → Dataformats. Leider kann man nicht einfach mit

```
Clipboard.GetDataObject().GetData(Dataformats.EnhancedMetafile)
```

das EnhancedMetafile auslesen. Dies wurde mir nach langen Telefonaten und E-Mails mit dem Microsoft-Support mitgeteilt. Es handelt sich hier um einen Bug im .NET-Framework, der erst in der nächsten Version behoben werden wird. Ich habe stattdessen die API anprogrammieren müssen, was in .NET eigentlich nicht mehr notwendig sein sollte.

Der Code für die API und das Clipboard minimiert:

```
Private objMetaFile As System.Drawing.Imaging.Metafile
Private Const CF_ENHMETAFILE = 14 'Das Object in der
    Zwischenablage ist ein Handle zu einer Enhanced Metafile

OpenClipboard(Me.Handle)
5 henhmetafile = GetClipboardData(CF_ENHMETAFILE)
If Not (IntPtr.Zero.Equals(henhmetafile)) Then
    objMetaFile = New System.Drawing.Imaging.Metafile(
        henhmetafile, True)
    PictureBox1.Image = objMetaFile
End If
10 CloseClipboard()
```

Die Fernsteuerung von Word lässt sich auf folgendes Codebeispiel minimieren:

```
Dim x As New Word.Application
Dim xx As Word.Document
xx = x.Documents.Open("C:\test.doc")
xx.SelectWholeStory
5 xx.Copy
xx.Close(False)
```

```
CType(x, Word.ApplicationClass).Quit(False)
x = Nothing
```

Beim alten Visual Basic (bis Version 6) funktionierte dieser Code:

```
If Clipboard.GetFormat(2) Or Clipboard.GetFormat(3) Or Clipboard
    .GetFormat(8) Then
    Picture1.ScaleMode = 1
    Picture1.AutoSize = True
    Picture1.AutoRedraw = True
5    Picture1.BackColor = &HFFFFFF
    Picture1.Refresh
    Picture1.Picture = Clipboard.GetData()
    Clipboard.Clear
    Clipboard.SetData Picture1.Picture
10 End If
```

Der hier angegebene Code konvertiert die Metadaten aus dem Clipboard in eine Grafik. Die Speicherung ginge damit aber auch.

Eine weitere Möglichkeit wäre das Auslesen des HTML-Quelltextes, den Word beim Kopieren in die Zwischenablage erstellt. Dabei werden eingebettete Objekte in Grafiken umgewandelt. Jedoch ist mir nicht bekannt, ab welcher Word-Version die Funktion eingebunden wurde.

5.9 Datenbank-Sicherheitskonzept

Zur Absicherung der Zugriffe auf die Datenbank wurde folgendes Modell entwickelt:

Die Steuerdatenbank „LernAss“ ist für den Benutzer „LernAss“ mit dem Passwort „LernAss“ mit den Rechten für SELECT, INSERT, UPDATE und DELETE, und für den Datenbankserver mit CREATE, DROP, GRANT und RELOAD Rechten versehen.

Die Aufgabendatenbanken müssen nur für den Dozenten einsehbar sein. Daher erfolgt der Zugriff für den Benutzer mit Passwort und den Rechten SELECT, INSERT, UPDATE und DELETE.

Ein Superuser hat das volle Zugriffsrecht auf die Datenbank. Das ist notwendig, damit im Fehlerfall auch wirklich auf alles zugegriffen werden kann.

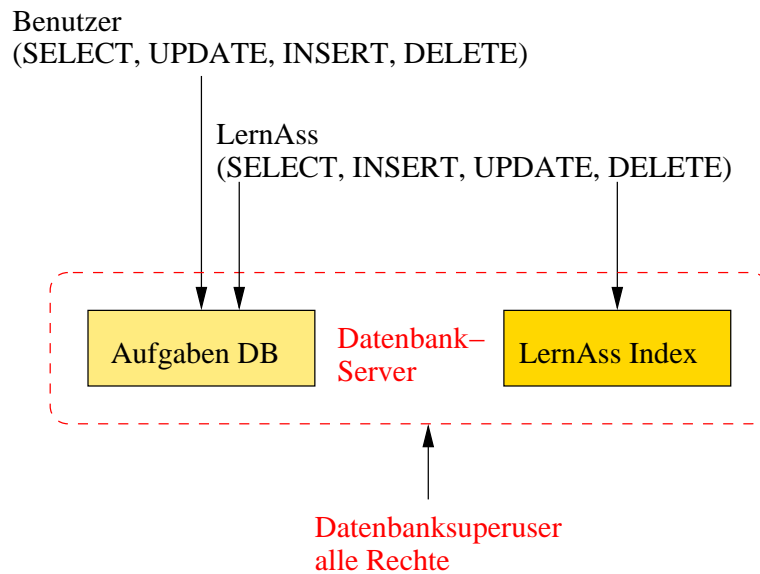


Abbildung 5.3: Datenbank Sicherheitskonzept

Die wichtigsten Rechte haben folgende Bedeutungen:

Tabelle 5.1: Die Rechte in der Datenbank

Recht	Bedeutung
CREATE	Recht zum Erstellen von Datenbanken und Tabellen
DELETE	Recht zum Löschen von Datensätzen
DROP	Recht zum Löschen von Datenbanken
FILE	Recht zum Öffnen von Dateien für den Upload in die Datenbank
GRANT	Recht zum Vergeben von Berechtigungen an Benutzer
INDEX	Recht zum Erstellen von Indizes über Tabellenspalten
INSERT	Recht zum Einfügen von Datensätzen in eine Datenbanktabelle
REFERENCES	Recht zum Erstellen von Referenzen
SELECT	Recht zum Anzeigen von Datensätzen aus der Datenbank
UPDATE	Recht zum Aktualisieren von Datensätzen in der Datenbank

6 LernAss Programme und Komponenten

Inhaltsangabe

6.1	Einleitung	65
6.2	LernAss Konverter	65
6.2.1	Aufbau	65
6.2.2	Bedienung	67
6.3	LernAss Web	69
6.3.1	Aufbau	69
6.3.2	Bedienung	71
6.4	LernAss Admin	78
6.4.1	Aufbau	78
6.4.2	Bedienung	78
6.5	LernAss Exerciser	86
6.5.1	Aufbau	86
6.5.2	Bedienung	86
6.6	LernAss Komponenten	97
6.6.1	ADODB_LIB	97
6.6.2	LA_EMF_Clip	97
6.6.3	MySQL_DB	97
6.6.4	MySQLClient	106

Abbildungen

6.1	LernAss Konverter	68
6.2	LernAss Web-Startseite	71

6.3	LernAss Web Aufgabendatenbank Ansicht	72
6.4	LernAss Web-Aufgabenübersicht	73
6.5	LernAss Web Multiple Choice-Test	74
6.6	LernAss Web Multiple Choice-Test Auswertung	75
6.7	LernAss Web-Aufgabenausdruckfunktion	76
6.8	LernAss Web-Suchfunktion	77
6.9	LernAss Admin Anmelden/Abmelden	79
6.10	LernAss Admin Datenbanken-Übersicht	80
6.11	LernAss Admin Datenbanken anlegen	81
6.12	LernAss Admin Datenbanken ändern	82
6.13	LernAss Admin Link-Übersicht	83
6.14	LernAss Admin Links anlegen	84
6.15	LernAss Admin Links ändern	85
6.16	LernAss Exerciser anmelden/abmelden	87
6.17	LernAss Exerciser Aufgaben	88
6.18	LernAss Exerciser Aufgabenbearbeitung	89
6.19	LernAss Exerciser Aufgabenverlinkung	90
6.20	LernAss Exerciser Aufgabenlevel	91
6.21	LernAss Exerciser Fachgebiete	92
6.22	LernAss Exerciser Aufgabenliste	93
6.23	LernAss Exerciser Aufgabenlisten	94
6.24	LernAss Exerciser Stichworte	95
6.25	LernAss Exerciser Links	96

Tabellen

6.1	LernAss V2.2 Schemadateiformat	67
6.2	Die Dateien von LernAss Web und ihre Funktion	70
6.3	Funktionsliste ADODB_LIB	97
6.4	Funktionsliste LA_EMF_Clip	97
6.5	MySQL_DB Datenbankverbindungsbehandlung	98
6.6	MySQL_DB WEB-Funktionen für die Datenbanken	98
6.7	MySQL_DB WEB-Link-Funktionen	98
6.8	MySQL_DB WEB-Aufgabenfunktionen	99

6.9	MySQL_DB WEB-Aufgabensuchfunktionen	99
6.10	MySQL_DB LernAss Admin-Datenbankfunktionen	100
6.11	MySQL_DB LernAss Admin / LernAss Exerciser-Linkfunktionen	100
6.12	MySQL_DB LernAss Exerciser-Extrafunktionen	100
6.13	MySQL_DB LernAss Exerciser-Aufgabenfunktionen	101
6.14	MySQL_DB LernAss Exerciser-Aufgabenverlinkung	101
6.15	MySQL_DB LernAss Exerciser Aufgabenlisten	102
6.16	MySQL_DB LernAss Exerciser-Lösungen	102
6.17	MySQL_DB LernAss Exerciser Multiple Choice	103
6.18	MySQL_DB LernAss Exerciser Source Up-/Download	103
6.19	MySQL_DB LernAss Exerciser-Bildfunktionen	103
6.20	MySQL_DB LernAss Exerciser-Hintfunktionen	104
6.21	MySQL_DB LernAss Exerciser-Topicfunktionen	104
6.22	MySQL_DB LernAss Exerciser Levelfunktionen	104
6.23	MySQL_DB Generelle Ausleseoperationen	105
6.24	MySQL_DB Datenbankrechteverwaltung	105
6.25	MySQL_DB Basisdatenbankbearbeitungsoperationen	105

6.1 Einleitung

Dieses Kapitel gibt einen Überblick und eine Anleitung über die LernAss Programme und Komponenten. Hierin enthalten sind Bedienungshinweise und Aufbaubeschreibungen, die die Funktionalität der Software erläutern.

6.2 LernAss Konverter

6.2.1 Aufbau

Der LernAss Konverter ist im Prinzip ein Eingabe- und Ausgabe-Programm, das direkt aus den „alten“ LernAss-Datenbanken liest und über eine ganz marginale Konvertierung in das neue Datenbank-Format durchführt.

Das Programm baut per Direktzugriff eine Verbindung zur Datenbank auf, und muss folglich auf dem Datenbankserver ausgeführt werden. Diese Einschränkung erachte

ich als sinnvoll, da man die alten Datenbanken ja nur auf dem Server benötigt und man sonst die Software unnötig verkomplizieren würde, wenn man nicht von diesem Szenario ausgeht.

Die Inkonsistenz mancher alter LernAss-Datenbanken erweist sich hier als problematisch. So sind zum Beispiel Doppeleinträge in der „Topics“-Tabelle für den Konverter eine Hürde. Man kann dieses Problem programmiertechnisch umgehen und es ist auch ein simpler Weg implementiert. Die Behebung eines solchen Problem es lässt sich mit MS-ACCESS am besten von Hand durchführen, denn es kann zum Beispiel ja auch ein Doppeleintrag mit unterschiedlicher Schreibweise vorkommen, der mit einem Programm schwerer zu erkennen ist, als von einem Menschen.

Beispiel 1:

Die Datenbank „Mathe“ von Herrn Prof. Rüdiger Ziehten hat das Gebiet Trigonometrie. Dieses wird durch einen Eingabefehler mit „Trig“ und „Trigo“ als Kurzform in der Datenbank gehalten und auch noch mit beiden Kurzformen referenziert. Der LernAss Konverter kennt nur die Langform und meldet einen Fehler, da zu der Langform zwei Schlüssel gemeldet werden.

Die Lösung ist, die Schlüsselinkonsistenz zu beheben. Man löscht den Schlüssel mit der Kurzform „Trigo“ und benennt alle „Trigo“ in „Trig“ um. Das muss man aber nicht, da der LernAss Konverter diesen Umstand berücksichtigt.

Beispiel 2:

Die Datenbank Physik von Herrn Prof. Dr. rer. nat. Manfred Merkel hat eine Inkonsistenz zwischen den Tabellen „Aufgaben“ und „Index“. Die Aufgabe mit der Nummer 331 taucht in der Tabelle „Aufgaben“ auf, nicht aber in der Tabelle „Index“. Dadurch haben meine SQL-Abfragen beim Konvertieren der Aufgaben in Verbindung mit beiden Tabellen die Aufgabe übergangen. Der hier verwandte Join erlaubt absichtlich (sonst fehlen Daten) nur Tupel, die in beiden Tabellen vorhanden sind.

Da ich bei den folgenden Abfragen (Multiple Choice, Lösungen und Aufgabenquellen) nicht mehr auf die Tabelle „Index“ angewiesen war, habe ich diese in den SQL-Abfragen auch nicht mehr verwendet. Dies stellte sich bei dieser Inkonsistenz als Fehler heraus. Die Fehlerbehebung verursacht hier einen deutlichen Performanceeinbruch beim Konvertieren, da ein Join über zwei Tabellen erfolgen muss.

Schemadaten:

Die LernAss V2.2-Schemadateien müssen vom Konverter geladen und konvertiert werden können. Hierzu habe ich das Dateiformat der Schemadatei entschlüsselt:

Tabelle 6.1: LernAss V2.2 Schemadateiformat

Zeile	Inhalt
1	Bezeichnung des Schematyps (LernAss-Vorauswahlschema V2.2)
2	Bezeichnung des Fachs
3	Bezeichnung des Dozenten
4 (1)	Aufgaben-ID
6 (2)	Fach
7 (3)	Fachgebiet kurz
8 (4)	Fachgebiet lang
9 (5)	Level
10 (6)	Boolean
11 (7)	Boolean
12 (8)	Boolean
13 (9)	Boolean
*	immer ein 9-Zeilenblock mit je einer Aufgabe

Der Aufgabenblock ist redundant – Die Zeilen zwei bis neun sind in der Datenbank wiederzufinden und werden aus Konsistenzgründen nicht verwendet.

Der LernAss Konverter benötigt Benutzernamen und Passwort eines Benutzers, der den Vollzugriff zur Datenbank hat (dieses wird einmal abgefragt). Groß- und Kleinschreibung wird unterschieden. Die Datenbank wird in die LernAss-Steuerdatenbank eingetragen und als aktiv markiert. Der Benutzername und das Passwort für die neue Datenbank werden auf den Namen des Dozenten festgelegt.

Der LernAss Konverter verwendet ADODB_LIB, LA_EMF_Clip, MySQL_DB und MySQLClient.

6.2.2 Bedienung

Wichtig! Der Konverter macht manchmal eine längere Pause, wobei Word ohne Dokument offen ist. Das ist normal und liegt an der Word API und auch an dem Konvertierungsvorgang, der bei großen Bildern auch länger als eine Minute dauern kann. Der Konverter macht sich intensiv den „TEMP“-Ordner des angemeldeten Benutzers zu Nutze. Durch die Bearbeitungsgeschwindigkeit und die Angewohnheiten von Word können nicht alle Tempdaten automatisch wieder gelöscht werden.

Daher ist es sinnvoll, nach einem Durchlauf des LernAss Konverters das temporäre Verzeichnis zu leeren.

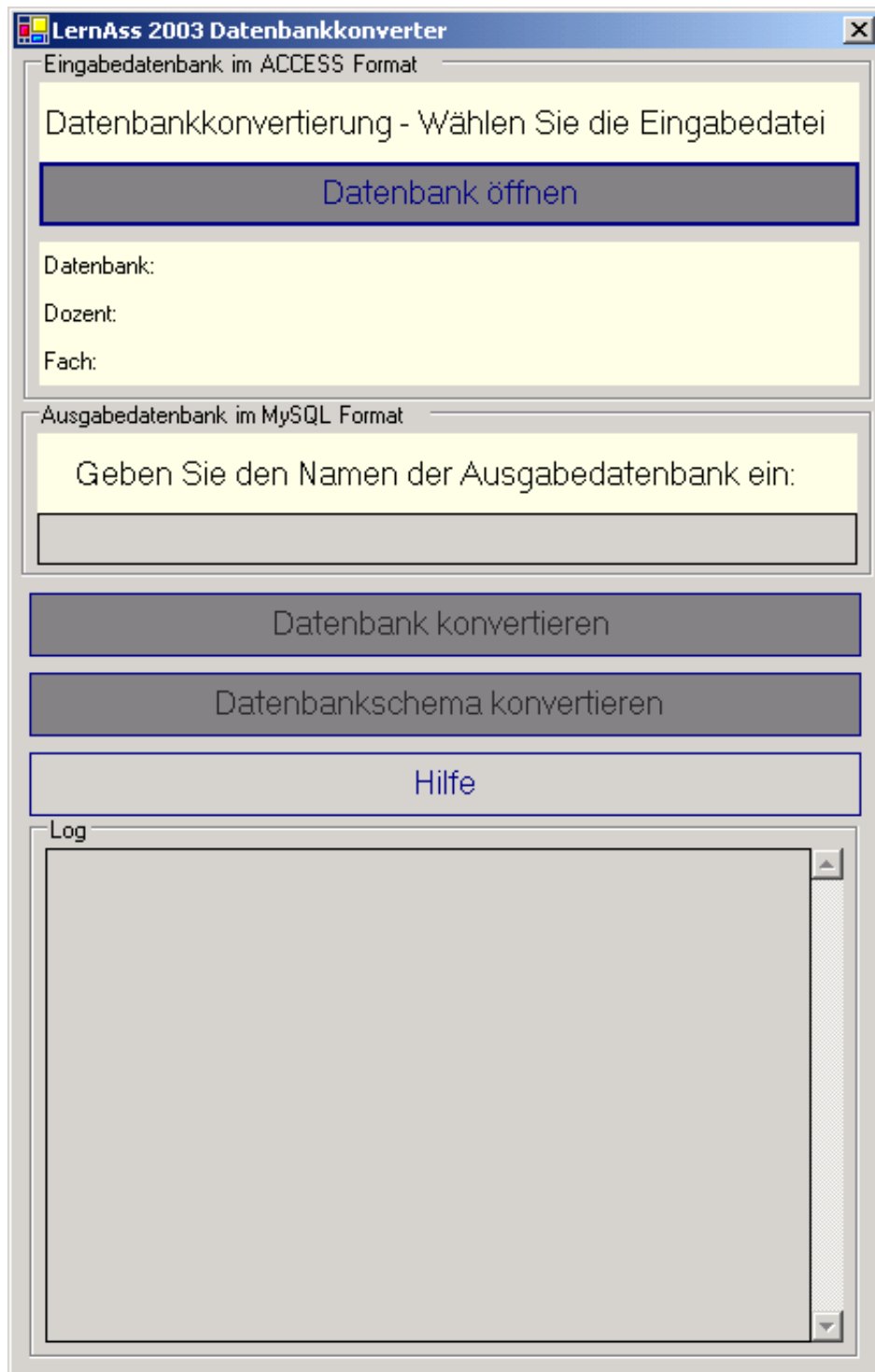


Abbildung 6.1: LernAss Konverter

Der Knopf „Datenbank öffnen“ dient zum Öffnen von alten LernAss-Datenbanken. Mit dem Knopf „Datenbank konvertieren“ fragt man beim ersten Aufruf nach dem Benutzernamen und Passwort für Vollzugriff auf die Datenbank und konvertiert die alte Datenbank in das neue Format. Der Knopf „Schemadaten importieren“ wird zum Importieren von „LernAss Vorauswahlschema V2.2“ benötigt. Der Knopf „Hilfe“ zeigt noch kurze Informationen an.

Zu beachten ist auch, dass der Konvertierungsvorgang nicht abgebrochen werden kann. Insbesondere beeinträchtigt dies die Verwendbarkeit der Zwischenablage während der Konvertierung.

6.3 LernAss Web

6.3.1 Aufbau

Die Kernfunktion der Webkomponenten ist die Visualisierung der Aufgaben. Dazu werden in einer Sessionstate die Datenbankverbindungen über jeweils eine Sitzung hinweg offen gehalten, sofern das per Flag „DB_Persistent=True“ angegeben wird (siehe Kapitel 5.6.1 auf Seite 57). Außerdem werden die benötigten Daten für alle Webseiten darin vorgehalten. Das sind URLs und Zugangsdaten, je nachdem, welche Funktion gerade verwendet wird.

Das Grundprinzip ist ein Framedesign, damit man die Seiten einfach gestalten und die Komponenten leichter austauschen kann.

Die Webseiten bauen per TCP/IP oder Direktzugriff eine Verbindung zur Datenbank auf, die Webseite läuft idealerweise auf dem Datenbankrechner.

Um die Cache-Funktion der Browser auszutricksen, musste ich „Picture.aspx“ und „DB_Picture.aspx“ auch die Datenbank-ID mitgeben. Sonst zeigen die Browser noch nicht angezeigte Aufgaben unter Umständen als angezeigt an.

LernAss Web verwendet MySQL_DB und MySQLClient

Tabelle 6.2: Die Dateien von LernAss Web und ihre Funktion

Datei	Funktion
Basics.vb	Grundfunktionen der Webseiten (Datenbankverbindungs-Auf- und -Abbau)
DB.Exercise.aspx	Anzeige der Aufgaben der ausgewählten Datenbank
DB.Exerciseout.aspx	Anzeige einer Aufgabe mit allen Lösungen
DB.Index.aspx	Anzeigen einer Datenbank in einem Frame mit DB_Links.aspx, DB_Main.aspx und Identity.aspx
DB.Links.aspx	Die Linksleiste im Datenbankmodus
DB.Main.aspx	Die Hauptseite im Datenbankmodus – Anzeige der möglichen Optionen
DB.Multiplechoice.aspx	Anzeige der Multiple Choice-Aufgaben der ausgewählten Datenbank
DB.Multiplechoiceout.aspx	Anzeige einer Multiple Choice-Aufgabe mit ihren Lösungen
DB.Multiplechoicetest.aspx	Auswertung des Multiple Choice-Tests
DB.Picture.aspx	Die Seite mit dem Bild einer Aufgabe, Lösung oder Multiple Choice-Lösung und der Druckmöglichkeit
DB.Print.aspx	Anzeige der Aufgaben der ausgewählten Datenbank zwecks Ausdruck
DB.Printout.aspx	Bearbeitung von DB_Print.aspx gesendeten Daten und Ausdruck
DB.Search.aspx	Suche in den Aufgaben
Default.aspx	Die Startseite von LernAss in einem Frame mit Links.aspx, Main.aspx und Identity.aspx
Exit.aspx	Abmeldeseite
Global.asax	Die Sessionstate-Daten und MySQL-Verbindungen
Identity.aspx	Die Corporate Identity Datei – oberer Frame
LernAssWeb.vsdisco	Discovery XML-Informationen
Links.aspx	Die Linkleiste im Begrüßungsbildschirm
Main.aspx	Das Hauptfenster im Begrüßungsbildschirm
Picture.aspx	Anzeigen eines Bildes aus einer Datenbank
Styles.css	CSS-Daten für die Applikation
Web.config	Konfiguration der Applikationsbehandlung

6.3.2 Bedienung

Zunächst wird LernAss Web durch den IIS von Microsoft aufgerufen. Das heißt man verwendet einen Browser und übergibt diesem die URL der LernAss Web-Applikation. Dadurch wird ein Sessionobjekt erzeugt, in dem LernAss Web wichtige Daten wie die Datenbankverbindung speichert. Die Funktionen von LernAss Web sind in einzelne ASPX-Seiten aufgeteilt, die mit Programmcode ausgestattet sind.

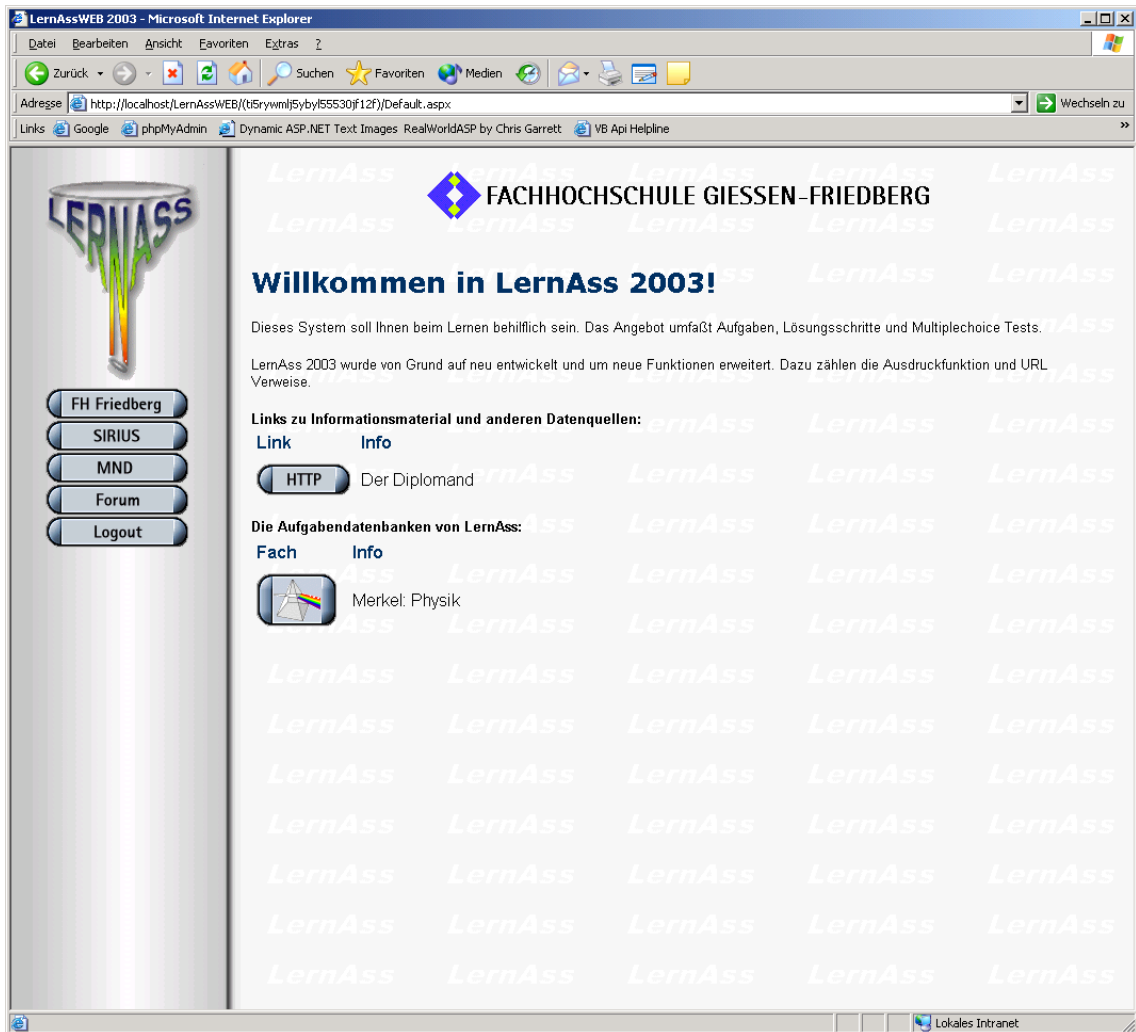


Abbildung 6.2: LernAss Web-Startseite

Die erste aufgerufene Seite ist ‚Default.aspx‘, die ein Framset aus ‚Links.aspx‘, ‚Identity.aspx‘ und ‚Main.aspx‘ aufbaut. Mit der ‚Links.aspx‘ Seite werden Navigationsmöglichkeiten gegeben. ‚Identity.aspx‘ ist als Identitätsmerkmal eingebaut – im Falle der LernAss-Anwendung in der Fachhochschule Gießen-Friedberg ist hier

das Fachhochschul-Logo eingebunden. „Main.aspx“ listet Daten aus der Datenbank auf, dazu zählen Links zu anderem Informationsmaterial und die Verknüpfung der Aufgabendatenbanken.

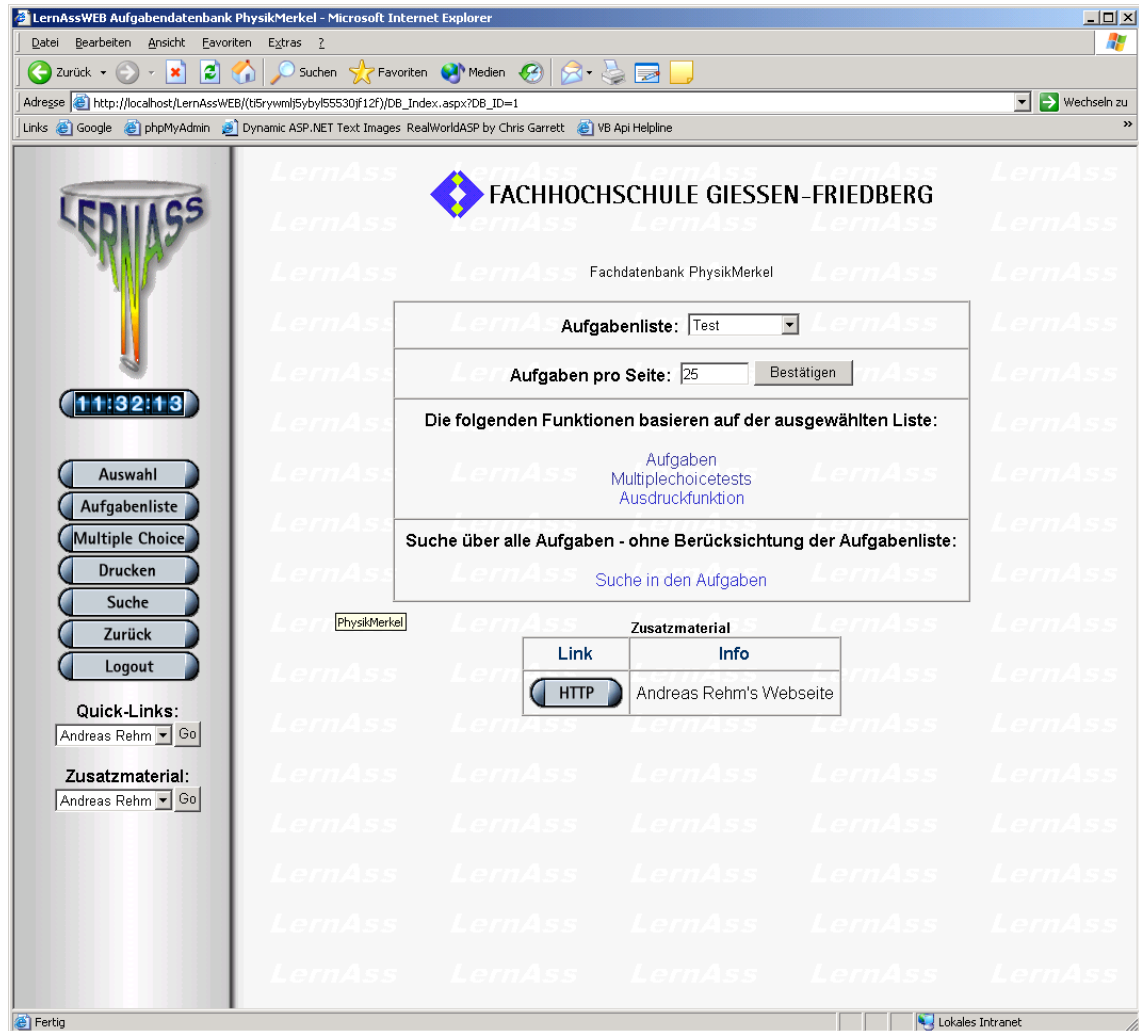


Abbildung 6.3: LernAss Web Aufgabendatenbank Ansicht

Eine Aufgabendatenbank wird mit der Seite „DB_Index.aspx“ geöffnet, welche ein Framset anlegt und die Datenbankverbindung aufbaut. Das Framset besteht aus den Seiten „DB_Main.aspx“, „DB_Links.aspx“ und „Identity.aspx“. „DB_Main.aspx“ gibt die möglichen Funktionen der Datenbank wieder. Außerdem werden hier auch wieder Links angezeigt, die in der Datenbank abgelegt sind. „DB_Links.aspx“ ist eine auf die Datenbank angepasste Linkleiste, die außerdem die Links der „LernAss“ und der geöffneten Aufgabendatenbank anzeigt.

Hier wählt man die gewünschte Aufgabenliste und die Anzahl der anzuzeigenden Aufgaben.

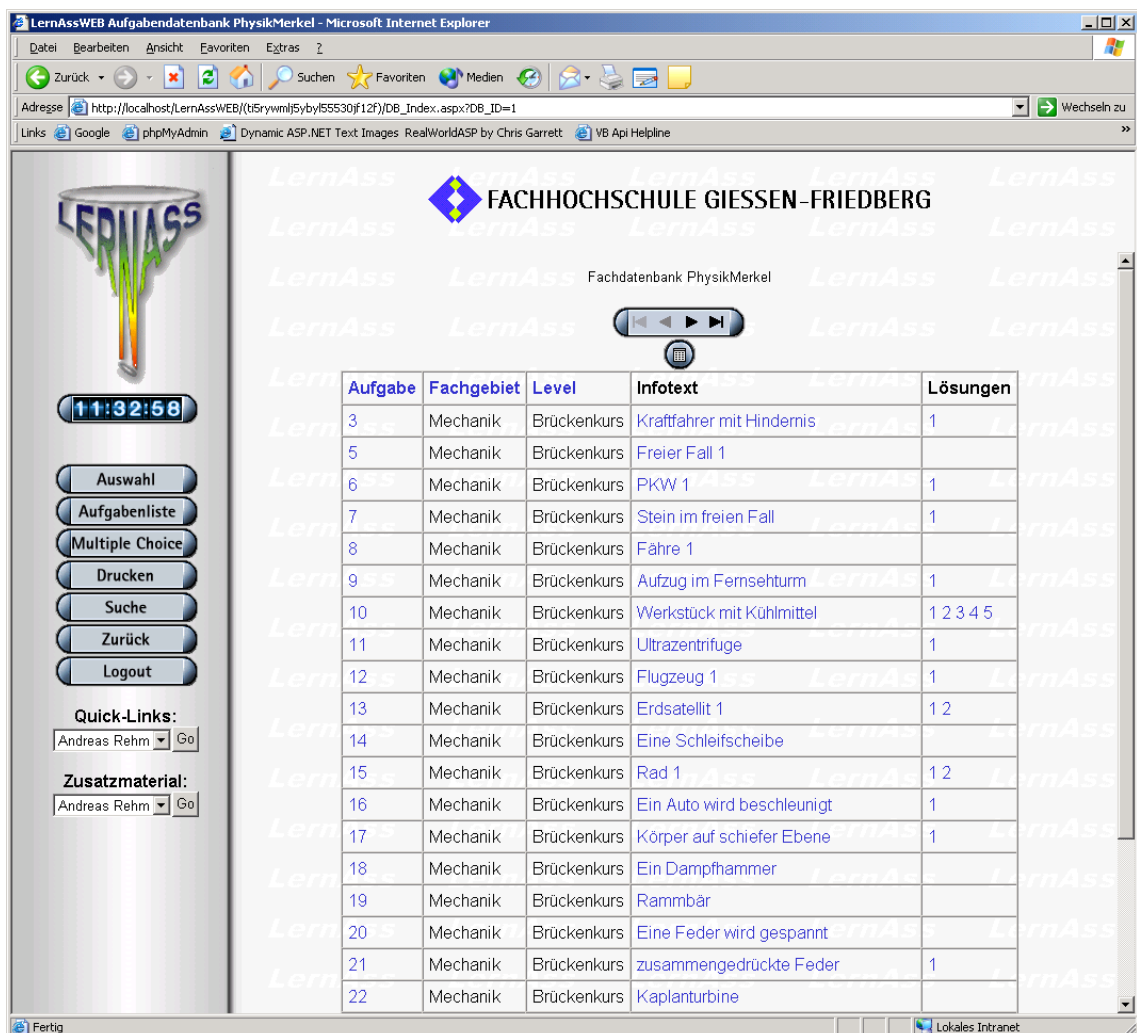


Abbildung 6.4: LernAss Web-Aufgabenübersicht

Die Aufgabenübersicht erlaubt die Navigation und die Anzeige der Aufgaben.

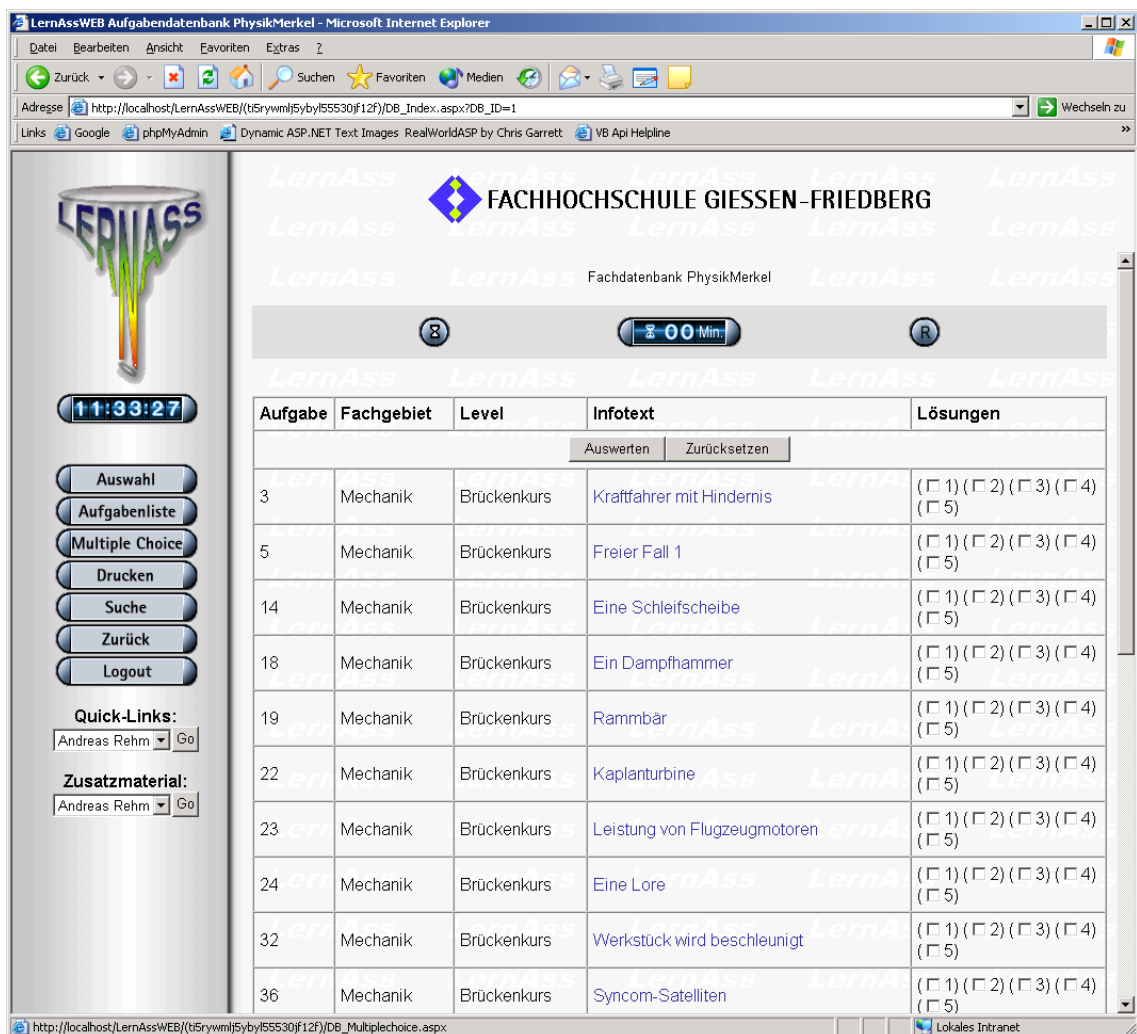


Abbildung 6.5: LernAss Web Multiple Choice-Test

Der Multiple Choice-Test ist im Prinzip genauso aufgebaut wie die Aufgabenübersicht. Hinzugekommen ist eine Stoppuhr.

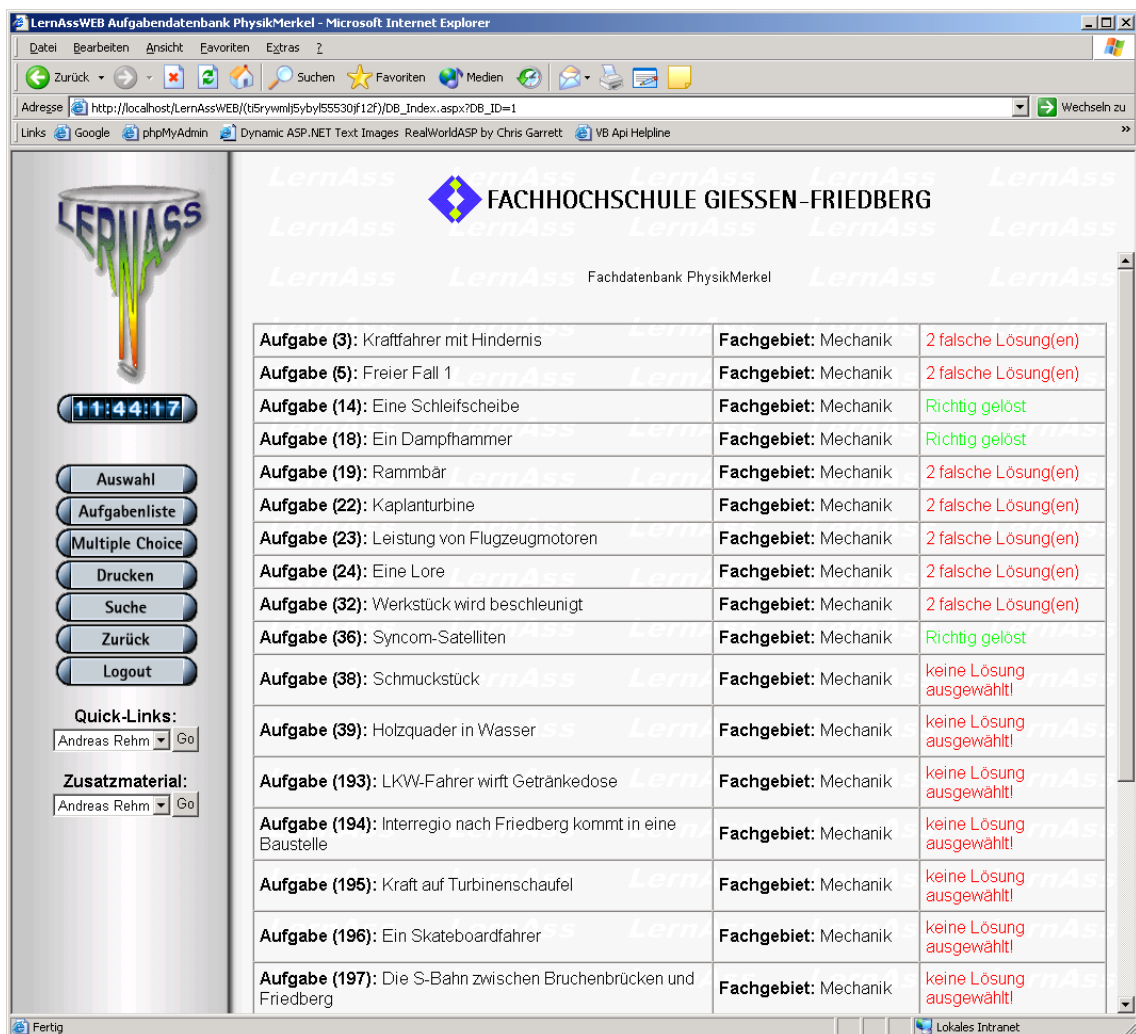


Abbildung 6.6: LernAss Web Multiple Choice-Test Auswertung

Die Auswertung eines Multiple Choice-Tests sieht aus wie abgebildet.

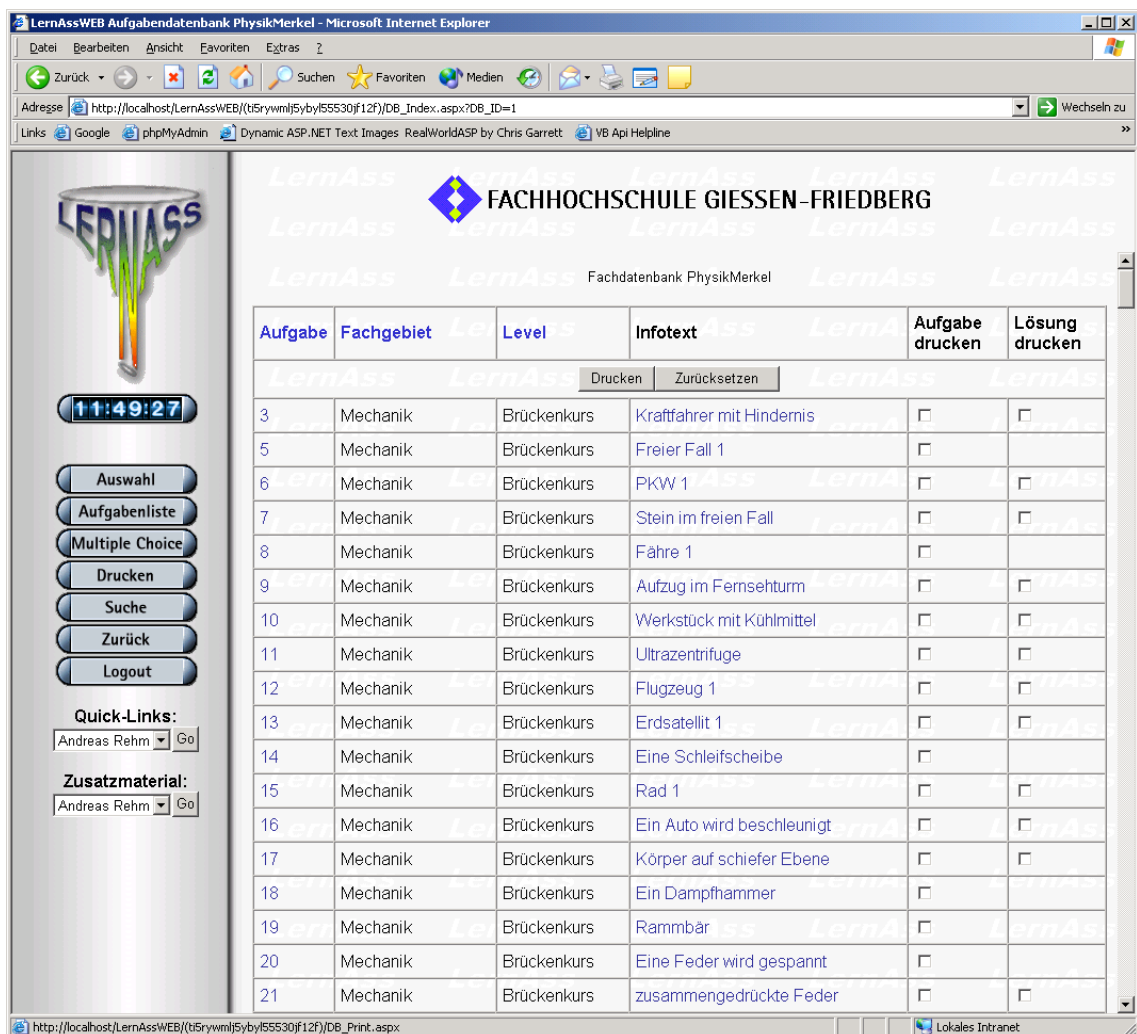


Abbildung 6.7: LernAss Web-Aufgabenausdruckfunktion

Die Aufgaben können auch über ein Formular ausgewählt und gedruckt werden.

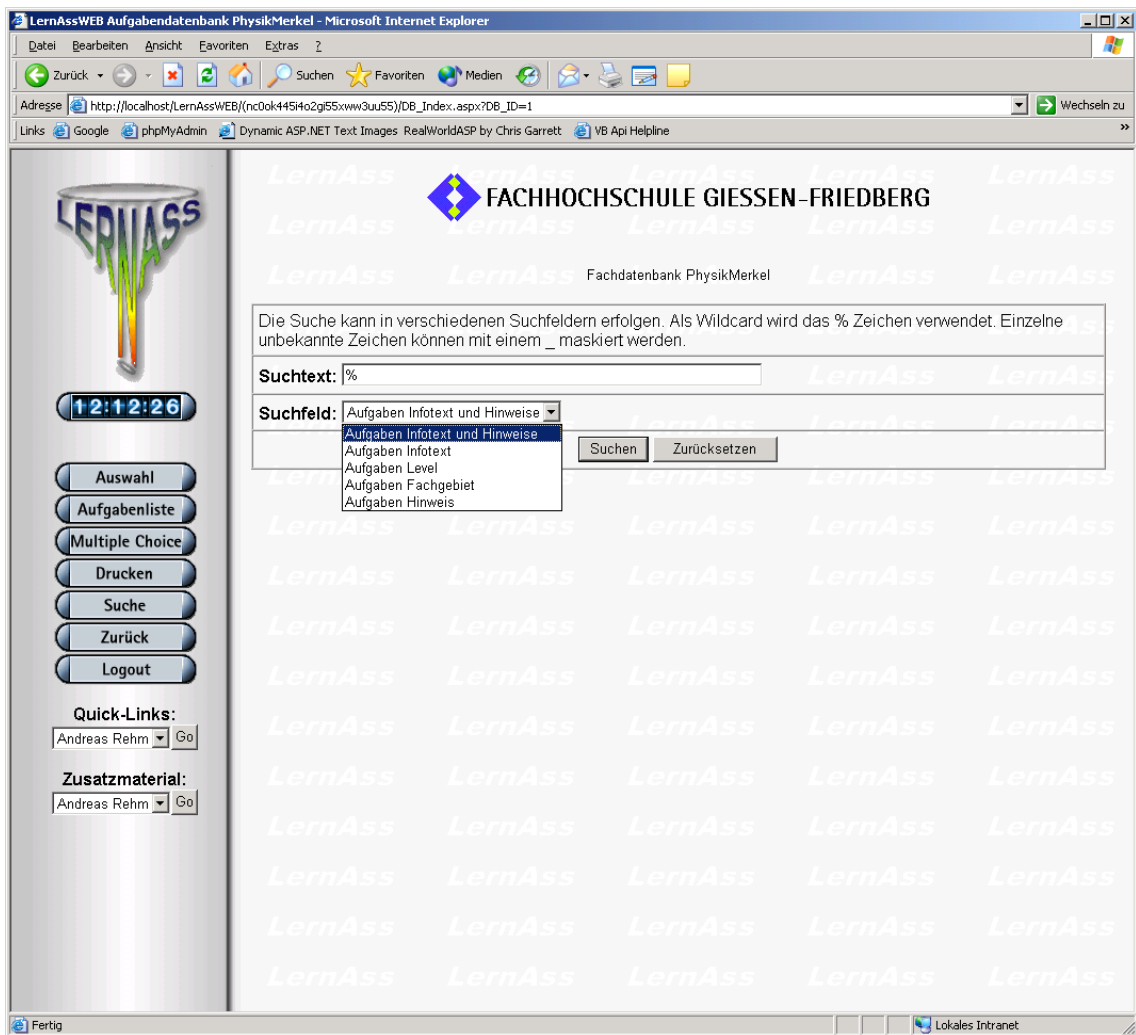


Abbildung 6.8: LernAss Web-Suchfunktion

Um die Suche nach einer Aufgabe zu erleichtern, ist ein Suchformular mit verschiedenen Suchmöglichkeiten vorhanden.

6.4 LernAss Admin

6.4.1 Aufbau

Das LernAss Admin-Programm dient zur Benutzer- und Datenbankverwaltung sowie zur Erstellung von Links auf der Hauptseite von LernAss Web. Hiermit werden die Aufgabendatenbanken in die LernAss-Steuerdatenbank eingepflegt und die Zugriffsberechtigungen für die Benutzer gesetzt.

Das Programm baut per TCP/IP oder Direktzugriff eine Verbindung zum Server auf, und muss folglich Zugriff darauf haben können.

LernAss Admin verwendet MySQL_DB und MySQLClient.

6.4.2 Bedienung

Um LernAss Admin verwenden zu können, muss man sich mit der LernAss-Zentraldatenbank verbinden. Der Zugriff muss unbedingt mit Vollzugriffsrechten ausgestattet sein, damit man neue Datenbanken anlegen kann.

Beim Start vom LernAss Admin erscheint daher erst einmal ein Anmelde-System.

LernAss Admin

Anmelden/Abmelden | Datenbanken | Links

Datenbank Login

Um mit dem LernAss Admin-Programm auf die Datenbank zugreifen zu können, benötigt man Benutzernamen und Passwort für den Administrator-Vollzugriff auf die Datenbank.

Anmelden

Benutzername:

Passwort:

Serverstandort:

Datenbankname:

Anmeldedaten merken

Passwort merken

Alle Datenbanken optimieren

LernAss Datenbank optimieren

Anmelden

Abmelden

Anmeldedaten löschen

Beenden

Abbildung 6.9: LernAss Admin Anmelden/Abmelden

An dieser Stelle meldet man sich mit dem „Anmelden“-Knopf an und gegebenenfalls mit dem „Abmelden“-Knopf wieder ab. Außerdem kann man hier bestimmen, dass die Einstellungen gespeichert werden. Sinnvoll ist hauptsächlich nur die Speicherung der Zugangsdaten ohne Passwort, jedoch ist die Möglichkeit zur Speicherung des Passwortes gegeben (Sicherheitsrisiko!!!).

Nachdem man auf den „Anmelden“-Knopf gedrückt hat wird man auf die „Datenbanken“-Übersicht weitergeleitet.



Abbildung 6.10: LernAss Admin Datenbanken-Übersicht

In der Datenbanken-Übersicht kann man neue Datenbanken anlegen, löschen oder auch ändern. Außerdem kann man hier die Optimierung der Datenbanken veranlassen. Insbesondere wird durch die Optimierung in allen LernAss-Aufgabendatenbanken jeder verwaiste Eintrag gelöscht.

Die Maske ändert sich je nach Operation, und beim Löschen einer Datenbank bekommt man vorher noch eine Sicherheitsabfrage.

The screenshot shows a web-based administrative interface for a database management system. The window title is "LernAss Admin Datenbearbeitungsfenster". The main heading is "Datenbank Wartungsoptionen". The interface is divided into several sections:

- Datenbanken:** A list of existing databases with checkboxes. Two are checked: "DD1Boergens2" and "PhysikMerkel".
- Action Buttons:** A vertical stack of buttons: "Datenbank anlegen", "Daten ändern", "Änderungen speichern", "Vorgang abbrechen", and "Datenbank löschen".
- Datenbank ID:** A field containing the value "21".
- Datenbank Name:** An empty text input field.
- Datenbank Standort:** A text input field containing "localhost".
- Datenbank Benutzer:** An empty text input field.
- Datenbank Passwort:** An empty text input field.
- Datenbank Informationstext:** An empty text area.

Abbildung 6.11: LernAss Admin Datenbanken anlegen

Beim Anlegen einer neuen Datenbank prüft das Programm, ob die Datenbank schon vorhanden ist. Sollte die Datenbank schon vorhanden sein, wird ein Import versucht (die Datenbank wird auf der Webseite angezeigt); ist das jedoch nicht der Fall, so wird die Datenbank mit allen Tabellen neu angelegt.



Abbildung 6.12: LernAss Admin Datenbanken ändern



Abbildung 6.13: LernAss Admin Link-Übersicht

In der Link-Übersicht kann man neue Links anlegen, Links löschen oder auch ändern. Die Maske ändert sich je nach Operation, und beim Löschen eines Links bekommt man vorher noch eine Sicherheitsabfrage.

LernAss Admin

Anmelden/Abmelden | Datenbanken | **Links**

Linkverwaltung

Links	Link hinzufügen
Andreas Rehm	Link ändern
	Link speichern
	Vorgang abbrechen
	Link löschen
Link ID	
1	
Name	Kurzinfo
Andreas Rehm	Der Diplomand
Webseitenadresse als URL	Bild URL
http://www.andreasrehm.de	Images/http.png

Hier kann man Links auf Webseiten ablegen. Diese können entweder lokal im InetPub-Verzeichnis oder auf entfernten Rechnern liegen.

Abbildung 6.14: LernAss Admin Links anlegen

Beim Anlegen eines Links wird auch die Link-URL geprüft. Wenn die URL schon in der Datenbank vorhanden ist, wird das Anlegen abgebrochen.



Abbildung 6.15: LernAss Admin Links ändern

Damit sind die grundlegenden Funktionen von LernAss Admin erklärt – viel mehr macht dieses Programm nicht. Es dient einzig und allein der Administration der LernAss Webseite und den darin aufgelisteten Datenbanken. Wichtig ist, dass man möglichst mit vollen Datenbankrechten zugreifen sollte. Um die Sicherheit zu erhöhen, begrenzt man den LernAss-Benutzer auf nur lesende Operationen und arbeitet hier nur mit dem Datenbankadministrationsbenutzer.

6.5 LernAss Exerciser

6.5.1 Aufbau

Der LernAss Exerciser ist das Programm zum Verwalten der Aufgabendatenbanken. Hiermit werden die Aufgaben in die Datenbank übertragen und die Einstellungen wie Verlinkung, Extras und Stichwörter gesetzt.

Das Programm baut per TCP/IP oder Direktzugriff eine Verbindung zur Datenbank auf und muss folglich Zugriff darauf haben können.

Der LernAss Exerciser verwendet LA_EMF_Clip, MySQL_DB und MySQLClient.

6.5.2 Bedienung

Beim Start vom LernAss Exerciser erscheint wie beim LernAss Admin erst einmal ein Anmeldesystem.

LernAss Exerciser

Anmelden/Abmelden | Aufgaben | Aufgabenverlinkung | Aufgabenlevel
Fachgebiete | Aufgabenliste | Aufgabenlisten | Stichworte | Links

Datenbank Login

Um mit dem LernAss Exerciser-Programm auf die Datenbank zugreifen zu können, benötigt man Benutzernamen und Passwort für den Dozenten-Zugriff auf die Datenbank.

Anmelden

Benutzername:

Passwort:

Serverstandort:

Datenbankname:

Anmeldedaten merken

Passwort merken

Abbildung 6.16: LernAss Exerciser anmelden/abmelden

An dieser Stelle meldet man sich an und gegebenenfalls wieder ab. Außerdem kann man hier bestimmen, dass die Einstellungen gespeichert werden. Die Möglichkeit zur Speicherung des Passwortes ist ein Sicherheitsrisiko. Nachdem man auf den „Anmelden“-Knopf gedrückt hat, wird man auf die „Aufgaben“-Übersicht weitergeleitet.

Man kann hier auch die Optimierung der Datenbank veranlassen. Dabei werden jedoch nur die Überhänge der Tabellen optimiert, nicht jedoch verwaiste Einträge entfernt.

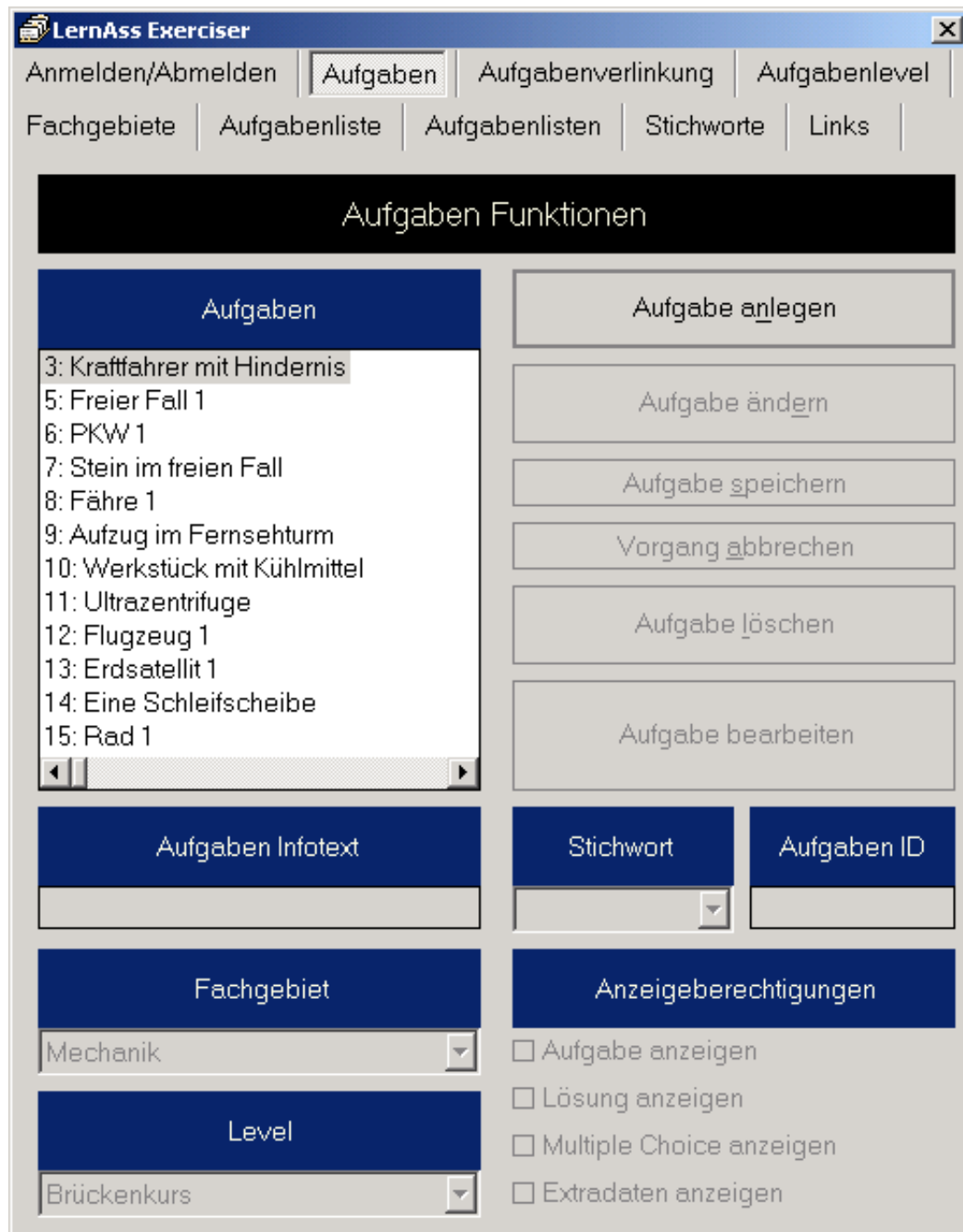


Abbildung 6.17: LernAss Exerciser Aufgaben

In der Aufgabenübersicht kann man die Aufgaben bearbeiten und neue Aufgaben hinzufügen. Wichtig ist vor allem der Knopf „Aufgabe bearbeiten“, mit dem man in das Fenster zur Aufgabenbearbeitung kommt.

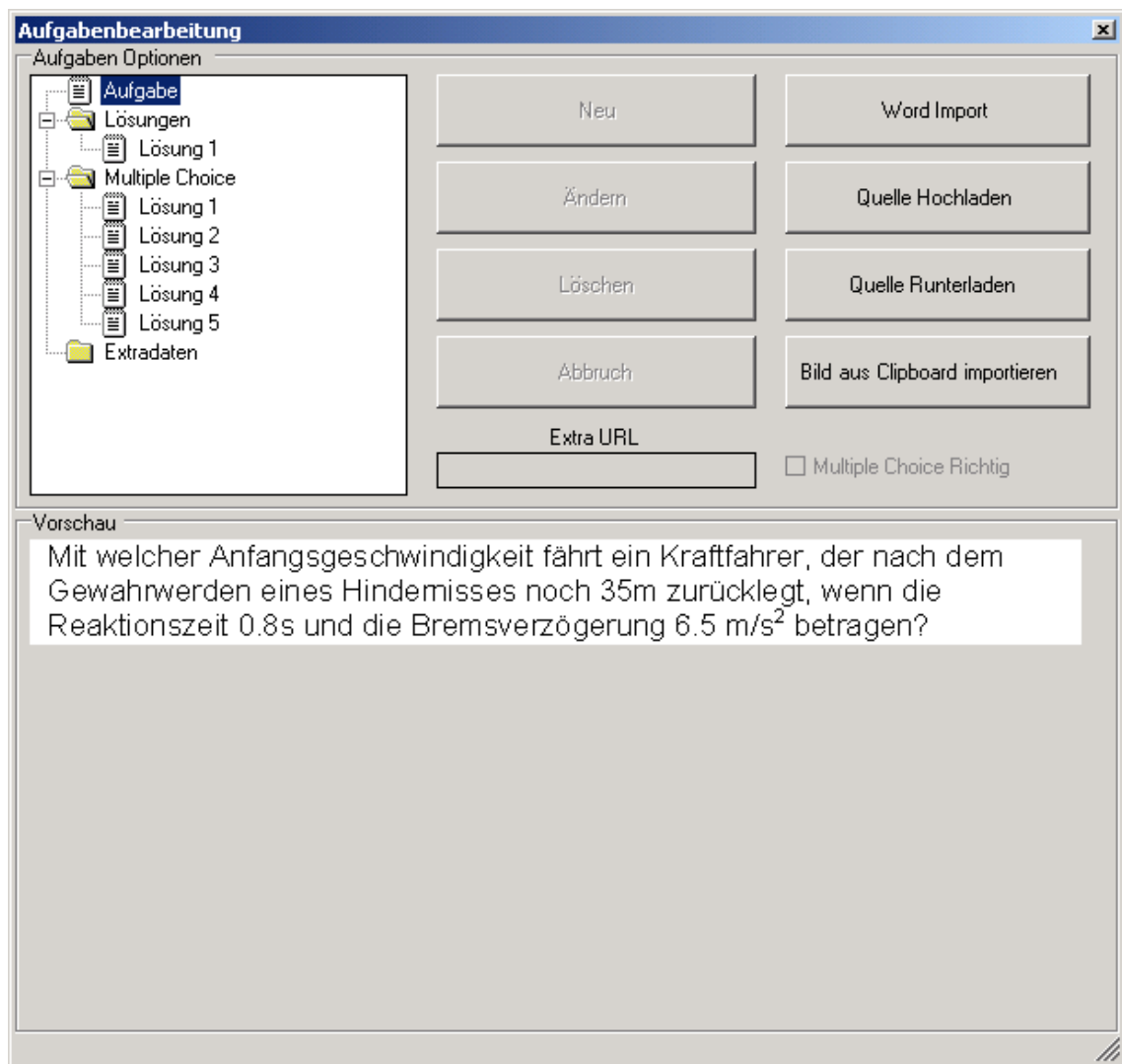


Abbildung 6.18: LernAss Exerciser Aufgabenbearbeitung

In der „Aufgabenbearbeitung“ kann man die Aufgabenquelle hochladen und das Bild zur Aufgabe importieren. Dies geht auch automatisiert mit dem „Word Import“. Außerdem kann man hier Lösungen, Multiple Choice-Lösungen und Extra-URLs definieren. Man bekommt eine Vorschau zu der ausgewählten Aufgabe, Lösung oder Multiple Choice-Lösung.

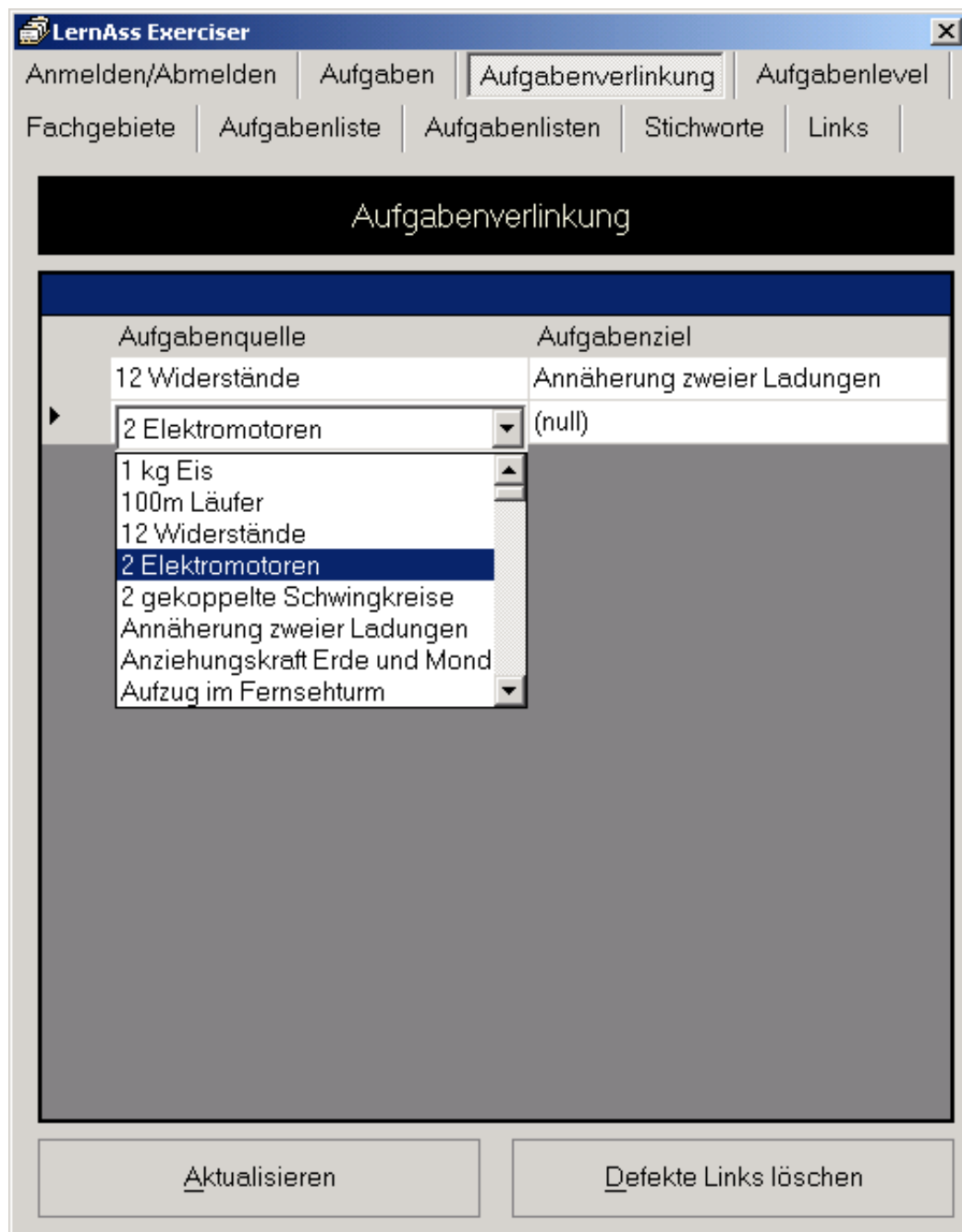


Abbildung 6.19: LernAss Exerciser Aufgabenverlinkung

Die Aufgaben in einer Aufgabendatenbank können über die „Aufgabenverlinkung“ miteinander verknüpft werden. Verwaiste Links können mit dem Button „Defekte Links löschen“ gelöscht werden.



Abbildung 6.20: LernAss Exerciser Aufgabenlevel

Für die Bearbeitung und Erstellung von Aufgaben sind bestimmte Voraussetzungen zu erfüllen. Zu den Voraussetzungen gehören die Level. In dem „Level“ Tab kann man diese anlegen, bearbeiten und löschen.

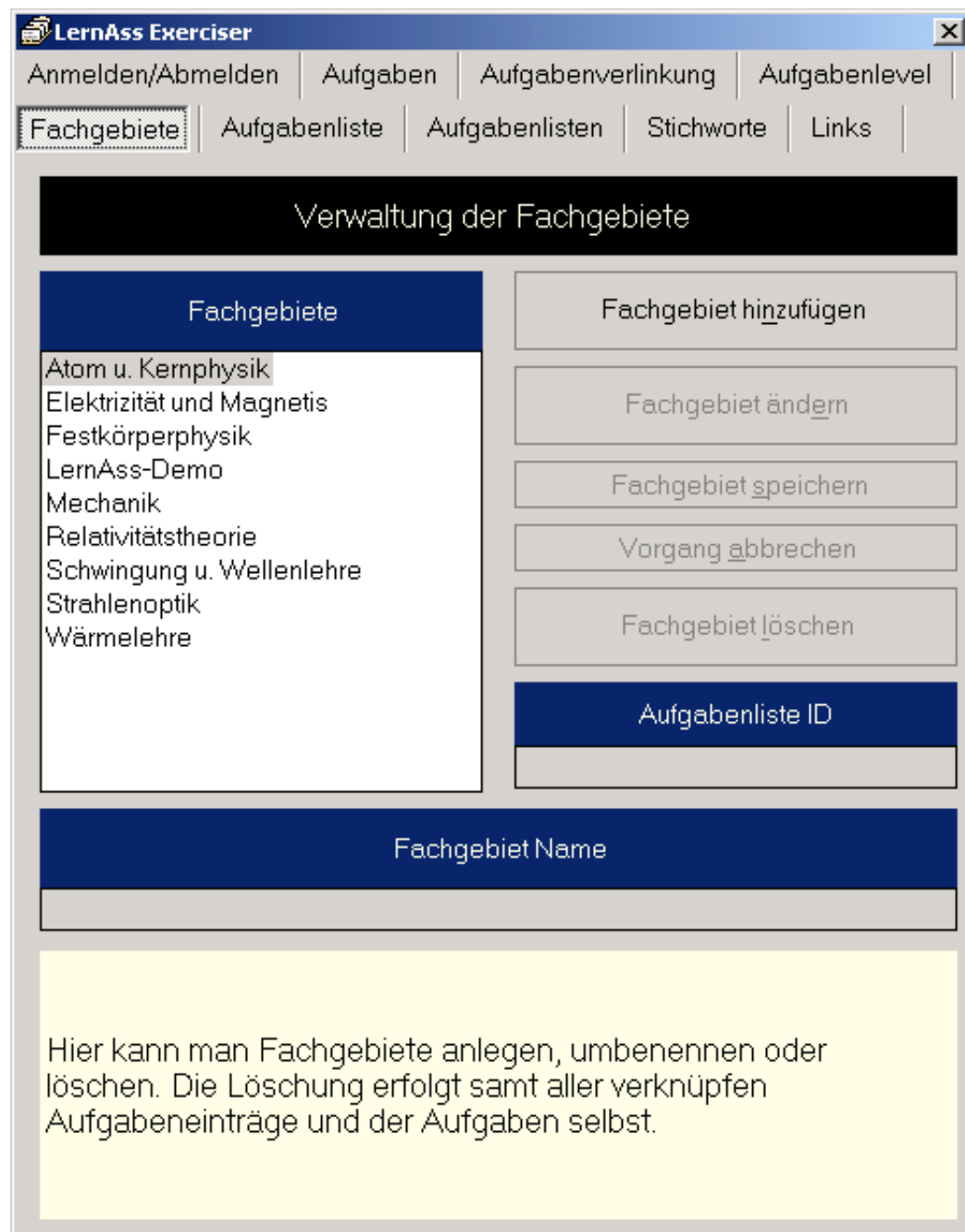


Abbildung 6.21: LernAss Exerciser Fachgebiete

Weitere Voraussetzung sind Fachgebiete. In dem „Fachgebiete“ Tab kann man diese anlegen, bearbeiten und löschen.

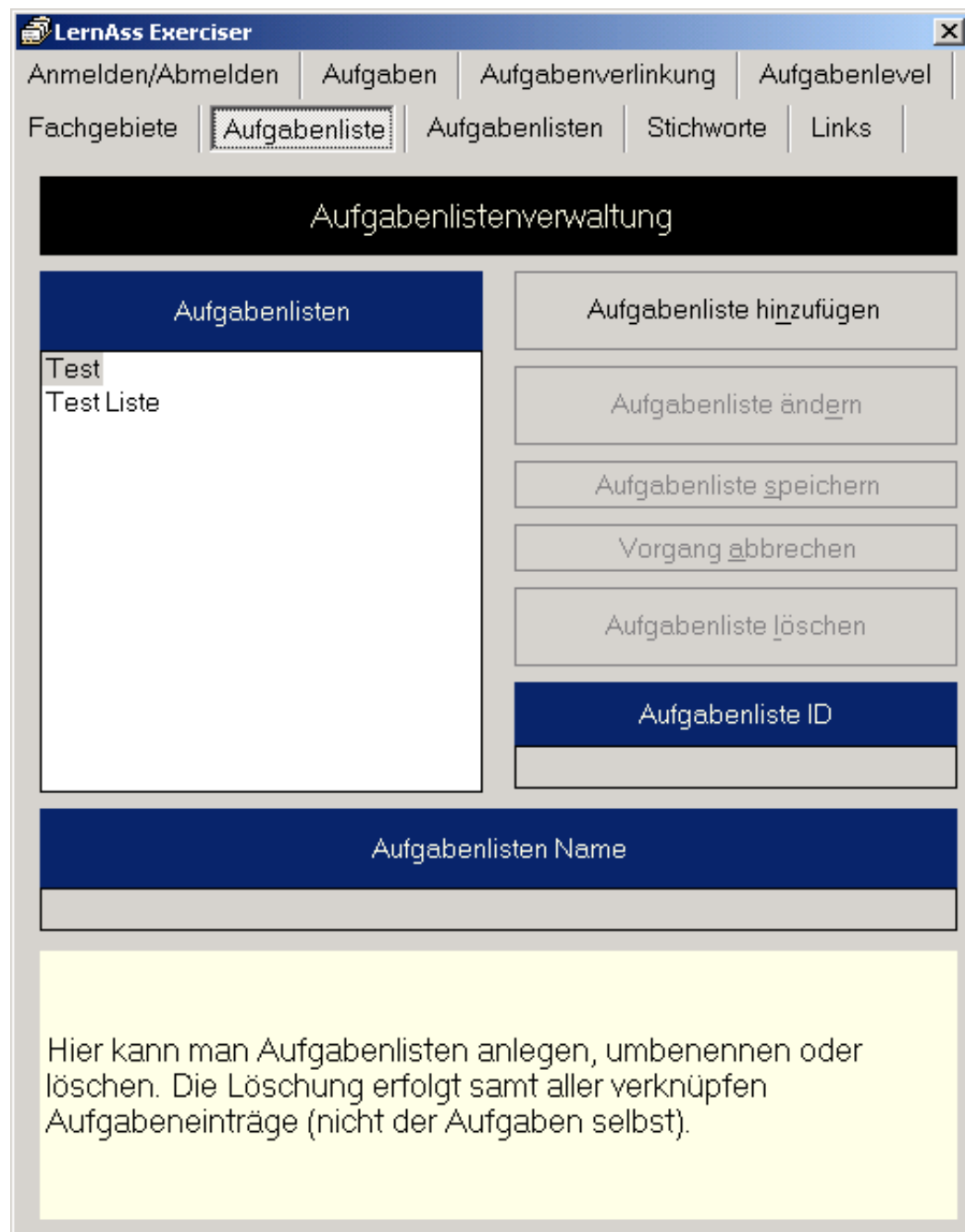


Abbildung 6.22: LernAss Exerciser Aufgabenliste

Aufgaben in LernAss können in verschiedenen Aufgabenlisten unterteilt werden. Die Aufgabenlisten werden hier verwaltet.

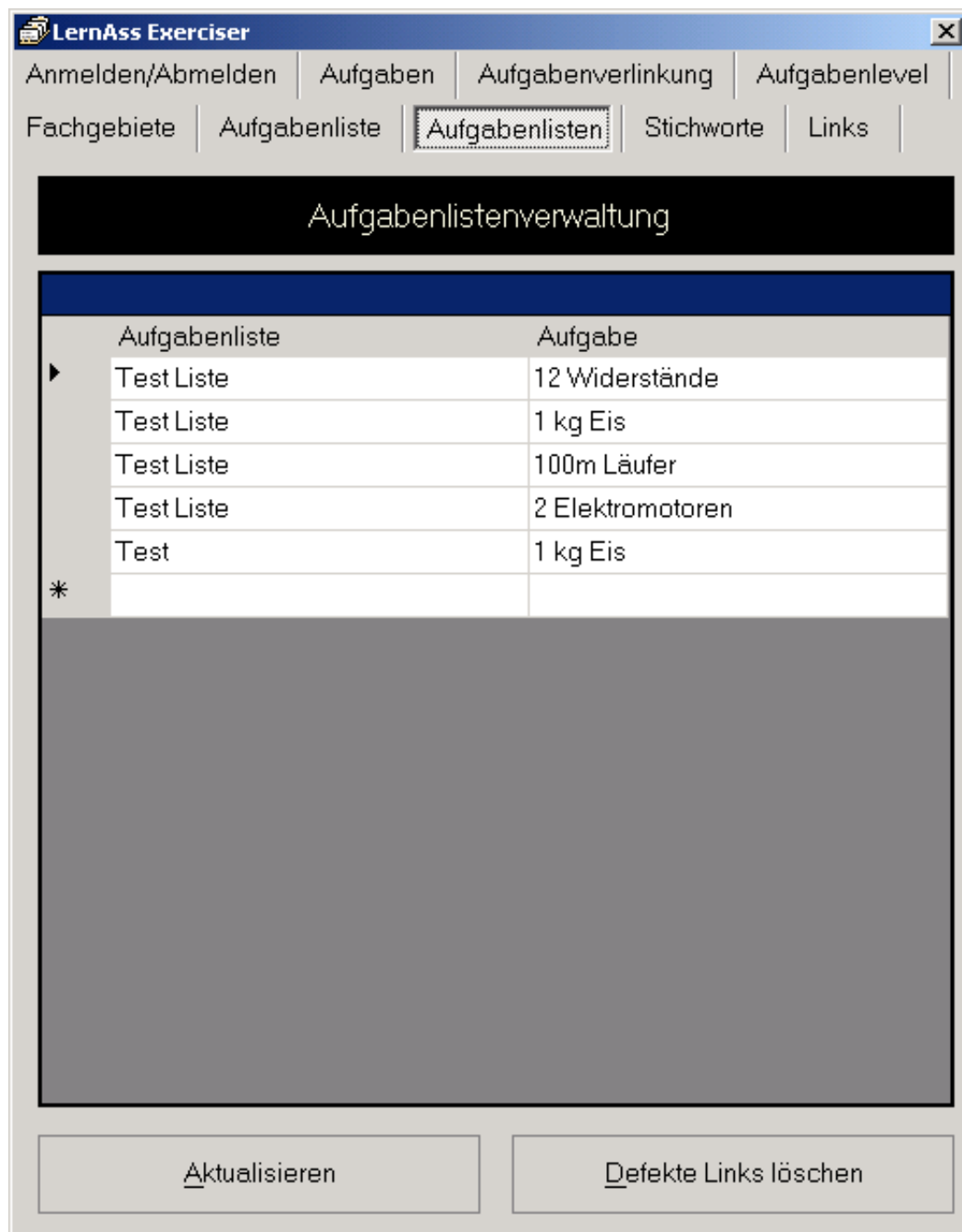


Abbildung 6.23: LernAss Exerciser Aufgabenlisten

Die Zuordnung von Aufgaben zu den Aufgabenlisten erfolgt auf der Tabseite „Aufgabenlisten“.

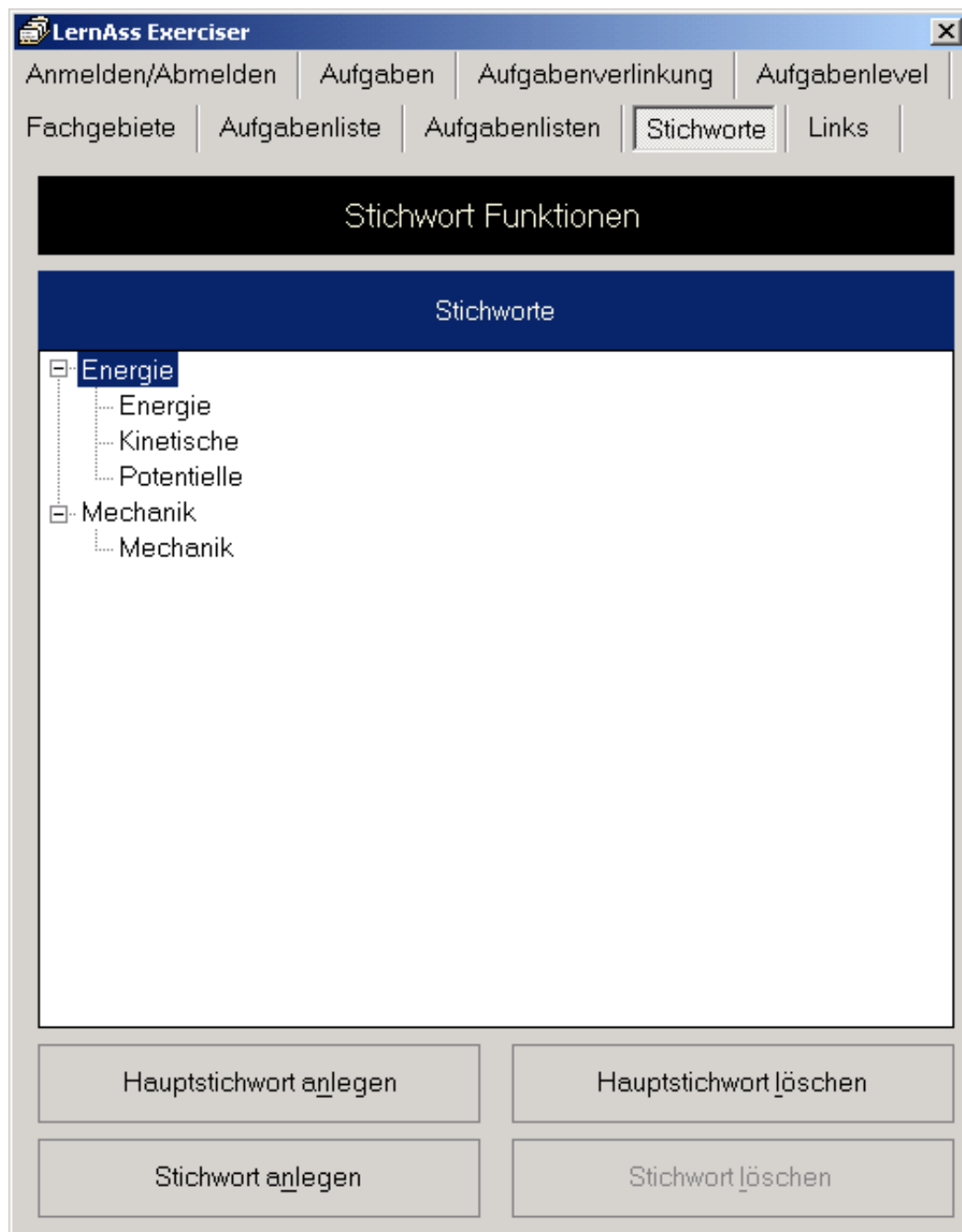


Abbildung 6.24: LernAss Exerciser Stichworte

Um die Suche nach Aufgaben zu vereinfachen, kann man Stichworte definieren. Die Stichworte werden auf der Tabseite „Stichworte“ verwaltet.



Abbildung 6.25: LernAss Exerciser Links

Die Aufgabendatenbank kann wie die LernAss-Datenbank Links aufnehmen und auf der Webseite anzeigen. Diese Links werden auf der Tabseite „Links“ verwaltet.

Um die Links immer im Blick zu haben, werden diese im Linkframe der Webansicht angezeigt, solange man sich in der Aufgabendatenbank befindet.

6.6 LernAss Komponenten

6.6.1 ADODB_LIB

Diese Komponente stellt die Funktionen für die alten LernAss-Datenbanken zur Verfügung. Dazu gehören das Auslesen von Daten, eine SQL Funktion und das Öffnen und Schließen der Datenbankverbindung.

Tabelle 6.3: Funktionsliste ADODB_LIB

Name	Funktion
open_DB	Öffnen der Datenbankverbindung und Rückgabe der ADODB.Connection-Struktur
close_DB	Schließen der Datenbankverbindung und Rückgabe von Fehlern
get_RS	Holen eines Recordsets aus der Datenbank und Rückgabe der ADODB_RS-Struktur
do_SQL	Direktes Ausführen von SQL-Befehlen und Rückgabe von Fehlern

Die ADODB.Connection Struktur enthält die Verbindung und gegebenenfalls eine Fehlerbeschreibung

Die ADODB_RS Struktur enthält das ADODB.Recordset und gegebenenfalls eine Fehlerbeschreibung

6.6.2 LA_EMF_Clip

Diese Komponente erledigt die Konvertierung der Metadaten aus dem Clipboard in einen Stream zur Speicherung in der Datenbank.

Tabelle 6.4: Funktionsliste LA_EMF_Clip

Name	Funktion
get_clip_emf	Öffnen der Zwischenablage und Auslesen der Metafiledaten, Rückgabe des Metafiles

6.6.3 MySQL_DB

Diese Komponente vereinfacht die Bedienung der Datenbank für LernAss Admin, LernAss Exerciser und LernAss Konverter-Programme. Sie stellt die Schnittstelle

zwischen dem MySQL-Dataprovider und den LernAss-Programmen dar. Die wichtigsten Funktionen werden hier abgebildet. Dazu gehören eine SQL-Ausführungsfunktion, eine Datenbank-Erstellungsfunktion (für die Aufgabendatenbanken) und das Öffnen und Schließen der Datenbankverbindung.

Die Funktionen sind durch Regions in Gruppen unterteilt. Der Code ist dadurch leichter zu lesen und eine Funktion schneller aufzufinden.

Tabelle 6.5: MySQL_DB Datenbankverbindungsbehandlung

Name	Funktion
open_DB	Öffnen der Datenbank anhand eines Verbindungsstrings und Rückgabe von Fehlermeldungen. Ohne Fehlermeldung ist die Verbindung geöffnet worden.
web_open_db	Öffnen der Datenbank anhand von Benuternamen, Passwort, Lokation und Datenbankname, Rückgabe von Fehlern
close_db	Schließen der Datenbankverbindung und Rückgabe von Fehlern

Tabelle 6.6: MySQL_DB WEB-Funktionen für die Datenbanken

Name	Funktion
get_DB_Data	Auslesen der Datenbankzugriffsdaten für eine Aufgabendatenbank aus der LernAss-Zentraldatenbank, Rückgabe als Dataset
get_DBs	Auslesen der Aufgabendatenbanken aus der LernAss-Zentraldatenbank, Rückgabe über Callback

Tabelle 6.7: MySQL_DB WEB-Link-Funktionen

Name	Funktion
get_Links	Auslesen der Links aus der aktuellen Datenbank, Rückgabe über Callback
get_Quick_Links	Auslesen der Links der aktuellen Datenbank, Rückgabe über Callback

Links können sowohl in der LernAss-Zentraldatenbank als auch in den Aufgabendatenbanken enthalten sein, und haben die selbe Tabellenstruktur

Tabelle 6.8: MySQL_DB WEB-Aufgabenfunktionen

Name	Funktion
get_Exercises	Auflisten von Aufgaben anhand der ausgewählten Aufgabenliste, Rückgabe über Callback
get_Exercise	Auslesen einer Aufgabe und den dazu gehörigen Daten, Rückgabe über Callback
get_Exercise_by_id	Auslesen einer Aufgabe mit Lösungen, Rückgabe eines Dataset über Callback
get_Exercise_List	Auslesen von Vergleichsdaten für den Ausdruck von Aufgaben, Rückgabe über Callback
get_Multiplechoice	Auslesen der Multiple Choice-Aufgaben anhand der ausgewählten Aufgabenliste, Rückgabe über Callback
get_Multiplechoice_List	Auslesen von Vergleichsdaten für den Multiple Choice-Test, Rückgabe über Callback
get_Multiplechoice_by_id	Auslesen einer Multiple Choice-Aufgabe mit Lösungen, Rückgabe eines Dataset über Callback
get_Picture	Auslesen der Binärdaten eines Bildes

Tabelle 6.9: MySQL_DB WEB-Aufgabensuchfunktionen

Name	Funktion
search_Exercise_Info	Suche nach Aufgaben anhand der Info, Rückgabe über Callback
search_Exercise_Topic	Suche nach Aufgaben anhand des Fachgebietes, Rückgabe über Callback
search_Exercise_Level	Suche nach Aufgaben anhand des Levels, Rückgabe über Callback
search_Exercise_Hint	Suche nach Aufgaben anhand eines Stichwortes, Rückgabe über Callback
search_Exercise_InfoHint	Suche nach Aufgaben anhand der Info und eines Stichwortes, Rückgabe über Callback

Tabelle 6.10: MySQL_DB LernAss Admin-Datenbankfunktionen

Name	Funktion
get_DBs	Auflisten der vorhandenen Aufgabendatenbanken, Rückgabe über Dataset
exists_database	Prüfung auf Existenz einer Datenbank, Rückgabe von Wahr oder Falsch
insert_db	Eine Datenbank in die LernAss-Zentraldatenbank einpflegen, Rückgabe von Fehlern
update_db	Zugangsdaten einer Datenbank ändern, Rückgabe von Fehlern
drop_db	Löschen einer Datenbank, Rückgabe von Fehlern
delete_db	Löschen einer Datenbank aus der LernAss-Zentraldatenbank, Rückgabe von Fehlern
activate_db	Aktivieren einer Aufgabendatenbank, Rückgabe von Fehlern
deactivate_db	Deaktivieren einer Aufgabendatenbank, Rückgabe von Fehlern
insert_metadata	Metadaten (Professor, Fach, aktuelle Aufgabenliste) in die Aufgabendatenbank eingeben, Rückgabe von Fehlern

Tabelle 6.11: MySQL_DB LernAss Admin / LernAss Exerciser-Linkfunktionen

Name	Funktion
insert_link	Einen Link anlegen, Rückgabe von Fehlern
update_link	Einen Link aktualisieren, Rückgabe von Fehlern
delete_link	Einen Link löschen, Rückgabe von Fehlern
get_Links_DS	Auslesen der Links, Rückgabe eines Dataset

Tabelle 6.12: MySQL_DB LernAss Exerciser-Extrafunktionen

Name	Funktion
insert_extra	Ein Extra anlegen, Rückgabe von Fehlern
update_extra	Ein Extra aktualisieren, Rückgabe des Erfolges
delete_extra	Ein Extra löschen, Rückgabe des Erfolges
get_Extra_DS	Auslesen der Extras, Rückgabe eines Dataset

Tabelle 6.13: MySQL_DB LernAss Exerciser-Aufgabenfunktionen

Name	Funktion
insert_exercise	Eine Aufgabe in eine LernAss-Aufgabendatenbank einpflegen, Rückgabe von Fehlern
update_exercise	Eine Aufgabe in einer LernAss-Aufgabendatenbank aktualisieren, Rückgabe des Erfolges
delete_exercise	Eine Aufgabe in einer LernAss-Aufgabendatenbank löschen, Rückgabe des Erfolges
upload_exercise_source	Eine Quelle zu einer Aufgabe in eine LernAss-Aufgabendatenbank einpflegen, Rückgabe der ID der Quelle
update_exercise_view	Ein Bild der Aufgabe zuordnen, Rückgabe des Erfolges
get_Exercise_DS	Auslesen der Aufgaben, Rückgabe eines Dataset

Tabelle 6.14: MySQL_DB LernAss Exerciser-Aufgabenverlinkung

Name	Funktion
get_Exercise_Link_DS	Auslesen der Aufgabenverlinkung, Rückgabe eines Dataset
set_Exercise_Link_DS	Aktualisierung der Aufgabenverlinkung über ein Dataset, Rückgabe des Erfolges
clear_defect_links	Löschen von verwaisten Einträgen, Rückgabe des Erfolges

Tabelle 6.15: MySQL_DB LernAss Exerciser Aufgabenlisten

Name	Funktion
get_Exercise_List_DS	Auslesen der Aufgabenlisten-Rückgabe eines Dataset
set_Exercise_List_DS	Aktualisierung der Aufgabenlisten über ein Dataset, Rückgabe des Erfolges
clear_defect_lists	Löschen von verwaisten Einträgen, Rückgabe des Erfolges
insert_schema	Ein Schema in einer LernAss-Aufgabendatenbank anlegen, Rückgabe des Erfolges
update_schema	Ein Schema in einer LernAss-Aufgabendatenbank aktualisieren, Rückgabe des Erfolges
delete_schema	Ein Schema in einer LernAss-Aufgabendatenbank löschen, Rückgabe des Erfolges
insert_schemaexercise	Eine Aufgabe zu einer Aufgabenliste hinzufügen, Rückgabe des Erfolges
delete_schemaexercise	Eine Aufgabe aus einer Aufgabenliste löschen, Rückgabe des Erfolges
get_actual_list	Auslesen der aktuellen Aufgabenliste, Rückgabe der Listen-ID
set_actual_list	Setzen der aktuellen Aufgabenliste, Rückgabe des Erfolges
get_Lists	Auslesen der Aufgabenlisten, Rückgabe über Callback
get_Lists_DS	Auslesen der Aufgabenlisten, Rückgabe as Dataset

Tabelle 6.16: MySQL_DB LernAss Exerciser-Lösungen

Name	Funktion
upload_solution_source	Eine Lösung zu einer Aufgabe in einer LernAss-Aufgabendatenbank hinzufügen, Rückgabe der ID der Quelle
update_solution_source	Eine Lösung zu einer Aufgabe in einer LernAss-Aufgabendatenbank aktualisieren, Rückgabe des Erfolges
delete_solution	Löschen einer Lösung, Rückgabe des Erfolges
update_solution_view	Ein Bild der Lösung zuordnen, Rückgabe des Erfolges
get_Next_Solution_Nr	Die größte Nummer einer Lösung auslesen, Rückgabe der größten Lösungsnummer
get_Solution_DS	Auslesen der Lösungen, Rückgabe eines Dataset

Tabelle 6.17: MySQL_DB LernAss Exerciser Multiple Choice

Name	Funktion
upload_multiplechoice_source	Eine Multiple Choice-Lösung zu einer Aufgabe in eine LernAss-Aufgabendatenbank einpflegen, Rückgabe von Fehlern
update_multiplechoice_source_correct	Eine Multiple Choice-Lösung als richtig oder falsch markieren, Rückgabe des Erfolges
delete_multiplechoice	Löschen einer Multiple Choice-Lösung, Rückgabe des Erfolges
update_multiplechoice_view	Ein Bild zur Multiple Choice-Lösung zuordnen, Rückgabe des Erfolges
get_Multiple_Choice_DS	Auslesen der Multiple Choice-Lösungen, Rückgabe eines Dataset

Tabelle 6.18: MySQL_DB LernAss Exerciser Source Up-/Download

Name	Funktion
update_source	Eine Quelle in der Datenbank aktualisieren, Rückgabe des Erfolges
download_source	Eine Quelle aus einer Datenbank auslesen, Rückgabe des Dataset
get_Source_DS	Auslesen der Metadaten einer Quelle, Rückgabe des Dataset

Tabelle 6.19: MySQL_DB LernAss Exerciser-Bildfunktionen

Name	Funktion
upload_picture	Ein Bild in eine LernAss-Aufgabendatenbank einpflegen, Rückgabe von Fehlern
update_picture	Ein Bild in einer LernAss-Aufgabendatenbank aktualisieren, Rückgabe des Erfolges
get_Picture_Byte	Ein Bild aus der Datenbank auslesen, Rückgabe als Byte Array

Tabelle 6.20: MySQL_DB LernAss Exerciser-Hintfunktionen

Name	Funktion
insert_hint	Ein Stichwort zur Datenbank hinzufügen, Rückgabe der ID der Quelle
delete_hint	Ein Stichwort löschen, Rückgabe des Erfolges
get_Hints_DS	Auslesen der Stichworte, Rückgabe als Dataset
get_Key_Hint_DS	Auslesen der Hauptstichworte (DISTINCT), Rückgabe als Dataset

Tabelle 6.21: MySQL_DB LernAss Exerciser-Topicfunktionen

Name	Funktion
insert_topic	Ein Topic in eine LernAss-Aufgabendatenbank einpflegen, Rückgabe von Fehlern
update_topic	Ein Topic in einer LernAss-Aufgabendatenbank aktualisieren, Rückgabe des Erfolges
delete_topic	Ein Topic in einer LernAss-Aufgabendatenbank löschen, Rückgabe des Erfolges
get_Topics_DS	Auslesen der Fachgebiete, Rückgabe als Dataset

Tabelle 6.22: MySQL_DB LernAss Exerciser Levelfunktionen

Name	Funktion
insert_level	Ein Level in eine LernAss-Aufgabendatenbank einpflegen, Rückgabe von Fehlern
update_level	Ein Level in einer LernAss-Aufgabendatenbank aktualisieren, Rückgabe des Erfolges
delete_level	Ein Level in einer LernAss-Aufgabendatenbank löschen, Rückgabe des Erfolges
get_Level_DS	Auslesen der Level, Rückgabe als Dataset

Tabelle 6.23: MySQL_DB Generelle Ausleseoperationen

Name	Funktion
get_Exercise_by_id	Auslesen der Aufgaben-ID anhand der Benutzer definierten ID
get_topic	Auslesen der ID eines Fachgebietes, Rückgabe der ID
get_level	Auslesen der ID eines Levels, Rückgabe der ID

Tabelle 6.24: MySQL_DB Datenbankrechteverwaltung

Name	Funktion
revoke	Benutzerrechte für eine LernAss-Aufgabendatenbank widerrufen, Rückgabe von Fehlern
grant	Benutzerrechte für eine LernAss-Aufgabendatenbank anlegen, Rückgabe von Fehlern

Tabelle 6.25: MySQL_DB Basisdatenbankbearbeitungsoperationen

Name	Funktion
get_DS	Ein Dataset aus der Datenbank auslesen, Rückgabe des Dataset
get_DR	Einen Datareader auf die Datenbank anlegen, Rückgabe des Datareader
do_SQL	Einen SQL-Befehl ausführen, Rückgabe von Fehlern
create_exercise_db	Anlegen der Tabellen einer LernAss-Aufgabendatenbank, Rückgabe von Fehlern
count_data	Zählen von Datensätzen mit einem Kriterium, Rückgabe der Anzahl
count_2_data	Zählen von Datensätzen mit zwei Kriterien, Rückgabe der Anzahl
optimze_tables	Optimieren der Tabellen in einer Aufgabendatenbank, Rückgabe des Erfolges
optimze_Lernsss	Optimieren der Tabellen in der LernAss-Datenbank, Rückgabe des Erfolges

6.6.4 MySQLClient

Der MySQL-Dataprovider von Christoph Wienands, geschrieben in C#, ist die in LernAss verwendete API für den Zugriff auf die MySQL Datenbank (siehe auch Kapitel [1.9.2](#) auf Seite [27](#)).

Die Komplexität und Funktionsvielfalt von Datenbankschnittstellen, hier insbesondere des MySQL_Client, erlaubt mir nicht, eine genaue Beschreibung der Funktionen abzugeben. Es kommen unter Umständen neue Funktionen hinzu oder die Funktionsweise ändert sich.

7 Installation des Servers

Inhaltsangabe

7.1	Vorwort	108
7.2	Windows 2000 Server	109
7.3	.NET Framework	109
7.4	MySQL	113
7.4.1	Installation	113
7.4.2	PHP	116
7.4.3	phpMyAdmin	121
7.4.4	Die Rechtevergabe mittels GRANT	122
7.4.5	Der SQL Code für die LernAss-Zentral-Datenbank	123
7.4.6	Der SQL Code für die LernAss-Aufgaben-Datenbank	124
7.4.7	Entfernen der GRANTs durch REVOKE	127
7.4.8	Einrichtung der Webapplikation	128

Abbildungen

7.1	.NET-Framework-Installation beginnen?	109
7.2	Durchführung der .NET-Framework-Installation	109
7.3	Windows Installer Update	110
7.4	Installation des .NET-Frameworks	110
7.5	Installation und Kompilierung des .NET-Framework	110
7.6	Ende der .NET-Framework Installation	111
7.7	Initialisierung des .NET-Framework-Updates	111
7.8	.NET-Framework-Update durchführen?	111
7.9	EULA des .NET-Framework-Updates akzeptieren	112
7.10	Beginn des .NET-Framework-Updates	112

7.11 Ende des .NET-Framework-Updates	112
7.12 Nach dem Update Rechner neu starten	112
7.13 MySQL-Installation beginnen?	113
7.14 Hinweise zur MySQL-Installation	113
7.15 Auswahl des Verzeichnisses	114
7.16 Auswahl der MySQL-Installationsvariante	114
7.17 Beginn der MySQL Installation	115
7.18 Ende der MySQL-Installation	115
7.19 MySQL „myini.ini“ in WinMySQLAdmin	116
7.20 Beginn der PHP-Installation	117
7.21 Rechtliche Hinweise zur PHP-Installation	117
7.22 Auswahl des Installationsmodus der PHP-Installation	118
7.23 Auswahl des Verzeichnisses der PHP-Installation	118
7.24 Konfiguration der PHP-E-Mail-Anbindung	119
7.25 Auswahl des Webservers für PHP	119
7.26 Beginn der PHP-Installation	120
7.27 Installationsvorgang	120
7.28 Auswahl des Webservers der PHP-Installation	121
7.29 Ende der PHP-Installation	121
7.30 Applikationseinstellung bei den Internetinformationsdiensten	128
7.31 WEB-Applikation erstellen	129

Tabellen

7.1 Server-Software – Voraussetzungen	109
---	-----

7.1 Vorwort

LernAss setzt im Prinzip vier Softwarepakete auf dem Server voraus. Das sind:

- LernAss Web (Die Lernanwendung)
- MySQL (die Datenbank)
- PHP und [phpMyAdmin](#) (für einfache MySQL-Administration)

- .NET Framework (für LernAss WEB wichtig – die Runtime-Bibliotheken)

Tabelle 7.1: Server-Software – Voraussetzungen

Windows 2000 (Server)	>= Servicepack 2
.NET-Framework	>= Servicepack 2
MySQL	>3.2.54
PHP	Version >4

7.2 Windows 2000 Server

Die Installation des Servers erfolgt mit den Standardkomponenten. Wichtig ist jedoch der komplette IIS mit allen Optionen!

Nach der Installation des Servers muss das .NET-Framework installiert werden. Dann muss die Installation des MySQL-Servers erfolgen (siehe Kapitel 7.4 Seite 113). Nach der MySQL-Installation kann LernAss Web installiert werden.

7.3 .NET Framework

Die Installation des .NET-Framework erfolgt in der Regel einfach und problemlos.



Abbildung 7.1: .NET-Framework-Installation beginnen?

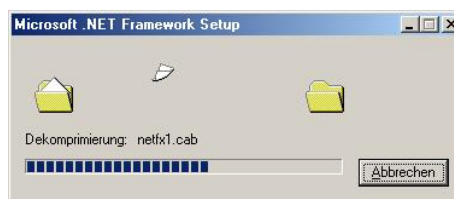


Abbildung 7.2: Durchführung der .NET-Framework-Installation

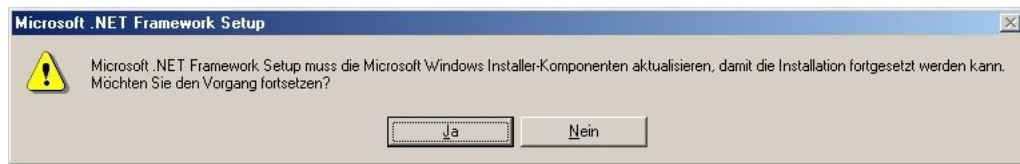


Abbildung 7.3: Windows Installer Update



Abbildung 7.4: Installation des .NET-Frameworks



Abbildung 7.5: Installation und Kompilierung des .NET-Framework



Abbildung 7.6: Ende der .NET-Framework Installation

Nach der Installation des .NET-Framework muss der Rechner neu gestartet werden. Es muss dann unbedingt noch das aktuellste Servicepack für VisualStudio .NET und das .NET-Framework aufgespielt werden. Zum Zeitpunkt der Diplomarbeit ist es das Servicepack 2 für das .NET-Framework.



Abbildung 7.7: Initialisierung des .NET-Framework-Updates



Abbildung 7.8: .NET-Framework-Update durchführen?

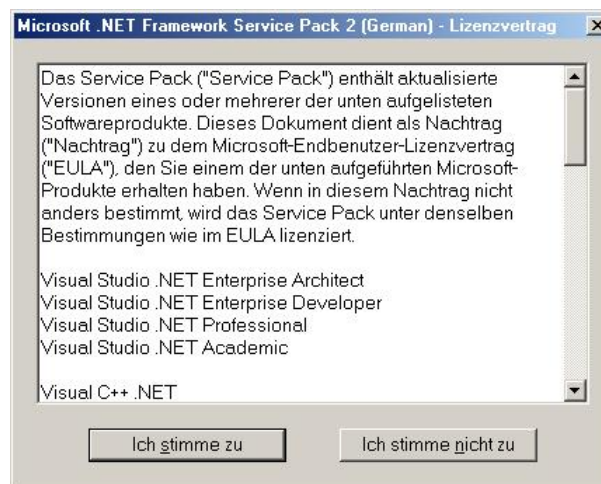


Abbildung 7.9: EULA des .NET-Framework-Updates akzeptieren



Abbildung 7.10: Beginn des .NET-Framework-Updates

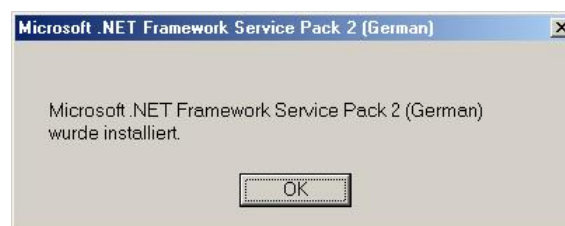


Abbildung 7.11: Ende des .NET-Framework-Updates



Abbildung 7.12: Nach dem Update Rechner neu starten

7.4 MySQL

7.4.1 Installation

Die Installation von MySQL ist einfach: Setup-Paket bei <http://www.mysql.com> (Abrufdatum 05.08.2003) herunterladen und installieren.

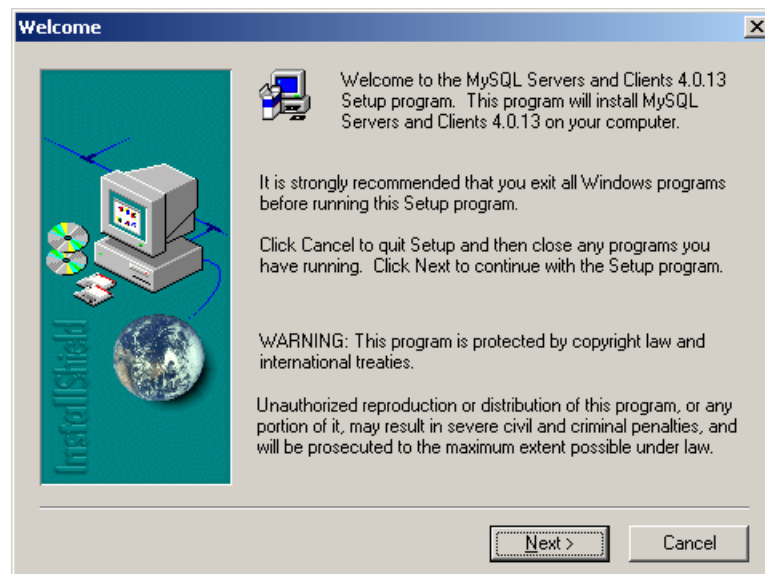


Abbildung 7.13: MySQL-Installation beginnen?

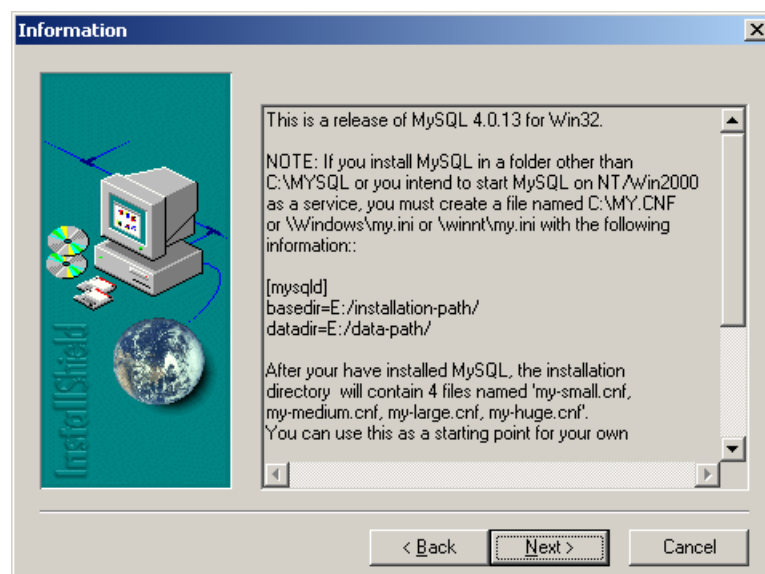


Abbildung 7.14: Hinweise zur MySQL-Installation

An dieser Stelle weist der Hersteller von MySQL auf wichtige Änderungen hin oder gibt Konfigurationsvorschläge. Man sollte sich diesen Text zumindest einmal durchlesen, damit man nachher nicht auf unerwartete Probleme mit dem installierten MySQL-Server stößt.

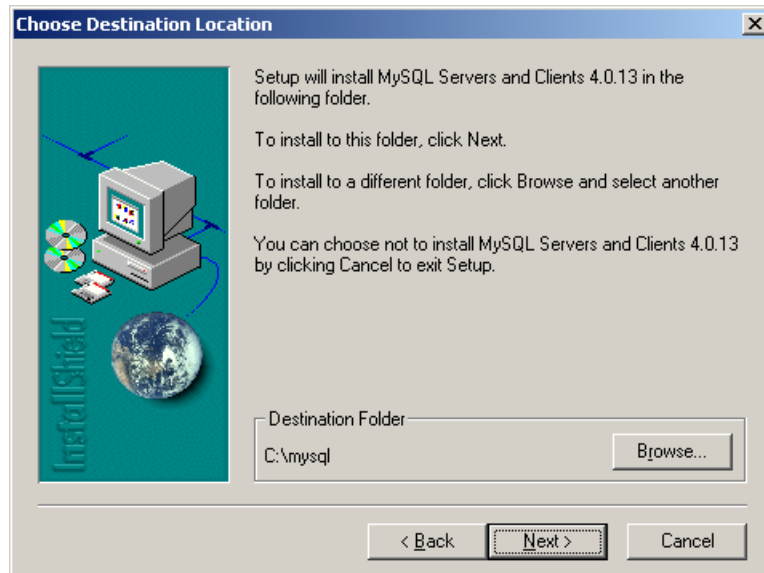


Abbildung 7.15: Auswahl des Verzeichnisses

MySQL wird standartmässig in „C:\mysql“ installiert. Es gibt keinen Grund, das zu ändern.

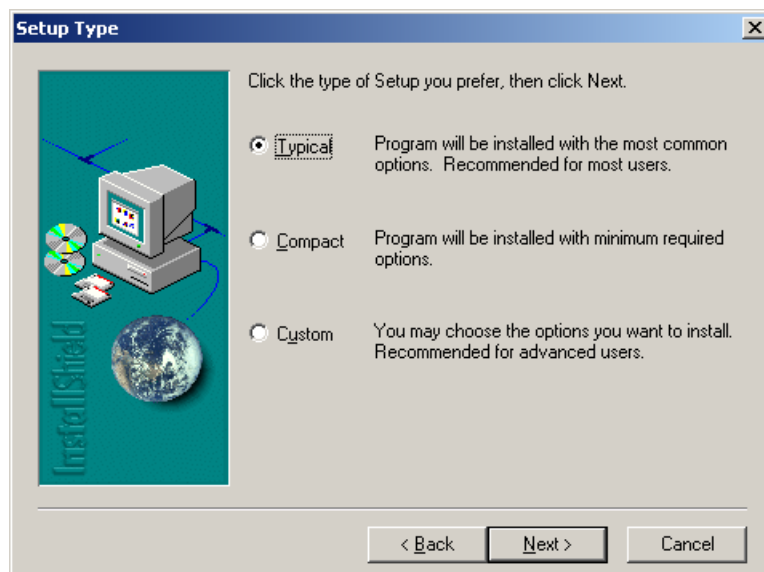


Abbildung 7.16: Auswahl der MySQL-Installationsvariante

Die „Typical“-Variante der MySQL-Installation ist am besten geeignet. Die Datenbank-Engine und die benötigten Dienstprogramme werden dadurch installiert.

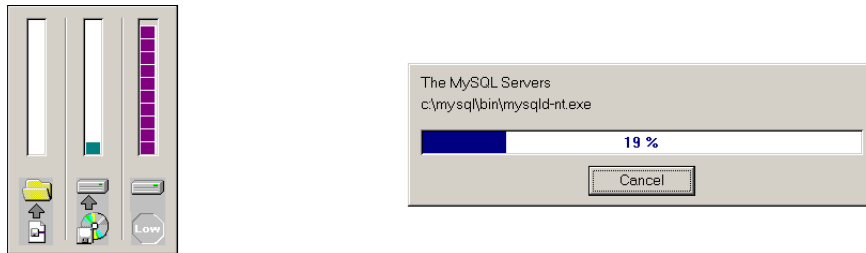


Abbildung 7.17: Beginn der MySQL Installation

Die Installation läuft zügig durch ...

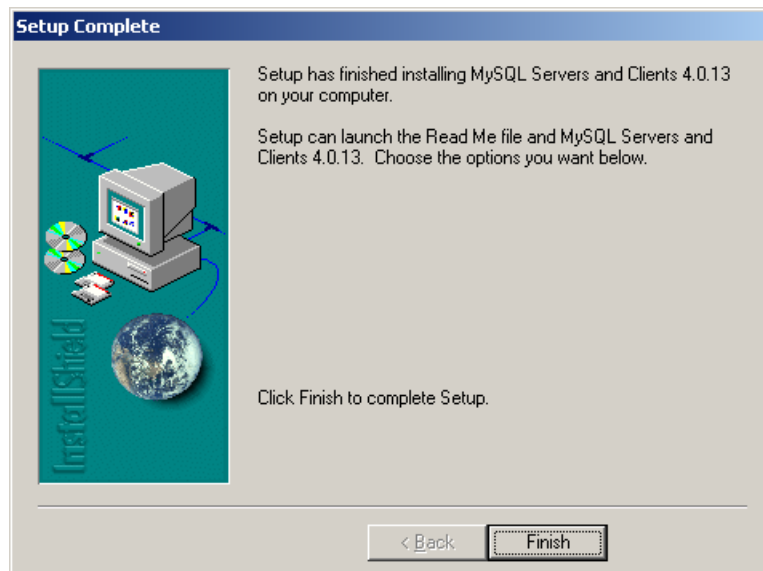


Abbildung 7.18: Ende der MySQL-Installation

Nach der Installation muss man den Puffer für Queries erhöhen. Dies geschieht in der Datei „mysql.ini“.

Die Datei mysql.ini sollte zusätzlich zu den Standardeinstellungen diese Einstellungen beinhalten:

```
[mysqld]
set-variable=key_buffer=16M
set-variable=max_allowed_packet=16M
set-variable=query_cache_limit=16M
```

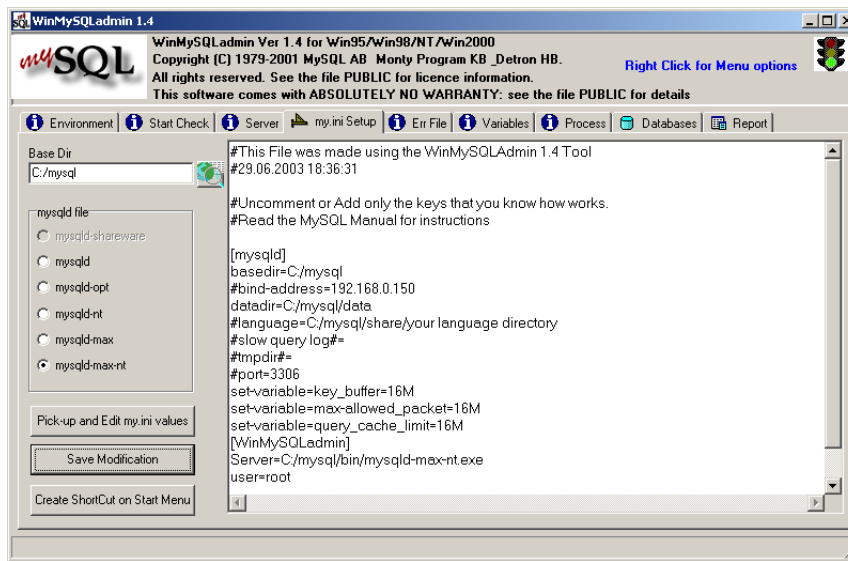


Abbildung 7.19: MySQL „myini.ini“ in WinMySQLAdmin

Nach der Installation des MySQL-Servers muss unbedingt die „libmysql.dll“ in das Windowsverzeichnis (%SYSTEMROOT%) kopiert werden! Beim ersten Start sollte der MySQL-Server nach Benutzernamen und Passwort fragen – dies ist für den Masterzugang zur Datenbank. Gut aufbewahren!!!

Ich verwende hier „root“ mit dem Passwort „LA2003pwd“. Das Passwort wird nur für die lokale Administration der Datenbank benötigt. Das heißt, LernAss ist nicht von diesem Passwort abhängig!

7.4.2 PHP

Um auf MySQL zugreifen zu können, müssen bestimmte Rechte vergeben werden (siehe Kapitel 7.4.4 auf Seite 122). phpMyAdmin und PHP sind dafür am besten geeignet. Diese Installation kann auf verschiedenen Wegen erfolgen. Der einfachste ist, die Dateien „php-#.##-Win32.zip“ und „php-#.##-installer.exe“ herunterzuladen, wobei # für die aktuelle Versionsnummer steht. Die Installationsschritte mit dem Installer sind wie folgt:



Abbildung 7.20: Beginn der PHP-Installation



Abbildung 7.21: Rechtliche Hinweise zur PHP-Installation

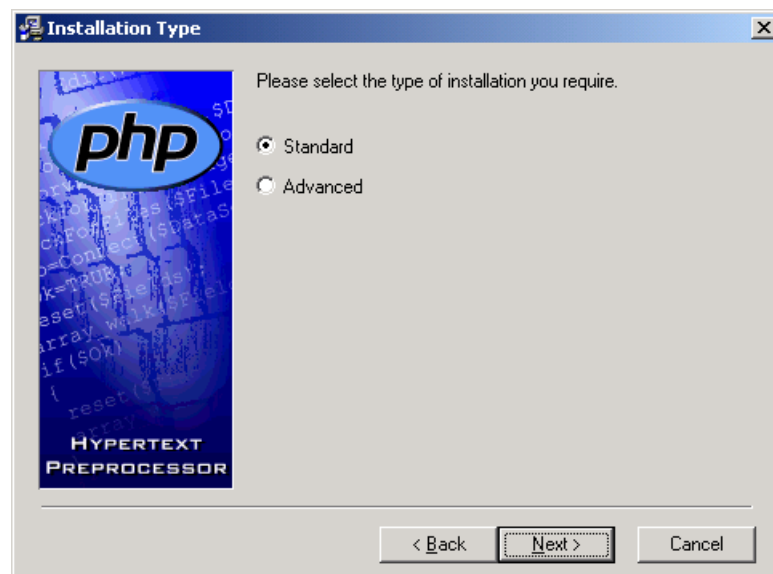


Abbildung 7.22: Auswahl des Installationsmodus der PHP-Installation

Hier belässt man die Auswahl auf „Standard“.

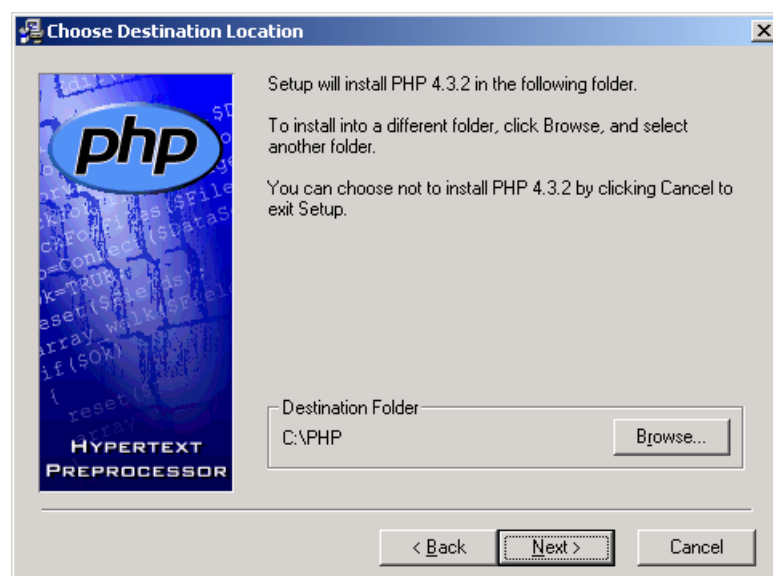


Abbildung 7.23: Auswahl des Verzeichnisses der PHP-Installation

Das Verzeichnis für PHP ist standardmäßig „C:\PHP“, dies belassen wir auch so.



Abbildung 7.24: Konfiguration der PHP-E-Mail-Anbindung

An dieser Stelle könnte man die Mail-Funktionalität an einen Mailserver mit entsprechender Mailadresse binden, die Grundeinstellung kann aber so belassen werden.

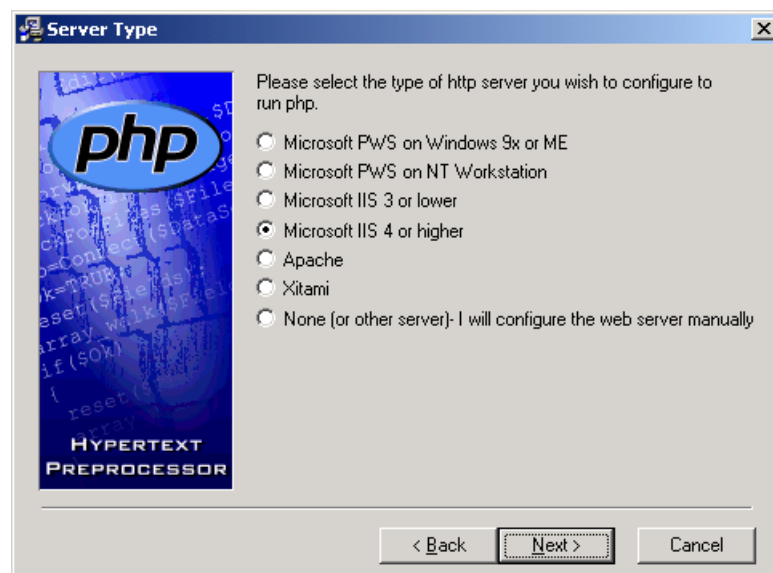


Abbildung 7.25: Auswahl des Webservers für PHP

Man belässt hier die automatische Auswahl auf IIS4.

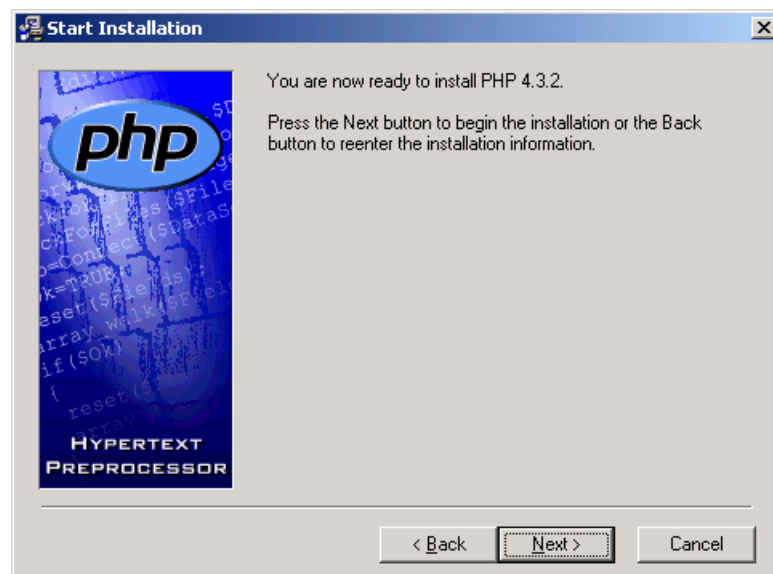


Abbildung 7.26: Beginn der PHP-Installation

Letzte Abfrage vor der Installation.

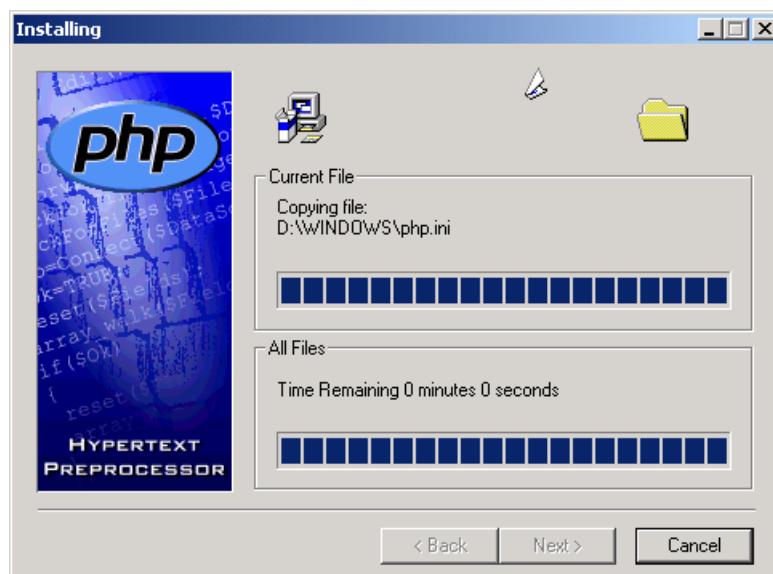


Abbildung 7.27: Installationsvorgang

Die Installation läuft durch.

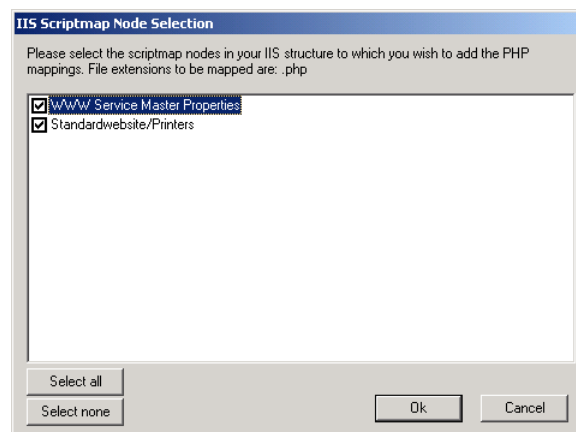


Abbildung 7.28: Auswahl des Webservers der PHP-Installation

Hier wählt man alle Einträge aus, wodurch alle Webverzeichnisse PHP-tauglich gemacht werden.

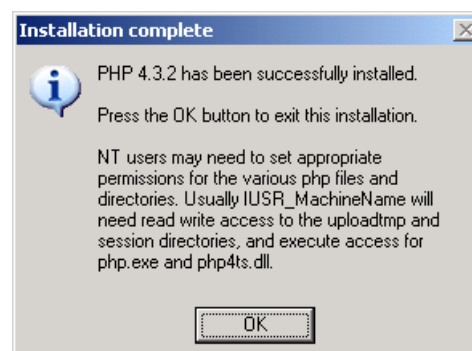


Abbildung 7.29: Ende der PHP-Installation

Die Installation ist hiermit abgeschlossen.

7.4.3 phpMyAdmin

PHPMyAdmin kann man bei <http://www.sourceforge.net/projects/phpmyadmin/> (Abrufdatum 05.08.2003) herunterladen. Das Paket entpackt man einfach in das Webserververzeichnis des IIS (C:\InetPub\wwwroot\).

Die Konfigurationsdatei von phpMyAdmin nennt sich „config.inc.php“. Man muss den Server hier korrekt eintragen.

```

$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin/';
$i = 0;
$i++;
$cfg['Servers'][$i]['host']           = 'localhost';
$cfg['Servers'][$i]['port']           = '';
$cfg['Servers'][$i]['socket']         = '';
$cfg['Servers'][$i]['connect_type']   = 'tcp';
$cfg['Servers'][$i]['controluser']    = '';
$cfg['Servers'][$i]['controlpass']    = '';
$cfg['Servers'][$i]['auth_type']      = 'config';
$cfg['Servers'][$i]['user']           = 'root';
$cfg['Servers'][$i]['password']       = 'LA2003pwd';
$cfg['Servers'][$i]['only_db']        = '';

```

„root“ und „LA2003pwd“ sind die Zugangsdaten zum MySQL-Server, die wir nach der Installation des MySQL-Servers angegeben haben.

7.4.4 Die Rechtevergabe mittels GRANT

In phpMyAdmin wählt man als zu bearbeitende Datenbank „mysql“ aus, und selektiert im neuen Frame „SQL“. In dem dann erscheinenden SQL-Eingabefenster müssen ein paar Arbeitsschritte erfolgen.

Löschen des Zugriffs auf alle „test*“ Datenbanken für jedermann, ausserdem Löschen der „test“ Datenbank:

```

DELETE from mysql.db where db = 'test%';
DROP DATABASE 'test';

```

Grant für LernAss zum Bearbeiten und Ansehen der Datenbanken, alle anderen User entfernen:

```

DELETE from mysql.user;
GRANT SELECT, INSERT, UPDATE, DELETE ON LernAss.* TO "LernAss"@
  localhost" IDENTIFIED BY "LernAss";
GRANT ALL PRIVILEGES ON *.* TO "root"@localhost" IDENTIFIED BY
  "LA2003pwd" WITH GRANT OPTION;

```

Der Benutzer LernAss kann Datenbanken erstellen, die Privilegien neu laden, GRANT und REVOKE ausführen und auf die Datenbank LernAss mit SELECT,INSERT,UPDATE und DELETE zugreifen.

Die neue erstellten Privilegien neu einlesen:

```
FLUSH PRIVILEGES;
```

Der komplette Grant Satz für LernAss – nur für einen neu installierten Server:

```
DELETE from mysql.db where db = 'test%';
DROP DATABASE 'test';
DELETE from mysql.user;
GRANT SELECT,INSERT,UPDATE,DELETE ON LernAss.* TO "LernAss"@
    localhost" IDENTIFIED BY "LernAss";
5 GRANT CREATE,RELOAD ON *.* TO "LernAss"@localhost" IDENTIFIED
    BY "LernAss" WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO "root"@localhost" IDENTIFIED BY
    "LA2003pwd" WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

7.4.5 Der SQL Code für die LernAss-Zentral-Datenbank

Nach dem Einstellen der Berechtigungen ist der letzte Schritt das Erstellen der LernAss Datenbank. Dazu verwendet man die SQL-Eingabe von phpMyAdmin und fügt diese Zeilen ein:

```
CREATE TABLE 'index' (
    'Key_Index' int(11) NOT NULL auto_increment ,
    'DB_activated' char(1) NOT NULL default '0',
    'DB_Info' text NOT NULL,
5 'DB_User' text NOT NULL,
    'DB_Password' text NOT NULL,
    'DB_Location' text NOT NULL,
    'DB_Name' text NOT NULL,
    PRIMARY KEY ('Key_Index')
10 ) TYPE=MyISAM PACK_KEYS=0 COMMENT='Zentrale_
    Datenbankverwaltungstabelle';
```

```
CREATE TABLE 'links' (
    'Key_Link' int(11) NOT NULL auto_increment ,
    'Link_Name' text NOT NULL,
15 'Link_Shortinfo' text NOT NULL,
    'Link_URL' text NOT NULL,
    'Link_Picture_URL' text NOT NULL,
```

```

PRIMARY KEY ( 'Key_Link ' )
) TYPE=MyISAM PACK_KEYS=0 COMMENT='Links ' ;

```

7.4.6 Der SQL Code für die LernAss-Aufgaben-Datenbank

Dieser SQL Code wird beim Anlegen von LernAss-Aufgaben-Datenbanken erstellt.

```

CREATE TABLE 'Extra ' (
    'Key_Extra ' int(10) unsigned NOT NULL auto_increment ,
    'Key_Exercise ' int(10) unsigned NOT NULL default '0' ,
    'Extra_URL ' text NOT NULL ,
5  'Extra_Name ' text NOT NULL ,
    PRIMARY KEY ( 'Key_Extra ' )
) TYPE=MyISAM COMMENT='Links_auf_Extradataen ' ;

CREATE TABLE 'Hint ' (
10  'Key_Hint ' text NOT NULL ,
    'Hint_Hints ' text NOT NULL ,
    FULLTEXT KEY 'Hint_Hints ' ( 'Hint_Hints ' )
) TYPE=MyISAM COMMENT='Stichworte ' ;

15 CREATE TABLE 'Index ' (
    'Key_Exercise ' int(10) unsigned NOT NULL auto_increment ,
    'Key_Exercise_ID ' int(10) unsigned NOT NULL default '0' ,
    'Key_Source ' int(10) unsigned NOT NULL default '0' ,
    'Key_Topic ' int(10) unsigned NOT NULL default '0' ,
20  'Key_Hint ' text NOT NULL ,
    'Key_View ' int(10) unsigned NOT NULL default '0' ,
    'Key_Level ' tinyint(3) unsigned NOT NULL default '0' ,
    'Exercise_Aktiv ' tinyint(1) unsigned NOT NULL default '0' ,
    'Exercise_Info ' text NOT NULL ,
25  'Exercise_Show_Solution ' tinyint(1) unsigned NOT NULL default
        '0' ,
    'Exercise_Show_MultipleChoice ' tinyint(1) unsigned NOT NULL
        default '0' ,
    'Exercise_Show_Extra ' tinyint(1) unsigned NOT NULL default '0'
    ,
    PRIMARY KEY ( 'Key_Exercise ' ) ,
    FULLTEXT KEY 'Exercise_Info ' ( 'Exercise_Info ' )
30 ) TYPE=MyISAM COMMENT='Hauptindex_ueber_die_Aufgaben ' ;

```

```
CREATE TABLE 'Level' (  
    'Key_Level' tinyint(3) unsigned NOT NULL auto_increment ,  
    'Level_Name' text NOT NULL,  
35 PRIMARY KEY ('Key_Level'),  
    FULLTEXT KEY 'Level_Name' ('Level_Name')  
) TYPE=MyISAM COMMENT='Aufgabenlevel';
```

```
CREATE TABLE 'Link' (  
40 'Key_LinkID' INT UNSIGNED NOT NULL AUTOINCREMENT ,  
    'Key_Link' INT UNSIGNED NOT NULL ,  
    'Key_Next_Link' INT UNSIGNED NOT NULL ,  
PRIMARY KEY ('Key_LinkID')  
) TYPE=MyISAM COMMENT='Aufgabenverlinkung';
```

```
45 CREATE TABLE 'Links' (  
    'Key_Link' int(10) unsigned NOT NULL auto_increment ,  
    'Link_Name' text NOT NULL,  
    'Link_Shortinfo' text NOT NULL,  
50 'Link_URL' text NOT NULL,  
    'Link_Picture_URL' text NOT NULL,  
PRIMARY KEY ('Key_Link')  
) TYPE=MyISAM COMMENT='Links';
```

```
55 CREATE TABLE 'Listexercises' (  
    'Key_Listexercise' INT UNSIGNED NOT NULL AUTOINCREMENT,  
    'Key_Listname' int(10) unsigned NOT NULL default '0',  
    'Key_Exercise' int(10) NOT NULL default '0',  
PRIMARY KEY ('Key_Listexercise')  
60 ) TYPE=MyISAM COMMENT='Aufgabenlisten_Daten';
```

```
CREATE TABLE 'Listname' (  
    'Key_Listname' int(10) unsigned NOT NULL auto_increment ,  
    'Listname_Name' text ,  
65 PRIMARY KEY ('Key_Listname')  
) TYPE=MyISAM COMMENT='Aufgabenlisten';
```

```
CREATE TABLE 'Metadata' (  
    'Teacher' text NOT NULL,  
70 'Subject' text NOT NULL,  
    'Actual_List' int(10) unsigned NOT NULL default '0'  
) TYPE=MyISAM COMMENT='Informationen_?ber_die_Datenbank';
```

```
CREATE TABLE 'MultipleChoice' (  
75 'Key_MultipleChoice' int(10) unsigned NOT NULL auto_increment ,  
    'Key_Exercise' int(10) unsigned NOT NULL default '0',  
    'Key_Source' int(10) unsigned NOT NULL default '0',  
    'Key_View' int(10) unsigned NOT NULL default '0',  
    'MultipleChoice_Is_Correct' tinyint(1) unsigned NOT NULL  
        default '0',  
80 PRIMARY KEY ('Key_MultipleChoice')  
    ) TYPE=MyISAM COMMENT='Multiple_Choice_Loesungen';
```

```
CREATE TABLE 'Solution' (  
    'Key_Solution' int(10) unsigned NOT NULL auto_increment ,  
85 'Key_Exercise' int(10) unsigned NOT NULL default '0',  
    'Key_Source' int(10) unsigned NOT NULL default '0',  
    'Key_View' int(10) unsigned NOT NULL default '0',  
    'Solution_Step' tinyint(3) unsigned NOT NULL default '0',  
    PRIMARY KEY ('Key_Solution')  
90 ) TYPE=MyISAM COMMENT='Loesungen_zu_den_Aufgaben';
```

```
CREATE TABLE 'Source' (  
    'Key_Source' int(10) unsigned NOT NULL auto_increment ,  
    'Source_Data' mediumblob NOT NULL,  
95 'Source_Name' text NOT NULL,  
    'Source_Type' text NOT NULL,  
    PRIMARY KEY ('Key_Source')  
    ) TYPE=MyISAM COMMENT='OLE_Daten_der_Aufgaben';
```

```
100 CREATE TABLE 'Topics' (  
    'Key_Topic' int(10) unsigned NOT NULL auto_increment ,  
    'Topic_Name' text NOT NULL,  
    PRIMARY KEY ('Key_Topic') ,  
    FULLTEXT KEY 'Topic_Name' ('Topic_Name')  
105 ) TYPE=MyISAM COMMENT='Fachgebiete_der_Aufgaben';
```

```
CREATE TABLE 'View' (  
    'Key_View' int(10) unsigned NOT NULL auto_increment ,  
    'View_Data' mediumblob NOT NULL,  
110 PRIMARY KEY ('Key_View')  
    ) TYPE=MyISAM COMMENT='Bilder_fuer_das_Internet';
```

7.4.7 Entfernen der GRANTs durch REVOKE

Sollte es einmal notwendig werden, kann man die GRANTs durch REVOKEs aufheben.

Das Zurücknehmen der Rechte erfolgt so:

```
REVOKE all privileges on testLa from test
```

Bei MySQL ist es jedoch besser, dass man die Einträge löscht, wenn der Benutzer keine Rechte mehr haben soll:

```
DELETE FROM MySQL.users WHERE User="Name"
```


7.4.8 Einrichtung der Webapplikation

Die Webapplikation kann man einfach in ein Webverzeichnis Kopieren. Damit die Webapplikation auch durch einfaches kopieren der Dateien funktioniert muss man die Applikation durch das „Internet-Informationdienste“ MMC Plugin einrichten:

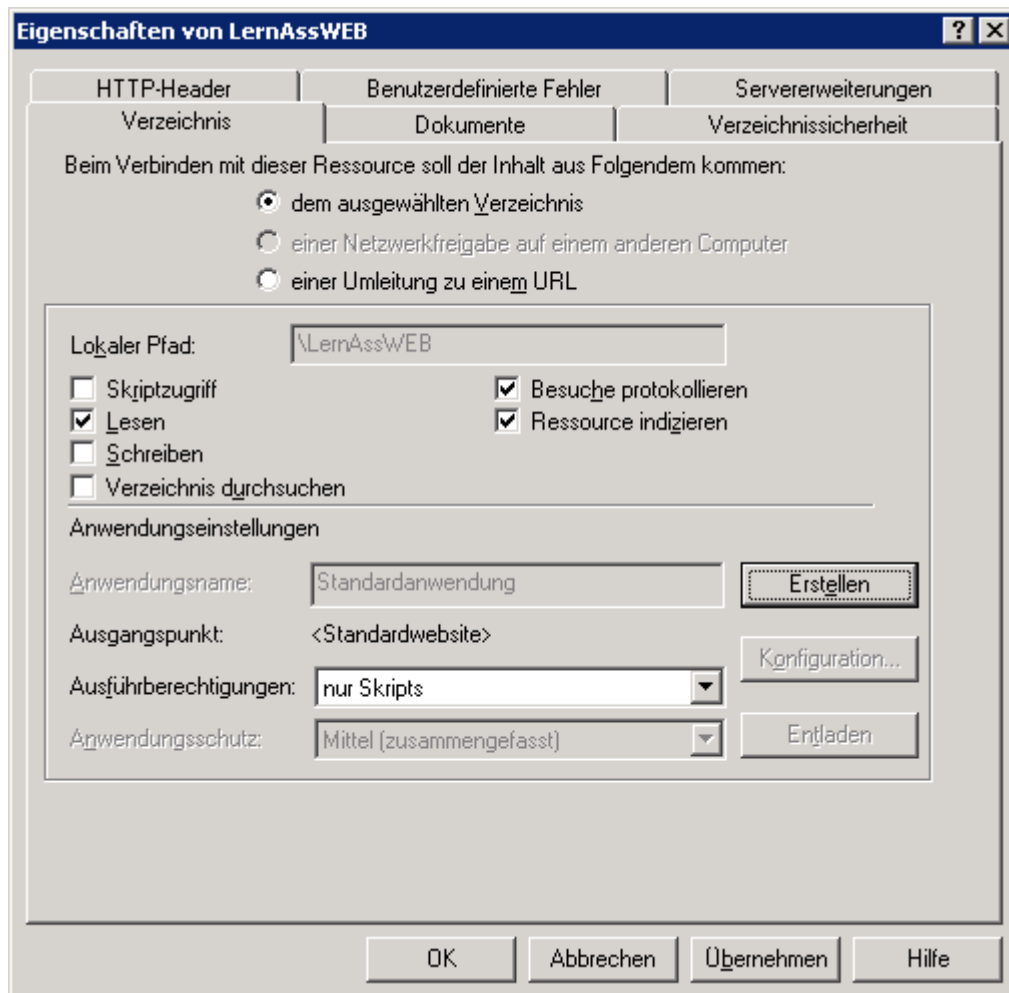


Abbildung 7.30: Applikationseinstellung bei den Internetinformationsdiensten

In dem Reiter „Verzeichnis“ muss man auf „Erstellen“ drücken, um die Applikation zu initialisieren:

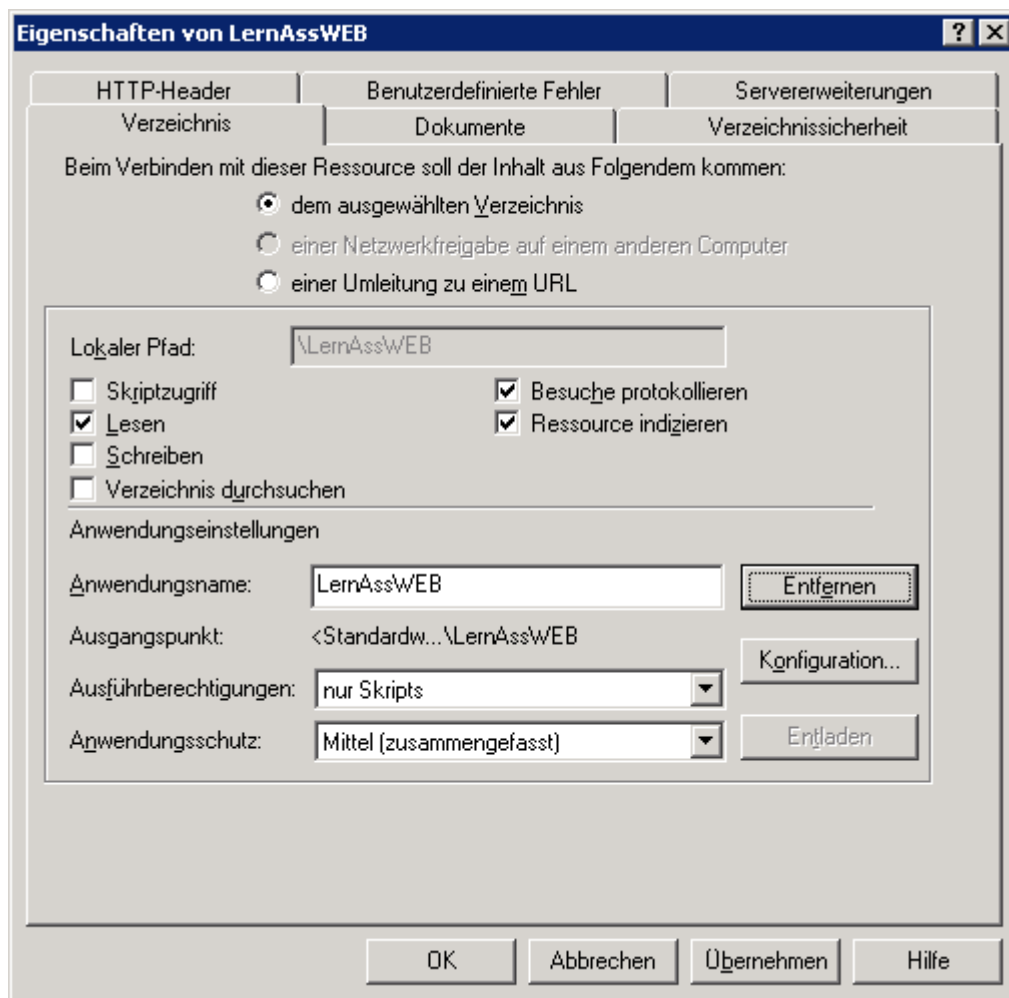


Abbildung 7.31: WEB-Applikation erstellen

A Flussdiagramme

Inhaltsangabe

A.1 Flussdiagramme	I
A.1.1 Aufgabeneingabe	III
A.1.2 LernAss Exerciser	IV
A.1.3 LernAss Konverter	V
A.1.4 LernAss Web	VI
A.2 Funktionsabläufe	VII
A.2.1 Datenbankfunktionen	VII
A.2.2 Aufgabenfunktionen	VIII
A.2.3 Zusatzdaten	VIII
A.2.4 Verweise	IX
A.2.5 Stichworte	IX

Abbildungen

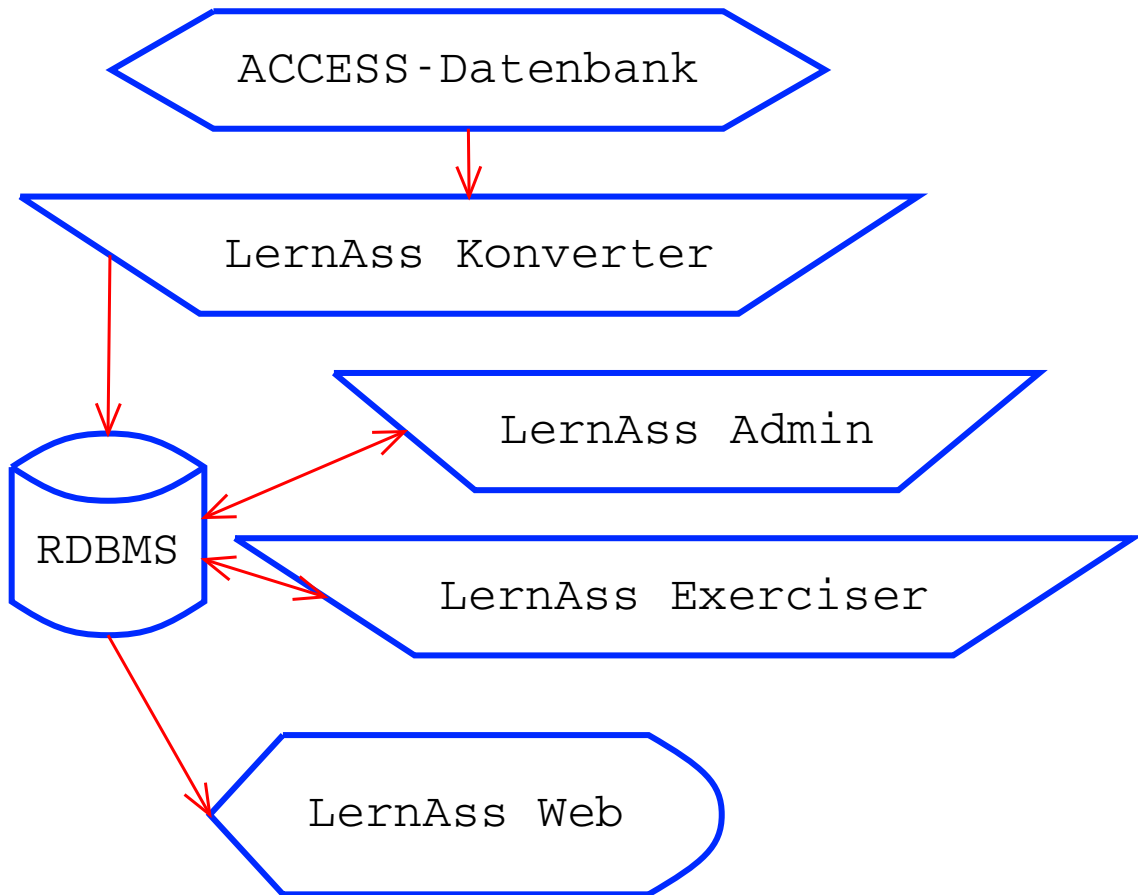
A.1 Datenfluss in LernAss	II
A.2 Aufgabeneingabe	III
A.3 LernAss Exerciser	IV
A.4 LernAss Konverter	V
A.5 LernAss Web	VI

A.1 Flussdiagramme

Die Flussdiagramme geben einen Überblick über die Funktionsweise der Programme. Anhand dieser Flussdiagramme wurden die Funktionen von LernAss 2003 programmiert.

Der generelle Datenfluss in LernAss sieht folgendermaßen aus:

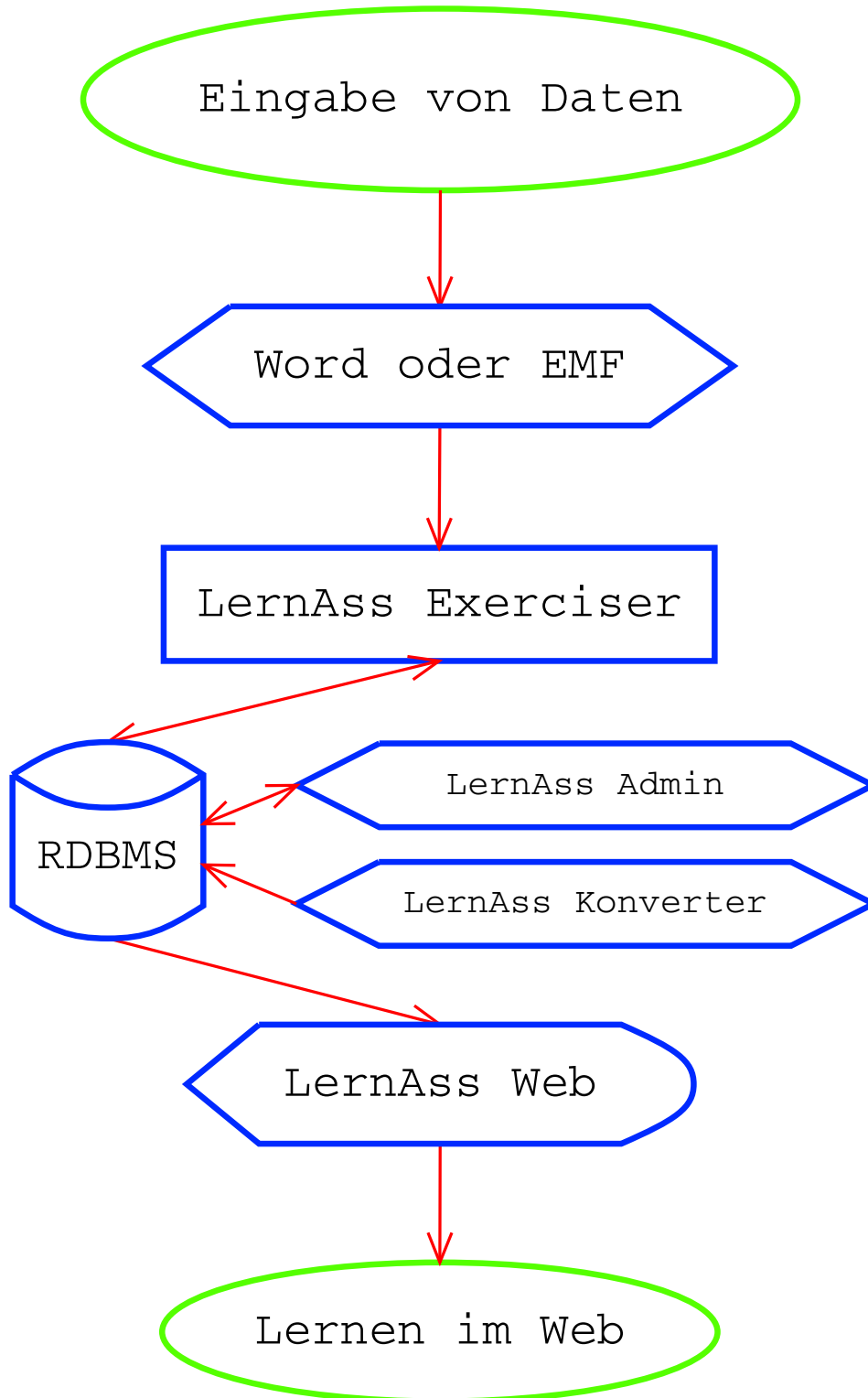
Abbildung A.1: Datenfluss in LernAss



Nachfolgend werden die Flussdiagramme detaillierter dargestellt.

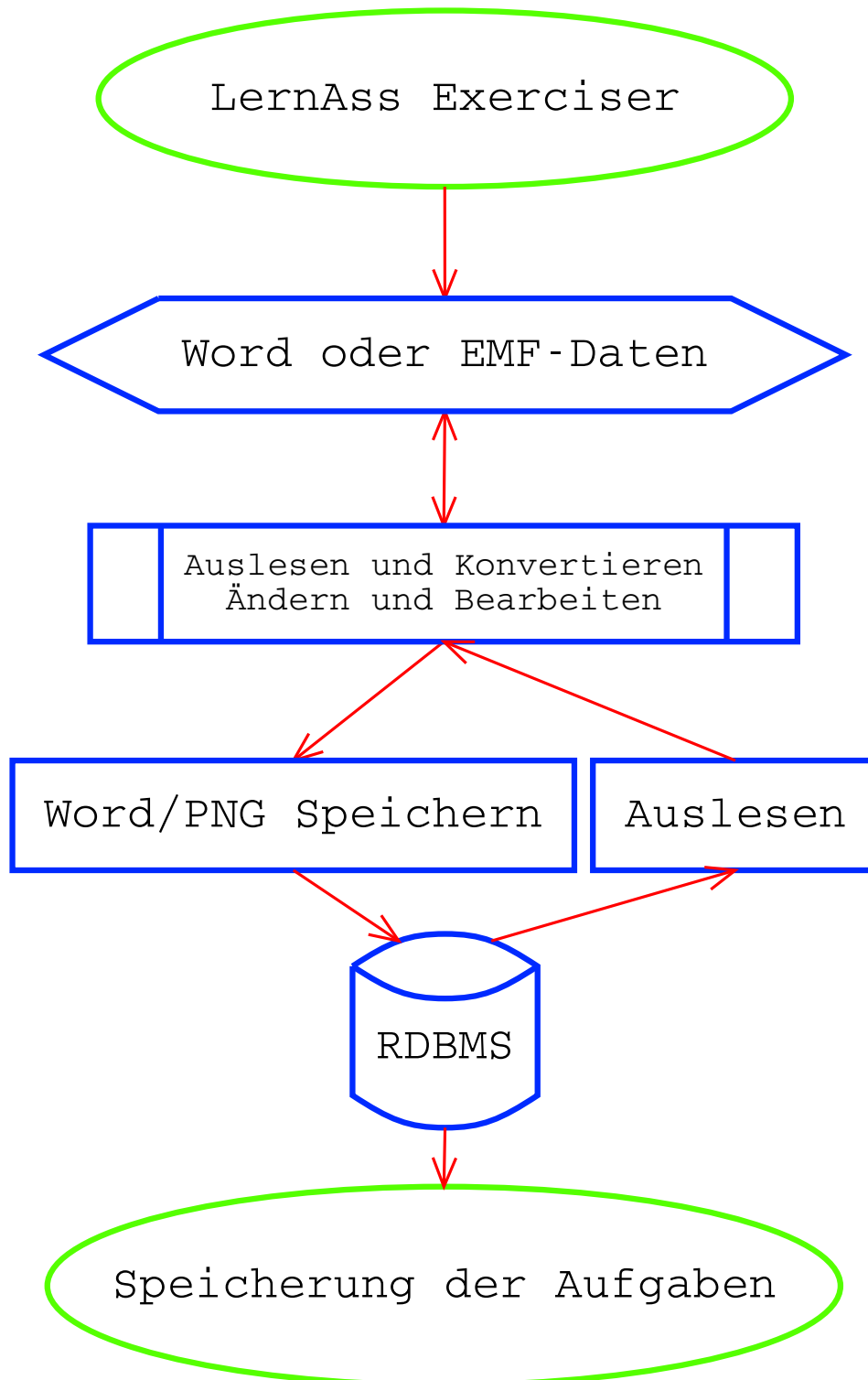
A.1.1 Aufgabeneingabe

Abbildung A.2: Aufgabeneingabe



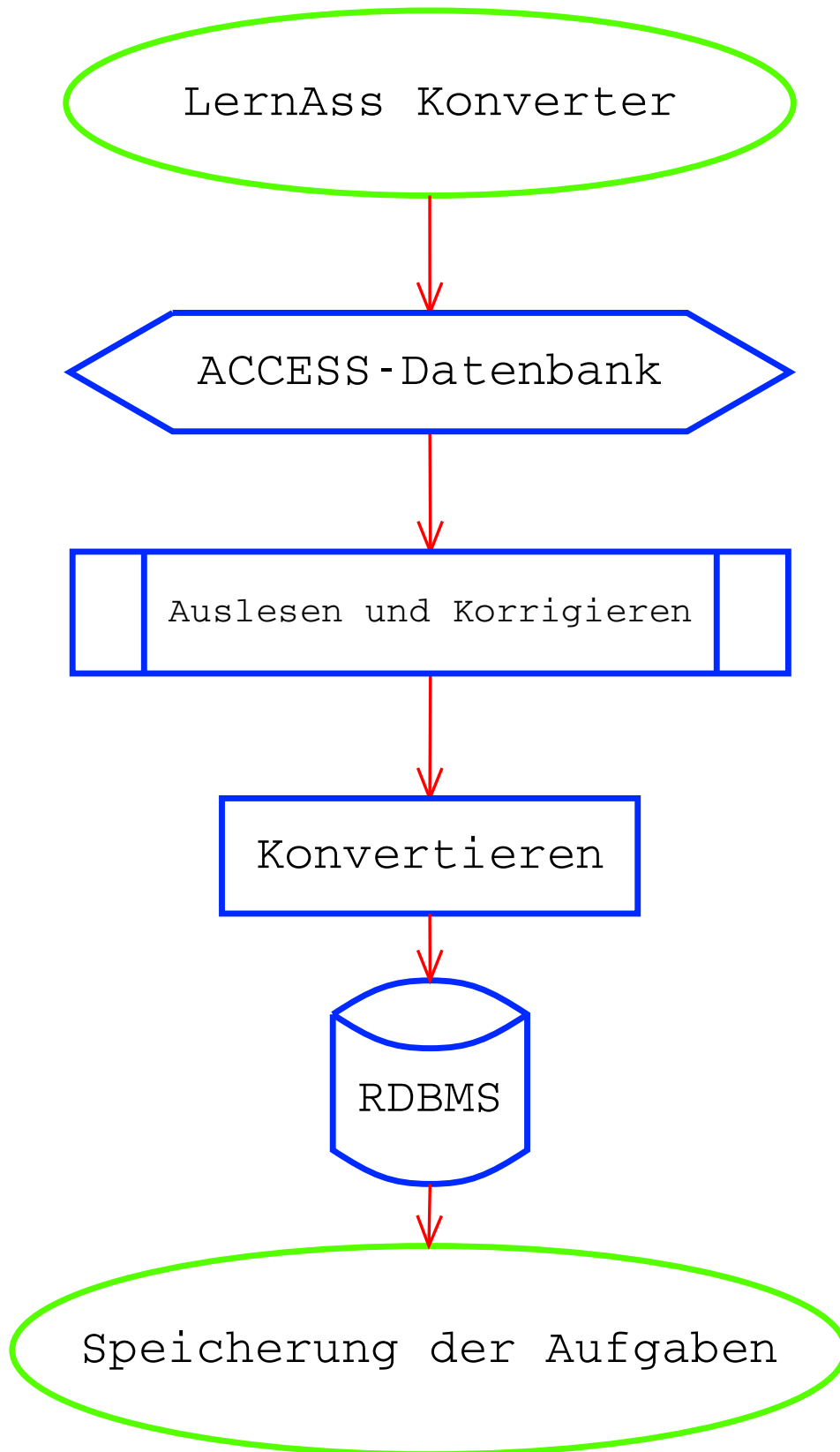
A.1.2 LernAss Exerciser

Abbildung A.3: LernAss Exerciser



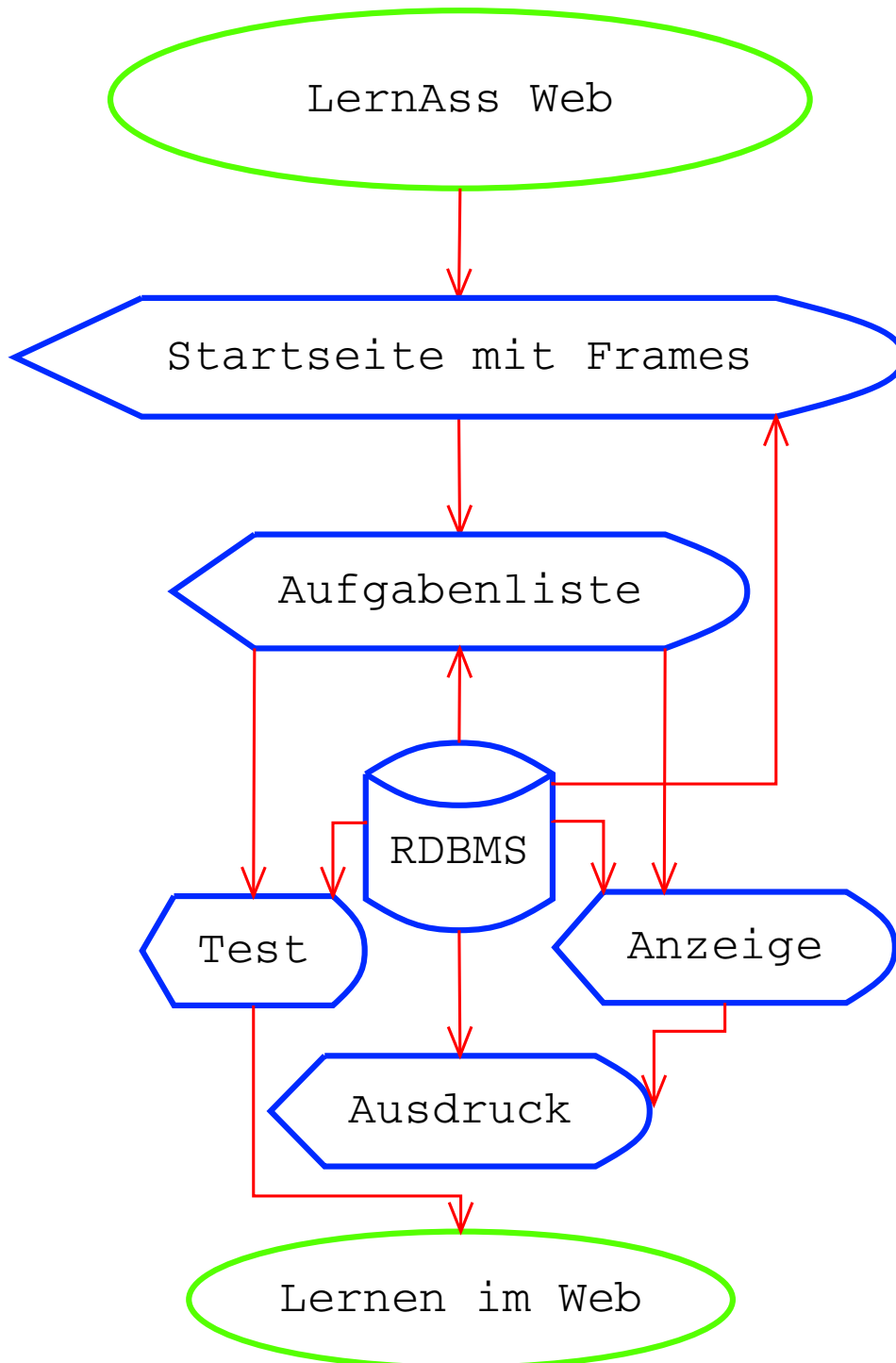
A.1.3 LernAss Konverter

Abbildung A.4: LernAss Konverter



A.1.4 LernAss Web

Abbildung A.5: LernAss Web



A.2 Funktionsabläufe

Nachfolgend werden noch einzelne Funktionsabläufe näher erläutert.

A.2.1 Datenbankfunktionen

Datenbank öffnen

- `ConnectionString` zusammensetzen (Benutzer, Passwort, Datenbank, Standort)
- Aufruf Datenbankobjekt kopieren
- Öffnen der Verbindung zur Datenbank mit dem `ConnectionString`
- Rückgabe des Erfolges oder Fehlschlages

Datenbank öffnen

- Prüfen, ob das Datenbankobjekt existiert
- Schließen der Datenbank

Datenbank in LernAss einpflegen

- Daten sammeln
- Prüfung auf Existenz des Datensatzes und der Datenbank → gegebenenfalls Import einer existenten Datenbank
- Gegebenenfalls Datenbank mit allen Tabellen anlegen
- Datenbank in der Tabelle „LernAss.Index“ eintragen mit Prüfung auf Vorhandensein

Datenbank löschen

- Sicherheitsabfrage und Frage, ob Datenbank gelöscht werden soll
- Datenbank aus der Tabelle „LernAss.Index“ entfernen
- Datenbank gegebenenfalls löschen

A.2.2 Aufgabenfunktionen

Aufgabe anzeigen

- Prüfung auf Anzeigerlaubnis (Auswahlliste bestimmt → Aktiv-Flag)
- Holen der Aufgabe aus der Tabelle „View“
- Prüfung und eventuell Anzeige von Zusatzdaten (Lösungen, Tests, Audio/Video/Grafik)
- Prüfung und eventuell Anzeige von Verweisen (Andere Aufgaben, Skriptstellen)
- Gegebenenfalls Verweis zur nächsten Aufgabe in der Datenbank

Aufgabe eingeben

- Dateneingabe über MS-Office oder anderen OLE-fähigen Provider
- Speicherung/Konvertierung über Programm mit OLE zu Bild-Konvertierung
- Hinweis auf weitere Bearbeitung (Zusatzdaten, Verweise)
- Bearbeitung von Verweisen und Zusatzdaten

Aufgabe löschen

- Prüfung ob eine andere Aufgabe noch auf diese Aufgabe verweist
- Löschen der Aufgabe

A.2.3 Zusatzdaten

Zusatzdaten einfügen

- Upload der Zusatzdaten (in die Datenbank)
- Setzen des MIME-Typs (gegebenenfalls erstellen)
- Zugehörigkeit festlegen (Key_Exercise)

Zusatzdaten löschen

- Ohne Prüfung löschen

A.2.4 Verweise

Verweis erstellen

- Verweise durch Auswahl bei den Aufgaben erstellen
- Speichern

Verweis entfernen

- Verweis ohne Prüfung entfernen

A.2.5 Stichworte

Hauptstichwort hinzufügen

- Prüfung auf Existenz des Hauptstichworts: wenn nicht existent, anlegen
- Erstes der Stichworte entspricht Hauptstichwort

Stichwort löschen

- Prüfung auf Existenz des Hauptstichwortes in der Aufgabenliste: wenn nicht existent, Löschung erlauben
- Löschung des Hauptstichwortes samt aller Stichworte

Stichwort hinzufügen

- Prüfung auf Existenz des Hauptstichwortes in der Aufgabenliste
- Löschung des Hauptstichwortes samt aller Stichworte

B Grafiken für LernAss Web

Inhaltsangabe

B.1 Einleitung	X
B.2 Erstellung der Bilder	X
B.3 Bilderablage auf dem Server	XIV

Abbildungen

B.1 Öffnen der Bildvorlage in Paintshop Pro 5	XI
B.2 Knopf Zeichensatz auswählen	XII
B.3 Knopf Bezeichnung eingeben	XIII
B.4 Knopf speichern	XIV

B.1 Einleitung

Die FH Gießen-Friedberg setzt AGFA Rotis als Schriftart zwecks Corporate Design voraus. Die Bearbeitung und Erstellung der Bilder erfolgte mit Paintshop Pro 5.

B.2 Erstellung der Bilder

Es existiert eine Vorlage: „vorlage_lang.png“. Diese Vorlage wird in Paintshop Pro 5 geöffnet.

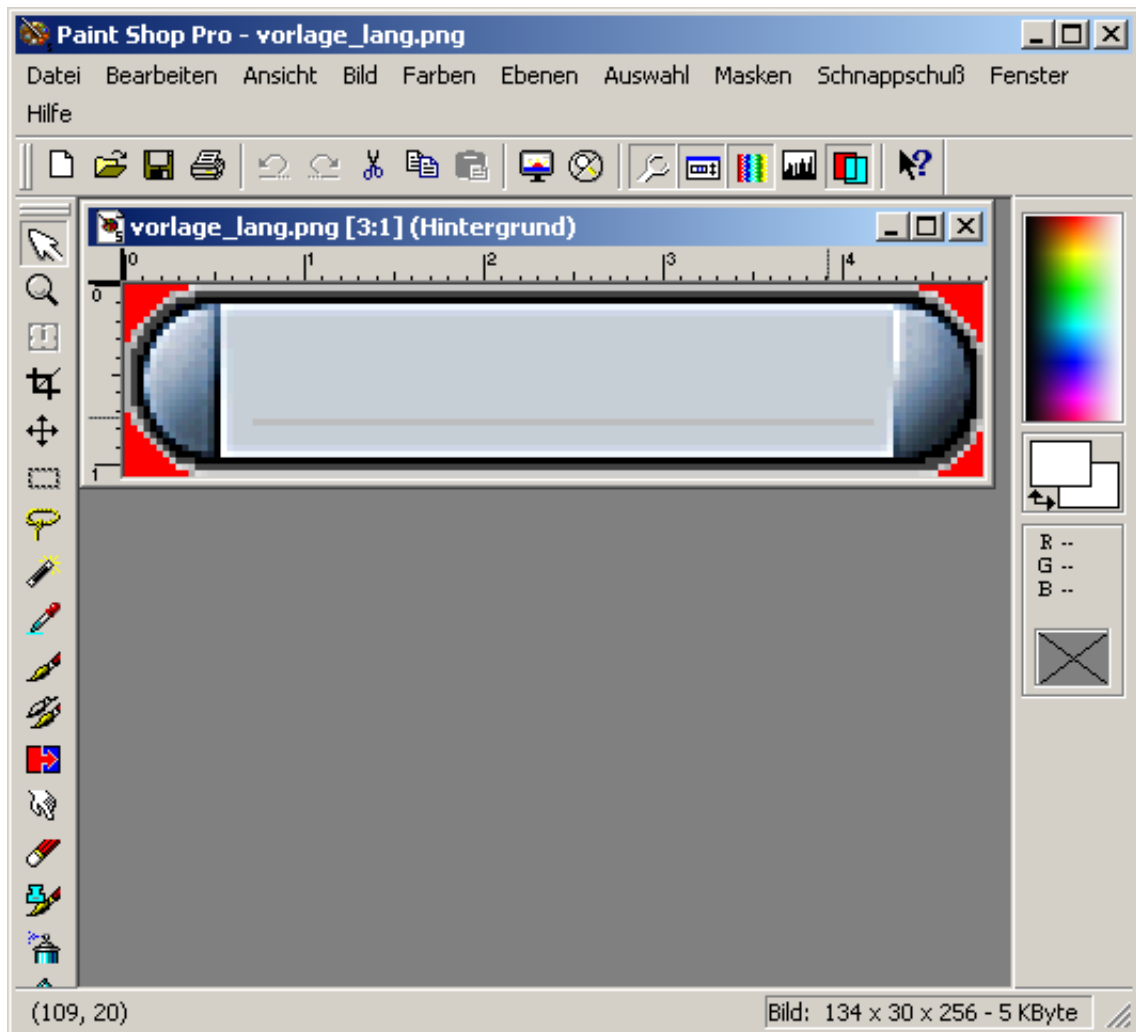


Abbildung B.1: Öffnen der Bildvorlage in Paintshop Pro 5

Unter dem Menüpunkt „Farben“ muss man die Farbanzahl auf 16 Millionen Farben erhöhen, damit das Aliasing beim Zeichnen der Schrift funktioniert.

Dann drückt man auf das „A“ für das Textwerkzeug. Dort stellt man den Zeichensatz „AGFA Rotis Sans Ex Bold“ in der Schriftgröße 12pt und Fett ein.

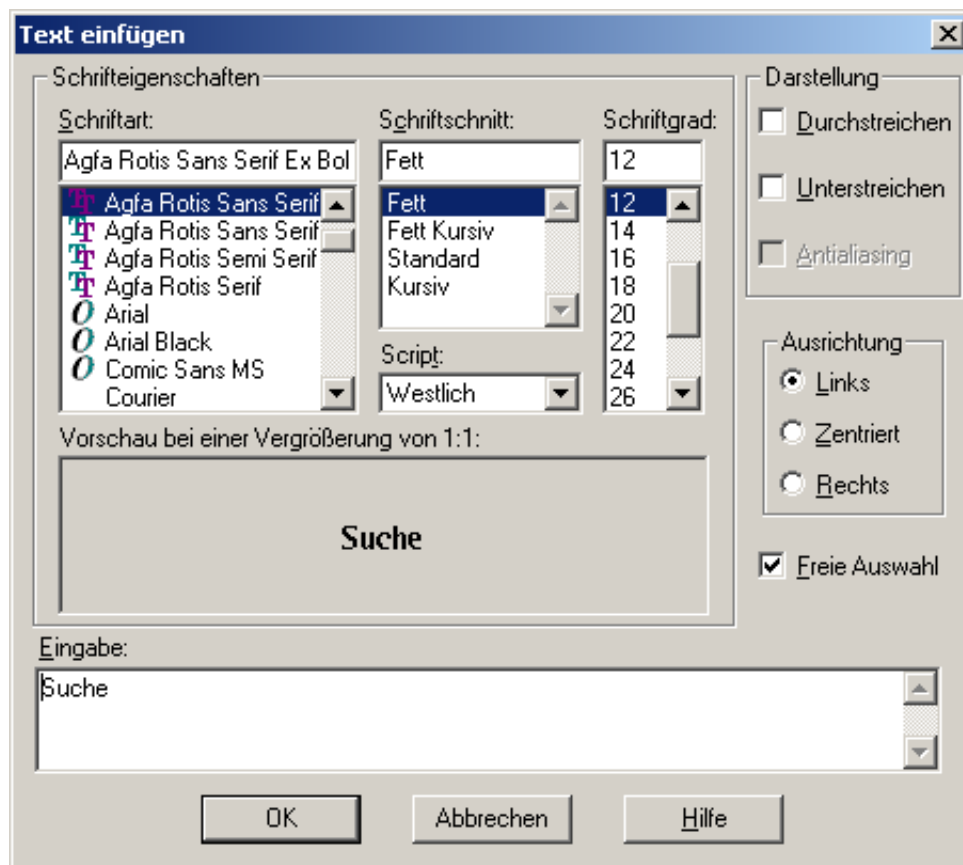


Abbildung B.2: Knopf Zeichensatz auswählen

Dann muss man die Bezeichnung anhand der Hilfslinie zentrieren.

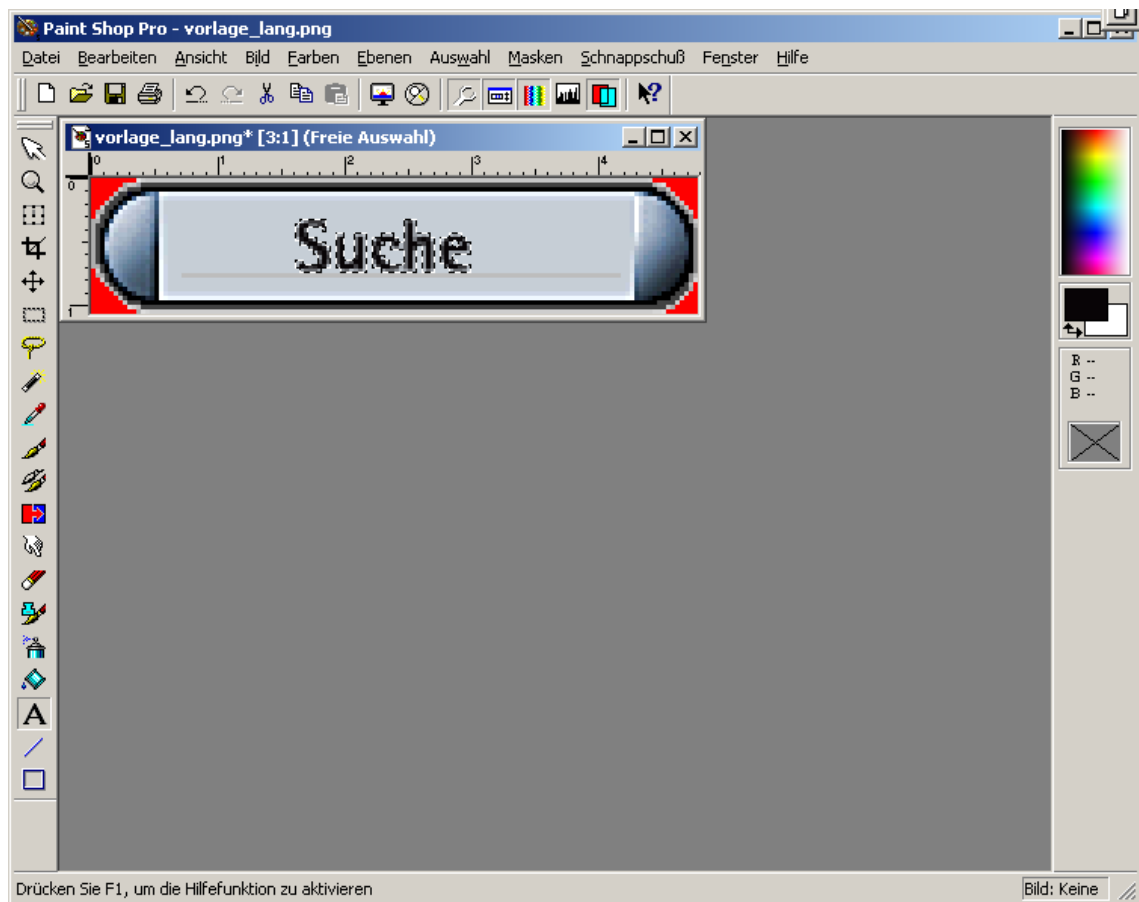


Abbildung B.3: Knopf Bezeichnung eingeben

Danach ist unter dem Menüpunkt „Auswahl“ die Option „Keine Auswahl“ zu wählen, und danach unter „Farben“ die Farbanzahl auf 256 zu verringern. Um die Erstellung abzuschließen, selektiert man mit dem Farbpicker die Hintergrundfarbe und füllt die Hilfslinie. Dann kann man das fertige Bild unter dem gewünschten Namen abspeichern.

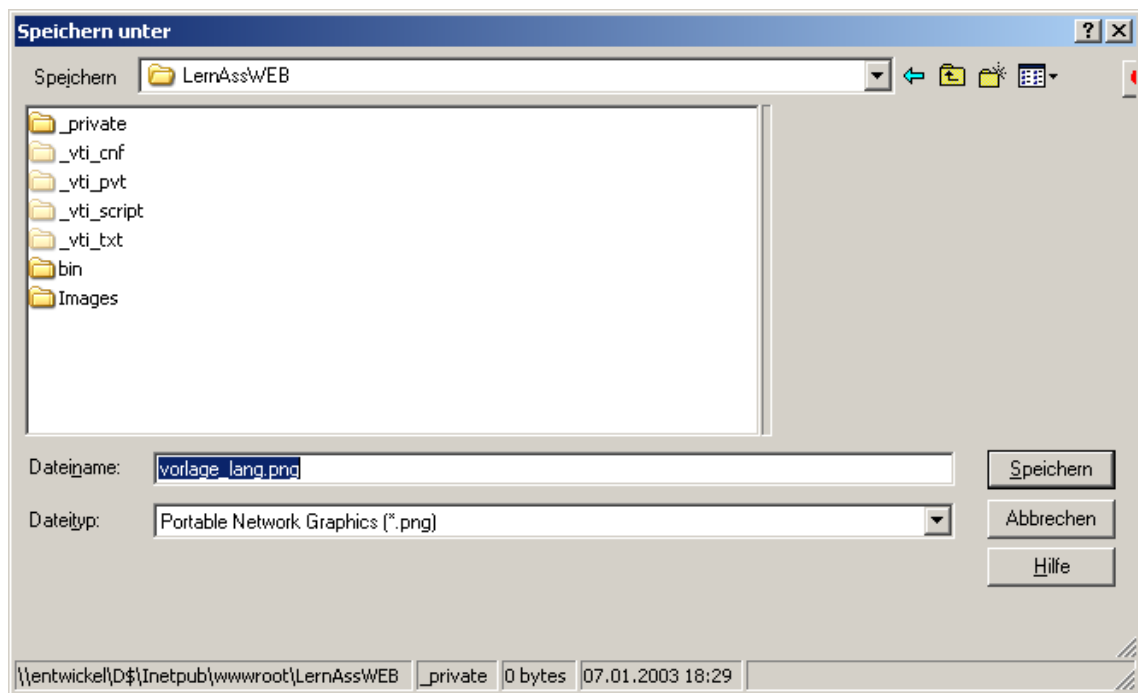


Abbildung B.4: Knopf speichern

Damit ist die Erstellung eines Bildes abgeschlossen.

B.3 Bilderablage auf dem Server

Im Sessioncache von LernAss Web muss auf die IURL (Bilder URL) verwiesen werden. Diese muss korrekt eingestellt sein.

Der Pfad zu den Bilder muss relativ zu LernAss Web sein. Beispiel: LernAss Web liegt in „C:\Inetpub\wwwroot\LernAssWEB“, dann muss die IURL „Images/“ oder ähnlich lauten.

C Internet Links

Inhaltsangabe

C.1 Allgemein	XV
C.2 E-Mailadressen	XV
C.3 .NET	XVI
C.4 Alternative Links zu .NET	XVI
C.5 Datenbanken	XVII
C.6 PHP	XVII

C.1 Allgemein

Alle hier folgenden Links waren am Abrufdatum 05.08.2003 erreichbar.

<http://www.andreasrehm.de>

Clipboardformatviewer – ein nützliches Tool zum Ansehen des Clipboards:

<http://www.codeproject.com/csharp/clipboardformatviewer.asp>

C.2 E-Mailadressen

E-Mail an Andreas Rehm:

<mailto:mail@andreasrehm.de>

E-Mail an Prof. Dr. rer. nat. Manfred Merkel:

<mailto:Manfred.Merkel@mnd.fh-friedberg.de>

E-Mail an Prof. Dr.-Ing. Wolf-Rainer Novener:

<mailto:wolf-rainer.novender@iem.fh-friedberg.de>

C.3 .NET



<http://www.microsoft.com>
<http://www.microsoft.com/germany/themen/net/>
<http://www.microsoft.com/germany/themen/net/plattform/>
<http://www.microsoft.com/germany/ms/entwicklerprodukte/visualnet/>
<http://www.sourceforge.net/projects/mysqldotnet>
<http://www.asp.net/webmatrix/>
<http://www.microsoft.com/myservices/>



<http://msdn.microsoft.com/germany/>

Clipboard Unterschied von VB6 zu VB.NET:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbcon/html/vxconchangestoclipboardobjectinvisualbasicnet.asp>

Microsoft Newsgroup VB.NET:

<http://support.microsoft.com/newsgroups/default.aspx?ICP=GSS3&NewsGroup=microsoft.public.de.german.entwickler.dotnet.vb&SLCID=DE&sd=GN&id=fh;DE;NEWSGROUPS>

C.4 Alternative Links zu .NET

OpenSource .NET Umgebung:

<http://go-mono.net>

OpenSource Entwicklungsumgebung:

<http://icsharpcode.net/OpenSource/SD/default.asp>

Datagrid und andere Beispiele sind auf dieser Seite zu finden:

<http://rustemsoft.com/Beginner.htm>

C.5 Datenbanken

<http://www.mysql.com>

<http://www.mysql.com/articles/dotnet/>

<http://www.sourceforge.net/projects/mysqldotnet>

C.6 PHP

<http://www.php.net>

PHP Webadministration:

<http://www.sourceforge.net/projects/phpmyadmin>

D Adressen

D.1 Andreas Rehm

Der Diplomand ist zu erreichen unter:

Webseite von Andreas Rehm:

<http://www.andreasrehm.de>

E-Mail-Adresse von Andreas Rehm:

<mailto:mail@andreasrehm.de>

Postalisch unter:	Telefonisch:
Andreas Rehm	Mobil : +49 173 7034684
Kurt-Moosdorf-Str. 36	Telefon: +49 6047 950180
D-63694 Limeshain-Rommelhausen	Fax: +49 6047 951156

D.2 Fachhochschule Gießen-Friedberg

Die Fachhochschule Gießen-Friedberg ist erreichbar unter:

www.fh-giessen-friedberg.de

Postalisch unter:	Telefonisch:
Fachhochschule Gießen-Friedberg	Telefon : +49 6031 604-0
University of Applied Sciences	Fax: +49 6031 604-180
Bereich Friedberg	
Wilhelm-Leuschner-Straße 13	
61169 Friedberg/Hessen	

D.3 Prof. Dr. rer. nat. Manfred Merkel

Der Referent ist zu erreichen unter:

Webseite des Fachbereiches MND:

<http://mnd-w1.fh-friedberg.de/>

E-Mail Adresse von Prof. Dr. rer. nat. Manfred Merkel:

<mailto:Manfred.Merkel@mnd.fh-friedberg.de>

Postalisch unter:	Telefonisch:
Fachhochschule Gießen-Friedberg Bereich Friedberg Zu Händen Herrn Prof. Dr. Manfred Merkel Wilhelm-Leuschner-Straße. 13 61169 Friedberg/Hessen	Telefon : +49 6031 604-426

D.4 Prof. Dr.-Ing. Wolf-Rainer Novender

Der Referent ist zu erreichen unter:

Webseite des Fachbereiches IEM:

<http://www.fh-friedberg.de/fachbereiche/iem/pers/prof.htm>

E-Mail Adresse von Prof. Dr.-Ing. Wolf-Rainer Novender:

<mailto:wolf-rainer.novender@iem.fh-friedberg.de>

Postalisch unter:	Telefonisch:
Fachhochschule Gießen-Friedberg Bereich Friedberg Zu Händen Herrn Prof. Dr.-Ing. Wolf-Rainer Novender Wilhelm-Leuschner-Straße. 13 61169 Friedberg/Hessen	Telefon : +49 6031 604- 229

E To do

E.1 Einleitung

LernAss 2003 ist gut, aber man kann es noch besser machen. Dieses Kapitel soll nur kurz aufzeigen, was noch an LernAss 2003 getan werden muss und was man noch machen kann.

Die hier genannten Punkte werden aus Zeitgründen nicht mehr vor Ende der Diplomarbeit in LernAss 2003 einfließen.

E.2 Was man noch verbessern kann

- Generelles Redesign in Richtung Benutzungsfreundlichkeit (mehrere CSS).
- Hilfesystem (sehr aufwändig).
- Weitere Daten anbieten (strukturelle Erweiterungen oder per URL).
- Benutzerverwaltung und Lernüberwachung
- Vereinheitlichung der Funktionen (statt Fehlerrückgabe als String eine Boolean Rückgabe und die Fehler in LAST_ERR von MySQL-DB)
- Entfernen der Übergabe des Datenbanknamens bei allen Funktionen (Trennung von LernAss-Zentraldatenbank und -Aufgabendatenbank sollte auch in Programmen sauber erfolgen)
- Vereinheitlichung der Begriffe/Spaltennamen der Datenbanktabellen (Groß/Kleinschreibung testen!)
- Mehrsprachigkeit
- Beschreibung der Aufgaben nach allgemeinen Lernstandards wie [LOM](http://www.adlnet.org), www.adlnet.org oder SCORM 1 und 2.

E.3 Was noch erledigt werden muss

- Funktionstest der Datenbankschnittstelle, wenn die Datenbank nicht lokal auf dem Server liegt, insbesondere in Hinsicht auf die Grants.
- Umstellung der Datenbank auf referenzielle Integrität – um die Funktion von LernAss 2003 sicherzustellen, muss alles durchgetestet werden, aber dieser Schritt ist möglichst erst mit einer MySQL Version 5 oder einer anderen Datenbank zu realisieren.
- Kontrolle und gegebenenfalls Korrektur aller Kommentare im Quellcode
- Kontrolle aller möglichen Fehlermeldungen (und Einführung eines Debug Flags)
- Datenbankbereinigungs- und Optimierungsprogramm (verwaiste Einträge entfernen, Tabellen optimieren, alle Datenbanken bearbeiten und nach verwaisten Datebankeinträgen suchen).

F Sourcecode

Der Sourcecode umfasst mehrere tausend Zeilen, weswegen er nur auf die CD gebrannt wird. Die Unmengen an Codes würden zu viele Seiten füllen und sich nur sehr schlecht lesen lassen. In der VisualStudio-Entwicklungsumgebung ist der Code wegen Regions (Regionen im Code können benannt und ausgeblendet werden) und Objekterkennung ausblendbar. Außerdem ist der Code dann mit Syntaxhighlighting versehen und man hat einen Objektbrowser.

Literaturverzeichnis

- [1] Sommer, Jürgen und Vortisch, Arne: LernAss V2.2. *Der Lernassistent für Windows. Diplomarbeit, eingereicht an der Fachhochschule Gießen-Friedberg, Bereich Friedberg, Juli 2000, 182 Seiten)*
- [2] Meier, Andreas: Relationale Datenbanken. *Leitfaden für die Praxis, 4., überarbeitete und erweiterte Auflage, Berlin [u.a.], 2001. XIII, 225 Seiten, ISBN 3-540-41468-1.*
- [3] Kofler, Michael: Visual Basic .NET. *Grundlagen, Programmier Techniken, Windows-Anwendungen, München [u.a.], 2002. 1075 Seiten, ISBN 3-8273-1982-X.*
- [4] Wienands, Christoph: mysql_dotnet. *MySQL Dataprovider für .NET, 2002.*
http://www.sourceforge.net/projects/mysql_dotnet/