

Projekt Schulanmeldung

Umsetzungsdokumentation des

***UnNameD* TEAM**

Mitglieder:

Stefan Wittmann (Projektleitung)

Bernhard Kapp

Mathias Stepper

Andreas Döbeling

Gliederung

01. Projektbeschreibung	S. 1
02. Zeitplanung	S. 2
03. ER-Modell	S. 3
04. Logisches Datenmodell	S. 4
05. SQL-Kommandos zum Erstellen der Tabellen	S. 5 - 8
06. SQL-Befehl zum Erzeugen eines VIEW aller Schülerinformationen	S. 9
07. Insert eines kompletten Datensatzes	S. 10 – 11
08. Source Code	S. 12
09. Testfälle	S. 13
10. Kontrollzettel	S. 14
11. Anlage: - Screenshots - Source Code	S. 15 ff.

Projektbeschreibung

Als Projektziel wurde eine funktionierende Anmeldemaske in Form einer Website angestrebt. Zum Schutz vor Robot-Einträgen sollte die tatsächliche Eingabeseite durch eine vorgelagerte Startseite geschützt werden. Diese Startseite wurde mit einem Captcha versehen. Erst nach korrekter Eingabe dieses Validierungsfeldes gelangt der User in den Eingabebereich.

Zur besseren Übersicht der einzugebenden Daten wurde die Html-Seite sektionsweise aufgeteilt. So werden bspw. erst persönliche Daten abgefragt; durch einen Klick auf den „Weiter“-Button werden auf einer nächsten Seite Daten zur schulischen Vorbildung abgefragt. Dadurch wird dem Anmelder ein leichter Überblick verschafft, anstatt ihn mit sämtlichen Eingabefeldern zu überfordern.

Für die korrekte Navigation innerhalb dieser Seiten wird eine Sessionverwaltung verwendet. Ein Datenverlust bereits erledigter Einträge ist somit nicht möglich.

Beim Bestätigen einer Eingabemaske erfolgt die Validierung der Felder. Sollten Daten angegeben worden sein die dem Datenschema nicht entsprechen (bspw. keine gültige E-Mail Adresse oder zu kurze Postleitzahl) wird der Besucher per Warnhinweis darauf hingewiesen und kann seine Eingaben erst nach einer Korrektur fortsetzen.

Die Übertragung der Daten in die Datenbank erfolgt erst nach erfolgreichem Abschluss aller Eingaben. Dadurch werden unvollständige Datensätze, beispielsweise durch Abbruch eines Eingabevorgangs, vermieden.

Nachdem die Angaben in die Datenbank geschrieben wurden erhält der Anwender eine Erfolgs-Bestätigung.

Über eine geschützte Seite erreichen Sekretariatsmitarbeiter eine CSV-Exportfunktionalität. Der Export bedient sich eines VIEWS, welcher alle Tabellenverknüpfungen auflöst und dadurch sämtliche Daten eines Schülerdatensatzes zur Verfügung stellt. Der VIEW kann ohne großen Aufwand, durch Anpassung des ihm zu Grunde liegenden SELECT-Statements, angepasst werden. Somit kann die Reihenfolge und das Auftreten der Felder in der CSV-Datei gesteuert werden.

Ebenfalls angedacht war eine Löschfunktion bereits exportierter Datensätze. Zur Identifizierung sollte ein Boolean Feld im Schülerdatensatz bei erfolgreichem Export auf TRUE gesetzt werden. Anschließend kann mittels Filterung auf dieses Feld eine Löschung erfolgen. Aus Zeitgründen konnte diese Funktion leider nicht mehr fertig gestellt werden.

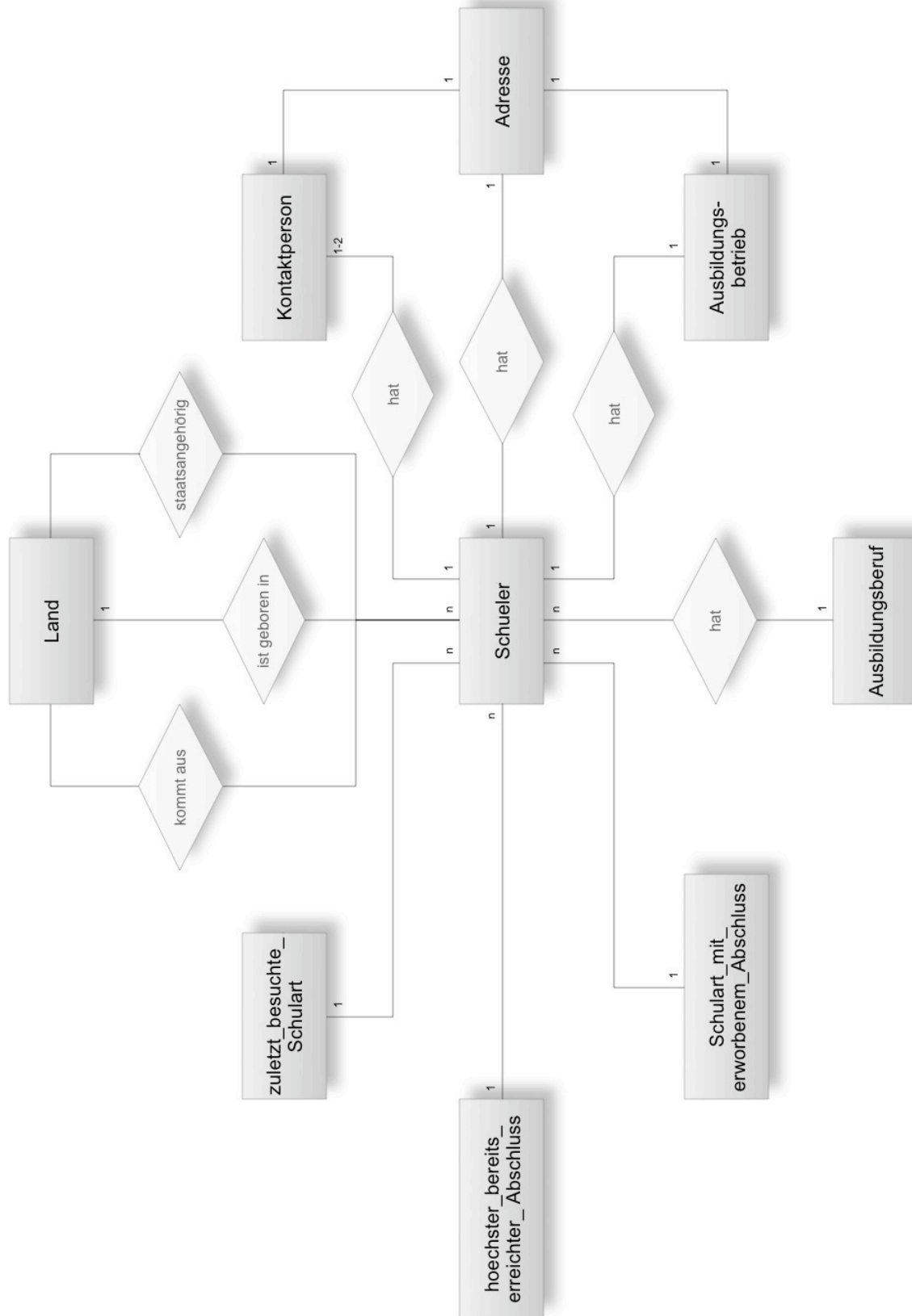
Zeitplanung

Alle Zeitangaben sind in Mann-Schulstunden angegeben.

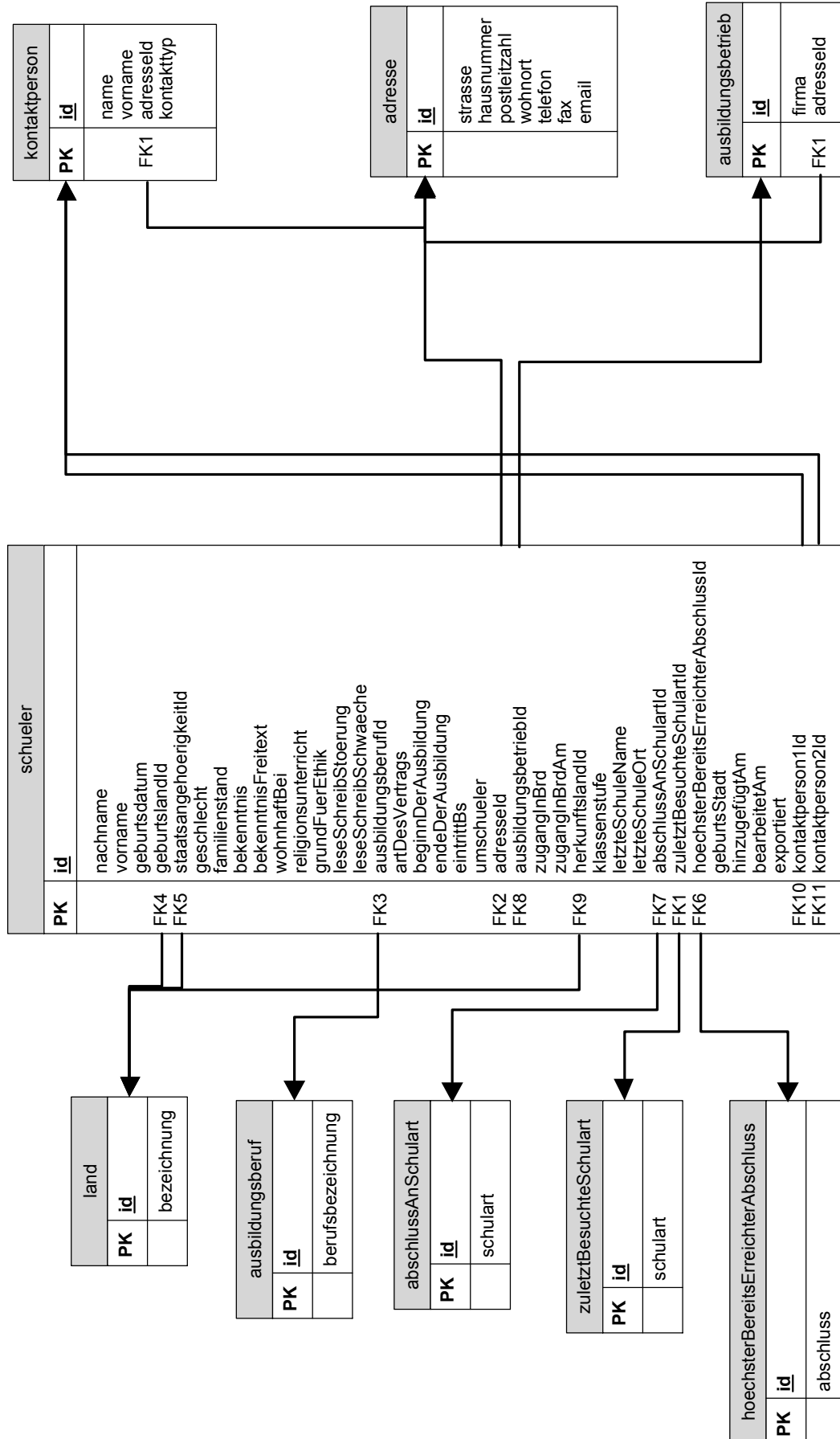
Als Basis werden 4 Leute x 38 Stunden = 152 Stunden angenommen.

	Beteiligte Personen	Zeitaufwand SOLL	Zeitaufwand IST
1. Datenbank			
1.1 Datenbankentwurf			
1.1.1 Analyse	Kapp Wittmann Stepper Döbeling	4 3 3 2	2 3 3 0,5
1.1.2 Design			
1.1.2.1 ER-Modell	Kapp Wittmann Stepper	1 2,5 2	1 2 2
1.1.2.2 Relationales Modell	Wittmann Stepper	4 4	5 4
1.1.3 Implementierung	Kapp Wittmann Stepper Döbeling	21,5 18 19 2	23 17 16 2
2. Oberfläche			
2.1 Pear-Umgebung aufbauen	Döbeling	3	4,5
2.2 Design	Kapp Wittmann Stepper Döbeling	1,5 0,5 1 3	2 0,5 1 3
2.3 Implementierung	Kapp Döbeling	3 22	3 22
3. Zusatzleistungen			
3.1 CSV-Datenexport	Wittmann Stepper	4 4	4 7
3.2 Captcha-Implementierung	Döbeling	6	6
4. Test	Kapp Wittmann Stepper	3 2 3	3 2,5 3
5. Dokumentation	Kapp Wittmann Stepper	4 4 2	4 4 2
Summe		152	152

ER-Modell



Logisches Datenmodell



SQL-Kommandos zum Erstellen der Tabellen

CREATE DATABASE schuelerverwaltung;

USE schuelerverwaltung;

CREATE TABLE `abschlussanschulart` (

 `id` int(10) unsigned NOT NULL AUTO_INCREMENT,

 `schulart` varchar(50) NOT NULL,

 PRIMARY KEY (`id`)

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE `adresse` (

 `id` int(10) unsigned NOT NULL AUTO_INCREMENT,

 `strasse` varchar(30) NOT NULL,

 `hausnummer` varchar(10) NOT NULL,

 `postleitzahl` varchar(5) NOT NULL,

 `wohnort` varchar(30) NOT NULL,

 `telefon` varchar(25) NOT NULL,

 `fax` varchar(25) DEFAULT NULL,

 `email` varchar(50) DEFAULT NULL,

 PRIMARY KEY (`id`)

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```
CREATE TABLE `ausbildungsberuf` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `berufsbezeichnung` varchar(80) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `ausbildungsbetrieb` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `firma` varchar(50) NOT NULL,  
  `adresseId` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `hoechsterbereiterreichterabschluss` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `abschluss` varchar(50) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `kontaktperson` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `name` varchar(30) NOT NULL,  
  `vorname` varchar(20) NOT NULL,  
  `kontakttyp` enum('Vater','Mutter','Vormund','Sonstige') NOT NULL,  
  `adresseId` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```



```
CREATE TABLE `land` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `bezeichnung` varchar(30) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `schueler` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `letzteSchuleName` varchar(50) NOT NULL,  
  `letzteSchuleOrt` varchar(50) NOT NULL,  
  `nachname` varchar(40) NOT NULL,  
  `vorname` varchar(40) NOT NULL,  
  `geburtslandId` int(10) unsigned NOT NULL,  
  `staatsangehoerigkeitId` int(10) NOT NULL,  
  `bekenntnisfreiText` varchar(40) DEFAULT NULL,  
  `wohnhaftBei` enum('Eltern','Vormund','eigene Wohnung') NOT NULL,  
  `leseSchreibStoerung` tinyint(1) DEFAULT NULL,  
  `leseSchreibSchwaechе` tinyint(1) DEFAULT NULL,  
  `zugangInBrд` enum('Aussiedler','Kriegsflüchtling','Asylant','Asylbewerber','sonstiger  
Zuzug') DEFAULT NULL,  
  `grundFuerEthik` enum('Austritt','Religionslosigkeit','kein Religionsunterricht') DEFAULT  
NULL,  
  `ausbildungsberufId` int(10) unsigned DEFAULT NULL,  
  `umschueler` tinyint(1) DEFAULT NULL,  
  `ausbildungsbetriebId` int(10) DEFAULT NULL,  
  `zuletztBesuchteSchulartId` int(10) unsigned NOT NULL,  
  `hoechsterBereitsErreichterAbschlussId` int(10) unsigned NOT NULL,  
  `adresseId` int(10) unsigned NOT NULL,  
  `herkunftslandId` int(10) unsigned DEFAULT '54',  
  `klassenstufe` enum('10','11','12') NOT NULL,
```

```

`abschlussAnSchulartId` int(10) unsigned NOT NULL,
`geburtsdatum` date NOT NULL,
`geschlecht` enum('männlich','weiblich') NOT NULL,
`zugangInBrdAm` date DEFAULT NULL,
`eintrittBs` date NOT NULL,
`geburtsStadt` varchar(50) NOT NULL,
`familienstand` enum('ledig','verheiratet','geschieden','getrennt lebend') NOT NULL,
`religionsunterricht` enum('rk','ev','Ethik') NOT NULL,
`beginnDerAusbildung` date NOT NULL,
`endeDerAusbildung` date NOT NULL,
`bekenntnis` enum('rk','ev','sonstige') NOT NULL,
`artDesVertrags` enum('Ausbildungsvertrag','BVJ','ohne Beruf/arbeitslos','ungelernte
Arbeitskraft','Umschüler','Teilnehmer an Lehrgang der Arbeitsverwaltung') NOT NULL,
`kontaktperson1Id` int(10) unsigned NOT NULL,
`kontaktperson2Id` int(10) unsigned DEFAULT NULL,
`hinzugefügtAm` datetime DEFAULT NULL,
`bearbeitetAm` datetime DEFAULT NULL,
`exportiert` tinyint(1) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `zuletztbesuchteschulart` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `schulart` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

SQL-Befehl zum Erzeugen eines VIEW aller Schülerinformationen

Der VIEW wird als Grundlage für den CSV-Dateien-Export verwendet. Die gesamten Datensätze werden aufgelöst und lassen sich somit leicht aus der Datenbank holen.

CREATE VIEW view_schueler AS

```
SELECT s.*, gl.bezeichnung geburtsland, sl.bezeichnung staatsangehoerigkeit, hl.bezeichnung  
herkunftsland, ab.firma, sa.strasse, sa.hausnummer, sa.postleitzahl, sa.wohnt, sa.telefon, sa.fax,  
sa.email, aba.strasse ab_strasse, aba.hausnummer ab_hausnummer, aba.postleitzahl ab_plz,  
aba.wohnt ab_ort, aba.telefon ab_telefon, aba.fax ab_fax, aba.email ab_email,  
abb.berufsbezeichnung ausbildungsberuf_bezeichnung, aas.schulart abschlussSchulart, zbs.schulart  
zuletztBesuchteSchulart, hbea.abschluss hoechsterAbschluss, kp1.name kp1_name, kp1.vorname  
kp1_vorname, kp1.kontakttyp kp1_kontakttyp, kpa1.strasse kpa1_strasse, kpa1.hausnummer  
kpa1_hausnummer, kpa1.postleitzahl kpa1_plz, kpa1.wohnt kpa1_ort, kpa1.telefon kpa1_telefon,  
kpa1.fax kpa1_fax, kpa1.email kpa1_email, kp2.name kp2_name, kp2.vorname kp2_vorname,  
kp2.kontakttyp kp2_kontakttyp, kpa2.strasse kpa2_strasse, kpa2.hausnummer kpa2_hausnummer,  
kpa2.postleitzahl kpa2_plz, kpa2.wohnt kpa2_ort, kpa2.telefon kpa2_telefon, kpa2.fax kpa2_fax,  
kpa2.email kpa2_email
```

```
FROM schueler s, ausbildungsbetrieb ab, adresse sa, adresse aba, adresse kpa1, adresse kpa2, land  
gl, land sl, land hl, ausbildungsberuf abb, abschlussAnSchulart aas, zuletztBesuchteSchulart zbs,  
hoechsterBereitsErreichterAbschluss hbea, kontaktperson kp1, kontaktperson kp2
```

```
WHERE s.ausbildungsbetriebId = ab.id
```

```
AND s.adresseId = sa.id AND ab.adresseId = aba.id
```

```
AND s.geburtslandId = gl.id
```

```
AND s.staatsangehoerigkeitId = sl.id
```

```
AND s.herkunftslandId = hl.id
```

```
AND s.ausbildungsberufId = abb.id
```

```
AND s.abschlussAnSchulartId = aas.id
```

```
AND s.zuletztBesuchteSchulartId = zbs.id
```

```
AND s.hoechsterBereitsErreichterAbschlussId = hbea.id
```

```
AND s.kontaktperson1Id = kp1.id AND s.kontaktperson2Id = kp2.id
```

```
AND kp1.adresseId = kpa1.id AND kp2.adresseId = kpa2.id;
```

Insert eines kompletten Datensatzes

Füllen der Tabelle Adresse mit der Adresse des Azubis:

```
INSERT INTO adresse (`strasse`, `hausnummer`, `postleitzahl`, `wohnort`,  
`telefon`, `email`, `id`) VALUES ('Musterstraße', '5', '91058',  
'Erlangen', '09133 123456', 'max@mustermann.de', 346)
```

Füllen der Tabelle Adresse mit den Daten d. 1. Kontaktperson:

```
INSERT INTO adresse (`strasse`, `hausnummer`, `postleitzahl`, `wohnort`,  
`telefon`, `email`, `id`) VALUES ('Musterstraße', '5', '91083',  
'Erlangen', '09133 123456', '', 347)
```

Füllen der Tabelle Kontaktperson mit den Daten der 1. Kontaktperson

```
INSERT INTO kontaktperson (`name`, `vorname`, `adresseId`, `kontaktyp`,  
`id`) VALUES ('Mustermann', 'Petra', 347, 'Sonstige', 348)
```

Füllen der Tabelle Adresse mit den Daten d. 2. Kontaktperson

```
INSERT INTO adresse (`strasse`, `hausnummer`, `postleitzahl`, `wohnort`,  
`telefon`, `email`, `id`) VALUES ('Müllerweg', '1', '91058', 'Erlangen',  
'09133 456789', 'ulrike@meyerversand.de', 349)
```

Füllen d. Tabelle Kontaktperson mit den Daten d. 2. Kontaktperson

```
INSERT INTO kontaktperson (`name`, `vorname`, `adresseId`, `kontaktyp`,  
`id`) VALUES ('Meyer', 'Ulrike', 349, 'Mutter', 350)
```

Füllen d. Tabelle Adresse mit den Daten d. Ausbildungsbetriebes

```
INSERT INTO adresse (`strasse`, `hausnummer`, `postleitzahl`, `wohnort`,  
`telefon`, `email`, `id`) VALUES ('Fabrikweg', '2', '91058', 'Erlangen',  
'09133 22558-99', 'azubi@müllerfabrik.de', 351)
```

Füllen d. Tabelle Ausbildungsbetrieb mit den Daten des Ausbildungsbetriebes

```
INSERT INTO ausbildungsbetrieb (`firma`, `adresseId`, `id`) VALUES  
('Müllerfabrik', 351, 352)
```

Füllen d. Tabelle Schüler

```
INSERT INTO schueler (`letzteSchuleName`, `letzteSchuleOrt`, `nachname`,  
`vorname`, `geburtslandId`, `staatsangehoerigkeitId`,  
`bekenntnisfreiText`, `wohnhafBei`, `leseSchreibStoerung`,  
`leseSchreibSchwaecher`, `zugangInBrd`, `grundFuerEthik`,  
`ausbildungsberufId`, `umschueler`, `ausbildungsbetriebId`,  
`zuletztBesuchteSchulartId`, `hoechsterBereitsErreichterAbschlussId`,  
`adresseId`, `herkunftslandId`, `abschlussAnSchulartId`, `geburtsdatum`,  
`geschlecht`, `zugangInBrdAm`, `eintrittBs`, `geburtsStadt`,  
`familienstand`, `religionsunterricht`, `beginnDerAusbildung`,  
`endeDerAusbildung`, `bekenntnis`, `artDesVertrags`, `kontaktperson1Id`,  
`kontaktperson2Id`, `hinzuuefuegtAm`, `id`) VALUES ('Bäckerfachschule',  
'Berlin', 'Mustermann', 'Max', '54', '54', NULL, 'eigene Wohnung', '0',  
'0', NULL, NULL, '7', '0', 352, '1', '1', 346, NULL, '1', '1972-07-11',  
'männlich', NULL, '2009-09-01', 'Dresden', 'verheiratet', 'rk',  
'2009-09-01', '2012-08-01', 'rk', 'Ausbildungsvertrag', 348, 350,  
'2009-02-02 21:13:55', 353)
```

Source Code

Im Anhang befinden sich sämtliche Source Codes. Auf Grund der Menge wurden die Codezeilen aus dieser Dokumentation ausgelagert.

Die Klasse „schueler“ ist zuständig für die Generierung des HTML-Formulars und dem Hinzufügen der eingegebenen Informationen in die mySQL-Datenbank. Darüber hinaus enthält sie zahlreiche weitere Methoden zur Steuerung und Validierung der Eingaben.

Die Klasse „session“ wird zur Sessionverwaltung und Seitensteuerung verwendet und übernimmt diverse benutzerbezogene Aufgaben wie beispielsweise die Ausnahmebehandlung (Errorhandling). Ebenfalls befindet sich hier die Logik und der Programmcode des Captchas.

Testfälle

Positiv-Testfälle:

Vorbedingung	Handlung	Erwartete Reaktion	Nachbedingung	Testfall erfolgreich
Auswählen des Feldes „Staatsangehörigkeit“ mit Wert „sonstige“	Auswählen eines Landes außer Deutschland	Freigabe der Felder „ZuganginBRD“ und Herkunftsland		<input type="checkbox"/>
Keine Vertragsart selektiert	Auswahl der Vertragsart „Ausbildungsvertrag“	Freigabe der Felder „Ausbildungsberuf“ und „Ausbildungsbetrieb“		<input type="checkbox"/>
Religionsunterricht nicht gewählt	Auswahl des Religionstypes „Ethik“	Freigabe des Feldes „GrundFuerEthik“		<input type="checkbox"/>
Startseite geöffnet	Wechseln der Teilseiten sowohl in Vorwärtsrichtung als auch in Rückwärtsrichtung	Getätigte Eingaben müssen noch vorhanden sein	Keine Veränderung der einzelnen Felddaten	<input type="checkbox"/>

Negativ-Testfälle:

Vorbedingung	Handlung	Erwartete Reaktion	Nachbedingung	Testfall erfolgreich
Geburtsdatumfeld nicht ausgefüllt	Eingabe eines falschen Datums in der Datumsmaske	Benutzer wird durch Dialog auf Fehler hingewiesen , mit Hinweis das falsches Geburtsdatum angegeben worden ist	Datum wird nicht gespeichert	<input type="checkbox"/>
Namensfeld nicht ausgefüllt	Eingabe eines zu langen Namens	Keine Eingabe über Feldlänge möglich		<input type="checkbox"/>
Keine Postleitzahl eingegeben	Eingabe eines Buchstaben anstatt einer Zahl in das Postleitzahlfeld	Benutzer wird durch Dialog auf Fehler hingewiesen, das PLZ nicht korrekt eingegebenen wurde		<input type="checkbox"/>
Auswahl eines freien beliebigen Feldes	Eingabe von SQL-Befehlen	Verwerfen des Feldinhaltes zur Absicherung gegen Code-Injektion		<input type="checkbox"/>

Kontrollzettel

1. Gruppeneinteilung	Gruppenmitglieder
	Stefan Wittmann (Projektleitung)
	Bernhard Kapp
	Mathias Stepper
	Andreas Döbeling
2. Datenbankentwurf	Kapp Wittmann Stepper Döbeling
3. Web-Site-Entwurf	Döbeling Kapp
4. Zusatzleistung	Döbeling Wittmann Stepper
5. Präsentation	Signum der Lehrkraft:
6. Dokumentation erstellt	Kapp Wittmann Stepper
7. Abnahme	Signum der Lehrkraft: