



universität  
wien

# Magisterarbeit

Titel der Magisterarbeit

„Softwareentwicklung für den Schulalltag  
am Beispiel einer Anwendung zur Planung  
der Nachmittagsbetreuung“

Verfasser

Bernhard Picher, Bakk. rer. soc. oec.

angestrebter akademischer Grad

Magister der Sozial- und Wirtschaftswissenschaften  
(Mag. rer. soc. oec.)

Wien, 2015

Studienkennzahl lt. Studienblatt:

A 066 922

Studienrichtung lt. Studienblatt:

Magisterstudium Informatikmanagement

Betreuer:

ao. Univ.-Prof. i.R. Dr. Erich Neuwirth



## **Plagiatserklärung**

Hiermit versichere ich, dass diese Diplomarbeit von mir persönlich verfasst ist und dass ich keinerlei fremde Hilfe in Anspruch genommen habe. Ebenso versichere ich, dass diese Arbeit oder Teile daraus weder von mir selbst noch von anderen als Leistungsnachweis andernorts eingereicht wurden. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen in gedruckter oder elektronischer Form sind gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und in der Bibliographie aufgeführt. Das Gleiche gilt für graphische Darstellungen und Bilder sowie für alle Internet-Quellen.

Wien, 2015

---





## **Danksagung**

An dieser Stelle möchte ich allen danken, die fachlich und persönlich zum Gelingen dieser Diplomarbeit beigetragen haben.

Mein Dank gilt ao. Univ.-Prof. i.R. Dr. Erich Neuwirth für die Betreuung während der Diplomarbeit und die freundliche Hilfsbereitschaft, die er mir stets entgegengebracht hat.

Ebenso gebührt all jenen Kolleginnen und Kollegen des Wienerwaldgymnasiums (Expositur des BG/BRG Purkersdorf) Dank, die im Zuge ihrer Tätigkeit in Zusammenhang mit der Nachmittagsbetreuung meine Software getestet und mir Feedback gegeben sowie Materialien zur Verfügung gestellt haben. Ohne ihre Bemühungen wäre diese Arbeit nicht zustande gekommen.

Danke!



## **Hinweis**

### **Nutzung der Materialien**

Für die in der Arbeit publizierten und auf dem beiliegenden Datenträger abrufbaren Materialien liegt das Urheberrecht bei Bernhard Picher. Die Daten dürfen ohne vorherige Rücksprache mit dem Urheber sowie dessen schriftliche Erlaubnis weder kopiert noch veröffentlicht werden. Bei einer erlaubten Weitergabe ist eine entsprechende Quellenangabe verpflichtend (© Bernhard Picher).

### **Datenschutz**

Alle in dieser Arbeit beispielhaft verwendeten persönlichen Daten sind frei erfunden. Bei Verwendung von Bildschirmfotos realer Daten werden die Bereiche mit sensiblen persönlichen Informationen durch lokale Unschärfe unkenntlich gemacht.



# Inhaltsverzeichnis

<b>1. Einleitung .....</b>	<b>11</b>
<b>2. Grundlagen .....</b>	<b>13</b>
2.1. Rechtliche Grundlagen .....	13
2.2. Begriffsdefinition .....	13
2.2.1. Mittagsbetreuung .....	13
2.2.2. Nachmittagsbetreuung .....	14
2.3. Rahmenbedingungen für die Entwicklung der Software .....	14
2.4. Umsetzung der Software .....	15
2.4.1. Art der Umsetzung .....	15
2.4.2. Verwendete Technologien .....	15
2.4.3. Alternative Umgebungsbedingungen .....	15
2.5. Umfang und Abgrenzung der Software .....	16
2.5.1. Die Software soll ... ..	16
2.5.2. Die Software muss nicht ... ..	16
2.6. Namensfindung .....	17
2.7. Verworfen Funktionen .....	17
<b>3. Datenquellen .....</b>	<b>19</b>
3.1. Verwendbare Datenquellen .....	19
3.2. Anbindung an Datenquellen .....	19
3.3. SchüSta .....	19
3.3.1. Datenspeicher .....	20
3.3.2. Ursprung der Daten .....	21
3.3.3. Exportoptionen .....	22
3.3.4. Export der Daten .....	22
3.3.5. Inhalt des Exports .....	24
3.3.6. Umstieg auf Sokrates .....	24
3.4. Untis .....	24
3.4.1. Datenspeicher .....	25
3.4.2. Datenbestand .....	25
3.4.3. Ursprung der Daten .....	26
3.4.4. Exportoptionen .....	26
3.4.5. Export der Daten .....	27
3.4.6. Inhalte des Exports .....	28
3.5. WebUntis .....	30
3.5.1. Inhalte des Exports .....	30
<b>4. Softwaredesign .....</b>	<b>31</b>
4.1. Aufbau .....	31

4.2.	„Model-View-Controller“ .....	31
4.2.1.	Modell .....	31
4.2.2.	Präsentation .....	31
4.2.3.	Steuerung .....	32
4.3.	Ordnerstruktur .....	32
4.4.	Namenskonventionen .....	33
4.5.	Datenbank .....	34
<b>5.</b>	<b>Bedienung .....</b>	<b>37</b>
5.1.	Semesterverwaltung .....	37
5.2.	Import von Daten .....	37
5.3.	Schulfreie Tage .....	40
5.4.	Verwalten der Anmeldung .....	40
5.5.	Anmeldestatistik .....	44
5.6.	Erkennung von Betreuungsstunden .....	45
5.7.	Festlegung der Kosten .....	45
5.8.	Anwesenheitsliste .....	46
5.9.	Export für die Buchhaltung .....	48
5.10.	Probleme bei der Verarbeitung der Daten .....	48
<b>6.</b>	<b>Beiliegender Datenträger .....</b>	<b>51</b>
6.1.	Inhalt des Datenträgers .....	51
6.2.	Installation .....	51
6.3.	Einspielen der Beispieldaten .....	52
6.4.	Zugangsdaten .....	52
<b>7.</b>	<b>Zusammenfassung .....</b>	<b>53</b>
7.1.	Wirkung der Software .....	53
7.2.	Zukünftige Erweiterungen und Anpassungen .....	53
7.3.	Persönliche Meinung .....	53
<b>8.</b>	<b>Darstellungsverzeichnis .....</b>	<b>55</b>
8.1.	Abbildungen .....	55
8.2.	Diagramme .....	55
<b>9.</b>	<b>Literaturverzeichnis .....</b>	<b>57</b>
<b>10.</b>	<b>Anhang .....</b>	<b>59</b>
10.1.	Abstract .....	59
10.2.	Lebenslauf .....	60

## 1. Einleitung

Die Beaufsichtigung von Schülerinnen und Schülern ist eine verantwortungsvolle Aufgabe. Im Gegensatz zu den über die Jahre relativ stabilen Klassenkonstellationen für den regulären Unterricht hat die nur von einigen Erziehungsberechtigten freiwillig gewählte Tagesbetreuung an bestimmten Tagen ein evidentes Problem: Die SchülerInnen werden in scheinbar willkürlicher und variierender Gruppierung von oft wechselnden Lehrkräften beaufsichtigt.

Diese Arbeit beschäftigt sich mit der Entwicklung einer Software zur Vereinfachung der Planung und Leitung der Tagesbetreuung im „Wienerwaldgymnasium“ in Tullnerbach (offiziell „Expositur des BG/BRG Purkersdorf“).

Allein durch das vielfältige Angebot an frei wählbaren Gegenständen ist der Nachmittagsunterricht vieler Schüler dieser Schule über alle Nachmittage der Schulwoche verteilt. Durch den Schulstandort mitten im Wienerwald und die damit verbundene beschränkte öffentliche Anbindung wird die Tagesbetreuung intensiv beansprucht. Um einen Überblick über die TagesbetreuungsschülerInnen zu bewahren, bedarf es mehr als die Organisation per einfacher Listen.





## **2. Grundlagen**

### **2.1. Rechtliche Grundlagen**

Das Schulorganisationsgesetz (SchOG, 2015) bietet mit § 8d öffentlichen Schulen die Möglichkeit, ganztägige Schulformen zu führen. Im Schulzeitgesetz wird mit § 8 Punkt j definiert, dass ganztägige Schulformen aus einer Lernzeit (gegenstandsbezogen oder individuell) sowie Freizeit bestehen. Im Schulunterrichtsgesetz (SchUG, 2015) wird in § 12a wiederum festgelegt, dass der Besuch des Betreuungsteiles einer ganztägig geführten Schulform einer Anmeldung bedarf, welche für das ganze Schuljahr Geltung besitzt. Schüler sind laut SchUG § 43 Abs. 1 verpflichtet, die Tagesbetreuung, zu welcher sie angemeldet sind, pünktlich zu besuchen. Des Weiteren schreibt das Schulzeitgesetz (Schulzeitgesetz, 2015) in § 5 Abs. 6 vor, dass die Tagesbetreuung bis mindestens 16:00 Uhr und bis 18:00 angeboten wird.

Aus diesen rechtlichen Grundlagen ergeben sich gut überschaubare Randbedingungen:

- SchülerInnen müssen zur Tagesbetreuung angemeldet werden
- Angemeldete SchülerInnen müssen verpflichtend anwesend sein
- Anwesende SchülerInnen müssen beaufsichtigt werden

Somit kann nach der in der ersten Schulwoche des Schuljahres stattfindenden Anmeldephase eine fest definierte SchülerInnenmenge in der Tagesbetreuung erwartet werden.

### **2.2. Begriffsdefinition**

Die Betreuung außerhalb der Regulärunterrichtszeit wird schulintern sowie in der Kommunikation mit Eltern mit zwei verschiedenen Begriffen bezeichnet. Einerseits gibt es die „Mittagsbetreuung“, welche auch von Seiten des Bundesministeriums so benannt wird (Mittagsbetreuung, 2014) und die Betreuung von SchülerInnen zwischen dem Vormittags- und Nachmittagsunterricht bezeichnet, andererseits gibt es die „Nachmittagsbetreuung“, welche eine Betreuung nach der letzten Unterrichtsstunde benennt. Der offizielle Begriff für die „Nachmittagsbetreuung“ ist je nach Art der Umsetzung ebenso „Nachmittagsbetreuung“ oder „Tagesbetreuung in verschränkter Form“ (Die ganztägige Schule, 2015).

Die unterschiedliche Bezeichnung der Betreuungszeiträume findet seinen Grund darin, dass die Berechnung der in Anspruch genommenen Betreuungszeit unterschiedlich gehandhabt wird.

#### **2.2.1. Mittagsbetreuung**

Die Mittagsbetreuung ist jener Teil der Tagesbetreuung, welcher den unterrichtsfreien Zeitraum zwischen zwei Unterrichtseinheiten überbrückt. Die frühestmögliche Einheit, in welcher die Mittagsbetreuung stattfinden darf, ist die 5. Stunde, welche in den meisten

Schulen um ca. 12:00 beginnt. Die Mittagsbetreuung kann bis maximal zwei Stunden an maximal zwei Tagen pro Woche in Anspruch genommen werden. Die Kosten für diesen Überbrückungszeitraum übernimmt das Bundesministerium. Daher ist diese Betreuungszeit für die Eltern kostenlos.

Üblicherweise wird die Mittagsbetreuung für Mittagessen jeglicher Art und zur Entspannung genutzt. Es findet eine reine Beaufsichtigung statt.

### **2.2.2. Nachmittagsbetreuung**

Die Nachmittagsbetreuung ist jener Teil der Tagesbetreuung, welcher nach der letzten Unterrichtsstunde stattfindet. Diese Betreuung muss laut Bundesministerium bis mindestens 16.00 Uhr aber bis maximal 18:00 Uhr angeboten werden (siehe 2.1). Wurde eine/ein Schülerin/Schüler zur Nachmittagsbetreuung angemeldet, so kann sie/er nicht mehr zur Mittagsbetreuung desselben Tages angemeldet werden. Die Nachmittagsbetreuung umfasst daher auch eine eventuelle Mittagsbetreuung.

Die Nachmittagsbetreuung ist kostenpflichtig. Die entstehenden Kosten berechnen sich pauschal anhand der in Anspruch genommenen Anzahl an Tagen der Betreuung pro Woche, jedoch unabhängig von der Anzahl der Stunden an diesen Tagen. Da das Bundesministerium einen Großteil der Kosten übernimmt, entstehen für die Eltern nur relativ geringe Unkosten in Form eines Betreuungsbeitrages, der üblicherweise maximal 88 Euro pro Monat bei wöchentlich fünftägiger Beanspruchung beträgt<sup>1</sup>.

Zu Beginn der Nachmittagsbetreuung wird meist Mittagessen konsumiert. Danach findet eine sogenannte „Lernstunde“ im Umfang einer Unterrichtseinheit für alle betreuten SchülerInnen statt. Diese Zeit muss zum Erledigen von allfälligen Hausaufgaben, Lernen für Prüfungen und Schularbeiten oder zur Revision genutzt werden. Damit diese Zeit sinnvoll betreut werden kann, sind zur Betreuung fachlich kompetente Lehrer eingeteilt.

## **2.3. Rahmenbedingungen für die Entwicklung der Software**

Nicht nur die Entwicklung einer Software erfordert viel Zeit und damit auch Kosten, sondern auch die sich im laufenden Betrieb nötigen Nachbesserungen bzw. Anpassungen, die Wartung und der Betrieb. Zu üblichen laufenden Ausgaben zählen unter anderem die Einrichtung der Software auf Arbeitsplätzen und Servern (sofern diese bereits vorhanden sind), die Behebung von Fehlern, die Erweiterung um notwendige Funktionen aber auch der Erwerb der zum Betrieb erforderlichen Lizenzen von Drittanbietern.

Aufgrund fehlender Gelder für die Finanzierung der Entwicklung sowie die nötige laufende Wartung der Software wurden folgende Rahmenbedingungen definiert:

- Lizenzfreie Entwicklungssoftware
- Lizenzfreie Frameworks

---

<sup>1</sup> Je nach Inanspruchnahme reduziert sich der Betrag auf 20%/40%/60%/80% bei 1/2/3/4 Tagen. In einem Schuljahr muss der Beitrag für 10 Monate gezahlt werden. Details zur Gestaltung des Betreuungsbeitrages kann dem aktuell gültigen Gesetzestext entnommen werden:

<https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10009945>

- Lizenzfreie Grafiken
- Minimaler Aufwand für Einschulung
- Keine Clientkonfiguration
- Einfache Backupmöglichkeit
- Gängige Software für Betrieb und Entwicklung

## **2.4. Umsetzung der Software**

Um den Rahmenbedingungen aus 2.3 Folge zu leisten, wurde die Umsetzung wie folgt beschlossen.

### **2.4.1. Art der Umsetzung**

Die Umsetzung erfolgt als Webanwendung. Dadurch, sowie durch den Einsatz möglichst kompatibler Standards in der Webentwicklung, wird der Forderung nach dem Entfall der Konfiguration der Clients nachgekommen. Eine zentralisierte Umsetzung vereinfacht auch wesentlich das Backup, da dieses nur an einer Installation durchgeführt werden muss. Für den Betrieb wird serverseitig ein Webserver und eine Datenbank benötigt. Clientseitig ist lediglich ein Browser notwendig, welcher mit jedem Betriebssystem geliefert wird bzw. aus einer Vielzahl an freien Alternativen gewählt werden kann.

### **2.4.2. Verwendete Technologien**

Auf Basis der Art der Umsetzung sowie der Rahmenbedingungen lassen sich viele Techniken zur Werkstellung wählen. Durch persönliche Vorlieben werden folgende Technologien eingesetzt.

#### **Software für Entwicklung**

PHP (PHP Group) – Open Source<sup>2</sup> Skriptsprache zur Erstellung dynamischer Webinhalte

CakePHP (Cake Software Foundation, Inc.) – Open Source PHP Framework zur einfachen Erstellung von Webanwendungen

#### **Umgebung für den Betrieb**

Debian (Debian Project) – Open Source Betriebssystem auf Basis von Linux

MySQL (Oracle) – Open Source Datenbank

Apache HTTP Server (The Apache Software Foundation, 1995) – Open Source Webserver

### **2.4.3. Alternative Umgebungsbedingungen**

Der Betrieb ist auf einer Vielzahl an Betriebssystemen möglich da die verwendeten grundlegenden Technologien PHP, MySQL und auch der Apache HTTP Server für alle gängigen Betriebssysteme frei verfügbar sind.

Die gewählte Betriebsumgebung ist für den erfolgreichen Einsatz daher nicht zwingend Voraussetzung. Einzig die Voraussetzungen und Empfehlungen des verwendeten

---

<sup>2</sup> Der Quelltext der Software steht frei zur Verfügung. Siehe auch: [http://de.wikipedia.org/wiki/Open\\_Source](http://de.wikipedia.org/wiki/Open_Source)

Frameworks sind zu beachten. Für die momentan aktuelle Version von CakePHP wird Apache als Webserver (mit mod\_rewrite) und MySQL, PostgreSQL, Microsoft SQL Server oder SQLite als Datenbank Server empfohlen (Installation, 2012). Auf anderen Systemen können Probleme oder zumindest ein Mehraufwand bei der Inbetriebnahme bzw. Entwicklung entstehen.

Wie bei allen Softwareprodukten muss damit gerechnet werden, dass mit zukünftigen Versionen eingesetzter Software und Entwicklungsumgebungen, Anpassungen an der entwickelten Software durchzuführen sind, um diese kompatibel zu halten.

## **2.5. Umfang und Abgrenzung der Software**

Bei der Entwicklung der Software müssen bereits vor dem Start klare Grenzen und Spezifikationen definiert und gesetzt werden. Ohne Grenzen droht die Gefahr, die Entwicklung in speziellen Bereichen extrem voranzutreiben und so die wichtigen, meist einfach zu erreichenden Ziele aus den Augen zu verlieren.

Gerade im Schulbereich sind Abgrenzungen für ein Softwareprodukt wichtig. Das mit der Software zum Arbeiten angehaltene Lehrpersonal ist nicht allein mit den durch die Applikation erreichten Erleichterungen wunschlos zufrieden, sondern stellt mitunter viele persönliche, unausgesprochene Erwartungen an die Software. Um dem Phänomen der Enttäuschung aus dem Weg zu gehen, wurde das Ziel der Software mit den betroffenen Personen ausführlich besprochen. Die in den Gesprächen zutage gekommenen Erwartungen wurden gemeinsam evaluiert, niedergeschrieben und in die weitere Planung integriert.

Diese Art der Vorgehensweise half insbesondere dabei, nach Fertigstellung der Entwicklung nicht auf Enttäuschung und Abneigung gegenüber der Software zu treffen.

Die folgenden Erwartungen wurden in zahlreichen Gesprächen und Beobachtungen festgelegt.

### **2.5.1. Die Software soll ...**

- ... die Verwaltungen der Anmeldungen zur Tagesbetreuung ermöglichen.
- ... druckbare Listen aller Anmeldungen an einem definierten Tag generieren.
- ... vorrangig im Wienerwaldgymnasium funktionieren aber eine Erweiterung bzw. Anpassung auf andere Umgebungsbedingungen ermöglichen.
- ... Daten für die Weiterverarbeitung in der Buchhaltung exportieren können.
- ... Schülerstammdaten aus der Software „SchüSta“ (Gmasz, 2012) importieren können (siehe 3) sowie für eine zukünftige Benützung von „Sokrates“ (bit Group GmbH, 2015) adaptierbar sein.
- ... Stammdaten sowie Stundenpläne aus der Software „Untis“ (Gruber & Petters GmbH, 2012) importieren können (siehe 3).

### **2.5.2. Die Software muss nicht ...**

- ... Anwesenheiten von Schülern online/live erfassen können.

- ... automatisch E-Mails an betroffene Personen versenden können.
- ... automatisch Daten aus anderen Verwaltungsprogrammen importieren.
- ... Vertretungslisten aus „Untis“ importieren können.

## **2.6. Namensfindung**

Ein klingender Name ist ein essentieller Bestandteil einer erfolgreichen Software. Gesucht wurde ein ansprechender, kurzer Name, der gut aussprechbar ist und auch in Sätzen verwendet werden kann. Natürlich durften keine Konflikte mit Markennamen auftreten. Der Wiedererkennungswert soll nach Möglichkeit hoch sein und die Assoziation zur Funktion soll möglichst aus dem Namen abgeleitet werden können.

Der gewählte Name „NabeTool“ setzt sich einerseits aus der im Schulbereich gängigen Abkürzung „NABE“ bzw. „Nabe“ für „Nachmittagsbetreuung“ und dem im Allgemeinen bekannten englischen Wort „Tool“ für „Werkzeug“ zusammen. Wenn auch nicht elegant, erfüllt der Name alle beschriebenen Voraussetzungen.

## **2.7. Verworfenne Funktionen**

Ursprünglich wurde eine Möglichkeit zur Zuteilung von Betreuungslehrern zu Schülergruppen sowie eine Anwesenheitsliste pro Lehrer überlegt. Grundsätzlich wurde die Idee von den BetreuungslehrerInnen gut geheißen, nach einer Beobachtungsphase aber wieder verworfen, da Schülergruppen in der Praxis nicht abgrenzbar sind. Üblicherweise teilen sich die betreuenden LehrerInnen zwischen diversen Örtlichkeiten auf. Diese Gruppen sind auch je nach Wetter unterschiedlich. Ist ein/eine LehrerIn mit einer Gruppe in der einen Woche bei Schlechtwetter in der Bibliothek, ist er/sie eventuell mit einer anders zusammengesetzten Schülergruppe bei schönem Wetter am Sportplatz.

Ebenfalls verworfen wurde die automatische Erkennung möglicher BetreuungslehrerInnen um diese manuell im NabeTool zuteilen zu können. Da die Lehrer komplett in „Untis“ verwaltet werden und dort bereits durch gesperrte Tage (dürfen nicht verplant werden) sowie weitere Randbedingungen wie Maximalbeschäftigung, Überschneidungen mit regulärem Unterricht und Mitverwendungen an anderen Schulen automatisch zugeordnet werden, wäre eine automatische Erkennung im NabeTool nur ein Nachbau vorhandener Funktionalität. Durch die Verwaltung in „Untis“ sehen LehrerInnen außerdem bereits in „WebUntis“ (Gruber & Petters GmbH, 2012), der Weboberfläche von „Untis“, ihre Betreuungsstunden.



### **3. Datenquellen**

#### **3.1. Verwendbare Datenquellen**

Die Planung der Tagesbetreuung basiert auf einem umfangreichen, bereits existierenden Pool an Daten. Diese setzen sich unter anderem aus den Stammdaten der SchülerInnen, welche in der Software „SchüSta“ verwaltet werden sowie Daten rund um den Stundenplan, die in „Untis“ verwaltet werden, zusammen. Des Weiteren werden Daten aus dem digitalen Klassenbuch „WebUntis“ importiert.

Diese bereits vorhandenen Daten existieren in Schulen normalerweise in digitaler Form. Eine Anbindung an diese Daten ist zwingend erforderlich um ein erneutes manuelles Eingeben und späteres ständiges manuelles Aktualisieren überflüssig zu machen.

#### **3.2. Anbindung an Datenquellen**

Der Transfer der Daten von externen Datenquellen in die Software soll möglichst einfach sein. Daher ist eine direkte Anbindung an die Daten wünschenswert. Leider empfiehlt der Entwickler von „SchüSta“ keine direkte Verwendung der Daten der zugrundeliegenden Datenbank sondern verweist stattdessen auf vorhandene Exportfunktionen (siehe 3.3.3). Eine direkte Verwendung der Daten aus „Untis“ wäre prinzipiell möglich, steht jedoch aufgrund der hervorragenden bereits verfügbaren Exportoptionen (siehe 3.4.4) nicht zur Diskussion. „WebUntis“ bietet zur direkten Anbindung an Daten eine eigene Schnittstelle. Diese unterstützt jedoch aktuell nur die Abfrage weniger Informationen und wird daher nicht verwendet. Die Programme werden in den folgenden Abschnitten genauer behandelt.

#### **3.3. SchüSta**

Das Programm „SchüSta“ (Gmasz, 2012) ermöglicht die Verwaltung von SchülerInnen, Klassen, LehrerInnen, Unterrichtsfächern und Noten einer Schule. Der Name „SchüSta“ ist eine Abkürzung und steht für „SCHÜlerSTAMmdaten“. Zu den in dieser „Standardsoftware“ im AHS-Bereich in Österreich verfügbaren Oberfläche gehören Masken für die Erfassung von Neuzugängen, die Bildung von neuen und die Verwaltung von bestehenden Klassen, die Planung von Abschlussprüfungen, die Ausgabe von Daten direkt an andere lokal installierte Programme oder den Drucker. Ebenso verfügbar sind Module für den Export von Daten.

Die Software spielt neben der Verwaltung der SchülerInnen auch eine große Rolle im Druck von Dokumenten. Sowohl Bescheide wie jene zur Benachrichtigung der Eltern über einen Leistungsabfall als auch Zeugnisse können mit „SchüSta“ gedruckt werden.

„SchüSta“ wird in Schulen hauptsächlich von der Administration und dem Sekretariat verwendet. Trotzdem steht es, je nach Art der Installation (siehe 3.3.1) Lehrkräften zum Nachschlagen oder Dokumentendruck zur Verfügung.

Mittlerweile ist ein Nachfolger von „SchüSta“ in Entwicklung und teilweise schon in Verwendung. Die neue Software „Sokrates“ löst „SchüSta“ in allen Funktionen ab. Diese Arbeit beschäftigt sich aufgrund der noch nicht fertigen Entwicklung nicht im Detail mit „Sokrates“. Das NabeTool unterstützt bereits „Sokrates“ beim Import von Daten. Spätere Abbildungen zeigen daher nicht ausschließlich auf „SchüSta“ ausgelegte Benutzerschnittstellen.

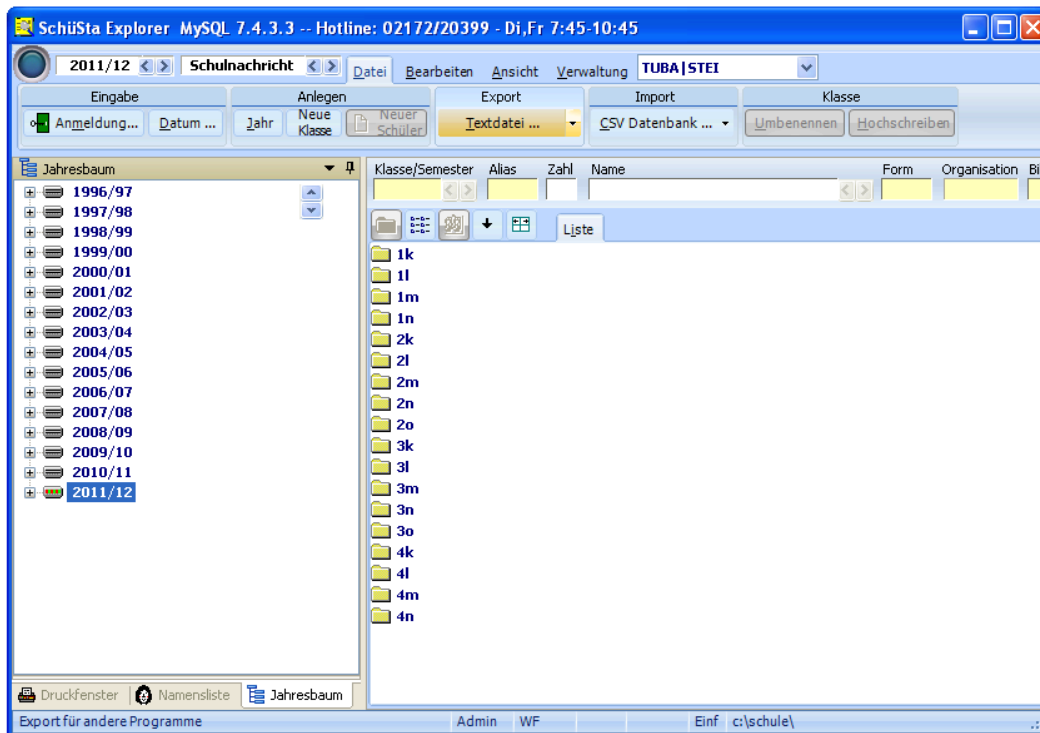


Abbildung 1: SchüSta. Überblick über Schuljahre und Klassen

### 3.3.1. Datenspeicher

Alle Daten werden mit einer der vielen von „SchüSta“ unterstützten Datenspeicher abgelegt. Aus Kostengründen wird häufig die Speicherung der Daten im Dateisystem gewählt. Für Mehrbenutzersysteme bietet der Hersteller die Verwendung einer Datenbank wie zum Beispiel MySQL an, welche als Erweiterungsmodul kostenpflichtig zu beziehen ist.

Der gewählte Datenspeicher hängt häufig von der Struktur der Verwaltung innerhalb der Schule ab. Die Speicherung der Daten im Dateisystem kann nur dann verwendet werden, wenn die Daten auf einem einzelnen Computer verwaltet und bearbeitet werden. Zwar gibt es Möglichkeiten zum Export einzelner Teilbereiche zur parallelen Bearbeitung auf anderen Rechnern und dem Wiederimport auf dem Hauptrechner, diese Option stößt aber in großen Schulen oder Schulen mit mehreren Standorten schnell an ihre Grenzen und wird demnach selten benutzt. In vielen Fällen wird oft auf die viel flexiblere Anbindung an eine Datenbank gesetzt.



### 3.3.2. Ursprung der Daten

Alle Daten, welche in „SchüSta“ vorhanden sind, wurden durch die Administration, das Sekretariat oder den Klassenvorstand der jeweiligen SchülerInnen einmalig manuell eingegeben oder, wenn bereits digital vorhanden, importiert.

Importiert werden können unter anderem Gegenstände und das Lehrerkollegium aus der Software „Untis“ (siehe 3.4), sofern diese Daten dort bereits angelegt wurden.

Neue SchülerInnen werden im Zuge der formalen Anmeldung im Sekretariat oder in der Administration vor Anfang des aktuellen Schuljahres in „SchüSta“ erfasst. Die Erfassung übernimmt meistens das Sekretariat. Die zu dokumentierenden Daten bestehen unter anderem aus Namen, Anschrift, Geschlecht, Geburtstag, Sozialversicherungsnummer, Staatsbürgerschaft, Erziehungsberechtigte und deren Kontaktmöglichkeiten sowie vielen anderen zum Teil schulspezifischen Informationen wie „Wunschklasse“, Schwerpunktwahl „BE/ME“ (Bildnerische Erziehung/Musikerziehung) und anderen (siehe Abbildung 2).

Nach Abschluss der Erfassung neuer SchülerInnen werden durch die Administration aus den neuen sowie aus den SchülerInnen der Schulstufen des Vorjahres – sofern es sich nicht um eine erste Schulstufe handelt – Klassen gebildet. Die Klassenbildung erfolgt neben den rechtlichen Rahmenbedingungen meist in Abstimmung mit SchülerInnen- bzw. Elternwünschen.

The screenshot displays the 'SchüSta Anmeldung' software window. The interface includes a menu bar with options like 'Editieren', 'Sicherheit', 'Verwaltung', 'Schüler', 'Modul starten', 'Export', 'Hilfe', 'Schülerliste', and 'Referenzlisten'. Below the menu is a toolbar with various icons. The main area is divided into several sections: 'Schülerstammdaten' (Student Data) on the left, 'Schulautonome Stammdaten' (School Autonomy Data) on the right, and 'Aufnahme' (Admission) at the bottom. The 'Schülerstammdaten' section contains fields for personal information such as name, gender, birth date, and social security number. The 'Schulautonome Stammdaten' section includes fields for school-related information like email, phone, and subject preferences. The 'Aufnahme' section at the bottom has fields for admission date, application number, and a table for 'Wohnort' (Residence) and 'Geschwister' (Siblings). The status 'Aufgenommen ja' (Admitted yes) is visible at the bottom of the 'Aufnahme' section.

Abbildung 2: Anmeldemaske in SchüSta für neue SchülerInnen

### 3.3.3. Exportoptionen

Ist die Datenquelle für „SchüSta“ eine Datenbank, ist ein direktes Auslesen der Stammdaten aus dieser theoretisch möglich, da die Betreuung der Datenbank und deren Zugriffsberechtigungen in der Hand der Schule liegt. Der Herausgeber der Software rät jedoch davon ab, da die Datenbank immer wieder ohne vorherige Ankündigung in ihrer Struktur verändert wird. Aus diesem Grund wird ein Datenexport über die in „SchüSta“ zur Verfügung stehenden Exportmöglichkeiten empfohlen.

Die Exportmöglichkeiten umfassen unter anderem (Auszug):

- BilDokmeldung (BilDok, 2015)
- WebUntis
- Textdatei

### 3.3.4. Export der Daten

Als maschinell einfach zu verarbeitendes Datenformat bietet sich der Export im Format „CSV“ – „comma separated values format“ - (Shafranovich, 2005) an, welcher in „SchüSta“ unter dem Punkt „Textdatei“ im Menü „Export“ aufgerufen werden kann.

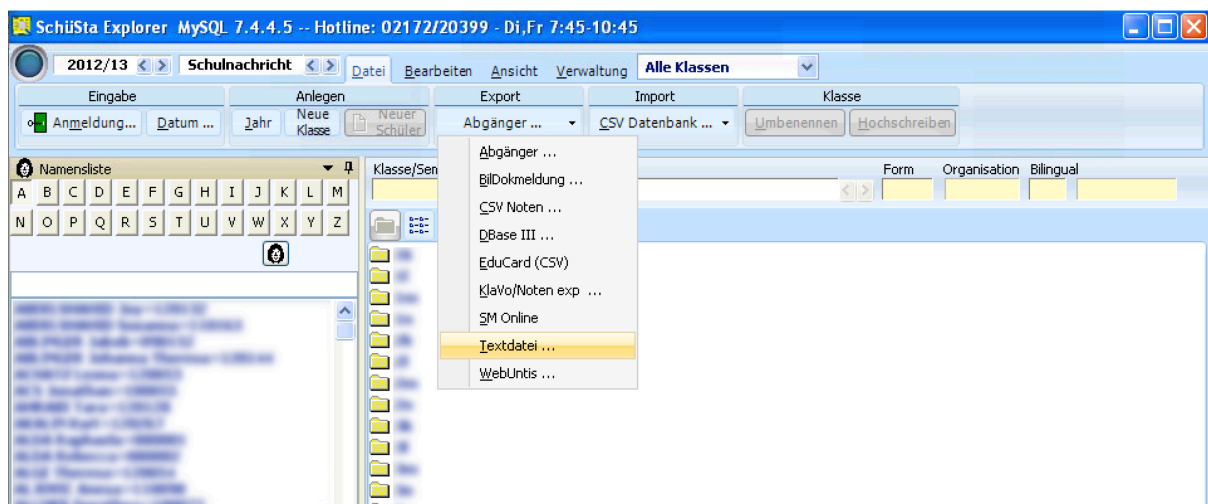


Abbildung 3: SchüSta Export Menü mit gewählter Exportoption „Textdatei“

In einer im Format „CSV“ strukturierten Textdatei werden Inhalte als Datensätze durch ein vorher zu definierendes Trennzeichen gespeichert. Einzelne Felder eines Datensatzes werden ebenfalls durch ein zuvor definiertes Zeichen getrennt. Die zwei verwendeten Zeichen müssen voneinander unterschieden sein. Jeder Datensatz sollte laut RFC 4180 (Shafranovich, 2005) die gleiche Anzahl an Feldern beinhalten. Die erste Zeile einer „CSV“ Datei kann Namen für die Felder der folgenden Datensätze beinhalten.

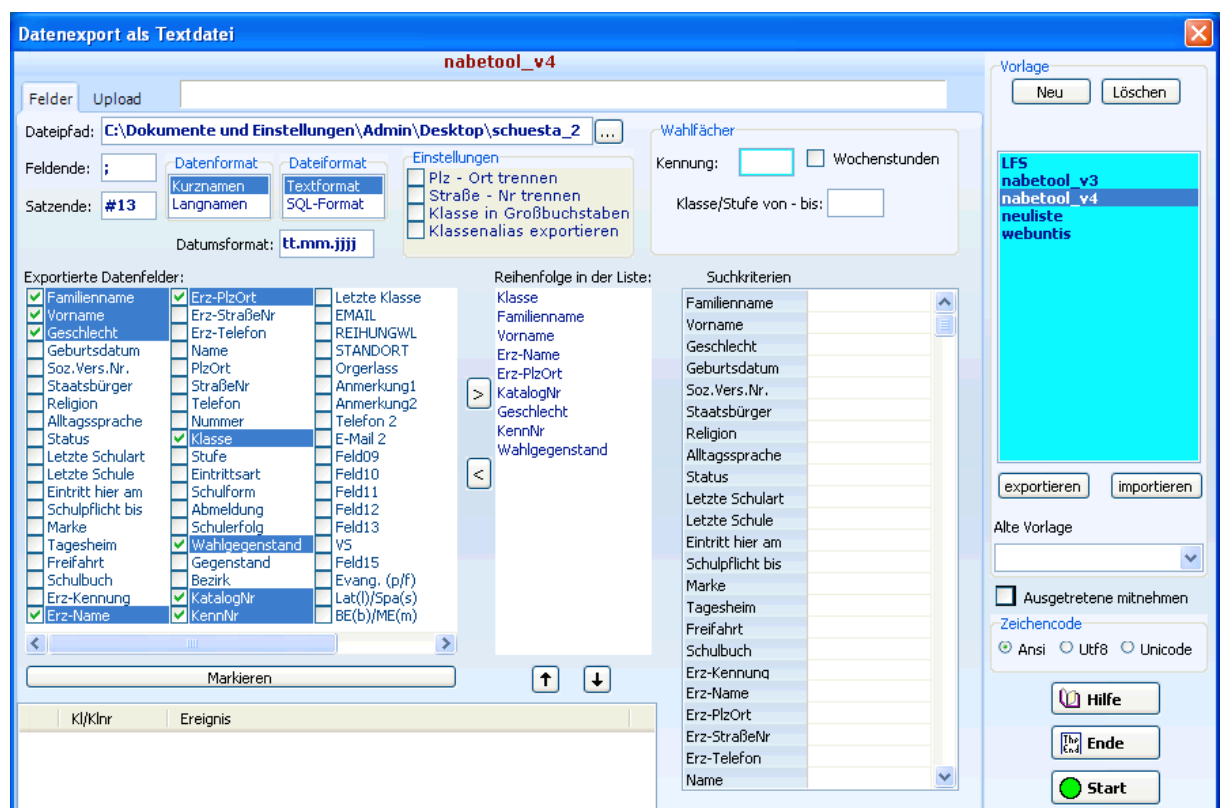
Trotz weiter Verbreitung wurde dieses Format formal nie definiert. Daher gibt es viele Umsetzungen. Üblich ist die Wahl der zwei Zeichen CR+LF („Wagenrücklauf“ gefolgt von „Zeilenumbruch“, üblicherweise im Jargon bekannt als „nächste Zeile“) als Trennzeichen für Datensätze und der Beistrich „“(Komma) als Trennzeichen für die einzelnen Felder eines Datensatzes.

„SchüSta“ unterstützt dieses Exportformat und bietet abgesehen von der Textkodierung auch an, die Trennzeichen den eigenen Bedürfnissen anzupassen (Felder „Feldende“ und „Satzende“). Mit dem Wissen über die Trennzeichen, die verwendeten Felder und den verwendeten Zeichensatz der Textdatei können die Daten maschinell problemlos zwischen Systemen transferiert werden.

Um Interpretationsprobleme beim Einlesen der aus „SchüSta“ exportierten Daten auf unterschiedlichen Betriebssystemen<sup>3</sup> zu umgehen, wird als Trennzeichen für Datensätze lediglich das Zeichen für „Wagenrücklauf“ verwendet, welches den dezimalen ASCII<sup>4</sup> Code 13 besitzt.

Als Trennzeichen zwischen einzelnen Feldern wird das Zeichen „Strichpunkt“ „;“ (Semikolon) gewählt, da Beistriche im Dateninhalt in einzelnen Feldern vorkommen, und auch hier Wert auf Kompatibilität gelegt wird.

Der Dialog zum Export von Daten in „CSV“ Dateien bietet eine Fülle an weiteren Optionen. Die wichtigsten und oft verwendeten sind die Wahl der zu exportierenden Datenfelder (siehe Abbildung 4 links mittig) sowie die Möglichkeit zum Speichern eingestellter Exportoptionen (siehe Abbildung 4 rechts außen). Letzteres speichert unter anderem die für den Export definierten Felder und Trennzeichen in einer wieder aufrufbaren Konfigurationsdatei. Das hilft dabei, die Struktur der exportierten Datei immer gleich zu halten, was eine Voraussetzung für den späteren Import ist.



<sup>3</sup> Unterschiedliche Betriebssysteme verwenden entweder nur CR oder LF oder aber auch eine Kombination aus den zwei Zeichen, um den Beginn einer neuen Zeile zu markieren. Siehe (Newline, 2003)

<sup>4</sup> American Standard Code for Information Interchange. Siehe (ASCII, 2001)

### 3.3.5. Inhalt des Exports

Die innerhalb des NabeTools benötigten und somit aus „SchüSta“ zu exportierenden Stammdaten der SchülerInnen setzen sich aus folgenden Feldern zusammen:

- Familienname
- Vorname
- Geschlecht
- Erz-Name (Name der Erziehungsberechtigten)
- Erz-PlzOrt (Postleitzahl der Erziehungsberechtigten)
- Klasse
- Wahlgegenstand (Französisch, Latein, ...)
- KatalogNr (Innerhalb des Klassenverbandes fortlaufende Nummer)
- KennNr (Innerhalb der Schule eindeutige fortlaufende Nummer)

Das wichtigste Feld ist das Feld „KennNr“. Dieses Feld beinhaltet die innerhalb einer Schule eindeutig vergebene SchülerInnen-Kennzahl. Eine/ein SchülerIn bekommt eine solche Kennzahl bei der Anmeldung an die jeweilige Schule zugewiesen. Da diese Kennzahlen nicht wiederverwendet werden, lassen sich durch sie SchülerInnen bei Verwaltungsaufgaben schnell und eindeutig identifizieren.

Da durch Klassenwechsel, Wiederholungen, Namensänderungen oder Neueintritte die Stammdaten von SchülerInnen bzw. die SchülerInnenkonstellationen der Klassen einer Schule nie über ein Jahr gleich bleiben, wird diese Kennzahl innerhalb des NabeTools benötigt, um bei Importen unterscheiden zu können, ob ein/eine SchülerIn neu angelegt oder ein/eine vorhandener/vorhandene SchülerIn aktualisiert werden soll.

### 3.3.6. Umstieg auf Sokrates

Der Nachfolger von „SchüSta“, „Sokrates“, ist bereits an einigen Schulen in Betrieb und wird in naher Zukunft an den meisten Schulen eingesetzt werden. Grundsätzlich ändert sich an den exportierbaren Daten inhaltlich wenig, lediglich das Format ändert sich von „CSV“ auf eine Struktur, die einzelne Datenfelder pro Zeile mit Tabulatoren trennt. Außerdem bietet „Sokrates“ weniger Wahlmöglichkeiten beim Export an und exportiert standardmäßig mehr Informationen.

Die am beigelegten Datenträger inkludierte Version der Software ist bereits für den Import von Daten aus „Sokrates“ vorbereitet.

## 3.4. Untis

Die Software „Untis“ (Gruber & Petters GmbH, 2012) bietet ein breites Spektrum an Funktionen zur Erstellung von Stundenplänen, Verwaltung von LehrerInnenstammdaten, Führung von Vertretungslisten sowie alle für die Abrechnung der Gehälter für das Lehr- und Betreuungspersonal notwendigen Funktionen. Die Software steht aufgrund einer vom

BMBF<sup>5</sup> 2001 gekauften Generallizenz allen mittleren und höheren Schulen zur Verfügung (UPIS, 2015).

Es gibt mehrere Module um die Basisfunktionalität zu erweitern. Zu diesen zählen unter anderem das Modul „Info-Stundenplan“ zur (auch automatischen) Ausgabe vertretungsrelevanter Daten auf Webseiten und das Modul „Pausenaufsichten“ zur Planung von Beaufsichtigung von SchülerInnen in Pausen.

„Untis“ wird fast ausschließlich von der Administration in Schulen verwendet. Abgesehen von einem eventuellen LehrerInnenteam, welches die Erstellung des Stundenplans übernommen hat, kommen Lehrkräfte nicht mit dieser Software in Berührung.

### 3.4.1. Datenspeicher

Alle Daten eines Schuljahres werden üblicherweise in einer einzelnen Datei im Dateisystem abgelegt. Um im Fehlerfall zu einer alten Version zurückkehren zu können wird empfohlen vor großen Änderungen bzw. am Tagesende eine Kopie der Datei abzulegen. Für große Schulen steht eine Mehrplatzversion zum Kauf zur Verfügung.

### 3.4.2. Datenbestand

„Untis“ trennt die Führung der Stammdaten vom „Tagesgeschäft der Administration“. Zu den Stammdaten zählen sowohl grundlegende Daten wie zum Beispiel der Beginn und das Ende eines Schuljahres, die Festlegung von Unterrichtszeiten aber auch die Definition von Unterrichtsfächern, LehrerInnen und Klassen. Die täglich anfallenden Daten hingegen umfassen Abwesenheiten von LehrerInnen, sich dadurch ergebende Änderungen des Stundenplans und gehaltene Vertretungen.

Name	Langname	Alias	Vorj.Name	Raum	Klassenlehrer	St.Tafel	Männl.	Weibl.	Studenten	Hauptf./Tag	Haupt
1K	1K			R.1K		1009	14	10	24	3	
1L	1L			R.1L		1009	14	10	24	3	
1MB	1M BE	1M		R.1M		1028	1	16	17	3	
1MM	1M ME	1M		R.1M		1017	3	7	10	3	
1N	1N			R.1N		1009	12	12	24	3	
2K	2K		1K	R.2K		1009	11	12	23	3	
2L	2L		1L	R.2L		1009	11	13	24	3	
2M	2M		1M	R.2M		1017	7	8	15	3	
2N	2N		1N	R.2N		1009	8	12	20	3	
3KG	3KG	3K	2K	R.3K		1102	4	4	8	3	
3KR	3KR	3K	2K	R.3K		1202	8	6	14		
3LG	3LG	3L	2L	R.3L		1102	2	5	7	3	
3LR	3LR	3L	2L	R.3L		1202	12	4	16		
3M	3M		2M	R.3M		1206	9	12	21	3	
3NG	3NG	3N	2N	R.3N		1102	5	6	11	3	
3NR	3NR	3N	2N	R.3N		1202	8	3	11	3	
3OG	3OG	3O	2O	R.3O		1102	6	6	12		

Abbildung 5: Stammdaten der Klassen in Untis

<sup>5</sup> Bundesministerium für Bildung und Frauen, bis 2014 Bundesministerium für Unterricht, Kunst und Kultur (BMUKK)

### **3.4.3. Ursprung der Daten**

Alle in „Untis“ zur Verfügung stehenden Daten wurden entweder per Hand eingegeben oder digital aus anderen Quellen importiert. Viele Daten wie die am Standort unterrichteten Fächer, die unterrichtenden Lehrkräfte, Angaben zu den Räumen oder stundenplanrelevante Eckdaten wie der Unterrichtsbeginn können aus dem Vorjahr übernommen werden. Die meisten Änderungen der Stammdaten finden am Schuljahresanfang oder bereits davor statt, wenn neue LehrerInnen aufgenommen und abgehende LehrerInnen ausgetragen und Schulklassen gebildet oder aufgelöst werden. Der Stundenplan wird ebenfalls vor oder knapp nach Schuljahresanfang erstellt und normalerweise während des laufenden Schuljahres nur minimal geändert (zB. bei Schwangerschaften, längeren Krankenständen, ...).

### **3.4.4. Exportoptionen**

„Untis“ bietet eine Fülle an Exportformaten an. Da die Software in einer Vielzahl an Ländern vertrieben wird und auch in vielen zur Abrechnung von Gehältern genutzt wird, hat der Hersteller demnach sehr viele landspezifische Exportformate, welche meist für eine spezielle maschinelle Weiterverarbeitung gedacht sind, integriert. Es gibt aber auch allgemeine Exportformate für den Export der meisten Stammdaten, welche ebenfalls für eine maschinelle Weiterverarbeitung geeignet sind.

Zu den allgemeinen Exportformaten zählt unter anderem der Export als Textdatei im Format „XML“ (W3C, XML Core Working Group, 2006). Dieses wird durch das Exportformat „XML-Import/Export“ in „Untis“ ermöglicht. Die Auszeichnungssprache „XML“ ermöglicht die Speicherung hierarchisch strukturierter Daten in Textdateien.

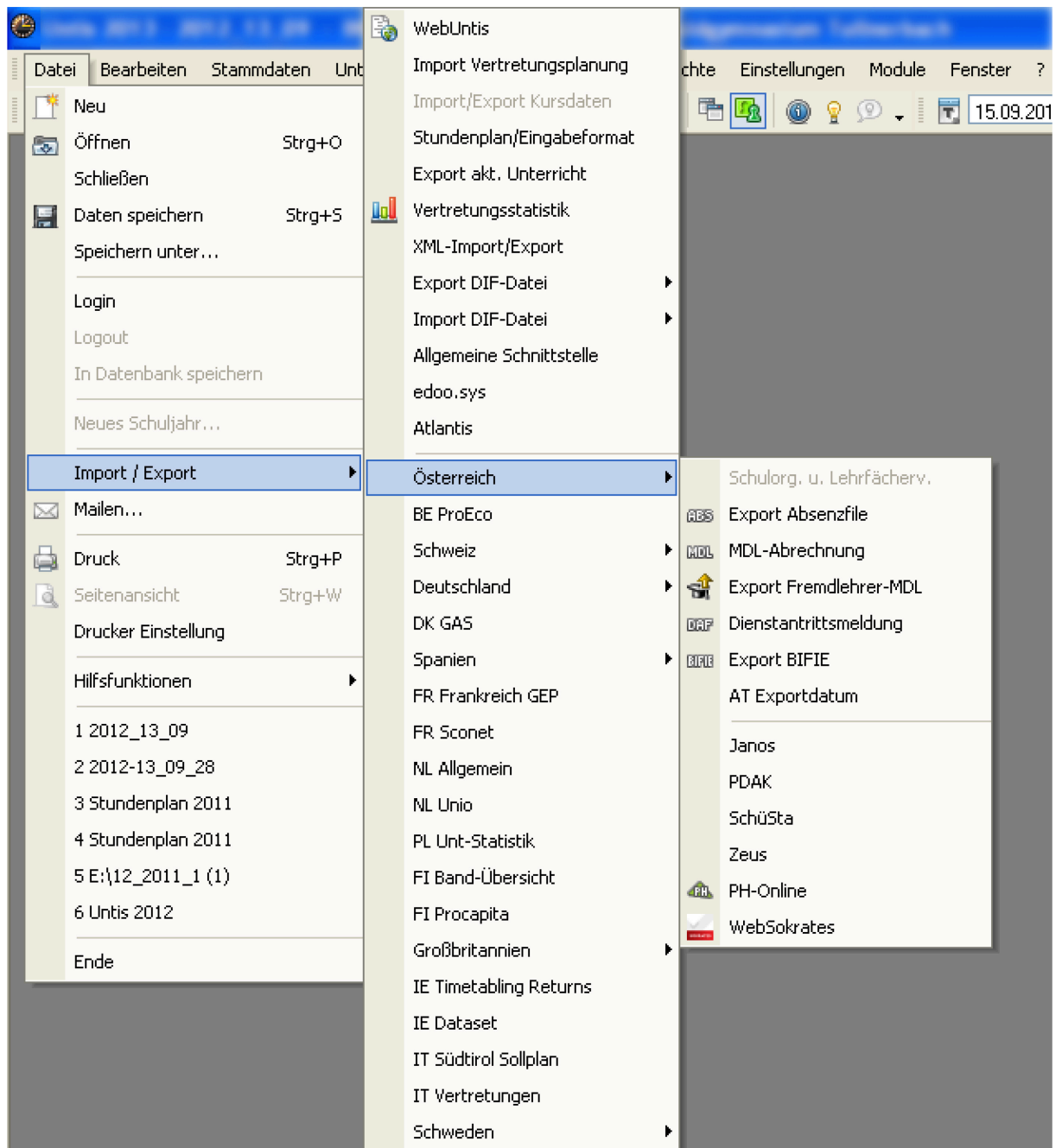


Abbildung 6: Untis Export Menü mit einer Vielzahl an Exportformaten

### 3.4.5. Export der Daten

„Untis“ bietet mit XML ein exzellentes Format als Exportmöglichkeit an. XML Daten sind strukturierte Daten und werden in Textdateien gespeichert. Diese Daten sind mit jedem Texteditor lesbar und können von einer Vielzahl an frei verfügbaren Parsern <sup>6</sup>

<sup>6</sup> Ein Parser ist ein Programm, das für die Umwandlung und Interpretation einer Eingabe in ein für die Weiterverarbeitung brauchbares Format zuständig ist.

weiterverarbeitet werden. Die Textdaten werden automatisch in UTF-8<sup>7</sup> gespeichert, um Probleme mit Sonderzeichen zu verhindern.

Im Dialog zum Export wird als einzige Option „Alias für Elemente verwenden“ angeboten. Der „Alias“ ist eine für viele Einträge frei definierbare Bezeichnung. Da diese Bezeichnung nicht eindeutig sein muss, aber in der exportierten XML Datei die eindeutige „ID“ ersetzt, führt das beim Import zu Problemen bzw. nicht erkannten Elementen. Diese Option darf daher nicht gewählt werden.

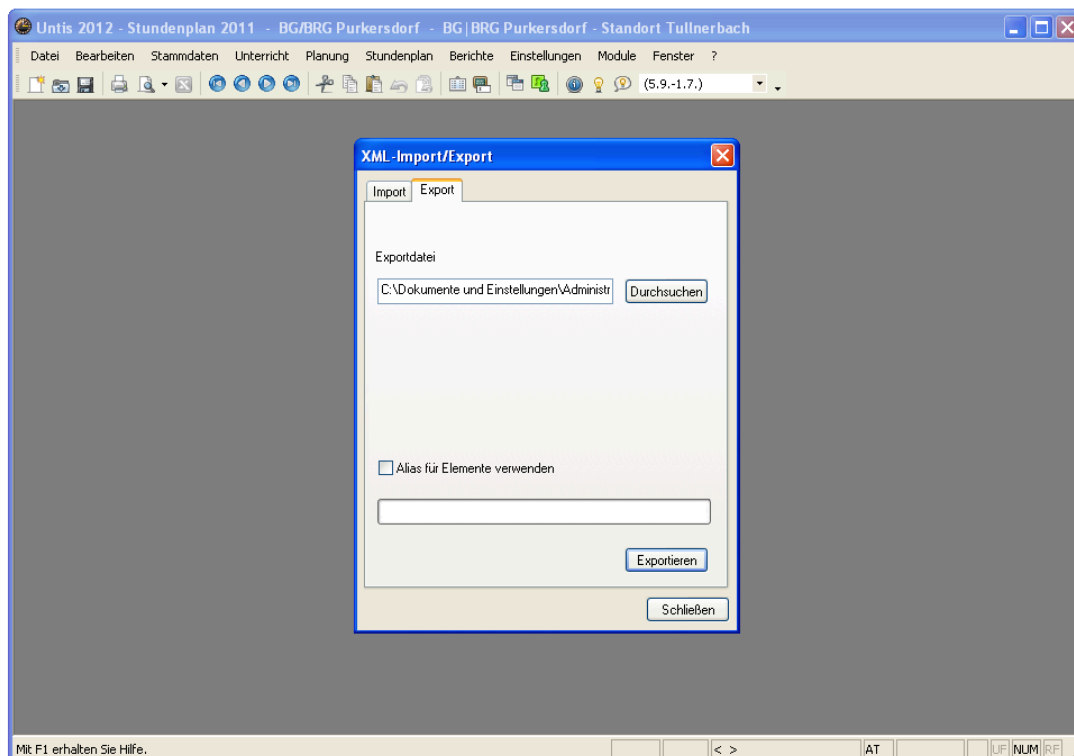


Abbildung 7: Untis XML Import/Export Dialog

### 3.4.6. Inhalte des Exports

Der Export aus „Untis“ enthält alle relevanten Daten zu Stundenplan und Personal. Ein (auf die Grundstruktur gekürzter) Export sieht wie folgt aus:

```
<?xml version="1.0" encoding="utf-8"?>
<document version="1.4" date="20140220" time="100136">
  <general> (...) </general>
  <timeperiods> (...) </timeperiods>
  <descriptions> (...) </descriptions>
  <departments> (...) </departments>
  <rooms> (...) </rooms>
  <subjects> (...) </subjects>
  <teachers> (...) </teachers>
  <classes> (...) </classes>
  <students/>
  <lessons> (...) </lessons>
  <reductions/>
```

<sup>7</sup> UTF-8 ist eine Kodierung für Textzeichen, die eine Vielzahl an Sonderzeichen unterstützt. Im Vergleich zu ASCII können zB. deutschsprachige Umlaute oder japanische Schriftzeichen abgebildet werden.



```

    <lesson_tables> (...) </lesson_tables>
</document>

```

An der Grundstruktur der exportierten XML Datei ist sofort erkennbar, welche Daten sie umfasst. Es sind sowohl allgemeine Schuldaten (<general>, <timeperiods>, <descriptions>) und Daten zur räumlichen Situation (<departments>, <rooms>) als auch alle stundenplanrelevanten Daten (<subjects>, <teachers>, <classes>, <lessons> ...) enthalten. Einige Bereiche wie zB. <students /> sind leer, da die SchülerInnenverwaltung in „SchüSta“ ausgelagert verwaltet wird.

Jeder Eintrag unterhalb der Hauptstruktur erhält eine eindeutige ID. Anhand dieser ID werden die Einträge referenziert und ergeben so eine logische Gesamtstruktur, den Stundenplan.

Beispiel: Das Fach D in der 1K, welches der Lehrer „ABCD“ unterrichtet, wird wie folgt definiert. Gut zu erkennen ist die strukturierte Auflistung der Abhaltungszeiten und -klassenräume:

```

(...)
<lesson id="LS_1900">
  <periods>4</periods>
  <lesson_subject id="SU_D"/>
  <lesson_teacher id="TR_ABCD"/>
  <lesson_classes id="CL_1K"/>
  <teacher_value>466800</teacher_value>
  <effectivebegindate>20130902</effectivebegindate>
  <effectiveenddate>20140629</effectiveenddate>
  <occurence>(...) </occurence>
  <times>
    <time>
      <assigned_day>2</assigned_day>
      <assigned_period>1</assigned_period>
      <assigned_starttime>0805</assigned_starttime>
      <assigned_endtime>0855</assigned_endtime>
      <assigned_room id="RM_R.1K"/>
    </time>
    <time>
      <assigned_day>3</assigned_day>
      <assigned_period>3</assigned_period>
      <assigned_starttime>1000</assigned_starttime>
      <assigned_endtime>1050</assigned_endtime>
      <assigned_room id="RM_R.1K"/>
    </time>
    <time>
      <assigned_day>4</assigned_day>
      <assigned_period>1</assigned_period>
      <assigned_starttime>0805</assigned_starttime>
      <assigned_endtime>0855</assigned_endtime>
      <assigned_room id="RM_R.1K"/>
    </time>
    <time>
      <assigned_day>5</assigned_day>
      <assigned_period>4</assigned_period>
      <assigned_starttime>1100</assigned_starttime>
      <assigned_endtime>1150</assigned_endtime>
      <assigned_room id="RM_R.1K"/>
    </time>
  </times>

```

</lesson>  
(...)

### 3.5. WebUntis

Die Software „WebUntis“ (Gruber & Petters GmbH, 2012) ist eine Onlineapplikation, welche SchülerInnen und LehrerInnen Zugriff auf den aktuellen Stundenplan sowie weitere stundenplanrelevante Daten bietet. Sie ist direkt mit „Untis“ in der Administration der Schule verbunden und spiegelt eine Auswahl an Informationen aus „Untis“ wider.

„WebUntis“ dient als Ersatz zum klassischen Klassenbuch und bietet viele Funktionen wie das Eintragen von Lehrstoff, das Erfassen von Abwesenheiten von SchülerInnen, das Planen von Tests und das Erstellen von SchülerInnengruppen.

Für die Nachmittagsbetreuung interessant sind jene Daten, die durch Gruppendefinitionen entstehen, da an vielen Unterrichtsgegenständen nicht immer alle SchülerInnen einer Klasse teilnehmen. Ein gutes Beispiel ist ein frei wählbarer Gegenstand am Nachmittag. Wählt ein/e SchülerIn diesen Gegenstand, hat er/sie einerseits automatisch das Recht (in den vom Gesetz vorgegebenen Grenzen, siehe 2.2.1) Mittagsbetreuung für die unterrichtsfreie Zeit zwischen Vormittags- und Nachmittagsunterricht in Anspruch zu nehmen, andererseits ergibt sich daraus, dass der/die SchülerIn in der Unterrichtszeit des gewählten Nachmittagsunterrichts nicht in der Nachmittagsbetreuung anwesend sein muss.

#### 3.5.1. Inhalte des Exports

Der Schülergruppen-Export aus „WebUntis“ ist eine reine Textdatei, welche die schülerspezifische Kennzahl, sowie die Bezeichnung der SchülerInnengruppe getrennt durch Tabulatoren auflistet. Alle anderen Informationen sind für die weitere Verarbeitung nicht relevant.

Die logische Verbindung zwischen Unterricht und SchülerInnengruppe wird dadurch hergestellt, dass im Export von „Untis“ bei jeder Unterrichtsdefinition als Zusatzinformation eine eventuelle SchülerInnengruppe angegeben wird.

Ein Auszug:

12345620120025	MITTEL Teresa	BSPM_3Ag3Cg3Cr	BSPM
12345620120025	MITTEL Teresa	RK_3Ag	RK
12345620120027	BURGER Johanna	BSPM_3Ag3Cg3Cr	BSPM
12345620120099	MANDL Katharina	BSPM_3Ag3Cg3Cr	BSPM
12345620120099	MANDL Katharina	RK_3Ag	RK
12345620120100	BAUER Mia	BSPM_3Ag3Cg3Cr	BSPM

## 4. Softwaredesign

Oberste Priorität beim Design der Software war der Erhalt einer gut überschaubaren Struktur. Dies wurde größtenteils durch den Einsatz des Frameworks CakePHP und das Umsetzen darin bereits vorgesehener Programmierkonventionen erreicht.

### 4.1. Aufbau

CakePHP hält sich an das gängige Architekturmuster „Model-View-Controller“ (MVC, siehe 4.2) bei der Softwarestrukturierung und bietet Entwicklern bereits in der Grundausstattung eine Vielzahl an vorhandenen Funktionen. Diese Funktionen umfassen unter anderem datenbankunabhängiges Auslesen von Datenbankinhalten, automatisches Abrufen von Inhalten logisch verknüpfter Tabellen und die Verwendung von Vorlagen zur Generierung von Webseiten. Damit diese Kernfunktionen verwendet werden können, muss einer vorgeschriebenen Konvention zur Namensgebung und Strukturierung der einzelnen Softwareteile gefolgt werden.

### 4.2. „Model-View-Controller“

Das Konzept „Model-View-Controller“ (Modell-Präsentation-Steuerung, MVC) ist mittlerweile beinahe ein Standard in der Software Entwicklung. Die Idee hinter dem Konzept ist, die Software in drei möglichst getrennte Bereiche aufzuteilen, von denen jeder einzelne ersetzt werden könnte, um die Software für andere Einsatzbereiche wiederverwendbar zu machen. Beispielhaft könnte das NabeTool durch Ersetzen einzelner Teile von einer Webanwendung zu einer Smartphone-Anwendung verändert werden.

#### 4.2.1. Modell

Das Modell beinhaltet bzw. verwaltet die Daten. Je nach Umsetzung ist es auch für die Kommunikation mit Datenquellen verantwortlich.

In CakePHP werden Modelle durch „Models“ implementiert. Durch sie wird auf Daten in der dahinterliegenden Datenbank zugegriffen. Änderungen werden ebenfalls über Modelle in die Datenbank eingepflegt.

#### 4.2.2. Präsentation

Die Präsentationsschicht erhält darzustellende Daten von der Steuerung und kennt das jeweilige Modell der empfangenen Daten. Sie nimmt Benutzereingaben entgegen, bearbeitet diese aber nicht selbst, sondern gibt sie an die Steuerung weiter.

In CakePHP findet die Präsentation mit Hilfe von „Views“ statt. Es können mehrere Arten der Präsentation in einer Software implementiert werden. So ist es möglich gleichzeitig eine Webansicht und eine Ausgabe als XML Datei für dieselbe Steuerung zu implementieren und je nach Aufruf die jeweilige Präsentation der Daten zu wählen.

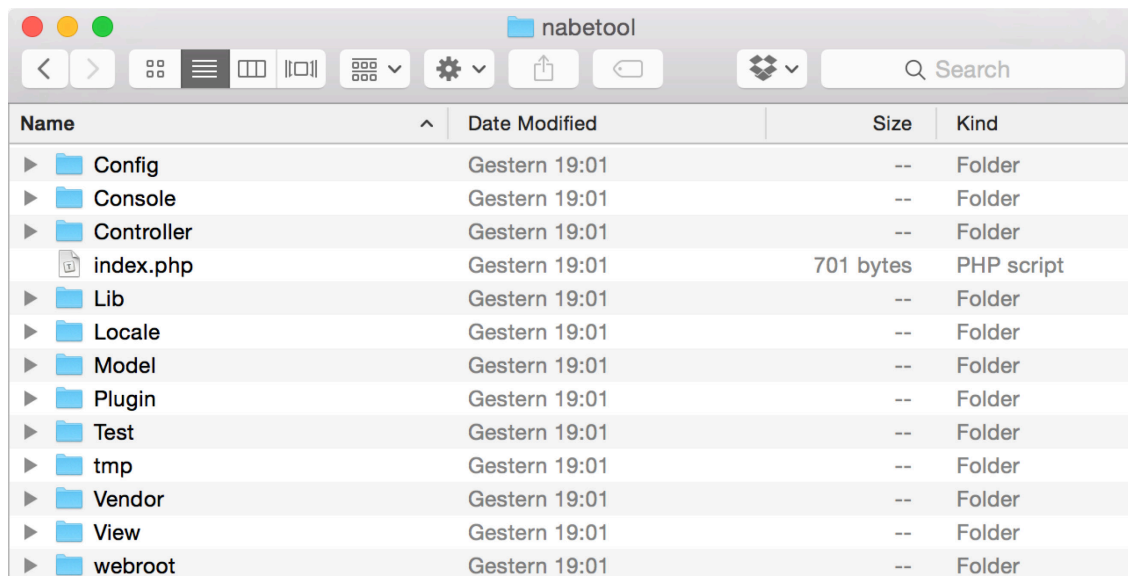
### 4.2.3. Steuerung

Die Steuerung verwaltet Präsentationen, reagiert auf Benutzereingaben und ist für den Datenfluss zwischen Präsentation und Modell verantwortlich. Außerdem stellt die Steuerung eine Sicherheitsebene dar und prüft Benutzereingaben auf ihre Rechtmäßigkeit.

In CakePHP befindet sich die Steuerung in „Controllern“. Die Prüfung von Benutzereingaben, zB. beim Anlegen neuer Einträge, wird zwar von der Steuerung angestoßen, jedoch findet die Kontrolle dort statt, wo die Daten ihre Struktur erhalten - in den Modellen.

### 4.3. Ordnerstruktur

Software wird in CakePHP auf mehrere Dateien in unterschiedliche, vordefinierte Bereiche geteilt. Diese Bereiche werden durch Ordner innerhalb einer Gesamtstruktur dargestellt.



Name	Date Modified	Size	Kind
▶ Config	Gestern 19:01	--	Folder
▶ Console	Gestern 19:01	--	Folder
▶ Controller	Gestern 19:01	--	Folder
index.php	Gestern 19:01	701 bytes	PHP script
▶ Lib	Gestern 19:01	--	Folder
▶ Locale	Gestern 19:01	--	Folder
▶ Model	Gestern 19:01	--	Folder
▶ Plugin	Gestern 19:01	--	Folder
▶ Test	Gestern 19:01	--	Folder
▶ tmp	Gestern 19:01	--	Folder
▶ Vendor	Gestern 19:01	--	Folder
▶ View	Gestern 19:01	--	Folder
▶ webroot	Gestern 19:01	--	Folder

Abbildung 8: Allgemeine Ordnerstruktur von CakePHP

Die Ordnerstruktur spiegelt in CakePHP die einzelnen Bereiche der Software wider. Alle Konfigurationsdateien wie die Zugangsdaten zur Datenbank, spezielle URLs und sonstige vordefinierte Werte sind in „Config“ untergebracht. In „Console“ müssen alle für einen Zugriff über ein Terminal relevanten Routinen untergebracht werden. „Controller“ beinhaltet die Business Logik der Software. „Lib“ und „Local“ beinhalten selbst erstellte Bibliotheken sowie Übersetzungen. Abstrahierte Datenmodelle von Tabellen der Datenbank sowie sonstige Datenmodelle und Datenbankanbindungen liegen in „Model“. Testroutinen werden in „Test“ erwartet, in „tmp“ liegen temporäre Daten wie Session IDs, Zwischenschritte diverser Berechnungen oder Cache-Inhalte. Der Ordner „Vendor“ ist für externe, importierte Bibliotheken vorgesehen. In „View“ müssen alle Templates für das Frontend abgelegt werden und „webroot“ beinhaltet die Scripts zur Bearbeitung von

Anfragen vom Frontend sowie vom Frontend benötigte Dateien wie Bilder und sonstige Ressourcen.

#### 4.4. Namenskonventionen

CakePHP wurde nach der Idee „convention over configuration“ („Konvention vor Konfiguration“) aufgebaut. Viele der bereits verfügbaren Funktionalitäten in diesem Framework stehen daher unmittelbar nach Umsetzen von Richtlinien und ohne weitere Konfiguration zur Verfügung. Eine Richtlinie ist unter anderem die Namensgebung von Programmteilen<sup>8</sup>. Namen sind durchgängig in englischer Sprache zu vergeben, da das Framework automatisch zwischen Plural und Singular der Bezeichnungen unterscheidet.

Soll die Software zum Beispiel Daten wie „Kommentare“ (engl. „comment“) zu „Beiträgen“ (engl. „post“) verwalten, muss zuerst ein passendes relationales Datenbankschema entworfen werden. Die Tabellen müssen mit „posts“ und „comments“ bezeichnet werden. Jede Tabelle muss neben den gewünschten Datenfeldern außerdem ein Feld „id“ besitzen, da dies zur eindeutigen Identifikation von CakePHP erwartet wird. Je nach Datenbank sollte dieses Feld zur Beschleunigung der Abfragen auch als Primärschlüssel definiert sein.

Da mehrere Kommentare zu Beiträgen gespeichert werden können sollen, muss im Sinne einer relationalen Datenbank in der Tabelle „comments“ ein Feld „post\_id“ erstellt werden. Dieses wird im nächsten Schritt mit genau diesem Namen von CakePHP erwartet.

Damit auf die Tabellen „posts“ und „comments“ über die Software zugegriffen werden kann, muss je Tabelle ein passendes Model<sup>9</sup> „Post“ und „Comment“ nach folgendem Schema angelegt werden.

Datei „Model/Post.php“:

```
App::uses('AppModel', 'Model');
class Post extends AppModel {
    public $name = 'Post';
    public $hasMany = 'Comment';
}
```

Datei „Model/Comment.php“:

```
App::uses('AppModel', 'Model');
class Comment extends AppModel {
    public $name = 'Comment';
    public $belongsTo = 'Post';
}
```

Die erste Zeile lädt jeweils (mit Framework-eigenen Methoden) die nötige Funktionsbibliothek, in Zeile zwei wird eine Klasse für das jeweilige Model angelegt, Zeile drei speichert den Namen für die spätere Verwendung im Framework und Zeile vier legt eine Beziehung zu anderen Klassen bzw. Tabellen fest.

---

<sup>8</sup> Details zur Namensgebung siehe <http://book.cakephp.org/2.0/en/getting-started/cakephp-conventions.html>

<sup>9</sup> Details zum „Model“-Teil des Frameworks siehe <http://book.cakephp.org/2.0/en/models.html>

Nach diesem Schritt steht im „Controller“-Teil der Software der abstrahierte Zugriff auf Beiträge und Kommentare zur Verfügung. Je nach Aufruf werden beim Abruf eines Beitrags automatisch alle zugehörigen Kommentare mitgeliefert.

Die Businesslogik befindet sich jeweils im sogenannten „Controller“<sup>10</sup>. In diesem Beispiel wird erwartet, dass ein Programmteil, in welchem Beiträge bearbeitet werden, in einem „Controller“ mit dem Namen „PostsController“ stattfinden. Ein einfacher Controller zum Auflisten aller Beiträge könnte wie folgt aussehen.

Datei „Controller/PostsController.php“:

```
App::uses('AppController', 'Controller');
class PostsController extends Controller {
    public function index() {
        $posts = $this->Post->find( 'all' );
        $this->set('posts', $posts);
    }
}
```

In der ersten Zeile wird wiederum die Funktionsbibliothek importiert und Zeile zwei erstellt die Controller-Klasse. Die Funktion „index“ ruft in der dritten Zeile alle Posts ab und gibt diese in Zeile vier an den „View“-Teil<sup>11</sup> der Software weiter, welcher auf Basis vorhandener Vorlagen im Ordner „View“ die Ausgabe erzeugt.

Durch das Anlegen der Klasse „PostsController“ mit der Funktion „index“ steht sofort ein URL Aufruf in folgender Form zur Verfügung:

```
http://[server]/posts/index
```

Durch diesen Aufruf wird das Framework initialisiert, die Businesslogik gestartet und die Datenbankverbindung aufgebaut. Anschließend werden alle Beiträge abgefragt und an den Präsentationsteil der Software weitergeben, welcher auf Basis verfügbarer Vorlagen eine Ausgabe erzeugt.

## 4.5. Datenbank

Als Datenbank wurde eine relationale Datenbank eingesetzt. Die Umsetzung erfolgte mit MySQL. Zur Modellierung der Datenbank wurde das original Entity-Relationship-Modell nach Peter Chen (Chen, 1976) eingesetzt.

Die Datenbank ist für den Import aller verfügbaren Daten aus „Untis“ und „SchüSta“ ausgelegt (siehe 3). Teile der Daten wie etwa „Raum“ (beschreibt ua. Stämmräume laut Stundenplanprogramm), „Zeitslot“ (beschreibt die möglichen Stunden einer Schulwoche) und deren Zuordnungen werden nur für eine eventuelle zukünftige Verwendung gespeichert aber in der Software nicht verwendet. Sie werden daher in dem folgenden Diagramm aus Platzgründen ausgespart.

Das Datenbankschema ist stark an die Importquellen angelehnt, um die Daten in möglichst derselben Struktur weiterverarbeiten zu können. Es gibt Platz für Verbesserung beim

---

<sup>10</sup> Details zum „Controller“-Teil des Frameworks siehe <http://book.cakephp.org/2.0/en/controllers.html>

<sup>11</sup> Details zum „View“-Teil des Frameworks siehe <http://book.cakephp.org/2.0/en/views.html>

Umstieg von „SchüSta“ auf „Sokrates“ oder eventuell bei einem großen Versionswechsel von „Untis“ und damit verbundenen Änderungen beim Export.

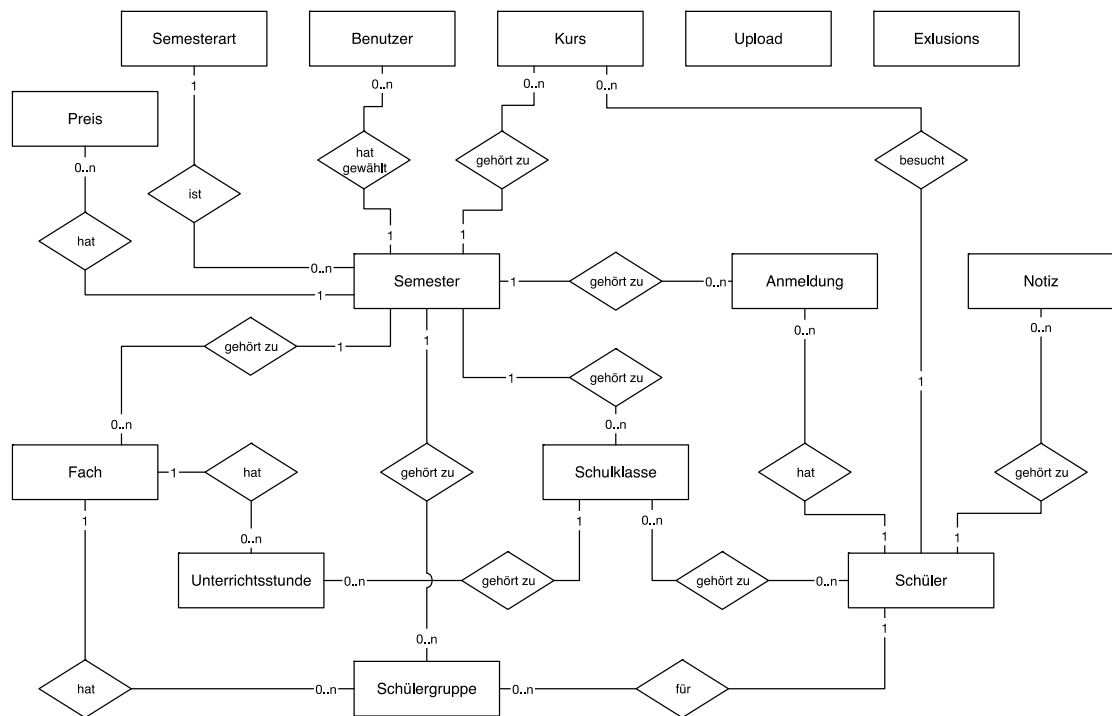


Abbildung 9: ER Modellierung nach Peter Chen (bereits vorbereitet für „Sokrates“)





## 5. Bedienung

Die Verwendung des NabeTools sieht den Import aller Stammdaten sowie Stundenpläne gefolgt von der Eingabe der Anmeldungen und der Ausgabe von Listen zur Verwendung durch das Betreuungspersonal vor. Jeder importierte Teil kann zu einem späteren Zeitpunkt erneut importiert werden um aktualisierte Daten einzuspielen.



Diagramm 1: Ablauf NabeTool

### 5.1. Semesterverwaltung

Außer den Stammdaten der SchülerInnen (siehe 5.2) und den schulfreien Tagen (siehe 5.3) werden alle eingegebenen oder importierten Informationen einzelnen Semestern zugeordnet. Daher ist pro Semester ein einmaliges Anlegen eines solchen erforderlich.

Es ist möglich, ein Semester anzulegen und alle Anmeldungen eines anderen Semesters zu importieren. So können einerseits diverse Situationen zu Testzwecken durchgespielt und andererseits Anmeldungen für das Sommersemester aus dem Wintersemester übernommen werden.

Das aktuelle Semester kann in der Kopfzeile rechts oben abgelesen und jederzeit unter dem Menüpunkt „Semester“ verändert werden.

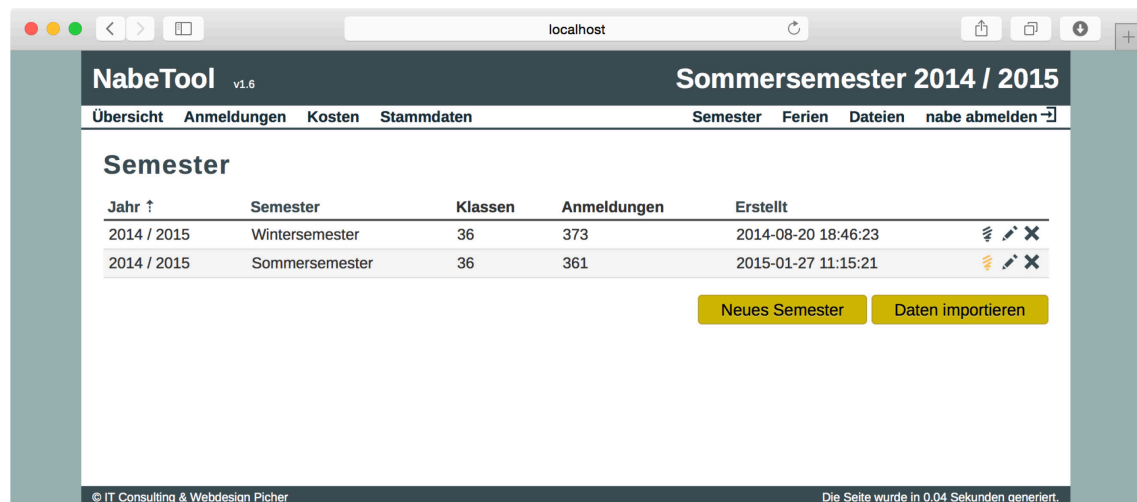


Abbildung 10: Semesterverwaltung

### 5.2. Import von Daten

Vor dem Einsatz des NabeTools zur Planung der Betreuungen müssen die notwendigen Basisdaten importiert werden. Zu diesen zählen die Stammdaten der SchülerInnen und die

Informationen über die ihnen zugeordnete Klasse. Ohne diese Daten können keine Anmeldungen verwaltet werden. Sie bilden daher die notwendige Grundlage.

Damit sowohl Stammdaten als auch Stundenpläne im NabeTool zur weiteren Verarbeitung zur Verfügung stehen, müssen die aus „Untis“, „SchüSta“ und „WebUntis“ exportierten Daten zuvor unter dem Menüpunkt „Dateien“ hochgeladen werden. Ihre Begründung findet diese Vorgehensweise darin, dass so die Administration die Daten zur Verfügung stellt und die Leitung der Nachmittagsbetreuung an einem anderen Ort damit weiter arbeiten kann.

Name ↓	Kommentar	Größe	Dateityp	Hochgeladen	
schülergruppen_2014-11-03.txt		103 KB	text/plain	2014-11-03 10:38:32	✕
sokrates_2015-01-27.txt	Sokrates 2015-01-27	99 KB	text/plain	2015-01-27 11:16:54	✕
sokrates_v2014-09-06.txt		95 KB	text/plain	2014-09-06 20:09:31	✕
sokrates_v2014-10-01.txt		97 KB	text/plain	2014-10-01 11:27:09	✕
sokrates_v2014-11-03.txt		97 KB	text/plain	2014-11-03 10:38:48	✕
sokrates_v2014-12-02.txt		98 KB	text/plain	2014-12-02 08:54:42	✕
sokrates_webuntis_2014-09-28.txt		96 KB	text/plain	2014-09-28 22:17:56	✕
StudentgroupStudents (2).txt	Stand 2014-09-19 um 09:32	111 KB	text/plain	2014-09-19 09:33:21	✕
StudentgroupStudents (3).txt	2014-09-22	114 KB	text/plain	2014-09-22 10:50:44	✕
StudentgroupStudents (4).txt	2014-09-22 15:48	115 KB	text/plain	2014-09-22 15:49:01	✕
StudentgroupStudents (5).txt		134 KB	text/plain	2014-10-01 11:31:28	✕
StudentgroupStudents 2015-02-08-1.txt		102 KB	text/plain	2015-02-08 22:28:22	✕
StudentgroupStudents-1.txt	Schülergruppen 2015-01-27	102 KB	text/plain	2015-01-27 11:35:22	✕
StudentgroupStudents.txt		100 KB	text/plain	2014-09-11 01:12:53	✕
untis_2014-09-05 ohne alias.xml		870 KB	text/xml	2014-09-06 22:42:32	✕

1 | 2

Hochladen

Abbildung 11: Dateiverwaltung

Nachdem die gewünschten Quelldaten hochgeladen wurden, können diese in der Semesterverwaltung (Menüpunkt „Semester“) in das aktuell aktivierte Semester eingespielt werden.

Sofern der Stundenplan noch nicht fertig ist, können aktualisierte Stundenplandaten jederzeit nachgereicht werden. Diese umfassen Unterrichtseinheiten und deren Unterrichtszeiteinheiten. Sie sind für die Verwaltung von Anmeldungen nicht relevant und daher im Notfall solange optional, bis Anwesenheitslisten, welche auf dem Stundenplan basieren, generiert werden sollen.

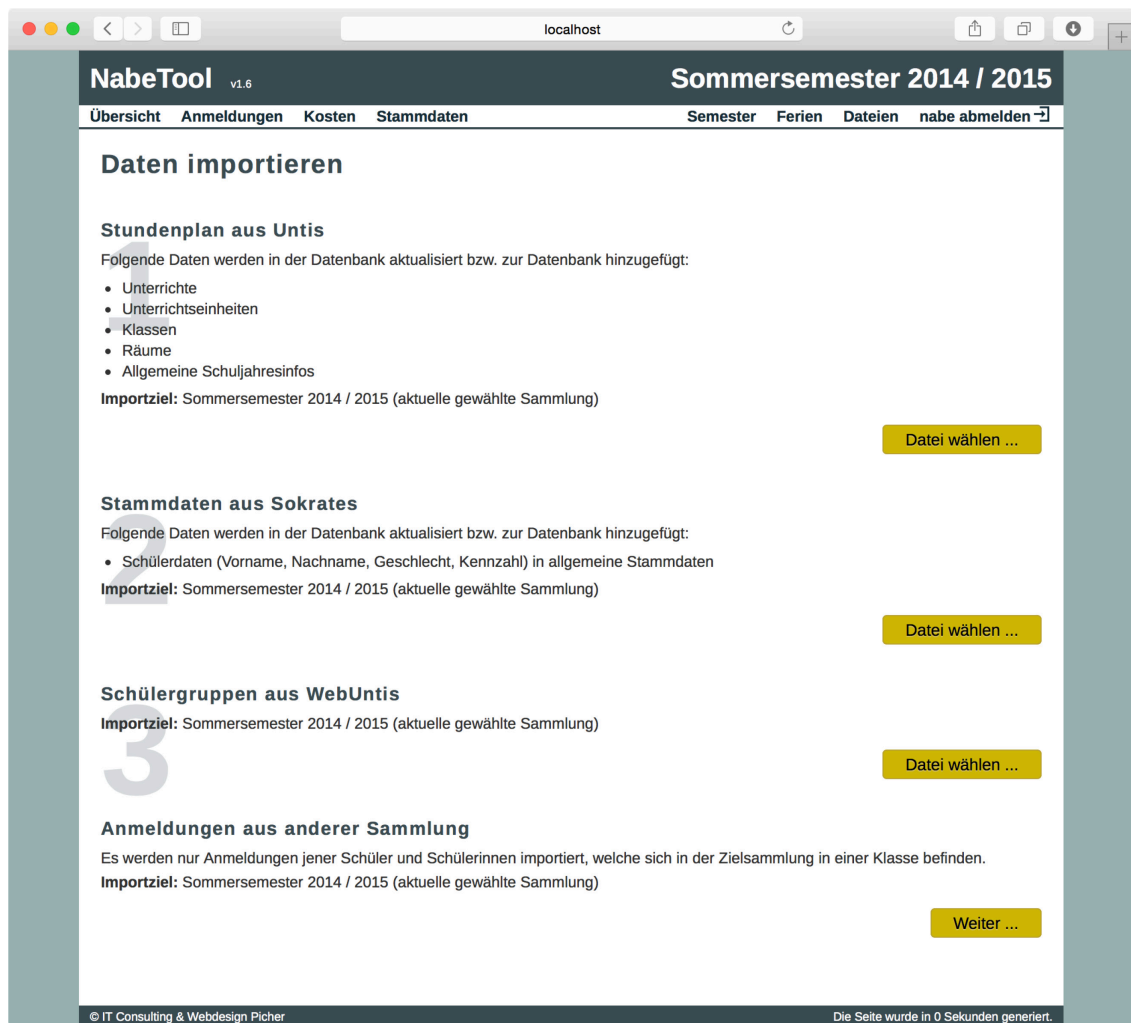


Abbildung 12: Importmöglichkeiten (bereits mit Import aus „Sokrates“)

Das NabeTool speichert beim Import nur die für die weitere Verarbeitung notwendigen Felder der Datensätze. Nicht benötigte Daten des Imports werden ignoriert. Alle Daten werden in der dem NabeTool zugrundeliegenden relationalen Datenbank gespeichert.

Beim Import der Basisdaten aus dem CSV Export von „SchüSta“ (siehe 3.3.3) erfolgt automatisch eine Strukturierung der Daten vor der Speicherung in der Datenbank. So ist zum Beispiel ein Abruf aller SchülerInnen einer Klasse zu einem späteren Zeitpunkt einfach möglich.

Beim Import der Daten aus „Untis“ werden die Daten ebenfalls strukturiert und in der relationalen Datenbank des NabeTools aufgenommen. Die Strukturierung ist hier wesentlich einfacher, da die Quelle bereits strukturierte Daten beinhaltet (siehe 3.4.4). Es muss lediglich eine Verknüpfung der Klassen aus „Untis“ mit den bereits aus „SchüSta“ importierten Klassen erfolgen.

Alle importierten Daten können jederzeit durch erneutes Importieren aktualisiert werden. Zu aktualisierende Stammdaten der SchülerInnen werden anhand der schülerspezifischen SchülerInnenkennzahl erkannt, neue SchülerInnen werden ergänzt. Der Stundenplan wird vor jedem erneuten Import gelöscht und neu eingespielt.

### 5.3. Schulfreie Tage

Schulfreie Tage sind laut Schulzeitgesetz (Schulzeitgesetz, 2015) gesetzliche Feiertage, vom BMBF festgelegte Ferien und schulautonom festgelegte schulfreie Tage<sup>12</sup>. An diesen Tagen findet keine Betreuung statt. Um die Übersichtlichkeit zu fördern, können auszuschließende Tage und Zeiträume im NabeTool definiert werden. So definierte Zeiträume werden bei allen druckbaren Listen ausgenommen und als Ferien aufgelistet.

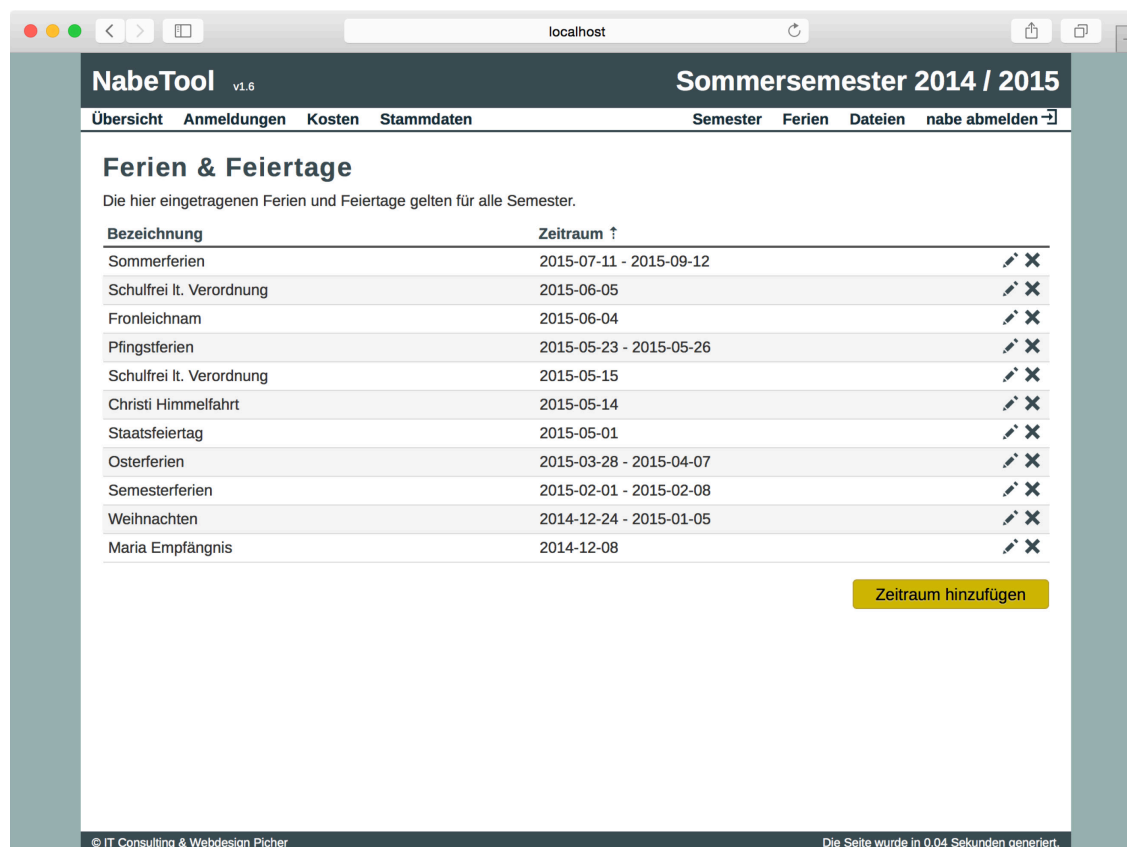


Abbildung 13: Schulfreie Tage

### 5.4. Verwalten der Anmeldung

Eine Anmeldung wird von der Verwaltung der Tagesbetreuung angelegt, sobald Erziehungsberechtigte ihr Kind zur Nachmittagsbetreuung und/oder zur Mittagsbetreuung angemeldet haben. Eltern füllen dazu jeweils einen Anmeldebogen für die Nachmittagsbetreuung und/oder für die Mittagsbetreuung aus. Bei der Mittagsbetreuung genügt die Angabe der in Anspruch zu nehmenden Tage. Bei der Nachmittagsbetreuung müssen die Eltern weiters eine Uhrzeit für die gewünschte Entlassung angeben. Außerdem können Eltern Anmerkungen wie zB. die abholenden Personen, der zu erreichende Bus oder ähnliches anfügen.

Die gesammelten Anmeldungen werden vom Verwalter der Tagesbetreuung auf Rechtmäßigkeit geprüft und unter dem Menüpunkt „Anmeldungen“ in das NabeTool

<sup>12</sup> Schulzeitgesetz § 2 Abs. 5: „Aus Anlässen des schulischen oder sonstigen öffentlichen Lebens kann das Klassen- oder Schulforum bzw. der Schulgemeinschaftsausschuss höchstens fünf Tage in jedem Unterrichtsjahr schulfrei erklären. (...)“

eingetragen. Besondere Aufmerksamkeit wird Anmeldungen mit Kommentaren zu rechtlich schwierigen Situationen wie Sorgerechtsstreitigkeiten geschenkt. Oft wird in diesen Fällen mit den Erziehungsberechtigten zusätzlich persönlich Rücksprache gehalten, wer das Kind wann abholen darf. Außerdem erfolgt ein Informationsabgleich mit der Direktion und allen betreuenden LehrerInnen. Um solche Fälle zu dokumentieren, gibt es bei jeder Anmeldung die Möglichkeit, „Ausweiskontrolle bei der Abholung“ anzuhaken. Das führt dazu, dass auf allen Anwesenheitslisten ein rotes Warndreieck bei der/dem betroffenen SchülerIn abgedruckt wird, um die betreuenden LehrerInnen an diesen Fall zu erinnern.

Klasse	Schüler ↓	☿	Typ	Tage	Notiz	Optionen	Mo	Di	Mi	Do	Fr	
3Ag		♀	MB	1					✓			✎ ✕  K
1B		♂	NABE	2	🕒			✓		✓		✎ ✕  K
1Db		♀	MB	1	📅			✓				✎ ✕  K
4Cg		♀	MB	1						✓		✎ ✕  K
3Br		♂	MB	1					✓			✎ ✕  K
2A		♀	NABE	2			✓	✓				✎ ✕  K
2A		♀	MB	3			✓		✓	✓		✎ ✕  K
4Br		♀	MB	1				✓				✎ ✕  K
1Db		♀	MB	1	📅			✓				✎ ✕  K
1B		♂	NABE	5	🕒		✓	✓	✓	✓	✓	✎ ✕  K
3Br		♂	MB	1					✓			✎ ✕  K
4Cg		♂	MB	1						✓		✎ ✕  K
2A		♂	MB	1			✓					✎ ✕  K
2D		♂	MB	1			✓					✎ ✕  K
3Db		♀	MB	2			✓		✓			✎ ✕  K
2D		♀	MB	1			✓					✎ ✕  K
2D		♀	NABE	2	🕒			✓	✓			✎ ✕  K
2B		♂	MB	1			✓					✎ ✕  K
3Br		♂	MB	1					✓			✎ ✕  K
4Ag		♀	MB	1				✓				✎ ✕  K
1Db		♀	MB	1				✓				✎ ✕  K

Abbildung 14: Übersicht aller Anmeldungen

**NabeTool** v1.6 **Sommersemester 2014 / 2015**

Übersicht Anmeldungen Kosten Stammdaten Semester Ferien Dateien nabe abmelden

### Anmeldung bearbeiten

SchülerIn\*

Type  
☒ Nachmittagsbetreuung\* ☐ Mittagsbetreuung\*

**Zeiten**

	Mo	Di	Mi	Do	Fr
Angemeldet	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bis Uhrzeit	<input type="text"/> : <input type="text"/>	16 : 30	<input type="text"/> : <input type="text"/>	<input type="text"/> : <input type="text"/>	<input type="text"/> : <input type="text"/>

☐ SchülerIn darf selbstständig auch früher gehen

**Abholung**

☒ SchülerIn darf die Nachmittagsbetreuung NUR in Begleitung verlassen

Abholung durch

☐ Ausweiskontrolle bei der Abholung

**Speichern**

© IT Consulting & Webdesign Picher Die Seite wurde in 0.06 Sekunden generiert.

Abbildung 15: Anmeldung erstellen/bearbeiten

Da es vorkommt, dass Erziehungsberechtigte die Formulare nicht richtig lesen und daher fehlerhaft ausfüllen, ist es besonders wichtig, bei jedem unklaren Fall Rücksprache zu halten, um die richtige Art der Anmeldung sowie die korrekte Anzahl an Tagen eingeben zu können. Diese Daten fließen sowohl in die für den Landesschulrat zu erstellende Statistik, als auch in die Verrechnung der für die Eltern entstehenden Kosten direkt ein. Da die Auswertung für den Landesschulrat direkten Einfluss auf die der Schule zur Verfügung stehenden Bezahlung der Betreuungslehrer hat, wirkt sich eine falsche Statistik hier sehr unangenehm aus.

Ist es notwendig, dass Angaben zu Anmeldungen in Rücksprache mit Erziehungsberechtigten adaptiert werden müssen, kann das zu jeder Zeit passieren. Um bei einem Telefongespräch alle nötigen Daten auf einen Blick ersichtlich zu machen, steht eine Übersichtsseite pro SchülerIn zur Verfügung. Allfällige Notizen können direkt erfasst werden. Erfolgte Änderungen fließen sofort in kommende Ausgaben ein.

**NabeTool v1.6** Sommersemester 2014 / 2015

Übersicht Anmeldungen Kosten Stammdaten Semester Ferien Dateien nabe abmelden

### Schülerdetails

**Details**

Kennzahl:

Geschlecht: ☐ ☒

**Anmeldungen**

Typ	Tag	Mo	Di	Mi	Do	Fr	Abholung	Interne Notiz	Kosten
NABE	3	✓ 16:00:00	✓ 16:00:00		✓ 16:00:00				€ 264,00
MB	1			✓					-

**Notizen**

Typ	Datum	Uhrzeit	Text
	2015-03-04	12:02:00	Ergänzt nach Rücksprache <input type="text"/> durch <input type="text"/>

[Notiz hinzufügen](#)

**Stundenplan**

	Mo	Di	Mi	Do	Fr	Sa	So
1	GPB	D	D	D	F		
2	D	M	E	E	M		
3	BE	E	F	F	E		
4	BE	ME	M	F	ME		
5	RE	BSPM		GPB	BIUK		
6	PH		GWK	PH			
7							
8			BSPM/SASS				
9			BSPM/SASS				
10			BSPM/SASS		UÜEM/UÜEM		

**Verlauf der Schulklassen**

Es werden die Schulklassen aller verwalteten Semester angezeigt.

Name	Schuljahr	Semester
3Cg	2014	Wintersemester
3Cg	2014	Sommersemester

© IT Consulting & Webdesign Picher Die Seite wurde in 0,06 Sekunden generiert.

Abbildung 16: Anmeldeübersicht eines/einer SchülerIn

Um Notizen schnell und unkompliziert erfassen zu können, stehen einige Kategorien zur Verfügung. Datum und Uhrzeit werden automatisch erfasst. Notizen stellen eine einfache Möglichkeit dar, Informationen zu Änderungen von Anmeldungen schriftlich zu erfassen und für eine eventuell nötige Recherche des Herganges von Änderungen im Diskussionsfall zur Verfügung zu stehen.

NabeTool v1.6 Sommersemester 2014 / 2015

Übersicht Anmeldungen Kosten Stammdaten Semester Ferien Dateien nabe abmelden

### Notiz hinzufügen

Type

☐ Telefonat\*\* ☐ Persönlich\*\* ☐ Email\*\* ☐ Intern\*\*

Date\*

25 May 2015

Time\*

14 : 53

Text\*

Speichern

© IT Consulting & Webdesign Picher Die Seite wurde in 0.04 Sekunden generiert.

Abbildung 17: Notiz hinzufügen

## 5.5. Anmeldestatistik

Der Landesschulrat benötigt zu Beginn jedes Schuljahres eine Anmeldestatistik um die zuzuweisenden Betreuungseinheiten für eine Schule festzulegen. Dazu muss erhoben werden, wie viele Anmeldungen – getrennt nach männlichen und weiblichen Schülern – zu ein, zwei, drei, vier und fünf Tagen eingegangen sind. Da anhand dieser Angaben die tatsächliche Bezahlung der LehrerInnen und somit die möglichen Betreuungszeiten berechnet werden, ist diese Auswertung existenziell für die Tagesbetreuung.



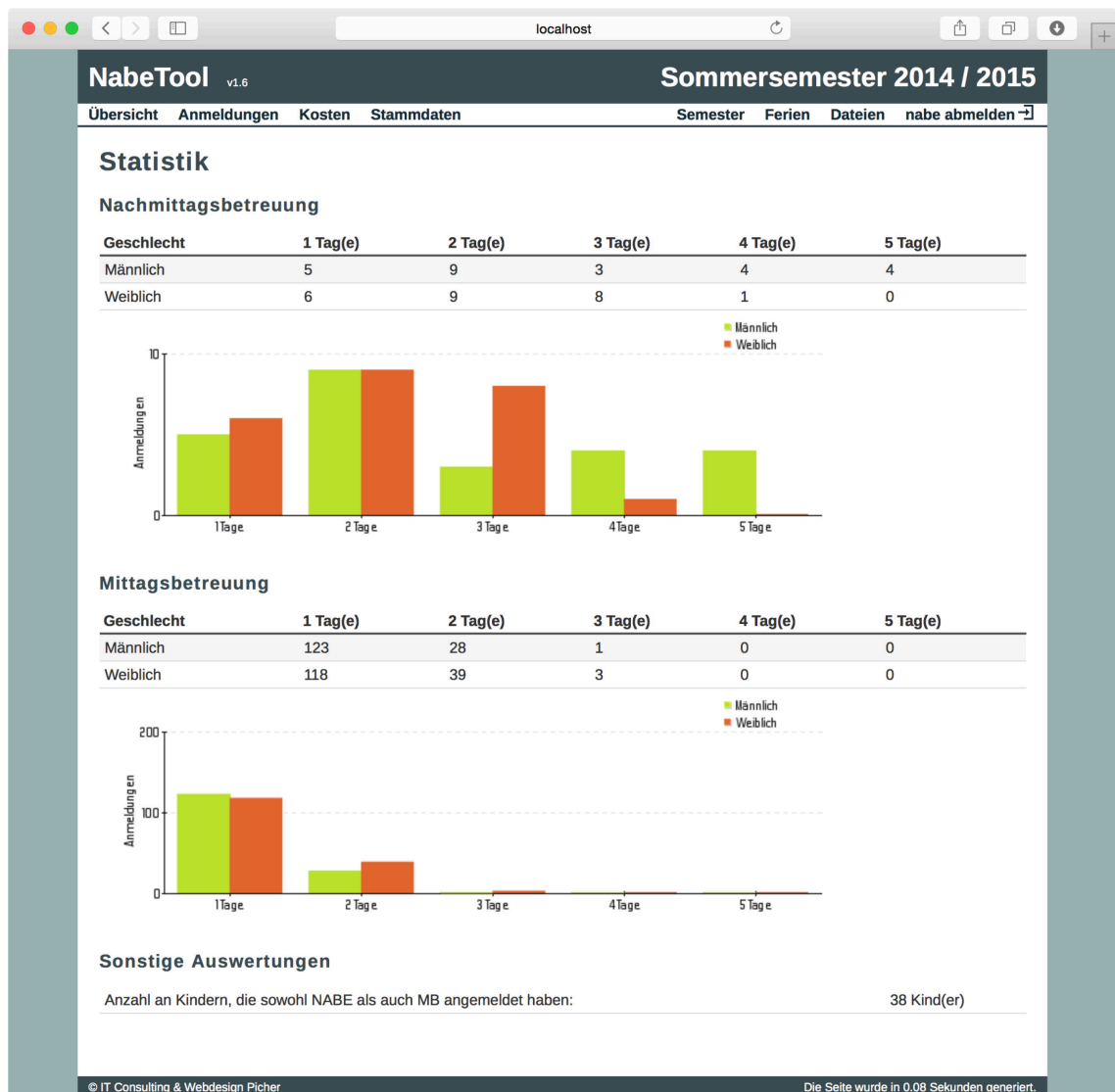


Abbildung 18: Automatische Erstellung der Anmeldestatistik für den Landesschulrat

## 5.6. Erkennung von Betreuungsstunden

Für den weiteren Ablauf werden von der Software automatisch jene Stunden aller angemeldeten SchülerInnen ermittelt, in welchen sie/er laut den jeweiligen Anmeldungen betreut werden muss. Hierbei wird direkt bei der Erkennung anhand der offiziellen Regelung vom Bundesministerium entschieden, welcher Typ an Betreuungsstunden vorliegt (siehe 2.2).

## 5.7. Festlegung der Kosten

Da die Betreuung von SchülerInnen während der Mittagsbetreuung kostenlos ist, muss dafür kein Preis festgelegt werden. Sehr wohl muss aber der pauschalisierte Betrag für die Nachmittagsbetreuung eingegeben werden (siehe 2.2.2). Dadurch wird unter anderem die Weiterverwendung der Daten für die Buchhaltung ermöglicht.

Die Kosten für die Nachmittagsbetreuung müssen einmalig pro Semester unter dem Menüpunkt „Kosten“ festgelegt werden.

Tag	Preis	Bearbeitet
1	€ 132,00	2015-02-26 11:37:37
2	€ 176,00	2015-02-26 11:37:43
3	€ 264,00	2015-02-26 11:37:49
4	€ 352,00	2015-02-26 11:37:55
5	€ 440,00	2015-02-26 11:37:59

Hinzufügen

Abbildung 19: Preise für Nachmittagsbetreuung

## 5.8. Anwesenheitsliste

Eine direkte Verwendung der Software zur Erfassung von Anwesenheiten ist aufgrund der nötigen Flexibilität innerhalb der Tagesbetreuung leider nicht möglich und wurde daher bereits in den Randbedingungen ausgeschlossen (siehe 2.5.2).

Damit die Anwesenheit schriftlich dokumentiert wird, ist es möglich, Listen in verschiedenen Formaten generieren zu lassen. Diese umfassen unter anderem eine Liste zur schnellen Erfassung der Anwesenheiten angemeldeter SchülerInnen der Nachmittagsbetreuung. Diese Liste beinhaltet auch die berechneten Anwesenheiten auf Basis des Stundenplans der einzelnen SchülerInnen (sofern diese bereits importiert wurde).

localhost

NabeTool v1.6

Sommersemester 2014 / 2015

Übersicht

Anmeldungen

Kosten

Stammdaten

Semester

Ferien

Dateien

nabe abmelden

Auswahl

Nachmittagsbetreuung

Mittagsbetreuung

Jänner

Februar

März

April

Mai

Juni

Juli

August

September

Oktober

November

Dezember

Montag

Dienstag

Mittwoch

Donnerstag

Freitag

Nachmittagsbetreuung Freitag im Mai

Kl.	Schüler	bis	08. May 2015 (KW 19)	22. May 2015 (KW 21)	29. May 2015 (KW 22)	Verlassen
1B		15:35	E 7 8	E 7 8	E 7 8	🕒
1B		15:20	E 7 8	E 7 8	E 7 8	🕒
1B		16:00	E 7 8 9	E 7 8 9	E 7 8 9	
1Cb		16:00	E 7 8 9	E 7 8 9	E 7 8 9	🕒
1Cb		16:00	E 7 8 9	E 7 8 9	E 7 8 9	
1Cb		16:00	E 7 8 9	E 7 8 9	E 7 8 9	🕒
1Cb		17:00	E 7 8 9	E 7 8 9	E 7 8 9	🕒
1Cb		15:30	E 7 8	E 7 8	E 7 8	
2B		14:45	E 7 8	E 7 8	E 7 8	🕒
2Eb		16:00	E 7 8 9	E 7 8 9	E 7 8 9	👤
2Eb		14:45	E 7 8	E 7 8	E 7 8	
2Eb		15:30	E 7 8	E 7 8	E 7 8	👤
4Br		16:00	E 7 8 9	E 7 8 9	E 7 8 9	🕒

Ferien

01. May

15. May

© IT Consulting & Webdesign Picher

Die Seite wurde in 0.19 Sekunden generiert.

Abbildung 20: Auswahl der zu druckenden Anwesenheitsliste

Nachmittagsbetreuung Freitag im Mai						
Kl.	Schüler	bis	08. May 2015 (KW 19)	22. May 2015 (KW 21)	29. May 2015 (KW 22)	Verlassen
1B		15:35	E 7 8	E 7 8	E 7 8	🕒
1B		15:20	E 7 8	E 7 8	E 7 8	🕒
1B		16:00	E 7 8 9	E 7 8 9	E 7 8 9	
1Cb		16:00	E 7 8 9	E 7 8 9	E 7 8 9	🕒
1Cb		16:00	E 7 8 9	E 7 8 9	E 7 8 9	
1Cb		16:00	E 7 8 9	E 7 8 9	E 7 8 9	🕒
1Cb		17:00	E 7 8 9	E 7 8 9	E 7 8 9	🕒
1Cb		15:30	E 7 8	E 7 8	E 7 8	
2B		14:45	E 7 8	E 7 8	E 7 8	🕒
2Eb		16:00	E 7 8 9	E 7 8 9	E 7 8 9	👤
2Eb		14:45	E 7 8	E 7 8	E 7 8	
2Eb		15:30	E 7 8	E 7 8	E 7 8	👤
4Br		16:00	E 7 8 9	E 7 8 9	E 7 8 9	🕒

Ferien

01. May

15. May

© IT Consulting & Webdesign Picher

Abbildung 21: Gedruckte Anwesenheitsliste der Nachmittagsbetreuung

Eine Voraussetzung für das praktische Handhaben dieser Listen zur Erfassung der Anwesenheiten ist einerseits die Übersichtlichkeit und andererseits die Nachvollziehbarkeit und somit die Zuordnung zu den Tagen. Daher werden unter anderem die SchülerInnen nach Klassen und Nachname sortiert und dieselben Wochentage eines Monats auf einem Ausdruck vereint. Dadurch ergibt sich schnell ein guter Überblick über den Verlauf der Anwesenheiten.

Die Anwesenheitsliste ist so gestaltet, dass die Anwesenheit von SchülerInnen durch Einkreisen der Stunde zu und als Beginn der Anwesenheit dokumentiert werden kann. Die SchülerInnen sind angewiesen, sich vor Verlassen der Tagesbetreuung bei der Lehrperson zu melden. Diese kontrolliert, ob die Tagesbetreuung laut Anmeldung verlassen werden darf und macht ein Häkchen um das Ende der Anwesenheit zu kennzeichnen.

Fehlen SchülerInnen, die laut Anwesenheitsliste aber anwesend sein sollten, werden vom/von betreuenden LehrerIn umgehend die Eltern verständigt. Unabhängig davon werden KlassenkollegInnen befragt und im Klassenbuch allfällige Stundenplanänderungen überprüft. Sollten die Eltern nicht erreichbar sein, wird der Anruf notiert und der/die LehrerIn hinterlässt nach Möglichkeit eine Mitteilung am Anrufbeantworter. Die fehlenden SchülerInnen werden daraufhin am Schulgelände gesucht und mithilfe von KlassenkollegInnen telefonisch kontaktiert.

## **5.9. Export für die Buchhaltung**

Um eine Weiterverarbeitung der Anmeldungen in der Buchhaltung zu ermöglichen, können alle Anmeldungen zur kostenpflichtigen Nachmittagsbetreuung eines Semesters gesammelt exportiert werden. Diese Liste umfasst Namen, Kontaktdaten der Eltern, Anzahl der Tage und berechnete Kosten der angemeldeten SchülerInnen. Diese Liste ist für die weitere Verwendung als Excel Datei (Microsoft, 2015) exportierbar.

## **5.10. Probleme bei der Verarbeitung der Daten**

Bei der internen Generierung der SchülerInnenstundenpläne zur Berechnung der Betreuungsstunden kann es an mehreren Stellen zu Problemen kommen.

Besuchen SchülerInnen einer Klasse Wahlfächer – dazu zählt zum Beispiel von einzelnen SchülerInnen in Anspruch genommener Instrumentalunterrichte – muss für diese Klasse pro SchülerIn ein individueller Stundenplan generiert werden. Diese Berechnung erfolgt auf Basis der aus „SchüSta“ importierten Wahlgegenstände, sowie dem Stundenplan dieser Gegenstände aus „Untis“. Da die Eintragung der Wahlgegenstände oftmals durch verschiedene Personen wie Klassenvorstände vorgenommen wird, kann es hier zu Dateninkonsistenz durch fehlende Einträge bzw. Fehleinträge kommen. Dadurch stimmen oft die berechneten Betreuungsstunden anfangs nicht.

Ein weiteres Problem ist die Benennung der Klassen. Werden Klassenverbände von der Administration im „SchüSta“ anders als in „Untis“ bezeichnet, kann eine automatische

Zuordnung der SchülerInnen zum Klassenstundenplan nicht mehr erfolgen. Das wird in der Praxis relativ schnell dann zu einem Problem, wenn in „SchüSta“ Klassen in deren Gesamtheit geführt werden, in „Untis“ jedoch für einzelne Klassenteile (zB. bei Trennung nach Schultypen) andere Stundenpläne entworfen werden und die Unterscheidung nur anhand eines benutzerdefinierten Feldes in „SchüSta“ getroffen wird.



## 6. Beiliegender Datenträger

Die Software sowie Beispieldaten sind der Arbeit auf einem Datenträger beigelegt.

### 6.1. Inhalt des Datenträgers

Ordner	Datei	Beschreibung
Software	nabetool.zip	Software komprimiert im ZIP Format
Datenbank	database.sql	SQL Befehle zum Anlegen der Grundstruktur der Datenbank (die Datenbank muss beim Import bereits existieren)
Beispieldaten	example.sql	SQL Befehle zum Importieren von Beispieldaten
Beispieldaten	sokrates.txt	Beispielhafter Export von Schülerstammdaten aus „Sokrates“
Beispieldaten	untis.xml	Beispielhafter XML Export aus „Untis“
Beispieldaten	webuntis.txt	Beispielhafter Export von Schülergruppen aus „WebUntis“

### 6.2. Installation

Im Folgenden wird davon ausgegangen, dass eine passende Umgebung für den Betrieb des NabeTools gegeben ist (siehe 2.4.2). Die beigelegten Installationsdateien wurden auf MySQL 5.0.11 mit InnoDB sowie Apache/2.2.29 mit PHP 5.6.7 getestet.

Folgende Schritte müssen durchgeführt werden, um die Software zu installieren. Alle Pfadangaben sind – wenn nicht anders angegeben – relativ zum gewählten Installationsverzeichnis zu verstehen. Alle Befehle werden im Installationsverzeichnis ausgeführt.

- 1) Anlegen einer neuen Datenbank am Datenbankserver und importieren der grundlegenden Datenbankstruktur mit Hilfe der Befehle in „database.sql“.
- 2) Importieren der Beispieldaten in aus „example.sql“ (optional).
- 3) Anlegen eines Datenbankbenutzers mit Zugriff auf die zuvor angelegte Datenbank.
- 4) Entpacken der Software aus „nabetool.zip“ in einen Unterordner des Stammverzeichnisses des Webservers.
- 5) Eingeben der Zugangsdaten für die Datenbank in der Datei „Config/database.php“
- 6) Vergeben von Schreibrechten für den Benutzer, unter welchem der Webserver läuft, für folgende Verzeichnisse und deren Unterverzeichnisse:
  - a. tmp
  - b. webroot/files

7) Installation der Software „Composer“<sup>13</sup>:

- a. Auf Mac OS X/Unix/Linux im Terminal:

```
$ curl -sS https://getcomposer.org/installer | php
```

- b. Auf Windows per Installationsdatei vom Hersteller:

<https://getcomposer.org/doc/00-intro.md#installation-windows>

8) Nachladen aller nötigen Abhängigkeiten des NabeTools mit „Composer“:

- a. Auf Mac OS X/Unix/Linux im Terminal:

```
$ php composer.phar install --no-dev
```

- b. Auf Windows in der Eingabeaufforderung:

```
> composer install --no-dev
```

### 6.3. Einspielen der Beispieldaten

Die Beispieldaten umfassen drei Klassen mit jeweils einigen Schülern. Die Stundenpläne dieser Klassen beinhalten sowohl wöchentlichen als auch zweiwöchentlichen sowie frei wählbaren Nachmittagsunterricht.

Die Daten liegen sowohl in SQL Befehlen zum sofortigen Import in die Datenbank als auch in jenem Format vor, das nach Exporten aus „Untis“, „WebUntis“ und „Sokrates“ zur Verfügung steht.

Zum Importieren der SQL Daten müssen lediglich die SQL Befehle in der Datei „example.sql“ in der in Punkt 6.2 angelegten Datenbank ausgeführt werden. Die anderen Daten können über die Anwendungsoberfläche importiert werden.

### 6.4. Zugangsdaten

Die Zugangsdaten für das NabeTool lauten wie folgt:

Benutzername: nabe

Kennwort: tool

Die Zugangsdaten können in der beiliegenden Version der Software nicht in der Benutzeroberfläche geändert werden, sondern sind fest vorgegeben. Für einen dauerhaft produktiven Einsatz muss diese Funktionalität ergänzt werden.

---

<sup>13</sup> „Composer“ ist ein Paketmanager für PHP mit welchem zuvor definierte Abhängigkeiten zu anderen Softwarepaketen erfüllt werden können. Homepage: <https://getcomposer.org>



## **7. Zusammenfassung**

### **7.1. Wirkung der Software**

Das NabeTool hat die Führung der Tagesbetreuung für das Personal wesentlich vereinfacht. Wo sich vorher Berge an An- und Ummeldungsbögen, händisch geschriebene, unübersichtliche Anwesenheitslisten und Unmengen an Notizzetteln gestapelt haben, liegt jetzt nur eine dünne Mappe mit der Anwesenheitsliste. Durch die transparente Struktur dieser Anwesenheitsliste und deren Aufbewahrungsort wurde den einzelnen BetreuerInnen die Ungewissheit über die zu erwartenden und zu beaufsichtigenden SchülerInnen genommen, welche durch die verschieden umfangreichen Anmeldungen ohne Listen nur mehr schlecht bis gar nicht mehr überblickbar war.

### **7.2. Zukünftige Erweiterungen und Anpassungen**

Mögliche Erweiterungen für die weitere Erleichterung der Führung der Tagesbetreuung umfassen neben der Möglichkeit der online Anwesenheitserfassung auch den Ausbau der Exportmöglichkeiten, die Erweiterung aller Ansichten um weitere Sortier- und Filteroptionen, den automatischen Versand von Anmeldebestätigungen an Erziehungsberechtigte per E-Mail sowie eine direktere und somit einfachere Anbindung an die Datenquellen.

### **7.3. Persönliche Meinung**

Meiner Meinung nach hat alleine die bisherige Entwicklung des NabeTools eine wesentliche Erleichterung der täglich wiederkehrend anfallenden Arbeiten der Tagesbetreuung gebracht. Vom Ansatz der Verzweiflung über die „Zettelwirtschaft“ zur schnellen und einfachen Verwaltung liegt jedoch ein langer Weg, dessen Ziel noch nicht erreicht ist. Jede noch kommende Erweiterung der aktuellen Funktionalität führt diesen Weg fort und ist – solange mir als Entwickler genug Zeit dafür zur Verfügung gestellt wird – dringend zu unterstützen.



## 8. Darstellungsverzeichnis

### 8.1. Abbildungen

Abbildung 1: SchüSta. Überblick über Schuljahre und Klassen .....	20
Abbildung 2: Anmeldemaske in SchüSta für neue SchülerInnen .....	21
Abbildung 3: SchüSta Export Menü mit gewählter Exportoption „Textdatei“ .....	22
Abbildung 4: SchüSta Export Dialog .....	24
Abbildung 5: Stammdaten der Klassen in Untis .....	25
Abbildung 6: Untis Export Menü mit einer Vielzahl an Exportformaten .....	27
Abbildung 7: Untis XML Import/Export Dialog .....	28
Abbildung 8: Allgemeine Ordnerstruktur von CakePHP .....	32
Abbildung 9: ER Modellierung nach Peter Chen (bereits vorbereitet für „Sokrates“) ....	35
Abbildung 10: Semesterverwaltung .....	37
Abbildung 11: Dateiverwaltung .....	38
Abbildung 12: Importmöglichkeiten (bereits mit Import aus „Sokrates“) .....	39
Abbildung 13: Schulfreie Tage .....	40
Abbildung 14: Übersicht aller Anmeldungen .....	41
Abbildung 15: Anmeldung erstellen/bearbeiten .....	42
Abbildung 16: Anmeldeübersicht eines/einer SchülerIn .....	43
Abbildung 17: Notiz hinzufügen .....	44
Abbildung 18: Automatische Erstellung der Anmeldestatistik für den Landesschulrat	45
Abbildung 19: Preise für Nachmittagsbetreuung .....	46
Abbildung 20: Auswahl der zu druckenden Anwesenheitsliste .....	47
Abbildung 21: Gedruckte Anwesenheitsliste der Nachmittagsbetreuung .....	47

### 8.2. Diagramme

Diagramm 1: Ablauf NabeTool .....	37
-----------------------------------	----



## 9. Literaturverzeichnis

ASCII. (01. 11 2001). Abgerufen am 14. 08 2015 von Wikipedia:

<http://en.wikipedia.org/wiki/ASCII>

bit Group GmbH. (2015). SOKRATES Web.

Bundeskanzleramt Österreich. (14. 08 2015). *Bundesgesetz über die Dokumentation im Bildungswesen Bildungsdokumentationsgesetz*. Abgerufen am 14. 08 2015 von Rechtsinformationssystem:

<http://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20001727>

Bundeskanzleramt Österreich. (14. 08 2015). *Bundesgesetz über die Ordnung von Unterricht und Erziehung in den im Schulorganisationsgesetz geregelten Schulen (Schulunterrichtsgesetz - SchUG)*. Abgerufen am 14. 08 2015 von Rechtsinformationssystem:

<http://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10009600>

Bundeskanzleramt Österreich. (14. 08 2015). *Bundesgesetz über die Unterrichtszeit an den im Schulorganisationsgesetz geregelten Schularten (Schulzeitgesetz 1985)*.

Abgerufen am 14. 08 2015 von Rechtsinformationssystem:

<http://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10009575>

Bundeskanzleramt Österreich. (14. 08 2015). *Bundesgesetz vom 25. Juli 1962 über die Schulorganisation*. (B. Österreich, Herausgeber) Abgerufen am 14. 08 2015 von Rechtsinformationssystem:

<http://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10009265>

Bundesministerium für Bildung und Frauen. (05. 03 2015). *Die ganztägige Schule*.

Abgerufen am 14. 08 2015 von <https://www.bmbf.gv.at/schulen/gts/index.html>

Bundesministerium für Bildung und Frauen. (10. 11 2014). *Mittagsbetreuung*.

Abgerufen am 14. 08 2015 von

<https://www.bmbf.gv.at/schulen/gts/mittagsbetreuung/index.html>

Bundesministerium für Bildung und Frauen. (15. 08 2015). *UPIS*. Abgerufen am 14. 08 2014 von <http://www.upis.at/>

Bundesministerium für Unterricht, Kunst und Kultur. (12. 9 2012). *Tagesbetreuung*.

Abgerufen am 20. 10 2012 von

<http://www.bmukk.gv.at/schulen/tagesbetreuung/index.xml>

Cake Software Foundation, Inc. (kein Datum). *CakePHP: the rapid development php framework*. Abgerufen am 19. 10 2012 von <http://cakephp.org/>

Cake Software Foundation, Inc. (2012). *Installation*. Abgerufen am 03. 01 2013 von <http://book.cakephp.org/2.0/en/installation.html>

Chen, P. (März 1976). The Entity-Relationship Model-Toward a Unified View of Data. *ACM Transactions on Database Systems, Vol. 1, No. 1.* , 9-36.

Debian Project. (kein Datum). *Debian -- The Universal Operating System* . Abgerufen am 19. 10 2012 von <http://www.debian.org/>

Gmasz, P. (27. 9 2012). SchüSta. (7.4.4.5) .

Gruber & Petters GmbH. (2012). Untis 2012.

Gruber & Petters GmbH. (2012). WebUntis 2012.

Microsoft. (14. 08 2015). Excel 2010.

Newline. (02. 06 2003). Abgerufen am 04. 01 2013 von Wikipedia: <http://en.wikipedia.org/wiki/Newline>

Oracle. (kein Datum). *MySQL :: The world's most popular open source database*. Abgerufen am 19. 10 2012 von <http://www.mysql.com/>

PHP Group. (kein Datum). *PHP: Hypertext Processor*. Abgerufen am 19. 10 2012 von <http://www.php.net>

Shafranovich, Y. (10 2005). *RFC 4180*. (T. I. Society, Herausgeber) Abgerufen am 14. 08 2015 von <http://tools.ietf.org/html/rfc4180>

The Apache Software Foundation. (1. 12 1995). *Apache HTTP Server Project*. Abgerufen am 19. 10 2012 von <http://httpd.apache.org/>

W3C, XML Core Working Group. (8 2006). *Extensible Markup Language (XML) 1.1 (Second Edition)*. (X. C. Group, Herausgeber) Abgerufen am 20. 10 2012 von <http://www.w3.org/TR/2006/REC-xml11-20060816/>

## **10. Anhang**

### **10.1. Abstract**

Diese Arbeit beschäftigt sich mit der Erstellung einer effizienten, sicheren und stabilen Anwendung zur Unterstützung administrativer Tätigkeiten im Zusammenhang mit der Nachmittagsbetreuung in Schulen. Durch den zeitlichen Druck im Schulalltag sowie den chronischen Geldmangel ergeben sich wichtige zu erfüllende Randbedingungen.

Durch meine Beschäftigung an einer Schule erfahre ich immer mehr, dass die Übernahme von Verwaltungstätigkeiten vielfältigster Art durch LehrerInnen nicht nur eine dringende und unumgängliche Maßnahme ist, den Schulalltag mit vorhandenen Mitteln zu ermöglichen, sondern leider auch immer öfter eine ehrenamtliche Tätigkeit, welche in Schulpausen oder in der Freizeit zu bewältigen ist. Die Kombination aus Zeitmangel, geforderter Flexibilität und unbekannter Materie ist oft das größte Problem.

Die Einteilung und Verwaltung der Nachmittagsbetreuung an unserer Schule ist ein Beispiel für eine zeitraubende Tätigkeit, welche die zuständige Lehrkraft leider zu oft an ihre zeitlichen Grenzen bringt. Da unsere Schule im Aufbau begriffen ist und somit jedes Jahr mehr Schüler und Schülerinnen den Standort besuchen, Mittagessen gehen wollen und nachmittags beaufsichtigt werden müssen, wird die Verwaltung zusehends immer umfangreicher. Dazu kommt die Lage der Schule mitten im Wienerwald mit allen verkehrstechnischen Schwierigkeiten. Unabhängig von den gegebenen Randbedingungen ist es außerdem notwendig, auf Änderungen im Stundenplan und unerwartete Ereignisse wie Krankheit schnell zu reagieren.

Ziel ist eine einfache, intuitiv zu bedienende Software, die einer Lehrperson ohne großen Mehraufwand die Administration der gesamten Nachmittagsbetreuung ermöglicht. Der Aufbau des Endproduktes soll eine problemlose Adaptierung an Abläufe in anderen Schulen zulassen.

## **10.2. Lebenslauf**

### **Ausbildung**

1990 – 1994	Volkschule 1130 Wien, Auhofstraße 49
1994 – 2002	Bundesgymnasium Wien 13 1130 Wien, Fichtnergasse 15
2002 - 2008	Universität Wien, Technische Universität Wien Bakkalaureatsstudium Informatikmanagement
seit 2008	Universität Wien Magisterstudium Informatikmanagement

### **Berufliche Tätigkeiten**

2000 - 2008	Mitarbeiter in der IT Medizinische Universität Wien, Klinisches Institut für Hygiene und Medizinische Mikrobiologie
2003	Grundwehrdienst Österreichisches Bundesheer, Militärstreifen-Lehrkompanie
2005 – 2010	Tutor für LVA „Einführung in die Praxis der Computersysteme“ Universität Wien, Fakultät für Mathematik
2008 – 2010	Wissenschaftlicher Assistent und Leitung der IT Medizinische Universität Wien, Institut für Hygiene und Angewandte Immunologie
seit 2010	Vertragslehrer für Informatik Landesschulrat Niederösterreich, BG BRG Purkersdorf Expositur Tullnerbach