



PROCESSREDESIGN

NMS Pabneukirchen

Diplomarbeit

Höhere Technische Bundeslehranstalt für Informatik
Fachgebiet Betriebswirtschaft und Management

Eingereicht von:	Patricia Berger Tanja Buchmair
Betreuungslehrer:	DI Christian Reisinger
Auftraggeber:	NMS Pabneukirchen Dir. Doris Kurus
Eingereicht am:	13. Mai 2015





Eidesstattliche Erklärung

Perg, am 13. Mai 2015

Ich erkläre hiermit an Eides statt, dass ich die vorliegende wissenschaftliche Arbeit selbstständig und lediglich mit den angegebenen Hilfsmitteln erstellt habe. Es wurden keine weiteren Mittel oder Ressourcen, als die in dieser Dokumentation genannten, verwendet. Die aus fremden Quellen direkt oder indirekt übernommenen Formulierungen und Zitate sind als solche kenntlich gemacht und befinden sich auf den letzten Seiten dieses Dokuments.

Patricia Berger

Tanja Buchmair





Danksagung

Wir möchten uns ganz herzlich bei allen Bedanken, die uns bei der Durchführung der Diplomarbeit unterstützt haben, vor allem Freunden, Familien und den geschätzten Klassenkollegen.

Danken möchten wir in erster Linie unseren Betreuungslehrer DI Christian Reisinger, der uns während der gesamten Durchführung mit seiner Fachkompetenz und seinen Ratschlägen zur Verfügung stand.

Weiters möchten wir Frau Doris Kurus, Direktorin der NMS Pabneukirchen danken, die uns diese Diplomarbeit ermöglicht hat. Sie stand uns bei Fragen immer zur Verfügung und gab uns hilfreiche Einblicke in das Schulsystem.

Herzlichen Dank!





Inhaltsverzeichnis

1. EINLEITUNG	1
1.1. KURZBESCHREIBUNG	1
1.2. ABSTRACT.....	2
2. PROJEKTDEFINITION	3
2.1. AUFTRAGGEBER	4
2.2. PROJEKTTEAM	5
2.2.1. <i>Patricia Berger</i>	5
2.2.2. <i>Tanja Buchmair</i>	5
2.3. PROJEKTUMFELD	6
2.3.1. <i>Schule</i>	6
2.3.2. <i>Betreuungslehrer</i>	6
2.3.3. <i>Ansprechpartner</i>	6
2.3.4. <i>Hierarchie</i>	6
3. AUSGANGSLAGE UND ZIELSETZUNGEN	7
3.1. AUSGANGSSITUATION	8
3.2. PROBLEMBEREICH	11
3.3. ZIELSETZUNG	12
3.4. BESCHREIBUNG DES SYSTEMS	13
4. FUNKTIONALE ANFORDERUNGEN	17
4.1. FUNKTIONALE ANFORDERUNGEN.....	18
4.1.1. <i>Legende für USE-CASE Diagramme:</i>	18
4.1.2. <i>Stundenplanerstellung</i>	18
4.1.3. <i>Vertretungsplanerstellung</i>	19
4.1.4. <i>Webpräsenz</i>	19
4.2. NICHT-FUNKTIONALE ANFORDERUNGEN	20
4.2.1. <i>Benutzbarkeit</i>	20
4.2.2. <i>Zuverlässigkeit und Stabilität</i>	20
4.3. MENGENGERÜST.....	20
4.3.1. <i>Stammdaten</i>	20
5. PROJEKTPLANUNG UND -MANAGEMENT	21
5.1. PROJEKTABLAUF.....	22
5.1.1. <i>Vorgehensmodell SCRUM</i>	23
5.1.2. <i>Vorgehensweise</i>	24
5.1.2.1. <i>Kommunikationsmodell</i>	24
5.1.2.2. <i>Abnahmeverfahren</i>	25
5.1.3. <i>Zeitplanung</i>	25
5.1.3.1. <i>Start- & Endzeitpunkt</i>	25
5.1.3.2. <i>Gantt-Zeitplan</i>	25
5.1.3.3. <i>Aufwandsschätzung</i>	27
5.1.4. <i>Projektstrukturplan</i>	28
5.1.5. <i>Verantwortungsmatrix</i>	29
5.2. RISIKOMANAGEMENT.....	30
5.3. RESSOURCENPLANUNG	32



5.3.1.	Hardware	32
5.3.2.	Software.....	33
5.3.3.	Menschliche Ressourcen	34
5.3.4.	Stundenstatistik	34
5.3.5.	Budget.....	34
5.4.	PROJEKTKOSTEN.....	35
5.4.1.	Kosten der Arbeitsleistung.....	35
5.4.2.	Lizenzkosten.....	36
5.4.3.	Sonstige Kosten.....	36
5.4.4.	Gesamtkosten.....	37
5.5.	LEISTUNGSUMFANG	38
5.6.	LIEFERUMFANG	39
6.	GRUNDLAGEN UND TECHNOLOGIEN	41
6.1.	HTML	42
6.2.	PHP	42
6.3.	MYSQL	42
6.4.	DROPBOX	42
6.5.	MICROSOFT OFFICE 2013	43
6.6.	GANTT PROJECT.....	43
6.7.	WORDPRESS	43
6.8.	JQUERY.....	43
6.9.	SCSS.....	43
6.10.	JAVASCRIPT.....	44
6.11.	MYSQL WORKBENCH.....	44
6.12.	SOURCE TREE	44
6.13.	FIREFOX FIREBUG	44
6.14.	APACHE HTTP SERVER	45
6.15.	BRACKETS	45
6.16.	VMWARE WORKSTATION	45
6.17.	UBUNTU	45
6.18.	ECLIPSE LUNA PHP DEVELOPER	45
7.	UMSETZUNG	47
7.1.	SYSTEMARCHITEKTUR.....	48
7.1.1.	TimeTable	48
7.1.2.	TimeOwl (Vertretungsplan)	48
7.1.3.	Webseite.....	48
7.2.	PROGRAMMARCHITEKTUR.....	49
7.2.1.	Index.php	50
7.2.2.	Controller Factory	51
7.2.3.	Controller.....	52
7.2.4.	Repository Factory	53
7.2.5.	Repository.....	54
7.2.6.	Datenbank	55
7.2.7.	View	55
7.2.8.	Template.....	55
7.2.9.	Model.....	56
7.3.	DATENMODELL	57
7.3.1.	Beschreibung der Datenbanktabellen	58
7.4.	DATENEXPORT (TIMETABLE) UND DATENIMPORT (TIMEOWL)	61



7.4.1. Export.....	61
7.4.2. Import.....	62
7.4.3. Stundenplananzeige laden.....	65
8. EVALUIERUNG.....	69
8.1. ERGEBNIS	70
8.1.1. Stundenplan.....	70
8.1.2. Vertretungsplan.....	71
8.1.3. Webpräsenz (Homepage).....	72
8.2. HINDERNISSE UND PROBLEME.....	73
8.2.1. Schulischer Alltag.....	73
8.2.2. Einhaltung der geplanten Stunden	73
8.3. WERTSCHÖPFUNG	74
8.4. RESÜMEE.....	74
9. GLOSSAR.....	75
10. VERZEICHNISSE	77
10.1. ABBILDUNGSVERZEICHNIS	77
10.2. TABELLENVERZEICHNIS	78
10.3. PROGRAMMCODEVERZEICHNIS	78
10.4. LITERATURVERZEICHNIS	80
11. ANHANG	81
11.1. ABNAHMEVERTRAG	81
11.2. STATUSBERICHT	82





1. Einleitung

1.1. KURZBESCHREIBUNG

Die Neue Mittelschule Pabneukirchen ist eine kleine, familiäre Schule, die derzeit ca. 20 Lehrkräfte beschäftigt. Der Verwaltungsaufwand ist auch in einer Schule dieser Dimension sehr hoch. Das Schuljahr beginnt für die Lehrkräfte meist ein paar Wochen früher, da die Stundenpläne festgelegt werden müssen. Auch während dem Jahr müssen Vertretungsstunden geplant werden, da hin und wieder eine Lehrperson krank wird oder eine Klasse einen Ausflug vor sich hat. Noch viele weitere Prozesse finden im Hintergrund statt, die ein Schüler oder ein Außenstehender gar nicht mitbekommt.

Daher ist das Ziel dieser Diplomarbeit, eine Prozessverbesserung im Rahmen des Organisationskreislaufes durchzuführen, um diese Abläufe zu optimieren und passende Unterstützungen zu finden.

So sollen die Probleme identifiziert und ein entsprechendes Konzept erstellt werden. Dadurch soll ein verbesserter Ablauf entstehen, um die Verwaltungstätigkeiten der Neuen Mittelschule Pabneukirchen zu unterstützen.



1.2. ABSTRACT

The new secondary school Pabneukirchen is a small and friendly school, which currently has about 20 teachers. The administrative expenditure is very high even in a school of this size. The school year starts for teachers usually a few weeks earlier because the timetables need to be set for the students. Also during the year, cover lessons must be planned as an occasional teacher is ill or a class has a trip before. Many other processes take place in the background, which does not even notice a student or an outsider.

Therefore, the aim of this thesis is to carry out a process improvement within the organization circuit in order to optimize these processes and find appropriate supports.

The problems should be identified and a suitable draft should be provided. An improved expiry should originate from it to support the administrative activities of the new secondary school Pabneukirchen.



2. Projektdefinition

WORUM GEHT ES IN DIESEM KAPITEL?

2.1. AUFTRAGGEBER	4
2.2. PROJEKTTEAM	5
2.2.1. <i>Patricia Berger</i>	5
2.2.2. <i>Tanja Buchmair</i>	5
2.3. PROJEKTUMFELD	6
2.3.1. <i>Schule</i>	6
2.3.2. <i>Betreuungslehrer</i>	6
2.3.3. <i>Ansprechpartner</i>	6
2.3.4. <i>Hierarchie</i>	6

Um die alltäglichen Prozesse im Schulablauf zu vereinfachen und effizienter zu gestalten und die Schulpräsenz zu verbessern, kontaktierte Dir. Doris Kurus die beiden Schülerinnen, Patricia Berger und Tanja Buchmair, um eine mögliche Kooperation mit der HTL Perg einzugehen, um dieses Vorhaben als mögliche Diplomarbeit zu realisieren.

Daraufhin wurde das Projekt „ProcessRedesign of NMS Pabneukirchen“ ins Leben gerufen. Ziel ist es, eine Prozessverbesserung im Rahmen der Abwicklung des Organisationskreislaufes durchzuführen. Nach der Identifikation der Problembereiche, in welchem redundante Abläufe stattfinden, wird eine Ist-Analyse durchgeführt sowie nach Lösungen gesucht und das passende Soll-Konzept entwickelt. Dieses beinhaltet eine Einführung einer Software für die Stundenplanerstellung sowie für die Vertretungsplanung, auch eine Einführung eines Content Management Systems für eine neue Homepage.

Das Projekt wird im Rahmen einer Diplomarbeit an der HTL in Perg mit Schwerpunkt Informatik im Schuljahr 2014/2015 durchgeführt und von DI Christian Reisinger betreut.

Der Großteil des folgenden Inhalts wurde aus dem Projekthandbuch entnommen.



2.1. AUFTRAGGEBER

Auftraggeber dieser Diplomarbeit ist die Neue Mittelschule Pabneukirchen¹ mit Frau Dir. Doris Kurus als Hauptverantwortliche.



Abbildung 1: Logo Neue Mittelschule Pabneukirchen

Die Neue Mittelschule Pabneukirchen besteht derzeit aus ungefähr 20 Lehrkräften und diversen Schulpersonal. Auch eine eigene Schulküche gehört zu der kleinen, familiären Schule. Die Lehranstalt wird von Frau Doris Kurus geleitet.



Abbildung 2: Neue Mittelschule Pabneukirchen

¹ <http://www.nms-pabneukirchen.at/>, 01.05.2015



2.2. PROJEKTTEAM

Diese Diplomarbeit wird im Rahmen der Matura an der HTL in Perg im Schuljahr 2014/2015 von den nachstehenden Diplomanden durchgeführt:

2.2.1. PATRICIA BERGER

<u><i>Persönliche Daten</i></u>	
Name:	Patricia Berger
Adresse:	Schweinberg 7a 3313 Wallsee
Klasse:	5AHIF
Geburtsdatum:	03.05.1996
E-Mail:	patricia.berger@wavenet.com

A portrait of Patricia Berger, a young woman with long, light-colored hair, wearing a green knitted beanie and a black top. She is looking slightly to the right of the camera.

Abbildung 3: Patricia Berger

2.2.2. TANJA BUCHMAIR

<u><i>Persönliche Daten</i></u>	
Name:	Tanja Buchmair
Adresse:	Linden 26 4352 Klam bei Grein
Klasse:	5AHIF
Geburtsdatum:	29.09.1994
E-Mail:	tanjabuchmair@gmail.com


A portrait of Tanja Buchmair, a young woman with long, straight brown hair, wearing a dark top. She is smiling slightly and looking towards the camera.

Abbildung 4: Tanja Buchmair



2.3. PROJEKTUMFELD

2.3.1. SCHULE

Diese Diplomarbeit wurde im Rahmen der Matura an der HTL in Perg, Schwerpunkt Informatik, im Schuljahr 2014/2015 durchgeführt.

2.3.2. BETREUUNGSLEHRER

Betreut wurde diese Diplomarbeit von unserem Lehrer DI Christian Reisinger. Er hat eine beratende Funktion und überprüft die Fortschritte der SCRUM-Prozesse (*siehe Kapitel 5.1.1*).

2.3.3. ANSPRECHPARTNER

In diesem Fall gab es hauptsächlich zwei Ansprechpartner:

- Dir. Doris Kuras
- Monika Berger

2.3.4. HIERARCHIE

Folgende Grundstruktur ergibt sich aus den teilhabenden Personen dieser Diplomarbeit:

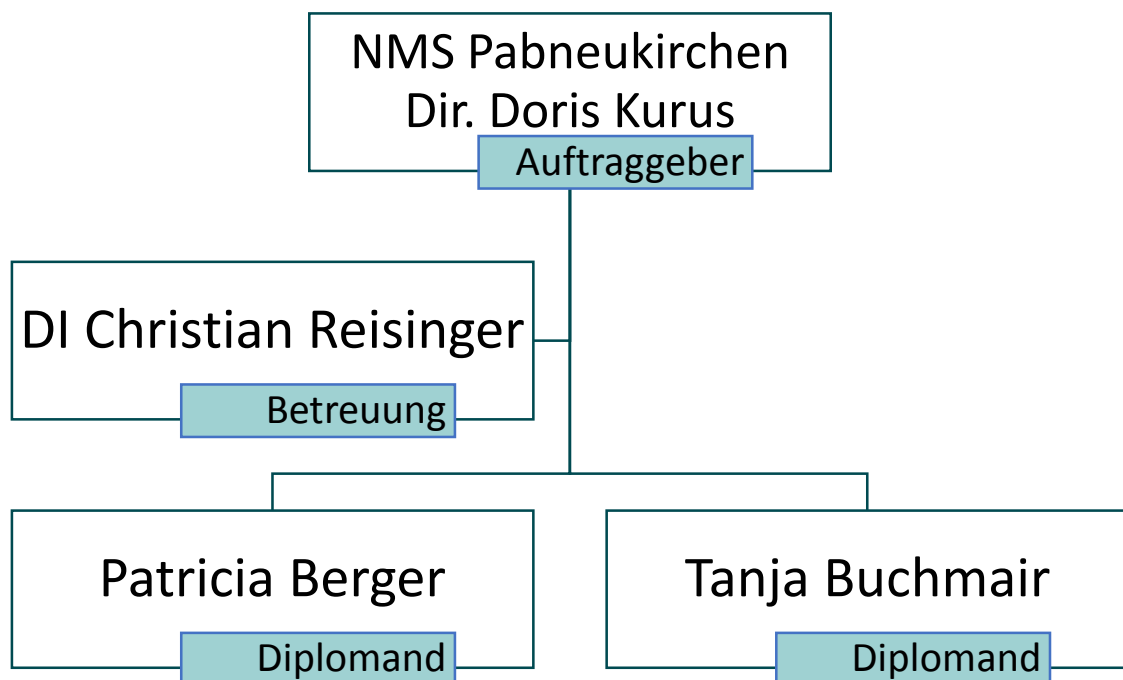


Abbildung 5: Projekthierarchie



3. Ausgangslage und Zielsetzungen

WORUM GEHT ES IN DIESEM KAPITEL?

3.1. AUSGANGSSITUATION	8
3.2. PROBLEMBEREICH	11
3.3. ZIELSETZUNG	12
3.4. BESCHREIBUNG DES SYSTEMS	13

Dieses Kapitel handelt von der Ausgangssituation des Auftraggebers und zeigt die gegenwärtigen Probleme auf. Um diesen entgegenzuwirken, müssen verschiedene Ziele hinsichtlich der Geschäftsprozesse und des Projektes definiert werden. Zum Abschluss wird das zu entwickelnde System beschrieben.

Der Inhalt dieses Kapitels wurde fast ausschließlich aus dem Pflichtenheft entnommen. Dieses Dokument bildet die Vertragsgrundlage zwischen Auftraggeber und Kunden. Im Pflichtenheft werden alle Anforderungen des Kunden festgehalten und auch die Lösung durch den Auftragnehmer sowie dessen Realisierung.



3.1. AUSGANGSSITUATION

Die derzeit durchgeführten Prozesse der Neuen Mittelschule Pabneukirchen sind mit enormen Aufwand verbunden. Um die derzeitigen Schwachstellen aufzuzeigen, wird eine Prozessverbesserung im Rahmen der Abwicklung des Organisationskreislaufes durchgeführt und die einzelnen Bereiche identifiziert und analysiert. Dazu wird folgender Kreislauf herangezogen:

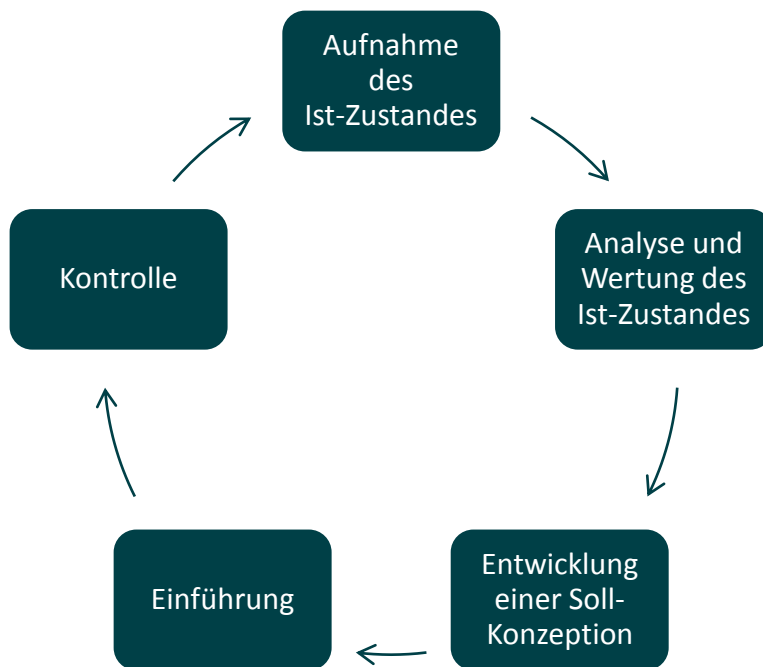


Abbildung 6: Organisationskreislauf

Aufnahme Ist-Zustandes

Um die Schwachstellen in den aktuellen Vorgängen aufzuzeigen, müssen diese zuerst in einzelne Bereiche definiert werden. Dazu muss der derzeitige Ablauf beobachtet werden. Der gesamte Prozess wird in kleine Bereiche untergliedert, damit diese voneinander abgeschottet analysiert sowie bewertet werden können.

Analyse und Wertung Ist-Zustandes

In diesem Schritt wird festgelegt, in welchen Prozessen eine Neugestaltung auch eine Verbesserung mit sich bringt. Nach der Bewertung und Analyse stellt man die einzelnen Bereiche gegenüber. Es wird analysiert, welche Prozesse häufig durchgeführt werden und eine effektive Verbesserung erzielt werden kann. Durch diese Bewertung haben wir drei Bereiche gewählt, in denen eine Neugestaltung sinnvoll sowie auch realisierbar ist. Im nachfolgenden werden diese Prozesse genau beschrieben und grafisch dargestellt.



- **Stundenplanerstellung / Vertretungsplan**

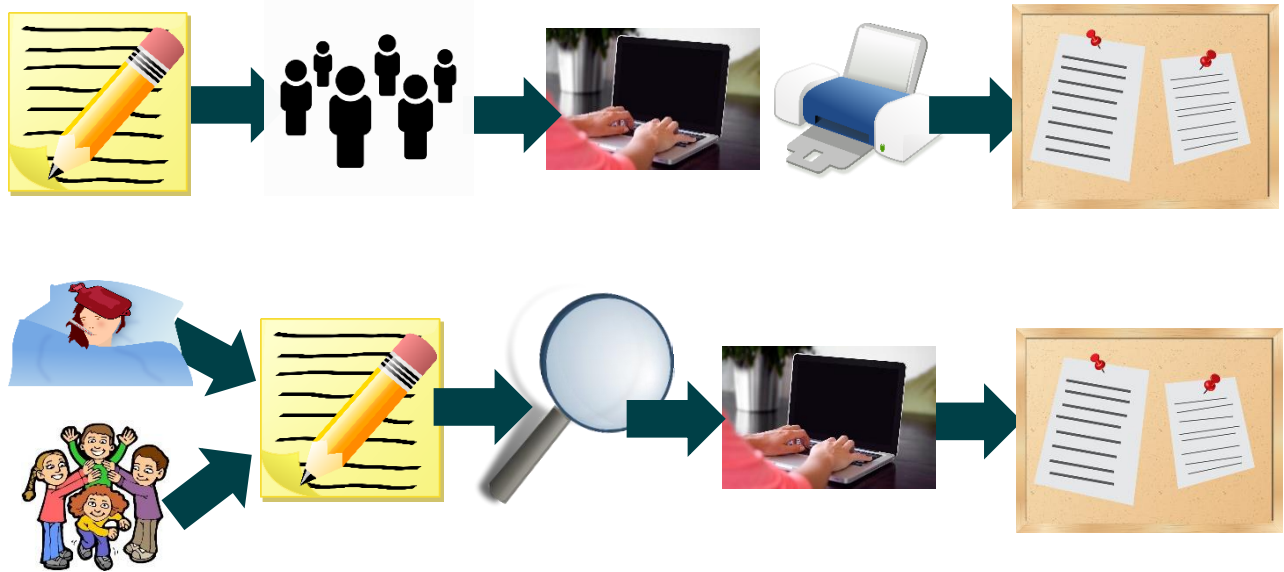


Abbildung 7: Prozesse des Ist-Zustandes

Die Neue Mittelschule Pabneukirchen wickelt derzeit die Stundenplanentwicklung per Hand ab. Auch die Einteilung der Vertretungsstunden wird handschriftlich festgelegt, elektronisch erfasst und im Lehrerzimmer ausgehängt, damit die betroffenen Lehrkräfte informiert sind und sich vorbereiten können.

Die Stundenpläne werden nach der handschriftlichen Erstellung digitalisiert und anschließend ausgehängt. Somit sind immer viele Dokumente im Umlauf, da bei jeder Änderung ein neuer Druck notwendig ist. Somit sind die einzelnen Lehrkräfte auf die Dokumente angewiesen und müssen ständig diese handschriftlichen Exemplare aktualisieren oder erneuern.

- **Homepage/Internetpräsenz**

Auch der Internetauftritt wurde vernachlässigt, da die Bedienung als Administrator kompliziert ist. Die Homepage entspricht nicht mehr dem heutigen Standard und wirkt durch die derzeitige Struktur sehr unübersichtlich.



Entwicklung Sollkonzeption

Nachdem die einzelnen Bereiche feststehen, welche erneuert werden sollen, muss ein passendes Konzept entwickelt werden. Die Prozesse werden neu strukturiert und alternative Systeme für die Optimierung gesucht oder gegebenenfalls entwickelt.

- **Stundenplanerstellung**

Da die derzeitigen Prozesse per Hand durchgeführt werden und somit mit enormen Dokumentenaufwand verbunden sind, beinhaltet das Konzept eine Einführung eines EDV-Systems für die Erstellung der Stundenpläne.

Die genaue Beschreibung dieses Soll-Konzepts ist im *Punkt 3.4* zu finden.

- **Vertretungsplan**

Der Prozess der Planung der Vertretungen wird per Hand durchgeführt. Sobald die Information über eine Abwesenheit vorhanden ist, werden die fehlenden Lehrkräfte notiert. Deren Unterrichtsstunden werden im Stundenplan gesucht und auf einen Handzettel festgehalten. Im nächsten Schritt werden die verfügbaren Lehrkräfte notiert und die Vertretung festgelegt. Diese werden anschließend digitalisiert und im Konferenzzimmer ausgehängt, damit die Lehrkräfte informiert sind. Das Konzept sieht eine Einführung eines EDV-Systems für die Erstellung und Verwaltung der Vertretungspläne.

Die genaue Beschreibung dieses Soll-Konzepts ist im *Punkt 3.4* zu finden.

- **Homepage/Internetpräsenz**

Die Internetpräsenz wird derzeit niedrig priorisiert, somit ist dieser Prozess veraltet, da dieser fast nie zum Einsatz kommt. Diese Vernachlässigung kommt aus der unübersichtlichen Administration der Homepage. Der damit verbundene Aufwand ist enorm, da die Eingabe erschwert ist durch die unübersichtliche Benutzeroberfläche sowie die Hochladezeiten oftmals mehrere Tage andauern. Das Konzept sieht daher vor, die alte Homepage zu verwerfen und ein benutzerfreundliches sowie einfaches CMS einzuführen

Die genaue Beschreibung dieses Soll-Konzepts ist im *Punkt 3.4* zu finden.

Einführung

Nach der Erstellung des Soll-Konzepts muss dieses auch eingeführt werden und das alte vorläufig ersetzen. Da das neue System für die meisten Mitarbeiter unbekannt ist, muss eine Einschulung stattfinden, um diesen Prozess durchführen zu können sowie das alltägliche Geschäft nicht zu beeinträchtigen. Diese Einführungsphase findet in einem bestimmt festgelegten Zeitintervall statt, um die Fortschritte zu beobachten.



Kontrolle

Nachdem das System bzw. die neuen Prozesse eingeführt worden sind, muss natürlich kontrolliert werden, ob diese neue Struktur die gewünschten Ergebnisse liefert. Hier entscheidet sich, ob das neue System beibehalten wird oder wieder auf den vorherigen Ablauf umgestellt wird. Falls dies der Fall ist, beginnt der Kreislauf wieder von vorne. Wenn sich dieser neue Prozess bewährt, wird dieser beibehalten und endgültig eingeführt.

3.2. PROBLEMBEREICH

Aufgrund der im *Punkt 3.1* beschriebener Ausgangslage kristallisieren sich folgende Probleme in der Neuen Mittelschule Pabneukirchen heraus:

REDUNDANTE ABLÄUFE

Viele Vorgänge, wie das Erstellen und Verwalten von Vertretungsplänen, werden durch die ausführenden Kräfte redundant und häufig durchgeführt. Minimale Änderungen sind in diesem Prozess mit hohem Aufwand verbunden.

INTERNETPRÄSENZ

Die Präsenz im Internet wird niedriger priorisiert, da die Homepage veraltet ist sowie die Administration erhebliche Aufwände verursacht.



3.3. ZIELSETZUNG

Ziel dieser Diplomarbeit ist es, die Prozesse im Rahmen der Abwicklung des Organisationskreislaufes zu verbessern. Nach der Identifikation der Bereiche Stundenplanerstellung, Vertretungsplanung, Homepage(-aktualisierung) und der genaueren Ist-Analyse der Situation wird nach Lösungen gesucht und das passende Soll-Konzept entwickelt.

Dieses beinhaltet eine Einführung einer bereits bestehenden Software für die Stundenplanerstellung, eine neue Software für die Erstellung der Vertretungspläne, speziell angepasst für die Neue Mittelschule Pabneukirchen, sowie die Einführung eines CMS².

Diese angeführten Komponenten bilden das System „PRoNMSP“.

Grundsätzlich wird zwischen Geschäfts- und Projektzielen unterschieden.

Geschäftsziele

- Zeitersparnis bei Erstellung der Stunden- sowie Vertretungspläne
- Hochladen von neuen Artikel durch Lehrer der NMS
- Verfügbarkeit der Artikel direkt nach dem Hochladen

Abbildung 8: Geschäftsziele

Ziele des Projekts

- EDV-unterstützte Stunden- und Vertretungsplanung
- Neu gestalteter Webauftritt unter Verwendung eines CM-Systems

Abbildung 9: Ziele des Projekts

² Content Management System



3.4. BESCHREIBUNG DES SYSTEMS

Im Rahmen der Abwicklung des Organisationskreislaufes wurden aus der Analyse und Bewertung des Ist-Zustandes folgende Bereiche festgelegt, in welchen die Prozesserneuerung stattfinden wird. Im Folgenden werden diese genauer beschrieben und auf die Entwicklung des Soll-Konzepts eingegangen. Diese Komponenten bilden das neue System „PRoNMSP.

STUNDENPLANERSTELLUNG / VERTRETUNGSPLANERSTELLUNG

Die derzeitige Erstellung des Stundenplans erfolgt grundsätzlich auf dem Papier. Diese Planung beginnt meist zwei Wochen vor Schulbeginn und nimmt auch noch die ersten zwei Schulwochen in Anspruch. Nachdem das Grobkonzept erstellt ist, wird der Stundenplan speziell für jede Lehrkraft per Hand niedergeschrieben. Daraufhin werden noch die letzten Kleinigkeiten beseitigt bzw. geändert und der Stundenplan schließlich digitalisiert. Sprich, dieser wird einerseits für die Klasse sowie für die Lehrkräfte geschrieben.

Die Vertretungsplanung erfolgt derzeit per Hand auf dem Papier. Wenn eine Lehrkraft verhindert ist oder sich eine Klasse auf Exkursion befindet, ist dies meist früher bekannt. Mit diesen Informationen werden auf dem Stundenplan die zu vertretenden Unterrichtsstunden notiert. Anschließend werden die Lehrkräfte herausgesucht, welche in diesen Stunden zur Verfügung stehen. Abschließend wird auf einem handschriftlichen Zettel die finale Belegung festgehalten und auf der Pinnwand ausgehängt, damit die Lehrkräfte informiert sind.

Durch die gründliche Analyse des Ist-Zustands haben wir beschlossen, diesen Bereich aufzugreifen und den Prozess zu erneuern. Das dazugehörige Soll-Konzept sieht vor, zuerst nach einer bestehenden Software zu suchen, welche den Anforderungen der Neuen Mittelschule gerecht werden.

Aufgrund der Benützung des Systems WebUntis, welches ein Werkzeug der Stundenplanerstellung mit vielen weiteren Funktionen darstellt, haben wir eine Richtlinie bezüglich solcher Art von Software, da diese in unserer Schule seit mehreren Jahren verwendet wird. Es ist gleichzeitig ein elektronisches Klassenbuch sowie ein Vertretungsplaner.

Obwohl dieses System den Zweck für die Neue Mittelschule Pabneukirchen erfüllen würde, haben wir uns gegen diese Software entschieden. Ausschlusskriterium ist der Punkt Kosten. Diese belaufen sich jährlich auf circa 700€. Für eine Schule mit einer Klassenanzahl von ungefähr 6-8 pro Schuljahr ist solch ein komplexes System überflüssig. Es würde Geld für Funktionen bezahlt werden, welche nicht einmal Verwendung finden. WebUntis ist nur im gesamten Paket erhältlich und nicht funktionsweise.



Soll-Konzept Stundenplan

Unser Betreuungslehrer gab uns einen Hinweis auf eine bestehende Software speziell für die Stundenplanerstellung, welche von einer Lehrkraft der HTL Perg entwickelt wurde. Nach Überprüfung und Absprache mit dem Entwickler werfen wir einen Blick auf die Software und analysieren diese auf die Anforderungen der Neuen Mittelschule Pabneukirchen.

Diese bestehende Software mit dem Namen „*TimeTable*“ wird für die NMS-Pabneukirchen zur Verfügung gestellt. Durch eine einmalige Zahlung darf dieses Programm auf Lebenszeit verwendet werden. Diese wird durch die Diplomanden speziell für den Auftraggeber angepasst. Verwendung findet diese Software für die Erstellung der Stundenpläne. Die einzelnen Gegenstände sind durch eine einfache Eingabemaske erstellt und müssen nur noch in die richtigen Wochenstunden verschoben werden. Kollisionen können durch die elektronische Erstellung vermieden werden, da diese Software den Benutzer darauf hinweist. Der Prozess wird somit enorm beschleunigt.

Soll-Konzept Vertretungsplan

Aufgrund der erfolglosen Suche nach einem vergleichbaren Produkt wie „WebUntis³“ fiel der Entschluss, solch ein System selber zu entwickeln. Diese Software wird speziell für den Auftraggeber entwickelt und nach dessen Anforderungen programmiert.

Der Benutzer kann mithilfe dieser Software die Vertretungsstunden verwalten. Diese ermöglicht es, die Stunden, welche besetzt werden müssen, eine neue Lehrkraft zuzuteilen. Auch Exkursionen und Abwesenheiten von Klassen können angegeben werden sowie die dazugehörigen Lehrkräfte. Das Programm weist den Benutzer darauf hin, welche Stunden durch diese Veranstaltungen ausfallen und werden somit farblich gekennzeichnet. Diese müssen durch den Benutzer nun neu belegt werden. Die Software gibt standardmäßig einen Überblick über die aktuelle Woche und deren Stundenplan sowie den aktuellen Suppliertundenzähler. Etwaige Kollisionen können somit vermieden werden sowie den Aufwand zu minimieren.



HOMEPAGE

Da die derzeitige Homepage eine sehr individuelle und komplexe Administration aufweist, wurde die Internetpräsenz niedrig priorisiert. So wurden keine weiteren Beiträge hochgeladen, da einerseits die Veröffentlichung der Artikel sehr kompliziert ist sowie die Hochladezeiten meist mehrere Tage in Anspruch nehmen oder gar nicht auf der Homepage aufscheinen. Somit wurde dieser Bereich vernachlässigt.

Durch die gründliche Analyse des Ist-Zustands haben wir beschlossen, diesen Bereich aufzugreifen und den Prozess zu erneuern. Das dazugehörige Soll-Konzept sieht vor, die alte Homepage zu verwerfen und ein neues, benutzerfreundliches sowie ein leicht zu bedienendes CMS bereitzustellen.

Durch das enorme Angebot an solchen Systemen führten wir einen Vergleich durch, welcher die drei größten Anbieter eines CMS darstellt. Dieser sieht wie folgt aus:

3 MONSTERS OF CMS Comparison of WordPress, Joomla, and Drupal			COST			MANUAL INSTALLATION TIME		
			Free	Free	Free	5 mins	10 mins	10 mins
POPULARITY			FREE THEMES			EASE OF MODERATION		
Released in 2003 by Matt Mullenweg	Released in 2005, forked from Mambo	Released in 2001 by Dries Buytaert	2,000+	900+	1,800+	★	★★	★★★
POPULARITY			FREE PLUGINS			LEVEL OF "SKILLS" NEEDED		
> 140 million downloads	> 30 million downloads	> 15 million downloads	27,000+	7,000+	24,000+	★	★★	★★
ONE CLICK INSTALLATION AVAILABILITY			UPDATES FREQUENCY			UPDATES FREQUENCY		
✓	✓	✓	42 days	36 days	51 days			

Durch diese Grundlage fällt unsere Entscheidung auf das CMS Wordpress, da dieses derzeit am Stärksten am Markt vertreten ist. Auch die Benutzerfreundlichkeit ist überzeugend sowie die einfache Bedienung der Administration, da diese selbstsprechend und sehr strukturiert ist.

Zugang für die Administration ist nur ausgewählten Personen gewährt. Die hauptverantwortliche Lehrkraft verwaltet den gesamten Inhalt der Homepage. Diese Homepage dient grundsätzlich der Information. Hier werden alle Termine bekannt gegeben sowie aktuelle Beiträge geteilt. Der Grundaufbau sowie die Übernahme alter Inhalte sind bereits durch die Diplomanden erfolgt.





4. Funktionale Anforderungen

WORUM GEHT ES IN DIESEM KAPITEL?

4.1. FUNKTIONALE ANFORDERUNGEN.....	18
4.1.1. <i>Legende für USE-CASE Diagramme:</i>	18
4.1.2. <i>Stundenplanerstellung</i>	18
4.1.3. <i>Vertretungsplanerstellung</i>	19
4.1.4. <i>Webpräsenz</i>	19
4.2. NICHT-FUNKTIONALE ANFORDERUNGEN	20
4.2.1. <i>Benutzbarkeit</i>	20
4.2.2. <i>Zuverlässigkeit und Stabilität</i>	20
4.3. MENGengerüst.....	20
4.3.1. <i>Stammdaten</i>	20

In den folgenden Seiten werden alle Anforderungen sowie Funktionalitäten, welche an die Applikation „PRoNMSP“ gestellt sind, ausführlich behandelt. Für einen erfolgreichen Abschluss des Projekts müssen diese erfüllt werden und weisen dadurch höchste Priorität auf.

Alle Anforderungen werden in diesem Kapitel detailliert beschrieben.

Der Inhalt dieses Kapitels wurde fast ausschließlich aus dem Pflichtenheft entnommen. Dieses Dokument bildet die Vertragsgrundlage zwischen Auftraggeber und Kunden. Im Pflichtenheft werden alle Anforderungen des Kunden festgehalten und auch die Lösung durch den Auftragnehmer sowie dessen Realisierung.



4.1. FUNKTIONALE ANFORDERUNGEN

4.1.1. LEGENDE FÜR USE-CASE DIAGRAMME:

Element	Darstellungsform	Funktion
Akteur	Strichmännchen	Greifen von außen in das System ein
Anwendungsfall (Use-Case)	Ellipse	Stellen ausführbare Aktionen, Funktionen, Anwendungen, usw. dar
Assoziation	Gerade	Visualisiert die Kommunikation zwischen Akteur und Anwendungsfällen
Systemgrenzen	Rechteck	Kennzeichnet die Grenzen des Systems

Tabelle 1: Legende Use-Case Diagramm

4.1.2. STUNDENPLANERSTELLUNG

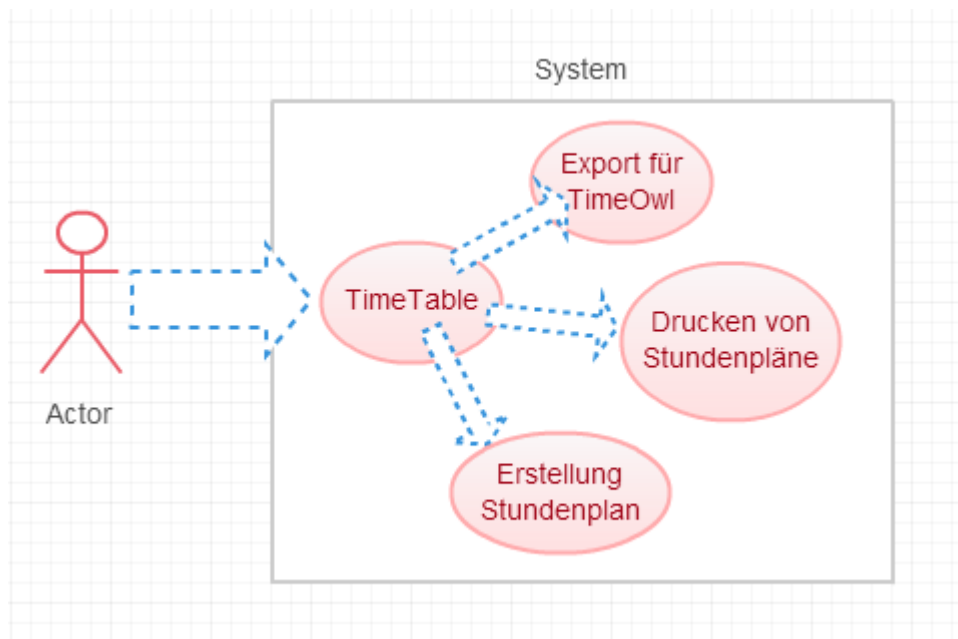


Abbildung 10: Use-Case Stundenplanerstellung



4.1.3. VERTRETUNGSPLANERSTELLUNG

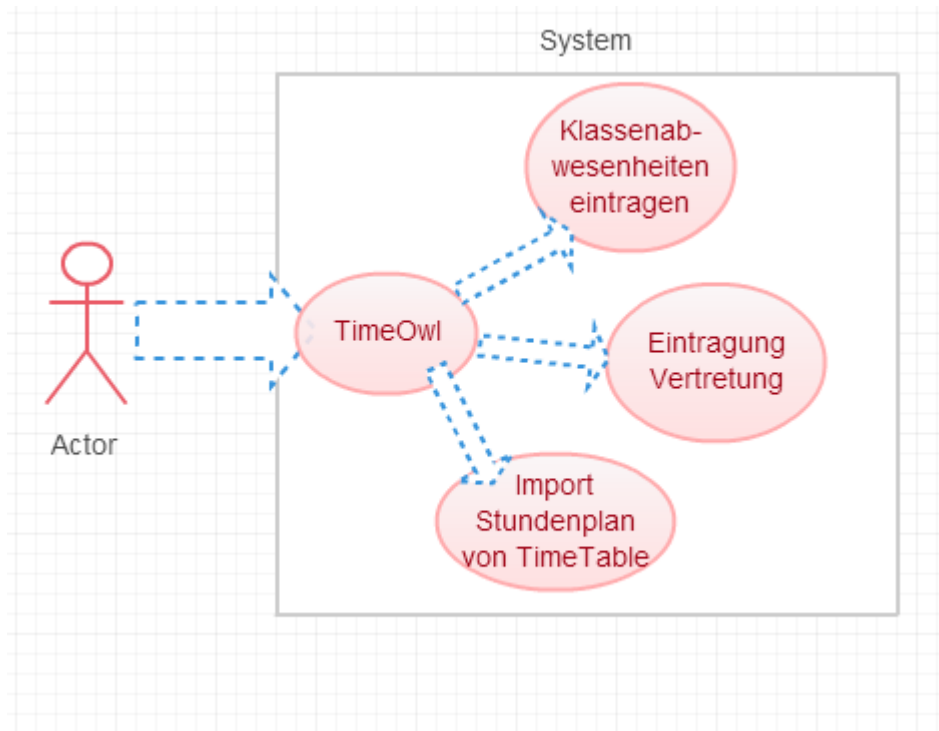


Abbildung 11: Use-Case Vertretungsplanerstellung

4.1.4. WEBPRÄSENZ

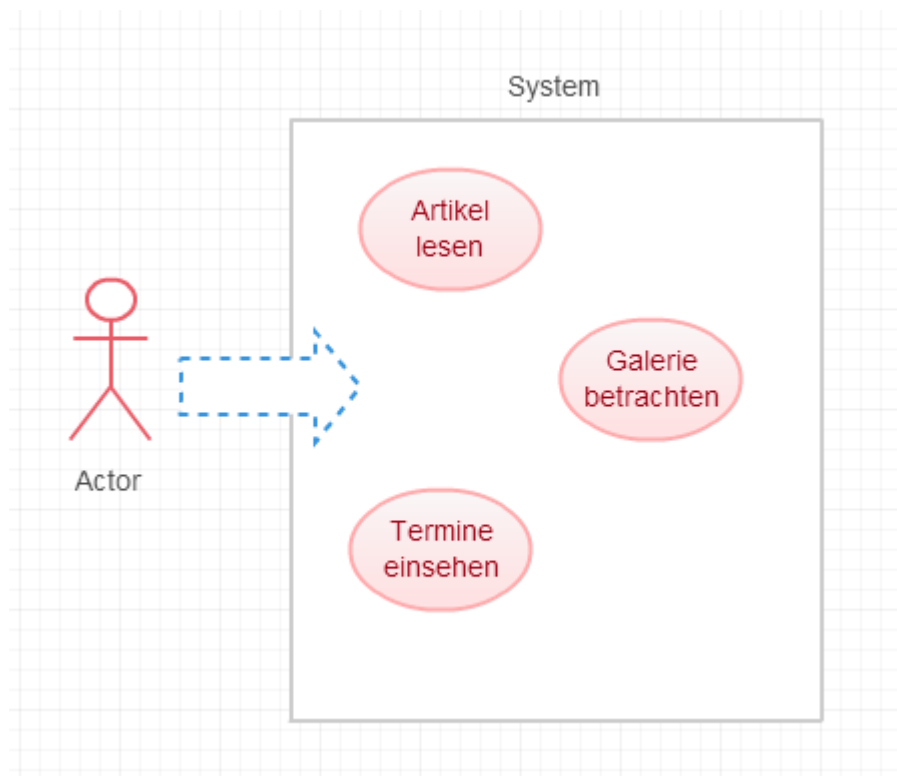


Abbildung 12: Use-Case Webpräsenz



4.2. NICHT-FUNKTIONALE ANFORDERUNGEN

4.2.1. BENUTZBARKEIT

Typ: Usability

Das System „PRoNMSP“ wird intuitiv benutzbar sein. Die Tätigkeiten werden erheblich erleichtert. Nach einer Einschulung und Einarbeitungsphase von 3-6 Stunden wird jeder Benutzer die Software bedienen und benutzen können.

Abbildung 13: Anforderungen - Benutzbarkeit

4.2.2. ZUVERLÄSSIGKEIT UND STABILITÄT

Typ: Performance, Funktionalität

Die Komponente Stundenplan von „PRoNMSP“ dient ausschließlich der Stundenplanerstellung. Die Korrektheit der Daten ist von der Benutzereingabe abhängig. Die Software weist zuverlässig auf Kollisionen hinsichtlich der Stundenbelegung hin und lässt diese erst gar nicht zu. Die Komponente Vertretungsplan von „PRoNMSP“ wird zuverlässig die richtigen Vertretungen vorschlagen und anzeigen sowie die angegebenen Informationen zu Exkursionen korrekt verarbeiten und darstellen (falsche Benutzereingaben sind vorbehalten). Das Programm ist 24h in voller Funktionalität verfügbar. Die Komponente Homepage ist mit uneingeschränktem Internetzugang zu 97% verfügbar.

Abbildung 14: Anforderungen – Zuverlässigkeit und Stabilität

4.3. MENGENGERÜST

4.3.1. STAMMDATEN

Die Neue Mittelschule Pabneukirchen hat derzeit ca. 20 Lehrkräfte und nochmals 7 weitere Angestellte. Diese Zahl steigt jährlich an.

Jedes Jahr ist viel Personal auf Fortbildung oder Exkursion, dadurch entstehen im Laufe des Schuljahres eine große Anzahl an Vertretungsstunden sowie der damit verbundene Aufwand.



5. Projektplanung und -management

WORUM GEHT ES IN DIESEM KAPITEL?

5.1. PROJEKTABLAUF.....	22
5.1.1. <i>Vorgehensmodell SCRUM</i>	23
5.1.2. <i>Vorgehensweise</i>	24
5.1.2.1. Kommunikationsmodell.....	24
5.1.2.2. Abnahmeverfahren.....	25
5.1.3. <i>Zeitplanung</i>	25
5.1.3.1. Start- & Endzeitpunkt.....	25
5.1.3.2. Gantt-Zeitplan.....	25
5.1.3.3. Aufwandsschätzung.....	27
5.1.4. <i>Projektstrukturplan</i>	28
5.1.5. <i>Verantwortungsmatrix</i>	29
5.2. RISIKOMANAGEMENT.....	30
5.3. RESSOURCENPLANUNG.....	32
5.3.1. <i>Hardware</i>	32
5.3.2. <i>Software</i>	33
5.3.3. <i>Menschliche Ressourcen</i>	34
5.3.4. <i>Stundenstatistik</i>	34
5.3.5. <i>Budget</i>	34
5.4. PROJEKTKOSTEN.....	35
5.4.1. <i>Kosten der Arbeitsleistung</i>	35
5.4.2. <i>Lizenzkosten</i>	36
5.4.3. <i>Sonstige Kosten</i>	36
5.4.4. <i>Gesamtkosten</i>	37
5.5. LEISTUNGSUMFANG.....	38
5.6. LIEFERUMFANG.....	39

Die Planung ist der Schlüssel zu einem gelungenen Projekt. Sobald alle Ziele und Aufgabenstellungen sowie Projektstruktur definiert sind, beginnt die Planungsphase. Wie wickle ich das Projekt ab, wie gehe ich vor. All diese Schritte müssen geplant werden, bevor überhaupt an dem Projekt gearbeitet werden kann.

Sobald alle Ziele und Aufgabenstellungen definiert worden sind, beginnt die Aufteilung der einzelnen Bereiche. Hier ist es wichtig, diese Ziele in kleine Teilbereiche zu teilen, unter den Projektmitgliedern aufzuteilen und die Verantwortlichkeit zu definieren. Danach kann mit der Abfolge der Arbeitspakete begonnen werden und daraus ergibt sich der Zeitplan. Danach beginnt die Umsetzung des Projektes.

Es wird auch eine fiktive Kostenplanung angegeben, um zu veranschaulichen, welche Kosten tatsächlich angefallen wären. Diese Kosten sind nur Schätzwerte.



5.1. PROJEKTABLAUF

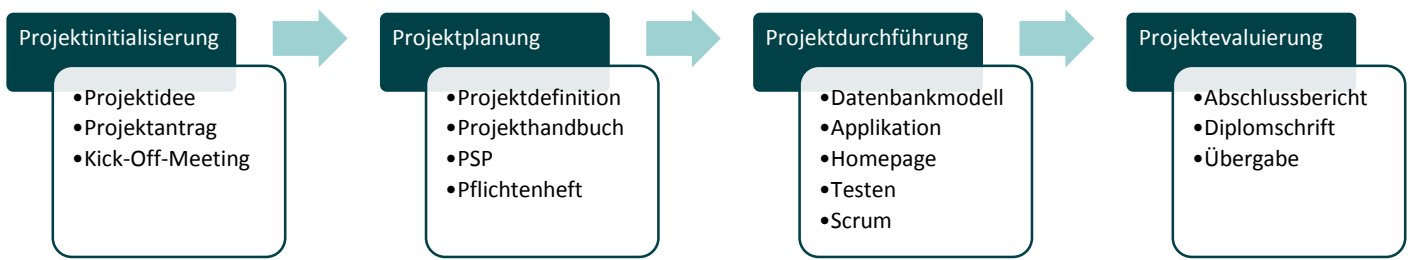


Abbildung 15: Projekttablauf

Die Projektinitialisierungsphase begann mit der Genehmigung des Diplomarbeitsantrags, nachdem dieser erstellt sowie eingereicht wurde. In dieser Phase ging es hauptsächlich darum, Dokumentenstandards zu definieren, sprich Vorlagen zu erstellen, Dokumentenablage einrichten sowie ein Logo entworfen. Das Projekt wurde offiziell mit dem Kick-Off-Meeting gestartet. Dieses wurde gemeinsam mit dem Auftraggeber abgehalten. Hier werden die einzelnen Anforderungen besprochen.

In der nachfolgenden Phase, der Projektplanung, ging es um die Definition und Verschriftlichung aller Anforderungen an das Projekt bzw. an die Applikation. Hier werden alle Grundsteine für die Projektdurchführung gelegt. Anhand dieser Dokumentationen orientiert man sich während der Durchführung, damit keine Unklarheiten im späteren Verlauf entstehen. Ein wichtiger Punkt in dieser Phase ist die Erstellung essenzieller Dokumente wie das Projekthandbuch, Pflichtenheft sowie der Projektstrukturplan (PSP).

Die Projektdurchführung beinhaltet alle Schritte, die für die erfolgreiche Erstellung der Applikation notwendig sind. Hier wurde mit der Modellierung der Datenbank begonnen. Diese bildet den Grundstein für die Anwendung. Dann wurde mit der Programmierung und Entwicklung der Software begonnen sowie die Homepage erstellt, designt und mit Daten befüllt. Während der Entwicklung wurden die einzelnen Komponenten stetig mithilfe von Firebug im Browser getestet.

Finale und somit letzte Phase bildet die Projektevaluierung. In diesem Abschnitt wurde das Projekt dem Auftraggeber Neue Mittelschule Pabneukirchen übergeben. Den Abschluss bildet die Präsentation der Diplomarbeit bei der schriftlichen Reife- und Diplomprüfung.



5.1.1. VORGEHENSMODELL SCRUM

Dieses Projekt wird nach dem Vorgehensmodell SCRUM⁴ abgewickelt. Dieses Modell kam im speziellen Bereich der Projektdurchführung im klassischen Projektablauf zum Einsatz (siehe Grafik *Abbildung 11*).

Scrum bildet ein Gesamtsystem aus verschiedensten Elementen wie Meetings, Rollen und Werten. Es wird ermöglicht, die einzelnen Aufgabenbereiche in kleine Stücke zu zerbrechen und diese Schritt für Schritt abzuarbeiten. So baut jeder Durchlauf auf den vorherigen Aufgaben auf. Durch die stetigen Meetings ist der Kunde immerzu miteinbezogen und bewertet die bisherigen Ergebnisse und kann sogleich Wünsche bzw. Änderungen bekannt geben. Scrum bildet somit einen Rahmen für effektive Zusammenarbeit im Team bei größeren bzw. komplexeren Projekten. Auch für kleinere Arbeiten bietet Scrum hilfreiche Elemente, man muss nicht den gesamten, komplexen Ablauf nutzen.

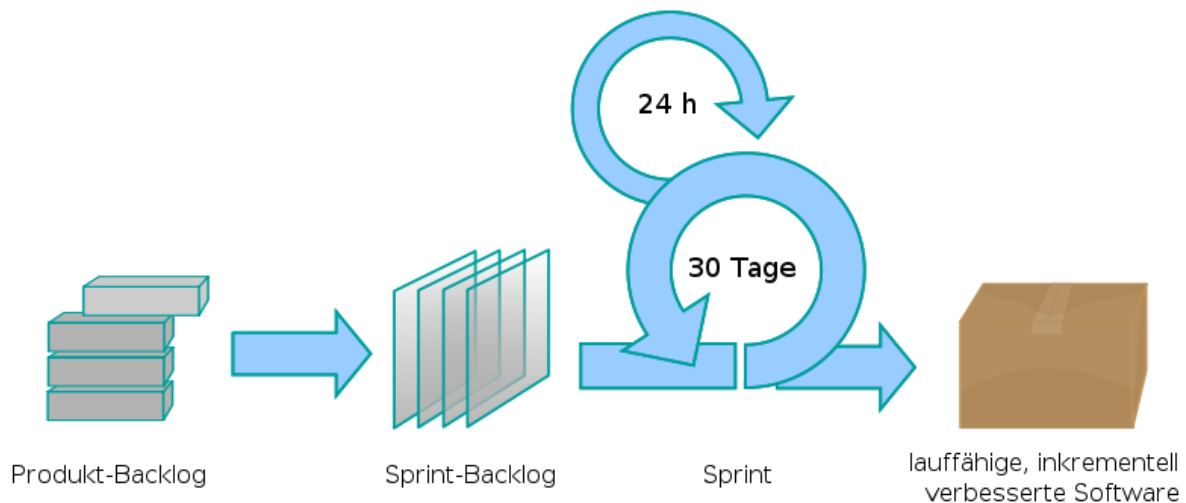


Abbildung 16: Prozessablauf SCRUM⁵

Produkt-Backlog

Hier werden alle Anforderungen vom Product Owner (Auftraggeber) festgehalten. Der Product Backlog bildet eine priorisierte und rein nutzerorientierte Liste mit Anforderungen, die bei der Entwicklung der Software berücksichtigt werden müssen.

Sprint-Backlog

Dieser beinhaltet jene Aufgaben, die in dem gerade stattfindenden Sprint zu erledigen sind.

Sprint

Der Sprint bildet den Arbeitszyklus von 30 Tagen ab. Nach dieser Anzahl an Arbeitstagen wird der Sprint beendet und das Ergebnis beim Meeting präsentiert.

⁴ <https://www.scrum.org/>

⁵ Bildquelle: <http://de.wikipedia.org/wiki/Scrum>



5.1.2. VORGEHENSWEISE

5.1.2.1. KOMMUNIKATIONSMODELL

Monatliche Meetings

Diese Meetings haben den Zweck, den aktuellen Status sowie die vorhandene Software zu besprechen. Diese wird getestet und dem Auftraggeber näher gebracht. Auch die nächsten Schritte werden besprochen und abgeklärt. Diese Form des Meetings wurde hauptsächlich mit dem Auftraggeber geführt und diente der monatlichen Absprache und Präsentation des derzeitigen Fortschritts.

Abbildung 17: Kommunikationsmodell – Monatliche Meetings

Vereinbarungen per E-Mail

Der Schriftverkehr herrscht vorwiegend zwischen den Diplomanden und dem Auftraggeber. Über dieses Kommunikationsmittel wurden die nächsten Schritte anvisiert sowie kurzfristig aufgetretene Fragen geklärt. Auch mit dem Betreuungslehrer wird dieser Schriftverkehr geführt. Dadurch werden Termine fixiert sowie aktuelle Dokumente übermittelt.

Abbildung 18: Kommunikationsmodell – Vereinbarung per E-Mail

Vereinbarungen per Telefon

Diese Telefonate dienen der Vereinbarung von Meetings, welche zwischen den Diplomanden sowie dem Auftraggeber stattfinden. Dieses Mittel diente rein der Kundgebung und Fixierung der nächsten Termine.

Abbildung 19: Kommunikationsmodell – Vereinbarung per Telefon



5.1.2.2. ABNAHMEVERFAHREN

Durch die agile Methode Scrum (Beschrieben im *Punkt 5.1.1*) entsteht nach den durchgeführten Sprints eine lauffähige Software. Dadurch kann der Auftraggeber diese stetig testen sowie Verbesserungen und Mängel aussprechen.

Das System „PRoNMSP“ mit den Komponenten Stundenplansoftware, Software für die Vertretungsplanung sowie Homepage, wird im Zuge eines Projektabnahmemeetings an die Neue Mittelschule Pabneukirchen übergeben. Eine Einschulung durch das Projektteam wird in diesem Rahmen durchgeführt. Auch die dazugehörigen Handbücher werden übergeben sowie durchgesprochen.

5.1.3. ZEITPLANUNG

5.1.3.1. START- & ENDZEITPUNKT

Das Projekt „Process-Redesign of NMS Pabneukirchen“ wird zu Beginn des Schuljahres Anfang September 2014/2015 gestartet.

Endzeitpunkt des Projektes ist der 13. Mai 2015. Die Übergabe des gesamten Systems sowie die Übergabe der wissenschaftlichen Arbeit erfolgt zu diesem Zeitpunkt.

5.1.3.2. GANTT-ZEITPLAN

Ein Gantt-Diagramm⁶ ist ein Instrument aus der Projektentwicklung. Dieses dient zur Planung von Zeitabläufen. Die zeitliche Abfolge von Arbeitsschritten wird auf einer Zeitachse in Form von Balken dargestellt. Weiters werden zu jedem Schritt der Start- sowie Endzeitpunkt definiert.

Auf der nächsten Seite wird das Diagramm des Projektes „PRoNMSP“ dargestellt, welches in der Projektplanungsphase erstellt worden ist.

⁶ <http://de.wikipedia.org/wiki/Gantt-Diagramm>

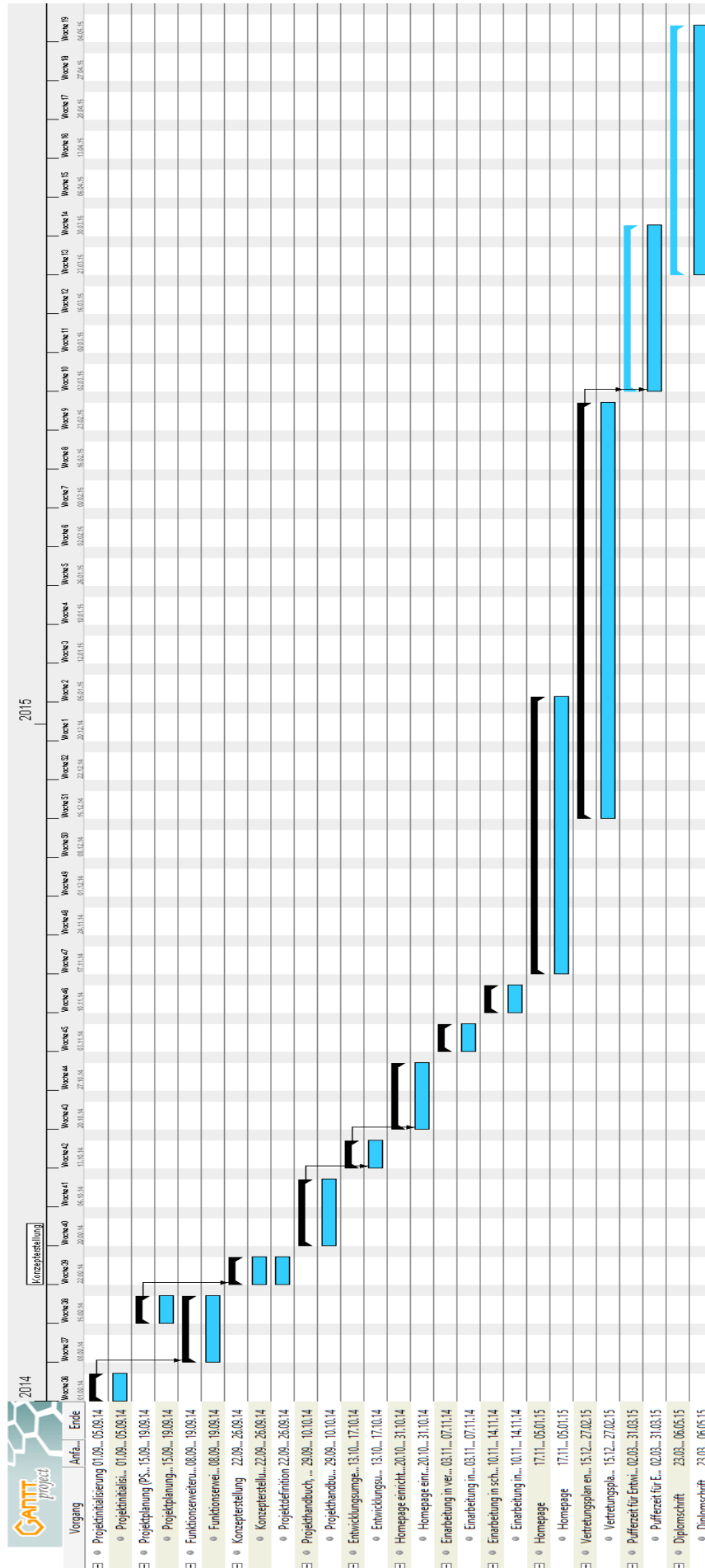


Abbildung 20: Gantt-Diagramm



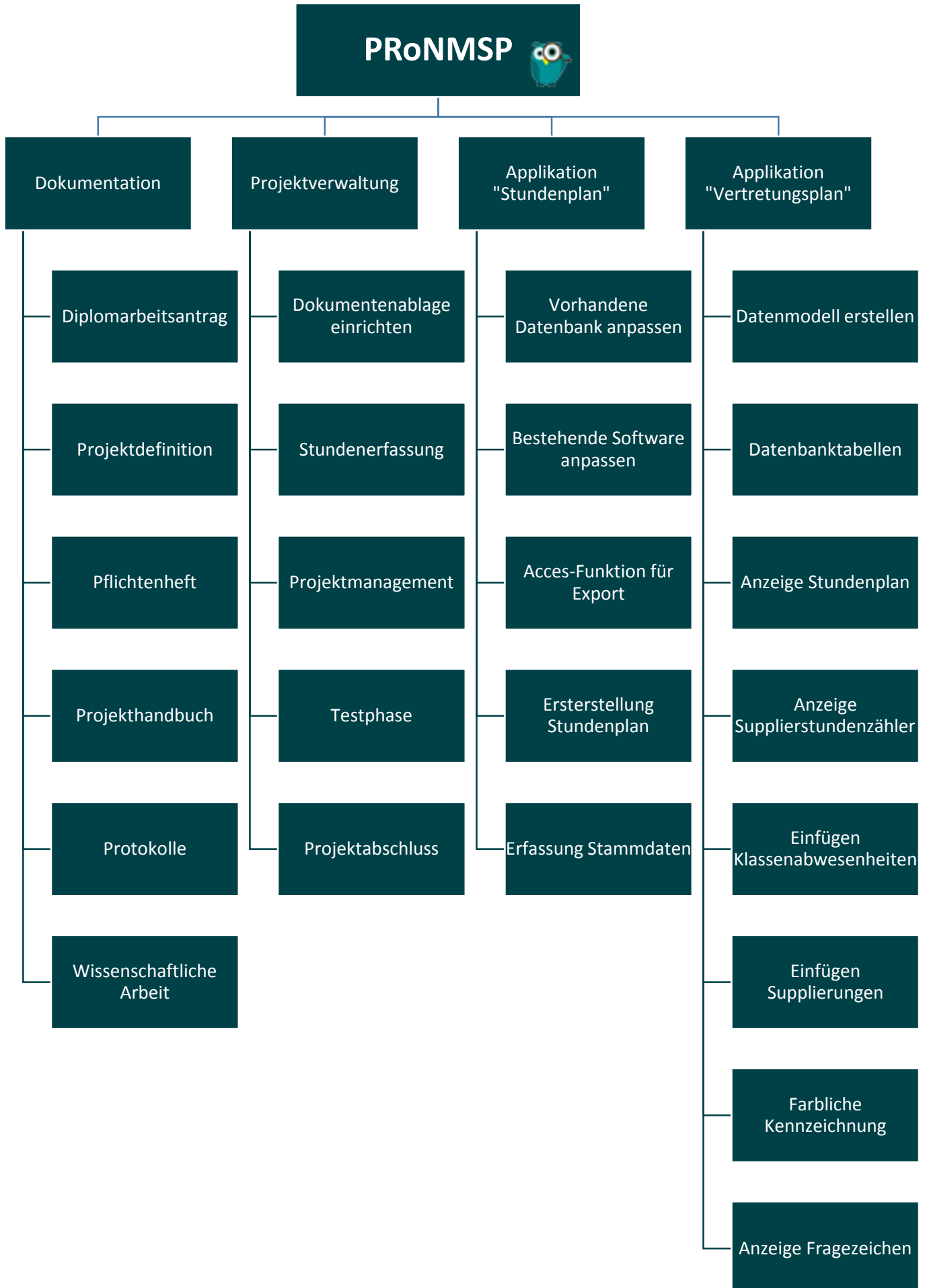
5.1.3.3. AUFWANDSSCHÄTZUNG

Name	Dauer (Stunden)
Projektinitialisierung (Namensfindung, Dokumentenvorlage, -standards, -ablage, Logo, etc.)	20
Projektplanung (PSP, Gantt-Plan, etc.)	10
Konzepterstellung	20
Projektdefinition	10
Projekthandbuch, Pflichtenheft	5
Entwicklungsumgebung einrichten	5
Homepage einrichten und konfigurieren	10
Einarbeitung in verwendete Technologien	10
Einarbeitung in Schulinterne Prozesse	15
Entwicklung:	
Software Vertretungsplan	90
Homepage	40
Funktionserweiterung Stundenplan	50
Modul- und Systemtests	10
Produktivtest mit Auftraggeber	10
Präsentationen und Wettbewerbe	10
Diplomarbeit	70
Summe	385

Tabelle 2: Aufwandsschätzung



5.1.4. PROJEKTSTRUKTURPLAN





5.1.5. VERANTWORTUNGSMATRIX

Durch die sogenannte IVM-Matrix soll die Verantwortlichkeit klar aufgeteilt werden. Diese soll verdeutlichen, welche Arbeitspakete unter welcher Verantwortung stehen. Die Beziehung zwischen Arbeitspaket und Projektbeteiligten soll dargestellt werden, jeder erhält somit eine bestimmte Rolle.

Diese wird durch Buchstaben gekennzeichnet. V bezeichnet den Verantwortlichen des Aufgabenbereichs. Mitarbeiter werden mit M bezeichnet und I steht für zu informierend. Wenn kein Buchstabe angeführt ist, betrifft denjenigen dieses Arbeitspaket nicht.

Arbeitspakete	Aufgabe	Beteiligte			
		Patricia Berger	Tanja Buchmair	DI Christian Reisinger	Auftraggeber
	Projektmanagement				
	Evaluierung der Geschäftsprozesse	V	M	I	I
	Kommunikation im Projekt Meetings	V	M	M	M
	Projektdokumente erstellen	M	V	I	I
	Bereitstellung von Daten	M	M	I	V
	Programmierung				
	Website	I	V	I	I
	Anpassung Stundenplansoftware	M	V	I	I
	Vertretungsplan-Applikation	V	M	I	I
	Datenbank	V	M	I	I
	Dokumentation	M	V	I	I

Tabelle 3: Verantwortungsmatrix



5.2. RISIKOMANAGEMENT

Mithilfe des Risikomanagement wird versucht, Risiken im Voraus zu identifizieren und zu analysieren. Daraus werden geeignete Maßnahmen definiert, damit im Laufe des Projekts, im Falle des Eintritts vorzeitig reagiert werden kann und die Folgen minimiert werden. Im Rahmen der Diplomarbeit wurden folgende Risiken identifiziert und ein Maßnahmenplan aufgestellt.

Nr.	Risiko	Eintrittswahrscheinlichkeit	Auswirkung
1	Unzureichende Erfahrung der Projektmitglieder	Mittel	Diverse Tasks nehmen mehr Zeit in Anspruch als geplant
2	Datenverlust	Niedrig	Alle verloren gegangenen Daten müssen neu erstellt werden
3	Komplexität unterschätzt	Hoch	Termine werden nicht eingehalten, Motivation sinkt
4	Schlechte Dokumentation	Mittel	Tätigkeiten können nicht nachvollzogen werden

Tabelle 4: Risikomanagement

UNZUREICHENDE ERFAHRUNG DER PROJEKTMITGLIEDER

Da das Projektteam Techniken verwendet, die vorher noch nicht bekannt waren, können schnell Probleme durch die Unerfahrenheit hervortreten. Dadurch ist die Lösungsphase von Problemen deutlich höher. Auch die Einarbeitungsphase benötigt mehr Zeit.

Gegenmaßnahme:

Um durch solche Zwischenfälle nicht aus dem Zeitplan zu geraten, sind sogenannte Pufferzeiten einzuberechnen. Vor allem bei den Einarbeitungsphasen, da hier der Aufwand höher sein wird als bei bereits bekannten Abläufen.



DATENVERLUST

Daten sind ein wichtiger Bestandteil der Projektarbeit. Durch diverse Dokumente sind alle Bestandteile sowie Anforderungen an die Applikation vermerkt. Wenn ein Datenverlust passiert, gehen alle wichtigen Dokumente der Diplomarbeit verloren, so auch die getätigten Arbeiten. Dies würde einen totalen Rückschlag bedeuten und die rechtzeitige Fertigstellung der Arbeit gefährden.

Gegenmaßnahme:

Um einen solchen Verlust zu vermeiden, wird eine Versionierungssoftware für die Dokumentenablage eingerichtet. Hier werden alle Dokumente online gespeichert und sind immer zugänglich. Zusätzlich werden die Dokumente auch am eigenen Rechner jedes Benutzers gespeichert. Somit sind die Daten mehrfach gesichert und ein Verlust kaum möglich.

KOMPLEXITÄT UNTERSCHÄTZT

Durch die Unerfahrenheit der Projektmitglieder im Bereich der Stundenplanerstellung sowie der Supplierstundenpflicht können schnell Probleme sowie Missverständnisse entstehen. Die Lösung dieser Hindernisse nimmt mehr Zeit in Anspruch als ursprünglich geplant.

Gegenmaßnahme:

Um solche aufwendigen Arbeiten zu vermeiden, ist eine Einschulung durch den Auftraggeber notwendig und eine anschließende Einarbeitungsphase, um das Besprochene auch zu verstehen. Durch das angeeignete Vorwissen können Unklarheiten beseitigt werden und eine zeitplanmäßige Arbeit erfolgen.

SCHLECHTE DOKUMENTATION

In der Projektplanungsphase entstehen viele wichtige Dokumente, die für die Entwicklung notwendig sind. Falls diese schlecht dokumentiert werden, können Fehler auftreten und bei der Entwicklung der Software zu Komplikationen führen. Auch vorher abgearbeitete Tätigkeiten können durch eine schlechte Dokumentenführung nicht mehr nachvollzogen werden.

Gegenmaßnahme:

In der Planungsphase ist es wichtig, ein Konfigurationsmanagement zu definieren. Dadurch werden Dokumentenstandards sowie die Dokumentenablage behandelt und erarbeitet. Eine gute Dokumentation kann nur durch eine gewissenhafte Schriftführung entstehen. Jedes anständige Dokument hilft der Projektarbeit und spart somit Zeit in der Nachvollziehung.



5.3. RESSOURCENPLANUNG

5.3.1. HARDWARE

Folgende Ressourcen sind bezüglich Hardware im Rahmen dieser Diplomarbeit notwendig:

ENTWICKLUNGSRECHNER

Die Entwicklung der Homepage sowie den programmtechnischen Teil erfolgt auf den vorhandenen Notebooks der Diplomanden. Somit sind keine weiteren Rechner notwendig.

DOKUMENTENABLAGE UND VERSIONISIERUNGS-SERVER

Um Dokumente und Softwarekomponenten abzulegen sowie sichern zu können, ist eine Software für die Dokumentenverwaltung notwendig. Hier bietet sich Dropbox an, welche kostenlos im Internet zur Verfügung steht. Sie aktualisiert und synchronisiert alle Dateien und Ordner automatisch, wenn Änderungen getätigt worden sind. Diese Aktion ist nur möglich, wenn ein Zugang zum Internet gegeben ist.

WEBSERVER

Durch die Abwälzung des Webdienstes auf ein drittes Unternehmen benötigen die Diplomanten zur Realisierung keinen eigenen Server. Somit besteht für den Auftraggeber kein Grund, einen eigenen Server anzuschaffen.



5.3.2. SOFTWARE

Für die Applikation notwendige Software wird im nachstehenden erläutert. Diese einzelnen Komponenten werden im *Kapitel 6* noch genauer beschrieben.

MICROSOFT WINDOWS 7 & 8⁷

Diese Diplomarbeit wurde unter der Verwendung des Betriebssystems von Microsoft Windows 7 sowie Microsoft Windows 8 realisiert.

MICROSOFT OFFICE 2013⁸

Mithilfe von Microsoft Word wurde die gesamte Projektdokumentation niedergeschrieben. Vor allem kamen besonders die Programme Outlook, Word, Excel und PowerPoint zum Einsatz.

JAVASCRIPT⁹

Für die Entwicklung der Software wurde die Sprache JavaScript verwendet.

PHP¹⁰

Ebenfalls wurde PHP für die Entwicklung der Software herangezogen.

WORDPRESS¹¹

Für die Erstellung der Homepage wurde das CMS¹² Wordpress verwendet, welches kostenlos im Internet zur Verfügung steht.

MYSQL¹³

Als Datenbank wird MySQL verwendet. Diese steht kostenlos im Internet zur Verfügung.

⁷ <http://windows.microsoft.com/de-at/windows-8/why-windows>

⁸ <http://products.office.com/de-AT/compare-microsoft-office-products>

⁹ <http://de.wikipedia.org/wiki/JavaScript>

¹⁰ <http://php.net/>

¹¹ <https://de.wordpress.org/>

¹² Content Management System

¹³ <https://www.mysql.de/>



5.3.3. MENSCHLICHE RESSOURCEN

Laut Richtlinien sind für eine Diplomarbeit an einer Höheren Technischen Lehranstalt mindestens 200 Arbeitsstunden pro Diplomand vorgeschrieben. Die daraus folgende Aufwandschätzung ist aus dem *Kapitel 5.1.3.3* zu entnehmen.

5.3.4. STUNDENSTATISTIK

Um die geleistete Arbeit genau darzulegen, wurde während der gesamten Durchführung des Projekts eine exakte Stundenaufzeichnung geführt. Diese dient der Gegenüberstellung der zuvor getätigten Aufwandschätzung.

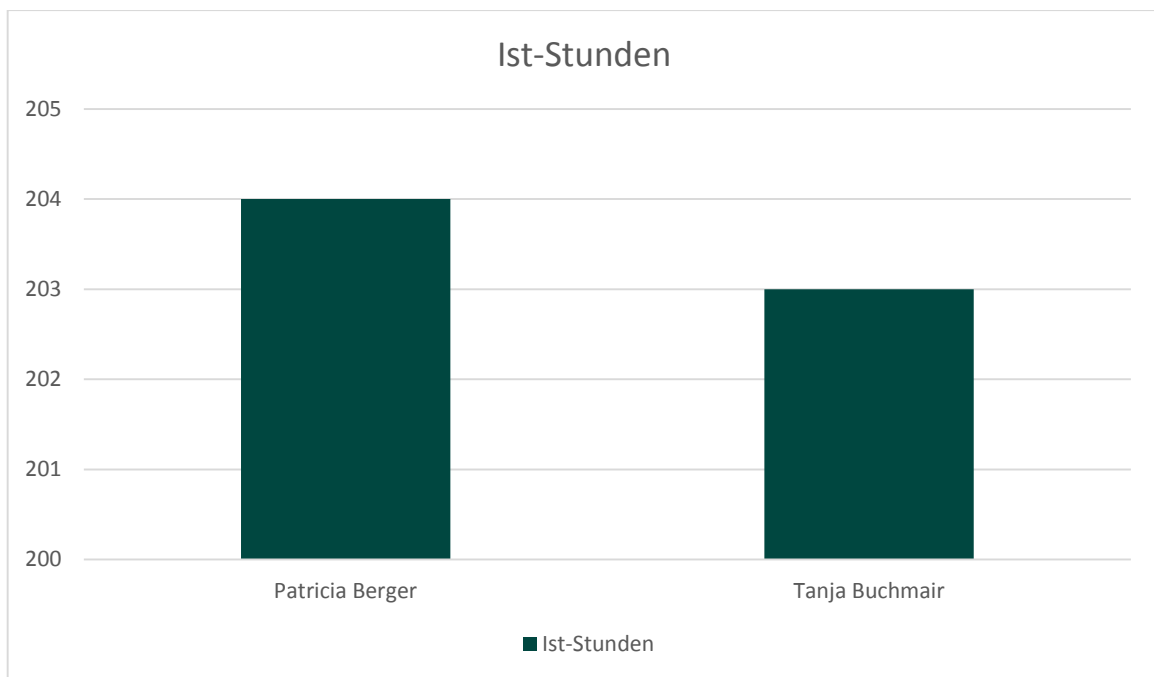


Abbildung 21: Ist-Stunden Statistik

Anhand dieser Grafik ist ersichtlich, dass insgesamt 407 Stunden in diese Diplomarbeit investiert wurde. Auch die geleistete Arbeit der Diplomanden ist sehr ausgeglichen.

5.3.5. BUDGET

Es wird kein Budget benötigt, da diese Arbeit im Rahmen der Matura an der HTL Perg durchgeführt wird, somit werden auch keine finanziellen Mittel anfallen.

Im nachfolgenden *Kapitel 5.4 Projektkosten* werden daher nur fiktive Kosten angenommen.



5.4. PROJEKTKOSTEN

Hier werden fiktive Kosten angenommen, die angefallen wären, wenn dieses Projekt in der Wirtschaft unter denselben Bedingungen durchgeführt werden würde. Es werden bei der Kostenaufstellung jene Kosten berücksichtigt, die für zwei Entwickler anfallen würden. Herangezogen werden die Kosten der Arbeitsleistung, Lizenzkosten sowie sonstige Kosten.

5.4.1. KOSTEN DER ARBEITSLEISTUNG¹⁴

Die nachfolgende Tabelle zeigt die Kosten der tatsächlich erbrachten Arbeitsleistung. Diese Kosten werden nicht geltend gemacht. Hier handelt es sich lediglich um Richtwerte, die Stundensätze wurden Branchenüblich gewählt. Die geleisteten Arbeitsstunden sowie deren Aufteilung sind aus dem *Kapitel 5.3* zu entnehmen.

Kostenträger	Stunden	Stundensatz	Summe
Patricia Berger	204	80€	€ 16.320,00
Tanja Buchmair	203	80€	€ 16.240,00
Gesamt	400		€ 32.560,00

Tabelle 5: Kosten der Arbeitsleistung

¹⁴ Der Stundensatz wurde aus der Informatikbranche übernommen



5.4.2. LIZENZKOSTEN¹⁵

Etliche Lizenzen waren von der Schule oder von uns bereits vorhanden, so entfallen diese Kosten. Sie sind nur als theoretische Zahlen anzusehen, da diese nicht geltend geworden sind. Um Kosten zu sparen, werden für die Entwicklung dieser Arbeit kostenlose Technologien verwendet, da ein großer Teil an Entwicklungssoftware frei zugänglich ist. Kosten könnten eventuell für diverse Betriebssysteme der Rechner anfallen.

Kostenträger	Stückzahl	Einzelpreis	Summe
Microsoft Windows 8.1	2	€ 279,00	€ 558,00
Microsoft Office 2013	2	€ 69,00/Jahr	€ 69,00
Gesamt			€ 627,00

Tabelle 6: Lizenzkosten

5.4.3. SONSTIGE KOSTEN

Kostenträger	Summe
Entwicklungsrechner	€ 1.600,00
Telefonie & Internet	€ 60,00
Fahrtkosten	€ 150,00
Büromaterial	€ 10,00
Webserver	€ 45,78
Sonstiges	€ 10
Gesamt	€ 1875,78

Tabelle 7: Sonstige Kosten

¹⁵ Diese Kosten wurden aus den jeweiligen Webshops der Hersteller entnommen (Stand 04/2015)



5.4.4. GESAMTKOSTEN

Folgende Gesamtkosten ergeben sich für die Diplomarbeit **PROCESSREDESIGN OF NMS PABNEUKIRCHEN** aus den im *Kapitel 5.4* angeführten Kosten:

Kostenträger	Summe
Arbeitsleistung	€ 32.560,00
Lizenzkosten	€ 627,00
Sonstige Kosten	€ 1875,78
Gesamt	€ 35.062,78

Tabelle 8: Gesamtkosten

Die Summe der fiktiv angenommenen Kosten beträgt **€ 35.062,78**. Der Großteil der Kosten fällt bei dem Kostenträger Arbeitsleistung an. Die Aufteilung lässt sich wie folgt veranschaulichen:

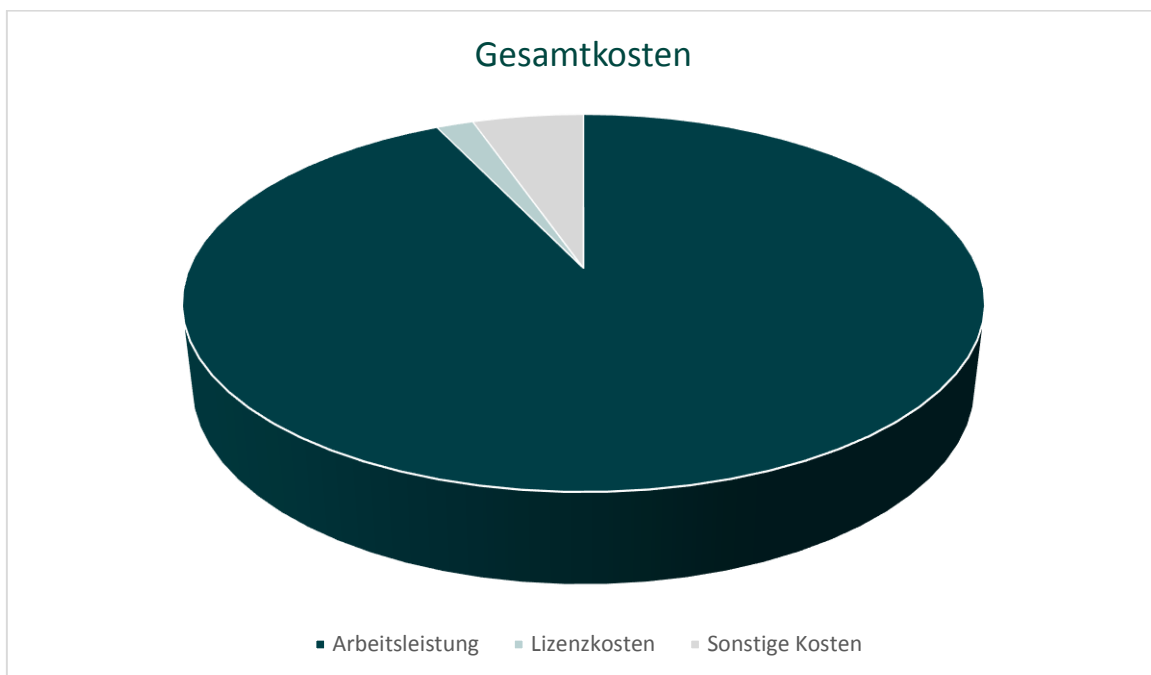


Abbildung 22: Fiktive Gesamtkosten



5.5. LEISTUNGSUMFANG

Der Leistungsumfang beschreibt die Menge aller Leistungen, die erbracht werden.

LEISTUNGSUMFANG HINSICHTLICH DER APPLIKATION „STUNDENPLAN“

Inscope	Out of scope
Verwaltung der Daten	Daten für neues Schuljahr 2015/16 einfügen
Funktionserweiterung (Speziell für Auftraggeber)	Wartung des Programmes

Tabelle 9: Leistungsumfang Applikation „Stundenplan“

LEISTUNGSUMFANG HINSICHTLICH DER APPLIKATION „VERTRETUNGSPLAN“

Inscope	Out of scope
Hinzufügen von Vertretung	Wartung nach einem Jahr
Bearbeiten von Vertretung	Erweiterung der Software
Liste der Supplierstundenanzahl	
Hinzufügen von Klassenabwesenheiten	
Löschen von Vertretungen	

Tabelle 10: Leistungsumfang Applikation „Vertretungsplan“

LEISTUNGSUMFANG HINSICHTLICH DER HOMEPAGE

Inscope	Out of scope
Startseite	Wartung der Homepage
Galerie	Befüllung mit neuen Daten
Wir über uns	
Leitung	
Veranstaltungskalender	
Speiseplan	
Archiv	
Übernahme vorhandener Daten	

Tabelle 11: Leistungsumfang Homepage



5.6. LIEFERUMFANG

Folgende Komponenten sind im Zuge der Diplomarbeit an den Auftraggeber zu liefern:

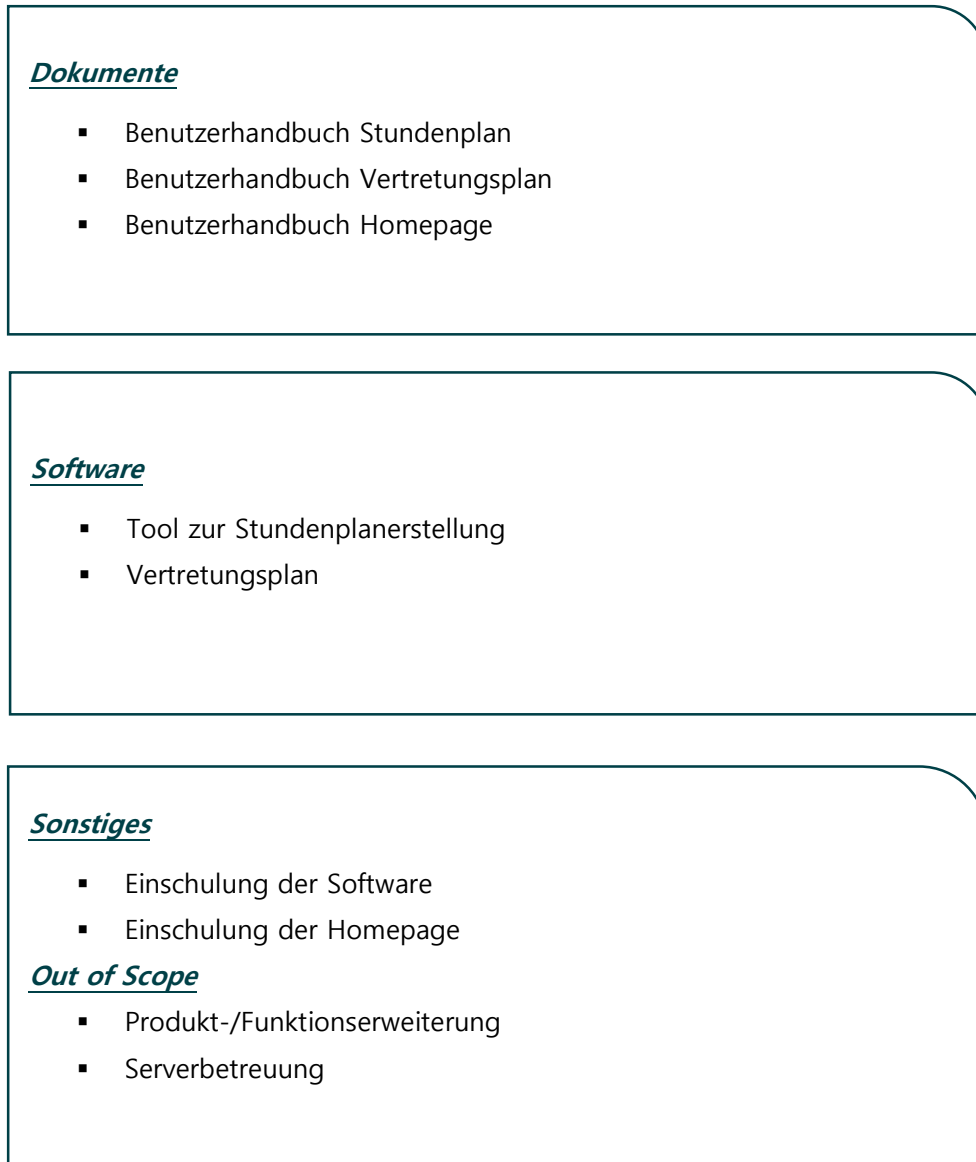


Abbildung 23: Lieferumfang





6. Grundlagen und Technologien

WORUM GEHT ES IN DIESEM KAPITEL?

6.1. HTML	42
6.2. PHP	42
6.3. MYSQL	42
6.4. DROPBOX	42
6.5. MICROSOFT OFFICE 2013	43
6.6. GANTT PROJECT	43
6.7. WORDPRESS	43
6.8. JQUERY	43
6.9. SCSS	43
6.10. JAVASCRIPT	44
6.11. MYSQL WORKBENCH	44
6.12. SOURCE TREE	44
6.13. FIREFOX FIREBUG	44
6.14. APACHE HTTP SERVER	45
6.15. BRACKETS	45
6.16. VMWARE WORKSTATION	45
6.17. UBUNTU	45
6.18. ECLIPSE LUNA PHP DEVELOPER	45

In diesem Kapitel werden alle verwendeten Technologien, Grundlagen und Software-Tools für die Entwicklung des Gesamtsystems gelistet. Alle, für die in der Diplomarbeit, verwendeten Komponenten werden genau beschrieben.



6.1. HTML

Die Hypertext Markup Language dient zur Übertragung von Texten und zusätzlichen Kommandos. Diese Sprache ist eine plattformunabhängige Dokumentbeschreibungssprache mit dem Haupteinsatzgebiet Web-Seiten. Die Formatierung der Texte wird nicht von HTML behandelt.



Abbildung 24: Logo HTML

6.2. PHP

Die Sprache Hypertext Preprocessor ist eine serverseitige Programmiersprache. Sie dient speziell der Webprogrammierung. Diese Sprache kann in eine HTML-Seite eingebettet werden und führt selbstständig die Befehle aus. Die Befehle werden direkt am Server beispielsweise in einen HTML-Code umgewandelt. Der Client erhält somit nur das Ergebnis, er sieht nicht, was der Entwickler im Hintergrund programmiert hat.



Abbildung 25: Logo PHP

6.3. MYSQL

MySQL ist ein kostenloses, einfaches, relationales Datenbankverwaltungssystem. Dieses hilft uns, große Datenmengen zu verwalten. Im Zuge der Diplomarbeit wurde die Version 5.5 verwendet.



Abbildung 26: Logo MySQL

6.4. DROPBOX

Dropbox ist ein kostenloser Cloud-Storage des Unternehmens Dropbox Inc. Das System dient der Online-Datenspeicherung sowie dem Austausch von Daten zwischen verschiedenen Personen, die Zugang auf den jeweiligen Speicher haben. Zugriff ist im Browser sowie durch Anwendungen möglich. Mit Apps ist der Zugang auch für mobile Geräte gegeben.



Abbildung 27: Logo Dropbox



6.5. MICROSOFT OFFICE 2013



Abbildung 28: Logo Microsoft Office

Microsoft Office 2013 ist ein kostenpflichtiges Produkt von Microsoft und ist ein Paket mit verschiedenen Komponenten. Die wichtigsten sind Word zur Textverarbeitung, Excel für die Tabellenkalkulation, Outlook für den E-Mail Verkehr sowie mit Access für die Datenverwaltung. Diese kamen im Rahmen der gesamten Diplomarbeit zum Einsatz.

6.6. GANTT PROJECT



Gantt Project ist ein kostenloses Tool zur Darstellung von Zeitplänen. Im Zuge der Diplomarbeit wurde der gesamte Ablauf mit diesem Tool realisiert. Dieser stellt den Ablauf der Arbeiten grafisch dar.

Abbildung 29: Logo Gantt Project

6.7. WORDPRESS



Wordpress ist ein kostenloses CMS¹⁶, mit welchem eine Homepage realisiert werden kann. Diese Software ist plattformunabhängig und sehr benutzerfreundlich. Aufgrund der Einfachheit wurde dieses System in der Diplomarbeit verwendet.

Abbildung 30: Logo Wordpress

6.8. JQUERY

JQuery stellt DOM¹⁷-Funktionen zur Verfügung und ist eine freie Java-Bibliothek. Diese bietet eine Schnittstelle für den Zugriff auf Datenbankobjekte. JQuery kommt in der Webprogrammierung zum Einsatz und glänzt durch die Browserkompatibilität. Durch die einfache und mächtige Syntax sind viele Elemente schnell realisierbar.



Abbildung 31: Logo jQuery

6.9. SCSS

SCSS ist eine neuere Version der Stylesheet-Sprache Sass¹⁸. Diese erleichtert die Erzeugung von Cascading Stylesheets. So gesehen ist SCSS eine neue Version von SASS, welche eine einfachere Syntax bietet sowie mehr Funktionalität.



Abbildung 32: Logo Sass

¹⁶ Content Management System

¹⁷ Document Object Model

¹⁸ Syntactically Awesome Stylesheets



6.10. JAVASCRIPT



JavaScript ist eine Programmiersprache und wird gemeinsam mit HTML verwendet, da diese eine Ergänzung zu dieser Sprache bildet. Sie wird im Zuge der Webprogrammierung eingesetzt, vor allem für Benutzerinteraktionen. Auch Elemente können somit täglich aktualisiert werden, da das Script beliebig oft aufgerufen werden kann.

Abbildung 33: Logo
JavaScript

6.11. MYSQL WORKBENCH



MySQL Workbench ist ein kostenloses Modellierungswerkzeug für Datenbanken und unterstützt bei der Planung sowie Realisierung dieser. Mithilfe dieses Tools kann die Datenbank auch gleich exportiert und verwendet werden. Es erleichtert die Erstellung erheblich, da das Modell grafisch aufbereitet wird.

Abbildung 34: Logo
MySQL Workbench

6.12. SOURCE TREE



Source Tree ist eine kostenlose Software und dient der Versionierung. Mithilfe dieses Tools kann gleichzeitig an einem Quellcode gearbeitet werden. Jeder Benutzer hat seinen eigenen Zweig und kann hier entwickeln, welche unabhängig voneinander sind. Durch Zusammenführen beider Pfade verschmelzen beide Arbeiten ineinander und der aktuelle Stand beider Entwickler ist hergestellt.

Abbildung 35: Logo
Source Tree

6.13. FIREFOX FIREBUG



Firefox Firebug ist eine kostenlose Erweiterung des Webbrowser Mozilla Firefox. Dieses Tool kam im Zuge der Diplomarbeit für die Fehlersuche sowie das Monitoring¹⁹ zum Einsatz.

Abbildung 36: Logo Firebug

¹⁹ Überwachung oder Beobachtung eines Prozesses



6.14. APACHE HTTP SERVER

Der Apache HTTP Server ist ein freies Produkt der Apache Software Foundation. Dieser Webserver wird meist in Verbindung mit der Programmiersprache PHP und Datenbanken wie MySQL verwendet. Hauptaufgabe ist die Auslieferung von Dateien an den Client wie zum Beispiel HTML-Seiten einer Webseite.



Abbildung 37: Apache Software Foundation

6.15. BRACKETS

Brackets ist freie Software welche unter der MIT-Lizenz²⁰ lizenziert ist. Dies ist ein Editor für die Entwicklung von Webanwendungen. Der Vorteil gegenüber anderen, kostenlosen Editors ist die Darstellung. Die unterschiedlichen Elemente werden farblich hervorgehoben und entsprechend gegliedert.



Abbildung 38: Logo Brackets

6.16. VMWARE WORKSTATION

VMware Workstation ist ein Emulator, der im Internet zum Download bereit steht. Mithilfe dieser Software ist es möglich, mehrere Betriebssysteme gleichzeitig zum Betreiben. Im Zuge dieser Diplomarbeit simulierte dieses Werkzeug das Betriebssystem UBUNTU mit einem dazugehörigen Server für Test- und Entwicklungszwecke.



Abbildung 39: Logo vmWare Workstation

6.17. UBUNTU

Ubuntu ist ein Betriebssystem auf Basis von Debian. Dieses System diente im Zuge der Diplomarbeit rein zu Testzwecken, da hier Testweise der Webserver realisiert worden ist.



Abbildung 40: Logo Ubuntu

6.18. ECLIPSE LUNA PHP DEVELOPER



Abbildung 41: Logo Eclipse Luna PHP

Eclipse Luna PHP Developer ist eine Entwicklungsumgebung von Eclipse Foundation, mit der aktuellen Version Luna²¹. Dieses System ist speziell auf die Programmiersprache PHP abgestimmt und bietet extra Support. Durch die Hilfestellungen wird die Entwicklung auf diesem abgestimmten System vereinfacht.

²⁰ <http://de.wikipedia.org/wiki/MIT-Lizenz>

²¹ Stand 04/2015





7. Umsetzung

WORUM GEHT ES IN DIESEM KAPITEL?

7.1. SYSTEMARCHITEKTUR.....	48
7.1.1. <i>TimeTable</i>	48
7.1.2. <i>TimeOwl (Vertretungsplan)</i>	48
7.1.3. <i>Webseite</i>	48
7.2. PROGRAMMARCHITEKTUR.....	49
7.2.1. <i>Index.php</i>	50
7.2.2. <i>Controller Factory</i>	51
7.2.3. <i>Controller</i>	52
7.2.4. <i>Repository Factory</i>	53
7.2.5. <i>Repository</i>	54
7.2.6. <i>Datenbank</i>	55
7.2.7. <i>View</i>	55
7.2.8. <i>Template</i>	55
7.2.9. <i>Model</i>	56
7.3. DATENMODELL	57
7.3.1. <i>Beschreibung der Datenbanktabellen</i>	58
7.4. DATENEXPORT (TIMETABLE) UND DATENIMPORT (TIMEOWL)	61
7.4.1. <i>Export</i>	61
7.4.2. <i>Import</i>	62
7.4.3. <i>Stundenplananzeige laden</i>	65

Dieses Kapitel beschäftigt sich mit der Entwicklung bzw. Umsetzung des zuvor erstellten Soll-Konzepts. Hier werden die einzelnen Punkte genau beschrieben, wie die Systeme entwickelt worden sind und wie die einzelnen Komponenten speziell an die Neue Mittelschule Pabneukirchen angepasst wurden.

Es wird ein Überblick über das Gesamtsystem gegeben. Genau eingegangen wird auf die eigens entwickelte Software „*TimeOwl*“, den Vertretungsplan. Die anderen Komponenten, die bestehende Software „*TimeTable*“ sowie das CMS für die Homepage, werden im Grundgerüst genannt und kurz angeschnitten, die einzelnen Softwarekomponenten werden nicht gesondert erklärt.



7.1. SYSTEMARCHITEKTUR

In unserem Fall beschäftigen wir uns mit drei verschiedenen Systemen, welche über unterschiedliche Zugriffsrechte und Arten verfügen.

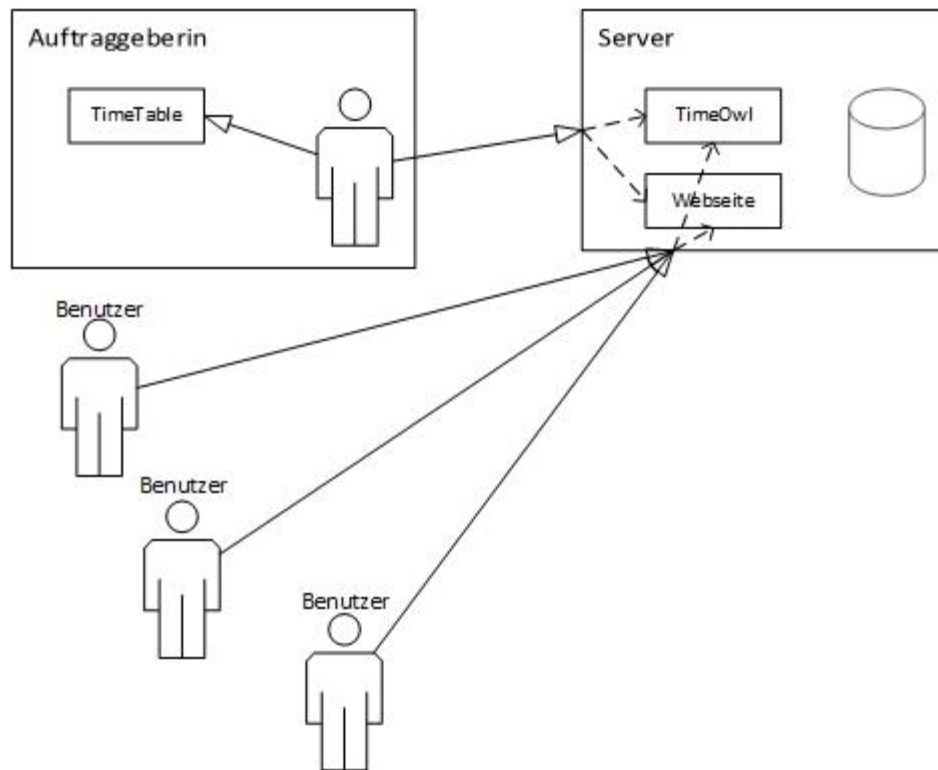


Abbildung 42: Systemarchitektur

7.1.1. TIMETABLE

Die Stundenplan Software wird lokal auf dem Rechner der Auftraggeberin installiert und ist auch nur über diesen benutzbar. *TimeTable* nutzt zur Datenspeicherung eine Access-Datenbank welche mit dem Produkt mitgeliefert wurde. Diese Datenbank liefert eine Export Funktion womit Daten für den Vertretungsplan exportiert werden können.

7.1.2. TIMEOWL (VERTRETUNGSPLAN)

Der Vertretungsplan und die zugehörige Datenbank liegen hingegen auf einem eigenen Server und somit ist der Vertretungsplan auch über jeden Rechner erreichbar – solange man über den benötigten Login verfügt. Hier ist nun auch (für Administratoren) eine Import Funktion enthalten, um die Daten vom zuvor erstellten Stundenplan ins System zu bringen.

7.1.3. WEBSEITE

Die Webseite liegt, ebenso wie der Vertretungsplan, auf dem Server und ist ebenfalls über jeden Rechner erreichbar – kann aber auch ohne Login angesehen werden.



7.2. PROGRAMMARCHITEKTUR

Die Abbildung stellt die eingesetzte Systemarchitektur des Vertretungsplans dar, welche aus mehreren Teilbereichen besteht. Da *TimeTable* bereits eine bestehende Software ist und die *Webseite* ein bereitgestelltes System, werden diese im Folgenden nicht genauer betrachtet.

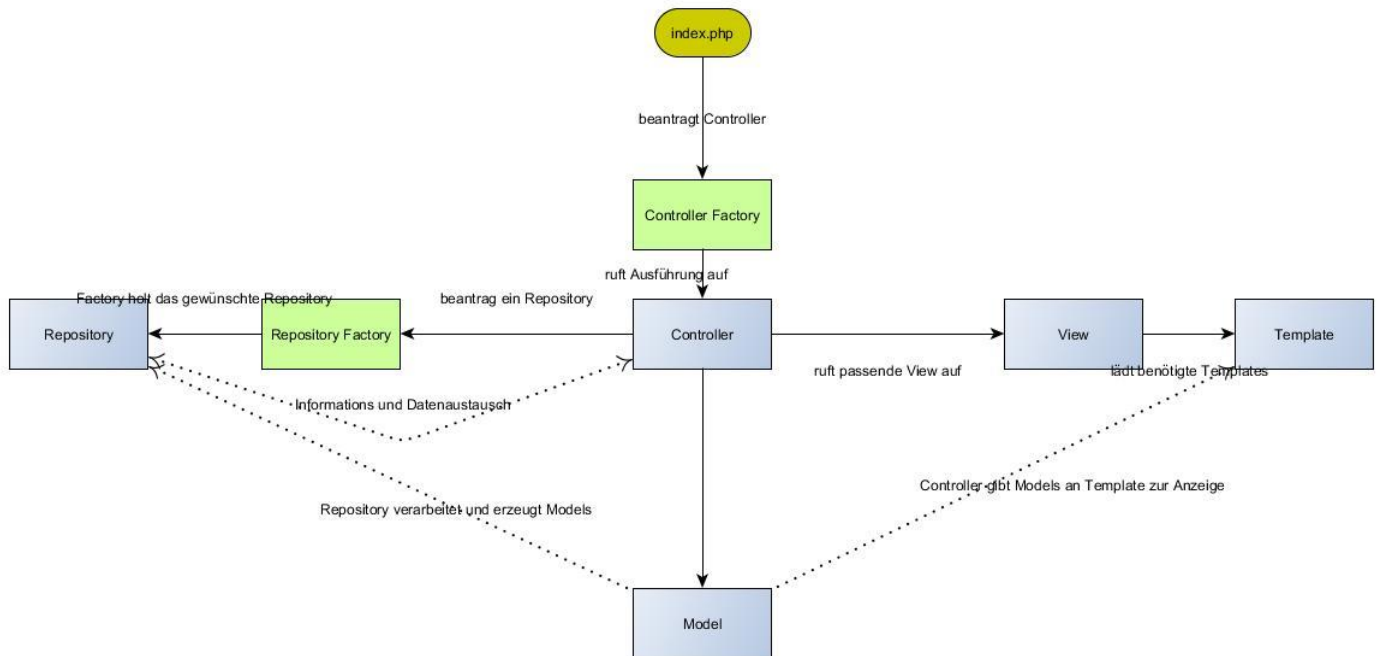


Abbildung 43: Programmarchitektur TimeOwl

Das Dateisystem der Anwendung ist passend zur Abbildung, da Controller + Factory, Repository + Factory, Model und View in eigens dafür vorgesehene Unterordner abgelegt werden und somit auch den passenden Namespace besitzen.



7.2.1. INDEX.PHP

Die „Einstiegsstelle“ in das System. Die Aufgabe der index.php ist es, zu erkennen ob der User eingeloggt ist oder nicht und auf den gewünschten Controller zuzugreifen und diesen zu starten.

```
if(isset($_GET['site'])){
    $site = ucfirst($_GET['site']);
    if(method_exists('\Controller\Factory', 'getCtrl'.$site)){
        $ctrl = \Controller\Factory::{'getCtrl'.$site}();
        if($ctrl instanceof Controller\LoggedInController){
            if($ctrl->isSucceed()){
                $ctrl->show();
            } else {
                $ctrl->showError();
            }
        } else {
            $ctrl->show();
        }
    } else{
        \Controller\Factory::getCtrlHome()->show();
    }
} else {
    if(\Controller\Factory::getCtrlIntern()->isSucceed()){
        \Controller\Factory::getCtrlIntern()->show();
    } else {
        \Controller\Factory::getCtrlHome()->show();
    }
}
```

Code Snippet 1: index.php



7.2.2. CONTROLLER FACTORY

Die Aufgabe der Controller Factory ist es, der *index.php* den gewünschten Controller zu überreichen. Eine Factory ist notwendig um eine abhängigkeitslose Schnittstelle zu erzeugen. Heißt also: egal woher, egal ob von unserem System oder einer anderen Schnittstelle, die *index.php* bekommt einen Controller ohne diesen erst suchen zu müssen.

```
namespace Controller;

final class Factory{
    protected static $ctrlHome = null;
    protected static $ctrlIntern = null;
    protected static $ctrlSaveLieferungen = null;
    protected static $ctrlGetFaecher = null;
    protected function __construct(){
        // Thou shalt not construct that which is unconstructable!
    }

    protected function __clone(){
        // Me not like clones! Me smash clones!
    }

    public static function getCtrlHome(){
        if(!isset(self::$ctrlHome)){
            self::$ctrlHome = new HomeController();
        }
        return self::$ctrlHome;
    }
    public static function getCtrlIntern(){
        if(!isset(self::$ctrlIntern)){
            self::$ctrlIntern = new InternController();
        }
        return self::$ctrlIntern;
    }
    public static function getCtrlSaveLieferungen(){
        if(!isset(self::$ctrlSaveLieferungen)){
            self::$ctrlSaveLieferungen = new SaveLieferungenController();
        }
        return self::$ctrlSaveLieferungen;
    }
    public static function getCtrlGetFaecher(){
        if(!isset(self::$ctrlGetFaecher)){
            self::$ctrlGetFaecher = new GetFaecherController();
        }
        return self::$ctrlGetFaecher;
    }
}
```

Code Snippet 2: Controller Factory



7.2.3. CONTROLLER

Die Controller stellen die Prozesse im System dar, jeder hat eigene Aufgaben und reagiert individuell auf verschiedenste Parameter. Sie sind auch das „Herzstück“ des Systems – denn ohne Controller läuft nichts. Ein Controller hat also die Aufgabe eingegebene Daten zu verarbeiten, Modelle aus der Datenbank zu laden oder zu speichern, Templates und Views zu wählen und Daten weiterzuleiten.

```
namespace Controller;
class InternController extends LoggedInController{
    public function __construct(){
        parent::__construct();
        $this->data = new \stdClass();
    }
    public function show(){
        $this->event = "event_timetable";
        $this->data->kw = isset($_GET['kw'])? $_GET['kw'] : date("W");
        $this->data->klassen = \Repository\Factory::getRepKlasse()->getAll();

        if(isset($_GET['supply'])){
            $stunden = \Repository\Factory::getRepStundenPlan()->getByKlasseAndKW($this->data->klasse, $this->data->kw);
            $stundenJSON = array();
            foreach($stunden as $stunde){
                $stundenJSON[] = $stunde->toJSON();
            }
            (new \View\JSONView()->output($stundenJSON);

        } else {
            $this->data->currKw = date("W");
            $this->data->stunden = \Repository\Factory::getRepStundenPlan()->getByKlasseOrderByDayAndHour($this->data->klasse);
            $this->data->lehrer = \Repository\Factory::getRepLehrer()->getAll();

            //Call Daddy
            (new \View\UserView()->output("intern", "Supplierplan", $this->data, $this->event);
        }
    }
}
```

Code Snippet 3: Controller

InternController.php

Im hier angeführten Code-Schnipsel, kann man zuerst erkennen wie Daten in einem Controller hinterlegt werden, anschließend auf übergebene Parameter reagiert und auf verschiedene Views verwiesen wird.



7.2.4. REPOSITORY FACTORY

Der Sinn der Repository Factory ist der selbige wie bei der Controller Factory. Jedoch ist es hier noch einmal wichtiger diese Schnittstelle zu gewährleisten da gegebenenfalls nicht nur verschiedene Repositories von verschiedenen Orten zum Einsatz kommen sondern auch verschiedene Datenbanken.

```
namespace Repository;

final class Factory{
    protected static $user = null;
    protected static $instance = null;
    protected static $repUser = null;
    protected static $repLehrer = null;

    public static function getInstance(){
        if(!isset(self::$instance)){
            $server = "localhost";
            $user = "root";
            $password = "*****";
            $database = "supplierplan";
            self::$instance = new \PDO('mysql:host=' . $server . ';dbname=' . $database .
';charset=utf8;', $user, $password);
            self::$instance->setAttribute(\PDO::ATTR_ERRMODE, \PDO::ERRMODE_EXCEPTION);

        }
        return self::$instance;
    }
    public static function getUser(){
        return self::$user;
    }
    public static function setUser($user){
        self::$user = $user;
    }
    public static function clearDatabase($newYear = false){
        self::getRepStundenplan()->deleteAll();
        self::getRepKlasse()->deleteAll();
        self::getRepFach()->deleteAll();
    }
    /**
     *
     * @return \Repository\RepositoryUser
     */
    public static function getRepUser(){
        if(!isset(self::$repUser)){
            self::$repUser = new RepositoryUser(self::getInstance());
        }
        return self::$repUser;
    }

    public static function getRepLehrer(){
        if(!isset(self::$repLehrer)){
            self::$repLehrer = new RepositoryLehrer(self::getInstance());
        }
        return self::$repLehrer;
    }
}
```

Code Snippet 4:Repository Factory



7.2.5. REPOSITORY

Die Aufgabe der Repositories ist es die Datenbank in einen konsistenten Zustand zu behalten und Datenbankoperationen zu erlauben. Um dies zu gewährleisten dürfen nur vorhandene Funktion für Datenbankinteraktionen genutzt werden. Die Repositories bieten Create, Update und Delete Funktionen, teils auch für mehrere Kriterien, und etwaige Arten zur Selektion.

```
namespace Repository;
class RepositoryFach{
    /**
     * @var \PDO
     */
    private $cube;

    /**
     * @var \Core\Lehrer[]
     */
    private $fachList = array();
    /* Dependency Injection */
    public function __construct($cube){
        $this->cube = $cube;
    }
    public function importFromCSV($file){
        $stmt = $this->cube->prepare("INSERT INTO Faecher (Fach_ID, numID, Langname)
VALUES(:id, :short, :long)");
        if (($handle = fopen($file, "r")) !== FALSE) {
            while (($data = fgetcsv($handle, 1000, ";")) !== FALSE) {
                $stmt->bindValue(':id', utf8_encode($data[1]));
                $stmt->bindValue(':short', $data[0]);
                $stmt->bindValue(':long', utf8_encode($data[2]));
                $stmt->execute();
            }
            fclose($handle);
        }
    }
    public function deleteAll(){
        $this->cube->query("DELETE FROM Faecher");
    }

    public function getIDByNumID($id){
        $stmt = $this->cube->prepare("SELECT Fach_ID FROM Faecher WHERE numID = :id");
        $stmt->bindValue(':id', $id);
        $stmt->execute();
        return $stmt->fetchColumn();
    }

    public function getAll(){
        $result = $this->cube->query("SELECT Fach_ID, Langname FROM Faecher WHERE Fach_ID !=
'EX'");
        while($data = $result->fetchObject()){
            if(!isset($this->fachList[$data->Fach_ID])){
                $this->fachList[$data->Fach_ID] = new \Core\Fach($data);
            }
            yield $this->fachList[$data->Fach_ID];
        }
    }
}
```

Code Snippet 5: Repository



7.2.6. DATENBANK

In unserem Fall handelt es sich hierbei um eine MySQL Datenbank, welche ausschließlich die Daten von *TimeOwl* enthält. Sie liegt auf dem gleichen System wie die Software und beinhaltet alle Daten die die Repositories erfolgreich senden.

7.2.7. VIEW

Die Aufgabe der View ist es die korrekte Anzeigeform zu präsentieren (mit oder ohne Design, das Design des Gasts oder eingeloggten User, usw.) und die zugehörigen Templates zu wählen. Die View ist stark vom Controller abhängig, da sie erstens von Controller gewählt wird und zweitens vom Controller gesagt bekommt welches Template nun benötigt wird.

Zusätzlich zu den Templates lädt die View noch Sprachdateien und zu den Template gehörende JavaScript-Skripte.

```
namespace View;
class UserView{
    public function output($templateName, $title, $data = null, $eventBindings = null){
        $user = \Repository\Factory::getUser ();
        include \Core\Config::getAppDir () . "/Template/User/_surrounding_user_start.php";
        include \Core\Config::getAppDir () . "/Template/User/" . $templateName . ".php";
        include \Core\Config::getAppDir () . "/Template/User/_surrounding_user_end.php";
    }
}
```

Code Snippet 6: View

UserView.php

Hier wird das Design in zwei Teilen eingefügt und dazwischen das vom Controller gewählte Template.

7.2.8. TEMPLATE

Templates sind aus Benutzersicht, in Verbindung mit der View, „die aktuelle Seite“. Sie kümmern sich also darum das die benötigten Daten (Model) anschaulich präsentiert werden und durch zusätzliche implementierbare JavaScript-Skripte durchaus auch veränderbar sind.

```
<div id="box">
    <form method="post" action="?site=intern" id="login">
        
        <span>Username</span>
        <input type="text" name="username" placeholder="username">
        <span>Passwort</span>
        <input type="password" name="password" placeholder="password">
        <input type="submit" value="Einloggen" id="submit">
    </form>
</div>
```

Code Snippet 7: Template aktuelle Seite



7.2.9. MODEL

Die Models sind die einzelnen Objekte an sich. In unserem Fall, also Lehrer, Klassen, Unterrichtseinheiten und dergleichen. Ihr Vorteil ist es, das jeder Objekttyp seine eigenen Funktionen besitzt und somit alles überschaubar bleibt. Weiters bringen Getter und Setter uns hier die Möglichkeit, Daten aus der Datenbank nachträglich zu laden bzw. in abgewandelter Form zu präsentieren aber auch vor Zugriffen zu schützen.

```
namespace Core;
class Lehrer{
    private $id;
    private $admin;
    private $name;
    private $password;
    private $stunden;
    private $currStunden;
    private $systempass;
    private $hash;

    public function __construct($dbo = null){
        if($dbo){
            $this->setAll($dbo);
        }
    }
    public function setAll($dbo){
        $this->id = $dbo->Lehrer_ID;
        $this->admin = (bool) $dbo->Administrator;
        $this->name = $dbo->Langname;
        $this->password = $dbo->Passwort;
        $this->stunden = $dbo->Stunden;
        $this->currStunden = $dbo->currStunden;
        $this->systempass = $dbo->SystemPass;
        if(!isset($dbo->hash)){
            $this->hash = md5($this->id);
        } else {
            $this->hash = $dbo->hash;
        }
    }
    public function getID(){
        return $this->id;
    }
    public function getAdmin(){
        return $this->admin;
    }
    public function getName(){
        return $this->name;
    }
    public function getPassword(){
        return $this->password;
    }
    public function getStunden(){
        return $this->stunden;
    }
    public function getSystempass(){
        return $this->systempass;
    }
    public function getCurrStunden(){
        return $this->currStunden;
    }
    public function toJSONArr(){
        return array(
            'id' => $this->id,
            'name' => $this->name
        );
    }
    public function toJSONArrWithStunden(){
        return array(
            'id' => $this->id,
            'name' => $this->name,
            'currStunden' => $this->currStunden,
            'stunden' => $this->stunden
        );
    }
}
```

Code Snippet 8: Model – Beispielobjekt Lehrer



7.3.1. BESCHREIBUNG DER DATENBANKTABELLEN

Die Lehrer Tabelle beinhaltet alle Lehrer der NMS Pabneukirchen und beinhaltet auch Login Daten für diese. Somit ist sie vergleichbar mit einer User Tabelle.

Tabelle Lehrer	
Feld	Erklärung
Lehrer_ID	Der Identifikationscode eines Lehrers
numID	Eine eindeutige Identifikationszahl welche nur für den Import benötigt wird um die Schlüssel korrekt miteinander zu verbinden.
Langname	Der vollständige Name eines Lehrers
Stunden	Jährliche Supplierreserve eines Lehrers
Passwort	Das vom Lehrer gewählte Passwort für den Login
Administrator	Flag um es sich um einen Administrator oder User handelt
SystemPass	Flag ob das Passwort noch vom System erstellt oder bereits vom User ist.
currStunden	Die bisher absolvierten Supplierstunden des aktuellen Jahrs
Hash	Ein Hash für den Login über Cookies

Tabelle Faecher	
Feld	Erklärung
Fach_ID	Der Identifikationscode eines Faches
numID	Eine eindeutige Identifikationszahl welche nur für den Import benötigt wird um die Schlüssel korrekt miteinander zu verbinden.
Langname	Der vollständige Name eines Faches

Tabelle Klassen	
Feld	Erklärung
Klasse_ID	Der Identifikationscode einer Klasse
numID	Eine eindeutige Identifikationszahl welche nur für den Import benötigt wird um die Schlüssel korrekt miteinander zu verbinden.



Die Stundenplan Tabelle beinhaltet sowohl reguläre Unterrichtseinheiten, wie auch Vertretungen und Exkursionen (aka. Klassenabwesenheiten)

Tabelle Stundenplan	
Feld	Erklärung
Stundenplan_ID	Der Identifikationscode eines Stundenplaneintrags
Tag	Tag der Woche von 1-5
Stunde	Unterrichtseinheit von 1-9
Lehrer	Fremdschlüssel der auf einen Lehrer verweist
Klasse	Fremdschlüssel der auf eine Klasse verweist
Fach	Fremdschlüssel der auf ein Fach verweist
Abwesend	Flag für spätere Implementierung einer Lehrer Abwesenheit
KW	Eintrag für die Kalenderwoche, NULL wenn es sich um eine reguläre Einheit handelt

Der LOG wird geführt um nach einer Stundenplanaktualisierung den Zähler trotzdem noch auf das ganze Jahr legen zu können. Sie ist somit ein sehr wichtiger Bestandteil des ganzen Systems.

Tabelle LOG_Supplierungen	
Feld	Erklärung
LOG_Supplierungen_ID	Der Identifikationscode des LOG Eintrags
Lehrer	Fremdschlüssel der auf den Lehrer verweist
Stundenplan_Stunde_ID	Fremdschlüssel der auf eine Unterrichtseinheit verweist, falls diese noch existiert.

Klassenabwesenheiten sind für uns auch als Exkursionen oder Ausflüge bekannt. Diese ermöglichen dem System jene Lehrer anzuzeigen, die nun durch diese Abwesenheit ausfallen und markiert diese mit einem ,???' in den jeweiligen Einheiten (siehe Abbildung so und so).

Tabelle Klassenabwesenheit	
Feld	Erklärung
Klassenabwesenheit_ID	Der Identifikationscode einer Klassenabwesenheit
Name	Name der Abwesenheit
Start	Startdatum
End	Enddatum



Da bei einer Klassenabwesenheit mehrere Klassen und Lehrer betroffen sein können, ist es hier von Nöten eine N:M Beziehung im Sinne einer assoziativen Tabelle zu realisieren.

Tabelle Klassenabwesenheit_has_Lehrer (assoziative Tabelle)	
Feld	Erklärung
Klassenabwesenheit_ID	Fremdschlüssel zur Klassenabwesenheit
Lehrer_ID	Fremdschlüssel zum Lehrer

Tabelle Klassenabwesenheit_has_Klassen (assoziative Tabelle)	
Feld	Erklärung
Klassenabwesenheit_ID	Fremdschlüssel zur Klassenabwesenheit
Klasse_ID	Fremdschlüssel zur Klasse

Zusätzlich zu diesen Tabellen gibt es noch drei temporäre Importtabellen, welche nur zum Import aus *TimeTable* benötigt werden und deswegen hier nicht extra aufgeführt werden.



7.4. DATENEXPORT (TIMETABLE) UND DATENIMPORT (TIMEOWL)

7.4.1. EXPORT

Um Daten in *TimeOwl* zu importieren müssen diese erst von *TimeTable* exportiert werden. Damit dies überhaupt möglich ist, musste für Access ein Export Modul entwickelt werden.

Zuerst wurden dafür passende Queries in Access erzeugt und diese mittels VisualBasic exportiert.

```
Sub exportQuery(str As String)
    DoCmd.TransferText acExportDelim, "Fach-Export", "Fach-Abfrage", str & "\fach.csv"
    DoCmd.TransferText acExportDelim, "Klasse-Export", "Klasse-Abfrage", str & "\klasse.csv"
    DoCmd.TransferText acExportDelim, "KlassenZ-Export", "KlassenZ-Abfrage", str &
"\klassenzuordnung.csv"
    DoCmd.TransferText acExportDelim, "Lehrer-Export", "Lehrer-Abfrage", str & "\lehrer.csv"
    DoCmd.TransferText acExportDelim, "Stundenplan-Export", "Stundenplan-Abfrage", str &
"\stundenplan.csv"
    DoCmd.TransferText acExportDelim, "UnterrichtZ-Export", "UnterrichtZ-Abfrage", str &
"\unterrichtszuordnung.csv"
End Sub
```

Da, das System dies in einen fixen Pfad speichert muss zuerst noch der Pfad ausgewählt werden. Da dies für den Auftraggeber möglichst einfach sein soll haben wir uns in diesem Fall für einen sogenannten *FileDialog* entschieden.

```
Public Function OrdnerEinstellen()
    Dim fd As FileDialog
    Set fd = Application.FileDialog(msoFileDialogFolderPicker)
    With fd
        If .Show = -1 Then
            exportQuery (fd.SelectedItems(1))
        Else
            End If
    End With
    Set fd = Nothing
End Function
```

Um das ganze nun aber noch einmal zu vereinfachen muss das Modul zum Start von Access aufgerufen werden. Dies wird bewerkstelligt in dem man das Makro „*AutoExec*“ nennt.

Der Nutzer erhält nun sechs Files welche in *TimeOwl* importiert werden müssen.



7.4.2. IMPORT

Um Dateien hochzuladen muss erst einmal der korrekte HTML-Part dafür vorhanden sein. Hier haben wir auch eine Prüfung eingebaut, welche die Dateinamen direkt auf Clientseite prüft um dem Nutzer zu helfen falls er eine Datei vergisst oder zu viel oder gar falsch auswählt.

```
<div id="box">
<form action="#" method="post" enctype="multipart/form-data">
  <label for="files">Bitte fach.csv, klasse.csv, klassenzuordnung.csv, lehrer.csv,
stundenplan.csv, unterrichtszuordnung.csv auswählen</label><br>
  <input id="files" name="files[]" class="multifile" type="file" accept=".csv" accept-
charset="utf-8" multiple><br>
  <div>
    <ul id="selectedFiles"></ul>
    <i>Alle Dateien müssen grün markiert sein und keine Datei darf fehlen</i><br>
    <span>Fehlende Dateien:</span><br>
    <ul id="missingFiles"></ul>
  </div>
  <input type="checkbox" id="newYear" name="newYear"><label for="newYear">Neues Schuljahr
starten</label><br>
  <input type="submit" value="Import starten">
</form>
</div>
```

Code Snippet 9: Prüfung Dateinamen - Eingabe

Die bereits erwähnte Prüfung wird nun mittels jQuery durchgeführt.

```
$(document).ready(function(){
  $('#files').change(function(e){
    $('#input[type="submit"]').attr('disabled', 'disabled');
    $('#selectedFiles').html("");
    $('#missingFiles').html("");

    var list = getFilesOfMultipleFileInput(e);
    var correctList = new Array('fach.csv', 'klasse.csv', 'klassenzuordnung.csv', 'lehrer.csv',
'stundenplan.csv', 'unterrichtszuordnung.csv');
    var correctedList = getDifference(list, correctList);
    // selectedFiles
    // missingFiles
    var allCorrect = true;
    for(var i = 0; i < correctedList.length; i++){
      switch(correctedList[i][1]){
        case 0: $('#selectedFiles').append('<li
class="file_correct">'+correctedList[i][0]+'</li>');
          break;
        case 1: $('#selectedFiles').append('<li
class="file_wrong">'+correctedList[i][0]+'</li>');
          allCorrect = false;
          break;
        case 2: $('#missingFiles').append('<li>'+correctedList[i][0]+'</li>');
          allCorrect = false;
          break;
      }
    }
    if(allCorrect){
      $('#input[type="submit"]').removeAttr('disabled');
    }
  });
});
```

Code Snippet 10: Funktion zur Prüfung der Dateinamen

**Zugehörige Funktionen:**

```
function getFilesOfMultipleFileInput(e){
    var list = new Array();
    $('#selectedFiles').html("");
    if(e.target.files){
        for(var i = 0; i < e.target.files.length; i++){
            list[list.length] = e.target.files[i].name;
        }
    }
    return list;
}
```

Code Snippet 11: Unterfunktion zur Prüfung der Dateinamen

```
function getDifference(list1, list2){
    var correctedList = new Array();
    for(var i = 0; i < list1.length; i++){
        var found = false;
        for(var j = 0; j < list2.length; j++){
            if(list1[i] === list2[j]){
                found = true;
            }
        }
        if(found){
            correctedList[correctedList.length] = new Array(list1[i], 0);
        } else {
            correctedList[correctedList.length] = new Array(list1[i], 1);
        }
    }

    for(var i = 0; i < list2.length; i++){
        var found = false;
        for(var j = 0; j < correctedList.length; j++){
            if(list2[i] === correctedList[j][0]){
                found = true;
            }
        }
        if(!found){
            correctedList[correctedList.length] = new Array(list2[i], 2);
        }
    }
    return correctedList;
}
```

Code Snippet 12: Unterfunktion zur Prüfung der Dateinamen

Nach absenden der Dateien werden diese nun an die jeweiligen Repositories übergeben und an die temporär Tabellen gesendet und anschließend bearbeitet.

**Lehrer-Import (Klassen-Import und Fach-Import sind im selben Stil):**

```
public function importFromCSV($file){
    $systempass = md5(md5("schule")."1234");
    $stmt = $this->cube->prepare("INSERT IGNORE INTO Lehrer (Lehrer_ID, numID, Langname, Passwort,
Hash) VALUES(:id, :short, :long, :pass, :ha)");
    if (($handle = fopen($file, "r")) !== FALSE) {
        while (($data = fgetcsv($handle, 1000, ";")) !== FALSE) {
            $stmt->bindValue(':id', utf8_encode($data[1]));
            $stmt->bindValue(':short', $data[0]);
            $stmt->bindValue(':long', utf8_encode($data[2]));
            $stmt->bindValue(':pass', $systempass);
            $stmt->bindValue(':ha', md5(utf8_encode($data[1])));
            $stmt->execute();
        }
        fclose($handle);
    }
}
```

Code Snippet 13: Import Lehrer von CSV

und der deutlich aufwendigere Stundenplan import:

```
public function importFromCSV($unterrichtsZ, $stundenplan, $klassenZ){
    $stmt = $this->cube->prepare("INSERT INTO tmp_unterrichtsZ (id,uid,lehrer,fach)
VALUES(:id, :uid, :l, :f)");
    if (($handle = fopen($unterrichtsZ, "r")) !== FALSE) {
        while (($data = fgetcsv($handle, 1000, ";")) !== FALSE) {
            $stmt->bindValue(':id', $data[0]);
            $stmt->bindValue(':uid', $data[1]);
            $stmt->bindValue(':l', $data[2]);
            $stmt->bindValue(':f', $data[3]);
            $stmt->execute();
        }
        fclose($handle);
    }
    $stmt = $this->cube->prepare("INSERT INTO tmp_stundenplan (uid, tag, stunde)
VALUES(:uid, :d, :h)");
    if (($handle = fopen($stundenplan, "r")) !== FALSE) {
        while (($data = fgetcsv($handle, 1000, ";")) !== FALSE) {
            $stmt->bindValue(':uid', $data[0]);
            $stmt->bindValue(':d', $data[1]);
            $stmt->bindValue(':h', $data[2]);
            $stmt->execute();
        }
        fclose($handle);
    }
    $stmt = $this->cube->prepare("INSERT INTO tmp_klassenZ (klasse, uzid) VALUES(:k, :uzid)");
    if (($handle = fopen($klassenZ, "r")) !== FALSE) {
        while (($data = fgetcsv($handle, 1000, ";")) !== FALSE) {
            $stmt->bindValue(':k', $data[1]);
            $stmt->bindValue(':uzid', $data[0]);
            $stmt->execute();
        }
        fclose($handle);
    }
    //befüllen der tmps abgeschlossen - beginne Übertragung in Stundenplan Tabelle
    $data = $this->cube->query("SELECT tag,stunde,lehrer,klasse,fach FROM tmp_unterrichtsZ LEFT
JOIN tmp_stundenplan ON tmp_unterrichtsZ.uid = tmp_stundenplan.uid LEFT JOIN tmp_klassenZ ON
tmp_unterrichtsZ.id = tmp_klassenZ.uzid");
    $stmt = $this->cube->prepare("INSERT INTO Stundenplan (Tag, Stunde, Lehrer, Klasse, Fach)
VALUES (:d, :h, :l, :k, :f)");
    while($row = $data->fetchObject()){
        $idLehrer = Factory::getRepLehrer()->getIDByNumID($row->lehrer);
        $idKlasse = Factory::getRepKlasse()->getIDByNumID($row->klasse);
        $idFach = Factory::getRepFach()->getIDByNumID($row->fach);

        $stmt->bindValue(':d', $row->tag);
        $stmt->bindValue(':h', $row->stunde);
        $stmt->bindValue(':l', $idLehrer);
        $stmt->bindValue(':k', $idKlasse);
        $stmt->bindValue(':f', $idFach);
        $stmt->execute();
    }
    $this->cube->query("DELETE FROM tmp_unterrichtsZ");
    $this->cube->query("DELETE FROM tmp_stundenplan");
    $this->cube->query("DELETE FROM tmp_klassenZ");
}
```

Code Snippet 14: Import Stundenplan von CSV



7.4.3. STUNDENPLANANZEIGE LADEN

Die Anzeige wird in zwei Schritten durchgeführt.

Schritt 1:

Direkt auf PHP Ebene ohne Asynchrone Requests.

Hier wird nur der normale Stundenplan gelesen und die Anzeige aufgebaut.

```
$this->event = "event_timetable";
$this->data->kw = isset($_GET['kw'])? $_GET['kw'] : date("W");
$this->data->klassen = \Repository\Factory::getRepKlasse()->getAll();
$this->data->klasse = isset($_GET['kl'])? $_GET['kl'] : $this->data->klassen[0]->Klasse_ID;

$this->data->currKw = date("W");
$this->data->stunden = \Repository\Factory::getRepStundenPlan()->getByKlasseOrderByDayAndHour($this->data->klasse);
$this->data->lehrer = \Repository\Factory::getRepLehrer()->getAll();

//Call Daddy
(new \View\UserView()->output("intern", "Supplierplan", $this->data, $this->event);
```

Code Snippet 15: Anzeige Stundenplan

Wie die Funktion bereits im Namen verrät, wird mit *getByKlasseOrderByDayAndHour()* der Stundenplan der Klasse geladen.

```
public function getByKlasseOrderByDayAndHour($klasse){
    $einheitenArr = array();
    $stmt = $this->cube->prepare("SELECT Stundenplan_ID, Tag, Stunde, Lehrer, Klasse, KW, Abwesend,
Fach FROM Stundenplan WHERE Klasse = :klasse AND KW IS NULL");
    $stmt->bindValue(':klasse', $klasse);
    $stmt->execute();
    while ($data = $stmt->fetchObject()){
        $einheitenArr[$data->Stunde][$data->Tag][] = new \Core\Supplierung($data);
    }
    return $einheitenArr;
}
```

Code Snippet 16: Laden des Stundenplans der einzelnen Klassen



Zusätzlich werden noch alle Lehrer geladen und anschließend an das *intern* Template über die *UIView* übergeben.

Ausschnitt aus dem intern-Template:

```
<div id="stunden">
  <?php
    for($i = 1; $i <= 9; $i++){
      for($j = 1; $j <= 5; $j++){
        if(isset($data->stunden[$i][$j])){
          $supplierung = false;
          $lehrerStunden = array();
          foreach($data->stunden[$i][$j] as $einheit){
            if($einheit->getKW()){
              $supplierung = true;
              $lehrerStunden[$einheit->getFachID()][] = $einheit->getLehrerID();
            }
            $lehrerStunden[$einheit->getFachID()] =
            array_unique($lehrerStunden[$einheit->getFachID()]);
          }
          if($supplierung){
            echo "<span class=\"stunde active supplier\" id=\"$j-$i\" data-day=\"$j\" data-hour=\"$i\">";
          } else {
            echo "<span class=\"stunde active\" id=\"$j-$i\" data-day=\"$j\" data-hour=\"$i\">";
          }
          foreach($lehrerStunden as $key => $val){
            echo "<p>.$key.<br>".implode(", ", $val)."</p>";
          }
          echo "</span>";
        } else {
          echo "<span class=\"stunde\" id=\"$j-$i\"></span>";
        }
      }
    }
  ?>
</div>
```

Code Snippet 17: Ausschnitt aus intern-Template



Schritt 2:

In der mit eingefügten JS-Datei wird nun *supplySupplierung()* aufgerufen.

```
function supplySupplierung(){
    var url = document.URL;
    url+= url.indexOf('?') === -1 ? '?' : '&';
    url+='supply=1';
    $.getJSON(url, function(result){
        if(!$.isEmptyObject(result)){
            $(result).each(function(){
                var id = '#'+this.tag+'-'+this.stunde;
                if($(id).hasClass('supplier')){
                    $(id+' p').append(', '+this.lehrer);
                } else {
                    $(id).addClass('supplier');
                    $(id).html('<p>'+this.fach+'<br>'+this.lehrer+'</p>');
                }
                if(!$(id).hasClass('active')){
                    $(id).addClass('active');
                }
            });
        }
    });
}
```

Code Snippet 18: Funktion SupplySupplierung

Diese fügt an die URL den Parameter *supply* an – worauf der Intern-Controller reagiert.

```
if(isset($_GET['supply'])){
    $stunden = \Repository\Factory::getRepStundenPlan()->getByKlasseAndKW($this->data->klasse,
    $this->data->kw);
    $stundenJSON = array();
    foreach($stunden as $stunde){
        $stundenJSON[] = $stunde->toJSON();
    }
    (new \View\JSONView())->output($stundenJSON);
}
```

Code Snippet 19: Aufruf Intern-Controller

Somit werden alle Vertretungen für die Klasse und Kalenderwoche selektiert und in JSON umgewandelt und gesendet (bzw. ausgegeben).

```
public function getByKlasseAndKW($klasse, $kw){
    $stmt = $this->cube->prepare("SELECT Stundenplan_ID, Tag, Stunde, KW, Abwesend, Lehrer, Klasse,
    Fach FROM Stundenplan WHERE Klasse = :k1 AND KW = :kw AND Abwesend = 0");
    $stmt->bindValue(':k1', $klasse);
    $stmt->bindValue(':kw', $kw);
    $stmt->execute();
    while($data = $stmt->fetchObject()){
        if(!isset($this->suppList[$data->Stundenplan_ID])){
            $this->suppList[$data->Stundenplan_ID] = new \Core\Supplierung($data);
        }
        yield $this->suppList[$data->Stundenplan_ID];
    }
}
```

Code Snippet 20: Selektion Vertretungen für Klasse und Kalenderwoche

SupplySupplierung ersetzt nun die alten Daten durch die neuen.

Dieses zwei-Schritt System verhindert eine zu kompliziertes filtern der Selektierten Daten. Weiters ist es somit möglich die an den Client gesendete Daten ohne Last auf den Server zu filtern und ggf. farblich anzupassen.





8. Evaluierung

WORUM GEHT ES IN DIESEM KAPITEL?

8.1. ERGEBNIS	70
8.1.1. <i>Stundenplan</i>	70
8.1.2. <i>Vertretungsplan</i>	71
8.1.3. <i>Webpräsenz (Homepage)</i>	72
8.2. HINDERNISSE UND PROBLEME.....	73
8.2.1. <i>Schulischer Alltag</i>	73
8.2.2. <i>Einhaltung der geplanten Stunden</i>	73
8.3. WERTSCHÖPFUNG	74
8.4. RESÜMEE.....	74

Zum Abschluss wird das Projektergebnis in diesem Kapitel diskutiert. Während der Durchführung dieser Diplomarbeit konnten wir viele wichtige Erfahrungen für die Projektentwicklung sowie Projektabwicklung, Erfahrungen und Erkenntnisse gewonnen werden.

Neben den Errungenschaften im technischen sowie wirtschaftlichen Bereich wird auch auf die Zukunftsaussicht des Projekts eingegangen.



8.1.2. VERTRETUNGSPLAN

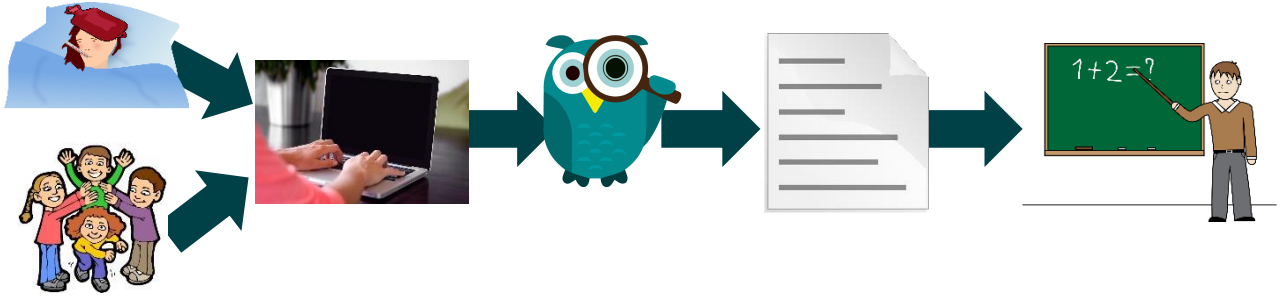


Abbildung 47: Soll-Ablauf Vertretungsplan

Eine Lehrkraft ist krank, auf Fortbildung oder eine gesamte Schulklasse befindet sich demnächst auf Exkursion. Die Direktorin trägt diese Informationen in das System „PRoNMSp“ ein.

Mit dieser Software hat die Direktorin nun einen Überblick, welche Unterrichtsfächer supliert werden müssen. Hier wurde vor allem auf die Benutzerfreundlichkeit sowie auf die Einfachheit der Oberfläche geachtet. Die wichtigsten Bereiche werden mit besonderen, farblichen Akzenten hervorgehoben.

Ein wichtiger Punkt im Zuge dieses Projekts war die Intelligenz der Software. Durch die eingegebenen Informationen teilt einem das Programm mit, welche Stunden vertreten werden müssen und listet einem sogar die verfügbaren Lehrkräfte für diese Unterrichtsstunde auf.

<input type="checkbox"/> Lindinger-Pesendorfer Rosmarie 0/20	<input type="checkbox"/> Bildnerische Erziehung
<input type="checkbox"/> Plaimer Monika 0/20	<input type="checkbox"/> Verb. Übung Berufsorientierung
<input type="checkbox"/> Ebner Josef 0/20	<input type="checkbox"/> Bewegung und Sport
<input type="checkbox"/> Reisinger Christine 1/20	<input type="checkbox"/> Biologie und Umweltkunde
<input type="checkbox"/> Aschauer Leopold 1/20	<input type="checkbox"/> Chemie
<input type="checkbox"/> Bindreiter Josef 1/20	<input type="checkbox"/> Deutsch
<input type="checkbox"/> Dedl Christoph 1/20	<input type="checkbox"/> Englisch
<input type="checkbox"/> Weichselbaumer Peter 2/20	<input type="checkbox"/> English Konversation
	<input type="checkbox"/> Ernährung und Haushalt
	<input type="checkbox"/> Geschichte und Sozialkunde/Politische Bildung
	<input type="checkbox"/> Ganz-Tages-Betreuung
	<input type="checkbox"/> Geographie und Wirtschaftkunde
	<input type="checkbox"/> Geometrisches Zeichnen

E	E	Aschauer Leopold	0/20
IA, LIRO_VODO	KRBA, VODO	Berger Monika	0/20
M	D	Ebner Josef	0/20
O, EBJO, WAJO	BEMO, HAAN	Hader Anna	0/20
D	M	Haider Raimund	0/20
E, BEMO, HAAN, WEPE	BIJO, EBJO, WAJO	Kapfensteiner Geneveva	0/20
GS	WE	Kranzl Barbara	0/20
WEPE	RECH, WEPE	Kurus Doris	0/20
		Lindinger-Pesendorfer Rosmarie	0/20
		Plaimer Monika	0/20
		Wahl Johanna	0/20
		Bindreiter Josef	1/20
REL	BSP	Bürscher Tanja	1/20
PLMO	HAAN, HARA	Dedl Christoph	1/20
GW	BSP	Reisinger Christian	1/20
WEPE	HAAN, HARA	Reisinger Christine	1/20
		Vogelhofer Doris	1/20
		Weichselbaumer Peter	1/20

	1A	1B	2A	2B	3	4	GTS 1	GTS 2	GTS 3	GTS 4	
1	29.10.-4.11.	6.11.-13.11.	15.11.-18.11.	19.11.-25.11.	26.11.-1.12.	BSP ASLE, HARA	ME WAJO	PH EBJO	E KRBA, LIRO_VODO	E KRBA, VODO	Aschauer Leopold 0/20
6	14.9.-20.9.	16.9.-22.9.	16.9.-22.9.	23.9.-1.10.	23.9.-30.9.	D BEMO, HAAN	M EBJO, WAJO	E BÜTA, KRBA, VODO	M BIJO, EBJO, WAJO	D BEMO, HAAN	Berger Monika 0/20
11	9.9.-15.9.	16.9.-22.9.	23.9.-29.9.	30.9.-5.10.	6.10.-12.10.	M BIJO, EBJO, WAJO	BSP HAAN, HARA	BSP HAAN, HARA	D ASLE, BEMO, HAAN, WEPE	M BIJO, EBJO, WAJO	Ebner Josef 0/20
16	14.10.-19.10.	20.10.-26.10.	27.10.-3.11.	4.11.-10.11.	11.11.-17.11.	D WEPE	E BÜTA, HARA, KRBA, VODO	D ASLE, BEMO, HAAN, WEPE	GS WEPE	WE RECH, WEPE	Hader Anna 0/20
21	18.5.-24.5.	25.5.-31.5.	1.6.-7.6.	8.6.-14.6.	15.6.-21.6.						Haider Raimund 0/20
26	22.6.-28.6.	29.6.-5.7.	6.7.-12.7.	13.7.-19.7.	20.7.-26.7.						Kapfensteiner Geneveva 0/20
31	27.7.-2.8.	3.8.-9.8.	10.8.-16.8.	17.8.-23.8.	24.8.-30.8.						Kranzl Barbara 0/20
36	31.8.-6.9.	7.9.-13.9.	14.9.-20.9.	21.9.-27.9.	28.9.-4.10.	BU BEMO	BU BEMO	INF RECHL	REL PLMO	BSP HAAAN, HARA	Kurus Doris 0/20
41	6.10.-12.10.	13.10.-19.10.	19.10.-25.10.	26.10.-1.11.	2.11.-8.11.	ME WAJO	GW WEPE	BE DECH	GW WEPE	BSP HAAAN, HARA	Lindinger-Pesendorfer Rosmarie 0/20
46	9.11.-15.11.	16.11.-22.11.	23.11.-29.11.	30.11.-6.12.	7.12.-13.12.						Plaimer Monika 0/20
51	14.12.-20.12.	21.12.-27.12.						GS VODO			Wahl Johanna 0/20

Abbildung 48: Software TimeOwl (Vertretungsplan)



8.1.3. WEBPRÄSENZ (HOMEPAGE)

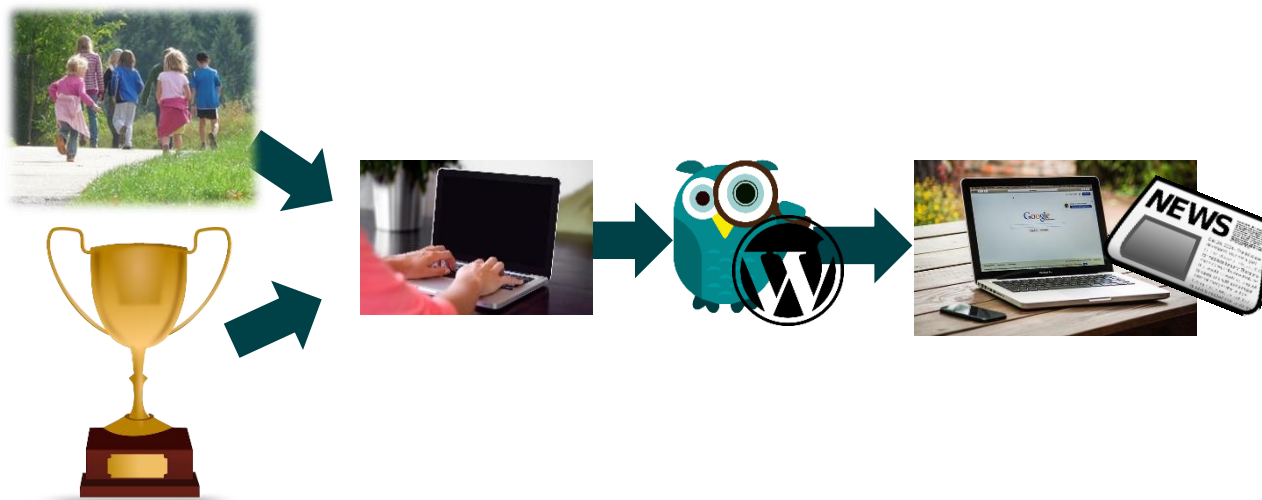


Abbildung 49: Soll-Ablauf Webpräsenz (Homepage)

Um alle aktuellen Ereignisse veröffentlichen zu können, werden bei Ausflügen oder bei Ereignissen in der Schule selbst stets Fotos gemacht. Am Ende dieses Tages kann die Lehrkraft ohne großen Aufwand einen Beitrag über das Geschehen veröffentlichen. Diese benötigt lediglich einen Computer mit Internetzugang sowie den dazugehörigen Benutzerdaten. Nach erfolgreicher Anmeldung muss der Beitrag nur noch geschrieben werden und die dazugehörigen Bilder hochgeladen werden. Mit dem Klick auf Veröffentlichen ist der Beitrag mit sofortiger Wirkung auf der Homepage verfügbar und für alle Besucher sichtbar.

Die Benutzereingabe ist mit diesem neuen System vereinfacht worden, da dieses für sich selbstsprechend ist. Auch dem Design der Homepage wurde ein neuer Anstrich gegeben.

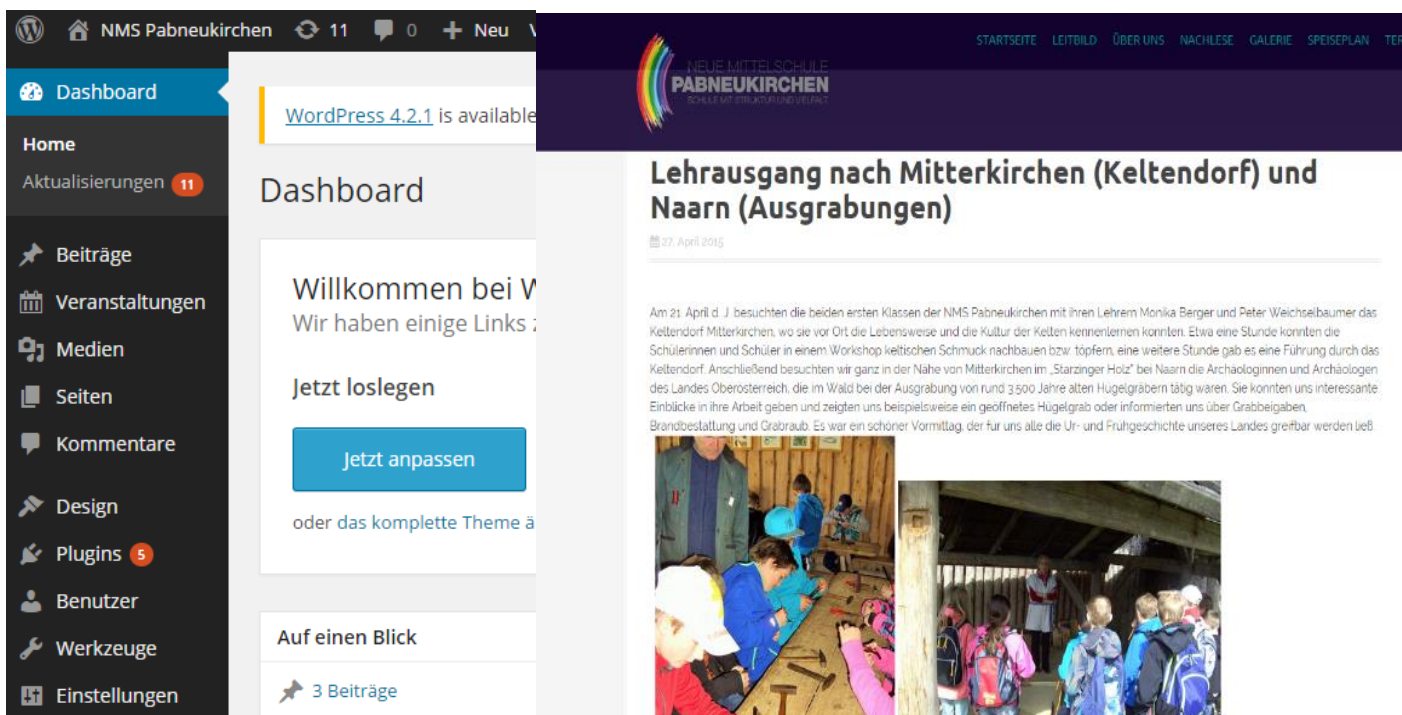


Abbildung 50: Homepage (aktuell)



8.2. HINDERNISSE UND PROBLEME

Im Laufe der Projektarbeit entstehen meist Probleme, welche in unserem Fall alle erfolgreich bewältigt wurden. Die folgenden Schwierigkeiten hatten keinerlei negativen Einfluss auf die Qualität oder den Fortschritt des Projekts. Die größte Herausforderung war die optimale Gestaltung unseres Zeitmanagements.

8.2.1. SCHULISCHER ALLTAG

Das größte Problem stellte der schulische Alltag dar. Neben den täglichen Aufwand für die Schule musste die Diplomarbeit durchgeführt werden, da diese außerhalb der schulischen Aktivitäten erstellt werden muss. Nur durch strenge Selbstdisziplin und Fleiß geht die Arbeit voran.

8.2.2. EINHALTUNG DER GEPLANTEN STUNDEN

Neben dem schulischen Alltag mussten natürlich die geplanten Stunden, welche im Rahmen der Projektinitialisierung festgelegt worden sind, auch eingehalten werden. Dies war eine der größten Herausforderungen, da im Laufe des Jahres viele unerwartete Ereignisse eintraten, wie Prüfungen oder Exkursionen. Da war eine Einhaltung der geplanten Wochenstunden nicht immer möglich. Diese mussten natürlich in den darauf folgenden Wochen wieder hereingearbeitet werden.

Folgende Tabelle soll einen Vergleich zwischen geplanten und verfügbaren Wochenstunden darstellen (Auszug aus der gesamten Tabelle, Zeitraum Februar 2015 in Kalenderwochen).

Diplomand	KW 6		KW 7		KW 8		KW 9	
	<i>Ist-Stunden</i>	<i>Soll-Stunden</i>	<i>Ist-Stunden</i>	<i>Soll-Stunden</i>	<i>Ist-Stunden</i>	<i>Soll-Stunden</i>	<i>Ist-Stunden</i>	<i>Soll-Stunden</i>
Berger Patricia	1.5	6	4.5	4	12.5	12	15	10
Buchmair Tanja	10	10	10	10	9	21	6	10

Tabelle 12: Vergleich Ist zu Sollstunden

Wie in der Tabelle ersichtlich treten einige Abweichungen der geplanten Stunden auf. Dies liegt meist am schulischen Alltag, da für die Prüfungen gelernt werden muss. Auch Krankheitsfälle können die Arbeit beeinträchtigen.



8.3. WERTSCHÖPFUNG

Durch die Entwicklung der Vertretungsplan-Software kann der Auftraggeber seine Lehrkräfte optimal einsetzen. Der gesamte Prozess, welcher auf Papier erfolgte, kann nun elektronisch abgewickelt werden, somit unterstützt unsere Software den umfassenden Ablauf. Durch die künstliche Intelligenz dieser Software wird die Dauer erheblich verkürzt und Kollisionen verhindert.

8.4. RESÜMEE

Die Durchführung dieser Diplomarbeit brachte sehr viele wertvolle Erfahrungen für uns beide. Vor allem die Zusammenarbeit mit der Neuen Mittelschule Pabneukirchen und die Beschäftigung mit den Abläufen an einer Schule aus der Sicht einer Lehrkraft bzw. Direktorin machte die Arbeit sehr interessant und abwechslungsreich.

Eine große Herausforderung stellte die Zeitplanung sowie deren Einhaltung dar. Die Planung sowie die Aufwandschätzung waren schnell erledigt, doch die geplanten Stunden auch wirklich einzuhalten war aufgrund des schulischen Alltags sehr schwierig. Oft kamen unerwartete Prüfungen oder Exkursionen in die Quere. Diese Ereignisse raubten natürlich sehr viel Zeit und schränkten die Arbeitszeit ein. Die verlorenen Stunden mussten natürlich in den nachfolgenden Tagen wieder eingearbeitet werden.

Auch das Risikomanagement hat seine Wirkung gezeigt. Der Fall „Unzureichende Erfahrung der Projektmitglieder“ ist eingetreten, da wir keine Kenntnisse im Bereich der Schulverwaltung hatten. Somit nahm die Einarbeitung in diesen Prozess deutlich mehr Zeit in Anspruch als ursprünglich geplant. Durch die im Voraus gesetzte Gegenmaßnahme, sogenannte Pufferzeiten einzubauen hatte dieses Risiko keinen Einfluss auf den Fortschritt bzw. Erfolg des Projektes und wurde somit erfolgreich bewältigt.

Der neuorganisierte Prozess der Stundenplanerstellung wurde im vollständigen Ausmaß abgeschlossen, wie in den Anforderungen verankert. Auch die Internetpräsenz, die Homepage, wurde im gesamten Umfang fertiggestellt. Die eigens entwickelte Software für die Vertretungsplanerstellung wurde in seinen Grundzügen fertiggestellt. Eine Komponente konnte aus Zeitgründen nicht mehr realisiert werden. Diese fehlende Funktion beeinträchtigt nicht die Funktionsfähigkeit dieser Software.

Der Auftraggeber hat nun die Unterstützung durch unser System „PRoNMSP“, welches es erlaubt, den Verwaltungsaufwand optimal abzuwickeln sowie den Aufwand zu minimieren. Die Abwicklung dieser Prozesse nimmt weniger Zeit in Anspruch und alle Dokumente sind zugleich digitalisiert verfügbar und für alle Lehrkräfte zugänglich.

Es kann somit das Augenmerk vermehrt auf den Schulalltag und die Schüler gelegt werden.



9. Glossar

Begriff	Erklärung
TimeTable	Bestehende Software für die Erstellung der Stundenpläne
TimeOwl	Eigens entwickelte Software für die Vertretungsplanung
Access-Datenbank	Relationales Datenbankmanagementsystem der Microsoft Office-Familie
VisualBasic	Objektorientierte Programmiersprache von Microsoft
Makro	Kleine Programme, die wiederkehrende Aufgaben automatisieren
Repository	Bezeichnung für Ort, wo Daten aufbewahrt werden für einen Nutzerzugriff oder Datenbankzugriff
CMS	Content Management System, welches die Administration von Webinhalten steuert
Query	Abfrage - Anfrage
PSP	Projektstrukturplan





10. Verzeichnisse

10.1. ABBILDUNGSVERZEICHNIS

Abbildung 1: Logo Neue Mittelschule Pabneukirchen	4
Abbildung 2: Neue Mittelschule Pabneukirchen.....	4
Abbildung 3: Patricia Berger.....	5
Abbildung 4: Tanja Buchmair.....	5
Abbildung 5: Projekthierarchie	6
Abbildung 6: Organisationskreislauf	8
Abbildung 7: Prozesse des Ist-Zustandes	9
Abbildung 8: Geschäftsziele.....	12
Abbildung 9: Ziele des Projekts.....	12
Abbildung 10: Use-Case Stundenplanerstellung	18
Abbildung 11: Use-Case Vertretungsplanerstellung	19
Abbildung 12: Use-Case Webpräsenz.....	19
Abbildung 13: Anforderungen - Benutzbarkeit.....	20
Abbildung 14: Anforderungen – Zuverlässigkeit und Stabilität.....	20
Abbildung 15: Projektablauf	22
Abbildung 16: Prozessablauf SCRUM.....	23
Abbildung 17: Kommunikationsmodell – Monatliche Meetings	24
Abbildung 18: Kommunikationsmodell – Vereinbarungen per E-Mail.....	24
Abbildung 19: Kommunikationsmodell – Vereinbarungen per Telefon	24
Abbildung 20: Gantt-Diagramm	26
Abbildung 21: Ist-Stunden Statistik	34
Abbildung 22: Fiktive Gesamtkosten	37
Abbildung 23: Lieferumfang	39
Abbildung 24: Logo HTML.....	42
Abbildung 25: Logo PHP.....	42
Abbildung 26: Logo MySQL.....	42
Abbildung 27: Logo Dropbox.....	42
Abbildung 28: Logo Microsoft Office.....	43
Abbildung 29: Logo Gantt Project.....	43
Abbildung 30: Logo Wordpress.....	43
Abbildung 31: Logo jQuery.....	43
Abbildung 32: Logo Sass	43
Abbildung 33: Logo JavaScript.....	44
Abbildung 34: Logo MySQL Workbench.....	44
Abbildung 35: Logo Source Tree	44
Abbildung 36: Logo Firebug.....	44
Abbildung 37: Apache Software Foundation.....	45



Abbildung 38: Logo Brackets.....	45
Abbildung 39: Logo vmWare Workstation	45
Abbildung 40: Logo Ubuntu.....	45
Abbildung 41: Logo Eclipse Luna PHP	45
Abbildung 42: Systemarchitektur	48
Abbildung 43: Programmarchitektur TimeOwl	49
Abbildung 44: Datenmodell TimeOwl.....	57
Abbildung 45: Soll-Ablauf Stundenplan.....	70
Abbildung 46: Software TimeTable (Stundenplan)	70
Abbildung 47: Soll-Ablauf Vertretungsplan	71
Abbildung 48: Software TimeOwl (Vertretungsplan)	71
Abbildung 49: Soll-Ablauf Webpräsenz (Homepage).....	72
Abbildung 50: Homepage (aktuell)	72

10.2. TABELLENVERZEICHNIS

Tabelle 1: Legende Use-Case Diagramm.....	18
Tabelle 2: Aufwandsschätzung	27
Tabelle 3: Verantwortungsmatrix.....	29
Tabelle 4: Risikomanagement	30
Tabelle 5: Kosten der Arbeitsleistung	35
Tabelle 6: Lizenzkosten	36
Tabelle 7: Sonstige Kosten	36
Tabelle 8: Gesamtkosten	37
Tabelle 9: Leistungsumfang Applikation „Stundenplan“	38
Tabelle 10: Leistungsumfang Applikation „Vertretungsplan“	38
Tabelle 11: Leistungsumfang Homepage	38
Tabelle 12: Vergleich Ist zu Sollstunden	73

10.3. PROGRAMMCODEVERZEICHNIS

Code Snippet 1: index.php.....	50
Code Snippet 2: Controller Factory.....	51
Code Snippet 3: Controller.....	52
Code Snippet 4:Repository Factory.....	53
Code Snippet 5: Repository	54
Code Snippet 6: View	55
Code Snippet 7: Template aktuelle Seite	55
Code Snippet 8: Model – Beispielobjekt Lehrer.....	56
Code Snippet 9: Prüfung Dateinamen - Eingabe	62
Code Snippet 10: Funktion zur Prüfung der Dateinamen.....	62
Code Snippet 11: Unterfunktion zur Prüfung der Dateinamen	63



Code Snippet 12: Unterfunktion zur Prüfung der Dateinamen	63
Code Snippet 13: Import Lehrer von CSV.....	64
Code Snippet 14: Import Stundenplan von CSV	64
Code Snippet 15: Anzeige Stundenplan.....	65
Code Snippet 16: Laden des Stundenplans der einzelnen Klassen	65
Code Snippet 17: Ausschnitt aus intern-Template.....	66
Code Snippet 18: Funktion SupplySupplierung.....	67
Code Snippet 19: Aufruf Intern-Controller	67
Code Snippet 20: Selektion Vertretungen für Klasse und Kalenderwoche	67



10.4. LITERATURVERZEICHNIS

- Neue Mittelschule Pabneukirchen
<http://www.nms-pabneukirchen.at> (Zugriff: 01.05.2015)
- Wikipedia
<https://de.wikipedia.org/wiki> (Zugriff: 01.05.2015)
- Stackoverflow
<http://stackoverflow.com/> (Zugriff: 01.05.2015)

- Schwab, Schwab-Matkovits, Schneider. EDV-Projektentwicklung. Wien: Manz Verlag, 2006
- Wilfried Schneider, Manfred Kronawetter, Betriebliche Organisation Band III mit SbX-CD, Auflage 2011

- PHPDoc
<http://php.net/docs.php> (Zugriff: 02.05.2015)
- Entwicklungsumgebung – Eclipse
<http://eclipse.org/downloads/> (Zugriff: 02.05.2015)
- Datenbank/Workbench – MySQL
<http://dev.mysql.com/downloads/> (Zugriff: 02.05.2015)
- Wordpress Hilfe
<https://de.support.wordpress.com/> (Zugriff: 02.05.2015)

- Computerlexikon für Fachbegriffe
<http://www.computerlexikon.com/> (Zugriff: 02.05.2015)

- Scrum
<https://www.projektmagazin.de/glossarterm/scrum> (Zugriff: 02.05.2015)



11. Anhang

11.1. ABNAHMEVERTRAG



PRoNMSP – Abnahme



Abnahme PRoNMSP

Mit diesem Dokument werden offiziell alle angeführten Inhalte des Projekts PRoNMSP an die Auftraggeberin übergeben.

Mit der Unterzeichnung dieses Dokuments verpflichtet sich die Auftraggeberin die Inhalte anzunehmen und keine weiteren Forderungen zu stellen.

Die Auftragnehmerinnen nehmen ab gewählten Stichtag keine Änderungen mehr vor.

Der Support wird für ein weiteres Jahr gewährleistet. Ebenso wird bei der Abnahme die Funktionsfähigkeit aller übergebener Inhalte geprüft und ggf. korrigiert.

Stichtag: 13.05.2015

Mit Unterzeichnung dieser Version wird der Vorgänger vom 22.04.2015 automatisch als nichtig erklärt.

Inhalte:

- Dokumente
 - Tutorial Homepage
 - Tutorial TimeOwl (Vertretungsplan)
 - Tutorial Stundenplanersteller
- Etwaige genutzte Grafikrechte
- Homepage
 - aktueller Stand
- TimeOwl
 - Fertigstellung mitinbegriffen
- Stundenplanersteller
 - aktueller Stand
- Access Makro für Export

Pabneukirchen, Pegg, 30.4.2015

Ort, Datum

Patricia Berger
Patricia Berger

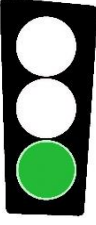
Tanja Buchmair
Tanja Buchmair



Doris Kuras



11.2. STATUSBERICHT

Projektteam	Status
Berger Patricia Buchmair Tanja	 Planmäßig
Datum	
23.03.2015	
Kurzbeschreibung	Berichtszeitraum
Tutorial Homepage Überarbeitung Homepage Supplierplan -Eintragen Supplierung -Sortierung Lehrer -Einfärben	01.03.2015 – 23.03.2015

Ergebnisse

Die Homepage ist auf aktuellstem Stand. Das dazugehörige Tutorial ist soweit fertiggestellt. Hier muss lediglich noch die Formatierung beendet werden.

Das Grundsystem des Supplierplans ist abgeschlossen. Nun folgen noch kleine Zusatzmodule und das Tutorial.

Weitere Schritte

- Fertigstellung Tutorial Homepage
- Einschulung für Verantwortliche der Homepage
- Schreiben Tutorial Supplierplan
- Fortführung Diplomschrift
- Modul der Klassenabwesenheiten
- Tutorial Supplierplan
- Lehrerspezifische Farben im Supplierplan