



DIPLOMARBEIT

HTL PERG FÜR INFORMATIK

Projektteam	Anika Seibezeder Matthäus Pfusterschmid
Projektbetreuer	Prof. Mag. Rupert Obermüller
In Zusammenarbeit mit Betreuer	HTL Perg Prof. Mag. Rosmarie Lindinger-Pesendorfer
Bearbeitungszeitraum	01.10.2015 – 08.04.2016



Danksagung

Unser Dank gilt all jenen, die uns bei der Umsetzung unserer Diplomarbeit unterstützt haben.

Besonders bedanken möchten wir uns bei Prof. Mag. Rupert Obermüller, der uns als Betreuungslehrer zur Seite stand und uns eine Anlaufstelle für technische und inhaltliche Fragen bot.

Auch gegenüber Prof. Mag. Rosmarie Lindinger-Pesendorfer möchten wir unseren Dank aussprechen. Als Lehrerin im Fach Deutsch an der HTL Perg, testete sie unsere Diplomarbeit und gab uns reichlich Feedback, welches uns bei der Realisierung von *anotesa* sehr förderte.

Inhaltsverzeichnis

1	<u>EINLEITUNG</u>	8
1.1	KURZBESCHREIBUNG	8
1.2	ABSTRACT	8
2	<u>PROJEKTDEFINITION</u>	9
2.1	AUFTRAGGEBER	9
2.1.1	KONTAKT	9
2.2	KONTAKTPERSON	9
2.2.1	KONTAKT	9
2.3	BETREUUNGSLEHRER	10
2.3.1	KONTAKT	10
2.4	HIERARCHIE	10
2.5	PROJEKTTEAM	10
2.5.1	DIPLOMANDEN	11
3	<u>ENTSTEHUNG</u>	12
3.1	IST-ZUSTAND	12
3.2	VERUNSICHERUNG BEI DER ENTSTEHUNG	12
3.3	IDEE	13
3.4	ZIELSETZUNG	14
3.5	FUNKTIONSUMFANG	15
3.5.1	FUNKTIONALITÄTEN	15
3.5.1.1	Use-Case-Diagramm	15
3.5.1.2	Funktionalitäten anotesa	15
3.5.2	FUNKTIONALITÄT-LEHRER	16
3.5.2.1	Login	16
3.5.2.2	Test anlegen	16
3.5.2.3	Eigene Tests bewerten	16
3.5.3	FUNKTIONALITÄT-SCHÜLER	17
3.5.3.1	Überschneidende Funktionalitäten	17
3.5.3.2	Bewertung ansehen	17
3.5.3.3	Notenspiegel einsehen	17
3.5.4	FUNKTIONALITÄT-ADMIN	18
3.5.4.1	Überschneidende Funktionalitäten	18
3.5.4.2	Tests löschen bzw. bearbeiten	18
4	<u>PLANUNG</u>	19



4.1	PROJEKTSTRUKTURPLAN	19
4.2	AUFGABEN UND VERANTWORTUNGEN	20
4.2.1	VERANTWORTUNGSMATRIX	20
4.2.2	AUFGABEN <i>ANOTESA</i>	20
4.3	MEILENSTEINE	21
4.4	ZEITPLANUNG	22
4.4.1	SCRUM	22
4.4.2	GANTT-PLAN	22
4.4.2.1	Beschreibung	22
4.4.2.2	GANTT-Plan <i>anotesa</i>	22
4.5	PFLICHTENHEFT	23
4.5.1	ZIELBESTIMMUNGEN	23
4.5.2	GRENZKRITERIEN	23
4.5.2.1	Musskriterien	23
4.5.2.2	Abgrenzungskriterien	23
4.5.3	PRODUKTEINSATZ	23
4.5.3.1	Anwendungsbereiche	23
4.5.3.2	Zielgruppen	23
4.5.4	PRODUKTKONFIGURATION	24
4.5.4.1	Software, Hardware	24
4.5.4.2	Schnittstellen	24
4.6	LIEFERUMFANG	25
4.6.1	SOFTWARE	25
4.6.2	DOKUMENTE	25
5	<u>SOFTWARE – VERWENDETE TECHNOLOGIEN</u>	26
5.1	C#	26
5.1.1	ASP.NET	27
5.2	HTML	27
5.3	CSS	28
5.4	JAVASCRIPT	29
5.4.1	JQUERY	30
5.4.2	AJAX	30
5.5	BOOTSTRAP	31
5.5.1	MORRIS.JS	32
5.5.2	DATETIMEPICKER.JS	32
5.6	WINDOWS SERVER 2012	33
5.7	GOOGLE DRIVE	34
5.8	VISUAL STUDIO	34
6	<u>PROGRAMMIERUNG</u>	35
6.1	DARSTELLUNG	35

6.1.1	LOGIN	35
6.1.2	NAVIGATION-BAR	36
6.1.2.1	Benachrichtigungen.....	36
6.1.2.2	User.....	37
6.1.2.3	Tasks (nur für Lehrer).....	37
6.1.2.4	Keine Daten vorhanden.....	38
6.1.3	DASHBOARD	38
6.1.3.1	Dashboard Schüler	38
6.1.3.2	Dashboard Lehrer	41
6.1.3.3	Dashboard Administrator	43
6.1.4	TESTS ANSEHEN	44
6.1.5	TEST-DETAILS.....	45
6.1.5.1	Lehrer	46
6.1.5.2	Administrator	48
6.1.5.3	Schüler.....	48
6.1.6	TESTS HINZUFÜGEN	49
6.2	INSTALLATION	50
6.2.1	SERVER.....	50
6.2.1.1	Add Roles and Features Wizard	50
6.2.1.2	IIS-Manager	51
6.2.1.3	Architektur Windows Server 2012 R2.....	52
6.2.2	WEBSEITE.....	53
6.3	RISIKEN	55
6.3.1	RISIKEN IM ALLGEMEINEN	55
6.3.2	RISIKEN ANOTESA.....	55
6.4	IMPLEMENTIERUNG.....	56
6.4.1	DATENMODELL.....	56
6.4.1.1	Übersicht Tabellen.....	57
6.4.2	ASP.NET	58
6.4.2.1	Test hinzufügen.....	58
6.4.2.2	Test bewerten	59
6.4.2.3	Test bearbeiten.....	59
6.4.3	JAVASCRIPT	60
6.4.3.1	Dashboard.....	60
6.4.3.2	Benachrichtigungen.....	61
6.4.3.3	Tasks.....	62
6.4.4	ANMELDUNG MIT IF-USERN.....	63
6.4.4.1	Web.Config	64
6.4.4.2	LoginController.....	65
7	AUFGETRETENE PROBLEME	66
7.1	ALLGEMEINE PROBLEME.....	66
7.1.1	NICHT KLAR DEFINIERTE LÖSUNG	66



7.2 PROBLEME WEBSEITE	66
7.2.1 LDAP-VERBINDUNG ERSTELLEN	66
7.2.2 FEHLERHAFTER ERREICHBARKEIT DER WEBSEITE	66
8 QUALITÄTSSICHERUNG	67
9 TESTUNG	68
9.1 TESTLAUF MIT SCHÜLER	68
9.1.1 DURCHFÜHRUNG	68
9.1.2 UMFRAGE	68
9.1.3 ANALYSE	69
9.1.4 REZENSION LEHRER	71
9.2 TESTFÄLLE	72
9.2.1 ÜBERSICHT TESTFÄLLE	72
9.2.2 TESTFALL 1 – EINLOGGEN MIT IF-USER	73
9.2.3 TESTFALL 2 – DASHBOARD FUNKTIONALITÄTEN	73
9.2.4 TESTFALL 3 – TEST ERSTELLEN	74
9.2.5 TESTFALL 4 – TEST BEARBEITEN	74
9.2.6 TESTFALL 5 – TEST BEWERTEN	75
9.2.7 TESTFALL 6 – TEST LÖSCHEN	75
9.2.8 TESTFALL 7 – KLIKK AUF BENACHRICHTIGUNGEN	76
9.2.9 TESTFALL 8 – KLIKK AUF EINEN TASK	76
10 IMPRESSUM	77
11 RESÜMEE	78
11.1 RESÜMEE – ANIKA SEIBEZEDER	78
11.2 RESÜMEE – MATTHÄUS PFUSTERSCHMID	78
12 ANHANG	79
12.1 ZUSTÄNDIGKEITEN	79
12.1.1 MATTHÄUS PFUSTERSCHMID	79
12.1.2 ANIKA SEIBEZEDER	79
12.2 VERZEICHNISSE	80
12.2.1 QUELLENVERZEICHNIS	80
12.2.2 ABBILDUNGSVERZEICHNIS	82
12.2.3 TABELLENVERZEICHNIS	83

Eidesstattliche Erklärung

Hiermit versichern wir, die vorliegende Arbeit selbstständig, ohne fremde Hilfe und ohne Benutzung anderer, als der von uns angegebenen Quellen, angefertigt zu haben. Alle Stellen, die wörtlich oder sinngemäß aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

Perg, _____ Unterschrift _____

Perg, _____ Unterschrift _____



Gender-Erklärung

Für die bessere Lesbarkeit werden in dieser Diplomschrift personenbezogene Bezeichnungen, die sich zugleich auf Frauen und Männer beziehen, generell nur in der im Deutschen üblichen maskulinen Form angeführt.

Dies soll jedoch keinesfalls eine Geschlechterdiskriminierung oder eine Verletzung des Gleichheitsgrundsatzes zum Ausdruck bringen.

Perg, _____ Unterschrift _____

Perg, _____ Unterschrift _____

1 Einleitung

1.1 Kurzbeschreibung

Beschäftigt mit dem Finden einer Diplomarbeit im 4. Jahrgang, hatten wir den Einfall, eine Web-Applikation dahingehend zu erstellen, dass Schularbeiten und Tests anonym abgehalten werden können. Die Webseite wird sowohl von Professoren, als auch von Schülern verwendet. Wobei ein Professor einen Test erstellt und die Schüler daraufhin ihre anonymen, identifizierbaren Tiernamen erhalten, welche sie bei der Arbeit angeben. Sofort nach der Beurteilung besteht für Schüler die Möglichkeit, auf ihren Laptops, PCs oder auch Smartphones, mittels einer Webseite, ihre Benotungen einsehen zu können. Dazu bekommen sie auch eine Benachrichtigung, dass dieser Test nun bewertet worden ist.

Ziel der Diplomarbeit ist, die Erleichterung der Beurteilung für Professoren und die Einfachheit der Testrückgabe für Schüler. Darüber hinaus gibt diese Methode *Prüfungen* anonym abzuhalten, den Schülern ein Gefühl einer fairer Benotung, bei der nur die Leistung im Vordergrund steht.

1.2 Abstract

Busy with finding a diploma thesis in the 4th grade, we had the idea to create a web application in which exams and tests can be conducted anonymously. The website is used by both professors and students. While a professor creates a test or an exam, the students receive their anonymous identifiable names, which identifies their work. Immediately after the assessment, students have the ability to view their evaluations on their laptops, PCs or smartphones via a website. They also get a notification that the test or exam has now been graded.

The aim of the thesis is to facilitate the assessment of tests and exams for professors and the simplicity of the test return for students. Moreover, the method, to hold examinations anonymously, gives the students an impression of fair marks, in which only the ability of the students is considered.



2 Projektdefinition

2.1 Auftraggeber

Die HTL Perg, die unter der Leitung von Herrn Direktor Dipl.-Ing. Christian Reisinger steht, ist bei unserer Diplomarbeit der Auftraggeber.

2.1.1 Kontakt

Höhere Technische Bundeslehranstalt Perg
Machlandstraße 48
4320 Perg

Tel.: +43 7262 53926
Email: office@htl-perg.ac.at



Abbildung 1: Logo HTL Perg

2.2 Kontaktperson

Frau Professor Mag. Rosmarie Lindinger-Pesendorfer stellt unsere Kontaktperson dar und wird dieses Projekt auch bei ihren Prüfungen einbinden.

2.2.1 Kontakt

Prof. Mag. Rosmarie Lindinger-Pesendorfer
E-Mail: r.lindinger@htl-perg.ac.at



Abbildung 2 : Prof. Mag. Rosmarie Lindinger-Pesendorfer

2.3 Betreuungslehrer

Als Programmierlehrer seit der vierten Klasse steht uns Herr Professor Mag. Obermüller auch als Betreuungslehrer zur Seite.

2.3.1 Kontakt

Prof. Mag. Rupert Obermüller
E-Mail: r.obermueller@htl-perg.ac.at



Abbildung 3 : Prof. Mag. Rupert Obermüller

2.4 Hierarchie

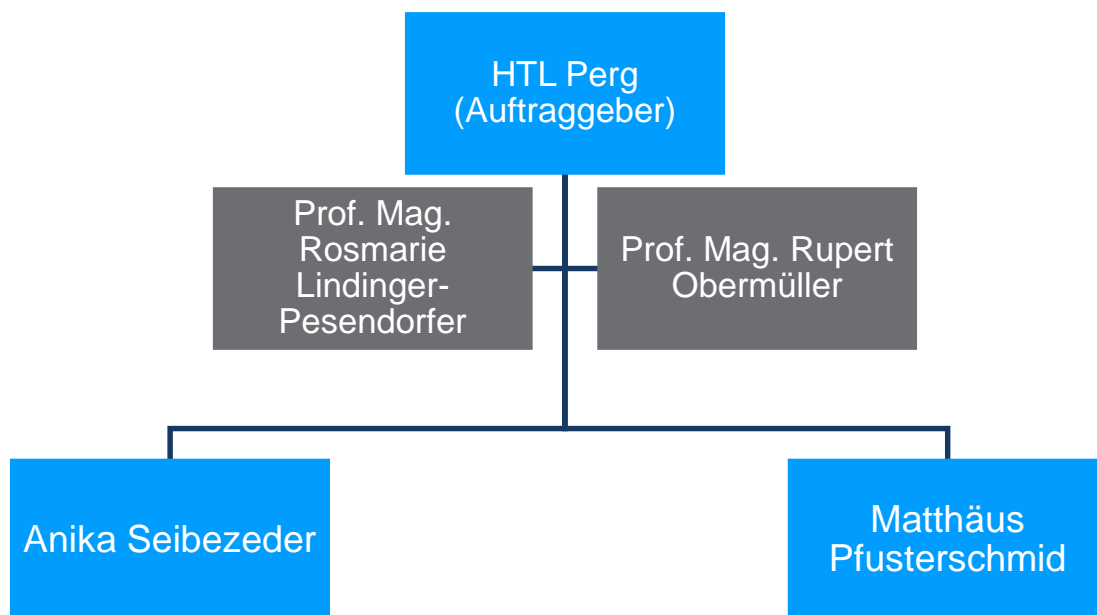


Abbildung 4 : Hierarchie

2.5 Projektteam

Im Rahmen der Reife- und Diplomprüfung an der HTL in Perg im Schuljahr 2015/2016 wurde diese Diplomarbeit von folgenden Diplomanden abgeschlossen:



2.5.1 Diplomanden



Abbildung 5: Anika Seibezeder

*Anika
Seibezeder*

Funktionalität der Webseite

Geburtsdatum	26.09.1996
Kontakt	Uferstraße 32 4432 Ernthofen ☎ 0681 20 30 52 55 ✉ anika.seib@gmail.com
Berufspraxis	August 2013 MAN Trucks & Bus Juli 2014 GRZ IT Center Juli 2015 GRZ IT Center



Abbildung 6 : Matthäus Pfusterschmid

*Matthäus
Pfusterschmid*

Server- und Datenanbindung

Geburtsdatum	23.11.1996
Kontakt	Umspannwerkstraße 9/3 4432 Ernthofen ☎ 0660 715 81 15 ✉ mpfusterschmid@gmail.com
Berufspraxis	Juli 2014 LineMetrics Juli 2015 Auris IT Consulting

3 Entstehung

3.1 Ist-Zustand

Bei schriftlichen Tests und Schularbeiten fühlen sich manche Schüler nicht fair beurteilt, ob zurecht oder unrecht. Auch Lehrkräften fällt es oft schwer, eine passende Note zu vergeben. So verbringen sie etliche Stunden mit der Beurteilung diverser schriftlicher Arbeiten.

Unsere Deutsch Professorin Prof. Mag. Rosmarie Lindinger-Pesendorfer muss genau diese Probleme bei jeder Schularbeit bewältigen: Viele Schüler bitten nach der Rückgabe um eine bessere Note, da sie behaupten, falsch beurteilt worden zu sein. Auch muss sie oft Texte von anderen Professoren lesen lassen, weil sie sich nicht auf eine Note festlegen kann.

Die Situation einer Schularbeitsrückgabe sieht so aus, dass eine komplette Unterrichtseinheit erhalten muss, um Arbeiten zurückzugeben. Meistens kommt es zur Verzögerung, weil die Schüler diverse Fragen zur Beurteilung haben.

Diese Angelegenheiten sind mühsam und kosten wertvolle Zeit, die man für den Unterricht verwenden könnte.

3.2 Verunsicherung bei der Entstehung

Während der Entstehung wurden viele Gedanken ausgetauscht. Dabei kamen auch teilweise verunsichernde Fragen auf. Die häufigste Frage war, wie man sicher gehen kann, dass die Schüler, den ihnen zugeteilten Tiernamen nicht austauschen um sich ihre Noten selbst richten zu können. Jedoch könnte man genauso die echten Namen vertauschen, welche man zur Identifikation des Schülers auf den Test schreibt. Um das Problem von vornherein zu beseitigen wird vermerkt, dass die Schüler den Tiernamen im Programm nicht ändern können und bei jedem neuen Test, einen neuen Tiernamen zugeteilt bekommen.



3.3 Idee

Die Idee der anonymen Testverwaltung entstand über mehrere Jahre. Bereits in den ersten HTL-Jahren waren etliche Schüler nicht davon überzeugt, fair benotet worden zu sein. Außerdem gab es bei Schularbeitsrückgaben immer wieder die Vermutungen, dass einige Schüler die Lieblinge der Professoren seien. Auch seitens der Professoren wird zugegeben, dass es ihnen manchmal schwer fällt, eine bestimmte Note zu vergeben, wenn ein Schüler im Unterricht überdurchschnittlich mitarbeitet und jede Übung mit Bravour meistert; bei der eigentlichen Leistungsfeststellung jedoch nicht so glanzvoll abschneidet. So kann es zu Notenschwankungen kommen, unabhängig von der Leistung.

Hauptsächlich nach dem Schreiben bzw. der Rückgabe einer Schularbeit wurde heftig diskutiert, wie man solche Probleme aus der Welt schaffen kann. Doch nur langsam kristallisierte sich die Idee, Arbeiten anonym zu schreiben, aus zahlreichen Gesprächen heraus. Die Anonymität soll den Schülern die Gewissheit geben, dass rein die Leistung der abgegebenen Schularbeiten oder Tests zählt, und keine persönlichen Einflüsse seitens der Professoren miteinwirken.

Beschäftigt mit dem Finden einer Diplomarbeit im 4. Jahrgang hatten wir den Einfall, dies mit einer Web-Applikation, welche von Professoren und Schülern verwendet wird, zu verbinden, um den Professoren die Testrückgabe zu erleichtern. Die Resonanz sowohl seitens der Professoren, als auch der Schüler war überwiegend positiv, wobei es keine eindeutige Gegenstimme zu der genannten Applikation gab.

Nach Rücksprache mit dem Direktor der HTL Perg, bekamen wir auch von oberster Stelle grünes Licht. Sowohl schulintern, als auch rechtlich wurden uns ab sofort keine Steine mehr in den Weg gelegt.

3.4 Zielsetzung

- Gesamtziel ist eine Webseite, welche den Lehrkräften die Beurteilung erleichtert und den Schülern ein Gefühl von fairer Benotung geben soll.
- Einfaches und ansprechendes Design ist Voraussetzung.
- Für Lehrer besteht die Möglichkeit
 - Tests zu erstellen
 - Tests zu beurteilen bzw. Noten zu vereinbaren
 - Tests einsehenzu können.
- Außerdem soll ein Überblick geschaffen werden, wie viel Zeit dem Lehrer zur Benotung noch bleibt. Dies wird mithilfe eines Ladebalkens, welcher nach sieben Tagen vollläuft, verwirklicht.
- Schüler müssen ihre einzelnen Testnoten einsehen können, sobald ein Lehrer die Noten eines Tests freigibt.
- Schüler sollen
 - eine Notenübersicht bzw.
 - einen Notenspiegeleinsehen können.
- Schüler müssen eine Benachrichtigung bekommen, falls ein Test bewertet worden ist.
- Für den Admin soll es möglich sein, alle Funktionen zu verwalten bzw. verwenden. Jeder vorhandene Test kann vom Admin bearbeitet oder gelöscht werden.



3.5 Funktionsumfang

3.5.1 Funktionalitäten

3.5.1.1 Use-Case-Diagramm

Das Use-Case-Diagramm dient zur Veranschaulichung der Funktionen des Programms und der Personen, die eine jeweilige Funktion anwenden können. Use Cases symbolisieren einen Vorgang, der durch den Benutzer ausgelöst wurde und zeigen Anwendungsfälle auf hohem Abstraktionsniveau, d.h. ohne sich um Details zu kümmern, die für die gezeigte Transaktion nicht relevant sind. [20]

3.5.1.2 Funktionalitäten anotesa

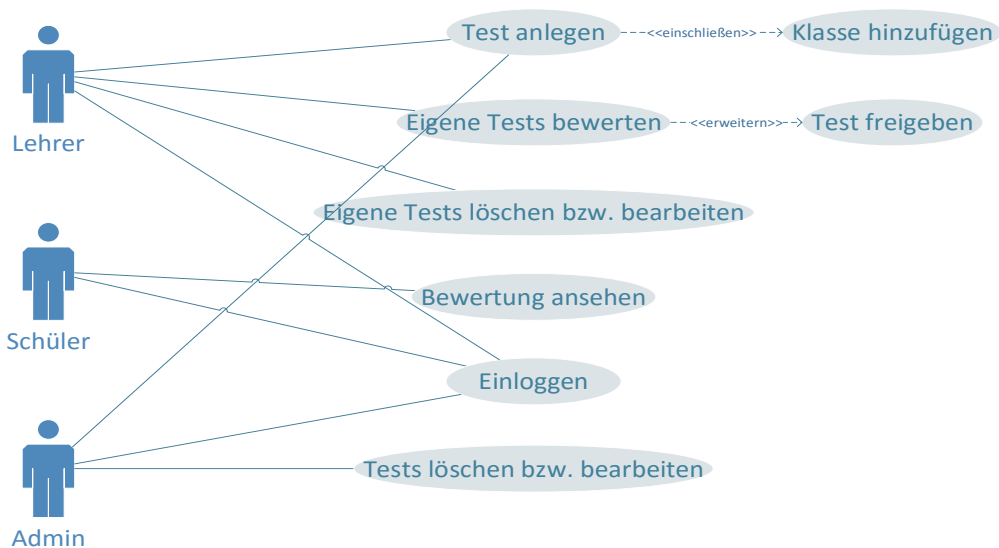


Abbildung 7 : Use-Case-Diagramm anotesa



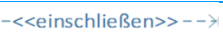
	<p>Akteur: Person oder Prozess, welche/welcher mit dem System interagiert. Symbol ist üblicherweise ein Strichmännchen oder andere Männchen ähnlicher Zeichnungen.</p>
	<p>Use Case: Intern (nicht dargestellt) können in einem Use-Case viele Aktionen ablaufen. Übliches Symbol ist das Oval; Use-Cases können auch als Klasse dargestellt werden.</p>
	<p>include/extend: Zeigen auf, dass zusätzlich zum Use-Case ein weiterer Anwendungsfall ausgeführt wird bzw. optional ausgeführt werden kann.</p>

Tabelle 1 : Use Case Legende

Daraus ergeben sich folgende Funktionalitäten:

3.5.2 Funktionalität-Lehrer

3.5.2.1 Login

Lehrer haben die Möglichkeit, sich mit ihren Login-Daten anzumelden. Dies kann im Netzwerk der HTL Perg oder zu Hause per VPN-Verbindung erledigt werden. Die Login-Daten entsprechen den Zugangsdaten der HTL Perg und sind streng vertraulich. Folgt die Anmeldung zum ersten Mal, so wird im Hintergrund ein spezifischer anotesa-User erstellt, der mit anderen Usern im Programm interagieren kann.

3.5.2.2 Test anlegen

Nach einem erfolgreichen Login erscheint das Dashboard. Auf der links platzierten *Navigation-Bar* hat der Benutzer die Möglichkeit, einen neuen Test anzulegen. Folgende Daten benötigt man, um einen Test zu erstellen: Gegenstand, Datum und Uhrzeit und Klasse; der Lehrer hat die Wahl zwischen allen Gegenständen, die ihm zugeteilt sind. Zusätzlich wird im Hintergrund jeder Schüler, der ausgewählten Klasse einem zufällig ausgewählten Tiernamen zugeordnet.

3.5.2.3 Eigene Tests bewerten

Diese Funktion sollte ausschließlich nach der Korrektur verwendet werden. Der Lehrer ruft die Seite mit allen Tiernamen, welche bei der Testerstellung zugeordnet worden sind, auf und aktualisiert die Noten der zugehörigen Schüler. Außerdem hat er die Möglichkeit, eine zusätzliche Notiz zu hinterlassen. Die Freigabe der Beurteilung, gilt als Bekanntgabe der Noten an die Schüler. Sofern die Note nicht programmtechnisch oder von Fremdeinwirkung geändert wurde, kann sich diese nicht mehr zum Negativen ändern.


Akteur	 Lehrer	Includes	extends
Use Case 1	Login		
Use Case 2	Test anlegen	Klasse und Gegenstand hinzufügen	
Use Case 3	Eigene Tests löschen bzw. bearbeiten		
Use Case 4	Eigene Tests bewerten		Test freigeben

Tabelle 2 : Use Case Lehrer



3.5.3 Funktionalität-Schüler

3.5.3.1 Überschneidende Funktionalitäten

Einige Funktionalitäten überschneiden sich mit denen der Lehrer. (siehe Login auf Seite 16)

3.5.3.2 Bewertung ansehen

Wurde von einem Lehrer die Beurteilung freigegeben, ist es dem Schüler möglich, seine eigene Beurteilung einzusehen. Nach dem Auswählen des jeweiligen Tests, hat der Schüler Einblick über die Benotung und einer gegebenenfalls schriftlichen Beurteilung.

3.5.3.3 Notenspiegel einsehen

Der Schüler hat die Möglichkeit, sich einen Überblick über alle Noten zu verschaffen. Dabei wird ein übersichtliches Diagramm erstellt, welches die Noten beinhaltet.


Akteur	 Schüler	Includes	extends
Use Case 1	Login		
Use Case 2	Bewertung ansehen		
Use Case 3	Notenspiegel ansehen		

Tabelle 3 : Use Case Schüler

3.5.4 Funktionalität-Admin

3.5.4.1 Überschneidende Funktionalitäten

Einige Funktionalitäten überschneiden sich mit denen der Lehrer. (siehe Login auf Seite 16 und Test anlegen auf Seite 16)

3.5.4.2 Tests löschen bzw. bearbeiten

Dem Admin ist es möglich, alle vorhandenen Tests zu löschen. Dies sollte nur vorgenommen werden, falls irrtümlich Tests erstellt worden sind oder Tests an ungültigen Daten erfasst wurden.


Akteur	 Admin	Includes	extends
Use Case 1	Login		
Use Case 2	Test anlegen	Klasse und Gegenstand hinzufügen	
Use Case 3	Tests löschen bzw. bearbeiten		
Use Case 4	Tests bewerten		Test freigeben

Tabelle 4 : Use Case Admin



4 Planung

4.1 Projektstrukturplan

Ein wesentlicher Bestandteil eines Projektes ist der Projektstrukturplan, der die Aufgaben des Teams in Abschnitte unterteilt. Visuell wird das Projekt in einzelne Blöcke unterteilt.

Der nachfolgende Projektstrukturplan beschreibt unser Projekt inklusive Dokumentation und Umfrage-Elemente. [20]

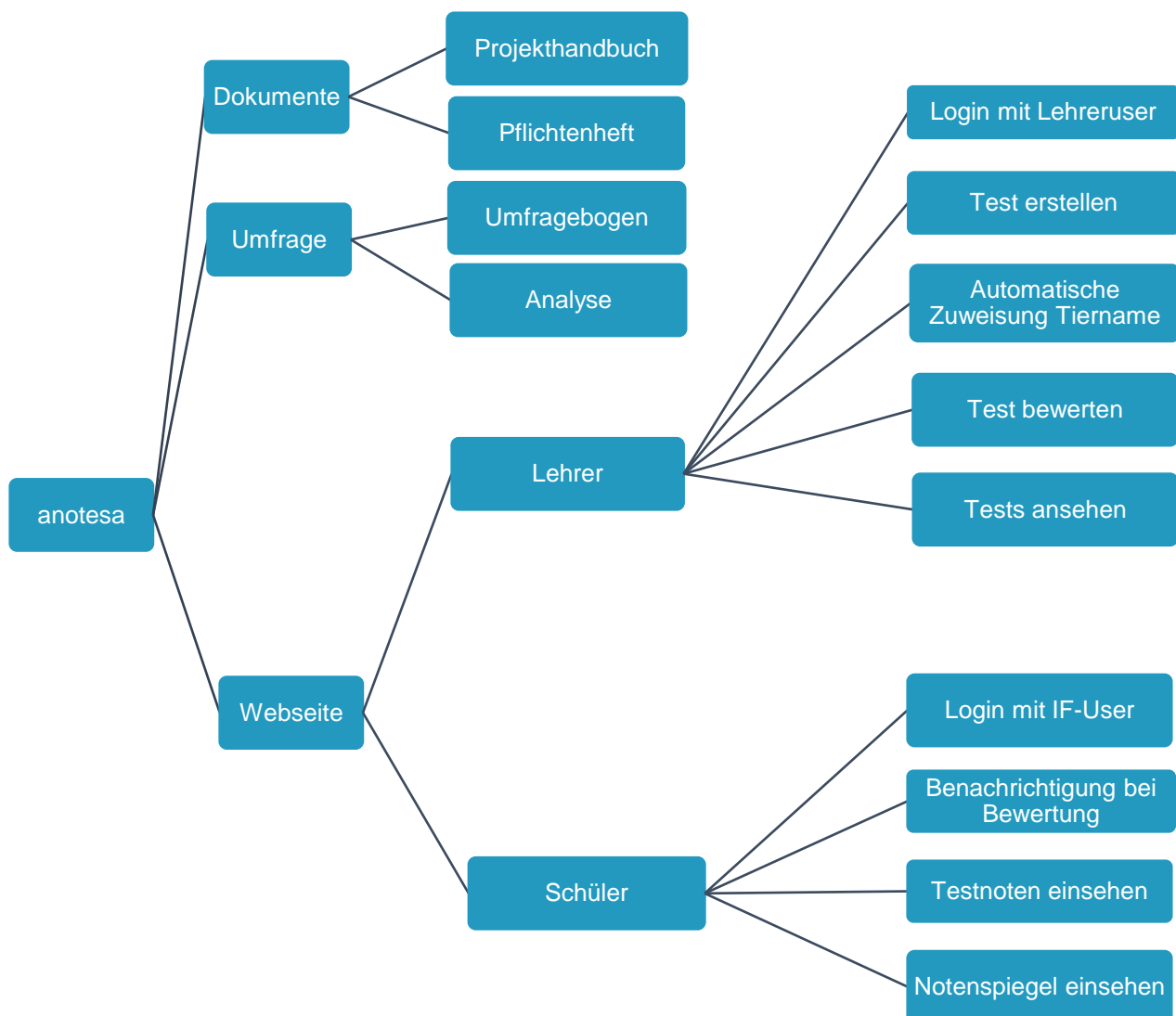


Abbildung 8 : Projektstrukturplan anotesa

4.2 Aufgaben und Verantwortungen

4.2.1 Verantwortungsmatrix

Die Verantwortungsmatrix ist eine Erweiterung der Funktionenmatrix. In den Kreuzpunkten werden zusätzlich die Verantwortlichkeiten für eine bestimmte Aufgabe dargestellt. Eine einfache Form der Verantwortungsmatrix ist die IVM-Matrix. Das am Kreuzungspunkt eingetragene Symbol oder Zeichen gibt Auskunft über die Art der Mitwirkung einer Person an einer Aufgabe. [20]

4.2.2 Aufgaben *anotesa*

In der folgenden IVM-Matrix soll erläutert werden, inwiefern sich die Zuständigkeiten auf die jeweiligen beteiligten Personen im Projekt aufteilen. Außerdem soll eine Übersicht gegeben werden, welche Personen informiert worden sind bzw. die Verantwortung einer bestimmten Aufgabe erhalten haben.

- I ... „wird informiert“
- V ... „**V**erantwortlicher“
- M ... „**M**itarbeiter“

Aufgabe	AS	MP	RO	RL
Projekthandbuch	M	VM		I
Pflichtenheft	M	VM	I	I
Server aufsetzen		VM	I	
Datenbank einrichten		VM	I	
Datenbankmodell erstellen	M	VM		
Login	VM		I	
Testerstellung	VM		I	
Testbewertung	VM		I	
Test einsehen	VM		I	
Umfrage	M	V		I
Analyse	I	V		M

Tabelle 5 : IVM-Matrix *anotesa*

Beteiligte Personen:

- AS ... Anika Seibezeder, Diplomand
- MP ... Matthäus Pfusterschmid, Diplomand
- RO ... Prof. Mag. Rupert Obermüller, Betreuungslehrer
- RL ... Prof. Mag. Rosemarie Lindinger-Pesendorfer



4.3 Meilensteine

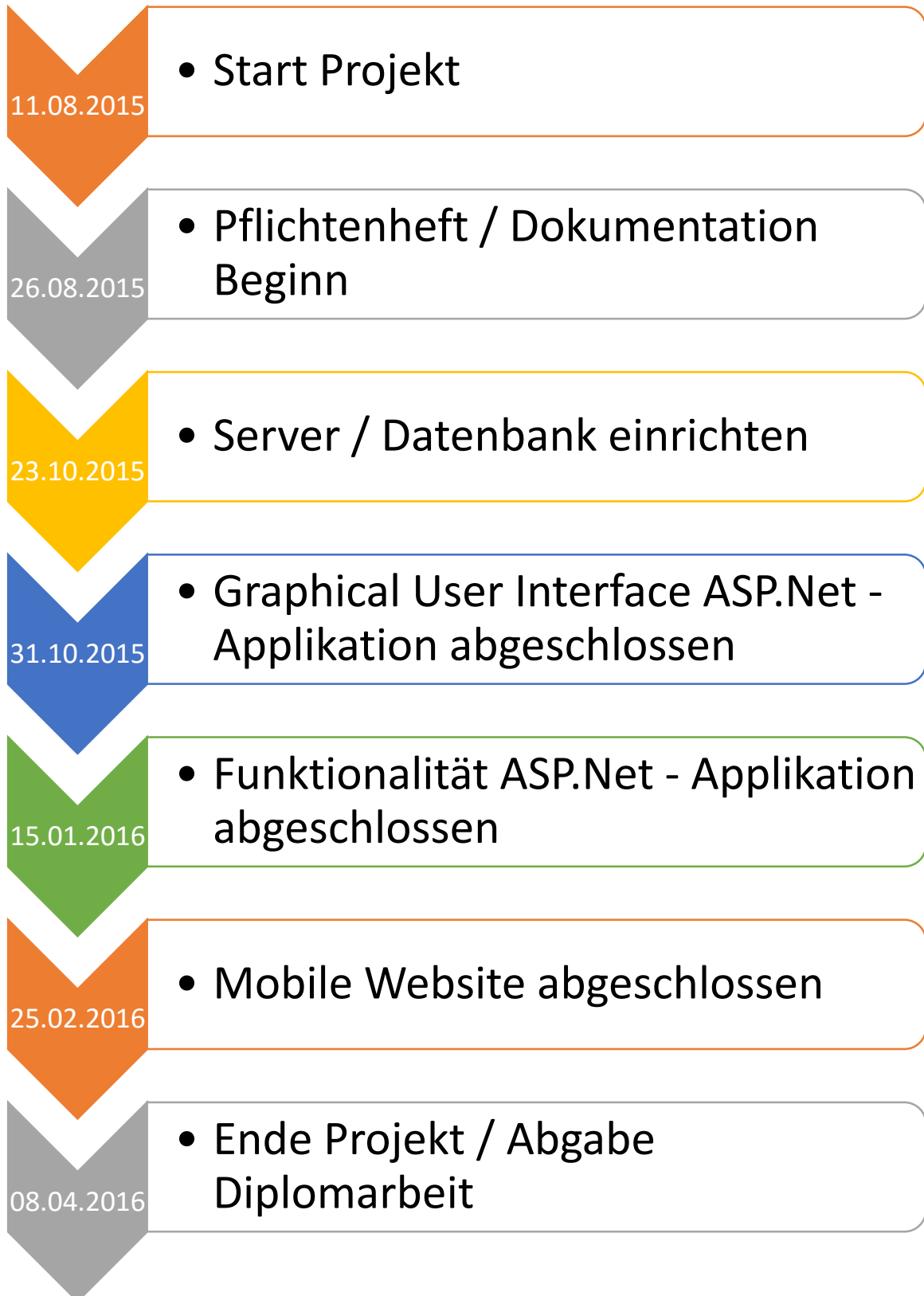


Abbildung 9 : Meilensteine anotesa

4.4 Zeitplanung

4.4.1 SCRUM

Scrum ist ein agiler Prozess, welcher häufig für die Produktentwicklung, vor allem Software-Entwicklung zum Einsatz kommt. Zusätzlich ist Scrum eine Projekt-Management-Methode, die im Projekt mit aggressiven Fristen, komplexen Anforderungen und ein gewisses Maß an Einzigartigkeit anwendbar ist. In Scrum Projekten wird mit einer Reihe von Iterationen, genannt Sprints, gearbeitet. Jeder Sprint ist in der Regel zwei bis vier Wochen lang. Vor jedem Sprint wird in einem Meeting festgelegt, was zu erledigen ist, sowie ein Überblick über den vorherigen Sprint geschaffen. Da wir mit SCRUM bereits seit der 3. Klasse vertraut sind, wandten wir diese Methode bei *anotesa* an. [20]

4.4.2 GANTT-Plan

4.4.2.1 Beschreibung

Balkendiagramme (Gantt-Diagramme) dienen der Planung, Steuerung und Überwachung von Projekten. In den Diagrammen kann abgelesen werden, wann Aktivitäten beginnen, wie lange sie dauern und wann sie enden.

Auf der horizontalen Achse (x-Achse) des Diagramms wird eine Zeitachse aufgetragen, welche den Zeitraum der gesamten Projektdauer umfasst. Entlang der senkrechten Achse (y-Achse) wird eine Liste der durchzuführenden Arbeitspakete oder Teilaufgaben (je nach gewünschter Detaillierung) erstellt. Wenn möglich, stehen früher durchzuführende Arbeiten weiter oben. [20]

4.4.2.2 GANTT-Plan *anotesa*

Der folgende Gantt-Plan ist ein Ausschnitt aus der Zeitplanung von *anotesa* und gibt die Hauptarbeiten am Server wieder.

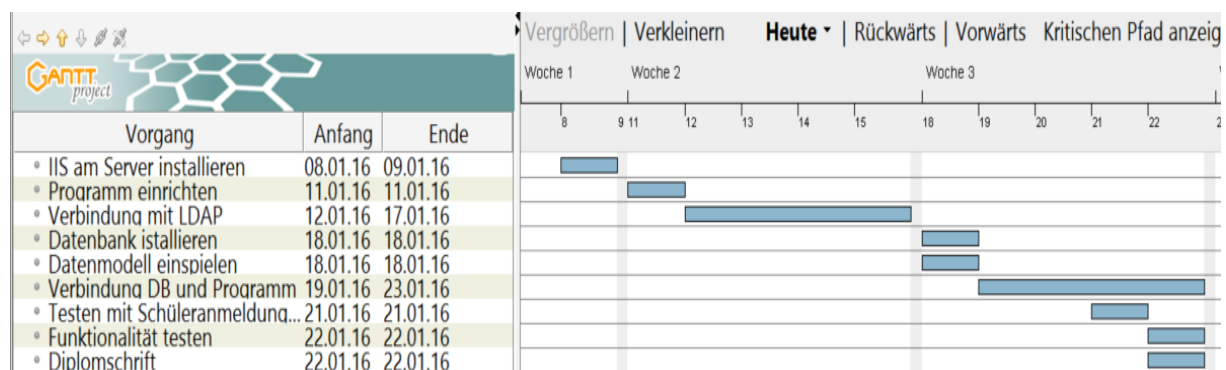


Abbildung 10 : GANTT-Plan *anotesa*



4.5 Pflichtenheft

Dieser Teil beschäftigt sich mit den wichtigsten Punkten eines Pflichtenhefts. Dazu gehören die Zielbestimmungen, Wunsch- und Abgrenzungskriterien sowie der Produkteinsatz und die Zielgruppen.

4.5.1 Zielbestimmungen

Das zu entwickelnde Programm stellt eine Möglichkeit dar, anonyme Tests zu verwalten. Dafür stellt das Programm ein ansprechendes Design, sowie Möglichkeiten, Noten, Schüler und Tests zu verwalten, zur Verfügung. (siehe Funktionalitäten auf Seite 15)

4.5.2 Grenzkriterien

4.5.2.1 Musskriterien

- Login-Funktion
- Verwaltung von Notendaten
- Darstellung muss einfach und übersichtlich sein
- Einblick der Schüler in
 - Ihre Noten
 - Ihre Tier-Zuordnungen
 - Benachrichtigungen

4.5.2.2 Abgrenzungskriterien

- Die Software steht ausschließlich in deutscher Sprache zur Verfügung
- Jegliche Schriftarten und Schriftgrößen sind fix vorgegeben
- Die Software wird ausschließlich unter dem Link <http://anotesa.html-perg.ac.at> verfügbar sein

4.5.3 Produkteinsatz

4.5.3.1 Anwendungsbereiche

Die Software wird zu Beginn in der Klasse 5AHIF im Deutschunterricht angewandt. Vielleicht wird das Programm in allen Klassen ebenfalls zur Anwendung kommen. Damit kann jeder Lehrer bei jedem Test die Noten selbstständig eintragen.

4.5.3.2 Zielgruppen

Jede Klasse hat Schüler, die ihre Notenverwaltung einsehen dürfen. Auch können Lehrer ihre anonymen Tests verwalten und Noten vergeben. Die berechtigten Benutzer müssen sich über Passwort und Benutzernamen identifizieren.

4.5.4 Produktkonfiguration

4.5.4.1 Software, Hardware

- **Software**
 - Browser
 - Microsoft SQL Server 2014
 - Microsoft Visual Studio
 - Microsoft SQL Management Studio
- **Hardware**
 - Windows Server 2012
 - Endgerät (PC, Smartphone, Tablet)

4.5.4.2 Schnittstellen

Tests

Tests werden von Lehrern oder Admins erstellt. Sie können bewertet, bearbeitet und gelöscht werden.

Noten

Die Noten werden per Hand eingegeben. Diese Daten werden auf der Datenbank abgespeichert.

Schülerdaten

Jeder Schüler muss sich mit seiner Matrikelnummer. und dem dazugehörigen Passwort anmelden. Auf der Datenbank wird dann ein Eintrag erstellt, der Verbindungen zu Benachrichtigungen und Tier-Zuordnungen erleichtern soll.

Lehrerdaten

Auch jeder Lehrer muss sich mit seinem Lehreruser und dem dazugehörigen Passwort anmelden. Auch dafür wird ein Eintrag auf der Datenbank erstellt, um Verbindungen zu erleichtern.

Benachrichtigungen

Nach einigen Aktionen, wie z.B. dem Erstellen oder Bewerten eines Tests, werden an die Schüler Benachrichtigungen gesendet. Diese sollen die Schüler über die Aktionen informieren.

Tier-Zuordnungen

Nach dem erfolgreichen Erstellen eines Tests, bekommt jeder Schüler ein Tier zugeordnet. Dadurch wird er beim jeweiligen Test identifiziert.

Datenbank

Auf der Datenbank werden alle Noten, die dazugehörigen Tests, Schülerdaten und auch Lehrerdaten gespeichert.



4.6 Lieferumfang

Hier wird erläutert, welche Dokumente und welche Software der Auftraggeber bzw. die Ansprechperson vom Diplomandenteam bekommen.

4.6.1 Software

Frau Prof. Mag. Rosmarie Lindinger-Pesendorfer erhält von uns, Anika Seibezeder und Matthäus Pfusterschmid, die Webseite *anotesa*, mit der sie anonyme Schularbeiten verwalten kann. Dazu erhält sie einen Zugang zu dem Administrator, um Tests jederzeit ändern, löschen oder bewerten zu können.

Auch Herr Prof. Mag. Rupert Obermüller erhält eine Kopie unserer Diplomarbeit auf einer von uns bereitgestellten CD.

Zusätzlich wird unser Projekt über den Link:

<http://anotesa.htl-perg.ac.at>

von überall, das heißt in der Schule und zu Hause, erreichbar sein und so für Lehrer die Benotung erleichtern. Schüler können dazu ihre Noten jederzeit abrufen.

4.6.2 Dokumente

Noch dazu erhält Herr Prof. Mag. Rupert Obermüller als Betreuungslehrer diese Diplomschrift.

5 Software – Verwendete Technologien

5.1 C#

C# ist eine von der Firma Microsoft für das *.NET-Framework* entwickelte und von der *European Computer Manufacturers Association* standardisierte Programmiersprache, die auf den Vorbildern C++ und *Java* aufbaut, aber auch etliche Weiterentwicklungen bietet.

Bei der Entwicklung des *.NET Frameworks* wurden die Klassenbibliotheken ursprünglich mit einem verwalteten *Code-Compiler-System* geschrieben, welches als *Simple Managed C (SMC)* bezeichnet wird. Im Januar 1999 gründete Anders Hejlsberg ein Team, mit dem Namen *Cool* ("C-like Object Oriented Language"), um eine neue Sprache zu entwickeln. *Microsoft* hat sich zuerst dazu entschieden den Namen *Cool* aufrecht zu erhalten, entschied sich jedoch aus Marketinggründen doch dafür, den Namen zu ändern. Mit der Zeit wurde das *.NET-Projekt* öffentlich gemacht und für Juli 2000 bei der *Professional Developers Conference* angekündigt, man hat die Sprache in C# umbenannt, und die Klassenbibliotheken und die *ASP.NET-Runtime* wurden zu C# portiert.

Hauptdesigner von C# und leitender Architekt bei *Microsoft* war Anders Hejlsberg, der zuvor mit dem Design von *Turbo Pascal*, *Delphi Embarcadero* (ehemals *CodeGear Delphi* und *Borland Delphi*) und *Visual J++* beteiligt war. In Interviews und technischen Zeitschriften, hat er erklärt, dass Mängel in den meisten wichtigen Programmiersprachen (beispielsweise C++, *Java*, *Delphi*, und *Smalltalk*) die Grundlagen der *Common Language Runtime (CLR)* stellten, die wiederum die Gestaltung der Sprache C# selbst beeinflussten.

Die Geschwungene-Klammer-Syntax von C# wird jedem sofort erkennbar sein, der mit C, C++ oder *Java* vertraut ist. Entwickler, die eine dieser Sprachen kennen, sind typischerweise in der Lage, mit C# zu beginnen, und innerhalb einer sehr produktiv mit C# zu arbeiten. Die C# Syntax vereinfacht viele der Komplexitäten von C++ und bietet leistungsstarke Funktionen wie *nullable-Wertetypen*, *enumerations*, *delegates*, *Lambda-Ausdrücke* und direkten Speicherzugriff, die in *Java* nicht umgesetzt werden können.

[1]



5.1.1 ASP .NET

Bei *ASP.NET* handelt es sich um eine schnelle und einfache Technologie zur Erstellung von dynamischen Webinhalten. Dynamische Webinhalte bestehen hierbei aus einer Kombination von Frontend-Technologien (z.B. *HTML5*, *CSS3* & *JavaScript*) und Server-Technologien (*ASP.NET Web Pages*, *Razor Engine*).

ASP.NET funktioniert mit dem *HTTP-Protokoll* und verwendet die *HTTP-Befehle* und *Richtlinien* um eine *Browser-Server bilaterale Kommunikation* und *Kooperation* aufzubauen.

ASP.NET ist ein Teil der *Microsoft.NET-Plattform*, und deren Anwendungen sind *Codes*, welche mit den erweiterbaren und wiederverwendbaren *Komponenten* oder *Objekte* im *.NET-Framework* geschrieben werden. Diese *Codes* können die gesamte *Hierarchie* von *Klassen* im *.NET-Framework* verwenden und können in den Sprachen *C#*, *Visual Basic*, *JScript* und *J#* geschrieben werden.

ASP.NET wird verwendet, um *interaktive, datengesteuerte Web-Anwendungen* über das *Internet* zu produzieren. Es besteht aus einer *Vielzahl* von *Steuerelementen*, wie beispielsweise *Textfelder*, *Buttons* und *Labels* zum *Erstellen*, *Konfigurieren* und zur *Manipulation* von *HTML-Seiten*. [2]

5.2 HTML

HTML ist keine *Programmiersprache*, sondern eine *Markupsprache*. Daher kommt auch der Name *HTML*, welcher *Hypertext Markup Language* bedeutet. *HTML* wird dazu verwendet, um *Webseiten* darzustellen und zu *designen*.

HTML war für mehrere Jahre in der *Versenkung*. Im *November 1990* hat man erstmals begonnen *erste Webseiten* zu erstellen, wobei zu diesem Zeitpunkt wenig bis kein *HTML-Standard* vorhanden war. Als Ergebnis bildete sich eine Gruppe namens *World Wide Web Consortium*, welche *Standards* für die *HTML-Codierung* festlegte. Diese gelten bis heute und zu diesen *Standards* zählen folgende Punkte:

- **Tag** – Wird verwendet um *Textstücke* zu markieren („mark-up“). Einmal markiert, wird der *Text* zu *HTML-Code*, der von einem *Web-Browser* interpretiert wird. *<tag>*
- **Element** – Ein komplettes *Tag*, mit einer *Öffnung* *<tag>* und *Schließung* *</tag>*.
- **Attribut** – Wird verwendet um den *Wert* des *HTML-Elements* zu bearbeiten. *Elemente* haben oft mehrere *Attribute*.

[3]

5.3 CSS

Um das Jahr 1994 beginnt das Web eine Plattform für das elektronische Publizieren mit immer mehr Interessenten zu werden. Ein entscheidender Teil einer Publishing-Plattform fehlt jedoch: Es gibt keine Möglichkeit, Dokumente zu formatieren. Zum Beispiel gibt es keine Möglichkeit, ein Zeitung-ähnliches Layout in einer Web-Seite zu beschreiben. Nachdem er mit personalisierten Zeitungspräsentationen am MIT Media Laboratory arbeitete, sah Håkon Wium Lie, ein norwegischer Informatiker, die Notwendigkeit einer Stylesheet-Sprache für das Web. [4]



Abbildung 11 : CSS

Bei CSS wird der Fokus auf die Formatierung und Darstellung von Informationen gerichtet. Dazu wird eine *W3C-Technologie* namens *Cascading Style Sheets (CSS)* verwendet, die es Autoren von Dokumenten ermöglicht die Präsentation von Elementen auf einer Webseite anzugeben, wie beispielsweise Schriftarten, Abständen oder Rändern. Dies erfolgt getrennt von der Struktur des Dokuments (*sectionheaders, body text, links, etc.*). Diese Trennung der Struktur von der Darstellung vereinfacht die Aufrechterhaltung und Änderung des Layouts eines Dokuments. Wenn man sich dazu entscheidet, dass alle *h1*-Elemente in beispielsweise *24pt Segoe UI* dargestellt werden sollen, hat man die Möglichkeit nur eine Stilregel zu aktualisieren und nicht jede *h1*-Instanz im HTML-Dokument zu ändern.

CSS muss nicht zwingend verwendet werden, jedoch sind die Vorteile der Verwendung bezüglich Zeitersparnis kaum überschaubar:

- Konsistenter Look-and-Feel auf der Website.
- Mit einem oder zwei *Stylesheets* können Farben, Schriftarten, Links und andere Elemente der Webseite gleich gehalten werden, ohne die gleichen Informationen zu jeder Seite der Website hardcoden zu müssen.
- CSS spart Zeit.
- Während es ein bisschen mehr Zeit in Anspruch nimmt, vor allem wenn ein *WYSIWYG-Editor (What you see is what you get-Editor)* verwendet wird, eine HTML Seite zu erstellen, geht es bei zusätzlichen Seiten umso schneller.
- Sie können das gesamte *Look-and-Feel* einer Website, durch eine kleine Änderung im Stylesheet, ändern.

[5]

5.4 JavaScript

JavaScript wird am häufigsten als client-seitige Skriptsprache zur Programmierung dynamischer Webseiten verwendet. Dies bedeutet, dass *JavaScript*-Code in eine *HTML*-Seite geschrieben wird. Wenn ein Benutzer eine *HTML*-Seite mit *JavaScript* anfordert, wird das Skript an den Browser gesendet, und es liegt im Ermessen des Browsers, etwas damit anzufangen.

Die Tatsache, dass das Skript in der *HTML*-Seite ist, bedeutet, dass die Skripte, von wem auch immer die Seite aufgerufen wird, gesehen und kopiert werden können. Dennoch kann diese Offenheit als ein großer Vorteil angesehen werden, da man diverse *JavaScript*-Codes welche man im *World Wide Web* findet, kopieren, und für die eigene Zwecke verwenden kann. So wird eine schnelle Verbreitung von *JavaScript* gewährt.



Abbildung 12 : JavaScript

JavaScript kann aber auch in anderen Kontexten als in einem Web-Browser verwendet werden. *Netscape* erstellt serverseitiges *JavaScript* als *CGI*-Sprache, die in etwa das Gleiche wie *Perl* oder *ASP* kann. Es gibt keinen Grund, warum *JavaScript* nicht verwendet werden kann um anspruchsvolle bzw. komplexe Programme zu schreiben. [6]

JavaScript wurde von *Netscape* in Zusammenarbeit mit *Sun Microsystems* entwickelt. Davor waren Browser einfache Programme, welche in der Lage waren, *Hypertext*-Dokumente anzuzeigen. *JavaScript* wurde später eingeführt, um der Erstellung von Webseiten mehr Möglichkeiten zu bieten und sie interaktiver zu machen. Die erste Version *JavaScript 1.0* wurde 1995 im *Netscape Navigator 2* veröffentlicht.

Zur Zeit der Veröffentlichung von *JavaScript 1.0*, dominierte der *Netscape-Navigator* den Browser-Markt. *Microsoft* kämpfte derzeit mit dem eigenen Browser, dem *Internet Explorer*, aufzuholen und folgte dem Beispiel von *Netscape* und führte eine eigene Skriptsprache namens *Visual Basic* ein. Zusammen mit dieser und einer Version von *JavaScript* mit dem Namen *JScript*, standardisierten *Netscape* und *Sun* die Sprache gemeinsam mit der *European Computer Manufacturers Association (ECMA)*. Das Ergebnis war *ECMAScript*; jedoch hielt sich dieser Name nicht lange und wir kennen diese Sprache heute als *JavaScript*. [7]

5.4.1 JQuery

jQuery ist ein JavaScript-Framework, dessen Ziel es ist, das Arbeiten mit *JavaScript* auf einer Website, einfacher zu gestalten. Man kann *jQuery* auch als Abstraktionsschicht beschreiben, da es die Funktionalität, welche in mehreren Zeilen geschrieben worden ist, nimmt, und sie in eine Funktion wickelt, welche mit einer einzigen Codezeile aufgerufen werden kann. Es ist wichtig anzumerken, dass *jQuery* *JavaScript* nicht ersetzen kann. Obwohl es einige syntaktische Verknüpfungen anbietet, ist der Code, der bei der Verwendung von *jQuery* geschrieben wird, nach wie vor *JavaScript*.



Abbildung 13 : jQuery

In diesem Sinne sollte man sich bewusst sein, dass man kein *JavaScript*-Experte sein muss um *jQuery* verwenden zu können. In der Tat versucht *jQuery* viele der komplizierten Dinge von *JavaScript*, wie *AJAX* Anrufe und *DOM*-Manipulation zu vereinfachen, so dass man diese Dinge vornehmen kann, ohne übermäßig viel über *JavaScript* zu wissen.

Es gibt eine Reihe von alternativen JavaScript-Frameworks, jedoch ist *jQuery* zurzeit die beliebteste, und scheint die am meisten erweiterbare zu sein, bewiesen durch die Tatsache, dass es *jQuery*-Plugins für nahezu jede Funktionalität gibt. [8]

5.4.2 AJAX

AJAX, die Abkürzung für *Asynchronous JavaScript and XML*, ermöglicht es Daten im Hintergrund zu laden und auf der Webseite angezeigt zu bekommen, ohne die Seite zu aktualisieren. Auf diese Weise können Webseiten erstellt werden mit der viel mehr Funktionen möglich sind. Beliebte Web-Anwendungen wie *Google Mail*, *Outlook Web Access*, und *Google Maps* verwenden *AJAX*, um ein schnell reagierendes, Desktop-ähnliches Erlebnis zu gewähren.

AJAX kann unter Umständen etwas problematisch sein, da verschiedene Browser unterschiedliche Implementierungen haben, welche *AJAX* unterstützen. Dafür kommt zum Beispiel *jQuery* zum Einsatz, wodurch nur eine einzige Zeile Code geschrieben werden muss. [9]



5.5 Bootstrap

Bootstrap ist ein Open Source front end-Framework für die schnellere und einfachere Web-Entwicklung. Es enthält *HTML* und *CSS*-basierte Design-Vorlagen für Komponenten der Benutzeroberfläche wie Formulare, Buttons, Tabellen, *Navigationen*, Auswahlmenüs, *Alerts*, *Tabs*, und viele andere sowie optionale *JavaScript*-Erweiterungen. *Bootstrap* bietet zusätzlich die Möglichkeit ein ansprechendes Layout mit viel weniger Aufwand zu erstellen.

Der größte Vorteil von *Bootstrap* ist, dass es mit einem Set von Werkzeugen für die Erstellung flexibler und reaktionsfähiger *Web-Layouts* sowie *Common-Interface-Komponenten* kommt. Darüber hinaus können unter der Verwendung der *Bootstrap-Data-APIs* erweiterte Schnittstellenkomponenten wie *Scrollspy* und *Typeahead* erstellt werden, ohne eine einzige Zeile *JavaScript* zu schreiben.



Abbildung 14 : Bootstrap

Weitere Vorteile bei der Verwendung von *Bootstrap* sind:

- Zeit – Man kann viel Zeit und Anstrengungen mit den von *Bootstrap* vorgegebenen Design-Vorlagen und Klassen sparen und sich auf andere Entwicklungsarbeiten konzentrieren.
- Design - Alle *Bootstrap* Komponenten teilen die gleichen Design-Vorlagen und Stile durch eine zentrale Bibliothek, so dass die Designs und Layouts der Webseite bei der Entwicklung konsistent bleiben.
- Einfach zu bedienen - *Bootstrap* ist sehr einfach zu bedienen. Wer mit den Grundkenntnissen in *HTML* und *CSS* vertraut ist, kann die Entwicklung mit *Bootstrap* starten.
- Kompatibel mit Browsern - *Bootstrap* ist kompatibel mit allen modernen Browsern wie *Mozilla Firefox*, *Google Chrome*, *Safari*, *Internet Explorer* und *Opera*.
- Open Source – *Bootstrap* ist als kostenloser Download erhältlich.

[10]

5.5.1 Morris.js

Morris.js ist eine *Open-Source-API* und wird zur Darstellung von Diagrammen verwendet. Die *API* bietet zahlreiche Methoden, um zum Beispiel Balken-, Linien- oder Donut Diagramme anzeigen zu lassen.

Erstellt werden kann beispielsweise ein Balkendiagramm mit `Morris.Bar(options)`, wobei `options` ein Objekt ist, welches mit den Konfigurationsoptionen gefüllt ist. Die folgenden Punkte sind die erforderlichen Optionen: [11]

Element	Die ID (oder eine Referenz auf) das Element, in welches das Diagramm einzufügen ist. Dieses Element muss eine Breite und Höhe definiert haben.
Daten	Dies ist ein Array von Objekten, mit <code>x</code> und <code>y</code> -Attribute. Die Reihenfolge, in der die Daten übermittelt werden, ist die Reihenfolge, in der die Balken angezeigt werden.
xkey	Eine Zeichenfolge, die den Namen des Attributs enthält, welches die <code>x</code> -Achse beschreibt.
ykeys	Eine Liste von Zeichenketten, die die Namen von Attributen enthält, die <code>Y</code> -Werte beschreiben.
labels	Eine Liste von Strings, für die Labels, die auf die Datenreihe aufgetragen werden (entsprechen der Option <code>ykeys</code>).

Tabelle 6 : Optionen Morris.js

5.5.2 Datetimepicker.js

Die *API Datetimepicker.js* wird verwendet zur Realisierung des Auswählens von Datumswerten. Es erleichtert den Benutzer die Eingabe eines Datums oder einer gewissen Zeit mit der Maus. Die Implementierung eines einfachen *Datetimepickers* sieht folgendermaßen aus:

HTML

```
<input id="datetimepicker" type="text" >
```

JavaScript

```
jQuery('#datetimepicker').datetimepicker();
```

[12]



5.6 Windows Server 2012

Windows Server 2012, oder auch *Windows Server 8*, ist die sechste Version von Windows Server. Es ist die Server-Version von *Windows 8* und ist der Nachfolger von *Windows Server 2008 R2*. Zwei Pre-Release-Versionen, eine Entwickler-Vorschau und eine Beta-Version, wurden während der Entwicklung veröffentlicht. Für Kunden wurde der Server ab dem 4. September 2012 veröffentlicht. [13]

Im Gegensatz zu seinem Vorgänger unterstützt der *Windows Server 2012* keine Itanium-basierten Computer. Verschiedene Merkmale wurden hinzugefügt oder verbessert im Vergleich zum *Windows Server 2008 R2* (ein Schwerpunkt wurde speziell auf Cloud Computing gesetzt), sowie eine aktualisierte Version von *Hyper-V*, eine *IP-Address-Management*-Rolle, eine neue Version des *Windows Task-Manager* und ein neues Dateisystem. [14]

Erforderliche Minimalanforderungen an das System:

- Prozessor: 1,4GHz, x64
- 512 MB Arbeitsspeicher
- 32 GB freier Speicher

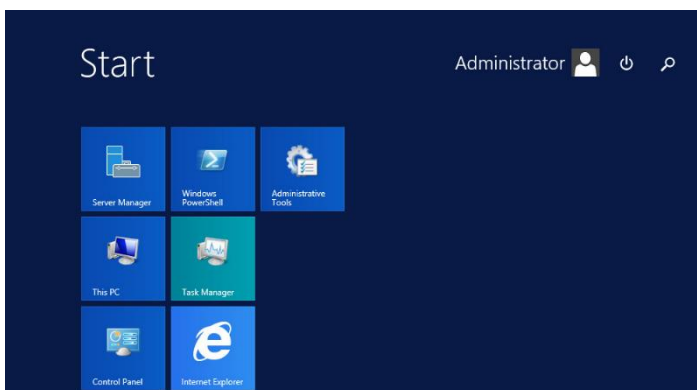


Abbildung 15: Screenshot des Startfesters eines Windows Servers

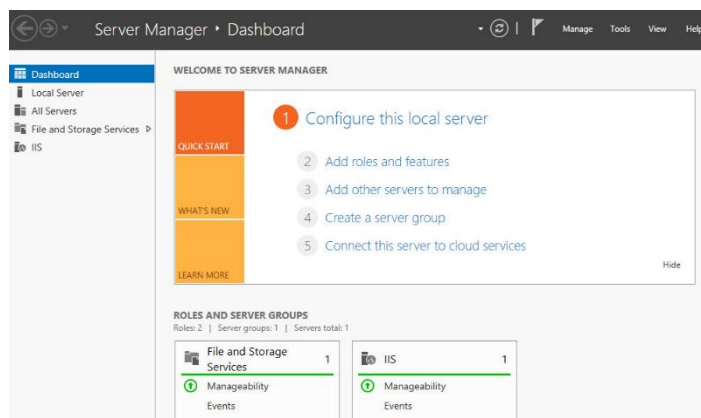


Abbildung 16: Screenshot des Server Manager

5.7 Google Drive

Google Drive ist ein Filehosting-Dienst des Unternehmens Google Inc.

In Form eines Cloudspeichers diente uns Google Drive als externes Speichermedium, auf welches sämtliche Dateien gespeichert wurden, die die Bearbeitung von beiden Diplomanden erforderten. Jedoch wurden nur Dokumente verwaltet, die nicht zum Programm zugehörig sind, wie z.B. Word Dokumente oder das Logo. [15]



Abbildung 17: Google Drive

5.8 Visual Studio

Als Entwicklertool für anotesa diente Visual Studio, eine Entwicklungsumgebung des Unternehmens Microsoft. Primär fiel die Entscheidung auf Visual Studio, da es nahezu alle verwendeten Programmiersprachen unterstützt: C# oder ASP.NET sowie HTML und JavaScript speziell für Web-Anwendungen. [16]



Abbildung 18: Visual Studio



6 Programmierung

6.1 Darstellung

6.1.1 Login

The image shows a login window titled 'Anmelden'. It contains a text input field with the value 'IF110042', a password input field with masked characters, and a green 'Log in' button.

Abbildung 19 : Login-Fenster

Wenn man unsere Diplomarbeit unter <http://www.anotesa.htl-perg.ac.at> aufruft, kommt man zuerst auf diese Seite. Hier gibt man die Matrikelnummer, falls man sich als Schüler anmeldet, oder den Lehrer-Usernamen, falls man sich als Lehrer anmeldet, und das dazugehörige Passwort, ein.

6.1.2 Navigation-Bar

Egal ob man sich als Lehrer, Schüler oder auch als Administrator (hat eigene Logindaten) anmeldet, die Navigation-Bar erscheint bei jeder weiteren Unterseite von anotesa am linken und oberen Rand.

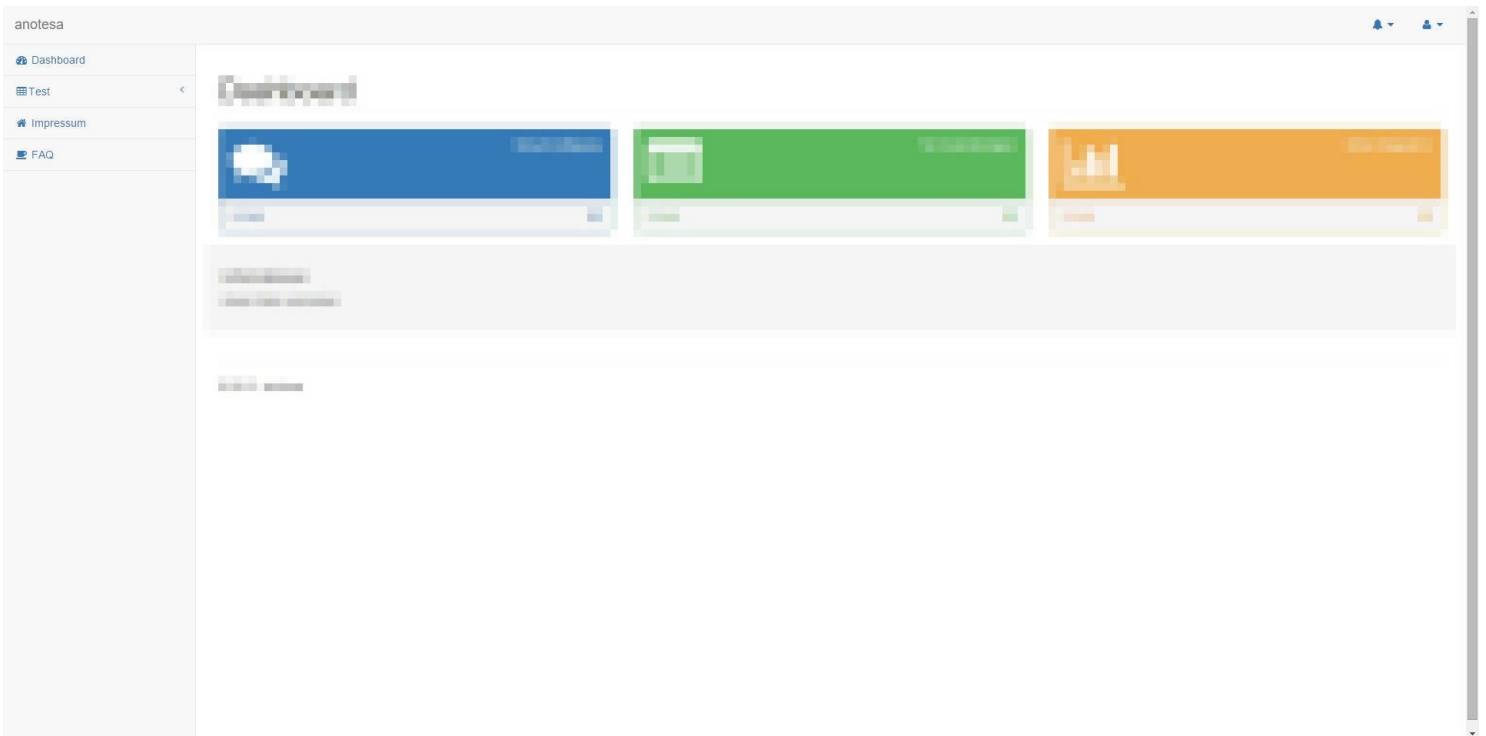


Abbildung 20 : Navigation-Bar

6.1.2.1 Benachrichtigungen

Sind Benachrichtigungen vorhanden, so erscheint das „Glocken-Symbol“ im rechten oberen Rand rot.

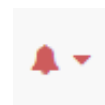


Abbildung 21 :
Glocken-Symbol

Klickt man nun auf dieses Symbol, erscheinen die Benachrichtigungen.

Am linken Rand jeder Benachrichtigung sieht man ein „Sprechblasen-Symbol“ und von wem man diese Benachrichtigung erhalten hat.

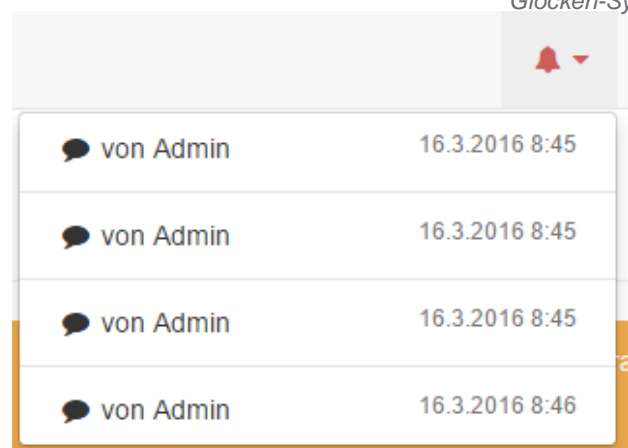


Abbildung 22 :
Benachrichtigungen



Man erhält Benachrichtigungen von dem Jemanden, der eine bestimmte Aktion ausgeführt hat (z.B. ein Test wird erstellt). Am rechten Rand jeder Benachrichtigung sieht man das Datum und die Uhrzeit der Erstellung der Benachrichtigung. Mit einem Klick auf eine Benachrichtigung kommt man zur Detailseite dieses Testes (siehe : Testdetails Lehrer - Bewerteter Test auf Seite 48).

6.1.2.2 User

Falls man eingeloggt ist, erscheint im rechten oberen Rand das User-Symbol in blau.

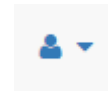


Abbildung 23 : User-Symbol

Klickt man nun auf dieses Symbol, erscheint ein Dropdownmenü mit den Punkten „Hilfe“ und „Logout“. Bei einem Klick auf „Hilfe“ erscheint ein weiteres kleineres Fenster, das ein Anleitungsvideo für anotesa wiedergibt. Mit einem Klick auf „Logout“ meldet sich man von anotesa ab und man wird zum Login-Fenster zurückgeleitet (siehe Login-Fenster auf Seite 35).

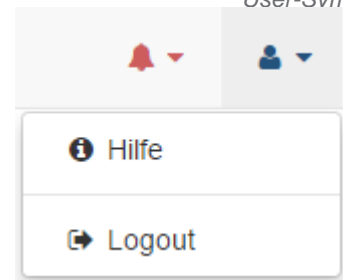


Abbildung 24 : User-Menü

6.1.2.3 Tasks (nur für Lehrer)

Wenn man sich als Lehrer einloggt, erscheint zusätzlich zu den oben genannten Menüpunkten noch der Punkt „Tasks“. Dieser erscheint oben rechts mit einem „Task-Symbol“.

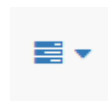


Abbildung 25 : Tasks-Symbol

Bei einem Klick auf das Symbol öffnet sich ein Menü mit den verschiedenen Tasks, die für die Lehrer noch zu erfüllen sind. Ein Task ist in unseren Fall ein offener Test, der noch bewertet werden muss.

Bei jedem offenen Test sieht man einen Zeitfortschritt, der entweder grün (wenn noch keine 60% der Zeit verschritten wurde), orange (wenn schon mehr als 60%, aber noch keine 80% der Zeit verschritten wurde) oder rot (wenn schon mehr als 80% der Zeit verschritten wurde) ist. Im

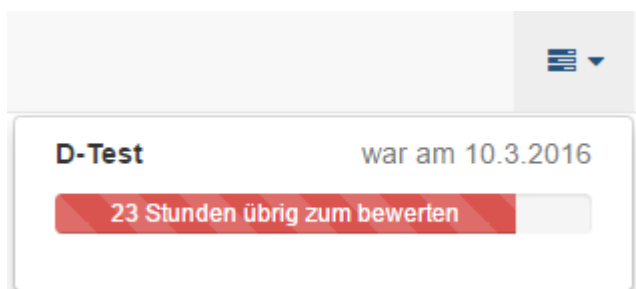


Abbildung 26 : Tasks

Zeitfortschritt kann man erkennen, wie viel Zeit noch zum Verbessern bleibt. Oben links erscheint bei jedem offenen Test der Name des Faches und oben rechts das Datum des Testes. Als Bemessungsgrundlage der Zeit zum Verbessern haben wir uns dafür entschieden, dass ein Test bis eine Woche nach dem Testtermin verbessert werden muss.

6.1.2.4 Keine Daten vorhanden

Sind keine Daten vorhanden (keine Tasks oder Benachrichtigungen), so erscheint beim Klick auf das jeweilige Symbol ein eigener Kontext.

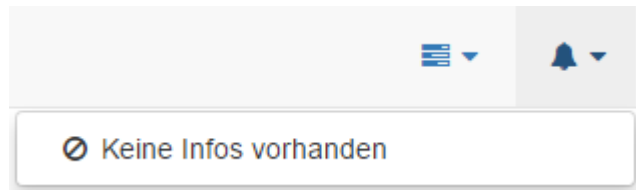


Abbildung 27 : Keine Daten vorhanden

Dieser beinhaltet ein „No-Symbol“ und den Text „Keine Infos vorhanden“.

6.1.3 Dashboard

Beim Dashboard handelt es sich um eine Startseite, die die Schüler, die Lehrer und den Admin über Neuigkeiten informieren soll. Dazu stehen ihnen eine Auswahl zwischen Benachrichtigungen und anderen Kategorien bereit. Beim jeweiligen Klick auf die Buttons, erscheint dynamisch der Inhalt dieser Kategorie.

Je nachdem, ob man sich nun als Lehrer, Schüler oder Administrator angemeldet hat, kommt eine unterschiedliche Ansicht des Dashboards:

6.1.3.1 Dashboard Schüler

Für Schüler erscheinen 3 Kategorien: Benachrichtigungen, Tier-Zuordnungen und Noten-Diagramm. Als Standard werden die Benachrichtigungen angezeigt. Die Darstellung erscheint wie folgt.

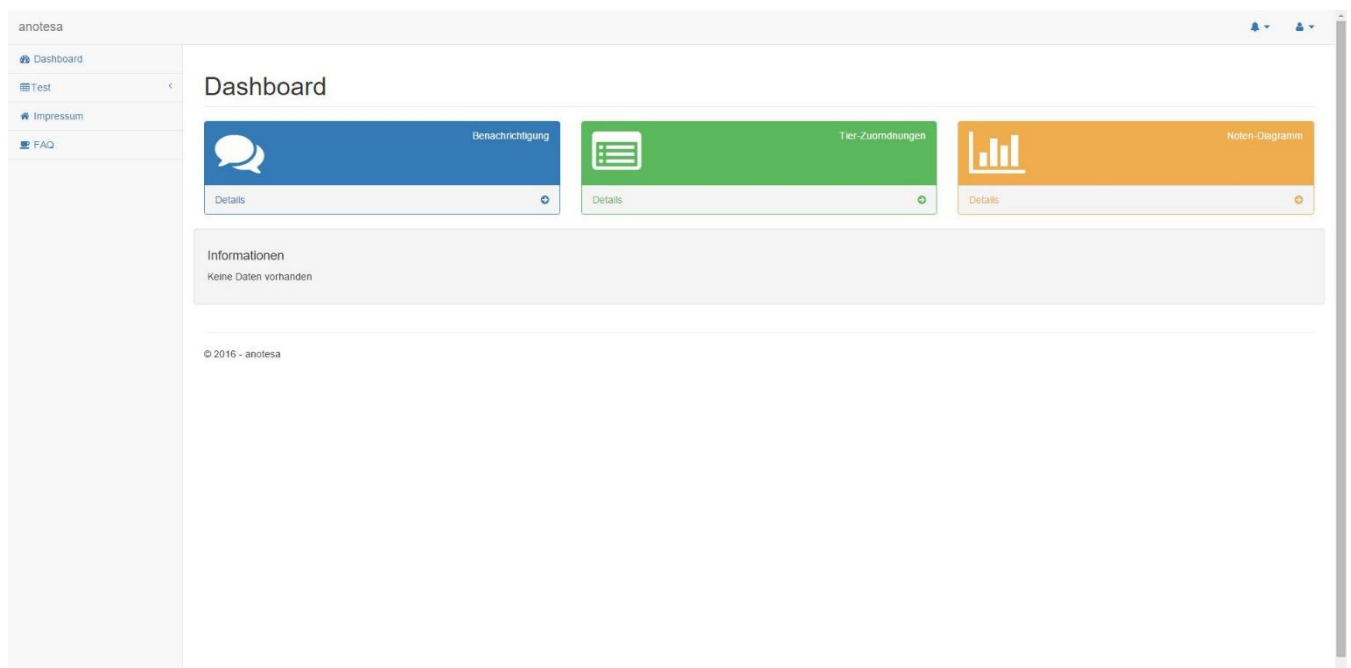


Abbildung 28 : Dashboard Schüler



Klickt man nun eine der Kategorien, geschieht folgendes:

Benachrichtigungen

Dashboard

© 2016 - anotesa

Abbildung 29 : Dashboard Schüler - Benachrichtigungen

Es erscheinen alle Benachrichtigungen eines Schülers. Diese können „Test wurde erstellt“, „Test wurde bearbeitet“ oder „Test wurde bewertet“ sein. Zusätzlich wird die Zeit der Erstellung der Benachrichtigung und der Absender mitgegeben.

Tier-Zuordnungen

Dashboard

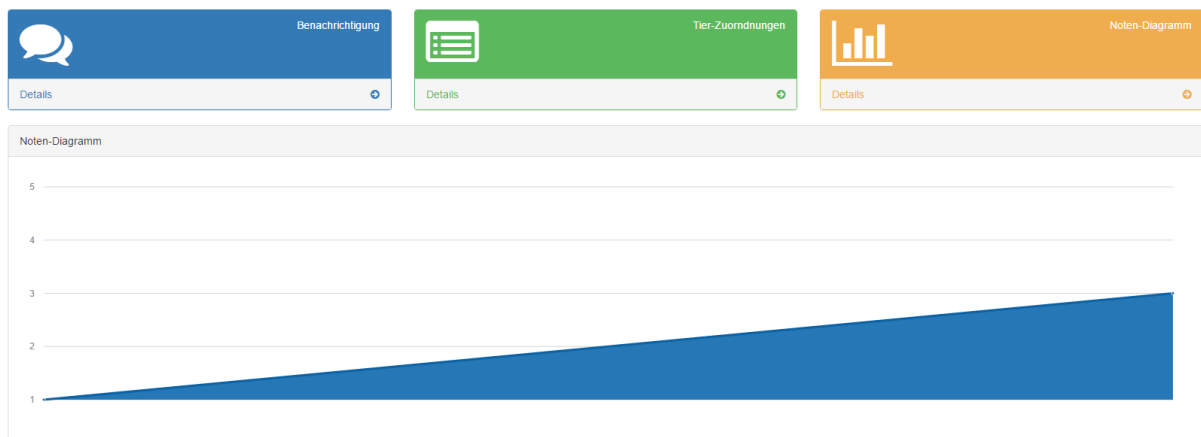
© 2016 - anotesa

Abbildung 30 : Dashboard Schüler - Tier-Zuordnungen

Es erscheinen alle Tier-Zuordnungen eines Schülers. Tier-Zuordnungen werden für jeden Test und jeden Schüler erstellt. Dieses Tier identifiziert den Schüler beim jeweiligen Test. Der Schüler muss sich das Tier gut merken und bei der Schularbeit statt seinem Namen den Tiernamen angeben.

Noten-Diagramm

Dashboard



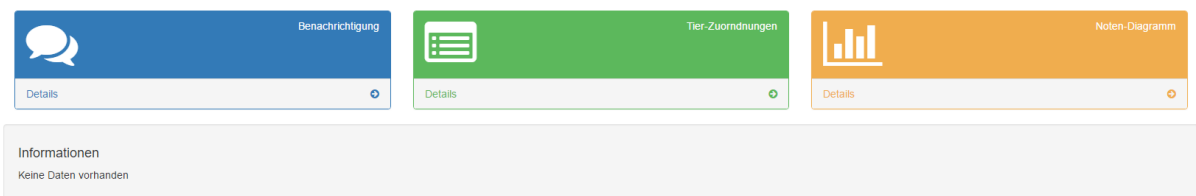
© 2016 - anotesa

Abbildung 31 : Dashboard Schüler - Notendiagramm

Es erscheinen alle Noten der vorherigen Tests oder Schularbeiten in einem Flächendiagramm. Die X-Achse beschreibt dabei den Zeitverlauf, die Y-Achse den Notengrad. Der Verlauf der Kurve zeigt den Verlauf der Noten an. Wenn man mit der Maus über das Diagramm fährt, erkennt man, dass es verschiedene Punkte gibt, die aus einem Datum und einer Note bestehen.

Keine Daten vorhanden

Dashboard



© 2016 - anotesa

Abbildung 32 : Dashboard Schüler - Keine Daten vorhanden

Sind keine Daten vorhanden (es wurde noch kein Test erstellt, es gibt keine neuen Benachrichtigungen oder es gibt keine Tests, die noch nicht bewertet wurden), so erscheint eine Information, die Auskunft darüber gibt, dass es eben keine Daten gibt. Dies ist sehr wichtig, da man so als Benutzer sofort weiß, dass es sich nicht um einen Fehler handelt.



6.1.3.2 Dashboard Lehrer

Für Lehrer erscheinen 2 Kategorien: Benachrichtigungen und Schüler-Noten-Diagramm. Als Standard werden die Benachrichtigungen angezeigt. Die Darstellung erscheint wie folgt.

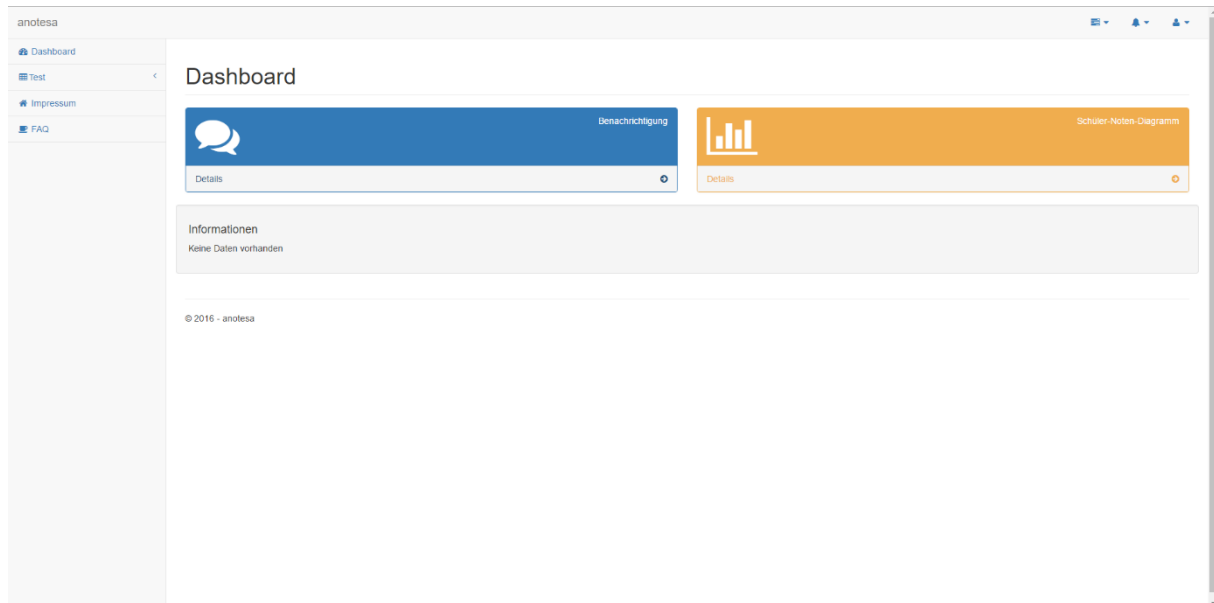


Abbildung 33 : Dashboard Lehrer

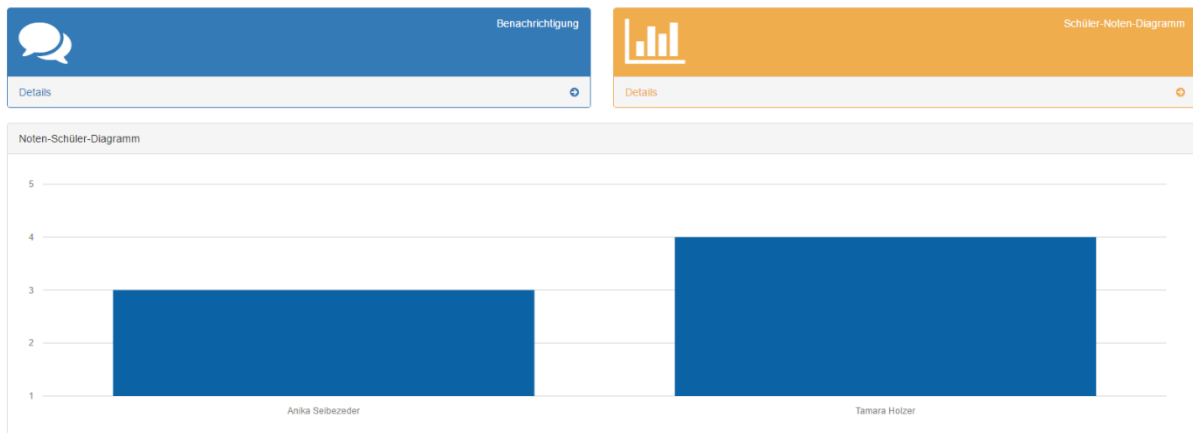
Klickt man nun eine der Kategorien, geschieht folgendes:

Benachrichtigungen

Die Benachrichtigungen erscheinen gleich wie bei den Schülern (siehe : Dashboard Schüler - Benachrichtigungen auf Seite 39). Dennoch erhält der Lehrer unterschiedliche Benachrichtigungen: Er bekommt eine Benachrichtigung, falls er nur mehr fünf, drei oder weniger als einen Tag Zeit hat zum Bewerten, wobei bis nach 1 Woche nach dem Test bewertet werden muss.

Schüler-Noten-Diagramm

Dashboard



© 2016 - anotesa

Abbildung 34 : Dashboard Lehrer - Schüler-Noten-Diagramm

Es erscheinen alle Noten des letzten Tests oder Schularbeit im Balkenformat. Die X-Achse beschreibt dabei die Schüler und die Y-Achse den Notengrad. So erkennt man als Lehrer gleich, wie der letzte Test oder die letzte Schularbeit ausgefallen ist. Fährt man mit der Maus auf einen Balken, erscheint ein kleines Infofenster. Dieses zeigt die Note und den Namen des Schülers noch einmal genau an.

Keine Daten vorhanden

(siehe : Dashboard Schüler - Keine Daten vorhanden auf Seite 40)



6.1.3.3 Dashboard Administrator

Wenn man sich nun als Administrator anmeldet, erscheinen 2 Kategorien: Benachrichtigungen und Schüler-Noten-Diagramm. Als Standard werden die Benachrichtigungen angezeigt. Die Darstellung erscheint wie folgt.

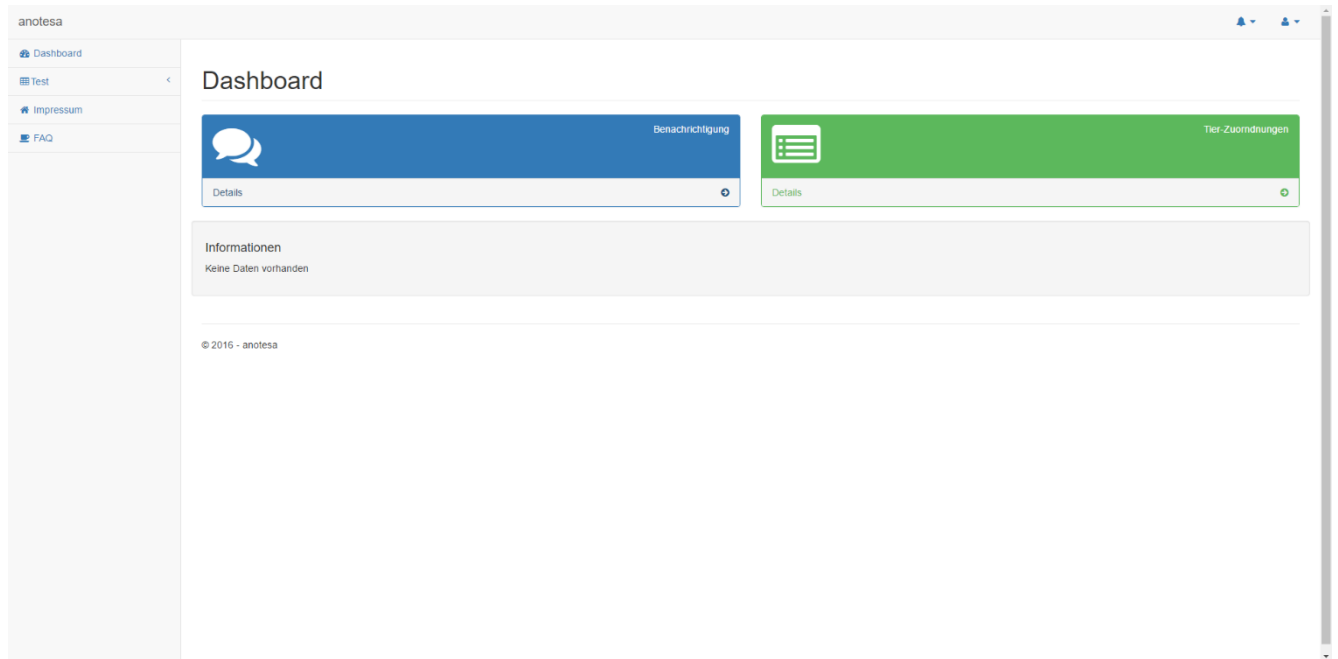


Abbildung 35 : Dashboard Administrator

Klickt man nun eine der Kategorien geschieht folgendes:

Benachrichtigungen

Grundsätzlich erscheinen die Benachrichtigungen gleich wie bei den Schülern (siehe : Dashboard Schüler - Benachrichtigungen auf Seite 39). Dennoch gibt es unterschiedliche Benachrichtigungstypen. Der Administrator erhält Benachrichtigungen, wenn ein Lehrer einen Test hinzufügt oder wenn ein Lehrer einen Test bearbeitet und bewertet.

Tier-Zuordnungen

Dashboard

Fach	Datum	Schulename	Tiername
REL	2.3.2016 11:27	Anika Seibezeder	Gorilla
GGP	17.3.2016 11:27	Anika Seibezeder	Grizzlybär
D	15.3.2016 11:45	Anika Seibezeder	Raupe

Informationen
Sie können jederzeit auf eine Zeile doppelklicken, um eine detaillierte Ansicht von dem jeweiligen Test zu bekommen

Abbildung 36 : Dashboard Administrator - Tier-Zuordnungen

Der Unterschied zu der Kategorie „Tier-Zuordnungen“ des Schülerdashboards (siehe : Dashboard Schüler - Tier-Zuordnungen auf Seite 39) liegt darin, dass beim Administrator alle Tier-Zuordnungen der Schüler dargestellt werden. Das heißt, er erhält von jedem Schüler die Tierzuordnung zum jeweiligen Test. Dadurch kann er den Schülern helfen, falls Jemand seine Tier-Zuordnung vergessen sollte.

Keine Daten vorhanden

(siehe : Dashboard Schüler - Keine Daten vorhanden auf Seite 40)

6.1.4 Tests ansehen

Egal ob man sich als Administrator, Schüler oder Lehrer anmeldet, der Weg um sich Tests anzusehen bleibt gleich: Man öffnet links bei der Navigation-Bar den Punkt „Test“ und wählt als Unterpunkt „Tests ansehen“ aus.

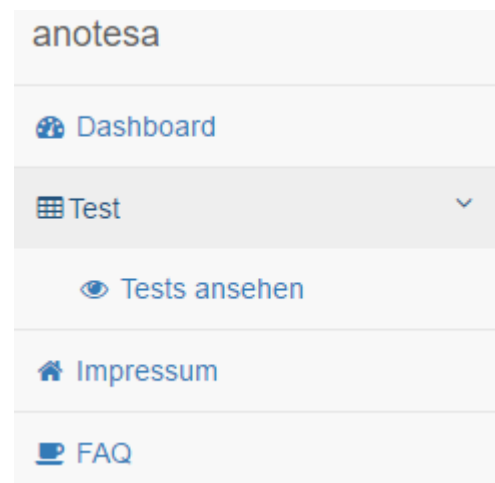
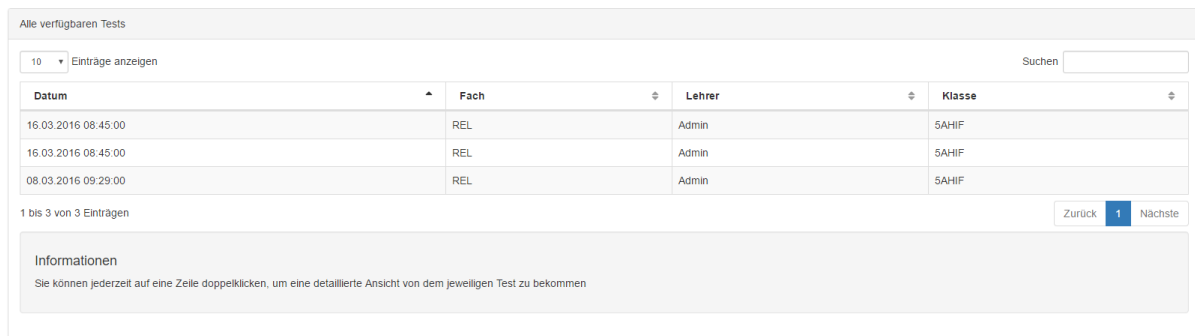


Abbildung 37 : Tests ansehen

Klickt man nun auf „Tests ansehen“ ergibt sich folgendes:



Tests anzeigen



Datum	Fach	Lehrer	Klasse
16.03.2016 08:45:00	REL	Admin	5AHIF
16.03.2016 08:45:00	REL	Admin	5AHIF
08.03.2016 09:29:00	REL	Admin	5AHIF

© 2016 - anotesa

Abbildung 38 : Tests anzeigen

Es erscheint eine Tabelle, die die verschiedenen Tests mit ihren Eigenschaften anzeigt. Die Spalten sind das Datum des Testes, das Fach des Testes, der Lehrer, der den Test erstellt hat und die Klasse. Ergeben sich nun sehr viele Einträge, kann man diese mit Hilfe des Suchfeldes, oder mit einem Klick auf eine Spalte filtern.

Oben links kann man dazu einstellen, wie viele Einträge pro Seite angezeigt werden sollen. Hat man mehr als 1 Seite, kann man mit den Buttons „Zurück“ und „Nächste“ die Seiten wechseln.

Zusätzlich gibt es noch ein Informationsfeld, das darüber informiert, dass man jederzeit auf eine Zeile klicken kann.

Unterschied bei Schülern

Es gibt bei den Schülern nur minimale Unterschiede bei der „Test ansehen“-Seite. Diese sind, dass bei jedem Test, die Note gleich dabei steht und falls ein Test noch nicht bewertet worden ist, wird das auch speziell gekennzeichnet.

6.1.5 Test-Details

Klickt man nun auf eine Zeile, so erscheint eine Detail-Seite des ausgewählten Tests. Diese ist bei Schülern, Lehrern und Administratoren unterschiedlich aufgebaut.

6.1.5.1 Lehrer

Falls ein ausgewählter Test noch nicht bewertet worden ist erscheint folgendes Bild:

Test

Details	
	D-5AHIF
Klasse	5AHIF
Fachbezeichnung	D
Vorname	Admin
Nachname	Admin
Datum	10.03.2016 10:11:00

[Zurück zur Liste](#)

© 2016 - anotesa

Abbildung 39 : Testdetails - Lehrer

Man erhält mit dieser Seite Informationen über den Test, wie z.B. den Klassennamen, das Datum des Testes, den Vor- und Nachnamen des Lehrers und die Fachbezeichnung. Im unteren Bereich sieht man 3 Buttons „Bewerten“, „Bearbeiten“ und „Löschen“ -

Falls man nun auf einen der Buttons klickt geschieht folgendes:

Bewerten

Test bewerten

Test-ID	Note	Feedback
Wal	Sehr Gut	<input type="text"/>
Clownfisch	Sehr Gut	<input type="text"/>

Abbildung 40 : Test bewerten

Es öffnet sich eine neue Seite, in der man für jedes Tier eine Note auswählen kann. Zusätzlich kann man ein Feedback eingeben, das den Schüler darüber informieren soll, ob er die Arbeit gut oder weniger gut gemacht hat.



Bearbeiten

Test bearbeiten

Gegenstand:	REL
Datum:	16.03.2016 10:25
Klasse:	1AFID
<input type="checkbox"/>	

Abbildung 41 : Test bearbeiten

Auf dieser Seite kann man einen Test bearbeiten, falls unabsichtlich falsche Daten beim Erstellen eingetragen worden sind.

Löschen

Löschen bestätigen

Sie sind dabei diesen Test zu löschen
Wollen Sie wirklich fortfahren?

Abbildung 42 : Test löschen

Beim Klick auf den „Löschen“-Button erscheint ein kleines Fenster mit der Frage „Sie sind dabei diesen Test zu löschen. Wollen Sie wirklich fortfahren?“. Falls man in dem kleineren Fenster nun auf „Löschen“ klickt, wird der Test gelöscht.

Bewerteter Test

Falls nun ein Test erfolgreich bewertet worden ist ergibt sich statt den Buttons folgendes Bild:

Noten		
	Note	Feedback
Tamara Holzer	1	Sehr gut gemacht!
Anika Seibezeder	3	Mittelmäßig

[Zurück zur Liste](#)

© 2016 - anotesa

Abbildung 43 : Testdetails Lehrer - Bewerteter Test

Man sieht nun die Namen der Schüler, die zu dieser Klasse gehören, ihre Note und ein Feedback zu ihrer Arbeit.

6.1.5.2 Administrator

Grundsätzlich ist die Seite Test-Details für den Administrator ident mit dem der Lehrer. Nur zeigt sie auch wenn der Test schon bewertet wurde Funktionen wie „Löschen“ und „Bearbeiten“ an.

6.1.5.3 Schüler

Nicht bewerteter Test

Details	
	D-5AHIF
Klasse	5AHIF
Fachbezeichnung	D
Vorname	Admin
Nachname	Admin
Datum	17.03.2016 13:30:00
Tierzuordnung	Chamaleon

[Zurück zur Liste](#)

© 2016 - anotesa

Abbildung 44 : Testdetails Schüler - Nicht bewerteter Test

Klickt man auf einem nicht bewerteten Test, so erscheint ein neues Fenster, welches Details zu diesem Test preisgibt.



Bewerteter Test

Details	
	D-5AHIF
Klasse	5AHIF
Fachbezeichnung	D
Vorname	Admin
Nachname	Admin
Datum	17.03.2016 13:30:00
Tierzuordnung	Chamäleon
Note	3
Feedback	Mittelmäßig

[Zurück zur Liste](#)

Abbildung 45 : Testdetails Schüler - Bewerteter Test

Ist ein Test nun bewertet worden und man öffnet die Detail-Seite dieses Tests, sieht man, dass zwei zusätzliche Zeilen hinzugefügt wurden: Note und Feedback.

6.1.6 Tests hinzufügen

Als Lehrer sowie auch als Administrator hat man die Möglichkeit einen Test zu erstellen.

Dazu klickt man am linken Rand auf „Test“ und auf die Unterkategorie „Test hinzufügen“.

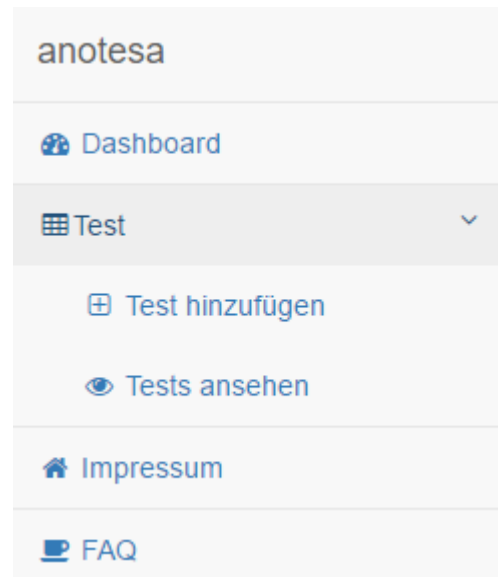


Abbildung 47 : Test hinzufügen - Navigationbar

Nun ergibt sich folgendes:

Test hinzufügen

Gegenstand:	REL
Datum:	16.03.2016 11:07
Klasse:	1AFID
<input type="button" value="✓"/>	

Abbildung 46 : Test hinzufügen

Beim Erstellen wählt man aus einer Liste von vorgegeben Schulfächern eines aus. Noch dazu fügt man noch ein Datum und eine Klasse hinzu. Die Auswahl des Datums wird mit Hilfe eines „Datetimepickers“ übersichtlicher.

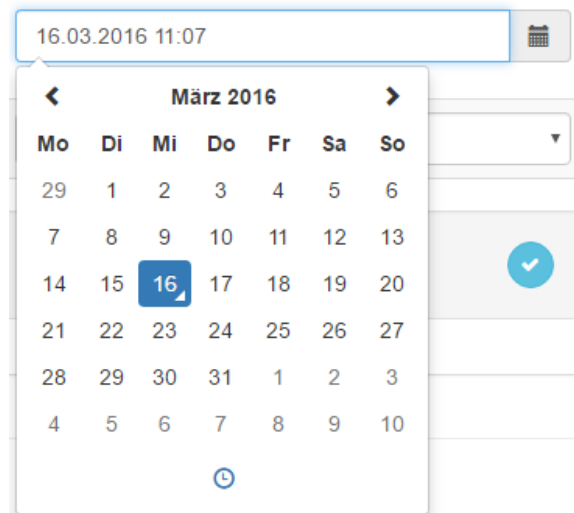


Abbildung 48 : Datetimepicker

6.2 Installation

6.2.1 Server

Für die Diplomarbeit *anotesa* wird von der HTL Perg ein *Windows Server 2012 R2* zur Verfügung gestellt, welcher zuerst installiert werden musste. Nach der Installation findet man sofort den Server Manager.

Da noch kein *IIS* vorinstalliert ist, wählt man zuerst *Add Roles and Features*, woraufhin sich ein Fenster öffnet.

6.2.1.1 Add Roles and Features Wizard

Die Installation von *IIS* beginnt beim Auswählen des Installationstyps. Hier wählt man *Role-based or feature-based* installation aus. Anschließend kommt man zur *Server Selection*, wo man den gewünschten Server selektiert, auf dem *IIS* installiert werden soll.

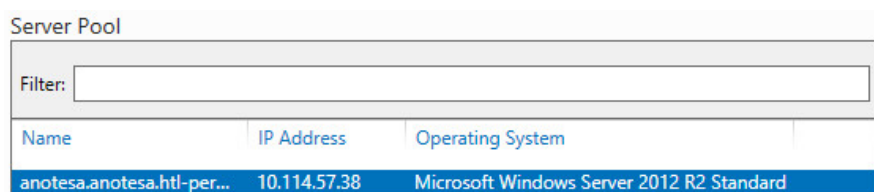


Abbildung 49 : Server Pool



Folglich wird man weitergeleitet zu den Serverrollen. Man fügt die gewünschten Rollen hinzu, welche benötigt werden. Im Fall von anotesa ist das lediglich der Web Server (IIS)

Letztlich fügt man noch die Features hinzu, welche das Programm, welches man später auf dem Web Server laufen lassen will, benötigt. Für anotesa werden folgende Features benötigt:

- *.NET Framework 3.5 Feature*
 - *.NET Framework 3.5*
- *NET Framework 4.5 Feature*
 - *NET Framework 4.5*
 - *ASP.NET 4.5*
 - *WCF Services*
 - *TCP Port Sharing*

6.2.1.2 IIS-Manager

Sobald der *IIS-Manager* startet, erhält man folgendes Bild:

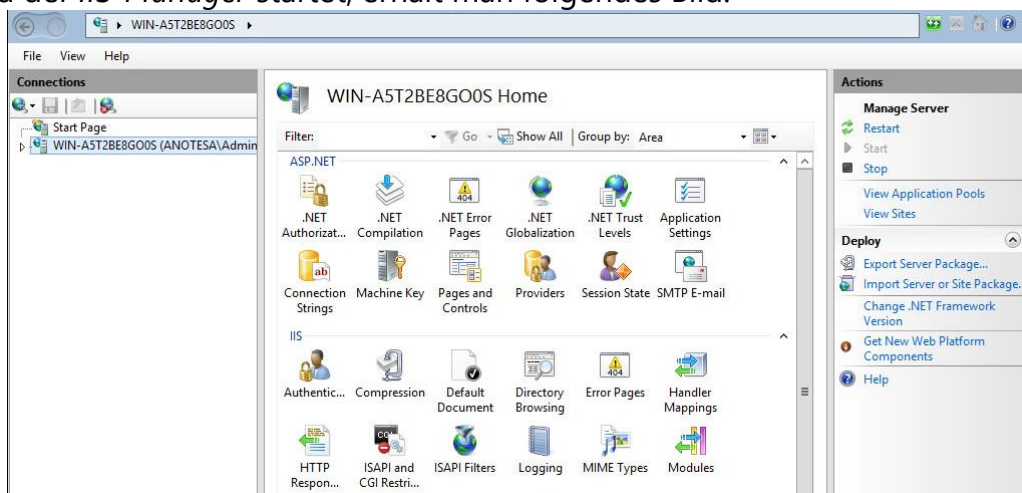


Abbildung 50 : IIS-Manager

Um das Programm später auch richtig laufen lassen zu können, muss das Programm am *Windows Server* unter dem Pfad *C:\inetpub\wwwroot* zu finden sein.

Für die Eintragung des Programmes, erstellt man am Server eine neue *Application* und bearbeitet sie folgendermaßen:

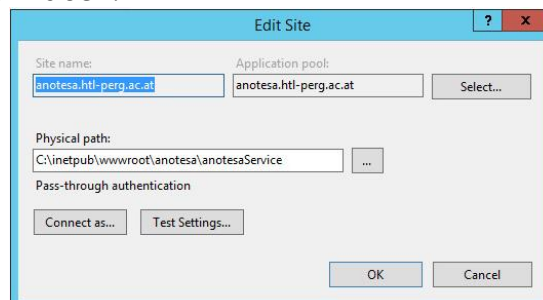


Abbildung 51 : Edit Site

Sicherzugehen ist, ob bei *Connect as...*, *Application user* ausgewählt ist. Das heißt, dass keine bestimmten Anmeldeinformationen verwendet werden und jeder auf das Programm Zugriff hat. Optional kann man bei *Test Settings...* die Einstellungen überprüfen, welche man für die Anwendung angegeben hat.

Wenn man schlussendlich im linken Fenster Actions auf *Browse anotesa.htl-perg.ac.at on *:80 (http)* klickt erscheint die *Application* und das Programm ist auch von anderen Clients mithilfe einer Netzwerkverbindung aufrufbar.

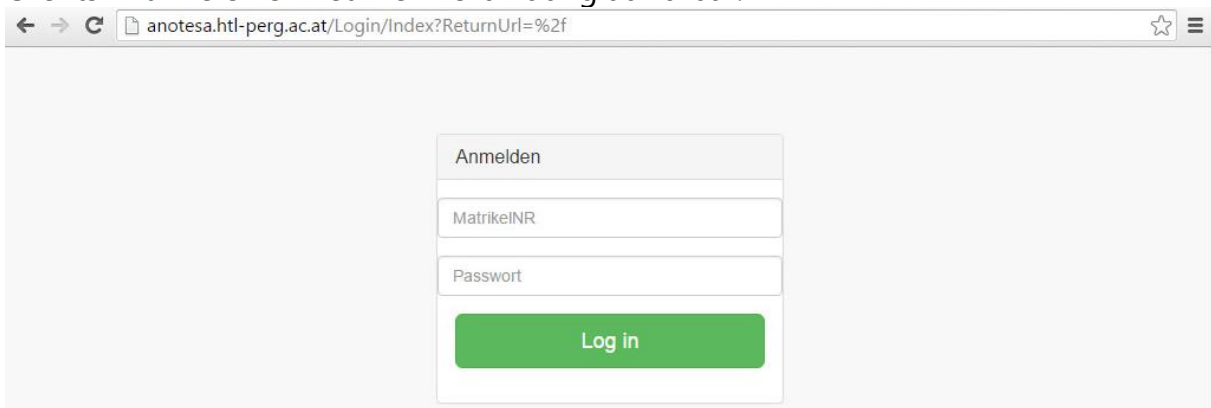


Abbildung 52 : Browse anotesa.htl-perg.ac.at

[17]

6.2.1.3 Architektur Windows Server 2012 R2

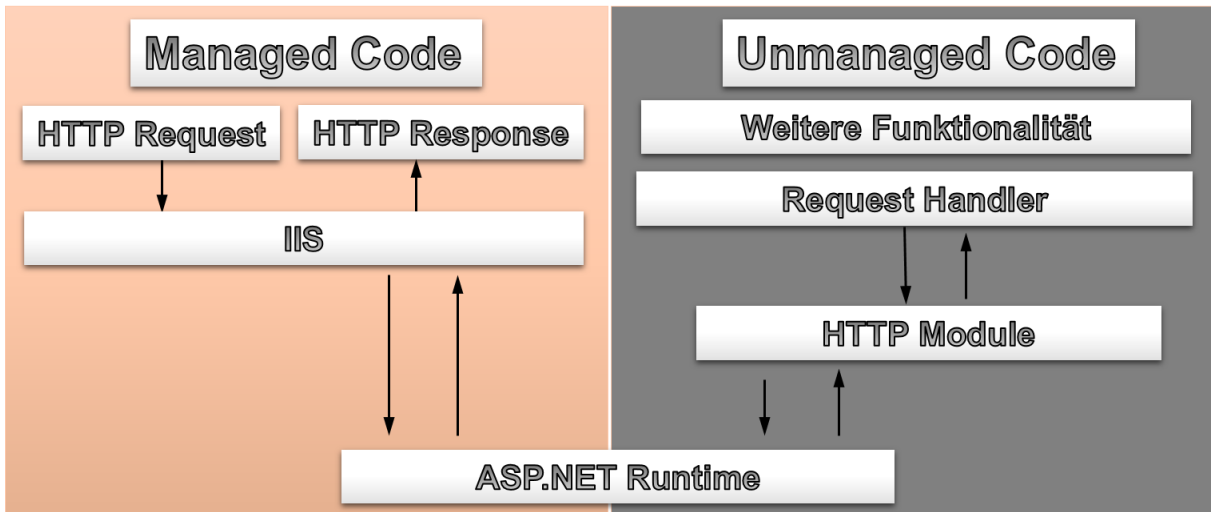


Abbildung 53 : ASP.NET Architektur



6.2.2 Webseite

Zuerst muss man Visual Studio installieren.

Die aktuellste Version findet man hier:

<https://www.visualstudio.com/downloads/download-visual-studio-vs>

Nachdem Visual Studio installiert ist, ladet man sich die neueste JQuery-Version herunter. Diese ist unter diesen Link zu finden:

<https://jquery.com/download/>

Danach lädt man sich Bootstrap herunter.

Unter diesem Link findet man die aktuellste Version:

<http://getbootstrap.com/getting-started/#download>

Falls man ein vorgefertigtes Design von Bootstrap verwenden möchte, kann man dies unter <http://startbootstrap.com/> suchen und herunterladen. Es ist aber darauf hinzuweisen, dass dieses Template bereits Bootstrap, JQuery und alle weiteren benötigten JavaScript-Dateien inbegriffen hat. Also muss man sich hier um keine weiteren Downloads mehr kümmern.

Sind nun alle Komponenten für die Webseite installiert, öffnet man Visual Studio und erstellt ein neues Projekt.

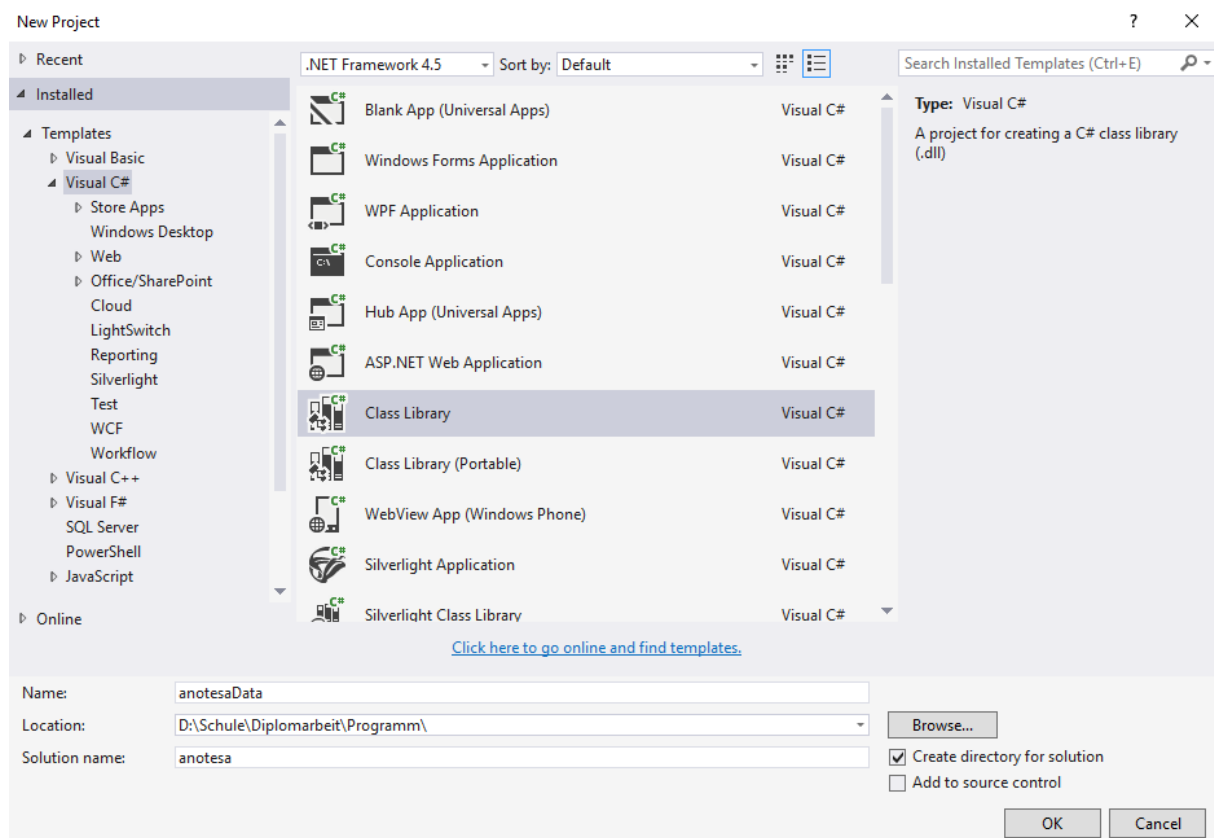


Abbildung 54 : Class Library Projekt

Man benötigt dabei zwei verschiedene Projekte, die in einer Solution sind. In diesem Fall haben wir ein „Class-Library“-Projekt namens „anotesaData“ und ein „ASP.NET Web Application“-Projekt namens „anotesaService“ angelegt. Für das 2. Projekt klickt man mit der rechten Maustaste auf die Solution und wählt Add → new Project aus.

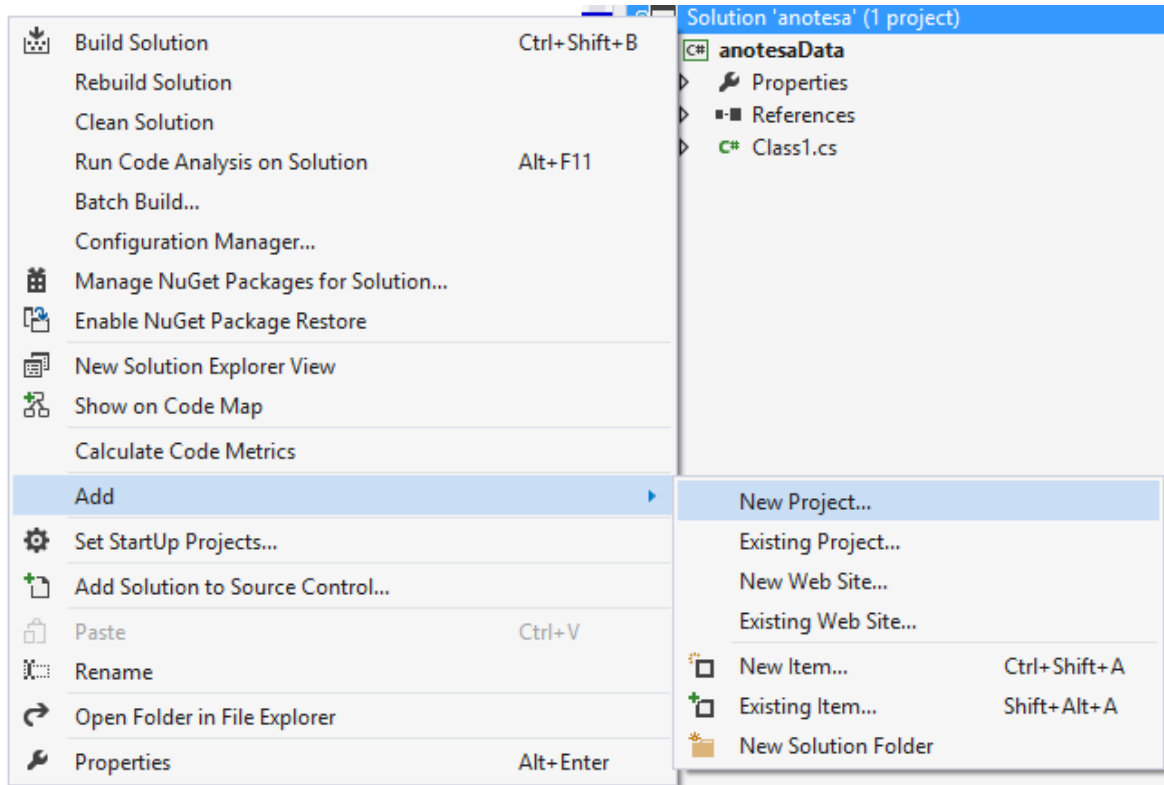


Abbildung 55 : Add New Project

Beim „ASP.NET Web Application“-Projekt muss man folgendes beachten:

Als Template wählt man MVC aus und setzt das Häkchen bei Web API. Zusätzlich klickt man noch auf „Change Authentication“ und wählt „No Authentication“ aus. Danach kann man mit dem Programmieren beginnen.

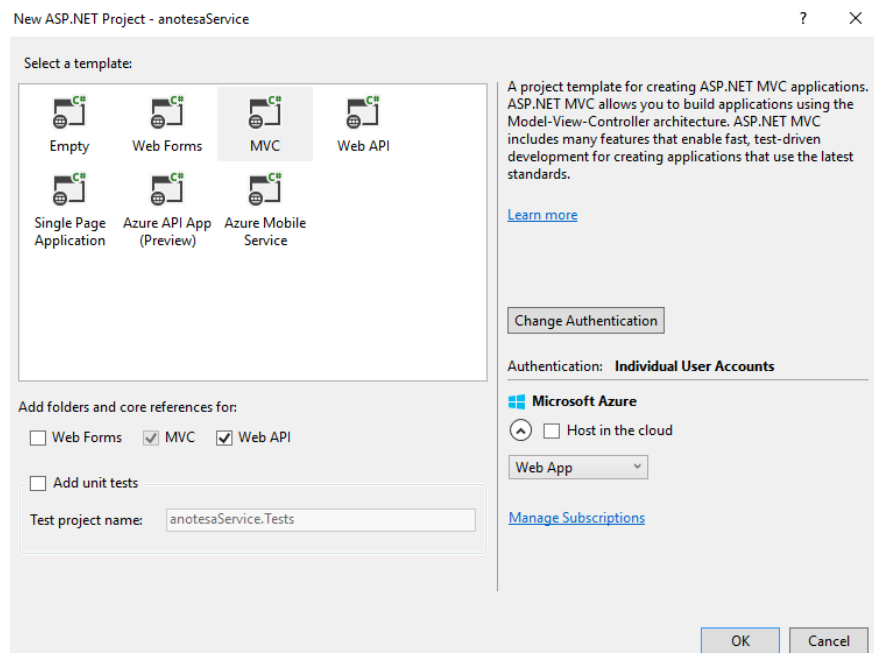


Abbildung 56 : MVC, API



6.3 Risiken

6.3.1 Risiken im Allgemeinen

In beinahe jedem Projekt gibt es gewisse Risiken, die man identifizieren, analysieren und bewerten sollte. Generell werden Risiken in vier verschiedene Gruppen geteilt: technische, wirtschaftliche, politische und soziokulturelle Risiken. [20]

6.3.2 Risiken Anotesa

Allgemeine Risiken sind beispielsweise die zeitlich befristete Fertigstellung und das Einhalten der geplanten Kosten. Da bei der Entwicklung von anotesa jedoch keine Kosten anfallen fällt dieser Teil weg. Anhand einer Matrix kann man die jeweiligen Risiken in Kategorien einteilen. Daraus kann man lesen welche Risiken man am ehesten beachten sollte. Dafür benötigt man die Eintrittswahrscheinlichkeit und den Einfluss auf den Projekterfolg beim Eintritt des Risikos.

Folgende Risiken waren bei unserer Diplomarbeit relevant:

		Eintritts- wahrscheinlichkeit	Einfluss auf Erfolg
1	Keine Fertigstellung zum geplanten Zeitpunkt	Niedrig	Hoch
2	Nichtannahme der Lehrer	Niedrig	Hoch
3	Negative Rezension der Lehrer	Niedrig/hoch	Niedrig
4	Negative Rückmeldungen der Schüler	Niedrig/Hoch	Niedrig
5	Andere Diplomarbeit beschäftigt sich mit dem selben Thema	Niedrig	Hoch

Tabelle 7 : Risikobewertung

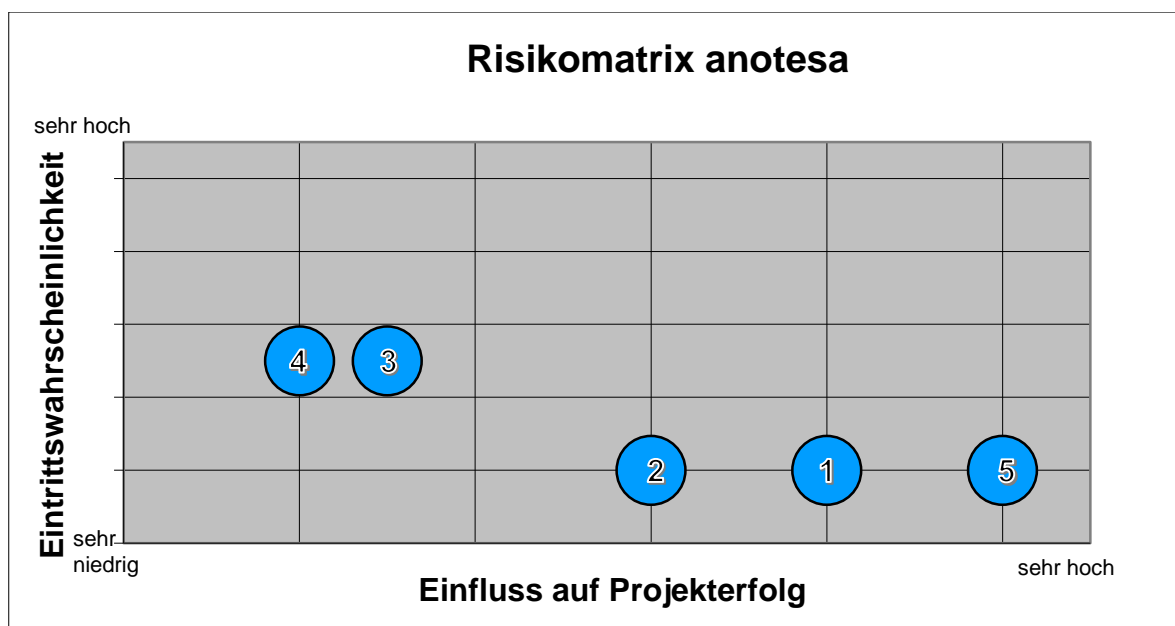


Abbildung 57 : Risikomatrix

6.4 Implementierung

6.4.1 Datenmodell

Unser Datenmodell beinhaltet 12 Tabellen.

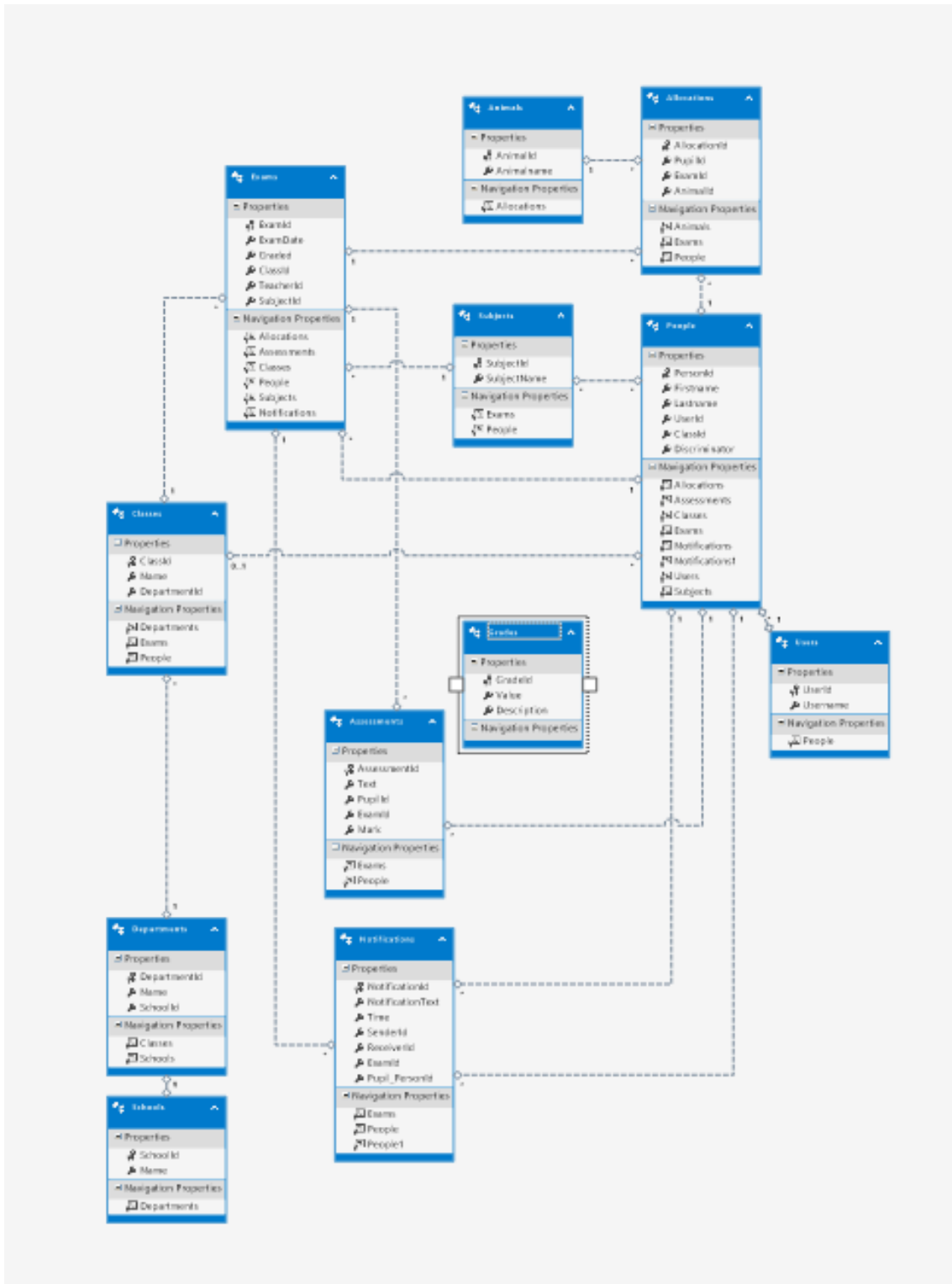


Abbildung 58 : Datenmodell anotesa



6.4.1.1 Übersicht Tabellen

Hier eine Übersicht der Tabellen mit deren Inhalt:

Tabellenname	Beschreibung
Animals	Beinhaltet 40 Tiernamen für die zufällige Zuteilung eines Tieres
Schools	Beinhaltet die verschiedenen Schulen, die unsere Diplomarbeit nutzen werden
Departments	Beinhaltet die verschiedenen Abteilungen der jeweiligen Schulen (z.B. Höhere Abteilung für Informatik)
Classes	Beinhaltet die Klassen einer Abteilung
People	Beinhaltet die Personen, die <i>anotesa</i> nutzen
Users	Beinhaltet für jede Person, die <i>anotesa</i> nutzt, den Usernamen
Exams	Beinhaltet alle erstellten Tests oder Schularbeiten
Grades	Beinhaltet die Noten von 1-5
Notifications	Beinhaltet alle erstellten Benachrichtigungen samt Sender und Empfänger
Subjects	Beinhaltet alle Fächer einer Schule
Allocations	Vereint die Tabellen <i>Animals</i> und <i>People</i> . Hier werden für die Schüler die Tiere zugeordnet
Assessments	Vereint die Tabellen <i>Grades</i> und <i>People</i> . Hier werden für die Schüler Noten zugeordnet
SubjectTeachers	Vereint die Tabellen <i>Subjects</i> und <i>Teachers</i> . Hier werden für die Lehrer die Fächer zugeordnet

Tabelle 8 : Übersicht Datenbanktabellen

6.4.2 ASP.NET

Funktionen wie das Bearbeiten oder Hinzufügen eines Testes wurden, gemäß dem Model-View-Controller-Prinzip, in einem eigenen Controller namens „ExamView“-Controller ausprogrammiert. Dort findet man nicht nur den logischen Aufbau einer Seite, sondern auch die Veränderung auf der Datenbank. Hier nun die wichtigsten Funktionen:

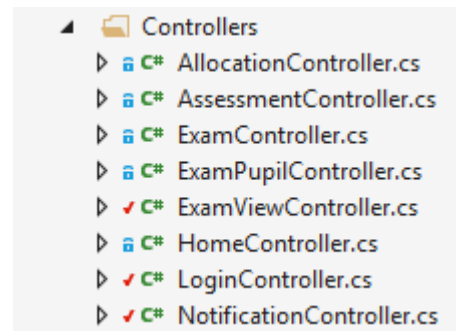


Abbildung 59 : ExamViewController

6.4.2.1 Test hinzufügen

Falls man nun einen Test hinzufügt (siehe : Test hinzufügen auf Seite 49), wird diese Methode aufgerufen. Zunächst wird ein neuer Test erstellt und auf der Datenbank gespeichert. Die Daten werden dafür von den Dropdownlisten bzw. Textfeldern geholt.

```
Exam ex = new Exam
{
    SubjectId = aem.SelectedSubject,
    ExamDate = aem.Date,
    ClassId = aem.SelectedClass,
    Graded = false,
    TeacherId = te.PersonId
};

uow.RepExam.Create(ex);
uow.SaveChanges();
```

Abbildung 60 : Code Test erstellen

Nachdem der Test erstellt wurde, wird für jeden Schüler der ausgewählten Klasse ein Tier zugeordnet. Dies wird ebenfalls auf der Datenbank gespeichert. Zusätzlich wird für jeden Schüler eine Benachrichtigung mit dem Inhalt „<Fach>-Test wurde angelegt“ erstellt.

```
//Zuweisungen der Schüler
IEnumerable<Animal> animals = uow.RepAnimal.Get();
IEnumerable<Pupil> pupils = uow.RepPupil.Get(p => p.ClassId == ex.ClassId);

foreach (Pupil p in pupils)
{
    Random rng = new Random();
    Allocation al = new Allocation
    {
        AnimalId = rng.Next(animals.First().AnimalId, animals.Last().AnimalId),
        PupilId = p.PersonId,
        ExamId = ex.ExamId
    };
    //Benachrichtigungen für Schüler
    Notification notification = new Notification
    {
        NotificationText = uow.RepSubject.GetById(aem.SelectedSubject).SubjectName + "-Test wurde angelegt",
        SenderId = te.PersonId,
        ReceiverId = p.PersonId,
        Time = DateTime.Now,
        ExamId = ex.ExamId
    };
    uow.RepAllocation.Create(al);
    uow.RepNotification.Create(notification);
    uow.SaveChanges();
}
```

Abbildung 61 : Code Test erstellen - Tierzuordnungen und Benachrichtigungen



6.4.2.2 Test bewerten

Dasselbe Prinzip geschieht beim Bewerten eines Testes (siehe : Test bewerten auf Seite 46): Zunächst werden die Noten mit dem jeweiligen Schüler gespeichert. Danach wird wieder eine Benachrichtigung an jeden Schüler geschickt. Diese beinhaltet „<Fach>-Test wurde bewertet“.

```
foreach (Pupil p in ex.Class.Pupils)
{
    gem.SelectedGrades = gem.SelectedGrades.Skip(i).ToList();
    gem.Feedbacks = gem.Feedbacks.Skip(i).ToList();
    //Notenbewertung für Schüler
    Assessment assessment = new Assessment
    {
        ExamId = gem.Exam.ExamId,
        PupilId = p.PersonId,
        Mark = gem.SelectedGrades.First(),
        Text = gem.Feedbacks.First()
    };
    uow.RepAssessment.Create(assessment);
    uow.SaveChanges();

    //Benachrichtigungen für Schüler
    Notification notification = new Notification
    {
        NotificationText = ex.Subject.SubjectName + "-Test wurde bewertet",
        SenderId = te.PersonId,
        ReceiverId = p.PersonId,
        Time = DateTime.Now,
        ExamId = examId
    };

    uow.RepNotification.Create(notification);
    uow.SaveChanges();
    i++;
}
```

Abbildung 62 : Code Test bewerten - Notenzuordnungen und Benachrichtigungen

6.4.2.3 Test bearbeiten

Falls man einen Test bearbeitet hat (siehe : Test bearbeiten auf Seite 47), wird zunächst geprüft, ob die ausgewählte Klasse dieselbe Klasse wie zuvor ist. Wenn dies nicht zutrifft werden alle Tierzuordnungen gelöscht.

```
foreach (Allocation al in eem.Exam.Allocations)
{
    uow.RepAllocation.Delete(al);
}
uow.SaveChanges();
```

Abbildung 63 : Code Tierzuordnungen löschen

Zusätzlich wird der Test bearbeitet und die veränderten Daten auf der Datenbank gespeichert. Dazu bekommt jeder Schüler wieder eine Benachrichtigung mit dem Inhalt „<Fach>-Test wurde bearbeitet“.

```
foreach (Pupil p in pupils)
{
    Notification notification = new Notification
    {
        NotificationText = uow.RepSubject.GetById(eem.SelectedSubject).SubjectName + "-Test wurde bearbeitet",
        SenderId = te.PersonId,
        ReceiverId = p.PersonId,
        Time = DateTime.Now,
        ExamId = eem.Exam.ExamId
    };
    uow.RepNotification.Create(notification);
    uow.SaveChanges();
}
```

Abbildung 64 : Code Test bearbeiten - Benachrichtigungen

6.4.3 JavaScript

Das Dashboard, die Benachrichtigungen und die offenen Tests in der *Navigation-Bar* (siehe : *Navigation-Bar* auf Seite 36), wurden mit JavaScript programmiert, da JavaScript die geeignete Dynamik mit sich bringt. Dafür wurden drei verschiedene JavaScript-Dateien erstellt: *dashboard.js*, *notification.js* und *task-filler.js*, um eine Übersichtlichkeit beizubehalten.

6.4.3.1 Dashboard

In der Datei *dashboard.js* wird der Inhalt des Dashboards, je nachdem welche Rolle angemeldet ist und welcher Button geklickt wurde, dynamisch aktualisiert.

Zuerst wird der Inhalt eines *Hiddenfields* gelesen, wo mit Hilfe einer *Session* gespeichert wird, welche Person gerade angemeldet ist. Danach kann man leicht überprüfen, ob dieses *Hiddenfield* nun leer oder beschrieben ist. Falls es beschrieben ist, speichert man sich nun diese Person.

```
var pupil = $("#pupil").val();
if (typeof pupil !== "undefined") {
    var pu = JSON.parse(pupil);
```

Abbildung 65 : JavaScript - Prüfung ob Schüler

Für jeden Klick auf einen Button wird nun mit Ajax der Content mit Hilfe des APIs geholt. Dafür wird ein Controller erstellt, der alle http-Kommandos wie zum Beispiel *get*, *getById*, *post* und *delete* ausprogrammiert hat. Hier ist der Content für die Benachrichtigungen am Dashboard: Dafür werden alle Daten für die Person vom *Notifications-Controller* geholt.

```
$.ajax({
    url: "/api/notification?personid=" + pu["PersonId"],
    success: notifications,
    error: noData($("#content"))
});
```

Abbildung 66 : Ajax - Notifications Api



Falls nun aber keine Daten zur Person gefunden werden, wird die `noData`-Funktion aufgerufen, die zur Darstellung der *keine Daten*-Seite beiträgt. (siehe : Dashboard Schüler - Keine Daten vorhanden auf Seite 40)

```
function noData(content) {
    content.empty();
    content.append("<div class='well'>"
        + "<h4>Informationen</h4>"
        + "<p>Keine Daten vorhanden</p></div>")
}
```

Abbildung 67 : JavaScript - Funktion "noData"

Findet die Ajax-Funktion doch Daten, so wird der Content aufgebaut. Dazu wird für jede Zeile ein Klick-Event hinzugefügt, das in diesem Fall zur `deleteNotifications`-Funktion führt. (siehe : JavaScript - Funktion "deleteNotifications" auf Seite 62)

```
html += "<div class='col-lg-12'>"
    + "<div class='panel panel-default'>"
    + "<div class='panel-heading'>"
    + "Benachrichtigungen</div>"
    + "<div class='panel-body'>"
    + "<div class='list-group'>";
$.each(data, function (index, item) {
    html += "<a href='#' class='list-group-item' data-exam='" + item.ExamId + "\" id='notification" + item.NotificationId + "'>";
    html += "<i class='fa fa-comment fa-fw'></i> " + item.NotificationText;
    html += "<span class='pull-center text-muted'><em> vom " + changeDate(item.Time) + "</em>";
    html += "<span class='pull-right text-muted small'><em> von " + item.SenderName + "</em>";
    html += "</span></a>";
});
html += "</div></div></div></div>";
content.append(html);

$("a[id^='notification']").click(function () {
    deleteNotifications($(this).attr("id"),$(this).attr("data-exam"));
})
```

Abbildung 68 : JavaScript - Aufbau Benachrichtigungen Dashboard

6.4.3.2 Benachrichtigungen

Die Abfrage, welche Person nun angemeldet ist, ist ident der Funktion des Dashboards (siehe : Dashboard Schüler auf Seite 38).

Nur der Aufbau der Benachrichtigungen ist anders. Zusätzlich kommt noch ein Farbenwechsel von Blau auf Rot, bei mehr als 0 Elementen, dazu.

```
$("#alert-icon").attr("style", "color:indianred");
$("#alert-down").attr("style", "color:indianred");
$.each(data, function (index, item) {
    html += "<li><a href='#' id='not" + item.NotificationId + "\" data-exam='" + item.ExamId + "\"><div>"
        + "<i class='fa fa-comment fa-fw'></i> von " + item.SenderName
        + "<span class='pull-right text-muted small'>" + changeDate(item.Time)+ "</span>"
        + "</div></a></li>"

    if (index < data.length - 1) {
        html += "<li class='divider'></li>";
    }
});
content.append(html);
$("a[id^='not']").click(function () {
    deleteNotifications($(this).attr("id"), $(this).attr("data-exam"));
})
```

Abbildung 69 : JavaScript - Aufbau Benachrichtigungen Navigation-Bar

Falls man nun auf eine Benachrichtigung klickt, ruft das Programm die „deleteNotifications“-Methode auf. Diese löscht die ausgewählte Benachrichtigung mit Hilfe von Ajax und wechselt dann zur Detail-Seite des Tests, welcher die Benachrichtigung anspricht.

```
$.ajax({
  url: "/api/notification/" + notificationid,
  type: "DELETE",
  success: function () {
    if (typeof $("#teacher").val() !== "undefined" || typeof $("#admin").val() !== "undefined") {
      location.href = "/ExamView/DetailExam?examId=" + examid;
    }
    if (typeof $("#pupil").val() !== "undefined") {
      location.href = "/ExamPupil/DetailExam?examId=" + examid;
    }
  }
});
```

Abbildung 70 : JavaScript - Funktion "deleteNotifications"

6.4.3.3 Tasks

Der Aufbau des Navigation-Bar-Reiters „Tasks“ ist wieder ident mit dem der anderen JavaScript-Dateien. Einzig und allein die Methode „animateUpdate“ ist noch hinzugekommen.

```
html += "<li><a href='#\" + exam.ExamId + "\"><div><p>"
+ "<strong>" + exam.SubjectName + "-Test</strong>"
+ "<span class='pull-right text-muted'> war am " + date.getDate() + "." + (date.getMonth() + 1) + "." + date.getFullYear() + "</span>"
+ "<div class='progress progress-striped active'>"
+ "<div id='pbar\" + exam.ExamId + "\" class='progress-bar progress-bar-success' role='progressbar' aria-valuenow='\" aria-valuemin='0\" aria-valuemax='100'>"
+ "<span id='text\" + exam.ExamId + "\"></span>"
+ "</div></div></div></a></li>";

if (index < data.length - 1) {
  html += "<li class='divider'></li>";
}
tasks.append(html);
animateUpdate(exam);
```

Abbildung 71 : JavaScript - Aufbau Tasks Navigation-Bar

In dieser Methode wird nun der Fortschritt der Zeit zum Bewerten ausgerechnet.

```
var now = new Date();
var timeDiff = now.getTime() - Date.parse(exam.ExamDate).getTime();
var perc = Math.round((timeDiff / maxTime) * 100);
var date = (Date.parse(exam.ExamDate).getTime() + maxTime);
var days = convertMS(date - now.getTime()).d;
var hours = convertMS(date - now.getTime()).h;
```

Abbildung 72 : JavaScript - Methode "animateUpdate"

Ist der Fortschritt nun unter 100%, wird die Methode „updateProgress“ aufgerufen.

Die Methode „updateProgress“ ändert nun den Fortschritt der *Progress-Bar* und ändert auch, aufgrund der Prozente, die Farbe der *Progress-Bar*. So ist diese bis zu 60 Prozent grün, von 60-80 Prozent orange und ab 80 Prozent rot. Zusätzlich werden dem Lehrer noch Benachrichtigungen geschickt, um ihn über die verbleibende Zeit zu erinnern.

```
if (percentage <= 60) {
  $("#pbar"+exam.ExamId).attr("class", "progress-bar progress-bar-success");
}
```

Abbildung 73 : JavaScript - Methode "updateProgress"



6.4.4 Anmeldung mit IF-Usern

Um einen Login mit den IF-Usern der Schüler bzw. den Usern der Lehrer durchführen zu können, benötigt man eine *LDAP*-Verbindung zum Server der Schule. Diese Verbindung haben wir im LoginController, sowie in der Web.Config konfiguriert.

LDAP

LDAP steht für *Lightweight Directory Access Protokoll*. Es ist ein Netzwerkprotokoll zur Abfrage von Informationen verteilter Verzeichnisdienste. Es befindet sich in der Anwendungsschicht des *ISO-OSI-Modells*. In der untenstehenden Tabelle werden die Schichten vereinfacht dargestellt:

Schichtname	Protokolle in der Schicht
Anwendung	LDAP
Transport	UDP, TCP
Internet	IP
Physikalisch	Zugang mit Ethernet, Token Bus, Token Ring, etc.

Tabelle 9 : *LDAP-Protokoll*

[18]

An dieses Protokoll orientiert sich das *Active Directory*. Das *Active Directory* erhält die verschiedenen Objekte (in unserem Fall die Schüler und Lehrer). Im *Active Directory* besitzt jedes Objekt zahlreiche Namensattribute, wie zum Beispiel den Anzeige- und Usernamen. Zusätzlich kann man mit Hilfe des *memberOf*-Attributs die Gruppen, in denen das Objekt verzeichnet ist, herausfinden.

Der *LDAP*-Pfad ist aber stets eindeutig; er identifiziert das Objekt innerhalb der Hierarchie des Verzeichnisses. Dabei beginnt er von der untersten Ebene und wandert bis zum Ursprung des Verzeichnisses (die Domäne).

In unserem Fall sieht der *LDAP*-Pfad so aus:

CN=Users,DC=main,DC=htl-perg,DC=at

Wobei *CN* für *Common Name*, also für den Objektnamen selbst, steht und *DC* für *Domain Component*, also für die einzelnen Abschnitte der Domäne. Wenn man nun unseren *LDAP*-Pfad von unten nach oben durchgeht, erhält man den Objektnamen Users in der Domäne „main.htl-perg.at“. [19]

6.4.4.1 Web.Config

In der Datei *Web.config*, die man unter *anotesaService* findet, muss man unter `<system.web>` eine neue Authentifizierung eintragen. Dazu wählt man den *Forms*-Modus und ändert den Namen der Forms auf *ADAuthCookie*. Zusätzlich muss man unter `<membership>` einen Provider mit dem richtigen *ConnectionString*, *Username* und *Password* eingeben. In diesem Fall sind der *connectionStringName* *ADConnectionString* und der *attributeMapUsername* *sAMAccountName*.

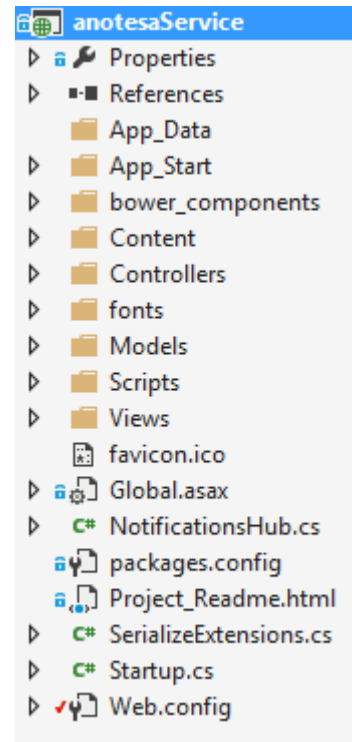


Abbildung 74 : LDAP - anotesaService

```
<system.web>
  <compilation debug="true" targetFramework="4.5" />
  <httpRuntime targetFramework="4.5" />
  <authentication mode="Forms">
    <forms name=".ADAuthCookie" loginUrl="~/Login/Index" timeout="45" slidingExpiration="false" protection="All" />
  </authentication>
  <membership defaultProvider="ADMembershipProvider">
    <providers>
      <clear />
      <add name="ADMembershipProvider" type="System.Web.Security.ActiveDirectoryMembershipProvider"
        connectionStringName="ADConnectionString"
        connectionUsername="mexxo" connectionPassword="mexxo"
        attributeMapUsername="sAMAccountName" />
    </providers>
  </membership>
</system.web>
```

Abbildung 75 : LDAP - system.web

Weiter unten fügt man noch einen neuen *ConnectionString* hinzu, den man bei den Providern weiter oben benötigt. Dieser lautet in unserem Schulnetzwerk: *LDAP://172.16.12.20/CN=Users,DC=main,DC=htl-perg,DC=at*

```
<connectionStrings>
  <add name="ADConnectionString" connectionString="LDAP://172.16.12.20/CN=Users,DC=main,DC=htl-perg,DC=at" />
</connectionStrings>
```

Abbildung 76 : LDAP - connectionString



6.4.4.2 LoginController

Im LoginController, der für einen erfolgreichen Login verantwortlich ist und den man unter *anotesaService – Controllers* findet, fragt man nun mit

```
if (Membership.ValidateUser(lm.Username, lm.Password))
```

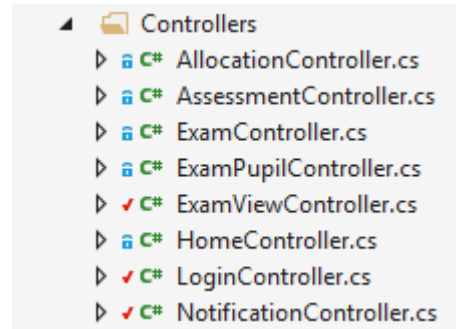


Abbildung 77 : LDAP - LoginController

ab, ob es einen User mit diesem Usernamen und Passwort überhaupt gibt und ob er sich erfolgreich anmelden kann. Falls ja, holt man sich mit der Methode *GetUserProperty*-Attribute wie den Anzeigenamen, die Beschreibung oder die dazugehörigen Gruppen. Dafür gibt man folgendes ein:

```
string Name = GetUserProperty(lm.Username, "DisplayName");
```

Abbildung 78 : LDAP - Aufruf GetUserProperty

Die Methode holt sich dann anhand des Usernamen alle Ergebnisse, die zu diesem Usernamen und Attribut passen, und gibt sie als String zurück. Nun kann man sich leicht aus dem String die gewünschten Daten zusammenschneiden.

```
public static string GetUserProperty(string accountName, string propertyName)
{
    DirectoryEntry entry = new DirectoryEntry();
    entry.Username = "mexxo";
    entry.Password = "mexxo";
    entry.Path = "LDAP://172.16.12.20/CN=Users,DC=main,DC=htl-perg,DC=at";
    entry.AuthenticationType = AuthenticationTypes.Secure;

    DirectorySearcher search = new DirectorySearcher(entry);
    search.Filter = "(SAMAccountName=" + accountName + ")";
    search.PropertiesToLoad.Add(propertyName);

    SearchResultCollection results = search.FindAll();
    if (results != null && results.Count > 0)
    {
        return results[0].Properties[propertyName][0].ToString();
    }
    else
    {
        return "Unknown User";
    }
}
```

Abbildung 79 : LDAP - GetUserProperty

Die Daten speichern wir dann in unserer Datenbank, um eine erleichterte Umgangsform mit den anderen Tabellen zu ermöglichen.

7 Aufgetretene Probleme

7.1 Allgemeine Probleme

7.1.1 Nicht klar definierte Lösung

Gerade am Anfang war es für uns schwer eine genaue Lösung zu definieren. So wurde etwa jede Woche etwas am Produkt verändert. Meist lag es daran, dass wir unsere Wünsche technisch nicht reibungslos umsetzen konnten.

Lösung:

Durch eine klare Definition der Diplomarbeit der beiden Diplomanden konnte dieses Problem schnell gelöst werden.

7.2 Probleme Webseite

7.2.1 LDAP-Verbindung erstellen

Die Anmeldung der Schüler mit ihren IF-Usern gestaltete sich doch schwieriger als Anfangs gedacht.

Lösung:

In einer kurzen Besprechung mit Herrn Dipl.-Ing. Dr. Michael Buchberger erklärte er uns die Vorgehensweise mit *LDAP*. Danach konnten wir dieses Problem rasch und einfach lösen.

7.2.2 Fehlerhafte Erreichbarkeit der Webseite

Die Installation der *Roles* von *ASP.Net 3.5* über **Control Panel\Programs and Features\Turn Windows features on or off** schlug fehl, da keine Verbindung zum *Microsoft Update Server* hergestellt werden konnte.

Danach haben wir das komplette *.Net Framework 3.5* heruntergeladen und versucht zu installieren. Da dieses jedoch wieder eine Verbindung zum *Microsoft Update Server* aufbauen wollte, lief dies auch schief.

Lösung:

<http://www.smart-arab.com/2013/12/install-net-on-windows-8-1-and-windows-10-offline/>

Diese Webseite beinhaltet ein Offline-Tool *DotNet Framework 2.0 and 3.5 Offline and Standalone Installer*, welches die Installation möglich macht. Nach der Überprüfung wurde erfolgreich festgestellt, dass die Rollen vorhanden sind und die Webseite über den Server aufrufbar ist.



8 Qualitätssicherung

Bei der Qualitätssicherung in einem EDV-Projekt wird großen Wert darauf gelegt, dass alle Eigenschaften und Merkmale des Projekts, ihre Richtigkeit haben. Speziell anotesa soll sowohl von Lehrern als auch Schülern verwendet werden. Somit ist es wichtig, dass die Benutzeroberfläche einerseits ein anspruchsvolles und schlichtes Design hat, im Gegenzug jedoch ein benutzerfreundliches Bedienungserlebnis bieten soll. Darüber hinaus sollen bei Klicken von Links, keine zu langen Reaktionszeiten entstehen.

Wichtig ist ebenfalls, dass die Daten, wie beispielsweise die Noten der Schüler, vertraulich behandelt werden. So wie bei der konventionellen Beurteilung soll ein Schüler nur seine eigene Note sehen können, der Lehrer jedoch alle vergebene Noten eines Tests oder einer Schularbeit.

9 Testung

9.1 Testlauf mit Schüler

9.1.1 Durchführung

Am Donnerstag, dem 29.10.2015, wurde die Methode von anotesa zum ersten Mal im Schulfach Deutsch in der Klasse 5AHIF getestet. Am Tag der Schularbeit wurde ein DIN A4 Blatt mit Tiernamen ausgeteilt, auf dem sich jeder Schüler einen bestimmten Namen aussuchte und sich merkte. Beim Test wurde nun nicht der eigene Name, sondern der Name des Tieres angegeben. Nach der Beurteilung wurde Frau Prof. Mag. Lindinger-Pesendorfer eine Übersicht mit allen teilnehmenden Schülern und dessen Tiernamen überreicht, damit auch in ihrem Notenverzeichnis die richtigen Schüler zugeteilt sind.

9.1.2 Umfrage

Nach der ersten Umsetzung der Methode von anotesa in der 5. Klasse wurde von den Schülern eine Umfrage ausgefüllt. In der Umfrage soll grundsätzlich erörtert werden, ob und inwiefern sich das Gefühl der Schüler beim Schreiben von Schularbeiten unter der Anwendung von anotesa verändert hat. Außerdem soll ein Überblick geschaffen werden, wie das Konzept generell ankommt und ob sie es weiter bzw. in anderen Schulfächern verwenden wollen. Im folgenden Abschnitt werden die Fragen und die möglichen Antwortmöglichkeiten angeführt.

Bitte bewerten Sie das Konzept von anotesa.										
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sehr schlecht									Sehr gut	

Bitte bewerten Sie Ihr Gefühl während dem Test/der SA.										
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sehr unsicher									Sehr sicher	

Bitte bewerten Sie Ihr Gefühl nach dem Test/der SA.										
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sehr unsicher									Sehr sicher	

Sind Sie der Meinung, dass sich mit der Durchführung des/der anonymisierten Tests/SA etwas Wesentliches verändert hat? (Eindrücke, Gefühle, etc.)										
Ja <input type="checkbox"/>					Nein <input type="checkbox"/>					
Bitte erläutern Sie:										

Würden Sie diese Methode auch in anderen Schulfächern verwenden wollen?										
Ja <input type="checkbox"/>					Nein <input type="checkbox"/>					



9.1.3 Analyse

An der Umfrage nahmen insgesamt 18 Personen teil. Es wurden alle Umfragebögen abgegeben und darunter war kein ungültiger Stimmzettel.

Allgemein kam das Konzept von anotesa sehr gut an. Lediglich ein kleiner Teil von etwa 3 Personen bewertet das Konzept nur mittelmäßig. Hervor ragt jedoch, dass es keine einzige negative Stimme gab. Man kann davon ausgehen, dass die Methode der anonymen Tests die Aufmerksamkeit der Schüler geweckt hat, und dass eine faire Benotung auch im Interesse der Schüler ist.

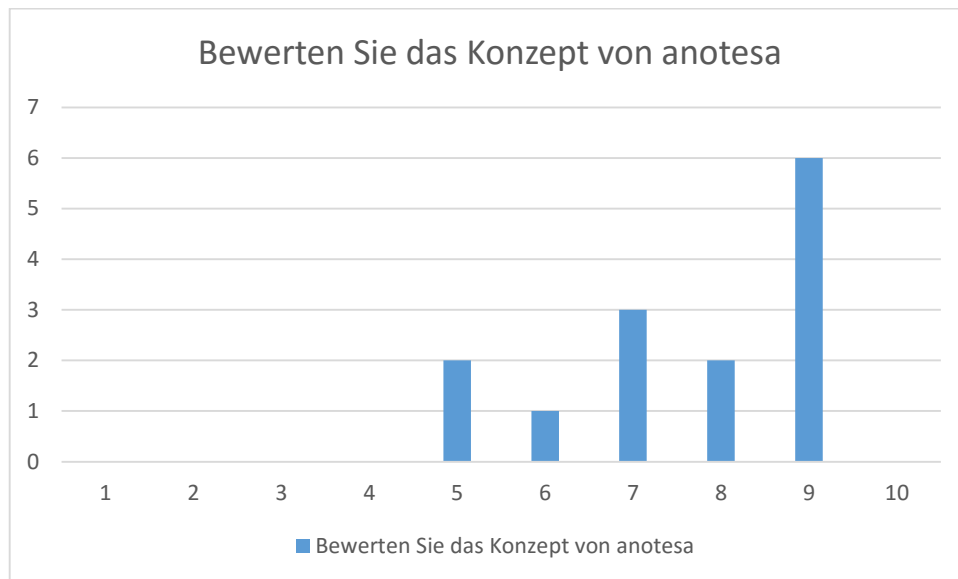


Abbildung 80 : Diagramm Umfrage 'Konzept'

Das Gefühl der Schüler während und nach der Schularbeit war überwiegend positiv. Der Großteil der Schüler gab an, trotz der Anonymität keine großen Einbußen gemacht zu haben. Zwischen ‚dem Gefühl während‘ und ‚nach der Schularbeit‘ gibt es keine großen Unterschiede. Erst nach der Beurteilung geht die Schere weiter auseinander. Mehrere Personen wurden besser benotet als sie erhofft haben. Jedoch gibt es auch einige, die mit einem besseren Ergebnis gerechnet haben. Nach Rücksprache mit den Schülern wurde jedoch dementiert, dass der Grund bei der Methode von anotesa läge.

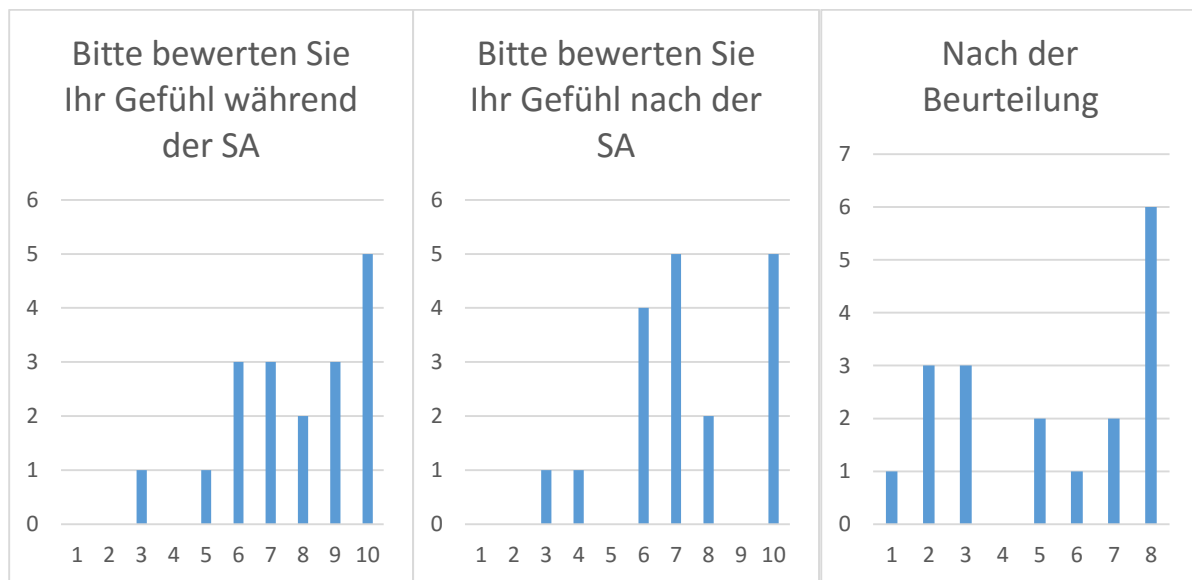


Abbildung 81 : Diagramm Umfrage 'Gefühl'

Die Mehrheit der teilnehmenden Personen ist der Meinung, dass sich bei der Durchführung der Schularbeiten etwas Grundlegendes geändert hat. Was man jedoch aus der 2. Frage - ob sie die Methode auch in anderen Schulfächern verwenden wollen würden - lesen kann ist, dass die Änderung bei den meisten Schülern positiv ankam.

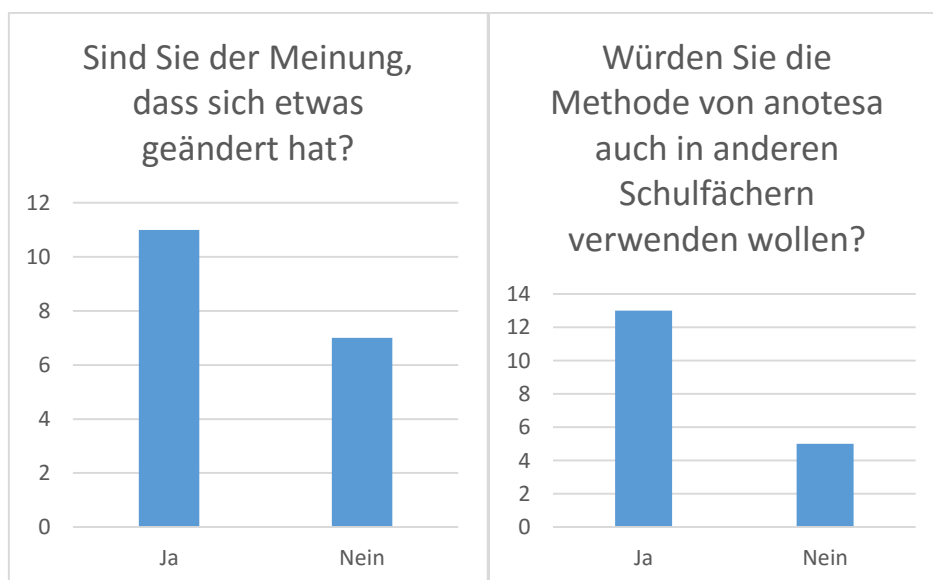


Abbildung 82 : Diagramm Umfrage 'Änderung'



9.1.4 Rezension Lehrer

Auch Frau Prof. Mag. Lindinger-Pesendorfer gab ein Feedback aus der Sicht der Lehrkräfte. Für sie war die Methode ebenfalls neu und führte diese zum ersten Mal durch. Bereits bei den ersten Schularbeiten merkte sie, dass das Beurteilen ohne jegliche persönliche Einflüsse leichter von Hand ging. Dennoch wäre nicht von Nachteil zu wissen, wer die besagte Arbeit gefertigt hat, um zu kontrollieren, ob diese nicht doch jeden Übungsaufsatz mit Bravur fertiggestellt hat, um bei der eigentlichen Benotung ein Auge zuzudrücken. Der große Vorteil sei jedoch, es werden rein die Arbeit bzw. das Können des Schülers benotet.

Als einzigen negativen Punkt merkte Frau. Prof. Lindinger-Pesendorfer an, dass sie während der Arbeitszeit nicht durch die Reihen gehen konnte, um die Anonymität zu gewährleisten. Somit besteht die Gefahr, dass einige Schüler unerlaubte Hilfsmittel benutzen. Dieses Problem kann jedoch aus der Welt geschaffen werden, indem sie doch durch die Reihen geht, aber nicht die einzelnen Bildschirme im Auge behält.

9.2 Testfälle

9.2.1 Übersicht Testfälle

Testfall Nummer	Beschreibung	Erwartetes Ergebnis	Erfolgreich?
1	Einloggen mit IF-User	Man kann sich erfolgreich mit seinem IF-User und Passwort anmelden	Ja
2	Dashboard-Funktionalitäten	Beim Klick auf einen Button wird der Content geändert	Ja
3	Test erstellen	Test wird erfolgreich erstellt und landet in der Datenbank	Ja
4	Test bearbeiten	Test wird erfolgreich bearbeitet und wird geändert auf die Datenbank geladen	Ja
5	Test bewerten	Test wird erfolgreich bewertet und die Noten werden auf die Datenbank geschrieben	Ja
6	Test löschen	Test wird erfolgreich gelöscht und aus der Datenbank genommen	Ja
7	Klick auf Benachrichtigung	Benachrichtigung wird gelöscht und man wird zur Detail-Seite des Testes geführt	Ja
8	Klick auf Task	Man wird zur Detail-Seite des Testes geführt	Ja

Tabelle 10 : Übersicht Testfälle



9.2.2 Testfall 1 – Einloggen mit IF-User

Schritte:

1. <http://anotesa.htl-perg.ac.at> aufrufen
2. IF-User und Passwort eingeben
3. Auf „Log in“-Button klicken

Erwartetes Ergebnis: Der Benutzer kann sich schnell mit seinem User anmelden

Ergebnis: schneller Login mit seinen Daten

Anmelden

IF110042

.....|

Log in

Abbildung 83 : Testfall 1 – Einloggen mit IF-User

9.2.3 Testfall 2 – Dashboard Funktionalitäten

Schritte:

1. <http://anotesa.htl-perg.ac.at> aufrufen
2. IF-User und Passwort eingeben
3. Auf „Log in“-Button klicken
4. Auf einen Der Dashboard-Buttons klicken (zum Beispiel Noten-Diagramm)

Erwartetes Ergebnis: Der Content wird anhand des Buttons angepasst.

Ergebnis: Content wird angepasst



Abbildung 84 : Testfall 2 – Dashboard Funktionalitäten

9.2.4 Testfall 3 – Test erstellen

Schritte:

1. <http://anotesa.htl-perg.ac.at> aufrufen
2. IF-User und Passwort eingeben (Lehrer oder Administrator)
3. Auf „Log in“-Button klicken
4. Links auf der Navigationbar den Reiter Test öffnen
5. Auf den Unterreiter „Test hinzufügen“ klicken
6. Einen Gegenstand, ein Datum und eine Klasse auswählen
7. Auf das blaue Häkchen drücken

Erwartetes Ergebnis: Der Test Test hinzufügen

wird mit den ausgewählten Daten auf der Datenbank gespeichert und man wird zur Testübersicht weitergeleitet.

Ergebnis: Test wird gespeichert und man wird zur Testübersicht weitergeleitet.

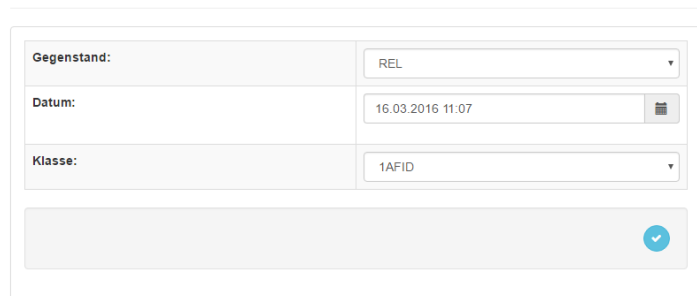


Abbildung 85 : Testfall 3 - Test erstellen

9.2.5 Testfall 4 – Test bearbeiten

Schritte:

1. <http://anotesa.htl-perg.ac.at> aufrufen
2. IF-User und Passwort eingeben (Lehrer oder Administrator)
3. Auf „Log in“-Button klicken
4. Links auf der Navigationbar den Reiter Test öffnen
5. Auf den Unterreiter „Tests ansehen“ klicken
6. Einen Test auswählen und auf die Zeile klicken
7. Auf den blauen Button „Bearbeiten“ klicken
8. Gewünschte Änderungen durchführen

Erwartetes Ergebnis: Der Test wird geändert auf der Datenbank gespeichert und man wird zur Testübersicht zurückgeleitet.

Ergebnis: Test wird geändert gespeichert und man wird zur Testübersicht zurückgeleitet.

Test bearbeiten

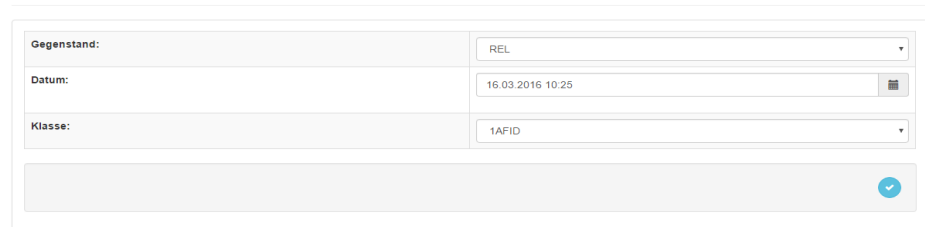


Abbildung 86 : Testfall 4 - Test bearbeiten

9.2.6 Testfall 5 – Test bewerten

Schritte:

1. <http://anotesa.htl-perg.ac.at> aufrufen
2. IF-User und Passwort eingeben (Lehrer oder Administrator)
3. Auf „Log in“-Button klicken
4. Links auf der Navigationbar den Reiter Test öffnen
5. Auf den Unterreiter „Tests ansehen“ klicken
6. Einen Test auswählen und auf die Zeile klicken
7. Auf den weißen Button „Bewerten“ klicken
8. Gewünschte Bewertung durchführen

Erwartetes Ergebnis: Alle Schüler der Klasse, die bei diesen Test zugeordnet waren, bekommen eine Note zugeteilt, die dann auf der Datenbank gespeichert wird. Man wird zur Testübersicht zurückgeleitet.

Test-ID	Note	Feedback
Wal	Sehr Gut	
Clownfisch	Sehr Gut	

Abbildung 87 : Testfall 5 - Test bewerten

Ergebnis: Noten werden gespeichert und man wird zur Testübersicht zurückgeleitet.

9.2.7 Testfall 6 – Test löschen

Schritte:

1. <http://anotesa.htl-perg.ac.at> aufrufen
2. IF-User und Passwort eingeben (Lehrer oder Administrator)
3. Auf „Log in“-Button klicken
4. Links auf der Navigationbar den Reiter Test öffnen
5. Auf den Unterreiter „Tests ansehen“ klicken
6. Einen Test auswählen und auf die Zeile klicken
7. Auf den roten Button „Löschen“ klicken
8. Beim Pop-Up „Löschen“ auswählen

Erwartetes Ergebnis: Gewünschter Test wird gelöscht und man wird zur Testübersicht zurückgeleitet.

Ergebnis: Gewünschter Test wird gelöscht und man wird zur Testübersicht zurückgeleitet.

Löschen bestätigen

Sie sind dabei diesen Test zu löschen

Wollen Sie wirklich fortfahren?

Abbrechen Löschen

Abbildung 88 : Testfall 6 - Test löschen

9.2.8 Testfall 7 – Klick auf Benachrichtigungen

Schritte:

1. <http://anotesa.htl-perg.ac.at> aufrufen
2. IF-User und Passwort eingeben
3. Auf „Log in“-Button klicken
4. Oben auf der Navigationbar das „Glocken-Symbol“ klicken
5. Eine Benachrichtigung auswählen und darauf klicken

Erwartetes Ergebnis: Die ausgewählte Benachrichtigung wird gelöscht und man wird zur Test-Detail-Seite des Testes weitergeleitet.

Ergebnis: Benachrichtigung wird gelöscht und man wird zur Test-Detail-Seite weitergeleitet.

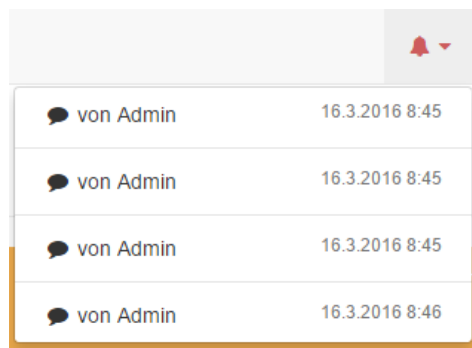


Abbildung 89 : Testfall 7 - Klick auf Benachrichtigungen

9.2.9 Testfall 8 – Klick auf einen Task

Schritte:

1. <http://anotesa.htl-perg.ac.at> aufrufen
2. IF-User und Passwort eingeben (Lehrer)
3. Auf „Log in“-Button klicken
4. Oben auf der Navigationbar das „Tasks-Symbol“ klicken
5. Einen Task auswählen und darauf klicken

Erwartetes Ergebnis: Man wird zur Test-Detail-Seite des Testes weitergeleitet.

Ergebnis: Weiterleitung zur Test-Detail-Seite

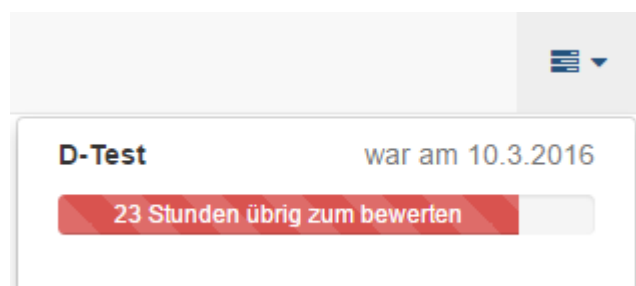


Abbildung 90 : Testfall 8 - Klick auf einen Task



10 Impressum

Projekttitel

anotesa – anonymous testing application

Schule

**Höhere Technische Bundeslehranstalt Perg
Machlandstr. 48
A-4320 Perg**

Schuljahr

2015-2016

Klasse

5AHIF

Projektteam

**Anika Seibezeder
Matthäus Pfusterschmid**

Betreuungslehrer

Mag. Rupert Obermüller

Auftraggeber

HTL Perg

Kontaktperson

Prof. Mag. Rosmarie Lindinger-Pesendorfer

11 Resümee

11.1 Resümee – Anika Seibezeder

Diese Diplomarbeit erlaubte es mir, mein Wissen im Bereich Webentwicklung mit JavaScript zu verbessern. Doch gerade am Beginn kostete es mich viel Zeit, mich in die entsprechenden Technologien, wie zum Beispiel Bootstrap und JavaScript, einzulesen. Dies verzögerte das Öffnen der vorgegebenen Termine.

Leider gab es bei unserer Diplomarbeit einen kleinen Mangel an Projektentwicklungsfähigkeiten und so kam es manchmal zu Kommunikationsproblemen im Team. Diese konnten mein Diplomarbeitspartner und ich aber immer lösen und danach waren wir beide sehr motiviert.

Hätte ich noch einmal die Wahl, eine Diplomarbeit zu schreiben, würde ich mich jederzeit wieder dafür entscheiden. Doch ich würde meinen Zeitplan gewissenhafter durchführen und die zu machende Arbeit gerecht aufteilen.

11.2 Resümee – Matthäus Pfusterschmid

Durch den Unterricht an der HTL Perg haben wir bereits reichlich Erfahrung im Bereich durchführbare Projekte gesammelt. Dies nun in Verbindung mit einer Diplomarbeit anzuwenden ist im Großen und Ganzen kein Neuland. Hat man jedoch im Hinterkopf, dass dieses Projekt nun im Rahmen des letzten Jahrgangs und vor allem nicht im Unterricht realisiert werden muss, kommen mir im Nachhinein Bedenken, dass Projektplanung ein wesentlicher Punkt ist, welcher nicht vernachlässigt werden darf. So kam es auch zum Ende hin teilweise zu Zeitschwierigkeiten.

Allgemein empfinde ich die Erstellung einer Diplomarbeit als sehr fördernd und es spiegelt vor allem wieder, was aus dem Unterricht, seien es technische oder projektspezifische Tätigkeiten, angewandt werden kann. Außerdem bekommt man einen besseren Bezug zum Fehler- und Problemlösen und zu neuen Technologien, welche man sich erarbeiten muss.



12 Anhang

12.1 Zuständigkeiten

12.1.1 [Matthäus Pfusterschmid](#)

- 1 Einleitung
- 2.5 Projektteam
- 3.1 Ist-Zustand
- 3.2 Verunsicherung bei der Entstehung
- 3.3 Idee
- 3.4 Zielsetzung
- 3.5 Funktionsumfang
- 4.2 Aufgaben und Verantwortungen
- 4.4 Zeitplanung
- 4.5 Pflichtenheft
- 5 Software – Verwendete Technologien
- 6.2 Installation
- 6.3 Risiken
- 8 Qualitätssicherung
- 9.1 Testlauf mit Schüler
- 11.2 Resümee – Matthäus Pfusterschmid

12.1.2 [Anika Seibezeder](#)

- 2.1 Auftraggeber
- 2.2 Kontaktperson
- 2.3 Betreuungslehrer
- 2.4 Hierarchie
- 3.1 Ist-Zustand
- 3.4 Zielsetzung
- 4.1 Projektstrukturplan
- 4.3 Meilensteine
- 4.5 Pflichtenheft
- 4.6 Lieferumfang
- 6.1 Darstellung
- 6.2.2 Webseite
- 6.4 Implementierung
- 7.1 Allgemeine Probleme
- 7.2 Probleme Webseite
- 9.2 Testfälle
- 10 Impressum
- 11.1 Resümee – Anika Seibezeder

12.2 Verzeichnisse

12.2.1 Quellenverzeichnis

- [1] C# / Microsoft Corporation
<http://jamshidhashimi.com/2014/07/30/history-of-csharp-language>
Jänner 2016
- [2] ASP.NET / Microsoft Corporation
<https://msdn.microsoft.com/de-de/library/4w3ex9c2%28v=vs.85%29.aspx>
Jänner 2016
- [3] HTML / World Wide Web Consortium
<http://www.tizag.com/htmlT/>
Februar 2016
- [4] CSS / World Wide Web Consortium
<http://www.inf.ufsc.br/~bosco/downloads/Livro-Java-Como-Programar-Deitel-Ed6/additional/addnlRefs/CSS/CSS.pdf>
Februar 2016
- [5] CSS / Håkon W Lie
<https://www.w3.org/People/howcome/p/cascade.html>
Februar 2016
- [6] JavaScript / Peter-Paul Koch
<http://www.quirksmode.org/js/intro.html>
Jänner 2016
- [7] JavaScript / Netscape, Sun Microsystems
http://link.springer.com/chapter/10.1007/978-1-4302-0062-8_1#page-2
Jänner 2016
- [8] JQuery
http://www.tutorialspoint.com/jquery/jquery_tutorial.pdf
März 2016
- [9] AJAX
<http://cglab.ca/~morin/teaching/2405/notes/ajax1.pdf>
März 2016
- [10] Bootstrap
<http://getbootstrap.com/getting-started/>
März 2016
- [11] Morris
<https://morrisjs.github.io/morris.js/index.html>
März 2016
- [12] Datetimepicker
<https://eonasdan.github.io/bootstrap-datetimepicker/>
Februar 2016
- [13] Windows Server 2012 / Microsoft Corporation
<https://www.microsoft.com/en-us/server-cloud/whats-new.aspx>
Februar 2016



- [14] | Windows Server 2012 / Microsoft Corporation
<https://technet.microsoft.com/en-us/library/hh831786>
Februar 2016
- [15] | Google Drive / Google Inc.
<http://www.wsj.com/articles/SB10001424052702303459004577362111867730108>
Jänner 2016
- [16] | Visual Studio / Microsoft Corporation
<https://msdn.microsoft.com/en-us/vstudio/ms788708.aspx>
Jänner 2016
- [17] | Installation IIS
<http://www.iis.net/learn/get-started/whats-new-in-iis-8/iis-80-using-aspnet-35-and-aspnet-45>
März 2016
- [18] | LDAP
https://de.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol
März 2016
- [19] | Active Directory
<https://www.faq-o-matic.net/2008/01/13/ldap-grundlagen-fuer-active-directory>
März 2016
- [20] | Use Case Diagramm
Verantwortungsmatrix
Zeitplanung
Risiken
Projektstrukturplan
Systemplanung und Projektentwicklung HTL III (MANZ Verlag) / Felix Schwab, Ingrid Schwab-Matkovits
November 2015

12.2.2 Abbildungsverzeichnis

ABBILDUNG 1: LOGO HTL PERG	9
ABBILDUNG 2 : PROF. MAG. ROSMARIE LINDINGER-PESENDORFER	9
ABBILDUNG 3 : PROF. MAG. RUPERT OBERMÜLLER.....	10
ABBILDUNG 4 : HIERARCHIE	10
ABBILDUNG 5: ANIKA SEIBEZEDER	11
ABBILDUNG 6 : MATTHÄUS PFUSTERSCHMID	11
ABBILDUNG 7 : USE-CASE-DIAGRAMM ANOTESA	15
ABBILDUNG 8 : PROJEKTSTRUKTURPLAN ANOTESA	19
ABBILDUNG 9 : MEILENSTEINE ANOTESA.....	21
ABBILDUNG 10 : GANTT-PLAN ANOTESA	22
ABBILDUNG 11 : CSS	28
ABBILDUNG 12 : JAVASCRIPT	29
ABBILDUNG 13 : JQUERY.....	30
ABBILDUNG 14 : BOOTSTRAP	31
ABBILDUNG 15: SCREENSHOT DES STARTFESTERS EINES WINDOWS SERVERS	33
ABBILDUNG 16: SCREENSHOT DES SERVER MANAGER.....	33
ABBILDUNG 17: GOOGLE DRIVE	34
ABBILDUNG 18: VISUAL STUDIO	34
ABBILDUNG 19 : LOGIN-FENSTER.....	35
ABBILDUNG 20 : NAVIGATION-BAR	36
ABBILDUNG 21 : GLOCKEN-SYMBOL	36
ABBILDUNG 22 : BENACHRICHTIGUNGEN.....	36
ABBILDUNG 23 : USER-SYMBOL	37
ABBILDUNG 24 : USER-MENÜ.....	37
ABBILDUNG 25 : TASKS-SYMBOL	37
ABBILDUNG 26 : TASKS	37
ABBILDUNG 27 : KEINE DATEN VORHANDEN	38
ABBILDUNG 28 : DASHBOARD SCHÜLER	38
ABBILDUNG 29 : DASHBOARD SCHÜLER - BENACHRICHTIGUNGEN	39
ABBILDUNG 30 : DASHBOARD SCHÜLER - TIER-ZUORDNUNGEN	39
ABBILDUNG 31 : DASHBOARD SCHÜLER - NOTENDIAGRAMM	40
ABBILDUNG 32 : DASHBOARD SCHÜLER - KEINE DATEN VORHANDEN.....	40
ABBILDUNG 33 : DASHBOARD LEHRER	41
ABBILDUNG 34 : DASHBOARD LEHRER - SCHÜLER-NOTEN-DIAGRAMM	42
ABBILDUNG 35 : DASHBOARD ADMINISTRATOR.....	43
ABBILDUNG 36 : DASHBOARD ADMINISTRATOR - TIER-ZUORDNUNGEN.....	44
ABBILDUNG 37 : TESTS ANSEHEN	44
ABBILDUNG 38 : TESTS ANZEIGEN	45
ABBILDUNG 39 : TESTDETAILS - LEHRER	46
ABBILDUNG 40 : TEST BEWERTEN	46
ABBILDUNG 41 : TEST BEARBEITEN	47
ABBILDUNG 42 : TEST LÖSCHEN.....	47
ABBILDUNG 43 : TESTDETAILS LEHRER - BEWERTETER TEST.....	48
ABBILDUNG 44 : TESTDETAILS SCHÜLER - NICHT BEWERTETER TEST	48
ABBILDUNG 45 : TESTDETAILS SCHÜLER - BEWERTETER TEST.....	49
ABBILDUNG 46 : TEST HINZUFÜGEN.....	49
ABBILDUNG 47 : TEST HINZUFÜGEN - NAVIGATIONBAR	49
ABBILDUNG 48 : DATETIMEPICKER	50
ABBILDUNG 49 : SERVER POOL.....	50
ABBILDUNG 50 : IIS-MANAGER.....	51
ABBILDUNG 51 : EDIT SITE.....	51
ABBILDUNG 52 : BROWSE ANOTESA.HTL-PERG.AC.AT	52
ABBILDUNG 53 : ASP.NET ARCHITEKTUR	52
ABBILDUNG 54 : CLASS LIBRARY PROJEKT	53
ABBILDUNG 55 : ADD NEW PROJECT	54
ABBILDUNG 56 : MVC, API	54



ABBILDUNG 57 : RISIKOMATRIX	55
ABBILDUNG 58 : DATENMODELL ANOTESA.....	56
ABBILDUNG 59 : EXAMVIEWCONTROLLER	58
ABBILDUNG 60 : CODE TEST ERSTELLEN	58
ABBILDUNG 61 : CODE TEST ERSTELLEN - TIERZUORDNUNGEN UND BENACHRICHTIGUNGEN.....	58
ABBILDUNG 62 : CODE TEST BEWERTEN - NOTENZUORDNUNGEN UND BENACHRICHTIGUNGEN.....	59
ABBILDUNG 63 : CODE TIERZUORDNUNGEN LÖSCHEN	59
ABBILDUNG 64 : CODE TEST BEARBEITEN - BENACHRICHTIGUNGEN	60
ABBILDUNG 65 : JAVASCRIPT - PRÜFUNG OB SCHÜLER.....	60
ABBILDUNG 66 : AJAX - NOTIFICATIONS API.....	60
ABBILDUNG 67 : JAVASCRIPT - FUNKTION "NOData"	61
ABBILDUNG 68 : JAVASCRIPT - AUFBAU BENACHRICHTIGUNGEN DASHBOARD	61
ABBILDUNG 69 : JAVASCRIPT - AUFBAU BENACHRICHTIGUNGEN NAVIGATION-BAR	61
ABBILDUNG 70 : JAVASCRIPT - FUNKTION "DELETENOTIFICATIONS"	62
ABBILDUNG 71 : JAVASCRIPT - AUFBAU TASKS NAVIGATION-BAR	62
ABBILDUNG 72 : JAVASCRIPT - METHODE "ANIMATEUPDATE"	62
ABBILDUNG 73 : JAVASCRIPT - METHODE "UPDATEPROGRESS"	62
ABBILDUNG 74 : LDAP - ANOTESASERVICE	64
ABBILDUNG 75 : LDAP - SYSTEM.WEB.....	64
ABBILDUNG 76 : LDAP - CONNECTIONSTRING.....	64
ABBILDUNG 77 : LDAP - LOGINCONTROLLER.....	65
ABBILDUNG 78 : LDAP - AUFRUF GETUSERPROPERTY.....	65
ABBILDUNG 79 : LDAP - GETUSERPROPERTY	65
ABBILDUNG 80 : DIAGRAMM UMFRAGE 'KONZEPT'.....	69
ABBILDUNG 81 : DIAGRAMM UMFRAGE 'GEFÜHL'	70
ABBILDUNG 82 : DIAGRAMM UMFRAGE 'ÄNDERUNG'	70
ABBILDUNG 83 : TESTFALL 1 – EINLOGGEN MIT IF-USER.....	73
ABBILDUNG 84 : TESTFALL 2 – DASHBOARD FUNKTIONALITÄTEN	73
ABBILDUNG 85 : TESTFALL 3 - TEST ERSTELLEN	74
ABBILDUNG 86 : TESTFALL 4 - TEST BEARBEITEN	74
ABBILDUNG 87 : TESTFALL 5 - TEST BEWERTEN.....	75
ABBILDUNG 88 : TESTFALL 6 - TEST LÖSCHEN	75
ABBILDUNG 89 : TESTFALL 7 - KLIICK AUF BENACHRICHTIGUNGEN.....	76
ABBILDUNG 90 : TESTFALL 8 - KLIICK AUF EINEN TASK	76

12.2.3 Tabellenverzeichnis

TABELLE 1 : USE CASE LEGENDE.....	15
TABELLE 2 : USE CASE LEHRER.....	16
TABELLE 3 : USE CASE SCHÜLER.....	17
TABELLE 4 : USE CASE ADMIN	18
TABELLE 5 : IVM-MATRIX ANOTESA	20
TABELLE 6 : OPTIONEN MORRIS.JS	32
TABELLE 7 : RISIKOBEWERTUNG.....	55
TABELLE 8 : ÜBERSICHT DATENBANKTABELLEN.....	57
TABELLE 9 : LDAP-PROTOKOLL	63
TABELLE 10 : ÜBERSICHT TESTFÄLLE	72