



HTL - Perg

Höhere Abteilung für Informatik

Diplomarbeit

HTL Perg Innovative Impulse System (HIS)

**Technologische Plattform zur Erfassung und Verwaltung von gedanklichen
Impulsen als integriertes Ideenmanagement für die HTL Perg**

Projektteam: Patrick Huber
Tobias Janko
Markus Luftensteiner
Roland Neulinger

Projektbetreuer: Direktor Prof. Dipl.-Ing. Christian Reisinger

In Zusammenarbeit mit der HTL Perg

Bearbeitungszeitraum: 01.10.2016 – 05.04.2017

Eidesstattliche Erklärung

Hiermit versichern wir, die vorliegende Arbeit selbständig, 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 _____
Patrick Huber

Perg, _____ Unterschrift _____
Tobias Janko

Perg, _____ Unterschrift _____
Markus Luftensteiner

Perg, _____ Unterschrift _____
Roland Neulinger

Danksagung

Dass Sie heute diese Diplomarbeit in Händen halten, ist das Ergebnis harter Arbeit, welche sich über einige Monate hingezogen hat. Sie wäre heute nicht geschrieben, wenn wir nicht einige Unterstützer gehabt hätten. Besonders möchten wir uns bei unserem Betreuungslehrer, dem Direktor der HTL Perg, Herrn Dipl.-Ing. Christian Reisinger bedanken. Er hielt uns im Lauf der Monate stets auf Trab und stand uns bei Fragen bis hin zum Endprodukt zur Verfügung. Weiters gilt unser Dank auch unserem Programmierlehrer, Herrn Dipl.-Ing. Christian Aberger, welcher uns bei technischen Fragen jederzeit Antworten lieferte, die bei der Lösung unserer Probleme schnell halfen.

Auch möchten wir uns bei unseren Kollegen Herrn Huber Lukas bei der Unterstützung zur Formatierung unserer Diplomarbeit bzw. bei programmiertechnischen Fragen und Herrn Michael Leonhardsberger für die Unterstützung bei der Implementierung des Webservers herzlich bedanken. Für das Korrekturlesen unserer Diplomarbeit möchten wir uns bei Frau Sonja Luger, MA, MBA herzlich bedanken, da unsere Diplomarbeit sonst nicht in dieser Qualität vorliegen würde.

Kurzfassung

Inhalt der vorliegenden Diplomarbeit ist ein speziell angefertigtes System für die Ideen-, Feedback- und Kritikeinbringung innerhalb der HTL Perg. Sowohl Schüler als auch Lehrer können gedankliche Impulse veröffentlichen. Anschließend wird der Impuls auf Integrität vom Redaktionsteam überprüft. Wenn dieser als in Ordnung befunden wurde, ändert sich der Status des Impulses und er ist nun für alle Benutzer sichtbar. Diese Benutzer, welche sowohl Lehrer als auch Schüler sind, können diesen nun lesen, Kommentare abgeben, ihn zur eigenen Watchlist hinzufügen oder sich als Verantwortlicher für die Umsetzung melden. Der Verantwortliche kümmert sich nun darum, dass der Impuls weiterverfolgt wird, die Umsetzung Fortschritte macht und informiert alle Benutzer, indem er den Status entsprechend aktualisiert.

Abstract

The content of this thesis consists of a specially tailored suggestion scheme for ideas, feedback and critics for the Higher Technical College of Engineering with a specialization in Informatics located in Perg, Austria. Students as well as teachers can publish thoughts. When submitted, the content is checked by an editorial team. If an impulse passes this gate, the state is changed and every user can see the submission. These users, mostly teachers and students, can now view details for every entry, write comments, add them to their own watchlist or sign up as a responsible user for implementation of an impulse. This person is then responsible for getting the proposed idea done and informs every user by changing the status accordingly.

1 Inhaltsverzeichnis

1	Inhaltsverzeichnis.....	4
2	Einleitung	6
2.1	Motivation	6
3	Projekt.....	7
3.2	Betreuungslehrer.....	15
3.3	Auftraggeber.....	16
3.4	Geschäftsziele.....	17
3.5	Projektziele	18
4	Ausgangssituation.....	19
5	Grundlagen.....	20
5.1	Theoretische Grundlagen	20
5.2	Technische Grundlagen	25
6	Darstellung der Ideenmanagement-Plattform HIIS.....	29
6.1	Webseite als Zugang zu HIIS.....	29
6.2	Android Applikation als Zugang zu HIIS.....	39
7	Programmbeschreibung	44
7.1	Architektur.....	44
7.2	Datenmodell für HIIS	46
7.3	Implementierungen für das HIIS.....	61
8	Projektvorgehen.....	76
8.1	Planung	76
9	Verwendete Ressourcen	79
9.1	Hardware	79
9.2	Software.....	79
9.3	Architekturen.....	87

9.4	Libraries	90
9.5	Orgware	92
9.6	Projektstrukturplan	99
9.7	Verantwortungsmatrix	100
10	Resümee und Ausblick	101
10.1	Projektdurchführung mittels Kanban	101
10.2	Kommunikation	101
10.3	Aufgetretene Probleme	102
10.4	Zukünftige Erweiterungen	103
11	Glossar	104
12	Anhang	108
12.1	Aufteilung der Diplomarbeiten niederschrift	108
12.2	Literaturverzeichnis	112
12.3	Abbildungsverzeichnis	116
12.4	Tabellenverzeichnis	119

2 Einleitung

2.1 Motivation

Die HTL Perg ist nicht nur eine Schule, sondern auch ein Unternehmen. Als solches hat es auch entsprechende Strukturen und Prozesse, sowie unterstützende Tools. Es sind bereits Systeme für betriebliches Vorschlagswesen installiert, allerdings entsprechen sie nicht dem State-of-the-Art anderer Unternehmen sowie dem modernen Verständnis von Ideenmanagement. Vor dieser umsetzungsorientierten Diplomarbeit fanden sich in der HTL Perg zwei getrennte Tools, ein betriebliches Vorschlagswesen für Schülerinnen und Schüler sowie eines für Lehrerinnen und Lehrer. Zudem waren Vorschläge in den Systemen nicht nachvollziehbar und die eine Zielgruppe wusste von Vorschlägen in der anderen Gruppe nicht Bescheid. Die Motivation für diese Diplomarbeit ist die dezentrale Verwaltung von Vorschlägen aller Beteiligten in der HTL Perg in einem Tool. Vorschläge sollen für alle internen Stakeholder einsehbar und deren Umsetzungsstatus so transparent sein, dass die Wahrscheinlichkeit der Realisierung steigt, wofür in den bisherigen Systemen die Motivation fehlte und Ideen leider häufig im Sand verliefen. Mit dieser Diplomarbeit wollen wir die bestehenden Systeme ablösen, digitalisieren, zentralisieren und auf einen einheitlichen State-of-the-Art Status, was integriertes Ideenmanagement angeht, bringen.

3 Projekt

3.1.1 Roland Neulinger

Tabelle 1: Persönliche Daten von Roland Neulinger

PERSÖNLICHE DATEN

NAME	Roland Neulinger
WOHNORT	Am Bühel 9 4312 Ried in der Riedmark
GEBURTSDATUM	07.01.1998
AUFGABENGEBIET	Entwicklung der Android- Applikation
E-MAIL	roland.neulinger@gmail.com



Die fünf Jahre in der HTL-Perg haben mir gezeigt wie wichtig die Entwicklung von Software in unserer Zeit wirklich ist. Jedoch musste ich auch erkennen, dass Software meist nicht nur aus einer einfachen Anwendung für den Benutzer besteht, sondern mehr dahintersteckt als man sich zunächst vorstellt. Neben den technischen Fächern hatte ich auch an kaufmännischen Spezialisierungen wie Betriebswirtschaft und Management Freude. In meinen Praktika durfte ich einerseits erste Eindrücke vom Arbeitsleben, andererseits zusätzliches Fachwissen erlangen. Ich plane nach der Schule mein theoretisches Wissen in der Praxis anzuwenden indem ich meinen Berufsweg als Softwareentwickler starte.

Erlernte Programmiersprachen/Technologien:

Tabelle 2: Kompetenzen von Roland Neulinger

PROGRAMMIERSPRACHEN	TECHNOLOGIEN
Java	Android-Entwicklung
C	Swift
C#	MySQL
Assembler	
HTML	
Angular2	
PHP	
JavaScript	
TypeScript	

Tabelle 3: Schullaufbahn von Roland Neulinger

SCHULLAUFBAHN	
Volksschule Piberschlag	2004
Volksschule Ried in der Riedmark	2005 – 2008
Hauptschule Ried in der Riedmark	2008 – 2012
Höhere Technische Bundeslehranstalt Perg	2012 – 2017
Praktikum in der Brauunion AG	Sommerferien 2016
Praktikum im Land Oberösterreich	Sommerferien 2017

3.1.2 Patrick Huber

Tabelle 4: Persönliche Daten von Patrick Huber

PERSÖNLICHE DATEN

NAME	Patrick Huber
WOHNORT	Klam 50 4352 Klam
GEBURTSDATUM	14.08.1998
AUFGABENGEBIET	REST Interface
E-MAIL	patrickhuber9@gmail.com



In den fünf Jahren, die ich in der HTL-Perg verbrachte, habe ich meinen Wissensstand vor allem im Bezug zur digitalen Welt stark verbessert. Aber auch abseits wurden uns viele wissenswerte Dinge und Kompetenzen gelehrt. Dies geschah nicht nur im Unterricht sondern auch in den Pflichtpraktika welche wir absolvieren mussten.

Ich absolvierte mein betriebswirtschaftliches Praktikum bei der Poschacher Baustoffhandel GmbH. Mein technisches Praktikum absolvierte ich bei der Fachhochschule Kufstein Tirol-Privatstiftung. In beiden Praktika konnte ich mit schneller Aufgabenerfassung und strukturierter Aufgabenabarbeitung punkten.

Nach der HTL werde ich in die Berufswelt einsteigen und dort die Erfüllung meines Lebens suchen.

Erlernte Programmiersprachen/Technologien:

Tabelle 5: Kompetenzen von Patrick Huber

PROGRAMMIERSPRACHEN	TECHNOLOGIEN
Java	Android-Entwicklung
C	Swift (iOS-Entwicklung)
C#	MySQL
Assembler	SAP
HTML	
Angular2	
PHP	
JavaScript	
TypeScript	
Python	
Lua	
XML	
SQL	

Tabelle 6: Schullaufbahn von Patrick Huber

SCHULLAUFBAHN	
Volksschule Klam	2004 – 2008
Hauptschule Saxen	2008 – 2012
Höhere Technische Bundeslehranstalt Perg	2012 – 2017
Praktikum für Softwareentwicklung	Sommerferien 2015
Praktikum für Buchhaltung/BWM	Sommerferien 2014

3.1.3 Tobias Janko

Tabelle 7: Persönliche Daten von Tobias Janko

PERSÖNLICHE DATEN	
NAME	Tobias Janko
WOHNORT	Bodendorf 185 4223 Katsdorf
GEBURTSDATUM	07.07.1997
AUFGABENGEBIET	Webseite: beinhaltet die Benutzersicht und die Redakteur-Ansicht
E-MAIL	tobiasjanko@yahoo.de


 A professional portrait of Tobias Janko, a young man with short brown hair and a light beard, wearing a dark suit jacket, a white shirt, and a dark bow tie. He is smiling slightly and looking towards the camera.

Während meiner Zeit in der HTL Perg habe ich sehr viel Erfahrung sammeln können was Softwareentwicklung und Netzwerkadministration betrifft. Durch die Praktikumswochen, die wir im Laufe der ersten vier Jahre absolvierten, war es mir möglich einiges vom Gelernten auch direkt in die Tat umzusetzen. In diesen Praktika konnte ich außerdem viel Erfahrung sammeln und lernen wie viel Flexibilität es braucht, wenn man im Job unmittelbar als Softwareentwickler tätig ist.

In unserer Schule ist der zweite Schwerpunkt Rechnungswesen und betriebliche Organisation. Auch in diesem Bereich wurde Praxiserfahrung gefordert, um in den fünften Jahrgang aufsteigen zu können. Ich habe mich persönlich im Laufe dieser fünf Jahre in den Bereichen Webentwicklung und Netzwerktechnik spezialisiert, was dann auch mein geplanter Berufseinstieg nach der Matura werden soll.

Tabelle 8: Kompetenzen von Tobias Janko

ERLERNTE PROGRAMMIERSPRACHEN/TECHNOLOGIEN

C	Ubuntu
C#	MySQL
Assembler	Angular2
HTML	Bash scripting
PHP	
JavaScript	
TypeScript	
Android	
Swift	

Tabelle 9: Schullaufbahn von Tobias Janko

SCHULLAUFBAHN

Volksschule Hochstraß	Oktober 2003 – Juli 2008
Hauptschule Ried in der Riedmark	Oktober 2008 – Juli 2012
HTL Perg	Oktober 2012 – Juli 2017
Praktikum für Programmieren E+E Elektronik	Sommerferien 2014
Praktikum für Rechnungswesen E+E Elektronik	Sommerferien 2015

3.1.4 Markus Luftensteiner

Tabelle 10: Persönliche Daten von Markus Luftensteiner

PERSÖNLICHE DATEN

NAME	Markus Luftensteiner
WOHNORT	Kolbing 16 4342 Baumgartenberg
GEBURTSDATUM	07.01.1998
AUFGABENGEBIET	Datenbank mit Stored Procedures und Testdaten
E-MAIL	luftensteiner98@gmail.com



Meine HTL-Zeit hat mich ideal auf die Anforderungen in der Berufswelt vorbereitet. Egal ob Programmieren und Softwareentwicklung, Betriebswirtschaft und Management oder Netzwerktechnik und verteilte Systeme, ich habe in allen Bereichen an Können und Wissen dazugewonnen. Meine Pflichtpraktika habe ich bei der Firma Robert Bosch AG und der ÖBB-Personenverkehr AG verrichtet und konnte aufgrund meines breiten Wissens und meiner zügigen und gewissenhaften Arbeitsweise bei meinen Arbeitskollegen und Vorgesetzten punkten. Bei Bosch war ich in der internen Tool-Entwicklung tätig und konnte unmittelbar an Entwicklungs- und Managementtools arbeiten, die dort zum Einsatz kommen. Bei den ÖBB war ich im IT-Projektcontrolling tätig und habe erkannt, wie wichtig Projektmanagement und Umsetzungsdisziplin in der Praxis ist. Derzeit sondiere ich die Studienangebote und interessiere mich für spezialisierte Informatik-Studiengänge in Wien.

Erlernte Programmiersprachen/Technologien:

Tabelle 11: Kompetenzen von Markus Luftensteiner


PROGRAMMIERSPRACHEN	TECHNOLOGIEN
Java	Android-Entwicklung
C	Swift (iOS-Entwicklung)
C#	MySQL
Assembler	WildFly
HTML	Arduino
Angular2	Banana Pi
PHP	
JavaScript	
TypeScript	
Python	
Perl	

Tabelle 12: Schullaufbahn von Markus Luftensteiner

SCHULLAUFBAHN	
Volksschule Baumgartenberg	2004 – 2008
Hauptschule Baumgartenberg	2008 – 2012
Höhere Technische Bundeslehranstalt Perg	2012 – 2017
Praktikum für Softwareentwicklung	Sommerferien 2014
Praktikum für Buchhaltung/BWM	Sommerferien 2015

3.2 Betreuungslehrer

Tabelle 13: Beschreibung von Direktor Reisinger (htl-perg.ac.at, 2017)

NAME:	PROFESSOR DIPL.-ING. CHRISTIAN REISINGER	
E-MAIL:	c.reisinger@htl-perg.ac.at	
GEGENSTÄNDE:	Betriebliche Anwendungen der DV (Übungen) (2AFID), Programmieren (5BHIF), Systemplanung und Projektentwicklung (3AHIF, 4AHIF, 5AHIF), Systemplanung und Projektentwicklung (Übungen) (5AHIF)	<p><i>Abbildung 1: Portraitbild von Direktor Reisinger (htl-perg.ac.at, 2017)</i></p>
WERDEGANG:	seit 2002: Lehrer an HTL Perg 1999 - 2007: CSC Austria / Projektmanager Projekt MVB 1995 - 1999: Qualitätsmanagementbeauftragter (ISO9001) (Servo Data, CSC Austria) 1986 - 1999: Servo Data / Programmier-, Analytiker- und Projektleitungstätigkeit in EDV-Projekten im Gesundheits- und Bankenbereich 1983 - 1988: Studium der Informatik an der TU Wien	
UNTERRICHT AN ANDEREN SCHULEN:	NMS Pabneukirchen	
WEITERE TÄTIGKEITEN:	Direktor an der HTL Perg Wettbewerbskoordination an der HTL Perg Erasmus+ Koordinator (internationale Schulprojekte und Schüleraustausch)	

3.3 Auftraggeber

3.3.1 HTL Perg

Kontakt

HTBLA Perg
Machlandstraße 48
4320 Perg
☎ 0 72 62 539 26



Abbildung 2: Logo der HTL Perg

Die Höhere Technische Bundeslehranstalt Perg ist darauf spezialisiert, die Schülerinnen und Schüler bereits während der Schulzeit auf das zukünftige Arbeitsfeld vorzubereiten. Dabei wird vor allem Augenmerk auf Informatik im Business-Umfeld gelegt. Als unmittelbare Kunden werden Schülerinnen und Schüler verstanden, die mindestens die Unterstufe (z.B. Neue Mittelschule, Gymnasium) erfolgreich absolviert haben. Je nach Ausbildungsdauer und Abschlussreife gelten an der HTL Perg unterschiedliche Aufnahmekriterien. Die Absolventinnen und Absolventen stehen unmittelbar nach der Ausbildungszeit als qualifizierte Fachkräfte dem Arbeitsmarkt zur Verfügung. Folgende zwei Optionen bietet diese praxisorientierte, berufsbildende höhere Schule an:

- **Höhere Abteilung für Informatik**

5-jährige Ausbildung mit Diplom- und Reifeprüfung mit praxisbezogenen Aspekten:

- **Schulprojekt**, das während des Projektunterrichts von 2-6 Schülerinnen bzw. Schülern innerhalb von 2 Jahren bearbeitet wird.
- **Ferialpraktika** von mindestens 8 Wochen bis zum Beginn des 5. Jahrgangs
- **Diplomarbeit**, die von 2-5 Schülerinnen bzw. Schülern außerhalb der Schulzeit betreut innerhalb eines Jahres mit mindestens 200 Stunden pro Teilnehmer vor der Graduierung einzureichen ist.

- **Fachschule für Informationstechnik**

4-jährig ohne Reifeprüfung mit praxisbezogenen Aspekten:

- **Ferial-/Betriebspraktikum:**
4 Wochen bis zum 4. Jahrgang | 15 Wochen im 4. Jahrgang bei einem einschlägigen Betrieb
- **Abschlussarbeit**, die von 2-5 Schülerinnen bzw. Schülern außerhalb der Schulzeit betreut innerhalb eines Jahres mit mindestens 80 Stunden pro Teilnehmer vor dem Abschluss einzureichen ist.

3.4 Geschäftsziele

Mit dem Projekt für diese Diplomarbeit möchte die HTL Perg ihren Umgang mit Ideen, Anregungen und Kritik im Sinne des unternehmerischen Human Resource Managements vor allem für Lehrerinnen und Lehrer sowie Schülerinnen und Schüler verbessern. Gerade an einer Schule, die praxisnahe Informatik im Business-Umfeld als ihre Mission ansieht, möchte vorgelebt werden, was auch unterrichtet wird:

- **Quantität der Impulse steigern**

Mit dem Projektergebnis wird sichergestellt, dass Lehrer und Schüler Ideen und Feedback abgeben und mit einer unkomplizierten, intuitiven, leicht verfügbaren Meldemöglichkeit die Anzahl der bekannt gegebenen Impulse steigt.

- **Qualität der Mindestinhalte und im Umgang mit Impulsen**

Durch die digitale Erfassung können Formulare für die Erfassung von Ideen und Feedback angelegt und Mindestangaben bei Bedarf festgelegt werden. Die Plattform soll prozessorientiert die Bearbeitung von Meldungen abbilden und durch ein Quality-Gate in Form eines Redaktionsteams die rasche Erstbewertung sicherstellen.

- **Nachvollziehbare Prozessdurchläufe sicherstellen**

Durch das Produkt wird die Koordination von Anregungen, Ideen und Kritik verbessert, egal ob sie von Lehrern oder Schülern kommt. Die Einmelder sollen jederzeit Einsicht in den Bearbeitungsstatus zu eigenen Impulsen haben.

- **Feedback sicherstellen und Umsatzquote steigern**

Unternehmerisches Ziel mit diesem Projektauftrag ist es jeder Impulsgeberin und jedem Impulsgeber jedenfalls Feedback zu Einmeldungen zu geben und natürlich die Umsetzung von Einmeldungen zu steigern. Damit bleibt die Motivation zur Teilnahme bei Lehrerinnen und Lehrern sowie Schülerinnen und Schülern erhalten.

- **Positiver Beitrag zum Employer Branding**

Mitarbeiter arbeiten gerne für Unternehmen, indem ihre Ideen und Kritik ernst genommen und behandelt wird. Darüber wird auch im Bekanntenkreis berichtet. Mit einem modernen integrierten Ideenmanagement steigt die Attraktivität der HTL Perg als Arbeitgeber bei Lehrerinnen und Lehrern als auch bei potenziellen Kundinnen und Kunden.

3.5 Projektziele

Für dieses konkrete Projekt zur Bereitstellung einer technologischen Plattform für ein integriertes Ideenmanagement für die wesentlichen internen Stakeholder der HTL Perg wurden folgende Projektziele besonders betont:

- **Termintreue**

Das Projektteam hält sich an die in den Projektplänen vereinbarten Termine, damit die rechtzeitige Übergabe und Inbetriebnahme der digitalen Plattform für ein integriertes Ideenmanagement nicht gefährdet wird.

- **Einhalten des Vorgehensmodelles**

Das in den Projektplänen definierte Vorgehensmodell wird vom Projektteam eingehalten, damit niemand Gefahr läuft das vereinbarte Ziel und den Projektscope aus den Augen zu verlieren.

- **Vereinbarte Abstimmungen einhalten**

Wir tun was wir sagen und wir sagen was wir tun. Sowohl Abstimmungen innerhalb des Projektteams als auch Vereinbarungen mit extern Beteiligten sind einzuhalten.

4 Ausgangssituation

Der Auftraggeber dieses Projektes für die Diplomarbeit, der Direktor der HTL Perg, ist mit dem Status-Quo zum betrieblichen Vorschlagswesen an seiner Schule unzufrieden und ist daher auf der Suche nach einer zeitgemäßen und zukunftsfähigen Ersatzlösung. Die Unzufriedenheit ergibt sich einerseits aus der Tatsache, dass es zwei getrennte, nicht vertrauliche Vorschlagswesen für Lehrerinnen und Lehrer sowie Schülerinnen und Schüler gibt als andererseits aus dem Mangel an systematischer, zeitnaher Behandlung und fehlender Feedback-Möglichkeit dieser analogen Systeme.

Für Lehrerinnen und Lehrer ist die Situation besonders unzufriedenstellend, da sie Feedback in Form von Zetteln auf einer Pinnwand einmelden können. Nicht nur, dass Zettel verloren gehen können, sie hängen auch unsystematisch über- und untereinander, was eine Abarbeitung erschwert und dem Lehrpersonal vor Augen führt wie unbemerkt die eigene Meldung eventuell bleibt. Wenn Zettel anonym aufgehängt werden kann auch kein Kontakt zum Einmelder aufgenommen werden falls es Rückfragen gibt oder eine begründete Ablehnung eines Vorschlags. Das wirkt sich natürlich negativ auf die Motivation aus künftig weitere Ideen, Anregungen oder Feedback in dieser Form einzubringen.

Für Ideen, Anregungen und Feedback von Schülerinnen und Schülern finden diese einen Briefkasten in der Garderobe. Dieses System hat im Wesentlichen die oben genannten Nachteile mit dem Unterschied, dass Zettel in der Box weniger wahrscheinlich verloren gehen.

Für die Direktion hat die Trennung in zwei Systeme für betriebliches Vorschlagswesen auch wesentliche Nachteile. Zunächst hängt der Bearbeitungsstart vom Zeitpunkt der Einsammlung von der Pinnwand und aus dem Briefkasten ab, dann gibt es keine Rückfragemöglichkeit, wenn die Einmeldung anonym stattfindet und zeitkritische Meldungen werden wahrscheinlich nicht fristgerecht erledigt. Der Bearbeitungsstatus lässt sich nur schwer nachvollziehen, wenn Zettel einem Ansprechpartner für die weitere Verfolgung übergeben wurden und zu Feedback an den Impulsgeber kommt es nur in den seltensten Fällen.

5 Grundlagen

5.1 Theoretische Grundlagen

Zum Verständnis dieser Diplomarbeit müssen zunächst folgende Begriffe eindeutig definiert werden.

Definition **betriebliches Vorschlagswesen**

„Ein Verbesserungsvorschlag ist eine eingereichte Anregung zur Verbesserung eines Zustandes (einer Arbeit, eines Materials, eines Werkzeuges, einer Organisation usw.) mit dem Ziel der Erhöhung der Wirtschaftlichkeit, Sicherheit, Zuverlässigkeit, Qualität, Umwelt, des Images usw. Der Verbesserungsvorschlag muss einen Lösungsansatz beinhalten, jedoch braucht er keine ausgearbeitete Lösung darzustellen.“ (Bronner, 2003, 97)

Definition **kontinuierlicher Verbesserungsprozess (KVP) bzw. KAIZEN**

„Das japanische Managementkonzept unterstützt Organisationen auf dem Weg (Kai) zum Guten (Zen) durch laufende Produkt- und Prozessverbesserungen. [...] Die Einbindung der Kreativität der Mitarbeiter zur Sicherung des Unternehmenserfolgs ist die Intention des KVP. Dies kann beinhalten, den Anteil der Wertschöpfung in allen Prozessen zu erhöhen und Verschwendungen zu minimieren, aber auch die Motivation und Leistungsfähigkeit der Mitarbeiter zu steigern. Eine konsequente Präsenz der Führungskräfte vor Ort an den Arbeitsplätzen zur Aktivierung, Koordination, Stabilisierung und Schaffung von Nachhaltigkeit des Verbesserungsprozesses wird benötigt.“ (Institut für angewandte Arbeitswissenschaft e. V., 2016, 42)

Definition **Qualitätszirkel**

Für diesen Begriff gibt es zwei relevante Definitionsquellen aus dem japanischen und deutschen Markt, wobei wir uns der gängigen Begriffssammlung im deutschsprachigen Raum anschließen und wir verstehen unter einem Qualitätszirkel eine kleine Gruppe von Mitarbeiterinnen und Mitarbeitern der unteren

Hierarchieebenen, die sich regelmäßig auf freiwilliger Grundlage treffen, um Probleme aus ihrem Arbeitsbereich zu bearbeiten. (vgl. Cuhls, 1993, 11)

Definition (interne) **Stakeholder**

Der Stakeholder-Ansatz geht davon aus, dass Unternehmen grundsätzlich mit vielfältigen Akteuren in Beziehung stehen, die ihrerseits Ansprüche (englisch: „stake“) an das Unternehmensverhalten stellen bzw. im weitesten Sinne ein Interesse am Unternehmensgeschehen haben. [...] Eigentümer, Mitarbeiterinnen und Mitarbeiter sowie Management werden als interne Stakeholder bezeichnet. (vgl. Riede, 2011, 25f)

Definition **integriertes Ideenmanagement**

Das integrierte Ideenmanagement beschäftigt sich mit Inputs aller internen Stakeholder vom Zeitpunkt der Entstehung der Idee über Feedback bis zu ihrer Realisierung in einer nachvollziehbaren Umgebung. Zentraler Punkt zu Beginn dieses Prozesses ist die Frage, wie geeignete Rahmenbedingungen geschaffen werden können, die die Kreativität und Teilnahmereitschaft der Beteiligten ansprechen und fördern und für einen offenen Ideenfluss im Unternehmen sorgen. Auch im weiteren Ablauf ist es entscheidend, wie die Kommunikation zwischen Einreicher und Beurteiler gestaltet wird. (vgl. Läge, 2002, 31)

Der Begriff Ideenmanagement wurde in den 1970er Jahren geprägt. Unter diesem Oberbegriff werden in der Praxis das betriebliche Vorschlagswesen, der kontinuierliche Verbesserungsprozess und Qualitätszirkel zusammengefasst und somit ist die Vorstufe zum Innovationsmanagement im Unternehmen etabliert. (vgl. Läge, 2002, 29)

Weitere Unterstützung kamen auf dem Weg zum integrierten Ideenmanagement in den 1980er Jahren vor allem aus Asien. Qualitätszirkel und kontinuierlicher Verbesserungsprozess fanden Einzug in westliche Unternehmen, wobei der Bezug zum bereits etablierten betrieblichen Vorschlagswesen noch gefunden werden musste. Heute ist das Verständnis zu integrierten Managementmethoden und Managementsystemen so gereift, dass die Vereinigung von Vorschlagswesen, Verbesserungsprozessen und

Qualitätszirkeln zu einem integrierten Ideenmanagement State-of-the-Art geworden ist. (vgl. Schat, 2017, 141)

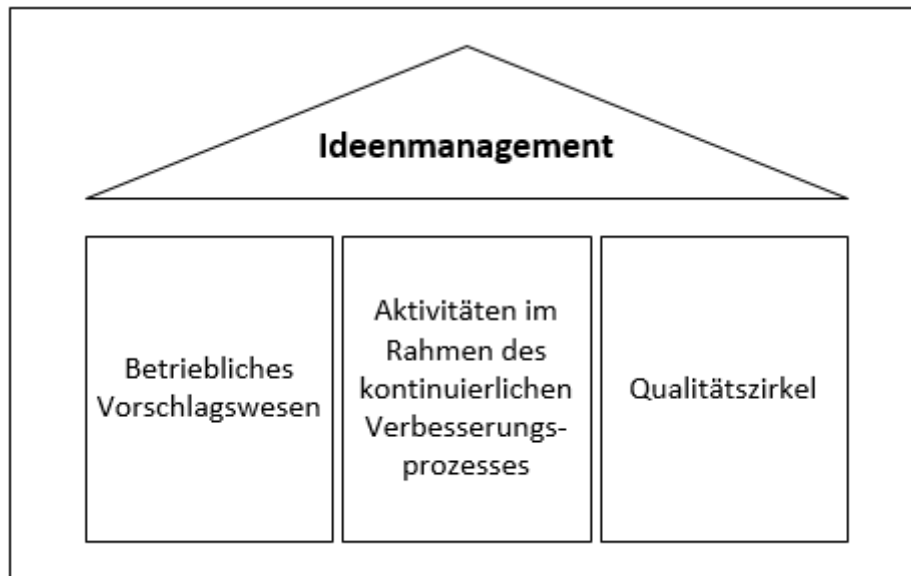


Abbildung 3: Säulen des Ideenmanagements (vgl. Läge, 2002, 30)

Wie auch in der HTL Perg bisher, so werden auch beim klassischen betrieblichen Vorschlagswesen Ideen von Mitarbeitern eher zufällig eingereicht, die Führungskräfte und interne Kommunikation haben eher wenig damit zu tun. Beim Ideenmanagement hingegen handelt es sich um eine Managementfunktion, bei der über Mittel der internen Kommunikation (z.B. Kampagnenmanagement) Mitarbeiter motiviert werden Ideen einzubringen und Führungskräfte für eine Prüfung der eingereichten Ideen sowie für Feedback zu bzw. Umsetzung der Mitarbeiterideen sorgen. Führungskräfte und Tools unterstützen das Einreichen von Verbesserungsvorschlägen um aktiv Veränderungen im Unternehmen anzustoßen. (vgl. Läge, 2002, 30)

Das integrierte Ideenmanagement wird sowohl organisatorisch, inhaltlich als auch strukturell verstanden. Zu letzterem ist wichtig, dass alle Mitarbeiter einer Organisation, egal ob Führungskraft oder Schüler, in ein Ideenmanagementsystem involviert wird. Was die beiden ersten Punkte betrifft hilft die folgende Abbildung der Veranschaulichung der organisatorischen und inhaltlichen Dimensionen des integrierten Ideenmanagements. Dabei handelt es sich um eine Schnittmenge aus dem Organisationsbereich Human Resource Management (Personalwesen), weil integriertes Ideenmanagement vorrangig nach

erfolgreicher Personalführung strebt, sowie den inhaltlichen Aufgaben des klassischen Qualitätsmanagements und den ersten Phasen des Innovationsmanagements. (vgl. Läge, 2002, 30f)

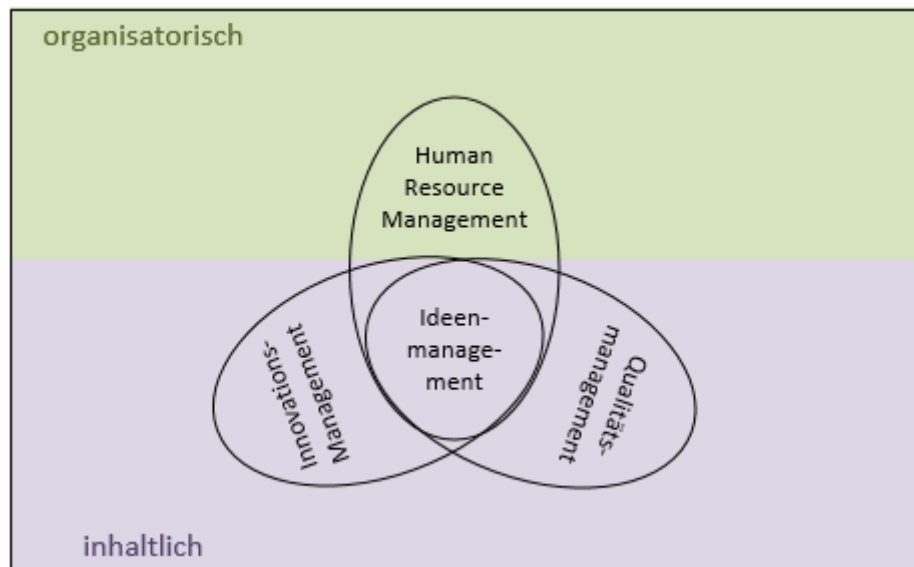


Abbildung 4: Bereiche des Ideenmanagements (vgl. Läge, 2002, 31)

5.1.1 Grundaufbau des neuen Ideenmanagementsystems der HTL Perg

Das zentrale Element im neuen HTL Perg Innovative Impulse System (HIIS), wie die Plattform für das integrierte Ideenmanagementsystem von nun an heißen soll, sind so genannte **Impulse**. Mit diesem Begriff werden alle Arten von Einmeldungen bezeichnet, ganz gleich ob Ideen, Anregungen, Kritik, Feedback, Weiterentwicklungs- oder Innovationsansätze. Damit ergeben sich Chancen im kontinuierlichen Verbesserungsprozess bis hin zu evolutionären und revolutionären Innovationen, die das Unternehmens- oder Lehrsystem der HTL Perg nachhaltig ändern könnten.

Zur Erfassung und Verwaltung von Impulsen braucht es natürlich **Benutzer**. Dabei handelt es sich in erster Linie um ordentliche, also derzeit aktive, Lehrerinnen und Lehrer sowie Schülerinnen und Schüler. Sie legen Impulse an und können den Status dazu in dieser Plattform nachvollziehen. Definierte Benutzer, die von der Direktion bestimmt werden, agieren in Form des **Redaktionsteams** als Gutachter bzw. (Erst-)Beurteiler und somit als erstes wesentliches Gate im Prozess nach der Einmeldung eines Impulses. Ein Mitglied des

Redaktionsteams muss einen Impuls freigeben, damit er im Prozess weiterkommt und im System für die Bearbeitung bzw. Weiterverfolgung angelegt wird. Das Redaktionsteam muss natürlich organisatorisch von der Direktion der HTL Perg ermächtigt sein nicht sinnvolle Impulse abzulehnen, wozu ein gemeinsames Verständnis von erwünschten Inhalten geschaffen werden muss.

Impulse, die vom Redaktionsteam für die Weiterverfolgung freigegeben wurden, sind für alle Benutzer in der **Vorschlagsliste** zugänglich. Dort können Impulse von anderen Benutzern kommentiert und mit Dateien ergänzt werden. Benutzer können sich hier auch Impulse zur persönlichen Watchlist hinzufügen und sich als Verantwortliche für die Weiterverfolgung bzw. Umsetzung deklarieren.

Impulse, Benutzer, Redaktionsteam und Watchlist sind demnach die wesentlichen Elemente in der digitalen Plattform für das integrierte Ideenmanagement der HTL Perg. Es zielt darauf ab so viele sinnvolle Impulse wie möglich zu erfassen, ihre Weiterverfolgung zu tracken und so viele wie möglich umzusetzen. Der unmittelbare Nutzen ist die Optimierung und Attraktivitätssteigerung der Arbeitsumwelt und -inhalte für das Lehrpersonal und die Schülerinnen und Schüler. Mittelbar besteht der Nutzen aber auch in der Motivationssteigerung die bei allen Beteiligten entsteht, wenn sie kreative Impulse einbringen können, die vielleicht nicht unmittelbar in den Kontext passen und Feedback dazu sichergestellt ist. Ob sich die neue Plattform in der Praxis bewährt und von allen Beteiligten akzeptiert wird, muss noch bewiesen werden. Vereinbarungsgemäß wurde die Inbetriebnahme von diesem Projekt ausgeschlossen, da sich die Fertigstellung der Diplomarbeit sonst verzögert hätte.

5.2 Technische Grundlagen

5.2.1 Datenbanken

Von den vielen Datenbank-Arten, die es gibt, werden hier einleitend drei näher betrachtet, die für das Projekt zu dieser Diplomarbeit besonders relevant hätten sein können: relationale, objektorientierte und graphenorientierte Datenbanken. Diese drei Datenbank-Arten unterscheiden sich in der Art und Weise wie bestimmte Daten gespeichert werden.

Relationale Datenbankmanagementsysteme haben den Vorteil, dass die Daten untereinander zu jedem Zeitpunkt aktuell und abgeglichen sind. Da bei uns diese Eigenschaft unbedingt gegeben sein muss, weil ansonsten eventuell Kommentare oder Dateien verloren gehen könnten, haben wir uns für den Einsatz einer relationalen Datenbank für die Implementierung eines integrierten Ideenmanagementsystems entschieden. Bei den anderen beiden Datenbank-Optionen sehen wir das unter Umständen gefährdet.

Bei objektorientierten Datenbanken hingegen ist die Konsistenz nicht von Interesse, da hier einfach nur Programmobjekte abgelegt werden und wieder von demselben Programm ausgelesen werden können, wenn diese wissen, wo genau die gesuchten Objekte liegen. Zu guter Letzt haben wir als Option die Graphendatenbanken ausgeschlossen, bei denen Daten, die in einem Objekt abgelegt sind, mittels eines sogenannten Pointers (Zeigers) referenziert sind. Das bringt enorme Geschwindigkeitsvorteile bei wachsender Datenmenge, was für unser System aber nicht zwingend erforderlich ist.

5.2.2 Mobile Applikationen

Was den Erfolg guter und häufig genutzter mobiler Anwendungen ausmacht, ist deren Einfachheit und zuverlässige Funktionalität. In diesem Sinne hochwertige Apps verbreiten sich über App-Stores bzw. Weiterempfehlung auch millionenfach weltweit. Mit der Smartphone-Verbreitung sind schnell zahlreiche Abnehmer für schnell programmierte Anwendungen des täglichen Bedarfs gefunden.

Im Wesentlichen können zwei Arten von mobilen Applikationen unterschieden werden:

- **Native Apps**

Native Apps werden speziell für ein Betriebssystem entwickelt. Deshalb laufen diese ausschließlich auf den Geräten, für die sie entwickelt wurden. Einer der größten Vorteile nativer Apps ist, dass alle Schnittstellen zur Hardware einheitlich funktionieren und die Ressourcen des Gerätes optimal genutzt werden.

- **Web Apps**

Eine Web App ist eine speziell programmierte HTML5 Webseite, die das Endgerät erkennt und den Inhalt optimiert dafür darstellt. Solche Anwendungen müssen nicht aus dem App-Store geladen werden sondern werden über einen Webbrowser abgerufen.

VORTEILE NATIVER APPS	VORTEILE WEB APPS
Sind für ein bestimmtes Betriebssystem optimiert	Funktionieren mit allen Betriebssystemen und auf allen Smartphones
Gerätehardware kann nur von nativen Apps genutzt werden	HTML5 ermöglicht die Offlinespeicherung von Daten
Zugriff auf bspw. Sensoren im Gerät ist möglich	Ist kostengünstiger, da es eine Anwendung für alle Betriebssysteme ist
Performance der Anwendung ist besser am jeweiligen Gerät	Weniger Abhängigkeit der einzelnen Programmiersprachen der Betriebssysteme

Abbildung 5: Vorteile Nativer und Web Apps

Wir haben uns bei der Umsetzung der mobilen Anwendung für HIIS für eine native App entschieden, weil diese:

- besser optimierbar ist als Web Apps,
- die Installation über den App-Store einfacher von der Hand geht als das Abrufen einer Web-App über einen mobilen Browser und
- wir Erfahrung im Umgang mit der für die native App erforderliche Programmiersprache aus dem Unterricht haben.

5.2.2.1 Marktdominierende native App-Plattformen

PLATTFORM	PROGRAMMIERSPRACHE	ENTWICKLUNGSUMGEBUNG
Android	Java / Android	Android Studio
iOS	Objective-C / Swift	xCode
Windows Phone	C++ / C#	Visual Studio

Abbildung 6: Marktdominierende Plattformen der nativen App-Entwicklung

Im Rahmen dieser Diplomarbeit können wir HIIS nicht für alle Plattformen entwickeln, deshalb haben wir anhand der Marktpräsenz hergeleitet, für welche Plattform die erste Umsetzung am meisten Sinn macht. Unser Ziel ist es so viele Benutzer wie möglich zu erreichen und somit analysieren wir in der folgenden Abbildung den Marktanteil der oben erwähnten Plattformen bei den Gerätebesitzern und somit potenziellen App-Anwendern.

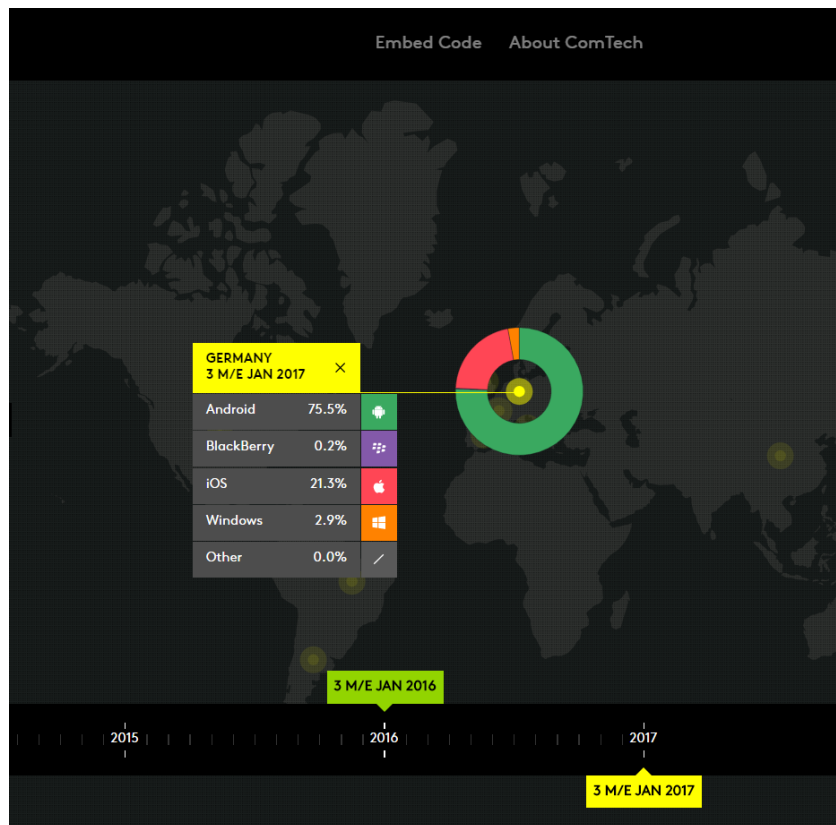


Abbildung 7: Marktanteil der verschiedenen Plattformen in Deutschland (www.kantarworldpanel.com, 2017)

Anhand der Marktanalyse wird deutlich, dass derzeit der Großteil (in etwa drei Viertel der am deutschen Markt eingesetzten Endgeräte) auf Android basiert. Daher haben wir uns dazu entschieden HIIS in Abstimmung mit unserem Projektauftraggeber in einem ersten Schritt für Android als mobile Anwendung umzusetzen.

6 Darstellung der Ideenmanagement-Plattform HIIS

6.1 Webseite als Zugang zu HIIS

Da die Vorteile eines webbasierten Zugangs zur integrierten Ideenmanagement-Plattform eindeutig überwiegen und die diesbezügliche Verfügbarkeit ausreicht, weil es sich um kein kerngeschäfts- bzw. sicherheitsrelevantes System handelt, werden Alternativen aus dem Offline-Bereich nicht näher analysiert und überlegt.

6.1.1 Funktion der Webseite für HIIS

Die Funktion der Webseite für das integrierte Ideenmanagement liegt in der Ansicht für Benutzer und in jener für Redakteure. In unserem Fall gibt es einen Redakteur, der zugleich auch Administrator des Systems ist. Es muss immer mindestens einen Redakteur geben, der dann wiederum andere Benutzer bestimmen kann, die dann ebenfalls als Redakteur fungieren können.

Ein Benutzer kann Impulse anlegen und sich für Impulse als Verantwortlicher eintragen. Ein Verantwortlicher ist ein Benutzer, der die Weiterverfolgung eines Impulses federführend übernimmt und versucht diesen bestmöglich umzusetzen. Außerdem ist es dem Benutzer möglich Kommentare zu einem Impuls zu verfassen und Dateien dazu anzuhängen.

Für den Redakteur besteht die Möglichkeit einen Impuls zu archivieren oder zu löschen, wenn die darin enthaltene Idee gegen interne Vorgaben und Richtlinien verstößt. Der Redakteur kann außerdem andere Benutzer zu Redakteuren ernennen, welcher dann auch dieselben Rechte und Funktionen hat.

Die Hauptfunktionen der Webseite sind:

- Eine **Übersicht** über **alle Impulse**
 - Die Auflistung erfolgt standardmäßig nach der Beliebtheit der eingegangenen Impulse
- **Detailansicht** mit **Kommentarfunktion** zu einzelnen Impulsen

- **Hinzufügen** eines Impulses
- **Löschen** eines Impulses
- **Bearbeiten** eines Impulses
- **Eigene Watchlist** (Favoritenlisten) **bearbeiten**

6.1.2 Ablauf auf der Webseite

Egal ob aus der Sicht eines Benutzers mit oder ohne Redakteursrechten, alle Benutzer finden den gleichen Ablauf auf der Online-Plattform des neuen integrierten Ideenmanagements HIIIS vor. Nach dem Login, was den persönlichen Datenschutz sicherstellt, entscheidet man sich nach der Home Seite für das Hinzufügen, Bearbeiten oder die Detailansicht eines Impulses oder die Übersicht in der eigenen Watchlist, in der auch in die Detailansicht der dort gelisteten Impulse gewechselt werden kann.

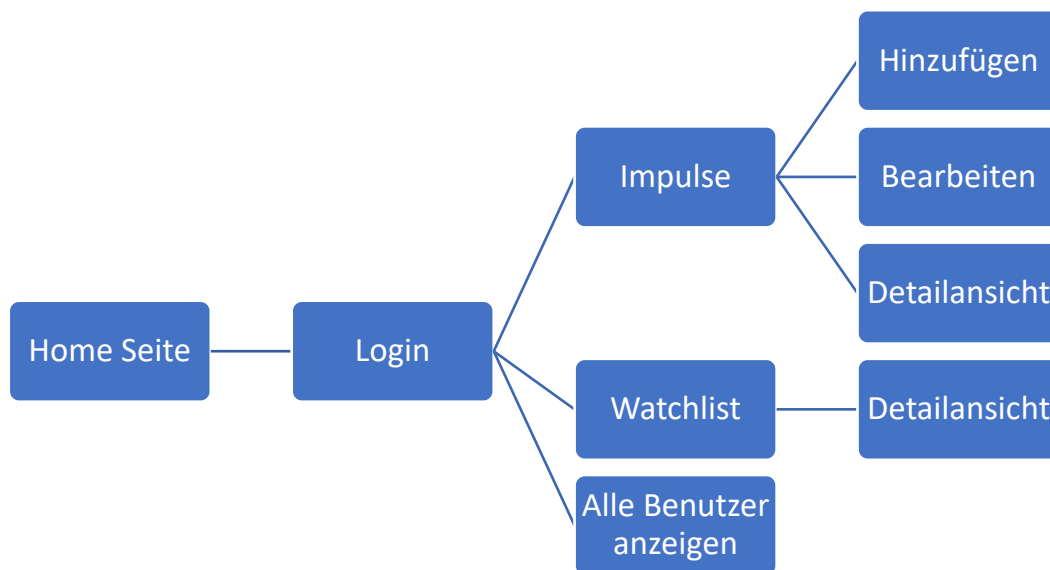


Abbildung 8: Ablauf der Webseite

6.1.3 Anwender der Webseite

Die Benutzer der Webseite sind sowohl die Schülerinnen und Schüler und auch die Lehrerinnen und Lehrer. Alle können sich mit ihrem persönlichen Schul-Login (Active Directory-Account) anmelden, somit muss kein separater Account erstellt werden um diese Plattform nutzen zu können. Grundsätzlich stehen dann allen Benutzern die gleichen

Funktionalitäten zur Verfügung, nur das Archivieren bzw. Löschen bereits eingereichter Impulse ist Redakteuren vorbehalten.

6.1.4 Funktionalitäten der Webseite

6.1.4.1 Home Seite

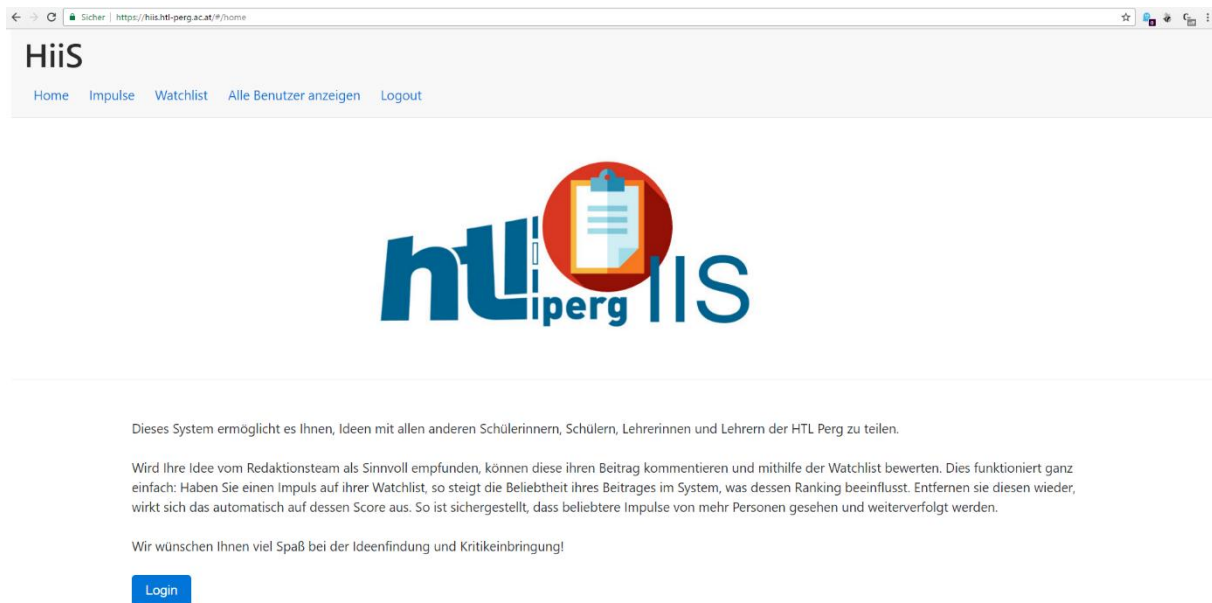


Abbildung 9: Home Seite Screenshot

Als Erstes sieht man die Home-Seite (verfügbar auf <https://hiis.htl-perg.ac.at>).

Wenn man sich jetzt einloggen möchte, kann man auf den „Login“-Button klicken und man wird direkt zur Login Seite weitergeleitet.

Auf der Home-Seite ist außerdem eine kurze Beschreibung über die Funktionen der Webseite und was man damit machen kann. Ein Logo wurde ebenfalls eingebunden.

Am oberen Rand findet man eine Menüleiste, die aber nicht nutzbar ist, solange man sich nicht eingeloggt hat.

6.1.4.2 Login

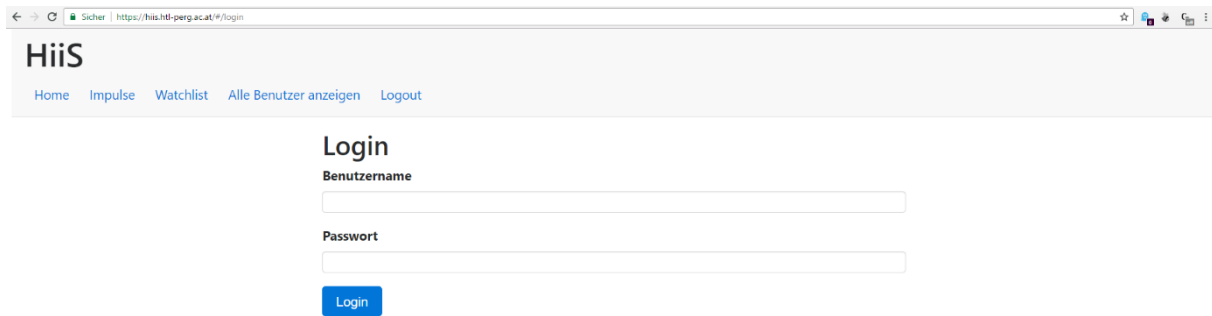
A screenshot of a web browser showing the login page for 'Hiis'. The browser's address bar shows the URL 'https://hiis.htl-perg.ac.at/#/login'. The page header includes the 'Hiis' logo and a navigation menu with links for 'Home', 'Impulse', 'Watchlist', 'Alle Benutzer anzeigen', and 'Logout'. The main content area is titled 'Login' and contains two input fields: 'Benutzername' (Username) and 'Passwort' (Password). Below the password field is a blue 'Login' button.

Abbildung 10: Login Screenshot

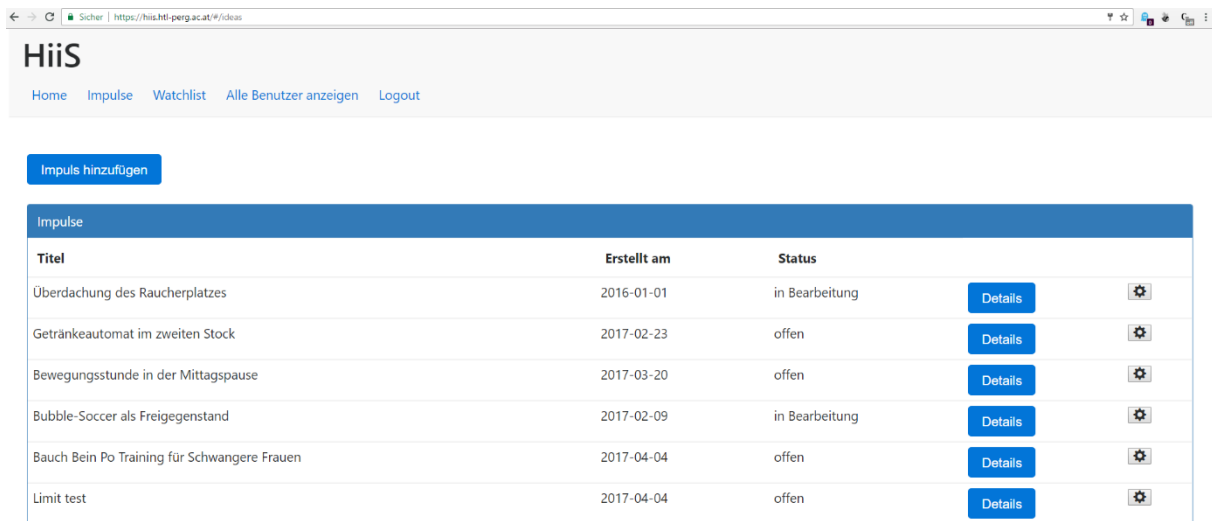
Zum Einloggen muss man seinen Username und sein Passwort eingeben. In unserem Fall kann man seinen IF-Schulbenutzer und das dazugehörige Passwort verwenden.

Durch Klicken auf den Login-Button wird man, sofern man die richtigen Werte eingegeben hat, auf die Impuls Seite weitergeleitet von dort aus kann man alles nutzen.

Beim Login wird ein Request an den Service, welcher auf dem Schulserver läuft, gesendet.

Im Login-Bereich werden die eingegebenen Werte abgeglichen und danach wird man nach positiver Prüfung der Eingabedaten weitergeleitet oder es wird einem der Zugriff verweigert, wenn der Benutzer nicht erkannt wurde. Weiters wird, wie oben im Bild zu sehen ist, ein Menü angezeigt. Darauf kann man allerdings nicht zugreifen, solange man nicht eingeloggt ist.

6.1.4.3 Impulse



Titel	Erstellt am	Status		
Überdachung des Raucherplatzes	2016-01-01	in Bearbeitung	Details	⚙️
Getränkeautomat im zweiten Stock	2017-02-23	offen	Details	⚙️
Bewegungsstunde in der Mittagspause	2017-03-20	offen	Details	⚙️
Bubble-Soccer als Freigegegenstand	2017-02-09	in Bearbeitung	Details	⚙️
Bauch Bein Po Training für Schwangere Frauen	2017-04-04	offen	Details	⚙️
Limit test	2017-04-04	offen	Details	⚙️

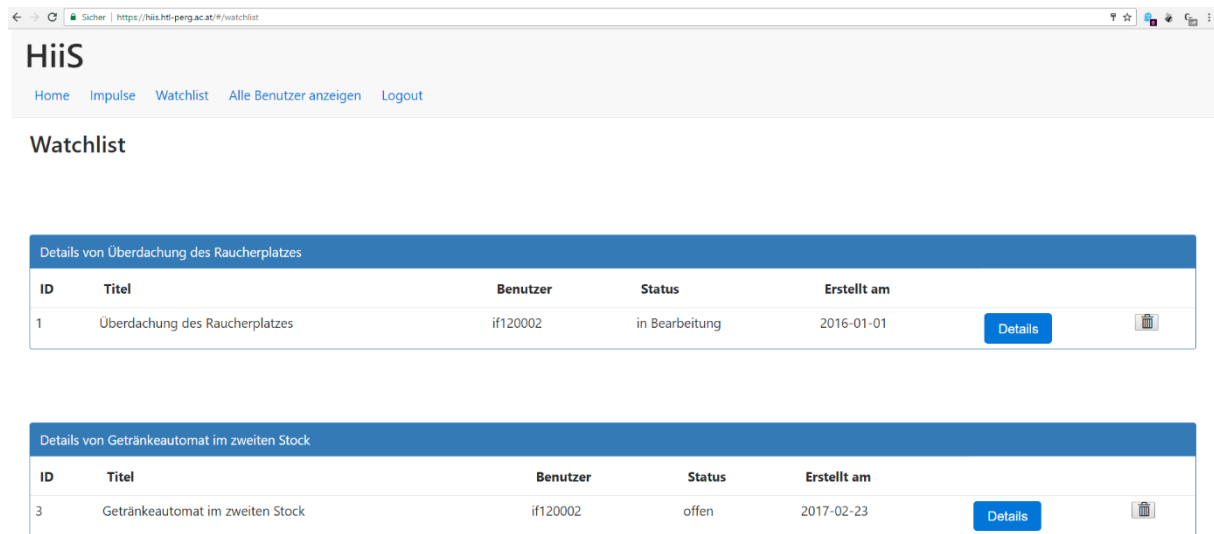
Abbildung 11: Impulse Screenshot

Die Impulse-Seite beinhaltet eine der wichtigsten Funktionen der Webseite. Hier werden alle bereits im System gepflegten Impulse angezeigt. Auf einen Blick sieht man den Titel, das Erstellungsdatum und den Status, in dem sich der Impuls gerade befindet. Außerdem kann man von hier aus auf die Detailansicht jedes Impulses weitergehen, wo man alle Informationen darüber einsehen und bearbeiten kann.

Für Redakteure ist es außerdem möglich, einen ausgewählten Impuls zu bearbeiten. Somit können Impulse archiviert oder gelöscht werden. Das funktioniert durch das Klicken auf das Zahnradchen-Icon, welches nur Redakteure haben, beim jeweiligen Impuls.

Alle Benutzer können außerdem von hier aus einen neuen Impuls anlegen, indem sie einfach auf den „Impuls hinzufügen“-Button klicken. Auch auf dieser Seite ist das Menü oben wieder vorhanden. Von dort aus wird man, wie schon zuvor auf der Home-Seite, durch klicken auf den jeweiligen Button auf die gewünschte Seite weitergeleitet.

6.1.4.4 Watchlist



The screenshot shows the Hiis Watchlist page. At the top, there is a navigation bar with links for Home, Impulse, Watchlist, Alle Benutzer anzeigen, and Logout. Below the navigation bar, the title 'Watchlist' is displayed. The main content area contains two table entries, each with a blue header bar indicating the title of the item.

Details von Überdachung des Raucherplatzes				
ID	Titel	Benutzer	Status	Erstellt am
1	Überdachung des Raucherplatzes	if120002	in Bearbeitung	2016-01-01

Details von Getränkeautomat im zweiten Stock				
ID	Titel	Benutzer	Status	Erstellt am
3	Getränkeautomat im zweiten Stock	if120002	offen	2017-02-23

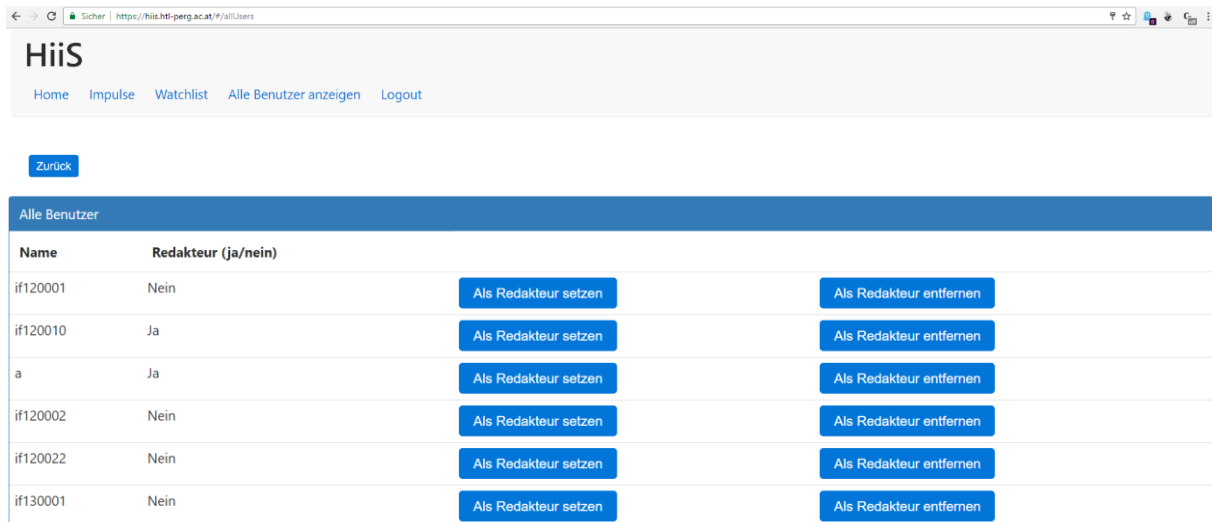
Abbildung 12: Watchlist Screenshot

Bei der Watchlist-Seite handelt es sich um eine Auflistung aller Impulse, die ein Benutzer verfolgen möchte. Hier erscheinen also alle Impulse, die ein Benutzer zu seiner Watchlist (als Favorit) hinzugefügt hat.

Von hier aus kann man wieder auf die Detail-Ansicht zu einem Impuls zugreifen um sich alle Informationen zu einem Impuls anzusehen. Dort kann man dann auch wieder Kommentare zum Impuls verfassen.

In der eigenen Watchlist-Ansicht ist es auch möglich, den Impuls zu entfernen, wenn man über diesem nicht längeren Bescheid wissen möchte. Natürlich ist es auch von hier aus über das Menü oben wieder möglich, auf die Home-Seite oder die Impulse-Übersicht weitergeleitet zu werden. Ausloggen funktioniert wie von den anderen Seiten gewohnt auch hier wieder.

6.1.4.5 Alle Benutzer anzeigen



Name	Redakteur (ja/nein)		
if120001	Nein	Als Redakteur setzen	Als Redakteur entfernen
if120010	Ja	Als Redakteur setzen	Als Redakteur entfernen
a	Ja	Als Redakteur setzen	Als Redakteur entfernen
if120002	Nein	Als Redakteur setzen	Als Redakteur entfernen
if120022	Nein	Als Redakteur setzen	Als Redakteur entfernen
if130001	Nein	Als Redakteur setzen	Als Redakteur entfernen

Abbildung 13: Alle Benutzer anzeigen Screenshot

Bei dieser Seite kann ein Redakteur jeden Benutzer als neuen Redakteur setzen oder vorhandene Redakteure entfernen, sodass sie wieder nur Benutzer sind.

Um das realisieren zu können, hat der Redakteur eine Tabelle mit allen Benutzern zu Verfügung. Durch einfaches Klicken auf „Als Redakteur setzen“ oder auf „Als Redakteur entfernen“, wird der Benutzer als Redakteur gesetzt oder entfernt.

Diese Seite kann nur ein Redakteur besuchen, damit nicht alle Benutzer Zugriff auf die Verwaltung der restlichen haben.

Auch hier gibt es wieder die Menüleiste mit der man jederzeit zu anderen Seiten navigieren. Durch Klicken auf „zurück“ kommt man zu der vorherigen Seite.

Impuls hinzufügen

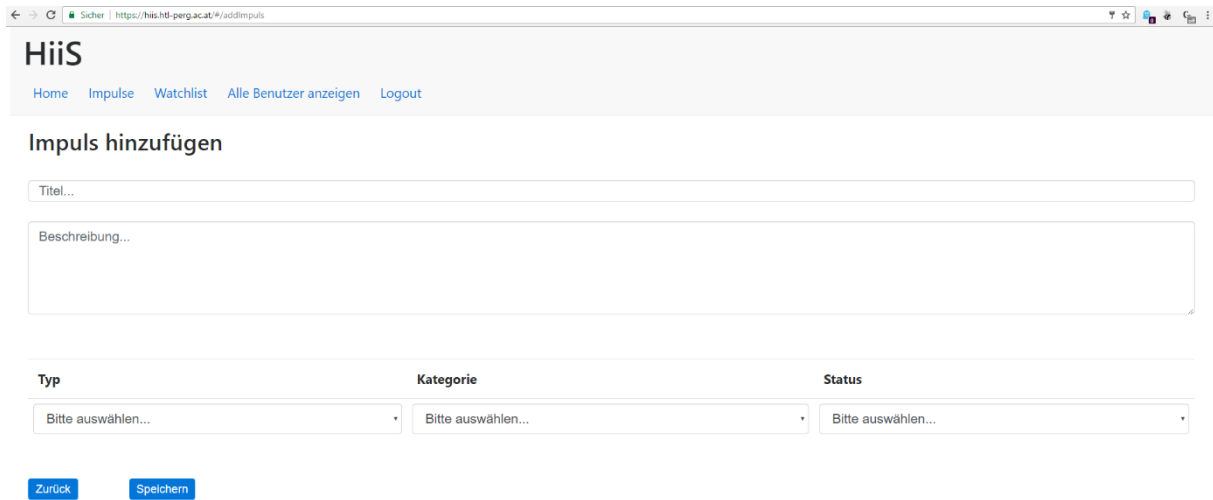
The screenshot shows a web browser window with the URL 'https://hii.htl-perg.ac.at/#/addimpuls'. The page title is 'HiiS'. The navigation menu includes 'Home', 'Impulse', 'Watchlist', 'Alle Benutzer anzeigen', and 'Logout'. The main heading is 'Impuls hinzufügen'. Below it are two text input fields: 'Titel...' and 'Beschreibung...'. Underneath are three dropdown menus labeled 'Typ', 'Kategorie', and 'Status', each with the placeholder text 'Bitte auswählen...'. At the bottom are two buttons: 'Zurück' and 'Speichern'.

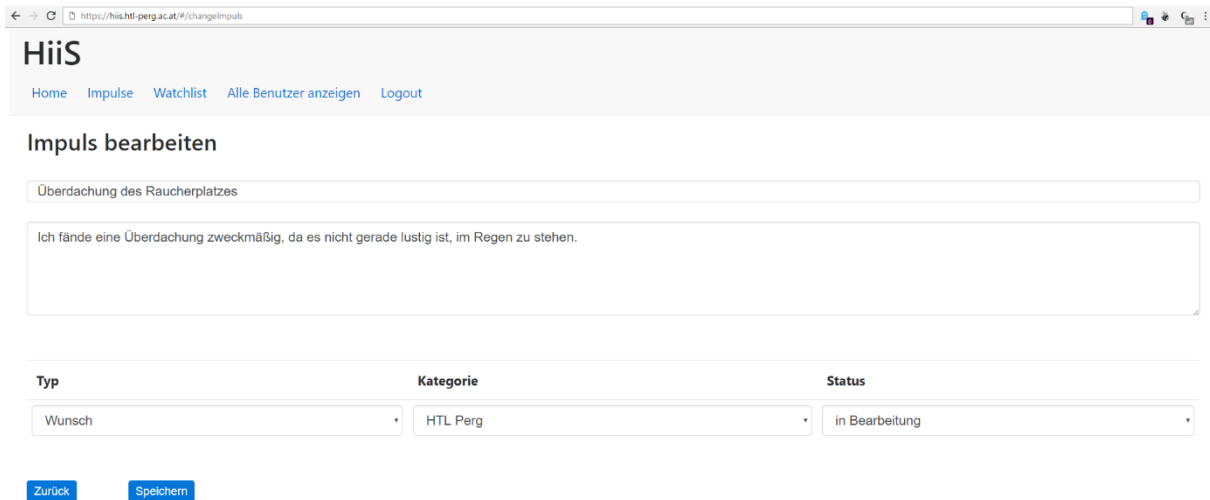
Abbildung 14: Impuls hinzufügen Screenshot

Ein Benutzer kann über diese Seite einen Impuls anlegen. Er muss dort den Titel und die Beschreibung in die offenen Felder schreiben. Durch die drei Dropdowns müssen Typ, Kategorie und Status ausgewählt werden.

Mit einem einfachen Klick auf „Speichern“ kann der Impuls am Ende erstellt werden. Wenn man das Erstellen verwerfen möchte, drückt man auf „zurück“ und der Impuls wird nicht erstellt.

Auch auf dieser Seite kann man mithilfe der Menüleiste zwischen den einzelnen Seiten navigieren.

6.1.4.6 Impuls bearbeiten



https://hii.s.htl-perg.ac.at/#/changeImpuls

HiiS

[Home](#) [Impulse](#) [Watchlist](#) [Alle Benutzer anzeigen](#) [Logout](#)

Impuls bearbeiten

Überdachung des Raucherplatzes

Ich fände eine Überdachung zweckmäßig, da es nicht gerade lustig ist, im Regen zu stehen.

Typ	Kategorie	Status
Wunsch	HTL Perg	in Bearbeitung

[Zurück](#) [Speichern](#)

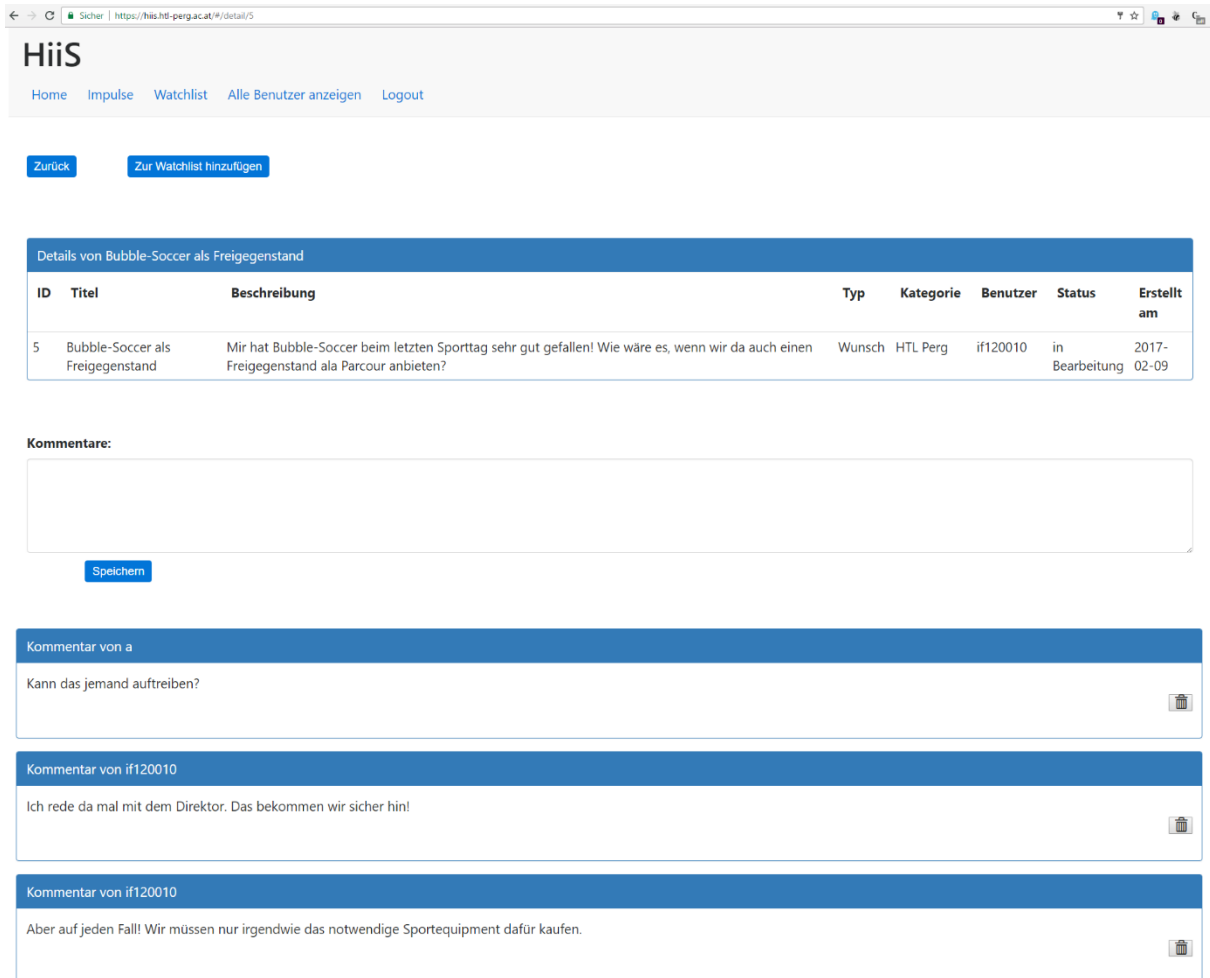
Abbildung 15: Impuls bearbeiten Screenshot

Redakteure können einen Impuls nach dessen Einreichung bearbeiten. Es werden die Daten des Impulses, den man ändern möchte, als Platzhalter in den jeweiligen Feldern angezeigt, damit man weiß, was vorher darin geschrieben stand. Zuerst sieht man den Titel, danach die Beschreibung des Impulses. Der Typ, die Kategorie und der Status können mit einem Dropdown ausgewählt bzw. geändert werden.

Wenn man den Impuls doch nicht ändern bzw. die vorgenommenen Änderungen verwerfen möchte kann man ganz einfach wieder abrechnen mit dem „Zurück“ Button. Falls man Änderungen speichern möchte, klickt man einfach auf den „Speichern“-Button.

Auch von hier aus kann man das Menü ganz oben für die Navigation verwenden, um auf eine andere Seite zu gelangen.

6.1.4.7 Detailansicht



Details von Bubble-Soccer als Freigegegenstand

ID	Titel	Beschreibung	Typ	Kategorie	Benutzer	Status	Erstellt am
5	Bubble-Soccer als Freigegegenstand	Mir hat Bubble-Soccer beim letzten Sporttag sehr gut gefallen! Wie wäre es, wenn wir da auch einen Freigegegenstand ala Parcour anbieten?	Wunsch	HTL Perg	if120010	in Bearbeitung	2017-02-09

Kommentare:

Kommentar von a

Kann das jemand aufreiben?

Kommentar von if120010

Ich rede da mal mit dem Direktor. Das bekommen wir sicher hin!

Kommentar von if120010

Aber auf jeden Fall! Wir müssen nur irgendwie das notwendige Sportequipment dafür kaufen.

Abbildung 16: Details Screenshot

Bei der Detailansicht handelt es sich um eine Übersicht über alle Details zu einem zuvor ausgewählten Impuls. Man sieht den Titel und die Beschreibung, sowie Typ, Kategorie, den Benutzer, den derzeitigen Status und das Erstellungsdatum des Impulses. An dieser Stelle kann man zusätzlich einen Kommentar hinzufügen, der bei Bedarf wiederum von einem Redakteur gelöscht werden könnte. Dies erfolgt einfach durch das Eintippen des gewünschten Textes in das Textfeld und den darauffolgenden Klick auf „Speichern“. Solche Kommentare, die von jedem Benutzer zu jedem Impuls erstellt werden können, werden in dieser Detailansicht angezeigt.

Von hier aus kann man den Impuls auch zu seiner persönlichen Watchlist hinzufügen, wenn man seinen Werdegang weiterverfolgen möchte. Ebenso wie auf jeder Seite kann auch hier das Menü im oberen Seitenbereich für die Navigation zu anderen Seiten der Online-Plattform verwendet werden.

6.2 Android Applikation als Zugang zu HIIIS

Da sich Benutzer mit ihrem Schul-Account in der mobilen Android-HIIS-App anmelden werden alle Änderungen und Aktualisierungen synchronisiert. Dadurch finden Benutzer alles, was sie in der App gemacht haben auch nach Anmeldung auf der HIIIS-Webseite wieder und umgekehrt.

6.2.1 Funktion der Android Applikation für HIIIS

Zu den Hauptfunktionalitäten der vor allem für die mobile Anwendung gedachte Android Applikation gehören, dass Impulse auch außerhalb der schulischen Infrastruktur entstehen können. Wenn es keine Möglichkeit gibt spontane Ideen umgehend zu notieren, ist die Wahrscheinlichkeit groß, dass Ideen, die Benutzern außerhalb der Schule einfallen, verloren gehen und nie in HIIIS erfasst werden. Um dies zu vermeiden können Benutzer die in diesem Projekt bereitgestellte HIIIS-Android-App am eigenen Android-Gerät verwenden, um den Impuls sofort zu erfassen und einzumelden. Ziel ist es, die App für Benutzer so einfach und intuitiv wie möglich zu halten. Deshalb wurden folgende Funktionalitäten entwickelt und unkompliziert umgesetzt:

- Anzeigen aller neuen Impulse
- Erstellen eines neuen Impulses
- Details zu einem Impuls anzeigen
- Impulse zur eigenen Watchlist hinzufügen
- Die eigene Watchlist anzeigen

6.2.2 Ablauf in der Android Applikation

Nach dem Start der HIIIS-Android-Applikation ist der persönliche Login erforderlich, was gerade außerhalb der schulischen Infrastruktur im Sinne des Datenschutzes besonders wichtig ist. Nach positivem Login gelangt der Benutzer automatisch in das Hauptmenü, von wo aus auf die Vorschlagliste und eigene Watchlist navigiert und ein neuer Impuls erfasst werden kann. Wenn diese Option gewählt wird gelangt man in die Detailansicht, wo man weitere Informationen zu einem Impuls einsehen kann.

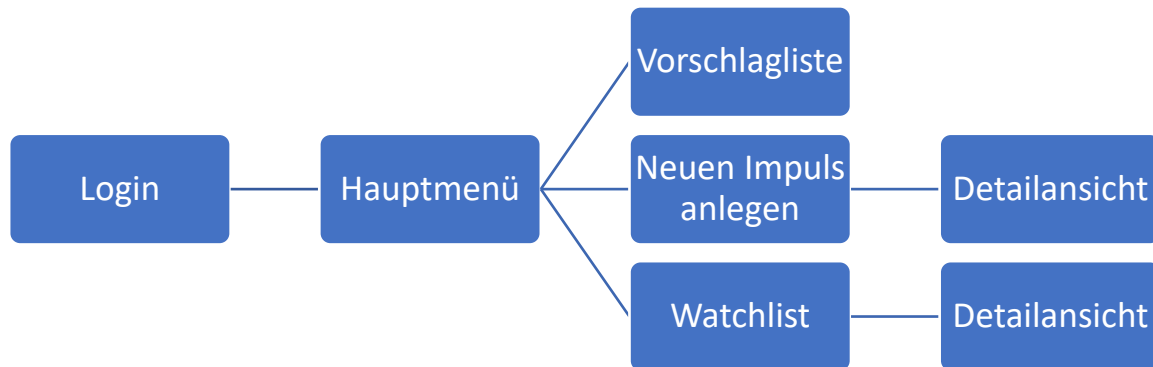


Abbildung 17: Ablauf der Android-App

6.2.3 Anwender der Android Applikation

Die Benutzer der IIS-Android-App sind sowohl die Schülerinnen und Schüler und auch die Lehrerinnen und Lehrer. Alle können sich mit ihrem persönlichen Schul-Login (Active Directory-Account) anmelden, somit muss kein separater Account erstellt werden um diese mobile Applikation nutzen zu können. Grundsätzlich stehen dann allen Benutzern die gleichen Funktionalitäten zur Verfügung. Im Gegensatz zur Webseite werden Benutzer mit Redakteursrechten keine zusätzlichen Funktionalitäten angeboten. Die mobile Anwendung ist nicht für die Bearbeitung und Verwaltung von Impulsen, sondern für das Anlegen und die Statureinsicht von Impulsen gedacht.

6.2.4 Funktionalitäten der Android Applikation

6.2.4.1 Login

Beim Starten der App werden Benutzer mit einem schlicht gehaltenen Bildschirm begrüßt. Hier können nun der Benutzername und das Passwort eingegeben werden. Besonders komfortabel ist die Tatsache, dass Benutzer sich mit ihrem Active Directory-Account einloggen, ohne sich um eine zusätzliche Registrierung kümmern zu müssen. Durch einen Klick auf den Login-Button verbindet sich die App mit dem Server.

Wenn die Verbindung aufgebaut werden konnte wird überprüft ob der eingegebene Benutzer gesperrt ist. Falls der Benutzer nicht durch einen Redakteur gesperrt wurde öffnet sich anschließend die Main Activity-Seite.

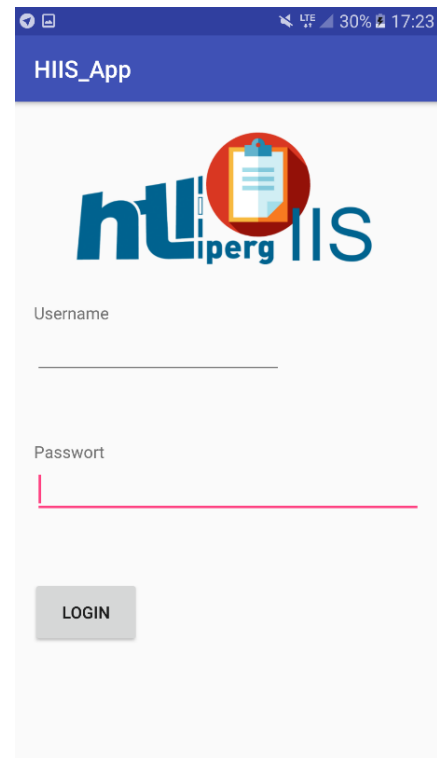


Abbildung 18: Login Activity

6.2.4.2 Main Activity Vorschlagsliste

Die Main Activity ist das Herzstück der Anwendung. Hier werden alle nicht abgeschlossenen Impulse, welche vom Redaktionsteam bereits auf Integrität überprüft wurden, angezeigt. Der Benutzer kann nun auf einen gewünschten Impuls klicken, um sich genauere Details dazu anzusehen. Im oberen Bereich des Bildschirms befindet sich ein Button mit einem „+“ Symbol, womit sofort in die Ansicht zur Erstellung eines neuen Impulses gewechselt werden kann.

Eine weitere Funktion ist der Watchlist-Button „Aus“ oder „Ein“. Mit diesem kann hier zwischen dem Watchlist-Modus und der gesamten Vorschlagslistenanzeige gewechselt werden. Im Watchlist-Modus werden nur Impulse angezeigt, die der Benutzer bereits zur eigenen Watchlist hinzugefügt hat.

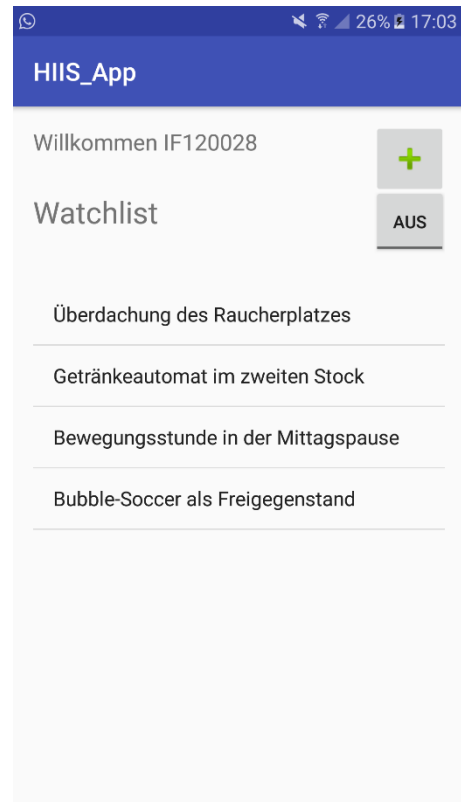


Abbildung 19: Main Activity

6.2.4.3 Create Impulse Activity

Auf dieser Activity-Seite können neue Impulse angelegt werden. Der Benutzer gibt einen sprechenden Titel und eine genauere Beschreibung für den neuen Impuls an. Zusätzlich kann ein Impuls-Typ und eine Impuls-Kategorie ausgewählt werden. Die Werte der Felder Typ und Kategorie entsprechen den aktuellen Werten in der Datenbank.

Wenn alle Felder befüllt wurden kann der Impuls abschließend mit einem Klick auf den „Neuer Impuls“-Button eingereicht werden.

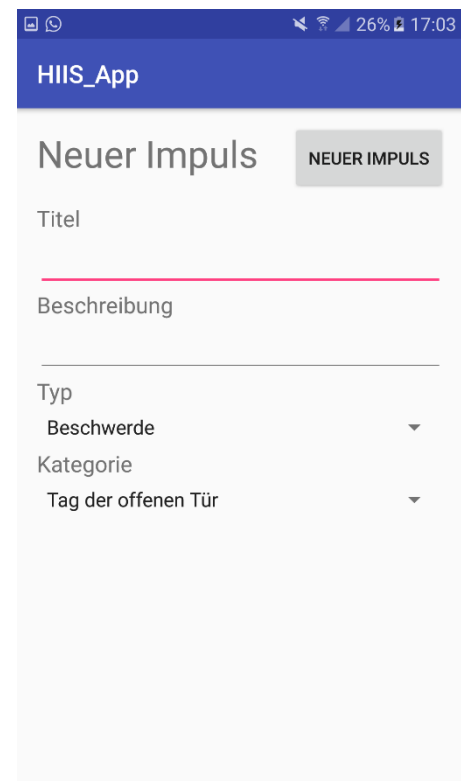


Abbildung 20: Create Impuls Activity

6.2.4.4 Detail Activity

Auf der Detail-Activity-Seite werden alle vorhandenen Details zu einem zuvor ausgewählten Impuls angezeigt. Eine weitere Funktion an dieser Stelle ist die Möglichkeit diesen Impuls mit einem Klick auf den „Zur Watchlist“-Button den Impuls zur eignen Watchlist hinzuzufügen um diesen später in der Main Activity im Watchlist-Modus jederzeit einsehen zu können.

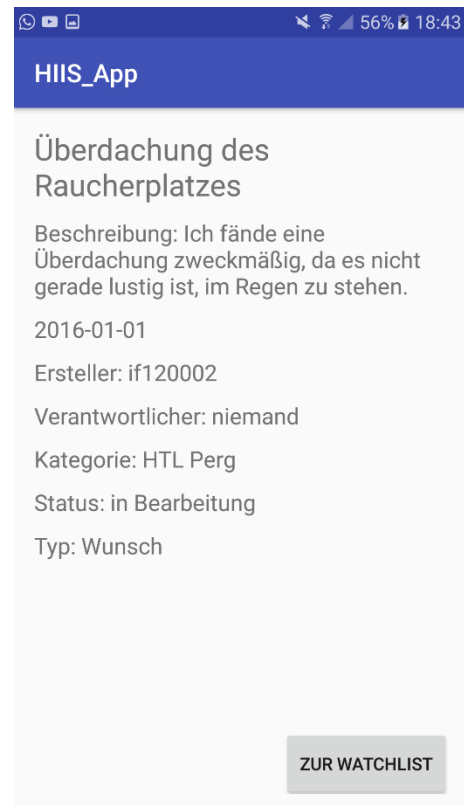


Abbildung 21: Detail Activity

7 Programmbeschreibung

7.1 Architektur

Für die Plattform zum integrierten Ideenmanagement HIIS wurde ein einfaches Architekturmodell gewählt, das so weit als möglich gemeinsame Grundlagen für die Webseite und die Android-Applikation verwendet. Wie die folgende Abbildung zeigt besteht die Architektur von HIIS aus vier Hauptkomponenten, wobei die beiden Benutzer-Plattformen über REST-Schnittstellen und der Webserver mittels Java Database Connectivity (JDBC) an die Datenbank angebunden ist.

- **Database Server**

Der Datenbankserver ist das Fundament der HIIS-Ideenmanagementplattform. Es speichert die Daten, die von den Endgeräten und vom Server verwendet werden.

- **Webserver**

Der Webserver dient als Vermittler zwischen dem Datenbankserver und den Endgeräten. Er benutzt die Stored Procedures des Datenbankservers und stellt mittels REST eine Schnittstelle für die Endgeräte dar.

- **Webseite**

Die Webseite unterscheidet zwischen Benutzern mit und ohne Redakteursrechten. Impulse können hier erfasst, angesehen und auch bewertet werden.

- **Mobile App**

Im Gegensatz zur Webseite wird hier nicht zwischen Benutzern und Redakteuren unterschieden. Die App dient lediglich zum Erstellen und Anzeigen von Impulsen.

Im folgenden Diagramm ist nun der Aufbau unseres Systems zu sehen. Die Android-App und die Web-Anwendung greifen über die vom Server zur Verfügung gestellte REST-Schnittstelle auf die Daten zu. Dieser Vorgang ist für beide Anwendungen nicht sichtbar, da der Server selbst mithilfe von Java Database Connectivity (JDBC) die Stored Procedures in der Datenbank aufruft.

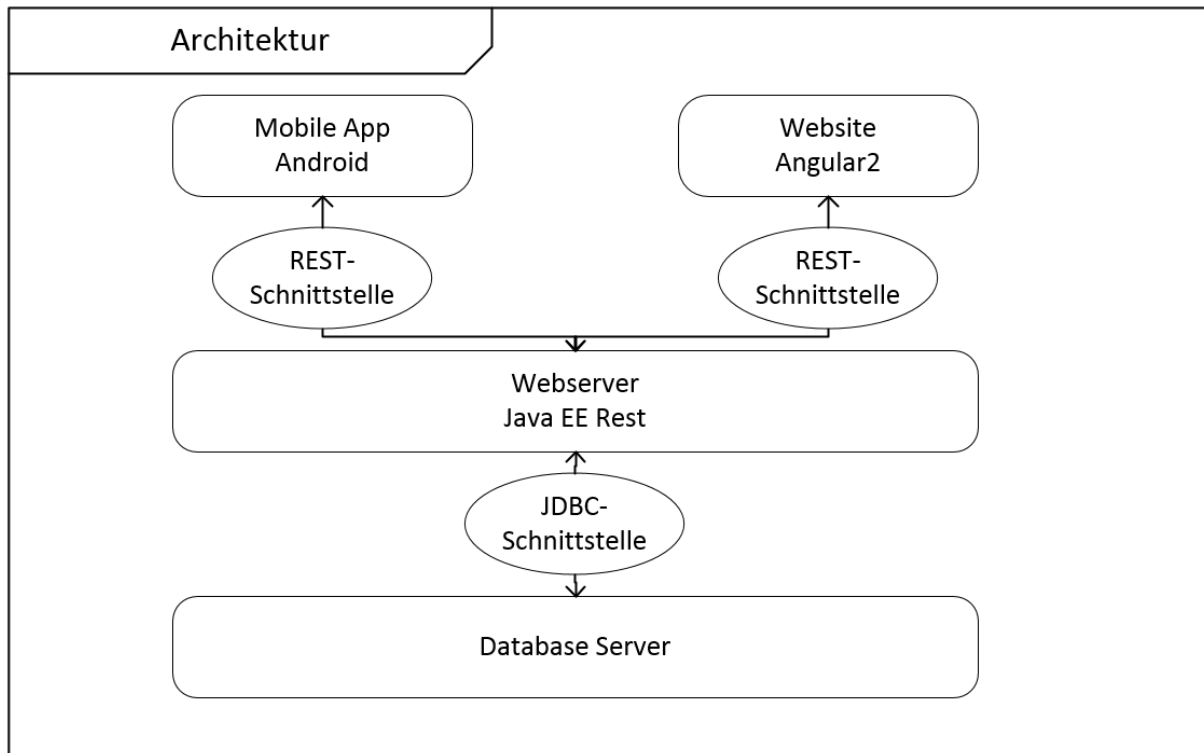


Abbildung 22: Architekturdiagramm von HIIS

7.2.1 Tabellen

7.2.1.1 Tabelle "Status"

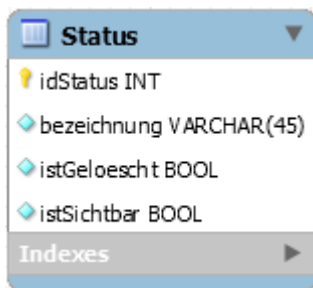


Abbildung 25: Tabelle "Status"

idStatus	bezeichnung	istGeloesch t	istSichtbar
1	offen	false	true
2	in Bearbeitung	false	true
3	zurückgestellt	false	true
4	archiviert	false	false
5	geloesch t	false	false
6	fertig	false	true

Abbildung 24: Standarddaten "Status"

Die Tabelle Status, die in den Abbildungen oben zu sehen ist, speichert grundsätzlich eine Bezeichnung der auswählbaren Status für einen Impuls und zwei Boolean-Werte, wobei einer vorgibt, ob ein Status in der Webanwendung bzw. in der App aufscheint und der andere gibt an, ob der Status selbst als gelöscht markiert wurde und nun nicht mehr in Verwendung ist. Identifiziert wird diese durch die Spalte "idStatus" und sie dient dazu, den momentanen Zustand eines Impulses zu zeigen (Beispiele hierfür sind: archiviert, offen, in Bearbeitung, etc.).

7.2.1.2 Tabelle "Typ"

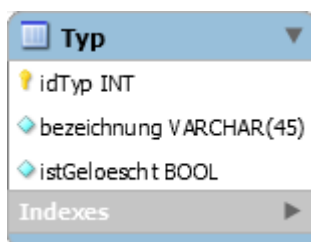


Abbildung 27: Tabelle "Typ"

idTyp	bezeichnung	istGeloesch t
1	Beschwerde	false
2	Kritik	false
3	Wunsch	false
4	Verbesserung	false

Abbildung 26: Standarddaten "Typ"

Bei der Tabelle Typ wurde im Gegensatz zum Status auf den Boolean "istGeloesch t" verzichtet, da hier kein Bedarf besteht einen Typus vor Benutzern zu verstecken. Ein Typ gibt an, wie dieser zu verstehen ist, also beispielsweise als Wunsch, Beschwerde oder Kritik. Natürlich lässt sich diese Liste, die in den Abbildungen oben zu sehen ist, nach Belieben erweitern.

7.2.1.3 Tabelle "Kategorie"

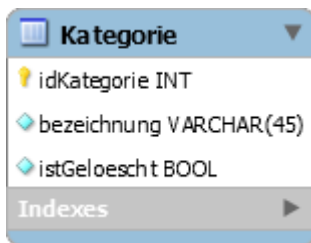


Abbildung 29: Tabelle "Kategorie"

idKategorie	bezeichnung	istGeloescht
1	Tag der offenen Tür	false
2	Messe Wels	false
3	HTL Perg	false

Abbildung 28: Standarddaten "Kategorie"

Diese Tabelle ist wie ihr Partner, der Typ, aufgebaut, wobei die Tabelle „Kategorie“ angibt, welchem Bereich ein Impuls angehört. Hier eignen sich beispielsweise der „Tag der offenen Tür“ in der HTL Perg oder die „Messe Wels“ als aktuelle Anlässe. Auch die Einträge in dieser Tabelle sind erweiterbar.

7.2.1.4 Tabelle "Impuls"



Abbildung 30: Tabelle "Impuls"

idImpuls	titel	beschreibung	Typ_idTyp	Kategorie_idKa	Benutzer_idBer	Status_idStatu	Verantwortlich	Erstellt
1	Überdachung des Raucherplatzes	Ich fände eine Überdachung zweckm...	3	3	4	2	null	2016-01-01
2	Werbungsmaßnahmen für TdoT	Der Tag der offenen Tür sollte besser...	2	1	2	4	2	2017-10-12
3	Getränkeautomat im zweiten Stock	Ein weiterer Getränkeautomat in der ...	3	3	4	1	2	2017-02-23
4	Bewegungsstunde in der Mittagspause	Aufgrund dessen, dass die Schüler (u...	3	3	3	1	null	2017-03-20
5	Bubble-Soccer als Freigegegenstand	Mir hat Bubble-Soccer beim letzten Sp...	3	3	2	2	4	2017-02-09

Abbildung 31: Testdaten "Impuls"

Bei der „Impuls“-Tabelle handelt es sich um das zentrale HIIS-Element, da hier fast alle anderen Informationen verlinkt sind bzw. zusammenfließen. Von sich selbst aus speichert diese Tabelle einen Titel, eine Beschreibung und das Erstelldatum zum Impuls. Wie auch alle anderen Tabellen ist sie über eine eigene ID eindeutig identifizierbar. Die Status, die ein Impuls annehmen kann, sind im State Transition Diagram ersichtliche.

7.2.1.5 Tabelle "Datei"

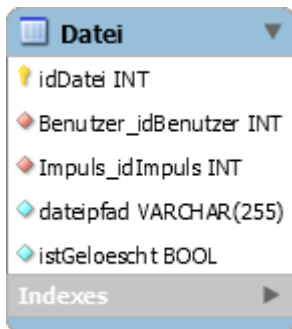


Abbildung 32: Tabelle "Datei"

idDatei	Benutzer_idBer	Impuls_idImpul	dateipfad	istGeloesch t
1	1	1	/path/to/file	false
2	2	2	/path/to/file	true

Abbildung 33: Testdaten "Datei"

In der Tabelle „Datei“ werden die IDs eines Benutzers und eines Impulses mit einem Dateipfad und einer Gelöscht-Markierung gespeichert. Somit ist sichergestellt, dass ein Benutzer mehrere Dateien zu einem Impuls hinzufügen kann bzw. dass mehrere Impulse dieselbe Datei referenzieren können.

7.2.1.6 Tabelle "Kommentar"

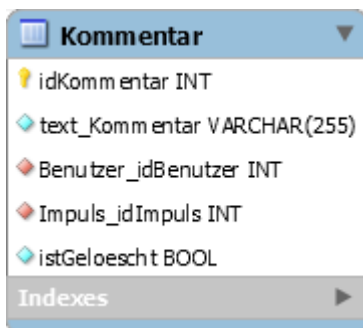


Abbildung 34: Tabelle "Kommentar"

idKommentar	text_Komm entar	Benutzer_idBer	Impuls_idImpul	istGeloesch t
1	Ich denke nicht, dass sich das umsetzen lässt...	1	1	false
2	Das finde ich auch!	2	2	true
3	Wozu? Ich finde das nicht notwendig.	1	2	false
4	Ich schließe mich an, dass das gut wäre und ic...	2	1	false
5	Material inwiefern?	4	1	false
6	Nun ja, ein paar Stützen und eventuell kann ic...	2	1	false
7	Das wäre ja perfekt! :)	4	1	false

Abbildung 35: Testdaten "Kommentar"

In dieser Entität „Kommentar“ lässt sich statt eines Dateipfades ein Textkommentar mit 255 Zeichen speichern. Der übrige Aufbau entspricht jenem der Tabelle "Datei".

7.2.1.7 Tabelle "Watchlist"

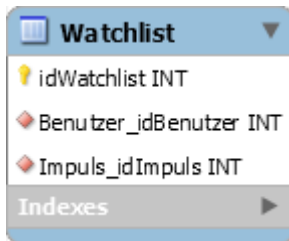


Abbildung 37: Tabelle "Watchlist"

idWatchlist	Benutzer_idBer	Impuls_idImpul
1	1	1
2	1	2
3	2	1
4	1	4

Abbildung 36: Testdaten "Watchlist"

In der Tabelle „Watchlist“ werden die IDs von Benutzern mit denen der Impulse vereint, um jedem Benutzer seine favoritenähnliche Watchlist-Funktionalität zu bieten. Somit kann jeder Benutzer jene Impulse auswählen, zu denen er auf dem aktuellsten Stand bleiben möchte.

7.2.1.8 Tabelle "Benutzer"

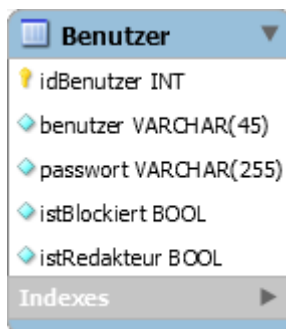


Abbildung 38: Tabelle "Benutzer"

idBenutzer	benutzer	password	istBlockiert	istRedakteur
1	if120001		false	false
2	if120010		true	true
3	a		false	false
4	if120002		false	false
5	if120022		false	false
6	if130001		true	false

Abbildung 39: Testdaten "Benutzer"

Diese Tabelle stellt die Grunddaten eines Benutzers zur Verfügung, die zum Teil auch beim Einloggen für HIIS benötigt werden. Diese umfassen den Benutzernamen, ein Passwort und zwei Booleans, von denen einer angibt, ob ein Benutzer höhere Rechte besitzt (Redakteur) und der andere, ob ein Benutzer durch einen Redakteur vom System ausgeschlossen wurde.

7.2.2 Views

7.2.2.1 View “vw_impuls_overview”

Diese View wird zur Anzeige von Impulsen inklusive der Identifikation des einmeldenden Benutzers und der Statusbezeichnung verwendet. Bei einigen Stored Procedures vereinfacht diese die Übersicht enorm, da man nicht immer alles einzeln zusammensuchen muss, sondern ganz einfach die View verwendet.

```
CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `hiis_Model`.`vw_impuls_overview` AS
  SELECT
    `i`.`Benutzer_idBenutzer`,
    `i`.`idImpuls`,
    `i`.`titel`,
    `i`.`Erstellt`,
    `b`.`benutzer`,
    `i`.`Status_idStatus`,
    `s`.`bezeichnung` AS `status_bezeichnung`
  FROM
    (`hiis_Model`.`Impuls` `i`
  JOIN `hiis_Model`.`Benutzer` `b` ON ((`i`.`Benutzer_idBenutzer` = `b`.`idBenutzer`))
  JOIN `hiis_Model`.`Status` `s` ON ((`i`.`Status_idStatus` = `s`.`idStatus`)))
```

Abbildung 40: Die View "vw_impuls_overview"

7.2.3 State Transition Diagram

Ein State Transition Diagram gibt an, wie sich ein bestimmter Status im Laufe der Zeit verändern kann. Dies ist für die nachfolgende Grafik relevant, da der Status eines Impulses variieren kann.

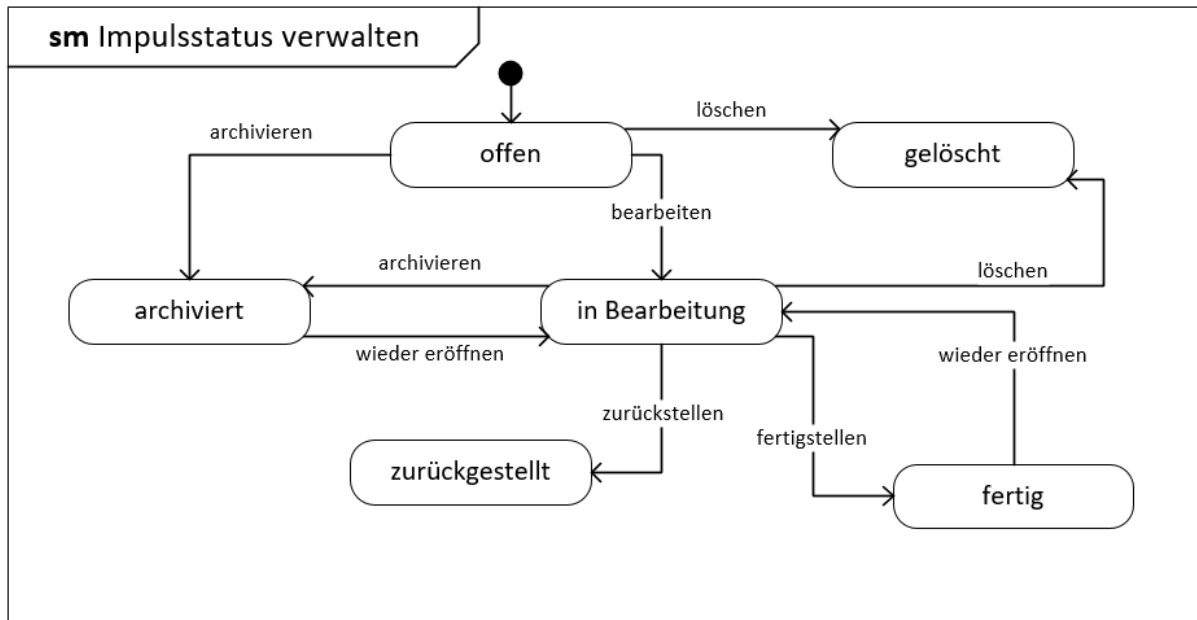


Abbildung 41: State Transition Diagram des Impulsstatus (Vorschlag)

In der Abbildung oben ist der ursprünglich geplante Verlauf zum Status eines Impulses dargestellt. Da dieser von Benutzern mit Redakteur-Status nach Belieben selbst erweitert werden kann, wird an dieser Stelle auf die vorhandene Grundmenge zurückgegriffen. Bei jedem neuen Impuls steht der Status auf "offen" bis jemand sich dessen annimmt bzw. zugewiesen bekommt oder ein Regisseur diesen (z.B. weil das Erstellungsdatum schon lange zurückliegt) ins Archiv verschiebt oder löscht, weil es sich um einen nicht konstruktiven oder unseriösen Impuls handelt. Nach der positiven Erstbewertung erlangt der Impuls im Status "in Bearbeitung", was für die längste Zeit im Lebenszyklus eines Impulses gilt, in dem Kommentare und Inputs (z.B. in Form von Dateien) dazu abgegeben werden können. In diesem Status als auch beim Status "offen" ist es möglich den Impuls auf die benutzereigene Watchlist zu setzen und somit dessen Beliebtheit im System zu steigern. Sollte sich diese Beliebtheit über einen längeren Zeitraum als zu gering herausstellen oder ein Impuls als nicht umsetzbar angesehen werden, so kann dieser im Status als "zurückgestellt" gekennzeichnet werden, wobei der Impuls weiterhin im Lesemodus aufrufbar bleibt. Oder einem Impuls wird der Status "gelöscht" zugewiesen, was zur Folge hat, dass dieser nur noch

mit besonderen Mitteln bzw. Rechten im System aufscheint. Für alle anderen Impulse gilt das Ziel diese in den Status „fertig“ zu bekommen, da dies eine erfolgreich umgesetzte Idee kennzeichnet. Falls erforderlich können fertige Impulse wieder in den Status „in Bearbeitung“ setzen, falls noch etwas ergänzt wird bzw. etwas in der Umsetzung gefehlt hat.

7.2.4 Mengengerüst

Wie in der Abbildung „Datenmodell“ (7.2. Fehler! Verweisquelle konnte nicht gefunden werden.) ersichtlich, haben alle IDs den Datentyp „int“, was einer Menge von 2.147.483.647 Werten entspricht. (dev.mysql.com, 2017) Da es relativ unwahrscheinlich ist, dass diese Menge (z.B. von Benutzern in der Eingabe) ausgereizt wird, hat sich die folgende Aufstellung betreffend das Mengengerüst jeweils ergeben.

7.2.4.1 Status, Typ und Kategorie

Der Zuwachs an Status, Typen und Kategorien ist schwer vorherzusagen, da diese vor allem von der Akzeptanz des Systems und der Anzahl der Benutzer abhängt. Im Allgemeinen lässt sich aber sagen, dass die meisten Datensätze in diesen drei Tabellen wiederverwendet werden, also ein Status, Typ und eine Kategorie auf mehrere Impulse angewendet werden, und ein allzu großer Zuwachs an Statureinträgen, Typen oder Kategorien in diesen drei Tabellen den nächsten Jahren nicht anzunehmen ist. Selbst bei einer Verhundertfachung in den Listenelementen aller drei Tabellen handelt es sich um 600 Werte, was für die aktuelle Dimensionierung des HIIS-Systems unproblematisch ist.

7.2.4.2 Watchlist

Die Watchlist könnte vom Zuwachspotenzial betrachtet eher problematisch werden, als Status, Typ und Kategorie. Bewertet am Status-Quo jedoch ist bei einer Benutzerzahl von 300 Schülerinnen und Schülern und ca. 20 Lehrerinnen und Lehrern (siehe Kapitel 7.1.11.5 Benutzer) nicht davon auszugehen, dass dieses Watchlist-Potenzial in den nächsten Jahren erschöpft ist.

7.2.4.3 Impuls

Die verfügbare Anzahl von mehr als zwei Milliarden Impulsen sollte ausreichend sein, um entsprechend viele Ideen umzusetzen, zu bearbeiten und zurückzustellen, bevor hier Handlungsbedarf, z.B. in Form von Löschen fertiger oder archivierter Impulse, besteht.

7.2.4.4 Kommentar

Zur Einführung des Systems bestehen auch hier keine Engpässe und das gilt voraussichtlich auch für die ersten paar Jahre danach. Andernfalls kann manuell eingegriffen werden und Kommentare, die z.B. an archivierten oder fertigen Impulsen hängen oder älter sind als ein bestimmtes Datum, entfernt werden.

7.2.4.5 Benutzer

Bei einem durchschnittlichen Benutzerzuwachs von hoch geschätzten 70 Personen pro Jahr besteht hier keine Gefahr, dass das System in allzu naher oder mittelfristiger Zukunft durch zu viele Benutzer ausgelastet ist.

7.2.4.6 Datei

Solange die Speicherkapazität des Servers ausreicht können über zwei Milliarden Dateien hochgeladen werden, was eine ideale Grundlage für anregende Diskussionen bei Impulsen bieten sollte.

7.2.5 Stored Procedures speziell für Redakteure

Eine Stored Procedure ist eine vordefinierte Methode im Datenbankmanagementsystem, welche vom Anwendungsserver aufgerufen und mit Parametern aus den Endbenutzeranwendungen ergänzt wird. Die Nachfolgenden stehen neben den Stored Procedures im Folgekapitel nur Benutzern mit aktivem Redakteurstatus zur Verfügung.

7.2.5.1 addRedakteur

Mithilfe der Prozedur "addRedakteur" kann ein als Redakteur registrierter Benutzer einem anderen Benutzer ebenfalls diese Privilegien einräumen.

7.2.5.2 blockUser

“blockUser” kann ebenfalls nur von Redakteuren verwendet werden um bestimmte Benutzer vom System auszuschließen. Diese Blockade kann mithilfe der Funktion „unblockUser“ wieder aufgehoben werden.

7.2.5.3 createKategorie

Um eine weitere Kategorie zu den bereits bestehenden hinzuzufügen wird diese Prozedur verwendet. Ein Benutzer ohne Redakteursrechte kann keine Kategorien anlegen, sondern muss einen Redakteur bitten dies zu tun um unnötige Kategorien in der Auswahl zu vermeiden.

7.2.5.4 createStatus

Wie die Kategorien kann die Statustabelle auch nur von Redakteuren erweitert werden, da ein Benutzer ohne Redakteursrechte nur aus den gewünschten, angebotenen Statusoptionen wählen können soll.

7.2.5.5 createTyp

Auch die möglichen Impuls-Typen sind, wie die Kategorien bzw. der Status, von Redakteuren vorgegeben und können von normalen Benutzern nur gesetzt, aber nicht selbst definiert werden.

7.2.5.6 deleteDatei

Unerwünschte bzw. irrtümlich gespeicherte Dateien können von einem Redakteur entfernt werden, wobei solche Dateien zwar gespeichert bleiben, aber nicht mehr im System zu Impulsen zugeordnet aufscheinen, um einen Datenverlust zu verhindern.

7.2.5.7 deleteImpuls

Nicht integrale Impulse können nach deren Erfassung von einem Redakteur gelöscht werden, sollten diese nicht verfolgenswert oder bei der Erstbewertung irrtümlich befürwortet worden sein.

7.2.5.8 deleteKategorie

Redakteure können Kategorien Benutzern nicht mehr anbieten, indem sie diese löschen. Gelöschte Kategorien existieren zwar weiterhin im System, werden Benutzern aber nicht mehr angeboten. Das kann erforderlich sein, wenn Kategorien obsolet geworden sind oder sich ein Redakteur beim Anlegen einer Kategorie vertippt hat.

7.2.5.9 deleteKommentar

Unsachliche oder unerwünschte Kommentare können von einem Redakteur entfernt werden, wobei diese nun nicht mehr angezeigt werden, aber weiterhin im System bleiben. Diese Funktion ist hilfreich, wenn Benutzer nicht von unerwünschten Kommentaren bzw. dem Benutzer, der diesen erstellt hat, beeinflusst werden sollen.

7.2.5.10 deleteStatus

Ein irrtümlich erstellter Status kann von einem Redakteur auch wieder gelöscht werden, damit keine Verwirrung in der Status-Tabelle entsteht.

7.2.5.11 deleteTyp

Genauso wie bei Kommentar und Status kann auch hier verfahren werden. Ein gelöschter Typ ist für normale Benutzer nicht mehr sichtbar.

7.2.5.12 removeRedakteur

Ein Redakteur kann einem Benutzer die Redakteursrechte entziehen und diesen somit zu einem normalen Benutzer machen, falls dieser irrtümlich ins Redaktionsteam gekommen ist oder der jeweilige Benutzer nicht mehr im Redaktionsteam sein kann, darf oder muss.

7.2.5.13 setStatusVisibility

Wenn ein Redakteur den „nicht-sichtbar“-Status eines Status-Attributs in einem Impuls setzt, so sind diese nur noch für Benutzer mit Redakteursrechten im System einsehbar.

7.2.5.14 showArchiviertImpulse

Alle Impulse mit dem Status „archiviert“ können speziell über diese Prozedur angezeigt werden. Da dieser Status von vornherein schon versteckt im System ist, können diesen nur Redakteure sehen.

7.2.5.15 showDeletedKommentareByBenutzer

Alle gelöschten Kommentare eines ausgewählten Benutzers können mit dieser Stored Procedure abgerufen werden. Es wird somit automatisch verhindert, dass Benutzer ohne Redakteursrechte diese sehen können.

7.2.5.16 showUsers

Mithilfe dieser Prozedur erhält man eine Liste aller momentan im System registrierten Benutzer, damit man beispielsweise eine bestimmte User-ID herausuchen kann, um einen neuen Redakteur festzulegen (siehe Punkt 7.2.5.1 „addRedakteur“)

7.2.6 Stored Procedures für alle Benutzer

Im Folgenden sind alle gespeicherten Prozeduren aufgelistet und erklärt, die absolut jedem Benutzer von HIIS zur Verfügung stehen.

7.2.6.1 addImpulsToWatchlist

Diese Stored Procedure ermöglicht es einem Benutzer, einen bestimmten Impuls auf seine eigene Watchlist zu setzen. Dies erhöht gleichzeitig die Beliebtheit des Impulses im System und verhindert, dass dieser verwaist.

7.2.6.2 changeStatusByImpulsId

Diese Procedure dient der Statusänderung eines Impulses. Dieser kann jeglichen Wert aus der Tabelle „Status“ annehmen und von allen Benutzern geändert werden.

7.2.6.3 impulsOverview

Hierbei handelt es sich um eine Erleichterung zur Anzeige aller Impulse, damit nicht jeder Impuls einzeln von der Datenbank abgefragt werden muss und somit die Auslastung zu stark steigen würde.

7.2.6.4 insertDatei

Grundsätzlich kann jeder Benutzer Dateien hochladen, die Impulsen zugeordnet werden. Ob diese jedoch zugänglich bleiben, entscheidet nach wie vor das Redaktionsteam.

7.2.6.5 insertKommentar

Kommentare können auch von jedem Benutzer erstellt werden, welche dann jeder andere Benutzer ansehen kann.

7.2.6.6 login

Um das System überhaupt benutzen zu können wird zum Einstieg einmal festgestellt, wer der Benutzer ist und ihm dann, falls er über diese verfügt, seine Redakteursrechte zugewiesen. Hat ein Redakteur einen Benutzer vom System ausgeschlossen, so wird diesem beim Login der Zugang verwehrt und er muss sich an einen entsprechenden Redakteur persönlich wenden.

7.2.6.7 newImpuls

Diese Stored Procedure ist sehr wichtig, da sie für die Erstellung eines Impulses verantwortlich ist und dann erst alles Weitere möglich wird. Kommentare oder Dateien können immer zu einem entsprechenden Impuls erstellt werden. Es ist derzeit nicht möglich, einen Impuls ohne Titel zu erstellen, sehr wohl aber ohne Beschreibung, da manche Ideen in wenigen Worten bereits erklärbar sind.

7.2.6.8 removeImpulsFromWatchlist

Ermöglicht es einem Benutzer, einen verfolgten Impuls von seiner Watchlist zu entfernen und somit seine Beliebtheit im System zu reduzieren.

7.2.6.9 setVerantwortlicherBenutzer

Jeder Benutzer kann sich selbst oder, falls er es für notwendig hält, einen anderen als Verantwortlichen für einen Impuls eintragen, damit dieser dafür sorgt, dass der Impuls entsprechend umgesetzt wird.

7.2.6.10 showAllKategorie

Diese Stored Procedure zeigt alle momentan im System gespeicherten Kategorien an, damit diese nicht einzeln abgefragt werden müssen.

7.2.6.11 showAllStatus

Diese Procedure dient zum Anzeigen aller momentan im System hinterlegten Status.

7.2.6.12 showAllTyp

Genauso wie Kategorie und Status dient diese Stored Procedure zum Anzeigen aller Typen in der Tabelle „Typ“, die alle im System bekannten Typen für alle Impulse ausgibt.

7.2.6.13 showBenutzerDaten

Diese Procedure zeigt zu einem bestimmten Benutzer die entsprechend hinterlegten Daten mit einem maskierten Passwort an.

7.2.6.14 showDatei

Jegliche Daten einer Datei inklusive dessen Speicherpfad sind über diese Prozedur abrufbar und können zur Anzeige verwendet werden.

7.2.6.15 showDateienByImpulsId

Alle Dateien, die an einen Impuls angehängt wurden, können direkt angezeigt werden und ersetzen somit das mehrfache Aufrufen der Prozedur „showDatei“.

7.2.6.16 showGeloeschteImpulse

Alle von Redakteuren gelöschten Impulse können über diese Prozedur abgerufen werden, damit man weiß, wer was ins System geschrieben hatte.

7.2.6.17 showImpuls

Diese Procedure zeigt alle Daten eines Impulses an inklusive Ersteller, Titel, Beschreibung, Typ, Status, Verantwortlichem, Ersteller und Erstelldatum. Wenn man einen speziellen Impuls sucht, ist diese Stored Procedure ideal.

7.2.6.18 showKategorieByImpulsId

Sollte man nur an Impulsen einer bestimmten Kategorie Interesse haben, so ist das Listen mit der Prozedur „showKategorieByImpulsId“ möglich.

7.2.6.19 showKommentar

Die Procedure zeigt einen bestimmten Kommentar mithilfe dessen Identifikationsnummer an.

7.2.6.20 showKommentare

Diese Procedure zeigt alle Kommentare zu einem bestimmten Impuls an. Dies können je nach Impuls einer oder mehrere sein.

7.2.6.21 showScore

Diese Stored Procedure gibt einen Score an, der sich daraus berechnet, wie viele Benutzer einen bestimmten Impuls auf ihrer Watchlist haben. Dies ermöglicht ein Ranking von beliebten Impulsen und solchen, die eher weniger Anklang bei den Benutzern in der Plattform zum integrierten Ideenmanagement finden.

7.2.6.22 showStatusByImpulsId

Diese Procedure gibt die Bezeichnung eines Status, der für einen bestimmten Impuls gesetzt wurde, zurück. Diese und ähnliche Funktionen können speziell für das Filtern verwendet werden.

7.2.6.23 showTypByImpulsId

Diese Funktion zeigt den Typ eines über die ID eindeutig identifizierten Impulses an.

7.2.6.24 showVerantwortlichkeitenById

Diese Procedure zeigt einem Benutzer alle Impulse an, für die er als Verantwortlicher eingetragen worden ist. Dies ermöglicht eine einfache Auflistung jener Impulse, für die ein Benutzer als umsetzungsverantwortlich nominiert wurde.

7.2.6.25 showWatchlist

Hier werden alle Impulse angezeigt, die ein Benutzer auf seiner Watchlist gespeichert hat. Dadurch ist dieser immer auf dem aktuellen Stand, was die Impulse betrifft, die ihn sehr interessieren.

7.2.6.26 unblockUser

Ein Redakteur kann einen bestimmten vom System ausgeschlossenen Benutzer wieder Zugang gewähren, indem er diese Funktion verwendet.

7.2.6.27 updateImpuls

Diese Prozedur wird verwendet, wenn sich die Daten eines Impulses geändert haben. Grundsätzlich kann nur der Ersteller eines Impulses, ein Redakteur oder der Verantwortliche die Daten eines Impulses verändern.

7.2.6.28 updateKommentar

Der Ersteller eines Kommentars kann diesen nachträglich mit dieser Funktion bearbeiten, sofern dieser nicht von einem Redakteur gelöscht wurde.

7.3 Implementierungen für das HIIIS

7.3.1 Implementierung Webservice

7.3.1.1 Erstellen des Java Persistence Architecture (JPA)-Projektes

Um ein JPA-Projekt zu erstellen und mit der Datenbank funktionsfähig zu machen, sind folgende Schritte notwendig:

- Ein Dynamic Web Project erstellen
- Mithilfe des Wizards JPA-Funktionalitäten hinzufügen
- Im Projekt selbst die Data Source (Datenquelle) setzen, auf die der Server zugreifen soll
- Implementierung der Anwendung selbst

```
@Entity
@NamedQuery(name="Benutzer.findAll", query="SELECT b FROM Benutzer b")
public class Benutzer implements Serializable
{
private static final long serialVersionUID = 1L;
@Id
private int idBenutzer;
private String benutzer;
private byte istBlockiert;
private String password;
// bi-directional many-to-one association to Datei
@OneToMany(mappedBy="benutzer")
private List < Datei > dateis;
// bi-directional many-to-one association to Impul
@OneToMany(mappedBy="benutzer")
private List < Impul > impuls1;
// bi-directional many-to-many association to Impul
@ManyToMany(mappedBy="benutzer")
private List < Impul > impuls2;
// bi-directional many-to-one association to Kommentar
@OneToMany(mappedBy="benutzer")
private List < Kommentar > kommentars;
public Benutzer ()
{}
//Setter and Getter
}
```

Abbildung 42: Die Klasse "Benutzer"

Zur leichteren Lesbarkeit wurden die Stellen, in denen im Codeauszug der Kommentar „Setter and Getter“ steht, entfernt. Hier waren in der Klasse jeweils die Setter- und Getter-Methoden noch angegeben. Weiters handelt es sich bei diesen Settern und Gettern um die Standardimplementierung, welche jederzeit durch ein einfaches „Generate Setters and

Getters“ erzeugt werden kann. Diese Annotations verhindern jedoch, dass die Klassen für andere Teile der Diplomarbeit verwendet werden können, da diese Annotations nur vom JPA erkannt werden. Daher werden diese Annotations in eine „orm.xml“-Datei ausgelagert. Wenn die Annotations nicht mehr in der Klasse vorhanden sind, so wird diese Klasse zu einem POJO, welche auch an anderen Stellen dieser Diplomarbeit verwendet werden kann. Außerdem bietet die externe Datei eine bessere Übersicht und sichert die Konfiguration, selbst wenn sich das Datenbankmodell ändert.

7.3.1.2 orm.xml

In der „orm.xml“-Datei werden auch die Aufrufe von Stored Procedures festgelegt.

Beispielhaft wird hier das Mapping eines Benutzers aus der „orm.xml“-Datei herausgegriffen und genauer beschrieben:

```
<entity class="at.ac.htlperg.hiis.model.Benutzer">
<table name="Benutzer"/>
<named-stored-procedure-query name="login" procedure-name="login" >
<parameter class="java.lang.String" name="inBenutzer" mode="IN"/>
<parameter class="java.lang.String" name="inPasswort" mode="IN"/>
<parameter class="java.lang.Integer" name="outId" mode="OUT"/>
</named-stored-procedure-query>
<named-stored-procedure-query name="showBenutzerDaten" procedure-name="showBenutzerDaten">
<parameter class="java.lang.Integer" name="inBenutzer idBenutzer" mode="IN"/>
<result-class>at.ac.htlperg.hiis.model.Benutzer</result-class>
</named-stored-procedure-query>
<named-stored-procedure-query name="unlockUser" procedure-name="unlockUser">
<parameter class="java.lang.Integer" name="inCurrentUserId" mode="IN"/>
<parameter class="java.lang.Integer" name="inIdBenutzer" mode="IN"/>
<parameter class="java.lang.Boolean" name="outSuccess" mode="OUT"/>
</named-stored-procedure-query>
<named-stored-procedure-query name="showUsers" procedure-name="showUsers">
<parameter class="java.lang.Integer" name="inCurrentUserId" mode="IN"/>
<result-class>at.ac.htlperg.hiis.model.Benutzer</result-class>
</named-stored-procedure-query>
<attributes>
<id name="idBenutzer">
<generated-value strategy="IDENTITY"/>
</id>
<basic name="benutzer"/>
<basic name="istBlockiert"/>
<basic name="passwort"/>
<basic name="istRedakteur"/>
<one-to-many name="watchlists" target-entity="at.ac.htlperg.hiis.model.Watchlist" mapped-by="benutzer"/>
<transient name="watchlists"/>
</attributes>
</entity>
```

Abbildung 43: Der Inhalt der Datei "orm.xml"

Die ersten beiden Zeilen geben an, auf welche Klasse und welche Tabelle sich das Mapping bezieht. Danach folgen die Beschreibungen einer Stored Procedure. Diese geben zuerst den Namen der Prozedur an. Danach folgt die Beschreibung der einzelnen Parameter. Ein Parameter enthält zuerst den Datentyp, danach den Namen und abschließend, ob es sich um einen Eingabe- oder Ausgabeparameter handelt. Wenn eine Stored Procedure eine ganze Tabellenzeile oder auch mehrere Tabellenzeilen ausgibt so muss zusätzlich noch eine „result-class“ definiert werden.

Nach den Beschreibungen der Prozeduren folgt die Beschreibung der einzelnen Attribute einer Tabelle. Diese Beschreibung kann von den automatisch generierten Annotations in den Entitätsklassen übertragen werden.

7.3.1.3 Data Access Object (DAO)

Um eine Stored Procedure aufrufen zu können, wird ein DAO benötigt, die für dieses Diplomarbeitenprojekt wie folgt ausgestaltet ist:

```
@Named
@RequestScoped
public class BenutzerDao
{
    @PersistenceContext
    private EntityManager em;

    private Benutzer findById ( int id )
    {
        StoredProcedureQuery spq = this.em.createNamedStoredProcedureQuery ( "showBenutzerDaten" );
        spq.setParameter ( "inBenutzer_idBenutzer", id );
        spq.execute ();

        return (Benutzer) spq.getSingleResult ();
    }

    public Benutzer login ( String username, String password )
    {
        StoredProcedureQuery spq = this.em.createNamedStoredProcedureQuery ( "login" );
        spq.setParameter ( "inBenutzer", username );
        spq.setParameter ( "inPasswort", password );
        spq.execute ();
        int retId = (int) spq.getOutputParameterValue ( "outId" );

        return ( retId == 0 ? null : this.findById ( retId ) );
    }

    @SuppressWarnings ( "unchecked" )
    public List < Benutzer > getBenutzerForRedakteur ()
    {
        StoredProcedureQuery spq = this.em.createNamedStoredProcedureQuery ( "showUsers" );
        spq.setParameter ( "inCurrentUserId", JwtInputData.getId () );
        spq.execute ();

        return spq.getResultList ();
    }
}
```

Abbildung 44: Das Data Access Object für die Benutzerdaten

In den einzelnen Methoden werden die Stored Procedures aufgerufen. Dabei wird zuerst eine Abfrage erstellt. Für eine Abfrage wird der Name der Stored Procedure festgelegt und danach werden die einzelnen Importparameter gesetzt. Schlussendlich wird die Abfrage ausgeführt. Die Ausgabeparameter können nun von der Abfrage extrahiert werden. Sollte es sich um Ausgabeparameter handeln, so können diese mit dem Namen des Parameters extrahiert werden. Sollte die Stored Procedure jedoch als Rückgabewert eine Klasse haben, so können diese als Ergebnisliste extrahiert werden.

7.3.1.4 Representational State Transfer (REST)

Um die Funktionalitäten der DAOs über die REST-Schnittstelle erreichen zu können, müssen JAX-RS-Quellen erstellt werden. Diese bekommen eine Zielentität zugewiesen.

```

@Path ( "/benutzers" )
@Produces ( "application/json" )
@Consumes ( "application/json" )
public class BenutzerEndpoint
{
    @Inject
    private BenutzerDao dao;
    private DefaultReturn < Benutzer > defRet = new DefaultReturn < Benutzer > ();

    @GET
    @Path ( "/login" )
    public Response login ( @QueryParam ( "benutzer" ) String benutzer,
        @QueryParam ( "passwort" ) String passwort )
    {
        Benutzer b;

        if ( ( b = this.dao.login ( benutzer, passwort ) ) != null )
        {
            JwtHandler handler = new JwtHandler ();
            return this.defRet.defaultReturn ( b,
                handler.generateJWT ( b.getIdBenutzer (), ( byte ) b.getIstRedakteur () == 1 ?
                    true : false ) );
        }
        else
        {
            return this.defRet.defaultReturn ( 401 );
        }
    }

    @RequestScoped
    @Secured ( Role.Redakteur )
    @GET
    @Path ( "/allUsers" )
    public Response showAllUser ()
    {
        return this.defRet.defaultReturn ( this.dao.getBenutzerForRedakteur () );
    }
}

```

Abbildung 45: Der Endpoint für den Benutzerdatenzugriff

Die Annotations, welche vor der Klasse definiert wurden, gelten für alle Methoden der Klasse, sofern eine Methode selbst nicht diese Annotation ändert um sie anzupassen. Die wichtigsten Annotations sind, um welche HTTP-Methode es sich handelt, mit welchem Pfad diese aufgerufen werden und welchen MediaType es sich handelt.

7.3.1.5 JSON Web Token erstellen

```

public String generateJWT ( int id, boolean isRedakteur )
{
    JwtHandler.key = new SecretKey ();
    Date now = new Date ( System.currentTimeMillis () );

    JwtBuilder builder = Jwts.builder ().setHeaderParam ( "typ", "JWT" ).setId ( "" + id )
        .claim ( "isRedakteur", isRedakteur ).setIssuedAt ( now )
        .signWith ( SignatureAlgorithm.HS512, JwtHandler.key.getKey () )
        .setExpiration ( new Date ( System.currentTimeMillis () + 18000000 ) );

    return builder.compact ();
}

```

Abbildung 46: Die Erstellung eines JSON Web Tokens

Im Auszug oben ist veranschaulicht, wie der JSON-Web-Token erstellt wurde. Als Parameter erwartet diese Methode zwei Werte: eine id und einen Boolean. Die id ist jene, welche dem aktuellen Benutzer zugewiesen ist und der Boolean gibt an, ob es sich beim Benutzer um einen Redakteur handelt oder nicht. Diese Informationen werden zusammen mit anderen Standardinformationen in einen JWT-Header zusammengepackt und als einfacher String zurückgegeben.

7.3.1.6 JSON Web Token verifizieren

```

@Override
public void filter ( ContainerRequestContext requestContext ) throws IOException
{
    String authHeader = requestContext.getHeaderString ( HttpHeaders.AUTHORIZATION );

    if ( authHeader != null && authHeader.startsWith ( "Bearer" ) )
    {
        try
        {
            String token = authHeader.substring ( "Bearer ".length () ).trim ();
            Claims claim = this.handler.validateMyAuth ( token );
            boolean isRedakteur = this.handler.getIstRedakteur ( claim );
            JwtInputData inputData = new JwtInputData ( this.handler.getUserId ( claim ),
                isRedakteur ? Role.Redakteur : Role.User );

            requestContext.setSecurityContext ( inputData );

            Method resourceMethod = this.resourceInfo.getResourceMethod ();
            List < Role > methodRoles = extractRoles ( resourceMethod );

            if ( methodRoles.isEmpty () )
            {
                Class < ? > resourceClass = this.resourceInfo.getResourceClass ();
                List < Role > classRoles = extractRoles ( resourceClass );

                if ( ! classRoles.isEmpty () )
                {
                    this.handler.checkPermission ( classRoles, token );
                }
            }
            else
            {
                this.handler.checkPermission ( methodRoles, token );
            }
        }
        catch ( SignatureException e )
        {
            // XXX syserr signature validation failed
            System.err.println ( e.getMessage () );
            requestContext.abortWith (
                Response.status ( Response.Status.UNAUTHORIZED ).entity ( "Sign wrong!" ).build () );
        }
        catch ( ExpiredJwtException e )
        {
            // XXX syserr toke expired
            System.err.println ( e.getMessage () );
            requestContext.abortWith (
                Response.status ( Response.Status.UNAUTHORIZED ).entity ( "Time expired" ).build () );
        }
        catch ( WrongRoleException e )
        {
            // XXX syserr wrongrole
            System.err.println ( e.getMessage () );
            requestContext.abortWith (
                Response.status ( Response.Status.UNAUTHORIZED ).entity ( "Time expired" ).build () );
        }
    }
    else
    {
        // XXX syserr not valid
        System.err.println ( "Not valid" );
        requestContext
            .abortWith ( Response.status ( Response.Status.UNAUTHORIZED ).entity ( "No sign!" ).build () )
    }
}

```

Abbildung 47: Die Verifizierung eines JSON Web Tokens

Um sicherzustellen, dass eine Anfrage nur von einem eingeloggten Benutzer kommt muss der Client den beim Login übergebenen JWT wieder zurück an den Server geben. Das Anmelde-Procedere ist in der Abbildung oben nachvollziehbar. Der Server liest diesen ein und entscheidet anhand des Aufbaus des Headers, ob dieser gültig ist und dieser tatsächlich vom Server erstellt wurde. Ist dem nicht der Fall, so bekommt der Client einen Unauthorized-Error zurück. Sollte der Header jedoch in Ordnung sein, so erkennt der Server diesen auch als gültig und erlaubt den Zugriff auf den Server. Weiters prüft der Filter ab, ob der Benutzer auch die entsprechenden Rechte am Server hat, um das gewünschte Interface aufzurufen. Sollte der Benutzer die erforderlichen Berechtigungen besitzen, wird die Anfrage weitergeleitet und der Client bekommt das Ergebnis der Anfrage zurückgesandt. Sollte die Berechtigung des Clients nicht ausreichen, so bekommt dieser einen Fehler, dass er nicht über ausreichende Berechtigungen verfügt.

7.3.2 Webseite

Bevor mit der Umsetzung der Webseite gestartet werden konnte musste sich das Projektteam einarbeiten, wie die technologischen Alternativen funktionieren. Es wurden mehrere Testseiten in Angular2 bzw. Typescript erstellt, um verschiedene Designs zu prüfen bzw. um eines zu finden, das passt. Außerdem wurde am Beginn statt WebStorm Visual Studio Code verwendet, da uns das im Programmierunterricht vermittelt wurde und wir uns schon damit arrangiert haben. Allerdings stellten wir bald auf WebStorm um, da diese Entwicklungsumgebung einige sehr hilfreiche Features bietet, welche Visual Studio Code nicht unterstützt (z.B. Code completion).

Designtechnisch wurde dann Bootstrap für Angular2 bzw. Typescript verwendet, da dieses einige sehr schön designte Vorlagen mitbrachte und so das Aussehen der Webseite mit deutlich weniger Aufwand verbesserte. Zusätzlich wurde SweetAlert2 verwendet um verschiedene Meldungen, wie zum Beispiel beim Hinzufügen eines Kommentars oder beim Entfernen eines Impulses aus der Watchlist, mit einem ansprechenden Design anzeigen zu können. Die Webseite besteht aus mehreren HTML-Files, welche alle mit so genannten Components verknüpft sind, wie im Folgenden zu sehen ist.

```

<div class="col-md-6 col-md-offset-3">
  <h2>Login</h2>
  <form name="form" (ngSubmit)="f.form.valid && login()" #f="ngForm" novalidate>
    <div class="form-group" [ngClass]="{ 'has-error': f.submitted && !username.valid }">
      <label for="username">Benutzername</label>
      <input type="text" class="form-control" name="username" [(ngModel)]="model.username" #username="ngModel" required />
      <div *ngIf="f.submitted && !username.valid" class="help-block">Benutzername muss eingegeben werden!</div>
    </div>
    <div class="form-group" [ngClass]="{ 'has-error': f.submitted && !password.valid }">
      <label for="password">Passwort</label>
      <input type="password" class="form-control" name="password" [(ngModel)]="model.password" #password="ngModel" required />
      <div *ngIf="f.submitted && !password.valid" class="help-block">Passwort muss eingegeben werden!</div>
    </div>
    <div class="form-group">
      <button [disabled]="loading" class="btn btn-primary">Login</button>
      
    </div>
    <div *ngIf="hasError" class="alert alert-danger">{{error}}</div>
  </form>
</div>

```

Abbildung 48: Login HTML

Im obigen Bild sieht man zum Beispiel den HTML-File der für die Login-Seite erstellt wurde.

```
@Component({
  selector: 'login',
  templateUrl: './login.component.html',
  styleUrls: ['./styles.css']
})

export class LoginComponent implements OnInit {

  model: any = {};

  public loading = false;

  public error = '';

  hasError: boolean = false;

  private timer;

  constructor(private loginService: LoginService, private router: Router, private errorMessage: ErrorMessageService){
    errorMessage.error$.subscribe(error => {
      this.hasError = true;
      this.error = error;

      if(this.timer!=null)
      {
        clearTimeout(this.timer);
      }
      this.timer = setTimeout(() => {
        this.hasError = false;
      }, 4000);
    });
  }

  ngOnInit(): void{
    this.loginService.logout();
  }

  login():void{
    this.loading = true;

    console.log(this.loginService.login(this.model.username, this.model.password))
    this.loginService.login(this.model.username, this.model.password)
      .subscribe(result => {

        console.log("Result: " + result);

        if(result === true){
          this.router.navigate(['./ideas']);
        }

      },error=>{
        this.loading = false;
        this.errorMessage.error("Wrong username or password!");
      });
  }
}
```

Abbildung 49: Login Component, ruft den Service auf

Im obigen Bild sieht man die zum Login gehörigen Component. Eine Component beinhaltet verschiedene Methoden die vom HTML-File aufgerufen werden. Außerdem wird von einem Component aus der dazugehörige Service aufgerufen, welches den Request an den Server sendet, falls Daten benötigt werden.

```

@Injectable()
export class LoginService{

    public url = 'https://hiis.htl-perg.ac.at/hiisWebserver/rest/benutzers/login';

    public token: string;
    public id: number;
    private watchlistService: WatchlistService;

    constructor(private http:Http, private router: Router){
        var currentUser = JSON.parse(localStorage.getItem('currentUser'));
        this.token = currentUser && currentUser.token;
    }

    login(username: string, password: string):Observable<boolean>{
        //console.log("LoginService " + username + ", " + password);
        return this.http.get(this.url + '?benutzer=' + username + '&password=' + password)
            .map((response: Response) => {

                console.log(response.status);

                let token = response.headers.get('JWT');
                let jsonObj = JSON.parse(response.text());

                if(token != null){
                    this.token = token;
                    //this.id = jsonObj['idBenutzer'];

                    localStorage.setItem('currentUser', JSON.stringify({id: jsonObj['idBenutzer'], token: token}));
                }
                return true;
            })
            .catch(this.handleError);
    }

    private handleError(error:any){
        return Observable.throw(false);
    }

    logout(){
        this.token = null;

        localStorage.removeItem('currentUser');
    }
}

```

Abbildung 50: Login Service, ruft die Request auf

Ein Service könnte zum Beispiel so wie in der obigen Darstellung aussehen. Da das hier die Code-Files für den Login sind wird in dem dazugehörigen Service ein Get-Request an den Server gesendet. Mitgesendet werden, wie bei allen anderen Request auch, ein JWT-Header, welcher zur Authentifizierung dient sodass der Server weiß, dass der eingeloggte Benutzer das Recht hat auf die Daten zuzugreifen.

Dieses oben beschriebene Konzept ist beim Programmieren mit Angular2 und Typescript Standard. Jede einzelne Seite, die dargestellt werden soll, benötigt eine Component und einen HTML-File. Aufgerufen wird eigentlich nur die index.html Datei. Durch Routing wird dann direkt zur ersten Seite weitergeleitet. Die im Folgenden gelb markierten „paths“ geben den Pfad an, der bspw. beim Klick auf einen Link aufgerufen wird.

```
@NgModule({
  declarations: [
    AppComponent,
    IdeasComponent,
    IdeaDetailComponent,
    LoginComponent,
    WatchlistComponent,
    AddImpuls,
    HomeComponent,
    AllUsersComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,

    RouterModule.forRoot([
      {
        path: '/',
        canActivate: [LoginGuard],
        component: HomeComponent
      },
      {
        path: '/home',
        component: HomeComponent
      },
      {
        path: '/ideas',
        component: IdeasComponent
      },
      {
        path: '/detail/:id',
        component: IdeaDetailComponent
      },
      {
        path: '/watchlist',
        component: WatchlistComponent
      },
      {
        path: '/addImpuls',
        component: AddImpuls
      },
      {
        path: '/login',
        component: LoginComponent
      },
      {
        path: '/allUsers',
        component: AllUsersComponent
      }
    ])
  ]
})
```

Abbildung 51: Projekt Module, kümmert sich um die einzelnen Weiterleitungen, zwischen den Seiten

Durch die Angaben der „paths“ werden die Routings gemanaged. Man muss dann nur beim Weiterleiten auf eine neue Seite den path unter einfache Hochkomma setzen.

```
if(result === true){
  this.router.navigate(['./ideas']);
}
```

Abbildung 52: Beispiel für ein Routing (Weiterleitung)

7.3.3 Android Applikation

7.3.3.1 Android Manifest

Die AndroidManifest.xml-Datei, die in der Abbildung unten nachzulesen ist, muss im Wurzelverzeichnis des Projektes sein. Sie beinhaltet wichtige Informationen über die Applikation für das Android Betriebssystem welche benötigt werden um die App zu starten.

Wichtige Funktionen dieser Datei sind zum Beispiel:

- Den Namen des Java Packages für die Anwendung zu definieren
- Definieren der Komponenten, unter anderem der Activities und der Services
- Es definiert, welche Berechtigungen die Anwendung haben muss
- Es deklariert die mindestens erforderliche Softwareversion von Android

(vgl. developer.android.com, 2017)

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="at.app.hiis.hiis_app">

<uses-permission android:name="android.permission.INTERNET" />

<application
    android:name=".AppController"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".LoginActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name=".MainActivity" />
    <activity android:name=".CreateImpulsActivity" />
    <meta-data
        android:name="com.google.android.gms.version"
        android:value="@integer/google_play_services_version" />
    <activity android:name=".DetailActivity"></activity>
</application>
</manifest>
```

Abbildung 53: Die Android Manifest Datei

7.3.4 HTTPS mit Volley

7.3.4.1 Einbinden der Bibliothek

Volley ist eine Bibliothek welche http- und https-Aufrufe einheitlich synchronisiert und verwaltet. Erster Schritt hierfür ist das Einbinden der notwendigen Bibliothek. In der build.gradle-Datei werden unter „dependencies“ die Abhängigkeiten eingefügt, so wie hier dargestellt.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.2.0'
    compile 'com.android.volley:volley:1.0.0'
}
```

Abbildung 54: Die Dependencies im Android Manifest

7.3.4.2 Erstellen einer Volley Singleton Class

Einer der besten Wege um die Volley-Objekte zu verwalten ist das Erstellen einer RequestQueue, indem man eine globale Klasse erstellt, so wie wir das hier gemacht haben.

```
public class LruBitmapCache extends LruCache<String, Bitmap> implements
    ImageCache {
    public static int getDefaultLruCacheSize() {
        final int maxMemory = (int) (Runtime.getRuntime().maxMemory() / 1024);
        final int cacheSize = maxMemory / 8;

        return cacheSize;
    }

    public LruBitmapCache() {
        this(getDefaultLruCacheSize());
    }

    public LruBitmapCache(int sizeInKiloBytes) {
        super(sizeInKiloBytes);
    }

    @Override
    protected int sizeOf(String key, Bitmap value) {
        return value.getRowBytes() * value.getHeight() / 1024;
    }

    @Override
    public Bitmap getBitmap(String url) {
        return get(url);
    }

    @Override
    public void putBitmap(String url, Bitmap bitmap) {
        put(url, bitmap);
    }
}
```

Abbildung 55: Definition der Klasse LruBitmapCache

Als Nächstes benötigt man einen Controller, welcher auf die eben erstellte Queue zugreift. Dazu erstellt man einen ApplicationController, wie jenen in der folgenden Abbildung.

```

public class ApplicationController extends Application {

    public static final String TAG = ApplicationController.class
        .getSimpleName();

    private RequestQueue mRequestQueue;
    private ImageLoader mImageLoader;

    private static ApplicationController mInstance;

    @Override
    public void onCreate() {
        super.onCreate();
        mInstance = this;
    }

    public static synchronized ApplicationController getInstance() {
        return mInstance;
    }

    public RequestQueue getRequestQueue() {
        if (mRequestQueue == null) {
            mRequestQueue = Volley.newRequestQueue(getApplicationContext());
        }
        return mRequestQueue;
    }

    public <T> void addToRequestQueue(Request<T> req, String tag) {
        // set the default tag if tag is empty
        // req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
        req.setTag("Tag");
        getRequestQueue().add(req);
    }

    public <T> void addToRequestQueue(Request<T> req) {
        req.setTag(TAG);
        getRequestQueue().add(req);
    }
}

```

Abbildung 56: ApplicationController Klasse

Zuletzt fügt man nun im AndroidManifest.xml im <application> tag die „INTERNET“ permission hinzu um Zugriff auf das Internet zu bekommen:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Abbildung 57: Internet Permissions im Android Manifest

7.3.4.3 Erstellen einer JSON Anfrage

Mithilfe von Volley kann man auch JSON-Anfragen versenden, was in der folgenden Abbildung zu sehen ist.

Es gibt zwei Arten von JSON Anfragen für Volley:

- **JSONArrayRequest**
Ein JSONArrayRequest erwartet nicht nur ein Objekt, sondern eine Menge an Objekten.
- **JsonObjectRequest**
Bei einem JsonObjectRequest wird eine Anfrage gesendet bei der das Objekt, welches man erhalten hat, nur ein einzelnes sein darf.

(vgl. www.androidhive.info, 2017)

```
// Tag used to cancel the request
String tag_json_obj = "json_obj_req";

String url = „http://„ +
getResources().getString(R.string.url)+"/hiisWebserver/rest/vwimpulsovervie
ws/impuls";

ProgressDialog pDialog = new ProgressDialog(this);
pDialog.setMessage("Loading...");
pDialog.show();

    JsonObjectRequest jsonObjReq = new JsonObjectRequest(Method.GET,
        url, null,
        new Response.Listener<JSONObject>() {

            @Override
            public void onResponse(JSONObject response) {
                Log.d(TAG, response.toString());
                pDialog.hide();
            }
        }, new Response.ErrorListener() {

            @Override
            public void onErrorResponse(VolleyError error) {
                VolleyLog.d(TAG, "Error: " + error.getMessage());
                // hide the progress dialog
                pDialog.hide();
            }
        }
    });

// Adding request to request queue
AppController.getInstance().addToRequestQueue(jsonObjReq, tag_json_obj);
```

Abbildung 58: Erstellen eines Volley Requests

8 Projektvorgehen

In diesem Kapitel ist das Vorgehen im Rahmen dieser Diplomarbeit festgehalten. Dabei handelt es sich um verbindlich abgestimmte Pläne zwischen dem Projektauftraggeber und dem Projektteam, die die inhaltlichen und terminlichen Rahmenbedingungen darstellen.

8.1 Planung

8.1.1 Meilensteine

05.11.2016	Projektinitialisierung Kick-off-Meeting abhalten Zeitplanung vereinbaren Verteilung der Aufgaben vornehmen Inhaltliche Anforderungen definieren
16.12.2016	Entwurf Entwicklungsumgebungen einrichten Webseite entwerfen Applikation entwerfen
08.01.2017	Datenmodell Datenmodell fertigstellen Testdaten erstellen Datenbank testen
24.02.2017	Prototyp Dezentrale Anwendungen bereitstellen (nur mit Grundfunktionalitäten)
29.03.2017	Implementierung Funktionale Anforderungen fertigstellen (Beginn der Übergabe)
05.04.2017	Diplomarbeit Wissenschaftlichen Arbeit abgeben

Tabelle 14: Meilensteine

8.1.2 Zeitplanung

Für die Zeitplanung wurde die in Microsoft Excel vorhandene Vorlage für Gantt-Projektpläne verwendet. Die verschiedenen Aktivitäten sind entlang der Zeilen eingetragen. Die Spalten repräsentieren die Zeiträume. Die Dauer eines Zeitraumes beträgt drei Wochen. Das Projektteam hat im Voraus den Soll-Beginn und die Soll-Dauer eingetragen. Im Verlauf des Projektes wurde zu jedem Punkt der Ist-Beginn und die Ist-Dauer hinzugefügt.

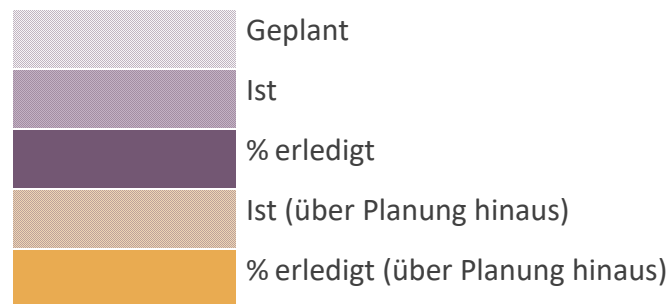
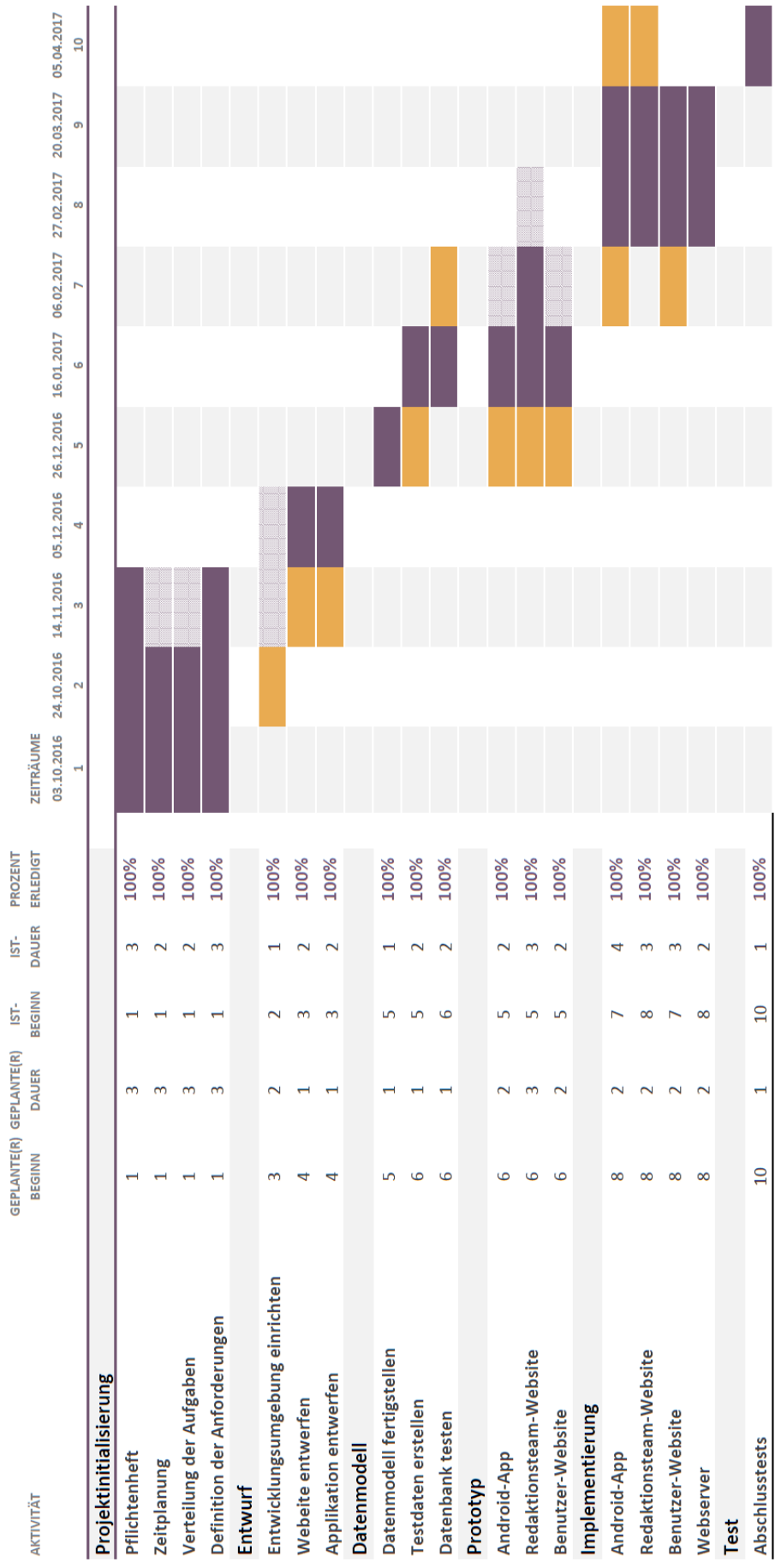


Abbildung 59: Gantt-Zeitplan Legende



9 Verwendete Ressourcen

9.1 Hardware

9.1.1 Server

Die HTL Perg hat dem Projektteam für diesen Auftrag ein Server zur Verfügung gestellt, damit die vereinbarten Inhalte erarbeitet werden konnten. Darauf hat ein Projektteammitglied einen Ubuntu Server in der Version 16.04 LTS (Long Term Support) aufgesetzt und ein anderes aufgrund seines Aufgabengebietes die Installation und Konfiguration des Wildfly-Applikationsservers übernommen. Der Zugriff auf diesen Server fand meist durch SSH (Secure Shell) statt.

9.2 Software

9.2.1 Eclipse

Eclipse ist eine plattformübergreifende IDE (Integrated Development Environment), welche in



Abbildung 60: Eclipse-Logo (eclipse.org, 2017)

der Diplomarbeit zur Entwicklung des REST-Applikationsservers verwendet wurde. Einer der größten Vorteile von Eclipse liegt im offenen Quelltext. Durch diese Transparenz der Eclipse Foundation (den Entwicklern von Eclipse) wird es jeder Person ermöglicht, Erweiterungen für die Software zu entwickeln und diese mit anderen Menschen zu teilen. (vgl. wikipedia.org, 2017)

Weiters zeichnet sich Eclipse durch seine standardmäßige Funktionsvielfalt aus. Diese umfassen einfache Aufgaben wie das Verwalten der Projektstruktur über eine Integration von Git bis hin zum automatischen Generieren von Code-Teilen. Zur Erleichterung der Entwicklung wurden die JBoss-Tools als Plug-in installiert. Auf dieses Plug-in wird im Folgenden noch genauer eingegangen.

9.2.2 JBoss-Tools

Die JBoss-Tools sind eine Sammlung von Eclipse Plug-ins und Funktionalitäten. Diese wurden primär für JBoss und JavaEE Entwickler erstellt. (vgl. wikipedia.org, 2017)



Abbildung 61: JBoss Tools Logo (design.jboss.org, 2017)

Das JBoss Plug-in bietet im Vergleich zu ähnlichen Lösungen vor allem zwei Hauptvorteile. Einerseits erleichtert es das Management des Servers und andererseits bietet das Plug-in zahlreiche Erweiterungen, welche für die Entwicklung einer REST-Schnittstelle essentiell sind. Dazu zählt vor allem die Möglichkeit, JAX-RS Quellen anzugeben. Auf diese wird im Verlauf der Diplomarbeit noch genauer eingegangen.

9.2.3 BlueGriffon

BlueGriffon ist ein Tool zur einfachen Gestaltung und Bearbeitung von Webseiten. Das Tool baut auf der Mozilla Application Suite auf und verwendet Gecko, die Rendering-Engine des Firefox.

BlueGriffon kompiliert im W3C Webstandard und bietet Unterstützung für HTML 4, XHTML 1.1, HTML 5 und XHTML 5.

Wir haben das Tool in der Diplomarbeit zur Erstellung des ersten funktionalen Designs verwendet. Dieses Design diente als grobe Grundlage für die Implementierung. (vgl. wikipedia.org, 2017)



Abbildung 62: BlueGriffon Logo (wikimedia.org, 2017)

BlueGriffon haben wir deshalb als Tool unserer Wahl genommen, weil es intuitiv und einfach zu benutzen ist.

9.2.4 WildFly

Als Applikationsserver wurde ein Wildfly-Server verwendet. Dieser ist in Java nach dem Java-EE-Standard implementiert und bietet somit eine gute Plattformunabhängigkeit. Dabei steht der Applikationsserver unter der GNU Lesser General Public License (LGPL). Dies bedeutet, dass der Server inklusive des Quelltextes kostenfrei heruntergeladen werden kann.



Abbildung 63: WildFly Logo (design.jboss.org, 2017)

Weiters bietet der Applikationsserver eine übersichtliche und logisch strukturierte Konfigurationsoberfläche in Form einer Webseite welche mit dem Server mitgeliefert wird. In dieser Oberfläche kann eine Datenbankverbindung eingerichtet werden. Um nun das Projekt auf dem Server zu aktivieren muss dies zuerst aus der Entwicklungsumgebung exportiert und anschließend auf den Server hochgeladen werden.
(vgl. de.wikipedia.org, 2017)

9.2.5 MySQL-Datenbank

MySQL ist eines der weltweit verbreitetsten relationalen Datenbankverwaltungssysteme. Es ist als Open-Source-Software sowie als kommerzielle Enterpriseversion für verschiedene Betriebssysteme verfügbar und bildet die Grundlage für viele dynamische Webauftritte. (de.wikipedia.org, 2017)



Abbildung 64: MySQL-Logo (de.wikipedia.org, 2017)

Aus eben diesem Grund haben wir MySQL als Datenspeicherungseinheit ausgewählt. Alle Daten, die wir speichern, haben Beziehungen und Referenzen untereinander, damit sie sinnvoll korrekt sind. Auch die Zuordnung von Benutzern zu Impulsen, Dateien und Kommentaren ist ein Grund, ein relationales Datenbankmanagementsystem wie MySQL zu verwenden.

9.2.6 MySQL Workbench

MySQL Workbench ist ein vereintes visuelles Werkzeug für Datenbankarchitekten, -entwickler und Datenbankadministratoren. Dieses Tool stellt Möglichkeiten zur Datenmodellierung, SQL-Entwicklung und umfangreiche Administrationswerkzeuge für die Serverkonfiguration, Benutzeradministration, Backup und noch viel mehr bereit. Es funktioniert plattformübergreifend auf Windows, Linux und Max OS X. (vgl. mysql.de, 2017)



Abbildung 65: MySQL-Workbench-Logo (quintetsolutions.com, 2017)

Aufgrund dessen, dass wir eine MySQL-Datenbank zum Speichern aller relevanten Daten gewählt haben, bot sich die MySQL-Workbench als Administrations- und Entwicklungstool an. Da diese Software aus dem gleichen Haus stammt wie die MySQL-Datenbank selbst, ist sichergestellt, dass diese miteinander funktionieren und keine ungewöhnlichen Probleme machen.

9.2.7 Android Studio

Android Studio bietet einige hilfreiche Tools für die Entwicklung von Apps für alle Geräte, welche auf Android basieren. Es bietet die Möglichkeit, den Code direkt am Android Gerät auszuführen oder einen Emulator, um es lokal am Rechner ausführen zu können. Ein weiteres Feature ist ein Code Editor, welcher Code completion und die Analyse des Programmcodes automatisch erfüllt. Weiters erlaubt es Codebeispiele von GitHub herunterzuladen um eigene Projekte zu erweitern. (vgl. developer.android.com, 2017)



Abbildung 66: Android Studio Logo (technicolit.com, 2017)

Aufgrund dessen, dass beispielsweise das Eclipse-Plugin für Android nicht mehr aktiv weiterentwickelt wird und Google auf diese Entwicklungsumgebung in Zukunft baut, haben wir uns für das Android Studio entschieden.

9.2.8 WebStorm

WebStorm ist eine Integrierte Entwicklungsumgebung (IDE). Es wurde von der Firma JetBrains entwickelt und unterstützt die Programmiersprachen JavaScript, HTML5, Node.js, Bootstrap, AngularJS, TypeScript und einige weitere. In meinem Fall habe ich WebStorm deshalb verwendet, weil es eine angenehme und übersichtliche Entwicklungsumgebung bietet. Außerdem unterstützt es Code completion und einige andere Features welche von VisualStudio Code nicht unterstützt wurden. WebStorm basiert auf IntelliJ IDEA, welche auch von JetBrains kommt, es stellt jedoch die JavaScript-spezialisierte Version dar, was ein weiterer Grund ist weshalb ich es verwendet habe. (vgl. wikipedia.org, 2017)



Abbildung 67: WebStorm Logo (worldvectorlogo.com, 2017)

9.2.9 Google Chrome Webbrowser

Google Chrome ist ein Webbrowser der Firma Google.

Chrome hat wie die anderen Webbrowser Tabs mit denen die Inhalte dargestellt werden können. Vor einigen Jahren war Google Chrome bei weitem schneller als seine Konkurrenz. Heutzutage ist dieser Vorsprung allerdings etwas verloren gegangen. Einen wichtigen Vorteil hat der Browser jedoch, weshalb ich ihn vor allen anderen

gegenüber bevorzugt habe. Google Chrome hat einen implementierten Adobe Reader und einen

Adobe Flash Player was den Umgang um Einiges vereinfacht. Außerdem finde ich persönlich sein Design ansprechender weshalb er auch der Standardbrowser ist.

(vgl. wikipedia.org, 2017)



Abbildung 68: Google Chrome Webbrowser Logo
(logobird.com, 2017)

9.2.10 Postman

Postman ist ein REST Client mit dem unter anderem Requests an einen Server gesendet werden können um Daten anzufordern. Wir haben ihn verwendet um zu prüfen, ob die REST Requests, die von der App und der Webseite auf den Server gesendet wurden, richtig funktionieren. Um zu vermeiden, dass

wir den Request direkt von der Webseite bzw. der App das erste Mal aufrufen müssen und dadurch zusätzliche Fehler riskieren, haben wir Postman verwendet. Außerdem kann er als Browser verwendet werden.



Abbildung 69: Postman Logo
(mwtestconsultancy.co.uk, 2017)

9.2.11 Visual Studio Code

Visual Studio Code dient vorrangig, ähnlich wie WebStorm, zur Entwicklung von Webanwendungen. Es unterstützt Programmiersprachen wie HTML, CSS, Java, JavaScript, etc.

Verwendet wurde es deshalb, weil wir es im

Programmierunterricht kennengelernt haben und wir uns damit auf einen Test vorbereitet haben. Der Umstieg zu

WebStorm erfolgte allerdings relativ schnell, da wir merkten, dass WebStorm einige Features mehr mitbringt im Gegensatz zu Visual Studio Code. Unter anderem eine übersichtlichere Entwicklungsumgebung und Code completion. (vgl. wikipedia.org, 2017)

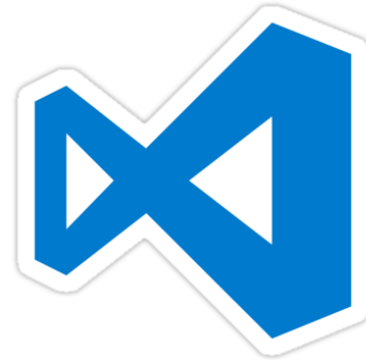


Abbildung 70: Visual Studio Code Logo (redbubble.com, 2017)

9.2.12 Advanced REST Client

Der Advanced Rest Client ist ein einfach strukturierter REST Client mit zahlreichen Funktionen und Features, wie REST-Aufrufe, aber das Tool bietet auch die Möglichkeit, sich auf einen Socket zu verbinden um diesen zu testen.

Vergleicht man nun den Advanced Rest Client mit Postman, so fällt auf, dass es keine gravierenden funktionalen Unterschiede gibt, jedoch bestehen Unterschiede im Aufbau der Benutzeroberfläche.

Im Projekt wurde der Advanced Rest Client zum Testen des REST Interfaces verwendet um die Korrektheit aller Funktionen sicherzustellen.

Aufgrund der Empfehlung eines Klassenkollegen haben wir neben Postman auch den Advanced Rest Client verwendet.



Abbildung 71: Advanced REST Client Logo (developer.myob.com, 2017)

9.2.13 Ubuntu

Ubuntu ist ein auf Debian basierendes Linux-Betriebssystem, welches auf PC's, Tablets und Smartphones installiert werden kann.



Abbildung 72: Ubuntu Logo (wikimedia.org, 2017)

Zusätzlich zu den angeführten Plattformen wird Ubuntu vor allem auf Servern verwendet. Hier zeichnet es sich vor allem durch den geringen Ressourcenverbrauch und die Sicherheit gegen Angriffe aus.

Ubuntu ist ein kostenloses Betriebssystem welches unter der GNU General Public License veröffentlicht wird. Es wird von Canonical veröffentlicht und gleichzeitig bietet Canonical auch kommerzielle Hilfestellung bei Problemen mit dem Betriebssystem an. (vgl. wikipedia.org, 2017)

9.3 Architekturen

9.3.1 HTTP-Header

Die HTTP-Header werden direkt nach der Request- oder Response-Zeile, welche sich in der ersten Zeile befinden, transportiert. Die einzelnen Header werden mit Doppelpunkten voneinander getrennt und bestehen aus einem Namen-Wert-Paar das in Klartext formatiert ist. Das Ende der Header wird mit einem leeren Feld gekennzeichnet.

Aus historischen Gründen wurden zu lange Header in mehrere Zeilen aufgeteilt. Ein zusammenhängender Header wird durch das Vorhandensein eines Leerraumes oder eines horizontalen Tabs als erstes Zeichen in der nächsten Zeile angezeigt. Einige Header, wie etwa der "Authorization"-Header, werden von der IETF standardisiert um die Interoperabilität zwischen verschiedenen Anwendungen zu erleichtern. (vgl. wikipedia.org, 2017)

9.3.2 REST

REST (Representational state transfer) oder RESTful Webservices bezeichnen eine Möglichkeit des Datentransfers zwischen Computersystemen über das Internet. Dabei kodiert REST nur den URI-Ort und Namen einer Ressource, nicht aber die Funktionalität welche der Web-Dienst anbietet.

Im Gegensatz zu anderen, ähnlichen Verfahren wie SOAP und WSDL, bietet REST den Vorteil, dass bereits ein Großteil der benötigten Infrastruktur im WWW implementiert und vorhanden ist. Dies bedeutet, dass eine einfache, statische Webseite bereits REST-konform ist. Bei dynamischen Seiten ist dies oft nicht der Fall, da bei diesen Seiten die automatisierte Verarbeitung der Informationen meist nicht möglich ist. Dies liegt an den unterschiedlichen Formaten und Inhalten, welche angezeigt werden sollten. Wenn das Format gleichbleiben würde, so wäre eine wichtige REST-Eigenschaft gegeben. (vgl. wikipedia.org, 2017)

9.3.3 Lightweight Directory Access Protocol (LDAP)

LDAP bezeichnet einen Industriestandard, welcher die Abfrage und Veränderung von verteilten Verzeichnisdiensten über ein IP-Netzwerk ermöglicht. Dabei können Informationen über Benutzer, Systeme, Netzwerke, Dienste und Anwendungen abgefragt werden.

Einer der Einsatzgebiete von LDAP liegt in der zentralen Userverwaltung. Dies bedeutet, dass alle Benutzerdaten in einer Datenbank gespeichert werden und alle Anwendungen über LDAP auf diese Datenbank zugreifen können, um die Authentifizierung von Benutzern zu ermöglichen. (vgl. wikipedia.org, 2017)

9.3.4 ORM

ORM (Object Relational Mapping) bezeichnet die Persistierung von Daten aus einem objektorientierten Programm in eine relationale Datenbank.

Würde man versuchen, Objekte direkt in eine relationale Datenbank zu speichern, so würde dies unweigerlich zu einem ORIM führen. Ein ORIM (Object Relational Impedance Mismatch) beschreibt den Konflikt zwischen der primären Identifikation eines Eintrages. In einem objektorientierten Programm wird ein Objekt eindeutig durch die Speicheradresse identifiziert wohingegen in einer relationalen Datenbank ein Datensatz eindeutig durch seinen primären Schlüssel identifiziert wird. Daher wird ORM verwendet um dies zu verhindern. Dabei verknüpft ORM die Werte aus einem Objekt mit den Werten einer Entität.

9.3.5 Java Persistence API

Die Java Persistence API bietet eine Schnittstelle zwischen Objekten in Java und einer Datenbank. Diese kümmert sich um den gesamten ORM-Prozess. (vgl. wikipedia.org, 2017)

Um eine Verbindung zwischen dem Java-Projekt und einer Datenbank herzustellen, wird zuerst im Applikationsserver eine Datenbankverbindung erstellt. Während dem Hinzufügen wird eine JNDI-Zeichenkette erstellt. Diese Zeichenkette muss man anschließend in der persistence.xml im „jta-data-source“ Element angeben. Dies ermöglicht den Zugriff der Anwendung auf die Datenbank.

9.3.6 JAX-RS

Bei JAX-RS (Java API for RESTful Web Services) handelt es sich um ein API, welches das Erstellen von Webservices nach dem REST Standard in Java erleichtert. Dabei wird der Großteil der REST-Funktionalitäten mittels Annotations geregelt. Weiters wird eine eigene spezielle Java-Klasse benötigt, welche die gesamte REST Schnittstelle erst ermöglicht. (vgl. wikipedia.org, 2017)

9.4 Libraries

9.4.1 Volley Library

Volley ist eine Open Source HTTP Bibliothek, welche es Android Apps vereinfacht, Zugriffe über das Internet schneller und einfacher zu erledigen. Mithilfe dieser Technologie können HTTP Transaktionen einheitlich priorisiert werden. Zusätzlich werden alle Anfragen in einem Cache zwischengespeichert. Dadurch müssen die Daten zum Beispiel beim Drehen des Bildschirms nicht neu geladen werden. Der Sourcecode von Volley ist auf GitHub frei zugänglich. (vgl. code.tutsplus.com, 2017)

9.4.2 JSON Web Token

Bei einem JSON-Web-Token handelt es sich um eine Authentifizierung mittels eines Tokens, welcher sich an den JSON-Standard anlehnt. Dieser wird im RFC7519 ausführlich beschrieben. (vgl. wikipedia.org, 2017)

Der JSON-Web-Token wurde platzsparend konzipiert, um in einem HTTP-Authorization-Header mitgesendet werden zu können. Dabei ist der Token wie ein JSON-Objekt aufgebaut und strukturiert.

Der Token unterteilt sich hierbei in drei Teile:

- dem Header
- der Payload und
- der Signatur

Der Header beinhaltet Metadaten über den Token an sich. Dies beschränkt sich meist auf den verwendeten Hashalgorithmus und auf die genaue Typbezeichnung des Tokens. Im Payloadbereich sind die gesamten Nutzdaten des Tokens als Claims gespeichert. Diese können beim Erstellen des Tokens frei gewählt werden. Zur vereinfachten Handhabung stehen jedoch einige vordefinierte Claims zur Verfügung.

Die Berechnung der Signatur gestaltet sich jedoch etwas aufwändiger. Dazu werden zuerst der Header und die Payload einzeln mithilfe des Base64url-Algorithmus kodiert.

Anschließend wird aus dem kodierten Header und der kodierten Payload ein MAC mit dem HMAC SHA-256-Algorithmus errechnet. Diese drei Teiltoken werden anschließend mit einem Punkt verbunden.

(vgl. tools.ietf.org, 2017)

Encoded

PASTE A TOKEN HERE

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJqdGkiOiIzIiwiaXNSZWRha3RldXIiOmZhbHN1LCJpYXQiOi0jE00TA20DE3NDgsImV4cCI6MTQ5MDY4MjM0OH0.TmvNeQaeinxMIyfgds1UVT0gkqLeKnW6VEwbzHP07riERpHsocKmwZkz1ptr21nPU9xHv6A0NGfEH1Yg
```

Decoded

EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE
<pre>{ "typ": "JWT", "alg": "HS512" }</pre>
PAYLOAD: DATA
<pre>{ "jti": "3", "isRedakteur": false, "iat": 1490681748, "exp": 1490682348 }</pre>
VERIFY SIGNATURE
<pre>HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload), secret) <input type="checkbox"/> secret base64 encoded</pre>

Abbildung 73: JSON-Web-Token Aufbau anhand eines HHS-Tokens (jwt.io/, 2017)

9.5 Orgware

9.5.1 Microsoft Office

Alle Dokumente, welche im Rahmen der Diplomarbeit erstellt wurden, sind mithilfe der Tools von Microsoft entstanden. Die folgenden Programme wurden verwendet:

9.5.1.1 Microsoft Word 2016

- Dokumentation
- Protokolle
- Notizen

9.5.1.2 Microsoft Excel 2016

- Lokale Zeitaufzeichnung
- Aufgabenlisten
- IMV-Matrix

9.5.1.3 Microsoft PowerPoint 2016

- Projektpräsentationen

9.5.1.4 Microsoft Visio 2013

- Diagramme

9.5.2 GitLab

GitLab ist eine Projektmanagement- und Versionskontrollsoftware welche auf Git aufbaut.



GitLab

GitLab wird von GitLab Inc.

entwickelt und steht in zwei Abbildung 74 GitLab-Logo (forum.gitlab.com, 2017)

Versionen zur Verfügung.

Einerseits in einer Business-Version und andererseits in einer Community-Version. Für die Diplomarbeit haben wir die Community-Version verwendet, welche auf einem gemieteten Server eines Projektmitgliedes installiert ist. Der Unterschied zwischen der kostenpflichtigen und kostenfreien Version liegt in der Funktionsvielfalt, welche für große Projekte und Unternehmen Vorteile bringt. Jedoch bietet die kostenfreie Version alle grundlegenden Features, welche für das Projektmanagement und die Versionskontrolle notwendig sind. Weiters ist eine Fehlermanagementlösung integriert.

Die Managementaspekte werden über die mitgelieferte Webseite abgewickelt. Diese bietet neben einer Zugriffskontrolle ein virtuelles Scrum-Board, welches für Kanban hilfreich war. Darüber hinaus kann man mit der Webanwendung das Repository verwalten.

GitLab selbst ist in Ruby implementiert und verwendet für die Webanwendung das Ruby on Rails Framework. Zum Speichern der Daten, welche in GitLab angezeigt werden, wird eine PostgreSQL-DB verwendet. Die Datenbank wird dabei beim Installieren von GitLab automatisch generiert. Nach der Installation können die Benutzer erstellt werden. Jeder Benutzer muss nach dem ersten Login ein Passwort eingeben um sich in Zukunft als diese Person zu authentifizieren. Danach kann ein SSH-Schlüssel eingegeben werden um sich mit dem git-Client über SSH zu verbinden. Weiters können Personen in Gruppen unterteilt werden um das Management zu erleichtern. Diese Personen und Gruppen können anschließend zu Repositories hinzugefügt werden um Zugriffe zu ermöglichen.

Außerdem bietet GitLab auf Repository-Ebene fein definierbare Zugriffskontrollen. Dabei kann man Berechtigungen für eine Person, für gesamte Gruppen, für Personen in einer Gruppe und für Personen in Repositories setzen. Wenn ein Benutzer ein Repository erstellt, muss dieser einen Namen eingeben, wie das Repository genannt werden soll. Beim Erstellen wird der Benutzer automatisch zum Administrator des Repositories. Der Administrator kann anschließend Personen und Gruppen hinzufügen, die Zugriffsrechte für jene setzen und alle anderen administrativen Aufgaben erledigen. Ein Repository kann entweder mit dem Namespace einer Gruppe oder einer Person erstellt werden. Der Namespace ist zwar unwesentlich, bietet aber einen Konfigurationsvorteil. Wenn man ein Repository erstellt und dies dem Namespace einer Gruppe zuweist, so bekommen alle Benutzer der Gruppe Zugriff auf das Repository mit ihren jeweiligen Berechtigungen in der Gruppe. Diese Zugriffsrechte können projektspezifisch angepasst werden.

9.5.3 GNU Image Manipulation

Program (GIMP)

GIMP ist ein Open Source
Bildbearbeitungsprogramm. Ein Vorteil
dieser Software ist, dass es sowohl verfügbar
für GNU/Linux Distributionen ist, als auch für
Windows und OSX. Es beinhaltet viele
hilfreiche Funktionen zur Bildbearbeitung.

Im Verlauf der Diplomarbeit wurde dieses
Tool zum Erstellen oder zum Ändern von Grafiken, wie z.B.: Das Logo, Plakate für
Veranstaltungen und sonstige Grafiken verwendet.
(vgl. code.gimp.org, 2017)



Abbildung 75: GIMP-Logo (gimp.org, 2017)

9.5.4 TortoiseGit

TortoiseGit stellt

- eigene Icons für den Windows Explorer, die über den Dateistatus aufschluss geben,
- ein kraftvolles Kontextmenü für die Versionierungssoftware Git
- und noch viel mehr

bereit. (vgl. tortoisegit.org, 2017)

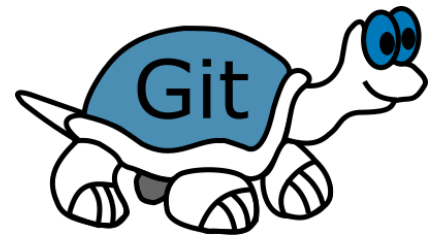
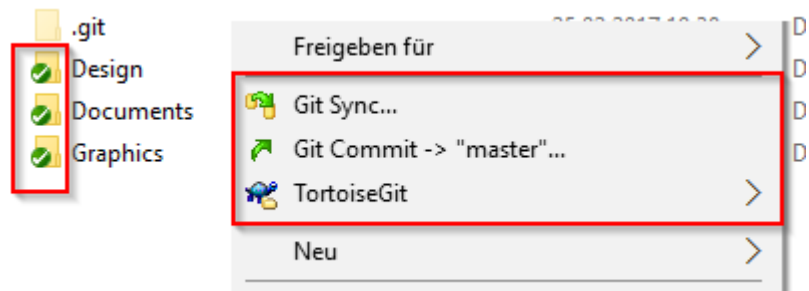


Abbildung 76: TortoiseGit-Logo
(tortoisegit.org, 2017)

Der Großteil unseres Projektteams hat mit TortoiseGit gearbeitet, was eine einfache Versionsverwaltung unter Windows ermöglichte. Ein grünes Häkchen neben dem Dateinamen bedeutet, dass die Datei/der Ordner mit dem Server synchronisiert ist und das Kontextmenü liefert alle Optionen, die zum normalen Gebrauch von Git notwendig sind.

Auch können Branches erstellt und unterschiedliche Dateiversionen miteinander verglichen (merged) werden.



9.5.5 GitKraken

Neben TortoiseGit wurde im Projekt auch GitKraken als Git-GUI-Client verwendet. Der



Vorteil von GitKraken

Abbildung 77 GitKraken Logo (cloudfront.net, 2017)

liegt in der

Plattformunabhängigkeit, denn GitKraken ist sowohl für Windows und Mac als auch für Linux verfügbar. Dabei ist die Benutzeroberfläche auf allen Plattformen ident. GitKraken bietet eine grafisch ansprechende Oberfläche welche die meisten Git-Funktionen implementiert. Dies erleichtert die Git-Interaktionen enorm und stellt einen flüssigen Workflow sicher. Weiters fördert GitKraken das Verständnis von Git während der Benutzung. GitKraken bietet auch eine Integration für die drei größten Git-Hosting Dienste (GitHub, Bitbucket, GitLab) und erlaubt es direkt in GitKraken diese zu verwalten. (vgl. gitkraken.com, 2017)

9.5.6 Kimai

Kimai ist ein freier open-source timetracker. Es loggt die Arbeitszeit bei bestimmten Projekten und gibt eine Zusammenfassung von allen Aktivitäten bei Bedarf aus. Jährlich, monatlich, pro Projekt, pro Aktivität et cetera.



Abbildung 78: Kimai-Logo (kimai.org, 2017)

Seine Stärke liegt in seiner Simpelheit. Aufgrund Kimai's Web-browser-basiertem Interface ist es plattformübergreifend lauffähig. Es kann sowohl als Webservice als auch als Einzelbenutzerprogramm auf dem eigenen Gerät installiert werden. Selbst ein Schließen des Webbrowsers stoppt die Zeitaufzeichnung nicht. Diese muss von irgendeinem Webbrowser aus nach einem Login angehalten werden. (vgl. kimai.org, 2017)

Aufgrund dessen, dass in der HTL Perg diese Zeittrackingsoftware bereits in allen Projekten verwendet wird, haben auch wir uns dafür entschieden, hier unsere Arbeitszeiten einzutragen.

9.5.7 Telegram

Telegram ist ein Messenger-Dienst, der es erlaubt, alle Nachrichten auf verschiedensten Geräten zu synchronisieren und Dateien in der Cloud abzuspeichern, sodass diese auch nach einiger Zeit weiterhin abrufbar und gleichzeitig gesichert sind. (vgl. telegram.org, 2017)



Abbildung 79: Telegram-Logo (telegram.org, 2017)

Aufgrund dieser und einiger weiterer Eigenschaften haben wir uns entschieden, diesen Service als Hauptkommunikationsmittel für unsere Diplomarbeit zu verwenden, da alle Dateien immer und überall abrufbar sind und jedes Teammitglied zu jeder Zeit über alles Bescheid wissen konnte.

9.5.8 Teamspeak

Teamspeak ist eine Voice over IP-Software, welche Wert auf Sicherheit legt und kristallklare Sprachqualität, Skalierbarkeit bis zu tausenden von Benutzern gleichzeitig und schier endlose persönliche



Abbildung 80: Teamspeak-Logo (www.teamspeak.com, 2017)

Einstellungsmöglichkeiten bietet. Zusammen mit kraftvollen Features und unglaublicher Performance ist Teamspeak das schweizer Armeemesser der Sprachkommunikation. (vgl. www.teamspeak.com, 2017)

Weil ein Klassenkollege einen Teamspeak-Server betreibt und auch ein Teamkollege einen solchen zur Verfügung stellt, haben wir dieses Medium für die entfernte Sprachkommunikation zur Abwicklung unserer Diplomarbeit verwendet. Selbst wenn ein Server einmal nicht zur Verfügung stand, konnten wir auf den anderen ausweichen und projektrelevante Tätigkeiten schnellstmöglich durchführen.

9.5.9 TeamViewer

Mithilfe von TeamViewer ist es möglich, seinen eigenen Bildschirm für andere Benutzer freizugeben, damit diese entweder nur zusehen können (Meeting-Funktion) oder ein bestimmter Benutzer den Rechner steuern kann (Fernsteuerungs-Funktion). Weiters ist der Versand von Chat-Nachrichten oder ein optionaler Video-Anruf verfügbar. Ein weiterer Vorteil von TeamViewer ist, dass es nicht installiert werden muss, sondern direkt ausgeführt werden kann, damit man in Sekundenschnelle eine Verbindung aufbauen kann. (vgl. [teamviewer.com](https://www.teamviewer.com), 2017)

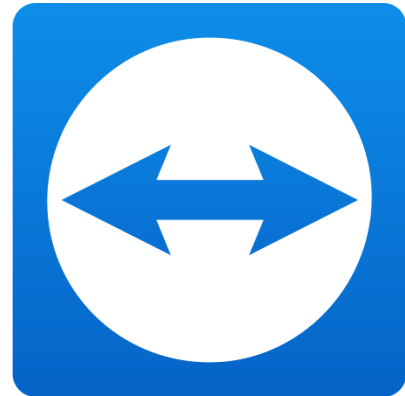
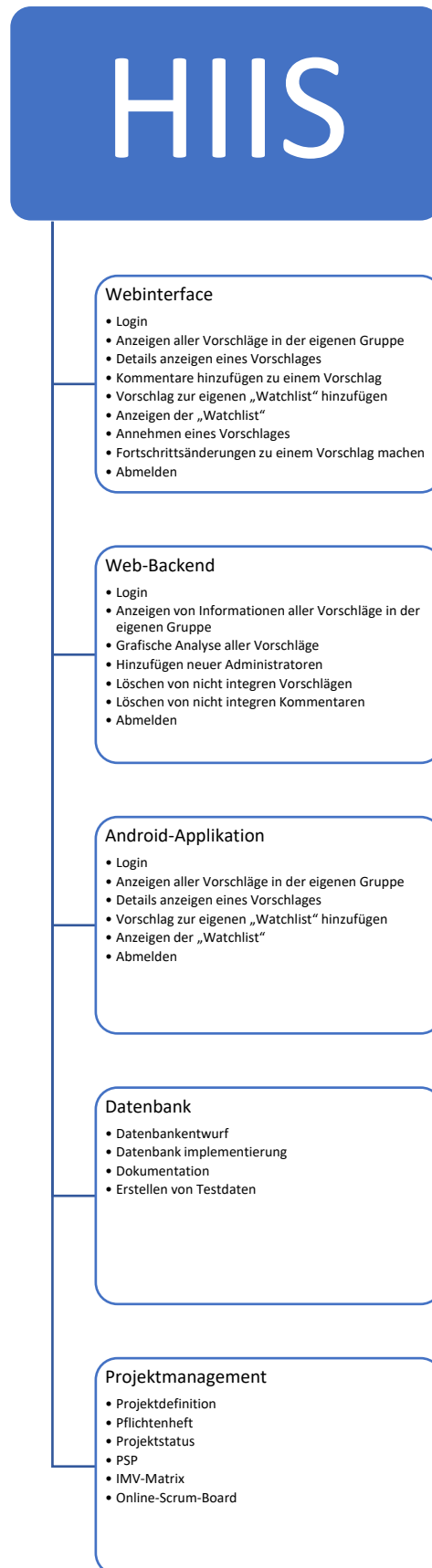


Abbildung 81: TeamViewer-Logo
([teamviewer.com](https://www.teamviewer.com), 2017)

Ohne TeamViewer hätten wir immer zu anderen Teammitgliedern fahren müssen, um diese zu unterstützen. Dies hat nicht nur Zeit gespart, sondern auch bessere Ergebnisse ermöglicht, da entsprechende Programmteile dem restlichen Projektteam ganz einfach präsentiert werden konnten.

9.6 Projektstrukturplan



9.7 Verantwortungsmatrix

Tätigkeiten	Roland Neulinger	Patrick Huber	Tobias Janko	Markus Luftensteiner
Allgemeine Tätigkeiten				
Kontakt mit Auftraggeber	V	M		
Datenbank Sicherung				V
Projektbeginn				
Kick-Off Meeting mit AG abhalten	V		M	
Verantwortungsmatrix	V	M	M	M
Projekt- Geschäftsziele definieren	M			V
Repository anlegen		V		
Pflichtenheft	V	M	M	
Benötigte HW und SW beschaffen		V		M
Beschreibung				
Schnittstellenbeschreibung		M		V
Architektur	V	M	M	M
Dokumente				
Projektstrukturplan	V	M	M	M
Meeting-Protokolle		M	V	
Projektstatusbericht	M	V	M	M
Zeitplan	V			M
Eintrag Diplomarbeitsdatenbank	V	M		
Grafiken	M		V	
Entwurf				
Datenbankentwurf				V
Entwurf der Webseite		M	V	
Entwurf der App	V			
Implementierung				
Aufsetzen des Servers				V
Datenbank	M	M	M	V
Datenbankschnittstelle			M	V
Webserver		V		
REST-Service	M	V	M	
Redaktionsteam-Webseite		M	V	
Benutzer-Webseite		M	V	
Android Applikation	V	M		

Tabelle 15: Verantwortungsmatrix

10 Resümee und Ausblick

In diesem Kapitel rekapitulieren wir, was wir im Rahmen dieses anspruchsvollen Projektes und beim Erstellen der Diplomarbeit gelernt haben. Das betrifft einerseits die Arbeitsmethoden, aber auch die Soft-Faktoren in der Zusammenarbeit.

10.1 Projektdurchführung mittels Kanban

Kanban ist ein Vorgehensmodell zur Softwareentwicklung, bei dem die Anzahl paralleler Arbeiten, der Work in Progress (WiP), begrenzt und somit kürzere Durchlaufzeiten erreicht und Probleme – insbesondere Engpässe – schnell sichtbar gemacht werden sollen. (de.wikipedia.org, 2017)

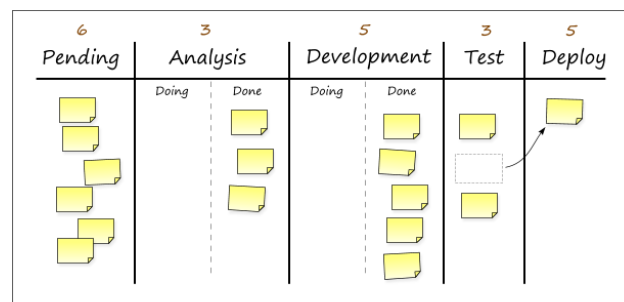


Abbildung 82: Beispielhaftes Kanban-Board (kanbanblog.com, 2017)

Wir haben uns aus eben diesem Grund für die Überwachung zur laufenden Projektdurchführung für die Arbeit mit Kanban entschieden. Alle zu Anfang definierten Tasks wurden von den einzelnen Teammitgliedern der sinnvollen Reihe nach abgearbeitet. Sollte ein bestimmter Task das Kanban-Limit überlasten, so wurde dies recht schnell sichtbar und andere Kollegen konnten unterstützen um die jeweilige Blockade zu lösen. Somit wurde ein weitgehend störungsfreier Ablauf der Diplomarbeit sichergestellt.

10.2 Kommunikation

Die Kommunikation mit unserem Auftraggeber Herrn Dipl. Ing. Prof. Reisinger funktionierte stets sehr gut. In regelmäßigen persönlichen Meetings mit dem Projektteam, bei denen wir alle offenen Punkte bzw. Fragen geklärt haben, gab es regen Austausch zwischen Auftraggeber und den Projektteammitgliedern. Wir haben unserem Auftraggeber den regelmäßigen Fortschritt aller unserer Teilbereiche präsentiert, wodurch er einen guten Überblick bekommen hat wie der Realisierungsgrad aktuell stand. Dadurch, dass unser

Auftraggeber gleichzeitig unser Professor in Systemplanung und Projektentwicklung ist, ergab sich auch im Rahmen des Unterrichts ab und zu Gelegenheit den Fortschritt zu diskutieren.

Die Kommunikation im Projektteam erfolgte in den meisten Fällen über Teamspeak oder persönlich in der Schule. Regelmäßige, systematische Projektteammeetings waren nicht erforderlich, da sich in der Klasse spontan Abstimmungen einberufen ließen. Wir haben mit dem jeweiligen Teammitglied direkt gesprochen, wenn Unklarheiten aufgetreten sind. Der persönliche Austausch im Projektteam war wichtig, weil wir dadurch motiviert in der persönlichen Umsetzung blieben. Es war jederzeit klar, dass Lösungen gemeinsam erreicht werden müssen um die Projektpläne nicht zu gefährden.

10.3 Aufgetretene Probleme

Zeitintensiv gestaltete sich die Startphase, in der wir uns in die zu verwendenden Tools und Utensilien (z.B. Programmiersprache, Umgang Entwicklungsumgebung) einarbeiten mussten und es ein paar Versuche brauchte um uns mit diesen neuen Themen vertraut zu machen. Herausfordernd war die intensive Fehlersuche, wenn sich welche eingeschlichen haben. Wir haben einander auch bei der Fehlersuche und -behebung unterstützt, was eine Notwendigkeit war, nachdem die Anwendung so stark vernetzt ist, dass wir starke Abhängigkeiten untereinander hatten. Wenn bei einem Task oder in einer Anwendung ein Fehler auftrat, so musste er nicht zwangsläufig dort verursacht sein. Zum Beispiel, wenn die Funktionalität eines Requests nicht richtig funktioniert hat, konnte es sein, dass die App oder die Webseite den Request nicht richtig absendet. Es konnte aber auch sein, dass der REST-Service auf dem Server den Request falsch verarbeitet und deshalb die Funktionalität nicht gegeben ist.

Ein weiteres Problem verursachten die regelmäßigen Updates vom REST-Service. Da wir bis Februar noch lokal gearbeitet haben mussten wir den neuen Service immer wieder neu in Eclipse einbinden. Das verursachte oft mehrmals täglich ein hoher Zeitaufwand - vor allem dann, wenn es mehrere Updates gab, die man zusammenfügen hätte können.

Bei der Webseite bestand anfangs das Problem, dass die Requests vom Browser selbst blockiert wurden, weshalb die Requests nicht funktionierten. Eine Zwischenlösung war die Browsererweiterung CORS Toggle, die die Requests durchließ. In der Endanwendung wurde ein Filter im REST-Service implementiert, der dieses Problem löste.

Beim Anbinden von LDAP, dem Benutzerverwaltungssystem der HTL Perg, in dem jede Schülerin und jeder Schüler sowie jeder Lehrer und jede Lehrerin einen persönlichen Account besitzt, gab es bei den ersten Versuchen Probleme mit dem Zugriff um die Daten zu erhalten. Gelöst haben wir es in Zusammenarbeit mit einem unserer Programmierprofessoren.

10.4 Zukünftige Erweiterungen

In der Projektdefinition wurde die Entwicklung einer iOS- und einer Windows-Phone-Applikation ausgenommen. Dadurch, dass die Dokumentation in dieser Diplomarbeit sehr detailliert ist, können jederzeit Apps für diese beiden Betriebssysteme entwickelt werden und so der Scope dieser integrierten Ideenmanagementplattform erweitert werden.

Auch empfehlen wir eine intensive Testphase mit wenigen Usern im echten Anwendungsumfeld sowie Optimierungen anhand von User-Experience-Erkenntnissen was die Oberflächen und Funktionen betrifft. Eine an diese anknüpfende Diplomarbeit könnte sich mit Testmanagement beschäftigen und so einen sinnvollen Abschluss und einen reibungslosen Go-Live des Systems sicherstellen sowie eine hohe Akzeptanz der Benutzer von Anfang an fördern.

11 Glossar

FREMDWORT	ERKLÄRUNG
Active Directory (AD)	Eine von Microsoft entwickelte und bereitgestellte Speicherstruktur, in denen Benutzerdaten und Ähnliches zur Abfrage bereitsteht.
Activity	Eine Activity ist im Android-Betriebssystem eine Schnittstelle für den Benutzer. Es bietet eine Benutzeroberfläche mit dem der Benutzer interagieren kann.
Algorithmus	Eine Reihe von Programmanweisungen, die einen bestimmten Zweck erfüllen sollen.
Bootstrap	Ein Werkzeugkasten zum Verschönern von Webseiten und um benutzerfreundlichere Funktionen bereitstellen zu können.
Data Access Object (DAO)	Eine Java-Klasse, die es ermöglicht, auf Datenbankinhalte zuzugreifen.
Dropdown	Eine vordefinierte Auswahlliste, aus der genau ein Wert ausgewählt werden kann.
Employer Branding	Ist die vom Unternehmen gezielt gestaltete Art und Weise, wie es im Arbeitsmarkt als Arbeitgeber wahrgenommen wird.
Extensible Markup Language (XML)	Ein Datenformat, das mithilfe von sogenannten „Tags“ Daten zur Speicherung und Übertragung organisiert.
Graduierung	(Schul-)Abschluss
Hashalgorithmus	Eine Reihe von Programmanweisungen, die eine Eingabe leicht verschlüsseln, diese Verschlüsselung aber nicht mehr leicht rückgängig zu machen ist.
Human Resource Management	Personalmanagement

Implementierung	Umsetzung
Impuls	Eine Einheit, die von der Eingabe eines Benutzers definiert wird. Kann eine Anregung, eine Kritik, eine Idee o.Ä. sein.
Internet Engineering Task Force (IETF)	Eine Organisation, die relevante Tätigkeiten und Spezifikationen zum Thema Internet bereitstellt und standardisiert.
Interoperabilität	Die Fähigkeit von Systemen, miteinander zu funktionieren bzw. zusammenzuarbeiten.
iOS	Ein Betriebssystem, welches von Apple entwickelt wurde für sämtliche Apple-Geräte.
JavaScript Object Notation (JSON)	Ein Datenformat, das die Übertragung von programmtechnisch wichtigen Daten ermöglicht und vom Aufbau her recht kompakt gestaltet ist.
Kategorie	Eine Kategorie gibt an, welcher Kategorie ein bestimmter Impuls zugeordnet werden kann. (Beispiel: Tag der offenen Tür)
Kommentar	Ein vom Benutzer verfasster Text, der eine Anregung o.Ä. sein kann, der sich an den Ersteller bzw. an den Verantwortlichen eines Impulses richtet.
Lightweight Directory Access Protocol (LDAP)	Ein Satz von Regeln, mit denen ein Active Directory-Server befragt werden kann, welcher dann beispielsweise Benutzer abgefragt werden können.
Persistierung	(Daten-)Speicherung
Project Scope	Tätigkeiten und Ergebnisse, die zu Beginn des Projektes definiert werden und am Ende vorliegen sollen. Definiert auch, was nicht im Projekt erledigt und gemacht wird.

Repository	Eine Speicherstruktur (welche auch auf einem entfernten Server sein kann), in der projektrelevante Daten und Quellcode gespeichert sind.
Request Queue	Eine Reihe von Abfragen, die der Reihe nach versendet bzw. abgearbeitet werden können.
REST bzw. Rest-Interface	Eine Schnittstelle, mit deren Hilfe Anwendungsprogramme Daten von einem Server empfangen können.
Socket	Eine Verbindungsmöglichkeit, die von Programmen für Programme bereitgestellt wird.
Stakeholder	Eine Person oder Gruppe, die ein berechtigtes Interesse am Verlauf oder Ergebnis eines Prozesses oder Projektes hat.
Status	Ein Impuls kann vorgegebenermaßen folgende Statusänderungen annehmen: offen, in Bearbeitung, archiviert, gelöscht, zurückgestellt und fertig. Dieser wird von Benutzern gesetzt.
Stored Procedure	Eine Methode, die in der Datenbank selbst gespeichert ist und von einem Anwendungsserver aufgerufen wird.
Typ	Ein Typ gibt an, welcher Art ein Impuls angehört.
Web-backend	Die Funktionalität, die vom Server im Hintergrund zur Verfügung bereitgestellt wird, um bspw. die Abfrage von Daten zu ermöglichen.
Webinterface	Ist die Benutzerschnittstelle im Browser.

WWW

World Wide Web → ein über das Internet aufrufbares System von elektronischen Dokumenten, sogenannten Webseiten.

12 Anhang

12.1 Aufteilung der Diplomarbeiteniederschrift

12.1.1 Markus Luftensteiner

- Beschreibung Betreuungslehrer
- Grundaufbau unseres Systems
- Technische Grundlagen von Datenbanken
- Abstract
- Architektur
- Datenmodell inklusive aller Unterpunkte
- State Transition Diagram
- Mengengerüst
- Stored Procedures
- TortoiseGit
- Kimai
- MySQL-Datenbank
- MySQL-Workbench
- Telegram
- Teamspeak
- TeamViewer

12.1.2 Patrick Huber

- GitLab
- Eclipse
- JBoss-Tools
- WildFly
- JSON Web Token
- HTTP-Headers
- REST
- ORM
- GitKraken
- Advanced Rest Client
- BlueGriffon
- Ubuntu
- Java Persistence API
- JAX-RS
- Implementierung Webservice

12.1.3 Tobias Janko

- WebStorm
- Postman
- Google Chrome Webbrowser
- Visual Studio Code
- Ausgangssituation
- Kommunikation
- Aufgetretene Probleme
- Zukünftige Erweiterungen
- Darstellung Webseite
- Implementierung Webseite

12.1.4 Roland Neulinger

- Danksagung
- Kurzfassung
- Motivation
- Auftraggeber
- Geschäftsziele
- Projektziele
- Ausgangssituation
- Technische Grundlagen von Mobilien Anwendungen
- Darstellung der Android Applikation
- Architektur
- Implementierung der Android Applikation
- Meilensteine
- Zeitplanung
- Android Studio
- Volley Library
- Microsoft Office
- Projektstrukturplan
- Verantwortungsmatrix

12.2 Literaturverzeichnis

Bronner, P. D.-I. (2003). Handbuch der Rationalisierung. expert verlag. Von

https://books.google.at/books?id=pvXXOuQrCGMC&pg=PA97&dq=definition+betriebliches+vorschlagswesen&hl=de&sa=X&redir_esc=y#v=onepage&q=definition%20betriebliches%20vorschlagswesen&f=false abgerufen

cloudfront.net. (03 2017). Von [https://dwa5x7aod66zk.cloudfront.net/assets/pack/logo-](https://dwa5x7aod66zk.cloudfront.net/assets/pack/logo-gitkraken-)
[c292786ac0b639afcda0a4d8531869e87cde5e40e255fe8684b0a19add839edd.jpg](https://dwa5x7aod66zk.cloudfront.net/assets/pack/logo-gitkraken-c292786ac0b639afcda0a4d8531869e87cde5e40e255fe8684b0a19add839edd.jpg)

abgerufen

code.tutsplus.com. (03 2017). Von <https://code.tutsplus.com/tutorials/an-introduction-to-volley--cms-23800> abgerufen

Cuhls, K. (1993). Qualitätszirkel in japanischen und deutschen Unternehmen. Von

https://books.google.at/books?id=pQLOBgAAQBAJ&pg=PA9&dq=definition+Qualit%C3%A4tszirkel&hl=de&sa=X&redir_esc=y#v=onepage&q=definition%20Qualit%C3%A4tszirkel&f=false abgerufen

de.wikipedia.org. (03 2017). Von

[https://de.wikipedia.org/wiki/Kanban_\(Softwareentwicklung\)](https://de.wikipedia.org/wiki/Kanban_(Softwareentwicklung)) abgerufen

de.wikipedia.org. (03 2017). Von <https://de.wikipedia.org/wiki/MySQL> abgerufen

de.wikipedia.org. (03 2017). Von

https://de.wikipedia.org/wiki/MySQL#/media/File:Logo_MySQL.svg abgerufen

design.jboss.org. (03 2017). Von

http://design.jboss.org/jbosstools/logo/jbosstools_horiz_600.png abgerufen

design.jboss.org. (03 2017). Von

http://design.jboss.org/wildfly/logo/final/wildfly_logo_stacked_600px.png abgerufen

dev.mysql.com. (03 2017). Von <https://dev.mysql.com/doc/refman/5.7/en/integer-types.html> abgerufen

developer.android.com. (03 2017). Von

<https://developer.android.com/guide/topics/manifest/manifest-intro.html>
abgerufen

developer.android.com. (03 2017). Von

<https://developer.android.com/studio/intro/index.html> abgerufen

developer.myob.com. (03 2017). Von

http://developer.myob.com/media/cms_page_media/285/advancedrest.jpg
abgerufen

eclipse.org. (03 2017). Von <https://www.eclipse.org/eclipse.org->

[common/themes/solstice/public/images/logo/eclipse-800x188.png](https://www.eclipse.org/eclipse.org-common/themes/solstice/public/images/logo/eclipse-800x188.png) abgerufen

forum.gitlab.com. (3 2017). Von

<https://forum.gitlab.com/uploads/default/original/1X/277d9badcbd723e913b3a41e64e8d2f3d2c80598.png> abgerufen

gimp.org. (04 2017). Von <https://www.gimp.org/images/frontpage/wilber-big.png> abgerufen

gitkraken.com. (03 2017). Von <https://www.gitkraken.com/features> abgerufen

htl-perg.ac.at. (03 2017). Von <http://htl-perg.ac.at/organisation/lehrer> abgerufen

htl-perg.ac.at. (03 2017). Von <http://htl->

[perg.ac.at/images/lehrerfotos/reisinger_christian.jpg](http://htl-perg.ac.at/images/lehrerfotos/reisinger_christian.jpg) abgerufen

Institut für angewandte Arbeitswissenschaft e. V. (2016). 5S als Basis des kontinuierlichen
Verbesserungsprozesses. Von

https://books.google.at/books?id=C5_tCwAAQBAJ&pg=PA41&dq=definition+KVP&hl=de&sa=X&redir_esc=y#v=onepage&q=definition%20KVP&f=false abgerufen

jwt.io/. (03 2017). Von <https://jwt.io/> abgerufen

kanbanblog.com. (03 2017). Von <http://kanbanblog.com/explained/image/kanban-board-2.png> abgerufen

kimai.org. (03 2017). Von <http://www.kimai.org> abgerufen

kimai.org. (03 2017). Von <http://www.kimai.org/assets/clocklogo.jpg> abgerufen

Läge, K. (2002). Ideenmanagement. Von

https://books.google.at/books?id=ofAjBgAAQBAJ&printsec=frontcover&dq=ideenmanagement&hl=de&sa=X&sqi=2&redir_esc=y#v=onepage&q&f=false abgerufen

logobird.com. (03 2017). Von logobird.com/new-google-chrome-logo/ abgerufen

mwtestconsultancy.co.uk. (03 2017). Von mwtestconsultancy.co.uk/postman-parsing-html-responses/ abgerufen

mysql.de. (03 2017). Von <https://www.mysql.de/products/workbench/> abgerufen

quintetsolutions.com. (03 2017). Von

http://cdn4.quintetsolutions.com/img/services/solutions/mysql_workbench_service_provider_india.jpg abgerufen

redbubble.com. (4 2017). Von redbubble.com/de/people/jtacrimson/works/22397279-visual-studio-code?p=sticker abgerufen

Riede, M. (2011). Determinanten erfolgreicher Stakeholderdialoge. Von https://books.google.at/books?id=_-iOpVlhwA8C&pg=PA26&dq=definition+interne+Stakeholder&hl=de&sa=X&redir_esc=y#v=onepage&q=definition%20interne%20Stakeholder&f=false abgerufen

Schat, H.-D. (2017). Erfolgreiches Ideenmanagement in der Praxis. Von https://books.google.at/books?id=mRRyDgAAQBAJ&pg=PA116&dq=betriebliches+vorschlagswesen+digital&hl=de&sa=X&redir_esc=y#v=onepage&q=vorschlagswesen&f=false abgerufen

teamviewer.com. (03 2017). Von <https://www.teamviewer.com/de/download/windows/> abgerufen

telegram.org. (03 2017). Von <https://telegram.org> abgerufen

tools.ietf.org. (03 2017). Von <https://tools.ietf.org/html/rfc7519> abgerufen

tortoisegit.org. (03 2017). Von <https://tortoisegit.org> abgerufen

upload.wikimedia.org. (03 2017). Von https://upload.wikimedia.org/wikipedia/commons/thumb/8/88/TortoiseGit_logo.svg/400px-TortoiseGit_logo.svg.png abgerufen

wikimedia.org. (03 2017). Von https://upload.wikimedia.org/wikipedia/en/1/19/Bluegriffon_Logo.png abgerufen

wikimedia.org. (03 2017). Von https://upload.wikimedia.org/wikipedia/commons/thumb/9/9d/Ubuntu_logo.svg/2000px-Ubuntu_logo.svg.png abgerufen

wikipedia.org. (03 2017). Von [https://en.wikipedia.org/wiki/Eclipse_\(software\)](https://en.wikipedia.org/wiki/Eclipse_(software)) abgerufen

wikipedia.org. (3 2017). Von <https://de.wikipedia.org/wiki/WildFly> abgerufen

wikipedia.org. (03 2017). Von https://en.wikipedia.org/wiki/JSON_Web_Token abgerufen

wikipedia.org. (03 2017). Von https://en.wikipedia.org/wiki/List_of_HTTP_header_fields abgerufen

wikipedia.org. (03 2017). Von https://de.wikipedia.org/wiki/Representational_State_Transfer abgerufen

wikipedia.org. (03 2017). Von https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol abgerufen

wikipedia.org. (03 2017). Von <https://de.wikipedia.org/wiki/WebStorm> abgerufen

wikipedia.org. (03 2017). Von wikipedia.org abgerufen

wikipedia.org. (03 2017). Von <https://de.wikipedia.org/wiki/BlueGriffon> abgerufen

wikipedia.org. (03 2017). Von [https://en.wikipedia.org/wiki/Ubuntu_\(operating_system\)](https://en.wikipedia.org/wiki/Ubuntu_(operating_system))
abgerufen

wikipedia.org. (03 2017). Von https://de.wikipedia.org/wiki/Java_Persistence_API abgerufen

wikipedia.org. (03 2017). Von

https://en.wikipedia.org/wiki/Java_API_for_RESTful_Web_Services abgerufen

wikipedia.org. (4 2017). Von wikipedia.org abgerufen

worldvectorlogo.com. (03 2017). Von worldvectorlogo.com/logo/webstorm-icon abgerufen

www.androidhive.info. (03 2017). Von <http://www.androidhive.info/2014/05/android-working-with-volley-library-1/> abgerufen

www.kantarworldpanel.com. (03 2017). Von

<https://www.kantarworldpanel.com/global/smartphone-os-market-share/> abgerufen

www.teamspeak.com. (03 2017). Von <https://www.teamspeak.com> abgerufen

12.3 Abbildungsverzeichnis

Abbildung 1: Portraitbild von Direktor Reisinger (htl-perg.ac.at, 2017).....	15
Abbildung 2: Logo der HTL Perg	16
Abbildung 3: Säulen des Ideenmanagements (vgl. Läge, 2002, 30)	22
Abbildung 4: Bereiche des Ideenmanagements (vgl. Läge, 2002, 31)	23
Abbildung 5: Vorteile Navtiver und Web Apps	26
Abbildung 6: Marktdominierende Plattformen der nativen App-Entwicklung	27
Abbildung 7: Marktanteil der verschiedenen Plattformen in Deutschland (www.kantarworldpanel.com, 2017)	28
Abbildung 8: Ablauf der Webseite	30
Abbildung 9: Home Seite Screenshot.....	31
Abbildung 11: Login Screenshot.....	32
Abbildung 12: Impulse Screenshot.....	33
Abbildung 13: Watchlist Screenshot	34
Abbildung 14: Alle Benutzer anzeigen Screenshot	35
Abbildung 15: Impuls hinzufügen Screenshot	36
Abbildung 16: Impuls bearbeiten Screenshot.....	37
Abbildung 17: Details Screenshot	38
Abbildung 18: Ablauf der Android-App.....	40
Abbildung 19: Login Activity.....	41
Abbildung 20: Main Activity	42
Abbildung 21: Create Impuls Activity	42
Abbildung 22: Detail Activity	43
Abbildung 23: Architekturdiagramm von HIIS	45
Abbildung 24: Übersicht über das Datenmodell.....	46
Abbildung 25: Standarddaten "Status"	47
Abbildung 26: Tabelle "Status"	47
Abbildung 27: Standarddaten "Typ"	47
Abbildung 28: Tabelle "Typ".....	47
Abbildung 29: Standarddaten "Kategorie".....	48
Abbildung 30: Tabelle "Kategorie"	48
Abbildung 31: Tabelle "Impuls".....	48

Abbildung 32: Testdaten "Impuls"	48
Abbildung 33: Tabelle "Datei"	49
Abbildung 34: Testdaten "Datei"	49
Abbildung 35: Tabelle "Kommentar"	49
Abbildung 36: Testdaten "Kommentar"	49
Abbildung 37: Testdaten "Watchlist"	50
Abbildung 38: Tabelle "Watchlist"	50
Abbildung 39: Tabelle "Benutzer"	50
Abbildung 40: Testdaten "Benutzer"	50
Abbildung 41: Die View "vw_impuls_overview"	51
Abbildung 42: State Transition Diagram des Impulsstatus (Vorschlag)	52
Abbildung 43: Die Klasse "Benutzer"	62
Abbildung 44: Der Inhalt der Datei "orm.xml"	63
Abbildung 45: Das Data Access Object für die Benutzerdaten	64
Abbildung 46: Der Endpoint für den Benutzerdatenzugriff	65
Abbildung 47: Die Erstellung eines JSON Web Tokens	65
Abbildung 48: Die Verifizierung eines JSON Web Tokens	66
Abbildung 49: Login HTML	68
Abbildung 50: Login Component, ruft den Service auf	69
Abbildung 51: Login Service, ruft die Request auf	70
Abbildung 52: Projekt Module, kümmert sich um die einzelnen Weiterleitungen, zwischen den Seiten.....	71
Abbildung 53: Beispiel für ein Routing (Weiterleitung)	71
Abbildung 54: Die Android Manifest Datei	72
Abbildung 55: Die Dependencies im Android Manifest	73
Abbildung 56: Definition der Klasse LruBitmapCache	73
Abbildung 57: ApplicationController Klasse.....	74
Abbildung 58: Internet Permissions im Android Manifest.....	74
Abbildung 59: Erstellen eines Volley Requests	75
Abbildung 60: Gantt-Zeitplan Legende	77
Abbildung 61: Eclipse-Logo (eclipse.org, 2017)	79
Abbildung 62: JBoss Tools Logo (design.jboss.org, 2017)	80

Abbildung 63: BlueGriffon Logo (wikimedia.org, 2017).....	80
Abbildung 64: WildFly Logo (design.jboss.org, 2017)	81
Abbildung 65: MySQL-Logo (de.wikipedia.org, 2017).....	82
Abbildung 66: MySQL-Workbench-Logo (quintetsolutions.com, 2017)	82
Abbildung 67: Android Studio Logo (technicolit.com, 2017)	83
Abbildung 68: WebStorm Logo (worldvectorlogo.com, 2017)	83
Abbildung 69: Google Chrome Webbrowser Logo (logobird.com, 2017).....	84
Abbildung 70: Postman Logo (mwtestconsultancy.co.uk, 2017).....	84
Abbildung 71: Visual Studio Code Logo (redbubble.com, 2017)	85
Abbildung 72: Advanced REST Client Logo (developer.myob.com, 2017).....	85
Abbildung 73: Ubuntu Logo (wikimedia.org, 2017)	86
Abbildung 74: JSON-Web-Token Aufbau anhand eines HUIS-Tokens (jwt.io/, 2017).....	91
Abbildung 75 GitLab-Logo (forum.gitlab.com, 2017).....	93
Abbildung 76: GIMP-Logo (gimp.org, 2017).....	94
Abbildung 77: TortoiseGit-Logo (tortoisegit.org, 2017).....	95
Abbildung 78 GitKraken Logo (cloudfront.net, 2017)	96
Abbildung 79: Kimai-Logo (kimai.org, 2017)	96
Abbildung 80: Telegram-Logo (telegram.org, 2017).....	97
Abbildung 81: Teamspeak-Logo (www.teamspeak.com, 2017)	97
Abbildung 82: TeamViewer-Logo (teamviewer.com, 2017)	98
Abbildung 83: Beispielhaftes Kanban-Board (kanbanblog.com, 2017)	101

12.4 Tabellenverzeichnis

Tabelle 1: Persönliche Daten von Roland Neulinger	7
Tabelle 2: Kompetenzen von Roland Neulinger	8
Tabelle 3: Schullaufbahn von Roland Neulinger	8
Tabelle 4: Persönliche Daten von Patrick Huber	9
Tabelle 5: Kompetenzen von Patrick Huber	10
Tabelle 6: Schullaufbahn von Patrick Huber	10
Tabelle 7: Persönliche Daten von Tobias Janko	11
Tabelle 8: Kompetenzen von Tobias Janko	12
Tabelle 9: Schullaufbahn von Tobias Janko.....	12
Tabelle 10: Persönliche Daten von Markus Luftensteiner	13
Tabelle 11: Kompetenzen von Markus Luftensteiner	14
Tabelle 12: Schullaufbahn von Markus Luftensteiner	14
Tabelle 13: Beschreibung von Direktor Reisinger (htl-perg.ac.at, 2017)	15
Tabelle 14: Meilensteine	76
Tabelle 15: Verantwortungsmatrix	100