

Universität Passau
Bachelorarbeit
eingereicht bei Prof. Dr. Tomas Sauer

Konzeption und Umsetzung eines webbasierten Systems zur effizienten
Organisation mathematischer Wettbewerbsaufgaben

19. November 2012

Julian von Heydebrand und der Lasa
Bachelor Internet Computing

Inhaltsverzeichnis

1	Systemübersicht	4
1.1	Zusammenfassung	4
1.2	Motivation	4
1.3	Zielsetzung	4
2	Funktionen	5
2.1	Infoseiten	5
2.2	Upload	6
2.3	Suche	8
2.4	Aufgabenkorb	8
2.5	Systemfunktionen	10
2.5.1	LaTeX	10
2.5.2	SVN	12
2.5.3	Mehrsprachigkeit	12
2.5.4	Konfiguration	13
3	Implementierung	15
3.1	Nutzeroberfläche	16
3.2	Datenhaltung	17
3.3	Systemstruktur	19
3.3.1	DownloadServlet	20
3.3.2	ServerStartStopListener	21
3.4	Serverumgebung	21
3.5	LaTeX	23
3.5.1	Datenextraktion	23
3.5.2	Download von Aufgaben	25
3.6	Sicherheit	27
3.7	Fehlerbehandlung	29
4	Vergleich	29
4.1	SMART	30
4.2	Mathematik-Wettbewerbe	31
4.3	Lösungsstrategien	32
5	Resümee	32

Abkürzungsverzeichnis

CC Creative Commons

CSS Cascading Style Sheets

HSQLDB HyperSQL DataBase

ISO International Organization for Standardization

JRE Java Runtime Environment

JSF JavaServer Faces

MVC Model View Controller

PDF Portable Document Format

SMART Sammlung Mathematischer Aufgaben als HypeRtext mit T_EX

SVN Apache Subversion

URL Uniform Resource Locator

XHTML Extensible Hypertext Markup Language

XSS Cross-Site-Scripting

Z-MNU Zentrum zur Förderung des mathematisch-naturwissenschaftlichen Unterrichts

Kapitalübersicht

Systemübersicht Das erste Kapitel gibt eine Zusammenfassung des Systems sowie eine Erläuterung der Motivation und Zielsetzung hinter dem System.

Funktionen Das zweite Kapitel bespricht alle wichtigen Funktionen, die das System sowohl seinen Nutzern bietet als auch intern verwendet. Zunächst wird auf die verschiedenen Unterseiten der Onlineplattform eingegangen und anschließend die Hintergrundprozesse und auch individuellen Anpassungsmöglichkeiten des Systems gezeigt.

Implementierung Das dritte Kapitel zeigt die verschiedenen verwendeten Technologien und bespricht in Teilen die konkrete Umsetzung der Plattform. So wird auf den zugrundeliegenden Aufbau des Systems eingegangen und erläutert welche Designphilosophie der Konzeption der einzelnen Elemente der Plattform zugrunde lagen.

Vergleich Das vierte Kapitel vergleicht die Plattform mit verschiedenen anderen Möglichkeiten Mathematikaufgaben zu verbreiten. Es wird dabei sowohl auf eine ähnliche Onlineplattform als auch auf Bücher und gedruckte Aufgabensammlungen eingegangen.

Resümee Das letzte Kapitel zieht ein Resümee der erstellten Plattform und gibt einen Ausblick auf mögliche Erweiterungen derselben und Anknüpfungspunkte.

1 Systemübersicht

1.1 Zusammenfassung

WorkSheetz ist eine Onlineplattform zur Verwaltung und Bereitstellung von frei zusammenstellbaren Aufgabenblätter und richtet sich sowohl an Schüler und Lehrer als auch alle anderen Mathefans, die gerne Aufgaben lösen oder auch erstellen. Jeder Nutzer kann sowohl neue Aufgaben hinzufügen als auch alle Bestehenden betrachten, miteinander kombinieren und herunterladen.

1.2 Motivation

So verschrien Mathematik als Stolperstein in der Ausbildung auch sein mag: ebenso fundamental wichtig ist dieses Fachgebiet in zahlreichen Berufsfeldern - nicht zuletzt auch der Informatik. Für alle diejenigen, die sich mit ihr beschäftigen - sei es aus Freude an der Disziplin an sich oder auch aus der reinen Notwendigkeit, als Grundlage für ein anderes Fach oder eine Prüfung, heraus - hat sich wohl früher oder später herauskristallisiert, dass Übung den Meister macht. An dieser Stelle möchte das Projekt *WorkSheetz* seine Nutzer unterstützen: Die Plattform bietet einen einfachen, leicht durchsuchbaren Zugang zu Mathematikaufgaben der verschiedensten Themengebieten, Schwierigkeits- und Klassenstufen. Sie soll durch eine Vielzahl von Aufgaben und Übungsformen seinen Nutzern dabei helfen selbstständig mathematische Arbeitsweisen zu erlernen und zu vertiefen.

1.3 Zielsetzung

Es soll eine Plattform geschaffen werden, die den Austausch von Mathematikaufgaben unter allen interessierten Nutzern ermöglicht. Grundsätzlich steht die Nutzung der Plattform allen Personen offen; die primäre

Zielgruppe sind jedoch Schüler der Jahrgangsstufen fünf bis dreizehn, die gerne zusätzlich zum Unterrichtsstoff Aufgaben lösen, oder auch deren Lehrer, die sich auf diesem Wege Ideen für innovative Aufgabentypen und verblüffende Lösungswege holen können.

Die Entscheidung eine Onlineplattform zu schaffen um dieses Ziel zu erreichen liegt nicht zuletzt an der Tatsache, dass in einem Flächenland wie Bayern die Zahl der Schüler, die mit einem regional angebotenen Mathematikurs erreicht werden können, relativ beschränkt ist.

Nicht zuletzt sollen auch konkret Nutzer angesprochen werden, die sich auf einen Wettbewerb wie zum Beispiel die Mathematik-Olympiade vorbereiten. Nötig ist eine solche gezielte Vorbereitung auf diese Mathematikwettbewerbe, da nicht alle Techniken für die Lösung der dort gestellten Aufgaben auch im Schulunterricht vermittelt werden.

2 Funktionen

Dieser Abschnitt widmet sich den Funktionen, die auf der Onlineplattform angeboten werden. Die vorgestellten Funktionalitäten lassen sich grob in zwei Bereiche gliedern: solche, die ein Nutzer direkt auf der Plattform sehen und nutzen kann und solche, die im Hintergrund des Systems arbeiten und nur indirekt vom Nutzer verwendet werden wenn dieser bestimmte Aktionen auf der Nutzeroberfläche durchführt.

2.1 Infoseiten

Die Informationsseiten geben auf der Plattform selbst einen Überblick über die für den Nutzer verfügbaren Funktionen. Außerdem finden sich dort Anleitungen und Hilfestellungen zur Benutzung der Webseite als auch gesetzlich verlangte Informationen wie das Impressum[?].

Die Informationen und Erklärungen auf diesen Seiten sind statisch eingebunden. Die entsprechenden Texte können von einem Administrator

angepasst werden. Die unterschiedlichen Einzelseiten umfassen:

Home Die Startseite begrüßt die Nutzer und gibt einen groben Überblick welche Möglichkeiten *WorkSheetz* bietet

About Stellt weiterführende Informationen zum Projekt und dessen Zielsetzung bereit

Impressum Zeigt das vorgeschriebene Impressum an

AGB Zeigt die Nutzungsbedingungen der Webseite an

Kontakt Zeigt die Kontaktinformationen der Seitenadministration an

Hilfe Zeigt eine Anleitung, wie das System verwendet werden kann, an

Der Inhalt dieser Informationsseiten ist grundsätzlich statisch. Die angezeigten Informationen werden aus einer Datei geladen; welche Datei genau das sein soll kann ein Administrator in den Sprachdateien¹ konfigurieren.

2.2 Upload

Die Seite *Upload* bietet dem Nutzer die Möglichkeit Aufgabenstellungen hochzuladen. Dieser Prozess findet über zwei hintereinander geschaltete Seiten statt: Auf der ersten der beiden Seiten kann lediglich die Datei mit der Aufgabenstellung hochgeladen werden. Auf der zweiten Seite kann der Nutzer anschließend sowohl zusätzliche Informationen als auch Hinweise und Lösungsvorschläge zur Aufgabenstellung bereitstellen. Falls die hochgeladene Aufgabenstellung diese Daten bereits enthalten hat, so wurden sie während des Uploads extrahiert und als voreingestellte Werte in die Eingabemaske der zweiten Seite eingetragen. Bei den zusätzlichen Informationen zu einer Aufgabe handelt es sich um folgende Werte:

¹Vgl. Kapitel Mehrsprachigkeit

- Autor Der Autor bzw. die Autoren der Aufgabe
- Gebiet* Der Bereich der Mathematik aus dem die Aufgabenstellung stammt
- Teilgebiet Ein oder mehrere Teilgebiete, die in dieser Aufgabe bzw. den Lösungen dazu berührt werden
- Thema Die konkrete Problemstellung dieser Aufgabe
- Klassenstufe* Die empfohlene Klassenstufe, bezogen auf das Gymnasium, zur Bearbeitung
- Schwierigkeitsgrad* Einer von drei Schwierigkeitsgraden von *Leicht* über *Mittel* bis hin zu *Schwer*
- Staffellauf Eignet sich die Aufgabe für einen Staffellauf bzw. Mannschaftswettbewerb: die Lösung ist kurz und der Schwierigkeitsgrad übersteigt *Mittel* nicht
- Schulstoff Gibt an ob das Wissen aus dem Unterricht zur Lösung der Aufgabe ausreicht oder ob eine Einführung nötig ist
- Quellen Angaben zur Herkunft der Aufgabenstellung und ob die Aufgabe bereits in Wettbewerben verwendet worden ist
- Lizenz Gibt an unter welchen Bedingungen die Aufgabe verwendet und auch weiterverbreitet werden darf. Es stehen die sechs normalen Creative Commons (CC)-Lizenzen sowie zusätzlich die *CC0*-Lizenz zur Verfügung. Die Auswahl aus diesen sieben Lizenzen ermöglicht es einem Autor zielgenau zu bestimmen für welche Zwecke und unter welchen Bedingungen seine Arbeit weiterverwendet werden darf: Von der Bedingung lediglich als ursprünglicher Autor erwähnt zu werden über zusätzliche Einschränkungen betreffend der Bearbeitung bis hin zum Ausschluss einer kommerziellen Verwertung[?]. Die *CC0*-Lizenz stellt einen Sonderfall dar bei dem der Autor auf alle Rechte an seiner so veröffentlichten Arbeit verzichtet[?].

Sprache Gibt an in welcher Sprache die Aufgabenstellung verfasst ist

Attribute, die mit einem Stern (*) gekennzeichnet sind, sind dabei Pflichtangaben.

2.3 Suche

Die Suchfunktion gliedert sich in eine Schnellsuche und eine erweiterte Suche auf.

Die Schnellsuche befindet sich jederzeit erreichbar am oberen Bildschirmrand und besitzt lediglich ein Eingabefeld mit dem in den Attributen Autor, Gebiet, Teilgebiet, Thema und Quellen gesucht werden kann.

Die erweiterte Suche besitzt eine eigene Unterseite und ermöglicht es dem Nutzer alle Metainformationen, die während des Uploads angegeben wurden, zu durchsuchen. Hierbei gibt es ein Eingabefeld analog zur Schnellsuche sowie eine weitere Eingabemöglichkeit für jedes weitere durchsuchbare Attribut. Eine exemplarische Suche ist in Abbildung 1 auf Seite 9 zu sehen.

Die Schnellsuche ist intern als erweiterte Suche realisiert bei der alle Attribute, außer der Textsuche, außer Acht gelassen werden.

2.4 Aufgabenkorb

Der Aufgabenkorb zeigt alle vom Nutzer ausgewählten Aufgaben mit den jeweils wichtigsten Informationen an. Alle Elemente, die sich in diesem Korb befinden können als ein gemeinsames Aufgabenblatt im Portable Document Format (PDF)-Format heruntergeladen werden. Der Nutzer kann außerdem noch festlegen ob bei den Aufgaben die jeweiligen Hinweise oder Lösungen, soweit vorhanden, mit auf dem Aufgabenblatt erscheinen sollen oder nicht.

Vor dem Erstellen des Aufgabenblattes kann der Nutzer noch die Reihenfolge, in der die Aufgaben auf dem Blatt erscheinen, ändern oder auch

Custom WorkSheetz Home About Upload Aufgabenkorb Suche

Suchbegriff

Klassenstufe bis

Schwierigkeitsgrad

Lizenz

Staffellauf

Schulstoff

Impressum / AGB / Kontakt

Abbildung 1: Screenshot der Detailsuche bei Auflösung 1024x768

einzelne Aufgaben entfernen.

2.5 Systemfunktionen

Zusätzlich zu den vorher genannten Funktionen, die direkt vom Nutzer aufrufbar sind, bietet *WorkSheetz* noch weitere Features, die nicht direkt von der Benutzeroberfläche aus verwendet werden können aber dennoch wichtige Aufgaben im Hintergrund erfüllen.

2.5.1 L^AT_EX

Alle Aufgaben mitsamt den zugehörigen Hinweisen und Lösungen werden im System als L^AT_EX²-Quellcode gespeichert und verwaltet.

In der Datei, die eine Aufgabenstellung enthält, befinden sich zugleich speziell definierte Befehle, die die Zusatzinformationen zu dieser Aufgabe speichern. Ein solcher Abschnitt innerhalb des Dokuments könnte wie folgt aussehen:

```
\id{42}
\autor{Mustermann, Max}
\gebiet{strategien}
\teilgebiet{folgen}
\teilgebiet{induktion}
\thema{ganze Zahlen}
\klasse{7}
\schwierigkeit{MEDIUM}
\staffellauf{0}
\sprache{de}
\quellen{}
\schulstoff{0}
```

²Textsatzprogramm mit Auszeichnungssprache, vgl. <http://www.latex-project.org/>

```
\einstelldatum{1352024304481} % entspricht 04.11.2012
\lizenz{CC_BY_ND} % Namensnennung, Keine Bearbeitung
```

Bei mehrwertigen Attributen wird das Attribut entsprechend oft eingefügt mit jeweils einem Datum. Bei den mehrwertigen Attributen handelt es sich um `autor`, `teilgebiet` und `thema`.

Diese Informationen werden vom System genutzt um die Aufgabe intern zu verwalten und auch um sie über entsprechende Suchanfragen auffindbar zu machen.

Weitere Teilbereiche des Dokuments umfassen die eigentliche Aufgabenstellung sowie die Hinweise und Lösungsvorschläge soweit diese vorhanden sind.

```
\aufgabe{
Wie lautet die Endziffer von \
 $2^{2007}+3^{2007}+5^{2007}$ 
}
```

```
\hinweis{
% Diese Aufgabe hat keinen Hinweis
}
```

```
\loesung{
Die Lösung ist \texttt{0}
}
```

Beim Hochladen eines Dokuments können diese Attribute bereits im Quelltext angegeben sein, müssen es aber nicht. Es ist möglich eine Aufgabe hochzuladen, die keines dieser Attribute im Quelltext besitzt. Alle fehlenden Zusatzinformationen werden automatisch, durch die während des Uploads in den jeweiligen Eingabefeldern bereitgestellten Informationen, ergänzt.

2.5.2 SVN

Mithilfe von Apache Subversion (SVN) und des dazugehörigen Archivs können alle Änderungen an den Aufgaben sowie deren zusätzlichen Dateien protokolliert werden. Das Archiv zeichnet alle Änderungen an den darin verwalteten Dateien auf und kann auch mehrere Versionen ein und derselben Datei enthalten. Hierzu wird nach jeder Änderung an einer der Dateien im Verzeichnis, das die Aufgaben beinhaltet, das SVN-Kommando angestoßen, das die geänderten Dateien ins Archiv überträgt. Hierbei werden jeweils nur tatsächlich geänderte Inhalte übertragen und gesichert um Speicherplatz zu sparen. Auf diesem Wege ist jederzeit gewährleistet, dass eine Sicherungskopie der Aufgabensammlung existiert. Zusätzlich stellt die Protokollierung aller Änderungen eine Absicherung gegen destruktive Nutzer, die bestehende Aufgaben zerstören oder massenhaft sinnlose neue Aufgaben hochladen, dar. In einem solchen Fall kann der Administrator diese Änderungen komfortabel rückgängig machen[?]. Das SVN-Kommando wird immer vom System im Anschluss an den Upload einer Aufgabe ausgeführt um die Änderungen an das Archiv zu übertragen. Es wird außerdem einmalig beim Systemstart aufgerufen um die aktuellste Version aus dem Archiv abzurufen.

Um die Sicherheit der Daten nochmals zu erhöhen kann ein Administrator zusätzliche Backups des kompletten Archivs erstellen um auch die Änderungsprotokolle sowie die verschiedenen Versionen der geänderten Dateien zu erhalten.

2.5.3 Mehrsprachigkeit

Die Plattform wurde dahingehend konzipiert auch in anderen Sprachen als Deutsch betrieben werden und auch mehr als eine Sprache zur selben Zeit unterstützen zu können. Hierzu wurden die tatsächlich angezeigten Zeichenketten und Nachrichten nicht direkt in den Programmcode eingebettet sondern aus einer zentralen Datei geladen. Pro unterstützter

Sprache existiert jeweils eine solche Datei, die alle in die entsprechende Sprache übersetzten Strings enthält. Unterstützung für weitere Sprachen kann so je nach Bedarf einfach hinzugefügt werden.

Um eine neue Sprache zu unterstützen muss lediglich die Datei

`Messages_XY.properties`

mit den übersetzten Zeichenketten in den Ordner

```
./WEB-INF/classes/de/unipassau/ws/messages/
```

des Projektverzeichnisses kopiert werden. Die Zeichen `XY` innerhalb des Dateinamens sind dabei jeweils durch den ISO-Code für die darin enthaltene Sprache zu ersetzen. Beispiele hierfür sind *de* für Deutsch oder *en* für Englisch. Abschließend muss noch in der Datei `faces-config.xml` innerhalb des `<locale-config>` Bereiches eine neue Zeile für die nun verfügbare Lokalisierung angelegt werden[?]. Für die imaginäre Sprache mit dem ISO-Code `XY` sieht diese Zeile wie folgt aus:

```
<supported-locale>XY</supported-locale>
```

Während der Großteil des Inhalts der Sprachdateien normale Zeichenketten sind, die auch so dem Nutzer präsentiert werden, so gibt es für die Inhalte der statischen Informationsseiten, in den Sprachdateien selbst, keine explizit anzuzeigenden Informationen. Stattdessen werden in der Sprachdatei die Pfade zu den Dateien angegeben, die die eigentlichen Inhalte dieser Infoseiten enthalten. Die Pfade sind jeweils relativ zum Projektverzeichnis.

2.5.4 Konfiguration

Die Plattform kann von einem Administrator in vielen Punkten nach den jeweiligen Bedürfnissen konfiguriert werden. Alle entsprechenden Änderungen können in der Datei

`./WEB-INF/config.properties`

vorgenommen werden, die sich im Projektverzeichnis befindet. Die Einstellungen werden beim Systemstart geladen und während der Laufzeit im Speicher vorgehalten.

Für den grundlegenden Systembetrieb und die Hintergrundprozesse werden folgende Einstellungen geladen:

`svn.user` Der Benutzername am SVN Server

`svn.pass` Das dazugehörige Passwort

`svn.server` Die Adresse des SVN Servers

`svn.path` Kommando zum Ausführen von SVN

`latex.path.pdf` Kommando zum Ausführen des \LaTeX -Compilers für PDF Dateien

`latex.path.png` Kommando zum Ausführen der Umwandlung der PDF Datei in eine Bilddatei

`latex.path.header` Zum Projektverzeichnis relativer Pfad der \LaTeX -Headerdatei mit den Makrodefinitionen

`log.level` Genauigkeit der Ereignisprotokollierung. Je nach Schwere des Fehlers wird *SEVERE* oder *WARNING* verwendet; Information haben den Grad *INFO*. Der jeweils niedrigere Grad enthält auch die Nachrichten aller höheren Grade

`log.path` Der Order in dem die Protokolldatei erstellt werden soll

`exercises.path` Das Arbeitsverzeichnis des Systems in dem alle Aufgaben gespeichert werden. Innerhalb dieses Verzeichnisses besitzt jede Aufgabe einen eigenen Ordner

Folgende Einstellungen, die direkten Einfluss auf die Nutzung der Plattform haben, werden geladen:

exercises.mingrade Die minimal wählbare Klassenstufe

exercises.maxgrade Die maximal wählbare Klassenstufe

exercises.topics Eine Liste der Schlüssel aller auswählbaren Gebiete. Einzelne Gebiete werden durch einen Strichpunkt getrennt. Jeder Schlüssel in der Liste benötigt eine Übersetzung in den Sprachdateien des Systems in der Form `topic_{Schlüsselname}`

exercises.subtopics Eine Liste der Schlüssel aller auswählbaren Teilgebiete. Einzelne Teilgebiete werden durch einen Strichpunkt getrennt. Jeder Schlüssel in der Liste benötigt eine Übersetzung in den Sprachdateien des Systems in der Form `subtopic_{Schlüsselname}`

3 Implementierung

Zur Umsetzung der Plattform wurde das Framework JavaServer Faces (JSF) gewählt, das der de-facto Standard der Webentwicklung in Java ist[?].

JSF erzwingt die Verwendung des Model View Controller (MVC)-Musters, das eine Aufgabenteilung zwischen *Model* mit der Anwendungslogik, *View* mit der Darstellung der Inhalte und *Controller* mit der Steuerung vorsieht. Diese Trennung ermöglicht es einzelne Teile auszutauschen ohne das komplette System ändern zu müssen[?].

Das JSF-Framework übernimmt lediglich die Aufgabe des Controllers und delegiert die Seitenaufrufe und Formulareingaben des Nutzers an die jeweils verantwortlichen *Backing Beans*, welche die eigentliche Logik des Systems enthalten. Die Schnittstelle, zwischen dem vom Framework bereitgestellten Controller und dem Model bzw. der View, ist die Datei

```
./WEB-INF/faces-config.xml
```

anhand deren Einstellungen der Controller weiß wie die Seitenaufrufe der Nutzer weitergeleitet werden müssen. Während des Betriebs der Webseite

ist keine Anpassung dieser Datei notwendig; lediglich bei einer Erweiterung oder Veränderung des Systems muss diese Datei bearbeitet werden.

Die gesamte Nutzeroberfläche der View, die Hintergrundlogik des Modells sowie die Konfiguration der notwendigen Schnittstellen wurden während dieser Arbeit anhand der Zielsetzung des Projektes entwickelt.

3.1 Nutzeroberfläche

Die Nutzeroberfläche wird, wie bei Onlineplattformen üblich, im Browser dargestellt. Die dazu verwendete Auszeichnungssprache ist Extensible Hypertext Markup Language (XHTML). In ihr werden die, zumeist dynamisch von der Programmlogik im Model des Servers generierten, Inhalte eingebettet. Die grafische Darstellung der angezeigten Webseite wird durch die Verwendung von Cascading Style Sheets (CSS) in Version 3 in gewissem Umfang angepasst um eine schönere Gestaltung der Inhalte zu erreichen.

Die Struktur jeder Unterseite leitet sich von einer generischen Vorlage ab um dem Nutzer über alle Teile der Plattform hinweg eine einheitliche Navigation zu ermöglichen. Die Datei

```
./pages/template.xhtml
```

im Projektverzeichnis definiert dabei die drei Bereiche des Layouts in welche die tatsächlichen Inhalte geladen werden: Menüleiste, Inhaltsbereich und Fußleiste. Die Menüleiste am oberen sowie die Fußleiste am unteren Rand der Webseite sind immer sichtbar und können von jeder Unterseite heraus angesteuert werden. Hier befinden sich alle wichtigen Links mit Hilfe derer der Nutzer jederzeit in die verschiedenen Teile der Webseite navigieren kann. Der Bereich zwischen der Menü- und Fußleiste zeigt den jeweils aktuellen Inhalt der Seite an.

Bei der grafischen Gestaltung der Webseite wurde darauf geachtet die Inhalte in einer möglichst gut lesbaren Art und Weise zu präsentieren.

Hierzu wurden unter anderem die verwendeten Schriftarten und Schriftgrößen angepasst als auch Eingabefelder und -knöpfe grafisch hervorgehoben. Außerdem wurden die Menü- und Fußleiste am oberen bzw. unteren Bildschirmrand fixiert und farblich vom Inhaltsbereich der Webseite abgesetzt.

In der Breite wurde die Webseite auf eine Bildschirmauflösung von 1024 Pixeln optimiert, so dass sowohl auf kleineren Monitoren alle Inhalte sichtbar sind als auch bei hohen Auflösungen das Lesen der Inhalte, aufgrund des zu lang gezogenen Textes, nicht erschwert wird. Das Seitenlayout mit Menü- und Fußleiste sowie dem gesondert hervorgehobenen Inhaltsbereich sind in Abbildung 2 auf Seite 18 zu sehen.

Durch den ausschließlichen Einsatz von standardisierten und weit verbreiteten Technologien für die formale Beschreibung der Nutzeroberfläche wird eine konsistente und fehlerfreie Darstellung der Webseite auch auf verschiedenen Endgeräten, Browsern und Betriebssystemen sichergestellt.

3.2 Datenhaltung

Ein wichtiges Kriterium für den Betrieb der Plattform war die Sicherheit der damit verwalteten Daten bzw. Aufgaben. Aus diesem Grund wurde als Speichermedium für die Aufgaben keine Datenbank gewählt, sondern einzelne Dateien auf der Festplatte, die automatisch nach jeder Änderung über SVN gesichert werden. Für jede Aufgabe werden die folgenden Dateien in einem jeweils separaten Unterverzeichnis des Arbeitsverzeichnis gehalten:

`exercise.tex` Diese Datei enthält den \LaTeX -Quelltext der Aufgabenstellung, die Metainformationen der Aufgabe sowie die eventuell vorhandenen Hinweise und Lösungsvorschläge

`preview.png` Diese Datei ist die vom \LaTeX -Interpreter ausgegebene Version der Aufgabenstellung, der Hinweise und Lösungswege. Sie wird zur Vorschau der Aufgabe auf der Webseite genutzt



Abbildung 2: Seitenlayout mit gesondert hervorgehobenem Inhaltsbereich bei Auflösung 1920x1080

[...] Weitere von der Aufgabenstellung eingebundene Dateien; zum Beispiel Zeichnungen von Graphen oder geometrischen Figuren

Obwohl die Daten komplett auf Festplatte vorliegen nutzt das System eine interne Datenbank um Suchanfragen besonders schnell ausführen zu können. Bei den dort gespeicherten Daten handelt es sich nicht um die tatsächlichen Aufgabenstellungen sondern lediglich um die Metainformationen der jeweiligen Aufgaben. Beim Systemstart wird diese Datenbank neu erstellt um die Konsistenz, mit den tatsächlich im Arbeitsverzeichnis auf der Festplatte verfügbaren Aufgaben, zu gewährleisten.

Es kommt dabei die HyperSQL DataBase (HSQLDB) zum Einsatz, die eine performante relationale SQL Datenbank in Java bereitstellt[?]. Die während der Laufzeit in der Datenbank vorgehaltenen Informationen befinden sich in insgesamt vier Tabellen. Die Tabellen `ws_teilgebiete`, `ws_themen` und `ws_authors` haben jeweils die Spalte `exercise_id` aus der Tabelle `ws_exercises` als Schlüssel gemeinsamen mit ihrer jeweils eigenen Spalte. Die mehrwertigen Attribute Autor, Teilgebiet und Thema wurden, wie aus Abbildung 3 hervorgeht, in eigene Tabellen ausgelagert. Die bei der Suche nötigen JOIN-Operation können aufgrund der Indizierung der Tabellen mit einem gemeinsamen Schlüssel performant durchgeführt werden.

3.3 Systemstruktur

Alle `*.xhtml`-Seiten befinden sich im Ordner `./pages/` im Projektverzeichnis während sich die Backing Beans mit dem Code und der Programmlogik im Ordner `./WEB-INF/classes/` befindet. Jede Unterseite, die Programmlogik für die Nutzer bereitstellt, steht in einer 1 : 1-Beziehung zu einer Backing Bean im Model. Hierunter fallen alle Seiten mit Ausnahme derjenigen, die das Layout der Webseite definieren und die statischen Inhalte für die Informationsseiten speichern.

Die CSS-Dateien, die die Definitionen für die grafische Gestaltung der

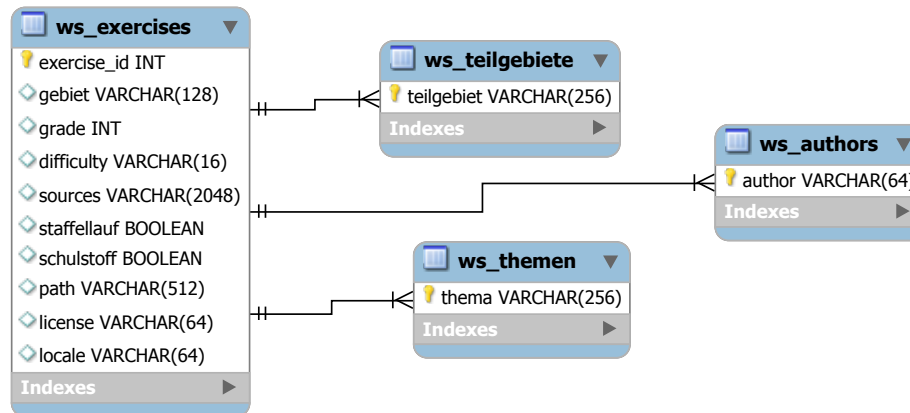


Abbildung 3: Schema der internen Datenbank

Webseite beinhalten, befinden sich in dem Ordner `./resources/css/` im Projektverzeichnis.

Die XHTML-Datei stellt die in der dazugehörigen Backing Bean gespeicherten Informationen dar. Ebenso werden Nutzereingaben auf der Webseite an die jeweilige Backing Bean übertragen wo diese verarbeitet werden. Beim Aufruf der XHTML-Seite wird automatisch im Hintergrund von JSF die dazugehörige Backing Bean erstellt.

Zusätzlich zu den bereits beschriebenen Teilen der View und des Models, die in den Arbeitszyklus von JSF eingebettet wurden, sind auch zwei weitere Klassen im Controller implementiert worden.

3.3.1 DownloadServlet

Das *DownloadServlet* liefert sowohl die PDF oder ZIP Dateien, die ein Nutzer erstellt hat, als auch die Vorschaubilder der einzelnen Aufgaben aus. Es extrahiert hierfür aus der übergebenen Uniform Resource Locator (URL) die zwei Parameter `type` und `id` anhand deren der Speicherort der auszuliefernden Datei eindeutig bestimmt werden kann:

- Die Vorschaubilder befinden sich immer unter dem selben Namen,

`preview.png`, im Verzeichnis der Aufgabe mit der eindeutigen, unter `id` angegebenen, Nummer

- Die ZIP-Dateien befinden sich immer im temporären Verzeichnis des Betriebssystems und ihr Dateiname ist mit der unter `id` gegebenen Nummer bezeichnet
- Die PDF-Dateien befinden sich ebenfalls im temporären Verzeichnis des Betriebssystems, allerdings in einem gesondert erstellten Ordner, dessen Name von der übergebenen `id` abhängig ist

Ein solcher Downloadlink könnte wie folgt aussehen:

`<Servername>.<tld>/WorkSheetz/DownloadPackage?type=PDF&id=42`

3.3.2 ServerStartStopListener

Der *ServerStartStopListener* ist ein *WebListener*, der sich durch die Annotation `@WebListener()` in der Java-Klassendatei automatisch bei der Ausführungsumgebung (z.B. Tomcat) registriert. Der Listener wird nun durch einen Funktionsaufruf informiert wenn das System gestartet bzw. gestoppt wird. Beim Systemstart werden aus der Konfigurationsdatei `config.properties` die Administrationseinstellungen ausgelesen und die extrahierten Werte zur Initialisierung verschiedener Klassen im System verwendet. Die Daten in der Konfigurationsdatei liegen als Schlüssel-Wert-Relationen mit einem Gleichheitszeichen als Trennzeichen vor.

3.4 Serverumgebung

Bei der Auswahl der tatsächlich verwendeten Implementierung von JSF muss lediglich darauf geachtet werden, dass die entsprechende Implementierung JSF 2.0 und Servlet 3.0 unterstützt. *WorkSheetz* arbeitet derzeit mit *Apache MyFaces*, das eine robuste Grundlage für die Plattform bietet und außerdem, über *MyFaces Tomahawk*, eine native Möglichkeit des Dateiuploads auf den Server zur Verfügung stellt.

Als Ausführungsumgebung kann *Apache Tomcat* ab Version 7 oder jede andere Umgebung, die Java 1.6 und die Voraussetzungen der gewählten JSF Implementierung unterstützt, verwendet werden.

Die Schnittstelle zwischen der Ausführungsumgebung und JSF ist die Datei `./WEB-INF/web.xml` im Projektverzeichnis. In ihr wird das JSF-Framework geladen als auch die Startseite der Plattform eingestellt. Außerdem wird hier das *DownloadServlet* registriert mit dem der Nutzer die PDF- und ZIP-Dateien herunterladen kann. Dazu wird zunächst die tatsächliche Klasse mit einem Namen versehen:

```
<servlet>
<servlet-name>DownloadServlet</servlet-name>
<servlet-class>
de.unipassau.ws.controller.DownloadServlet
</servlet-class>
</servlet>
```

Anschließend wird festgelegt bei dem Aufruf welcher URL das mit diesem Namen versehene Servlet aufgerufen werden soll:

```
<servlet-mapping>
<servlet-name>DownloadServlet</servlet-name>
<url-pattern>/DownloadPackage</url-pattern>
</servlet-mapping>
```

Analog hierzu ist auch die Einbindung von JSF in die Ausführungsumgebung realisiert. Bei der Festlegung der URL, bei der JSF aktiv werden soll, wurde allerdings kein statischer Wert verwendet sondern ein Muster, das alle XHTML-Seiten zur Anzeige über JSF ausliefert:

```
<url-pattern>*.xhtml</url-pattern>
```

Durch die Wahl von JSF - und damit effektiv Java - als Basis für die Plattform kann diese auf praktisch allen modernen Webservern ausgeführt

werden, da die Grundvoraussetzung - eine installierte Java Runtime Environment (JRE) - für alle gängigen Betriebssysteme erfüllt ist.

3.5 \LaTeX

Die auf der Plattform verfügbaren Aufgaben, sowie die jeweils dazugehörigen Hinweise und Lösungsvorschläge, liegen als \LaTeX -Code vor. Neue Aufgabenstellungen, die ein Nutzer hochlädt, müssen ebenfalls in diesem Format abgefasst sein. Auch wenn durch die Festlegung auf \LaTeX , aufgrund der längeren Einarbeitungszeit, einige potenzielle Autoren abgeschreckt werden könnten, so überwiegen dennoch die Vorteile dieser Entscheidung. \LaTeX bietet, insbesondere für das Erstellen von mathematischen Texten, eine komfortable Möglichkeit Formeln zu schreiben und in den Fließtext zu integrieren.

Jeder Nutzer kann die Aufgaben zur weiteren Verwendung entweder als \LaTeX -Quellcode oder als fertig kompiliertes PDF herunterladen. Eine Aufgabe, die als Quellcode heruntergeladen wurde, kann der Nutzer bearbeiten, um z.B. einen Fehler zu korrigieren oder eine Formulierung zu präzisieren, und anschließend auf die Plattform hochladen um die Aufgabe auch für alle anderen Nutzer zu aktualisieren. Die Entscheidung, ob eine hochgeladene Aufgabe neu ist oder die Aktualisierung einer bereits Bestehenden darstellt, wird aufgrund des Vorhandenseins einer eindeutigen Identifikationsnummer gefällt. Falls es sich um eine neue Aufgabe handelt werden die vom Nutzer eingegebenen zusätzlichen Informationen wie z.B. Autor, Schwierigkeitsgrad und Klassenstufe mit der eigentlichen Aufgabenstellung, den Hinweisen und der Lösung zusammengeführt und abgespeichert.

3.5.1 Datenextraktion

Beim Hochladen einer Aufgabe wird die komplette Datei analysiert und in eine interne Repräsentation überführt. Dabei werden alle Attribute ent-

weder mit den vorhandenen Daten aus der hochgeladenen Datei oder, falls die entsprechenden Werte nicht vorhanden sind, mit einem Platzhalter befüllt. Die gespeicherten Daten umfassen alle bereits im Kapitel Upload auf Seite 6 aufgeführten Attribute und zusätzlich noch den vollständigen Pfad zu dem Ordner in dem die Aufgabenstellung auf der Festplatte gespeichert ist als auch das Datum der Erstellung der Aufgabe. Diese beiden zusätzlichen Werte werden erst nach dem Upload im System gesetzt.

Die einzelnen Attribute werden aus der hochgeladenen Datei durch die Verwendung von *regulären Ausdrücken* extrahiert. Alle im Dokument gespeicherten Werte weisen das folgende Muster auf:

```
\Attributname{Wert (auch mehrzeilig)}
```

Die Begrenzung erfolgt durch die geschweiften Klammern. Innerhalb des Wertes können weitere geschweifte Klammern auftreten. Eine solche Sprache ist allerdings nicht mehr regulär und kann daher auch nicht mehr mit einem regulären Ausdruck vollständig korrekt erkannt werden. Um dennoch die Werte der verschiedenen Attribute schnell innerhalb des Dokument finden zu können, wird eine maximale Verschachtlung von geschweiften Klammern angenommen und zum Programmstart ein entsprechender regulärer Ausdruck generiert. Die maximal angenommene Verschachtelung kann dabei im Quelltext eingestellt werden. Der reguläre Ausdruck wird anhand folgender Grammatik aufgebaut:

$G = (N, T, P, S)$

$T = \{ '(, ', ')', '\{, \}', '\.', '*', '\?', '\\'\} \cup$
 $\{\text{Namen aller Attribute}\}$

$N = \{S, A, B\}$

$P = \{$

$S \rightarrow \\A\{B\}$

$A \rightarrow \text{Namen eines Attributs}$

$B \rightarrow ((.*?)(\{B\})?)*$

$B \rightarrow .*?$

}

Die Anzahl der unterstützten verschachtelten Klammerungen ist davon abhängig wie oft die erste Produktion von `B` gewählt wird bevor mit der zweiten Produktion von `B` die Rekursion beendet wird. Die Namen aller Attribute sind grundsätzlich frei wählbar, allerdings für die konkrete Implementierung der Plattform auf die im Beispieldokument auf Seite 10 verwendeten Namen festgelegt.

Bei jedem Systemstart wird, wie im Abschnitt Datenhaltung bereits erwähnt, die Datenbank neu aufgebaut. Hierbei wird das Arbeitsverzeichnis der Plattform durchlaufen und alle darin enthaltenen Aufgaben analysiert und deren Attribute analog zur Methode, die auch bei hochgeladenen Dateien angewandt wird, extrahiert und in die Datenbank eingefügt.

3.5.2 Download von Aufgaben

PDF Wenn ein Nutzer ein PDF-Dokument aus den in seinem Aufgabenkorb liegenden Dokumenten erstellt, wird im temporären Verzeichnis des Servers ein neuer Ordner erstellt und innerhalb dieses Ordners eine neue Datei in der alle im Aufgabenkorb des Nutzers ausgewählten Aufgaben mit dem `\input`-Befehl eingebunden werden. Um sicherzustellen, dass der gewählte Ordnername eindeutig ist, wird der aktuelle Zeitstempel in Millisekunden mit in den Namen integriert und außerdem im Code, der den Namen generiert, durch das `synchronized`-Schlüsselwort sichergestellt, dass keine zwei Ordner zur selben Zeit erstellt werden können.

In die temporäre \LaTeX -Datei wird zunächst der Inhalt der vom System verwendeten Makro-Datei kopiert. In ihr werden unter anderem drei Schalter definiert, die dazu dienen die Anzeige von Aufgabenstellung, Hinweisen und Lösung jeweils einzeln an- und abzuschalten. Die Definition eines solchen Schalter sieht, exemplarisch am Beispiel der Aufgabenstellung, wie folgt aus:

```
\newif\ifSHOWEXERCISE
```

```
\newcommand{\aufgabe}[1]{  
  \ifSHOWEXERCISE{#1}\else{}\fi  
}
```

Alle Tags, die lediglich der Speicherung von zusätzlichen Informationen dienen, werden so definiert, dass sie nicht in der Ausgabe auftauchen wie hier am Beispiel der eindeutigen Identifikationsnummer gezeigt:

```
\newcommand{\id}[1]{}
```

Der nächste Schritt bei der Erstellung des zu kompilierenden Dokuments ist das Setzen der Schalter anhand der vom Nutzer eingestellten Werte wie z.B.:

```
\SHOWEXERCISEtrue  
\SHOWHINTSfalse  
\SHOWSOLUTIONtrue
```

Im nun folgenden `document` Bereich der Datei wird der Titel des Arbeitsblattes, sofern der Nutzer einen solchen angegeben hat, innerhalb des Tags `abtitel` platziert, der in der Makrodatei standardmäßig als

```
\newcommand{\abtitel}[1]{\centering{\huge{#1}}}
```

definiert ist. Die Definition eines eigenen Tags für den Titel des Arbeitsblattes bietet den Systemadministratoren in Zukunft eine komfortable Möglichkeit das Erscheinungsbild durch eine kleine Änderung der Makrodatei anzupassen.

Anschließend werden noch alle ausgewählten Aufgaben innerhalb einer `enumerate`-Umgebung per `input` als jeweils eigenständiges Element eingebunden.

Abschließend wird der Befehl zum Kompilieren des Dokuments aufgerufen. Der dazu gestartete Prozess erhält in den übergebenen Umgebungsvariablen den Schlüssel `TEXINPUTS` mit einer Auflistung aller Quellverzeichnisse, so dass die per `input` eingebundenen Aufgabendateien auch

korrekt inkludiert werden können. Sobald der externe Prozess geendet hat wird überprüft ob dort ein Fehler aufgetreten ist, was durch einen Rückgabewert ungleich Null signalisiert werden würde. Falls ein Fehler aufgetreten ist, wird der Nutzer darüber informiert, andernfalls wird ihm die erstellte PDF-Datei zum Herunterladen angeboten.

Quelltext Ein Nutzer kann außerdem den \LaTeX -Code aller Aufgaben in seinem Aufgabenkorb herunterladen. Dies schließt auch weitere von den Aufgaben verwendete Dateien, wie z.B. Skizzen von geometrischen Figuren, ein. Die ausgewählten Dateien werden dem Nutzer in einem dynamisch generierten ZIP-Archiv zum Download angeboten. Hierbei werden die Aufgaben und zusätzlichen Dateien in ihren jeweiligen Ordnern belassen und komplett in das Archiv gepackt. Auch die Attribute, mithilfe derer die Aufgaben intern verwaltet und kategorisiert werden, bleiben in den heruntergeladenen Aufgabendateien bestehen. Neben den eigentlichen Aufgabenstellungen wird zusätzlich noch eine Datei erstellt und gepackt, die analog zu der temporären Datei bei der Erstellung des PDF-Dokuments, alle Makros sowie die einzelnen Aufgaben als Input enthält, so dass der Nutzer die Aufgaben auch lokal auf seinem Rechner kompilieren kann.

3.6 Sicherheit

WorkSheetz ist frei über das Internet zugänglich und wird daher früher oder später das Ziel eines Angriffs werden. Dabei wird in der Regel versucht über Schwachstellen in der Programmierung der Plattform an nicht-öffentliche Informationen zu gelangen oder auch einfach nur möglichst viel Schaden am System anzurichten. Der Ansatzpunkt solcher Angriffe sind immer Nutzereingaben, sei es nun ein Texteingabefeld, eine URL mit Übergabeparametern oder ein Dateiupload.

Alle Nutzereingaben werden auf potentiell schädliche Eingaben über-

prüft und diese gegebenenfalls eliminiert. Als Beispiel lässt sich hier die Suchfunktion und deren Freitextfeld anführen: Die Datenbanksoftware kennt bestimmte Sonderzeichen (z.B. Anführungszeichen oder Prozentzeichen), die wenn sie ungefiltert in eine Datenbankabfrage eingefügt werden, dazu führen können, dass ein Nutzer im schlimmsten Fall beliebige Befehle auf der Datenbank ausführen kann. Bevor die Nutzereingabe in die Datenbankabfrage eingefügt wird, werden diese Sonderzeichen durch den regulären Ausdruck `[\'|\"|%]` entfernt.

Eine andere Stelle, an der die Nutzereingaben geprüft werden müssen, ist die Erstellung eines Aufgabenblattes. Hier kann in einem Freitextfeld der gewünschte Titel des zu erstellenden Arbeitsblattes eingegeben werden. Da dieser in das \LaTeX -Dokument kopiert wird, darf er allerdings keine dort als Sonderzeichen definierten Zeichen enthalten. Um solche Eingaben herauszufiltern wird ein *Validator* verwendet. Der Validator überprüft bereits sehr früh, noch bevor die eigentliche Programmlogik im Model aufgerufen wird, ob die Eingabe den im Validator implementierten Spezifikationen genügt. Falls dies nicht der Fall ist, wird eine Ausnahme ausgelöst und die Programmlogik gar nicht erst aufgerufen. Außerdem wird der Nutzer mit einer Fehlermeldung auf die ungültige Eingabe aufmerksam gemacht. Der Validator wird dabei durch die Verwendung des folgenden Attributes im Quelltext der Seite selektiv für ein spezielles Eingabefeld spezifiziert:

```
<f:validator validatorId="SheetTitleValidator" />
```

Die dort angegebene *validatorId* wird automatisch der Klasse mit der Annotation

```
@FacesValidator(value = "SheetTitleValidator")
```

zugeordnet. In dieser Klasse wird in ihrer `validate(...)`-Methode die tatsächliche Überprüfung der Nutzereingabe vorgenommen.

Ein anderes Angriffsszenario, Cross-Site-Scripting (XSS), zielt darauf ab ausführbare Inhalte - in der Regel JavaScript - auf die Webseite hochzuladen, so dass andere Nutzer, die diesen Inhalt später betrachten, den Schadcode unbewusst ausführen. Ein solcher Angriff wird durch den konsequenten Einsatz des `outputText`-Elements von JSF in den XHTML-Dateien, die für die Darstellung der Inhalte verantwortlich sind, verhindert. Alle HTML-Sonderzeichen, die in einem über dieses Element ausgegeben Text enthalten sind, werden automatisch in eine harmlose Darstellung umgewandelt.

3.7 Fehlerbehandlung

Während des Betriebes eines Softwaresystems ist es im Allgemeinen unvermeidlich, dass Fehler auftreten. Diese Fehler werden, falls eine Erholung davon möglich ist, abgefangen. Einen so gefangenen Fehler wird in aller Regel mit dem Grad **SEVERE** protokolliert und dem Nutzer wird eine Nachricht angezeigt, dass seine Aktion aufgrund eines Fehlers nicht abgeschlossen werden konnte. Diese Nachricht an den Nutzer enthält keine systemrelevanten Informationen, die Rückschlüsse auf die verwendeten Technologien zulassen um einem möglichen Angreifer keinen Ansatzpunkt zu bieten.

4 Vergleich

WorkSheetz ist nicht die erste Onlineplattform mit der Idee mathematische Aufgaben online zu verwalten und zum Download anzubieten. Hier wird *WorkSheetz* mit weiteren Aufgabensammlungen, die sowohl in Buchform oder auch online vorliegen, und anderen Formen den Austauschs von Mathematikaufgaben verglichen.

4.1 SMART

Zunächst soll die Plattform mit der Sammlung Mathematischer Aufgaben als Hypertext mit T_EX (SMART), die von der Universität Bayreuth und dem Zentrum zur Förderung des mathematisch-naturwissenschaftlichen Unterrichts (Z-MNU) betrieben wird, verglichen werden. SMART stellt nach eigenen Angaben mehr als 5500 Aufgaben in den Bereichen Mathematik und Physik für das Gymnasium und Mathematik für die Realschule bereit. Es ist ebenfalls eine Onlineplattform, die den Download von Aufgabenblättern im PDF-Format ermöglicht[?].

SMART setzt bei der Erstellung von neuen Aufgaben auf ein kleines Team und bietet interessierten Helfern keine Möglichkeit direkt online neue Inhalte hochzuladen. Potentielle Autoren werden dazu aufgefordert sich bei der Leiterin des Arbeitskreises bzw. bei der Z-MNU zu melden. An dieser Stelle setzt *WorkSheetz* auf einen leichten Zugang für neue Autoren und eine möglichst geringe Einarbeitungszeit in die Erstellung von Aufgaben, die direkt in die Plattform eingepflegt werden können. Durch diese Kultur der Offenheit gegenüber seinen Nutzern soll eine lebendige Gemeinschaft entstehen, die die Sammlung pflegt und erweitert.

SMART bietet dem Nutzer die Möglichkeit die Aufgaben über eine mehrstufige Baumstruktur zu durchsuchen. Hierbei geht allerdings, aufgrund der Feingliedrigkeit der angebotenen Möglichkeiten und der fehlenden Anzeige der Position innerhalb des Baumes, schnell die Übersicht verloren. Außerdem ist die Einordnung von Aufgaben oft nicht eindeutig möglich wodurch ein Nutzer eventuell, für ihn interessante Inhalte, nicht findet. Die zweite von SMART angebotene Möglichkeit die Aufgaben zu finden, ist eine Suchfunktion, die allerdings nur eine Volltextsuche mit einem weiterem Kriterium, der Klassenstufe, erlaubt. Hier ist zu erwarten, dass - je nach gewähltem Suchbegriff - entweder eine deutlich zu große oder zu kleine Menge an Ergebnissen gefunden wird.

WorkSheetz verzichtet auf eine Option die Aufgabensammlung durch

eine Baumstruktur zu erforschen sondern setzt auf eine gut ausgearbeitete Suchfunktion. Der Nutzer kann sowohl eine detaillierte Suche als auch eine Schnellsuche, deren Ergebnisse anschließend mit einer Detailsuche verfeinert werden können, durchführen. Die erweiterte Suchfunktion bietet dem Nutzer die Möglichkeit genau diejenigen Aufgaben zu finden, die er benötigt, indem er den Suchbereich aller vorhanden Attribute passend eingrenzt.

4.2 Mathematik-Wettbewerbe

Es gibt zahlreiche mathematische Wettbewerbe deren Teilnehmerfeld sich von Grundschulern bis zu Schülern der Oberstufe erstreckt. Diese Wettbewerbe wie zum Beispiel der *Landeswettbewerb Mathematik Bayern* oder der *Bundeswettbewerb Mathematik* sowie die verschiedenen Runden der Mathematikolympiade - von Schulrunden über Regional- und Landesrunden bis hin zur Bundesrunde - bieten eine oft bunte Mischung von verschiedenen Aufgaben und Themengebieten. Die besten Teilnehmer der Bundesrunde der Mathematikolympiade dürfen sogar in der deutschen Mannschaft an der Internationalen Mathematikolympiade teilnehmen[?].

Die meisten dieser Wettbewerbe veröffentlichen die gesammelten Aufgaben eines Jahres mitsamt deren Lösungen auf ihren Internetauftritten oder auch in Buchform. Eine solche Aufgabensammlung ist in der Regel nach den verschiedenen Runden des Wettbewerbes und Klassenstufen sortiert. Das gezielte Üben eines bestimmten Aufgabentyps ist nicht möglich.

WorkSheetz dahingegen bietet eine Sortierung der Aufgaben in verschiedene Gebiete, Teilgebiete und Themen an womit ein Nutzer über die Suchfunktion gezielt Aufgabentypen oder Problemstellungen finden kann, die ihn besonders interessieren oder herausfordern. Zudem kann die Plattform in aller Regel mit einer deutlich größeren Auswahl von Aufgaben aufwarten als ein Buch es je könnte.

4.3 Lösungsstrategien

Auf dem Markt für Mathematikliteratur gibt es unter anderem Bücher, die gezielt verschiedene Strategien zur Lösung von Aufgaben besprechen und anhand von Beispielen die verschiedenen zu Grunde liegenden Prinzipien erklären. In der Regel erfolgt diese Besprechung anhand von bereits bekannten Wettbewerbsaufgaben, die allerdings, im Gegensatz zu einer einfachen Sammlung derselben, nach dem anwendbaren Lösungsweg geordnet sind.

Solche Bücher bieten zwar einen guten Einstieg in die Materie und auch einen einfachen Weg möglichst viele verschiedene Techniken zumindest anzuschneiden. Letztendlich ist allerdings die Menge der darin enthaltenen Aufgaben begrenzt und auch eine Mitarbeit an den Aufgabenstellungen und Lösungsvorschlägen, durch das Klarstellen von unter Umständen missverständlich formulierten Passagen oder dem Hinzufügen eines neuen Lösungsvorschlages, ist nicht möglich[?].

Die in Buchform gereichten Einführungen in Lösungsstrategien haben sicherlich ihre Daseinsberechtigung, allerdings entsteht erst durch die Zusammenarbeit von vielen engagierten Nutzern ein großartiges Ergebnis: eine von der Community aktiv gepflegte Plattform kann stets die aktuellsten Aufgaben zur Verfügung stellen. Auch hier kann eine solche Plattform durch die potentiell unbegrenzte Menge an verfügbaren Aufgaben für jeden Schwierigkeitsgrad den Nutzern die Möglichkeit bieten immer wieder etwas neues zu entdecken.

5 Resümee

WorkSheetz ist alles in allem eine solide Plattform für den Austausch von und über mathematischen Aufgaben. Die Benutzung der Webseite alleine für die Suche und den Download von Aufgaben ist intuitiv möglich, so dass einem schnellen Einstieg in die Übungen nichts im Wege steht. Eben-

so erfordert das Hochladen von Aufgaben keine besonderen Kenntnisse und auch die Quelldateien müssen vor dem Upload nicht speziell angepasst werden. Das Ziel eine leichte Benutzbarkeit der Seite zu gewährleisten kann als vollständig erreicht betrachtet werden und, verbunden mit den geringen Anforderungen an Autoren, darf davon ausgegangen werden, dass sich relativ viele Nutzer dazu begeistern lassen werden an der Aufgabensammlung mitzuarbeiten. Eben diese Nutzer, die an der Pflege der Aufgaben mitarbeiten, sind die wichtigste Komponente der ganzen Plattform. Um sie zu unterstützen wäre es denkbar eine Möglichkeit der direkten Kommunikation unter ihnen zur Verfügung zu stellen z.B. durch die Integration einer Diskussionsplattform oder von Social-Media-Plugins, die die Seite um eine Kommentarfunktion oder ähnliches ergänzen. Zusätzlich könnte ein Accountsystem eingeführt werden um eine gewisse Übersicht über die verschiedenen Autoren zu erhalten und auch um diese besser an die Plattform zu binden.

Eine Umsetzungsmöglichkeit wäre etwa die Möglichkeit sich via *Facebook Connect* auf der Plattform einzuloggen wodurch sowohl ein Accountsystem als auch die Kommunikationsmöglichkeit zwischen den Nutzern der Plattform integriert wäre. Eine solche Lösung würde auch die Bekanntheit der Plattform steigern, sobald die Facebook-Freunde eines Nutzers dessen Aktivität auf *WorkSheetz* auf seiner Pinnwand sehen[?]. Selbstverständlich ist auch die Einbindung von anderen sozialen Netzwerken, die eine solche Möglichkeit für externe Webseiten anbieten, möglich.

Eine Erweiterung der Plattform hinsichtlich des unterstützten Themenspektrums ist ebenfalls denkbar: obwohl die Plattform derzeit ausschließlich zur Verwaltung von mathematischen Aufgaben gedacht ist, so ist es mit nur geringem Aufwand möglich das ganze System so anzupassen, dass auch komplett andere Fachgebiete damit arbeiten können. Hier kommt ein großer Vorteil von \LaTeX , auf dem die angebotenen Aufgaben basieren, zum Tragen: Es lässt sich mit nur wenigen Handgriffen um viele Zusatzpakete erweitern, die z.B. Unterstützung für Notensatz oder

Lautschrift mitbringen. Um solche neue Einsatzgebiete unterstützen zu können muss nur das passende Paket, das diese Funktionalität bietet, zur Installation des L^AT_EX-Interpreters hinzugefügt werden. Einfache thematische Anpassung z.B. der Eingabeaufforderung beim Dateiupload runden die Erweiterung auf der Nutzeroberfläche ab.

Bei einer zukünftigen Erweiterung der Plattform ist das Hinzufügen einer Möglichkeit eines Massenimports von Aufgaben aus anderen Sammlungen, die in einem inkompatiblen Format vorliegen, ebenfalls ein Ansatzpunkt. Hierzu müsste ein Modul erstellt werden, welches das Aufgabenformat, in dem die hinzuzufügenden Aufgaben vorliegen, analysieren kann. Sobald die Inhalte der neuen Aufgabe von dem Modul erfasst wurden kann es, über eine Schnittstelle im System, diese zur bestehenden Aufgabensammlung hinzufügen.

Hiermit versichere ich, dass die vorliegende Arbeit von mir selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt wurde. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Diese Arbeit hat weder in dieser oder ähnlich Form noch auszugsweise einer anderen Prüfungsbehörde vorgelegen.

Passau, den 19.11.2012 /