

TGM

E-Bibliothek

v1.2

Nenad Gligorevic

2015

WEXSTRASSE

Inhalt

1.) Aufgabenstellung	3
2.) Planung (Aufwandschätzung).....	4
Aufwandschätzung für Benutzerfunktionen	4
Planungsergebnis	5
3.) Datenbankdesign.....	5
3.1) XML-Schema.....	6
3.2) Propel	6
Ergebnis:	7
4.) Aufgabeneinteilung	8
Nenad Gligorevic	8
Mirza Kuljancic	8
Sefa Öztürk	8
Rohat Anil	8
Hüseyin Bozkurt.....	9
Antonio Papp	9
5.) Skizze Design	9
6.) GUI.....	9
6.1) v1	9
6.2) v2	10
7.) Klassen - Konzept	11
7.1) v1	11
7.2) v2	11
8.) Implementierung.....	11
8.1) Benutzer registrieren	12
8.2) Auf Benutzer Eingaben reagieren	13

<i>Version</i>	<i>Datum</i>	<i>Status</i>	<i>Kommentar</i>
1.0	28.04.15	Dokumentenerstellung	Aufbau des Dokumentes
1.1	30.04.15	Implementierung (angefangen) 3.2	3.2 Propel Vorgehensweise Implementierung
1.2	05.05.15	Klassen - Konzept	UML eingefügt Implementierung 8.1
1.3	16.06.15	Dokumentenfertig- stellung	

1.) Aufgabenstellung

Um die in den verschiedenen fachspezifischen Fächern erworbenen Kenntnisse zu demonstrieren und zu festigen soll ein gemeinsames Abschlussprojekt fächerübergreifend realisiert werden. Insbesondere soll dabei Augenmerk auf die Kompetenzen „User-Interface-Design“, „verteilte Systeme“, „Web Services“ sowie „Softwareentwicklung“ gelegt werden.

Es soll ein Anwendung für eine Online-Bibliothek und Lese-Seite für quelloffene Bücher erstellt werden. Der Zugriff auf die Inhalte dieser Bibliothek soll sowohl von Desktop-Systemen (Webseite) als auch von mobilen Geräten (Webseite oder App) möglich sein.

2.) Planung (Aufwandschätzung)

Aufwandschätzung für Benutzerfunktionen

TASK	Beschreibung	Geschätzte Zeit in h
Zielbestimmung	Das Lastenheft zerlegen und das benötigte Visualisieren --> Verständlichkeit	2
Klassen Konzept (UML)	Die Überlegung wieviel Klassen es gibt und welche Aufgaben diese haben	4
XML - Schema	XML-Schema erzeugen und Propel Befehle ausführen	1
Aufgabeneinteilung	Wer welche Funktionen übernimmt	3
GUI Erstellung	Die Erstellung der GUI ohne Funktionalität, Skizze (Blatt + Photoshop)	5
Registrierung	Die Funktionen werden implementiert	5
Login & Logout	Die Funktionen werden implementiert	5
Passwort zurücksetzen	Die Funktionen werden implementiert	4
OAuth	Die Funktionen werden implementiert	4
User löschen	Die Funktionen werden implementiert	4
Benutzerrechte verwalten	Die Funktionen werden implementiert	5

Geschätzte Zeit in h

42 h

Planungsergebnis

TASK	Geschätzte Zeit in h	Zeit in h	Kommentar
Zielbestimmung	2	5	Das Zielerfassen hat in der Gruppe länger gedauert als geplant
Klassen Konzept (UML)	4	2	Schneller als geplant
XML - Schema	1	1	Keine Probleme
Aufgabeneinteilung	3	4	Verständnisprobleme
GUI Erstellung	5	3	Keine Probleme Schneller da als Gruppe agiert
Registrierung	5	10	Ungeplante Funktionen sind dazugekommen
Login & Logout	5	10	--/--
Passwort zurücksetzen			
OAuth			
User löschen	4		
Benutzerrechte verwalten			

Benötigte Zeit in h

h

3.) Datenbankdesign

Welche Daten benötigen wir ?

ID

Benutzername

Passwort

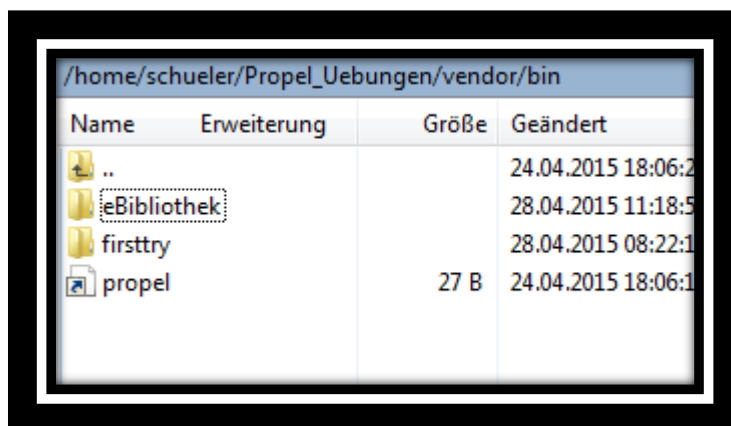
rolle

3.1) XML-Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<database name="ebibliothek" defaultIdMethod="native">
  <table name="user" phpName="Book">
    <column name="UserID" type="integer" required="true" primaryKey="true" autoIncrement="true"/>
    <column name="benutzername" type="varchar" size="255" required="true" />
    <column name="passwort" type="varchar" size="24" required="true"/>
    <column name="rolle" type="integer" required="true"/>
  </table>
</database>
```

3.2) Propel

Verzeichnis anlegen eBibliothek :



Name	Erweiterung	Größe	Geändert
..			24.04.2015 18:06:2
eBibliothek			28.04.2015 11:18:5
firsttry			28.04.2015 08:22:1
propel		27 B	24.04.2015 18:06:1

Befehle eingeben:

propel model:build

propel sql:build

Das in composer.json einfügen

```
"autoload": {
  "classmap": [ "generated-classes/" ]
}
```

dann folgenden Befehl eingeben:

php /composer.phar dump-autoload

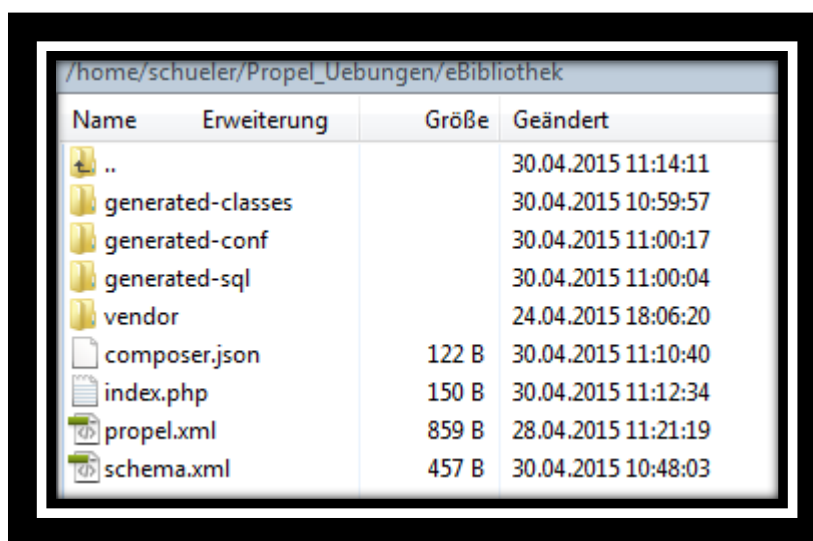
dann:

propel config:convert

Folgenden 2 Zeilen sind notwendig um auf die erzeugten Propel Verzeichnisse zugreifen zu können:

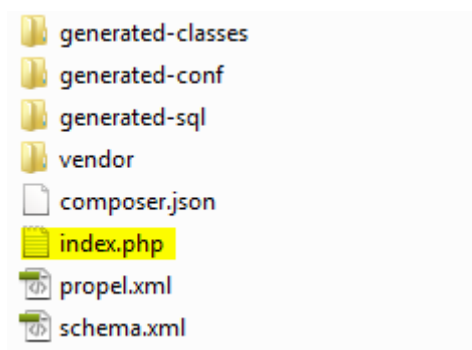
```
// setup the autoloading  
require_once '/path/to/vendor/autoload.php';  
  
// setup Propel  
require_once '/generated-conf/config.php';
```

Ergebnis:



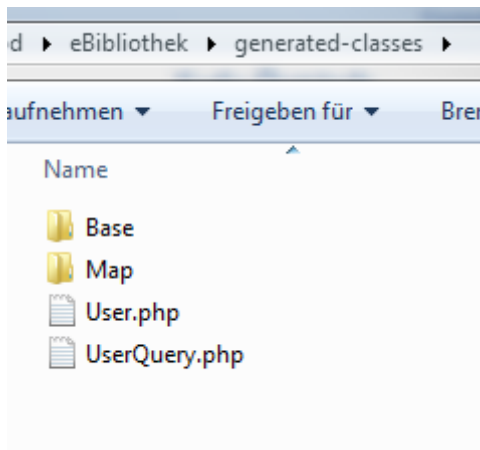
Name	Erweiterung	Größe	Geändert
..			30.04.2015 11:14:11
generated-classes			30.04.2015 10:59:57
generated-conf			30.04.2015 11:00:17
generated-sql			30.04.2015 11:00:04
vendor			24.04.2015 18:06:20
composer.json		122 B	30.04.2015 11:10:40
index.php		150 B	30.04.2015 11:12:34
propel.xml		859 B	28.04.2015 11:21:19
schema.xml		457 B	30.04.2015 10:48:03

Das File index.php ist unser Ausgangsfile aus diesem navigieren wir uns zu unseren Daten.



generated-classes			
generated-conf			
generated-sql			
vendor			
composer.json			
index.php			
propel.xml			
schema.xml			

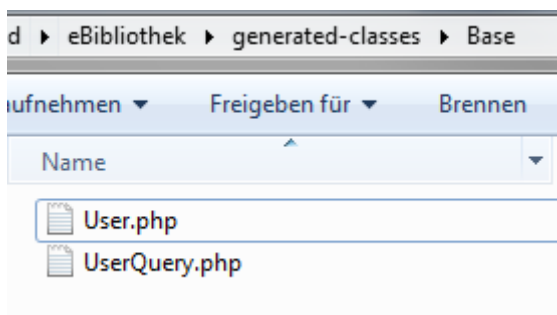
Wir greifen ausschließlich auf den Base Ornder Zu



In diesem befinden sich 2 Files:

- User.php
- UserQuery.php

Nur auf diese beiden Files soll und darf zugegriffen werden.



4.) Aufgabeneinteilung

Nenad Gligorevic

- Organisation
- Benutzer registrieren

Mirza Kuljancic

- Benutzer einloggen
- Benutzer ausloggen

Sefa Öztürk

- OAuth (Facebook..)

Rohat Anil

- User löschen

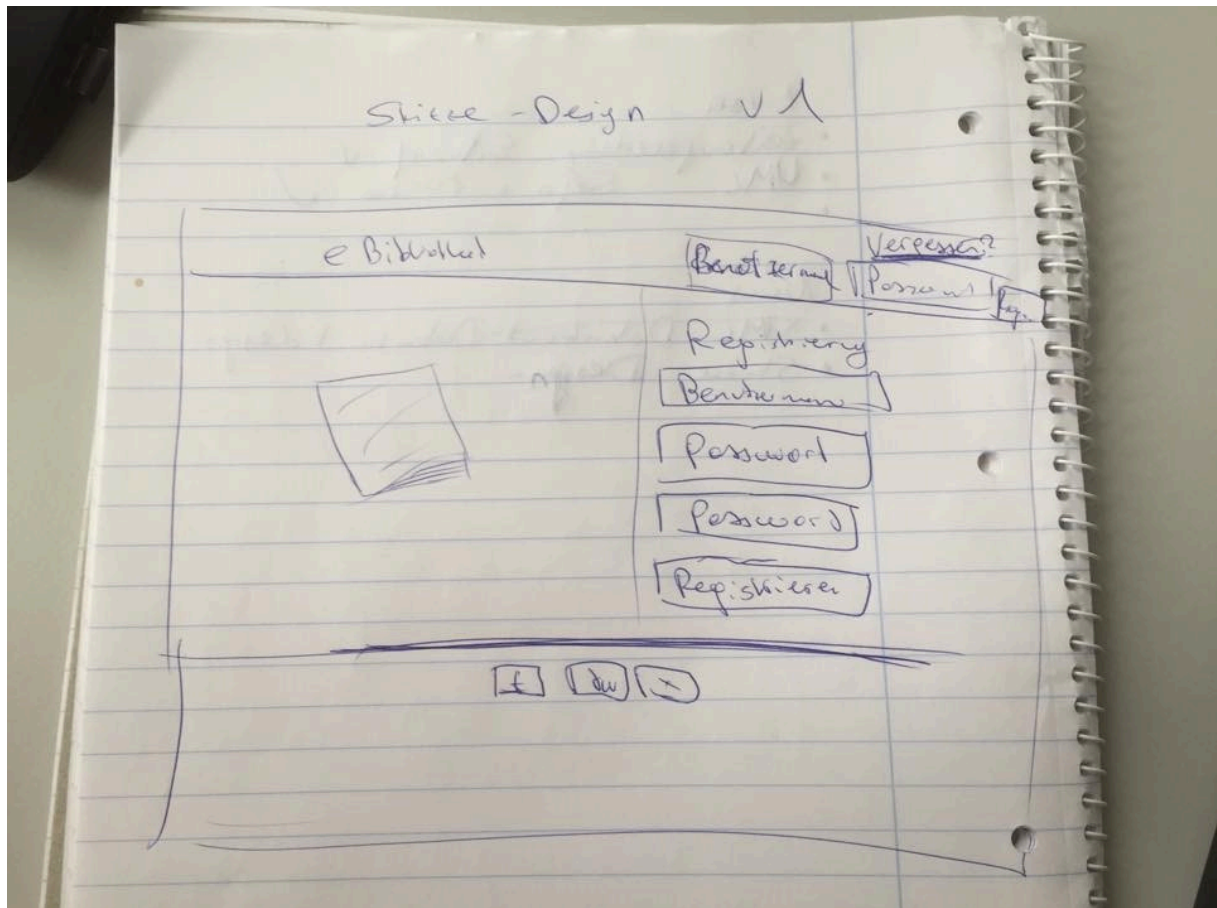
Hüseyin Bozkurt

- Passwort zurücksetzen

Antonio Papp

- Benutzerrechte verwalten

5.) Skizze Design



6.) GUI

6.1) v1

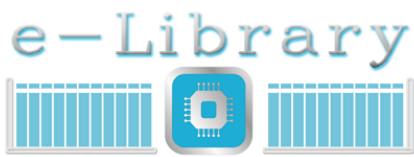
e-Library

Benutzername

Password



Sign in



Registrierung

Benutzername

Password

Password

Registrieren

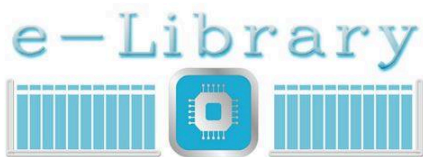
© TGM , Wexstrasse 19-23

6.2) v2

Benutzername:

Password

Anmelden



Registrierung

Benutzername

Password

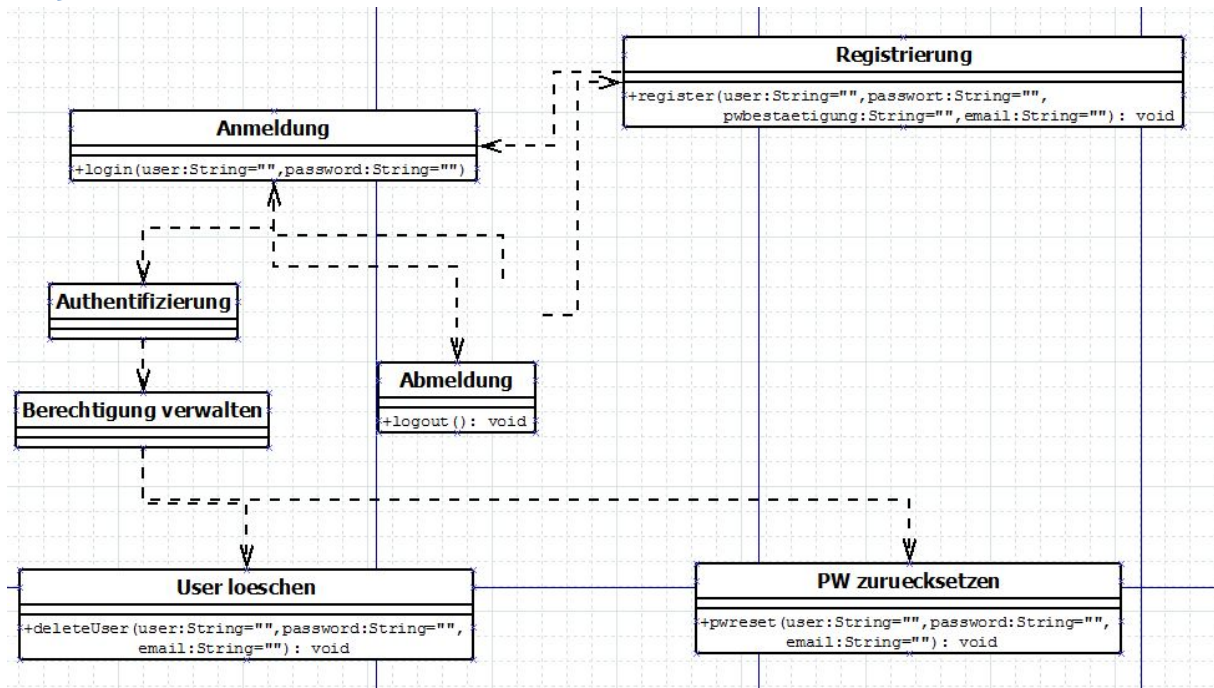
Password verifizieren

Registrieren

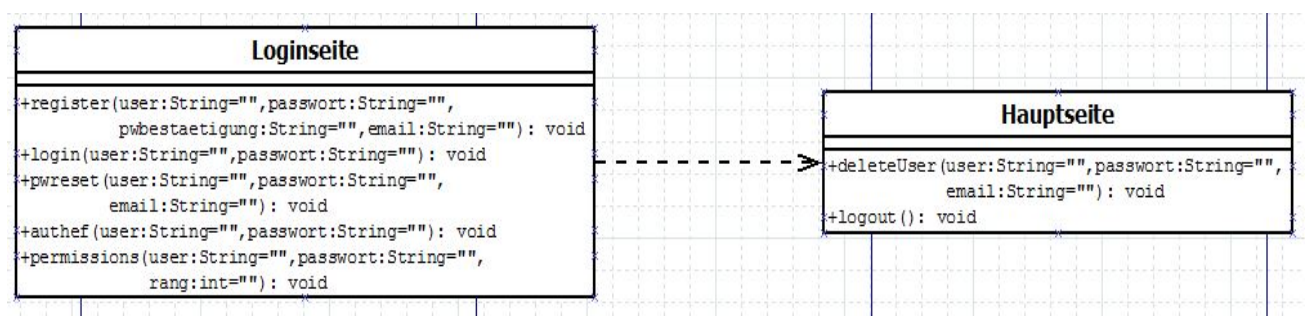


7.) Klassen - Konzept

7.1) v1



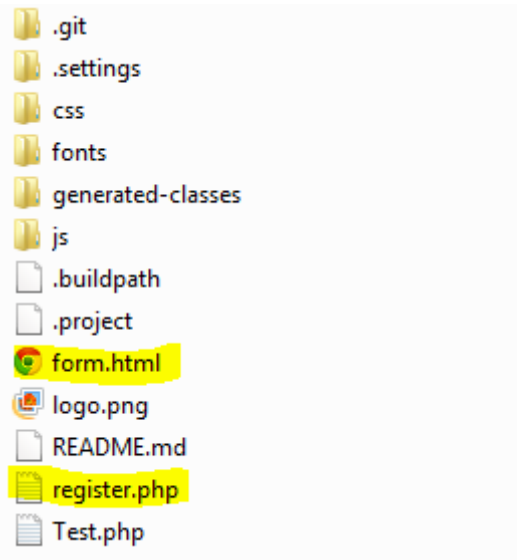
7.2) v2



8.) Implementierung

Es gibt ein form -File indem sich die GUI befindet

Und ein Register -File indem die Funktionalität steckt.



8.1) Benutzer registrieren

```
function Benutzer_registrieren($bname,$pw) {  
    $user = new User();  
  
    $user -> setBenutzername($bname);  
    $user -> setPassword($pw);  
    $user -> setRolle(0);  
  
    $user -> save();  
}
```

8.2) Auf Benutzer Eingaben reagieren

Benutzer Eingabe	Funktioniert	Funktioniert nicht
1.) Leer	X	
2.) Pw stimmen nicht überein	X	
3.) Richtig	X	
4.) Benutzer existiert bereits	X	
5.) Sonderzeichen meiden	X	
6.) Eingabenlänge	X	

1.) Leere Eingabe

e-Library

ngligorevic

.....

?

Sign in

e-Library

Registrierung

Benutzername

Passwort

Passwort wiederholen

Eingaben sind leer

Registrieren

© TGM , Wexstrasse 19-23

2.) Passwörter stimmen nicht überein

e-Library

ngligorevic

.....

?

Sign in

e-Library

Registrierung

Benutzername

Passwort

Passwort wiederholen

Passwörter stimmen nicht überein!










Registrieren

© TGM , Wexstrasse 19-23

3.) Richtige Eingabe

Resultat:

Ein neuer Benutzer

<div>← T →</div>				UserID	benutzername	passwort	rolle
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	1	testuser	testuser	0
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	2	testadmin	testadmin	2
<input type="checkbox"/>	 Bearbeiten	 Kopieren	 Löschen	7	jovan	123	0

Wird auf die nächste Seite gebracht dazwischen findet eine Animation statt damit man nicht in der DB nachschauen muss ob ein Benutzer angelegt wurde.



4.) Benutzername existiert bereits

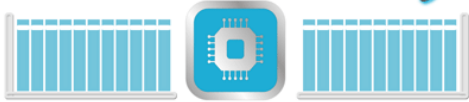
e-Library

nlgigorevic

.....

Sign in

e-Library



Registrierung

Benutzername

Passwort

Passwort wiederholen

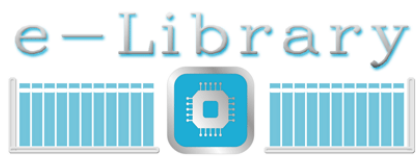
Dieser Benutzername existiert bereits

Registrieren

5.) Sonderzeichen meiden

e-Library

[Sign in](#)



Registrierung

Nur Buchstaben und Zahlen

© TGM , Wexstrasse 19-23

6.) Eingabelänge

e-Library

[Sign in](#)



Registrierung

Passwort mind. 6 Zeichen

Benutzername mind. 4 Zeichen

© TGM , Wexstrasse 19-23