



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

Web Applications Documentation

Hochschule für Technik und Wirtschaft Berlin
21.12.2017

Stephan Nordheim

Kristine Schwabauer

Intro	2
1. Requirements	3
2. Setting Up	5
3. How-To-Use	6
4. Development	8
Libraries	8
API	8
Patterns	8
Datenbank	8
Fazit	10
Abbildungsverzeichnis	11

Intro

Im Rahmen der Web Application Veranstaltung bei Prof. **Gefei Zhang** wurde uns die Aufgabe gestellt, eine Web Applikation, die den Prozess eines Peer Review Systems darstellen soll, zu entwickeln.

Im Kapitel 2 werden die Anforderungen an die Applikation beschrieben.

1. Requirements

- 2.1** Der User muss sich über eine Registrierungsseite anhand seiner E-Mail-Adresse und der Vergabe eines Passworts registrieren können.
- 2.2** Der User soll sich einloggen können.
- 2.3** Nachdem sich der User eingeloggt hat, sollte eine Liste angezeigt werden, die den Status der Reviews zeigt ("under review"/"accepted"/"rejected").
- 2.4** Der User darf die IDs der Reviewers nicht sehen.
- 2.5** Paper submission, review submission und das Zuweisen der Reviewers sollte access control geschützt werden.
- 2.6** Der Conference Chair soll in der Lage sein über eine eigene Seite registrierte Nutzer als Reviewers zur Conference hinzufügen können.
- 2.7** Jedes Paper soll mindestens drei Reviewers assigned werden.
- 2.8** Der User darf sein eigenes Paper nicht reviewen.
- 2.9** Jeder registrierte Nutzer soll die Möglichkeit haben auf einer bestimmten Seite n-Paper zu submitten.
- 2.10** Das Paper soll aus einem Titel und dem Abstract bestehen.
- 2.11** Der User sollte die Möglichkeit haben das Paper zu bewerten zwischen -2 (totally unacceptable) bis 2 (great paper).

2. Setting Up

Bevor die Flask Web Applikation genutzt werden kann, muss das richtige Set Up existieren. Hierzu kann das Projekt direkt von Github ¹ gecloned bzw. heruntergeladen werden.

Anschließend müssen folgende Schritte in der Konsole ausgeführt werden, welche nochmal in der ReadMe.md des Projektes nachgelesen werden können:

```
$ virtualenv venv  
$ source venv/bin/activate  
$ pip install -r requirements.txt
```

Abbildung 1: Befehle zur Erstellung der richtigen Umgebung

```
1. python manage.py db init  
2. python manage.py db migrate  
3. python db_create.py
```

Abbildung 2: Befehle zur Erstellung und zum Füllen der Datenbank

Mit dem dritten Befehl der Abbildung 1 werden die nötigen Packages in die Umgebung geladen. Der dritte Befehl der Abbildung 2 füllt die Datenbank mit zwei Usern, dem Admin und einem normalen User.

¹ https://github.com/krstnschwpwr/peer_review_system

3. How-To-Use

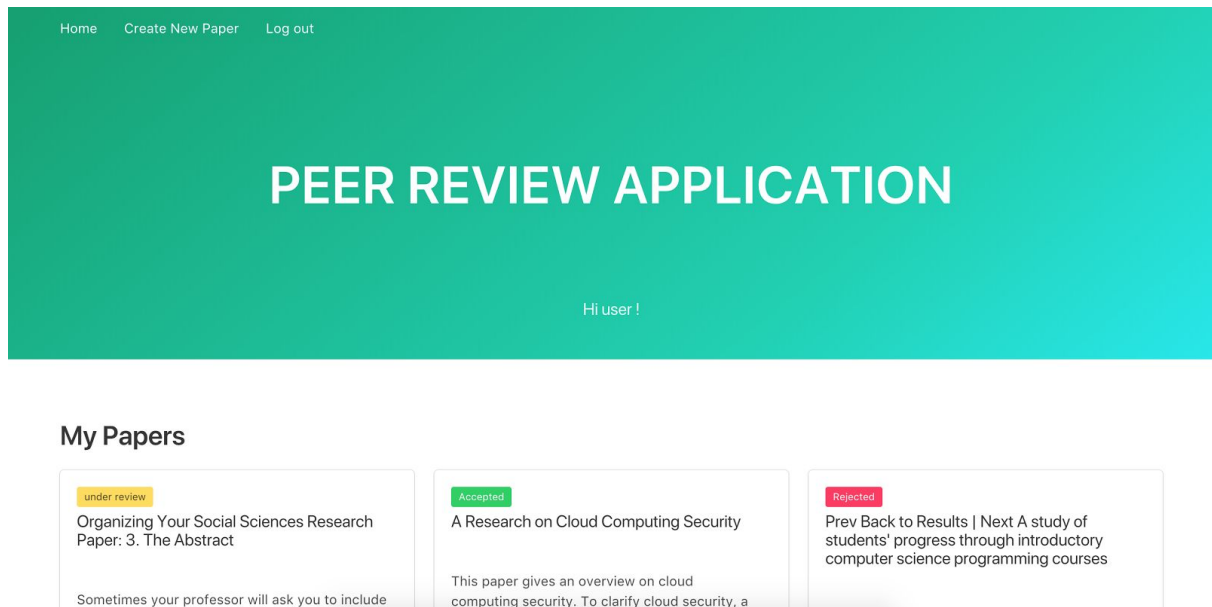


Abbildung 3: Darstellung der User-Seite

Die Applikation ist für zwei User-Gruppen gedacht. Der **Admin** übernimmt die Rolle des Conference Chairs und die **User** die der Paper-Autoren. Im Folgenden werden die Funktionen der einzelnen User Gruppen genauer erläutert.

User:

User kommt auf die Startseite ("/") und kann sich entweder mit seinem Namen, Email-Adresse und Passwort registrieren oder direkt mit seinen registrierten Daten (Username und Passwort) anmelden bzw. einloggen.

Nachdem sich der User erfolgreich eingeloggt hat, landet er auf seiner eigenen Seite ("/mypage"). Schnell hat er die Möglichkeit zwischen verschiedenen Seiten bzw. Funktionalitäten im Menü links wechseln, wie zum Beispiel ein neues Paper zu erstellen, zum Startbildschirm zu gelangen sowie sich sicher auszuloggen.

Die Seite ist in zwei Sections gegliedert. Zum Einen sieht der User die selbst erstellten Papers und zum Anderen die Papers, die ihm vom Conference Chair zur Review zugeteilt wurden.

Die Hauptfunktionalität der Seite besteht jedoch daraus, die vom User erstellten Papers mit dem aktuellen Status ("Under Review"/"Accepted"/"Rejected") anzuzeigen. Somit hat er einen sehr guten Überblick über die von ihm selbst erstellten und verfassten Papers. Außerdem hat er die Möglichkeit jedes Paper in einer detailed View anzusehen oder es direkt zu entfernen.

Bei denen ihm zugeteilten Paper hat der User die Aufgabe, dieses zu bewerten und zwar nach einem Gradingssystem von -2 bis 2, wobei 2 die höchste Bewertung ist.

Sobald der User auf das Menü-Feld "Create New Paper" klickt, gelangt er auf die Seite ("/paper/new"), auf der die Möglichkeit hat ein Paper zu erstellen. Dieses besteht sowohl auf einem Textfeld für den Titel, als auch einer Textarea für den Abstract.

Admin:

Als Conference Chair oder Admin gelangt dieser nach dem Einloggen direkt auf die ("/admin") Seite, anders als beim User ("/mypage").

Genau wie beim User wird ihm eine Navigation links oben angezeigt, bestehend aus folgenden Menü-Punkten: Home, Submission und Logout.

Die Hauptfunktionalität der ("/admin")-Seite besteht daraus, einen Überblick über alle erstellten Papers zu haben, sowie deren Status zu sehen. Außerdem hat der Admin die Möglichkeit hier für jedes angezeigte Paper die Reviewer, die das Paper bewerten sollen, auszuwählen. Nachdem diese Funktion ausgeführt wurde, gelangt das Paper sowohl auf die Seite der jeweiligen Reviewer als auch auf die ("/submissions")-Seite und somit aus der Liste des Admins.

Die Submissions-Seite ("/submission") beschreibt die freigegebenen Papers. Diese können von da an bewertet werden. Der Admin wird auf dieser Seite darüber informiert. Nachdem mindestens 3 User an der Review teilgenommen haben, kann der Conference Chair die Bewertungen vergleichen und letztendlich entscheiden, ob das Paper akzeptiert oder abgelehnt wird.

4. Development

Im Folgenden werden benutzte Patterns und Methoden sowie Schwierigkeiten beschrieben und genauer erläutert.

Libraries

Folgende Libraries wurden für die Entwicklung dieser Anwendung verwendet:

- baseguide (CSS-Framework)
- bulma (CSS-Framework)
- jQuery
- Flask-Marshmallow
- Flask-Login
- und weitere Flask Libraries, die in der requirement.txt Datei nachgelesen werden können.

API

Die einzelnen Api-Endpunkte können der views.py entnommen werden.

Patterns

Die Flask Applikation verhält sich nach dem Model-View-Controller Pattern (models.py/views.py/template), mit dem Ziel den Entwurf der Applikation so flexibel wie möglich zu gestalten und die Arbeit bei Änderungen oder Erweiterungen zu erleichtern.

Datenbank

Die Datenbank der Peer Review System Applikation besteht insgesamt aus drei Tabellen und hat folgende Felder und Relationen:

- **Papers:**
 - id
 - title
 - abstract
 - status
 - author_id
 - rating
- **Reviewers:**

- id
- paper_id
- reviewer_id
- rating
- **User:**
 - id
 - name
 - email
 - password

1:User -> n:Papers

1:Paper -> n: Reviewers

Fazit

Die uns gestellte Aufgabe mit dem im ersten Kapitel aufgelisteten Anforderungen, ein Peer Review System mit Python und Flask zu entwickeln, konnte erfüllt werden sowie alle Anforderungen umgesetzt werden.

Die Entwicklung der App war für uns sehr interessant und hat viel Spaß gemacht. Wir haben dabei viel dazulernen sowie Erfahrung sammeln können.

Auf die weitere Aufgabe sind wir schon sehr gespannt.

Abbildungsverzeichnis

Abbildung 1: Befehle zur Erstellung der richtigen Umgebung

Abbildung 2: Befehle zur Erstellung und zum Füllen der Datenbank

Abbildung 3: Darstellung der User-Seite