Projekt Modul 133  
Knowledgebase-System

«Scriky»



**Allgemeine Informationen**

Bearbeiter: Simon Gadola  
Dozent: Stéphanie Ramseyer  
Abgabedatum: 8. November 2019

Inhalt

[1 Einleitung 3](#_Toc24145666)

[1.1 Projektrahmen 3](#_Toc24145667)

[1.2 Projektumgebung 3](#_Toc24145668)

[1.3 Kontaktdaten 3](#_Toc24145669)

[2 Initialisierung 4](#_Toc24145670)

[2.1 Projektidee 4](#_Toc24145671)

[2.2 Projektnamen 4](#_Toc24145672)

[3 Konzept 5](#_Toc24145673)

[3.1 Detailanforderungen 5](#_Toc24145674)

[3.2 Use-Case Diagramm 5](#_Toc24145675)

[3.3 Mockups 5](#_Toc24145676)

[4 Realisierung 6](#_Toc24145677)

[4.1 Projektstruktur 6](#_Toc24145678)

[4.2 Persönliche Erfahrungen & Bemerkungen 6](#_Toc24145679)

[5 Anhang 7](#_Toc24145680)

[5.1 Referenzen 7](#_Toc24145681)

# Einleitung

## Projektrahmen

Im Rahmen des Moduls 133, welches an der Computerschule Bern (CsBe) von Frau Stéphanie Ramseyer unterrichtet wird, haben wir den Auftrag erhalten mit Hilfe eines Frameworks der Wahl ein kleines Projekt als Webapplikation zu konzipieren und zu realisieren.

Im Weiteren ein paar Metadaten zum Projekt:

## Projektumgebung

Framework: Django und das MVC-Prinzip  
Programmiersprachen: *Python* - plus *HTML*, *CSS* und *JS*  
Entwicklungsumgebung: *Jetbrains PyCharm*

## Kontaktdaten

Ich bin anhand der folgenden Kontaktdaten erreichbar, vorzugsweise per Mail:

Adresse: Simon Gadola  
Statthalterstrasse 70  
3018 Bern

Telefon: 079 726 82 26

Mail: simon.gadola@csbe.ch

# Initialisierung

## Projektidee

Ein Knowledgebase-System bzw. Website auf welcher Benutzer sich ein- und ausloggen können, und wenn angemeldet, in der Lage sind Wiki-mässig Notizen zu Einträgen zusammenzutragen und niederzuschreiben. Je nach dem soll evtl. sogar eine Funktions ??? eingebaut werden.  
Formulare werden zum Schutz vor dem CSRF-Angriff (**C**ross **S**ite **S**cript **F**orgery) ein sog CSRF-Token

## Projektnamen

Es wurden folgende Namen als möglicher Projektname evaluiert von denen sich am Schluss **Scriky** durchgesetzt hat:

* Handy KnoNo
* Scribb(l)Wiki
* Scriki
* Scriky
* Scribky
* Scriwki
* Scriwky

# Konzept

## Detailanforderungen

Folgende Features wurden als Must-have-Anforderung und als Nice-to-have-Anforderung resp. definiert:

Must-have

* Login
* Artikelseiten erstellen
* Artikelseiten bearbeiten
* CSRF-Token zum Schutz von Formularen

Nice-to-have

* Artikelseiten formatieren (Schriftart, -grösse, asdasd (Fett, kursiv, unterstrichen))
* Artikelseiten verlinken (intern sowie extern)
* Medien wie Bild-, Video-, und Audio-Dateien einbinden

## Use-Case Diagramm

Folgendes Use Case Diagramm wurde für das Projekt erstellt:

## Mockups

Folgende Mockups wurden für das Projekt erstellt:

# Realisierung

## Projektstruktur

Das Projekt ist so aufgebaut, dass man sich über die Links im Header erst anmelden bzw. registrieren muss, und danach ist man in der Lage über das Seitenmenü die Artikel entweder zu erstellen, zu bearbeiten, zu löschen.

## Persönliche Erfahrungen & Bemerkungen

Mit der Arbeit mit PyCharm und Django Ich habe festgestellt, dass man die Webapplikation auch von aussen testen lassen kann. Dazu sind lediglich 2 Einstellungen in der Einstellungs-Datei settings.py im Ordner der Hauptapp des Projekts (in meinem Fall heisst der Ordner scriky) vorzunehmen sind:

* Die IP-Adresse mit welcher der Server gestartet wird muss in der Startkonfiguration in der Eigenschaft *host* auf den Wert «0.0.0.0» gesetzt werden. So Listened («hört») der Server auf alle IP-Adressen, jedoch limitiert auf die Einträge der Variable im nächsten Punkt.
* Die Liste der Variable ALLOWED\_HOSTS muss um die gewünschten IP-Adressen erweitert werden, nur so lässt der Server auch die fremden IPs zu.  
  Mit dem Wert ‘\*’ als einziger Eintrag der Liste

## Schlussreflexion

Trotz einigen Mühen und Schwierigkeiten insbesondere mit dem CSS-Styling-Framework «Bootstrap» und mit Django, konnte ich dennoch bereits vorhandenes Wissen einsetzten und auch erweitern.

Leider hat es mir nicht gereicht das Projekt komplett fertigzustellen, jedoch plane ich es im privaten Rahmen noch weiter zu ergänzen und komplettieren.

# Anhang

## Referenzen

Folgende Referenzen wurden zum Nachschlagen von Informationen konsultiert:

* <https://simpleisbetterthancomplex.com/tutorial/2017/02/18/how-to-create-user-sign-up-view.html>
* <https://wiki.selfhtml.org/wiki/CSS/Eigenschaften/Flexbox>
* <https://getbootstrap.com/docs/4.3>
* <https://docs.djangoproject.com>