



TECHNISCHE HOCHSCHULE NÜRNBERG  
GEORG SIMON OHM

---

**Entwicklung einer internen Social Media  
Plattform mit personalisierbarem Dashboard für  
Studierende**

---

**Esther Kleinhenz**

Matrikelnummer: 2649270

**Bachelorarbeit**

zur Erlangung des akademischen Grades

**Bachelor of Engineering**

Media Engineering

**Erstprüfer:** Prof. Dr. Oliver Hofmann  
**Zweitprüfer:** Prof. Dr. Matthias Hopf

Nürnberg, 28. September 2018

---

Hiermit erkläre ich, dass die vorliegende Arbeit von mir selbständig verfasst und nicht anderweitig für Prüfungszwecke vorgelegt wurde, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche oder sinngemäSse Zitate als solche gekennzeichnet wurden.

Nürnberg, 28. September 2018

Katja Cornelia Hader  
E-mail: [haderka56442@th-nuernberg.de](mailto:haderka56442@th-nuernberg.de)

Studiengang Wirtschaftsinformatik  
Georg Simon Ohm  
Technische Hochschule Nürnberg  
KeSSlerplatz 12  
90489 Nürnberg  
Deutschland

---

## Abstract

---

Content of this Bachelor thesis is ....

**Keywords:** bla,bla Georg Simon Ohm, Wirtschaftsinformatik



---

# Inhaltsverzeichnis

---

<b>Abstract</b>	<b>i</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Ausgangssituation . . . . .	1
1.2 Ziel der Arbeit . . . . .	1
<b>2 Framework</b>	<b>3</b>
2.1 Django . . . . .	3
2.2 Erweiterung . . . . .	3
2.2.1 Taggable-Manager . . . . .	3
2.3 Bootstrap . . . . .	3
<b>3 Prototyp</b>	<b>5</b>
3.1 Organisation . . . . .	5
3.1.1 Verwaltung im Administrator-Backend . . . . .	5
3.1.2 Berechtigung der User . . . . .	5
3.2 Funktion . . . . .	5
3.2.1 Abonnieren . . . . .	5
3.2.2 Filtern . . . . .	5
3.2.3 Benachrichtigung . . . . .	5
<b>4 Ergebnis</b>	<b>7</b>
4.0.1 Evaluierung . . . . .	7
<b>5 Zusammenfassung und Ausblick</b>	<b>9</b>
<b>Referenzen</b>	<b>11</b>



## Einleitung

---

Die vorliegende Arbeit beschäftigt sich mit der wachsenden E-Mail-Flut in den Postfächern der Studierenden und wie man diese reduzieren kann. Schon seit geraumer Zeit ist bekannt, dass das Versenden von Informationen über elektronische Post nicht nur Vorteile mit sich bringt. Wie der Spezialist für Gesundheitsprozessberatung in einem Bericht der Mitteldeutschen Zeitung erwähnt, macht es die stets wachsende E-Mail-Menge unmöglich sich vernünftig mit den Informationen zu befassen“([Ver13]). Nicht nur am Arbeitsplatz sondern auch in Hochschulen wird Gebrauch gemacht, weitere Empfänger oder sogar ganze Verteiler mit in die Kopie einer E-Mail zu integrieren. Um die Prioritäten der Informationen besser bilden zu können, sollen Studierende selbst entscheiden, welche Nachrichten relevant sind. Hierfür wird eine Social Media Plattform mit personalisierbarem Dashboard implementiert.

### 1.1 Ausgangssituation

Alle Informationen der Fakultät Elektrotechnik Feinwerktechnik Informationstechnik, kurz efi, werden über die globalen Verteiler des Hochschulinternen Postfaches versendet. Viele dieser Daten sind jedoch nur für eine geringe Schnittmenge der Empfänger relevant und lassen sich nur schwer priorisieren. Das ständig überlastete Postfach muss somit regelmäßig gepflegt werden. Einen massiven Administrativen Aufwand bedeutet es, E-Mails zu filtern und nach persönlichem Ermessen zu verwalten. Zudem leidet die Nachhaltigkeit der Informationen. Mochten die Empfänger ältere E-Mails abrufen, mussten diese meist schon entfernt werden um Platz für den neuen, eintreffenden E-Mail-Verkehr zu schaffen. Diese Situation führt dazu, dass Empfänger die Informationen meist nicht lesen und sofort entfernen. Die Ersteller dieser haben keinerlei Möglichkeiten zu überprüfen ob und wie viele Studierende und Dozenten eingehende Nachrichten öffnen und lesen.

### 1.2 Ziel der Arbeit

Ziel der Arbeit ist es, durch die Einbindung einer Social Media Plattform den Speicheraufwand des Hochschulpostfaches für Studierende der Efi-Fakultät zu reduzieren. Die Flut an

## 1. Einleitung

---

E-Mails soll durch das Verwenden eines personalisierte Dashboard gedrosselt werden. Hierbei wird zunächst der Fokus auf die grundlegenden Funktionen der Website wert gelegt. Zudem sollen die Verfasser der Informationen benachrichtigt werden, in welchem Umfang die hochgeladenen Informationen bereits abonniert und gelesen wurden.

### Framework

---

Um die Website-Erweiterung realisieren zu können, wird zunächst festgelegt welche Programmierschnittstellen verwendet werden. Im Web-Backend fällt die Wahl auf die objekt-orientierte Sprache Python, das ausschließlich Serverseitig anwendbar ist. Der Programmaufbau Pythons macht den Code leicht lesbar und der einfache Syntax ermöglicht eine strukturierte Implementierung der Website([Ndu17]). Ein entscheidende Vorteil hierbei ist das dazugehörige Web-Framework Django, auf das im folgenden Kapitel genauer eingegangen wird.

### 2.1 Django

Django ist eine Erweiterung, die auf einer Model-View-Presenter (MVP) Architektur basiert. Ähnlich wie der Model-View-Controller sind die Interaktionen zwischen Model und View die Auswahl und Ausführung von Befehlen und das Auslösen von Ereignissen. Da die View aber hier bereits den Großteil des Controllers übernimmt, ist der MVP eine Überarbeitung. Der Teil, der Elemente des Modells auswählt, Operationen durchführt und alle Ereignisse kapselt, ergibt die Presenter-Klasse([She09]). auf die Daten vom Model zurück und das Model teilt der View mit, wenn sich Daten ändern. Da nun aber nahezu alle Aufgaben des Controllers von der View übernommen werden,

### 2.2 Erweiterung

#### 2.2.1 Taggable-Manager

### 2.3 Bootstrap



### Prototyp

---

#### **3.1 Organisation**

3.1.1 Verwaltung im Administrator-Backend

3.1.2 Berechtigung der User

#### **3.2 Funktion**

3.2.1 Abonnieren

3.2.2 Filtern

3.2.3 Benachrichtigung



---

**Ergebnis**

---

4.0.1 Evaluierung



---

**Zusammenfassung und Ausblick**

---

Zusammenfassung...



---

## Referenzen

---

- [Lei13] Ingo Leipner. Stress für beschäftigte: Wie kann man die e-mail-flut bekämpfen. 2013. <http://www.mz-web.de/wirtschaft/e-mail-flut-mails-bearbeiten-kommunikation-stress-zeit-sparen>.
- [Ndu17] Nnenna Ndukwe. Python is the back-end programming language of the future and here's why. 2017. <https://medium.com/@nnennahacks/https-medium-com-nnennandukwe-python-is-the-back-end-programming-language-of-the-future-heres-why>.
- [She09] Alexy Shelest. Model view controller, model view presenter, and model view viewmodel design patterns. 2009. <https://www.codeproject.com/Articles/42830/Model-View-Controller-Model-View-Presenter-and-Mod>.