

hallo und herzlich willkommen zur Präsentation meiner Bachelorarbeit mit dem thema:
Entwicklung einer internen social media Plattform mit personalisierbarem
Dashboard für Studierende

1. Motivation

- E-Mail-System der Hochschule wird genauer betrachtet
- Zu viele irrelevante Informationen überfluten das E-mail-Postfach
- es fällt besonder auf, dass
 - Durch das Verwenden von globalen Verteilern werden Unmengen an Informationen an einzeln Personen weitergeleitet

Problemstellung

Folgende Probleme treten auf:

- die Menge an E-Mails ist schwer priorisierbar und administrierbar
- die Nachhaltigkeit von Informationen ist nicht garantiert
 - ist das Postfach überlastet - schnelles löschen
 - Suche nach alten Mails nicht zuverlässig/möglich

Forschungsfrage

Kann die E-Mail-Flut der Technischen Hochschule mit Hilfe einer Social Media Plattform gedrosselt und die Nachhaltigkeit der Informationen gewährleistet werden?

- um Frage beantworten zu können, ..

Es wird ein Prototyp implementiert, der mithilfe von Tags das abonnieren von Informationen für Studierende ermöglicht

Um Prototypen zu entwickeln wurde verschiedene Frameworks verwendet

2. Verwendetes Framework

als Tagcloud visualisiert, zeigt eine Übersicht über die verwendeten Frameworks
—> Wichtigkeit dieser wird durch Größe angedeutet

- Django (kurze Einführung)
- Taggabel Manager
- Hilfsbibliotheken, die im Prototyp eingebunden wurden
 - django-taggit-templatetags
 - django-hitcount
 - django-post-office

Django

Django basiert auf Python, einer ...

- | | |
|----------------------|---|
| Python: | Django: |
| • modularer Aufbau | • Umfangreiche Extras |
| • dynamische Sprache | - Administrator-Oberfläche (später mehr dazu) |
| • Objekt-Orientiert | - gestaffelte Zugangsberechtigungen (staff_status, superuser_status) |
| • einfach lesbar | • Schutz vor Sicherheitslücken |
| | • SQL-Injection, Cross-Site-Scripting und sicheres Authentifizierungssystem |

Taggable Manager

einer der wichtigsten Erweiterungen für den Prototyp ist django-taggit

- klassische Bibliothek für Django
- mit pip (Django Packagemanager installieren und einbinden)

– Was kann django-taggit eig?

wie die documentation sagt „ ... designed to make adding tagging to your project easy and fun“

-> beschreibt es schon ziemlich gut

- vereinfacht das Hinzufügen von Tags in Modellen

Vorteile:

- API stellt verschiedene Methode zur Verfügung (add, remove, clear)
- „slug“:
 - ist der eindeutige Identifikator eines Tags (Vergleichbar mit einem Primary Key)
 - automatisch generiert

Hilfsbibliotheken

- django-taggit-templatetags
 - im Frontend integrieren
 - auf verschiedene Modelle anwendbar
 - zB. Listen, Tagcloud
- django-hitcount
 - einfacher Zähler
- django-post-office
 - für wöchentliche Updates des Prototypen
 - ermöglicht Erstellen von E-Mail Templates
 - Logt versenden von Mails (Kontrolle)

3. Prototyp

zunächst **Datenmodellierung** um den Prototyp besser verstehen zu können (vereinfacht)

- Django Authentifizierungssystem mit den gestaffelten Zugangsberechtigungen
- wird erweitert mit Custom User um tags für den einzelnen realisieren zu können
- Post-Model mit einzelnen Feldern das über author mit User-Model verbunden wird

• **Video vorbereitet**

- zeigt Abonnier-Funktion aus Sicht eines Studenten
- begleitend dazu erklären

Video: (Pausieren cmd+K)

Login: Dashboard des Studenten wird angezeigt (nur ein Tag abonniert im Moment)

Übersicht aller Artikel:

- möchte man zum Beispiel alle Artikel die den Tag „test“ enthalten sehen
- einfach klicken -> gefilterte Übersicht

Detailansicht des Artikels:

Tags suchen und abonnieren:

- klicken auf Suche — zweigeteilte Ansicht
- Links Tag-Cloud
- Suche nach „hi“ -> werden alle Artikel angezeigt die diesen Tag enthalten
- Tag „hi“ soll abonniert werden
- auf sichern klicken
- Anzeige bestätigt Eingabe
- neuer Tag wird sofort hier angezeigt, optisches Feedback

Auf Dashboard werden aktualisiert die Artikel gezeigt

Student kann sich dann Abmelden

• Funktionen (Ansicht für Benutzer mit staff-Berechtigung)

- zwei Menü-Punkte sind identisch zur Studenten-Ansicht (Dashboard und Suche)
- Django Admin-Backend:
 - Django eigene Benutzeroberfläche
 - Metadaten aus Code werden visuell dargestellt
 - User-Verwaltung, Tag-Verwaltung und Post-Verwaltung
- Editor um neue Artikel anzulegen: (siehe Bild)
 - Titel, Text, Tags und Datum der Veröffentlichung (falls nicht angegeben) —>
- Entwurfsliste für nicht veröffentlichte Artikel (siehe Bilder)
 - Detailansicht -> Publish, Bearbeiten, Löschen

4. Evaluation

- **Einleitend zur Evaluation möchte ich die Forschungsfrage wiederholen:**
 - Kann die E-Mail-Flut der Technischen Hochschule mit Hilfe einer Social Media Plattform gedrosselt und die Nachhaltigkeit der Informationen gewährleistet werden?
 - habe mich dafür entschieden anhand eines Probanden der Hochschule die Problematik zu verdeutlichen
 - durch eine beispielhafte Berechnung zeigen, dass sich durch den Prototyp die E-Mail-Flut reduzieren lässt
 - **Beispielhafte Berechnung unter folgenden Voraussetzungen:**
 - Sommersemesters 2018
 - 8. Hochschulsemester
 - Bachelor Media Engineering
 - Summe der E-Mails ca. 265
 - darunter Verteiler wie EFI-intern, Studierende, Fachschaft
 - Summe aller relevanten E-Mails des Probanden
 - Relevante Infos:
 - Prüfungsrelevante Mails
 - Sekretariat
 - Veranstaltungen
 - International Office usw.
 - > nur die Hälfte ist relevant!
- Achtung: Relevanz subjektiv!

- **Ergebnis:**
 - **geht man davon aus, dass**
 - wöchentlichen Updates per Mail User über Neuigkeiten im Pototpyne informieren -> 28
 - Vergleicht man Updates mit aktuellem E-mail-Verkehr wird deutlich
 - Prototyp unter idealen Voraussetzungen die E-Mail-Flut stark eindämmen kann !!

5. Schlussbetrachtung

Betrachtet man das Resultat dieser Arbeit kann man folgendes Schlussfolgern:

- E-Mail-Flut kann durch die Verwendung des Prototypen reduziert werden
- Prototyp zeigt, das Nachrichten einfach gepflegt und abonniert werden können
- Weiter Funktionen müssen noch ergänzt werden
 - Eigenschaften von Tags erweitern (Erstellungsdatum, Gruppierungen usw.)
 - Usability optimieren (Hinzufügen und Löschen von Tags als grafische Schaltfläche realisieren)