

Pflichtenheft

Zur Vorlesung Systemadministration

im Bachelor Studiengang Angewandte Informatik

Sommersemester 2017

bei Herr Prof. Dr. Eggendorfer

Umsetzen eines Webmonitoring-Dienstes

März 2017

Inhaltsverzeichnis

1. Einleitung	3
1.1 Motivation	3
1.2 Ziele	3
1.3 Marktanalyse	3
2. Anforderungen	4
2.1 Funktionale Anforderungen (must-have)	4
2.2 Nichtfunktionale Anforderungen (nice-to-have)	4
3. Gantt-Plan	5
3.1 Vorgänge	5
Dokumentation	5
Projekt	5
3.2 Diagramm	5

1. Einleitung

1.1 Motivation

Jeden Tag durchstöbern wir das WorldWideWeb auf der Suche nach Neuigkeiten. Wir schauen uns dabei vielleicht Nachrichtenseiten, die neuesten Updates in GitHub-Projekten, öffentliche Bekanntgaben, Foren oder Preise von Produkten an.

Ist uns die aktuelle Veränderung von Informationen einer Website besonders wichtig, so ist es nicht immer möglich, darüber automatisch informiert zu werden.

Wartet man bspw. mehrere Tage auf die Freigabe zu einem heiß begehrten Verkauf oder dessen Preisänderungen, so ist man mehrmals am Tag gezwungen dies zu Überprüfen.

Diese Lücke soll ein automatisierter Prozess füllen, welcher die kontinuierliche Überprüfung von Webseiten übernehmen kann.

1.2 Ziele

Als Ziel dieser Arbeit soll ein Dienst erstellt werden, welcher Benutzern die Möglichkeit geben soll, Webseiten oder Teilbereiche dieser auf Veränderungen zu überwachen.

Der Benutzer soll über eine Weboberfläche die Möglichkeit bekommen die jeweilige Webseite anzugeben und (falls gewünscht) den entsprechenden Teilbereich auszuwählen.

Findet eine Veränderung statt, so wird der Nutzer des Dienstes über einen ausgewählten Kommunikationsweg (bspw. Email) in Kenntnis gesetzt.

1.3 Marktanalyse

Systeme wie das von visualping.io, distill.io, followthatpage.com oder auch uptrends.de greifen genau diesen Gedanken auf, automatisiert Webseiten zu überwachen. Mit unterschiedlicher Grundmotivation nutzen die vorhandenen Produkte unterschiedliche technologische Ansätze für den Vergleichsprozess.

Der größte Nachteil dieser etablierten Anbieter ist, aus unserer Sicht vor allem, die etwas umständliche Bedienung (für nicht technisch Interessierte) und der Registrierungszwang.

Dies macht es für die meist vereinzelte Nutzung als Nischenprodukt unattraktiv und umständlich.

Durch ein einfaches Design und schlichte Handhabung möchten wir mit innovativer Leistung unser Produkt aus der Masse hervorstechen lassen.

2. Anforderungen

2.1 Funktionale Anforderungen (must-have)

- Der Dienst ist über das Internet erreichbar
 - Domain wird registriert
 - Weiterleitung muss eingerichtet werden
- Der Dienst besitzt eine Weboberfläche
 - Server (Apache/httpd/nginx) wird aufgesetzt und eingerichtet
 - Webapp für den Dienst wird implementiert
 - Implementierung in Python + Flask + JS
 - Datenbank Server wird eingerichtet
- Ein Hintergrundprozess prüft die gesamte Webseite auf Veränderungen
 - implementierung Python + Cron
- Prüfungen werden in unterschiedlichen Intervallen durchgeführt
 - Cron Jobs müssen automatisch erstellt werden
- HTTP und HTTPS wird unterstützt
- GET und POST Parameter können an Webseiten gesendet werden
- Nutzer des Dienstes können per Email informiert werden
 - Email Server wird aufgesetzt (bsp exim4)

2.2 Nichtfunktionale Anforderungen (nice-to-have)

- der Dienst ist skalierbar
 - Datenbank wird Cluster-skalierbar gemacht
- Ein Hintergrundprozess prüft Webseiten auf grafische Veränderungen, wahlweise nur in benutzerdefinierten Teilbereichen
- Prüfungen können wahlweise mit Adblocker durchgeführt werden
- Benutzer werden über alternative Kanäle informiert
 - Messenger
 - Push Notifications
- Benutzer können sich Registrieren
 - über Facebook/Google/VK/Twitter
- Registrierte Benutzer können mehrer Prüfvorgänge verwalten

3. Gantt-Plan

3.1 Vorgänge

Dokumentation

- 14.03. - 24.03. Projektskizze
- 01.04. - 07.04. Pflichtenheft
- 13.04. - 21.04. Finales Pflichtenheft
- 22.04. - 19.05. Betaversion der Ausarbeitung
- 27.05. - 09.06. Finale Ausarbeitung
- 10.05. - 16.06. Präsentation

Projekt

- 14.03. - 28.03. Recherche / Systemaufbau bestimmen
- 29.03. - 19.04. Webmonitoring Hintergrundprozess
- 20.04. - 03.05. Weboberfläche / Server einrichten
- 04.05. - 11.05. Benachrichtigung per Email
- 12.05. - 19.05. Testing

3.2 Diagramm

