

# Softwaretechnik

## WERBUNG ANALYSE

1. Anforderungsbestimmung
2. Lösungsvorschläge
3. Beschreibung der Lösung
4. Testen
5. Wie benutzt man den Software?
6. Schluss

### 1. Anforderungsbestimmung

#### Beschreibung des Problems

Eine Doktorantin in Psychologie will die Werbung im Internet analysieren und die Konsequenzen bei den Nutzern bewerten.

Dafür wird sie verschiedene Personen zu ihrem Büro einladen und sie fördern im Internet 30 Minuten lang zu surfen. Als wäre es ein ganz normaler Tag.

Nach dieser Stunde wird die Untersuchende manche Frage stellen, um etwas mehr über sie zu kennen.

Aufgabe unseres Systems ist die Daten der Werbung, die die Untersuchende im Internet finden, zu speichern. Diese Daten werden später von der Doktorantin bewertet und analysiert.

#### Genaue Daten, die gebraucht werden:

Man braucht die URL's, die die Untersuchende besucht haben und man braucht die Werbung-URL, die gezeigt wurden.

Unter URL muss man den ganzen link verstehen.

Beispiel:

<http://www.nytimes.com/2014/07/05/sports/at-wimbledon-white.html?hp&action=click&pgtype=Homepage&version=LargeMediaHeadlineSum&module=photo-spot-region>

(Nicht nur [www.nytimes.com](http://www.nytimes.com))

#### Technischer Bereich:

Die Daten müssen in einem leserlichen und gemütlichen Format herauskommen, der leicht zu einer Excel-Datei oder similar exportierbar ist.

#### Beschreibung der Zielgruppe:

Das System muss schon an den Computern installiert sein, und die Untersuchenden werden mit den Software nicht interagieren müssen. Die Untersuchende wissen, dass Daten über sie gesammelt werden, also ist eine langsamere Verbindung kein wichtiger Problem.

Die Doktorantin hat Windows und Linux-Computerkenntnisse (trotzdem hat sie Windows lieber, da sie Linux nur in der Universität benutzt hat). Als Psychologiestudentin hat sie auch mit statistischer Software und dabei auch ein bisschen mit Datenbanken in Excel gearbeitet.

## **2. Lösungsideen**

Da ein Kontroll über die Quelltexte und das Besuch der verschiedenen Seiten geführt werden soll, wird es klar, dass wir ein Addon benutzen sollen.

Addons sind Programme, die im Browser installiert werden, und die mit ihm zusammenarbeiten können.

Erste Lösungsidee sei ein Addon selber zu entwickeln: lese ein Quelltext, suche die Werbung und speichert diese Daten. Problem hier ist die große Komplexität des Projekts.

Zweite Lösungsidee ist das Addon „Adblockplus“ zu benutzen. Glücklicherweise ist es FreeSoftware und somit kann man den Quelltext lesen und sogar beliebig verändern.

Wenn man die beiden Ideen betrachtet, wird es klar, dass die zweite die optimalste ist.

## **3. Beschreibung der Lösung**

### **Beschreibung von Adblockplus**

Adblockplus ist ein Open Source Projekt. Man kann den Quelltext herunterladen und es beliebig ändern.

Nach dem Herunterladen des Quelltextes, muss man sich einlesen, um ein Überblick des Projekts zu haben und eine optimale Lösung zu finden, um die Werbung zu speichern. Dokumentation zum Quelltext ist nicht vorhanden.

Es ist in Javascript geschrieben. Die Funktionsweise ist die folgende:

Eine Seite wird geladen. Der Quelltext wird in einem Baum mit verschiedenen Knoten gespeichert. Diese Knoten werden rekursiv untersucht ob sie Werbung beinhalten. Wenn es der Fall ist, werden sie nicht geladen.

Um zu wissen ob Werbung vorhanden ist, werden Filtern benutzt. Diese können vom User gemacht werden, oder kann man die schon vorhandene nehmen.

Um den Quelltext zu verändern und nachher es starten zu können, muss man die „Add-on SDK“ von Firefox herunterladen und installieren. Dafür muss man die Dokumentation dafür lesen:

<https://developer.mozilla.org/en-US/Add-ons/SDK/Tutorials>

Um den Aufbau der Add-ons zu verstehen, ist es nützlich sich verschiedene Punkte dieser Tutoriale zu lesen.

### **Beschreibung der Lösung**

Es ist eine Möglichkeit unseren Problem zu lösen, den exakten Platz zu finden, wo es gesagt wird, dass die Werbung nicht geladen werden soll.

Um diesen Punkt zu finden, wird das Code durchgelesen und die mögliche Funktionen werden mit der Entwickler-Konsole von Firefox durchsucht (wann sie aufgerufen werden, was sind die Variablen, die als Parameter haben...etc). Dies ist zu machen, weil die Kommentare im Quelltext sehr inhaltslos sind.

Nach dem die Funktion gefunden ist, die die Werbung findet, muss man sich eine schlaue Form ausdenken, die Daten zu speichern.

Mögliche Formen, die eingefallen sind, sind:

- In einer Textdatei
- In einer Cookie
- In einer Datenbank

Textdateien würde die einfachste Form sein, und auch die einfachste für die User.

Nach einer Suche in Google wie dies in JavaScript gemacht wird, wird es klar, dass dies für Sicherheitsgründe unmöglich ist. Also muss man die nächste Möglichkeit versuchen.

Cookies sind nicht für Addons gedacht. Also kommt man direkt zum letzten Punkt: die Daten in einer Datenbank speichern.

Es ist nicht die optimalste Lösung, da die Benutzerin erst lernen muss, wie man die Datenbank benutzt. Trotzdem ist es nicht eine so große Last, im Vergleich zum Auszudenken, wie man die JavaScript-Verhinderung Dateien zu speichern, zu umgehen.

Für dies, wird XAMPP installiert. Es ist eine Zusammenstellung von freier Software: PHP, MySQL-Datenbanken und Apache-Server.

Es wird eine Datenbank zusammengestellt mit einer Tabelle, die die Werbung-Daten speichern wird.

Um die Tabelle zu füllen, wird ein PHP-Skript erstellt.

Dieser PHP Skript wird vom Add-On mit den richtigen Parametern aufgerufen.

Und somit ist die Verbindung zwischen den Add-On und die Datenbank schon erstellt.

### **Vor- und Nachteile der Lösung**

Nachteile der Implementation ist die Schwierigkeit der Installation (wie nachher erklärt wird).

Dies ist kein großes Problem, da es nur in ein oder zwei Computern installiert wird, also kann direkt von den Entwicklern installiert

werden.

Weiterer Problem, man braucht Linux und Firefox. Dies kann ein Problem für die Untersuchendepersonen, da sie nicht an diesem System gewohnt sind. Trotzdem, da man nur im Internet surfen muss, und Firefox als einer der verbreitesten Browser gilt, soll dies auch nicht als großer Nachteil gelten.

Ein signifikanter Vorteil ist, dass der Projekt billig ist. Da es auf Freier Software basiert ist, muss man kein Geld für weiteres Software ausgeben. Die Datenbanken sind umsonst, und der Quelltext worauf es basiert kann man direkt aus Internet herunterladen. Also muss man nur die Arbeit der Entwickler als Kosten betrachten.

Weiterer Vorteil ist, dass die Daten nicht missbraucht werden. Da es Freier Software ist, kann man betrachten, dass die Daten nicht woanders geschickt werden.

#### 4. Testen

Der neue Add-On ist leicht zu testen:

- Das Surfen im Internet muss ohne große Zeitverzögerung erfolgen
- Die Werbung muss gezeigt werden
- Die Daten der Werbung müssen in der Datenbank gesammelt werden

#### Testergebnisse

Werbung wird richtig angezeigt und die Daten werden richtig gespeichert.

Es folgen die Ergebnisse der Load-Zeiten:

Webseite	Zeit ohne Addon (in ms)	Zeit mit Addon (in ms)
nytimes.com	<ul style="list-style-type: none"><li>• 29749</li><li>• 26126</li></ul>	<ul style="list-style-type: none"><li>• 36718</li><li>• 31020</li></ul>
abc.go.com	<ul style="list-style-type: none"><li>• 43614</li><li>• 39469</li></ul>	<ul style="list-style-type: none"><li>• 48624</li><li>• 47373</li></ul>
rollingstones.com	<ul style="list-style-type: none"><li>• 22027</li><li>• 19150</li></ul>	<ul style="list-style-type: none"><li>• 20517</li><li>• 20819</li></ul>

\*Die Daten kommen aus dem Addon „app.telemetry Page Speed Monitor“

Man kann eine leichte Verzögerung vom 20% in den Seiten mit Werbung sehen. (rolligstones.com hat keine Werbung). Es ist nicht ein großes Hindernis, man kann damit leben.

Trotzdem ist die User-Experience nicht optimal, da sich viele Pop-Ups öffnen und dies macht, dass Firefox nicht schnell funktioniert.

Beim Sprechen mit der Doktorantin, erklärt sie es sei kann Problem: sie will, dass die Software billig ist und in einer Woche umsetzbar ist.

## **5. Wie benutzt man den Software?**

Damit die Software funktioniert muss XAMPP installiert sein.

Es muss eine Datenbank mit den Namen „adblockplus“ geben.

Diese Datenbank muss eine Tabelle mit den „adblock\_data“ haben.

In `/opt/htdocs` muss es ein Verzeichnis mit dem Namen „adblock“.

Darin muss die Datei „adblock.php“ sein.

Die Eingangsdaten sind in dieser Datei beinhaltet (Password, Tabellennamen,...). Diese kann man verändern.

### **Installation:**

1. Öffne Konsole
2. Navigiere zum Verzeichnis, wo sich die Addon-SDK befindet  
(`cd ../../addons`)
3. Tippe: `source bin/activate`
4. Navigiere zum Verzeichnis wo der neue Quelltext vom neuen Adblock ist: (`cd ../../adblockstat`)
5. Tippe: `python build.py autoinstall 8888`

### **Program starten:**

1. Initialisiere XAMPP
  1. Öffne Konsole
  2. Tippe: `sudo /opt/lampp/lampp start`
  3. Tippe Passwort
2. Öffne Firefox
3. Sei sicher, dass Adblockplus aktiviert ist

### **Daten sehen:**

1. Tippe in Firefox: `localhost/phpmyadmin`
2. Klicke auf „adblockplus“ und dann auf „adblockdata“
3. Um die Daten zu exportieren:
  1. Klicke auf „Mehr“
  2. Klicke auf „Export“
  3. Wähle das gewünschte Format

## **6. Schluss**

Die Doktorantin wolle eine Software, die schnell und billig umsetzbar sei.

Der vorliegende Projekt hat diese beide Charakteristiken.

Schnell, weil es in wenig Zeit gemacht wurde.

Billig, weil alles auf Freies Software basiert.