1. Allgemeines

Der Prototyp des Produktes wurde im Rahmen einer Softwarestudie entwickelt und stellt eine beschränkte Funktionalität bereit, die an einen Presseclipping Dienst angelehnt ist. Der prototypische Entwurf legt den grundsätzlichen Aufbau der Software fest. Das Ziel des Prototyps ist die Einarbeitung in die grundlegenden Konzepte und Technologien, welche im weiteren Verlauf der Implementierung verwendet werden sollen. Während der Implementierung unterliegt das Konzept ständiger Überarbeitung und Verfeinerung.

1. Produktübersicht

Der Prototyp basiert auf Java und stellt folgende Funktionalitäten bereit:

● Registrieren von neuen Benutzern unter Angabe von Name, Email-Adresse und Passwort

● Anmelden eines Benutzers

● Anzeigen des Pressespiegels für ein Profil

● Anzeige aller Profile eines Benutzers und Auswahlmöglichkeit

● Abmelden des Benutzers und damit Beenden der Session

Die entsprechende Web-Applikation kann Anfragen eines aktuellen Webbrowsers entgegennehmen und diese dynamisch beantworten. Die Benutzerschnittstelle ist dabei der HTML-Browser. Serverseitig wird eine Java-fähige Umgebung benötigt.

1. Stuktur- und Entwurfsentscheidungen für das Gesamtsystem

Die Software kann in drei Hauptbereiche aufgeteilt werden. Das Frontend ist eine Website, die über einen HTML-Browser erreicht werden kann. Über diese Website kann der Benutzer mit der Anwendung interagieren und darüber kann er auch die Ergebnisse einsehen. Das Backend besteht aus einer Datenbank und der Java-Anwendung. Die Anfragen, die vom Benutzer über die Website ausgelöst werden, werden an die Java-Anwendung weitergeleitet und dort verarbeitet. Die benötigten Daten bezieht die Anwendung von der Datenbank. Ist die Anfrage fertig bearbeitet, wird das Ergebnis an die Website zurückgegeben.  
Dieses Konzept entspricht dem MVC-Entwurfsmuster. Dabei entspricht die Datenbank dem Model, die Website stellt die View dar und die Java-Anwendung ist der Controller. Das Entwurfsmuster basiert auf der Trennung der Software in diese drei Bereiche. Dadurch ist der Quellcode leichter nachzuvollziehen und das Programm ist besser zu warten. Außerdem steigen die Wiederverwendbarkeit und die Testbarkeit der Software.   
Jede der Komponenten stellt eine Schnittstelle zur Verfügung, über die die anderen Komponenten damit interagieren können, ohne deren Implementierung zu kennen. Wird die Komponente verändert, funktioniert die Software immer noch ohne Mehraufwand, solange die Schnittstelle nicht verändert wird.

1. Stuktur- und Entwurfsentscheidungen für die einzelnen Punkte

Die Datenbank enthält alle nötigen Daten. Dazu gehören die Benutzeraccounts und deren jeweilige Profile mit Filter, aber auch die Feeds selbst. Der Controller erstellt entsprechende SQL-Abfragen, um die für ein Profil passenden Feeds von der Datenbank zu holen, und liefert dem Client die erhaltenen Daten. Ein Update der Datenbankeinträge wird ebenfalls vom Controller gestartet. Dabei werden zuerst die in der Datenbank vorhandenen Newsfeed mit den neuen Feeds von den Nachrichtenseiten abgeglichen. Alle neuen Feeds werden dann mit den Filtern der Nutzerprofile verglichen und gegebenenfalls mit dem Profil verlinkt, sodass bei der nächsten Abfrage der Feed an den Nutzer geschickt werden kann. Um mehr Performance zu erreichen, wird für den Update-Prozess Multithreading eingesetzt. Dafür werden in einem Thread die Feeds an den Feedenlarger geschickt und in einem zweiten Thread werden die Filter auf die vom Feedenlarger zuletzt erhaltenen Daten angewandt.  
Bei der Benutzerregistrierung und bei der Anmeldung werden die SQL-Abfragen von der Website gestellt und laufen nicht über den Controller, um diesen nicht zusätzlich zu belasten.