

Softwaretechnikpraktikum
SS 2015
Datum: 15.06.2015

Gruppe: swp15-aae
Betreuer: Prof. Gräbe
Tutor: Klemens Schöllhorn

Projektteam:
Felix Albroscheit
Dorian Dahms
Paul Eisenhuth
Martin Lechner
Christian Seidemann
Ruth von Borell
Franz Wendt

Entwurfsbeschreibung Gesamtprojekt

Inhaltsverzeichnis

1	Allgemeines	2
1.1	Installationsanleitung	2
1.2	Backupkonzept	2
2	Produktübersicht	3
2.1	Drupal	3
2.2	Layout	3
2.2.1	Darstellung Öffentliches Profil	3
2.2.2	Darstellung Kalender	3
2.2.3	Darstellung Events	3
2.2.4	Darstellung Newsfeed	3
2.2.5	Darstellung Interaktive Karte	3
2.3	Inhalte	4
2.4	Funktionalitäten	4
2.4.1	Nutzerregistrierung	4
2.4.2	Kommentarfunktion	4
3	Grundsätzliche Struktur- und Entwurfsprinzipien	4
3.1	MVC/PAC-Modell	4
3.2	Arbeitsteilung	5
3.2.1	Layout	5
3.2.2	Backend	5
3.2.3	Daten	5
4	Struktur- und Entwurfsprinzipien einzelner Pakete	5
4.1	Datenebene	5
4.2	Darstellungsebene	5
4.3	Eigene Drupal-Module	5

4.4	Eigenes Drupal-Theme	6
5	Datenmodell	6
6	Testkonzept	6
6.1	Komponententests	6
6.2	Integrationstests	6
6.3	Systemtests	7
7	Glossar	7

1 Allgemeines

Die Stadtteilplattform Leipziger Osten soll mit umfangreichen Funktionen ausgestattet sein, um gute Nutzbarkeit zu gewährleisten. Akteure haben hier die Möglichkeit, ein öffentliches Profil zu erstellen, ihre Veranstaltungen einzutragen und auf einer anschaulichen Karte und einem Terminkalender mit vielen Filterfunktionen der Öffentlichkeit anzeigen zu lassen. Dadurch soll der Stadtteil Leipziger Osten attraktiver gemacht und Bausteine für Synergien einzelner Akteure gebildet werden. Diese grundlegende Struktur der Plattform wird auf Basis von Drupal realisiert. Dies ermöglicht einen einfachen Ausbau der Webseite mittels Modulen, die für Drupal zu Verfügung stehen. Neben den vorhandenen werden auch eigene Module entwickelt, beispielsweise um die benutzten Module (Kalender, Karte) mit der Datenstruktur zu verknüpfen.

Zur Strukturierung der Datensätze wird RDF benutzt und auf die bereits vorhandenen Ontologien von Leipzig Data zurückgegriffen. Um externe Daten im RDF-Format einzubinden, wird ein Konzept zur Nutzung geeigneter RDF-Ontologien entwickelt. Dadurch wird eine Grundlage für das weitere Arbeiten mit RDF und damit für die Orientierung des Projekts am Konzept Linked Open Data geschaffen. Das Projekt steht also ganz im Sinne von Open Data und bezieht damit viele Datensätze aus bereits erstellten freien Datenständen zu Akteuren der Region.

1.1 Installationsanleitung

Drupal muss auf einem Webserver installiert werden. Anweisungen hierfür sind unter <https://www.drupal.org/documentation/install> zu finden. Bei erfolgreicher Installation müssen die Drupal-Module: Kalender, Ctools, Date und View installiert werden. Es wird ein eigenständiges Modul und Thema entwickelt, so dass diese lediglich in den entsprechenden Drupal-Ordner kopiert und via Backend aktiviert werden müssen. Dabei wird die Datenstruktur (ohne Inhalte) in die Datenbank importiert. Plattforminterne Inhalte könnten über ein Datenbank-Dump erfolgen, welche als Datei im SQL-Format bereit gestellt wird. Externe Inhalte von Leipzig Open Data könnten über eine Funktion im Backend bzw. mittels täglicher CRON-Jobs in die Datenbank integriert werden.

1.2 Backupkonzept

Um ein sicheres Arbeiten zu gewährleisten, wird nicht auf der Drupalinstanz des Servers gearbeitet. Jedes Teammitglied installiert sich eine eigene Drupalinstanz auf seinem lokalen Arbeitsrechner. Der „core“ wird dabei nicht verändert, d.h. Originalthemes und -module werden nicht verändert. Sollten wir Änderungen vornehmen müssen, dann werden diese in einem abgeleiteten

eigenen Modul implementiert, gleiches gilt für Themes. Die bearbeiteten Module werden über das git-Repository verwaltet und in Drupal verlinkt, so dass Fehlschläge rückgängig gemacht werden können.

2 Produktübersicht

2.1 Drupal

Im Zuge des Vorprojektes wurde bereits Drupal installiert, sowie Bausteine für die Karte und das Layout gesetzt. Für das Aufsetzen der Website mit den bereits genannten Funktionen (Kalender, Eventseite, Akteurprofil) werden im Hauptprojekt bündelweise Module implementiert, sowie die RDF-Struktur eingebettet. Hierbei kann auf vorhandene Module zurückgegriffen werden, welche entsprechend angepasst werden bzw. über ein eigenes Modul miteinander verknüpft werden.

2.2 Layout

2.2.1 Darstellung Öffentliches Profil

Ein öffentliches Profil beinhaltet zentrale Informationen zum Akteur (Name, Adresse, E-Mail, Webseite, Telefonnummer) und eventuell Bild- und Videomaterial. Das Profil soll mit den Events, dem Kalender und der interaktiven Karte verknüpft werden. Diese soll auf dem Profil ebenso dargestellt werden. Desweiteren soll eine Kommentarfunktion für User bereitgestellt werden.

2.2.2 Darstellung Kalender

In einem Kalender werden Events zeitlich dargestellt. Über Filteroptionen können hierbei „Wunsch-Events“ angezeigt werden. Von dem Kalender aus sollen die Eventseiten und die öffentlichen Profile erreichbar sein.

2.2.3 Darstellung Events

Events werden auf einer eigenen Seite dargestellt. Diese beinhaltet Name, Datum, Ort, Zielgruppe, Sparte, die Verknüpfung zum öffentlichen Profil des Erstellers und der Darstellung des Events auf der interaktiven Karte. Die Events sollen über den Kalender, das öffentliche Profil des Erstellers und über die Übersichtskarte auf der Startseite erreichbar sein. Desweiteren soll eine Kommentarfunktion für User bereitgestellt werden.

2.2.4 Darstellung Newsfeed

Der Newsfeed soll auf der Startseite dargestellt werden und zeigt Neuigkeiten „aus dem Viertel“, der Webseite und für den Stadtteil relevante Nachrichten aus Leipzig.

2.2.5 Darstellung Interaktive Karte

Eine interaktiver Karte soll Informationen zu Akteuren und Events ortsbezogen darstellen. Die Darstellung wird mit OpenStreetmap realisiert. Die interaktive Karte sollte sowohl als Ganzes auf einer eigenen Seite präsentiert werden, als auch in entsprechenden Ausschnitten auf den

Seiten von Akteuren und Events eingebunden werden. Über Filtermöglichkeiten sollen angezeigte Events aussortiert bzw. hervorgehoben werden. Die Darstellung von Ortsangaben auf der Karte sollte weitestgehend automatisch anhand von Veranstaltungs- bzw. Kontaktadressen erfolgen. Für einige Zwecke kann jedoch auch eine manuelle Darstellung durch einen Veranstalter sinnvoll sein, beispielsweise wenn eine Veranstaltung an einem Ort stattfindet, der nicht durch eine Adresse beschreibbar ist: Für bestimmte Orte innerhalb eines Parks wäre die Eingabe der exakten Koordinaten durch den Veranstalter beim Erstellen des Events die bessere Lösung.

2.3 Inhalte

Eine Datenintegration der K.I.L.O.-Daten ist zur Zeit nicht möglich. Die Daten sind zum größten Teil veraltet und sollen daher nicht ausgegeben werden. Allerdings soll uns in der Zukunft eine Excel-Tabelle übergeben werden, die interessierte Akteure enthalten soll (eventuell mit Daten, die manuell eingetragen werden müssen). Daher ist die Hauptquelle für Inhalte die Datensätze des Leipzig Open Data Projekts, welche noch über einer sinnvollen RDF-Struktur übernommen werden müssen. Inhalte von Akteuren außerhalb der übergebenen K.I.L.O.-Daten oder des Leipzig Open Datas Projekts müssen von diesen selbst eingetragen werden.

2.4 Funktionalitäten

2.4.1 Nutzerregistrierung

Ein User hat die Möglichkeit sich auf der Plattform zu registrieren, um ein öffentliches Profil zu erstellen. Von diesem aus kann er Events erstellen, welche, wie das Profil, von Besuchern der Webseite eingesehen werden können. Das Formular zur Registrierung muss von allen Seiten der Plattform aus über einen entsprechenden Link erreichbar sein, ebenso wie die Login/Logout-Funktion.

2.4.2 Kommentarfunktion

Die Kommentarfunktion soll von Besuchern und registrierten Usern genutzt werden können, um eigene Meinungen auf der Event und Profilseite zu verfassen und sich auszutauschen. Kommentare registrierter Nutzer sollen direkt veröffentlicht werden, während die von nicht registrierten Nutzern zwar gespeichert, aber noch nicht publiziert werden, um eine Vorabmoderation zu ermöglichen.

3 Grundsätzliche Struktur- und Entwurfsprinzipien

3.1 MVC/PAC-Modell

Für das Projekt adaptieren wir das MVC-Modell. Im Model werden die Datensätze verwaltet. Der Controller reagiert dann auf Nutzeranfragen, wie einer Suche und leitet diese an das Model weiter. Das Model verarbeitet diese Anfrage und gibt das Ergebnis zur Darstellung an den View weiter. Drupal selbst baut auf eine erweiterte Form des MVC-Modells auf. Im Presentation-Abstraction-Control oder PAC Modell entspricht Presentation dem View, und Abstraction dem Model. Im Gegensatz zu MVC findet diese Dreiteilung der Architektur im PAC jedoch für jeden Systemteil („Agent“) statt, der eine spezifische Aufgabe erfüllt. Diesen Agents entsprechen die einzelnen Drupalmodule. Die Kommunikation / Zusammenarbeit dieser stellt einen kritischen

Punkt für das Gelingen des Projekts dar. Dem Controller-Bereich des selbstgeschriebenen Moduls sollte daher besondere Aufmerksamkeit geschenkt werden. Da es im Sinne des PAC-Modells ist, einzelne „Agents“ für bestimmte Aufgaben zu haben, sollte eventuell in Betracht gezogen werden, das selbstgeschriebene Modul in mehrere aufzuteilen, sodass es nicht mit verschiedenen Funktionalitäten überladen ist.

3.2 Arbeitsteilung

Das Team arbeitet in der Implementierungsphase in kleineren Gruppen, um eine parallele aber personell möglichst getrennte Entwicklung von Layout, Backend, und Datenstruktur zu ermöglichen.

3.2.1 Layout

Der/die Layouter sind mit der Gestaltung der Plattform und ihrer interaktiven Elemente beauftragt. Das Layout wird fortschreitend mit den anderen Ebenen der Plattform entwickelt, um auf eventuelle Änderungen von deren Seiten aus reagieren zu können.

3.2.2 Backend

Die Backend-Gruppe ist für die Installation, das Einrichten der Drupal-Instanz verantwortlich. In der Implementierungsphase arbeitet dieser Teil des Teams an der Entwicklung eines eigenen Drupal-Moduls und der Verknüpfung aller Module und Funktionalitäten des Systems.

3.2.3 Daten

Dieser Aufgabenbereich umfasst im aktuellen Fall die „Akquise“ externer Daten für die Plattform, die Entwicklung eines Konzepts für deren Notation in RDF, sowie ihre tatsächliche Überführung in dieses Format.

4 Struktur- und Entwurfsprinzipien einzelner Pakete

4.1 Datenebene

Die Daten werden in ein RDF-Format überführt oder stehen bereits als solches zur Verfügung. Die Strukturierung bzw. Bezeichnungen orientieren sich an den Ontologien von Leipzig Data.

4.2 Darstellungsebene

Es soll eine Hauptseite geben, die die interaktive Karte, den Newsfeed, sowie eine Übersicht über anstehende Events und Navigationsleiste beinhaltet. Weiterhin soll es Seiten geben für: Profile, Events, den Kalender, die Registrierung, die Karte, FAQ und dergleichen.

4.3 Eigene Drupal-Module

Um die Zusammenarbeit vorgefertigter Drupal-Module zu ermöglichen, und um soweit möglich auf fehlerhafte / schwer integrierbare Module zu verzichten, wird ein Teil der Funktionalitäten

in vom Entwicklungsteam selbst geschriebenen Drupal-Modulen umgesetzt. Der genaue Umfang der oder des Moduls steht noch nicht fest.

4.4 Eigenes Drupal-Theme

Um bei der Gestaltung des Layouts freie Hand zu haben und eigene Ideen umsetzen zu können, wird für die Stadtteilplattform ein eigenes Drupal-Theme entwickelt. Es existiert dazu bereits ein Entwurf: <http://grinch.pavo.uberspace.de/lo/>. Dieser wird noch auf die Drupaltheme überführt und erfüllt bereits das Merkmal „Responsive“.

5 Datenmodell

Das Projekt orientiert sich an der RDF-Ontologie des Leipzig Data Projekts.

Wir streben an, dass die Daten in einer Graphdatenbank abgespeichert werden, da es das Konzept von Open Data besser umsetzt und es später den Rückfluss zu Leipzig Data erleichtert. Allerdings benutzen wir übergangsweise eine relationale Datenbank, da sie die Einarbeitung erleichtert und die Entwicklung unserer Module beschleunigt.

6 Testkonzept

Um in der Implementierungsphase effizient arbeiten zu können, dürfen die Tests selbst nicht unnötig viel Zeit in Anspruch nehmen. Da ein großer Teil des Teams weder Erfahrung mit Drupals Test-Modul, noch mit Softwaretests im allgemeinen hat und daher extra geschult werden müsste, werden nicht alle Teammitglieder Tests durchführen. Stattdessen werden die im folgenden erläuterten Tests von den am Backend arbeitenden Mitgliedern ausgeführt, die das zu testende Material selbst geschrieben haben.

6.1 Komponententests

Bei der Implementierung einzelner Komponenten (Module) ist es wichtig, diese auf Funktionalität und Sicherheit zu testen. Auch wenn wir viel mit bereits vorhandenen Drupalmodulen realisieren werden, sollte jedes einzelne Modul noch einmal geprüft werden. Nur dadurch ist ein einwandfreies Funktionieren gesichert. Tests werden in Testprotokollen dokumentiert, um allen Output festzuhalten und Änderungen nachvollziehen zu können. Für die Komponententests benutzen wir das interne Drupalmodul „Testing“.

Im Rahmen des Vorprojekts wurden bisher noch keine Komponententests durchgeführt.

6.2 Integrationstests

Auch wenn alle Komponenten richtig funktionieren kann es zu Schwierigkeiten beim Zusammenwirken kommen. Es ist daher wichtig, bei dem Einfügen einer Komponente deren Zusammenspiel mit bereits implementierten Komponenten zu testen. Dies wird ebenfalls protokolliert und mittels „Testing“ geprüft.

Im Rahmen des Vorprojekts wurden bisher noch keine Integrationstests durchgeführt.

6.3 Systemtests

Nach allen Komponententests und Integrationstests muss abschließend das Gesamtsystem geprüft werden. Es wird getestet, ob das Produkt innerhalb der späteren Nutzungsumgebung funktioniert. Für immer gleiche Anfragen benutzen wir das Tool „Selenium“, welches auch bereits in dem Modul „Testing“ integriert ist. Dieses automatisiert Browseranfragen und gestattet auch Stresstests, da beliebig viele Anfragen auf einmal gestellt werden können. Auch das wird protokollarisch festgehalten.

Im Rahmen des Vorprojekts wurden bisher noch keine Systemtests durchgeführt.

7 Glossar

Stadtteilplattform:

Eine interaktive (Online)-Plattform, welche der Organisation, Verschönerung, Attraktivität, Vermittlung, „News-Verbreitung“ und vielem mehr dienen soll. Die Plattform sollte so aufgesetzt sein, dass sie in gewisser Weise selbst funktioniert und mit Inhalten bespielt wird. Das heißt Nutzer können sich registrieren, Inhalte kommentieren sowie für im Viertel aktive Akteure öffentliche Profile anlegen und über diese Veranstaltungen und Angebote veröffentlichen, ohne dass alles von einem Betreiber der Seite im Voraus einzeln kontrolliert werden muss. Aus inhaltlichen, gesetzlichen, datenschutzrechtlichen Gründen kann eine Nachmoderation, oder im Fall unregistrierter Nutzer auch eine Vorabmoderation jedoch nötig sein, siehe „Nutzer“/„Seitenbetreiber“). Ziel der Plattform ist es, eine übersichtliche Website zu gestalten, die mittels Interaktiver Karte, Kalender, etc. den Stadtteil mit seinen Akteuren attraktiv macht.

Nutzer:

Nutzer sind zunächst jegliche Besucher der Plattform, die diese zu Informations- und Präsentationszwecken nutzen. Auch alle anderen auf der Plattform aktiven Menschen (z.B. von Betreiberseite) können zusätzlich in der Rolle des Nutzer auf ihr unterwegs sein. Nutzer können alle öffentlichen Seiten der Plattform aufrufen, sowie auf Akteursprofilen und Veranstaltungsseiten Kommentare hinterlassen - diese müssen jedoch vor ihrer Veröffentlichung von einem Moderator freigeschaltet werden. Nutzer haben zusätzlich die Möglichkeit, sich auf der Plattform mit einem eigenen Account zu registrieren. Dies ermöglicht ihnen, Kommentare ohne vorherige Moderation veröffentlicht zu bekommen. Weiterhin ermöglicht eine Registrierung das Anlegen eines öffentlichen Profils für einen im Stadtteil aktiven Akteur anzulegen, den der Nutzer auch im realen Leben vertritt. Dies muss im Zweifelsfall von den Seitenbetreibern überprüft und verifiziert werden.

Akteur:

Unter Akteuren sind zunächst Folgende zu verstehen: Veranstalter, Vereine, Initiativen und Privatpersonen, die im Leipziger Osten auf verschiedene Art aktiv sind. Im Rahmen der Architektur unserer Plattform bezeichnet „Akteur“ auch die Repräsentation einer solchen Institution, die online von einem oder mehreren Nutzern, die der Institution auch in der realen Welt angehören, (Akteursinhaber & Akteursmitglieder) vertreten und verwaltet wird. Die öffentliche Repräsentation eines Akteurs auf der Plattform ist das zugehörige Akteursprofil.

Akteursinhaber/Akteursadmin:

Registrierter Nutzer, der für einen realweltlichen Akteur ein öffentliches Profil auf der Stadtteilplattform anlegt, und für dieses Administratorrechte besitzt. Diese umfassen vollständige Bearbeitungsrechte für die Inhalte des Profils (ausgenommen Moderation von Nutzerkommentaren), sowie das Recht die Löschung des Profils vorzunehmen, bzw. zu veranlassen, oder den Adminstatus an einen anderen registrierten Nutzer oder die Plattformbetreiber abzugeben.

Akteursmitglied:

Das öffentliche Profil eines Akteurs soll auch von mehreren Personen verwaltbar sein. Akteursmitglieder sind registrierte Nutzer, denen vom Akteursadmin beschränkte Bearbeitungsrechte zugewiesen wurden. Diese können beispielsweise die Bearbeitung von (bestimmten) Inhalten einschließen, aber Akteursmitgliedern keine Löschung des Profils zu ermöglichen.

Akteursprofil / Kurzdarstellung:

Von Nutzern verwaltete Selbstdarstellung von Akteuren auf der Stadtteilplattform, ähnlich eines Profils in sozialen Netzwerken. Dieses sollte mindestens folgende Daten umfassen: Name, Beschreibung, Adresse, sonstige Kontaktmöglichkeit (E-Mail, Facebook...), Sparte, Zielgruppe. Optional sind Bilder, etc. Um ein Profil für einen Akteur auf der Plattform anzulegen, benötigen die ihm zugehörigen Nutzer eine entsprechende Zugangsmöglichkeit über einen registrierten Account.

Seitenbetreiber:

Inhaber und Betreiber der Stadtteilplattform, insbesondere nach Übergabe des Projekts. Der Seitenbetreiber kann eine Person oder eine Institution, z.B. ein Verein sein. In der Praxis können im Seitenbetreiber auch verschiedene administrative Aufgaben vereint sein, beispielsweise technische Administration, Nutzerverwaltung, sowie Moderation und redaktionelle Tätigkeiten. Eine andere personelle und inhaltliche Verteilung der administrativen Aufgaben ist ebenfalls denkbar.