

PR Anwendungsentwicklung i.d. Medieninformatik

EventJunkie

Status Report

WS 2014/15 - Gruppe PK2 Team: PK2

Contributors:

| MatNr. | Name | e-mail |
|----------|---------------------|----------------------------|
| a1125503 | Oliver Guggenberger | a1125503@unet.univie.ac.at |
| a1207617 | Philipp Hierman | a1207617@unet.univie.ac.at |
| a1151917 | Sandra Markhart | a1151917@unet.univie.ac.at |

December 14, 2014

Contents

| | | |
|----------|--|----------|
| 1 | Progress | 2 |
| 1.1 | Summary | 2 |
| 1.2 | Git-Repository | 2 |
| 1.2.1 | Namen auf GitHub | 2 |
| 1.3 | Status per Milestone | 3 |
| 1.4 | Activities per Person | 3 |
| 1.4.1 | Sandra Markhart | 3 |
| 1.4.2 | Oliver Guggenberger | 4 |
| 1.4.3 | Philipp Hiermann | 5 |
| 1.5 | Report on working hours per person | 6 |
| 1.5.1 | Sandra Markhart | 6 |
| 1.5.2 | Oliver Guggenberger | 6 |
| 1.5.3 | Philipp Hiermann | 6 |
| 2 | Risk Analysis | 6 |
| 3 | Outlook | 7 |
| 3.1 | Milestones and Schedules | 7 |
| 3.2 | Planned Effort per Person | 7 |
| 3.2.1 | Sandra Markhart | 7 |
| 3.2.2 | Oliver Guggenberger | 7 |
| 3.2.3 | Philipp Hiermann | 8 |

1 Progress

1.1 Summary

Die Seite ist auf <http://eventjunkie.funatika.org> abrufbar.

Alle Basisfunktionalitäten wie Event erstellen/anzeigen/updaten/löschen/verlinken, sowie die Flickr/Twitter/GoogleMap/GoaBase-APIS wurden implementiert. "Simple Search" wurde für normale und GoaBase-Events implementiert, Advanced-Search nur für normale Events mit der ausgewählten Checkbox "Site" ohne "GoaBase". Beim Erstellen und Updaten von Events sind alle benötigten Felder vorhanden und werden im Event-Model validiert. Mit "Add Field" können Social-Media-Felder hinzugefügt, welche im Social-Media-Model validiert werden. Über die dazugehörige API werden die Social-Media-Daten abgefragt und in ein socialmedia-Array geschrieben. Events können auch von nicht-angemeldeten Benutzern mit Social-Media-Seiten verlinkt werden.

Ausstehend für die zweite Iteration sind noch Advanced-Search für GoaBase und beide Seiten zusammen, Ajax-Validierungen und die Administration von Events durch den Admin. Auch muss noch die Facebook-API implementiert werden. Um normale Events und GoaBase-Events besser durchsuchen zu können, sollen die GoaBase-Events in einer eigenen Datenbank-Tabelle abgespeichert werden, und nach einer bestimmten Cache-Zeit gelöscht und über die API neu abgefragt werden. Dadurch soll es uns leichter möglich sein, dass Events, egal ob von der Seite oder von GoaBase durchsucht und sortiert werden können. Somit ist es auch möglich die Pagination übergreifend für beide Event-Typen einzubauen. Auch soll die Image-Gallery mit JavaScript umgesetzt werden.

1.2 Git-Repository

Das Git-Repository ist auf <https://github.com/reiko1991/eventjunkie> zu finden. Dort sind unser aktuelles Projekt und alle zugehörigen Commits zu finden.

1.2.1 Namen auf GitHub

guggi Oliver Guggenberger

fips1986 Philipp Hiermann

reiko1991 Sandra Markhart

1.3 Status per Milestone

| Meilenstein | Datum (von) | Datum (bis) |
|--|-------------------|-------------------|
| 1 Anforderungen und des Designs Das Anforderungen und Design Dokument wurde erstellt und der Meilenstein ist zu 100% abgeschlossen und wurde zeitgerecht gestartet und beendet. | 14.10.2014 | 9.11.2014 |
| 2 Prototyp, Statusbericht Fast alle geplanten Funktionalitäten wurden implementiert und der Statusbericht erstellt. Einzelne Funktionalität (Advanced Search) wurden nicht implementiert und werden stattdessen bis zur finalen Demo hinzugefügt. Insgesamt wurden aber 95% des Meilensteins erreicht. | 24.10.2014 | 14.12.2014 |
| 3 Präsentation, Endbericht Präsentation und Endbericht müssen noch erstellt werden. Zudem auch eine AJAX-Validierung, GoaBase-Event-Table in der Datenbank und Adminsitration der Events durch den Admin. | 20.01.2014 | 25.01.2014 |
| 4 Finale Demo Einzelne Funktionalitäten müssen noch implementiert und getestet werden. | 15.12.2014 | 31.01.2014 |
| 5. Präsentation der Demo | 31.01.2014 | 31.01.2014 |

1.4 Activities per Person

1.4.1 Sandra Markhart

Events: Create/Update/Delete/View/Link ist jetzt möglich. Create ermöglicht das Erstellen von Events, Update das Bearbeiten und Delete das Löschen. Mit "View" wird ein Event mit einer bestimmten ID angezeigt. Dabei werden auch die zugehörigen Social-Media-Objekte aus dem Cache gelesen. Link-Events ermöglicht es einen Social-Media-Link zu einem Event zu erstellen, auch wenn dieser einem nicht gehört.

- Event-Table und SocialMedia-Table in der Datenbank erstellt.
- Erste Version der Event-Index-Seite mit Suchformular erstellt, auf der die normalen Events nach Datum gereiht aufgelistet werden. Zusätzlich Top-Events nach Clicks und New-Events nach Erstellungsdatum.
- Advanced Search/Simple Search für ausschließlich normale Events.
- Merken der Auswahl Advanced Search/Simple Search nach neuem Request.

- Views und Controller-Actions für Create/Update/Delete/View/Link erstellt.
- Gallery-Link mit Gallery zu den jeweiligen SocialMedia-Bildern der Seite.
- AddField-Funktion um SocialMedia-Felder zum Formular mittels JavaScript hinzuzufügen.
- Validierung von Events nach Datum, Bildformat usw. und von SocialMedia-Links beim Erstellen/Updaten im Event-Model.
- Link-Events-with-SocialMedia für nicht-angemeldete User.

SocialMedia APIs: Erstellen eines socialmedia-Objekts mit der SocialMediaApi aus Twitter-Links/Hashtags und Flickr-Links. Zugriff auf die GoogleMaps-API.

- Cache von Social-Media-Objekten mittels FileCache für 5 Minuten. Dieser wird ebenfalls beim Updaten für das jeweilige Event gelöscht und die Daten neu von Flickr/Twitter etc. abgefragt.
- SocialMediaApi-Klasse erstellt um ein SocialMedia-Objekt mit "images" und "comments" anzulegen. Doppelte Bilder werden dabei entfernt und Kommentare nach Datum sortiert.
- FlickrApi-Klasse erstellt um Gallery/Set/Photo-Links zu parsen und alle zugehörigen Bilder (thumbnail/original) zu erhalten. Dabei wurden Requests über eine REST-Schnittstelle gestellt und die zurückgelieferten JSON-Objekte geparsed.
- TwitterApi-Klasse erstellt um Tweets nach einem Hashtag-Link oder Hashtag zu durchsuchen und zugehörige Tweets und Bilder der Tweets aus dem erhaltenen JSON-Objekt auszulesen. Tweets werden zusätzlich nach Hashtags, Benutzern und anderen Links geparsed und diese durch eine Verlinkung ersetzt. Für die Authentifizierung mit OAuth wurde die PHP-Library "j7mbo/twitter-api-php" verwendet.
- Adressen werden mit Hilfe der Geolocating API (JavaScript und serverseitig) im Event-Model geparsed und die Longitude/Latitude-Daten abgespeichert.
- Erstellen einer Karte mittels der GoogleMaps-API und aufgelistete Events im Event-View und auf der Event-Index-Seite auf der Karte mit einem Marker, der die Daten des Events enthält anzeigen. Dies wurde mit der GoogleMaps-API und JavaScript umgesetzt.

1.4.2 Oliver Guggenberger

User: Geeignetes Usermanagement ausgesucht und in Applikation integriert (Amnah Usermanagement).

Events: Events von GoaBase werden auf der Eventseite angezeigt. Dadurch ist die Eventseite nie leer, auch wenn keine internen Events existieren. Wenn es jedoch interne Events gibt, werden die GoaBase-Events mit den internen Events, nach Datum angezeigt. Die GoaBase-Events werden aus dem Cache gelesen, falls diese schon einmal aufgerufen worden sind.

- Erstellung des EventContollers.
- Weiterleitung von SiteController → EventController
- Suche in Advanced und Simple Search aufgeteilt
- GoaBaseEvents integriert
- Interne Events mit Goabase-Events vermischen und nach Datum sortiert.
- GoaBase view erstellt (Für Detail Ansicht eines Goabase-Events)

GoaBaseApi: Schnittstelle zu GoaBase. Laden von beliebig vielen Goaparties oder von Detail Informationen zu einer gewissen Party mithilfe der Party Id. Wenn eine Liste von Parties geladen wird, sind die einzelnen Parties so integriert, dass diese vom Typ ein Event Objekt sind und somit den gleichen Typ haben wie interne Events.

- GoabaseApi-Klasse erstellt
- Detail Informationen zu einer bestimmten Party mit Party-Id abfragen
- Partieliste mit GoaBase-Events. Diese haben den Typ Events wie interne Events.
- Start und End Datum wird nach dem Format der internen Parties angepasst.
- Erhaltenen GoaBase Koordinaten in brauchbare Adresse parsen (Google maps Api).

Info: Informationsseite erstellt. Informationen bzw Overview über EventJunkie werden in der Infoseite angezeigt.

Contact: Kontaktseite erstellt. Über ein Kontaktformular kann Kontakt mit den Administrator aufgenommen werden.

Generell: EventJunkie auf private Domain installiert bzw. Daten migriert (Datenbank und Tabellen). Internetadresse: <http://eventjunkie.funatika.org/>

1.4.3 Philipp Hiermann

Events: Bei Eingabe einer Zeichenkette in die Suchmaske, werden dazu passend Events als möglich Sucheingaben angezeigt. Dies soll asynchron erfolgen.

- Geeignete Funktion im Eventcontroller zum Laden der Daten erstellt

- Javascript-Callbacks zur Behandlung der geholten Daten erstellt

FacebookApi: Die Klasse FacebookApi stellt die Schnittstelle zur API von Facebook dar. Ziel ist es, dass zumindest Kommentare zu einem Event auf unsere Seite sichtbar sein sollen.

- Klasse FacebookApi erstellt
- Geeignete Funktion zum Parsen von Facebook-Links erstellt

1.5 Report on working hours per person

1.5.1 Sandra Markhart

Die genauen Stunden wurden von mir nicht dokumentiert, jedoch kann der Aufwand mit den Commits zu unserem Git-Repository nachvollzogen werden. Ich denke aber, dass die geplanten 65 Stunden erreicht wurden.

1.5.2 Oliver Guggenberger

Ich habe meine geleisteten Stunden ebenfalls nicht mitdokumentiert, jedoch kann der Aufwand mit den Commits zu unserem Git-Repository nachvollzogen werden. Die geplanten 65 wurden bestimmt erreicht.

1.5.3 Philipp Hiermann

Auch bei mir gilt: Die Arbeitsstunden wurden nicht notiert. Jedoch habe ich mit großer Wahrscheinlichkeit keine 65 Stunden daran gearbeitet (eher 40).

2 Risk Analysis

Was kann passieren, das das Projekt gefährdet?

- **Risiko:** Ausfall von Team-Mitgliedern
Gegenmaßnahmen: Jedes Team-Mitglied kennt sich im gesamten Projekt aus. Dadurch können die Aufgaben eines ausgefallenen Teammitgliedes an die verbleibenden Mitglieder aufgeteilt werden.
- **Risiko:** Wir nehmen uns zu viel vor
Gegenmaßnahmen: Iterativer Entwicklungsansatz (je eine überschaubare Teilmenge an Features); regelmäßige Diskussion und Feedback mit Übungsleiter. Derzeit eher unwahrscheinlich, da das Projekt schon recht weit fortgeschritten ist.
- **Risiko:** Ausfall von benötigter Infrastruktur (unwahrscheinlich, beherrschbar)
Gegenmaßnahmen: Abhängigkeit von zentraler Infrastruktur minimieren: Verteiltes Source-Code-Management mit git; Anwendung kann einfach lokal getestet und vorgezeigt werden (lokaler Apache)

- **Risiko:** Das Projekt findet keinen Anklang bei der gewünschten Zielgruppe
Gegenmaßnahmen: Seite intuitiv designen, hohe Latenzzeiten vermeiden um eine annehmbare Performance gewährruleisten.
- **Risiko:** Jeder hat seine eigenen Vorstellungen und möchte diese umsetzen, Kompromisse sind sehr schwer einzugehen.
Gegenmaßnahmen: "In den sauren Apfel beißen", versuchen, die eigenen Vorstellungen mit denen der anderen zu vereinen und das Beste aus beidem zu machen.

3 Outlook

3.1 Milestones and Schedules

Diesselben wie im "Requirements und Design"-Dokument inklusive der erwähnten Punkte bei Punkt 1.1 Summary und 1.3 Status per Milestone.

3.2 Planned Effort per Person

3.2.1 Sandra Markhart

Events:

- Anzeigen von eigenen Events in einer Liste um schneller auf diese zugreifen zu können.
- SocialMedia-Bilder bei Klick nicht mehr statt der Seite öffnen sondern in einem kleinen Fenster (eventuell mittels JavaScript).

Admin:

- Auflisten aller Events in einer Tabelle und Löschen von Events durch den Admin ("Manage Events").

Allgemein:

- Testen und Fehler beheben.
- Mitarbeit an der Präsentation und dem Endbericht.

Da die meisten geplanten Punkte schon in der ersten Iteration implementiert wurden, rechne ich mit einem maximalen Aufwand von 40 Stunden.

3.2.2 Oliver Guggenberger

Events:

- Advanced Search implementieren

Allgemein:

- Testen und Fehler beheben.

- Mitarbeit an der Präsentation und dem Endbericht.
- EventJunkie im Internet deployen und aktuell halten.

3.2.3 Philipp Hiermann

Events:

- Suchmaske im Bezug auf Ajax verbessern

FacebookApi:

- Api fertig implementieren

Allgemein:

- Testen und Fehler beheben.
- Mitarbeit an der Präsentation und dem Endbericht.