

HOCHSCHULE HEILBRONN
FAKULTÄT WIRTSCHAFT 1

STUDIENGANG ELECTRONIC BUSINESS

Konzeption und Implementierung einer Metabildergalerie

Vorgelegt bei

Prof. Dr. Michael Gröschel

Von

David König Matrikelnummer: 160923

Im

Sommersemester 2008

Management Summary

Das Internet erlebte in den letzten Jahren einen regelrechten Boom, der viele unterschiedliche Geschäftsmodelle hervorgebracht hat. Im Wesentlichen sorgte das sogenannte Web 2.0 dafür, dass viele User am Netz partizipierten.

Ein wesentliches Web 2.0-Erfolgsmodell sind die Bilderdienste. Mittlerweile existieren viele Bilderdienste, die dem Benutzer eine Plattform bieten, Sammlungen von Bildern zu verwalten. Diese Plattformen sind sehr komfortabel und glänzen durch deren einfache Bedienung.

Möglich machte diese Entwicklung die AJAX-Technologie, mit der es möglich ist, im Hintergrund Abfragen an den Server zu senden, ohne dass eine Seite neu geladen werden muss. AJAX steht für Asynchrones Javascript mit XML. Realisiert wird die Abfrage durch das XMLHttpRequest-Objekt, dessen Ergebnisse mittels Javascript wieder in die Seite integriert werden.

Ein weiterer Aspekt der Bilderdienste ist ein Web-Service, über den, mit der freigegebenen API, auf die Daten zugegriffen werden kann. Dies eignet sich hervorragend um die vom Bilderdienst gehosteten Fotos in eigene Applikationen zu integrieren.

Unter den Bilderdiensten gibt es starke Vertreter, wie Flickr oder Picasa. Diese heben sich durch die Einsatzvielfalt oder durch besondere Upload-Tools hervor, besitzen aber auch Nachteile. Meistens wird die kostenlose (private) Nutzung eingeschränkt, wie z.B. durch die Beschränkung der Datenmenge für einen bestimmten Zeitraum.

Deshalb besitzen Foto-affine Personen mehrere dieser Accounts. Die Idee dieser Diplomarbeit ist es einen Versuch zu unternehmen, beliebige Bilderdienst-Accounts in einem Meta-Account zu aggregieren.

Ruby on Rails hat während der Web 2.0 Phase stark an Bedeutung gewonnen und eignet sich sehr gut für die Entwicklung von Web 2.0 Projekten, da Rails von Haus aus über eine AJAX-Bibliothek verfügt, mit der sehr schnell brauchbare Ergebnisse erzielt werden.

Die Diplomarbeit beschreibt die Konzeption und Implementierung eines Meta-Galerie-Prototyps. Die Meta-Galerie ist Teil des photoBizKit-Projekts und soll in späteren Phasen mit anderen Komponenten verzahnt werden. Ziel ist die Erstellung eines Photo Business Kits, mit dem der Benutzer seine vom Bilderdienst gehosteten Bilder in einen Online-Printservice übertragen kann. Für die verkauften Bilder bekommt er dann eine Provision, die er auch selbst bestimmen kann.

Im ersten Schritt wurde in einem kleinen Team, mit den beteiligten Personen, eine genaue Anforderungsanalyse durchgeführt. Ergebnis war ein Lastenheft, das die Anforderung grob umreißt.

In der Konzeptionsphase wurden die erhobenen Anforderungen modelliert. Dazu kam eine Methode zum Einsatz, die ein Geschäftsprozessmodell in Use-Case-Diagramme wandelt. Die

Use-Cases waren wiederum Basis für die Aktivitäten. Vorteil dieser Methode ist, dass alle Anforderungen durchgereicht werden, man sagt auch, die Semantische Lücke wird geschlossen. Weitere Modelle waren kleine Klassendiagramme und ein paar Wireframes für das Layout.

Wireframes helfen Funktionen auf Webseiten zu strukturieren. Sie bieten den Vorteil, dass sie sehr schnell erstellt werden können. Modifikationen sind jederzeit möglich. Sie geben aber nicht das endgültige Layout vor, helfen nur beim Visualisieren der Elemente.

Im Implementierungskapitel wird beschrieben wie der Prototyp umgesetzt wurde. Die wichtigsten Domänen werden mit Controllern, Models und Actions beschrieben. Außerdem wird die Konfiguration der Umgebung und der Routen beschrieben. Abschließend wird erläutert, wie der SelectionManager, eine zentrale Klasse des Prototyps, funktioniert.

Die Entwicklung mit Ruby on Rails ist einfach und macht Spaß. Die natürlich-sprachigen Eigenschaften der Skript Sprache erleichtern das Programmieren sehr. Natürlich ist die Einstiegshürde ziemlich hoch, da man sich erst Ruby und dann Rails aneignen muss, nicht zu vergessen, die Javascript-Frameworks, die fest im Rails-Framework verankert sind.

In der nächsten Phase der Metagaleries werden das endgültige Layout und die Texte integriert. Der Prototyp muss um die Eigenschaften der Mehrsprachigkeit erweitert werden und in ferner Zukunft erfolgt die Verzahnung mit den anderen photoBizKit-Modulen.

Inhaltverzeichnis

Management Summary	II
Inhaltverzeichnis.....	IV
Abbildungsverzeichnis	VII
Listingverzeichnis.....	VIII
Tabellenverzeichnis	IX
Abkürzungsverzeichnis	X
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	2
1.3 Vorgehensweise.....	2
2 Wissenschaftliche Grundlagen	3
2.1 Ruby	3
2.1.1 Einleitung	3
2.1.2 Installation	3
2.1.3 Notations-Konventionen in Ruby.....	4
2.1.4 Objektorientierung.....	5
2.1.5 Datentypen.....	5
2.1.6 Container, Blöcke und Iteratoren.....	6
2.1.7 Expressions	6
2.1.8 Fehlerbehandlung	8
2.1.9 Threads und Prozesse	9
2.1.10 Unit Tests.....	9
2.1.11 RubyGems	9
2.2 Ruby on Rails	10
2.2.1 Einleitung	10
2.2.1 Installation	11
2.2.3 Model-View-Controller	11
2.2.4 Ein Rails Projekt.....	12
2.2.5 Active Record - Model	13
2.2.6 Action Pack	14
2.2.7 Namenskonventionen	14
2.2.8 Konfiguration	15

2.2.9 AJAX.....	16
2.2.10 Sicherheit.....	17
2.3 Online Bilderdienste.....	17
2.3.1 Einleitung.....	17
2.3.2 Eigenschaften der Bilderdienste.....	17
3 Die photoBizKit gallery.....	18
3.1 Anforderungen.....	18
3.1.1 Fachliche Anforderungen.....	19
3.1.2 Technische Anforderungen.....	22
3.2 Konzeption.....	23
3.2.1 Einleitung.....	23
3.2.2 Modellierung des Geschäftsprozesses.....	23
3.2.3 Modellierung der Use-cases.....	25
3.2.3.1 Bilderdienste verwalten.....	25
3.2.3.2 Selektion verwalten.....	26
3.2.3.3 Galerien verwalten.....	27
3.2.4 Modellierung der Aktivitäten.....	28
3.2.4.1 Bilderdienstverwaltung – Neuer Bilderdienst.....	28
3.2.4.2 Selektionenverwaltung – Auswahl Bilderdienste.....	30
3.2.5 Modellierung der Architektur (Klassendiagramme).....	31
3.2.6 Wireframes.....	32
3.2.6.1 Bilderdienstverwaltung – Übersicht.....	33
3.2.6.2 Selektionenverwaltung – Neue Selektion.....	34
3.2.6.3 Galerieverwaltung.....	35
3.3 Implementierung.....	36
3.3.1 Einleitung.....	37
3.3.2 Konfiguration der Rails-Umgebung.....	37
3.3.3 Architektur.....	39
3.3.3.1 Einleitung.....	39
3.3.3.2 Das Main-Layout.....	39
3.3.3.3 User - HomeController.....	41
3.3.3.4 Hoster – HosterController.....	42
3.3.3.5 Selection – SelectionController.....	44
3.3.3.6 Gallery - GalleryController.....	47

3.3.3.7 Show – ShowController	48
3.3.3.8 SelectionManager - Businessklasse.....	50
3.4 Migration	54
4 Fazit und Ausblick.....	55
Anhang	V
Lastenheft Metagalerie V1.0	V
Geschäftsprozess-/ Use-case-/ Activitydiagram.....	XV
Quellcode der Anwendung	XLVII
Controller.....	XLVII
Helpers.....	LIV
Models	LVII
Views.....	LVII
Layouts	LXII
Konfiguration	LXIX
SelectionManager	LXXIII
Literaturverzeichnis	LXXXIV
Internetquellen.....	LXXXV
Eidesstattliche Erklärung.....	LXXXVI