Konzeption und Entwicklung einer CD Tauschbörse mit Ruby Rails

Christian Bunk
Alexander Miller
Christian Sandvoß
Antonia Ziegler

Inhalt

1. Aufgabenstellung
2. Framework
3. Projektmanagement
4. Implementierung
5. Veröffentlichung
6. Extras
7. Zusammenfassung

AUFGABENSTELLUNG

Aufgabe und Zielstellung

- Konzeption und Entwicklung einer CD-Tauschbörse
- Rich-Applikation (RIA)
- Framework: Ruby on Rails
- Projektmanagement und Dokumentation

FRAMEWORK

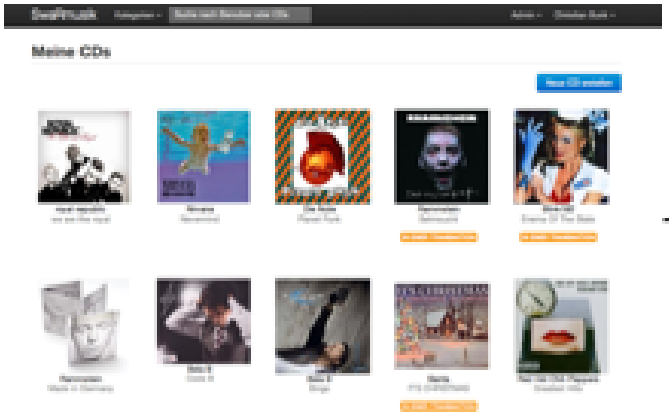
Ruby

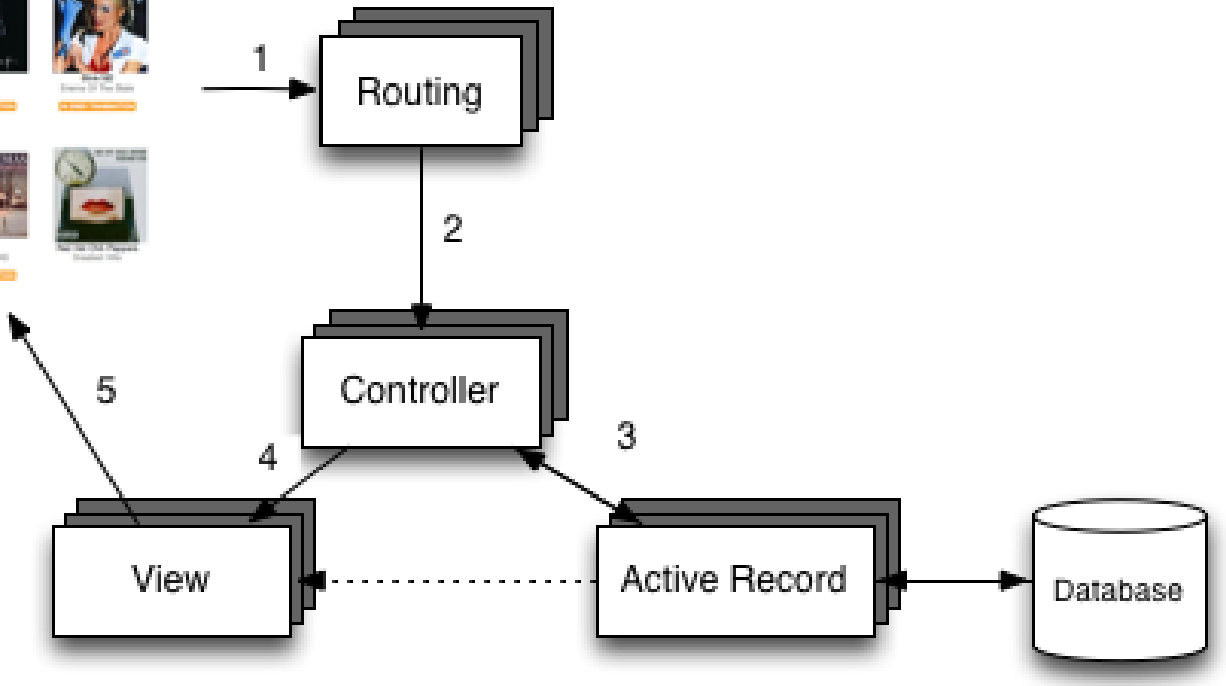
- Entwickelt von Yukihiro Matsumoto (1995)
- Version 1.9.3
- Interpretiert, plattformunabhängig
- Vollständig Objektorientiert `5.times{...}`
- Große Community
- Sehr viel Dokumentation

Ruby on Rails

- Webframework auf der Sprache Ruby
- Prinzipien
  - Don‘t Repeat Yourself (DRY)
  - ConventionOverConfiguration
- MVC bestehend aus fünf Komponenten:
  - Active Support
  - Action Pack
  - Action Mailer
  - Active Ressource
  - ActiveRecord
- WEBrick als Webserver für Development

Rails: Request & Response





Routing

- Definieren von Routen `resources :users`

HTTP Verb	Path	Action
GET	/users	index
GET	/users/new	new
POST	/users	create
GET	/users/:id	show
GET	/users/:id/edit	edit
PUT	/users/:id	update
DELETE	/users/:id	destroy

ActiveRecord (1)

- Abbildung von Objekten und deren Attribute in einer Datenbank
- Hinzufügen, Entfernen und Ändern von Objekten wird direkt in der Datenbank ausgeführt.
- SQLite, MySQL, PostgreSQL, Microsoft SQL Server
- 3 Environments (Development, Production, Test)
- Development / Test: Sqlite
- Production: PostgreSQL

ActiveRecord (2)

- Bezeihungen
  - `belongs_to`
  - `has_many`
- Validierung Attribute
  - `Presence`
  - `unique`

ActiveRecord (3) Migrations

- Komfortable Möglichkeit Datenbank organisiert zu ändern.
- Migrationen halten bestimmte Änderungen in der DB fest.
- Migration: `rake db:migrate`
- Rollback: `rake db:rollback`

Scaffolding

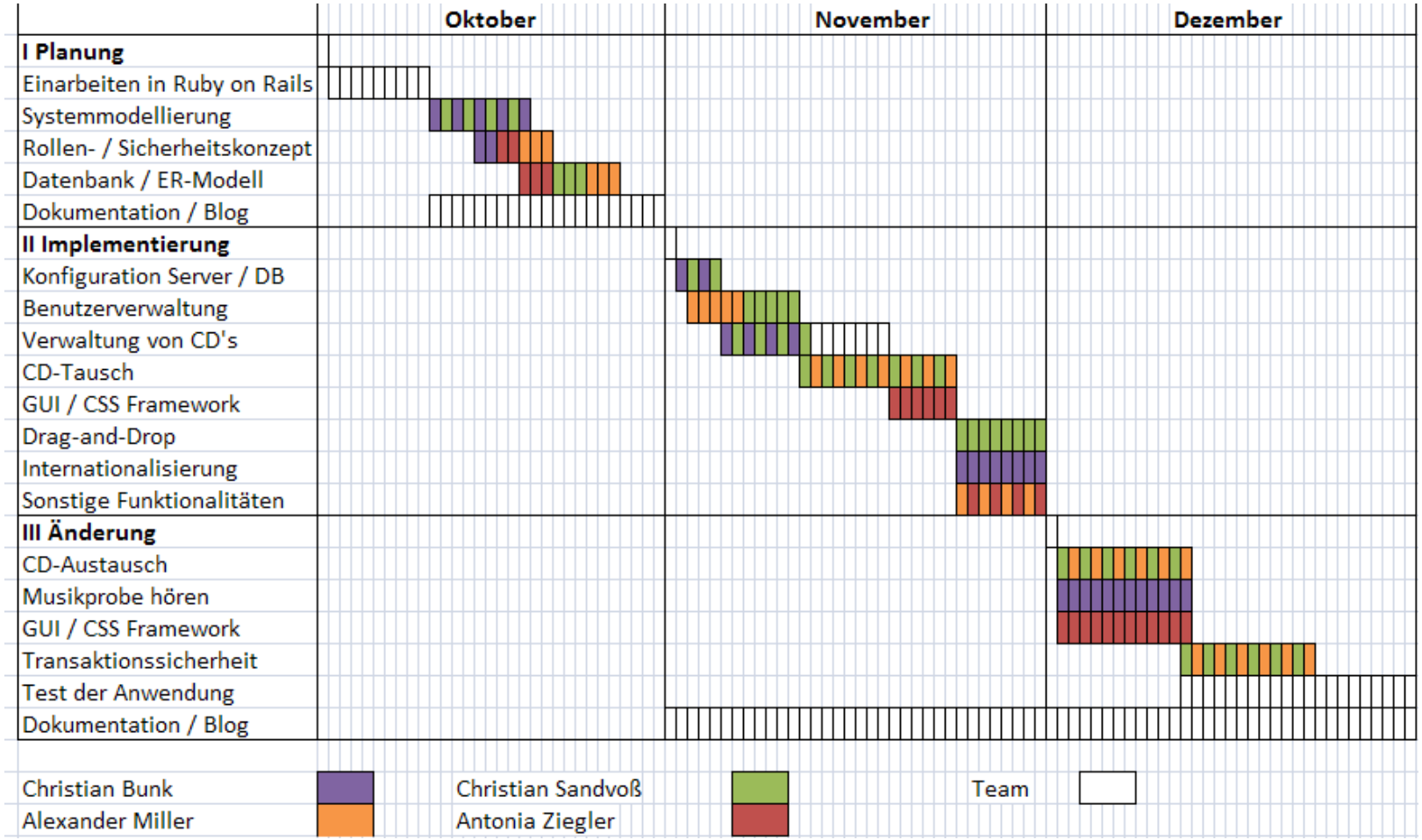
- Auf Knopfdruck Controller Model und Views erstellen.
- Gut für prototypische Entwicklung.
- Das haben wir NICHT gemacht!

PROJEKTMANAGEMENT

Projektmanagement

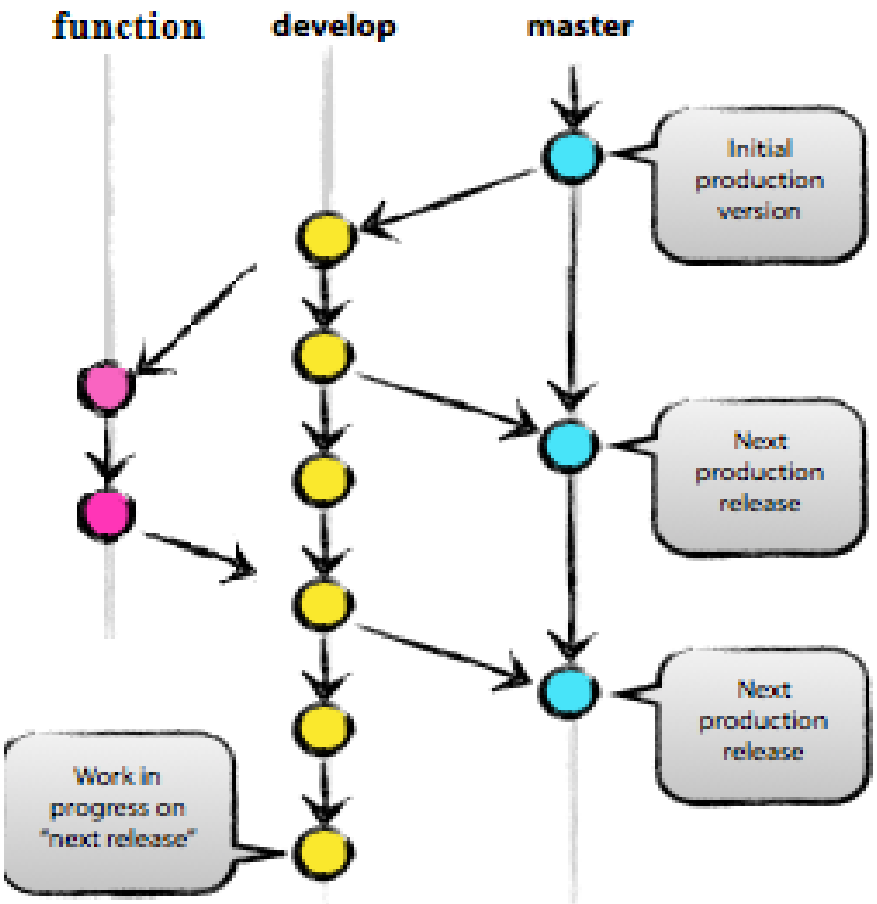
- Treffen 2 mal wöchentlich
- Bericht über Fortschritt und Probleme
- Austausch von Ideen
- Meilensteinplanung
- Server
- User-Stories für jede Funktionalität
- Tickets für jede User-Story
- Versionsverwaltung von Quellcode

Meilensteinplanung



Versionsverwaltung von Quellcode

- GitHub-Repository
  - Master
  - Develop
  - Transaction
  - …
- Deployment
  - Apache Webserver
  - Ruby 1.9.2 und Rails 3.1.1
  - PostGreSQL
  - Capistrano
  - http://kallisto.f4.htw-berlin.de



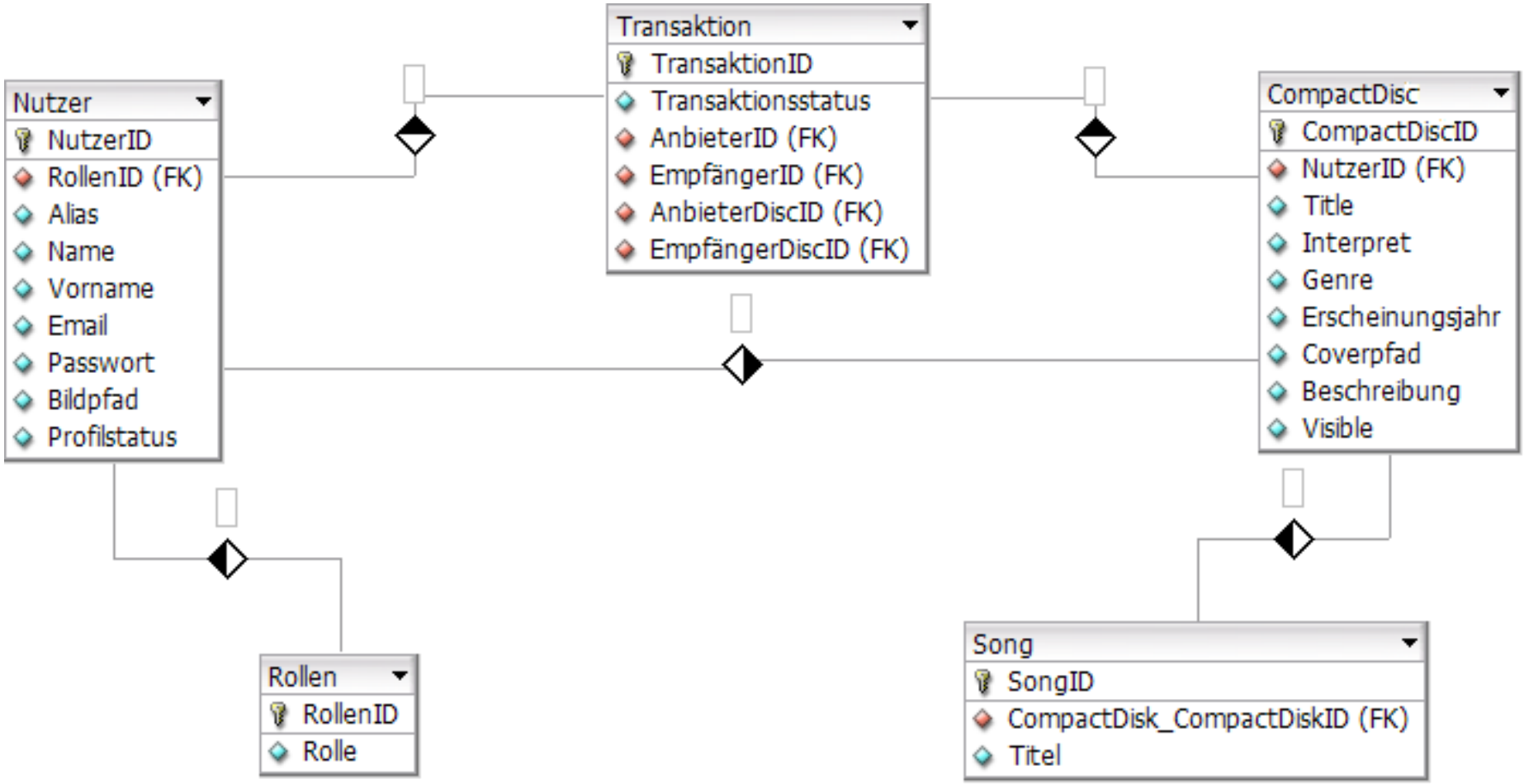
http://nvie.com/posts/a-successful-git-branching-model/

IMPLEMENTIERUNG

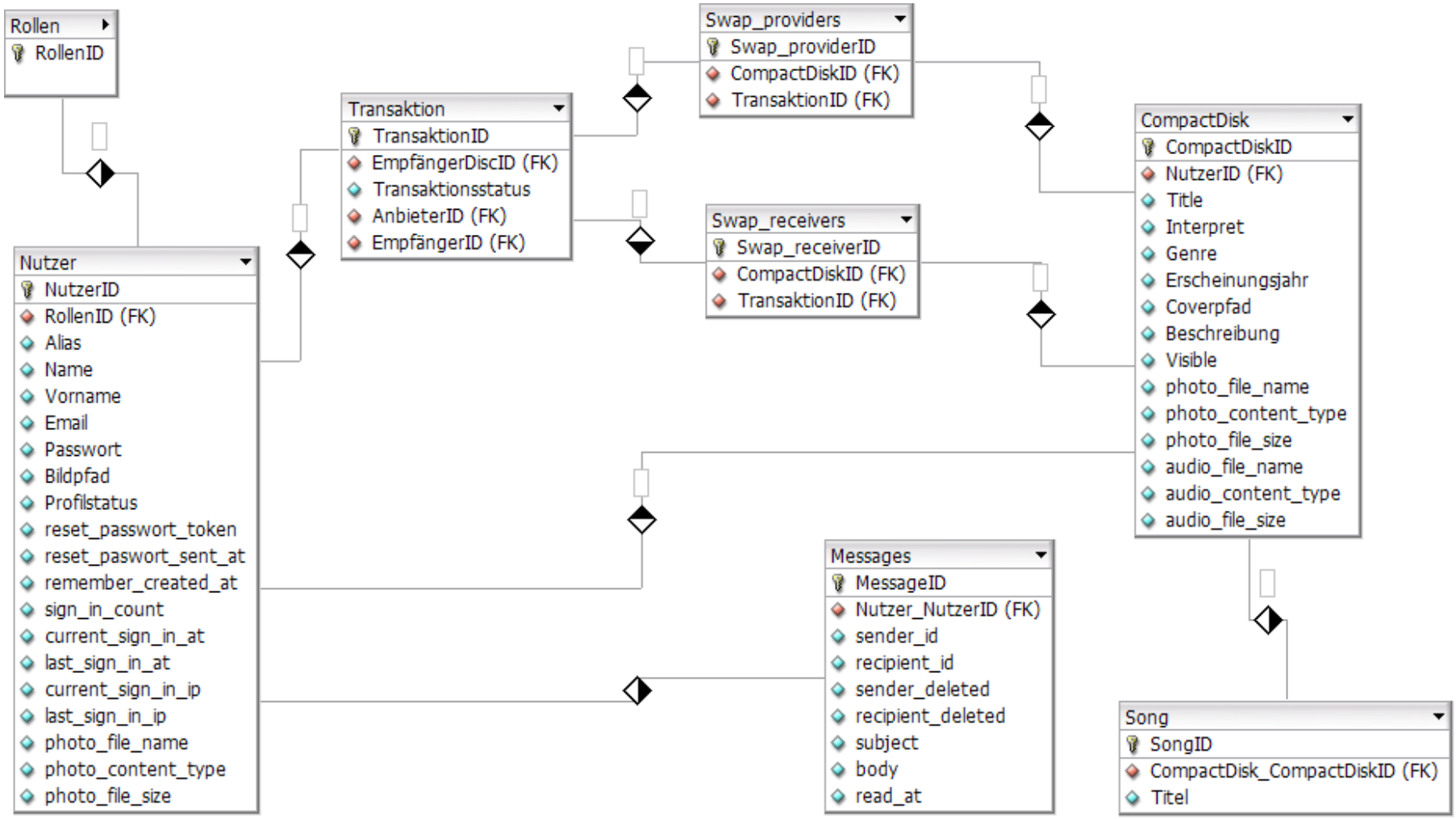
Herangehensweise

1. Erstellen des Models
2. Validierung des Models mit Unit Tests
3. Funktionalität und View

Konzeption der Datenbasis



Änderungen im Datenbankschema



Benutzerverwaltung

- Erst eigene Implementierungen, dabei festgestellt
  → nur Grundgerüst der Autorisierung-Funktionalität
  → mehrere Sicherheitsrisiken
- sicherer und gut dokumentierter Plug-In
- Plug-Ins evaluiert
  → Devise

Devise- Plugin

- Rails keine Komponenten zur Authentifizierung
- flexible Lösung zur Benutzerauthentifizierung für Rails
- besteht aus 12 Modulen, nach MVC Prinzip implementiert und nach Bedarf kombiniert

- Devise NUR Benutzerauthentifizierung

Rollenverwaltung

- Rollenmanagement zu realisieren,
  - →Nutzern mit unterschiedliche Rechten
  - →Dadurch die Anwendung zu schützen
- Definition von
  - mehreren Gruppen (Gast, Benutzer, Admin)
  - Jeder angemeldete Nutzer wird automatisch der Gruppe „Nutzer“ zugewiesen

Transaktionen (1)

- Sollte durch Messaging-Systems realisiert werden
- Gem für die interne Kommunikationbzw. den Nachrichtenaustauch
  - simple-private-message

Transaktionen (2)

- Sobald Anfrage auf einer CD existiert, wird diese für andere Nutzer gesperrt
- Da durch Transaktionssicherheit
  - nur CDs die nicht in einer Transaktion sind können getauscht werden

Transaktionen (3)

1. Mittels Javascript / jQuery IDs der CDs auslesen
2. Daten werden an Controller geschickt, der dann Nachricht generiert und verschickt
3. Empfänger erhält Nachricht und hat verschiedene Möglichkeiten
4. Beim Modifizieren (wieder zu 1.)
5. Annehmen oder Ablehnen

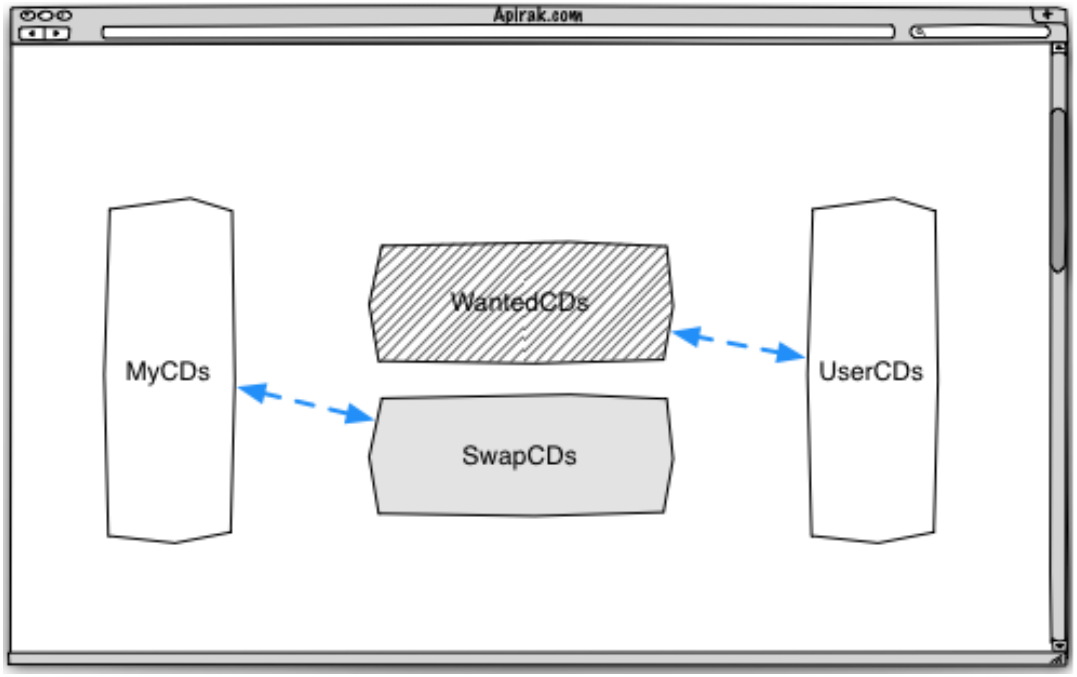
- Bei Annehmen: Tauschen (User_IDs in CDs tauschen)
- Daten müssen geparsed und gesplitted werden, da sie als String vorliegen (z.B. CD IDs)

Transaktionen (4)

- Anfrage
  - Subject: „4,7;32,8“
  - Body: „Anfrage“
- Modifikation
  - Subject: „4,7 ; 32,8 ; 3,44 ; 3 “
  - Body: „Modifikation“

Drag and Drop

- Einleiten eines Tausches/Transaktion durch Drag and Drop
- Funktionalität wurde durch jQuery realisiert



Graphic User Interface

- Erste Entwürfe in analoger Form
- Mokups mittels Adobe Photoshop
- Umsetzung mittels HTML5 und CSS
- Nach Woche 7 und ersten Test wurde das Layout und Design als unzureichend deklariert
  → Wert auf Usability
  →Evaluierung von Plug-ins, Templates und Frameworks

→Bootstrap von Twitter

Bootstrap (Twitter)

- Bootstrap ist ein spezielles Toolkit, das im Wesentlichen ein HTML- und CSS-Template bereitstellt, in das Twitter seine Erfahrungen einfließen ließ
  - praxiserprobte Frontend-Entwurfsmuster
  - Grid-System
  - Styling für typographische Elemente
  - Formulare und Buttons Designs
  - viele Lösungsansätze zur Realisierung von Javascript Effekten → Tooltips und Popover

Usability

- Einheitlichkeit
- Wiedererkennung
- Einfacheit
- Intuitive Bedienung

VERÖFFENTLICHUNG

Deployment

- Automatisiertes Deployment mit Capistrano
- Deployment mit einem Kommando
  - Auscheken der Daten
  - Ausführen von Migrationen
  - Ausführen der Tests
  - Symlink setzen
  - Server neu starten

Serverkonfiguration

- Webserver Apache 2.2
- Datenbank PostgreSQL

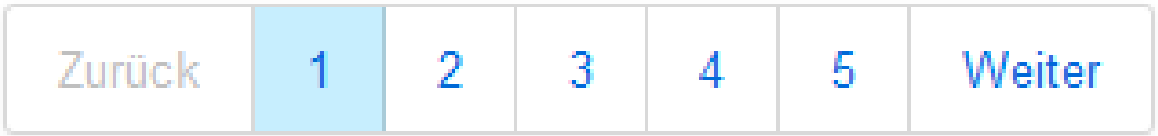
EXTRAS

Seitennavigation

- CDs können mehrere hundert sein
  - →Diese sollen nicht alle auf einer Seite angezeigt werden

→*WillPaginate –Plug-in*

  - →Begrenzt die Anzahl der CDs auf einer Seite und bietet eine Seitennavigation



MusicBrainz

- Plugin rBrainz stellt API für Anfragen
- Eingabe des Album und Interpret
- Auslesen von Erscheinungsjahr und Tracks
- Generierung einer Cover-URL
- Amazon Standard IdentificationNumber (ASIN)

```
http://images.amazon.com/images/P/<ASIN>.jpg
```

File Upload

- Upload von Bild und Audio
- Plugin Paperclip
- Einfache Konfiguration im Model:
- Speicherung im Dateisystem

```
has_attached_file :photo, :styles => { :normal => "150x150>", :small => "70x70>" },
                  :url  => "/system/users/:id/:style/:basename.:extension",
                  :path => ":rails_root/public/system/users/:id/:style/:basename.:extension",
                  :default_url => "/assets/user_default_:style.png"

validates_attachment_size :photo, :less_than => 5.megabytes
validates_attachment_content_type :photo, :content_type => ['image/jpeg', 'image/png']
```

Unterstützung von Audio im Browser

Brower	Ogg Vorbis	MP3
FireFox	✓	
Safari		✓
Chrome	✓	✓
Opera	✓	
Internet Explorer		✓

- Systemtool ffmpeg
- Konvertierung nach .ogg und .mp3

Background Jobs

- PluginResque
- Erstellen von Background Jobs
- Benötigt für Audiokonvertierung und Versand von Emails
- Abarbeiten von Jobs in einer Queue
  - Konvertierung der Musik
  - Versenden der eMails
- Läuft als Dämon
- Über Webinterface aufrufbar

Internationalisierung

- jede gute Webanwendung mehrere Sprachen unterstützen sollte, um möglichst viele Nutzer zu erreichen
- in Rails über eine *localize*Datei
- In den Views werden nun Variablen für die Bezeichnungen eingetragen

Tests

Features

- Suche nach Benutzern (Name, Vorname, eMail, Alias) und CDs (Title, Künstler, Songs)
- Autocomplete der Suche
- Doppelte Einträge werden in Suchfeld entfernt
- Benutzer kann entscheiden ob eigene CDs in der Suche aufgelistet werden
- Ansichten
  - Alle CDs
  - Neueste CDs (zeigt die 10 letzten CDs an)
  - Tollste CDs
  - Aktivste Benutzer (zeigt die 10 Aktivsten Benutzer)

Features

- Upload von Cover via URL
- Nutzen von MusicBrainz für Eintragung von Songs, Erscheinungsjahr und Cover-URL
- Eintragen von CD-Songs
- Konvertierung von DemoSongs in .oggund .mp3 (im Hintergrund)
- Benachrichtigung via eMail über neue Tauschangebote
- Abschalten von Benachrichtungen
- Benutzer kann alle 2 Stunden eine CD liken
- Tooltips zu jeder CD (Popovers)

ZUSAMMENFASSUNG

RésuméPlugins

- Plugins sinnvoll?
- Meistens ja, da ausgereift und sicherer
- Nein, wenn Funktionalität systemkritisch ist
  - Mehr Funktionalität
  - Updates
- Teilweise sperrig, wenn eigene Funktionalität eingebracht werden soll
- Plugin muss der gewünschten Funktionalität entsprechen
- Gute Dokumentation vorausgesetzt

RésuméRails

- Gut für prototypische Entwicklung
- Gut für größere Projekte

Danke für Ihre Aufmerksamkeit
Fragen?

Christian Bunk
Alexander Miller
Christian Sandvoß
Antonia Ziegler