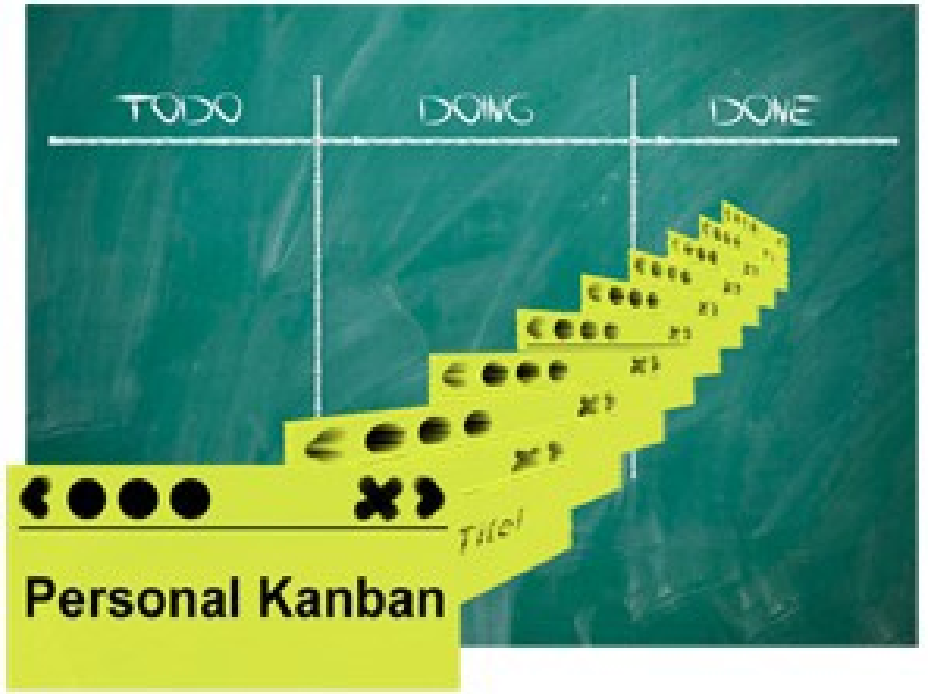
Personal Kanban



BE THE PUPPET MASTER

Inhaltsverzeichnis

1 Gegenstand und Grund des Projektauftrages

2 Ist-Analyse

2.1 Problem

2.2 Derzeitige Lösungen

3 Soll-Konzept

3.1 Lösungsansatz

3.2 Architektur

3.2.1 Client

3.2.2 Server

Inhaltsverzeichnis

4 Implementierung

4.1 Vorbereitung

4.2 Kommunikation zwischen Server & Client

4.3 Server

4.4 Client

5 Fazit

6 Quellen

1 Gegenstand und Grund des Projektauftrages

- Erstellen einer Webapplikation zur Verbesserung des Zeitmanagements
  - **Personal Kanban**
  - Umsetzung auf Grundlage Kanban
- Gründe
  - Selbst Probleme im Zeitmanagement
  - Erfahrungen mit Kanban im betrieblichen Umfeld

2 Ist-Analyse

2.1 Problem

- Pool von Aufgaben, Terminen, Meetings usw.
- Mehrere Personen beteiligt
  - Komplexität des Organisationsaufwandes wächst stark
    - → Übersicht leidet, Wichtiges wird vergessen
    - → hoher Zeit-/Kostenaufwand → Frust
- Unser Ziel: Komplexität auflösen durch Kanban-Steuerung

2.2 Derzeitige Lösungen

- To-Do-Liste
  - Aufgabenliste die abgehakt wird
- Getting Things Done
  - Kontextbezogene Aufgabenliste
  - Aufgaben, Termine und Projekte
- Kanban Zeitmanagement
  - Boards mit Spalten und Tickets
  - WIP und Priorisierung

3 Soll-Konzept

3.1 Lösungsansatz

- Wahl des Kanban-Zeitmanagement-Systems
  - Visualisierung
  - Begrenzung der Menge angefangener Arbeiten
- Board- und Gruppenmanagement
- Viele wichtige Infos auf einen Blick
- Statistische Auswertungen für Messbarkeit

3.2 Architektur

3.2.1 Client

- MVC-Architektur → Emberjs-Framework
  - Seperation of Concerns
- Model
  - EmberDataModels: JavaScript Objekte
    - Widerspiegelung unserer Datenbanktabellen
  - EmberDataStorage: Puffern und Persistieren der Model-Daten
    - DataStorageAdapter

3.2.1 Client

- View
  - Template-Engine: Handlebars
    - Grundgerüst und Komponenten
    - Zugriff auf Model-Daten und Controllerfunktionen
    - Nachteil: Alle Templates in einer Datei → Grunt
  - Design: Less
    - Erweiterung CSS Standard
    - Vermindert Redundanzen (DRY)

3.2.1 Client

- Controller
  - EmberRouter
    - Bestimmung Pfad und URL-Paramter
    - Zuordnung der Anfrage zu richtigen EmberController, EmberDataModel und Handlebars-Template
  - EmberController
    - Zusätzliche Funktionen für Templates (z.B. Filterung)
    - Formulare auswerten
    - XMLHttpRequest an Server schicken

3.2.2 Server

- Backend-Solution-Stack: WAMP
- Hauptaufgaben
  - Requests behandeln
  - Schnittstelle zur Datenbank
  - Response an Client geben
- Abstrakter und modularer Aufbau
  - Erweiterbarkeit und Anpassungsfähigkeit

3.2.2 Server

- Datenbankabstraktion (ORM)
  - ActiveRecord-Framework
- Nutzung verschiedener Designprinzipien
  - Separation of Concerns, Single-Responsibility-Principle, Dependency Injection
- Umsetzung durch Design Pattern
  - Factory Pattern
  - Observer-Subject Pattern

4 Implementierung

4.1 Vorbereitung

- Zentrales Versionsverwaltungssystem (git)
  - Vorteil: Arbeit protokolliert und archiviert
- Gemeinsame Entwicklungsumgebung
  - Codingstandards
- Aufgabenteilung anhand geplanter Features
  - Featurebranches
  - Github Issuetrackingsystem

4.2 Kommunikation zwischen Server & Client

- REST Paradigma
  - Adressierbarkeit
  - Unterschiedliche Repräsentationen
  - Zustandslosigkeit (Protokoll)
  - Vorgesehene Operationen (HTTP-Verb)
    - PUT, GET, POST, DELETE
- Benutzen des Clientseitig integrierten RESTAdapter

4.2 Kommunikation zwischen Server & Client

- Im RestAdapter werden Actions bereitgestellt die REST-Operationen repräsentieren

Action	HTTP Verb	URL
Find	GET	/people/123
Find All	GET	/people
Update	PUT	/people/123
Create	POST	/people
Delete	DELETE	/people/123

4.3 Server

- Aufsetzen der Datenbank
  - Kein zentraler Server → Schemas versionieren
  - Problem: Credentials nicht quelloffen versionieren
- Einbinden des ActiveRecord-Frameworks
- REST-API implementieren
  - .htaccess anpassen

4.3 Server

- RequestHandler
  - Verarbeitung und Aufbereitung der Request-Parameter
- ModelFactory
  - Datenbankverbindung initialisieren
  - Anhand des injizierten Modelnamens, Modelklasse dynamisch instantiieren & REST-Operationen ausführen

4.3 Server

- Modelklassen
  - Jede Datenbanktabelle benötigt repräsentative Klasse
  - Queries ausführen
- ResponseFactory
  - Rückgabe des Ergebnisses an Client als JSON-String
  - Mögliche Errors aufbereiten

4.4 Client

- Personal Canban Route
  - Laden der Tickets und Boards wenn User eingeloggt ist
  - Weiterleitung zum Mainboard
  - Ist der User nicht eingeloggt, wird er zum Login weitergeleitet

4.4 Client

- Login
  - On-the-fly Eingabevalidation
  - Bestanden: einloggen und aufrufen der PersonalCanbanRoute
  - Nicht Bestanden: Fehleranzeige
- Registrierung
  - POST-Request um User anzulegen und Weiterleitung zum Login (5s delay)

4.3 Client

- Boardansicht
  - Variable Route (URL)
  - Einstellungsmöglichkeiten für das ausgewählte Board
    - Infos anzeigen
    - Einstellungen ändern
    - Neue Tickets erstellen

4.4 Client

- Ticketkomponente
  - Darstellung & Verarbeitung von Tickets
  - Ansichten
    - Basic
    - Details
    - Edit
    - Delete

5 Fazit

- SOLL-IST-Vergleich
  - Die meisten Features konnten umgesetzt werden
    - Terminierte Tickets hervorheben, statistische Auswertungen und Nachverfolgung nicht geschafft
  - Geplante Designprinzipien eingehalten
  - Modularer Aufbau
    - Erweiterbarkeit und Anpassbarkeit gegeben

5 Fazit

- Ausblick
  - Fehlende Features implementieren
  - Refactoring nach SEO-Gesichtspunkten
- Schlusswort
  - Viel gelernt, aber noch viel Potential im Projekt
  - Web-Applikationen zukunftsweisend
  - Zukünftige Projekte → test-driven development

Hands-On

6 Quellen

- http://www.worldpropertychannel.com/news-assets/Architect-Report.jpg
- http://devangelist.de/wp-content/uploads/2013/04/single-page-application.jpg
- http://nextgenerationtrust.com/wp-content/uploads/Diagram-and-pen-iStock_000013941242Medium1.jpg
- http://4.bp.blogspot.com/-6ptLDX7jTxU/UdQcETLQkmI/AAAAAAAAELE/26L3cY03tWw/s400/goal_puzzle_0.jpg
- http://goshenchurch.com/wp-content/uploads/2012/02/Hands-On.jpg
- http://emberjs.com/guides/models/the-rest-adapter/