

# CAS FEE 2017 HERBST INFORMATIONEN ZU PROJEKTEN PRÜFUNG, NOTEN

11. Sept. 2017

Markus Stolze [mstolze@hsr.ch](mailto:mstolze@hsr.ch)

Silvan Gehrig [sgehrig@hsr.ch](mailto:sgehrig@hsr.ch)



**HSR**

HOCHSCHULE FÜR TECHNIK  
RAPPERSWIL

FHO Fachhochschule Ostschweiz

# PROJEKT INFORMATIONEN

# Unterstützung und Hilfe beim Projekt

## ■ Face2Face Unterstützung bei Projekten / Übungen

- Jeden Mittwoch um 16.00. Raum 1.213 / TBD
- Voranmeldung an [sgehrig@hsr.ch](mailto:sgehrig@hsr.ch)
  - Bei grösseren Problemen

## ■ Fragen per E-Mail an [sgehrig@hsr.ch](mailto:sgehrig@hsr.ch)

- Jederzeit

# PROJEKT 1

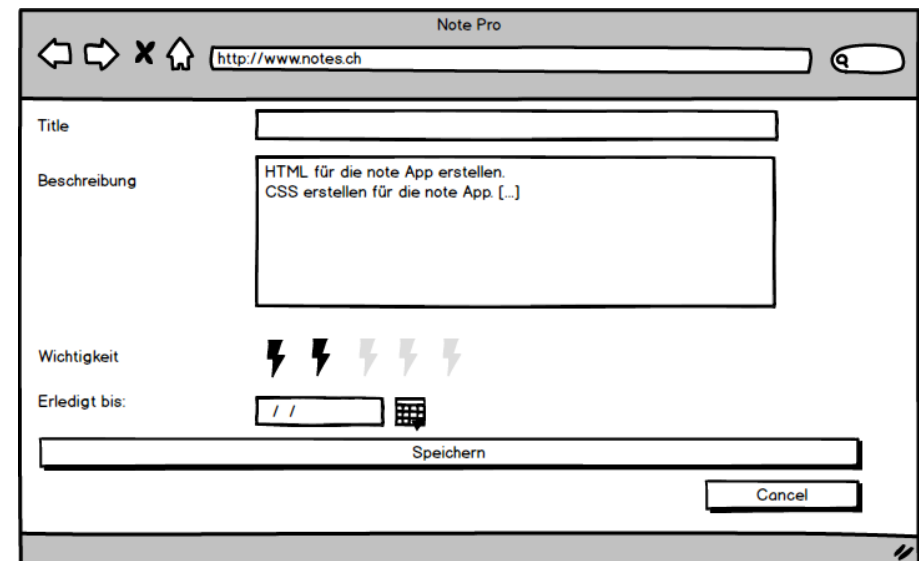
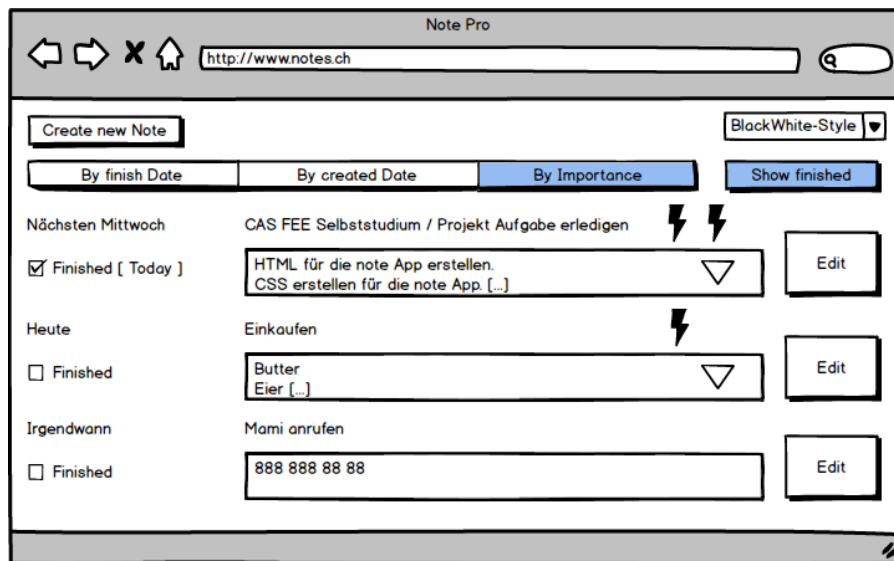
# Projekt 1: Notizen Verwalten

## ■ Ziel von Projekt 1

- Grundlagen festigen von CSS/JS/HTML

## ■ Aufgabe

- Erstellen einer Notizenapplikation



# Projekt 1

- **Startet heute.**  
**Aktion heute: Eintrag in Einschreibeliste (Name und GitHub URL)**  
<https://docs.google.com/spreadsheets/d/1Udio-8XKxlzmUcDp-g3agcKUrGaYM3U5H-OLdhThX6c/edit?usp=sharing>
- **Bewertung: Pass/Fail (must pass)**
- **Arbeit alleine, aber öffentlich ([Github Account](#) melden -> [Liste im Moodle](#))**
- **Abgabetermin 16.11.2017 – Mitternacht (Abend vor Angular Ganztage)**
  - Branch erstellen mit dem Namen «Abgabe»
  - E-Mail erstellen mit folgendem Inhalt:
    - Subject: [CAS FEE] Abgabe {{Gruppennummer}}
    - Link zum Branch
    - Ein ReadMe auf GitHub, falls dieses notwendig ist.
- **Bei anderen Projekten nachzuschauen ist erlaubt, die Lösung muss aber eine Eigenleistung sein**
- **Erfolgreiche Abgabe von Projekt1 ist Voraussetzung zum Start von Projekt 2**

# Projekt 1 – Einschränkungen

- Das Projekt sollte gut strukturiert sein aber weitgehend auf die Nutzung von Frameworks verzichten
  - Nutzung eines SPA Frameworks (Angular oder React+...) **nicht** erlaubt
    - Nutzung des MVC Patterns trotzdem sinnvoll
  - Nutzung von Layout / Styling-Framework wie Bootstrap nicht erlaubt
    - Flexbox und Grid nutzen
  - Nutzung von simpler Templating Engine (z.B. Handlebars) ist verlangt.
  - Nutzung von jQuery ist erlaubt
- Ältere Browser müssen nicht unterstützt werden.



# Projekt 1 – «Bewertung»

- **Das Projekt 1 wird in 5 Kategorien angeschaut. Jede Kategorie muss erfüllt sein.**
  - Funktionsumfang
  - Architektur
  - JS Qualität
  - CSS Qualität
  - HTML Qualität
- **Es wird ein generelles Feedback über alle Projekte geben.**
- **Bei größeren Problem in einem Projekt erhält man 2 Wochen Zeit um diese zu beheben.**



# Projekt 1 – «Bewertung»

## ■ Funktionsumfang

- Der Funktionsumfang ist in den Wire-Frames dargestellt. Diese beinhalten u.a.
  - Anzeigen, editieren und erfassen von Notizen
  - Sortieren von Notizen
  - Filtern von „abgeschlossenen“ Notizen
  - Abspeichern der Daten auf dem Server
  - Wechseln des Styles

## ■ Allgemeine Code Guidelines

- Unterstützung modernen Browsers & Features (Ältere Browser können vernachlässigt werden)
  - CSS3+, HTML5+, ES6+
- Saubere Trennung von Struktur (HTML/HBS), Logik (JS) und Darstellung (CSS)
- Sauberen Code
  - DRY: Kein Copy-Paste-Code (auch keine 'ähnlicher Code'!)
  - Keine langen Methoden
  - Sprechende, konsistente Benennung von Variablen
- Kein CSS / JS im HTML
- Übersichtliche Projekt-Struktur

# Projekt 1 – «Bewertung»

## ■ Architektur

- REST: Server und Client kommunizieren über JSON bzw. **x-www-form-urlencoded**
- Client:
  - Server-Calls nur im Service Layer
  - Routing/Event-Handling nur im Controller
  - Rendering/DOM Manipulation nur in der View
  - View und Controller können im gleichem File definiert werden.
  - Kein HTML Zusammenbasteln aus String => Handlebars verwenden
- Server
  - Memory-Storage- / DB-Zugriffe nur im Service
  - Controller stellt Actions/Request-Handlers zur Verfügung
  - Router: Verknüpfung von Routen und Actions/Request-handlers

# Projekt 1 – «Bewertung»

## ■ JavaScript

- Nutzung einer Template Engine auf dem Client (z.B. Handlebars)
- Keine Console Pollution
- Kein auskommentierter Code
- Kein «global namespace pollution»

## ■ CSS

- FlexBox
  - keine Float-, Table- oder Inline Layouts
  - keine unnötigen Klassen & ID's (Elemente über Struktur/Name selektieren)
- Inline Styles & Inline-Style-Klassen sind nicht erlaubt

## ■ HTML

- Korrekter Einsatz von Semantischem HTML
- Sinnvolle HTML Validation

# Projekt 1 – Ablauf

Woche	Aufgabe-Grob	Details
1	Projekt im Github eintragen.	
2	HTML Gerüst erstellen für die WireFrames inkl. CSS.	Ändern der Wireframes ist erlaubt. Flex-layout im Projekt ausprobieren.
3	«Create New Note» auf der Detail-Seite implementieren. Die Daten in den «LocalStorage» einpflegen. Navigation zwischen den beiden HTML-Seiten. Auf der Hauptseite die Anzahl darstellen.  Style Switcher implementieren.	Einen zweiten sinnvollen Style ausdenken.
4	Hauptseite ausprogrammieren: Anzeigen der Einträge / Filtern / Sortieren Handlebars verwenden für das Rendern der Einträge.  Detailseite ausprogrammieren: Erfassen / Editieren	Flex-layout im Projekt einsetzen.
5	JavaScript optimieren. Patterns anwenden. Nutzen von Klassen für die Datenhaltung.	(Revealing) Module Pattern für die "Datenklassen" erstellen. IIFE anwenden.
6	Client Modularisierung fortführen. Node-Module erstellen zum Verwalten der Daten auf dem Server.	Bonus: Neue Einträge sollen auf andern Browser sichtbar werden. z.B. durch Polling.
7	Die REST API vom Server implementieren. Diese im Client anbinden.	
8+9	Finalisieren & Abgabe	

**CODEWARS**

# Pilot 2017: Codewars

The screenshot shows the Codewars dashboard. At the top, there's a navigation bar with the Codewars logo, a star icon, a bell icon, a user profile picture, and a rank badge showing '6 kyu' and '136' points. Below the navigation bar, the main content area is divided into several sections. On the left, there's a sidebar with icons for 'Your Next Challenge...', 'JavaScript', 'Rank Up', and a document icon. The 'Your Next Challenge...' section shows a challenge titled 'Perimeter of squares in a rectangle' with a '5 kyu' difficulty. The challenge description says: 'The drawing shows 6 squares the sides of which have a length of 1, 1, 2, to see that the sum of the perimeters of these squares is:  $4 * (1 + 1 + 8) = 4 * 20 = 80$ '. Below this, it asks: 'Could you give the sum of the perimeters of all the squares in a rectangle n + 1 squares disposed in the same manner as in the drawing:'. There are buttons for 'TRAIN' and 'SKIP'. Below the challenge, there's a section for 'Upgrade To Codewars Red!' with a 'LEARN ABOUT RED' button. At the bottom, there's a section for 'Allies' with a table showing the user's position, name, rank, and clan.

Position	User	Clan
1	6 kyu mstolze	Hochschule für Technik Rapperswil

## Kata

In our dojo, kata are real code challenges focused on improving skill and technique. Some train programming fundamentals, while others focus on complex problem solving. Each kata is crafted for and by the community.

## Kyu/Dan Ranks

Each kata on the site is set to a Kyu/Dan rank, based on its subject area and difficulty. The community collectively determines rank in the Beta Process. Every new Codewarrior (user) on Codewars starts being ranked as 8 Kyu.

What are Kyu and Dan? The terms are borrowed from a system in Japanese martial arts, which is in turn borrowed from the game of Go. Kyu indicates the number of degrees away from master level (Dan). This is why they count downward. Once you reach master level, we count upward.

Currently Dan level content is reserved for later. You will only find Kyu level content to train on for now.

## Earning Ranks

You can advance through the ranks by completing kata at or above your rank - the harder the kata the faster you advance.

# Codewars

- **Anmelden mit HSR oder eigener E-Mail**
  - Nachteil HSR: geht verloren nach Abschluss
  - Vorteil HSR: Automatisch(?) im HSR Clan
- **Account Info Abgabe mit Projekt1 Info Sheet**
- **Zum Start von Projekt 2 muss Level 7kyu erreicht sein**
- **Zum Abschluss von Projekt 2 muss Level 6kyu erreicht sein**
- **Unregelmässig: Leaderboard & Erfahrungsaustausch**



# PROJEKT 2

## Projekt 2: Single Page Applikation

- **Erfolgreiche Abgabe von Projekt1 ist Voraussetzung zum Start von Projekt2**
- **Erreichen von CodeWars Level 7kyu ist Voraussetzung zum Start von Projekt2**
- **Projekt 2 wird in 2er Gruppen bearbeitet. 3er Gruppe ist möglich, keine Einzelarbeit -> Prüfungszulassung**
- **Ziel von Projekt 2**
  - Vertiefen von Advanced Themen
  - Erstellen einer Single Page Anwendung unter Nutzung eines aktuellen SPA Frameworks (Angular oder React+Redux)

### Themen

- **Freie Projektwahl mit Mindestanforderungen:**
  - User-Management
  - Server mit Daten (REST-Routes)
  - Die Themenwahl muss von einem Betreuer bestätigt werden
  - Umsetzung entweder mit Angular oder React+Redux
- **Beispiel-Themen:**  
Reddit (Einfach), Webshop, Buchhaltungssystem, Online-Schach (Schwer)

## Projekt 2

### ■ Ideen, Themen und Gruppenfindung ab: 17.11 ("UX Tag")

- Repos anderer Mitglieder anschauen: Stil, Skill-Level,
- Diskutieren: Zieltechnologien, Projekt-Ideen,  
Geplante Herausforderung: Halbmarathon, Marathon, Iron-Man, ...

### ■ Abgabe 22. April (Sonntag nach Präsentation/Prüfung)

### ■ Präsentation: 17. April 2018

### ■ Weitere Informationen am 17.11 ("UX Tag")

- z.B. Zeldman on Twitter: "Lists is a gallery of real content to populate your mockups."  
→ <https://twitter.com/zeldman/status/890965212790222852>

# PRÜFUNG & NOTEN

# Zulassung zur Prüfung

## ■ Zugelassen zur Prüfung ist:

- Wer 80% oder mehr des Unterrichts besucht hat
- Wer CodeWars Level 6kyu erreicht hat
- Wer Projekt 1 erfolgreich abgegeben hat (Bewertung "pass")
- Projekt 2 im Team erfolgreich abgegeben hat (Note 4 oder besser)

# Prüfung

- **Datum: 17. April**
- **Start 15:30**
- **Closed Book, keine Computer**
- **1 Stunde**
- **Besteht aus Multiple Choice und «1 Satz Antwort-Fragen»**
  - Fragen, welche man als Front-End-Engineer in einem Bewertungsgespräch beantworten können sollte.

# Noten

## Notenverteilung

Prüfung	40%
Projekt 2	60%



# Reglement

- **Studienreglement und Prüfungsrichtlinien-2017 im Moodle verfügbar**