

## Miniprojekt WE 3

Ihre Miniprojekt-Aufgabe besteht darin, eine Finance-Webapplikation in Angular sowie React zu programmieren. Es soll möglich sein, **Geld-Transaktionen zwischen verschiedenen Konten** zu veranlassen.

Die Grundlagen bilden die Wireframes unten, welche den Funktionalitätsumfang zeigen. Zusätzlich definiert das Video „Mini-Projekt-WED3.mp4“ die dynamische Ansicht der Webseite mit weiteren, konkreten Anforderungen.

Ihre Aufgabe ist den kompletten Funktionsumfang der Wireframes zu implementieren und die im Video visualisierten Feinheiten zu berücksichtigen. Die jeweiligen Vorlagen sind von der Technologie (Angular / React) abhängig und werden in separaten Dokumenten im Übungsordner beschrieben. Die Server-Funktionalität wird durch eine Node.js Lösung zur Verfügung gestellt und liegt dieser Mini-Projekt Beschreibung bei. Die **Server API steht im Readme.md File** innerhalb des GitHub Repos unter <https://github.com/IFS-Web/HSR.WED3.Server> Files beschrieben.

Sie können am Aussehen der Finance-Webapplikation Anpassungen vornehmen. Beachten Sie, dass das Design im Mindesten **fluid** sein sollte. Sie dürfen externe Design-Libraries wie **Bootstrap / Material Design / Semantic UI / ...** einsetzen.

### Login / Registrieren

Finance Portal

http://finance/welcome

Home Registrieren

Willkommen

Benutzername:

Passwort:

Login

http://finance/ verweist auf http://finance/welcome.  
Registrier-Eingabe analog der Loginmaske.

### Dashboard

Finance Portal

http://finance/dashboard

Home Konto Bewegungen Logout Michael Gfeller

Neue Bewegung

Von 111111 [750 chf]

Zu

chf

Überweisen

Letzte Bewegungen

Von	Zu	Betrag	Kontostand Neu
111111	33333	-50	750
111111	22222	-100	800
44444	11111	+50	900

Alle Anzeigen

Das Zielkonto (Zu) wird überprüft und der Name des Empfängers wird angezeigt - falls Nummer stimmt. Ansonsten Fehler.  
Die letzten Bewegungen (rechts) müssen nach der erfolgreichen Transaktion updated werden.

## Konto Bewegungen

Finance Portal

Home Konto Bewegungen Logout Michael Gfeller

Konto Bewegungen

Filter 2017 Februar

Datum	Von	Zu	Betrag	Kontostand Neu
29.01.2017	111111	33333	-50	750
28.01.2017	111111	22222	-100	800
28.01.2017	44444	111111	+50	900

Filterung der Bewegungen nach Jahr und Monat.

## Durchführung

Die Arbeit ist in 2er Teams durchzuführen.  
Ausnahmen sind nur unter expliziter Bewilligung erlaubt.

Bitte registrieren Sie Ihre Gruppe bis spätestens 15. März im folgenden Dokument:

<https://docs.google.com/spreadsheets/d/1iSjsSqecY9H42afJIL7HGkdePKwOhaxyY6nkLKa7ZZQ/edit#gid=0>

## Review / Abgabe

Sie werden als Gruppe Ihre Lösung während der letzten Übungsstunde des Semesters den Übungsleitern präsentieren. Alle Gruppenteilnehmer müssen anwesend sein.  
Zusätzlich schicken Sie dem Übungsleiter Ihre Lösung als Zip-File per E-Mail (ohne den Ordner "node\_modules").

Das Review der Lösung findet in der Woche 14 (Thema „Praxisvorträge“) statt.

Für den Review-Termin können Sie sich im folgenden Dokument einschreiben:

<https://docs.google.com/spreadsheets/d/1iSjsSqecY9H42afJIL7HGkdePKwOhaxyY6nkLKa7ZZQ/edit#gid=1433436071>

## Notenabgabe

Am Ende des Reviews erhalten Sie ein Feedback über Ihr Projekt (genügend, oder ungenügend).

## Checkliste / Anforderungen

Folgende Punkte müssen erfüllt sein:

		Bemerkungen / geforderte Eigenschaften
<b>User Management</b>	Login	<ul style="list-style-type: none"> <li>• Validation vom Input (Benutzername / Passwort <b>required</b> und länger als <b>drei Zeichen</b>).</li> <li>• Login muss auf dem vom Server gelieferten <b>JWT Token</b> basieren.</li> <li>• Das JWT Token im Local-Storage ablegen.</li> <li>• Das erfolgreiche Login endet auf dem Dashboard.</li> </ul>
	Registrierung	<ul style="list-style-type: none"> <li>• Validation von Vornamen, Nachnamen, Benutzernamen / Passwort / analog Login.</li> <li>• Passwort Confirmation Client-Seitig überprüfen.</li> <li>• User wird auf dem Server angelegt und automatisch eingelogged.</li> </ul>
	Logout	<ul style="list-style-type: none"> <li>• Löscht das Client-seitig gespeicherte JWT Token.</li> <li>• Benutzer befindet sich nach Logout auf der Welcome-Page.</li> <li>• Das JWT Token aus dem Local-Storage löschen.</li> </ul>
<b>Transaction Management</b>	Zahlung auslösen	<ul style="list-style-type: none"> <li>• Neue Zahlungen auf dem Dashboard auslösen.</li> <li>• Gültige Zahlungsempfänger (Vorname / Nachname) automatisch vom Server abfragen.</li> <li>• Validation vom Input (nur Beträge &gt; 0.05 CHF erlaubt).</li> <li>• Vom Server bestätigte Transaktionen mittels einer „Transaction Erfolgreich“ -Info abschliessen.</li> </ul>
	Zahlungs-übersicht darstellen	<ul style="list-style-type: none"> <li>• Die letzten drei Transaktionen auf dem Dashboard darstellen.</li> <li>• Möglichkeit, auf die Ansicht mit sämtlichen Transaktionen zu wechseln.</li> <li>• Nach einer Transaktion soll diese in der Übersicht dargestellt werden.</li> </ul>
	Alle Zahlungen darstellen und filtern	<ul style="list-style-type: none"> <li>• Zahlungen nach Monaten filtern.</li> <li>• Filtermöglichkeit nach Monat des entsprechenden Jahres (die letzten drei Jahre auflisten).</li> <li>• Den Filter mithilfe der Server-API anwenden.</li> </ul>
<b>Einschränkungen</b>	Fluides Design	<ul style="list-style-type: none"> <li>• Die Cards auf dem Dashboard sollen im Verhalten fluid sein. Dazu können Libraries (siehe <i>Sonstiges</i>) verwendet werden.</li> </ul>
	Sonstiges	<ul style="list-style-type: none"> <li>• Der vorbereitete Server sowie die Vorlagen (Angular / React spezifisch) sollen verwendet werden.</li> <li>• CSS Libraries (z.B. Bootstrap) dürfen eingesetzt werden. Diese müssen sich aber auf reines CSS beschränken oder explizit in der Vorlesung erwähnt/behandelt werden.</li> <li>• Vermeiden Sie Copy-Paste Code. Verwenden Sie Komponenten um den UI Code zu reduzieren und Services für wiederverwendbare Programm-Logik.</li> </ul>