**Kleinprojekt: E-Banking App**



# Problemstellung

Du bist Programmierer im ARZ Innsbruck. Für einen neu zu gewinnenden Kunden soll eine E-Banking-App programmiert werden. Ein erster voll funktionsfähiger Prototyp mit reduzierter Basisfunktionalität soll den Kunden überzeugen, die Programmierung und das Hosting der App im ARZ durchführen zu lassen. Du und ein weiterer Mitarbeiter aus der Abteilung werden mit der Erstellung des Prototyps beauftragt, für den vom Abteilungsleiter in Absprache mit dem potentiellen Kunden einige funktionale Anforderungen und ein Basis-Szenario definiert wurden.

# Hauptszenarien

Kunden registrieren sich über die App per mobilem Device oder PC bei der Bank und erhalten ein Konto. Kunden können über ihr Konto Überweisungen auf Konten innerhalb derselben Bank tätigen. Kunden können am Bankschalter über einen Bankangestellten Beträge ein- und auszahlen. Kunden können ihre Kontobewegungen und Kontostände detailliert abfragen bzw. durchsuchen.

# Funktionale Anforderungen

Folgende funktionale Anforderungen gelten für das Projekt:

1. Kunden sollen sich selbst registrieren können.
2. Kunden und alle anderen Benutzer des Systems sollen sich einloggen können.
3. Nach einem Login sollen Benutzer solange eingeloggt bleiben, bis sie sich wieder abmelden oder aber den Browser schließen.
4. Benutzer haben verschiedene Rollen: Es gibt zwei Rollen: „Kunde“, „Angestellter“.
5. Angestellte werden vorab „händisch“ in die Datenbank eingetragen (es gibt also keinen Registrierungsprozess).
6. Kunden haben genau 1 Konto, das sie gleich bei der Registrierung bekommen. Für dieses Bankkonto werden folgende Informationen gespeichert: Kontostand, alle Kontobewegungen (Einzahlungen, Auszahlungen), IBAN, BIC und Verfüger (=Kunde).
7. Eingeloggte Kunden sollen ihre Kontodaten (Kontonummer, Kontostand, Eingänge und Ausgänge incl. allen Überweisungsdetails) abfragen können.
8. Eingeloggte Kunden sollen Überweisungen (=Kontobewegung) auf andere Kontonummern der gleichen Bank durchführen können. Überweisungen an Kontonummern außerhalb der Bank werden später implementiert. Eine Überweisung hat folgende Informationen: IBAN und BIC Absender, IBAN und BIC Empfänger, Zahlungsreferenz, Verwendungszweck, Betrag in Euro auf 2 Kommastellen, Datum, Uhrzeit.
9. Kunden sollen ihre Überweisungen nach Datum durchsuchen können (genau, von – bis).
10. Kunden sollen ihre Überweisungen nach Beträgen (genau, von – bis) durchsuchen können.
11. Kunden sollen ihre Überweisungen nach allen anderen Text-Informationen durchsuchen können.
12. Angestellte (Kassa) können im Namen von Kunden Beträge auf das Konto einzahlen und abheben (Barbehebung).
13. Kunden erhalten bei Einzahlung / Auszahlung über einen Bankbeamten (Kassa in der Bank) einen entsprechenden Beleg (ausgedruckt).

# Zusätzliche Anforderungen

1. Responsive Bootstrap-UI
2. Mehrschichtanwendung (UI, Modelklassen, ggf. zusätzliche Klassen für Businesslogik (Serviceklassen), DB-Zugriff mit Active-Record)
3. Vollständige Sourcecode Dokumentation
4. Codeversionierung mit Git und GitLab (Tools: <https://tortoisegit.org/> , <https://gitforwindows.org/>)
5. Scrum-Workmanagement mit GitLab (<https://about.gitlab.com/2018/03/05/gitlab-for-agile-software-development/>, <https://www.youtube.com/watch?v=CiolDtBIOA0>)
6. UI-Design in Adobe-XD (<https://www.youtube.com/watch?v=m-vji_2H3W8&t=814s>, <https://www.youtube.com/watch?v=s4SXQb6e0uM&t=1564s>)
7. Arbeit im Pair-Programming-Modus

# Liefervereinbarung

Abzugeben sind:

1. Projektplanung, Anforderungsanalyse und Systementwurf lt. Wordvorlage
2. Quellcode der Anwendung als GitLab-Repository-URL
3. SQL-Dump der Datenbank

# Zielumgebung

PHP-Mehrschichtanwendung incl. Bootstrap UI und MySQL-Datenbankanbindung.