Fachhochschule der Wirtschaft

-FHDW-

Paderborn

**Dokumentation**

Thema:

**Online-Banking**

Verfasser(in):

Alexander Heumann

Jannis Isaak

Sebastian Lühnen

Anwendungsentwicklung II

Eingereicht am:

13.12.2017

# Inhaltsverzeichnis

Inhaltsverzeichnis II

Abbildungsverzeichnis III

Tabellenverzeichnis IV

1 Fachkonzept 1

1.1 Fachliche Grundlagen 1

1.2 Use-Case-Diagramm 1

1.3 Benutzeroberflächenkonzept 2

1.4 Produktmodell 3

**1.4.1** **Funktionale Anforderungen** 3

**1.4.2** **Nichtfunktionale Anforderungen** 3

1.5 Qualitätsanforderungen 4

2 Fachkonzept 5

2.1 Benutzerlogin 5

2.2 Nutzer (App) 5

2.3 Verwaltung (App) 5

3 Technisches Konzept 6

3.1 Klassendiagramm 6

3.2 Datenbankmodell 6

4 Implementierung 7

4.1 Verwendete Programmiersprache 7

4.2 SQLite Datenbank 7

4.3 JDBC 7

4.4 Weitere Frameworks / Technologien die wir benutzen 7

4.5 Vorstellung des Praxisunternehmens 8

4.6 Ist-Analyse 8

4.7 Schwächenanalyse 8

4.8 Sollkonzept 8

4.9 << Umsetzung oder Implementierung >> 8

4.10 Kosten-Nutzen-Analyse 8

5 Test 9

# Abbildungsverzeichnis

Abbildung 1 Use-Case-Diagramm der Verwaltung 1

Abbildung 2 Use-Case-Diagramm des Nutzers 2

# Tabellenverzeichnis

[Tabelle 1 Quälitätsanforderungen 4](#_Toc500811338)

# Fachkonzept

## Fachliche Grundlagen

Diese Dokumentation wurde im Rahmen einer Gruppenarbeit für das Fach Anwendungsentwicklung I erstellt. Das zu entwickelnde Produkt ist eine Bankverbindung von einem Android System aus. Dazu soll eine App entwickelt werden, die sich auf einen Server verbindet und von welcher man Geld abheben und überweisen kann. Weiterhin soll es eine Webanwendung geben, bei welcher gesehen werden kann, welches Konto, welche Überweisungen und Abbuchungen getätigt wurden. Außerdem soll es möglich sein den Namen des Kontoführenden zu ändern.

## Use-Case-Diagramm

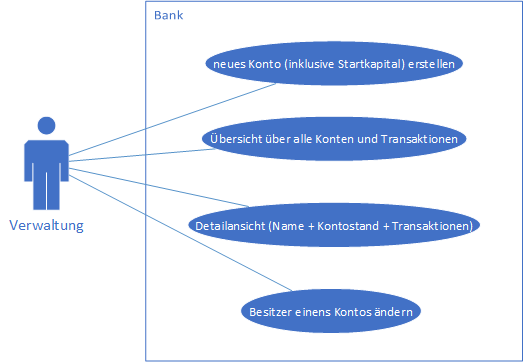


Abbildung Use-Case-Diagramm der Verwaltung

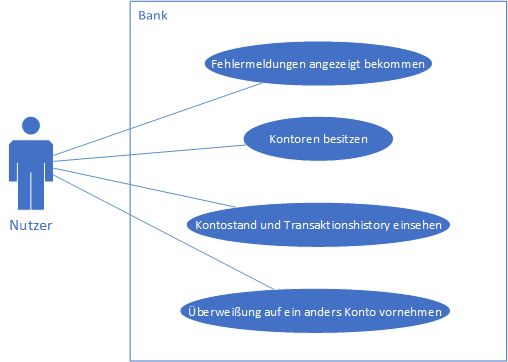


Abbildung Use-Case-Diagramm des Nutzers

## Benutzeroberflächenkonzept

Login

Startseite (App)

Neue Transaktion

Detailansicht

Startseite (Verwaltung)

## Produktmodell

### **Funktionale Anforderungen**

#### Nutzer

/ UC 1010 / Als Nutzer möchte ich ein oder mehrere Konten besitzen können.

/ UC 1020 / Als Nutzer möchte ich meinen Kontostand und meine Transaktionshistorie einsehen können.

/ UC 1030 / Als Nutzer möchte ich eine Überweisung auf ein anders Konto vornehmen können.

/ UC 1040 / Als Nutzer möchte ich Fehlermeldungen angezeigt bekommen.

#### Verwaltung

/ UC 2010 / Als Verwaltung möchte ich ein neues Konto (inklusive Startkapital) erstellen können.

/ UC 2020 / Als Verwaltung möchte ich eine Übersicht über alle Konten und Transaktionen haben.

/ UC 2030 / Als Verwaltung möchte ich ein Konto auswählen können um in eine Detailansicht (Name + Kontostand + Transaktionen) zu erhalten.

/ UC 2040 / Als Verwaltung möchte ich den Besitzer Einens Kontos ändern können.

### **Nichtfunktionale Anforderungen**

/ NFC 3000 / Die Validierung von eingaben soll von Server vorgenommen werden.

/ NFC 3010 / Der Client soll nur öffentliche Schnittstellen vom Server verwenden

/ NFC 3020 / Der Server soll konfigurierbar sein.

/ NFC 3030 / Der Server soll auch von andere Gruppen über deren Client nutzbar sein.

/ NFC 3040 / Der Client für den Nutzer soll als App entwickelt werden.

/ NFC 3050 / Ein Konto soll einem Inhaber zugeordnet werden können.

/ NFC 3060 / Ein Konto hat eine eindeutige 4-stellige Kontonummer.

/ NFC 3070 / Kontostand kann serverseitig durch die Transaktionshistorie ermittelt werden.

/ NFC 3080 / Nur das Bank Konto darf überzogen werden bei Überweisungen.

/ NFC 3090 / Transaktionen können nur zwischen zwei verschiedenen Konten vorgenommen werden.

/ NFC 3100 / Umsetzung der Web-Oberfläche soll in einer angular2 Webanwendung realisiert werden.

/ NFC 3110 / Zugriff auf die Datenbank soll über JDBC erfolgen.

/ NFC 3120 / Rückgabe der Transaktionshistorie soll im JSON Format erfolgen.

/ NFC 3130 / Aktionen von Nutzer/Verwaltung sollen in der Konsole und in einer Log-Datei gespeichert/ausgegeben werden.

## Basismaschine

Tabelle Basismaschine

|  |  |
| --- | --- |
| Windows | Smartphone |
| 1,6GHz Prozessor | Android 5 oder höher |
| 512MB Ram | 1080 x 1920 Auflösung |
| Ab FireFox v. 12 | Internetfähig (Wlan, Mobilfunk) |
| Ab Chrome v. 41 |  |
| Min Java 7 |  |

## Qualitätsanforderungen

Tabelle Quälitätsanforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Produktqualität | sehr wichtig | wichtig | weniger wichtig | nicht relevant |
| Funktionalität | \* |  |  |  |
| Zuverlässigkeit |  | \* |  |  |
| Benutzbarkeit |  | \* |  |  |
| Effizenz |  |  | \* |  |
| Änderbarkeit |  |  | \* |  |
| Übertragbarkeit | \* |  |  |  |

# Projektplan

## Benutzerlogin

Der Login in die App geschieht durch eine Maske in welcher die Kontonummer des Kontos eingegeben wird. Für den Zugriff auf ein Konto wird also nur die Kontonummer benötigt und kein Passwort.

## Nutzer (App)

Der Benutzer soll innerhalb der App angezeigt bekommen wie viel Geld er momentan auf seinem Konto vorhanden hat, sowie seine komplette Transaktionshistorie. In dieser werden alle seine bisher getätigten Transaktionen angezeigt. Weiterhin soll der Nutzer in der Lage sein eine neue Überweisung ausführen zu können. Falls diese Überweisung Fehler ausweißt soll er eine Fehlermeldung erhalten.

## Verwaltung (App)

Die Verwaltung hat Zugriff auf alle Konten und kann dort sehen welche Transaktionen diese getätigt haben. Weiterhin kann die Verwaltung auch neue Konten erstellen und die Besitzer bestimmen. Bei bereits vorhandenen Konten kann die Verwaltung auch im Nachhinein den Besitzer eines Kontos ändern. Außerdem soll es eine Detailansicht für die Verwaltung geben bei welcher der Besitzer, der Kontostand und die Transaktionen eines Kontos angezeigt werden.

# Technisches Konzept

## Klassendiagramm

In diesem Klassendiagramm sieht man alle Klassen in welchem Zusammenhang diese miteinander stehen und welche Methoden diese beinhalten.

## Datenbankmodell

Durch das Datenbankmodell kann man erkennen, wie die Datenbank hinter der Anwendung zusammengestellt ist. Außerdem kann man dadurch besser seine Programmierung anpassen, da man bereits weiß in welcher Tabelle welche Daten vorzufinden sind.

# Implementierung

## Verwendete Programmiersprache

Zur Entwicklung des Servers wurde die Entwicklungsumgebung Eclipse benutzt, um eine gemeinsames Programmieren zu versichern wurde eine Versionsverwaltung mithilfe von Git benutzt.

Für die Entwicklung der App wurde Android Studio benutzt, da dieses die offizielle Entwicklungsumgebung für Android Apps ist. Die verwendete Programmiersprache ist sowohl beim Server als auch bei der App Java.

## SQLite Datenbank

Die Datenbank ist eine SQLite Datenbank welche lokal auf dem Rechner liegen muss, damit das Programm sich damit verbinden kann.

## JDBC

JDBC wird für die Datenbankschnittstelle für die Java-Plattform, welche für rationale Datenbanken benutzt wird. Da wir eine SQLite Datenbank innerhalb des Projektes benutzen, muss auch der dafür vorgesehene Datenbanktreiber benutzt werden.

## Weitere Frameworks / Technologien die wir benutzen

Weitere Frameworks / Technologien die wir benutzen:

• HTML

• Atom.io

• GitHub

• REST-Framework Jersey

• Webserver Jetty

# Zeitablauf Gegenüberstellung

# Test

Die Tests von dem Programm wurden während der Entwicklung von der einzelnen Person durchgeführt. Falls Fehler aufgekommen sind, wurden diese meistens versucht zu beheben bevor wieder eine neue Version in das gemeinsame Git geladen wurde. Trotzdem wurden nicht im Ansatz alle möglichen Fehlerquellen gefunden und behoben, da dies viel zu viel Zeit in Anspruch nehmen würde.

# Fazit

## Bewertung

## Fazit

Die Aufgabe, welche innerhalb der Gruppenarbeit erarbeitet werden sollte war passend ausgewählt und nicht so groß das es nicht möglich gewesen wäre diese komplett zu schaffen. Allerdings hat es am Zeitmanagement der Gruppenmitglieder gefehlt und daher war zu Ende hin ein großer Zeitdruck entstanden und es war nicht mehr möglich alle am Anfang geplanten Ziele zu erreichen.

Trotzdem war das Projekt sehr lehrreich und man hat wieder einmal die Wichtigkeit von gutem Zeitmanagement mitbekommen. Außerdem hat man gemerkt wie wichtig es ist Aufgaben klar zu verteilen und sich selber Meilensteine zu setzten, welche dann auch jeder aus der Gruppe pünktlich einhalten sollte. Zum Ende hin hat man gemerkt das man niemals die Zeit vergessen sollte die solch ein Projekt in Anspruch nimmt.