**Software Engineering**

**Projekt : Bank Management System**

**Use Case : Kontostand**

Kurzbeschreibung: Mitarbeiter will den Kontostand abrufen.

Primärer Aktor: Service Mitarbeiter

Vorbedingung: Der Kunde ist registriert und hat den Mitarbeiter seine Zugangsdaten übergeben.

Nachbedingung: Überprüfung des Kontostands

**Erfolgsszenario:**

1. Mitarbeiter wählt „view account information“ aus.
2. Mitarbeiter gibt Sicherheitsnummer ein.
3. Das System validiert die Sicherheitsnummer.
4. Das System gibt den Kontostand aus.
5. Use Case endet erfolgreich.

**Erweiterungen:**

(1-3)a. Mitarbeiter Unterbricht den Vorgang.

(1-3)a.1. Use Case endet Erfolglos.

3a. System validiert die Zugangsdaten NICHT und gibt eine Fehlermeldung aus.

Schritt 3.a wiederholt sich solange die Zugangsdaten falsch sind.

3a||a. Mitarbeiter Drückt „-1“:

3a||a.1 System Unterbricht den Vorgang und kehrt zum Menü Auswahl zurück.

3a||a.2 Use Case endet erfolglos.

**Use Case: Geld einzahlen**

Kurzbeschreibung: Mitarbeiter will auf einem Konto einzahlen.

Primärer Aktor: Service Mitarbeiter

Vorbedingung: Mitarbeiter hat die Zugangsdaten der Kunde und hat auch das Geld von ihm erhalten

Nachbedingung: Der eingegebene Betrag wurde dem Konto hinzugefügt.

**Erfolgsszenario:**

1. Mitarbeiter wählt „Deposit into account“ aus.
2. Mitarbeiter gibt die Zugangsdaten ein.
3. Das System validiert die Zugangsdaten.
4. Der Mitarbeiter gibt den Beitrag ein.
5. Use Case endet erfolgreich.

**Erweiterungen:**

(2-4)a. Mitarbeiter Unterbricht den Vorgang.

(2-4).a.1. Use Case endet erfolglos.

3a. Das System validiert die Zugangsdaten NICHT.

3a.1 System gibt die Fehlermeldung aus und fragt wieder nach der Nummer.

Schritt 3a.1 wiederholt sich solange die Sozial Nummer Falsch ist.

3a.1||a. Mitarbeiter drückt „-1“.

3a.1||a.1.System kehrt zum Menü zurück.

3a.1||a.2 Use Case endet Erfolglos.

Use Case: Geld auszahlen

Kurzbeschreibung: Der Mitarbeiter will Geld für den Kunden auszahlen.

Primärer Aktor: Service Mitarbeiter

Vorbedingung: Der Kunde übergibt dem Service Mitarbeiter seine zugangsdaten.

Nachbedingung: Der Betrag wurde vom Konto abgezogen.

**Erfolgsszenario:**

1. Mitarbeiter wählt „withdraw from account“ aus.
2. Mitarbeiter gibt die Zugangsdaten ein.
3. Das System validieren die Zugangsdaten.
4. Mitarbeiter gibt den auszuzahlenden Betrag ein
5. Das System validiert den Betrag
6. Use Case endet Erfolgreich

**Erweiterungen:**

(2-4)a. Mitarbeiter Unterbricht den Vorgang.

(2-4)a.1. Use Case endet erfolglos.

3a. System validiert die Zugangsdaten NICHT.

3a.1. System gibt die Fehlermeldung aus und fragt wieder nach der Zugangsdaten.

Schritt 3a.1 wiederholt sich solange die Zugangsdaten falsch sind.

3a.1||a. Mitarbeiter drückt „-1“ aus:

3a.1||a.1. System unterbricht den Vorgang und kehrt zum Menü.

3a.1||a.2. Use Case endet erfolglos.

5a. Der vorhandene Betrag reicht nicht:

5a.1. System gibt die Fehlermeldung aus und kehrt zum Menü.

5a.2. Use Case endet erfolglos.

**Use Case : Konto Löschen**

Kurzbeschreibung: Service Mitarbeiter will ein Konto löschen.

Primärer Aktor : Service Mitarbeiter.

Vorbedingung: Kunde ist registriert und hat den Mitarbeiter seine Zugangsdaten übergeben.

Nachbedingung: Das Konto ist vom System gelöscht.

**Erfolgsszenario:**

* 1. Mitarbeiter wählt „Erase Account“.
  2. Mitarbeiter gibt die Zugangsdaten ein.
  3. System validiert die Zugangsdaten (stellt fest dass das Konto existiert).
  4. System löscht das Konto.
  5. Use Case endet erfolgreich.

**Erweiterung:**

(1-4)a. Mitarbeiter unterbricht den Vorgang.

(1-4)a.1 Use Case endet erfolglos.

3a. System validiert die Zugangsdaten NICHT.

3a.1 System gibt die Fehlermeldung aus und fragt wieder nach der Zugangsdaten.

Schritt 3a.1 wiederholt sich solange die Zugangsdaten falsch sind.

3a.1||a. Mitarbeiter drückt „-1“ aus.

3a.1||a.1. System unterbricht den Vorgang.

3a.1||a.2. Use Case endet erfolglos.

**Use Case: Konten entfernen**

Kurzbeschreibung: Der Service Mitarbeiter will die Datei (die gespeicherten Konten) aus der Datenbank löschen.

Primärer Aktor : Service Mitarbeiter

Vorbedingung: Die Datei muss vorhanden sein und den Mitarbeiter muss der Datei name kennen.

Nachbedingung: Die Datei ist aus der Datenbank löscht.

**Erfolgsszenario:**

* + 1. Mitarbeiter wählt „Remove account“.
    2. Mitarbeiter gibt den Name ein.
    3. System validiert den Name.
    4. System löscht die Datei von der Datenbank.
    5. {Datenbank pflegen}
    6. Use Case endet erfolgreich.

**Erweiterungen:**

2a. Mitarbeiter unterbricht den Vorgang.

2a.1 Use Case endet erfolglos.

3a.1. System validiert den Name NICHT.

3a.2. System gibt die Fehlermeldung aus und kehrt zum Menü

3a.3. Use Case endet erfolglos.

**Use Case : Konten Laden**

Kurzbeschreibung: Service Mitarbeiter will die vorhandenen Konten aus der Datenbank laden.

Primärer Aktor: Service Mitarbeiter

Vorbedingung: Die Datei muss vorhanden sein und der Mitarbeiter muss den Name Kenne.

Nachbedingung: Die Datei (Konten) sind vom Mitarbeiter aus der Datenbank geladen.

**Erfolgsszenario:**

* + - 1. Mitarbeiter wählt „Load account“.
      2. Mitarbeiter gibt den Name ein.
      3. System Validiert den Name.
      4. System Löscht die Datei.

**Erweiterungen:**

2a. Mitarbeiter unterbricht den Vorgang.

2a.1 Use Case endet erfolglos.

3a. System validiert den Name NICHT.

3a.1 System gibt die Fehlermeldung aus und kehrt zum Menü zurückt.

3a.2. Use Case endet erfolglos.

**Use Case: Konten archivieren**

Kurzbeschreibung: Service Mitarbeiter will neue Konten archivieren.

Primärer Aktor: Service Mitarbeiter

Vorbedingung: Die Verbindung zur Datenbank muss sichergestellt werden.

Nachbedingung: Die Konten wurden sicher auf die Datenbank archiviert.

**Erfolgsszenario:**

* + - * 1. Mitarbeiter wählt „Archive Accounts“
        2. Mitarbeiter gibt der Dateiname ein.
        3. System archiviert die Konten.
        4. {Datenbank pflegen}.
        5. Use Case endet erfolgreich.

**Erweiterungen:**

2a. Mitarbeiter Unterbricht den Vorgang.

2a.1. Use Case endet erfolglos.

**Use Case: Datenbank pflegen**

**Kurzbeschreibung:** Das System will die eigegebenen Parameter im Datenbank Speichern

**Primärer Aktor:** System

**Vorbedingung:** Verbindung zum Datenbank muss sichergestellt werden.

**Nachbedingung:** Die Daten müssen so gesichert werden dass sie später noch abrufbar sind.

**Erfolgsszenario:**

System prüft die Verbindung zum Datenbank.

System Validiert die Verbindung.

System speichert die eingegebenen Daten mit ihrem Parameter( Name zum Beispiel)

Use Case endet erfolgreich.

**Erweiterungen:**

**2a.** System validiert die Verbindung NICHT.

**2a.**1 System Unterbricht den Vorgang.

**2a.**2 Use Case endet erfolglos.