Begleitdokument Programmieren II Projekt

Jan Ackermann, Luis Eckert, Joshua Kristof, Kai Pistol, Cynthia Winkler, Marcel Wölke

TINF21CS2

Funktionalität

Unser Programm bildet eine einfache Banking-Applikation ab. Endnutzer (Kunden der Bank) können sich einloggen, Überweisungen tätigen, Geld abheben und neue Konten eröffnen.

Es stehen drei Kontentypen zur Verfügung:

- Girokonto
- Tagesgeldkonto (Geld kann nur an anderes Konto desselben Besitzers ausgezahlt werden, keine Verschuldung möglich)
- Aktiendepot (Geld kann nur vom Konto desselben Besitzers ein-/ausgezahlt werden, Verschuldung möglich)

Für Girokonten und Aktiendepots kann vom Bankmitarbeiter ein Überziehungsrahmen festgelegt werden, über welchen sich der Nutzer bis zu einem gewissen Betrag verschulden kann.

Konten haben eine eindeutige ID mit welcher sie Geld empfangen bzw. an andere Konten überweisen können. Über eine Eingabemaske kann eine Transaktion getätigt werden.

Außerdem können Kunden neue Konten eröffnen, diese müssen allerdings erst vom zuständigen Bankmitarbeiter freigeschalten werden.

Bankmitarbeiter haben zugeordnete Kunden, die von ihnen betreut werden. Für diese Kunden können Sie Konten freischalten und Geld die auf Konten einzahlen (analog zur Einzahlung von Bargeld im realen Leben).

Neue Nutzer können nur von Bankmitarbeitern angelegt werden.

Außerdem haben beide Nutzertypen (Bankmitarbeiter, Kunde) zugeordnete Kundendaten (Name, Adresse, Geburtsdatum, …) welche Sie selbstständig ändern können.

Die Oberfläche soll leicht zu bedienen sein, sowohl für den Endkunden als auch für den Bankmitarbeiter.

Ordnerstruktur

controller/	Controller die von der GUI aufgerufen werden
controller/interfaces	Interfaces welche die von der GUI erwartete API beschreibt
controller/mockups	Test-Funktionalität für die Entwicklung, nicht mehr relevant in fertigem Projekt
data/models	Datenstrukturen (z.B. Nutzer, Konto)
main/	Für Programmaufruf notwendige Programmlogik
utils/	Diverse Hilfsklassen, z.B. Krypto und Datenvalidierung
utils/mysql	Klassen zur Kommunikation mit der MySQL Datenbank

Oberfläche (GUI)

Die Oberfläche ist in Java Swing geschrieben und ist entsprechend Plattformunabhängig. Wir haben die Oberfläche erfolgreich unter verschiedenen Linux-Distributionen und Microsoft Windows getestet.

Beim Starten des Programms wird der Nutzer mit einer Login-Maske begrüßt. An dieser Stelle hat der Nutzer bewusst keine Möglichkeit sich zu registrieren, da diese Funktionalität Bankmitarbeitern vorgehalten ist.

Nach dem Login sind die angezeigten Menüeinträge abhängig von der Rolle des Nutzers. Bankmitarbeiter können die ihnen zugeordneten Kunden sehen, während Nutzer ihre Konten angezeigt bekommen.

Einige Optionen (z.B. “Profil bearbeiten”) werden jedem Nutzer unabhängig der Rolle angezeigt.

Nutzernamen, Kontonamen und Datenvalidierung

Beim Anlegen von neuen Nutzern und Konten wird automatisch eine eindeutige ID generiert. Über diese sind Nutzer und Konten eindeutig zuordbar und werden für das Login und Überweisungen genutzt.

Beim Anlegen eines neuen Kontos / Nutzer wird die eindeutige ID einmalig angezeigt und sollte notiert werden. Zur einfacheren Notation wird die ID beim Erstellen eines neuen Kontos automatisch in die Zwischenablage des Betriebssystems kopiert.

Beim Anlegen eines neuen Nutzers werden einige Validierungen vorgenommen:

- Passwörter müssen min. 8 Zeichen lang sein, einen Großbuchstaben, eine Zahl und ein Sonderzeichen beinhalten
- Vor-/Nachnamen dürfen nur Zeichen von a-z sowie öüäß beinhalten.
- Postleitzahlen müssen vier- oder fünfstellig sein.

Datenbank

Die Anwendung nutzt eine zentrale, im Internet gehostete mySQL Datenbank. Diese wird von großzügigerweise von Joshua Kristof auf privaten Kosten bereitgestellt.

Die Kommunikation mit der Datenbank erfolgt über den JDBC mySQL Treiber.

Beim Öffnen der Anwendung wird geprüft, ob die notwendigen Tabellen existieren. Sollte dies nicht der Fall sein, werden diese automatisch angelegt. Passwörter werden in der Datenbank nicht in Klartext gespeichert, sondern mit einem zufälligen Salt versehen und anschließend ein kryptografischer Hash mit der Hashfunktion SHA-512 erzeugt. Dadurch wird eine versehentliche Veröffentlichung der Passwörter auch bei unberechtigtem Zugriff auf die Datenbank verhindert.