Handbuch „Cash Register“

Ein einfaches RFID gestütztes Bezahlsystem mit Datenbankanbindung als geschlossenes System

Moritz Hilberg 733760 Lukas Köhler 734188

28. Juni 2015

Inhaltsverzeichnis

1 Installation und unterstützte Hardware

1.1 Installation und Starten des Programms

Um das System zu installieren, genügt es, den Ordner *AbgabeProtokoll/JarExport/* auf das eigene Betriebssystem zu kopieren und die Datei *CashRegister.jar* zu starten. Diese öffnet dann direkt das User-Interface des Programms. Wichtig ist, dass eine JRE installiert ist, da sonst *CashRegister.jar* nicht gestartet werden kann. Weitere genutzte Bibliotheken wie *jSSC* oder *sqlite3* müssen nicht manuell installiert werden, da diese bereits im *.jar* Paket liegen und auch dort referenziert werden.

1.2 Bibliotheken und Datenbank

jSSC ist für das Lesen und Schreiben einer seriellen Verbindung und *sqlite3* für das Lesen und Schreiben einer Datenbank. Die Datenbank für das System befindet sich in dem Ordner *AbgabeProtokoll/JarExport/database/* und kann bei Bedarf manuell mit einem Datenbank-Programm wie z.B. dem passenden sqlitebrowser (*http://sqlitebrowser.org/*) geöffnet und beschrieben werden. Die Datenbank umfasst alle User mit Namen und einer passenden Tag-ID, um sie später einem passenden RFID-Tag zuordnen zu können, Transactions, welche alle getätigten Transaktionen in Bezug auf ein User enthält und Values, welche die Geldscheine anhand weiterer RFID-Tags simuliert und jeweils einen Betrag und eine Tag-ID enthält.

1.3 Konfigurationsdatei

Weiter umfasst das Programm eine Konfigrationsdatei (*AbgabeProtokoll/JarExport/configFile*), welche die Befehle für die serielle Schnittstelle enthält. Befehl und zu erwartende Antwort sind durch ein Semikolon getrennt. Die Datei enthält die Befehle, welche bei Start des Programms als Setup an die serielle Schnittstelle geschickt werden und die Befehle die benötigt werden um einen RFID-Tag auszulesen. Die zu erwartende Antwort wird dabei jeweils mit der zurückgekommenen Antwort verglichen.

1.4 Betriebssystem

Das Programm wurde unter Linux entwickelt und getestet, kann jedoch plattformunabhängig unter jedem Betriebssystem mit einer funktionsfähigen JRE gestartet werden, da sämtliche Bibliotheken selbst im *.jar* Paket liegen und diese ebenfalls platformunabhängig sind.

1.5 Reader-Board

Um RFID-Tags zu lesen, wurde das *Atmel ATA 2270-EK1* benutzt. Da ein Bezahlsystem auf keine hohen Reichweiten ausgelegt sein soll, entschieden wir uns bei dem Vergleich

eines LF- beziehungsweise HF-Systems für Ersteres. Auf die Konfiguration und Installation des *Atmel ATA 2270-EK1* wird hier nun nicht weiter eingegangen, da dies detailert in Termin 3 des Praktikums durchgeführt wurde.

2 Benutzerführung und Funktionen

Abbildung 1 zeigt den erfolgreich eingelesenen Tag-ID *moritzTagID*, welcher im Feld *Customer ID* nochmals dargestellt ist. Weiterhin ist der Name in dem Feld *Name* und der aktuelle Kontostand in *Balance* zu sehen. Dieser beträgt in diesem Fall zurzeit 0. Das grüne Label unten weist hinzukommend auf das erfolgreiche Einlesen und Erkennen hin. *Counter* zeigt, wie oft derselbe Tag hintereinander eingelesen wurde, falls dieser sich länger in der Reichweite des Lesegerätes befindet.

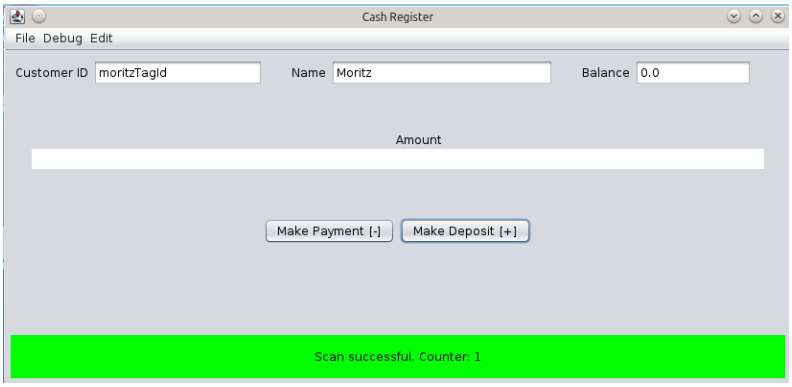

Abbildung 1: Tag-ID *moritzTagID* wurde eingelesen

In Abbildung 2 wurde um Geld einzuzahlen zuvor auf *Make Deposit* geklickt. In diesem Beispiel wurden Value-Tags mit 20.0 und 5.0 hintereinander eingelesen. Wie zu sehen ist, werden diese direkt summiert. Wird die Aktion mit einem Klick auf *Ok* bestätigt, wird die Transaktion ausgeführt und anschließend in der Datenbank gespeichert.

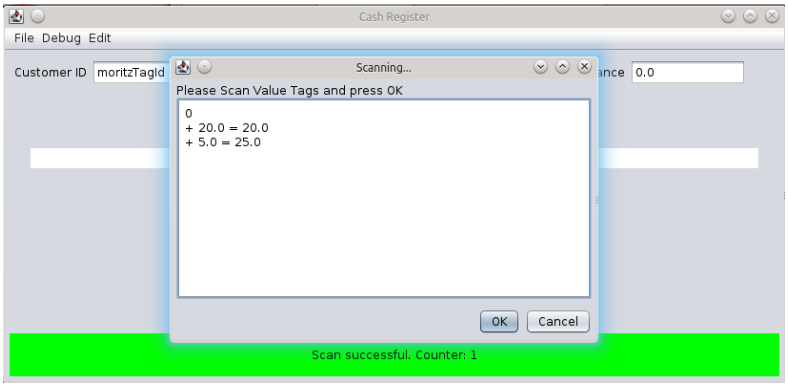

Abbildung 2: Make Desposit

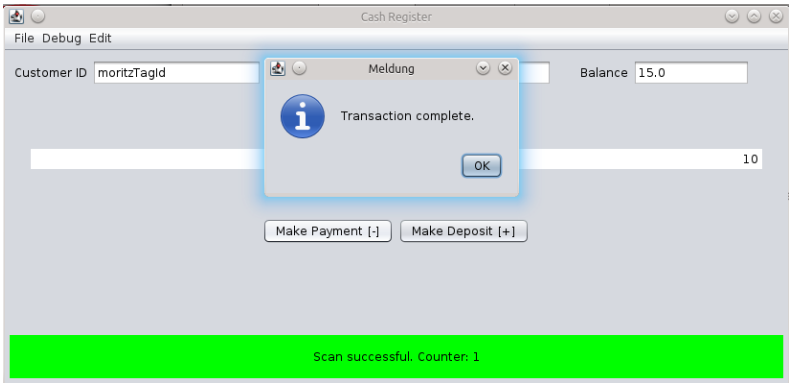

Abbildung 3: Make Payment

Abbildung 3 zeigt das Gegenteil zu Abbildung 2. Hier wird nun Geld ausgezahlt, indem der Benutzer seinen gewünschten Betrag in das Feld *Amount* einträgt und anschließend mit *Make Payment* die Transaktion bestätigt. Der Betrag wird dann vom Konto abgezogen und in die Datenbank gespeichert.

In Abbildung 4 wurde ein Tag eingescannt, dessen ID der Datenbank beziehungsweise dem Programm nicht bekannt ist. Somit wird im unteren Feld in rot bestätigt, dass es sich hier um eine unbekannte ID handelt. *Make Deposit* und *Make Payment* hat dann keine Funktion.

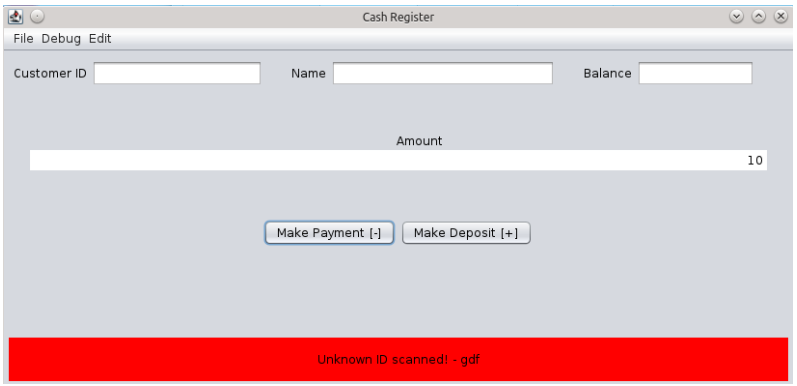

Abbildung 4: Unbekannte Tag-ID

In der Menü-Leiste kann unter *File* die Verbindung gestartet, getrennt und überprüft werden. *Debug* gibt die Möglichkeit eine Tag-ID einzugeben, um so die Funktionsfähigkeit des Programms zu kontrollieren, ohne einen echten Reader anschließend zu müssen.
ACHTUNG: Dies ist momentan nicht funktionsfähig, da das Programm auf die Nutzung im Praktikum spezialisiert wurde.
Unter *Edit* können die in Abbildung 5 gezeigen Einstellungen abgerufen und geändert werden.

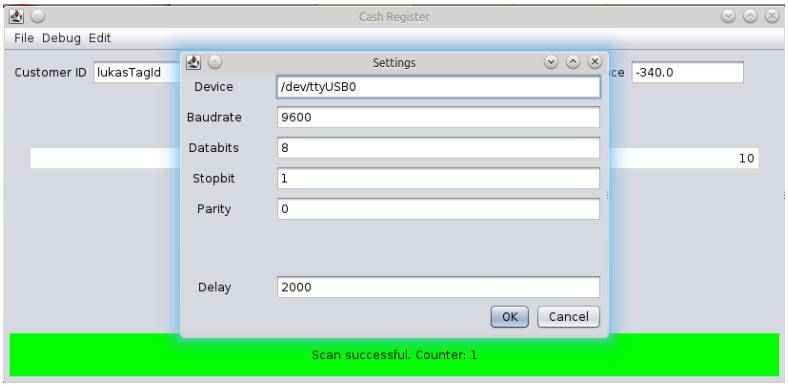

Abbildung 5: Settings

Hier sind alle Einstellungen der seriellen Schnittstelle vorzunehmen.

3 Beschreibung der Realisierung mittels UML Klassendiagrammen

Das Programm ist nach dem Paradigma des Model-View-Controller Patterns aufgebaut. Die meisten Klassen des Controllers und der View sind als Singletonpattern realisert. Daher erfolt die Referenzübergabe über den Aufruf getInstance() der jeweiligen Klasse. Das Model braucht als Einzige Zugriff auf die Datenbank und ist daher von SQLConnection abgeleitet.

4 Anhang

Abbildung 6: MVC - Model

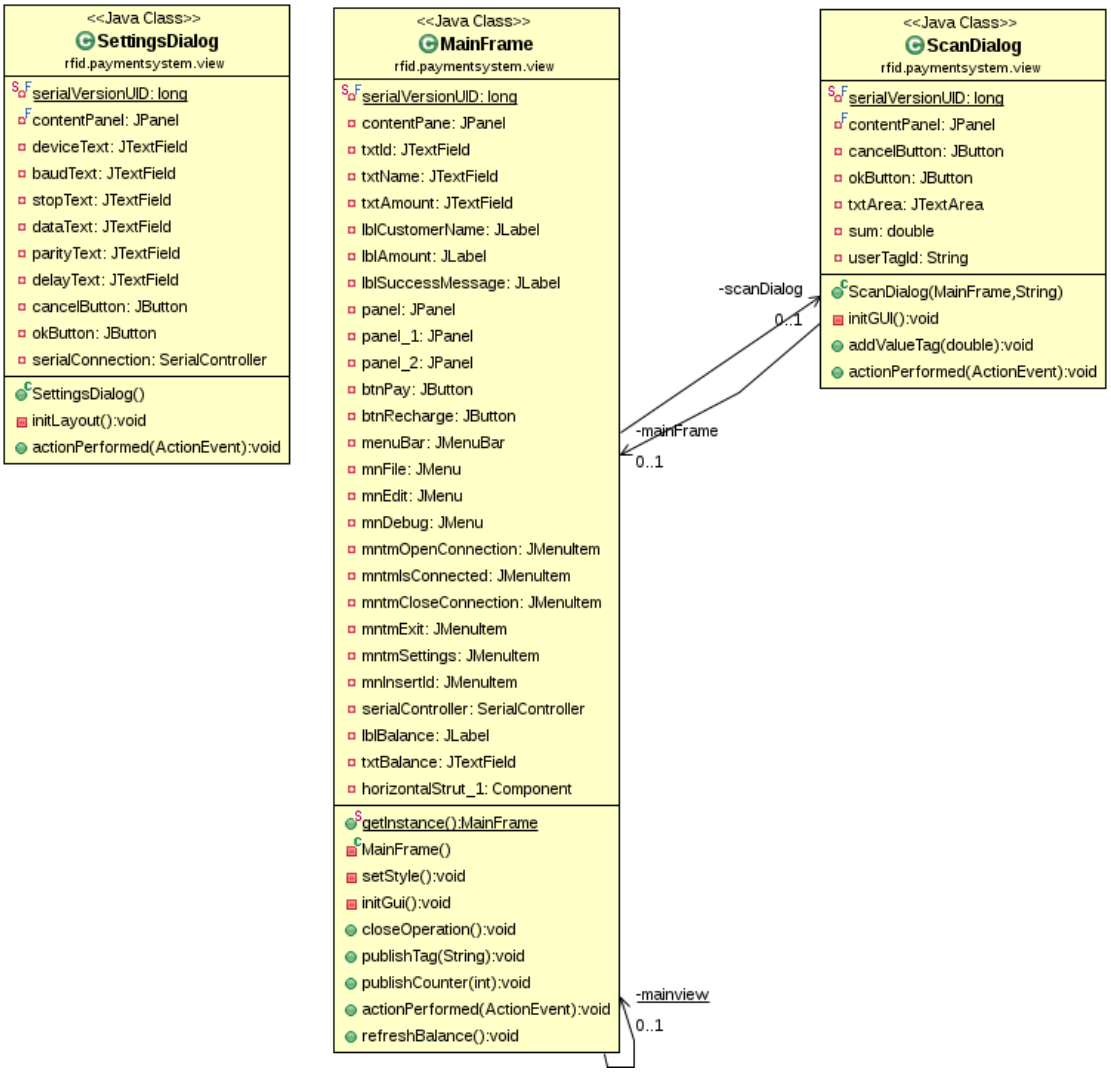

Abbildung 7: MVC - View

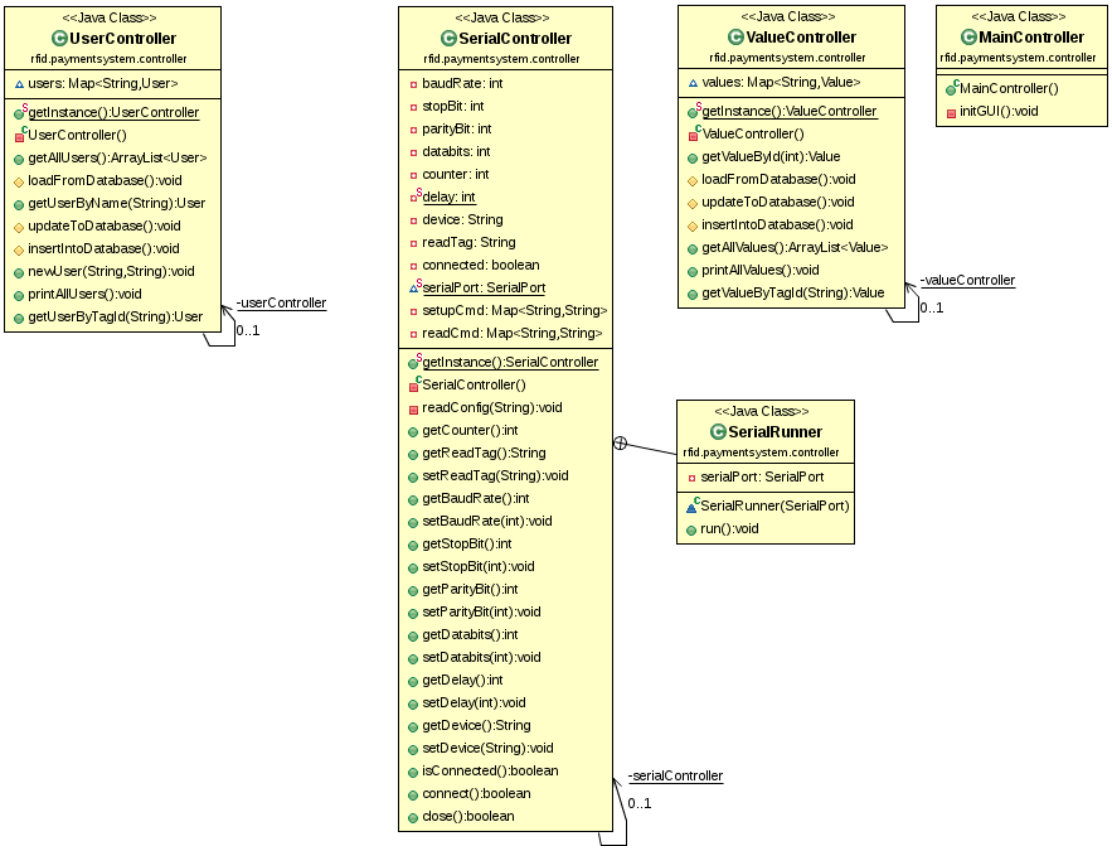

Abbildung 8: MVC - Controller