**Entwicklung eines Lagerabrufs- und Verwaltungssystems mit Datenbank und Android-Anwendung zum Einsatz in der Gastronomie**

Studienarbeit

3. Studienjahr

Modul T3200

des Studienganges Mechatronik

an der Dualen Hochschule Baden-Württemberg

am Standort Stuttgart

von

Marvin Mai und Daniel Schifano

Bearbeitungszeitraum 12 Wochen

Betreuer der dualen Hochschule Prof. Dr.-Ing. Johannes Moosheimer

# Ehrenverantwortliche Erklärung

Gemäß § 5 (2) der „Studien- und Prüfungsordnung DHBW Technik“ vom 06. November 2013.

Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen**,** als die angegebenen Quellen und Hilfsmittel verwendet.

|  |  |  |  |
| --- | --- | --- | --- |
| Stuttgart, 09.01.2018 |  |  |  |
| Ort, Datum |  | Unterschrift |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Stuttgart, 09.01.2018 |  |  |  |
| Ort, Datum |  | Unterschrift |  |

Inhaltsverzeichnis

[Ehrenverantwortliche Erklärung 1](#_Toc508705489)

[Abbildungsverzeichnis 3](#_Toc508705490)

[1 Anforderungen 4](#_Toc508705491)

[1.1 Aufgabenstellung 4](#_Toc508705492)

[2 Pflichtenheft 5](#_Toc508705493)

[3 Pflichtenheft 5](#_Toc508705494)

[3.1 Neue Funktionen 5](#_Toc508705495)

[3.2 Anwendungsfälle 7](#_Toc508705496)

[3.2.1 Anwendungsfalldiagramm Android-Anwendung 7](#_Toc508705497)

[3.2.2 Android- Anwendung 7](#_Toc508705498)

[3.2.3 Anwendungsfalldiagramm Datenbankmanager 11](#_Toc508705499)

[3.2.4 Datenbankmanager 12](#_Toc508705500)

[3.3 Produktdaten 16](#_Toc508705501)

[3.4 Workflows 17](#_Toc508705502)

[3.4.1 Tische 17](#_Toc508705503)

[3.4.2 Bestellungen 18](#_Toc508705504)

[3.4.3 Artikelverwaltung 19](#_Toc508705505)

[3.4.4 Warenein- und -ausgänge 20](#_Toc508705506)

[3.4.5 Bedienung 21](#_Toc508705507)

[3.4.6 Anmeldedaten 22](#_Toc508705508)

[3.5 Benutzeroberfläche 23](#_Toc508705509)

[3.5.1 Benutzeroberfläche der Android-Anwendung 23](#_Toc508705510)

[3.5.2 Benutzeroberfläche des Datenbank-Managers 25](#_Toc508705511)

[4 Entwurf 26](#_Toc508705512)

[5 Implementierung 27](#_Toc508705513)

[5.1 Implementierung des Datenbank-Systems 27](#_Toc508705514)

[5.1.1 Klasse DatabaseService\_Interface 27](#_Toc508705515)

[5.1.2 Klasse RestApiController 28](#_Toc508705516)

[5.2 Implementierung der Android-Anwendung 29](#_Toc508705517)

[6 Test 30](#_Toc508705518)

[6.1 Test des Datenbank-Systems 30](#_Toc508705519)

[6.2 Test der Android-Anwendung 31](#_Toc508705520)

[7 Fazit 32](#_Toc508705521)

[8 Anhang 33](#_Toc508705522)

[8.1 Installationsanweisung 33](#_Toc508705523)

[8.1.1 Datenbank-System 33](#_Toc508705524)

[8.1.2 Android Applikation 35](#_Toc508705525)

[9 Testdokumentation 36](#_Toc508705526)

# Abbildungsverzeichnis

[Abbildung 1 Entwurf der überarbeiteten Datenbank 16](#_Toc508705527)

[Abbildung 2 GUI Android-Anwendung 23](#_Toc508705528)

[Abbildung 3 Gui Datenbank-Manager 25](#_Toc508705529)

# Anforderungen

## Aufgabenstellung

Laut Aufgabenstellung soll ein Lagerabrufs- und Verwaltungssystem mit Datenbank und Android-Anwendung entwickelt werden. Das System besteht aus einem zentralen Computer, der über ein Netzwerk mit beliebig vielen Handheld-Geräten verbunden werden kann. Das Netzwerk wird von einem Router via Wireless Local Area Network (WLAN) zur Verfügung gestellt.

Die Grundstruktur des Systems ist bereits aufgebaut und getestet.   
Durch eine Optimierung der Datenbank und der Schnittstelle zu den Handheld-Geräten, soll es zu einem Ressourcen-effizienterem Austausch von Daten zwischen Datenbank und Handheld-Geräten kommen. Die Optimierung wird zusätzlich zur Umsetzung verschiedener Erweiterungen benötigt.   
Ebenfalls soll das System dahingehen erweitert werden, dass ein Login für verschiedene Mitarbeiter möglich ist. Diese Login Daten müssen auf der Datenbank gegengeprüft werden. Sind diese dort hinterlegt, kann der Mitarbeiter auf das System zugreifen. So wird sichergestellt, dass nur befugtes Personal mit dem System arbeiten kann.

Zusätzliche Erweiterungen des Lagersystems sind im Pflichtenheft vermerkt.

# Pflichtenheft

## Neue Funktionen

Die folgenden Funktionen sollen im Rahmen dieser Studienarbeit erarbeitet und im bisherigen System implementiert werden:

|  |  |  |  |
| --- | --- | --- | --- |
| Nr | Bezeichnung | Beschreibung | Betroffene Systeme und deren Änderung  (APP: Android-App,  DBM: Datenbankmanager,  DBS: Database-System) |
| 1 | Mitarbeiter-login | * Jeder Service-Mitarbeiter muss sich in der App mit einem Login (Name und Passwort) identifizieren. * Passwort und Name werden im Datenbankmanager definiert. * Die Zugangsdaten werden in der Datenbank abgelegt. | * APP: In einem neuen Login-Fenster soll das Eingeben von den Login-Daten und das Einloggen durch eine Abfrage beim Server möglich sein. Die Verifizierung soll durch den http-Headers bei jeder Anfrage umgesetzt werden. * DBS: Der Server muss so angepasst werden, dass alle Anfragen durch eine Verifizierung im http-Header eindeutig überprüft werden können. * DBM: Ein neuer Tab muss erstellt werden, in dem das Anlegen von Login-Daten möglich ist. |
| 2 | Artikel-kategorien | * Alle Artikel sollen Kategorien zugeordnet werden. (Getränke, Hauptspeisen...) * In der App sollen die Artikel in Kategorien geordnet angezeigt werden. | * APP: Es soll beim Befüllen der Artikelanzeige nach Kategorien sortiert werden. Diese sollen logisch angeordnet werden (Vorspeise vor Hauptgang usw.). * DBS: DatabaseService und Netzwerkschnittstelle muss angepasst werden. * DBM: Beim Anlegen eines Artikels muss es möglich sein, eine Kategorie auszuwählen. |
| 3 | Artikel-kommentare | * Es soll bei einer Bestellung möglich sein, jedem Artikel einen Kommentar hinzuzufügen. Das dient bspw. dem Vermerken von ungewünschten Zutaten. * In der App soll dies durch ein langes Drücken auf den Artikeleintrag passieren. Anschließend geht ein Eingabefenster auf, in dem der Kommentar eingegeben werden kann. Anschließend wird der Artikel mit Kommentar der Bestellung hinzugefügt. | * APP: Implementieren der Funktion, einem bestimmten Artikel einen Kommentar hinzuzufügen. * DBS: Es wird eine neue Tabelle angelegt, in der zu einer Bestellung Artikel mit zugehörigem Kommentar zugeordnet wird. Die Datenbank-Schnittstelle wird nur soweit geändert, dass beim Abholen von Bestellungen diese durch eine Kommentartabelle ergänzt werden. * DBM: Es werden die Kommentare zusätzlich zu den restlichen Informationen angezeigt. |
| 4 | Rechnungs-split | * Bei einer offenen Bestellung soll es möglich sein, dass alle Gäste separat Zahlen können. * Dafür soll in einer Übersicht durch die Bedienung die zu Zahlenden Artikel ausgewählt werden können und der Teilbetrag angezeigt werden. | * APP: Die neue Übersicht und Funktion muss implementiert werden und auf das neue Datenbank-Modell angepasst werden. * DBS: Es muss eine neue Tabelle erstellt und angesprochen werden, in der die bestellten Artikel separat abgelegt werden. Diesen wird ein „Bezahlt“-Attribut zugordnet. * DBM: -- |

## Anwendungsfälle

### Anwendungsfalldiagramm Android-Anwendung



### Android- Anwendung

|  |  |
| --- | --- |
| **Bezeichnung** | AW1b - Mitarbeiterlogin |
| Priorität | A |
| Ziel | Der Mitarbeiter loggt sich in der Android-Anwendung mit personifizierten Daten ein. Damit sind alle folgenden Aktionen mit seinem Konto authentifiziert und können ihm zugeordnet werden. |
| Vorbedingung | Der Mitarbeiter ist nicht eingeloggt und hat Zugriff auf Login-Daten, die in der Datenbank hinterlegt wurden. |
| Nachbedingung | Die Anmeldung wird mit einer Meldung bestätigt. |
| Nachbedingung im Sonderfall | 1. Login-Name existiert nicht in der Datenbank. 2. Das Passwort ist falsch. 3. Es besteht keine Verbindung zur Datenbank. |
| Akteure | Bedienung |

|  |  |
| --- | --- |
| **Bezeichnung** | AW2b – Passwort ändern |
| Priorität | D |
| Ziel | Der Mitarbeiter ändert in der Android-Anwendung sein Passwort. |
| Vorbedingung | Der Mitarbeiter ist eingeloggt und hat sein Passwort erneut eingegeben. Das neue Passwort wurde zwei Mal eingegeben. |
| Nachbedingung | Das Passwort erfüllt Sicherheitskriterien und wurde erfolgreich in der Datenbank geändert. |
| Nachbedingung im Sonderfall | 1. Das Passwort erfüllt die Sicherheitskriterien nicht. 2. Es besteht keine Verbindung zu Datenbank. |
| Akteure | Bedienung |

|  |  |
| --- | --- |
| **Bezeichnung** | AW3b – Artikel als produziert markieren |
| Priorität | B |
| Ziel | Wenn die Bedienung Artikel einer Bestellung von der Küche abholt, markiert sie diese in einer Übersicht in der Android-Anwendung als produziert. |
| Vorbedingung | Es existiert eine Bestellung mit noch nicht produzierten Artikeln, die in der Küche zubereitet wurden. |
| Nachbedingung | Die Artikel werden in der Datenbank als produziert markiert. |
| Nachbedingung im Sonderfall | Es wird eine Fehlermeldung angezeigt, dass keine Verbindung zur Datenbank aufgebaut werden konnte. |
| Akteure | Bedienung |

|  |  |
| --- | --- |
| **Bezeichnung** | AW4b – Rechnung splitten |
| Priorität | C |
| Ziel | Von einer bereits existierenden Bestellung soll ein Teilbetrag vom Kunden bezahlt werden. Dafür soll von der Bedienung in einer separaten Übersicht die zu zahlenden Artikel ausgewählt werden können. Der zu zahlende Betrag wird angezeigt. |
| Vorbedingung | Es existiert für den Tisch, an dem der Kunde sitzt, eine Bestellung mit noch nicht bezahlten Artikeln. |
| Nachbedingung | Der bestellte Artikel wird als bezahlt markiert und erscheint nicht mehr in der Übersicht der zu zahlenden Artikel in der Android-Anwendung. |
| Nachbedingung im Sonderfall | Es besteht keine Verbindung zur Datenbank und die Artikel wurden in der Datenbank nicht als bezahlt markiert. |
| Akteure | Bedienung |

|  |  |
| --- | --- |
| **Bezeichnung** | AW5b – Artikel einsehen (neu) |
| Priorität | F |
| Ziel | Die Artikel werden in einer Übersicht dargestellt, in der diese nach Kategorien sortiert sind. |
| Vorbedingung | 1. Ein Tisch wurde ausgewählt. 2. In der Datenbank wurden Artikel mit entsprechenden Kategorien hinterlegt. |
| Nachbedingung | Die Artikel werden richtig und sortiert angezeigt. |
| Nachbedingung im Sonderfall | Es wird eine Fehlermeldung angezeigt, dass keine Verbindung zur Datenbank aufgebaut werden konnte. |
| Akteure | Bedienung |

|  |  |
| --- | --- |
| **Bezeichnung** | AW6b – Bestellten Artikel kommentieren (optional) |
| Priorität | E |
| Ziel | Einem Artikel soll ein Kommentar hinzugefügt werden, in dem eine Nachricht für die Küche hinterlegt wird. |
| Vorbedingung | Es wurde ein Artikel ausgewählt, der einer Bestellung hinzugefügt werden soll. |
| Nachbedingung | Der Kommentar wurde dem Artikel hinzugefügt und dieser in die Datenbank geschrieben. |
| Nachbedingung im Sonderfall | -- |
| Akteure | Bedienung |

### Anwendungsfalldiagramm Datenbankmanager



### Datenbankmanager

|  |  |
| --- | --- |
| **Bezeichnung** | AW7b - Login-Daten anlegen |
| Priorität | A |
| Ziel | Damit sich die Bedienungen im System authentifizieren können, müssen Login-Daten erstellt werden.  Die Login-Daten müssen nach dem Anlegen ausgedruckt werden, damit sie der Bedienung übergeben werden können. |
| Vorbedingung | Eine neue Bedienung wurde eingestellt und benötigt Zugriff auf das System |
| Nachbedingung | Die Login-Daten wurden erfolgreich in der Datenbank angelegt und ausgedruckt. |
| Nachbedingung im Sonderfall | Es wird eine Fehlermeldung angezeigt, dass keine Verbindung zur Datenbank besteht. |
| Akteure | Gastronom |

|  |  |
| --- | --- |
| **Bezeichnung** | AW8b – Login-Daten einsehen |
| Priorität | A |
| Ziel | Die Login-Daten aller Bedienungen können eingesehen werden. |
| Vorbedingung | Der Gastronom will alle Login-Daten einsehen. Er öffnet den Reiter um die Login-Daten einzusehen |
| Nachbedingung | Der Gastronom kann alle Login-Daten einsehen. |
| Nachbedingung im Sonderfall | Es besteht keine Verbindung zur Datenbank. Es wird eine Fehlermeldung angezeigt. |
| Akteure | Gastronom |

|  |  |
| --- | --- |
| **Bezeichnung** | AW9b – Login-Daten bearbeiten |
| Priorität | A |
| Ziel | Die Login-Daten einer Bedienung müssen bearbeitet werden. Der Name ändert sich oder das Passwort soll geändert werden. |
| Vorbedingung | Es existieren Login-Daten für die entsprechende Bedienung und es besteht der Bedarf diese zu ändern.  Die bereits bestehenden Daten wurden erfolgreich in der Anwendung eingegeben. |
| Nachbedingung | Die Login-Daten wurden erfolgreich geändert und in der Datenbank hinterlegt. |
| Nachbedingung im Sonderfall | Das geänderte Passwort erfüllt nicht die Sicherheitskriterien.  Es besteht keine Verbindung zu Datenbank. Es wird eine Fehlermeldung ausgegeben. |
| Akteure | Gastronom |

|  |  |
| --- | --- |
| **Bezeichnung** | AW10b – Login-Daten löschen |
| Priorität | A |
| Ziel | Die Login-Daten einer Bedienung werden gelöscht, wenn diese nicht mehr benötigt werden. |
| Vorbedingung | Die Bedienung benötigt die Login-Daten nicht mehr. Beispielsweise aufgrund von einer Kündigung.  Der Gastronom löscht die Daten über einen Rechtsklick in der Darstellung. |
| Nachbedingung | Die Login-Daten wurden erfolgreich gelöscht und die Bedienung kann sich damit nicht mehr im System authentifizieren. |
| Nachbedingung im Sonderfall | Es wird eine Fehlermeldung angezeigt, dass keine Verbindung zur Datenbank besteht. Die Login-Daten werden nicht gelöscht |
| Akteure | Gastronom |

|  |  |
| --- | --- |
| **Bezeichnung** | AW11b – Bedienungs-Mitarbeiterdaten anlegen |
| Priorität | A |
| Ziel | Jede Bedienung wird mit ihren persönlichen Daten in der Datenbank hinterlegt. Dazu werden vom Anwender in einem Formular die Daten eingegeben. |
| Vorbedingung | Es muss ein neuer Mitarbeiter angelegt werden, wenn eine neue Bedienung eingestellt wird. |
| Nachbedingung | Die Daten der Bedienung wurden in der Datenbank hinterlegt. |
| Nachbedingung im Sonderfall | Es wird eine Fehlermeldung angezeigt, dass keine Verbindung zur Datenbank aufgebaut werden konnte. |
| Akteure | Gastronom |

|  |  |
| --- | --- |
| **Bezeichnung** | AW12b – Bedienungs-Mitarbeiterdaten einsehen |
| Priorität | A |
| Ziel | In einer tabellarischen Übersicht sollen die Mitarbeiterdaten aller Bedienungen dargestellt werden. |
| Vorbedingung | Es besteht eine Verbindung zur Datenbank. |
| Nachbedingung | Es werden alle vorhandenen Mitarbeiterdaten dargestellt. |
| Nachbedingung im Sonderfall | Es wird eine Fehlermeldung angezeigt, dass keine Verbindung zur Datenbank aufgebaut werden konnte. |
| Akteure | Gastronom |

|  |  |
| --- | --- |
| **Bezeichnung** | AW13b – Bedienungs-Mitarbeiterdaten bearbeiten |
| Priorität | A |
| Ziel | Die Daten werden vom Anwender bearbeitet. In einem Formular werden die bisherigen Daten durch neue überschrieben. |
| Vorbedingung | Es existiert ein Eintrag, dessen Daten bearbeitet werden können. |
| Nachbedingung | Die Daten wurden erfolgreich bearbeitet und werden aktualisiert in der tabellarischen Ansicht dargestellt. |
| Nachbedingung im Sonderfall | Es wird eine Fehlermeldung angezeigt, dass keine Verbindung zur Datenbank aufgebaut werden konnte. |
| Akteure | Gastronom |

|  |  |
| --- | --- |
| **Bezeichnung** | AW14b – Bedienungs-Mitarbeiterdaten löschen |
| Priorität | A |
| Ziel | Die Daten eines Mitarbeiters müssen gelöscht werden, weil dieser nicht mehr angestellt ist. Dafür wird vom Anwender mit einem Rechtsklick auf den Eintrag das Kontextmenü „Löschen“ ausgewählt. |
| Vorbedingung | Einem Mitarbeiter wurde gekündigt. |
| Nachbedingung | Die Daten wurden gelöscht. In der tabellarischen Übersicht wird der Eintrag nicht mehr angezeigt. |
| Nachbedingung im Sonderfall | Es wird eine Fehlermeldung angezeigt, dass keine Verbindung zur Datenbank aufgebaut werden konnte. |
| Akteure | Gastronom |

|  |  |
| --- | --- |
| **Bezeichnung** | AW15b – Bestellte Artikel einsehen |
| Priorität | A |
| Ziel | Die Artikel, die zu einer Bestellung gehören, werden angezeigt. Dafür wird mit Doppelklick oder Rechtsklick auf die Bestellung die Detail-Ansicht der Bestellung in einem neuen Fenster geöffnet. |
| Vorbedingung | Die tabellarische Übersicht der Bestellungen wurde geöffnet. |
| Nachbedingung | Die Details mit den bestellten Artikeln wurden in einem neuen Fenster angezeigt. |
| Nachbedingung im Sonderfall | Es wird eine Fehlermeldung angezeigt, dass keine Verbindung zur Datenbank aufgebaut werden konnte. |
| Akteure | Gastronom |

## Produktdaten



Abbildung 1 Entwurf der überarbeiteten Datenbank

## Workflows

Im Folgenden wird in Form von Workflows der Kontext zwischen den Anwendungsfällen und der Datenbankstruktur hergestellt. Dafür werden Geschäfts- und Prozessabläufe in Flussdiagrammen dargestellt und deren Auswirkung auf die Datenbankinhalte definiert.

### Tische



### Bestellungen



### Artikelverwaltung



### Warenein- und -ausgänge



### Bedienung



### Anmeldedaten



## Benutzeroberfläche

### Benutzeroberfläche der Android-Anwendung

#### Bestellvorgang



Abbildung 2 GUI Android-Anwendung

#### Bestellte Artikel als produziert markieren (AW3b)



### Benutzeroberfläche des Datenbank-Managers



Abbildung 3 Gui Datenbank-Manager

# Entwurf

## Optimierung des Datenbank-Services

## Authentifizierung mit Login-Daten

In der neuen Datenbank-Tabelle zur Speicherung von Login-Daten werden ein Login-Name und ein zugehöriges Passwort abgelegt, das in einen 256-bit-Hash-Code verschlüsselt wird. Der Hash-Code wird mit einem Hash-Algorithmus (deutsch: Streuwertfunktion) generiert. Dieser Algorithmus erzeugt aus einem String einen zufälligen Integerwert, der keinen Rückschluss auf den ursprünglichen String erlaubt, allerdings bei selbem Eingabewert immer den gleichen Wert enthält. Es ist mit heute bekannten Mitteln nicht möglich, aus dem in der Datenbank gespeicherten Hash-Code auf das Passwort zu schließen. Theoretisch erreichbar ist dies nur mit Brute-Force-Attacken, also dem Eingeben von vielen verschiedenen Eingabewerten, bis der resultierende Hash-Code mit dem zu entschlüsselnden Hash-Code übereinstimmt. Aufgrund der benötigten Rechenleistung ist dies aber kein realistisches Szenario. Ziel der Verschlüsselung ist es, das zu keiner Zeit außer im RAM des Programms das Passwort in Klartext gespeichert wird. Somit ist es nicht möglich, das Passwort abzurufen oder zu rekonstruieren. Da bei der Authentifizierung immer ein String erwartet wird, der anschließend mit dem Hash-Algorithmus verschlüsselt wird, bringt einem potentiellen Angreifer auch das Abgreifen des schon verschlüsselten Hash-Codes nichts.

In der Netzwerkschnittstelle der Rest-API, über die die Android-Anwendung auf die Datenbank zugreift, wird bei jeder Schnittstellenfunktion eine Authentifizierung verlangt, bevor Daten angenommen oder ausgegeben werden. Die Login-Daten werden im http-Header erwartet. In der Android-Anwendung wird über ein Eingabefeld Login-Name und Passwort eingegeben, das Passwort in den Hashcode verschlüsselt und anschließend bei jeder Anfrage an die Netzwerkschnittstelle im Header mit übergeben.

# Implementierung

## Implementierung des Datenbank-Systems

Im Folgenden sind ausgewählte Klassen des Datenbank-System exemplarisch dokumentiert. Unter dem folgenden Link ist die gesamte Dokumentation abrufbar: <https://kassensystem.github.io/DatabaseSystem/>

### Klasse DatabaseService\_Interface

### Klasse RestApiController

## Implementierung der Android-Anwendung

Im Folgenden sind zwei ausgewählte Klassen der Android-Anwendung exemplarisch dokumentiert. Diese exemplarische Dokumentation ist stark gekürzt. Die gesamte Dokumentation ist unter folgendem Link abrufbar: <https://nunay.github.io/Kassensytem-AndroidApplikation/JavaDoc/>

# Test

## Test des Datenbank-Systems

## Test der Android-Anwendung

# Fazit

# Anhang

## Installationsanweisung

Im Folgenden wird beschrieben, wie man das Datenbank-System auf einem Computer installiert und die Android Applikation auf einem Android fähigem Smartphone installiert.

### Datenbank-System

Die folgenden Schritte müssen durchgeführt werden, um das Datenbank-System auf einem Computer zu installieren und betreiben.

1. Wenn nicht vorhanden, JRE (Java Runtime Environment) installieren.
2. Installation des MySQL Community Servers

Download des MySQL-installers von:

<https://dev.mysql.com/downloads/windows/installer/5.5.html>

Folgende Komponenten installieren:

* MySQL-Server
* MySQL-Workbench
* MySQL-Notifier

1. Imortieren der Datenbank Strukturen

* MySQL-Workbench öffnen: Management -> Data Import/Restore
* "Import from Dump Project Folder":

Die Datei "Datenbank Import/Dump20171128.sql" auswählen

* Im Drop-Down-Menü "Dump Structure Only" auswählen
* Imporieren mit "Start Import"

1. Anlegen eines neuen Users für den Database-Service

* MySQL-Workbench: Management -> Users and Privileges
* Den foglenden User anlegen:
  + Login Name: DatabaseService
  + Password: password
  + Im Tab "Administrative Roles": "DBManager" auswählen
* Anwenden mit "Apply"

1. Installieren des Druckertreibers

* Download des Treibers: <https://download.epson-biz.com/modules/pos/index.php?page=single_soft&cid=5131&pcat=3&scat=31>
* APD\_507\_T88V.exe im Ordner "Druckertreiber APD\_507\_T88V\_EWM" starten
* Installationsanweisungen folgen

1. Download des aktuellsten DatabaseSystems

* <https://github.com/Kassensystem/DatabaseSystem/releases/latest>
* Zum Download auf "Source Code (zip)" klicken
* Entpacken der Dateien

1. Download der aktuellsten ManagerApplication

* <https://github.com/Kassensystem/ManagerApplication/releases/latest>
* Zum Download auf "Source Code (zip)" klicken
* Entpacken der Dateien

1. Anpassen der Firewall

* Um dem Server eine Kommunikation im lokalen Netzwerk zu ermöglichen die folgenden Änderungen durchführen:
  + Windows-Firewall öffnen
  + Firewall komplett deaktivieren

1. Starten des Servers

* In entpackten Dateien des DatbaseSystems:
  + Datei "start.bat" starten
  + Das Kommandozeilenfenster geöffnet lassen
  + Zum Beenden des Servers das Fenster schließen.

1. Starten der ManagerApplication

* In entpackten Dateien der ManagerApplication:
  + Datei "start.bat" starten
  + Oder “kassensystem\_manager.exe” im Pfad: "\out\artifacts\kassensystem\_managerApplication\bundles\kassensystem\_managerApplication"

### Android Applikation

Die folgenden Schritte müssen durchgeführt werden, um die Applikation auf einem Smartphone zu installieren und zu betreiben.

1. Voraussetzung: Betriebssystem: Android, Version > 5
2. Installation von Apps aus unbekannten Quellen zulassen:

*Einstellungen* öffnen 🡪 Unterpunkt *Sicherheit* auswählen 🡪 *Unbekannte* *Herkunft* muss aktivieren

1. Download der APK auf das Smartphone:

<https://github.com/Nunay/Kassensytem-AndroidApplikation>

1. Heruntergeladene Datei öffnen und installieren

Ordner *Dateien* öffnen 🡪 *app-release.apk* öffnen 🡪 installieren

1. Applikation starten

### 

# Testdokumentation

Im Folgenden sind die Tests der öffentlichen Methoden der Software-Module dokumentiert. Es wird jeweils der Anwendungsfall und die getestete Methode beschrieben, außerdem Normalablauf und Sonderfälle, bspw. einem Laufzeitfehler wie Verbindungsprobleme oder falsche Eingaben. Anschließend wird das zu erwartenden und das tatsächliche Testergebnis für den Normalablauf und Sonderfall dokumentiert. In einigen Fällen wurden die Testergebnisse gekürzt.