**Diplomarbeit**

## 📄 **Projektbeschreibung: Wechselgeld-App**

### **Titel der Diplomarbeit:**

**Wechselgeld-App – Digitalisierung von Bargeldwechselvorgängen im stationären Handel**

### **Kurzbeschreibung:**

Ziel des Projekts ist die Entwicklung zweier **(oder nur eine?)** sich ergänzender Android-Apps – eine für Verkäufer:innen und eine für Kund:innen –, die den Umgang mit Bargeld effizienter gestalten. Anstatt Münzen als Wechselgeld auszubezahlen, wird der Betrag digital gutgeschrieben und kann beim nächsten Einkauf eingelöst werden. Die Apps kommunizieren lokal über Bluetooth, speichern Guthaben verschlüsselt auf dem Gerät und optional zusätzlich auf einem Server.

Diese Lösung soll vor allem für Geschäfte mit vielen Stammkunden (z. B. Bäckereien, Kioske, Imbissbuden) eine einfache und bargeldfreundliche Alternative bieten.

## ✅ **Anforderungen an die Wechselgeld-App (Java-Version)**

### 🔧 **Funktionale Anforderungen**

#### 📱 Allgemein:

* Zwei verschiedene Android-Apps: **Verkäufer-App** und **Kunden-App**
* Kommunikation über **Bluetooth** (ohne Server-Abhängigkeit)
* **Offline-Betrieb** möglich (lokale Datenhaltung)

#### 🛍 Verkäufer-App:

* Übersicht der verfügbaren Kunden (Bluetooth-Reichweite)
* Auswahl eines Kunden für die Abrechnung
* Rechnungsbetrag eingeben (manuell oder via QR-Scan)
* Eingabe des erhaltenen Betrags (z. B. 20 €-Schein)
* Automatische Berechnung des neuen Guthabens
* Speicherung des neuen Guthabens (lokal + optional Cloud)
* Anzeige des Backup-Status

#### 👛 Kunden-App:

* Empfang und Anzeige des Rechnungsbetrags
* Anzeige des vorhandenen Guthabens
* Bestätigung der Zahlung und Speicherung des neuen Guthabens
* Übersicht über alle Guthaben bei verschiedenen Geschäften (bilateral)

#### 🔐 Kommunikation:

* Bluetooth-Verbindung zwischen Geräten (Java Android API)
* Sichere Datenübertragung (verschlüsselte JSON-Objekte)
* Identifikation über **UUID + Anzeigename**

#### 💾 Datenhaltung:

* Lokale Speicherung aller Daten in **SQLite** oder mit **Room (Java-kompatibel)**
* Struktur: Kundennamen, UUID, aktuelles Guthaben, Zeitstempel
* **Optionale Sicherung** der Daten auf einem Server (verschlüsselt)

### 🚫 **Nicht-Funktionen**

* Keine Übertragbarkeit von Guthaben zwischen verschiedenen Geschäften
* Keine Transaktionshistorie notwendig
* Kein Online-Zahlungssystem / keine Integration mit Kassensystem
* Keine Gebühren pro Transaktion

**🚀 Projektaufbau & Rollenverteilung**

**👥 Teamgröße: 3 Personen**

**🕒 Projektdauer: 4 Wochen (30.06.2025 – 30.07.2025)**

**👤 Person A – Android Frontend-Entwicklung (Verkäufer- & Kunden-App)**

**Aufgaben:**

* UI mit **XML-Layouts + Java Activities/Fragments**
* Benutzerführung, Interaktionen, Darstellung der Abrechnung
* Bluetooth-Verbindung herstellen und verwalten
* UI für Betragseingabe und Anzeige von Guthaben

**Tools & Frameworks:**

* **Java (Android)**
* **Android Bluetooth API**
* **XML Layouts + AppCompat Components**
* **MVVM oder MVP Pattern (leichtgewichtig)**

**👤 Person B – Datenhaltung, Verschlüsselung & Backup**

**Aufgaben:**

* Speicherung von Guthaben lokal (pro Beziehung: Kunde ↔ Geschäft)
* Lokale Datenbank mit **Room (Java-Support möglich)** oder **SQLite direkt**
* AES-Verschlüsselung der Guthabensdaten
* Server-Backup vorbereiten und (optional) einrichten

**Tools & Frameworks:**

* **SQLite oder Room mit Java**
* **Java Cryptography API (AES/RSA)**
* **Server (optional): Java Spring Boot ODER Node.js + MongoDB**

**👤 Person C – Projektmanagement, Testing & Dokumentation**

**Aufgaben:**

* Projektstruktur, Aufgabenverteilung & Git-Management
* Testing der App (Manuelle UI-Tests & einfache Unit-Tests)
* Screenshots, Ablaufdiagramme, Flussdiagramme, Dokumentation schreiben
* Präsentation & Live-Demo vorbereiten

**Tools:**

* **GitHub** (Projekte, Issues, Versionskontrolle)
* **JUnit** für einfache Logiktests
* **Draw.io / Figma** für Diagramme & Mockups
* **Google Slides / PowerPoint** für Abschlusspräsentation

**🔧 Technologiestack (Java-basiert)**

| **Bereich** | **Tool/Technologie** |
| --- | --- |
| App Entwicklung | Java, Android SDK |
| UI | XML Layouts, AppCompat Views |
| Kommunikation | Bluetooth API (Java) |
| Lokale Speicherung | SQLite (mit oder ohne Room) |
| Verschlüsselung | AES / RSA mit javax.crypto |
| Backup (optional) | Java Spring Boot + MongoDB |
| Versionskontrolle | Git + GitHub/GitLab |

**🔐 Bluetooth-Kommunikation (Java) – Grobablauf**

1. Verkäufer-App scannt nach Geräten (Kunden-Apps)
2. Geräte koppeln sich über Bluetooth
3. Übertragung: UUID + Rechnungsbetrag → Kunde
4. Kunde akzeptiert & Betrag wird verarbeitet
5. Rückmeldung & Speicherung des neuen Guthabens