General Project

**Simple Salesman – Anwendung zur Adress- und Notizverwaltung im D2D-Vertrieb**

Ziel

**Effiziente Verwaltung und Dokumentation von Door-to-door-Verkaufsaktivitäten durch eine leichtgewichtige, maßgeschneiderte Anwendung.**

* Datenimport: Automatisierter wöchentlicher Import von Adress- und Auftragsdaten aus Excel-Dateien in eine PostgreSQL-Datenbank.
* Datenerhalt: Bestehende Datensätze bleiben erhalten und sind pro Adresse mit Notizen erweiterbar.
* DSGVO-konform: Vollständig lokal verarbeitet, mit verschlüsselter Speicherung und Zugriffsschutz.
* Kleines Team, großer Effekt: Optimiert für ein kleines Vertriebsteam (2 Verkäufer) und einen Entwickler.

**Kern-Nutzen**

* Zeitersparnis: Vermeidung manueller Eingabe – Reduktion der Dokumentationszeit um bis zu 75 %.
* Produktivitätssteigerung: Spart dem Vertriebsteam (2 Personen) ca. 4 Stunden pro Woche.
* Kostenersparnis: Bei 25 €/Stunde sind das jährlich über 5.000 €.
* Datenqualität: Einheitliche, strukturierte Datenbasis für Auswertungen und gezielte Folgeaktionen.
* Transparenz & Kontrolle: Direkter Überblick über Besuchs Status, Notizen, Aufträge.
* KI-Einsatz (optional): Strukturierte Textvorschläge durch ein kleines, lokal eingebettetes Modell.

**Repräsentativer Entwickler Wert**

**Diese Software zeigt praxisnahe Kompetenz in:**

* Full-Stack-Entwicklung (Spring Boot, PostgreSQL, REST-API, HTML/JS)
* Automatisierung von Geschäftsprozessen im Kleinbetrieb
* Umsetzung gesetzlicher Anforderungen (DSGVO, KI Act)
* Effiziente Architektur und Code-Struktur auch für kleine Teams

Kern Funktionen

* Import von Excel-Dateien (wöchentlich)
* Speicherung und Abgleich der Daten in PostgreSQL
* Notizverwaltung pro Adresse
* Web-Frontend zur Auswahl, Anzeige und Bearbeitung
* Texteditor für die Felder
* Export von Excel-Dateien (Backup)
* Night Mode
* Modernes Design (ohne Frameworks)
* Wetter für die ausgewählte Region per REST
* Optional: Easter Eggs - Motivation Balken
* Optional: Ai Text editor - Jira als Beispiel

Technology Stack

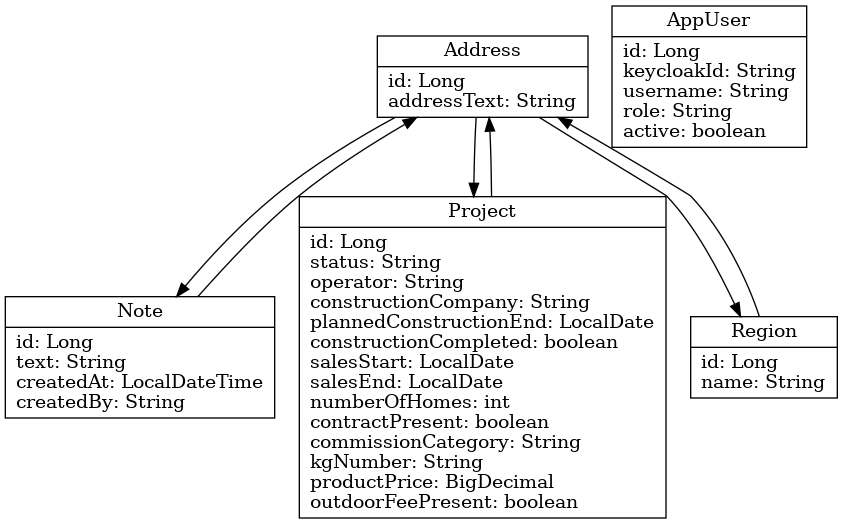
* **Frontend:** HTML, CSS, JavaScript
* **Backend:** Spring Boot (REST-API, Datenimport via Apache POI)
* **Datenbank:** PostgreSQL (verschlüsseln - inklusive Hardware)
* **Authentifizierung:** [**OAuth 2.0 authorization using Keycloak**](https://www.redhat.com/en/blog/oauth-20-authentication-keycloak)

Systemarchitektur (synchron oder asynchron)

* **Frontend:** UI zur Anzeige von Adressen & Notizen, Datei-Upload, API-Nutzung
* **Backend:** Datenimport, REST-API, Geschäftslogik, Kommunikation mit PostgreSQL
* **Datenbank:** Strukturierte Speicherung aller Daten mit normalisierten Relationen

**Entitäten und Beziehungen (vereinfacht)**

* **Region → Adresse**: 1:N
* **Adresse → Projekt**: 1:N
* **Projekt → Status, Betreiber, Baufirma**: jeweils 1:N



**Software Struktur**

| src/  └── main/  ├── java/  │ └── com/  │ └── simple salesman/  │ ├── SimpleSalesmanApplication.java  │ ├── config/ # Security, CORS, Swagger-Konfiguration  │ ├── controller/ # REST-Controller  │ ├── service/ # Geschäftslogik  │ ├── repository/ # JPA-Repositories  │ ├── entity/ # JPA-Entities  │ ├── dto/ # Data Transfer Objects  │ └── util/ # Helfer Klassen (z. B. Excel-Import, Mapper)  └── resources/  ├── application.properties # Konfiguration (DB, Security, Logging)  ├── static/ # Statische Inhalte (CSS, JS, Icons)  ├── templates/ # HTML (für Thymeleaf oder statisch)  ├── schema.sql # Optional: Initiales DB-Schema  ├── data.sql # Optional: Beispieldaten  └── banner.txt # Optional: CLI-Start Banner |
| --- |

**Ablauf**

1. Benutzer lädt Excel-Datei hoch
2. Backend verarbeitet und gleicht Daten mit PostgreSQL ab
3. Frontend zeigt aktuelle Daten interaktiv an
4. Notizen können ergänzt, bearbeitet oder gespeichert werden

Weitere Aspekte

* **Testen:** Unit- und Integrationstests mit JUnit
* **Dokumentation:** API- und Benutzerdokumentation
* **Planung & Management:** Git, Gantt, CI/CD, Refactoring
* **Werkzeuge:** draw.io/Lucidchart für ERM und UML, GitHub für Versionierung
* **Umgebung:** Zielumgebung definieren

**Stakeholder Analyse:**

1. Anwender - Verkäufer (funktionale anforderungen)
2. Admin - Support funktionen für den Verkäufer
3. DSGVO - Gesetzeslage und der Verantwortlicher
4. KI Act - Gesetzeslage und der Verantwortlicher (lokales Modell)
5. Lizensen - Apache Lizenz beachten (sich an diese halten)

**Links:**

* <https://www.fb.gv.at/Info/rtr.html>
* <https://artificialintelligenceact.eu/>
* <https://www.rtr.at/>
* <https://www.wko.at/vlbg/information-consulting/kuenstliche-intelligenz>
* <https://www.wko.at/datenschutz/uebersicht>

**Funktionale Anforderungen:**

**USER - Anwender funktionen**

AF1: Als Verkäufer möchte ich eine Excel-Datei hochladen können, um meine Adressdaten effizient zu importieren.

AF2: Als Verkäufer möchte ich meine Adressen und Projekte anzeigen können, um meine Vertriebsaktivitäten zu überblicken.

AF3: Als Verkäufer möchte ich zu jeder Adresse Notizen hinzufügen, bearbeiten und speichern können.

AF4: Als Verkäufer möchte ich den Status eines Projekts aktualisieren können, um den Fortschritt nachvollziehen zu können.

AF5: Als Verkäufer möchte ich Adressen und Projekte filtern und durchsuchen können, um gezielt Informationen zu finden.

AF6: Als Verkäufer möchte ich meine Daten als Excel-Datei exportieren können, um ein Backup zu erstellen.

AF7: Als Verkäufer möchte ich das aktuelle Wetter für eine bestimmte Region aufrufen können, um meine Routen besser zu planen.

AF8: Als Verkäufer möchte ich optional KI-generierte Textvorschläge erhalten, um meine Notizen effizienter zu formulieren.

**ADMIN - Admin Support funktionen:**

AD1: Als Admin möchte ich Benutzerkonten verwalten können, um den Zugang zur Anwendung zu kontrollieren.

AD2: Als Admin möchte ich Fehlerprotokolle einsehen können, um Probleme schnell zu identifizieren.

AD3: Als Admin möchte ich fehlerhafte Datenimporte rückgängig machen können, um Datenkorrektheit sicherzustellen.

AD4: Als Admin möchte ich System-Backups und Wiederherstellungen durchführen können.

AD5: Als Admin möchte ich eine Übersicht über alle Verkäufer und ihre Daten erhalten.

AD6: Als Admin möchte ich gesetzlich relevante Informationen verwalten können, z. B. zur DSGVO.

**DSGVO – Datenschutzanforderungen**

DS1: Die Anwendung muss alle Zugriffe und Änderungen an personenbezogenen Daten protokollieren.  
DS2: Alle personenbezogenen Daten müssen verschlüsselt gespeichert werden.  
DS3: Nutzer dürfen nur auf ihre eigenen Daten zugreifen können.  
DS4: Personenbezogene Daten müssen auf Anfrage vollständig gelöscht werden können (Recht auf Vergessenwerden).  
DS5: Die Anwendung muss eine Datenschutzerklärung anzeigen und die Einwilligung der Nutzer dokumentieren.

**KI Act – Anforderungen an den KI-Einsatz**

KI1: KI-generierte Texte müssen als solche gekennzeichnet werden.  
KI2: Die Nutzung von KI-Funktionen muss abschaltbar sein (Opt-out).  
KI3: Die KI darf keine personenbezogenen Daten speichern oder weiterverarbeiten.  
KI4: Verwendete KI-Modelle müssen dokumentiert sein, inkl. Zweck und Versionsnummer.

**Lizenz-Anforderungen – Apache License 2.0**

LI1: Ein Lizenzhinweis zur Apache License muss im Quellcode enthalten sein.  
LI2: Copyright-Hinweise externer Bibliotheken müssen übernommen werden.  
LI3: Alle verwendeten Open-Source-Komponenten müssen dokumentiert werden.  
LI4: Änderungen am Code müssen in einem Änderungsverlauf (CHANGELOG.MD) nachvollziehbar dokumentiert werden.

***Verpflichtende und optionale Anforderungen***

***Muss: AF1–AF4, DS1–DS5, KI1–KI4, LI1–LI4, NF1–NF5  
Kann: AF5, AF6, AF7, AD1–AD5  
Wunsch: AF8, AD6***

***PM:***

***Marktanalyse und Benchmarking***

* *Vergleich mit bestehenden Lösungen der Konkurrenz*
* *Identifikation potenzieller Alleinstellungsmerkmale (z. B. DSGVO-/KI-Compliance, einfache Bedienung, Geschwindigkeit)*

***DEV:***

* *Performance fixes*
* *Deployment config manual*
* *Feedback einholen und zur Optimierung der Anwendung nutzen*
* *UML (Unified Modeling Language) - erstellen*

**Zeiten:**

* Davor - 10 h
* 28.05 - 4 h
* 29.05 - 6 h
* 31.05 - 3 h
* 09.06 - 5 h
* 15.06 - 4 h
* 16.06 - 4 h
* 17.06 - 6 h
* 18.06 - 7 h
* 20.06 - 2 h
* 21.06 - 3 h