

LF08 – Datenbank-Projekt Nr. 2 mit C#

1. Projekt-Beschreibung

Dieses LF08-Projekt hat zum Inhalt, einen Teil des Prozesses zur Einführung des vom Kreis Steinburg geplanten elektronischen Identifikationssystems für Abfallbehälter in Ansätzen nachzubauen. Im Infoblatt des Kreises werden die Ziele des Projekts grob wie folgt beschrieben¹:

- *Von Anfang März bis Mitte Juli diesen Jahres werden alle Restabfall-, Bio- und Papierbehälter, die im Kreis Steinburg stehen, mit einem elektronisch auslesbaren Identifikationssystem (Chip) ausgerüstet.*
- *Mit dem neuen System wird es zukünftig möglich sein, die jeweiligen Abfalltonnen einem Grundstück genau zuzuordnen und festzustellen, ob die Anzahl und Größe der Behälter vor Ort stimmen. Die ordnungsgemäße Leerung wird dokumentiert und eventuell auftretende Probleme werden vermerkt.*
- *Mit dem Identifikationssystem werden Arbeitsabläufe effektiver und transparenter. Entsorgungstouren lassen sich besser planen und der Behälterbestand lässt sich besser verwalten. Auf dem Chip sind keine personenbezogenen Daten gespeichert.*

Die sog. „Verchipung“ der Abfallbehälter wird wie folgt beschrieben:

- *Behälter, die bereits in Gebrauch sind, werden direkt am Grundstück mit einem Chip ausgerüstet.*
- *Im Rahmen der Chipausrüstung wird ein Transponder-Chip in den unteren Rand der Tonne eingesetzt und ein Behälteretikett auf die linke Seite des Abfallbehälters geklebt.*
- *Nachdem der Chip angebracht wurde, bekommt die Tonne das Behälteretikett. Das Etikett enthält die Adresse, die Behältergröße, die Abfallart, eine Behälternummer und einen Barcode. Mithilfe des Behälteretiketts, das dauerhaft auf den Tonnen bleibt, findet über einen Strichcode eine Verknüpfung des Transponders mit dem Grundstück statt. Sobald das Etikett angebracht wurde, ist die Chipausrüstung abgeschlossen.*

Aufgrund dieser Beschreibungen kann davon ausgegangen werden, dass für das Identifikationssystem eine eigene Datenbank neu entwickelt werden muss, in der neben den eigentlichen Abfallbehältern auch die Entleerungen usw. gespeichert werden können.

Es kann weiter davon ausgegangen werden, dass dem Kreis Steinburg ein sog. ERP-System² (SAP o.ä.) zur Verfügung steht, mit dem u.a. die Kundendaten verwaltet und die jährlichen Abfallrechnungen (s.u.) erstellt werden. Dieses ERP-System wird dafür benutzt, die für die Verwaltung der Abfallbehälter benötigten Informationen in Dateien zu exportieren (jedes ERP-System beinhaltet eine entsprechende Funktion). Diese exportierten Informationen werden dann in einem ETL-Prozess in das neue Datenbanksystem importiert und bilden damit die Basis der folgenden „Verchipung“.

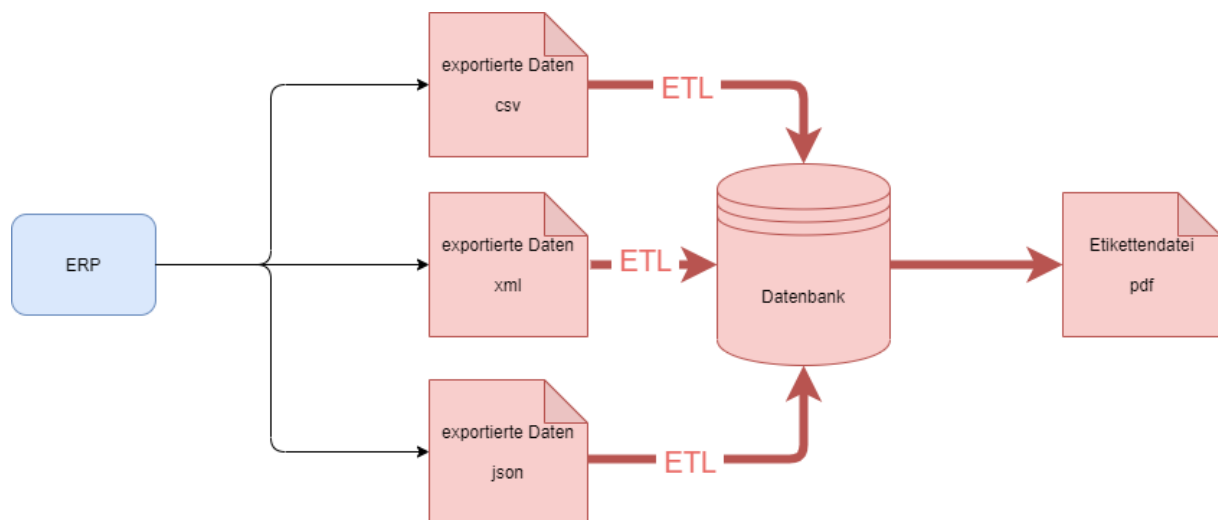
¹ S. Infoblatt vom Kreis

² <https://www.datenbanken-verstehen.de/erp/erp-grundlagen/erp-systeme-praxis/>

Aus dieser Datenbank werden dann automatisch die Etiketten erstellt, die dann auf die Abfallbehälter geklebt werden, wenn die „Verchipung“ stattfindet. In diesem Projekt werden die Etiketten als pdf generiert, die danach mit Hilfe eines Etikettendruckers ausgedruckt werden könnten.



Der ganze Ablauf lässt sich wie folgt graphisch zusammenfassen, wobei die rot dargestellten Elemente die Prozessteile sind, die in diesem LF08-Projekt umgesetzt werden sollen:



2. Exportierte Daten vom ERP-System

Die vom ERP-System erzeugten Rechnungen sehen wie folgt aus:

Itzehoe, 04.12.2020

Objekt-Nr.: 3.028.1345.021

Kunden-Nr.: 123456

Jahreserstbescheid über die Abfallgebühr 2021

Grundstück: A-Weg 0, 12345 Teststadt

Sehr geehrte Damen und Herren,
für das o. g. Grundstück werden auf der Grundlage der Satzung über die Erhebung von
Gebühren für die Abfallentsorgung im Kreis Steinburg in der zurzeit geltenden Fassung
folgende Gebühren festgesetzt. Die Gebühren gelten vorbehaltlich der Beschlussfassung
durch den Kreistag am 17.12.2020.

Monate (Anzahl)	Berechnungs- grundlagen	Anzahl / Menge	Gebühr pro Monat	Jahresgebühr gesamt
Jan. - Dez. (12)	Grundgebühr Wohnung	1	4,90 €	58,80 €
Jan. - Dez. (12)	120l Biotonne	1	6,54 €	78,48 €
Jan. - Dez. (12)	240l Papiertonne	1		
Festsetzungsbetrag:				137,28 €

In diesem Projekt wird daher davon ausgegangen, dass das ERP-System die Daten als csv-Dateien exportiert, wie sie beispielhaft dargestellt ist:

```
obj_nr;ob_kunde_nr;obj_str;obj_haus_nr;obj_plz;obj_ort;abfallart;volumen;anzahl
2.2345.21.234;346665;Aweg;1;11111;Astadt;Bio;120;1
2.2345.21.234;346665;Aweg;1;11111;Astadt;Papier;240;1
2.2345.21.235;346665;Aweg;2;11111;Astadt;Bio;120;1
2.2345.21.235;346665;Aweg;2;11111;Astadt;Papier;240;1
3.4665.34.323;567432;Bweg;2;22222;Bstadt;Bio;80;2
3.4665.34.323;567432;Bweg;2;22222;Bstadt;Restmüll;120;1
3.4665.34.323;567432;Bweg;2;22222;Bstadt;Papier;240;1
```

Diese Dateien müssen durch einen ETL-Prozess automatisiert in die Zieldatenbank importiert werden können.

3. Zieldatenbank

Wie bereits erwähnt, soll für das Verwaltungssystem eine neue Datenbank entwickelt werden. In diesem Projekt soll nur das eigentliche Verwalten der Abfallbehälter und deren „Verchipung“ berücksichtigt werden. Zukünftige Aspekte wie das Verwalten der einzelnen Leerungen oder das Planen von Routen der Müllwagen sollen hier nicht berücksichtigt werden.

Für die zu implementierende SQLite-Datenbank gilt, dass sie in der dritten Normalform vorliegen muss.

4. Etikettendateien

Wie bereits erwähnt, wird für die „Verchipung“ der einzelnen Abfallbehälter pro Behälter ein Etikett benötigt, das die Objektnummer, die Objektadresse, die Behältergröße, die Abfallart, die Behälternummer und einen Barcode mit der Behälternummer enthält.

Für jedes Objekt soll eine pdf-Datei erzeugt werden, die die Informationen aller Abfallbehälter eines Objekts wie folgt beispielhaft dargestellt enthält:

Objekt-Nr: 2.2345.21.234
Straße: AWeg
Nr: 1
PLZ: 11111
Ort: Astadt



Tonnen-Nr: 6456785
Abfallsorte: Bio
Volumen: 120

Objekt-Nr: 2.2345.21.234
Straße: AWeg
Nr: 1
PLZ: 11111
Ort: Astadt



Tonnen-Nr: 6456786
Abfallsorte: Papier
Volumen: 240

Das eigentliche „Verchipen“, also das Einscannen der barcodes und die Zuordnung zu den Ids der Tranponder an den Abfallbehältern wird in diesem Projekt nicht berücksichtigt, sondern es wird hier lediglich die ganze Vorarbeit zum „Verchipen“ geleistet.

5. Anforderungsbeschreibung

Die oben beschriebenen Vorgänge sollen automatisiert ablaufen, wofür neben der Zieldatenbank ein entsprechendes Konsolen-Programm mit dem Namen ksa.exe zu entwickeln ist.

Generell gilt: Wird das Programm auf andere Weise aufgerufen als hier beschrieben ist, ist dies ein fehlerhafter Programmaufruf. Es erfolgt dann eine entsprechende Meldung.

Die in die Datenbank zu importierenden Dateien können sich im selben Verzeichnis wie die Datei ksa.exe befinden. Eine Berücksichtigung einer Pfadangabe muss nicht implementiert werden.

Die Datenbankstruktur ist komplett selbst zu entwickeln. Die Datenbank befindet sich im selben Verzeichnis wie das Programm, d.h. Pfadangaben zur Datenbank müssen nicht

berücksichtigt werden. Das Programm muss die Datenbank nicht selbst erstellen. Wird das Programm zum ersten Mal benutzt, befindet sich die (noch leere) Datenbank bereits im Programmverzeichnis.

5.1 Grundanforderung – Bereitstellen der Daten in einer Datenbank

Die Grundanforderung besteht im Wesentlichen darin, dass die Software die Daten in den csv-Dateien in eine Datenbank importiert und für die dann folgende Etiketten-Erzeugung bereitstellt.

Es muss möglich sein, die Namen der Projektgruppenmitglieder auszugeben:

Namenausgabe	Gibt einfach die Namen aller Projekt-Beteiligten aus.
Aufruf	ksa.exe /n
Beschreibung	/n – steht für names
Beispiel	betl.exe /n Hansi Müller – Max Mustermann – Tony Test

Das Importieren der csv-Dateien erfolgt folgendermaßen:

Import	Import aller csv-Dateien im aktuellen Verzeichnis in die Datenbank.
Aufruf	ksa.exe /csvimp
Beschreibung	/csvimp – steht für Import
Beispiel	ksa.exe /csvimp data.csv erfolgreich importiert!

5.2 Etiketten-Generierung

Etiketten	Generierung aller pdf-Dateien
Aufruf	ksa.exe /etk
Beschreibung	/etk – steht für Etikett
Beispiel	ksa.exe /etk Alle pdf-Dateien werden im aktuellen Verzeichnis gespeichert. Eine pdf-Datei pro Objekt. Der Dateiname ist die entsprechende Objektnummer.

Für das Generieren der pdf-Dateien wird die Bibliothek itext7³ empfohlen, die per NuGet-Manager wie bekannt dem VS-Projekt hinzugefügt werden muss. Der unten angegebene link führt zur entsprechenden Seite, die sehr reich ist an Beispielcode. Diese Bibliothek bietet u.a. das Einbinden von vorhandenen barcodes als gif-Dateien an. Die barcode-Dateien können z.B. mit der Bibliothek NetBarcode erstellt werden, die ebenfalls per NuGet hinzugefügt werden muss. Ein anderer Weg ist es, mit itext7 selbst die barcodes generieren zu lassen und in das pdf direkt einzubauen⁴. Es kann aber auch jede andere Bibliothek verwendet werden.

³ <https://kb.itextpdf.com/home/it7kb/ebooks/itext-7-jump-start-tutorial-for-net/chapter-1-introducing-basic-building-blocks-net>

⁴ <https://stackoverflow.com/questions/54004849/itext7-for-net-barcode>

5.3 XML-Import-Erweiterung

Neben reinen csv-Dateien können die Daten auch im xml-Format aus dem ERP-System exportiert werden.

Import als xml	Import aller xml-Dateien im aktuellen Verzeichnis in die Datenbank.
Aufruf	ksa.exe /xmlimp
Beschreibung	/xmlimp – steht für Import von xml-Dateien
Beispiel	ksa.exe /xmlimp data.xml erfolgreich importiert!

Die hier zu importierenden xml-Dateien haben folgende Struktur:

```
<?xml version="1.0"?>
<entries>
  <entry>
    <objekt_nr>2.2345.21.234</objekt_nr>
    <kunden_nr>346665</kunden_nr>
    <obj_strasse>Aweg</obj_strasse>
    <obj_haus_nr>1</obj_haus_nr>
    <obj_plz>11111</obj_plz>
    <obj_ort>A</obj_ort>
    <abfallart>Bio</abfallart>
    <volumen>120</volumen>
    <anzahl>1</anzahl>
  </entry>
  <entry>
    <objekt_nr>2.2345.21.234</objekt_nr>
    <kunden_nr>346665</kunden_nr>
    <obj_strasse>Aweg</obj_strasse>
    <obj_haus_nr>1</obj_haus_nr>
    <obj_plz>11111</obj_plz>
    <obj_ort>A</obj_ort>
    <abfallart>Papier</abfallart>
    <volumen>240</volumen>
    <anzahl>1</anzahl>
  </entry>
  ...
</entries>
```

5.4 JSON-Import-Erweiterung

Neben reinen csv-Dateien können die Daten auch im json-Format aus dem ERP-System exportiert werden.

Import als json	Import aller json-Dateien im aktuellen Verzeichnis in die Datenbank.
Aufruf	ksa.exe /jsonimp
Beschreibung	/jsonimp – steht für Import von json-Dateien
Beispiel	ksa.exe /jsonimp data.xml erfolgreich importiert!

Die hier zu importierenden json-Dateien haben folgende Struktur:

```
[
  {
    "objekt_nr": "2.2345.21.234",
    "kunden_nr": "346665",
    "obj_strasse": "Aweg",
    "obj_haus_nr": "1",
    "obj_plz": "11111",
    "obj_ort": "A",
    "abfallart": "Bio",
    "volumen": "120",
    "anzahl": "1"
  },
  {
    "objekt_nr": "2.2345.21.234",
    "kunden_nr": "346665",
    "obj_strasse": "Aweg",
    "obj_haus_nr": "1",
    "obj_plz": "11111",
    "obj_ort": "A",
    "abfallart": "Papier",
    "volumen": "240",
    "anzahl": "1"
  },
  ...
]
```

6. Abgabe

Das Projekt ist als Verzeichnis mit der **auf Schulrechnern(!!) lauffähigen** Datei ksa.exe sowie allen benötigten Dateien der verwendeten Bibliotheken und der leeren Datenbank abzugeben.

Dieses Verzeichnis wird zum Testen des Programms verwendet!
Zusätzlich dazu ist das VisualStudio-Projekt abzugeben.

Das Projekt wird nicht alleine bearbeitet.

**Die Abgabe erfolgt über den moodle-Kurs am: 07.06.2022
zum Ende der Unterrichtsstunde um: 13:10 Uhr.**