

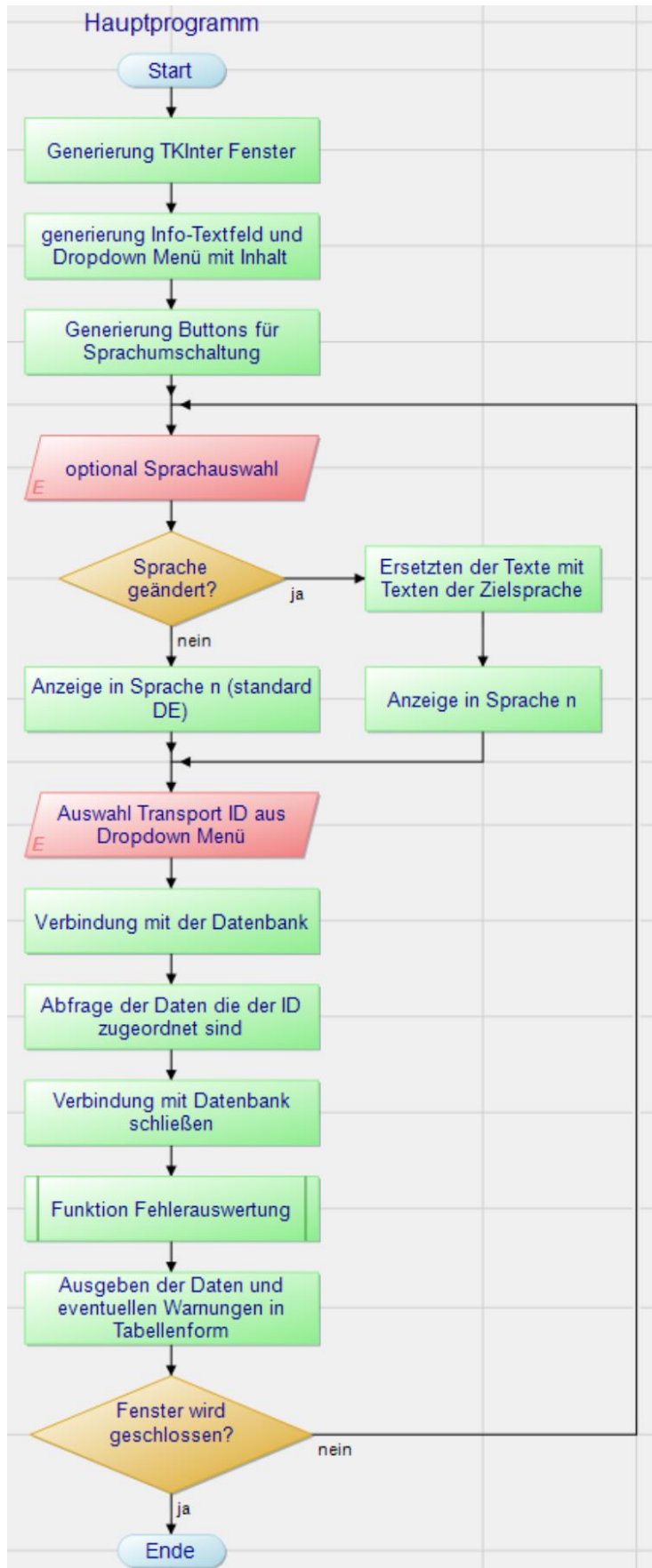


**BILDUNGSZENTRUM
TECHNIK UND GESTALTUNG
OLDENBURG**

Coolchain – Projekt

PROJEKTVORSTELLUNG DER KÜHLKETTENÜBERWACHUNG
CHRISTIAN V., RIKO, LARS, HICHAM, DAVID

Programmablaufplan Hauptprogramm



Importierte Bibliotheken

```
import pyodbc
import customtkinter as ctk
from tkinter import messagebox
from datetime import timedelta, datetime
```

Pyodbc ermöglicht den Datenbankzugriff.

Verwendete Funktionen aus der Bibliothek sind:

.connect()	Verbindung
.cursor()	Beschreibt den Fortschritt während der Abfrage
.fetchall()	Alle Daten laden
.close()	Verbindung schließen
.Error()	

Customtkinter ist eine Erweiterung von Tkinter. Diese Bibliothek ermöglicht die Generierung von simplen Benutzeroberflächen.

Verwendete Funktionen aus der Bibliothek sind:

- .CTkFrame()
- .CTkLabel()
- .CTkOptionMenu()
- .CTkButton()
- messagebox.

Verbindung mit der Datenbank

```
# Def Verbindung Datenbank
def fetch_data():
    transport_id = dropdown_transport_id.get()
    if not transport_id:
        messagebox.showerror(lang["Fehler"], lang["Bitte eine valide Transport ID eingeben."])
        return

    try:
        conn = pyodbc.connect(conn_str)
        cursor = conn.cursor()
        cursor.execute('SELECT transportstation, category, direction, datetime FROM coolchain1 WHERE transportid = ?', (transport_id,))
        results = cursor.fetchall() # results: hier eine Matrix aller Daten
        display_results(results, transport_id)
    except pyodbc.Error as e: #Fehlerabfang, Meldung im Fenster ausgeben
        messagebox.showerror(lang["Fehler bei Datenbankzugriff. Netzwerkverbindung prüfen."], str(e))
    finally:
        if conn:
            conn.close()
```

Es wird eine Verbindung zur Datenbank hergestellt, die enthaltenen Daten werden in der Variable 'results' hinterlegt anschließend wird die Verbindung getrennt.

Anzeigetabelle erzeugen

```
# Def Daten Anzeigen
def display_results(results, transport_id):

    for widget in frame_results.winfo_children():
        widget.destroy()

    if results:

        headers = [lang["Ort"], lang["Kategorie"], lang["Richtung"], lang["Zeitstempel"], lang["Dauer"], lang["Warnung"]]
        for i, header in enumerate(headers):
            label = ctk.CTkLabel(frame_results, text=header, font=("Arial", 12, "bold"))
            label.grid(row=0, column=i, padx=10, pady=5)
```

Hier wird die Anzeigetabelle mit sechs Spalten aufgesetzt und die Schriftart und Größe festgelegt. Diese Tabelle wird später im Programmfenster angezeigt.

Programmfenster aufsetzen

```
# Sprache bei Start
lang = LANGUAGES["DE"]

# UI Fenster aufsetzen: Dark mode, blaue Akzente, Name, Größe
ctk.set_appearance_mode("Dark")
ctk.set_default_color_theme("blue")
root = ctk.CTk()
root.title("Kühlketten Überwachung")
root.geometry("1200x700")

# Überschrift Eingabebox Knopf
label_transport_id = ctk.CTkLabel(root, text=lang["Transport ID eingeben:"], font=("Arial", 14))
label_transport_id.pack(pady=(60, 10))
dropdown_transport_id = ctk.CTkOptionMenu(
    root,
    values=transport_ids,
    font=("Arial", 12),
    width=600,
    fg_color="black", # Hintergrundfarbe des drop down menu
    button_color="black", # Farbe des Schalters
    button_hover_color="darkgray", # Farbe wenn über den Schalter/Knopf gefahren wird
    text_color="white", # Textfarbe
    command=lambda choice: fetch_data() # Direkte Aufruf der Funktion nach Auswahl
)
dropdown_transport_id.pack(pady=(10, 20))

# Rahmen für Liste
frame_results = ctk.CTkFrame(root, width=860, height=300)
frame_results.pack(pady=20, padx=20, fill="both", expand=True)

# Knöpfe für Lokalisierung
button_language_1 = ctk.CTkButton(root, text="EN", command=set_english, width=50)
button_language_1.place(relx=1.0, rely=0.0, anchor="ne", x=-80, y=20)

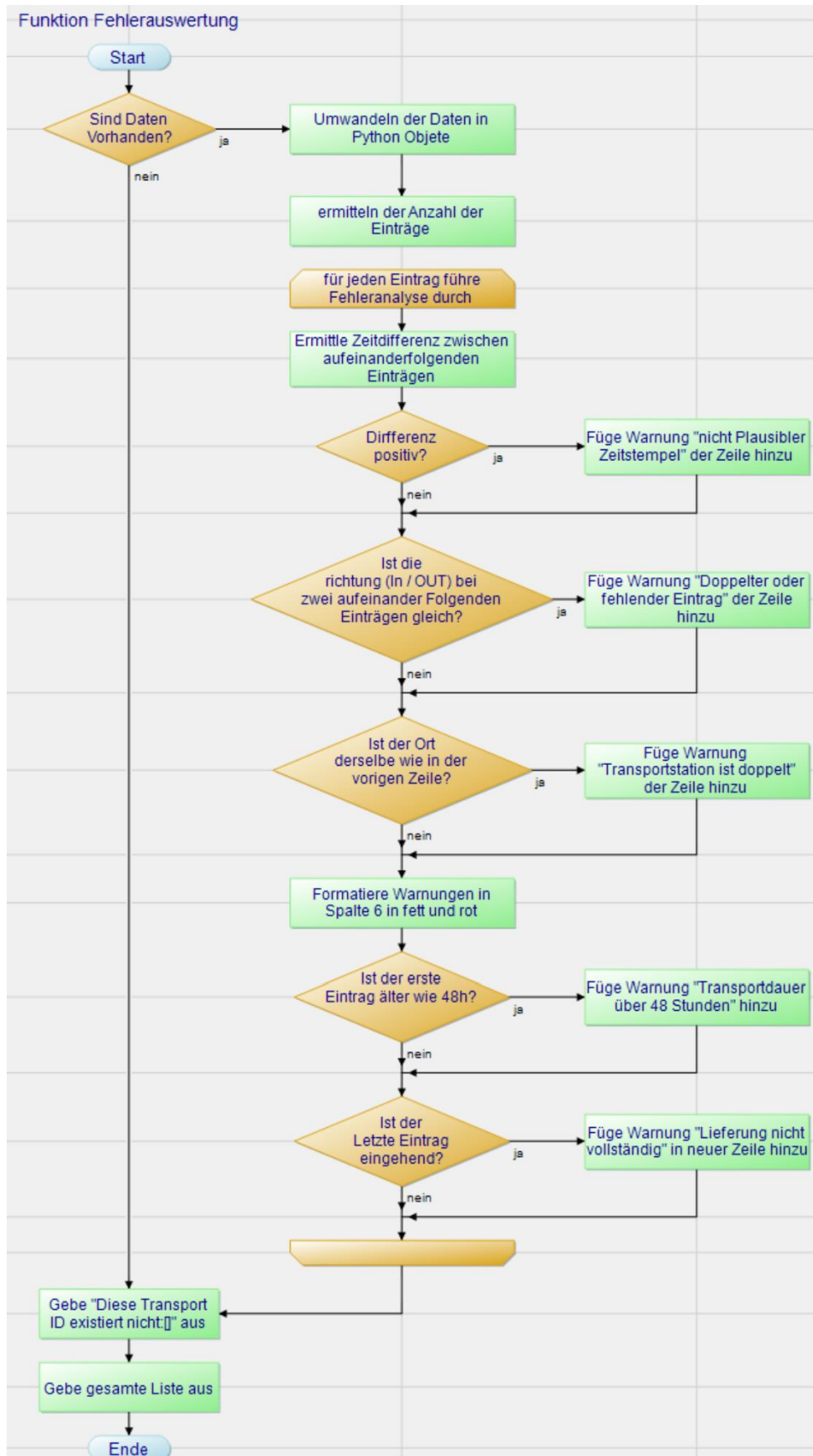
button_language_2 = ctk.CTkButton(root, text="AR", command=set_arabic, width=50)
button_language_2.place(relx=1.0, rely=0.0, anchor="ne", x=-20, y=20)

# Programmstart
root.mainloop()
```

Hier wird das Anzeigefenster aufgesetzt, es wird die Sprache beim Start festgelegt dann wird das Programmfenster definiert (Größe in px*px, Hintergrundfarbe, Überschrift etc.). Im zweiten Teil wird ein Dropdownmenü mit allen möglichen ID's erstellt.

Dann werden die Buttons zur Sprachwahl konfiguriert. Zum Schluss 'root.mainloop()' damit das Programmfenster offen bleibt (ähnlich zu 'while TRUE').

Programmablaufplan Fehlerauswertung



Fehlerauswertung

```
previous_datetime = None
previous_direction = None
first_datetime = results[0][3]
last_datetime = None
previous_location = None

for row_index, row in enumerate(results, start=1):
    transportstation, category, direction, current_datetime = row

    last_datetime = current_datetime
    last_direction = direction
    warnung = " "

    if previous_datetime:
        time_difference = current_datetime - previous_datetime
        time_diff_str = str(time_difference)

        if time_difference.total_seconds() < 1:
            warnung = lang["Nicht plausibler Zeitstempel"]

        if direction == "'in'" and time_difference > timedelta(minutes=10):
            warnung = lang["Übergabezeit über 10 Minuten"]

    else:
        time_diff_str = "N/A"
```

Dies ist die Fehlerüberprüfung auf plausible Zeiten. Diese dient als Beispiel für die gesamte Fehlerauswertung.

Oben im Bild werden leere Variablen erstellt. Diese werden für jeden Durchlauf mit den Werten aus der Datenbank gefüllt. Es werden die Variablen als sortierte Tabelle ausgegeben. Jede Zeile wird mit der vorherigen Zeile verglichen und anhand der Berechnungen auf Fehler im Ablauf der Kühltaste überprüft.

Tritt ein Fehler auf, wird dieser der Zeile mit Fehler hinzugefügt.

Sprachumschaltung

```
# lokalisierung DE
def set_german():
    global lang
    lang = LANGUAGES["DE"]
    update_gui_language()
```

Die Sprachumschaltung hat drei Funktionen hinterlegt. Für jede Sprache eine. Die Funktion wird ausgeführt, wenn man im Programmfenster auf den Button für eine andere Sprache klickt.

```
def update_gui_language():
    label_transport_id.configure(text=lang["Transport ID eingeben:"])

    if lang == LANGUAGES["DE"]:
        button_language_1.configure(text="EN", command=set_english)
        button_language_2.configure(text="AR", command=set_arabic)
```

Als erstes wird dann das Fenster aktualisiert damit die Buttons der beiden anderen Sprachen gezeigt werden und der Text über dem Eingabefenster übersetzt ist.

```
# hinterlegung der Sprachen
LANGUAGES = {
    "DE": {
        "Transport ID eingeben:": "Transport ID eingeben:",
        "Daten prüfen": "Daten prüfen",
        "Ort": "Ort",
        "Kategorie": "Kategorie",
        "Richtung": "Richtung",
        "Zeitstempel": "Zeitstempel",
        "Dauer": "Dauer",
        "Warnung": "Warnung",
        "Nicht plausibler Zeitstempel": "Nicht plausibler Zeitstempel",
        "Übergabezeit über 10 Minuten": "Übergabezeit über 10 Minuten",
        "Doppelter oder fehlender Eintrag": "Doppelter oder fehlender Eintrag",
        "Transportstation ist doppelt": "Transportstation ist doppelt",
        "Transportdauer über 48 Stunden": "Transportdauer über 48 Stunden",
        "Diese Transport ID existiert nicht: ": "Diese Transport ID existiert nicht: ",
        "Fehler": "Fehler",
        "Bitte eine valide Transport ID eingeben.": "Bitte eine valide Transport ID eingeben.",
        "Fehler bei Datenbankzugriff. Netzwerkverbindung prüfen.": "Fehler bei Datenbankzugriff. Netzwerkverbindung prüfen.",
        "Lieferung nicht vollständig. Zeit seit letztem Eintrag: ": "Lieferung nicht vollständig. Zeit seit letztem Eintrag: "
    },
    "EN": {
        "Transport ID eingeben:": "Enter Transport ID:",
        "Daten prüfen": "Check Data",
        "Ort": "Location",
        "Kategorie": "Category",
        "Richtung": "Direction",
        "Zeitstempel": "Timestamp",
        "Dauer": "Duration",
        "Warnung": "Warning",
        "Nicht plausibler Zeitstempel": "Not plausible timestamp",
        "Übergabezeit über 10 Minuten": "Handover time over 10 minutes",
        "Doppelter oder fehlender Eintrag": "Duplicate or missing entry",
        "Transportstation ist doppelt": "Transport station is duplicate",
        "Transportdauer über 48 Stunden": "Transport duration over 48 hours",
        "Diese Transport ID existiert nicht: ": "This Transport ID does not exist: ",
        "Fehler": "Error",
        "Bitte eine valide Transport ID eingeben.": "Please enter a valid Transport ID.",
        "Fehler bei Datenbankzugriff. Netzwerkverbindung prüfen.": "Error in database access. Check network connection.",
        "Lieferung nicht vollständig. Zeit seit letztem Eintrag: ": "Delivery not complete. Time since last entry: "
    },
    "AR": {
        "Transport ID eingeben:": "ادخل رقم تعريف النقل:",
        "Daten prüfen": "تحقق البيانات",
        "Ort": "المكان",
        "Kategorie": "الفئة",
        "Richtung": "الاتجاه",
        "Zeitstempel": "الختم الزمني",
        "Dauer": "المدة",
        "Warnung": "تحذير",
        "Nicht plausibler Zeitstempel": "الختم الزمني غير معقول",
        "Übergabezeit über 10 Minuten": "وقت التسليم أكثر من 10 دقائق",
        "Doppelter oder fehlender Eintrag": "تكرار أو سجل مفقود",
        "Transportstation ist doppelt": "محطة النقل مكررة",
        "Transportdauer über 48 Stunden": "مدة النقل أكثر من 48 ساعة",
        "Diese Transport ID existiert nicht: ": "هذا رقم تعريف النقل غير موجود: ",
        "Fehler": "خطأ",
        "Bitte eine valide Transport ID eingeben.": "الرجاء إدخال رقم تعريف النقل صالح.",
        "Fehler bei Datenbankzugriff. Netzwerkverbindung prüfen.": "خطأ في الوصول إلى قاعدة البيانات. تحقق من اتصال الشبكة.",
        "Lieferung nicht vollständig. Zeit seit letztem Eintrag: ": "التسليم غير مكتمل. الوقت منذ السجل الأخير: "
    }
}
```

Für jede Sprache ist eine Übersetzungstabelle im Code hinterlegt. Wir haben im Code eine Grundsprache, der Teil vor dem Doppelpunkt, mit der dazugehörigen Übersetzung. Wir übersetzen also jede Ausgabe einmal in die gewählte Sprache.

```
"EN": {  
  "Transport ID eingeben:": "Enter Transport ID:",  
  "Daten prüfen": "Check Data",  
}
```

Beispiel aus der Englisch Übersetzungstabelle.

```
"AR": {  
  "Transport ID eingeben:": "أدخل معرف النقل:",  
  "Daten prüfen": "تحقق من البيانات",  
}
```

Beispiel aus der Arabisch Übersetzungstabelle.

Anhang

Trello-Board:

<https://trello.com/b/Y2F2RZ9f/iot-kuehlkettenueberwachung-g3>

GitHub:

https://github.com/rkpls/Projekt_Kuehlkette