U06 project (car parking ticket-machine)

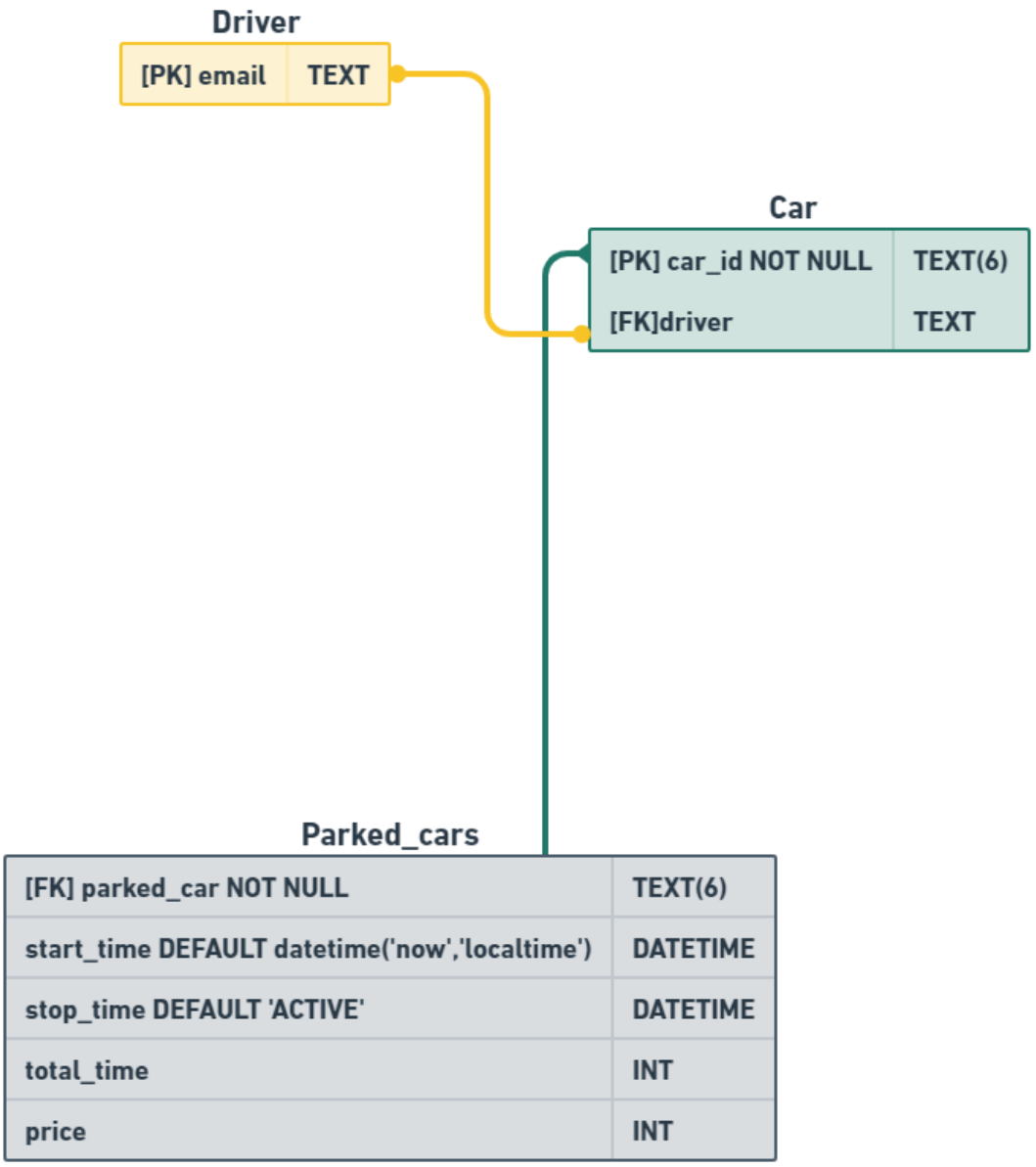
Concept: This project is about building a GUI application for a car parking ticket-machine with Python/Tkinter. To store data related to the application I will be using SQLite for the database. The application is made for *one* parking lot with 50 parking spots. The parking lot has a price list depending on parking duration, the ticket itself will be mailed to the "car driver" (Mailhog) as a receipt in PDF-format after the user ends the parking stay.

Features and functionality:

- **View price list (button)**
  - Prices: 0 - 60 minutes are free, 61 minutes onwards → 0.25kr/min.
  - Messagebox(showinfo) shows the price list when the '**View price lis**t'-button is clicked.
- **Start parking (button)**
  - Start parking-button opens up a new window and disables main menu buttons.
  - Inside the window the user will be asked to enter a 6 character registration number for the car (***xxx123(x)***) and click the '**start parking**'-button.
  - The entered registration number gets checked if it's valid:
    - If it's valid and unique it will be inserted into the car table and parked_cars table with the entered reg num and start time (localtime). Messagebox(showinfo) shows reg num + start time as confirmation. Amount of parking spaces gets subtracted by one.
    - If the registration number is in the wrong format (*123xxx or x234xx).* → Messagebox (showerror).
    - If the entered registration is valid and not unique → Messagebox(showerror).
- **See status for parked car (button)**
  - See status for parked car-button opens up a new window and disables main menu buttons.
  - Inside the window the user will be asked to enter a 6 character registration number for the car (***xxx123(x)***) and click the '**check status**'-button.
  - The entered registration number gets checked if it's valid:
    - If the registration number is valid and registered in the db → return reg num, start time, stop time('active'), total parking time, price for the chosen car.
    - If the registration number is in the wrong format (*123xxx or x234xx).* → Messagebox (showerror)

        - If the entered registration is valid and not entered in the db → Messagebox(showerror)
- **Stop parking (button)**
    - Stop parking-button opens up a new window and disables the main menu buttons.
    - Inside the window the user will be asked to enter a 6 character registration number for the car (***xxx123(x)***) and click the '**stop parking**'-button.
    - The entered registration number gets checked if it's valid:
        - If the registration number is valid and registered in the db → Messagebox(showinfo) shows reg num, start time, stop time, total time, price as confirmation for the chosen car. Update column in parked_cars for the chosen car with stop time, total time and price.
        - If the registration number is in the wrong format (*123xxx or x234xx).* → Messagebox (showerror)
        - If the entered registration is valid and not entered in the db → Messagebox(showerror)
    - When the parking is successfully stopped:
        - Create a label with text that asks the user for an email address.
        - Create an entry box for the user to enter the email address (mandatory to fill).
        - Insert the entered email address to the driver table where the reg num is connected to the driver.
    - Send receipt to "driver" (Mailhog):
        - Receipt has to include reg num, start time, stop time, total time, price (including moms).
        - When the receipt is sent → Delete data connected to the stopped reg num from every table.
        - Close the stop parking window and activate the main menu buttons.
        - Increase total parking spaces by one.

Database structure:



Sprint 1:

- Create database **DONE**
- Create tkinter main menu (**root**) **DONE**
    - Pictures for header and iconbitmap **DONE**
    - Time label **DONE**
        - Function to keep time dynamic **DONE**
    - Date label **DONE**
    - Total parking spaces label **DONE**
- View price list button **DONE**
- Start parking button **DONE**
- See status for parked car **DONE**
- Pycodestyle compliance (--ignore=E501) **DONE**
- flake8 compliance (--ignore=E501) **DONE**
- pylint compliance (--disable=C0301,W0603,R0915) **DONE**

TODO:

Sprint 2:

- Stop parking button
- Disable start parking-button if total parking spaces = 0
- Create a dockerfile
- Mailhog setup