CAR MANAGEMENT SYSTEM

Analysis & Documentation

Index:

Introduction:

Car management system are computer-based solution designed to improve the management of cars associated operations. This system involves the integration of software, hardware, and communication technologies to help managers monitor and control their vehicles effectively. Car management system can provide a range of benefits, including improved efficiency. These systems are widely used by transportation companies, logistics companies, and other organizations that manage large fleets of cars. With advanced features such as real-time tracking, predictive maintenance, and driver behavior monitoring, car management system has become an essential tool for companies seeking to optimize their operations and improve their bottom line.

Problem definition:

The problem that car management systems aim to solve is the difficulty of keeping track of important information related to a car's information. Car owners and fleet managers often struggle to manage the various tasks associated with owning or managing a vehicle. Without a centralized system for managing this information, important tasks may be overlooked.

Ultimately, the problem that car management systems aim to solve is the need for a comprehensive, user-friendly system for managing the various aspects of car ownership and fleet management. By providing a single platform for managing important information related to a vehicle's lifecycle, car management systems can help improve safety, reduce costs, and optimize performance.

Problem scope:

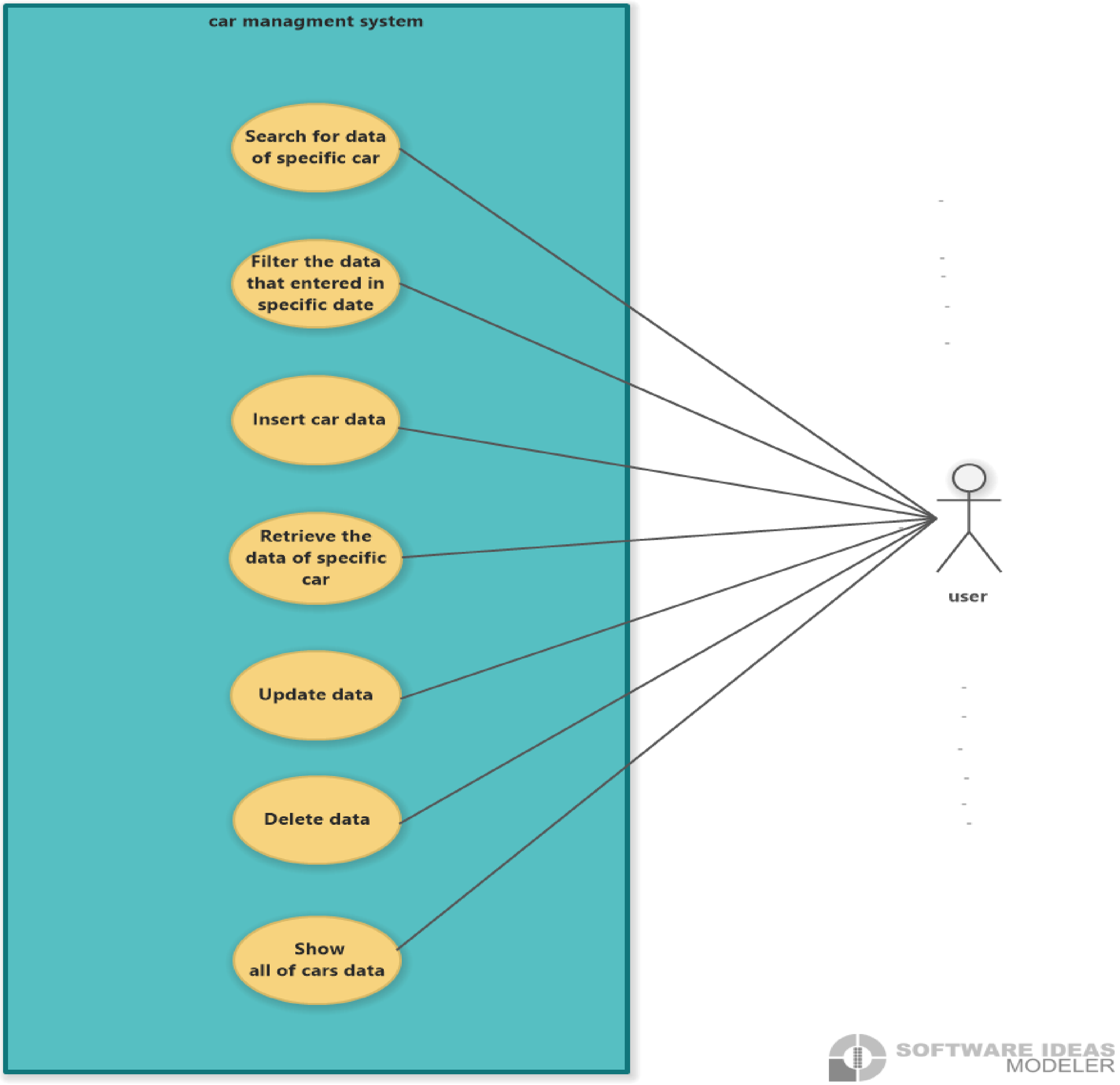
The project scopes the people who are searching for car details, salesman who works in car shop, Car owners, fleet managers, companies seeking to optimize their operations and improve their bottom line

Problem solving:

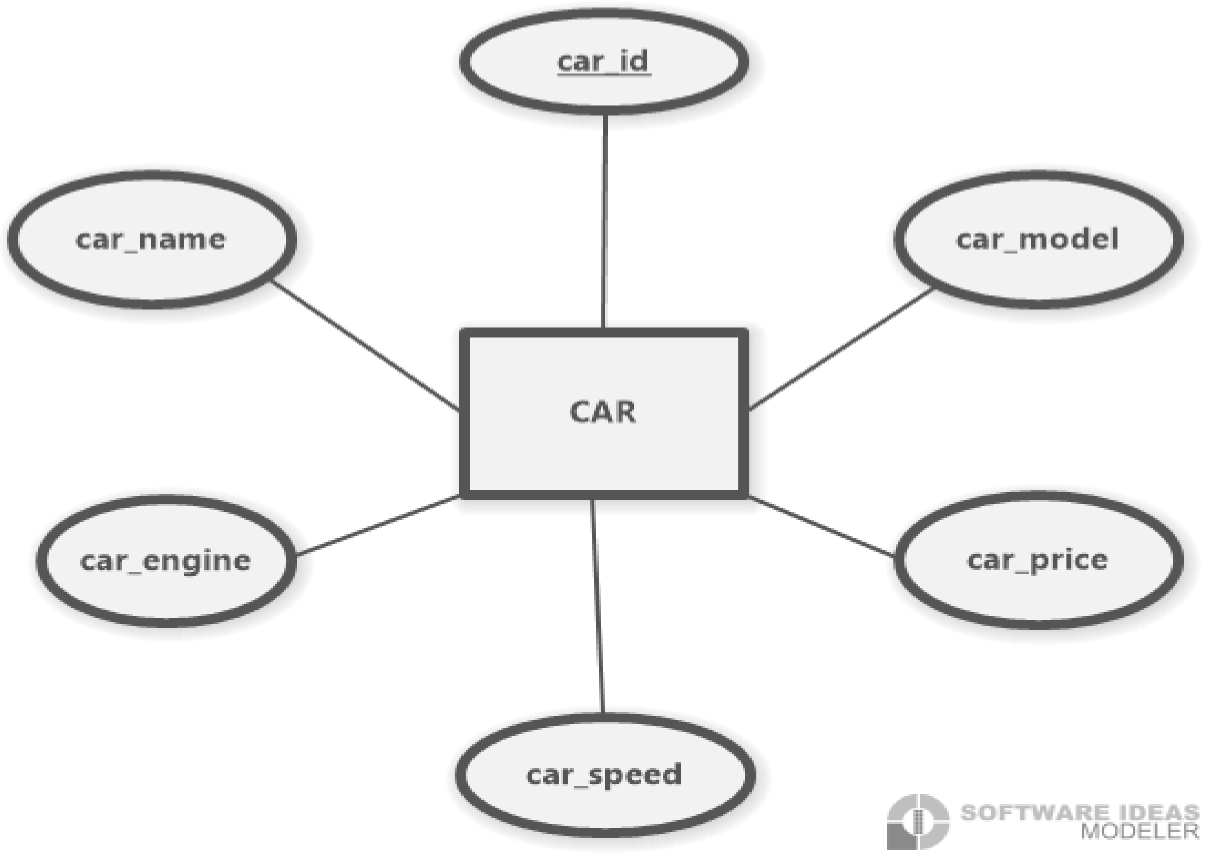
We solve the problem by making a system that enables the user to insert data of the cars, recover and update for data of specific car, delete data of a car, show the data of all cars, search for a specific car with the id given to the car and filter the data according to the date in which it is inserted

Analysis Diagrams:

1- Use Case Diagram


car managment system
Search for data of specific car
Filter the data that entered in specific date
Insert car data
Retrieve the data of specific car
Update data
Delete data
Show all of cars data
user
SOFTWARE IDEAS MODELER

2-ERD


car_id
car_name
car_model
CAR
car_engine
car_price
car_speed
SOFTWARE IDEAS MODELER

Code:

```
import tkinter as tk
from PIL import ImageTk,Image
import time
from tkinter import ttk
from tkinter import messagebox
from tkcalendar import Calendar, DateEntry
from datetime import datetime
import sqlite3

conn = sqlite3.connect('Car.db')

window = tk.Tk()
window.geometry('850x655-200-60')
window.title('Car Managment System')
window.configure(background='#6A6A6A')
window.resizable(width=False, height=False)

icon = tk.PhotoImage(file='car_img.png')
window.call('wm','iconphoto',window._w,icon)

emp_id,emp_name,emp_age = tk.StringVar(),tk.StringVar(),tk.StringVar()
emp_email,emp_phone,emp_salary = tk.StringVar(),tk.StringVar(),tk.StringVar()

cur = conn.cursor()
cur.execute("""CREATE TABLE IF NOT EXISTS emp_data (
   ID TEXT NOT NULL PRIMARY KEY ,Name TEXT NOT NULL,
   Age TEXT NOT NULL,Email TEXT NOT NULL,
   Phone TEXT NOT NULL,Salary TEXT NOT NULL,
   Date_save TEXT NOT NULL)
   """)
cur.close()

def get_ids_list():
   cur = conn.cursor()
   ids_query = "SELECT ID FROM emp_data"
   ids_list = cur.execute(ids_query)
   my_ids = ids_list.fetchall()
   int_ids = [x[0] for x in my_ids]
   return int_ids

emp_ids_list = get_ids_list()

def insertData():
   cur = conn.cursor()
   Car_id = emp_id.get(); Car_name = emp_name.get()
   Car_age = emp_age.get(); Car_email = emp_email.get()
   Car_phone = emp_phone.get(); Car_salary = emp_salary.get()
   date_emp = str(datetime.now().date())

   if Car_id == '' or Car_name == '' or Car_age == '' or Car_email == '' or
Car_phone == '' or Car_salary == '':
      messagebox.showwarning('Missed values', 'There are some blank fields
please check again')
```

```
   else:
      insertQuery = "INSERT INTO emp_data
(ID,Name,Age,Email,Phone,Salary,Date_save) VALUES (?,?,?,?,?,?,?)"
      val = (Car_id,Car_name,Car_age,Car_email,Car_phone,Car_salary,date_emp)
      cur.execute(insertQuery,val)

      if insertData:
         messagebox.showinfo("Car Insertion","Car has been inserted
successfully")

emp_id.set('');emp_name.set('');emp_age.set('');emp_email.set('');emp_phone.s
et('');emp_salary.set('')
      conn.commit()
      get_data()

def updateData(rows):
   trView.delete(*trView.get_children())
   for item in rows:
      trView.insert('','end',values=item)

def get_data():
   global myData
   cur = conn.cursor()
   Car_id = emp_id.get()
   #sel_data = ("SELECT * FROM emp_data WHERE ID=?")
   #cur.execute("SELECT * FROM emp_data WHERE ID=?"+Car_id)

   query = "SELECT ID,Name,Age,Email,Phone,Salary FROM emp_data"
   cur.execute(query)
   myData = cur.fetchall()
   updateData(myData)

def search_Car():
   cur = conn.cursor()
   search_name = searchEnt.get()

   if search_name in emp_ids_list:
      query = "SELECT ID,Name,Age,Email,Phone,Salary FROM emp_data WHERE ID
LIKE ? OR Name LIKE ?"
      cur.execute(query,(search_name,search_name))
      myData = cur.fetchall()
      updateData(myData)
   else:
      messagebox.showinfo("ID does not exist","This ID does not exist in the
database")

def retrive_Car():
   cur = conn.cursor()
   Car_id = emp_id.get()
   my_data = "SELECT Name,Age,Email,Phone,Salary FROM emp_data WHERE ID = ?"
   emp_ids_list = get_ids_list()

   cur.execute(my_data,(Car_id,))
   selected_data = cur.fetchall()

   if Car_id in emp_ids_list:
      for emp_row in selected_data:
            emp_name.set(str(emp_row[0]))
            emp_age.set(str(emp_row[1]))
            emp_email.set(str(emp_row[2]))
            emp_phone.set(str(emp_row[3]))
            emp_salary.set(str(emp_row[4]))
    else:
        messagebox.showinfo("ID does not exist","This ID does not exist in the
database")

def delete_Car():
    cur = conn.cursor()
    Car_id = emp_id.get()

    ids = cur.execute("SELECT ID FROM emp_data")
    list_ids = [str(''.join(item)) for item in ids]

    if not Car_id in list_ids:
        messagebox.showinfo("Error","Car does not exist")

    else:
        delMessage = messagebox.askquestion("Delete Car","Are you sure that you
want to delete this Car ?")
        if delMessage == 'yes':
            deleteQuery = ("DELETE FROM emp_data WHERE ID = ?")

            deleteCar = cur.execute(deleteQuery, (Car_id,))
            messagebox.showinfo("Delete Car","Car has been deleted succesfully")

emp_id.set('');emp_name.set('');emp_age.set('');emp_email.set('');emp_phone.s
et('');emp_salary.set('')
            get_data()
        conn.commit()

App_title_frm = tk.Frame(window,width=830,height=60,bg='#4F5A60')
App_title_frm.place(x=10,y=10)

app_date = datetime.now()
x_date = app_date.strftime('%Y-%m-%d')
#x_time = app_date.strftime('%H:%M:%S')

date_lbl =
tk.Label(App_title_frm,text=x_date,bg='#4F5A60',fg='#E8FEFF',font=('Arial
Greek',10,'bold'))
date_lbl.place(x=70,y=18)

def app_time():
    string = time.strftime('%H:%M:%S %p')
    time_lbl.config(text = string)
    time_lbl.after(1000, app_time)

time_lbl = tk.Label(App_title_frm,bg='#4F5A60',fg='#E8FEFF',font=('Arial
Greek',10,'bold'))
time_lbl.place(x=712,y=18)

app_time()

# ========================================== Menu Barr
```

```
==========================================
def light_theme():
   App_title_frm.config(bg='#D2FFF7')

window.config(background='#F4FFFF');date_lbl.config(bg='#D2FFF7',fg='#555F5F'
)

my_app_title.config(bg='#D2FFF7',fg='#555F5F');time_lbl.config(bg='#D2FFF7',f
g='#555F5F')

search_frm.config(bg='#D1FFEC');search_lbl.config(bg='#C8FFEA',fg='#363D3D')

search_btn.config(bg='#8FDEC2',fg='#363D3D'),searchEnt.config(bg='#A8D7D9',fg
='#131414')
   filter_btn.config(bg='#BDB7FB',fg='#363D3D')
   buttons_box.config(bg='#D9F2FA');filter_date_frm.config(bg='#F4FFFF')

Car_data.config(bg='#C9FFF8',fg='#2B5667');Car_list.config(bg='#F4FFFF',fg='#
31766D')

   filter_lbl.config(bg='#F4FFFF',fg='#323739')
   from_date_lbl.config(bg='#F4FFFF');to_date_lbl.config(bg='#F4FFFF')
   cal1.config(background='#F4FFFF');cal2.config(background='#F4FFFF')

idlbl.config(bg='#C9FFF8',fg='#323739');namelbl.config(bg='#C9FFF8',fg='#3237
39')

agelbl.config(bg='#C9FFF8',fg='#323739');emaillbl.config(bg='#C9FFF8',fg='#32
3739')

phonelbl.config(bg='#C9FFF8',fg='#323739');salarylbl.config(bg='#C9FFF8',fg='
#323739')


idEnt.config(bg='#B2D4DF',fg='#131414');nameEnt.config(bg='#B2D4DF',fg='#1314
14')

ageEnt.config(bg='#B2D4DF',fg='#131414');emailEnt.config(bg='#B2D4DF',fg='#13
1414')

phoneEnt.config(bg='#B2D4DF',fg='#131414');salaryEnt.config(bg='#B2D4DF',fg='
#131414')

   insertCar.config(bg='#77F6E9',fg='#353C3B')
   updateCar.config(bg='#82E8D6',fg='#353C3B')
   retrieveCar.config(bg='#98B6F5',fg='#353C3B')
   deleteCar.config(bg='#FFD393',fg='#353C3B')
   show_data.config(bg='#A5FFCF',fg='#353C3B')

def dark_theme():
   App_title_frm.config(bg='#4F5A60')

window.config(background='#6A6A6A');date_lbl.config(bg='#4F5A60',fg='#E8FEFF'
)

my_app_title.config(bg='#4F5A60',fg='#D1FFF9');time_lbl.config(bg='#4F5A60',f
g='#E8FEFF')
```

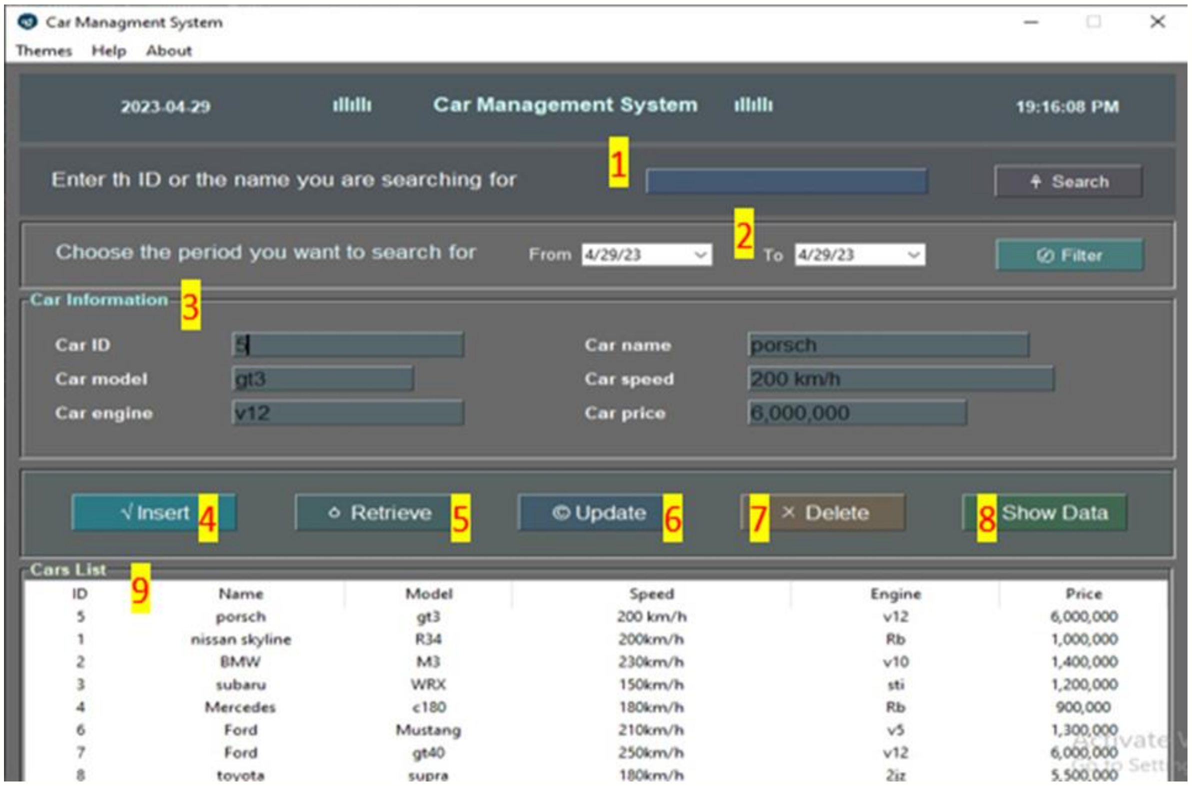
```
   search_frm.config(bg='#53595D');search_lbl.config(bg='#53595D',fg='white')


search_btn.config(bg='#56545D',fg='white'),searchEnt.config(bg='#445A72',fg='
black')
   filter_btn.config(bg='#497D7D',fg='white')


buttons_box.config(bg='#5C6463');filter_date_frm.config(bg='#6A6A6A',fg='whit
e')

Car_data.config(bg='#6A6A6A',fg='#A4FFFB');Car_list.config(bg='#6A6A6A',fg='#
E7FFD8')
   filter_lbl.config(bg='#6A6A6A',fg='#DFF9FF')
   from_date_lbl.config(bg='#6A6A6A');to_date_lbl.config(bg='#6A6A6A')
   cal1.config(background='#536063');cal2.config(background='#536063')


idlbl.config(bg='#6A6A6A',fg='white');namelbl.config(bg='#6A6A6A',fg='white')

agelbl.config(bg='#6A6A6A',fg='white');emaillbl.config(bg='#6A6A6A',fg='white
')

phonelbl.config(bg='#6A6A6A',fg='white');salarylbl.config(bg='#6A6A6A',fg='wh
ite')

idEnt.config(bg=ent_color,fg='#131414');nameEnt.config(bg=ent_color,fg='#1314
14')

ageEnt.config(bg=ent_color,fg='#131414');emailEnt.config(bg=ent_color,fg='#13
1414')

phoneEnt.config(bg=ent_color,fg='#131414');salaryEnt.config(bg=ent_color,fg='
#131414')

   insertCar.config(bg='#2D7A87',fg='white')
   updateCar.config(bg='#435C6B',fg='white')
   retrieveCar.config(bg='#476265',fg='white')
   deleteCar.config(bg='#6C6354',fg='white')
   show_data.config(bg='#416853',fg='white')

def help_window():
   help_win = tk.Toplevel()
   help_win.geometry('500x400+350+150')
   help_win.title('Car Administration System')
   help_win.configure(background='#6A6A6A')
   help_win.resizable(width=False, height=False)

   help_win.mainloop()

def about_app():
   pass

def contact_app():
   pass

menubarr = tk.Menu(window)
```

```
window.config(menu=menubarr)

file_menu = tk.Menu(menubarr, tearoff = 0)
menubarr.add_cascade(label = "Themes", menu = file_menu)
file_menu.add_command(label = "Light Mode",command=light_theme)
#file_menu.add_separator()
file_menu.add_command(label = "Dark Mode",command=dark_theme)

Help_menu = tk.Menu(menubarr, tearoff = 0)
menubarr.add_cascade(label = "Help", menu = Help_menu)
Help_menu.add_command(label = "Help options",command = help_window)

about_menu = tk.Menu(menubarr, tearoff = 0)
menubarr.add_cascade(label = "About", menu = about_menu)
about_menu.add_command(label = "About",command = about_app)
about_menu.add_command(label = "Contact",command = contact_app)

my_app_title = tk.Label(App_title_frm,text='ıllıllı\tCar Management
System\tıllıllı',bg='#4F5A60',
   fg='#D1FFF9',font=('Arial Greek',12,'bold'))
my_app_title.place(x=222,y=15)

search_frm = tk.Frame(window,width=830,height=60,bg='#53595D')
search_frm.place(x=10,y=75)

search_lbl = tk.Label(search_frm,text='Enter th ID you are searching
for',bg='#53595D',
   fg='white',font=('Arial Greek',13))
search_lbl.place(x=20,y=16)

searchEnt = tk.Entry(search_frm,width=30,font=('Arial Greek',12),
   bg='#445A72',fg='#131414',relief=tk.RIDGE)
searchEnt.place(x=350,y=19)

search_btn = tk.Button(search_frm,width=12,padx=2,pady=2,
   font=("Arial Greek",10),bd=1,text="♀
Search",bg="#56545D",fg='white',relief=tk.RIDGE,command=search_Car)
search_btn.place(x=700,y=16)

filter_date_frm = tk.LabelFrame(window,width=830,height=60,bg='#6A6A6A',bd=3,
   font=('Arial Greek',9,'bold'),fg='white')
filter_date_frm.place(x=10,y=140)

Car_data = tk.LabelFrame(window,width=830,height=150,bd=3,font=('Arial
Greek',10,'bold'),
   text='Car Information',bg='#6A6A6A',fg='#A4FFFB')
buttons_box = tk.LabelFrame(window,width=830,height=80,bd=3,font=('Arial
Greek',10,'bold'),
   bg='#5C6463')
Car_list = tk.LabelFrame(window,width=830,height=150,bd=3,font=('Arial
Greek',10,'bold'),
   text='Cars List',bg='#6A6A6A',fg='#E7FFD8')

Car_data.place(x=10,y=200)
buttons_box.place(x=10,y=358)
Car_list.place(x=10,y=438)
```

```
# =============================== Filter Per Date ==========================
filter_lbl = tk.Label(filter_date_frm,text='Choose the period you want to
search for',bg='#6A6A6A',
   fg='#DFF9FF',font=('Arial Greek',13))
filter_lbl.place(x=20,y=12)

cal1 = DateEntry(filter_date_frm,
width=12,background='#536063',foreground='white',borderwidth=1)
cal1.place(x=400,y=16)

from_date_lbl = tk.Label(filter_date_frm,text='From',font=('Arial
Greek',10),bg='#6A6A6A',fg='white')
from_date_lbl.place(x=360,y=15)

cal2 = DateEntry(filter_date_frm,
width=12,background='#536063',foreground='white',borderwidth=1)
cal2.place(x=553,y=16)

to_date_lbl = tk.Label(filter_date_frm,text='To',font=('Arial
Greek',10),bg='#6A6A6A',fg='white')
to_date_lbl.place(x=528,y=16)

def filter_per_date():
   cur = conn.cursor()
   filter_last_date = str(cal1.get_date())
   filter_next_date = str(cal2.get_date())

   query = "SELECT ID,Name,Age,Email,Phone,Salary FROM emp_data WHERE
Date_save BETWEEN ? AND ?"
   cur.execute(query,(filter_last_date,filter_next_date))
   myData = cur.fetchall()
   updateData(myData)

filter_btn = tk.Button(filter_date_frm,width=12,padx=2,pady=2,
   font=("Arial Greek",10),bd=1,text="⟳
Filter",bg="#497D7D",fg='white',relief=tk.RIDGE,command=filter_per_date)
filter_btn.place(x=698,y=13)

# =============================== Filter Per Date ==========================


def update_Car():
   global int_ids
   cur = conn.cursor()
   Car_id = emp_id.get()
   update_query = "UPDATE emp_data SET
Name=?,Age=?,Email=?,Phone=?,Salary=?,Date_save=? WHERE ID=?"

   values_update =
(emp_name.get(),emp_age.get(),emp_email.get(),emp_phone.get(),emp_salary.get(
),
      str(cal1.get_date()),Car_id)
   cur.execute(update_query,values_update)

   if Car_id in emp_ids_list:
      messagebox.showinfo("Car Update","Car has been updated successfully")

emp_id.set('');emp_name.set('');emp_age.set('');emp_email.set('');emp_phone.s
et('');emp_salary.set('')

   else:
      messagebox.showinfo("Car Update","Car has been updated successfully")
   conn.commit()
   get_data()

# ========================= Car Information start =========================
ent_color = '#56666B'
idlbl = tk.Label(Car_data,text='Car ID',font=('Arial
Greek',10,'bold'),bg='#6A6A6A',
   fg='white')
idlbl.place(x=20,y=20)

idEnt = tk.Entry(Car_data,width=18,textvariable=emp_id,font=('Arial
Greek',12),
   bg=ent_color,fg='#131414',relief=tk.RIDGE)
idEnt.place(x=150,y=20)

namelbl = tk.Label(Car_data,text='Car name',font=('Arial
Greek',10,'bold'),bg='#6A6A6A',
   fg='white')
namelbl.place(x=400,y=20)

nameEnt = tk.Entry(Car_data,width=22,textvariable=emp_name,font=('Arial
Greek',12),
   bg=ent_color,fg='#131414',relief=tk.RIDGE)
nameEnt.place(x=520,y=20)

agelbl = tk.Label(Car_data,text='Car model',font=('Arial
Greek',10,'bold'),bg='#6A6A6A',
   fg='white')
agelbl.place(x=20,y=50)

ageEnt = tk.Entry(Car_data,width=14,textvariable=emp_age,font=('Arial
Greek',12),
   bg=ent_color,fg='#131414',relief=tk.RIDGE)
ageEnt.place(x=150,y=50)

emaillbl = tk.Label(Car_data,text='Car speed',font=('Arial
Greek',10,'bold'),bg='#6A6A6A',
   fg='white')
emaillbl.place(x=400,y=50)

emailEnt = tk.Entry(Car_data,width=24,textvariable=emp_email,font=('Arial
Greek',12),
   bg=ent_color,fg='#131414',relief=tk.RIDGE)
emailEnt.place(x=520,y=50)

phonelbl = tk.Label(Car_data,text='Car engine',font=('Arial
Greek',10,'bold'),bg='#6A6A6A',
   fg='white')
phonelbl.place(x=20,y=80)
```

```
phoneEnt = tk.Entry(Car_data,width=18,textvariable=emp_phone,font=('Arial
Greek',12),
   bg=ent_color,fg='#131414',relief=tk.RIDGE)
phoneEnt.place(x=150,y=80)

salarylbl = tk.Label(Car_data,text='Car price',font=('Arial
Greek',10,'bold'),bg='#6A6A6A',
   fg='white')
salarylbl.place(x=400,y=80)

salaryEnt = tk.Entry(Car_data,width=17,textvariable=emp_salary,font=('Arial
Greek',12),
   bg=ent_color,fg='#131414',relief=tk.RIDGE)
salaryEnt.place(x=520,y=80)

# ============================ Car Information End =======================
trView =
ttk.Treeview(Car_list,columns=(1,2,3,4,5,6),show="headings",height=8)
trView.pack(fill='both')

trView.heading(1,text="ID"); trView.heading(2,text="Name")
trView.heading(3,text="Model");trView.heading(4,text="Speed")
trView.heading(5,text="Engine");trView.heading(6,text="Price")

trView.column(1,width=80,anchor="n");trView.column(2,width=150,anchor="n")
trView.column(3,width=120,anchor="n");trView.column(4,width=200,anchor="n")
trView.column(5,width=150,anchor="n");trView.column(6,width=120,anchor="n")

insertCar = tk.Button(buttons_box,width=12,padx=2,pady=2,
   font=("Arial Greek",12),bd=1,text="√
Insert",bg="#2D7A87",fg='white',relief=tk.RIDGE,command=insertData)
insertCar.place(x=35,y=20)

updateCar = tk.Button(buttons_box,width=12,padx=2,pady=2,
   font=("Arial Greek",12),bd=1,text="©
Update",bg="#435C6B",fg='white',relief=tk.RIDGE,command=update_Car)
updateCar.place(x=355,y=20)

retrieveCar = tk.Button(buttons_box,width=12,padx=2,pady=2,
   font=("Arial Greek",12),bd=1,text="ǒ
Retrieve",bg="#476265",fg='white',relief=tk.RIDGE,command=retrive_Car)
retrieveCar.place(x=195,y=20)

deleteCar = tk.Button(buttons_box,width=12,padx=2,pady=2,
   font=("Arial Greek",12),bd=1,text="X
Delete",bg="#6C6354",fg='white',relief=tk.RIDGE,command=delete_Car)
deleteCar.place(x=515,y=20)

show_data = tk.Button(buttons_box,width=12,padx=2,pady=2,
   font=("Arial Greek",12),bd=1,text="⦸ Show
Data",bg="#416853",fg='white',relief=tk.RIDGE,command=get_data)
show_data.place(x=674,y=20)


window.mainloop()
```

Output:



2023-04-29 Car Management System 19:16:08 PM

Enter th ID or the name you are searching for

Choose the period you want to search for From 4/29/23 To 4/29/23

Car ID	5	Car name	porsch
Car model	gt3	Car speed	200 km/h
Car engine	v12	Car price	6,000,000

ID	Name	Model	Speed	Engine	Price
5	porsch	gt3	200 km/h	v12	6,000,000
1	nissan skyline	R34	200km/h	Rb	1,000,000
2	BMW	M3	230km/h	v10	1,400,000
3	subaru	WRX	150km/h	sti	1,200,000
4	Mercedes	c180	180km/h	Rb	900,000
6	Ford	Mustang	210km/h	v5	1,300,000
7	Ford	gt40	250km/h	v12	6,000,000
8	toyota	supra	180km/h	2iz	5,500,000

1- Search bar: Where you can search for specific car using ID
2-Filter bar: Where you can filter the car data according to the date inserted
3- Car information: Where car information appears
4- Insert Button: After completing car data press insert to insert the data
5- Retrieve Button: To retrieve car data
6-Update Button: To update the data of specific car
7-Delete Button: To delete car after inserting car id
8- Show Data Button: Shows the data of all cars
9- Cars list: Contains list of all cars in database

Conclusion:

We propose a new design for the car information management system. This system will provide the benefit for customer’s car lovers, and Anyone interested in cars. The system currently available provides information on all types of cars. In the current system, customers can search and insert data car online. Our proposed system will provide car information systems. Interested car buyers can use this system to help them choose their suitable car from many cars in the databases with this approach, users can Search update delete, and insert new car info into the system. This design will help normal people to start to search for car information. Finally, this system provides good business for organizations and car owners, and for customers, it will be a good service.