



Fenerbahçe Üniversitesi
Endüstri Mühendisliği
İstanbul, Türkiye

COMP205 NESNEYE YÖNELİMLİ PROGRAMLAMA

NÜFUS YÖNETİM SİSTEMİ

Proje Teslim Raporu

Hazırlayanlar:

Hüseyin Can Alkan- huseyin.alkan@stu.fbu.edu.tr

Eray Badem- eray.badem@stu.fbu.edu.tr

İçindekiler

1.Çalışmanın Özeti	3
1.1 Özetçe.....	3
1.2 Anahtar Kelimeler	3
1.3 Abstract	3
1.4 Keywords.....	3
2. Giriş	3
1.1 Projenin Amacı.....	3
3. Sistem Mimarisi.....	3
2.1 Geliştirme Araçları	3
2.2 Tasarım Mimarisi	4
4. Sonuçlar	7
5. Proje Ekibi	7

1.Çalışmanın Özeti

1.1 Özetçe

Çalışmanın özeti bir nüfus yönetim sistemi geliştirmektir. Bu sistemde nesneye yönelik programlamanın işlevleri kullanılarak kişi ekleme, kişi silme, kişi bilgileri güncelleme ve kişi listesi görüntüleme gibi özellikler oluşturulmuştur.

1.2 Anahtar Kelimeler

Kişi ekleme, kişi silme, kişi bilgileri güncelleme ve kişi listesi görüntüleme

1.3 Abstract

The summary of the study is to develop a population management system. In this system, features such as adding contact, deleting contact, updating contact information and viewing contact list were created by using the functions of object oriented programming.

1.4 Keywords

Adding contact, deleting contact, updating contact information and viewing contact list

2. Giriş

1.1 Projenin Amacı

Projenin amacı bir nüfus yönetim sistemi geliştirmektir. Yönetici tarafından yeni bir vatandaş kaydı yapılabilir, kayıtlı vatandaşların bilgileri listelenebilir, güncellenebilir veya silinebilir. Kaydedilen bilgiler veritabanında bir tablo olarak tutulur ve dosyaya kaydedilir.

3. Sistem Mimarisi

2.1 Geliştirme Araçları

- PyCharm Community Edition
- Sqlite3 veritabanı kütüphanesi

2.2 Tasarım Mimarisi

```
import sqlite3

class Kisiler:
    def __init__(self):
        self.connect = sqlite3.connect("kisiler.db")
        self.cursor = self.connect.cursor()
        self.verileriOku()
        print("Veritabanındaki vatandaş sayısı:", len(self.verileriOku()))

    def verileriOku(self):
        self.cursor.execute("SELECT * FROM kisiler")
        kisiler = self.cursor.fetchall()
        return kisiler

    def dbGuncelle(self):
        self.connect.commit()

    def tablo_olustur(self):
        self.cursor.execute("CREATE TABLE IF NOT EXISTS kisiler (kimlikNo INT, Adı TEXT, SoyAdı TEXT, BabaAdı TEXT, AnneAdı TEXT, DoğumYeri TEXT, MedeniDurumu TEXT, KanGrubu TEXT, KütükŞehir TEXT, Kütükİlçe TEXT, İkametgahŞehir TEXT, İkametgahİlçe TEXT)")
        self.dbGuncelle()
```

Programımız “anaEkran.py” dosyasını çalıştırarak başlar.

- “proje.py” dosyasının içerisinde öncelikle sqlite3 kütüphanesini import ederek başlar.
- Sonrasında “Kisiler” adında bir sınıf oluşturur. Bu sınıfın constructorında veritabanı bağlantısını gerçekleştirir ve bir imleç oluşturur. Veritabanındaki kişi sayısı ekrana yazdırılır.
- “verileriOku()” fonksiyonunda imleci “execute()” fonksiyonu ile çalıştırıp, “Select * from” sorgusunu kullanarak “kisiler” tablosundaki bütün verileri alır ve “fetchall()” fonksiyonu ile bunu “kisiler” adlı bir listeye atar.
- “dbGuncelle()” fonksiyonu, sqlite3 kütüphanesinin “commit()” metodunu, veritabanını güncellemek için kullanır.
- “tablo_olustur()” fonksiyonu imleci çalıştırıp, “Create Table If not Exists” sorgusunu kullanarak “kisiler” adında bir tablo olup olmadığını kontrol eder. Eğer yok ise “kisiler” adında bir tablo oluşturur ve gerekli değişkenleri tanımlar. En sonda da “dbGuncelle()” fonksiyonu çağırılarak veritabanı güncellenir.

Ana Menü

```
from proje import Kisiler
class anaEkran(Kisiler):
    def __init__(self):
        super().__init__()
    def menu(self):
        print("VATANDAŞ VERİTABANI".center(100, "*").upper())
        print("1-) Vatandaş ekleme\n"
              "2-) Vatandaş silme\n"
              "3-) Vatandaş bilgi güncelleme\n"
              "4-) Vatandaş listeleme\n"
              "5-) Vatandaş arama\n")
```

```

        "6-) Çıkış\n")
    while True:
        try:
            choice = int(input("Your choice: "))
            if int(choice) < 1 or int(choice) > 6:
                print("Lütfen 1-6 arası bir sayı giriniz.")
                continue
            break
        except ValueError:
            print("Yanlış giriş!! Lütfen tam sayı bir değer giriniz.")
    return choice
def calistir(self):
    choice = self.menu()
    if choice == 1:
        self.kisi_olustur()
    if choice == 2:
        self.kisi_silme()
    if choice == 3:
        self.bilgiGuncelle()
    if choice == 4:
        self.veri_listele()
    if choice == 5:
        self.arastir()
    if choice == 6:
        self.cikis()

calistir = anaEkran()
while True:
    calistir.calistir()

```

“anaEkran.py” dosyasında “Kisiler” sınıfından kalıtım yapan bir “anaEkran” sınıfı bulunur. Sınıfın içinde bulunan “menu()” fonksiyonunda kullanıcıya seçenekler listelenir ve bir değer alınır, değer istenen aralıkta değil ise hata mesajı iletilir. “calistir()” fonksiyonunda kullanıcının seçim yaptığı değerin karşılığı olan fonksiyon çalıştırılır.

1-) Kişi Oluşturma

```

def kisi_olustur(self):
    self.kimlikNo = self.kimlikNoGirme()
    for i in self.verileriOku():
        if i[0] == self.kimlikNo:
            print("{} kimlik numaralı kişi kayıtlıdır.".format(self.kimlikNo))
            self.cikis()
    self.adi = input("Veritabanına girilecek olan kişinin adını giriniz:
").lower().capitalize()
    self.soyadi = input("Veritabanına girilecek olan kişinin soyadını giriniz:
").lower().capitalize()
    self.babaAdi = input("Veritabanına girilecek olan kişinin baba adını giriniz:
").lower().capitalize()
    self.anneAdi = input("Veritabanına girilecek olan kişinin anne adını giriniz:
").lower().capitalize()
    self.dogumYeri = input("Veritabanına girilecek olan kişinin doğum yerini giriniz:
").lower().capitalize()
    self.medeniDurumu = input("Veritabanına girilecek olan kişinin medeni durumu:
").lower().capitalize()
    self.kanGrubu = input("Veritabanına girilecek olan kişinin kan grubunu giriniz:
").lower().capitalize()
    self.kutukSehir = input("Veritabanına girilecek olan kişinin kütük şehrini giriniz:
").lower().capitalize()
    self.kutukIlce = input("Veritabanına girilecek olan kişinin kütük ilçesini giriniz:
").lower().capitalize()
    self.ikametgahSehir = input("Veritabanına girilecek olan kişinin ikametgah şehrini giriniz.").lower().capitalize()
    self.ikametgahIlce = input("Veritabanına girilecek olan kişinin ikametgah ilçesini giriniz: ").lower().capitalize()
    self.cursor.execute("INSERT INTO kisiler VALUES(?,?,?,?,?,?,?,?,?,?,?,?)", (self.kimlikNo, self.adi, self.soyadi, self.babaAdi, self.anneAdi, self.dogumYeri, self.medeniDurumu, self.kanGrubu, self.kutukSehir, self.kutukIlce, self.ikametgahSehir, self.ikametgahIlce))
    self.dbGuncelle()
    print("Vatandaş veritabanına eklendi.")

```

“kişi_olustur()” fonksiyonu, öncelikle kişinin kimlik numarası istenir. Eğer kimlik numarası veritabanında kayıtlı değilse gerekli olan diğer bilgiler istenir. Veritabanına “INSERT INTO” komutu ile veriler “kisiler” tablosuna kaydedilir.

2-) Kişi Silme

```
def kisi_silme(self):
    self.kimlikNo = self.kimlikNoGirme()
    for i in self.verileriOku():
        if i[0] == self.kimlikNo:
            self.cursor.execute("Delete from kisiler where kimlikNo= ?", (self.kimlikNo,))
            self.dbGuncelle()
            print("Vatandaş veritabanından silindi.")
            break
    else:
        print("Kişi bulunamadı")
```

“kisi_silme()” fonksiyonu, öncelikle silinecek kişinin kimlik numarası istenir. Eğer veritabanından alınan listede yoksa ekrana “Kişi bulunamadı.” yazısı bastırılır. Varsa “Delete” komutuyla kişi veritabanından silinir.

3-) Bilgi Güncelleme

```
def bilgiGuncelle(self):
    self.kimlikNo = self.kimlikNoGirme()
    for i in self.verileriOku():
        if i[0] == self.kimlikNo:
            operations =
["kimlikNo", "Adı", "SoyAdı", "BabaAdı", "AnneAdı", "DoğumYeri", "MedeniDurumu", "KanGrubu", "KütükŞeh
ir", "Kütükİlçe", "İkametgahŞehir", "İkametgahİlçe"]
            while True:
                try:
                    updateSelect = int(input("1-) Kimlik No\n2-) Adı\n3-) Soyadı\n4-) Baba Adı\n5-)
Anne Adı\n6-) Doğum Yeri\n7-) Medeni Durumu\n8-) Kan Grubu\n9-) Kütük Şehir\n10-) Kütük İlçe\n11-)
İkametgah Şehir\n12-) İkametgah İlçe\nSeçiniz: "))
                    if int(updateSelect) < 1 or int(updateSelect) > 12:
                        print("Lütfen 1 ile 12 arasında bir sayı giriniz.")
                        continue
                    break
                except ValueError:
                    print("Yanlış giriş!! Lütfen tam sayı bir değer giriniz.")
            yeniDeger = input("{} için yeni değeri giriniz: ".format(operations[updateSelect-
1])).lower().capitalize()
            self.cursor.execute("UPDATE kisiler SET {} = '{}' WHERE
kimlikNo='{}' ".format(operations[updateSelect - 1], yeniDeger, self.kimlikNo))
            self.dbGuncelle()
            print("{} kimlik nolu vatandaşın bilgileri güncellendi.".format(self.kimlikNo,))
            break
    else:
        print("Kişi bulunamadı")
```

“bilgiGuncelle()” fonksiyonu kimlik numarası istenerek başlar. Eğer veritabanında kimlik numarası mevcutsa, kullanıcıya hangi bilgiyi güncellemek istediği sorulur. Kullanıcı girişi yapıldıktan sonra yeni değer istenir. Daha sonra “UPDATE” komutu ile veritabanı güncellenir.

4-) Verileri Listeleme

```
def veri_listele(self):  
    print("Vatandas Listesi Hazirlanıyor ...")  
    for i in self.verileriOku():  
        print(i)
```

“veri_listele()” fonksiyonu, “verileriOku()” fonksiyonunu kullanarak verileri ekrana bastırır.

5-) Kişi Araştırma

```
def arastir(self):  
    self.kimlikNo = self.kimlikNoGirme()  
    for i in self.verileriOku():  
        if int(i[0]) == int(self.kimlikNo):  
            print(i)  
            break  
    else:  
        print("Kişi bulunamadı")
```

“arastir()” fonksiyonu , kullanıcı tarafından girilen kimlik numarasını veritabanında arar. Eğer varsa kişi bilgilerini ekrana bastırır. Yoksa “Kişi bulunamadı.” yazısı ekrana bastırılır.

6-) Çıkış

```
def cikis(self):  
    self.connect.close()  
    exit(0)
```

“cikis()” fonksiyonu, veritabanını kapatarak programı sonlandırır.

4. Sonuçlar

- Python'da nesneye yönelimli programlamanın detaylarını öğrendik.
- *sqlite3* veritabanı kütüphanesini araştırarak SQL kullanımını öğrendik ve kodumuzda etkili bir şekilde kullanabildik.
- Latex sistemini kullanarak CV oluşturmayı öğrendik.

Youtube Linki: <https://youtu.be/myvaNJ9MKtg>

Projenin Github Linki: <https://github.com/hpoohdini12/Nufus-Yonetim-Sistemi>

5. Proje Ekibi

Hüseyin Can Alkan:

01.12.2001 yılında İstanbul'da doğdum. 2019 yılında Recep Güngör Anadolu Lisesi'nden mezun oldum. Fenerbahçe Üniversitesi Endüstri Mühendisliği bölümünde 2. Sınıf olarak eğitimime devam etmekteyim. Giriş seviyesinde C programlama dili ve Python programlama diline de hakimim.

Eray Badem:

26.07.2001 yılında Yalova’da doğdum. 2019 yılında Yalova Fen Lisesi’nden mezun oldum. Fenerbahçe Üniversitesi Endüstri Mühendisliği bölümünde 2. Sınıf olarak eğitime devam etmekteyim. Giriş seviyesinde C programlama dili ve Python programlama diline de hakimim.



Hüseyin Can Alkan
huseyin.alkan@stu.
fbu.edu.tr

[https://github.com/
hpoohdini12](https://github.com/hpoohdini12)

Adres
İstanbul
Türkiye

Hüseyin Can Alkan

Endüstri Mühendisi

Hakkımda Fenerbahçe Üniversitesi'nde 2. sınıf Endüstri Mühendisliği öğrencisiyim. Hobi olarak kodlama ve basketbol ile ilgileniyorum.

Eğitim

2015 - 2019 , Recep Güngör Anadolu Lisesi

2019 , Fenerbahçe Üniversitesi

Endüstri Mühendisliği

Diller

Türkçe(Ana dil),İngilizce(Orta seviye)

Yetenekler

Programlama

- Python
- C

Bilgisayar Yazılımları

- MySQL
- iOS
- Android

Projeler

Nüfus Yönetim Sistemi: Yönetici tarafından yeni bir vatandaş kaydı yapılabilir, kayıtlı vatandaşların bilgileri listelenebilir, güncellenebilir veya silinebilir. Kaydedilen bilgiler veritabanında bir tablo olarak tutulur ve dosyaya kaydedilir.

İlgi Alanları

Kitap okuma,basketbol,kodlama,



Eray Badem
eray.badem@stu.fbu.
edu.tr

Adres
Yalova
Türkiye

Eray Badem

Endüstri Mühendisi

Hakkımda Fenerbahçe Üniversitesi'nde 2. sınıf Endüstri Mühendisliği öğrencisiyim. Hobi olarak video oyunları, kodlama,futbol ve basketbol ile ilgileniyorum.

Eğitim

2015 - 2019 , Yalova Fen Lisesi

2019 , Fenerbahçe Üniversitesi
Endüstri Mühendisliği

Diller

Türkçe(Ana dil),İngilizce(Orta seviye)

Yetenekler

Programlama

- Python
- C

Bilgisayar Yazılımları

- MySQL
- iOS
- Android

Projeler

Nüfus Yönetim Sistemi: Yönetici tarafından yeni bir vatandaş kaydı yapılabilir, kayıtlı vatandaşların bilgileri listelenebilir, güncellenebilir veya silinebilir. Kaydedilen bilgiler veritabanında bir tablo olarak tutulur ve dosyaya kaydedilir.

İlgi Alanları

Kitap okuma,basketbol,kodlama,video oyunları