



GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

ELECTRONICS ENGINEERING

INF 212

Algorithms and Programming II

Project 2 Report

Name – Surname	Burak Topçu
Student Number	151024064

İçindekiler

Bölüm 1) Projenin Amacı ve Görevler

- a) Projenin Amacı
- b) Görevler.....

Bölüm 2) Problemin Tanımı ve Çözümü

- a) Problemin Tanımı ve Analizi
- b) Problemin Çözümü.....

Bölüm 3) Implementation (Uygulama)

- a) Tasarım ve İşlevler
- b) Menü İşlemleri için Fonksiyonlar
- c) Akış Diyagramı
- d) Menüdeki İşlevlerin Açıklanması
- e) Değişken ve Fonksiyon İsimlendirmesi.....
- f) Programın Test Edilmesi
- g) Kontrol Fonksiyonları
- h) Program Kısıtlamaları ve Olası Hatalar

Bölüm 4) Ekran Görüntüleri ve Kaynak Kod

- a) Ekran Görüntüleri.....
- b) Kaynak Kod.....

Bölüm 1: Projenin Amacı ve Görevler

a) Projenin Amacı: Genel olarak projenin amacı, bir eczane entanveri otomasyon yazılımı oluşturmaktır, bu yazılım eczane bilgilerini ve eczanelerde bulunan ilaçların bilgilerini veritabanı dosyalarında saklama, bu veritabanlarında eczane ve ya ilaç bazlı arama/görüntüleme yapabilme özelliklerine sahip olmalıdır. Ayrıca eczane ve ilaç bilgilerine ekleme/düzenleme yapılabilmesi ve veritabanları güncellenmelidir. Yazılım yönetici ve müşteriler tarafından kullanılacaktır bu yüzden yönetimsel işlemler için ayrı bölüm oluşturulmalı ve buraya sadece yetkilendirilmiş kişilerin erişimi sağlanmalıdır, müşteriler ise kendi bölümlerine kısıtlamasız olarak erişebilir olmalıdır.

b) Görevler

Görevler Grubu 1: Yönetimsel

- 1) Eczane Bilgileri için Veritabanı Oluşturulması
- 2) Eczanenin İlaç Envanteri için Veritabanı Oluşturulması
- 3) Eczane Ekleme ve Veritabanı Güncelleme
- 4) İlaç Ekleme ve Veritabanı Güncelleme
- 5) İlaç Düzenleme ve Veritabanı Güncelleme
- 6) Yönetimsel Görevler için Yetki Doğrulama

Görevler Grubu 2: Müşteri İşlemleri

- 1) Tüm Eczane İlaç Veritabanlarında İlaç Arama
- 2) Eczane Veritabanını Listeleme
- 3) Veritabanında Eczane Arama
- 4) Seçilen Eczane İlaç Veritabanında İlaç Arama

Bölüm 2) Problemin Tanımı ve Çözümü

a) Problemin Tanımı ve Analizi: Eczane ve ilaç verileri birden fazla değişkenden (nesneden) oluşur, bu durumda bu nesne ve bağlı/üye fonksiyonları tek bir çatı altında tutan bir şablon/nesne yapısı kullanılmalıdır. Ayrıca bu veriler düzenli bir şekilde dosyaya yazılarak tekrar okunabilir olmalıdır. Oluşturulan bu veritabanı dosyaları içerisinde arama ve düzenleme işlemleri gerçekleştirilmelidir.

Problem Girdileri:

Eczane Bilgileri -> İsmi(string), Numarası(integer), Adresi(string)

İlaç Bilgileri -> Numarası(integer), Sayısı(integer), İsmi(string), Fiyatı(double)

Üye Fonksiyonlar:

Eczane için

- Eczane oluşturma fonksiyonu -> **eczn_olustur()**
- Eczane yazdırma fonksiyonu -> **eczn_goster()**
- Eczane için ilaç liste oluşturma -> **eczn_ilac_sayfasi()**
- Ayrıca tekil olarak değişken değerleri atamak /değiştirmek ve öğrenmek için fonksiyonlar (**set ve get fonksiyonları**)

İlaç için

- İlaç oluşturma fonksiyonu -> **ilac_olustur()**
- İlaç yazdırma fonksiyonu -> **ilac_goster()**
- İlaç düzenleme fonksiyonu -> **ilac_duzenle()**
- Ayrıca tekil olarak değişken değerleri atamak /değiştirmek ve öğrenmek için fonksiyonlar (**set ve get fonksiyonları**)

Problem Çıktıları:

Eczane listesi veritabanı dosyası (.dat formatında) -> **(eczane_listesi.dat)**

Her bir eczane için ilaç listesi veritabanı dosyası (.dat formatında) -> **(eczaneismi_ilac.dat)**

Aranan ilaç ve ya eczanenin bilgilerinin gösterilmesi

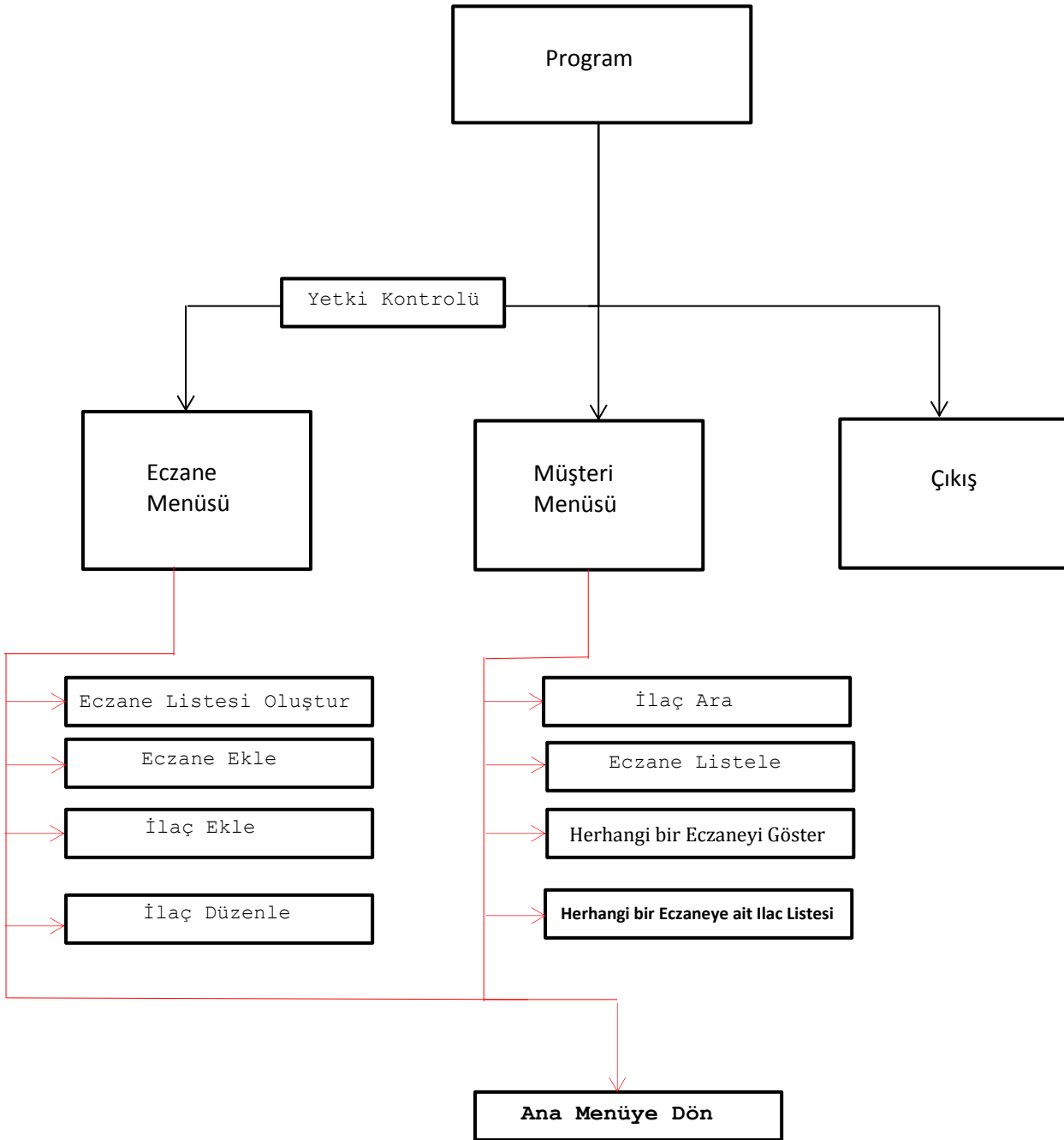
Veritabanı Okuma/Yazma İşlemleri için Fonksiyonlar

- İstenilen sayıda eczane ile eczane listesi ve bağlı ilaç listesi oluşturmak için fonksiyon -> **ecznListesiOlustur()**
- Eczane oluşturup veritabanına eklemek için fonksiyon -> **eczn_ekle()**
- İlaç oluşturup veritabanına eklemek için fonksiyon -> **ilac_ekle()**
- Eczane veritabanını okumak ve yazdırmak için fonksiyon -> **eczaneleri_goster()**
- Eczane ilaç veritabanını okumak ve yazdırmak için fonksiyon -> **ilac_listele()**
- İstenilen eczanenin verilerini dosyadan bulup yazdıran fonksiyon -> **eczane_goster()**
- İstenilen ilacı veritabanlarında arayıp bularak yazdıran fonksiyon -> **ilac_ara()**

Problemin Çözümü: Problemin analizi kısmında belirttiğimiz birden fazla değişken/nesne ve üye fonksiyonu tek bir çatı altında toplayan veri tipi C++ dilinde **sınıflar** yani “**class**”lardır, eczane ve ilaç verileri ve üye fonksiyonlar için sınıf oluşturarak yukarıda tanımladığımız girdi ve üye fonksiyonları bu sınıflar içerisinde tanımlayabiliriz. Veritabanı okuma/yazma/oluşturma işlemleri için ise **fstream** kütüphanesinin **ifstream** ve **ofstream** class’larını kullanacağız. Bahsettiğimiz diğer fonksiyonları ise bunların dışında ayrıca tanımlıyacağız. Görevler bölümünde değindiğimiz işlevler için menü ve benzeri diğer detayları program tasarımı ve akış diyagramı bölümlerinde aktaracağız. **Sonuç olarak problemi temelde nesne yönelimli programlamanın yapı taşı olan class veri tipini kullanarak çözüyoruz.**

Bölüm 3) Implementation (Uygulama)

a) Tasarım ve İşlevler: Kodlamaya başlamadan önce programın genel yapı ve işlevlerini belirleyelim.



b) Menü İşlemleri için Fonksiyonlar

Görevler için tanımlanan fonksiyonlar **if-else if-else** yapısı ile oluşturulan menülerde ilgili seçenekte kullanılmış ve bu menüler fonksiyon olarak oluşturulmuştur, böylece kullanıcı yapmak istediği işlevi seçerek bunun için oluşturulmuş işlevi kullanacaktır.

Ana Menü -> **menu_goster()**

Eczane Menüsü -> **eczane_menusu()**

Müşteri Menüsü -> **musteri_menusu()**

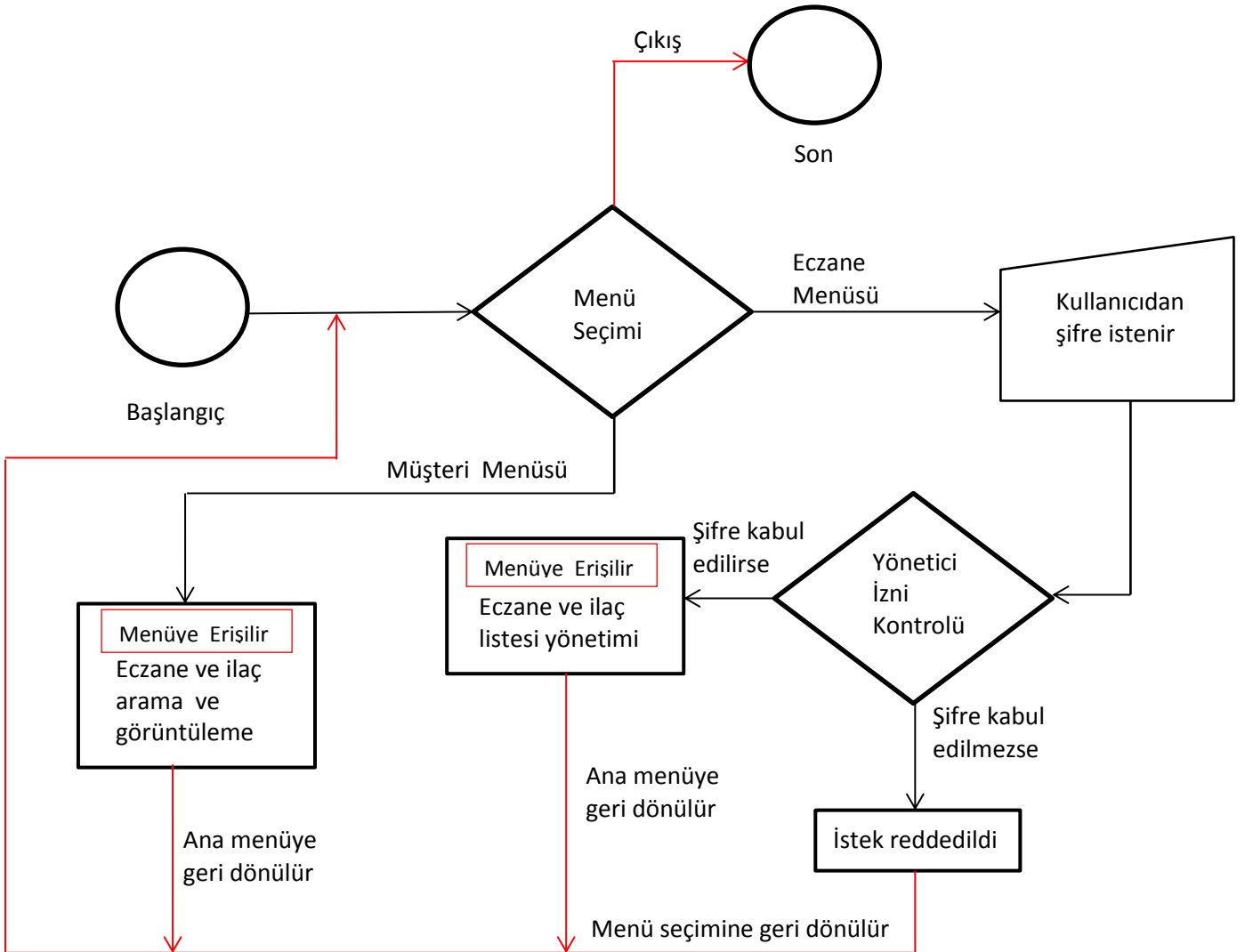
Eczane menüsü için bir **şifre** belirlenmiş ve kullanıcı bu menüye ulaşmak isterse önce şifre sorularak geçerli bir şifre girip girmediği kontrol edilecektir.

İndeksleme İşlemleri için Yardımcı Fonksiyonlar:

eczn_sayisi () -> Eczane listesinde kayıtlı eczane sayısını veren ek fonksiyon.

obje_sayisi () -> İstenen eczaneye kaç ilaç kayıtlı olduğunu veren ek fonksiyon.

c) Akış Diyagramı:



d)Menüdeki İşlevlerin Açıklanması

Eczane Menüsü

- | | |
|------------------------|---|
| Eczane Listesi Oluştur | → Kullanıcıya kaç adet eczane gireceği sorulur ve liste oluşturularak girdiği eczaneler eklenir. Eklediği her bir eczaneden sonra ilaç listesini doldurmak isteyip istemediği sorulur, kullanıcı ister bu adımda ister daha sonra ilaç ekle bölümünden eczanenin ilaç listesine ilaç ekleyebilir. |
| Eczane Ekle | → Eczane ismi, adresi, numarası kullanıcıdan alınır ve listeye eklenir. |
| İlaç Ekle | → İlaç bilgileri kullanıcıdan alınır ve listeye eklenir. |
| İlaç Düzenle | → İlaç bilgisi değiştirmek istenen eczanenin ismi ve değiştirilmek istenen ilacın ID'si kullanıcıdan alınır ve ilaç bulunarak yeni değerler alınarak ilaç bilgisi veritabanında güncellenir. |

Müşteri Menüsü

- | | |
|--|--|
| İlaç Ara | → Aranılan ilacın ismi kullanıcıdan alınır ve tüm eczane ilaç listelerinde aranarak sonuçlar gösterilir. |
| Eczane Listele | → Veritabanındaki tüm eczaneler listelenir. |
| Herhangi bir Eczaneyi Göster | → Kullanıcıdan ismi alınan eczanenin bilgileri gösterilir. |
| Herhangi bir Eczaneye ait İlaç Listesi | → Kullanıcıdan ismi alınan eczanenin ilaç bilgileri gösterilir. |

e) Değişken ve Fonksiyon İsimlendirmesi: Değişken ve fonksiyonların isimleri projede belirtildiği şekilde direkt dökümandan alınarak kullanılmıştır. Genel olarak değişken/fonksiyon işlevi ile bağlantılı kolay anlaşılır isimlendirme kullanılmıştır.

f) Programın Test Edilmesi: 4 adet eczane oluşturulup her birine ait ilaç listelerine aynı isim ve ID farklı fiyat ve sayıda 5 adet ilaç eklenerek koddaki her bir fonksiyon test edilmiştir.

g) Kontrol Fonksiyonları:

Yeni ilaç ve eczane oluştururken, kullanıcının girdiği isim ve ya ID ile zaten kayıtlı bulunan nesne var mı diye kontrol edilir, kullanıcı aynı isimde iki tane eczane oluşturursa bu karışıklığa yol açar, bu kontrolü kullanıcıya bırakmadan biz kontrol ederek bu durumun önüne geçiyoruz.

kopyakontroleczane() -> Yeni eczane ile kayıtlı eczaneler arasında ID ve isim çakışması var mı kontrol eder.
kopyakontrolilac() -> Yeni ilaç ile kayıtlı ilaçlar arasında ID ve isim çakışması var mı kontrol eder.

h) Program Kısıtlamaları ve Olası Hatalar: Eczane ve ilaç isimleri ile adres bilgileri tek kelimelik string olarak tanımlanmıştır. Bunu arttırmak için koddaki değişiklik yapmak gerekir. Bir ilaç düzenlendiğinde sadece o ilacın bulunduğu satır değil tüm veritabanı güncellenmektedir, bu yöntem listede çok sayıda ilaç olduğunda bellek kullanımının artmasına sebep olur, bu işlev daha verimli şekilde yeniden tasarlanabilir. Ayrıca projede istenen şartlar gereği ilacın sadece fiyat ve sayı bilgileri düzenlenebilir.

Bölüm 4) Ekran Görüntüleri ve Kaynak Kod

a) Ekran Görüntüleri:

1)Ana Menü

```
Eczane Programina Hosgeldiniz!  
1. Eczane Girisi  
2. Musteri Girisi  
3. Cikis
```

```
Seciminiz:
```

2)Eczane Menüsü

```
Eczane Programina Hosgeldiniz!  
1. Eczane Girisi  
2. Musteri Girisi  
3. Cikis
```

```
Seciminiz: 1  
Sifrenizi giriniz: admin  
Sifre kabul edildi.
```

```
Eczane Girisi  
1. Eczane Listesi Olustur  
2. Eczane Ekle  
3. Ilac Ekle  
4. Ilac Duzenle  
5. Ana Menuye Don  
Seciminiz:
```

Kullanıcı şifresini girerek "Eczane Menüsü'ne" ulaşır.

2.a) Eczane Listesi Olustur

```
Eczane Girisi  
1. Eczane Listesi Olustur  
2. Eczane Ekle  
3. Ilac Ekle  
4. Ilac Duzenle  
5. Ana Menuye Don  
Seciminiz: 1  
Kac adet eczane gireceksiniz?  
1  
Eczane numarasini girin:  
1  
Eczane ismini girin :  
fidan  
Eczane adresini girin:  
Maltepe  
Eczane listesi ve eczanenin ilac listesi olusturuldu.  
Ilac listesini doldurmak ister misiniz?(E/H)  
Daha sonra 'Ilac ekle' bolumunden de doldurabilirsiniz.  
Cevabininizi girin (E/H).  
h
```

Kullanıcı bu seçenekte istediği sayıda eczane ile eczane listesi oluşturabilir, her bir eczaneyi oluşturduktan sonra ilaç listesini de doldurmak isteyip istemediği sorulur.

2.b) Eczane Ekle

```
Eczane Girisi  
1. Eczane Listesi Olustur  
2. Eczane Ekle  
3. Ilac Ekle  
4. Ilac Duzenle  
5. Ana Menuye Don  
Seciminiz: 2  
Eczane numarasini girin:  
2  
Eczane ismini girin :  
derman  
Eczane adresini girin:  
Kadikoy  
Eczane listeye eklendi.
```

Kullanıcının girdiği eczane listeye eklenir ve ilaç sayfasında oluşturulur, kullanıcı daha sonra ilaç ekle bölümünden eczaneye ilaç ekleyebilir.

2.c)İlaç Ekle

```
Eczane Girisi
1. Eczane Listesi Oluştur
2. Eczane Ekle
3. İlaç Ekle
4. İlaç Düzenle
5. Ana Menüye Don
Seciminiz: 3
İlaç eklenmek istenen eczanenin ismini giriniz:
fidan
İlaç ID giriniz:
1
İlaç sayısını giriniz:
6
İlaç fiyatını giriniz:
13
İlaç ismini giriniz:
parol
İlaç listeye eklendi.
```

Kullanıcıdan bilgileri alınan ilaç, ismi alınan eczaneye eklenir.

2.d)İlaç Düzenle

```
Eczane Girisi
1. Eczane Listesi Oluştur
2. Eczane Ekle
3. İlaç Ekle
4. İlaç Düzenle
5. Ana Menüye Don
Seciminiz: 4
İlaçları düzenlemek istenen eczanenin ismini giriniz:
fidan
Düzenlemek istediğiniz ilacın ID'sini girin':
1
Düzenlemek istediğiniz ilacın bilgileri:
isim: parol ! ID: 1 ! Sayı: 6 ! Fiyat: 13
Sistem Notu: İlaç ID'si ve ismi değiştirilemez.
Yeni ilaç sayısını giriniz:
12
Yeni ilaç fiyatını giriniz:
15
İlaç bilgileri başarılı bir şekilde düzenlendi.
```

Kullanıcıdan ismi alınan eczanenin listesinde bulunan ilaç düzenlenir.

3)Müşteri Menüsü

```
Eczane Programına Hoşgeldiniz!
1. Eczane Girisi
2. Müşteri Girisi
3. Çıkış

Seciminiz: 2

Müşteri Girisi
1. İlaç Ara
2. Eczane Listesi
3. Herhangi bir Eczaneyi Göster
4. Herhangi bir Eczaneye ait İlaç Listesi
5. Ana Menüye Don
Seciminiz:
```

Kullanıcı müşteri menüsüne şifresiz olarak erişir.

3.a)İlaç Ara

```
Musteri Girisi
1. Ilac Ara
2. Eczane Listesi
3. Herhangi bir Eczaneyi Goster
4. Herhangi bir Eczaneye ait Ilac Listesi
5. Ana Menuye Don
Seciminiz: 1
Aradiginiz ilacin adini girin:
parol
Ilacin bulundugu eczane:fidan
Bu eczanede fiyatı: 15 ! Bu eczanede sayisi:12
Ilacin bulundugu eczane:derman
Bu eczanede fiyatı: 11 ! Bu eczanede sayisi:10
Ilacin bulundugu eczane:hayat
Bu eczanede fiyatı: 16 ! Bu eczanede sayisi:3
Ilacin bulundugu eczane:deniz
Bu eczanede fiyatı: 15 ! Bu eczanede sayisi:8
Aradiginiz ilac 4 tane eczanede bulundu.
```

Kullanıcıdan aramak istediği ilacın adı alınır ve tüm eczanelerde aranarak bulunan sonuçlar listelenir, hangi eczanede bulunduğu ve bulunduğu eczanedeki sayısı ve fiyatı gösterilir.

3.b) Eczane Listesi

```
Musteri Girisi
1. Ilac Ara
2. Eczane Listesi
3. Herhangi bir Eczaneyi Goster
4. Herhangi bir Eczaneye ait Ilac Listesi
5. Ana Menuye Don
Seciminiz: 2
-----> Eczane listesi yazdiriliyor <-----
Eczane ismi: fidan ! Eczane ID: 1 ! Eczane adresi: Maltepe
Eczane ismi: derman ! Eczane ID: 2 ! Eczane adresi: Kadikoy
Eczane ismi: hayat ! Eczane ID: 3 ! Eczane adresi: Pendik
Eczane ismi: deniz ! Eczane ID: 4 ! Eczane adresi: Kartal
-----> Yazdirma islemi bitti <-----
```

Tüm eczaneler listelenir.

3.c) Herhangi Bir Eczaneyi Göster

```
Musteri Girisi
1. Ilac Ara
2. Eczane Listesi
3. Herhangi bir Eczaneyi Goster
4. Herhangi bir Eczaneye ait Ilac Listesi
5. Ana Menuye Don
Seciminiz: 3
Aradiginiz eczanenin ismini giriniz:
derman
-----> Eczane yazdiriliyor <-----
Eczane ismi: derman ! Eczane ID: 2 ! Eczane adresi: Kadikoy
-----> Yazdirma islemi bitti <-----
```

Kullanıcıdan ismi alınan eczanenin bilgileri gösterilir.

3.d) Herhangi bir Eczaneye ait Ilac Listesi

```
Musteri Girisi
1. Ilac Ara
2. Eczane Listesi
3. Herhangi bir Eczaneyi Goster
4. Herhangi bir Eczaneye ait Ilac Listesi
5. Ana Menuye Don
Seciminiz: 4
Ilac lari listelenmek istenen eczanenin ismini giriniz:
derman
-----> Ilac listesi yazdiriliyor <-----
isim: parol ! ID: 1 ! Sayi: 10 ! Fiyat: 11
isim: augmentin ! ID: 2 ! Sayi: 20 ! Fiyat: 21
isim: aspirin ! ID: 3 ! Sayi: 25 ! Fiyat: 31
isim: dolorex ! ID: 4 ! Sayi: 30 ! Fiyat: 21
isim: nurofen ! ID: 5 ! Sayi: 35 ! Fiyat: 11
-----> Yazdirma islemi bitti <-----
```

Kullanıcıdan ismi alınan eczanenin ilaç listesi gösterilir.

b) Kaynak Kod

Kütüphanelerin import edilmesi:

```
#include<iostream>
#include<fstream>
#include<string>
#include<stdio.h>
#include<stdlib.h>
using namespace std;
```

Yönetici şifresinin belirlenmesi:

```
const string& yonetici_sifresi="admin";
```

Eczane sınıfının oluşturulması:

```
class eczn
{
    private:
        int eczn_id;
        string eczn_isim;
        string eczn_adres;
    public:
        eczn();
        eczn(int id_, string isim_, string adres_);
        eczn(const eczn &obj);
        eczn& operator=(const eczn &obj);

        void eczn_olustur();
        void eczn_goster();
        void eczn_ilac_sayfasi();

        void setID_Num(int id_);
        void setIsim(string isim_);
        void setAdres(string adres_);

        int getID_Num();
        string getIsim();
        string getAdres();
};
```

İlaç sınıfının oluşturulması:

```
class ilac{
private:
    int ilac_id;
    int ilac_sayisi;
    double ilac_fiyati;
    string ilac_isim;
public:
    ilac();
    ilac(int id_,int sayi_,double fiyat_,string isim_);
    ilac(const ilac &obj);
    ilac& operator=(const ilac &obj);

    void ilac_olustur();
    void ilac_goster();
    void ilac_duzenle();

    void setID_Num(int id_);
    void setSayi(int sayi_);
    void setFiyat(double fiyat_);
    void setIsim(string isim_);

    int getID_Num();
    int getSayi();
    double getFiyat();
    string getIsim();
};
```

Menülerde kullanılacak fonksiyonların prototiplerinin tanımlanması:

```
void menu_goster();
void eczane_menusu();
void muster_i_menusu();

void ecznListesiOlustur();
void eczn_ekle();
void ilac_ekle();
void ilacDuzenle();
void ilac_ara();
void eczaneleri_goster();
void eczane_goster();
void ilac_listele();

int eczn_sayisi ();
int obje_sayisi (string ecznIsmi);
int kopyakontroleczane(string isim_kopya,int id_kopya);
int kopyakontrolilac(string isim_kopya,int id_kopya,string ecznIsmi);
```

Eczane sınıfındaki constructor ve üye fonksiyonların tanımlanması:

```
eczn::eczn()
{
    eczn_id = 0;
    eczn_isim = "\0";
    eczn_adres = "\0";
}
eczn::eczn(int id_, string isim_, string adres_)
{
    eczn_id = id_;
    eczn_isim = isim_;
    eczn_adres = adres_;
}
eczn::eczn(const eczn &obj)
{
    eczn_id = obj.eczn_id;
    eczn_isim = obj.eczn_isim;
    eczn_adres = obj.eczn_adres;
}
eczn &eczn::operator=(const eczn &obj)
{
    eczn_id = obj.eczn_id;
    eczn_isim = obj.eczn_isim;
    eczn_adres = obj.eczn_adres;
    return (*this);
}

void eczn::eczn_olustur()
{
    cout<<"Eczane numarasini girin:"<<endl;
    cin>>eczn_id; cin.clear(); fflush(stdin);
    cout<<"Eczane ismini girin : "<<endl;
    cin>>eczn_isim; cin.clear(); fflush(stdin);
    cout<<"Eczane adresini girin: "<<endl;
    cin>>eczn_adres; cin.clear(); fflush(stdin);
}
void eczn::eczn_goster()
{
    cout<<"Eczane ismi: "<<eczn_isim<<" | Eczane ID: "<<eczn_id<<" | Eczane
adresi: "<<eczn_adres<<endl;
}
void eczn::eczn_ilac_sayfasi()
{
    string dosyaadi;
    dosyaadi = eczn_isim+"_ilac.dat";
    fstream filex;
    filex.open(dosyaadi.c_str(), ios::out);
    filex.close();
}

void eczn::setID_Num(int id_){eczn_id = id_;}
void eczn::setIsim(string isim_){eczn_isim=isim_;}
void eczn::setAdres(string adres_){eczn_adres = adres_;}

int eczn::getID_Num(){return eczn_id;}
string eczn::getIsim(){return eczn_isim;}
string eczn::getAdres(){return eczn_adres;}
```

İlaç sınıfındaki constructor ve üye fonksiyonların tanımlanması:

```
ilac::ilac()
{
    ilac_id = 0;
    ilac_sayisi = 0;
    ilac_fiyati = 0;
    ilac_isim = "\\0";
}
ilac::ilac(int id_, int sayi_, double fiyat_, string isim_ )
{
    ilac_id = id_;
    ilac_sayisi = sayi_;
    ilac_fiyati = fiyat_;
    ilac_isim = isim_;
}
ilac::ilac(const ilac &obj)
{
    ilac_id = obj.ilac_id;
    ilac_sayisi = obj.ilac_sayisi;
    ilac_fiyati = obj.ilac_fiyati;
    ilac_isim = obj.ilac_isim;
}
ilac &ilac::operator=(const ilac &obj)
{
    ilac_id = obj.ilac_id;
    ilac_sayisi = obj.ilac_sayisi;
    ilac_fiyati = obj.ilac_fiyati;
    ilac_isim = obj.ilac_isim;
    return (*this);
}

void ilac:: ilac_olustur()
{
    cout<<"İlaç ID giriniz: "<<endl;
    cin>>ilac_id;cin.clear(); fflush(stdin);
    cout<<"İlaç sayisini giriniz: "<<endl;
    cin>>ilac_sayisi;cin.clear(); fflush(stdin);
    cout<<"İlaç fiyatini giriniz: "<<endl;
    cin>>ilac_fiyati;cin.clear(); fflush(stdin);
    cout<<"İlaç ismini giriniz: "<<endl;
    cin>>ilac_isim;cin.clear(); fflush(stdin);
}
void ilac::ilac_goster()
{
    cout<<"isim: "<<ilac_isim<<" | ID: "<<ilac_id<<" | Sayi:
"<<ilac_sayisi<<" | Fiyat: "<<ilac_fiyati<<endl;
}
void ilac::ilac_duzenle()
{
    cout<<"Sistem Notu: İlaç ID'si ve ismi degistirilemez."<<endl;
    cout<<"Yeni ilaç sayisini giriniz: "<<endl;
    cin>>ilac_sayisi;cin.clear(); fflush(stdin);
    cout<<"Yeni ilaç fiyatini giriniz: "<<endl;
    cin>>ilac_fiyati;cin.clear(); fflush(stdin);
}
```

```

void ilac::setID_Num(int id_){ilac_id = id_;}
void ilac::setSayi(int sayi_){ilac_sayisi=sayi_;}
void ilac::setFiyat(double fiyat_){ilac_fiyati=fiyat_;}
void ilac::setIsim(string isim_){ilac_isim=isim_;}

int ilac::getID_Num(){return ilac_id;}
int ilac::getSayi(){return ilac_sayisi;}
double ilac::getFiyat(){return ilac_fiyati;}
string ilac::getIsim(){return ilac_isim;}

```

Eczane menüsünün oluşturulması:

```

void eczane_menusu()
{
    while (1)
    {
        int secim,adet;
        cout<<"Eczane Girisi"<<endl<<
        "1. Eczane Listesi Olustur"<<endl<<
        "2. Eczane Ekle"<<endl<<
        "3. Ilac Ekle"<<endl<<
        "4. Ilac Duzenle"<<endl<<
        "5. Ana Menuye Don"<<endl<<
        "Seciminiz: "<<endl;
        cin>>secim;
        if(secim> 5 || secim<1){
            cout<<"Gecersiz bir secenek girdiniz."<<endl;
        }
        else {
            if (secim==1){
                cout<<"Kac adet eczane gireceksiniz?"<<endl;
                cin>>adet;
                if(adet>0){
                    for (;adet>0;adet=adet-1){
                        ecznListesiOlustur();
                    }
                }
                else {cout<<"Gecersiz giris."<<endl;}
            }

            if(secim==2)
                eczn_ekle();
            if(secim==3)
                ilac_ekle();
            if(secim==4)
                ilacDuzenle();
            if(secim==5)
            {
                break;
            }
        }
    }
}

```

Müşteri menüsünün oluşturulması:

```
void muster_i_menusu()  
{  
    while (1)  
    {  
        int secim;  
        cout<<"Musteri Girisi"<<endl<<  
        "1. Ilac Ara"<<endl<<  
        "2. Eczane Listesi"<<endl<<  
        "3. Herhangi bir Eczaneyi Goster"<<endl<<  
        "4. Herhangi bir Eczaneye ait Ilac Listesi"<<endl<<  
        "5. Ana Menuye Don"<<endl<<  
        "Seciminiz: "<<endl;  
        cin>>secim;  
  
        if (secim==1)  
            ilac_ara();  
        else if (secim==2)  
            eczaneleri_goster();  
        else if (secim==3)  
            eczane_goster();  
        else if (secim==4)  
            ilac_listele();  
        else if (secim==5)  
        {  
            break;  
        }  
        else  
            cout<<"Gecersiz giris, lutfen gecerli bir secim  
giriniz."<<endl;  
    }  
}
```

Ana menünün oluşturulması:

```
void menu_goster()  
{  
    string sifre;int secim;  
    while (1)  
    {  
        cout<<"1. Eczane Girisi"<<endl<<  
        "2. Musteri Girisi"<<endl<<  
        "3. Cikis"<<endl<<  
        "Seciminiz: "<<endl;  
        cin>>secim;  
  
        if (secim==1)  
        {  
            cout<<"Sifrenizi giriniz: ";  
            cin>>sifre;cin.clear(); fflush(stdin);  
            if (sifre==yonetici_sifresi){  
                cout<<"Sifre kabul edildi."<<endl;  
                eczane_menusu(); }  
            else{  
                cout<<"Gecersiz sifre girildi."<<endl;  
            }  
        }  
    }  
}
```

Eczane menüsü
İçin şifre
kontrolü

```

        else if(secim==2)
            musteri_menusu();
        else if(secim==3){
            cout<<"Programdan cikmak icin bir tusa basin..."<<endl;
            system("pause"); break;}

        else
            cout<<"Gecersiz giris, lutfen gecerli bir secim
giriniz."<<endl;
    }
}

```

Eczane listesi oluşturma fonksiyonunun tanımlanması:

```

void ecznListesiOlustur()
{
    eczn a;char c;
    a.eczn_olustur();
    int kopya_kontrol=kopyakontroleczane(a.getIsim(),a.getID_Num());
    if(kopya_kontrol==0){
        a.eczn_ilac_sayfasi();
        ofstream my_file1;
        my_file1.open("eczane_listesi.dat", ios::app);
        (my_file1 << a.getID_Num()<<" "+a.getIsim()+" " << a.getAdres()<<endl);
        my_file1.close();
        cout<<"Eczane listesi ve eczanenin ilac listesi olusturuldu."<<endl;
        cout<<"Ilac listesini doldurmak ister misiniz?(E/H)"<<endl;
        cout<<"Daha sonra 'Ilac ekle' bolumunden de doldurabilirsiniz."<<endl;
        cout<<"Cevabininizi girin (E/H)."<<endl;
        scanf("%s",&c);
        if(c == 'E' || c =='e'){
            A:
                ilac ilac1;
                ilac1.ilac_olustur();
                ofstream my_file2;
                my_file2.open((a.getIsim()+"_ilac.dat").c_str(),ios::app);
                my_file2<<ilac1.getID_Num()<<" "<<ilac1.getSayi()<<"
"<<ilac1.getFiyat()<<" "<<ilac1.getIsim()<<endl;
                my_file2.close();
                cout<<"Daha fazla ilac eklemek istiyor musunuz?(E/H)"<<endl;
                scanf("%s",&c);
                if(c == 'E' || c =='e'){goto A;}
                else{}
            }
        else{}}
        else { cout<<"Bu id ve ya isimde eczane zaten var."<<endl;
            cout<<"ID ve isim her bir eczane icin benzersiz olmalı."<<endl;}
    }
}

```


Eczane ekleme fonksiyonunun tanımlanması:

```
void eczn_ekle()
{
    eczn a;
    a.eczn_olustur();
    int kopya_kontrol=kopyakontrolieczane(a.getIsim(),a.getID_Num());
    if(kopya_kontrol==0){
        a.eczn_ilac_sayfasi();
        ofstream my_file1;
        my_file1.open("eczane_listesi.dat", ios::app);
        (my_file1 << a.getID_Num()<<" "+a.getIsim()+" " << a.getAdres()<<endl);
        my_file1.close();cout<<"Eczane listeye eklendi."<<endl;}
        else { cout<<"Bu id ve ya isimde eczane zaten var."<<endl;
            cout<<"ID ve isim her bir eczane icin benzersiz olmalı."<<endl;}
    }
}
```

İlaç ekleme fonksiyonunun tanımlanması:

```
void ilac_ekle()
{
    string ecznIsmi;
    cout<<"İlaç eklenmek istenen eczanenin ismini giriniz:"<<endl;
    cin>>ecznIsmi;cin.clear(); fflush(stdin);
    ifstream myFile((ecznIsmi+"_ilac.dat").c_str());
    if(myFile.fail()){
        cout<<"Girilen isimde eczane ve ya eczanenin ilac listesi yok."<<endl;}
    else {
        myFile.close();
        ilac ilac1; ilac1.ilac_olustur(); int kopya_kontrol;
        kopya_kontrol=kopyakontrolilac(ilac1.getIsim(),ilac1.getID_Num(),ecznIsmi);
        if(kopya_kontrol==0){
            ofstream my_file2;
            my_file2.open((ecznIsmi+"_ilac.dat").c_str(),ios::app);
            my_file2<<ilac1.getID_Num()<<" "<<ilac1.getSayi()<<"
"<<ilac1.getFiyat()<<" "<<ilac1.getIsim()<<endl;
            my_file2.close(); cout<<"İlaç listeye eklendi."<<endl;}
            else { cout<<"Bu id ve ya isimde ilac zaten var."<<endl;
                cout<<"ID ve isim her bir ilac icin benzersiz olmalı."<<endl;}}
    }
}
```

İndeksleme işlemleri için yardımcı fonksiyonların tanımlanması:

```
int eczn_sayisi ()
{
    int eczn_sayisi = 0;
    string satir;
    ifstream myfile("eczane_listesi.dat");
    while (getline(myfile, satir)){ ++eczn_sayisi;}
    return eczn_sayisi;
}
int obje_sayisi (string ecznIsmi)
{
    int obje_sayisi = 0;
    string satir;
    ifstream myfile((ecznIsmi+"_ilac.dat").c_str());
    while (getline(myfile, satir)){ ++obje_sayisi;}
    return obje_sayisi;
}
```

Bir eczanenin ilaç listesini yazdırma fonksiyonu:

```
void ilac_listele()
{
    string ecznIsmi;
    cout<<"İlaclari listelenmek istenen eczanenin ismini giriniz:"<<endl;
    cin>>ecznIsmi;cin.clear(); fflush(stdin);
    int n=obje_sayisi(ecznIsmi);
    if (n==0){ cout<<"Aradiginiz isimde eczane ve ya eczaneye kayitli ilac
yok."<<endl;}
    else{
        cout<<"-----> Ilac listesi yazdiriliyor <-----
---"<<endl;
        ilac ilac [n];
        ifstream myfile((ecznIsmi+"_ilac.dat").c_str());
        int id,sayi;double fiyat;string isim;

        for(int i = 0; i < n; i++){
            if(myfile >> id >> sayi >>fiyat>>isim)
            {
                ilac[i].setID_Num(id);
                ilac[i].setSayi(sayi);
                ilac[i].setFiyat(fiyat);
                ilac[i].setIsim(isim);
                ilac[i].ilac_goster();
            }
            cout<<"-----> Yazdirma islemi bitti <-----
-----"<<endl;
            myfile.close(); }
    }
```

İlaç düzenleme fonksiyonu:

```
void ilacDuzenle()
{
    string ecznIsmi; int temp_id,index;
    index=-1;
    cout<<"İlaclari duzenlemek istenen eczanenin ismini giriniz:"<<endl;
    cin>>ecznIsmi;cin.clear(); fflush(stdin);
    int n=obje_sayisi(ecznIsmi);
    if (n==0){ cout<<"Aradiginiz isimde eczane ve ya eczaneye kayitli ilac
yok."<<endl;}
    else{
        cout<<"Duzenlemek istediginiz ilacin ID'sini girin':"<<endl;
        cin>>temp_id;cin.clear(); fflush(stdin);
        ilac ilac [n];
        ifstream myfile((ecznIsmi+"_ilac.dat").c_str());
        int id,sayi;double fiyat;string isim;

        for(int i = 0; i < n; i++){
            if(myfile >> id >> sayi >>fiyat>>isim)
            {
                ilac[i].setID_Num(id);
                ilac[i].setSayi(sayi);
                ilac[i].setFiyat(fiyat);
                ilac[i].setIsim(isim);
                if (id==temp_id){
```

```

        index=i;
    }
}
if (index== -1){ cout<<"Aradiginiz ID'ye sahip ilac
bulunamadi."<<endl; }
else{
    cout<<"Duzenlemek istediginiz ilacin bilgileri:"<<endl;
    ilac[index].ilac_goster();
    ilac[index].ilac_duzenle();
    fstream my_file;
    my_file.open((ecznIsmi+"_ilac.dat").c_str(), ios::out);
    for(int i = 0; i < n; i++)
        if(my_file << ilac[i].getID_Num()<<" "<<ilac[i].getSayi()<<"
"<<ilac[i].getFiyat()<<" "<<ilac[i].getIsim()<<endl)
        {}
        cout<<"Ilac bilgileri basarili bir sekilde duzenlendi."<<endl;
        myfile.close(); my_file.close();}
}

```

İstenilen bir eczanenin bilgilerini göstermek için fonksiyon:

```

void eczane_goster()
{
    string eczn_isim;int index=-1;
    cout<<"Aradiginiz eczanenin ismini giriniz:"<<endl;
    cin>>eczn_isim;cin.clear(); fflush(stdin);
    int n1=eczn_sayisi();
    ifstream myfile3("eczane_listesi.dat");
    int ID; string name; string adress;

    eczn eczn [n1];
    for(int i = 0; i < n1; i++){
        if(myfile3 >> ID >> name >> adress)
        {
            eczn[i].setID_Num(ID);
            eczn[i].setIsim(name);
            eczn[i].setAdres(adress);
            if (eczn[i].getIsim()==eczn_isim){
                index=i;
            }
        }
    }
    if(index== -1){ cout<<"Aranan isimde eczane yok."<<endl;}
    else{
        cout<<"----->          Eczane yazdiriliyor          <-----
-----"<<endl;
        eczn[index].eczane_goster();
        cout<<"----->          Yazdirma islemi bitti          <-----
-----"<<endl;
    }
    myfile3.close();
}

```

Tüm eczaneleri listelemek için fonksiyon:

```
void eczaneleri_goster()
{
    int n1=eczn_sayisi();
    if (n1==0){cout<<"Sisteme henuz eczane eklenmemis."<<endl;}
    else{
        cout<<"-----> Eczane listesi yazdiriliyor <-----
        -----"<<endl;
        ifstream myfile3("eczane_listesi.dat");
        int ID; string name; string adress;

        eczn eczn [n1];
        for(int i = 0; i < n1; i++){
            if(myfile3 >> ID >> name >> adress)
            {
                eczn[i].setID_Num(ID);
                eczn[i].setIsim(name);
                eczn[i].setAdres(adress);
                eczn[i].eczn_goster();
            }
        }
        cout<<"-----> Yazdirma islemi bitti <-----
        -----"<<endl;
        myfile3.close();
    }
}
```

İlaç arama fonksiyonu:

```
void ilac_ara()
{
    string ilac_ismi; int kontrol=-1;
    cout<<"Aradiginiz ilacin adini girin:"<<endl;
    cin>>ilac_ismi;cin.clear(); fflush(stdin);
    int n1=eczn_sayisi(); string eczn_gecici;
    ifstream myfile3("eczane_listesi.dat");
    int ID; string name; string adress;
    int id,sayi;double fiyat;string isim;
    eczn eczn [n1];
    for(int i = 0; i < n1; i++){
        if(myfile3 >> ID >> name >> adress)
        {
            eczn[i].setID_Num(ID);
            eczn[i].setIsim(name);
            eczn[i].setAdres(adress);
        }
    }

    for(int i=0;i<n1;i++){
        eczn_gecici=eczn[i].getIsim();
        int n=obje_sayisi(eczn_gecici);
        ilac ilac [n];
        ifstream myfile((eczn_gecici+"_ilac.dat").c_str());
        for(int i = 0; i < n; i++){
            if(myfile >> id >> sayi >>fiyat>>isim)
            {
                ilac[i].setID_Num(id);
```

İlaç arama fonksiyonu basitçe üç for döngüsünden oluşur, birinci döngüde eczane listesinden tüm eczaneler okunur, ikinci döngüde teker teker eczanelerin ilaç listeleri okunur ve üçüncü döngüde eczane ilaç listelerinde kullanıcının girdiği ilaç ismi ile eşleşen ilaç varsa bu eczanedeki bilgileri ve eczane ismi yazdırılır. Son olarakda ilacın kaç eczanede bulunduğu dair bilgi yazdırılır.

```

        ilac[i].setSayi(sayi);
        ilac[i].setFiyat(fiyat);
        ilac[i].setIsim(isim);
        if (ilac_ismi==ilac[i].getIsim()){
            kontrol=kontrol+1;
            cout<<"Ilacin bulunduđu eczane:"<<eczn_gecici<<endl;
            cout<<"Bu eczanede fiyatı: "<<ilac[i].getFiyat()<<" | Bu eczanede
sayısı:"<<ilac[i].getSayi()<<endl;
        }
    }myfile3.close(); myfile.close();}
    if (kontrol==1){cout<<"Aradığınız ilac hic bir eczanede
bulunamadi."<<endl;}
    else{ cout<<"Aradığınız ilac "<<(kontrol+1)<<" tane eczanede
bulundu."<<endl;}
}

```

Aynı isim ve id ile eczane eklenmesini önlemek için fonksiyon:

```

int kopyakontrolieczane(string isim_kopya,int id_kopya)
{
    int kontrol=0;
    int n1=eczn_sayisi();
    if (n1==0){return kontrol;}
    else{
        ifstream myfile3("eczane_listesi.dat");
        int ID; string name; string adress;
        eczn eczn [n1];
        for(int i = 0; i < n1; i++){
            if(myfile3 >> ID >> name >> adress)
            {
                eczn[i].setID_Num(ID);
                eczn[i].setIsim(name);
                eczn[i].setAdres(adress);
                if(eczn[i].getID_Num()==id_kopya||eczn[i].getIsim()==isim_kopya){
                    kontrol=1;}
            }
        }myfile3.close();}
    return kontrol;
}

```

Aynı isim ve id ile ilaç eklenmesini önlemek için fonksiyon:

```

int kopyakontrolilac(string isim_kopya,int id_kopya,string ecznIsmi)
{
    int kontrol=0;
    int n=obje_sayisi(ecznIsmi);
    if (n==0){ return kontrol;}
    else{
        ilac ilac [n];
        ifstream myfile((ecznIsmi+"_ilac.dat").c_str());
        int id,sayi;double fiyat;string isim;

        for(int i = 0; i < n; i++){
            if(myfile >> id >> sayi >>fiyat>>isim)
            {
                ilac[i].setID_Num(id);
                ilac[i].setSayi(sayi);
                ilac[i].setFiyat(fiyat);
                ilac[i].setIsim(isim);
            }
        }
    }
}

```

```
if(ilac[i].getID_Num()==id_kopya||ilac[i].getIsim()==isim_kopya){kontrol=1;  
}  
    }} myfile.close();}  
    return kontrol;  
}
```

Oluşturulan fonksiyonların kullanılacağı ana fonksiyonun tanımlanması:

```
int main()  
{  
    cout<<"Eczane Programina Hosgeldiniz!"<<endl;  
    menu_goster();  
    return 0;  
}
```