

ÖZET

Projemizin adı market ürün ve müşteri kayıt otomasyonudur. Bu proje fikri oluşurken aklımızdan geçenler, günlük hayatta insanların işlerini kolaylaştıracak ürünler üretmeye adım atmaktı. Adında da anlaşılacağı üzere market otomasyonumuz marketlerde satıcı ve alıcı arasında yapılan alış-verişlerin dijital bir ortama taşıyarak satıcının raporlarının tek bir çatı altında toplanmasını amaçlamaktır. Programımızı incelediğimizde üç ana başlığa ayırdığımızı görebiliriz. Birinci kısımda ürünler ile ilgili işlemler bulunmaktadır. Burada marketin içinde bulunan ürünlerin kayıt altına alınması, listelenmesi güncellenmesi, silinmesi ve aranması gibi başlıklar bulunmaktadır. İkinci kısımda müşteri ile satıcı arasında iletişimi sağlayan satış işlemleri yer almaktadır. Birinci kısımda tanımladığımız ürünlerin her birine ayrı ayrı kod vererek satış kısmında çok kolay bir şekilde satılan ürünün kodunun girilmesi ve adedinin girilmesiyle satış yapılan ürünlerin toplam tutarı ekrana bastırılmaktadır. Üçüncü kısımda ise marketten alışveriş yapan müşterilerin kaydı tutulmaktadır. Bu kısımda müşterinin kaydı yani müşteriye ait bilgilerin yanında zincir marketler için kullanım kolaylığı sağlamak adına hangi şubeden alış-veriş yapıldığı da belirtilmektedir. Eğer müşteri ödeme yapmadıysa ödeme yapılmadığını veya yapılan alışveriş tutarından daha az bir ödeme yapıldıysa kalan borç tutarı bilgisi gösterilecek müşterinin gecikmede mi cezada mı yoksa güncel kategoride mi olduğu sistem tarafından gösterilmektedir.

1. GİRİŞ

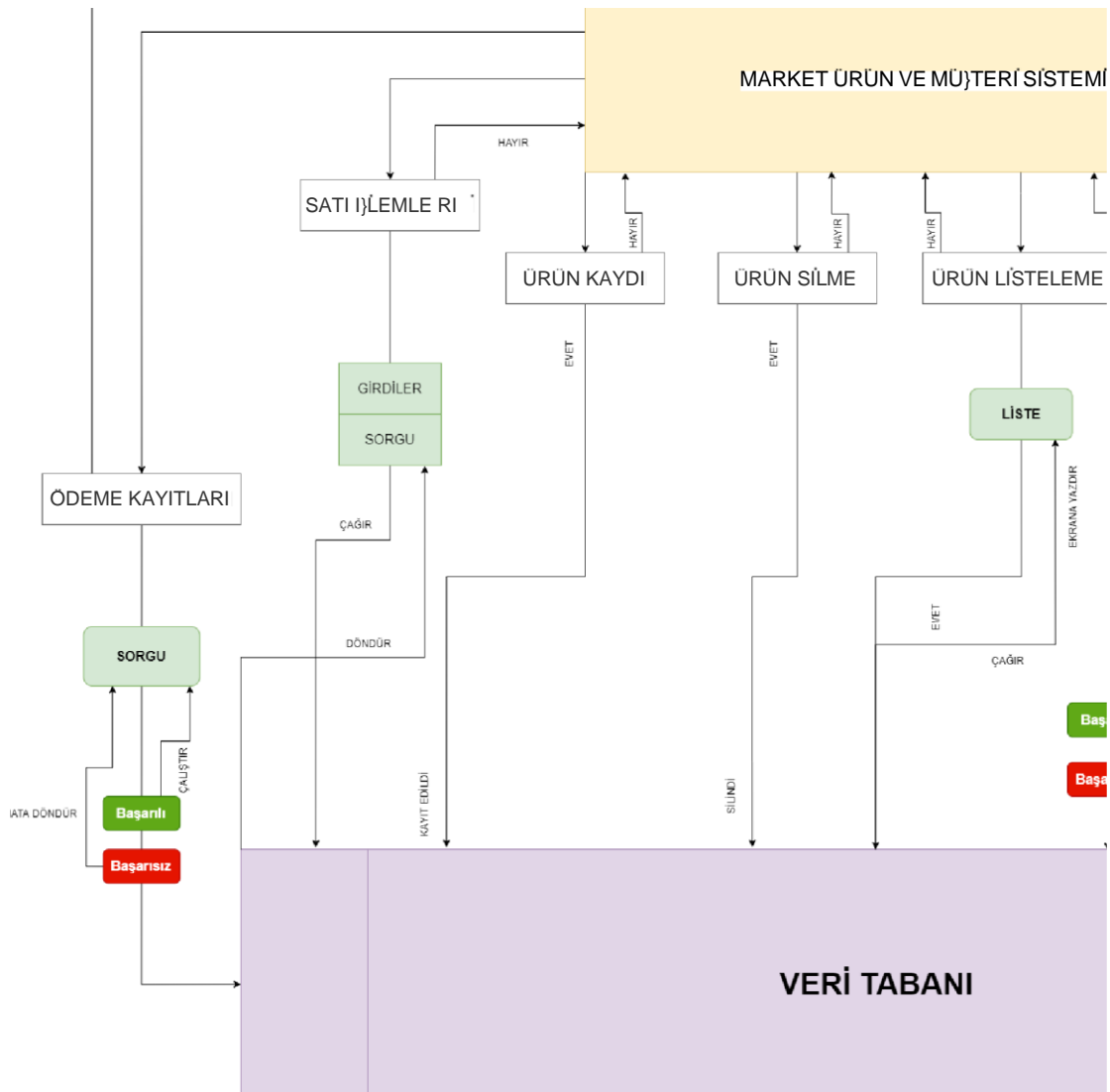
Projemiz marketlerde satıcıların işlerini kolaylaştırmak adına yapılmıştır. Marketin sattığı ürünler kayıt altında ve ödeme yapmayan müşteriler kayıt altında bulundurarak eski usul bakkal defteri mantığının daha güvenilir bir şekilde çalışmasını, dijital ortamda kayıt altına alarak sağlamaktır.

Fatura kaydı tutma, müşterilerin bilgilerini tutma ve ürün kaydı tutmamızı kolaylaştıran konsol uygulamamızı yaptık.

2. GERÇEKLEŞTİRİLEN PROJE ÇALIŞMASI

Projenin akış diyagramı (Şekil.1) aşağıda vurgulanmıştır. Diyagram doğrultusunda, kodlanmış olan konsol uygulamasının çalışma mantığı ve veri tabanı bağıntısı anlatılmaktadır. Kısa ve öz olmasının akış diyagramındaki önemine binaen, kullanıcı dostu bir ara yüz, projenin gerçek hayatta da kullanıma sunulabilecek bir esnekliğinin olmasını sağlamaktadır. Ürün ve müşteri kolay takip; istenildiğinde çağrılacak veriler kullanımının pratik açısından değerini arttırmaktadır.

Ara yüzden yapılan işlemler, veri tabanı çağrışımı ve kodların yerinde kullanılmasıyla ilgili kayıtları bulmak, derlemek ve güncellemek üzerine kurulmuştur.



3. SONUÇ ve KAZANIMLAR

Bu projeyi yaparken C dili üzerinde pek çok konuyu öğrenmiş olduk. C diline daha çok hakim olduk. Projemizin gerçek hayata uyarlanabilir olması aslında soyut işlemler yaptığımız bilgisayarlarımız somut dünya gereksinimlerini karşıladığını gördük. C'de dosya işlemlerinin nasıl çalıştığını, göstericilerin(pointer) ve fonksiyonların nasıl tanımlanması gerektiğini öğrendik. Çeşitli fonksiyon tiplerini görmüş olduk. Döngüler, koşul ifadeleri gibi ders içi konularımızın bir program içinde nasıl çalıştığını görmüş olduk. Programlama bilgimizin artmasının yanı sıra grup içi senkronize çalışmamızı geliştirmiş olduk.

4. KAYNAKÇA

Programlamayı C İle Öğreniyorum

Codewithc.com

Youtube yazılım bilimi kanalı

<https://www.sanfoundry.com/c-programming-examples-file-handling/>

<http://kovan.ceng.metu.edu.tr/~sinan/ceng230/index.html>

https://www.bilgigunlugum.net/prog/cprog/c_dosya

<https://www.reviversoft.com/tr/file-extensions/dat>

<https://caylakyazilimci.com/>

stackoverflow.com

github.com

EK

PROGRAM KODLARIMIZ

```
#include<stdio.h> // standart input output dosyalarını ekler.
```

```
#include<string.h> //karakter dizileri ile ilgili fonksiyon, veri türü ve makro tanımlamaları içerir.
```

```
#include<conio.h> //çoğunlukla MS-DOS derleyicileri tarafından konsol girdi/çıkı işlemleri için sunulan bir başlık dosyasıdır.
```

```
#include<stdbool.h> //ikili sayı sistem veri türü değerleri ile ilgili tanımlamalar C99 standartları ile birlikte standart kütüphaneye katılan stdbool.h başlık dosyasında yapılmıştır.
```

```
#include<ctype.h> //karakterlerle ilgili fonksiyon tanımlamaları Standart Kütüphanedeki ctype.h başlık dosyasında yer almaktadır.
```

```
#include<stdlib.h> //dinamik bellek yönetimi, rasgele sayı üretimi, çevre ile iletişim, tamsayı aritmetiği, arama, sıralama ve dönüştürme gibi çeşitli genel amaçlı fonksiyonlar tanımlanmıştır.
```

```
FILE *fo,*fg,*fp,*fp1;
```

```
//FILE, stdio.h dosyası içinde tanımlanmış olup dosyanın boyutu, aktif konumu ve giriş modları gibi dosya hakkındaki değişik değerleri içeren bir yapıdır.
```

```
//fgetc() ve fputc() fonksiyonları: genel yapısı fp*,fonksiyonu kullanarak dosyaya bir karakter yazabilir veya dosyadan bir karakter okuyabiliriz
```

```
//fgetc() fonksiyonu, aşağıda belirtilen 2 farklı durumda, EOF değerini geri verir:
```

//1. Bir hata meydana geldiğinde

//2. Dosya sonuna gelindiğinde

//fgetc() fonksiyonu EOF değerini geri verdiğinde, yukarıdaki durumlardan hangisine göre bu değeri geri verdiğini bilemeyiz. EOF değerinin fgetc() fonksiyonu tarafından hangi nedenle geri verildiğini anlayabilmek amacıyla aşağıda ana yapıları verilen feof() ve ferror() fonksiyonlarını programlarımızda kullanabiliriz

struct urunler //Türleri farklı olan verileri tek bir isim altında toplayabilmemiz için ise, yapı (structure) adı verilen kullanıcı tanımlı veri türlerini kullanabiliriz.

```
{
    int kodu;
    int durum;
    char adi[15];
    char cinsi[15];
    float alis;
    float satis;
}; struct urunler urun,gec;
```

struct musteriler

```
{
    int hesap;
    int durum;
    char adi[20];
    char soyadi[20];
    char adres[50];
    char mail[20];
    long long int telefon;
}; struct musteriler islem1, gec1;
```

void input(); //Fonksiyon prototipi

void writefile();

void search();

void output();

```
void urun_listeleme();
```

```
struct tarih{
    int ay;
    int gun;
    int yil;
};
```

```
struct hesap {
    int sayi;
    char ad[100];
    char subeAdi[100];
    char hesapdurumu;
    float toplamtutar;
    float kalantutar;
    float odeme;
    struct tarih odemetarihi;
}musteri;

int tl,sl,ts;
```

```
void urun_kayit()
```

$$\{$$

```
int secim1;
```

[illegible]

```
printf("\n * * * * * \n\n");
```

fo = fopen("urun.txt", "a"); // "a" komutu = Bir metin dosyasını ekleme yapmak için açar. Dosya yok ise oluşturulur.

```
urun_kayit();
```

```

    }
    if (secim1 == 2)
    {
        system("cls");
        int main();
    }

}

void urun_guncelleme()
{
    int kodu;
    int secim5;
    int bas=1, urun_varmi=0;

    fo = fopen("urun.txt", "r"); // "r" = Okuma için bir metin dosyası açar. Dosya mevcut
    olmalıdır.

        printf("Guncellenecek urunun kodunu girin :");
        scanf("%d", &kodu);

// fscanf = Dosya akışından (stream) veri okur.

// Dosya akışından okunan değerleri, format parametresi ile gösterilen karakter dizisindeki
format tanımlayıcılarının sırasına ve veri yapısına uygun olarak, üç nokta (...) ile ifade edilen ve
format parametresinden sonra aynı sırayla yer alan değişken parametrelere aktarır.

        while(fscanf(fo, "%d %s %s %f %f %d", &urun.kodu, &urun.adi,
&urun.cinsi, &urun.alis, &urun.satis, &urun.durum)!= EOF) // EOF dosyanın bittiği haberini bize
vermektedir.

        {
            if(urun.kodu=kodu && urun.durum==1)
            {
                printf("Urun Basari ile Eslesti\n");
                urun_varmi=1;
                break;
            }
            else
                printf("Urun Eslesmedi Bekleyin...\n");
        }
}

```

```
printf("Arama islemi sona erdi.\n");
```

```
fclose(fo);
```

```
fo=fopen("urun.txt","r");
```

```
fg=fopen("gecici.txt","w+"); // "w+" = Okuma ve yazma için bir metin dosyası oluşturur.
```

```
if(urun_varmi==1)
```

```
{
```

```
    printf("Yeni ad :");
```

```
    scanf("%s", &gec.adi);
```

```
    printf("Yeni cins :");
```

```
    scanf("%s", &gec.cinsi);
```

```
    printf("Yeni alis fiyatı :");
```

```
    scanf("%f", &gec.alis);
```

```
    printf("Yeni satis fiyatı :");
```

```
    scanf("%f", &gec.satis);
```

```
        while(fscanf(fo, "%d %s %s %f %f %d", &urun.kodu, &urun.adi,  
&urun.cinsi, &urun.alis, &urun.satis, &urun.durum)!= EOF)
```

```
        {
```

```
            if(urun.kodu==urun.kodu && urun.durum==1) //ürün kodunun  
eşleşip eşlemediğinin kontrolü burada yapılır.
```

```
            continue;
```

```
            if(bas==1)
```

```
            {
```

```
                fprintf(fg, "\n%d %s %s %f %f 1", kodu, gec.adi,  
gec.cinsi, gec.alis, gec.satis);
```

```
                bas=0;
```

```
            }
```

```
        fprintf(fg, "\n%d %s %s %f %f %d", urun.kodu, urun.adi, urun.cinsi,  
urun.alis, urun.satis, urun.durum);
```



```
}
```

```
fclose(fo);
```

```
fclose(fg);
```

```
printf("\n Kayit guncellendi\n\n");
```

```
remove("urun.txt");
```

```
rename("gecici.txt","urun.txt"); //güncelleme işlemi yeni dosyaya yazdırılır.
```

```
}
```

```
printf("\n Baska ne yapmak istersiniz?\n\n");
```

```
printf("1-Yeni Guncelleme \n");
```

```
printf("2-Ana Menu\n");
```

```
printf ("\n *****\n");
```

```
scanf("%d", &secim5);
```

```
if (secim5 == 1)
```

```
{
```

```
    system("cls");
```

```
    urun_guncelleme();
```

```
}
```

```
if (secim5 == 5)
```

```
{
```

```
    system("cls");
```

```
    int main();
```

```
}
```

```
}
```

```
void urun_silme()
```

```
{
```

bool kayit=false; //Bool veri tipi: "Doğru" ve "Yanlış" olmak üzere iki değer alan veri tipidir. Uygulama içerisinde true(doğru) ve false(yanlış) değer ataması yapılmaktadır. Kontrol işlemlerinde sıklıkla kullanılmaktadır.

```
int kodu;
```

```
int secim3;
```

```
fo=fopen("urun.txt","r");
```

```

fg=fopen("gecici.txt","w+");
printf("Dikkat bu islem asla geri alinamaz!!!\nSilme istediginiz urunun kodunu giriniz:");
scanf("%d", &kodu);

while(fscanf(fo, "%d %s %s %f %f %d", &urun.kodu, &urun.adi, &urun.cinsi,
&urun.alis, &urun.satis, &urun.durum)!= EOF)
{

    if (kodu==urun.kodu && urun.durum==1)
    {
        kayit=true;
        fprintf(fg, "\n%d %s %s %f %f 0", urun.kodu, urun.adi, urun.cinsi,
urun.alis, urun.satis);
        system("cls");
        printf("Kayit basariyla silindi.");
        printf("\n Baska ne yapmak istersiniz?\n\n");
        printf("1-Yeni Silme Islemi \n");
        printf("2-Ana Menu\n");

        printf("\n *****\n");
        scanf("%d", &secim3);
        if (secim3 == 1)
        {
            system("cls");
            urun_silme();
        }
        if (secim3 == 2)
        {
            system("cls");
            int main();
        }

    }

    else
    {
        fprintf(fg, "\n%d %s %s %f %f %d", urun.kodu, urun.adi, urun.cinsi,
urun.alis, urun.satis, urun.durum);

```

```

        }

    }

    if(kayit==false)
    {
        system("cls");

        printf("Boyle bir urun yok!\n");
        printf("\n Baska ne yapmak istersiniz?\n\n");
        printf("1-Yeni Silme Islemi \n");
        printf("2-Ana Menu\n");


        printf("\n *****\n");
        scanf("%d", &secim3);
        if (secim3 == 1)
        {
            system("cls");
            urun_silme();
        }
        if (secim3 == 2)
        {
            system("cls");
            int main();
        }

    }

    fclose(fo);
    fclose(fg);
    remove("urun.txt");
    rename("gecici.txt","urun.txt");
}

```

```

void urun_listeleme()
{
    int secim4;

    fo=fopen("urun.txt","r");

```

while(!feof(fo)) // Bu ifade dosyanın sonuna gelinip gelinmediğini kontrol eder.
Geldiğinde durur.

```
{  
    fscanf(fo,"%d %s %s %f %f %d", &urun.kodu, urun.adi, urun.cinsi, &urun.alis,  
&urun.satis, &urun.durum);  
    if(urun.durum!=0)  
    {  
        printf("\nListelenen Urunler\n");  
        printf ("\n *****\n");  
        printf("\n KODU: %d ADI: %s CINSI: %s ALIS: %.2f SATIS: %.2f", urun.kodu,  
urun.adi, urun.cinsi, urun.alis, urun.satis);  
    }  
}  
fclose(fo);  
printf ("\n *****\n");  
printf("\n\n Baska ne yapmak istersiniz?\n\n");  
printf("1-Ana Menu\n");  
  
printf ("\n *****\n");  
scanf("%d", &secim4);  
if (secim4 == 1)  
{  
    system("cls");  
    int main();  
}  
  
}
```

void urun_sorgulama()

```
{  
  
    int kodu;  
    int secim5;  
    fo=fopen("urun.txt","r");
```

```

printf("Sorgulanacak olan urunun kodunu girin:");
scanf("%d",&kodu);

while(fscanf(fo,"%d %s %s %f %f %d",&urun.kodu, &urun.adi, &urun.cinsi, &urun.alis,
&urun.satis, &urun.durum)!=EOF)
{
    if(urun.kodu==kodu && urun.durum==1)
    {
        printf("Urun basariyla eslesti.\n%s %s %.2f %.2f\n", urun.adi, urun.cinsi, urun.alis,
urun.satis);
        break;
    }
    else
        printf("Urun bulunamadi.Bekleyiniz...\n\n");
}
printf("Sorgulama islemi sona erdi.\n");
printf("\n\n Baska ne yapmak istersiniz?\n\n");
printf("1-Ana Menu\n");
printf("2-Yeniden Sorgulama\n");

printf ("\n *****\n");
scanf("%d", &secim5);
if (secim5 == 1)
{
    system("cls");
    int main();
}
if (secim5 == 2)
{
    system("cls");
    urun_sorgulama();
}

```

```
}
```

```
void satis()
```

```
{
```

```
    int kodu;
```

```
    int adet;
```

```
    float tutar;
```

```
    char devam = 'E';
```

```
    while(devam!='H')
```

```
    {
```

```
        fo=fopen("urun.txt","r");
```

```
        printf("Urunun kodunu girin:");
```

```
        scanf("%d",&kodu);
```

```
        while(fscanf(fo,"%d %s %s %f %f %d", &urun.kodu, &urun.adi, &urun.cinsi,  
&urun.alis, &urun.satis, &urun.durum)!=EOF)
```

```
        {
```

```
            if(urun.kodu==kodu && urun.durum==1)
```

```
            {
```

```
                printf("Adet girin :");
```

```
                scanf("%d", &adet);
```

```
                tutar+=urun.satis*adet; //tutar hesaplamasının tanımı
```

```
                printf("Tutar = %.2f dir\n", tutar);
```

```
            }
```

```
            else
```

```
                printf("Urun bulunamadi.Bekleyiniz...\n");
```

```
        }
```

```
        printf("Devam etmek isiyor musunuz (E/H) :");
```

```
        scanf("%s", &devam);
```

```
        system("cls");
```

```
}
```

```
printf("Odemeniz gereken toplam tutar %.2f dir.", tutar);
```

```
}
```

```
void musteri_ekleme()
```

```
{
```

```
    int secim6;
```

```
    fp = fopen("musteri.txt","a"); // "a": Bir metin dosyasını ekleme yapmak için açar. Dosya yok ise oluşturulur.
```

```
    printf("Hesap No:");
```

```
    scanf("%d",&islem1.hesap);
```

```
    printf("Musteri Adi:");
```

```
    scanf("%s",&islem1.adi);
```

```
    printf("Musteri Soyadi:");
```

```
    scanf("%s",&islem1.soyadi);
```

```
    printf("Adresi:");
```

```
    scanf("%s",&islem1.adres);
```

```
    printf("Mail:");
```

```
    scanf("%s",&islem1.mail);
```

```
    printf("Telefon:");
```

```
    scanf("%11d",&islem1.telefon);
```

```
    fprintf(fp, "\n %d %s %s %s %s %11d 1", islem1.hesap, islem1.adi, islem1.soyadi, islem1.adres, islem1.mail, islem1.telefon);
```

```
    fclose(fp);
```

```

system("cls");
printf ("\n *****\n");
printf ("\n Kayit basariyla olusturuldu.\n Baska ne yapmak istersiniz?\n\n");
printf ("1-Yeni Kayit \n");
printf ("2-Ana Menu\n");

printf ("\n *****\n");
scanf ("%d", &secim6);
if (secim6 == 1)
{
    system("cls");
    musteri_ekleme();
}
if (secim6 == 2)
{
    system("cls");
    int main();
}
}

```

//Bu kısımda müşterilerimizin bakiyelerini de düzenleyebildiğimiz bir yapı oluşturmak istedik. Ancak aldığımız küçük bir hatayı çözemediğimiz için yorum satırı olarak bırakmak istedik.

```

/*void musteri_guncelleme()
{
    int hesap;
    int kosul = 1;
    int musteri_varmi=0;
    float tutar;

    fp = fopen("musteri.txt","r");

    printf("Guncellenecek Musteri Hesap Numarasi Girin:");

```



```

scanf("%d", &hesap);

while(fscanf(fp, "%d %s %s %f %f %f %s %d", &islem1.hesap, &islem1.adi, &islem1.soyadi,
&islem1.bakiye, &islem1.odeme, &islem1.guncel, &islem1.tarih, &islem1.durum) != EOF);
{
    if(islem1.hesap==hesap && islem1.durum==1)
    {
        printf("Musteri kaydi basariyla bulundu.\n");
        musteri_varmi = 1;

    }
    else
        printf ("Musteri kaydi bulunamadi, lutfen bekleyin...\n");
}

printf ("Arama islemi sona erdi. \n");

fclose(fp);
fp = fopen("musteri.txt", "r");
fp1 = fopen("gecici_musteri.txt", "a");

if(musteri_varmi==1)
{
    printf("Guncel borcunuz %.2f dir.\n", islem1.guncel);
    printf("Odeme tutarini giriniz:");
    scanf("%f",&tutar);
    islem1.guncel = islem1.guncel - tutar;
    printf("Kalan borcunuz %.2f dir\n", islem1.guncel);
    printf("Tarih (gg/aa/yyyy) :");
    scanf("%s",islem1.tarih);
    gec1.hesap = islem1.hesap;
    gec1.bakiye = islem1.bakiye;
    gec1.odeme = islem1.odeme;
    gec1.guncel = islem1.guncel;

    fprintf(fp1, "\n %d %s %s %f %f %f %s 1", gec1.hesap, islem1.adi, islem1.soyadi,
gec1.bakiye, gec1.odeme, gec1.guncel, gec1.tarih);

```

```

    }

}*/

void musteri_sorgulama()
{
    int secim7;

    int hesap;

    fp = fopen("musteri.txt", "r");

    printf("Sorgulanacak Musterinin Hesap Numarasini Giriniz:");

    scanf("%d",&hesap);

    while(fscanf(fp, "%d %s %s %s %s %11d %d", &islem1.hesap, &islem1.adi,
&islem1.soyadi, &islem1.adres, &islem1.mail, &islem1.telefon, &islem1.durum) != EOF);
    {

        if(islem1.hesap == hesap && islem1.durum == 1)

        {

            printf("Musteri Basariyla Eslesti\n%d %s %s %s %s %11d ", islem1.hesap,
islem1.adi, islem1.soyadi, islem1.adres, islem1.mail, islem1.telefon);

        }

        else

            printf("\nMusteri Bulunamadi\n");

    }

    printf("\nSorgulama islemi sona erdi.\n");

    printf ("\n *****\n");

    printf("\n Baska ne yapmak istersiniz?\n\n");

    printf("1-Yeni Sorgulama \n");

    printf("2-Ana Menu\n");

    printf ("\n *****\n");

    scanf("%d", &secim7);

    if (secim7 == 1)

    {

        system("cls");

```

```

    musteri_sorgulama();
}
if (secim7 == 2)
{
    system("cls");
    int main();
}
}

void musteri_silme ()
{
    bool musteri = false;
    int hesap;
    int secim8;
    int yanit;
    fp = fopen("musteri.txt", "r");
    fp1 = fopen("gecici_musteri.txt", "w+"); //silme işleminde aracı olması için.
    printf("Dikkat! Bu işlem geri alınamaz! \n Ana menüye dönmek için -> 0 \n İşleme devam etmek için -> 1 tuslayınız:");
    scanf("%d", &yanit);
    system("cls");
    //if (yanit == 0) gerek var mı?

    if (yanit == 1)
    {
        printf("Silmek istediğiniz müşterinin hesap numarasını giriniz:");
        scanf("%d", &hesap);
        while(fscanf(fp, "%d %s %s %s %s %11d %d", &islem1.hesap, &islem1.adi, &islem1.soyadi, &islem1.adres, &islem1.mail, &islem1.telefon, &islem1.durum) != EOF);
        {
            if (hesap == islem1.hesap && islem1.durum == 1)
            {
                musteri = true;
                fprintf(fp, "\n %d %s %s %s %s %11d 0", islem1.hesap, islem1.adi, islem1.soyadi, islem1.adres, islem1.mail, islem1.telefon);
                printf("Musteri başarıyla silindi.\n");
            }
        }
    }
}

```

```

printf("\n Baska ne yapmak istersiniz?\n\n");
printf("1-Yeni Silme Islemi \n");
printf("2-Ana Menu\n");

printf ("\n *****\n");
scanf("%d", &secim8);
if (secim8 == 1)
{
    system("cls");
    musteri_silme();
}
if (secim8 == 2)
{
    system("cls");
    int main();
}

    }
    else
    {

        fprintf(fp, "\n %d %s %s %s %s %11d %d", islem1.hesap,
islem1.adi, islem1.soyadi, islem1.adres, islem1.mail, islem1.telefon, islem1.durum);

    }

}

if (musteri == false)
{

    printf("Boyle bir musteri kaydi bulunamadi.\n");

printf("\n Baska ne yapmak istersiniz?\n\n");
printf("1-Yeni Silme Islemi \n");
printf("2-Ana Menu\n");

printf ("\n *****\n");
scanf("%d", &secim8);
if (secim8 == 1)
{

```

```

        system("cls");
        musteri_silme();
    }
    if (secim8 == 2)
    {
        system("cls");
        int main();
    }

    }

}

fclose(fp);
fclose(fp1);
remove("musteri.txt");
rename("gecici_musteri.txt", "musteri.txt");
}

void musteri_listeleme()
{
    fp=fopen("musteri.txt","r");

    while(!feof(fp)) // bu dosyanın sonuna gelinip gelinmediğini kontrol eder.
    geldiğinde durur.
    {
        fscanf(fp,"%d %s %s %s %s %11d %d", &islem1.hesap, islem1.adi, islem1.soyadi,
        &islem1.adres, &islem1.mail, &islem1.telefon, &islem1.durum);
        if(islem1.durum!=0)
        {
            printf("\n%d %s %s %s %s %11d", islem1.hesap, islem1.adi,
            islem1.soyadi, islem1.adres, islem1.mail, islem1.telefon);
        }
    }
    fclose(fp);
}

```

$$\}$$

```
int main ()
```

 $\{$

```
int secim;
```

```
int i,n;
```

```
char ch;
```

while (1)

$$\{$$
[illegible]

```
printf("\n * * * * *");
```

```
printf ("\n Ana Menu\n");
```

```
printf ("1- Urun kaydi \n");
```

```
printf ("2- Urun guncelleme\n");
```

```
printf ("3- Urun silme\n");
```

```
printf ("4- Urun listesi alma\n");
```

```
printf("5- Urun sorgulama\n");
```

```
printf("6- Satis islemleri\n");
```

```
printf("7- Musteri Odeme(Fatura) Kayitlari\n");
```

```
printf("8- Musteri Ekleme\n");
```

```
printf("9- Musteri Sorgulama\n");
```

```
printf("10- Musteri Silme\n");
```

```
printf("11-Musteri Listeleme\n");
```

```
printf ("0- Cikis\n");  
printf ("***** \n");  
printf ("Yapmak istediginiz islemi tuslayiniz:\n");  
scanf("%d", &secim);
```

```
if (secim == 1)  
{  
    urun_kayit();  
}
```

```
if (secim == 2)  
{  
    urun_guncelleme();  
}
```

```
if (secim == 3)  
{  
    urun_silme();  
}
```

```
if (secim == 4)  
{  
    urun_listeleme();  
}
```

```
if (secim == 5)  
{  
    urun_sorgulama();  
}
```

```
if (secim == 6)  
{  
    satis();  
}
```

```

if (secim==7)
{
system("cls");

printf("\n\n\n\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2
MUSTERI ODEME TAKIP
SISTEMI\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\n\n\n")
;

printf("=====\n");
printf("\n1:Yeni Musteri Hesabi ekleme\n");
printf("\n2:Musteri hesabi sorgulama \n");
printf("\n3:exit\n");
printf("\n=====\n");
do{
printf("\nYapmak istediginiz islemi seciniz?");
ch=getche(); // getche() Fonksiyonu klavyeden girilen tek bir karakteri okur
}while(ch<='0' || ch>'3');
switch(ch){
case '1':

printf("\nGirilecek hesap sayisi:");
scanf("%d",&n);
for(i=0;i<n;i++){
input();
// İNPUT BURDA ÇAĞRILMIŞ
if(musteri.odeme>0)

musteri.hesapdurumu=(musteri.odeme<musteri.toplamtutar)? 'B': 'D';// C Dilinde "?"
operatöründen sonra gelen işlemlerde eğer koşul doğruysa soldakini doğru değilse sağdaki işlemi
yapar

else

musteri.hesapdurumu=(musteri.kalantutar>0) ? 'B' : 'D';

musteri.kalantutar=musteri.toplamtutar - musteriodeme;
//müşterilerin borç durumunu hesaplayan kısım

writefile();
// WRITEFILE BURDA ÇAĞRILMIŞ
}

main();

```



```
case '2':
```

```
    printf("Arama yontemi seciniz\n");
```

```
    printf("\n1 --- Musteri numarasi ile arama\n");
```

```
    printf("2 --- Musteri adi ile arama\n");
```

```
    search();
```

```
    // SEARCH BURADA ÇAĞRILMIŞ
```

```
    ch=getche();
```

```
    main();
```

```
    system("cls");
```

```
}
```

```
}
```

```
if(secim==8)
```

```
{
```

```
    musteri_ekleme();
```

```
}
```

```
if (secim==9)
```

```
{
```

```
    musteri_sorgulama();
```

```
}
```

```
if (secim==10)
```

```
{
```

```
    musteri_silme();
```

```
}
```

```
if (secim==11)
```

```
{
```

```
    musteri_listeleme();
```

```
}
```

```
if (secim == 0)
```

```

        break;

    }

return 0;

}

```

//Windows işletim sistemi içerisinde kullanılan bir dosya uzantısıdır. "Data" yani "Veri" kelimesinin kısaltılmışıdır ve içerisinde genellikle yazılımlara dair veri kodlarını içerirler.

//Birçok program DAT dosyalarını oluşturur, açar ve referans alır. Sadece uygulama tarafından kullanılmak üzere tasarlanmış ve kullanıcı tarafından manuel olarak açılmaz.

```

void input()
{
    FILE *fp=fopen("dosya.dat","rb"); // Binary dosyalarında rb modu mevcut dosyayı hem
    okuma hem de yazma için açar. Dosya mevcut değilse geriye NULL değeri dönderir.

    fseek(fp,0,SEEK_END); // fseek fonksiyonu dosyaya yazma ve dosyadan okuma
    işleminden önce istenilen kayda ulaşmak için kullanılır. SEEK_END Başlangıç noktasının dosyanın
    sonu olduğu anlamına gelir.

    tl=ftell(fp);

    sl=sizeof(musteri); //unary operatör türüdür. Önüne yazıldığı oprandın bellekte
    kapladığı alanın byte cinsinden miktarını üretir.

    ts=tl/sl;

    fseek(fp,(ts-1)*sl,SEEK_SET); // Başlangıç noktasının dosyanın başı olduğu anlamına gelir.

    fread(&musteri,sizeof(musteri),1,fp);

//Her biri size parametre değeri kadar bayt boyutunda olan count parametre değeri kadar
    elemanı stream parametresi ile gösterilen akıştan okur ve bunları ptr parametresi ile gösterilen
    belleğe yükler.

//Akışın konum göstergesi, okunan toplam bayt miktarı kadar (size * count) ile ileri alınır.

```

```

printf("\n        Musteri no:%d\n",++musteri.sayi);

fclose(fp);

printf("\n        Musteri adi:");

scanf("%s",musteri.ad);

printf("        Sube:");

scanf("%s",musteri.subeAdi);

printf("        Odenecek tutar:");

scanf("%f",&musteri.toplamTutar);

```

```

        printf("        Odeme Miktari:");
        scanf("%f",&musteri.odeme);
        printf(" Odeme Tarihi(mm/dd/yyyy):");

scanf("%d/%d/%d",&musteri.odemetarihi.ay,&musteri.odemetarihi.gun,&musteri.odemetarihi.yi
l);

        return;
    }

void writefile()
{
    FILE *fp;
    fp=fopen("dosya.dat","ab");
    fwrite(&musteri,sizeof(musteri),1,fp);
    fclose(fp);
    return;
}

void search()
{
    char ch;
    char nam[100];
    int n,i,m=1;
    FILE *fp;
    fp=fopen("dosya.dat","rb");
    do{
        printf("\nseciminizi giriniz:");
        ch=getche();
    }while(ch!='1' && ch!='2');
    switch(ch){
        case '1':
            fseek(fp,0 /* Başlangıç noktasının dosyanın başı olduğu anlamına gelir.
            */,SEEK_END); // SEEK_END Başlangıç noktasının dosyanın sonu olduğu anlamına gelir
            tl=ftell(fp);
            sl=sizeof(musteri);
            ts=tl/sl;

```

```

do{
    printf("\nmusteri numarinizi giriniz:");
    scanf("%d",&n);
    if(n<=0 || n>ts)
        printf("\nyanlis musteri numarası\n");
    else{
        fseek(fp,(n-1)*sl,SEEK_SET);
        fread(&musteri,sl,1,fp);
        output();
    }
    printf("\n\nyeniden denemek ister misiniz?(e/h)");
    ch=getche();
}while(ch!='e');
fclose(fp);
break;
case '2':
    fseek(fp,0,SEEK_END);
    tl=ftell(fp);
    sl=sizeof(musteri);
    ts=tl/sl;
    fseek(fp,(ts-1)*sl,SEEK_SET);
    fread(&musteri,sizeof(musteri),1,fp);
    n=musteri.sayi;

```

```

do{
    printf("\nMusteri adi giriniz:");
    scanf("%s",nam);
    fseek(fp,0,SEEK_SET);
    for(i=1;i<=n;i++)
    {
        fread(&musteri,sizeof(musteri),1,fp);

```

if(strcmp(musteri.ad,nam)==0) ///strcmp: Str1 parametresi ile gösterilen karakter dizisini str2 parametresi ile gösterilen karakter dizisi ile karşılaştırır. Eğer her iki karakter dizisindeki tüm karakterler aynı ise sıfır değeri, aksi takdirde sıfırdan farklı bir değeri geri döndürür.

//Fonksiyon str1 ve str2 parametreleri ile gösterilen karakter dizilerini ilk karakterden itibaren

karşılaştırmaya başlar. Birbirlerine eşitse, karakterler farklı olana veya karakter dizisinin sonunu gösteren boş bir karakterle karşılaşılan kadar karşılaştırma işlemine devam eder.

//Eğer str1 ve str2 parametreleri boş karakter ile sonlandırılmış karakter dizilerini gösteren bir işaretçi değilse, anlamsız bir sonuç elde edilir.

```
        {
            output();
            m=0;
            break;
        }
    }
    if(m!=0)
        printf("\n\nMusteri adi bulunamadi.\n");
        printf("\nYeniden denemek ister misiniz?(e/h)");
        ch=getche();
    }while(ch=='e');
    fclose(fp);
}
return;
}
```

```
void output()
{
    printf("\n\n  Musteri no:           :%d\n",musteri.sayi);
    printf("  Musteri Adi           :%s\n",musteri.ad);
    printf("  sube adi                 :%s\n",musteri.subeAdi);
    printf("  Odenmesi gereken tutar      :%.2f\n",musteri.toplamtutar);
    printf("  Odeme Miktari              :%.2f\n",musteri.odeme);
    printf("  Kalan tutar                :%.2f\n",musteri.kalantutar);
    printf("  Odenen tarih\n\n");
    printf(":%d/%d/%d\n\n",musteri.odemetarihi.ay,musteri.odemetarihi.gun,musteri.odemetarihi.yil);
    switch(musteri.hesapdurumu)
    {
        case 'B':
            printf("\t Hesap Durumu: BORCLU \n\n");
```

```
        printf("\t Borc miktari: %.2f",musteri.kalantutar);  
        break;  
    case 'D':  
        printf("\t Hesap Durumu: BORC BULUNMAMAKTADIR \n\n");  
        break;  
    default:  
        printf("HATA\\n\n");  
    }  
    return;  
}
```