T.C

İSTANBUL ESENYURT ÜNİVERSİTESİ

SİNEMA REZERVASYON OTOMASYONU

AMACI

* En iyi hizmetle müşterilerin sorunsuz bir şekilde bilet alıp sinemanın keyfini çıkarmalarını sağlamak
* Personelin daha basit ve işlevsel şekilde çalışmasını sağlamak ve denetilmesini sağlamak
* *Rezarvasyon işlemlerinde hem personelin hemde müşterinin madur olmamasını ve sistemsel hataların en aza indirgenmesini sağlamak*

HAZIRLADIĞIMIZ SİSTEMDE;

* 4 ana işlemden oluşur;

1-Rezervasyon yap

2-Rezervasyon iptal

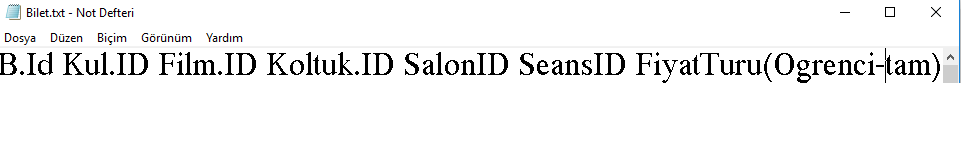
3-Film yorumları

4-Üye işlemleri

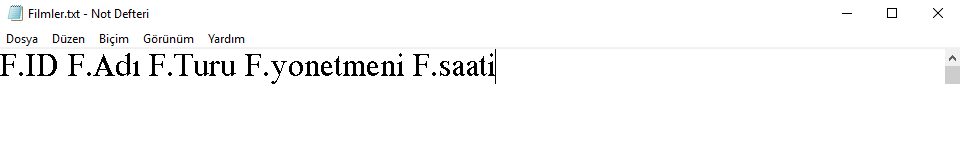
* Kullanıcılar üye olması zorunludur.
* Kullanicilar üye olmadan yapacağı tek şey izleyeceği filmin yorumlarını görebilir.
* Kullanicilar üye olduktan sonra ise rezervasyon yapabilir,yaptığı rezervasyonun iptali gerçekleştirebilir,filmi izledikten sonra filme yorum yazabilir,yorumları görebilir ve üye işlemlerini gerçekleştirebilir.
* Rezervasyon yap işleminde yapılanlar ise filmi,salonu,seans seçtikten sonra oturumu açmışsa koltuk seçme işlemine gider , koltuk seçimini yaptıktan sonra rezervasyonu onaylarsa rezervasyonu gerçekleştirmiş olur.
* Rezervasyon iptal işleminde ise yapmış oldugu rezervayonu silebilir.
* Üye işlemlerinde yapacak işlemler ise Kullanici oluşturma,şifremi unuttum ve bloke kaldırma işlemidir.

VERİLERİ TUTULMASI İÇİN GEREKEN METİN BELGELERİ

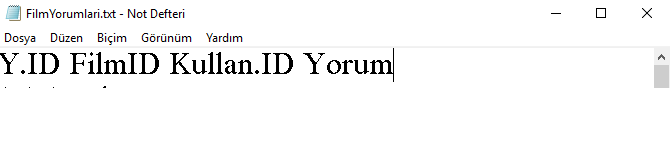
1-Bilet.txt



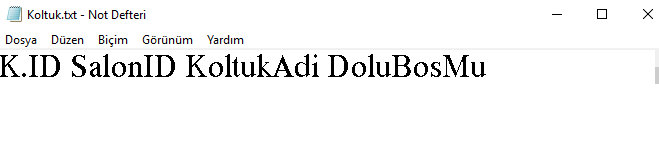
2-Filmler.txt



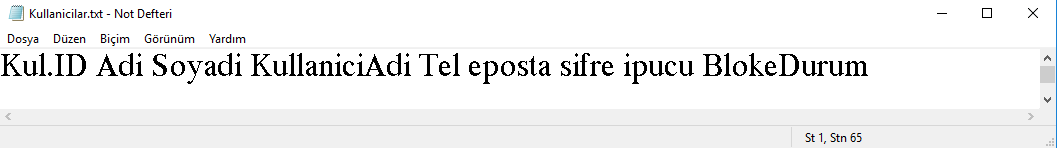
3-FilmYorumları.txt



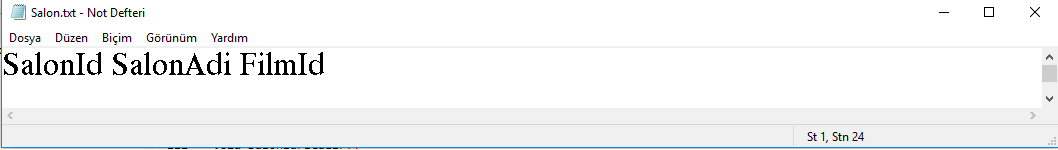
4-Koltuk.txt



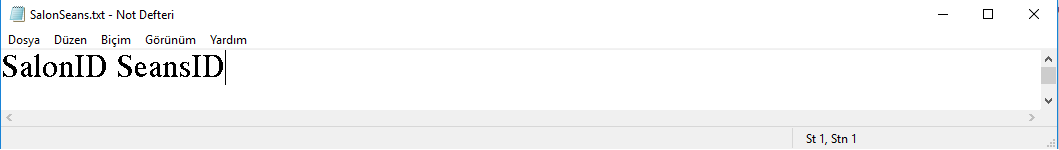
5-Kullanicilar.txt



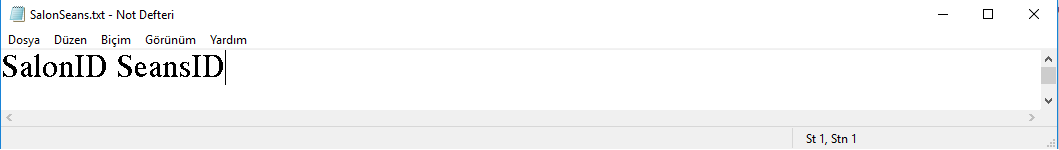
6-Salon.txt



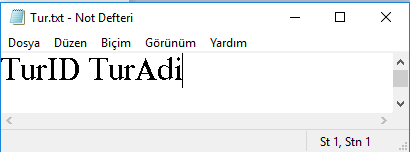
7-SalonSeans.txt



8-Seans.txt



9-Turler.txt



STRUCTLAR

struct Bilet{

int Id;

int KullaniciId;

int FilmId;

int KoltukId;

int SalonId;

int SeansId;

//Tam fiyat mı öğrenci fiyatı mı

int FiyatTuru;

};

struct Yorumlar{

int Id;

int FilmId;

int KullaniciId;

char Yorum[1000];

};

struct Kullanici{

int Id;

char Adi[100];

char Soyad[100];

char Sifre[100];

char Ipucu[20];

char KullaniciAdi[100];

char Telefon[100];

char eposta[100];

int Durum;

};

struct Tur{

int Id;

char Adi[100];

};

struct Filmler{

int Id;

char Adi[100];

int FTuru;

char FYonetmeni[100];

int Saati;

};

struct Salon{

int Id;

char Adi[100];

int FId;

};

struct Seans{

int Id;

char SeSaati[100];

};

struct Koltuk

{

int Id;

int SalonID;

char KoltukNo[3];

int Durum;

};

struct SalonSeans

{

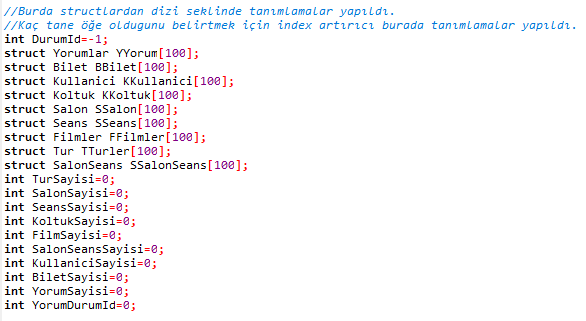
int SalonId;

int SeansId;

char Tarih[11];

};

GLOBAL TANIMLAMALAR



STRUCTLARA VERİ CEKME

//Biletleri cağırma, islemi bilet.txt verileri BBilet structa atıyoruz.

void BiletleriCagir()

{

FILE \*fp;

char cDosyaAdi[100] ="Bilet.txt";

if((fp=fopen(cDosyaAdi,"r"))== NULL)

{

printf("Dosya acilamadi.");

}

else

{

while(!feof(fp))

{

fscanf(fp,"%d %d %d %d %d %d %d\n",&BBilet[BiletSayisi].Id,&BBilet[BiletSayisi].KullaniciId,&BBilet[BiletSayisi].FilmId,&BBilet[BiletSayisi].KoltukId,&BBilet[BiletSayisi].SalonId,&BBilet[BiletSayisi].SeansId,&BBilet[BiletSayisi].FiyatTuru);

BiletSayisi++;

}

fclose(fp);} }

//FilmYorumlar.txt verileri YYorum struct dizine atıldı.

void YorumlariCagir()

{

FILE \*fp;

char cDosyaAdi[100] ="FilmYorumlari.txt";

if((fp=fopen(cDosyaAdi,"r"))== NULL)

{

printf("Dosya acilamadi.");

}

else

{

while(!feof(fp))

{

fscanf(fp,"%d %d %d %s\n",&YYorum[YorumSayisi].Id,&YYorum[YorumSayisi].FilmId,&YYorum[YorumSayisi].KullaniciId,YYorum[YorumSayisi].Yorum);

YorumSayisi++;

}

fclose(fp);

}

}

//Tur.txt alınan veriler TTurler struct atandı.

void TurleriCagir()

{FILE \*fp;

char cDosyaAdi[100] ="Tur.txt";

if((fp=fopen(cDosyaAdi,"r"))== NULL)

{printf("Dosya acilamadi.");}

else

{ while(!feof(fp))

{

fscanf(fp,"%d %s\n",&TTurler[TurSayisi].Id,TTurler[TurSayisi].Adi);

TurSayisi++; }fclose(fp);}

}