

Metaland Proje Raporu

Taha Furkan Genç
Öğrenci No :210201077

Emir Muharrem Acar
Öğrenci No: 210201059

Özet

Bu rapor 2022-2023 öğretim yılı Kocaeli Üniversitesi 2. sınıf öğrencileri Taha Furkan Genç ve Emir Muharrem Acar tarafından Programlama Laboratuvarı 2 dersinin 3. projesi için hazırlanmıştır. İçerikte projeden genel anlamıyla bahsedilmiş olup gerçekleştirilen eylemler, verimlilik, sunum, analiz ve projenin hedefteki istenilen amaca hizmet etme şekli ve miktarından bahsedilmiştir.

I. GİRİŞ

A. Projenin Genel Tanımı

Metaland projesi bir emlak oyunudur. Projenin ana amacı veri tabanı çalışmasıdır. Oyunda Yönetici ve Oyuncu olmak üzere 2 adet kullanıcı tipi vardır. Belirtilen boyutlara göre arsalar yerleştirilir veya oluşturulur ve oyuna başlanır. Oyundaki temel mekanikler şu şekildedir: kullanıcı girişi yapmak, kullanıcı kaydı yapmak, Oyun alanı belirlemek , istenilen gün kadar gün atlamak, Yöneticide işe girmek , Herhangi bir arsada işe girmek, işten çıkmak, herhangi bir arsayı satın almak, herhangi bir işletmeden eşya satın almak, herhangi bir boş arsaya işletme kurmak.

B. Giriş

Bu kısımda projede istenilenler ve bize verilen bilgiler doğrultusunda adımların belirlenmesi sağlanacaktır. Adımlar şöyle belirlenmiştir:

- Veri tabanı Kurulumu
- Veri tabanı bağlantıları
- Oyun içerisindeki karakterlerin oluşturulması
- Kullanıcı girişi ve kaydedilişi
- Oyun içerisindeki arsaların oluşturulması
- Oyun içerisindeki yöneticinin oluşması
- Arsa satın alma
- Arsalara işletme kurma
- İşe girme
- Ürün satın alma
- Gün atlama

II. YÖNTEM

A. Veri Tabanı Kurulumu

Projenin ana amacı veri tabanı yönetimidir. Oyuncularla ve oyun içerisindeki arsalarla ilgili olan bilgilerin veri tabanında tutulması gerekiyor. Veri tabanı kurarken MySQL'den yardım aldık. Rapora ek sayfalarda kurduğumuz veri tabanının ER diyagramını ve Normalizasyon işlemlerini görebilirsiniz.

B. Veri Tabanı Bağlantıları

Veri tabanı bağlantılarını programın ilerleyen geliştirme süreçlerinde veya versiyon yenilemelerinde daha rahat kullanabilmek için ayrı bir class içerisinde temel veri tabanı işlevlerini yerine getiren metodların olduğu bir "Bağlantı" sınıfı oluşturmayı süreç için çok daha mantıklı olarak gördük.(Fig. 1) Bu Bağlantı sınıfında kullanılan metodlar şu şekildedir :

```
static ResultSet baglantiyap(String baglantiSQLkod)
static void kayitekle(String sqlkodu)
static void guncelle(String sqlkodu)
static void kayitsil(String sqlkodu)
```

Bu metodlar statik olarak yerleştirildiğinden dolayı yeni nesne oluşturma mecburiyetimiz yoktu ve bu geliştirme sürecinde çok olumlu etki etti. Veri tabanına erişebilmek için öncelikle Connection nesnesi oluşturmamız ve bu nesneye database adresimizi , şifremizi girmemiz gerekiyor. Daha sonrasında bir Statement nesnesi oluşturup bu nesneye verdiğimiz SQL kodunu ilgili metotla çağırıp ResultSet oluşturmamız gerekiyor.

```
class baglanti{
    2 using;
    static Connection myConn;
    3 using;
    static Statement myStat;

    4 using;
    static ResultSet baglantiyap(String baglantiSQLkod){
        ResultSet myRs;
        try {
            myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/prolab3denemesi", "root", "furkan254");
            myStat = myConn.createStatement();
            myRs = myStat.executeQuery(baglantiSQLkod);
            //myRs = myStat.executeQuery("SELECT * FROM prolab3denemesi.kullanicilar;");

        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
        return myRs;
    }

    5 using;
    static void kayitekle(String sqlkodu){
        System.out.println(sqlkodu);
        try {
            myStat.executeUpdate(sqlkodu);
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Fig. 1. Bağlantı Class

C. Oyun İçerisindeki Kullanıcıların Oluşturulması

Oluşturduğumuz ResultSet nesnesinin verilerini alabilmek için bir döngü açıp son satıra gelene kadar dönmelerini sağlamaktır. Bu döngüler her adımda bir satır okur. Oluşturduğumuz Kullanıcı sınıfının yapıcısına verdiğimiz parametreleri ResultSet ile veriyoruz. Bu sayede her kullanıcının bir Kullanıcı nesnesi vardır.(Fig. 2) Bu kullanıcı nesnesindeki değişkenleri programı yazarken hem Java kodunda hem de SQL kodunda kolay bir şekilde değiştirebilmemiz

```

public static void main(String[] args) throws SQLException {
    ResultSet myRs = baglanti.baglantiYap() baglanGösteri("SELECT * FROM probab3dennesi_kullanicilar probab3dennesi_calisanlar cal
    while (myRs.next()){// kullanicilara baglar
        kullanicilar.baglantiYap().addNew Kullanicisi
        myRs.getString(columnLabel "kullanicilar"), myRs.getString(columnLabel "kullaniciladi"),
        myRs.getString(columnLabel "kullanicisifre"), myRs.getString(columnLabel "yircerebiklari"),
        myRs.getString(columnLabel "para"), myRs.getString(columnLabel "esya"),
        myRs.getString(columnLabel "ponukabaram"), myRs.getString(columnLabel "gunayisi"),
        myRs.getString(columnLabel "calistirmaisaid"), myRs.getString(columnLabel "masa"));
    }

    // ARSILAR ALANI
    myRs = baglanti.baglantiYap() baglanGösteri("SELECT * FROM probab3dennesi_arsalar probab3dennesi_isletme probab3dennesi_isletmeserviste
    arsaArayList = new ArrayList<>();
    while (myRs.next()){// arsalari olusturur
        arsaArayList.add(new Arsa(
            myRs.getString(columnLabel "isance"), myRs.getString(columnLabel "isimkullanicil") myRs.getString(columnLabel "arsaistat"),
            myRs.getString(columnLabel "isletmeisale"), myRs.getString(columnLabel "isletmeisalestatu"),
            myRs.getString(columnLabel "gunayisat"), myRs.getString(columnLabel "isletmeisalestatu"),
            myRs.getString(columnLabel "isletmeisalestatu"), myRs.getString(columnLabel "kapsite"), myRs.getString(columnLabel "isletmeisalestatu");
        }
    }
}

```

```

3 import
public void setEsyamiktari(int esyamiktari) {
    baglanti.guncelle( newNode("UPDATE kullanicilar SET esyar=" +esyamiktari+" WHERE idkullanicilar="+IDkullanicli+""));
    baglanti.guncelle( newText("Eyar esyamiktari :="+esyamiktari );
        this.esyamiktari = esyamiktari;
    }
}

5 import
public void setGunlukkezanc(int gunlukkezanc) {
    baglanti.guncelle( newNode("UPDATE kullanicilar SET gunlukkezanc="+gunlukkezanc+" WHERE idkullanicilar="+IDkullanicli+""));
    gunlukkezancLabel.setText("Gonluk kezanc : "+ gunlukkezanc);
        this.gunlukkezanc = gunlukkezanc;
    }
}

1 import
public void setGunsayisi(int gunsayisi) {
    baglanti.guncelle( newNode("UPDATE kullanicilar SET gunsayisi="+gunsayisi+" WHERE idkullanicilar="+IDkullanicli+""));
    gunsayisiLabel.setText("Gun Sayisi : "+ gunsayisi);
        this.gunsayisi = gunsayisi;
    }
}

2 import
public void setIDcalisilanarsa(int IDcalisilanarsa) {
    baglanti.guncelle( newNode("UPDATE calisanlar SET calisilanarsa="+IDcalisilanarsa+" WHERE idkullanicli="+IDkullanicli+""));
    if(IDcalisilanarsa==1){
        IDcalisilanarsaLabel.setText("Calisilan arsa yok");
    }else {
        IDcalisilanarsaLabel.setText("Calisilan Arsa ID : "+ IDcalisilanarsa);
    }
        this.IDcalisilanarsa = IDcalisilanarsa;
    }
}

6 import
public void setMaas(int maas) {
    baglanti.guncelle( newNode("UPDATE calisanlar SET maas="+maas+" WHERE idkullanicli="+IDkullanicli+""));
    maasLabel.setText("Maas :"+maas);
        this.maas = maas;
    }
}

```

D. Yönetici/Kullanıcı Girişi ve Kaydedilişi

değerler ise daha öncesinde belirlenmiş sabit değerler verilerek oluşturulur.

Arsaları birleştirdiğimiz bir SQL kodunu Bağlantı sınıfımızdaki bağlantı metoduna veriyoruz. Bu metod bize Arşalarımızı tutan ResultSet'i verir. Bu ResultSeti bir while döngüsü ile okur ve daha öncesinde belirlemiş olduğumuz Arsa sınıfının yapıcı metodunun parametrelerine bu değerleri göndererek yeni nesneler oluştururuz ve bu nesneler bizim arşalarımız olur. (Fig.2)

Arsa al tuşuna tıklanıldığı zaman tıklayan kullanıcının almak istediği arsanın numarasını girmesi gerekiyor. Eğer kullanıcı 2 arsaya sahip ise işlem başlatılmaz. Girilen ID numarasına sahip olan arsanın sahibi sonrasında çıkan OptionPane paneline 'onay' yazarsa satın alım gerçekleşir. Bu işlemlerin yalancı kod açıklaması aşağıdaki gibidir: sahip olunan arsa sayısının 2 veya 2'den fazla olup olmadığına bakılır;

Arsaları tutan ArrayList , bir for döngüsü ile dönülür;
Eğer arsa numarası bir indexteki arsa nesnesi ile birlesirse o
index tutulur:

İşletme kur tuşunun yaptıkları şöyledir : Kullanıcı öncelikle açılan OptionPane içinde işletme kurmak istediği arsayı seçer.

Arsaları tutan ArrayList içinde Linear Search yaparak bahsi geçen arsayı buluruz ve arsaya sahip olup olmadığımızı kontrol ederiz. Eğer arsaya sahip değilsek bu işlemi yapamayız. Eğer arsaya sahipsek o arsada bir işletme olup olmadığını kontrol ederiz. Eğer bu arsada bir işletme varsa herhangi bir işletme kuramayız. Eğer bu alanda işletme yoksa yeni açılan OptionPane içinde açmak istediğimiz işletmenin türünü gireriz. Eğer bedeli karşılayabiliyorsak işletme arsaya kurulur. İşletme arsaya kurulduktan sonra arsanın bedeli işletmenin kurulum maliyetinin yarısı kadar artar.

H. İşe Girme

İşe Gir tuşu aslında hem işe girme hem de işten çıkma görevi görür. Eğer ilk başta açılan JOption Pane' ye girilen integer değer, oyuna kayıtlı olan bir arsayı temsil edebilir ihtimali var ise bu arsa aranır ve indexi , bulunduğu taktirde , tutulur. Başlıca aradığımız ihtimaller şu şekildedir : yöneticide çalışıp boşa geçmek isteyebilir yöneticide çalışıp arsaya geçmek isteyebilir boşta olup boşta kalmak isteyebilir boşta olup yöneticiye geçmek isteyebilir boşta olup arsaya geçmek isteyebilir arsada olup boşa geçmek isteyen olabilir arsada olup yöneticiye geçmek isteyen olabilir arsada olup başka arsaya geçmek isteyen olabilir yöneticide çalışıp yöneticiye geçmek isteyen olabilir Eğer bir arsada çalışmıyorsak ve işe girmek istediğimiz arsada yer varsa bu arsada işe başlarız. Eğer bu arsadaki işletme bir mağaza ise , çalışan kişinin eşyaları gün içinde düşmez. Eğer bu arsadaki işletme bir market ise , çalışan kişinin yiyecekleri gün içinde düşmez. Eğer bu arsadaki işletme bir emlak ise , çalışan kişinin parası gün içinde düşmez.

I. Ürün Satın Alma

Ürün satın alma tuşuna öncelikle ürün satın almak istediğimiz arsanın ID numarası girilir. Eğer bu arsanın sahibi kullanıcının kendisi değilse ve bu arsaya sahip bir kullanıcı varsa yeni açılan OptionPane içine kaç adet üründen almak istediğimizi gireriz. Eğer bu sayıdaki ürünü satın almak için yeterli miktarda paramız yoksa satın alım gerçekleştirilemez. Eğer bu sayıdaki ürünü satın almak için yeterli miktarda paramız varsa kullanıcının ürün satın aldığı arsaya bakılır. Eğer arsadaki işletme bir mağaza ise girilen sayı kadar eşya, kullanıcının hanesine eklenir ve bu alışverişin bedeli olan para kullanıcın bakiyesinden eksiltir. İşletmeye sahip olan kullanıcının bakiyesine aynı miktarda para eklenir. Eğer arsadaki işletme bir market ise girilen sayı kadar yiyecek, kullanıcının hanesine eklenir ve bu alışverişin bedeli olan para kullanıcın bakiyesinden eksiltir. İşletmeye sahip olan kullanıcının bakiyesine aynı miktarda para eklenir. Eğer Arsada bir işletme yoksa veya arsadaki işletme türü emlak ise alım yapılmaz.

J. Gün Atlama

Yöneticide yer alan butona tıklandığı zaman çıkan OptionPane içine yazılan pozitif değer kadar gün atlanır. Her

gün atlama fonksiyonu çağırıldığında kontrol edilmesi gereken bir takım unsurlar vardır. Eğer kullanıcının yiyeceği , eşyası veya parası bitmişse bu kullanıcı oyunu kaybetmiş olur ve arayüzden atılır. Kullanıcıların çalıştığı arsalarla bakılır. Eğer bu arsadaki işletme bir mağaza ise , çalışan kişinin eşyaları gün içinde düşmez. Eğer bu arsadaki işletme bir market ise , çalışan kişinin yiyecekleri gün içinde düşmez. Eğer bu arsadaki işletme bir emlak ise , çalışan kişinin parası gün içinde düşmez. Aynı şekilde eğer bakılan kullanıcı bir arsada değilse yöneticide çalışıyorsa aynı işlem yönetici için de uygulanır.Sırası ile yöneticide çalışılan işletmeye bakılır. Eğer bu işletme bir mağaza ise , çalışan kişinin eşyaları gün içinde düşmez. Eğer bu işletme bir market ise , çalışan kişinin yiyecekleri gün içinde düşmez. Eğer bu işletme bir emlak ise , çalışan kişinin parası gün içinde düşmez. Eğer bir arsada tam kapasite çalışılan bir gün geçerse işletmenin xp seviyesi 1 artar. Eğer bu seviye bir haftalık seviyeye gelirse işletme seviye atlar ve kapasitesi güncellenir. Xp seviyesi ise yenilenir.

III. SONUÇ

A. Programın Yapabildikleri ve İşlevsellikler Hakkında

Programın yöntemler ve kod ifadeleriyle de anlattığı üzere yerine getirilmesi istenen işlevlerin bir kısmını yapabilir olup bunlar : Veri tabanı Kurmak , Veri tabanı bağlantıları yapmak, Oyun içerisindeki karakterlerin oluşturulmak ve oyuna kaydetmek , Kullanıcı girişi ve kaydedilişini sağlamak , Oyun içerisindeki arsaların oluşturulmasını sağlamak , Oyun içerisindeki yöneticinin girişini sağlamak, Yöneticinin sahip olması gereken özellikleri atamak , Arsa satın almak , Arsalarla işletme kurmak , Bir arsada veya yöneticide işe girmek veya işten çıkmak , Ürün satın alma ticaretini sağlamak ve gün atlama fonksiyonlarını sağlamak.

IV. DENEYSEL ÇIKTILAR



Fig. 4. 0.arsanın bilgileri

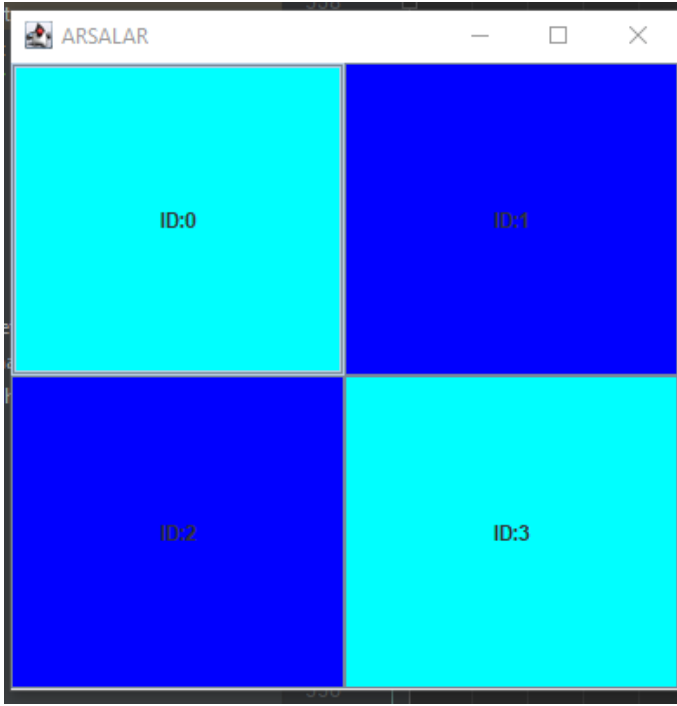


Fig. 5. arsalar

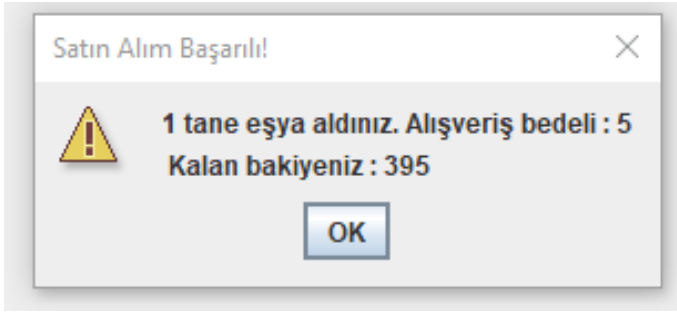


Fig. 6. örnek warning mesajı

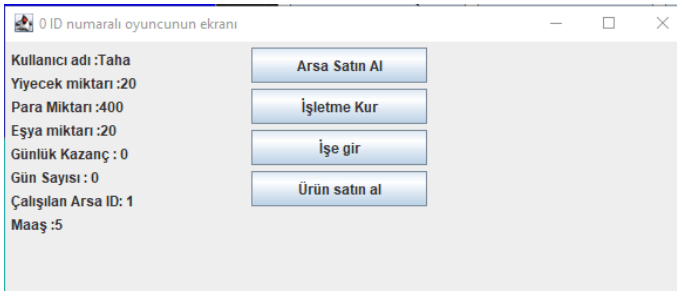


Fig. 7. oyuncu arayüzü

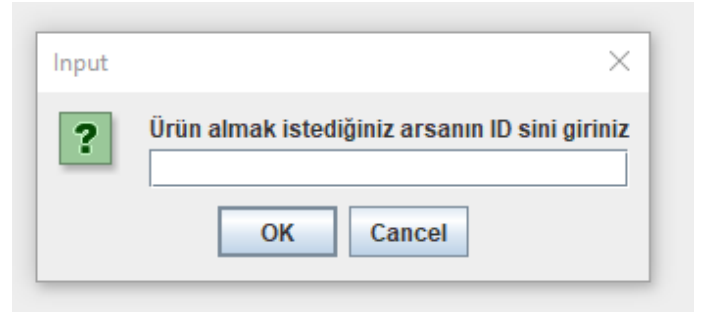


Fig. 8. ürün alma optionpane

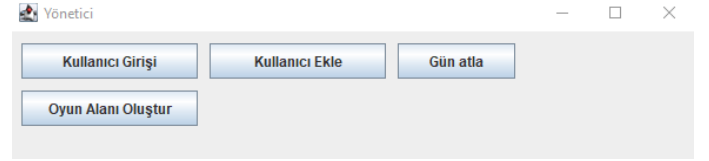


Fig. 9. yönetici ekranı

<https://www.youtube.com/watch?v=IFJKnU6wjJo>

<https://www.youtube.com/watch?v=aNpw7rGmUTA>

<https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>

<https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>

<https://bilgisayarkavramlari.com/2009/03/05/ucuncu-normal-sekil-third-normal-form-3nf/>

<https://bilgisayarkavramlari.com/2009/01/16/ikinci-normal-sekil-second-normal-form-2nf/>

<http://www.javaturk.org/global-degisenler-global-variables-ve-java/>

(Bazı kaynakların URL'sinde LaTeX komutlarını çalıştıran karakterler olduğundan dolayı rapora eklenememiştir.)

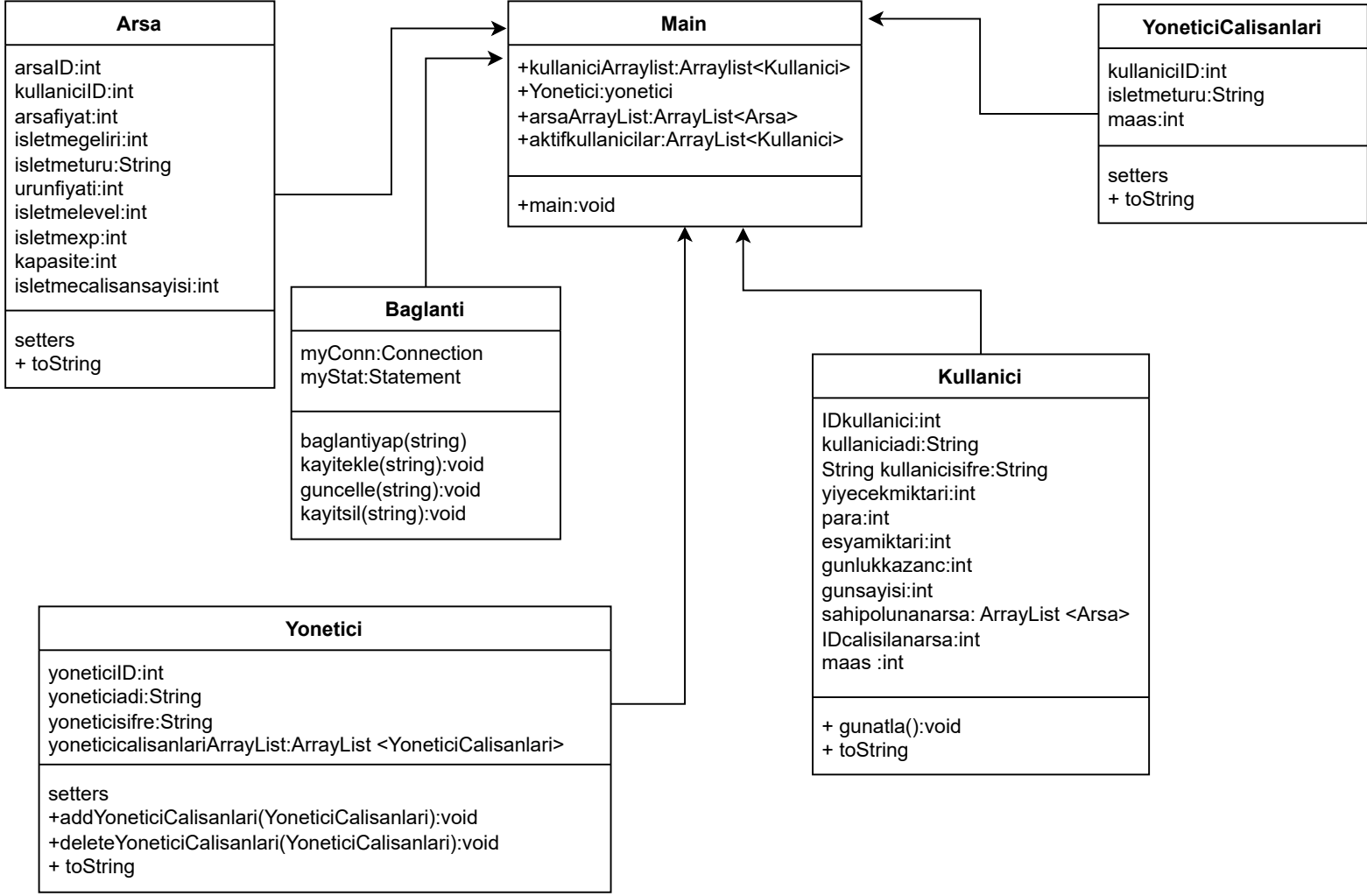
A. Deneyisel Çıktıların Açıklamalarıyla Verilmiş Halleri

V. KAPANIŞ VE KAYNAKÇA

Çıktılar yüklenen dosya üzerindeki işlemlerden alınmıştır.

A. Kaynakça

<https://www.youtube.com/watch?v=kCsFkNYahP0>



kullaniciilar
idkullaniciilar
kullaniciadi VARCHAR(45)
kullanicisifre VARCHAR(45)
yiyecekmiiktari
para
esya
gunlukkazanc
gunsayisi
Indexes

yonetici
idyonetici
yoneticiadi VARCHAR(45)
yoneticisifresi VARCHAR(45)
Indexes

arsalar
idarsa
idkullanici
arsafiyat
isletmegelir
Indexes

isletmeseviyetablosu
seviye
kapasite
Indexes

calisanlar
idkullanici
calisilanarsaid
maas
Indexes

oyunalani
idarsa
xkoordinat
ykoordinat
Indexes

isletmeyonetici
idyonetici
idkullanici
isletmeturu VARCHAR(45)
maas
Indexes

isletme
idarsa
isletmeturu VARCHAR(45)
urunfiyati
isletmelevel
isletmexp
isletmecalisansayisi
Indexes

ARSALARIN NORMALLEŞTİRİLMESİ

	idarsa	idkullanici	arsafiyat	isletmegelir	idarsa	isletmeturu	urunfiyati	isletmelevel	isletmexp	isletmecalisansayisi	seviye	kapasite
▶	0	3	100	10	0	emlak	0	1	0	1	1	3
	1	1	75	5	1	market	5	1	0	1	1	3
	2	2	75	5	2	market	5	1	0	1	1	3
	3	2	75	5	3	magaza	5	1	0	1	1	3

Seviye ve kapasite kısmı iyi fonksiyonel bağımlılık içerir bu yüzden onları ayrı bir tabloya almamız gerekir

	idarsa	idkullanici	arsafiyat	isletmegelir	idarsa	isletmeturu	urunfiyati	isletmelevel	isletmexp	isletmecalisansayisi	seviye	kapasite
▶	0	3	100	10	0	emlak	0	1	0	1	1	3
	1	1	75	5	1	market	5	1	0	1	1	3
	2	2	75	5	2	market	5	1	0	1	1	3
	3	2	75	5	3	magaza	5	1	0	1	1	3

	seviye	kapasite
▶	0	0
	1	3
	2	6
	3	12
	4	24
	5	48
	6	96
*	NULL	NULL

Farklı seviyeler de gösteriliyor. Bu tablo ile birleştiriliyor.

	idarsa	idkullanici	arsafiyat	isletmegelir	idarsa	isletmeturu	urunfiyati	isletmelevel	isletmexp	isletmecalisansayisi	seviye	kapasite
▶	0	3	100	10	0	emlak	0	1	0	1	1	3
	1	1	75	5	1	market	5	1	0	1	1	3
	2	2	75	5	2	market	5	1	0	1	1	3
	3	2	75	5	3	magaza	5	1	0	1	1	3

Yukarıda da görüldüğü gibi bazı durumlarda bu kısımlar da tekrarlı olabiliyor. O yüzden seçili alanları da ayrı bir tabloya alıyoruz

	idarsa	idkullanici	arsafiyat	isletmegelir
▶	0	3	100	10
	1	1	75	5
	2	2	75	5
	3	2	75	5
*	NULL	NULL	NULL	NULL

Geriye kalan kısımlar aşağıdadır

	idarsa	idkullanici	arsafiyat	isletmegelir	idarsa	isletmeturu	urunfiyati	isletmelevel	isletmexp	isletmecalisansayisi	seviye	kapasite
▶	0	3	100	10	0	emlak	0	1	0	1	1	3
	1	1	75	5	1	market	5	1	0	1	1	3
	2	2	75	5	2	market	5	1	0	1	1	3
	3	2	75	5	3	magaza	5	1	0	1	1	3

Bu kısmı da işletme tablosunda ayırdık

	idarsa	isletmeturu	urunfiyati	isletmelevel	isletmexp	isletmecalisansayisi
▶	0	emlak	0	1	0	1
	1	market	5	1	0	1
	2	market	5	1	0	1
	3	magaza	5	1	0	1
*	NULL	NULL	NULL	NULL	NULL	NULL

KULLANICILARIN NORMALLEŞTİRİLMESİ

	idkullanicilar	kullaniciadi	kullanicisifre	yiyecekmiktarı	para	esya	gunlukkazanc	gunsayisi	idkullanici	calisilanarsaid	maas
	0	Taha	taha	25	365	22	0	0	0	1	5
	1	Emir	emir	20	460	23	5	0	1	2	10
	2	icardi	gol	23	425	20	10	0	2	3	10
	3	ferdi	kayseri	20	400	20	10	0	3	0	15

Yukarıda seçili alanı farklı bir tabloya aldık

	idkullanici	calisilanarsaid	maas
▶	0	1	5
	1	2	10
	2	3	10
	3	0	15
*	NULL	NULL	NULL

Geri kalan ise aşağıdaki gibidir:

	idkullanicilar	kullaniciadi	kullanicisifre	yiyecekmiktarı	para	esya	gunlukkazanc	gunsayisi
▶	0	Taha	taha	25	365	22	0	0
	1	Emir	emir	20	460	23	5	0
	2	icardi	gol	23	425	20	10	0
	3	ferdi	kayseri	20	400	20	10	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

YÖNETİCİ ÇALIŞANLARI

Result Grid			Filter Rows:
	idyonetici	yoneticiadi	yoneticisifresi
▶	420	admin	admin123
*	NULL	NULL	NULL

Yöneticinin çalışanları eğer aynı tabloda tutulursa yukarıdaki değerler tekrar edecekti. O yüzden aşağıda görüldüğü gibi bir ayırma yaptık

Result Grid				Filter Rows:
	idyonetici	idkullanici	isletmeturu	maas
*	NULL	NULL	NULL	NULL