

Meta-Land Ticaret Oyunu

Umut YİĞİT
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi 2023
200201041

Betül BODUR
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi 2023
210201069

I. ÖZET

Bu dokümanda Programlama Laboratuvarı II dersimizin 3. projesi olan Meta-Land Ticaret Oyunu Projesi için oluşturulan algoritmalar ve database açıklanmıştır. Dokümanda problemin tanımı verilip bu problemler hakkında araştırmalar ve yöntemler sunulmuştur. Araştırmalar ve yöntemlerde bahsedilen class ve metotların ayrıntılı açıklaması bulunmaktadır.

Index Terms—database, nesne, arraylist

II. PROBLEM TANIMI

Bu projede, sanal ortamda arsa ve market ile ilgili ticari aksiyonları yerine getirmesi için belli kurala göre veritabanı oluşturulması istenmiştir. Oluşturulan veritabanı en az 3NF'yi sağlayacak şekilde oluşturulmalıdır. Oyun oynanırken veritabanında oluşabilecek değişikliklerin kaydedilebilmesi için oluşturulan veritabanı ile oyunun programlandığı ide arasındaki iletişim sağlanması gerekmektedir.

Meta-Land sisteminde oyuncu ve yönetici olmak üzere 2 farklı rol olmalıdır. Adından da anlaşılacağı üzere yönetici rol sistem içerisindeki tüm oyuncuların verilerini görebilme, kontrol edebilme ve oyun kurallarının parametrelerini belirleyebilmektedir. Oyuncu rolü ise diğer oyuncuların kısıtlı bilgilerini harita üzerinde görebilmektedir. Oyunun kuralları şu şekildedir :

Oyun karasel alanlara bölünmüş şekilde oyun arayüzünde gösterilmesi gerekmektedir. Karasel alanlar, oyundaki arsa ve üzerine kurulacak olan işletme alanlarına karşılık gelmektedir. Oyunun kaç kaçlık karasel alandan oluşacağını yönetici belirlemektedir. Oyuna yeni dahil olan oyuncu için otomatik ve sabit miktarda yemek, para ve eşya verecektir. Bu 3 özellik oyuncu tarafından görülebilmelidir. Oyunda yemek, para, eşya miktarında günlük olarak sabit miktarda azalma gerçekleşmelidir. Yiyecek ve eşya'dn biri tükendiğinde oyuncu oyunu kaybetmiş olmaktadır. Her bir kullanıcı bir işletmede çalışabilme hakkına sahiptir. Çalışma saati, günlük kazanacağı ücret, çalışmak için anlaşma yapacağı işletme sahibi belirlemektedir. Oyuncu çalışma saatlerinde herhangi bir ticari işlem yapamamaktadır. Oyun başında tüm alanlar yöneticiye ait olup sınırsız oyuncu kapasitesine sahip market, mağaza ve emlak bulunmalıdır. Markette çalışan bir oyuncunun günlük yiyecek miktarı, mağazada

çalışan bir oyuncunun günlük eşya miktarı ve emlakçıda çalışan oyuncunun günlük para miktarında çalışma süresi boyunca sabit eksiltme yapılmayacaktır. Oyuncular sahip oldukları para miktarı ile işletme sahiplerinin verdiği fiyatlara göre marketlerden yiyecek, mağazalardan eşya veya emlakçıdan işletme satın alma, kiralama işlemlerini yapabilmektedir. Emlakçı kendi belirlediği oranda alıcı-kiralayıcı ve satıcı-kiraya verenden bir komisyon ücreti almaktadır. Oyuncu arsa sahibi olamadan işletme kuramamaktadır. Oyuncu arsa sahibi olduğunda işletme kurması için Yönetici Emlak noktasından işlem yaparak inşa edebilmektedir. Oyuncu üzerinde işletme olmayan en fazla 2 arsaya sahip olabilmektedir. Her işletme kurulduğunda 1. seviye olarak başlamaktadır. 1. seviye işletmelerde maksimum 3 kişi çalışabilmektedir. Her seviye arttığında bu sayı ikiye katlanmaktadır. 1. seviyeye sahip işletme müşteri oyuncularla yaptıkları ticari işlemlerden elde ettiği gelirler dışında yöneticinin belirlediği sabit bir gelire sahiptir. Bu sabit gelir ilerleyen seviyelerde de yöneticinin belirlediği oranda arttırılacaktır. Bir işletme 1 hafta boyunca tam kapasite çalıştığında bir üst seviyeye geçmektedir. Bu proje java programlama dili ile yazılmıştır

III. ARAŞTIRMALAR VE YÖNTEMLER

Bu projede öncelikle, proje dosyasında verilen bilgiler doğrultusunda veritabanı oluşturulmuştur. Oluşturulan bu veritabanı normalizasyon işleminden geçirilerek son haline gelmiştir.

Aynı fonksiyon içerisinde farklı tablolardan veri çekilmesi gerektiğinde açılmış bir resultSet kapanmadığı için yeni resultSet oluşturularak bu sorun çözülmüştür.

Yukarıdaki sorunu çözmek için oluşturulan result tarafından yapılan işlem eğer bitirilmemişse bağlantı kurmak için oluşturulan statement yeniden kullanılamıyor ve burada da yeni bir statement oluşturularak bu durum çözülmüştür.

IV. GELİŞTİRME ORTAMI

- Bu proje JAVA programlama dili ile yazılmıştır.

- JDK Sürümü: 19
- NetBeans 15 kullanıldı.
- Veritabanı yönetim sistemi için MySQL kullanıldı.

V. KOD BİLGİSİ

A. Class'lar

Programın sınıfları şu şekilde özetlenmiştir:

MetaLand Sınıfı:

Ana class'tır. Arayuz sınıfından **gui** nesnesini oluşturmaktadır. Arayuz sınıfındaki **components metodu**u çağırılmaktadır. **components** metoduyla oluşturulan arayüz bileşenleri Arayuz sınıfında bulunan **goster metodu** ile görünürlüğü aktifleştirilir.

Arayuz Sınıfı:

JTextField ,JMenuItem ,JButton ,Calendar ,JComboBox ,JPopupMenu ,JLabel ,JRadioButton ,ButtonGroup ,TimerTask ve Timer tipinde nesneler oluşturulmuştur. Arsa ve Kullanici tipinde arraylistler oluşturulmuştur. Ve uygulama için gerekli değişkenler oluşturulmuştur. TimerTask class'ındaki run() adlı abstract class'ı override edilmiştir. run metodu ile SQLException hatası yakalanmadığı sürece sırasıyla miktarGuncelle() metodu ve maasYatir() metodu çağrılmıştır.

- arsaEkranı metodu

Veritabanında executeQuery metoduyla alan tablosuna ulaşılmıştır. Oyun alanını oluşturmak için alan tablosundaki tüm veriler while döngüsü kullanılarak gezilmiştir. Her bir konum için Arsa tipinde nesneler oluşturulmuştur. Oluşturulan her nesne için x ve y koordinatları belirlenmiştir. Nesnelerin arsa mı, market mi, emlak mı ve mağaza mı olduklarını anlamak için alan tablosundaki alanTuru sütunundaki veriler karşılaştırılmıştır. Eğer nesne; market ise o nesnenin isletmeID'si 1 dir, emlak ise o nesnenin isletmeID'si 2 dir, mağaza ise o nesnenin isletmeID'si 3 dir. İşletme olan alanlar arayüzde sarı olarak gözükmektedir. Karasel bölgelere tıklanıldığında olması gerekenler ActionListener ile belirtilmiştir. Eğer arsaya tıklanırsa sırasıyla bunlar gerçekleşmektedir:

Arsaların işletme olup olmadığını ve arsanın sahip id'si ile giriş yapılan id'nin aynı olup olmadığı kontrol edilmektedir. Mouse'un x ve y eksenine göre lokasyon alıp hangi karesel alanda olduğu bulunmaktadır. X ve Y koordinatları alındıktan sonra $8*y+x$ ile hesaplama yapılarak tüm arsaların kayıt edildiği Arsa tipindeki areas arraylistinden arsaların id'lerine erişilmektedir. İlan için arsaya tıklanıldığında ilanKey true olmaktadır ve arsanın id'si alınıp ilanKey false olmaktadır. Ve ilan için gerekli olan isciSayisi, gunSayisi, maasBilgisi ve ilanOnay adlı JTextField ve JButton görünürlükleri aktifleştirilmektedir. Eğer

iskur true ise ilk false değerine çevrilir ve arsanın id'si alınmaktadır. İşletmelerin türüne göre PopUp menüde çıkanlar değişmektedir. İşletmenin id'si 1 ise markettir ve bulunluğu karasel alana tıklanıldığında PopUp menüsünde ilanlar menü itemi ve yemekAl menü itemi görünür olmaktadır. Emlak için ilanlar menü itemi ,arsaAl menü itemi ve isletmeKur menü itemi görünür olmaktadır. Mağaza için ise ilanlar menü itemi ve eşyaAl menü itemi görünür olmaktadır. Bu menü itemleri , JPopupMenu tipinde oluşturulan işlemler adlı nesneye eklenmektedir.

Kullanıcının yemek, esya, para, ad, soyad ve id'si arayüz üzerinde gözükmektedir. kullanıcıID, 0 değilse setMiktar() metodu çağrılmaktadır. kullanıcıID, 0 ise yöneticiİşlemler() metodu çağrılmaktadır.

- components metodu

Kullanıcıların oyuncu mu , yönetici mi olduğunu oyuncu ise hangi oyuncu olduğunu anlamak için kullanıcı adı ve şifrenin TextField'ları ve butonlarının görünürlüğü aktifleştirilmiştir. Ve oyun başladığında oyuncu ve yöneticinin görmesini istediğimiz JPopupMenu, JMenuItem, JTextField, JLabel ve butonların konumları belirlenmiştir. Arayüzde TextField'lara girilen bilgileri almak için buton bulunmaktadır ve her bir buton için basıldığında gerçekleşmesi istenilen olaylar ActionListener yapısıyla belirlenmiştir.

Kullanıcı adı girildiğinde onay butonuna basıldığında, girisOnay() metodu çağrılmaktadır.

- girisOnay metodu

Başlangıç ekranında kullanıcı adı ve parolası girildiğinde onay butonuna basıldığında bu metot çağrılmaktadır. Bu metotta kullanıcının girdiği kullanıcı adı veritabanında bulunup bulunmadığı kontrol edilmektedir. Kullanıcı girişte direkt ad soyad şeklinde girmektedir. Bu bilgi veritabanında 2 ayrı sütundan oluşmaktadır bu yüzden iki veriyi birbirinden ayırmamız gerekmektedir bunu trim metoduyla gerçekleştirilmiştir. Ad, soyad ve parola veritabanında bulunan veriler ile eşleştğinde başlangıç ekranında gözükmekte olan butonlar ve JTextField'ların görünürlüğü pasifleştirilmektedir. Kullanıcı adını ararken aynı zamanda Adminin oyuncular arasında geçiş yapabilmesi için JComboBox'u da doldurulmaktadır. Ve her bir kullanıcı için ; kullanıcıNo, kullanıcıAdı ve kullanıcıSoyadı parametre olarak Kullanıcı tipinde nesne oluşturulur. Oluşturulan bu nesneler Kullanıcı tipinde oluşturulan arraylist'e eklenmektedir. Kullanıcı bilgileriyle ilgili işlemler Adminin kullanıcılar arasında geçiş yaptığına verileri doğru bir şekilde çekebilmesini sağlamaktadır. Veritabanındaki kullanıcıların hepsi gezildikten sonra arsaEkranı() metodu çağrılmaktadır. arsaEkranı() metodundaki işlemler bittikten sonra oyunun zaman akışını başlatan zamanBaslat() metodu çağrılmıştır.

- goster metodu

Frame ekranının boyut, arkaplanı ve frame'in görünürlüğü aktifleştirilmiştir.

- maasYatir metodu

Veritabanında calisan tablosu sorgulanır.While döngüsü içerisinde tablodaki verilerin işletme ID'si alınmaktadır,her bir işletmenin başlangıç tarihi bilgisi alınmaktadır, her bir işletmenin iş bitiş tarihi bilgisi alınmaktadır,çalışılan gün sayısı bilgisini almaktadır.Eğer bitiş tarihi ile baaşlangıç tarihi arasında çalışılan gün sayısı kadar fark var ise isletme ve calisan tablosuna ulaşıp kullanıcının işi bittiği için gerekli güncellemeler veritabanında yapılmaktadır.Kullanıcı işte çalışıyorsa mevcut para miktarına erişmek için miktar tablosuna erişilmiştir ve maaşı eklenerek yeni para miktarı veritabanında güncellenmiştir.

- miktarAyarla metodu

SQL sorgusuyla miktar tablosuna erişilmektedir.miktar tablosunda kullanıcıNo sütunundaki veri ile kullanıcıID eşit ise kullanıcıya yemek,esya ve para miktarı güncellenmektedir.

- miktarGuncelle metodu

Veritabanındaki miktar tablosuna ulaşılmaktadır.miktar tablosundan kullanıcının kullanıcıNo'suna erişilmiştir ve kullanıcının yemek,eşya ve para miktarı giriş yapan kullanıcının miktarlarına eşitlenmiştir.Eğer calisan tablosundan erişilen işletme türü mağaza veya emlak ise oyuncunun yemek sayısı 25 azalmaktadır. Eğer ki işletme türü market veya emlak ise eşya miktarı 25 azalmaktadır.calisan tablosunda kullanıcı yoksa oyuncunun yemek ve eşya miktarı 25 azaltılmaktadır. Her zaman para miktarı 25 azalmaktadır.Arayüzdeki ve veritabanındaki yemek,eşya ve para miktarları güncellenmektedir.Yemek miktarı veya eşya miktarı veya para miktarı sıfır veya altına düşüyse ve oyuncu daha önce elenmediyse oyuncu elenmektedir.Kullanıcıya ait olan arsaların görünürlüğü pasifleştirilmiştir.

- setMiktar metodu

Veritabanından miktar tablosuna executeQuery() metodu kullanılarak ulaşılmıştır.Giriş yapan kişinin id'si ile miktardaki id karşılaştırılıp giriş yapan kişinin id'sindeki bilgilere ulaşılmalıdır.Bunun için miktar tablosundaki kullanıcıNo sütunu while döngüsü yardımıyla gezilmiştir.Kişinin id'si, giriş yapanın id'si ile eşlendiğinde integer tipindeki değişken bir artmaktadır.Eğer artmassa veritabanında giriş id'sine ait miktar tablosundaki veriler vasrsayılan şekilde 100,100,1000,0 şeklinde atanmaktadır. Kişinin eşya,yemek ve para miktarı veritabanından okunup JLabel'lara yazı olarak

girilmektedir.Eğer eritabanından okunan yemek miktarı veya eşya miktarı veya para miktarı 0'a eşit veya küçükse ve elendi boolean değeri 0'a eşit ve giriş id'si 0 değilse, elendi boolean değeri 1 olmaktadır ve arayüzde ad soyad bilgisinin yanına elendiğini belirten yazı yazılmaktadır.Kullanıcıya ait olan arsalar ve işletmeler'in id'si adminin id'sine eşitlenmektedir.miktarAyarla() metodu çağrılmaktadır.

- yöneticiIslemler metodu

Admin'in değerleri değiştirebileceği JButton, Oyuncuların adının ve soyadının bulunduğu JComboBox , Oyunuculara ait eşya, para ve yemek miktarını ayarlayabilmek için JTextField , oyuncuları verilerini görebilmek için onay butonu, Zamanın ne kadar ileri alınacağı belirtilen JTextField ve bu zamanı onaylamak için olan butonun görünürlüğü aktifleştirilmektedir.miktarAyarla() metodu çağrılmaktadır.

- zamanBaslat metodu

Timer class'ında bulunan schedule() metodu kullanılmıştır gunKontrol adlı görevi belirtilen sabit olan 30000ms'den sonra tekrarlanması için kullanılmıştır.

- zamanYolculugu metodu

Zamanın ne kadar ileri alınacağı girilen JTextField'dan gelen veriyi parametre olarak almaktadır.Girilen gün sayısı kadar sırasıyla miktarGuncelle() metodu ve maasYatir() metodu çağrılmaktadır.

Arsa Sınıfı:

- Arsa metodu

Parametre olarak mouse'tan alınan x ve y koordinatlarını,alan no'su ve alan sahibinin id'sini almaktadır.

- kiraIslemi metodu

Kullanici Sınıfı:

- Kullanici metodu

Kullanıcının adı,soyadı ve id'sini parametre olarak almaktadır.

B. UML DİYAGRAMI

Kısım ektedir.[fig.1]

VI. İSTATİSTİK

Kullanılan class'lar ve kullanım amaçları:
java.util

ArrayList oluşturmak,kullanıcıdan veri almak için kullanılmıştır.

java.awt

Yazı tipi, yazı rengi, actionlist gibi olayları düzenlemek için kullanılmıştır.

javax.swing

Buton, görüntü , çerçeve,.. gibi arayüzü oluşturmak için gerekli nesneleri oluşturmak için kullanılmıştır.

java.sql

Veritabanına erişmek için kullanılmıştır.

VII. SONUÇLAR

Sonuç olarak, veritabanı oluşturup normalizasyon işlemlerini gerçekleştirerek proje için uygun veritabanı tasarlanmıştır.Tasarlanan veritabanında güncelleme,ekleme,veri alma gibi işlemler yapılabilmektedir.Proje dosyasınad verilen isterlerin bir çoğu yerine getirilmiştir.

REFERENCES

- [1] <https://www.guru99.com/database-normalization.html>
- [2] <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>
- [3] <https://www.yusufsezer.com.tr/java-jdbc/>
- [4] <https://alvinalexander.com/java/java-mysql-update-query-example/>
- [5] <https://www.youtube.com/watch?v=XJUTERs-aFE>
- [6] <https://www.youtube.com/watch?v=lxaylm9CZRA>
- [7] <https://www.youtube.com/watch?v=GNYG3WZMXnY>
- [8] https://www.youtube.com/watch?v=vVLo_aCKFEclst=PL9OZDx-_1-s39SjOvT-fr-lABr7XfSjdN
- [9] <https://www.youtube.com/watch?v=6X28VC1bPLI>
- [10] <https://www.youtube.com/watch?v=vIGU31SBQzIlist=PLXyRAIMUJB4HufIPZX-3ouSA8AXGP2hqindex=2>
- [11] <https://www.youtube.com/watch?v=A4Wql24PcLk>
- [12] <https://stackoverflow.com/questions/44387119/error-1046no-database-selected>
- [13] <https://stackoverflow.com/questions/5260462/can-i-add-an-action-listener-to-a-jlabel>
- [14] <https://www.javatpoint.com/java-actionlistener>
- [15] <https://medium.com/tableplus/how-to-import-export-mysql-databases-a1eb0c8aea2d>
- [16] <https://stackoverflow.com/questions/68176631/how-to-solve-error-java-sql-sqlexception-before-start-of-result-set>
- [17] <https://stackoverflow.com/questions/75263127/i-have-a-sql-error-java-sql-sqlexception-operation-not-allowed-for-a-result-set>
- [18] <https://stackoverflow.com/questions/32671535/operation-not-allowed-after-resultset-closed-mysql-java>
- [19] <https://stackoverflow.com/questions/6260855/how-to-clear-reset-a-jframe>
- [20] <https://stackoverflow.com/questions/23316517/jdbc-how-to-clear-the-contents-of-a-result-set-before-using-it-again>
- [21] <https://stackoverflow.com/questions/8826247/java-sql-sqlexception-before-start-of-result-set>
- [22] <https://www.javatpoint.com/java-jcombobox>
- [23] https://www.youtube.com/watch?v=5-RGCRc4_ts
- [24] <https://stackoverflow.com/questions/41748009/how-can-i-display-time-to-my-game>
- [25] <https://www.kodlamamerkezi.com/java/java-timer-zamanlayici-kullanimi/>
- [26] <https://www.tutorialspoint.com/jdbc/jdbc-update-records.htm>
- [27] <https://blog.burakkutbay.com/jdbc-preparedstatement-nesneleri.html/>
- [28] <https://stackoverflow.com/questions/42950607/arraylistjbutton-adding-action-listeners-with-arraylist>
- [29] <https://docs.oracle.com/javase/8/docs/api/java/awt/MouseInfo.html>
- [30] <https://www.youtube.com/watch?v=hi1C-QtOeDo>
- [31] <https://www.javatpoint.com/java-jpopupmenu>
- [32] <https://www.geeksforgeeks.org/calendar-class-in-java-with-examples/>
- [33] https://www.w3schools.com/sql/sql_insert.asp
- [34] <https://www.geeksforgeeks.org/jradiobutton-java-swing/>
- [35] <https://digitallibrary.sdsu.edu/islandora/object/sdsu>
- [36] <https://www.youtube.com/watch?v=mUtAPbb1ECMlist=PLLGlmW7jT-nTr1ory9o2MgsOmmx2w8FB3index=2>

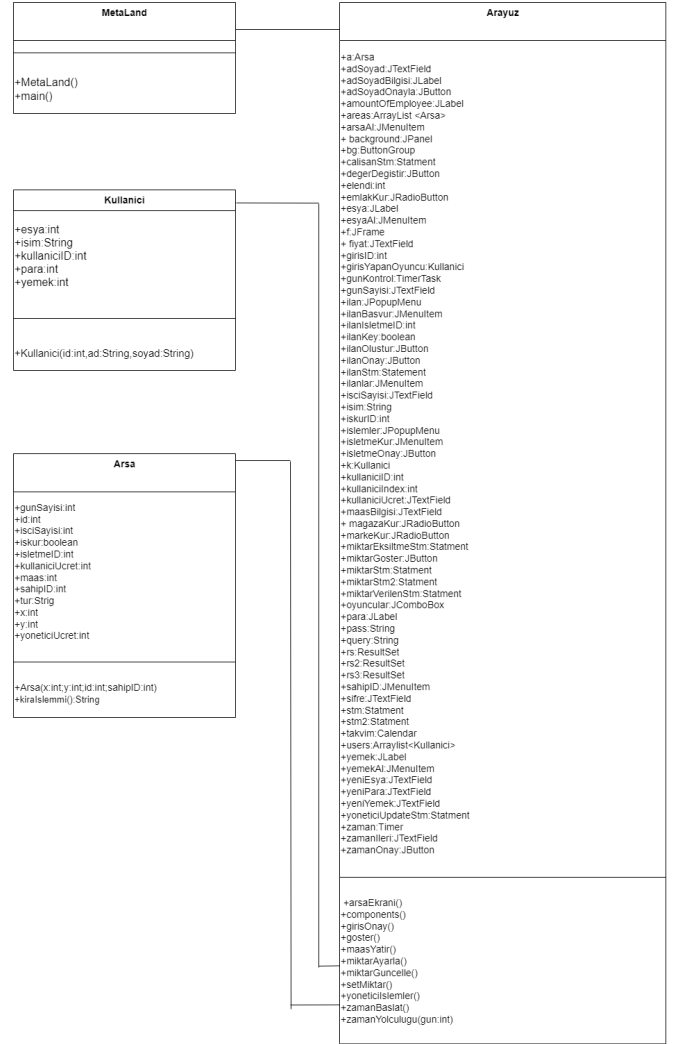


Fig. 1. UML DİYAGRAMI

A. Normalizasyon Adımları

kullanici																							
kullaniciNo	kullaniciAdi	kullaniciSoyad	kullaniciFirma	kullaniciCemex	kullaniciEya	kullaniciPara	baslangicEya	baslangicCemex	baslangicPara	oyunBaslangicTarihi	alanBoyutu	alanNo	alanTuru	alanSahip	gunlukYiyecekGideri	gunlukEyaGideri	gunlukParaGideri						
1	Umut	Yigit	abc123	85	126	5700	100	100	1000	20.05.2023	64	4	Arsa	1	25	25	25						
2	Betül	Bodur	kou2023	120	30	3400	100	100	1000	18.05.2023	64	5	Market	2	25	25	25						
3	Furkan	Göz	ceng123	400	210	8000	100	100	1000	17.05.2023	64	6	Arsa	3	25	25	25						
4	Kubra	Erat	proLab*3	346	234	200	100	100	1000	2.05.2023	64	7	Magaza	4	25	25	25						
isletme																							
isletmeTur	yoneticUcret	kullaniciUcret	calismaBaslang	calismaBitis	calisanSayisi	seviye	kapasite	calisanSayisi	yiyecekUcret	esyaUcret	isletmeFiyat	kiralikFiyat	komisyon	emlakislemi	kirasuresi	satistarihi	kiralamaTarihi	kiraBitisTarihi	emlakID	sabitGelir	gelirOran	mevcutSeviye	baslangicTarihi
Market	0	800	24.05.2023	18.05.2023	3	72	1	1	30	0	3000	200	80			10.03.2020						1	10.03.2020
Magaza	0	800	28.05.2023	6.06.2023	1	24	2	6	1	50		800	80			4.04.2022						2	4.04.2022

Fig. 2. Normalizasyon 1

kullanici													
kullaniciNo	kullaniciAdi	kullaniciSoyad	kullaniciSifresi	kullaniciYemek	kullaniciEya	kullaniciPara	baslangicEya	baslangicCemex	baslangicPara	oyunBaslangicTarihi	gunlukYiyecekGide	gunlukEyaGideri	gunlukParaGideri
1	Umut	Yigit	abc123	85	126	5700	100	100	1000	20.05.2023	25	25	25
2	Betül	Bodur	kou2023	120	30	3400	100	100	1000	18.05.2023	25	25	25
3	Furkan	Göz	ceng123	400	210	8000	100	100	1000	17.05.2023	25	25	25
4	Kubra	Erat	proLab*3	346	234	200	100	100	1000	2.05.2023	25	25	25

alan

alanBoyutu	alanNo	alanTuru	alanSahip
64	4	Arsa	1
64	5	Market	2
64	6	Arsa	3
64	7	Magaza	4

isletme

alanNo	isletmeTur	yoneticUcret	kullaniciUcret	calismaBaslangic	calismaBitis	calismaGunSayisi	calisanSaat	seviye	kapasite	calisanSayisi	yiyecekUcret	esyaUcret
5	Market	0	500	14.05.2023	18.05.2023	3	72	1	3	1	30	0
7	Magaza	0	800	28.05.2023	6.06.2023	1	24	2	6	1	0	50

emlak

alanNo	isletmeFiyat	kiralikFiyat	komisyon	emlakislemi	kiraSuresi	satisTarihi	kiralamaTarihi	kiraBitisTarihi	emlakID	sabitGelir	gelirOran	mevcutSeviye	baslangicTarihi
5	5000	300	80			10.03.2020						1	10.03.2020
7	3000	800	90			4.04.2022						2	4.04.2022

Fig. 3. Normalizasyon 2

alan			kullanici				calisan									
alanNo	alanTuru	alanSahip	kullaniciNo	kullaniciAdi	ullaniciSoyad	kullaniciSifresi	yunBaslangicTarihi	kullaniciNo	calismaBaslangic	calismaBitis	calismaGunSayisi	calisanSaat	isletmeTur	isletmeID		
4	Arsa	1	1	Umut	Yigit	abc123	20.05.2023	1	14.05.2023	18.05.2023	3	72	Market	5		
5	Market	2	2	Betül	Bodur	kou2023	18.05.2023	3	28.05.2023	6.06.2023	1	24	Magaza	7		
6	Arsa	3	3	Furkan	Goz	ceng123	17.05.2023									
7	Magaza	4	4	Kubra	Erat	proLab*3	2.05.2023									
miktari			isletme													
kullaniciNo	kullaniciYemek	kullaniciEya	kullaniciPara	arsaSayisi	alanNo	isletmeTur	yoneticUcret	kullaniciUcret	seviye	kapasite	calisanSayisi	yiyecekUcret	esyaUcret	ilanVerildi	ilanGun	
1	85	126	5700	1	5	Market	0	500	1	3	1	30	0	1	10	
2	120	30	3400	1	7	Magaza	0	800	2	6	1	0	50	0	0	
3	400	210	8000	1												
4	346	234	200	1												
emlak			gider													
alanNo	isletmeFiyat	kiralikFiyat	komisyon	emlakislemi	sabitGelir	gelirOran	mevcutSeviye	baslangicTarihi	kullaniciNo	gunlukYiyecekGide	gunlukEyaGideri	gunlukParaGideri	alanBoyutu	baslangicEya	baslangicCemex	baslangicPara
5	5000	300	80			1		10.03.2020	1	25	25	25	64	100	100	1000
7	3000	800	90			2		4.04.2022	2	25	25	25				
									3	25	25	25				
									4	25	25	25				
islem																
alanNo	kiraSuresi	satısTarihi	kiralamaTarihi	kiraBitisTarihi	emlakID											
6						5		10.03.2020								
7								4.04.2022								

Fig. 4. Normalizasyon 3