MetaLand Simülasyon Oyunu

1st Fatih Serhat TURAN
Kocaeli Üniversitesi
Bilgisayar Mühendisliği
210202100

2nd Mustafa SÜRMELİ
Kocaeli Üniversitesi
Bilgisayar Mühendisliği
210202042

I. Açıklama

Kocaeli Üniversitesi Bilgisayar Mühendisliği Programlama Labaratuvarı 2 3.proje hakkında rapor yazısı.

II. Özet

A. Proje Açıklaması

Kullanıcıların emlak alabildiği iş kurabildiği çalışabildiği, kendi içinde ekonomik bir sistemi olan bazı kurallara dayalı metaland oyunu.Oyuncular belirtilen bölge içindeki boş arazileri satın alabilir,buradaki menkullerin tipini değişebilir buradan ilan açabilir,bu ilan üzerinden kullanıcıları işe alabilir belirli bir gelir karşılığında belirli bir süre çalıştırabilir.Ayrıca menkul satabilir veya satın alabilir.Haritanın bir de yöneticisi vardır.Yönetici istediği boyutta harita oluşturabilir, bu harita üzerinde bir çok yetkiye sahiptir kullanıcıların yemek eşya ve paralarını manipüle edebilir.Ayrıca kullanıcıların yemek, eşya ve paraları biterse oyunu kaybederler ve bir daha erişim sağlayamazlar.

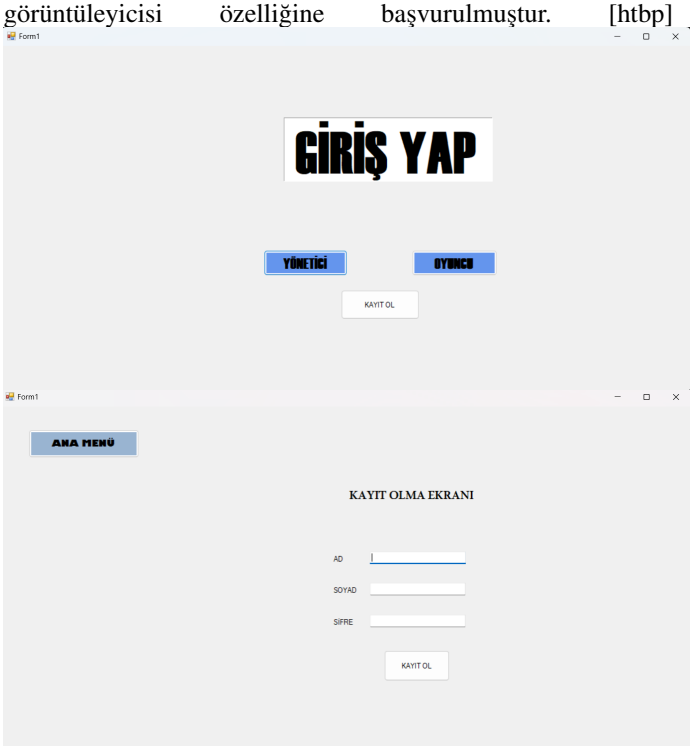
B. Geliştirme Süreci

Öncellikle bizden istenen metaland simülasyonu oyunu için c kullandık.C kullanmamızdaki temel etken bizden istenenleri rahatlıkla karşılaması, windows form uygulaması kullanımının visual studio üzerinden kullanılmasının kolay olması ve daha öncesinde c diline aşina olmamız oldu.Ayrıca bizden istenen veri tabanı işlemleri için mssql server kullandık, mssql kullanımızdaki temel etkense hem daha öncesinde kullanımına aşina olmamız hem de mssqlin arayüz ve kullanımının bize sunulan diğer seçenekler olan mysql ve postgresql'den daha kolay olması oldu.

Tasarladığımız Metaland platformunda bulunan kullanıcılar için tasarlanacak olan veritabanı, varlıklar ve işlemlerle ilgili bilgilerin düzenli bir şekilde yönetilmesine yardımcı olacaktır. Bu sayede kullanıcılar, ihtiyaç duydukları bilgilere daha kolay bir şekilde ulaşabileceklerdir.Veritabanı tabloları oluşturulurken bizden istendiği şekilde 3nf normalizasyonuna uygun olmasına dikkat ettik.

Oluşturduğumuz windows form uygulamasında tek bir form üzerinden tüm işlemleri kontrol etmek için Usercontrol sınıflarıyla çalıştık . Usercontrol sınıfı form classı üzerine tetklenen ve üzerinde yeniden kullanılmasına olanak sağlayacak şekilde form1 classı üzerine gerekli durumlar oluştuğunda çağırılmaktadır.Bu controller hem kod yazımını daha düzenli ve verimli hale getirmektedir hem de kullanıcı için daha düzenli bir arayüz sunmaktadır.

Projede oluşturduğumuz ER diyagramı mssql'in bize sunduğu veritabanı diagramı görüntüleme özelliği kullanarak elde edilmiştir. Aynı zamanda bizden istenen UML diyagramı içinse c geliştirmek için kullandığımız ortam olan visual studio'nun sınıf görüntüleyicisi özelliğine başvurulmuştur. [htbp]



III. Yöntem

A. Giriş

Giriş ekranında giriş yapan kişiden yönetici girişi mi kullanıcı girişi mi yapmak istedğine göre seçtirilir.Veya kayıt olma ekranına gidilir.Yaptığı seçeneğe göre ekranlara yönlendirilir.yönlendirilen ekranda isim ve şifre istenir.Ardından bu değerler kontrol edilir eğer oyuncu veya yönetici tabloları kontrol edilir eğer kullanıcı varsa giriş yapılır yoksa işlem başarısız olur.Ayrıca burada oyuncunun kaynakları kontrol edilir eğer tükenmişse giriş yapmasına izin verilmez.

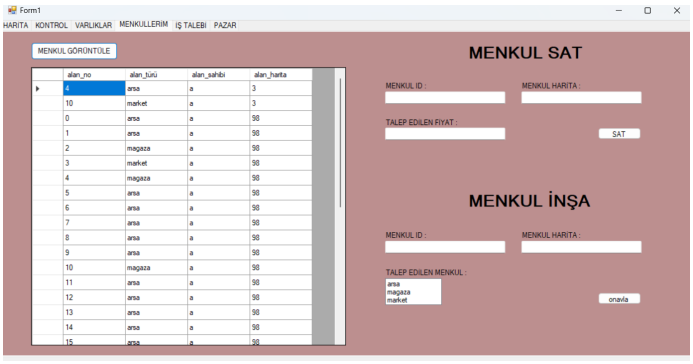
B. Kayıt Olma

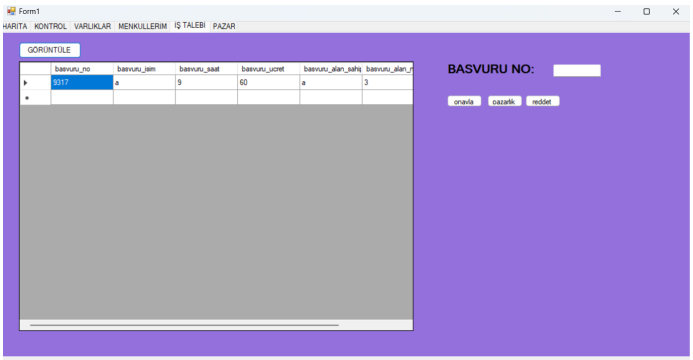
Kayıt olma ekranında kullanıcıdan bilgiler alınır.Ardından metaLand adlı Db de kullanıcı tablosuna kaydedilir.Ve kul-

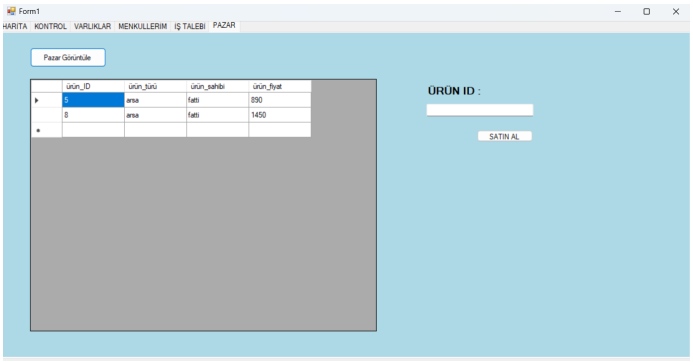
lanıcıya kayıt oluşturuldu bilgisi verilir.

C. kullanıcı panel

Burada yöneticiye harita id sorulur.Ve Seçilen harita ekrana verilir.Kontrol ekranında yönetici harita numarasını ve boyutunu girerek harita oluşturabilir.Haritaya kayıtlı olanları sorgular ve gün sonunu verir.Oyuncu panelinden farkı kontrol kısmıdır.Menkullerim kısmında kullanıcıya ait menkuller bilgileriyle birlikte verilir.Ve bu ekranda kişi menkul seçip satabilir.Ayrıca burada menkul inşa edilebilir.Varlıklar kısmında kişinin yemek para ve eşya miktarı gösterilir.İş talebi kısmında iş ilanları görüntülenebilir uygunsa kabul edilebilir değilse pazarlık yapılabilir.Pazar kısmında satılıktaki menkuller satın alınabilir.







D. harita-oyun

Harita üzerinde gezinme yapılabili, beğenilen ya da seçilen yerlerin satın alımı yapılabilir is başvurusunda bulunulabilir.

'türVer' adlı yöntem, 'alanNo' adlı bir tamsayı parametresi alır. Sağlanan bağlantı dizesini kullanarak bir SQL veritabanına bağlanır, bir bağlantı açar ve 'alaninfo' adlı bir tablodan veri seçmek için bir SQL komutu çalıştırır. Komut, 'alanharita' alanının değeri 'singleton.haritaID' değişkeninin değerine eşit ve 'alanno' alanının değeri sağlanan 'alanNo' parametresine eşit olan tüm verileri seçer.

Tabloda eşleşen bir kayıt bulunursa, yöntem kayıttan 'alantürü' alanının değerini alır, bir dizeye dönüştürür ve 'returnValue' adlı bir değişkene atar. Son olarak, veritabanı bağlantısını kapatır ve 'returnValue' değerini döndürür.

'fiyatver' methodu bir tamsayı ve bir dize parametresi alır. Bağlantı dizesini kullanarak bir SQL veritabanına bağlanır, bir bağlantı açar ve 'alanTür' parametresinin değerine göre 'magaza' veya 'market' tablosundan veri seçmek için bir SQL komutu çalıştırır. Komut, 'magazaharita' veya 'marketharita'

alanının değeri 'singleton.haritaID' değişkeninin değerine eşit ve 'magazano' veya 'marketno' alanının değeri sağlanan 'alanNo' parametresine eşit olan tüm verileri seçer.

Tabloda eşleşen bir kayıt bulunursa, yöntem kayıttan 'magazaücret' veya 'marketücret' alanının değerini alır, bir dizeye dönüştürür ve 'returnValue' adlı bir değişkene atar. Son olarak, veritabanı bağlantısını kapatır ve 'returnValue' değerini döndürür.

Yani bu method, sağlanan 'alanNo' parametresine ve 'alanTür' parametresine bağlı olarak bir SQL veritabanı tablosundan fiyat değerini alır. 'yöneticibilgiver' methodu 'talepbilgi' adlı bir dize parametresi alır ve kullanıcının 'esyamiktarı' veya 'yemekmiktarı' değerlerini döndürür. Bağlantı dizesini kullanarak bir SQL veritabanına bağlanır ve 'kullanıciinfo' tablosundan 'kullanıciadı' alanı 'singleton.oyuncuAd' değişkeninin değerine eşit olan verileri seçmek için bir SQL komutu çalıştırır.

Eğer 'talepbilgi' değeri 'esya' ise, SQL komutu 'esyamiktarı' alanının değerini seçer; eğer 'talepbilgi' değeri 'yemek' ise, SQL komutu 'yemekmiktarı' alanının değerini seçer.

Daha sonra, SQLDataReader nesnesi ile kayıtlar okunur ve bulunan kayıtlarda 'esyamiktarı' veya 'yemekmiktarı' değeri 'returnValue' değişkenine atanır. Bu değer son olarak bir tamsayıya dönüştürülür ve 'returnValue' olarak döndürülür.

Son olarak, veritabanı bağlantısı kapatılır.

Bu method, kullanıcının eşya veya yemek miktarını döndürür ve bu bilgilerin diğer işlemlerde kullanılması için kullanılabilir.

"yöneticiParaAzalt()" methodu, öncelikle bir SqlConnection objesi oluşturur ve bağlantı dizesini kullanarak bir SQL veritabanına bağlanır. Daha sonra, "yöneticiParaBul()" adlı başka bir method kullanarak yöneticinin parasını bulur ve bir değişkene atar. Ardından, kullanıcının seçtiği ürün adetini ve fiyatını kullanarak hesaplayarak, kullanıcının parasını azaltır ve veritabanına kaydeder.

"yöneticiParaBul()" methodu ise, bir tamsayı değeri döndürür ve bir SqlConnection objesi oluşturur. Daha sonra, bağlantı dizesini kullanarak bir SQL veritabanına bağlanır ve bir SQL komutu çalıştırarak yöneticinin parasını bulur. Bu değeri bir değişkene atar, veritabanı bağlantısını kapatır ve değişkeni döndürür.

Bu iki method birlikte çalışarak, kullanıcının ürün satın alma işlemleri sırasında, yöneticinin parasını azaltmaktadır. Bu sayede, kullanıcıların alışveriş yaparken yeterli miktarda para olup olmadığını kontrol etmek için bu method kullanılabilir.

Yöneticinin parasını azaltmak için kullanılan bu method, diğer işlemler için de kullanılabilir. Örneğin, yöneticinin para miktarını kontrol etmek için "yöneticiParaBul()" methodu kullanılabilir. Bu sayede, yöneticinin para miktarı değiştiğinde, diğer işlemlerde bu değişiklikten yararlanılabilir.

Overall, bu method, kullanıcının ürün satın alma işlemleri sırasında yöneticinin parasını azaltmak için kullanılabilir ve yöneticinin para miktarını kontrol etmek için de diğer işlemlerde kullanılabilir. Bu sayede, daha iyi bir alışveriş deneyimi sağlanabilir ve işlemler daha akıcı hale getirilebilir.

'sahiparaartır' methodu bir kullanıcının veya yöneticinin parasını artırmak için tasarlanmıştır. Method, "sahipBul()" adlı bir method kullanarak öncelikle kullanıcının veya yöneticinin kimliğini belirler ve bir değişkene atar. Daha sonra, kullanıcının seçtiği ürün adetini ve fiyatını kullanarak hesaplayarak, kullanıcının parasını arttırır ve veritabanına kaydeder.

Bu method, kullanıcının veya yöneticinin parasını arttırmak için kullanılabilir ve bu bilgilerin diğer işlemlerde kullanılması için kullanılabilir. Yönetici veya kullanıcı kimliğine bağlı olarak, "yöneticiinfo" veya "kullanıciinfo" tablolarından ilgili alanlar seçilir ve kullanıcının parası "paramiktarı" veya "yöneticipara" alanına kaydedilir.

Overall, bu method, kullanıcının veya yöneticinin parasını arttırmak için kullanılabilir ve diğer işlemlerde kullanılabilir. Bu sayede, daha iyi bir alışveriş deneyimi sağlanabilir ve işlemler daha akıcı hale getirilebilir.

'kullanıcıparabul' methodu, veritabanında kullanıcı adı 'alanSahip.Text' olan kullanıcının para miktarını döndürür. Veritabanına bağlanarak, 'kullanıciinfo' tablosundan kullanıcının para miktarını seçer ve bir değişkene atar. Bu değişken son olarak bir tamsayıya dönüştürülür ve 'returnValue' olarak döndürülür.

Bu method, kullanıcının para miktarını kontrol etmek için kullanılabilir ve diğer işlemlerde de kullanılabilir. Örneğin, kullanıcının para miktarına bağlı olarak, kullanıcının alabileceği ürünlerin sayısı kontrol edilebilir. Bu sayede, daha iyi bir alışveriş deneyimi sağlanabilir ve işlemler daha akıcı hale getirilebilir.

'sahipBul()' methodu, veritabanında yönetici adı 'alanSahip.Text' olan yöneticinin veya kullanıcının kimliğini belirler ve bir dize olarak döndürür. Bağlantı dizesini kullanarak bir SQL veritabanına bağlanır ve 'yöneticiinfo' tablosundan 'yöneticiad' alanı sağlanan 'alanSahip.Text' değerine eşit olan verileri seçmek için bir SQL komutu çalıştırır. 'COUNT(*)' fonksiyonu kullanılarak kayıtlar sayılır ve sayı değeri bir değişkene atanır. Eğer kayıt sayısı 0'dan büyükse, yönetici kimliği belirlenir ve 'yönetici' dizesi döndürülür; aksi takdirde, kullanıcı kimliği belirlenir ve 'kullanıcı' dizesi döndürülür.

Bu method, kullanıcının veya yöneticinin kimliğini belirlemek için kullanılabilir ve diğer işlemlerde de kullanılabilir. Örneğin, yönetici veya kullanıcı kimliğine bağlı olarak, yönetici veya kullanıcı özelliklerine erişim kontrol edilebilir. Bu sayede, daha iyi bir kullanıcı deneyimi sağlanabilir ve işlemler daha akıcı hale getirilebilir.

'ürünalclick' methodu, kullanıcının ürün satın alma işlemlerindeki temel işlevleri gerçekleştirir. Öncelikle, kullanıcının seçtiği ürün türüne (mağaza ya da market) ve adetine göre, kullanıcının parasını azaltır ve ürün miktarını artırır.

Bu işlem, öncelikle 'yöneticiBilgiVer()' methodunu kullanarak, kullanıcının seçtiği ürünün fiyatını ve türünü belirler. Daha sonra, kullanıcının parasını azaltmak için 'yöneticiParaAzalt()' methodunu çağırır ve kullanıcının parasını günceller.

Ardından, kullanıcının seçtiği ürün miktarını 'ürünAdet' parametresinde belirleyerek, 'kullanıciinfo' tablosundaki

‘esyamiktarı‘ veya ‘yemekmiktarı‘ alanını günceller. Bu işlem, ‘Update‘ SQL komutunu kullanarak gerçekleştirilir.

Son olarak, kullanıcının parasını artırmak için ‘sahipParaArtır()‘ methodunu çağırır ve işlemin başarılı olduğunu belirten bir mesaj kutusu görüntüler.

Bu method, kullanıcının ürün satın alma işlemlerini gerçekleştirmesini sağlayarak, alışveriş deneyimini daha iyi hale getirir ve işlemleri daha akıcı hale getirir. Ayrıca, yöneticinin parasını kontrol etmek için ‘yöneticiParaBul()‘ methodu ve kullanıcının para miktarını kontrol etmek için ‘kullanıcıParaKontrol()‘ methodu gibi diğer methodlarla birlikte kullanılabilir.

Overall, bu method, bir alışveriş işlemi sırasında gereken temel işlevleri yerine getirerek, kullanıcıların daha iyi bir alışveriş deneyimi yaşamasını sağlar ve işlemlerin daha hızlı ve akıcı bir şekilde gerçekleştirilmesini sağlar.

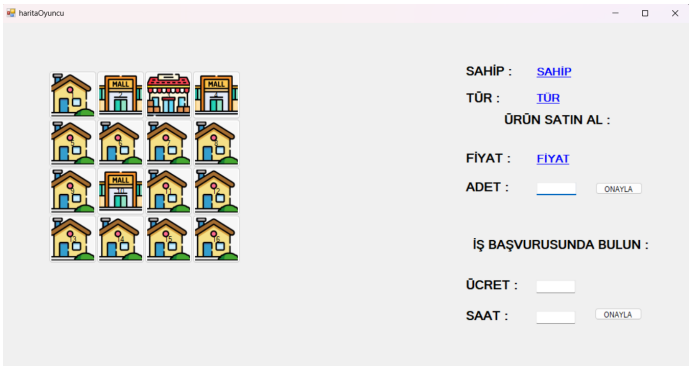
’isbasvuruonaylaclick’ methodu, bir iş başvurusu bilgisi oluşturmak için kullanılır. İş başvuruları, ”basvuruinfo” adlı bir tabloda saklanır ve bu method, sağlanan parametreleri kullanarak yeni bir iş başvurusu kaydı oluşturur.

Öncelikle, SqlConnection nesnesi oluşturulur ve bağlantı dizesi kullanılarak bir SQL veritabanına bağlanılır. Daha sonra, ”basvuruinfo” tablosuna yeni bir kayıt eklemek için bir SQL INSERT komutu çalıştırılır.

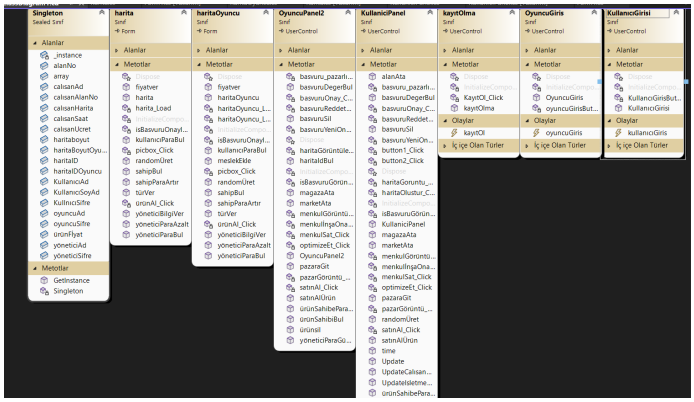
Komut, sağlanan parametreleri kullanarak, ”basvuruno”, ”basvuruisim”, ”basvurusaat”, ”basvuruucret”, ”basvurualansahip”, ”basvurualanno” ve ”basvurualanharita” alanlarını doldurur.

Son olarak, SqlCommand nesnesi kullanılarak komut çalıştırılır ve işlem başarılı olduğunda bir mesaj kutusu görüntülenir.

Bu method, yeni bir iş başvurusu kaydı oluşturmak için kullanılır. İş başvuruları, ”basvuruinfo” adlı bir tabloda saklanır ve bu method, sağlanan parametreleri kullanarak yeni bir iş başvurusu kaydı oluşturur.



E. UML diyagramı :



F. ER diyagramı :

REFERENCES

[1] https://www.youtube.com/watch?v=2EkMrrX9sYY&t=8038s
[2] https://www.youtube.com/watch?v=L86eqtdC2as&t=3633s
[3] https://learnsql.com/blog/sql-basics-cheat-sheet/
[4] https://www.youtube.com/watch?v=q_F4PyW8GTg
[5] https://dotnet.microsoft.com/en-us/learn/csharp
[6] https://learn.microsoft.com/en-us/dotnet/csharp/
[7] https://www.youtube.com/watch?v=UMc1HLyBHSQ
[8] https://www.youtube.com/watch?v=PzP8mw7JUzI
[9] https://www.youtube.com/watch?v=qLNLWw82NeE&list=PLyQXlWxYAh8-TkeUrN8viS8K69lNK1pDn
[10] https://www.youtube.com/watch?v=oSeYvMEH7jc
[11] https://www.youtube.com/watch?v=QXQN8NkINfw&list=PLAeYh-ds6kE-VvUUziIn7_pPnwUxcsteF
[12] https://www.youtube.com/watch?v=vYDyGxoq9JU
[13] https://www.youtube.com/watch?v=HXV3zeQKqGY
[14] https://www.youtube.com/watch?v=iiQPXcQHaz8&list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73_
[15] https://www.youtube.com/watch?v=GHVkRo3CRo4
[16] https://www.phind.com
[17] https://bard.google.com/?hl=en
[18] https://chat.openai.com