

Programlama Laboratuvarı II

Proje 3

Furkan Demirsoy 210201052-Önder Alp Cıvcır 200201083
Kocaeli Üniversitesi Bilgisayar Mühendisliği

I. ÖNSÖZ

Bu proje Kocaeli Üniversitesi Bilgisayar Mühendisliği Programlama Laboratuvarı II dersi 3. projesidir. Bu raporda bu projeyi nasıl yaptığımız, yaparken nelerden faydalandığımız ve karşılaştığımız hataları nasıl çözdüğümüz gibi birçok konuda bilgilendirme yapılacaktır.

II. GİRİŞ

Bu proje kapsamında sanal ortamda arsa ve market ile ilgili (alım-satım-kiralama-işletme vb.) ticari aksiyonları yerine getiren bir platformun veritabanı sistemini tasarlamamız ve bu veritabanı üzerinde gerekli işlemleri gerçekleyen bir oyun geliştirmemiz amaçlanmaktadır. Proje C++/Java/Python/C dil-

lerinden biri kullanılarak form veya internet tabanlı bir proje olarak gerçekleştirilebilir. Veri tabanı yönetim sistemi için MySQL, PostgreSQL veya MsSQL kullanılabilir ve biz geliştirdiğimiz proje için PostgreSQL veritabanı kullandık ayrıca form uygulaması olarak geliştirdik

III. ÖZET

A. Proje açıklaması

Tasarlanacak veritabanı, metaland platformu üzerindeki kullanıcılar için varlıklara yönelik

hesaplara ve tüm işlemlere ilişkin bilgileri organize bir şekilde yönetilmesine yardımcı olacaktır. Bu sayede kullanıcıların ihtiyacı olan bilgilere daha kolay ulaşabilmesi sağlanmış olacaktır.

Bir veritabanı tasarımının ilk aşamasında sistemin ihtiyaçlarının belirlenmesi ve depolanacak bilgi türlerinin tanımlanması için Varlık-İlişki (ER) diyagramı oluşturulmalıdır. ER diyagramı sistem içerisinde var olabilecek varlıkların ve aralarındaki ilişkilerin görsel olarak ifade edilmesi için kullanılır. Geliştirme sırasında, ER diyagramı gereksinimlerin daha açık ve özlü bir şekilde haritalanmasına yardımcı olmaktadır. Bizde problemlerin çözümü

için bir veritabanı tasarımı yapmanız istenmektedir. Tablo sayısı tüm tablolar en az 3NF normalizasyon formuna uyacak şekilde Bize bırakılmıştır.

- Oyun Tanımı ve Kuralları ile İlgili Bilgiler: Meta-land sisteminde oyuncu ve yönetici olmak üzere 2 farklı rolde kullanıcı bulunacaktır. Yönetici sistem içerisindeki tüm oyuncuların verilerini görebilme, kontrol edebilme ve oyun kurallarının sayısal parametrelerini belirleme yetkisine sahip kullanıcı rolüdür. Oyuncu ise oyunu oynamak için sisteme giriş yapan kullanıcıların sahip olduğu roldür. Oyun aşağıdaki kurallara göre işletilecektir:

- Oyun Kuralları:
- Oyun karesel alanlara bölünmüş bir grid alan veya harita üzerinden görselleştirilerek oynanacaktır. Bu karesel alanlar oyun içerisindeki arsalara ve üzerinde kurulacak işletme alanlarına karşılık gelecektir. Bu alanın kaç kareden oluşacağını yönetici belirleyecektir. (Örneğin 3x3 veya 4x5 vb.)
- Sistem oyuna dahil olan yeni oyunculara başlangıçta belirli bir miktar para, yiyecek ve eşya verecektir (Bu üç özellik sayısal değerler ile gösterilecektir).
- Günlük olarak her oyuncudan sahip olduğu yiyecek, eşya ve para miktarından sabit miktarda eksiltme yapılacaktır. Yiyecek veya eşya miktarlarından biri tamamen tükenen oyuncu oyunu kaybetmiş olacaktır.
- Her kullanıcı bir işletmede çalışma hakkına sahip olacaktır. Çalışma saatleri ve günlük kazanacağı para miktarı mağaza sahibiyle yaptığı anlaşmaya göre belirlenecektir. Çalışma saatleri içerisinde oyuncunun başka bir ticari işlem yapması mümkün olmayacaktır
- Oyun ilk başlatıldığında tüm alanlar (arsalar) yöneticiye ait olacaktır. Ayrıca yöneticiye ait sınırsız oyuncu kapasitesine sahip birer market, mağaza ve emlak bulunacaktır.
- Markette çalışan bir oyuncunun günlük yiyecek miktarı, mağazada çalışan bir oyuncunun günlük eşya miktarı ve emlakçıda çalışan bir oyuncunun ise günlük para miktarında çalıştığı süre boyunca sabit eksiltme yapılmayacaktır.
- Oyuncular sahip oldukları parayla işletme sahiplerinin verdiği fiyata göre marketlerden yiyecek, mağazalardan eşya ya da emlak noktalarından başka bir oyuncuya ait arsa veya işletme (market, mağaza, emlak) satın alabilecek veya işletmeyi kiralayabilecektir. Emlakçı kendi belirlediği bir miktar üzerinden alıcı-kiralayıcı ve satıcı-kiraya verenden bir komisyon ücreti alacaktır
- Bir oyuncunun sıfırdan işletme kurabilmesi için öncelikle bir arsa sahibi olması gerekmektedir.

Sonrasında bu arsa üzerine market, mağaza ya da emlak işletmelerinden herhangi birisini yöneticinin belirlediği para değeri karşılığında yönetici emlak noktasından işlem yaparak inşa edebilecektir

- Oyuncu bir emlak aracılığıyla başka bir girişimcinin sattığı arsayı alabileceği gibi yönetici emlak noktası üzerinden de bedeli karşılığında satın alabilir.
- Oyuncu üzerine işletme kurulmamış en fazla 2 arsaya aynı anda sahip olabilir. Sistem böyle bir oyuncuya daha fazla boş arsa satışına izin vermeyecektir.
- Her işletme ilk kurulduğunda 1. seviye olarak başlayacak ve bünyesinde en fazla 3 oyuncu çalıştırma kapasitesine sahip olacaktır. Sonraki her seviye için işletmenin oyuncu çalıştırma kapasitesi ikiye katlanarak artacaktır. (1. seviye - 3 oyuncu; 2. seviye-6 oyuncu; 3. seviye 12 oyuncu)
- 1. seviyeye sahip bir işletme müşteri oyuncularla yaptığı ticari işlemlerden elde ettiği gelirler dışında yöneticinin belirlediği sabit bir gelire sahip olacaktır. Bu sabit gelir işletmenin her bir üst seviye geçişinde yine yöneticinin belirlediği bir oran üzerinden artırılacaktır.
- Bir işletme bir hafta boyunca tam kapasite çalışması halinde bir sonraki seviyeye geçecektir
- Demo sırasında oyundaki diğer tüm kullanıcıların işlem yapmadığı ve tüm parametrelerin sabit kaldığı varsayılarak istenilen gün sayısı kadar tarih ileriye alınmalı ve böylece oyun hızlandırılarak ileri bir tarihteki sonuçlar görüntülenebilmelidir
- Oyuncu oyun sırasında kendisine ait oyun bilgilerini ve detaylarını (giderler, satın almalar, harcamalar, varlıklar, bütçe durumu, tüm geçmiş tercihler ve aksiyonlar) açılır bir pencere üzerinden görüntüleyebilmeli, benzer şekilde yönetici ise tüm oyuncuların bilgilerini izleyebilmelidir
- Tüm bu veriler ve veri hareketleri veritabanında tutulmalı ve oyunun ilerleyişine göre

güncellenmelidir. Arayüzde gösterilen bilgilerin tamamı yine bu veritabanını üzerinden yapılan gerekli sorgularla elde edilmelidir.

- Tasarlanan arayüzde oyuncunun kendi yaptığı tüm işlemler gerek oyuncuya göre gerekse işletmeye göre filtrelenebilir ve kronolojik olarak gösterilmelidir.

IV. YÖNTEM

A. Araştırma

İlk olarak projeyi java ve swing kütüphanesi ile yapmaya karar verdik ve java ile veritabanı bağlantısı(JDBC) sağlamak için Apache netbeans uygulamasının içerisinde hazır olarak var olan PostgreSQL JDBC Driver ı kullandık. JDBC bağlantısı

sağlandıktan sonra Veritabanı tabloları içerisinde bilgi eklemek veya bilgi silmek,bulmak,listelemek gibi işlemleri dinamik bir şekilde gerçekleştirmek için bir takım araştırmalarda bulunduk ve bu araştırmaların sonunda Entity-Manager Pattern adı verilen her tablonun entity paketi içerisinde bir class olarak temsil edildiği ve her attribute ün de türüne göre liste içerisinde private field oluşturulduğu ve get,set metotları ile bilgilerin alındığı ayrıca her entity classı için de manager paketi içerisinde entity.name.manager ismiyle isimlendirilen sınıflarda da o entity türüne göre list,find,delete,update,insert ismi verilen fonksiyonlar oluşturularak dinamik bir şekilde veritabanı işlemlerinin oluşturulmasına karar verildi.Bu pattern kullanımı sayesinde her class bir entity i temsil eder ve entity içerisindeki her attribute de bir nesne haline gelir ve bu sayede nesneler üzerinde rahatlıkla yapılan işlemler veritabanına bağlandı.

B. Arayüz ve Oyun

Arayüzün oluşturulması için Swing kütüphanesi kullanıldı program ilk çalıştırıldığında karşımıza çıkan log in ekranı ile kullanıcıları user name ve password bilgisine göre girişi sağlandı ayrıca sign up butonu ile kullanıcıların kayıt oluşturulması sağlandı bu sayede boş bilgi ya da yanlış bilgi

giren kişilerin sisteme alınmaması ve giriş esnasında yaptıkları hatalar da ContentPane ekranları ile kullanıcılara bildirildi

Giriş ekranının back-end kısmının oluşturulmasında com.MetaLand.Game.entity paketi içerisinde SQL sunucusundaki Kullanıcı tablosunu temsil eden User class ı kullanıldı.Bunun için her kullanıcı önce kayıt yaptırmak zorunda bırakıldı bu kayıt sırasında textbox lar içerisinde yazılan bilgiler alınarak user.manager classı içerisindeki insert metodu ile veritabanına kaydedildi bu kayıt gerekli Sql komutlarıyla sağlandı Kayıt işlemi tamamlandıktan sonra girişin

yapılması için GUI class ı içerisinde o anda ki user name ve password bilgisi veritabanında olan kişi yine user.manager class ı içerisindeki list fonksiyonu ile tespit edildi bu fonksiyon çalışırken gelen kullanıcı bilgileri içerisinde password ve user name bilgisi girilen bilgiyle aynı olan kişi geldiğinde döngüyü break ile keserek o kişinin userNo bilgisini yani unique primary key numarasını aşağıda çağrılan diğer classlara ait construct ve fonksiyonlara parametre olarak gönderdik. Oyunun iç kısmının arayüzünün

olluşturulmasında ise yöneticiye ait panel ve diğer kullanıcılara ait panel olmak üzere 2 ye böldük bunun için yönetici paneli oyun kuralları kısmında belirtilen günlük giderleri belirleme,oyun boyutunu belirleme,kullanıcı bilgilerini izleme,tarihi ileri alma gibi işlemlerin gerçekleştirilebileceği şekilde tasarlandı. Oyunun kullanıcıların oynayacağı kısmı

ise arsların ve işletmelerin gösterimi ve kullanıcının mülk,para,yemek,eşya gibi bilgilerini görebileceği bölgeler ve oyunu oynaması için örnek olarak yiyecek,eşya almak işe girmek ve mülk satın almak gibi işlemleri yapabileceği butonlar eklendi

V. TABLO NORMALİZASYON İŞLEMLERİ

Tabloların Normalizasyonu için 3.Normal Form kullanılmıştır ve bunun için aşağıdaki işlemler uygulanmıştır:

- İlk olarak tabloların 1. ve 2.Normal Forma getirilmesi ve tablolardaki unique özelliğin sağlanması ve veri tekrarının oluşmaması için ve tabloların birbirlerine bağlanmaları için 2 adet temel primary key in oluşturulmasına karar verildi bunlar kullanıcı no ve alan no olarak belirlendi.Bu sayede her tuples içerisinde sadece 1 veri bulunması sağlandı ayrıca kullanıcı ve alan numaraları oluşturularak fonksiyonel bağımlılıklar azaltılmış ve her kullanıcı ya da her alan bulunduğu tabloda sadece 1 row da bulunacak şekilde oluşturulmuş oldu.
- Daha sonra tablolardaki veri tutarlılığının sağlanması için diğer tablolardaki Alan no ve Kullanıcı no Attribute leri Kullanıcı tablosu içerisindeki kullanıcı no ya ve Alan tablosu içerisindeki Alan no Attributelerine Foreign key olarak bağlandı bu sayede veritabanını bilmeyen birisi bu tablolara bir veri girmek istediğinde eğer kullanıcı tablosu içerisinde bulunmayan bir id girerse program bunu bir hata olarak kabul eder ve bu bilgiyi sisteme girmez.
- Son aşama olarak tabloların 3.Normal Forma uygun hale getirilmesi için fonksiyonel bağımlılıkların azaltılması sağlandı bunun için alan boyutu,gunluk gider ve işletme kapasitesi tabloları ayrıldı ve alan boyutu bilgisi alan tablosuna bazı id ler ile eklendi bu sayede örnek olarak alan boyutu bilgisi değiştirilmek istendiğinde tüm alanların alan boyutu bilgilerinin değiştirmeye gerek kalmadan direkt alan boyutu tablosunda id nin bağlı olduğu alan boyutunu değiştirerek elde edebiliriz ve bu tabloyu 3.Normal Forma fonksiyonel bağımlılıkları azalttığı için ulaştırmaktadır.

VI. DENEYSEL SONUÇLAR

Veritabanı ile java bağlantısı sağlanmaya en başta MSSQL kullanıldı ancak apache netbeans MSSQL JDBC Driver ı bulamadı daha sonra MYSQL denendi ancak bu da olmadı sonuçta Apache Netbeans içerisinde hazır bulunana PostgreSQL Driver ı tanıdı ve bu sebepten postgres kullanıldı buna ek olarak program yazılırken en başta entity-manager patterni kullanılmadı ve bunun sonucunda dinamik bir şekilde veri hareketi durumunun çok zor olacağı tespit edildi ve bu sebepten entity-manager patterni kullanıldı.

VII. KAYNAKÇA

- <https://www.javatpoint.com/java-jdbc>
- <https://www.postgresql.org/>
- <https://www.freecodecamp.org/news/java-string-to-int-how-to-convert-a-string-to-an-integer/>
- <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>
- <https://www.guru99.com/database-normalization.html>

VIII. SONUÇ

JDBC ile Java ve veritabanı bağlantısının nasıl sağlanacağı ve entity-manager patterni öğrenildi ayrıca SQL komutları ve problem çözme becerisinin gelişmesi sağlandı.

IX. ENTITY RELATIONSHIP DİYAGRAMI

Diyagram aşağıda verilmiştir

