

Restoran Bilgi Sistemi

Ege Üniversitesi Bilgisayar Mühendisliği
Veri Yapıları

Proje-3

Hüseyin YAŞAR 05-06-7657

Didem KAYALI 05-06-7669

Umut BENZER 05-06-7670 <http://www.ubenzer.com/>

Özlem GÜRSES 05-07-8496

Teslim tarihi: 22 Aralık 2008



1. Programcı Kataloğu

1.1-Platform ve Dil

Projenin tamamı tüm çalışanlar tarafından yapılmıştır, ancak birleştirimi ve düzenlenmesi Umut Benzer'e aittir. Dolayısıyla proje resmi olarak, Windows Vista Ultimate SP1 üzerinde 4.3 Ganymede kullanılarak gerçekleştirilmiştir.

Projenin yazımında kullanılan dil JAVA 5 (JDK 1.6)dir.

1.2-Problem

Projede her restoran için belli bilgilerin tutulacağı ve değişiklik yapılabileceği bir restoran bilgi sistemi oluşturulması istenmektedir. Restoranlara ilişkin bilgiler restoranın adı, tam adresi, telefon, faks, e-posta, internet adresini kapsamaktadır. Ayrıca her restoran için bir de mönü bilgisi gerekmektedir. Mönünün de kendi içinde kategorilere ayrılması istenmiştir. Yemek bilgileri (yemek adı, fiyatı, tarifi, malzemeler) kategorilere göre düzenlenecektir. Bu bilgiler üzerinde ekleme, silme ve güncelleme işlemleri yapılabilir olmalıdır.

Ayrıca sistemin müşterilere yönelik kısmında çeşitli arama imkanları da sağlanmalıdır. Kullanıcılar restoran adına, yemek adına, verilen bir fiyat aralığına, içindeki malzemeye ve illere göre arama yapabilmelidirler.

1.3-Sınıflar ve Metotlar

Console: Java'nın konsolunun proje çalışanları tarafından yetersiz bulunması dolayısıyla çalışanlardan Umut Benzer bu proje için kendisi bir konsol tasarlamıştır. Bahsi geçen yetersizlik, program çalıştırıldığında ekran temizleme sisteminin dahi olmaması, ya da ekrana yazdırılacak bir bilginin çok uzun bir kodla yazdırılması gibi durumlardır. Daha sonraki sınıflarda kullanılacak konsolların metotları console sınıfı içerisinde yer almaktadır. Söz konusu metotlar görsel editör tarafından otomatik olarak yaratılmıştır.

Console içerisinde kullanılan metotlar:

`public void write(String metin)`, `System.out.println` in yaptığı gibi ekrana bilgi yazdırılmasını sağlar.

`public String read(String msg, boolean acceptNull)` metodunda `msg` ekrana yazdırılacak metni alır. `acceptNull` ise geriye null değeri döndürülüp döndürülmediğini kontrol eder.

`public int readInt(String msg)` metodu aynı bir önceki gibi ekrandan veri alır ancak alınacak veri kesinlikle integer tipinde olmalıdır. Aksi halde sürekli veriyi ister. Kullanıcının integer dışında değer girmesi durumunda programın çökmesi gibi bir durum söz konusu değildir.

`public boolean isParsableToInt(String i)` parametre değeri olarak girilecek i'nin integer olup olmadığını kontrol eder.

`public float readFloat(String msg)` tıpkı 3. Metot gibi veri alır ancak alınan verinin float tipinde olmasını şart koşar.

`public boolean isParsableToBoolean(String i)`, girilen i'nin boolean tipinde olup olmadığını kontrol eder.

`public void cls()` metodu ekranı temizler.

`public void beklet()`, "devam etmek için bir tuşa basın." Diyerek kullanıcıyı bekleten metottur.

Kategori: Kategori bilgilerini tutmak için kullanılan sınıftır. Sınıf içerisinde kullanılan metot, adı verilen kategorideki yemekleri `Vector<String>` tipinde bir vektörde döndüren `public Vector<String> getYemek(String kategoriAdi)` metodudur.

Main: Verilerin diskten okunup üzerinde tüm işlemlerin yapılması ve programın kapatılmasına kadar olan her şey bu sınıf tarafından koordine edilir. Mainde kullanılan metotlar:

`public static int save(String dosyaAdi)`, dosyaAdi ile verilen dosyaya tüm verileri diske yazan metottur.
`public static int load(String dosyaAdi)`, daha önce diske yazılmış verileri okuyor ve yerlerine yerleştirilen metottur.

`public static String hakkında()`, metodu String tipinde hakkında bilgisini ekrana yazdırır.

`public static void main(String[] args)`, ana main metodudur. menü, alt menüler, bilgi alımı, ilgili metotların çağrılması, ilk başta diskten okuma ve en sonda diske yazma gibi işlemler bu metot içerisinde gerçekleştirilir.

`public static boolean restoran_ekle(String restoranAdi, String il, String adres, String telefon, String faks, String eposta, String www, YemekR yemekler)`, verilen parametre bilgileri ile sisteme yeni bir restoran ekleyen metottur.

`public static boolean restoran_guncelle(String oldRestoranAdi, String newRestoranAdi, String newIl, String newAdres, String newTelefon, String newFaks, String newEposta, String newWWW)`, yine parametre değerleri ile verilen restoran ile ilgili bilgilerin güncellenmesini sağlayan metot olup, bir bilgi güncellenmek istenmediğinde null göndererek o bilginin güncellenmemesini sağlar.

`public static boolean restoran_sil(String restoranAdi)`, eğer silinmek istenen restoran mevcutsa true göndererek silme işlemini yaptığını belirten metottur.

`public static boolean is_kategori_exists(String kategoriAdi)`, istenilen kategorinin sistemde olup olmadığına bakar. Böylelikle ekleme ya da silmenin yapılmasına karar verilir.

`public static boolean kategori_sil(String kategoriAdi)`, eğer mevcutsa kategoriyi silen metottur.

`public static boolean kategori_ekle(String kategoriAdi)` metodu, kategori eklemek için kullanılır.

`public static boolean malzeme_ekle(String malzemeAdi)`, malzeme eklenmesini sağlayan metottur.

`public static boolean yemek_duzenle(String oldYemekAdi, String newYemekAdi, String newEkBilgi, String newKategorisi)`, verilen parametrelerle yemek bilgilerinde düzenleme yapar.

`public static boolean is_yemek_exists(String yemekAdi)`, eklenmek ya da silinmek üzere olan yemeğin var olup olmadığına bakar. Ekleme ve silme metotları içinde çağrılır.

`public static boolean yemek_sil(String yemekAdi)`, sistemde mevcutsa yemeği silme metottur.

`public static boolean yemek_ekle(String yemekAdi, String ekBilgi, Vector<String> olduguRestoranlar, String kategorisi, Vector<String> icindekiMalzemeler)`, tüm bilgileriyle sisteme yeni yemek ekler.

Restoran sınıfı, bir restorana ait bilgileri tutan sınıf olup, yapıcı metot dışında metot barındırmamaktadır.

Tree: sınıfı ağaç yapısını tutan sınıftır. Metotları:

`public Restoran getRestoran(String restoranAdi)`, adı verilen restoranı ağaç içerisinde bulup getirir. Eğer böyle bir ağaç yoksa null döndürür.

`public void addRestoran(Restoran gelenRest)`, ağaca yeni restoran ekler.

`public boolean removeRestoran(String restoranAdi)`, adı verilen restoran varsa ve silindiyse true, aksi hallerde false döndüren metottur.

`public Vector<Restoran> preOrder()`, ağacı preorder dolaşp tüm restoranları preorder sırada bir vektöre koyup sonra bu vektörü döndüren metottur. `public Vector<Restoran> inOrder()` ve `public Vector<Restoran> postOrder()` aynı işlemleri inorder ve postorder olarak yapan metotlardır.

`public int depthMeter (Restoran localRoot)` metodu verilen restoran nesnesinin ağacın hangi derinliğinde olduğunu döndürür.

Yemek sınıfı sistemdeki yemekleri vektör olarak tutar, metodu yoktur.

YemekR sistemde bulunan bir yemeğin ilgili restorana eklenmesini ya da o restorandan silinmesini sağlayan sınıftır. Restoran ve yemek bazında fiyat tutar ve her restoran için bir YemekR sınıfı bulunur . kullanılan metotlar:

`public void addYemek(String yemekAdi, float fiyati)`, adı ve fiyatı verilen bir yemeğin o restorana eklenmesini sağlayan metottur.

`public void removeYemek(String yemekAdi)`, adı verilen yemeğin silinmesini sağlar.

`public void addYemek(YemekRYapi yemek)` direk YemekRYapi nesnesi göndermek için kullanılan metottur.

`public boolean hasYemek(String yemekAdi)` yemek eklemekten ya da silmeden önce yemeğin varlığını adına göre kontrol eden metottur.

`public float getYemekFiyati(String yemekAdi)` metodu girilen yemekadına göre o restorandaki yemeğin fiyatını döndürür.

`public void changeYemek(String oldYemekAdi, String newYemekAdi)` sistemdeki yemeğin adı değiştiğinde restoranlardaki adının da değişmesini sağlayan metottur.

`public void changeYemek(String yemekAdi, float newFiyati)`, kullanıcının yemeğin fiyat bilgisini de değiştirmesini sağlar.

`public YemekR yemekFiltre(float fiyatAltLimit, float fiyatUstLimit)`, metodu o restorana ait fiyat aralığı belirliyor. iki fiyat aralığı verildiğinde o aralıktaki yemekleri bir YemekR nesnesi içerisinde gönderiyor.

YemekRYapi bir restorana ait yemekleri tutan ve metodu olmayan sınıftır.

1.4-Veri Yapıları

Programda ağaç ve vektör yapıları kullanılmıştır. Ayrıca malzeme ve yemek listesinin tutulması için Hash tablosu kullanılmıştır. Veri yapılarının kullanımı açıkça aşağıdaki tabloda görülebilir.

Tree

```
|
|Restoran
| |
| |String ad
| |    |String il
| |String adres
| |String telefon
| |String faks
| |String eposta
| |String www
| |YemekR yemekler
| | |
| | |Vector
| | | |
| | | |YemekRYapi
| | | | |
| | | | |String yemekAdi
| | | | |float fiyatı
```

Console

Main

HashTable

```
|
|Yemek
| |
| |String ekBilgi
| |String kategorisi
| |Vector
| | |
| | |String icindekiMalzemeler
| | |
| |Vector
| | |
| | |String olduguRestoranlar
```

HashTable

```
|
|Malzemeler
| |
| |Vector
| | |
| | |String olduguYemekler
```

Vector

```
|
|Kategori
| |
| |String kategoriAdi
| |Vector
| | |
| | |String aitYemekler
```

1.5-Dosya Özellikleri

ObjectInputStream ve ObjectOutputStream nesneleri ile nesneler direk bellekten alınıp diske yazılır. Böylece metin metin eklemeye gerek kalmaz. Aynı durum yükleme için de geçerlidir. Onun için tüm bilgiler tek dosyada tutulabilir. Yani içindeki tüm bilgilerle birlikte nesne diske yazılmış olur. Dosyanın okunması ve yazılması programın başında ve sonunda yapılır. Programın işleyişi sırasında dosya kullanılmaz her şey bellekte gerçekleşir. En sonunda restoran.cokmezbilisim dosyasına yazılır. Söylediğimiz gibi dosya programın başında okunur, sonunda yazılır ve bu işlemleri Maindeki **load** ve **save** metotları yürütür.

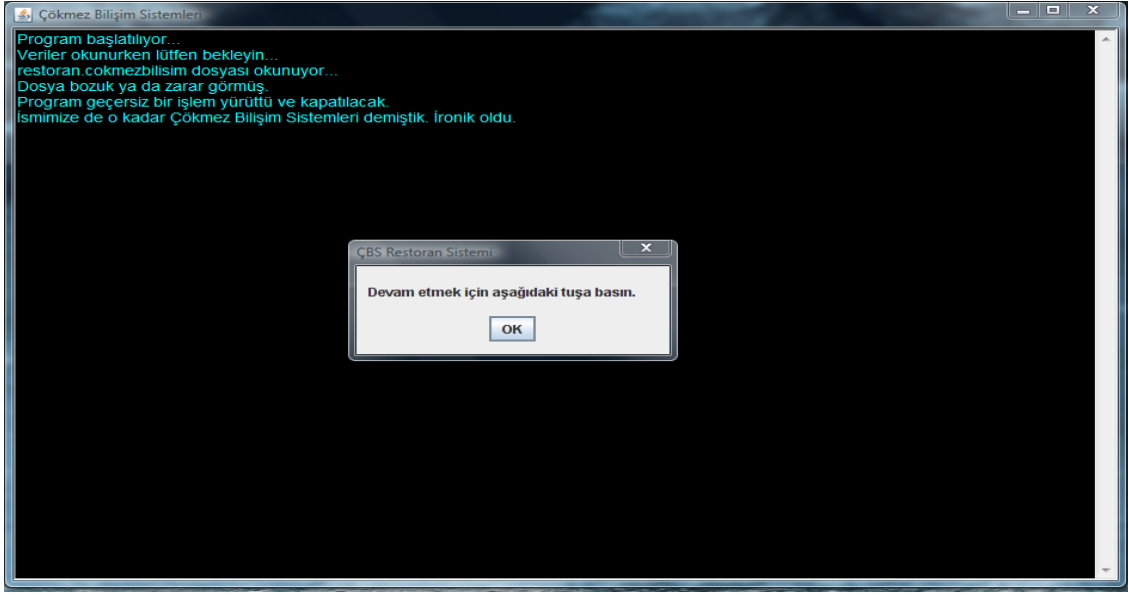
1.6-Çalışma Süreleri

Projenin tamamlanması, karar verme süreci dahil olmak üzere toplamda yaklaşık 35 saat almıştır. Bu ortalama geçen süre içerisinde her bir çalışan kendi esnekliğine bağlı olarak günde en az 4 saatlik çalışma yapmıştır. Ancak proje elemanları tarafından üretilen kod parçalarında oluşan hataların düzeltilmesi, derlenmesi ve main ve console gibi oldukça zorlu sınıfların yazılması gibi işler Umut Benzer'e ait olduğu için proje için en çok çalışan eleman olup tam bir verimle 35 saatlik çalışma yapmıştır.

2. Kullanıcı Kataloğu

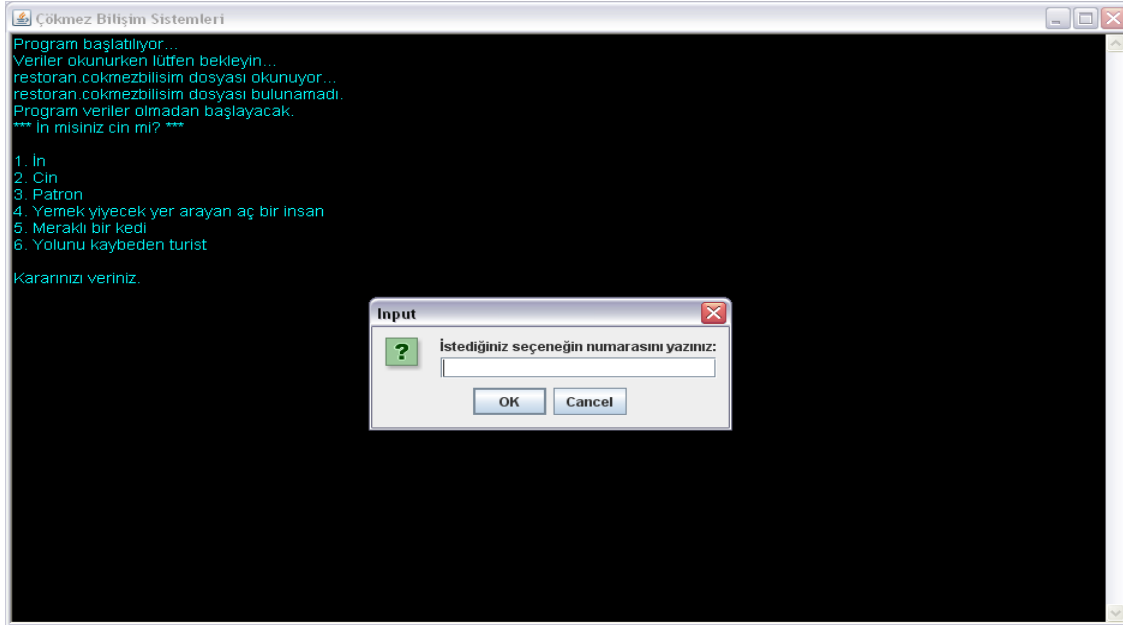
2.1-Programın İşletimi

Program yetkili kişiler ve müşterilere göre iki ayrı bölüm içermektedir. Yetkili kişiler sisteme yeni bilgiler ekleyebilir, sistemdeki bilgilerde değişiklik yapabilir ve sistemden bilgileri silebilirlerken müşteriler sadece belirli kriterlere göre arama ve listeleme yapabilmektedirler. Müşteriler beklentilerine göre çok çeşitli kriterlere göre arama listeleme yapabilmektedirler ve menü seçenekleri müşteriler açısından kolay anlaşılabilir şekilde düzenlenmiştir. Kullanıcı menüler içerisinde dolanarak işlemlerini gerçekleştirip programdan çıkar.



2.2-Kullanım Kılavuzu

Çökmez Bilişim Sistemleri'nden yeni nesil restoran sistemleri...
Siz de Çökmez Restoran Sistemlerini tercih edin, karlı çıkın!
ÇRS başlatıldığında bir menü sunulur.



Menüdeki ilk iki şık espri amaçlı oluşturulmuştur.

Menüde üçüncü şık patron girişi içindir. Burada gerekli işlemlerin yapılabilmesi için kullanıcı adının ve şifrenin doğru girilmesi gerekir.(**Kullanıcı adı: daydin, Şifre: ant**) Restoran sahipleri veyahut yetkili kişiler burada karşlarına gelen menüde restoran işlemleri için biri, yemek kategorileri hakkında işlemler için ikiyi, yemek işlemleri için üçü, yemek malzemeleri hakkındaki işlemleri için dördü ve sistemi terk etmek için beşi seçerler.

Yetkili kişiler, restoran işlemleri için açılan menüde; biri seçerek açtıkları yeni restoranlarını sisteme ekleyebilir, ikiyi seçerek restoranlarına yeni bir yemek çeşidi ekleyebilir, üçü seçerek restoranlarından yemek çeşidi silebilir, dördü seçerek restoran bilgilerinde düzenleme yapabilir, beşi seçerek yemek fiyatlarında düzenleme yapabilir, altıyı seçerek restoranlarını silebilir ve yediyi seçerek bir üst menüye çıkabilirler.

Yetkili kişiler, yemek kategori işlemleri için açılan menüde; biri seçerek sisteme yeni bir kategori ekleyebilir, ikiyi seçerek sistemdeki herhangi bir kategoriye yeniden adlandırabilir, üçü seçerek sistemdeki herhangi bir kategoride bulunan yemekleri listeleyebilir, dördü seçerek adı verilen kategoriye ve bu kategorideki yemekleri sistemden kaldırabilir ve beşi seçerek bir üst menüye çıkabilirler.

Yetkili kişiler, yemek işlemleri için açılan menüde; biri seçerek sisteme yeni bir yemek ekleyebilir, ikiyi seçerek sistemde var olan yemeklerin bilgilerini düzenleyebilir, üçü seçerek sistemde var olan bir yemeğe yine sistemde kayıtlı olan bir malzeme eklemesi yapabilir, dördü seçerek sistemde kayıtlı olan bir yemeğin içindeki malzemeleri listeleyebilir, beşi seçerek sistemdeki bir yemeği menülerinde bulunduran restoranları listeleyebilir, altıyı seçerek sistemdeki bir yemekten herhangi bir malzemeyi çıkarabilir, yediyi seçerek sistemden bir yemek çeşidi silebilir ve sekizi seçerek bir üst menüye çıkabilirler.

Yetkili kişiler, yemek malzeme işlemleri için açılan menüde; biri seçerek yemeklerde kullanılmak üzere sisteme yeni bir malzeme ekleyebilir, ikiyi seçerek girdikleri malzemenin hangi yemeklerde bulunduğunu öğrenebilir ve üçü seçerek bir üst menüye çıkabilirler.

Yetkili kişiler Yönetici menüsünden çıkış için beşi seçmelidirler.

Menüde dördüncü şık müşteri menüsüne açılan şıktır. Müşteri menüsünden bir seçilerek adı girilen bir restoranın sistemde var olup olmadığı aranır. Restoran sistemde kayıtlıysa; şehri, adresi, telefo-

nu,elektronik postası, internet adresi ve m n s  g r nt lenir. Men den; iki se ilerek adı verilen bir yeme in  ıktı ı restoranlar,    se ilerek girilen bir fiyat aralı ındaki yemekler ve bulundu u restoranlar, d rt se ilerek girilen bir malzemenin bulundu u yemekler ve yemeklerin  ıktı ı restoranlar, be  se ilerek girilen bir ilde bulunan t m restoranlar ve bilgileri listelenebilir. Men den be  se ildi inde bir  st men ye ge i  yapılır.

Men de be inci  ık, program yapım ıları ve s r m hakkında bilgi almak i in kullanılır.

Men de altıncı  ık, programdan  ıkmak i in kullanılır.

2.3-Programın Kısıtlamaları

Programın  o u kısmında cancel i lemi  alı mamaktadır. Bunun yerine ok'e direk tıklamanız cancel i lemi yerine ge mektedir.

Bir restorana sistemde kayıtlı olmayan bir yemek eklenememektedir.  nce sisteme yemek eklemek sonrasında restorana eklemek gerekmektedir. Ancak aynı durum herhangi bir yeme e malzeme eklemekte ge erli olmamaktadır, sistemde var olmayan bir malzemeyi yeme e eklemeye  alı tı ınızda program malzemeyi otomatikman sisteme de eklemektedir.

Yemek, malzeme ve kategori isimleri b y k k   k harf duyarlıdır, dikkat edilmesi gerekir.