**REZERVASYON SINIFLANDIRMA SİSTEMİ**

*Gözde ÖRGÜ - Büşra ERKAN*

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[gozde.orgu@gmail.com](mailto:gozde.orgu@gmail.com) [busraerkan39@gmail.com](mailto:busraerkan39@gmail.com)

**Özet**

*Rezervasyon Sınıflandırma Sistemi projesinde farklı kullanıcıların farklı zamanda ve yerlerde yapmış oldukları rezervasyonların kayıtlarını içeren bir rezervasyon verisi ağaç ve bağlı liste yapıları kullanılarak temel işlevleri yerine getiren bir uygulamanın yapılması beklenmektedir.*

**1.Giriş**

Rezervasyon Sınıflandırma Sistemi projesinde rezervasyon kayıtlarını içeren bir rezervasyon verisi ağaç ve bağlı liste kullanılarak gösterilir.

Proje Rezervasyon, RezervasyonBilgileri, Heap ve MainClass sınıflarından oluşur.

RezervasyonBilgileri sınıfında tanımlanan kullanıcıId, yerId, zaman, enlem, boylam, sehir, kategori ve altkategori değişkenlerinde rezervasyon verisinde verilen bilgiler tutulmaktadır. Heap sınıfında ağacın altında bulunan kategori düğümlerinden sonra rezervasyon işlemini yapan müşteri bilgilerinin yer aldığı kullanıcı düğümleri Max-Heap algoritmasına göre oluşturulup ağaca eklenmiştir. MainClass sınıfında dosyadanOku() fonksiyonu çağırılır ve kullanıcıya, ağaca veri eklemesi ya da silmesi için bir menü sunulur. Ağaç oluşturma, yazdırma, veri ekleme ve veri silme işlemleri Rezervasyon sınıfında gerçekleşir.

Tasarlanan sistemde kayıt bilgileri “rezervasyon.txt” dosyasından çekilir ve bu bilgilerle ağaç yapısı oluşturulur. Ağaç “rezervasyon” kök düğümünden başlar. Aşağı doğru kategoriler ve alt kategoriler için kayıtlardaki sıra dikkate alınarak alt düğümler oluşturulmaktadır. Ağacın altında bulunan kullanıcı düğümlerinde Max-Heap algoritması kullanılmıştır.

Kullanıcı kendisine sunulan menü ile kategori ve kullanıcı ile ilgili işlemler, sorgu ve listeleme işlemlerini de gerçekleştirebilir.

**2.Temel Bilgiler**

Program Java dilinde geliştirilmiş olup tümleşik geliştirme ortamı olarak NetBeans IDE kullanılmıştır.

**3.Tasarım**

Rezervasyon Sınıflandırma Sistemi Projesi’nin programlanma aşamaları aşağıda belirtilen başlıklar altında açıklanmıştır.

**3.1 Algoritma**

Ağaçta müşteri bilgilerinin yer aldığı kullanıcı düğümleri için Max-Heap Algoritması kullanılmıştır.

**3.2 Yöntem**

Bu proje dört sınıftan oluşmaktadır. Rezervasyon sınıfında Tree ve Node; RezervasyonBilgileri sınıfında RezervasyonBilgileri ve RezervasyonGrup sınıfları bulunmaktadır.

Kayıt bilgilerini tutmak için RezervasyonBilgileri ve RezervasyonGrup sınufları kullanılmıştır. Rezervasyon sınıfında dosyadanOku() fonksiyonu ile kayıt bilgileri proje dosyası içindeki “rezervasyon.txt” belgesinden okutulur. Okutulan bilgiler ağaç yapısını oluştururken kullanılması için ArrayList’lere atılmıştır.

agacOlustur() fonksiyonunda ilk önce “rezervasyon” isimli kök düğüm oluşturulmuştur. Ağaç oluşturulurken kategori isimlerinin ilk harfinin ‘M’ harfinden küçük olup olmamasına bakılır. Küçükse sol alt ağaca, büyükse sağ alt ağaca eklenir. Ağaca kategori eklenirken aynı isimde bir kategori var mı diye kontrol edilir. Eğer aynı isimde bir kategori bulunuyorsa o kategorinin alt kategorisi kontrol edilir. Alt kategori varsa altKategoriEkle() fonksiyonu çağırılır. altKategoriEkle() fonksiyonunda da ağaca ekleme yapılırken alt kategori isimlerinin ilk harfinin ‘M’ harfinden küçük olup olmadığına bakılır. Küçükse sol alt ağaca, büyükse sağ alt ağaca eklenir. Aynı isimde bir alt kategori var mı diye kontrol edilir. Varsa işlem yapılmaz, yoksa yeni alt kategori eklenir ve kategoriRezervasyon() fonksiyonuna gidilir. Bu fonksiyonda txt den okutulan bilgileri tuttuğumuz rglist kullanılarak o alt kategorideki rezervasyon sayıları ArrayList’e eklenir. Kullanıcı düğümlerinin ağaçlarının oluşturulması için HeapTreeOlustur() fonksiyonu çağırılır. Bu fonksiyonda önce ağacın kökü oluşturulur. Düğüm oluşturulması için HeapTreeOlusturRecursive() fonksiyonu çağırılır. Bu fonksiyonda ekleme soldan sağa doğru yapılır. Ağacın Max Heap Algoritması’na

uygunluğunun kontrolü için her ekleme yapldığında kontrolMaxHeap() fonksiyonu çağırılır. Fonksiyonda eklenen eleman eklendiği kök düğümden büyük mü diye kontrol edilir ve büyükse yer değiştirme işlemi yapılır.

MainClass sınıfında kullanıcıya bir menü sunulur. Kullanıcının isteğine göre kategori ekleme, kategori silme, alt kategori ekleme, alt kategori silme, kullanıcı ekleme, kullanıcı silme, kullanıcıya göre kategori listeleme, kategoriye göre kullanıcı listeleme, rezervasyon yerine göre kullanıcı listeleme ve kullanıcıya göre rezervasyon listeleme işlemleri yapılır.

**3.3 Kullanılan Fonksiyonlar**

*dosyadanOku()*

rezervasyon.txt dosyasındaki veriler çekilir ve ArrayList’lere atılır

*agacOlustur()*

rezervasyon.txt’den okutulan bilgilerle kategoriler ağaca eklenir.

*altKategoriEkle()*

Ağaçtaki kategori düğümlerine rezervasyon.txt’den okutulan bilgiler eklenir.

*kategoriRezervasyon()*

Kullanıcı düğümlerinde kullanılacak olan rezervasyon sayıları bulunur.

*HeapTreeOlustur()*

Heap ağacı oluşturulmaya başlanır.

*HeapTreeOlusturRecursive()*

Max Heap ağacının sağına ve soluna eklemeler yapılıyor.

*kontrolMaxHeap()*

Eklenen elemanın ait olduğu kökten büyük olup olmadığı kontrol edilir. Büyükse değiştirme işlemi yapılır.

*treeNode()*

Ağacın sağ ve sol düğümlerini yazdıran fonksiyon.

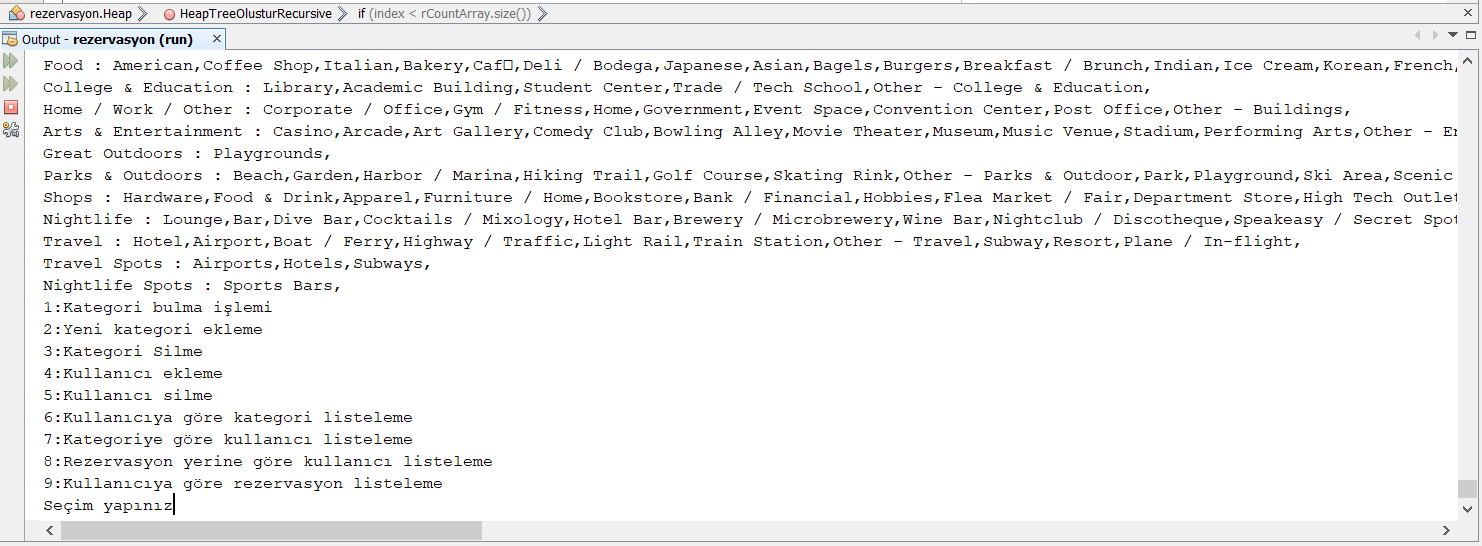
*display()*

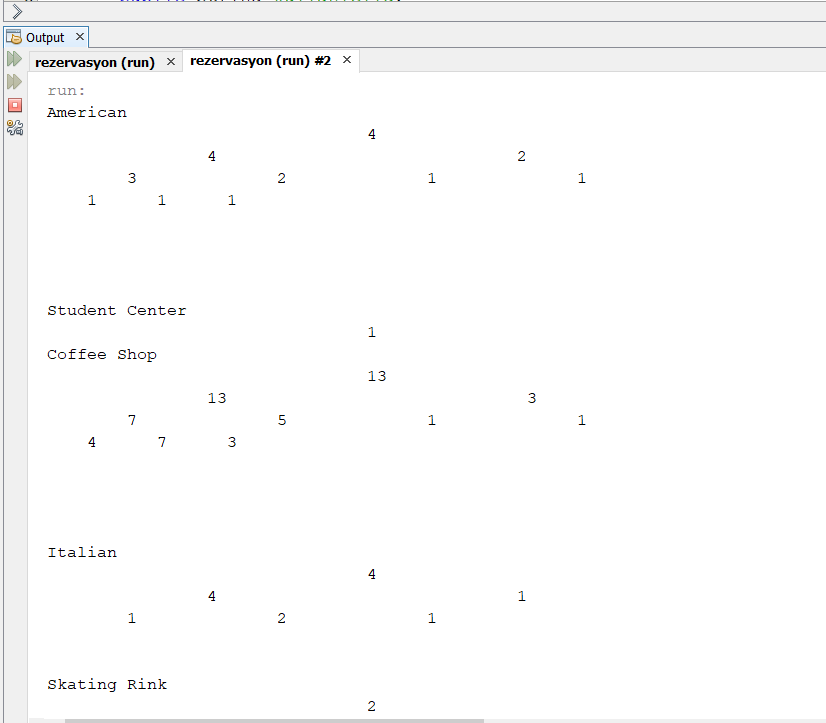
Kullanıcı düğümlerinin ağacı, ağaç yapısına uygun olarak ekrana bastırılır.

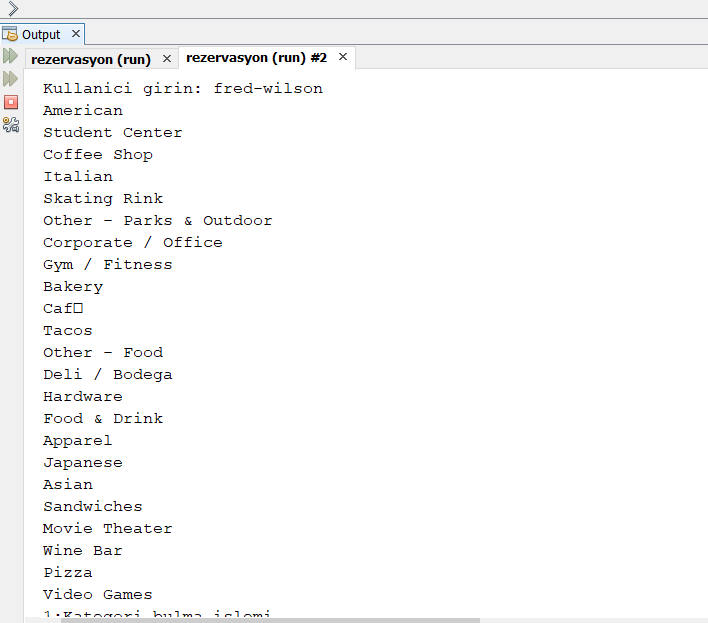
**4.Akış Şeması**



**5.Sonuç**

****

****

****

**6.Kaynakça**

display() fonksiyonu aşağıdaki siteden alınmıştır.

<http://homepage.divms.uiowa.edu/~sriram/21/spring07/code/heap.java>

<https://www.kodlamamerkezi.com/java/java-for-each-dongusu-kullanimi/>

<https://www.youtube.com/channel/UCyo2F4E2_hNz_b6t63RWH4w>