

Kocaeli Üniversitesi
Bilgisayar Mühendisliği Bölümü
BLM 209 Programlama Lab. I, 2018-2019 Güz
Proje 4
(Proje Son Teslim Tarihi: 4 Ocak 2019)

REZERVASYON SINIFLANDIRMA SİSTEMİ

Bu proje kapsamında farklı kullanıcıların farklı zamanda ve yerlerde yapmış oldukları rezervasyonların kayıtlarını içeren bir rezervasyon verisi üzerinde ağaç (tree) ve bağlı liste (linked list) yapıları kullanılarak temel işlevleri yerine getiren bir uygulamanın yapılması beklenmektedir.

Amaç: Bu proje sayesinde projeyi yapacak kişinin veri yapıları konusunda daha önce edinmiş olduğu teorik bilgileri pratiğe dönüştürmesi ve istenenlere çözüm bulabilmesi amaçlanmaktadır.

1. Problem

Bir firma farklı alanlarda hizmet veren işletmeler için müşterilerin (hizmet alan) rezervasyon işlemlerini, internet üzerinden yapabilmelerine imkan sağlayan bir sistem sunmaktadır. Bu firma rezervasyon bilgilerini içeren sıralı verilerini, işletmelerin hizmet alanlarına göre sınıflandırmak ve bu sayede verilerle ilgili istenen sorgu ve analizlerin daha etkin bir şekilde yapılabileceği bir model geliştirmeyi planlamaktadır. Firmanın bu istekleri doğrultusunda sizden bir yazılım geliştirmeniz talep edilmektedir.

Kişilerin Rezervasyon Bilgilerini Tutan Dosyadaki Kayıtların Formatı

Rezervasyon kayıtları “rezervasyon.txt” dosyası içinde aşağıdaki formatta tutulmaktadır:

<kullanıcı id><,><yer id><,><zaman><,><enlem><,><boylam><,><şehir><,><kategori:altkategori>

Örnek kayıt:

fred-wilson,giorgione-new-york,2010-10-24T00:22:49,40.726144,-74.008348,New York,Food:Italian

2. Modelin Oluşturulması

Tasarlanan sistemde “rezervasyon.txt” dosyasından çekilen kayıt bilgileriyle meydana getirilecek ağaç yapısı 3 aşamada oluşturulmalıdır:

1) İlk aşamada ağacın üst kısmında, “rezervasyon” kök düğümünden başlayarak aşağı doğru kategoriler ve alt kategoriler için kayıtlardaki hiyerarşik sıraya göre alt düğümler oluşturulmalıdır. Her kategori düğüm yapısı aşağıdaki bilgileri tutmalıdır:

- **Kategori ismi:** Kategorinin “rezervasyon.txt” dosyasındaki ismi
- **Alt kategori sayısı:** Kendinden hemen sonra gelen çocuk düğümlerin sayısını gösterir

- **Kullanıcı sayısı:** Kategorinin altında bulunan toplam kullanıcı sayısını gösterir
- **Rezervasyon sayısı:** Kategorinin altında bulunan toplam rezervasyon sayısını gösterir
- **Kategori yolu:** Kategori düğümünün rezervasyon.txt dosyasındaki gösterimi
- **Derinlik:** Kategorinin ağacın kök düğümünden itibaren kaçınıcı dereceden çocuk düğümü olduğunu gösterir.

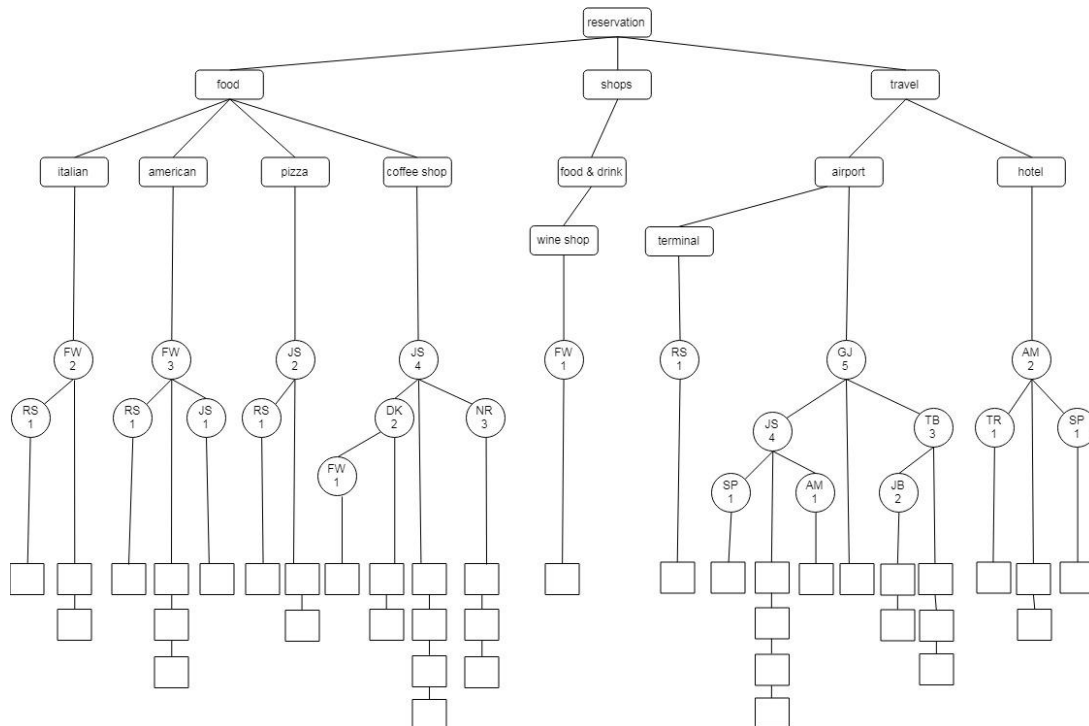
2) İkinci aşamada ağacın altında bulunan kategori düğümlerinden sonra rezervasyon işlemini yapan müşteri bilgilerinin yer aldığı kullanıcı düğümleri **Max-Heap Algoritmasına** göre oluşturulacak ayrı bir ağaca eklenmelidir. Bu ağacın kök düğümü ise ilgili kategorinin bir çocuk düğümü olacak şekilde bağlanmalıdır. Her bir kullanıcı ağaçlarını oluştururken kullanıcıların o kategorideki toplam rezervasyon sayıları dikkate alınmalıdır. Kullanıcı düğümleri aşağıdaki bilgileri içermelidir:

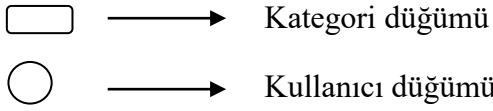
- **Kullanıcı id :** “rezervasyon.txt” dosyasından çekilen “kullanıcı id” bilgisi
- **Kategori ismi:** “rezervasyon.txt” dosyasından çekilen en alt kategori bilgisi
- **Rezervasyon sayısı:** Bir kullanıcının o kategoriye ait toplam rezervasyon sayısı

3) Modelin oluşturulmasındaki bu son aşamada her bir kullanıcı düğümüne kullanıcıların o kategorideki rezervasyonları, bağlı liste yapısı kullanılarak eklenmelidir. Bağlı listelerdeki her bir düğüm aşağıdaki bilgileri içermelidir:

- **Yer id:** “rezervasyon.txt” dosyasından çekilen “yer id” bilgisi
- **Rezervasyon zamanı:** “rezervasyon.txt” dosyasından çekilen “zaman” bilgisi
- **Enlem:** “rezervasyon.txt” dosyasından çekilen “enlem” bilgisi
- **Boylam:** “rezervasyon.txt” dosyasından çekilen “boylam” bilgisi
- **Şehir:** “rezervasyon.txt” dosyasından çekilen “şehir” bilgisi

Örnek Ağaç Yapısı:





Not: Tasarlanan bu modelde tüm bağlantılar çift yönlü olarak düşünülmelidir.

3. İsterler

1) Kategori ile ilgili aşağıdaki işlemler ağaç yapısı üzerinde gerçekleştirilmelidir:

- **Kategori bulma:** Kategorinin ismi ile ağaç üzerinde arama yapılarak kategori bilgileri ekrana basılmalı sonrasında ise kullanıcıya bu kategorinin altına yeni bir kategori ekleme ya da mevcut kategoriyi silme işlemleri için seçenek sunulmalıdır.

- **Yeni kategori ekleme:** Ekleme işlemi yapmadan önce ekleme işlemin yapılacağı kategorinin altında aynı isimle bir kategorinin olup olmadığı sorgulanmalıdır. Eğer aynı isme sahip bir kategori varsa kullanıcıya uyarı mesajı verilerek işlem iptal edilmelidir. Eğer yoksa altına eklenmesi istenen kategori bulunarak onun bir alt kategorisi olacak şekilde yeni kategori eklenmelidir.

- **Kategori silme:** Kategori silme işlemi sırasında silinecek kategori düğümüne doğrudan bağlı kullanıcı düğümleri varsa onlar da kategoriyle beraber silinecektir. Ayrıca silinecek kategorinin altında bulunan kategoriler silinmemeli bunun yerine silinecek kategorinin bir üst düğümüne bağlanmalıdır.

Not: Kategori silme ve ekleme işlemleri için ilgili kategorilerin bilgilerinin güncellenmesi unutulmamalıdır.

2) Kullanıcı ile ilgili işlemler:

- **Kullanıcı ekleme:** Belirlenen bir kategoriye yeni bir kullanıcı eklenmelidir. Ekleme işlemi yapmadan önce aynı kullanıcı id'ye sahip bir kullanıcının olup olmadığı sorgulanmalıdır. Aynı kullanıcı id'ye sahip bir kullanıcı var ise işlem iptal edilmelidir. Aksi durumda yeni kullanıcı ilgili kategori altındaki Max-Heap ile oluşturulmuş kullanıcı ağacına eklenmelidir.

- **Kullanıcı silme:** 3 durum söz konusudur. 1. Durumda belli bir kategorinin altında yer alan tüm kullanıcılar silinmelidir. 2. durumda ise belli bir kategori altındaki belli bir kullanıcı bulunarak silinmelidir. 3. durumda ise belli bir kullanıcının tüm kategorilerdeki kullanıcı düğümleri silinmelidir. Bu üç seçenek kullanıcıya sunulmalıdır.

3) Sorgu ve listeleme işlemleri:

- **Kullanıcıya göre kategori listeleme:** Kullanıcının yapmış olduğu tüm rezervasyonların ait olduğu kategoriler tekrarsız olarak listelenmelidir (en alt kategori).

- **Kategoriye göre kullanıcı listeleme:** Belli bir kategorideki tüm kullanıcıların bilgileri listelenmelidir (kullanıcı id).

Notlara Bak

- **Rezervasyon yerine göre kullanıcı listeleme:** Aynı yerde rezervasyon yaptırmış olan tüm kullanıcılar listelenmelidir (kullanıcı id).

- **Kullanıcıya göre rezervasyon listeleme:** Belli bir kullanıcıya ait tüm kategorideki rezervasyon yerleri tekrarsız olarak listelenmelidir (yer id, enlem, boylam, şehir, son rezervasyon zamanı).

3. Notlar

- Proje C, C# veya Java programlama dillerinden biri kullanılarak yapılmalıdır.
- Proje ile ilgili bilgilendirme toplantısına katılımınız ve anlaşılmayan noktaları mutlaka sormanız, sunumda problem yaşamamanız için çok önemlidir.

4. Ödev Teslimi

- Proje sunum gününde raporu (hard copy) teslim etmeniz gerekmektedir. Rapor IEEE formatında (önceki yıllarda verilen formatta) 4 sayfa, akış diagramı veya yalancı kod içeren özet, giriş, yöntem, sonuç ve kaynakça bölümünden oluşmalıdır.
- Dersin takibi projenin teslimi dahil edestek.kocaeli.edu.tr sistemi üzerinden yapılacaktır. edestek sisteminde belirtilen tarihten sonra getirilen projeler kabul edilmeyecektir.
- Proje ile ilgili sorular edestek sistemindeki forum üzerinden Arş. Gör. Abdurrahman Gün veya Arş. Gör. Ekin Ekinci'ye sorulabilir.
- Demo tarihleri daha sonra duyurulacaktır.
- Demo sırasında algoritma, geliştirdiğiniz kodun çeşitli kısımlarının ne amaçla yazıldığı ve geliştirme ortamı hakkında sorular sorulabilir.
- Kullandığınız herhangi bir satır kodu açıklamanız istenebilir.