

İK YÖNETİM SİSTEMİ

Mert MAHANOĞLU

Bilişim Sistemleri Mühendisliği Bölümü

Kocaeli Üniversitesi

mahanoglumert73@gmail.com

Özet

İnsan kaynakları bilgi sistemi kuramında İkili Arama Ağacı (Binary Search Tree) ve Bağlı Liste (LinkedList) yapılarının koordinasyonu sağlanarak geliştirilen projenin; sisteme kayıt, bilgi görüntüleme, istenilen bilgileri silme ve sistemde kayıtlı olan bilgileri güncelleme işlemlerini belirli filtre kategorileri ile süsleyip tasarlanması gerçekleştirilmiştir. Bağlı liste yapısının sınıf ve metotları tasarlanırken hazır bağlı liste sınıfından destek alınmamıştır. İkili arama ağacı ve bağlı listenin devinimi senkronize edilmiş olup akıcılık esas alınmıştır. Ayrıca kaydedilen güncel verilerin dosyaya yazdırılıp anlaşılır satırlar aralığında tutulması sağlanmıştır. Projenin sonucu olarak beklenen çıktı, dinamik çalışabilecek bir insan kaynakları yönetim sistemi elde etmek olmuştur. Bu bağlamda da ana amaç, proje kriterleri dikkate alınarak programın olabildiğince sorunsuz çalışması olarak belirlenmiştir.

Projenin niteliği, kariyer ve iş başvuru erişilebilirliği için hazırlanmış bir prototip olarak sunulmuştur. Kişinin, kendi bilgilerinin yanı sıra platformda bulunan diğer bireylerin bilgilerini de görüntüleyerek kolektif bir ortam amaçlanmıştır. Projenin tasarlanması ve geliştirilmesi sırasında kullanılan stüdyo Netbeans olup program Java dilinde yazılmıştır. Projenin tasarım ve geliştirme sürecinde faydalanan kaynaklardan herhangi bir hazır kod kullanılmamıştır. Projenin mimarisinde esinlenen kaynaklar Kaynakça bölümünde numaralandırılarak belirtilmiştir.

1. Giriş

İnsan kaynakları bilgi sistemi kapsamında, iş başvurusu yapabilmek yetkinliği ile başvuru yapan kişilere ait kişisel veya genel bilgilerin sistemde kaydedilip üzerinde değişiklikler yapılarak güncel, taze verinin tutulması. Herhangi bir veri tabanı bağlantısına ihtiyaç duyulmadan verilerin dosya bazlı tutulup korunması. Ayrıca bu prosedür izlenirken kullanılan ikili arama ağacı ve bağlı liste yapılarının özgün ve elverişli olması gerekmektedir. Birbiriyle senkron çalışıp kişinin girmiş olduğu verileri doğru yörengede taşıyıp istenilen formatta kullanıcıya görüntülenmesi talep edilmektedir.

2. Temel Bilgiler

Projeyi gerçekleştiren kullanılan teknolojiler ve sistemler ;
İşletim Sistemleri : Microsoft Windows , Mac OS X
Stüdyo: VSCode , Netbeans , Draw.io(Akış Şeması için)
Program dili : Java
İletişim : Google Meet

3. Geliştirilen Mimari

3.1 Kullanılan sınıflar ve metotları

3.1.1 Node2 Sınıfı

Yazılmış olan bu sınıf programın temelini oluşturmaktadır. Program boyunca geçici olarak bellekte tutulacak bilgiler bu sınıf içerisindeki değişken ve yapıcı metotlar sayesinde tutulmaktadır. Raporun ilerleyen kısımlarında bahsedilecek olan ikili arama ağacı ve bağlı liste yapısı ,bu sınıftan üretilen nesneler ile oluşacaktır. Sınıfın ana görevi ise kendisine diğer sınıflar üzerinden yapıcı metodun aracılığıyla gelen bilgileri, kendi içerisinde bulunan değişkenlere eşitlemek ve böylece veri kaybını önlemektir.

3.1.2 Tree Sınıfı

Proje kriterlerinde belirtilmiş olan kişiye ilişkin (ad,adres,telefon,e-posta,doğum tarihi,yabancı dil,ehliyet) bilgilerini, daha önce çalıştığı işyerlerinin bilgilerini (adı, adresi, pozisyon veya görevi ,süresi) ve Eğitim Durumu yani mezun olduğu okulların bilgilerini (okul adı, bölümü, başlangıç ve bitiş tarihleri yıl olarak, not ortalaması) bir ikili ağaç yapısında tutma işlevi görmektedir. Gerçekleşen bu tutma işlevinin gerçekleşmesi sırasında yukarıda belirtilmiş olan **Node2** sınıfı üzerinden nesneler üretilmiş ve bu nesneler aracılığı ile gelen her nesnenin kişi adı karşılaştırılmıştır. Gelen kişi adının alfabeedeki yerine göre, ağaçtaki kişi adından daha önce olması durumunda ağacın dal diye ifade edilen sol dalına eklenmiştir. Tam tersi durumda ise ağacın sağdaki dalına eklenmiştir ve bu durum yinelemeli(recursive) bir yapı içerisinde gerçekleşmiştir. Ayrıca bu sınıf içerisinde proje kriterlerinde belirtilmiş olan ağacın sıralama yöntemleri (inorder,preorder,postprder) yine yinelemeli(recursive) metotlar aracılığıyla işleme alınmıştır. Ağacın derinliği diye ifade edilen dallanma sayısını bulmak ve ayrıca ağaçtaki eleman sayısını bulmak amacıyla sınıf içerisinde metotlar bulundurmaktadır.

Metotlardan ilki olan Ekle() metodu yukarıda bilgileri verilmiş olan "Tree" sınıfının içerisinde bulunmaktadır. Metot içerisine kişiye ilişkin bilgiler, eğitim ve işyeri bilgilerini, her bilgi için bir değişken olmak üzere parametre olarak istemektedir. Bu bilgilerin ikili arama ağacına eklenerek tutulması istenmektedir. Çalışan bu tutma işlevinin gerçekleşmesi sırasında yukarıda belirtilmiş olan **Node2** sınıfı üzerinden nesneler üretilmiş ve bu nesneler aracılığı ile gelen her nesnenin kişi adı karşılaştırılmıştır. Gelen kişi adının alfabeedeki yerine göre, ağaçtaki kişi adından daha önce olması durumunda ağacın dal diye ifade edilen sol dalına eklenmiştir. Tam tersi durumda ise ağacın sağdaki dalına eklenmiştir ve bu durum yinelemeli(recursive) bir yapı içerisinde gerçekleşmiştir.