ASANSÖRLERDEKİ TALEP YOĞUNLUĞUNUN MULTITHREAD İLE KONTROLÜ PROJESİ

Osman Şimşek
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
180202048
simsekosman15@hotmail.com

Yener Emin Elibol
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
180202054
yenereminelibol@gmail.com

Asansörlerdeki Talep Yoğunluğunun Multithread ile kontrolü Projesi, bir AVM nin asansöründeki yoğunluğu multithread kullanarak diğer asansörlerle birlikte azaltmaktır.

GİRİŞ

Asansör Yoğunluk Projesi, paralel çalışması gereken bir takım işlemlerin programlama üzerinde verilen konsept üzerinde uygulanmasını amaçlamaktadır. Projede ki biz de istenen problem sürekli olarak AVM ye giriş yapan kişilerin ve AVM'den çıkmak isteyen kişilerin her zaman asansör kapasitesinin üstüne çıkması ve bu kapasite aşımını diğer asansörleri aktif/pasif hale getirerek dengelemektir. Diğer bir problem ise paralel olarak AVM'ye bir kişinin girmesi, AVM'de bulunan bir kişinin AVM'den çıkmak istemesi ile oluşan yoğunluğun paralel bir şekilde çalışacak algoritmalar üzerinde oturaklı bir algoritma yazılmasıdır. Yazılmış olan algoritmalar paralel olarak çalıştığı için aynı değişken üzerinde birden fazla işlem yapılmaktadır ve bu da katlarda veya asansörlerde bulunan kişi sayıları üzerinde anomaliler oluşturmaktadır ve son olarak karşılaşılan problem ise Kullanıcı Ara yüzü üzerinde güvenli bir şekilde değişiklik yapmaktır.

I. TEMEL BİLGİLER

Program C# programlama dilinde gerçekleştirilmiş olup Windows Form projesi olarak oluşturulmuştur. Geliştirme ortamı olarak Visual Studio kullanılmıştır.

II. TASARIM

A. Kullanıcı Arayüzü

Program çalıştırıldığı anda form elemanları oluşturulur ve program arayüzü temel olarak iki parçadan oluşur. Bu Parçalar;

- Kullanıcı arayüzündeki ilk kısım olarak her bir kata ait kuyruk bilgileri ve bu kuyrukta yer alan müşterilere ait bilgiler bulunmaktadır. İlk kattan farklı olarak diğer katlarda o an da bulunan müşteri sayısını da göstermektedir. Kullanıcı arayüzündeki ilk kısımdan birisiide diğer katlardan zemin kata gelip AVM den çıkış yapan müşterilerin sayısını göstermktedir.
- Kullanıcı arayüzünde yer alan ikinci parça ise AVM de bulunan her bir asansöre ait bilgiler yer almaktadır. Bu bilgiler asansörün aktif olup olmadığı, kaçıncı katta olduğu, çalışma modunun ne şekilde olduğu, kapasitesi, içinde bulunan müşteri sayisi ve bu müşterilerin gösterilmesi gibi çeşitli kontroller ve değerler bulunmaktadır.

Yazılım Mimari

Program, Kat, Grup ve Queueu yapılarından oluşmaktadır. Programda 3 farklı thread görev almaktadır. Bunlar girisThread, cikisThread ve asansorThread'dir.

- Public Class Kat:
  Kat bilgilerini tutmak için oluşturulmuştur. İçerisinde kat'a gelen kişilerin tutulduğu gelenKisiler isimli kuyruk yapısı, kattan zemin kata gitmek isteyenler için gidenKisler isimli kuyruk yapısı, kat numarası, katta bulunan kişi sayısı ve Form ekranında bilgileri bastırmak için yer alan birkaç fonksiyondan oluşmaktadır.

- Public Class Grup:
  AVM ye gelen müşterilerin bilgilerinin tutulması sağlanır. İçerisinde kişi sayisi ve kat sayisi bilgilerini tutan iki değişken yer almaktadır. girisThread'i her çalıştığında aslında bir grup sınıfından oluşturulmuş müşteri oluşturur.

- Public void MusteriEkle:
  Giriş katına her 500ms de 1 ile 10 kişi arasında kişi alır ve bu kişileri giriş katında kuyruğa sokar.

- Public void MusteriCikart:
  Her 1000ms de herhangi bir kattan 1 ile 5 kişi arasında kişiyi AVM den çıkması için bulunduğu kattan(1 ve 4. Katlar arasında olmak zorunda) kuyruğa sokar.

- Public void AsansorHareket:
  Asansörlerin çalışma mantığını içinde barındıran metoddur. Öncelikle asansörün aktif olup olmadığını kontrol eder. Asansör aktif ise bulunduğu katta inmesi gereken insan olup olmadığını kontrol eder ve inmesi gereken kişileri asansörden indirir. Asansörden indirme işlemi bittikten sonra asansöre alma işlemi yapılır bu noktada asansörün kapasitesi ve asansörün içinde aktif olarak bulunan kişi sayısına göre asansöre binme işlemi yapılır ve asansöre binecek kişiler o katta kuyrukta bekleyen kişilerden seçilir. Asansöre binme işlemi bittikten sonra asansörün ulaşması gereken hedef nokta tekrardan belirlenir ve belirlenen hedefe göre asansörün gittiği yön belirlenir ve bir sonraki kata geçmesi için 200 ms beklenir.

B. Proje İşleyişi

Projede 3 farklı thread görev almaktadır. Bunlardan ilki olan girisThreadi 500ms aralıklarla 1 ile 10 arasında rastgele sayıda kullanıcıyı 1 ile 4 arasında herhangi bir kata gitmesi için kuyruğa alır. Kuyruğa alınan insanları yukarılara çıkartmak ve yukarıdan inmek için talep eden insanları zemin kata indirmek için görev alan asansorThread'i bulunmaktadır. asansorThread'i her katta 200 ms saniye durucak şekilde çalışmaktadır. Projemizde 1 tane asansor çalışmaktadır. Asansor giriş kattan en fazla 10 müşteri alıcak şekilde müşteri alır ve müşteri gitmek istediği katlara götürür. Aradından istedikleri katlara giden müşteriler cikisThreadi ile katlarda bulunan müşteriler inmek için çıkış kuyruğuna aktarılırlar. cıkışThread'i 1000 ms aralıklarla 1 ile 5 müşteri arasında müşteriyi çıkış kuyruğuna alır. Asansor müşterileri talep ettikleri katlara bıraktıktan sonra çıkış kuyruklarını kontrol eder ve inmek isteyen müşterileri gider ve alır. En fazla 10 müşteri olucak şekilde çıkış kuyruklarından müşteri alabilir. Aldığı müşterileri zemin kata götürür ve yeni müşteriler biner.

C. Kazanımlar

- C# da Thread classının ve içinde ki methodlarını kullanmayı öğrendik..
- C# da Kuyruk yapısının kullanımını öğrendik
- C# da Thread ve Multithread kullanımı öğrendik
- Asenkron, senkron farkını öğrendik.

KAYNAKÇA

[1] https://stackoverflow.com/

[2] https://www.geeksforgeeks.org/c-sharp-interface/

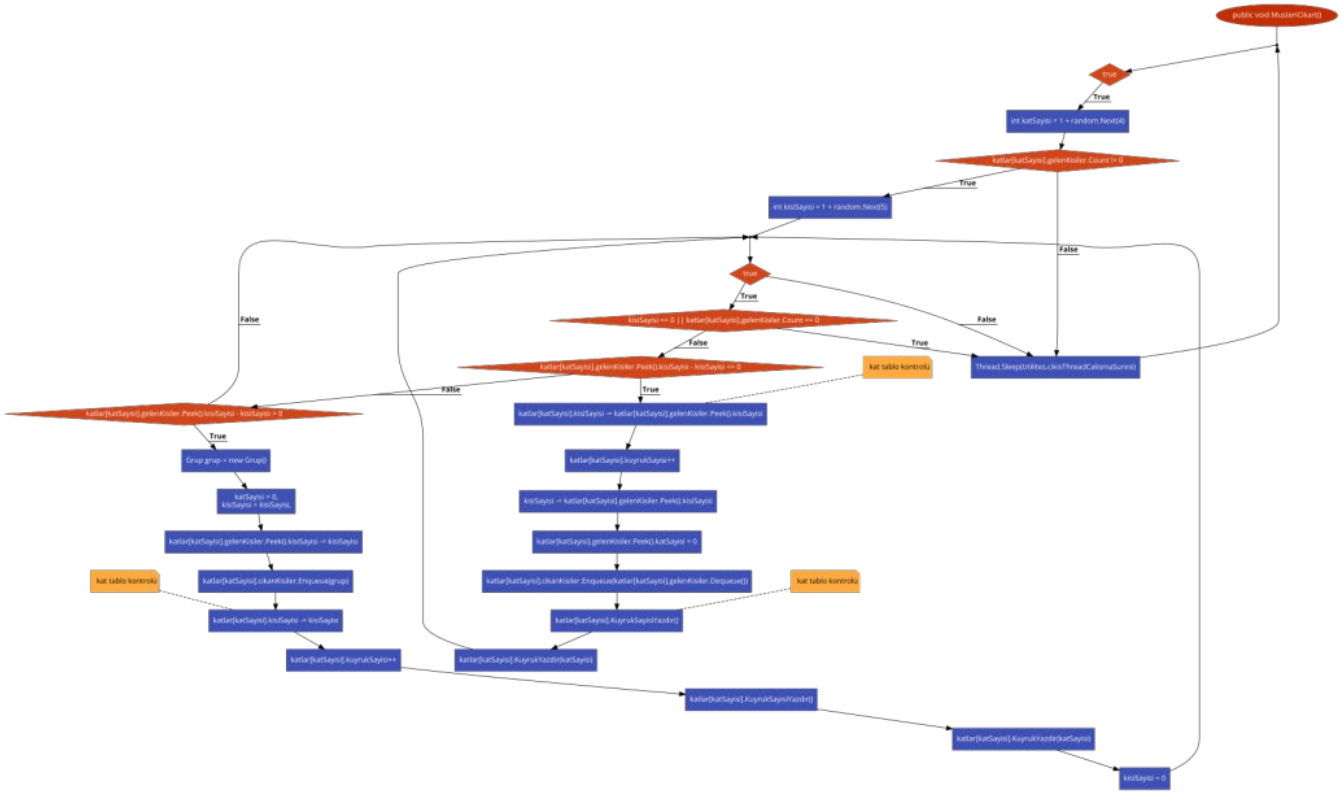
[3] https://docs.microsoft.com/tr-tr/dotnet/api/system.threading.thread?view=net-5.0

[4] https://www.geeksforgeeks.org/c-sharp-multithreading/

[5] https://docs.microsoft.com/tr-tr/dotnet/csharp/language-reference/builtin-types/struct

[6] https://docs.microsoft.com/en-us/dotnet/desktop/winforms/controls/how-to-make-thread-safe-calls-to-windows-forms-controls?view=netframeworkdesktop-4.8

D. Akış Diyagramı



MusteriCikart fonksiyonuna ait akış diagramı.

E. Uml Diagramı