ASANSÖRLERDEKİ TALEP YOĞUNLUĞUNUN MULTITHREAD İLE KONTROLÜ

Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi

Faruk Arığ
180202009

Sefa Öztürk
180202036

Özet

Bu projeyi geliştirirken java dilini ve özelliklerini kullandım. Projede bir sunucuya gelen isteklerdeki aşırı yoğunluğu, multithread kullanarak asansörlerle birlikte azaltmaktır. Asansör kapasitelerinin 10 üzerine çıktığı anda iki katına çıktığında yeni yaratılan threadin kapasitesi artmakta ve o kapasite bitene kadar thread çalışmaya devam etmektedir ve bütün thread kapasitelerini canlı olarak görebilmekteyiz.

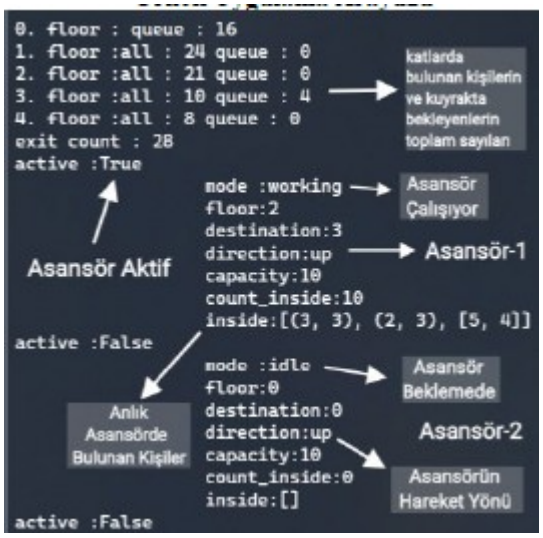
1 Problem Tanımı

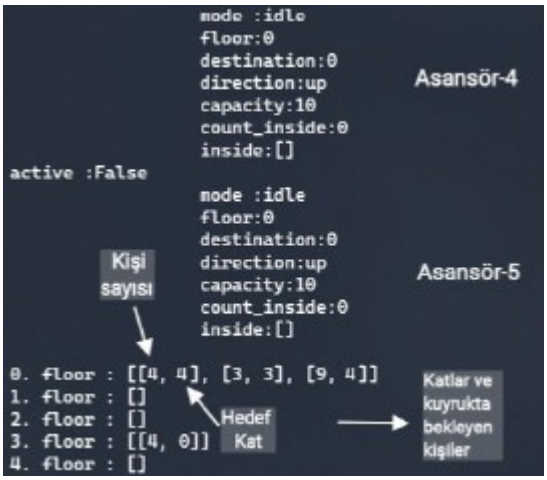
Main thread kapasitemiz 10 ve 500 ms zaman aralıklarıyla [1-10] arasında rastgele sayıda istek kabul etmektedir. İstek olduğu sürece 1000 ms zaman aralıklarıyla [1-5] arasında rastgele sayıda isteğe geri dönüş yapmaktadır. Child thread kapasitemiz ise 10 dur. 500 ms zaman aralıklarıyla [1-10] arasında rastgele sayıda ana asansörden istek almaktadır. İstek olduğu sürece 10 ms zaman aralıklarıyla [1-10] arasında rastgele sayıda isteğe geri dönüş yapmaktadır.
Child threadler 10 olduğunda ise yeni threade kapasiteleri 2 katına çıktığındaalt child thread e aktarmaktadır.
Multithread kullanarak bütün threadlerin istek alıp geri dönüşünü sağlamak. Thread kapasitelerini canlı göstermek.

2.1 Tasarım

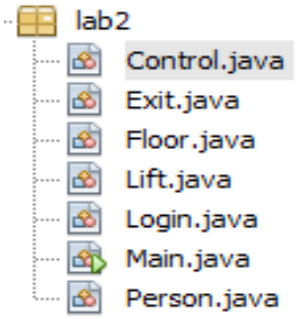
Proje de katların yazdığı ve sırasıyla asansörlein devamlı çalışan mainthread ve liftler yer almaktadır. Açıklamarında ise yeni gelen kişiler yani kuyruklar ve ihtiyaç duyulan threadler görüntülenmektedir. Bunların yanında ise kapasite leri, çalışma durumu, hareketyönü, aktflik durumu, asansör içerisindeki insan sayısı gibi bilgiler yer almaktadır.

2.2 Yazılım Mimarisi





Yukarıda görünen mimari yapıyı kod yapısında geliştirmek için kullandık. Bu yapıyı şu şekilde koda döktük.



Control classı ise yapılan her işlem ve değişkenlerin yazdırıldığı yapı olarak yazdık.. Lift,login ve exit class ları da her class ta yapılacak olan işlemleri içermektedir.

2.3 Örnek Proje Çıktıları

```
0. Floor : 0        queue : 10
1. Floor : 7        queue : 1
2. Floor : 8        queue : 0
3. Floor : 3        queue : 0
4. Floor : 5        queue : 3
```

Örnek 1. Katlar ve kuyrukta bekleyenler

```
active : true
                    mode : working
                    floor : 2
                    destination : 0
                    direction : up
                    capacity : 10
                    count_inside : 4
                    inside : [[1 0][1 3][2 4]]
```

Örnek 2. Herhangi bir asansörün bilgisi

```
0. Floor : [[4 1][1 2][3 3][2 4]]
1. Floor : [[1 0]]
2. Floor : []
3. Floor : []
4. Floor : [[3 0]]
```

Örnek 3. Exit Threadinden sonraki durumlar

3 Genel Yapı

İlk önce main classını kurdum ve her 200 de bir istek almasını bu isteğe de her 1000 ms de bir kez geri dönüş yapmasını sağladım.

Main, control ve login classlarında ise her 500ms de bir kez ana asansörden istek almakta ve her 200 de bir kez ana asansöre geri dönüş sağlamaktadır. Eğer alt asansörler kapasitelerinden herhangi biri 10 u aşarsa capasity fonksiyonu ile kapasite kontörü yapılıp yeni bir lifts add fonksiyonu ile yaratılmaktadır. 10 kapasiteyi aşan sub thread kapasitesinin yarısı lifts kapasiteye aktarılarak lift threadi her 500 ms de bir anasunucudan istek alarak her 200ms de bir kez isteğe geri dönüş yapmaktadır..Uygulamada bulunan her thread in kapasitesi canlı takip edilmekte ve console output ile gösterilmek.

4 Yapılan Araştırmalar

Bu proje en çok hata aldığımız ve hatanın ne olduğunu dahi anlayamadığımız projelerden biri oldu.

Threadlerin birbiriyle çakışması,senkronize olmama durumları,deadlock,projenin çok yavaş çalışması gibi birçok problemle karşı karşıya kaldık.

Öncelikle ana sunucu ve alt sunucuları oluşturduk ve çalıştırdık.Doğru çalıştırıyordu ancak 500 ms aralıklarla gelmesi gereken sunucu istekleri düzenli olmamakla birlikte çok daha fazla aralıklarla geliyor 5-10 saniye beklediğimiz oluyordu.Bunun senkronize olmadığından kaynaklı olduğunu düşündük ve classlarımızı sleep ile çağırdık

4.1 Karşılaşılan Sorunlar ve Çözümleri

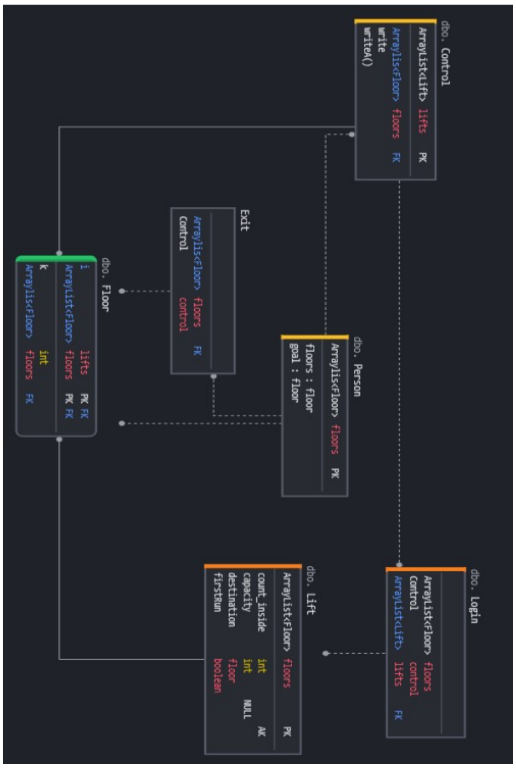
İllegalmonitorstateexception threadleri listeden stop işlemi ile durduğumuzda bu hata ile kaşılaştık hatanın sebebi stop işlemini yanlış yerde yapmamızdı.

Oluşturduğumuz sunucuları yönetme konusunda problem yaşadık bu problemi arraylistlerden yararladık.Bu şekilde asansörlerimizi yönetimi kolaylaştı.

4.2 Kazanımlar

Sunucu İstek Yoğunluğunun Multithread İle Kontrolü Projesinin bize kantıkları;
* Threadler yardımıyla birden fazla işin bir arada nasıl yapılacağını öğrendik.
* Birden fazla threadin nasıl yönetileceğini ve aynı alana erişimi sırasında karşılaşılan eş zamanlılık hatalarının nasıl önüne geçiceleceği öğrendik
*Birden fazla classı Threadten extends ederek oop mantığındaki eksiklerimizi tamamladık.
* Bilgisayarın birden fazla işi nasıl yaptığı hakkında daha fazla fikir sahibi olduk

4.2 Diyagram



5 Referanslar

[1] https://emrahmete.wordpress.com/2011/10/06/javada-thread-yapisi-ve-kullanimi-hakkinda-ipuclari/

[2] https://www.generacodice.com/es/articolo/693507/Understand-the-concept-of-MultiThreading-in-Javahttps://stackoverflow.com/questions/29987378/java-multithreading-difference-in-child-thread-running-time

[3] https://stackoverflow.com/questions/13786083/create-two-threads-one-display-odd-other-even-numbers

[4] https://www.subjectcoach.com/tutorials/detail/contents/quick-walk-through-the-advanced-concepts-in-java-part-3-of-series/chapter/multithreading

[5] https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example

[6] https://www.logicbig.com/tutorials/core-java-tutorial/java-multi-threading/thread-wait-notify.html

[7] https://emrahmete.wordpress.com/2011/10/06/javada-thread-yapisi-ve-kullanimi-hakkinda-ipuclari/

[8] https://medium.com/gokhanyavas/javada-multithreading-bbc6a9181772

[9] http://www.csharpnedir.com/articles/read/?id=593

[10] https://coderanch.com/t/475739/certification/Thread-java-lang-IllegalMonitorStateException-Iam

[11] https://stackoverflow.com/questions/33642758/how-to-create-dynamic-threads-in-java

[12] https://ramazanbiyikci.com.tr/java-thread-ayni-anda-coklu-islem/