|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ | DOOB Nedir | TARİH : 10.06.2019 |
| HAVELSAN 30 yıldır Komuta Kontrol, Simülasyon, Eğitim ve Bilgi Sistemlerini Türk Silahlı Kuvvetleri ve tüm dünyada kullanılmak üzere geliştirmektedir. DOOB ürün ailesi, HAVELSAN’ın engin saha ve mühendislik tecrübesinin sonucunda geliştirilmiş ürünlerdir. DOOB ürün ailesi, Komuta ve Kontrol Bilgi Sistemlerinin temel niteliklerini içeren ve stratejik, operasyonel ve taktik komuta seviyelerinde kullanılmak üzere tasarlanmış üç farklı uygulamadan oluşur. DOOB uygulamalarının temel amacı tüm komuta seviyelerindeki durumsal farkındalığı arttırmaktır.  DOOB uygulaması farklı uluslararası ve milli standartlar vasıtasıyla farklı kaynaklardan toplandığı bilgileri entegre edip, Cari Harekât Takibi Modülü aracılığıyla bu bilgileri birleştirilmiş bir arayüzle sunarak durumsal farkındalığı arttırır. Bunun yanında, farklı fonksiyonel alanlarda hiyerarşik ve eş zamanlı planlama ve bu planların icra safhalarının farklı fonksiyon alanlarına uygun olarak takibini de destekler. DOOB milli ve NATO ihtiyaçlarını desteklediği gibi özel gereksinimlere kolayca uyarlanabilir ve genişletilebilir. DOOB ürün ailesi, HAVELSAN tarafından geliştirilmiş bir CBS çözümü olan TMAP ile desteklenmektedir.  DOOB web tabanlı bir uygulamadır ve modern web tarayıcısı haricinde, istemci makinesine herhangi bir kurulum gerektirmeden çalıştırılabilir. Kullanım uzmanları, DOOB ürünlerinin tasarım ve geliştirilmesi safhalarında yer aldıkları için her üç ürünün de kullanımı modern web uygulamalarına benzer, kolay ve hızlıdır. DOOB modüler bir uygulamadır ve farklı modülleri ayrı olarak kurulabilir. Bu nedenle kurulumu, bakımı ve yönetimi kolaydır.  DOOB Tactical uygulaması komuta yerlerinde taktik seviye komutanlar tarafından kullanılmak üzere geliştirilmiştir. DOOB Tactical ile taktik kullanıcılar DOOB tarafından üretilen planların basit halini alabilirler. Bununla birlikte kendi planlarını yapabilir ve cari harekatı takip edebilirler. Taktik kullanıcılar semboloji standartlarına uygun olarak genel durumsal farkındalığa bilgi girişi yaparak katkı sağlayabilir ve bu bilgiyi DOOB ve DOOB Mobile ile paylaşabilirler. DOOB Tactical, DOOB Mobile ile entegre olduğu için kendi birliğine ait DOOB Mobile kullanıcılarını izleyebilir.  SmartOps olarak da bilinen DOOB Mobile uygulaması, erler için geliştirilmiş ve bu seviyedeki komuta kontrol faaliyetlerini gerçekleştirmeyi hedefleyen taktik seviye bir uygulamadır. Android işletim sistemi tabanlı mobil uygulamadır. Güvenli VPN üzerinden temel olarak konum ve kalp atışı bilgilerini paylaşır, kendi takımındaki diğer erlerin aynı şekilde durumlarını takip eder ve bağlı olduğu DOOB Tactical komutanının konumunu görebilir. Bunun yanında canlı video yayımı, mesajlaşma, ses, fotoğraf ve diğer doküman paylaşım özellikleri de mevcuttur.  DOOB ürün ailesi kendi arasında entegre bir şekilde çalışır. Bu yüzden DOOB Tactical ve DOOB Mobile tarafından yayınlanan bütün bilgiler DOOB tarafından izlenebilir.  DOOB web tabanlı ve dağıtık mimariye sahip bir uygulama olduğu için tek bir sunucu üzerinden birçok kullanıcıya destek verebileceği gibi dağıtık mimari sayesinde bütün bir orduyu destekleyebilir. Bu şekilde kolay bir şekilde kurulup, kolay bir şekilde yönetilebilir ve genişletilebilir. DOOB ürününün geliştirilmesinde modern ve açık kaynak kodlu web ve uygulama çatıları kullanılmış ve herhangi bir şekilde rahat veya paralı bir kütüphane kullanılmamıştır. DOOB kullanıcının aşina olduğu diğer web uygulamalarına yakın bir kullanım kolaylığı sağlar. | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ | DOOB Tactical Projesi | TARİH : 11.06.2019 |
| DOOB Tactical uygulaması komuta yerlerinde taktik seviye komutanlar tarafından kullanılmak üzere geliştirilmiştir. DOOB Tactical ile taktik kullanıcılar DOOB tarafından üretilen planların basit halini alabilirler. Bununla birlikte kendi planlarını yapabilir ve cari harekatı takip edebilirler. Taktik kullanıcılar semboloji standartlarına uygun olarak genel durumsal farkındalığa bilgi girişi yaparak katkı sağlayabilir ve bu bilgiyi DOOB ve DOOB Mobile ile paylaşabilirler. DOOB Tactical, DOOB Mobile ile entegre olduğu için kendi birliğine ait DOOB Mobile kullanıcılarını izleyebilir.  Umman ülkesine verilmek için geliştirilen Tactical projesine dahil oldum. Askeri birliklerin ve tablet üzerinde kurulan yazılım ile birbirlerini takip etmesi ve düşman birimlerin yazılımda haritada işaretlenerek birimlerin hareket kabiliyetlerini iyileştirmektedir.  Proje hakkında yeterli sözel bilgiyi edinip gün sonuna doğru kod incelemesine başladım. | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ | DOOB Tactical Projesi | TARİH : 12.06.2019 |
| Tüm gün kod incelemesine devam ettim. | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ | DOOB Tactical Projesi | TARİH : 13.06.2019 |
| Maven, ilk versiyonu 30 Mart 2002’de çıkan proje bağımlılık yönetim aracıdır. Hatta direk kendisine “Proje Yöneticisi” denilebilir. Ant’a bir alternatif olarak gösterilen aynı Ant gibi bir JDT(Java Development Tool) ‘dur. Bağımlılık yönetiminin yanında modüler bazda proje geliştirme imkanı, plugin desteğiyle proje yönetimini daha güçlü kılma ve benzeri imkanlar da sunar.  Maven, proje geliştirirken proje içerisinde bir standart oluşturmamızı, geliştirme sürecini basitleştirmemizi, dokümantasyonumuzu etkili bir şekilde oluşturmamızı, projemizdeki kütüphane bağımlılığını ve IDE bağımlılığını ortadan kaldırmamızı sağlayan bir araçtır. Bizi internette sayfa sayfa dolaşıp ihtiyacımız olan library(kütüphane)’nin ihtiyacımız olan versiyonunu arama zahmetinden kurtaran bir çatıdır. Yani projeniz örneğin log4j.jar ve Spring.jar gibi kütüphanelere ihtiyaç duyuyorsa; bu ihtiyacı ilgili jar dosyalarını bularak ilgili yerlere kopyalamak mavenin sorumluluğundadır. Pom.xml (maven in proje ayar dosyası) içerisinde log4j nin şu versiyonu ve Spring’ın şu versiyonu kullanmak istiyorum demeniz yeterlidir. Kullanabilmemiz için çeşit çeşit konfigürasyonlar yapmamıza gerek yoktur. Böylece Projelerinizi istediğiniz yere kolayca taşımanıza + kütüphanelerin yönetilmesine imkân sağlar.  Örneğin; Spring Core kütüphanesini projemize dahil edebilmek için aşağıdaki XML kodunu pom.xml dosyamıza eklememiz gerekiyor.  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-core</artifactId>  <version>4.3.2.RELEASE</version>  </dependency>  Kısaca etiketlerin özelliklerine değinmek gerekirse;   * **<dependency>** : Bir bağımlılık temsil eder. İçerisinde bağımlılığa ait özellikleri barındırır. * **<groupId>** : Bağımlılığın hangi organizasyona ait olduğunu ifade eder. Burada org.springframework aslında birçok kütüphane barındıran bir organizasyondur. Java dünyasında groupId genellikle projenin web adresi ile alakalıdır. Eğer projenizin web adresi “blogprojem.com” ise groupId “com.blogprojem” olabilir. Bu bir şart değil, sadece genel bir yazılı olmayan kuraldır. * **<artifactId>** : Belirtilen organizasyon içinde hangi kütüphanenin getirileceğini ifade eder. * **<version>** : Anlaşılacağı üzere hangi versiyonun getirileceğini ifade eder. | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ | DOOB Tactical Projesi | TARİH : 14.06.2019 |
| Birimlerin birbirlerini takip etmesi için lan üzerinde anlık iletişim gereksinimi üzerinde düşünüldü. İki farklı çözüm bulundu. Bu çözümler:   * Ortak lan’a bağlı cihazların ip adresleri veri tabanına kaydedilerek, anlık iletişim sırasında veri tabanına eklenen ip adreslerine mesaj göndermek * Ortak lan’a broadcast mesajı atarak cihazların ip adreslerini veri tabanına kaydedilerek, anlık iletişim sırasında veri tabanına eklenen ip adreslerine mesaj göndermek.   Bulunan çözümlerden 2.si seçilerek gerekli araştırmalar neticesinde dökümantasyon hazırlanarak gün bitirildi. | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ | DOOB Tactical Projesi | TARİH : 17.06.2019 |
| Broadcast mesajı bire bütün bir iletişim türüdür. Amaç oluşturulan paketi ağdaki tüm düğümlere göndermektir. Noktadan noktaya iletişimin aksine, hedef ana bilgisayarın ip adresini bilmek zorunda değiliz. Bunun yerine, bir yayın adresi kullanılır. IpV4 protokolü uyarınca, bir yayın adresi, ağa bağlı cihazların paketleri almak için etkinleştirildiği mantıksal bir adrestir. Örneği verilecek kod resminde yerel ağın yayın adresi olan 255.255.255.255 numaralı belirli bir ip adresini kullanırız.  İletişim tekniği için UDP terçih edildi. Java’da java.net paketi, UDP protokolü aracılığıyla iletişim kurmak için kullanılabilecek DatagramPacket ve DatagramSocket sınıflarını sunar. UDP tipik olarak ses/video akışı, ağ keşfi vb. gibi düşük gecikmenin garantili teslimattan daha önemli olduğu senaryolarda kullanılır. Datagram ise ağ üzerinden, varış, varış zamanı ve içeriği garanti edilmeyen, bağımsız ve kendi kendine yeten bir mesajdır.  Aşağıdaki örnek kodda UDP protokolü kullanılarak oluşturulan paketin ağdaki tüm düğümlere göndermek hedeflenmiştir. 4445 değeri datagramın göderildiği uzak ana bilgisayardaki port numarasıdir. Diğer bir deyişle sanal veri yolu numarasıdır.  Aşağıdaki örnek kodda ise ağ ortamından gönderilen paketleri almak hedeflenmiştir.  Projede verilen kod örneklerinden yararlanılarak UDP kullanılarak veri gönderme modülü eklendi. Veri alma ve test işlemleri yarına bırakıldı. | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ | DOOB Tactical Projesi | TARİH : 18.06.2019 |
| Projeye UDP veri alma modülü eklendi. Veri alma ve gönderme işlevleri anlık olması için thread yapısı kullanılmaya karar verildi. Tabletlere yüklenilen uygulama çeşitli ayarlarda test edildi. Veri almada problemler çıktı. Gerekli notlar alınarak 7. iş günü bitirildi. | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ | DOOB Tactical Projesi | TARİH : 19.06.2019 |
| Tüm gün veri alma ve gönderme modüllerinde değişiklikler ve testler yapıldı. Gün sonunda karşılaşılan problemlere çözüm bulunamadığından broadcast yapısının değiştirilmesine karar verilip 8. iş günü bitirildi. | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ |  | TARİH : |
|  | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ |  | TARİH : |
|  | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ |  | TARİH : |
|  | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ |  | TARİH : |
|  | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ |  | TARİH : |
|  | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ |  | TARİH : |
|  | | |
| KONTROL SONUCU | | |

|  |  |  |
| --- | --- | --- |
| KISIM |  | YAPRAK NO : |
| YAPILAN İŞ |  | TARİH : |
|  | | |
| KONTROL SONUCU | | |