Bölümler

1. Yazılım şirketinde kimler çalışır?
   1. Yazılım şirketinde çalışanlar ne iş yapar?
   2. HBYS nedir?
   3. Yazılım takım lideri ne iş yapar?
   4. Yazılım geliştirme yaşam döngüsü nedir?
   5. Agile ve scrum nedir?
   6. Sprint nedir?
   7. Kanban board nedir?
   8. Azur DevOps nedir , ne işe yarar?
   9. Developer türleri nelerdir?
2. Framework ne demektir?
   1. Github nedir?
   2. UI nedir?
   3. Jenkins nedir?
   4. Pipeline nedir?
   5. Kubernetes nedir?
   6. Dashboard nedir?
   7. Programlama dili nedir?
   8. Compiler nedir?
   9. Assembly nedir?

3 Debug nedir?

1.0 Yazılım şirketinde kimler çalışır?

Web tasarımcısı , grafik tasarımcısı , web programlamacı ,bilişim teknolojileri (BT)

* 1. Yazılım şirketinde çalışanlar ne iş yapar?

Web tasarımcısı: Resim, logo gibi bir çok tasarımsal ürünlerin tasarımını yapan bilişimcilerdir.

Grafik tasarımcısı: web programcısı, web uygulamaları aracılığıyla neredeyse tüm işlerle ilgilenir ve başlıca görevleri şöyle sıralanır;

* Kurumun web sitesi ya da uygulamaları ile ilgili ihtiyaçlarını belirlemek,
* Gerekli görülen algoritmaları hazırlamak ve kullanıma hazır hale getirmek,
* Web sitesinin görsel tasarımıyla ilgili fikir vermek,
* Web sitesinin tasarımıyla ilgili çalışmak,
* Web ile ilgili gelişmeleri takip etmek,
* Web sayfası ya da uygulamasının verilerini düzenlemek.

Bilişim teknolojileri(BT): Yazılım geliştiriciler, programlar, uygulamalar ve web siteleri oluşturarak yazılım çözümleri uygular. Genellikle geliştirme araçları kullanarak kod yazar ve test ederler. Çalışma, hangi sistemin çözümüne ihtiyaç duyulduğunu değerlendirmek ve tanımlamak için müşteriler ve meslektaşlarınızla konuşmayı içerebilir.

* 1. HBYS nedir?

Hastane Bilgi Yönetim Sistemleri (HBYS); Bilişim dilinde tıbbi süreçler sonucunda ortaya çıkan verilerin oluşturulması, işlenmesi ve depolanmasını sağlayan yazılım ve donanımlar bütünüdür.

* 1. Yazılım takım lideri ne iş yapar?

Yazılım takım liderinin ana görevi projelerin tasarlanması sürecinde kullanılan teknolojiyi, araçları ve teknikleri denetlemektir. Aynı zamanda yazılım mühendislerinden oluşan ekibi yönetmekle yükümlü olan yazılım takım liderinin diğer görevleri arasında;

* Markanın ya da işletmenin yetkilileri ve mühendisleriyle yakın olarak çalışmak ve tasarım sürecinde gereksinimleri karşılamak,
* Geliştirme standartlarına, politikalara ve prosedürlere bağlı kalarak ekibi yönetmek,
* Yönetimin tavsiyeleri doğrultusunda değişiklikleri uygulayarak verimli ve etkili stratejiler geliştirmek,
* Bilişim teknolojileri standartlarını ve prosedürlerini yüksek kalitede yazılım geliştirmek için uygulamak.
* Takımdaki bütün mühendislerinin var olan bilişim teknolojileri standartlarını takip etmesini ve anlamasını sağlamak,
* Yeni takım elemanlarının uyumu için eğitimler vermek ve takım mühendislerini etkili olarak yönetmek,
* Takım elemanlarını denetlemek, eğitmek ve rehberlik etmek, yer alır.
  1. Yazılım geliştirme yaşam döngüsü nedir?

SDLC mümkün olan en kısa sürede, en yüksek kalitede ve en düşük maliyetle yazılım üreten bir süreçtir. SDLC, bir organizasyonun iyi test edilmiş ve üretimde kullanıma hazır yüksek kaliteli yazılımı hızla üretmesine yardımcı olan iyi yapılandırılmış bir aşama akışı sağlar.

SDLC altı aşamadan oluşur:

**1.Aşama: Planlama**

“Ne istiyoruz?” SDLC’ nin bu aşamasında ekip, analiz edilen gereksinimlerin uygulanması için gereken maliyeti ve kaynakları belirler. Ayrıca ilgili riskleri detaylandırır ve bu riskleri azaltmak için alt planlar sunar.

Başka bir deyişle, ekip projenin fizibilitesini ve projeyi en düşük riski göz önünde bulundurarak nasıl başarılı bir şekilde uygulayabileceklerini belirler.

**2. Aşama: Tasarım**

Ekip, ürün için geniş bir dizi gereksinim ve hedef üzerinde anlaşmaya vardıktan sonra “Bunu nasıl inşa edeceğiz?” Sorusuna cevap vermelidir.

Bir ürün ekibi için bu aşama, önerilen çalışmanın öncelik sırasının belirlenmesini, ürünün yol haritasının oluşturulması aşamasıdır. Bu, hem geliştirme hem de ürün ekiplerindeki herkesin amaçlarının daha net bir resmini elde etmesine yardımcı olacaktır.

**3.Aşama : Uygulama veya Kodlama**

Bu, mühendislik ekibinin ürünü fiilen kodladığı aşamadır. Bu aşamada geliştirme ekibi, yol haritasında iletilen üst düzey genel bakışı bir dizi taktiksel görev, bitiş tarihi ve günlük çalışma programlarına dönüştürür.

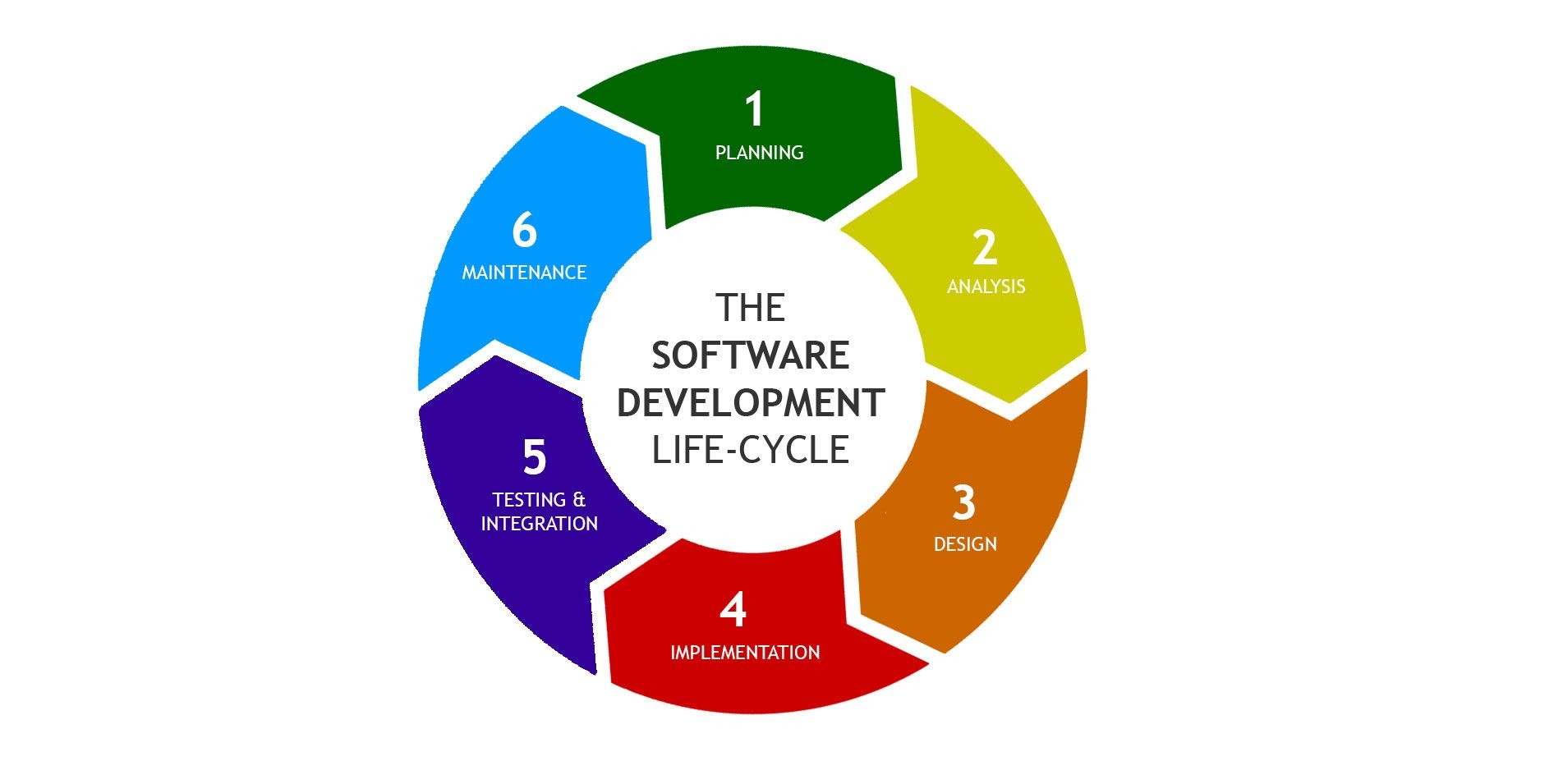
**4. Aşama: Test**

Ekip, yazılımın bir sürümünü tamamladıktan sonra, onu bir test ortamına yayınlar. Burada, QA ekibi ve geliştiriciler, herhangi bir kusur, hata veya diğer sorunları tespit etmek için uygulamanın tüm alanlarını test eder.

**5. Aşama: Dağıtım**

Bu aşamada ekip, tüm kusurları düzelttiğinden ve yazılımın üzerinde anlaşmaya varılan hedeflere ve özelliklere göre oluşturulduğundan emindir. Bu, ürünün genel olarak müşterilerin satın alması ve kullanması için hazır olduğu anlamına gelir.

**6. Aşama: Bakım**

Yazılımın artık canlı olması ve müşteriler tarafından kullanılmasıyla birlikte, geliştirme ekibinin odak noktası onu korumak olur. Geliştiricilerin iyileştirmeler, hata düzeltmeleri ve yeni özellikler için istekleri karşılamaya hazır olmaları gerekir. 

* 1. Agile ve scrum nedir?

Agile modeli proje yönetimi, yazılım geliştirme sürecinde karşılaşılan problemleri çözmek üzere **sık aralıklarla parça parça yazılım teslimatını** ve **değişikliği teşvik eden** bir yazılım geliştirme modeli.

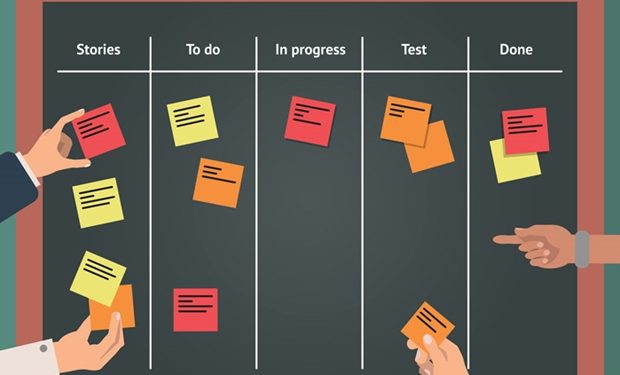
Scrum, agile proje yönetme metodolojilerinden biridir. Projeyi küçük parçalara ayırarak yönetmeyi önerir. Bu sayede her parça tek tek incelenip gerekli değişiklikler yapılabilir. Bir parçada olan sorun daha kolay bulunabilir.Scrum'da başlangıç ve bitiş süreleri belirlenmiş spritler ile yapılır.

* 1. Sprint nedir?

Aylar boyunca lineer bir geliştirme süreci sonunda ürün çıkarmak yerine, hızla değişen ihtiyaçlara cevap verebilmek adına, haftalık çalışma planlamaları “**sprint**” adı verilen çalışma süreleri içinde gerçekleştirilir. Her sprint’te yapılması gereken işler, kişiler tarafından sahiplenilir ve sprint sonunda ürün olarak çıkar.

* 1. Kanban board nedir?

Kanban, profesyonel hizmetlerde işleri/çalışmaları organize etme ve yönetme metodudur. Kanban, hem süreci (iş akışını) hem de bu süreçten geçen fiili işi görselleştirir. Kanban'ın amacı, sürecinizdeki olası problemleri belirlemek ve bunları düzeltmektir, böylece iş, optimum hızda veya verimde gerçekleşebilir.



* 1. Azur DevOps nedir , ne işe yarar?

**Azure Devops** uçtan uca proje geliştirme yapabilmeniz için tasarlanmış bir bulut tabanlı **Devops** ürünüdür. Azure Devops sadece kod geliştirmek için değil, scrum projelerinizi yönetmek, Kanban boardlarınızı yönetmek, kod versiyonlamanızı yapmak, projenizi dev,test, prod ortamlarına deploy etmek (dağıtmak) , test süreçlerinizi yönetmek gibi uçtan uca gerekli olan tüm proje aşamaları için araçlar sunar.

* 1. Developer türleri nelerdir?

Bir web sitesini açtıınızda karşınıza menüler, kampanyaların yer aldığı görsel geçişleri, renkler, fontlar, görsel yerleştirmeleri çıkar. Bu ana sayfa, **bir front-end** developer eseridir. Kısaca, bir web sitesinde kullanıcının görebildiği, etkileşime geçebildiği her şey, yazılımın front-end aşamasında ortaya çıkar. HTML, CSS, JavaScript gibi kodlama dilleriyle oluşturulan front-end kısmı, Backbone, Bootstrap, AngularJS Foundation, ve EmberJS gibi platformların da kullanımı ve jQuery, LESS gibi kütüphanelerden faydalanılarak ortaya çıkar.

**Backend**’in Türkçe karşılığı “Arkayüz”dür. Kullanıcıların görmediği arka yüz (Server-side) geliştiren kişidir. Yani sistemin mimarisini oluşturan, veri tabanı yönetimini planlayan, sunucu ayarlamalarından sorumlu, sistemin maksimum seviyede verimli ve hızlı çalışmasını sağlayan kişidir. Back-end teknolojilerinide Python, PHP, Ruby, Java, C#, ASP.NET, MySQL, MS Sql, MongoDB kullanılır diye özetleyebiliriz.

**Full stack developer** bir web sitesinin veya mobil uygulamanın hem front-end hem back-end kısımlarıyla çalışan kişi, web tasarımcısıdır.

Mobil Uygulama Geliştiricileri mobil cihazlar için yazılımlar hazırlarlar. Mobil geliştiriciler, Java, Objective-C ve C ++ gibi nesne yönelimli programlama dillerine yakından sahiptir. Mobil uygulama geliştiricileri mobil uygulamalar için değişiklikleri kodlar, test eder, hata ayıklar, izler ve değişiklikleri belgelerler. Mobil uygulama geliştiricileri, bir veya daha fazla mobil işletim sisteminin geliştirme ortamlarında çalışmak konusunda yeteneklidir.

**Desktop developer** masaüstü (MacOS, Windows, Linux) işletim sistemlerine uygulamalar yapan geliştiricelere denir.

**Game developer** bir oyunun kodlanmasında görev alan developer tipidir.Genellikle, C, C++ ve Java’yı, Mobil cihazlarda ise yine Java’yı ve Swift’i kullanırlar.

**Gömülü yazılım mühendisleri**, cihazlar üzerinde (microcontrollers, real-time systems, electronic interfaces, set-top boxes, consumer devices, iOT devices) çalışacak olan yazılımların kodlanmasında görevlidir. Yazılımın, cihaza ve cihazın mimarisine bağlı olarak geliştirilmesi gömülü yazılımın ilk ve en önemli basamağıdır.

**Veri bilimci**;(data scientist)bilgi odaklı keşifler yapabilen, bulunduğu ortama dair mevcut durumu tanımlayan, sınıflandırıcı olan ve durum hakkında gelecekle ilgili tahminleri yazılım kullanarak yapabilen kişidir. Farklı araçlar kullanarak elindeki veriyi betimler. Yapılan çalışmalar hakkında çıkarımlar yapar. Öngörüsel, tahminsel ve önleyici modelleme yapar.

**QA developer** test mühendisleri olarak da geçmektedir. Bir ürünün test edilmesinde ve hatalarının tespit edilmesinde görev alırlar.

2.0 Framework ne demektir?

Framework; yazılım ya da uygulama geliştirmek için deneyimli mühendisler ve yazılımcılar tarafından geliştirilen, optimize edilen ve test edilerek kullanıma sunulan bir yazılımdır. Geliştiricilere ufak detaylarla uğraşmadan alanlarında ustalıklarını konuşturacakları bir çerçeve çizdiği için çok yönlüdür, güvenilirdir ve verimlidir.

Geliştiriciler, hangi alanda bir uygulama ya da yazılım geliştirdiklerine bağlı olarak farklı framework çerçevelerinden birini tercih edebilirler. Herhangi bir framework, geliştiricilere bazı temel düzenlemelerin olduğu bir alan sunar. Geliştirici, bu alan üzerinde çalışmaya başladığı için işin en büyük yükünden daha başlamadan kurtulmuş olur.

**Framework türleri:**

* Angular framework
* Laravel framework
* Apache Spark framework
* PyTorch framework
* TensorFlow framework
* Ionic framework
* Xamarin framework
* Flutter framework

2.1 Github nedir?

Github, dünyanın en büyük geliştirici topluluklarından birisi olup, git versiyon kontrol sistemini kullanarak yazılım geliştirme projeleri için web tabanlı bir bulut depolama servisidir.

Ayrıca, Github yazılım geliştiricileri için bir sosyal ağ platformudur. Github sayesinde, yazılım geliştiriciler, kendileri gibi yazılımla uğraşan kişilerin projelerine göz atabilir onları takip edebilirler.

## Github İçin Bilmeniz Gereken Terimler

**Repository:** Repository yani depo, github profilinizin altında açacak olduğunuz dosya dizinleri yani depolardır. Kısaca projeleriniz. Bu depolar gizli olabileceği yani sadece siz görebileceğiniz gibi herkese açık olarak da ayarlayabilirsiniz.

**Branch:** Bir kullanıcının çalıştığını projenin farklı versiyonlarına erişmesini istiyorsanız Branch oluşturabilirsiniz. Kullanıcı projesine bir yenilik eklemek istediği zaman, yaptığı değişiklik projesinin çalışmasını olumsuz etkileyebilir bu olumsuzluktan etkilenmemesi için **branch** kullanılır.

**Origin:** Çalışmanın ana dalı olan **Origin (kaynak)** olarak ifade edilir. Aktif branch da origin olabilir. İlk oluşturulan Branch ise **Master** olarak ifade edilir.

**Fork:** Başka bir geliştiricinin oluşturduğu projeyi kendi hesabınıza kopyalamanıza Fork adı verilir. Başka bir kullanıcının projesini kendi hesabınıza alıp (Fork) üzerinde çalışarak istediğiniz değişiklikleri yapabilirsiniz.

**Clone:** Github üzerindeki bir projeyi bilgisayarınıza kopyalayıp yani indirmenize yarar.

**Commit:** Fork'un ne olduğunu öğrendik. Bir proje için fork yaptınız diyelim yani projeyi kopyaladınız ve üzerinde değişiklik yapacaksınız projeye ekleyeceğiniz yeni kodları da içeren çalışmanın Github'taki kopyasına da iletilmesi için öncelikle **"Commit"** işlemini yapmanız gerekmektedir. Bu işlemi yapmanız demek çalışmanızın Github bulut sistemine gönderilmek üzere paketlenmesi demektir.

**Ignore:** Commit işlemini yaparken bazı dosyalarınızın paketlenmesini ve gönderilmesini istemiyor olabilirsiniz. Bu istemediğiniz dosyaları, ignore edebilirsiniz. Böylece o dosyalar paketlenip gönderilmez.

**Push:** Commit işlemi ile paketlediğiniz çalışmayı Github sunucularına Push edilme işlemi yani gönderilmesine denir.

**Issues:** Sorunlar olarak Türkçe'ye çevrilen Issues, bir durum veya sorun karşılığında bilgilendirme yapmak için kullanılmaktadır.

2.2 UI nedir?

UI, kullanıcı ara yüzü anlamına gelmektedir ve kullanıcıların sitede rahatça dolaşması ve sitede daha uzun süre kalması site içeriklerine daha rahat ulaşması gibi deneyim odaklı iyileştirmeler amaçlanır. UI ne demek diye sorduğumuzda aklımıza ilk gelecek olan şey özetle :  kullanıcı konforu odaklı görsel tasarım gelmelidir.

2.3 Jenkins nedir?

Jenkins, Sürekli Entegrasyon (CI: Continuous Integration) yöntemi için kullanılan java ile yazılmış açık kaynak kodlu bir otomasyon sunucusudur. Yazılım geliştirme süreçlerini otomatize etmemize yarar. Jenkins, belirli bir sunucuda ve portta çalışır, belirlenen kaynaktan projeye ulaşır ve istenen işlemleri gerçekleştirir. Sonuçlarını belirlenen kişilere iletir. Bu sayede projemiz sürekli test edildiği için hatalar hızlıca tespit edilir ve çalışır bir vaziyette tutulur. Jenkins, manuel olarak yapılan build, test ve deploy gibi işlemleri uygular ve bu süreçte yaşanabilecek tüm aksaklık ve iletişim eksikliğini en aza indirir.

2.4 Pipeline nedir?

Pipelines yazılım teslim sürecini aşamalara ayırmaktadır. Her aşamada, yeni işlevselliğin doğrulanması ve hataların kullanıcılarınızı etkilemesini önlemek için yeni özelliklerin kalitesini farklı bir açıdan doğrulamayı amaçlamaktadır. Pipelines ekibe geri bildirim sağlamalı ve yeni özelliklerin sunulmasında yer alan herkese değişiklik akışı konusunda görünürlük sağlamalıdır.

2.5 Kubernetes nedir?

**Kubernetes** ilk olarak **Google** tarafından geliştirilen bir hizmettir. Daha sonrasında **Open Source**olarak herkesin kullanımına açılmıştır.  Kubernetes artık containerlar ve dağıtılmış uygulama için pazar lideri ve endüstri standartlarını düzenleme aracıdır.

2.6 Dashboard nedir?

**Dashboard,** bilgi akışını veya içeriğini özetlemek amacıyla kullanılan, grafikler ve tablolar yoluyla belirli bir durumu açıklamaya yarayan göstergeler ekranını eder. Amacı en kısa sürede, en az etkileşim ve düşünme gereksinimi ile gerekli olan bilginin sunulmasıdır. Örneğin, bir sunucu trafiğini, kullanıcı etkinliklerini, satış departmanındaki çalışan performansını, çağrı merkezi yoğunluğunu, ürün stok durumunu, trafik akışını, günlük ölüm-doğum sayılarını bu bilgilendirme ekranları sayesinde kolay bir şekilde takip edebiliriz.

2.7 Programlama dili nedir?

Programlama dili, yazılımcının bir [algoritmayı](https://tr.wikipedia.org/wiki/Algoritma) ifade etmek amacıyla, bir [bilgisayara](https://tr.wikipedia.org/wiki/Bilgisayar) ne yapmasını istediğini anlatmasının tektipleştirilmiş yoludur. Programlama dilleri, [yazılımcının](https://tr.wikipedia.org/wiki/Yaz%C4%B1l%C4%B1m) bilgisayara hangi [veri](https://tr.wikipedia.org/wiki/Veri) üzerinde işlem yapacağını, verinin nasıl depolanıp iletileceğini, hangi koşullarda hangi işlemlerin yapılacağını tam olarak anlatmasını sağlar.

* 1. Compiler nedir?

**Compiler**(Derleyici), geliştiricilerin herhangi bir programlama dilini kullanarak yazdığı kaynak kodu bilgisayarın anlayabileceği makine diline yani 0 ve 1’lere çeviren aracı yazılımdır. Derleyici sayesinde geliştiriciler farklı programlama dillerini kullanarak aynı işlevi yerine getiren yazılımlar üretebilirler.

* 1. Assembly nedir?

Çevirme dili ya da assembly dili : bir [bilgisayarda](https://tr.wikipedia.org/wiki/Bilgisayar) tüm işlemleri [işlemci](https://tr.wikipedia.org/wiki/%C4%B0%C5%9Flemci) gerçekleştirir ve işlemcinin de, [makine dili](https://tr.wikipedia.org/wiki/Makine_dili) denen kendine has bir dili vardır. İşlemci yalnızca bu dili anlar ve bu dili kullanarak anlaşırsınız. Fakat bu dili öğrenmek ve kullanmak çok zordur. Bu nedenle insanların anlayabileceğimiz bir dilde konuşup ardından işlemcinin diline çeviren yazılımlar geliştirilmiştir. Bunlara derleyici denir. Derleyiciler de bir dile sahiptir fakat işlemcinin diline göre çok daha kolaydır. İşte bu derleyici dillerinden biri de Assembly'dir.

1. Debug nedir?

**Debugging** yani **hata ayıklama**, bir yazılım kodunun beklenmedik şekilde davranmasına veya çökmesine neden olabilecek mevcut ve olası bug-ları algılama ve kaldırma işlemidir.