

İlkay Vural – Yazılım Geliştirme Ve Ortam Araçları Vize Ödevi

H5210025



Selçuk Üniversitesi

Tarihçe: Konya'nın ilk üniversitesi olan Selçuk Üniversitesi adını ve sembolünü Anadolu Selçuklularından almıştır. Kartal sembolü, Doğu ve Batı medeniyetlerinde gücü ve görkemi sembolize etmiştir. Göklerin hâkimi olarak da ifade edilen kartal, diğer kuşlara göre daha yüksek uçabildiği için güneşe yakınlığıyla ifade edilmiştir. Hava ve ateş elementlerini simgeleyen kartal, hükümdarların sembolü olmuştur. Bu fikri destekler nitelikteki arkeolojik buluntu, Anadolu Selçuklu sarayı olan Kubâdâbât Sarayı kazılarında bulunan çini karo üzerindeki çift başlı kartal figürünün göğsünde "Sultan" yazmasıdır. Kartalın çift başlı olarak sembolleştirilmesinden mevcut konumundan daha da erişilmez bir kategoriye yerleşmiş olduğu anlaşılmaktadır. Yönetimin başında bulunanlar hem dinin (Halife) hem de devletin temsilcisi sıfatına sahip olmalarından dolayı çift başlı kartal ile temsil edilmiştir. Çift başlı kartal figürünün aynı zamanda Doğu ve Batı'ya hâkimiyeti simgelediği düşünülmüştür. Konya'da üniversite açılması 1955 yılında gündeme gelmiştir. Bu tarihte üniversite için TBMM'de kanun tasarısı hazırlanmıştır. Tasarı, milletvekillerinin yarısından fazlası tarafından imzalanmıştır ancak Milli Eğitim Komisyonu'ndan geçememiştir. 1962'de MEB'e bağlı olarak açılan Selçuk Eğitim Enstitüsü ve Yüksek İslam Enstitüsü ile üniversiteye sahip olma yolunda ilk ciddi adım atılmıştır. Daha sonra bu adımın güçlendirilmesi ve geliştirilmesi için 1968 yılında Konya'da Üniversite Kurma ve Yaşatma Derneği kurulmuştur. 11 Nisan 1975'de kurulmuştur. 1976-1977 eğitim-öğretim yılında Fen Fakültesi ve Edebiyat Fakültesi olmak üzere 2 fakülte, 7 bölüm, 327 öğrenci ve 2 kadrolu öğretim üyesi ile faaliyete geçmiştir. Selçuk Üniversitesi için atılım yılı 1982 olmuştur. 20 Temmuz 1982 tarih ve 41 sayılı Kanun Hükmündeki Kararname ile ilk etapta üniversitenin çekirdeğini oluşturan Fen ve Edebiyat Fakülteleri birleştirilerek Fen-Edebiyat Fakültesi'ne, Selçuk Yüksek Öğretmen Okulu'nun Eğitim Fakültesi'ne, Konya Devlet Mühendislik-Mimarlık Akademisi'nin Mühendislik-Mimarlık Fakültesi'ne, Konya Yüksek İslam Enstitüsü'nün İlahiyat Fakültesi'ne dönüştürülmesine karar verilmiştir. Bugün bünyesinde 23 fakülte, 7 enstitü, 5 yüksekokul, 1 konservatuvar, 22 meslek yüksekokulu, 54 araştırma ve uygulama merkezi bulunan Selçuk Üniversitesi, altmış sekiz bini aşkın öğrencisi ile Türkiye'nin en büyük eğitim kurumları arasında yer almaktadır.

Selçuk Üniversitesi bünyesinde kurulduğu tarihten günümüze kadar 11 adet rektör görev almıştır. Selçuk Üniversitesi'nin ilk rektörü Prof. Dr. Ali Rıza Çetik'tir (1975-1979). 2020' den günümüze kadar ise Prof. Dr. Metin Aksoy Selçuk Üniversitesi'nin rektörlüğünü sürdürmektedir. Prof. Dr. Metin Aksoy' dan bahsetmemiz gerekirse Londra Stantoon Scholl of Economics'te İngilizce eğitimi (2006-2007) aldıktan sonra 2007 yılında Selçuk Üniversitesi Uluslararası İlişkiler Bölümünde Yrd. Doç. Dr. olarak göreve başladı. Aynı üniversitede 2013 yılında Doçent unvanını aldı ve 2018 Yılında Profesör oldu. Sivil Toplum kuruluşlarıyla işbirliği içinde oldu. Akademik Birlik Platformu kurucusu olarak görev aldı. Ulusal ve yerel gazetelerde köşe yazıları kaleme aldı. İngilizce, Almanca, Rusça ve Kırgızca bilen Aksoy, evli ve üç çocuk babasıdır. Prof. Dr. Metin Aksoy, 2547 sayılı yükseköğretim kanununun 13'üncü maddesi ile 3 sayılı Cumhurbaşkanlığı kararnamesinin 2'nci, 3'üncü ve 7'nci maddeleri gereğince 23 Haziran 2020 tarih ve 202/303 sayılı kararla Selçuk Üniversitesi Rektörü olarak atandı. Halen bu görevini sürdürmektedir. Rektörün öğrencilere mesajı şu şekildedir; Selçuk Üniversitesi'ni bilimsel, yenilikçi, sanatsal, kültürel ve sosyal sorumluluk projelerine açık bir kurumsal yapı olarak daha da güçlü kılacağız. Projelerimizin devamlılığında disiplinler arası çalışma ortamına ihtiyaç olduğunun bilinciyle teknoloji ve sanat üst kurullarımızla eğitim, kültür, sanat, sağlık ve spor projelerimizi topluma faydalı olacak şekilde hayata geçireceğiz. Birlikte yürüyeceğimiz öğrencilerimize de ilk günden gönülümü açıyorum. "Selçuk Üniversiteli" olmanın ayrıcalığını her fırsatta yaşayacaklarını ve geleceğe emin adımlarla yürüyeceklerini temin ederim. Birlikte çok güzel işlere imza atacağımızdan eminim.

Üniversitenin idari kadrosunda rektör yardımcıları da bulunmaktadır. Prof. Dr. Emrullah Eken, Prof. Dr. İlhan Çiftçi, Prof. Dr. Ahmet Tuğrul Polat rektör yardımcılarıdır. Genel sekreter ise Prof. Dr. Kamil Beşoluk'tur.

Selçuk Üniversitesi'nin bünyesinde bulundurduğu fakülteleri sıralamak gerekirse; Dış Hekimliği Fakültesi, Eczacılık Fakültesi, Edebiyat Fakültesi, Fen Fakültesi, Güzel Sanatlar Fakültesi, Hukuk Fakültesi, İktisadi ve İdari Bilimler Fakültesi, İletişim Fakültesi, Mesleki Eğitim Fakültesi, Sağlık Bilimleri Fakültesi, Tıp Fakültesi, Teknik Eğitim Fakültesi, Veteriner Fakültesi, Ziraat Fakültesi, Sanat ve Tasarım Fakültesi, Teknoloji Fakültesi, Turizm Fakültesi, Akşehir İktisadi ve İdari Bilimler Fakültesi, Akşehir Mühendislik ve Mimarlık Fakültesi, Beyşehir Ali Akkanat İşletme Fakültesi, Beyşehir Ali Akkanat Turizm Fakültesi Selçuk Üniversitesi fakülteleridir.

Selçuk Üniversitesi bünyesinde çokça yüksekokul, enstitüler, araştırma ve uygulama merkezleri de bulundurmaktadır.

Selçuk Üniversitesi'nin idari kadrosuna göz attığımızda çok sayıda idari ve akademik personel bulunduğunu görebiliriz. Personel Daire Başkanlığı Başkanı Selahattin Şen, Şube Müdürü Semrah Engez, Şube Müdürü Erol Öğüt, Avukat Fatma Çoker Verimli yönetimde yer alan personellerdir.

Selçuk Üniversitesi'nin kendine ait sloganı vardır. Bu 'Değişen ve Değiştiren Üniversitedir.'

Süreç Adımları:

Planlama : İlk olarak nasıl bir adım izleyeceğimi planladım. Hangi adımları geçip kodlamaya başlamaya varacağımı belirledim. Bu ürünü geliştirdiğimde ne kadarlık bir maliyete sebep olacağını ve sonrasında ne kadarlık bir kar bırakır bunu belirledim. Her proje kar bırakıyor denilemez. Bu yüzden önceden planlama yapmak yararımıza olur. Geliştireceğimiz yazılımın hukuki yönden uyulması gereken kuralları var mı bunları araştırdım. Benzer geliştirilen sistemlerin piyasada ne durumda olduklarını araştırdım. Bu proje en çok ne amaçla kullanılıyor,nerelerde en çok dikkat etmemiz gerekiyor bunları belirledim. Yazılımın nerelerde kullanılacağını belirledim web ve mobilde kullanılacaktır. Yapacağımız proje içereceği kategorileri planladım. Kısacası temel ihtiyaçları burda belirledim.

Analiz : Bu adım genel olarak yazılım gereksinimlerinin belirlendiği adımdır.Kısacası ihtiyaçlarımızı belirlediğimiz adım. Ben bu adımda öncelikle bu sitenin nasıl bir sistemde olması gerektiğini parçalara ayırdım.Uyacağımız prensipleri belirledim. Bu yazılım geliştirirken ekte bulunması gereken kişileri belirledim. Örnek olarak Proje Yöneticisi, Yazılım Mühendisi, Grafik tasarımcı vb. Bu sistemi oluştururken nasıl bir planla ilerlemem gerektiği hakkında detaylı araştırma yaptım. Sistem hem öğrencinin hem yönetimin tüm gereksinimlerini karşılamalı ve üniversite sisteminde verilebilecek tüm bilgiyi vermeliydi. Diğer üniversite sistemlerindeki analiz ederek bu sistemden ne istediğimi net şekilde belirledim. Sınıflar ve nesneler arasındaki ilişkileri genel olarak belirledim. Bir üniversite sistemi çok dinamik olmalıydı. Kod güvenliğini nasıl sağlayacağımıza karar verdim. Depo private oluşturulacak yetkisiz kişiler kodlara erişemeyecek. Proje kritik noktalarda dallar ile ayrılacağını belirledim. Dökümantasyonun nasıl olacağına karar verdim. Dökümantasyonsuz hiçbir iş yapılmayacak her kodun amacı dokümantasyona eklenecek. Versiyon sisteminde git kullanacağımızı belirledim. Projenin hangi yazılım süreç modeli ile geliştirileceğini belirledim. Çevik(Agile) yazılım süreç modeli kullanılacak. Proje ekibinin nasıl koordine olacağına karar verdim. Haftalık adım adım toplantılar ve herhangi bir sorunda e- posta ile iletişim sağlanacak. Proje yöneticisi her şeyi takip edecek. Böylece analiz adımını tamamladım.

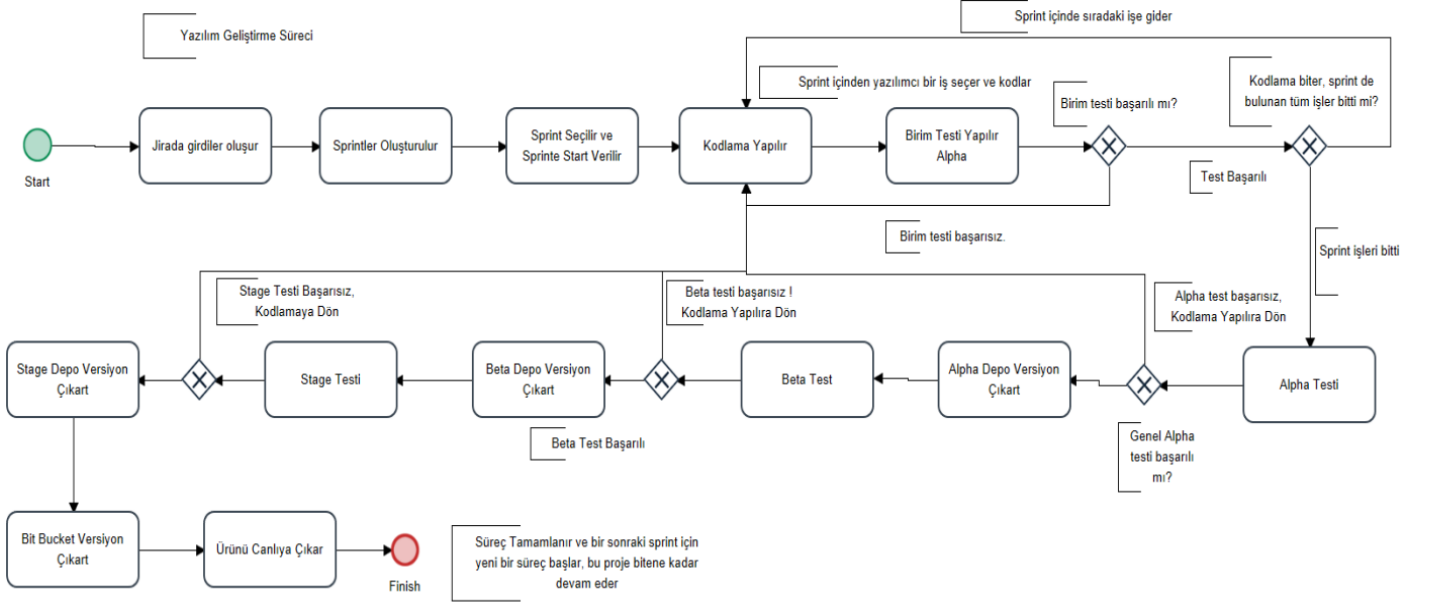
Tasarım : Tasarım adımında analiz adımından yararlanarak projesini çizdim. Bu adımda en büyük amacım müşterinin isteklerini karşılamaktı. Sistemde bulunacak özelliklerin nasıl olacağını ve bunları nasıl gerçekleştireceğimi belirledim. Ardından kağıt üzerinde algoritmaları ve uml diyagramlarını belirledim. Kategoriler arası ilişkileri belirledim. İlişkileri ve özellikleri belirlediğimizde işimiz kolaylaşır. Projenin çoğu kısmı belkide uml ve use caselere bağlıdır. Kullanıcıların erişebilecekleri alanları belirleriz. Arayüzlerini oluşturdum. Wireframeleri kağıt üzerinde çizdim. Sisteme ait bir adet veri tabanı tasarlayacağımı belirledim. Veri tabanında tutacağım veri yapıları ile sistemi kafamızda kuracaktık. Sistemi Html,css,javascript kullanarak hazırlayacağıma karar verdim veri tabanında ise mssql kullanacağımı belirledim.

Gerçekleme : Bütün yapacağımız işleri ilk önce havuzda topladık. Daha sonra analizini, tasarımını belirlediğim sistemi belirlediğim dillerle yazmaya başladık. Bu adımda her zaman solid prensiplerine uymaya dikkat ettik. Çünkü ileride ne doğuracağını bilerek yazmalıydık kodları ve maliyeti en düşük seviyede tutmalıydık. Kural farklılıkları olabilirdi, müşterilerin yeni talepleri olabilirdi, müşterinin zamanında eksik anlattığı şeyler olabilirdi, bizim yanlış veya eksik anladıklarımız olabilirdi, yazılımın doğasında değişim var ve biz bunu kabul etmek zorundayız. Bu yüzden yazılım geliştirme prensiplerinin doğru bir şekilde uygulanmış olması gerekmektedir ve bizde bunu isteyerek ilerledik. Her zaman temiz kod prensibiyle ilerledik ve ektende istediğim buydu. Temiz kod derken bahsettiğim her kod kendi bloğu içinde yazılmalı ve çözümlenmelidir. Bu test işimizde kolaylaştıracaktı. Kolaya kaçmadan okunaklı olmasını istiyordum kodların çünkü her şey daha anlaşılır olduğur ve düzenlemesi daha kolay olduğu için. Adım adım projeyi kodladık. Grafiklerini hazırladık, altyapısını kurduk, düzenli bir şekilde yolumuza devam ettik

Test (Sınama) : Bu adımda sistemi geliştirirken ilerledikçe oluşabilecek hataların maliyeti artacağı için bu sistem geliştirilirken her zaman test edilip kodların sinanması gerektiğini belirledim. Testin bir diğer önemi proje için hataları hızlıca belirlemek olduğu için projede önemli yer tutuyordu. Test senaryolarını oluşturduk. Bu adımda olması gerekenleri belirledik.Test senaryomuzu saymamız gerekirse; Grup,Test Case No, Kullanım Senaryosu No, Test Amacı veya Başlığı, Ön Koşullar, Öncelik,Test Adım No, Girdiler, Test Verisi, Beklenen Çıktılar, Gerçekleşen Çıktılar, Durum, Notlar, Hata No ve Hata bizim baştan aşağı test senaryomuz bu şekildeydi. Alpha, Beta ve Stage ortamlarından test edilmesine karar verdik. Projeyi her açıdan sınavıp müşteriye istenen kalitede teslim edebilmek açısından. Yaptığımız testleri dökümantasyonladık. Hataları ayıklamak ve bulmak için. Bulduğumuz hatalar oldu tabi ki. Yazım hatalarından tut, kullanılan yanlış kodlara kadar. Test aşaması bu açıdan bize çok yol gösterici oldu. Test adımını böyle tamamladım.

Bakım : Geliştirdiğimiz projeyi faaliyete geçirdikten sonra birkaç değişiklik yapma gereğinde bulunduk. Birkaç analiz ve tasarım hataları belirledik ve bakım aşamasında bunların üstüne giderek onları düzeltmeye özen gösterdik. Bakım aşaması teslimden önceki son aşamamız olduğu için projeyi bu aşamada çokça test edip oluşabilecek en ufak hataya bile önlemlerimizi almaya özen gösterdik. Kodlama hatalarına iyice göz attık. Oluşturduğumuz senaryoları tamamladık ve teslim hazır hale getirdik.

Teslim : Geliştirdiğimiz projeyi en ince ayrıntısına kadar test ederek, bakımını yaparak istediğimiz şekilde müşteriye teslimini gerçekleştirdik.



Yazılım Süreç Adımlarında kısmen üstteki tablo benim süreç adımlarımın özeti oldu diyebilirim

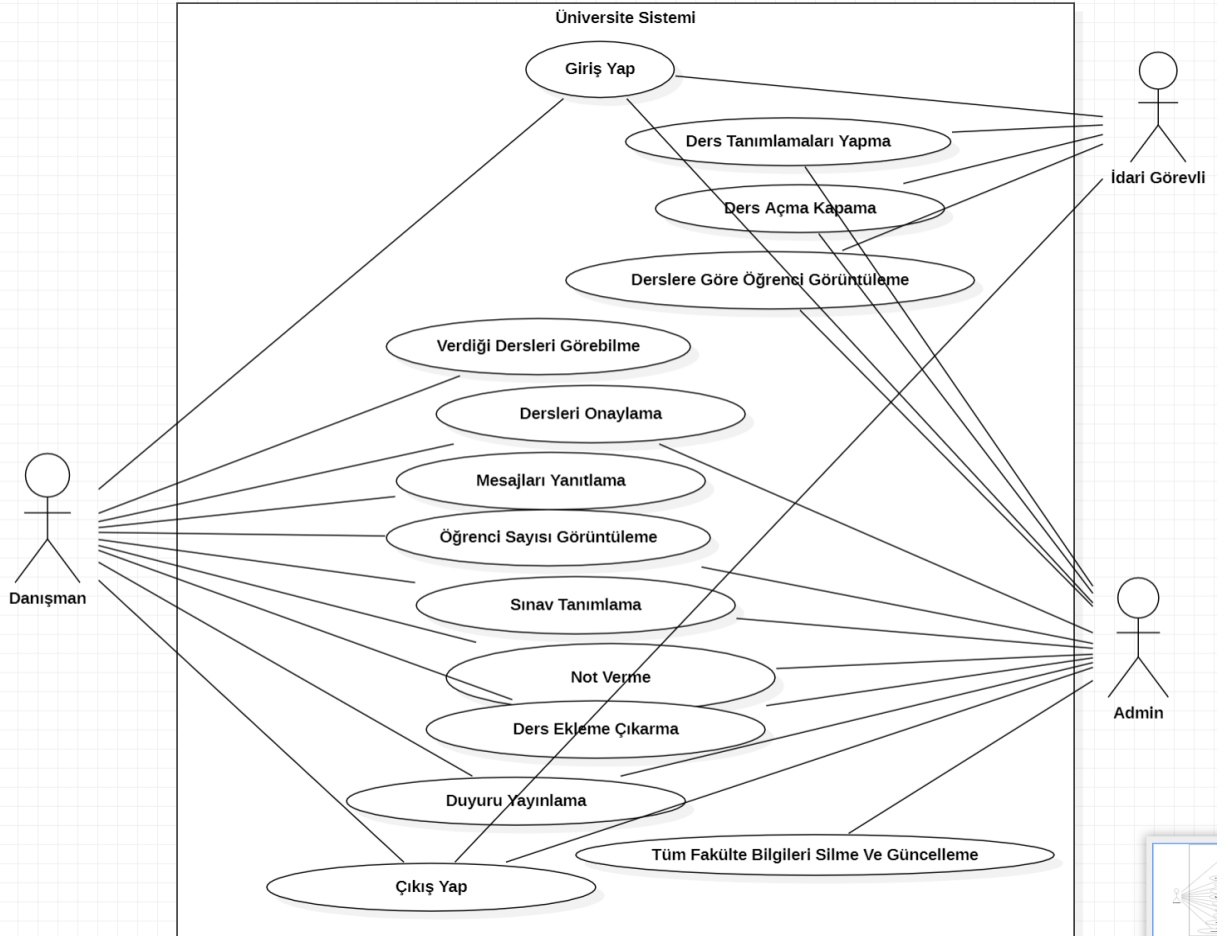
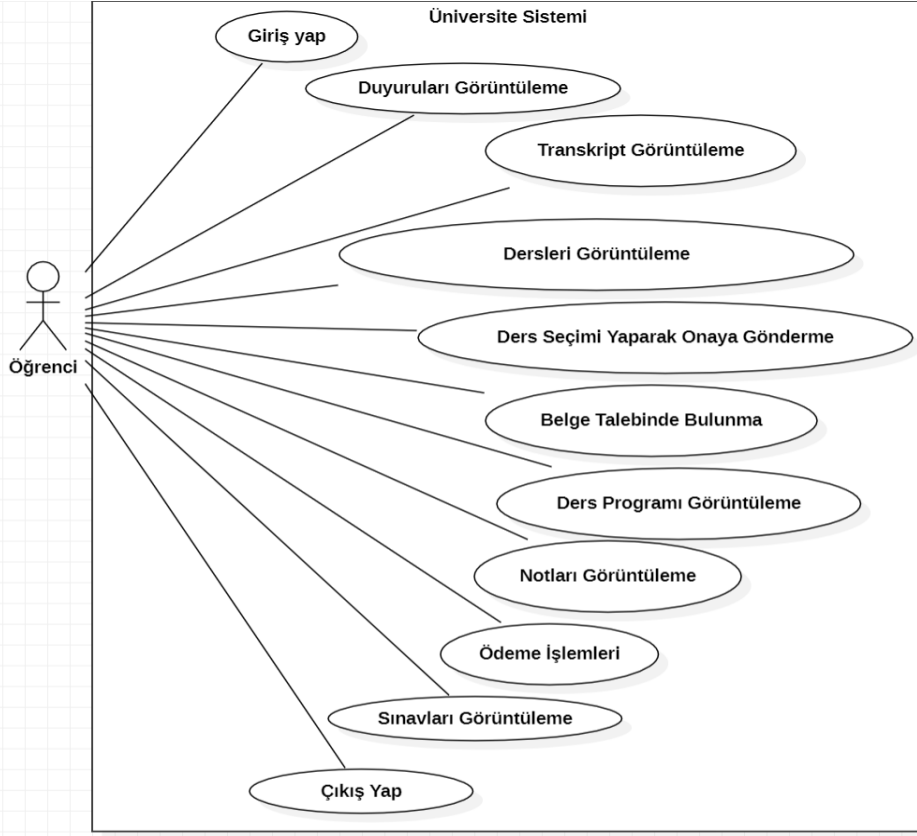
Kullandığım Yazılım Süreç Modeli : Çevik (Agile) Süreç Modeli

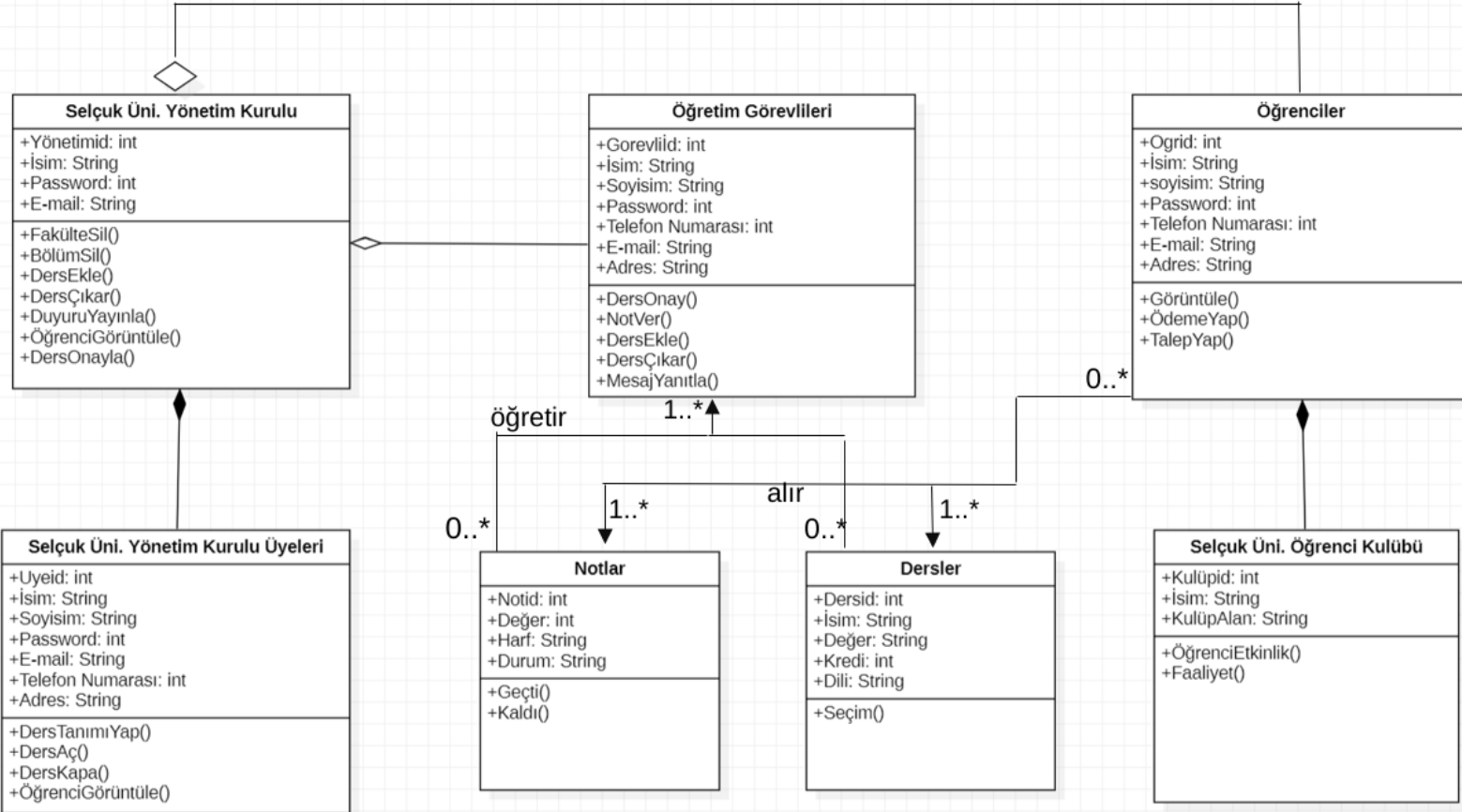
Bu yazılım süreç modelini tercih etmemde tek bir neden yok bir çok neden var bunları açıklamam gerekirse. Kısaca çevik modeli açıklamam gerekirse değişen gereksinimlere, önceden belirlenmeyen durumlara kısacası yazılım ürününü etkileyebilecek her türlü duruma esneklik kazandıran sistem çevik sistemdir. 4 ana maddeye göre: Süreçler ve araçlar değil; bireyler ve etkileşimler, Kapsamlı dokümantasyon değil; çalışan yazılım, Kontrat görüşmesi değil; müşteri ile işbirliği, Planı takip etmektense değişikliklere yanıt vermek her zaman önce gelir. Çevik agile yöntem bu gereksinimlerin hepsini karşılar. Yazılım sürekli gelişmelidir ilkesine dayanarak hareket ettim bu projede ilerlerken. Evet belki yazılım her zaman birbirine uyum sağlayabilir çok düzenli gidebilir ama her zaman üstüne koyarak gelişmelidir ve öncelikle yazılım doğru çalışmalıdır diye düşündüm. Tabiki her süreci yönetecek bir ekip vardır ortada ve bu ekibinde belli özellikleri vardır. Ben çevik süreci seçerken en başta ekibin özelliklerine dikkat ettim. Bu ekibi oluştururken uyum içinde olabilecek insanları seçmeye özen gösterdim. Müşteriye olumlu yaklaşacak, projeye her zaman bir şeyler katmaya çalışacak, yüz yüze görüşme yapabilecek, her verilen işe uyum sağlayabilecek bir ekip oluşturmaya özen gösterdim. Çünkü ekip bana göre yazılım en önemli tarafıdır. Ekibin uyumlu olması demek, her şeyin uyumlu olması demektir. Çalışan yazılım üreterek müşteriye zamanında teslim etmek, birbirleri ve müşteriyle iç içe olmaları, hem teknik hem tüm proje hakkında karar vermeleri çevik süreci yöneten ekibin özelliklerinden olduğu için bu yöntemi tercih ettim. Çevik yöntemi tercih etmemin nedenlerini sizlere saymakla bitiremem. Diğer metotlarda zaman dilimleri ay olarak alınırken çevik metotta ise bu süre haftaya düşüyor. Bu kısa süre içerisinde kısa kısa ilerleyerek müşteriyle iletişime geçip geri bildirim alınarak daha verimli hale getirilebileceğimiz için ekip arkadaşlarımla bu tarzda ilerledik. Çevik yönetimin hem avantajları hem dezavantajları vardı evet bizim için. Avantajları düşük risk parça parça geliştirerek ilerlediğimiz için proje başlangıcındaki ciddi riskler erkenden fark ettik, sürekli var olan değişim bu sistemde değişimi müşterinin avantajına çevirdik, yüksek kalite her şeyi düzeninde parça parça yaptığımızda sonunda yüksek kaliteli bir ürüne vardık ve bu da her türlü hem müşteri hem satıcı tarafından avantaja dönüştü, proje esneklik kazandı, müşteriye projeye dahil ettik ve şeffaflığı sağladık. Avantajları olduğu gibi dezavantajları da bulunuyor tabi ki. Bu yöntemi kullanacak olanlar bunlara dikkat etmeli. Uygun olmayan ekip oluşabiliyor ve bu da oluşacak projeyi çok farklı yerlere götürüyor. Müşteri kararsız ve net değilse gelişme süreç yanlış yönlenebiliyor. Bu dezavantajlara karşılık bizim aldığımız önlemler vardı. En başta oluşturduğumuz ekipte proje yöneticisi olarak bulunduğum için her zaman müşteriyle iletişim halinde oldum, her isteğini en doğru şekilde planlamaya çalıştım ve oluşturduğumuz ekibi sürekli denetledim. Son olarak benim bu süreci seçmemdeki en büyük nedenim 'Kısa zamanda çok başarılı bir ürün' ortaya çıkarabileceğimiz içindi.

Çevik framework aracı olarak ise scrum kullanmayı tercih ettim. Scrum ilk olarak yazılım projelerinin yönetilmesinde kullanılan, proje ekibinin çevik davranarak kompleks süreçleri sadeleştirmesini ve üretkenliğini artırmasını hedefleyen bir proje yönetim frameworküdür. Scrum framework kullanmamında pek çok nedeni var. Bunlardanda bahsetmem gerekirse; Scrum projelerde ki karmaşıklığı en aza indirmek, şeffaflığı arttırmak ve risk oranını en aza düşürmek için 3 temel prensip üzerine kurulmuştur. Bunlar sırasıyla; Şeffalık, Denetleme ve Uyarlamadır. Scrum kısmında roller kısmı da bulunmaktadır. Bu kısmı iş paylaşımı gibi düşünebiliriz. Geliştirme takımı ve müşteri arasındaki iletişimi sağlayan, takımı rahatsız eden ve verimli çalışmalarını engelleri ortadan kaldıran ve geliştirme takımı vardır onlar ise bir ekiptir üzerlerine düşen bütün görevleri yapmakla sorumludurlar. Bende scrum kullanacağımı belirlerken bu ekipleri oluşturabileceğime karar verdikten sonra scruma karar verdim. Proje benden başladığı için düşündüğümde müşteri ve takımla iletişimi sağlayan olduğunda düzeni, takımı rahatsız eden engelleri ortadan kaldıran rolle proje aşamasında huzuru, ekibi oluşturduğumda ise projenin düzgün ilerlemesini sağlayabilirdim. Bu yöntem bu açıdan beni cezbedti. Küçük şeyleri düzenlediğimizde bize projede büyük sonuçlar doğurdu. Ben bu yöntemle projenin daha düzenli ilerleyeceğini düşündüğüm için çevik agile ve framework kısmını tercih ettim. Scrum müşteriye güven sağlamak açısından çok önemlidir. Scrum toplantılar kısmında ilk olarak bir içerik planlama toplantısı yaptık. Proje içeriğini planladık. Sonrasında günlük olarak toplantılar yaptık. Proje yöneticisi, yazılım mühendisi, grafikeri vs hepsini her gün bir araya topladık. Günler geçtikçe projede çok büyük aşama kaydettik. Sprint öncesi yaptığımız toplantılar oluşabilecek bütün riskleri engelliyordu. Scrum kısmında bulunan ürün iş listesi sayesinde içeriğin tüm özelliklerine göre ilerledik. Scrumda kullandığımız grafik sayesinde ne kadarlık ilerleme kaydettiğimizi ve ne kadarlık işimiz kaldığını anlayabiliyorduk. Proje sonunda ekibimizde bulunan bütün arkadaşlarımızı toplayarak ortaya çıkardığımız ürünü değerlendirme toplantısı yaptık. İlerlediğimiz yolda eksik bir şey yaptık mı bunları tartıştık çünkü müşteriye teslim etmeden önce yaptığımız son toplantıydı. Bu kısımda son kez projeyi derleyip çalıştırdık ve her şey istediğimiz gibi gitti. Ardından müşteri ve ben proje yönetici olarak bir toplantı gerçekleştirdim. Oluşturduğumuz projeyi kendisine gösteren istediği bütün gereksinimleri

tamamladığımızı projenin sorunsuz çalıştığını. Müşteri kendisinde bu sistemi test etti. Ve bu sürecin sonunda olumlu şekilde projeyi tamamladık. Kullandığımız süreci düzgün yöneterek ortaya dinamik bir proje çıkardık.

Use Case Diagram





UML DİYAGRAM (CLASS)