# ÖZET

Yüksek Lisans Tezi

UML İLE SANAL KAMPÜS MODELLEMESİ

Sait Ali UYMAZ

Selçuk Üniversitesi Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Sirzat KAHRAMANLI

2007, 101 Sayfa

Jüri: Prof. Dr. Ahmet ARSLAN

Prof. Dr. Sirzat KAHRAMANLI

Yrd. Doc. Dr. Mesut GÜNDÜZ

Bu çalışma lisans ve yüksek lisans eğitim veren bir üniversitenin uzaktan eğitim sisteminin Uml ile modellenmesi amacıyla yapılmıştır. Bu amaçla sistem aktörleri ve rolleri belirlenmiştir. Yazılımın gereksinimleri ayrıntılı olarak ortaya koyduktan sonra süreçler, veri yapıları, yazılım ve donanım bileşenleri üzerinde durulmuştur. Örnek modellemede Uml diyagramlarının yedi çeşidi; kullanım durum diyagramları, sınıf diyagramları, durum diyagramları, etkinlik diyagramları, ardışık sıra diyagramları, bileşen diyagramları ve yaygınlaştırma diyagramları kullanılmıştır.

Anahtar Kelimeler: uzaktan eğitim, sanal kampüs, uml, uml diyagramları

ABSTRACT

Master Thesis

VIRTUAL CAMPUS MODELLING
WITH UML DIAGRAMS

Sait Ali UYMAZ

Selçuk University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Prof. Dr. Sirzat KAHRAMANLI

2007, 101 Page

Jury: Prof. Dr. Ahmet ARSLAN

Prof. Dr. Sirzat KAHRAMANLI

Yrd. Doc. Dr. Mesut GÜNDÜZ

This study has been presented for modelling the distance learning system of a university, which educates undergraduate and postgraduate students, by uml. For this reason, system actors and their roles have been determined. After necessities of the software were produced processess, data constructions, software and hardware components have been accented. In sample model the seven kinds of uml diagrams which are usecase diagrams, class diagrams, state diagrams, activity daigrams, sequence diagrams, component diagrams and deployment diagrams have been used.

Key Words: distance learning, virtual campus, uml, uml diagrams

# İÇİNDEKİLER


**1 GİRİŞ ..... 2**
**2 LİTERATÜR ARAŞTIRMASI ..... 3**
**3 MATERYAL VE METOD ..... 8**
3.1 Uzaktan Eğitim ..... 8
3.1.1 Neden uzaktan eğitim ..... 9
3.1.2 Uzaktan eğitimin tarihçesi ..... 10
3.1.3 İnternet tabanlı uzaktan eğitim ..... 10
3.1.4 İnternet tabanlı uzaktan eğt. sis. sahip olması gereken özellikler ..... 12
3.2 Uml ..... 15
3.2.1 Sınıf diyagramları ..... 17
3.2.2 Nesne diyagramları ..... 17
3.2.3 Durum diyagramları ..... 17
3.2.4 Sıra etkileşim diyagramları ..... 17
3.2.5 Etkinlik diyagramları ..... 17
3.2.6 Kullanım durum diyagramları ..... 18
3.2.7 Bileşen diyagramları ..... 18
3.2.8 Yaygınlaştırma diyagramları ..... 19
3.2.9 İletişim diyagramları ..... 19
3.2.10 Zamanlama diyagramları ..... 19
3.2.11 Etkileşim tanıtma diyagramları ..... 19
3.2.12 Karma yapı ..... 19
3.2.13 Paket diyagramları ..... 20
**4 SANAL KAMPÜSÜN MODELLENMESİ ..... 21**
4.1 1. Aşama: Kullanıcı Bakışı ..... 22
4.1.1 Aktörler ..... 22
4.1.2 Kullanım durumu diyagram ve senaryoları ..... 26
4.2 2. Aşama: Yapısal Bakış ..... 85
4.3 3. Aşama: Davranış Bakışı ..... 87
4.4 4. Aşama: Gerçekleştirme Bakışı ..... 94
4.5 5. Aşama: Ortam Bakışı ..... 96
**5 SONUÇ ..... 98**
**6 KAYNAKLAR ..... 100**

# 1 GİRİŞ

Uzaktan eğitim, 1950'li yıllarda ülkemize eğitici filmlerle girmiş ve diğer teknoloji kaynaklarının kullanımı ile giderek yaygınlaşmıştır. Zaman içerisinde mektup ile öğretim, açık öğretim, televizyon ve son olarak da internet teknolojileri uzaktan eğitimde yaygınlaşmaya başlamıştır. İnternet teknolojileri uzaktan eğitim yöntemleri arasına katıldıktan sonra uzaktan eğitim yaklaşımları da değişmiştir. Uzaktan eğitim artık teknoloji odaklı bir öğretim modeli olmaktan çok öğrenci merkezli pedagojik amaçların gerek çoklu katılıma ve gerekse bireysel öğrenime hizmet ettiği bir öğretim ve öğrenim faaliyeti olmuştur.

İnternet tabanlı uzaktan eğitimle birlikte yazılımcılarda artık bu işin bir parçası olmuş ve eğitimin internet üzerinden daha iyi nasıl verilebilir noktasında çalışmalara başlamışlardır. İnternet tabanlı uzaktan eğitimi sadece ders materyallerinin internette yayımlama olarak değil animasyonlar, öğretici ve öğrencileri görüntülü konferanslar ile buluşturma, soru cevaplar, bölüm sonu eğitici testler, bilgiye hızlı ulaşma teknikleri ve öğrenciye uygun içerikler ile destekleme gibi daha etkin yöntemler gelişmektedir. Bu yöntemler yazılım geliştirme sürecini etkilemiştir ve kodlamadan önce diğer büyük projelerde olduğu gibi sağlıklı bir analiz ve modelleme yapılmasını zorunlu kılmıştır.

Bu noktada Uml (Birleşik modelleme dili) programlama dilinden bağımsız olması ve birçok şematik gösterimi ile iyi bir analiz ve modelleme aracı olmuştur. Uml bir standart olarak Object Management Group (OMG) tarafından dünyaya yayılmaktadır. Yapılan bu tez çalışmasında internet tabanlı uzaktan eğitimin Uml diyagramları ile modellenmesi gerçekleştirilmiştir.

# 2 LİTERATÜR ARAŞTIRMASI

Eğitimde yeni teknolojilerin geliştirilmesi ve uygulanması çok önemlidir. Internet bilgisayar ağı ve uzaktan eğitim, geniş multimedya bilgilerinin sınırsız transferine ve eğitimin yer ve zamandan bağımsız hale gelmesine imkân verir. Öğrencilerin geri beslemesi uzaktan eğitimin kalitesini arttırabilir. Diyalog temelli uzaktan eğitim sistemi öğrencilerin bilgisinin değerlendirilmesine olanak sağlar. Sistem; öğrencilerin geri bildirimlerinin analizlerine, eğitim verilecek bölgenin yapısının ve yeni öğretim içeriklerinin üretilmesinin araştırılmasına dayandırılmıştır. Bu öğretim ilerlemesi ve bilgi kalitesinin gelişimini takip eder. Bir uzaktan eğitim sistemi bütünleşik veritabanı ve www teknolojileri yoluyla uygulanır. (Kovacic ve Skocir 2003)

Uzaktan eğitim internet ve eğitimdeki reformdaki ilginin artmasıyla eğitim yazılımlarına olan ihtiyaçta katlamalı olarak büyümektedir. West Virginia üniversitesinde bir web tabanlı etkileşimli kurs aracı geliştirilmiştir. Bu araç eğitmen, yönetici ve öğrencilerin internet aracılığı ile etkileşimli iletişimde bulunmaları için farklı bileşenler sağlar. Bu çalışma bu aracın bir nesne tabanlı yazılım mimari modelini sunmaktadır. (Ammar ve ark. 2001)

İnternet teknolojilerinin gelişimi, uzaktan eğitim sistemlerinin nasıl tasarlanması ile ilgili birçok teori ve metodun gelişmesine yardım etmiştir. Uzaktan eğitim sistemleri internet veya web üzerinden öğrenci ve öğretmen arasında eğitim alıp veren istemci sunucu sistemleri gibi düşünülebilir. Uml uygulamanın analizinden tasarımına dek geliştirme süresince yazılım modellemesinde kullanılan görsel bir tekniktir. Bu yazıda uzaktan eğitim sisteminin modellenmesi ve geliştirilmesinin Uml ile nasıl olacağı gösterilmiştir. (Markovic ve ark. 2005)

Binlerce yıl önce Sokrat Öğrencileri ile onları tartışarak kurallar ve kavramlar koymak için sadece eleştiri kullanarak bugünün felsefi temelini kurdu. Bu yaklaşım Socratic yöntem olarak adlandırıldı, bir öğretme-öğrenme stratejisi olduğunu ayrıca

ispatladı. Günümüzde web tabanlı olduğu halde işbirliği ile oluşturulan uzaktan eğitim sistemleri kurslar için çevrimiçi içeriğin gelişimi ve yeniden gösterimi için endüstri standartları haline geliyor. Teknolojilerin yüksek kalitede uzaktan eğitim içeriği sağlamak için farklı pedagojik stratejilerin desteği ile nasıl doğru olarak uygulandığı ile ilgili tekrarlayan sorulara yanıt bulmak için socratic metottan bazı özellikler alınabilir. Bu yazıda Piaget gibi birçok etkileşimli araçlar üzerinde büyük ölçüde yeniden kullanılabilen küçük parçacıklı öğrenme nesnelerinden derleme bir örnek gösterilmiştir. (Silveira ve ark. 2003)

Bu dokümanda ticari boyuttaki Uml diyagramlarının uygulama alanlarının tersine mühendislik yaklaşımı ile yeniden kurulumu tanıtılmaktadır. Bu yaklaşım kısmi olarak uygulama alanının nesne tabanlı yaklaşım modelini sağlayan Uml sınıf diyagramlarının yeniden kurulumuna izin verir, sıralı diyagramları belirlenmiş ticari nesnelerin üzerindeki etkileşimlerini ve web uygulamaları tarafından sağlanan kullanıcı fonksiyonlarının kulanım tipi diyagramların modellendirir. Kaynak kod analizinden faydalanan sezgisel kriterler diyagramların iyileştirilmesi için kullanılır. Bu kriterleri uygulamak için kullanılan araçlar üretilmiş ve bunları güncellemek için yapılan deneyler konu çalışmalarının yardımı ile yürütülmüştür. Deneysel sonuçlar planlanan yaklaşımın uygulanabilirliğini ve geçerliliğini göstermiştir. (Lucca ve ark. 2003)

Son 20 yıldan beri, yazılım mühendisliği ve gelişmiş sistemlerin tasarımı inşaat mühendisliği ve yerbilimi fakültelerinde müfredatın bir parçasıdır. Hem lisans hem de yüksek lisans seviyesindeki öğrencilere birbiriyle bağlantılı derslerin çeşitleri sunulur. Bunun başlangıcında program için prosedüre bağlı yazılım geliştirme tasarımı tercih edilir. Programın bütünü zamanında eğitimsel uygulamalar için elverişli olmuş fakat kademe kademe güncelliğini yitirmiş ve bu kursların yeniden yapılandırılması için teşvik edici hale gelmiştir. Yeni program öğrencilere modern tasarım anlayışını ve programlama tekniklerini nesnel tabanlı programlar olan Java ve Uml'nin kurallarını sunar. Bununla birlikte gerçek yaşam yazılım geliştirme durumları için Uml modellerinin soyut ve diğerlerine oranla karmaşık içeriği öğrencilerin zorlanmasına sebep oldu.

Bu sorunu aşmak için kursta sadece mevcut Uml editörü dâhilinde ki modelleme tekniklerinin altkümeleri sunuldu. Ana fikir öğrencilerin aşağıda verilen 3 modeli çıkarması: (1) fiziksel nesne modeli, (2) Kullanıcı etkileşimleri ve (3) sistemin tepkileri. Bu amaç için, sadece sınıf diyagramları ve kullanıcı durum diyagramlarının özelliklerinin kullanıldığı Uml araçlarından sadeleştirilmiş bir modelleme yöntemi geliştirildi. Yüksek lisans kursları boyunca gelişmiş tasarım sistemlerine değinerek diğer ileri özellikler sunulur. (Beheshti ve Dado 2003)

Yazılım projesi yönetiminin ana hedefi bir yazılım ürününün zamanında yapılacağının maliyet limitlerinin ve uygun kalitenin sağlanacağını temin etmektedir. Ana problem karmaşık tasarımı gerçekleştirmek için ihtiyaç duyulan güç kaynak için yeterli ve etkili ödenek ve çevrenin gelişimi için uygun tahminlerde bulunmaktır. Uygun bir proje planı, kendi ideal alt görev gelişim planlamasına sahip olmalıdır. Sonuç olarak, proje yönetim metodolojisi, projenin gelişimi süresince oluşacak risklerin üstesinden gelmelidir. Proje yönetim aktiviteleri ölü noktaların tespiti için gelişim işlemlerin çok erken evrelerinde başlamalıdır ve sürekli olarak bu işlemlerin gelişim ile uyum sağlamalıdır.

Uml'yi projelendirme dili olarak kullanmanın en büyük faydalarından biri uygulamanın en başındaki evrelerinden beri kullanılıyor olmasıdır. Bir sistemin dinamik hareketlerinin modellenmesi özelliğiyle Uml'nin niteliklerinin işletilmesi iş akışlarını canlandırabilir. Bu yöntem ayrıca işlem gelişiminin kendini tanımlaması için kullanılabilir. Bu yöntemde ana amacımız, bütün ana faktörlerin proje ve Uml tabanlı tasarım üzerine eklentilerin raporlanabilmesi ve Uml diyagramlarına dayanan bir yazılım projesi optimizasyon probleminin matematiksel modelini oluşturmaktır. (Doban ve Pataricza 2003)

Bilgi toplumu eğitimsel organizasyon ve yönetimle ilgili yaklaşımlardaki hızlı değişimi uygulamaktır. Yeni yaklaşımlar ICT'nin geniş bir alanda kullanılmasına dayandırılmış ve iletişim genellikle internet sayesinde sağlanır hale gelmiştir. Uzaktan eğitim zaman ve mekân engelini ortadan kaldırmakta, eğitimde ki klasik hazır bulunma sınırlandırmakta ve uzaktaki öğrencilerin bağımsız olarak çalışmalarına imkân sağlamaktadır. Bu makalede klasik ve sanal üniversitelerin sistemleri karşılaştırılmıştır.

Ayrıca sanal üniversitelerin ana elementlerini ve özel işlemlerini içeren modeller sunulmuştur. Ayrıca önemli katılımcılar ve onların rolleri tanımlanmıştır. Modelin oluşturulması Rup ve Uml dili ve bazı Uml diyagramlarının kullanımına dayandırılmıştır. Modelin ana amacı klasik ve sanal üniversitelerin seçilmiş ekonomik göstergelerini karşılaştırmak, hesaplamak ve niceliklerini belirtmektir. (Kadavová ve ark. 2006)

Son yıllarda bu alanda ki uzman yazılımcılar tarafından sanal sınıflar oluşturmak amacıyla çeşitli modelleme uygulamaları tasarlanmıştır. Bunlardan bir tanesini göz önünde bulundurursak, bu makalenin yazarları, seviyesi günden güne artan uzaktan eğitime dayandırılmış olan modelin zorlukları ve olabilirliği üzerinde çalışmıştır. Bu yazarlar yordam program ve materyallerin öğretimi için kullanılan yüksek seviyeli eğitimin ihtiyaçlarına kıyasla sanal sınıfların bazı önemli sorunlarını ve metodolojik unsurlarını incelediler. Asıl amaç, bu özel modelleme için tahsis edilen model nesnelerinin tanımlanmasına izin veren karmaşık durumları açıklamaktır. Bu yazı aşağıda gösterildiği gibi düzenlenmiştir. İnternet ortamında sanal öğretim prosedürünün tanıtımıyla başlar bunu takiben sanal eğitimin yerel ve evrensel görünümü tartışılır ve sanal eğitimin asıl avantajlarının farkına varılır. Daha sonra bahsi geçen kursun değerlendirmesini amaçlayan uygulama modeli belirlenir. Son olarak yüksek eğitim egzersizlerinin sanal sınıflarda ki uygulamalarının olabilirliğinin sonucuna varılır. (Kartyas ve Gati 2005)

İnternet hızla gelişirken, WWW (World Wide Web) de öğretimin aktarılması için güçlü, etkileşimli ve dinamik bir araç haline gelmektedir. Birçok yüksek öğretim kurumu eğitim ve öğretim sağlamak için web'i kullanmaktadır. Web-tabanlı öğretim sağlayan bu kurumlardan birçoğu web'in eğitim ve öğretim için uygun bir araç olduğu gerçeğini algılamaktadır. Bir eğitim ve öğretim aracı olarak web, web-tabanlı öğretim gibi iyi tasarlanmış kaynakların oluşturulmasını destekleyecek potansiyele sahiptir. Bu çalışmada web-tabanlı eğitimin teorisi incelenmiştir. Ayrıca örnek bir web-tabanlı ders tasarlanmış, geliştirilmiş ve tamamlanmıştır. Bu çalışma iki kısımdan oluşmaktadır. Birinci kısımda web tabanlı öğretim ve uzaktan öğretimde web konusu incelenmektedir. İkinci kısımda ise web tabanlı öğretim için kullanıcı arabirimi tasarımı, uzaktan eğitim

uygulamalarının tasarımı ve kullanışlılık testleri ile bir web tabanlı kursun oluşturulması ve tamamlanması anlatılmaktadır. (Bay ve Tüzün 2002)

İnternet teknolojilerinin hızlı gelişimi, uzaktan eğitime yönelik yeni olanaklara ve uygulama alanlarına da yol açmaktadır. Bu açıdan, internet, uzaktan eğitim açısından da yeni bir ortam ve araç olarak karşımıza çıkmaktadır. İnternet tabanlı uzaktan eğitim uygulamaları, öğretmen merkezli eğitim paradigmasından öğrenci merkezli bir alana doğru gerçek bir değişimi de ifade etmektedir. Bu bağlamda, nesnelci öğretimden oluşturmacı öğrenme yaklaşımlarına doğru da bir anlayış değişikliğinin meydana geldiği söylenebilir. Bu bildiride, geleneksel öğretim tasarım modelleri ele alınarak, nesnelci ve oluşturmacı stratejileri içeren karma bir tasarım modeli üzerinde durulmaya çalışılacaktır. (Atıcı ve Gürol 2001)

# 3 MATERYAL VE METOD

## 3.1 Uzaktan Eğitim

Bu eğitim ile ilgili pek çok tanım vermek mümkündür. Bu konuyla ilgilenen veya bu yöntemi kullananlar kendilerine göre tanımlama yapmıştır. Örnek vermek gerekirse;

***The Pennsylvania State University**, (1998)* tarafından yapılan tanımda, uzaktan eğitim, öğrenmeyi destekleyen yöntemleri ve teknolojileri içeren eğitimsel bir sistemdir.[19]

***Idoha Üniversitesi**, (2001)* tarafından yapılan tanımda, uzaktan eğitim, en temel anlamda bir öğretici ile öğrenicinin fiziksel bir uzaklıkta ayrılmasından dolayı meydana gelen bir gereksinimdir. Geleneksel eğitimin yerini alan ses, video, bilgisayar verisi ve yayımlanan yayınlar gibi teknolojiler uzaktan eğitim ile geleneksel eğitim arasında köprüdür. Bu yollarla verilen eğitim programları yetişkinlere aldıkları eğitime ek olarak yeni eğitim fırsatları sunar. Zaman, mesafe ya da fiziksel engelliler için de bu sistem, bulundukları iş-ev ortamında bilgilerini arttırma, eğitimlerini sürdürme olanağı demektir.[20]

***California Distance Learning Project,** (CDLP 2004)* tarafından yapılan tanımda, uzaktan eğitimin öğrenciyle eğitsel kaynaklar arasında bağlantı kurularak gerçekleştirilen bir eğitim sistemi olduğu belirtilmekte, uzaktan eğitim programlarının herhangi bir eğitim kurumuna kayıtlı bulunmayan kimselere de eğitim imkânı sağlamasının son dönemde öğrencilere tanınan eğitim imkânlarını artıran bir durum olduğunun altı çizilmektedir. Uzaktan eğitim programının bir başka yönü de mevcut kaynaklardan yeterince faydalanarak gelişen teknolojiyi de yakından takip etmek zorunda olmasıdır.[21]

***United States Distance Learning Association,** (USDLA 2004)* tarafından yapılan tanımda, uzaktan eğitim uydu, video, ses, grafik, bilgisayar, çoklu ortam teknolojisi gibi

araçların yardımıyla, eğitimin uzaktaki öğrencilere ulaştırılmasıdır. USDLA, öğretmen ve öğrencinin birbirlerinden coğrafi olarak uzak olduğunu belirterek bu eğitim programında elektronik araçların ya da yazılı materyal ve matbu malzemelerinin kullanılması gerektiğinin altını çizer.[21]

***Alkan (1987),*** uzaktan öğretimi şöyle tanımlamaktadır: geleneksel öğretme-öğrenme yöntemlerinin sınırlılıkları nedeniyle sınıf içi etkinliklerini yürütme olanağının bulunmadığı durumlarda, eğitim etkinliklerini planlayanlar ve uygulayıcılar ile öğrenciler arası iletişim ve etkileşimin özel olarak hazırlanmış öğretim üniteleri ve çeşitli ortamlar yoluyla belirli bir merkezden sağlandığı bir öğretim yöntemidir. [13]

Genel tanımlar vermek yanıltıcı olsa da, uzaktan eğitime genel bir tanım vermek gerekirse öğretici ile öğrencinin aynı anda aynı mekânda olmasını gerektirmeyen eğitim şeklidir. Teknolojinin gelişmesi ile uzaktan eğitimin tanımı da zamanla değişmekte ve örgün eğitimle arasındaki farklar giderek artmaktadır.

### 3.1.1 Neden uzaktan eğitim

Uzaktan eğitime geçişteki en önemli sorulardan biri "niçin uzaktan eğitim" sorusudur. Uzaktan eğitimin tarihsel gelişimi, özellikle teknoloji araçlarının bu eğitime dâhil olması niçin uzaktan eğitim sorusuna verebileceğimiz cevapların sayısını arttırmaktadır. Bu soruya verilebilecek cevaplar;

- Geniş bir öğrenen kitlesine ulaşmak
- Derslere devam edemeyen özürlü öğrencilere ulaşmak
- İş hayatı nedeniyle zaman sıkıntısı çeken insanlara ulaşmak
- Öğrencilerin kayaklara erişim ve araştırmacı özelliklerini arttırmak
- Her öğrenciye özel kişisel eğitim imkânı vermek
- Monolog ortamlara hapis olmamak
- Farklı kültürlerden farklı konulara hâkim öğrencilerle tanışmak
- Konferans şeklinde farklı konuşmacılarla öğrenme ortamı oluşturmak
- Geleceğe ayak uydurabilmek.

### 3.1.2 Uzaktan eğitimin tarihçesi

Uzaktan eğitim çalışmaları 200 yıldan daha eski yıllara kadar uzanmaktadır. Örneğin, 1728 yılında Boston Gazetesi'nde mektup ile stenografi dersleri verildiğine ilişkin reklâmlar bulunmuştur. 1890'lı yıllarda Avustralya'daki Queensland Üniversitesi kampüs dışına açık bir eğitim programı yürütmüştür. Benzer bir programı da 1920'lerde Columbia Üniversitesi gerçekleştirmiştir. 1930'lara gelindiğinde radyo artık pek çok okul tarafından bir uzaktan eğitim aracı olarak kullanılmaya başlanmıştır. 1950'lerde ise Amerika'da özellikle askeri amaçlı olarak kullanılan uzaktan eğitim için kâğıt tabanlı iletişim ortamı kullanılmıştır. Teknolojik gelişmeler sayesinde günümüze gelindiğinde ise disketler, videokasetler, cd-romlar, uydu yayınları, video konferanslar ve Internet, uzaktan eğitim çalışmalarında önemli bir yer edinmişlerdir.

### 3.1.3 İnternet tabanlı uzaktan eğitim

Teknoloji geliştikçe hem tanım hem de model olarak örgün eğitimden uzaklaşan uzaktan eğitimin şu anki şartlarda en uç noktası internet tabanlı uzaktan eğitim sistemleridir.

İnternet Tabanlı Uzaktan Eğitim Sistemleri internete dayalı tekniklerin tamamını içinde barındırmaktadır. Bilgilerin statik gösterimi için HTML sayfalar, iletişimin sağlanması için e-mailler, farklı kültür ve coğrafyalardan farklı bilgi birikimlerini paylaşmak için tartışma listeleri, öğrencilerden öğretim hakkında anında geri dönüş almak için anketler testler vs, öğrencinin tercihlerine göre anlatım metotlarını değiştirebileceğimiz dinamik sayfalar kullanılmaktadır.

İnternet tabanlı uzaktan eğitimin en önemli avantajları arasında sanal bir kampüs yaratılabilmesi ve hem eşzamanlı hem de eşzamansız olarak eğitime olanak vermesi yer almaktadır. Öğrenciler sistem dâhilindeki içeriğe istedikleri zaman ulaşabilmekte ve kaynaklardan istedikleri ölçüde faydalanabilmektedirler. Sağlanan bu esneklik, maliyet avantajları ile birleştiğinde ideal bir model oluşmasına olanak tanımaktadır [12].

İnternet tabanlı uzaktan eğitimin günümüzde kabul görmesinin ve popülaritesinin giderek artmasının en temel nedeni eğitimin zamandan ve mekândan bağımsız oluşudur. İnternet tabanlı uzaktan eğitimin bu esnek ve bağımsız yapısı iş hayatı nedeniyle zaman sıkıntısı çeken ya da eğitimin verildiği yerde fiziki olarak bulunamayan kişiler için önemli bir tercih nedeni oluşturmuştur. İnternet tabanlı uzaktan eğitimin ön plana çıkan zaman ve mekân avantajları diğer önemli üstünlüklerini kimi zaman gölgede bırakmıştır. Bu üstünlükler sistemin tercih edilmesinde önemli roller oynayabilecek niteliktedir [14].

Bu nitelikler kişiselleştirilebilir eğitim, öğrenci merkezli eğitim, öğrenci yönetimli eğitim ve düşük maliyetli eğitim olarak karşımıza çıkmaktadır. Kişiselleştirilebilir eğitimde, verilen eğitimin şirket, bölüm, grup hatta kişiye göre özelleştirilmesi sağlanmaktadır. Öğrenciye verilecek eğitimin, öğreticinin kapasitesine göre değil, öğrencinin gereksinimlerine göre belirlenebilmesini öğrenci merkezli eğitim sağlamaktadır. Öğrenci yönetimli eğitimin getirisi ise öğrencinin çevrim içi topluluklar oluşturmasına olanak tanıyarak, öğrencinin kendisi için içerik ya da program oluşturabileceği bir ortam hazırlayabilmesidir.

Maliyet açısından bakıldığında İnternet tabanlı uzaktan eğitimin maliyetinin geleneksel örgün eğitim maliyetinin ortalama yarısı kadar olduğu görülmektedir. Öte yandan İnternet tabanlı uzaktan eğitimin etkileşimli eğitim özelliği ve güncel içerik sunması da göz ardı edilmemelidir [19]

İnternet tabanlı uzaktan eğitim ile ilgili örnekleri incelediğimizde temel olarak iki farklı yaklaşım ortaya çıkmaktadır. Bu yaklaşımlar akademik programlar ve sertifika programları olarak ifade edilebilir. Üniversite ve özel sektör bünyesinde verilen çok çeşitli sertifika programları bulunmaktadır. Sayıca üstün olan bu sertifika programlarını lisans, ön lisans ve yüksek lisans programları izlerken, doktora programlarının sayısı diğerlerine göre bir hayli düşüktür.

İnternet tabanlı uzaktan eğitimden bahsedebilmemiz için birçok fonksiyonu içinde barındıran bir eğitim yönetim sisteminin bulunması gerekmektedir. Bu bağlamda sadece ders içeriğinin elektronik ortamda bulunması yeterli değildir. Ders içeriğine web üzerinden erişilebilmesi İnternet tabanlı uzaktan eğitimin önemli unsurlarından biridir, ancak hiçbir zaman tamamı anlamına da gelmemektedir. Eğitim Yönetim Sistemi, ders

içeriklerinin hazırlanmasından öğrenci kayıtlarının tutulmasına, sistemin kullanım saatleri ve sıklıkları gibi istatistikî bilgilerden, öğrencilerin başarı durumuna kadar birçok bilginin sistem üzerinden alınmasına ya da web ortamına aktarılmasına olanak sağlamalıdır. Bu açıdan bakıldığında İnternet tabanlı uzaktan eğitim sistemlerinin sahip olması gereken özellikler sistemin genelini de tanımlar niteliktedir.

### 3.1.4 İnternet tabanlı uzaktan eğt. sis. sahip olması gereken özellikler

İnternet tabanlı uzaktan eğitim sistemlerinin web üzerindeki basit eğitim içeriklerinden ayrılabilmesi için sahip olması gereken temel özellikler vardır. Bu özelikler eğitim sisteminin amacına ve hedef kitlesine göre kimi zaman değişiklikler gösterse de genel hatlarıyla aşağıdaki fonksiyonları barındırması gerekmektedir.

**1.** Kullanıcıların tanımlanması ve yönetilmesi:

Hazırladığımız İnternet tabanlı Eğitim Sistemi hakkında genel bilgi, tanıtımlar ve eğitimin nasıl işlediğine dair tanıtıcı eğitimler herkese açık olabilir fakat ders içerikleri görüntüleme, hazırlama, istatistikî bilgiler ve değerlendirmeler sisteme belli bir güvenlik modülünden geçen kullanıcılar için açık olmalıdır. Bu güvenlik modülü de yönetilebilir olmalıdır. Sistemdeki yöneticiler bu sisteme giriş yetkisi olan eğitici ve öğrencileri tanımlayabilmeli düzenleyebilmeli ve silebilmelidir.

**2.** Ders içeriklerinin hazırlanması:

İnternet tabanlı Eğitim Sisteminin temelini oluşturan ders içeriklerinin hazırlanması ya da hazırlanmış içeriklerin web ortamına aktarılması sistem içerisinden yapılabilmelidir. Hazır bir şablon kullanılabileceği gibi, içeriğin oluşturulmasında farklı programları da kullanmak mümkündür.

**3.** Derslerin yönetilmesi:

Öğrenci ders yüklerinin kontrol edilmesi, hangi dönem hangi dersi almaları gerektiği ya da hangi dersi aldıkları gibi bilgilerin takip edilebilmesi gerekmektedir. Tüm bu bilgiler ışığında öğrencinin belirli bir programı takip etmesi ve bitirmesi sağlanabilir. Bu sayede sistem genelinde aktif olan derslerin kullanım yoğunluğu da takip edilmiş olmaktadır.

**4.** Öğrenciye özel programların açılması:

İnternet tabanlı Eğitim Sisteminin en önemli avantajlarından birinin esneklik olduğundan daha önce bahsetmiştik. Bu esneklik öğrenciye özel, öğrencinin gereksinimlerine göre anlatım senaryolarını değiştirebileceğimiz programların oluşturulabilmesiyle ön plana çıkan bir özellik haline gelmektedir. Eğitim programı zamandan bağımsız olarak tasarlanabildiğinden, dönemlik, aylık hatta haftalık ders yükleri farklı şekilde belirlenebilir. Seçmeli derslerin sınıf mevcuduna göre açılıp açılmama durumu gibi sorunlar bu sistemde yer almaz.

**5.** Ödev ve proje verilmesi/teslimi:

Öğrencilere ödev ve projelerin verilmesi, bu çalışmalar ile ilgili içerik ve açıklamaların öğrencilere aktarılması, tamamlanan çalışmaların toplanıp değerlendirilmesi gibi işlemlerin yapılabilmesi gerekmektedir. Tüm bu işlemlerin tek bir merkezden yapılması, sorumlu kişilerin üzerindeki iş yükünü azaltacağı gibi, sürece de hız kazandıracaktır.

**6.** Sınav ve testlerin hazırlanması ve uygulanması:

İnternet tabanlı Eğitim Sistemi uygulamalarında dönem içinde aktarılan bilginin öğrenci tarafından ne derecede alınabildiği ortaya konmalıdır. Bütün eğitim sistemlerinde olduğu gibi İnternet tabanlı Eğitim Sisteminde de bu çalışma sınav ve testler yoluyla yapılmaktadır. Bu çalışmalarda iki farklı yöntem genel olarak tercih edilmektedir. Bunlardan biri dönem/eğitim sonunda öğrencilerin bir merkezde toplanarak sınava tabi tutulmalarıdır. Bu sistem farklı ülkelerden sisteme dâhil olan kullanıcılar için uygun bir yöntem değildir. Bu durumda çevrimiçi sınavlar devreye girmektedir. Öğrenciler terminaller yardımıyla merkezden gelen soruları yanıtlamaktadırlar. İki yöntemin beraber kullanıldığı sistemler de mevcuttur. Her iki yöntemde de (ya da ikisini de uygulayan sistemlerde) eğitim süresince öğrencinin kendi bilgi düzeyini test etmesi gerekmektedir. Genel değerlendirmede kullanılacak testlerin yanı sıra, sadece deneme amaçlı olarak testlerin oluşturulabilmesi ve bu testlerin eğitim sistemi üzerinden öğrenciye sunulabilmesi de gerekmektedir.

**7.** Öğrenci davranışlarının izlenmesi ve incelenmesi:

İnternet tabanlı eğitim sistemlerini başarıya taşıyacak en önemli çalışmalardan biri şüphesiz sistemin ne derece etkin kullanıldığının gözlenebilmesidir. Bunun yolu

kullanıcıların sistem içerisinde davranışlarının izlenebilmesinden geçer. Öğrencilerin günün hangi saatinde sistemden ne ölçüde yararlandıkları, hangi ders içeriklerinde ne kadar vakit geçirdikleri gibi bilgilerin sistem üzerinden takip edilebilmesi gerekmektedir. Elde edilen verilerin belirli istatistikî bilgiler halinde sorumlu kişilere aktarılması yine sistemin sorumluluğunda olmalıdır.

**8.** Öğrencilerin başarı durumlarının değerlendirilmesi:

Eğitimin sonunda hem sistemin başarısını, hem de öğrencinin başarısını öğrenci başarı durum değerlendirmesi ortaya koyacaktır. Bu değerlendirme aynı zamanda, diploma, sertifikasyon ya da başarı belgesine öğrencinin hak sahibi olup olmadığını da belirleyecektir. Başarı durumlarının değerlendirilmesi eğitim programında daha sonraki aşamalarda ön koşulun yerine getirilip getirilmediğinin de bir göstergesi olacaktır. Tüm bu çalışmalar sistemin sorumlulukları arasında yer almaktadır.

**9.** Etkileşimli iletişim ortamlarının oluşturulması ve yönetilmesi:

İnternet tabanlı Eğitim Sisteminin önemli avantajlarından birisi de birçok değişik internet tabanlı iletişim sistemini kendi bünyesinde barındırıyor olmasıdır. Tartışma grupları, sohbet odaları, akışkan video ve ses aktarımı gibi kullanıcı etkileşimi sağlayabilecek ara yüz teknolojilerinden en üst düzeyde fayda sağlanması, sistemin sahip olması gereken özelliklerin başında gelmelidir.

## 3.2 Uml

"Unified Modelling Language" 'in kısaltması olan UML Türkçeye "Birleşik Modelleme Dili" olarak çevrilebilir. UML bir programlama dili olmaktan ziyade iş sistemlerinin nasıl modellenebileceğini belirleyen ve açıklayan yöntemlerin bir araya toplanmış halidir. Doğal olarak daha çok yazılım geliştiriciler tarafından kullanılmaktadır. UML ile yapılan modellemeler her zaman yazılım projelerinde kullanılmak zorunda değildir. Mesela bir iş sistemin yapısını sade ve anlaşılır şekilde ortaya çıkarmak için "Yaygınlaştırma Diyagramı" kullanılabilir. "Sınıf Diyagramı" vasıtası ile Nesnel Yönelimli Programlamada temel teşkil eden sınıflar oluşturulabilir ve böylelikle sınıfları geliştirmek sağlanan ek görsellik ile daha kolay hale gelebilir.

Yazılım teknolojisi geliştikçe yazılan programların karmaşıklığı ve zorluğu giderek artmaktadır. Donanım ve yazlımın iç içe girdiği, büyük ağ sistemlerinin giderek arttığı bir dönemde doğaldır ki biz programcıların yazacağı programlarda büyüyecektir. Yazacağımız programlar çok karmaşık olacağı için kod organizasyonu yapmamız zor olacaktır. Hele birçok programcının çalışacağı projelerde bu nerdeyse imkânsız hale gelmiştir. Bu yüzden standart bir modelleme ve analiz diline ihtiyaç duyarız. Programımızın analiz ve tasarım aşamasında modellemeyi güzel yaparsak ileride doğabilecek birçok problemin çıkmasına engel olmuş oluruz. UML daha çok nesneye dayalı programlama dilleri için uygundur. Problemlerimizi parçalara ayırabiliyorsak ve parçalar arasında belirli ilişkiler sağlayabiliyorsak UML bizim için biçilmiş kaftan gibidir. Mesela bir ATM siteminde müşteriyi, banka memurunu ve ATM makinesini ayrı parçalar halinde düşünebiliriz. Müşteri ATM makinesinden para çeker, banka memuru ATM makinesine para yükler. Ama banka memuru ile müşteri arasında doğrudan bir ilişki yoktur. Bu tür ilişkiler UML 'de çeşitli diyagramlarla gösterilir.

UML 'in doğuşu son yıllarda yazılım endüstrisindeki en büyük gelişmelerden biri olarak kabul edilebilir. UML 1997 yılında yazılımın, diyagram şeklinde ifade edilmesi için bir standartlar komitesi tarafından oluşturuldu. Daha önce hemen her daldaki

mühendislerin standart bir diyagram çizme aracı vardı. Ve şimdi de programcıların UML 'si var.

UML ile hazırlanmış bir yazılım hem daha az maliyetli hem daha etkili ve daha uzun ömürlü olur. UML ile dokümantasyonu yapılmış bir programın sonradan düzenlenmesi daha kolay olur. Bütün bunlar UML kullanmamız için yeterli sebeplerdir. Kısaca, UML'nin faydalarını maddeler halinde sıralarsak;

**1-)** Öncelikle programımız kodlanmaya başlamadan önce geniş bir analizi ve tasarımı yapılmış olacağından kodlama işlemi daha kolay olur. Çünkü programdan ne beklediğimizi ve programlama ile neler yapacağımızı profesyonel bir şekilde belirleriz UML ile.

**2-)** Programımızda beklenmedik bir takım mantıksal hataları minimuma indirgemiş oluruz.

**3-)** Tasarım aşaması düzgün yapıldıysa tekrar kullanılabilen kodların sayısı artacaktır. Bu da program geliştirme maliyetini büyük ölçüde düşürecektir.

**4-)** UML diyagramları programımızın tamamını kapsayacağı için bellek kullanımını daha etkili hale getirebiliriz.

**5-)** Programımızın kararlılığı artacaktır. UML ile dokümanlaştırılmış kodları düzenlemek daha az zaman alacaktır.

**6-)** Ortak çalışılan projelerde programcıların iletişimi daha kolay hale gelir. Çünkü UML ile programımızı parçalara ayırdık ve parçalar arasında bir ilişki kurduk.

Bir sistemin geliştirilmesi kabaca aşağıdaki aşamalardan geçmektedir. İlk iki aşamada UML büyük ölçüde rol oynar.




Şekil 3.1. Yazılım Geliştirme Aşamaları

UML bir programlama dili değildir. Bir diyagram çizme ve ilişkisel modelleme dilidir. Bu diyagramlara kısaca göz atmak gerekirse;

### 3.2.1 Sınıf diyagramları

Gerçek dünyada eşyaları nasıl araba, masa, bilgisayar şeklinde sınıflandırıyorsak yazılımda da birtakım benzer özelliklere ve fiillere sahip gruplar oluştururuz. Sınıf diyagramları kendi baslarına son derece yalın yapılardır ve görsel olarak çizilmeleri, ancak aralarındaki ilişkilerle birlikte incelendiklerinde anlamlıdır. UML'deki sınıf diyagramları sistemin durağan yapısını tanımlamak için kullanılır.

### 3.2.2 Nesne diyagramları

Bir nesne(object) sınıfın (class) bir örneğidir. Bu tür diyagramlarda sınıfın yerine gerçek nesneler kullanılır. Sınıf diyagramlarının doğruluğunun test edilmesi için kullanılan nesne diyagramları sistemin belirli bir andaki durumunu gösterirler.

### 3.2.3 Durum diyagramları

Gerçek nesnelerin herhangi bir zaman içindeki durumunu gösteren diyagramlardır. Mesela, Ali nesnesi insan sınıfının gerçek bir örneği olsun. Ali'nin doğması, büyümesi, gençliği ve ölmesi Durum Diyagram'larıyla gösterilir.

### 3.2.4 Sıra etkileşim diyagramları

Sınıf ve Nesne diyagramları statik bilgiyi modeller. Hâlbuki gerçek zamanlı sistemlerde zaman içinde değişen aktiviteler bu diyagramlarla gösterilemez. Bu tür zamanla değişen durumları belirtmek için sıra etkileşim diyagramları kullanılır.

### 3.2.5 Etkinlik diyagramları

Çok parçacıklı uygulamalar geliştiriliyor ve bu uygulamaların algoritmik mantığı UML ile gösterilmek isteniyorsa bu diyagramlar kullanılmaktadır. İş akışını, bir süreci

veya karmaşık bir algoritmayı UML ile modellemek için yine bu diyagramlar kullanılabilir. Etkinlik diyagramları sayesinde is süreçleri gözden geçirilerek, gereksiz şekilde ardışık yapılan işlemlerin paralel olarak yapılabileceği belirlenebilir. Ayrıca; bir veya daha fazla kullanım senaryosunun ayrıntıları bu diyagramlar sayesinde ortaya konulabilir.

Bir nesnesinin durumu zamanla kullanıcı tarafından ya da nesnenin kendi içsel işlevleri tarafından değişebilir. Bu değişim sırasını etkinlik diyagramlarıyla gösteririz.

### 3.2.6 Kullanım durum diyagramları

Programımızın davranışının bir kullanıcı gözüyle incelenmesi Kullanım Durum diyagramlarıyla yapılır. Gerçek dünyada insanların kullanacağı bir sistemde bu diyagramlar büyük önem taşırlar. Diğer bir deyişle üzerinde çalıştığımız uygulama ile uygulamadan etkilenen dış kullanıcılar (aktör) arasında geçen iş hedeflerine uygun bir dizi etkileşime kullanım durumu denir. Kullanım Durumları sistemin kabaca ne yaptığı ile ilgilenir, kesinlikle nasıl ve neden yapıldığını incelemez.

### 3.2.7 Bileşen diyagramları

Özellikle birden çok geliştiricinin yürüttüğü projelerde sistemi bileşen dediğimiz parçalara ayırmak, geliştirmeyi kolaylaştırır. Sistemi öyle modellememiz gerekir ki her geliştirici ötekinden bağımsız olarak çalışabilsin. Genel olarak; tek basına çalışabilen alt modüller bileşen olarak düşünülebilir. Bu diyagramlarla bileşenler arasındaki bağımlılıklar belirlenebilir ve bileşenler arasındaki organizasyon net olarak ortaya konabilir. Bu tür modellemeler Bileşen Diyagramlarıyla yapılır.

### 3.2.8 Yaygınlaştırma diyagramları

Bu tür diyagramlarla sistemin fiziksel incelenmesi yapılır. Mesela bilgisayarlar arasındaki bağlantılar, programın kurulacağı makineler ve sistemimizdeki bütün aletler Yaygınlaştırma Diyagramında gösterilir.

### 3.2.9 İletişim diyagramları

Önceki sürümlerde işbirliği diyagramları (collaboration) olarak adlandırılan diyagramlardır. Bir sistemin amacının yerine gelmesi için sistemin bütün parçaları işlerini yerine getirmesi gerekir. Bu işler genellikle birkaç parçanın beraber çalışmasıyla mümkün olabilir. Bu tür ilişkileri göstermek için iletişim diyagramları kullanılır. Sistemin durağan ve dinamik davranışlarını birlikte gösterir.

### 3.2.10 Zamanlama diyagramları

Zamanla bir veya birden fazla öğenin değer veya durum değişimini göstermek için kullanılır. Zamanlanmış olaylar, zaman ve onları etkileyen kısıtlamaların sürekliliği arasındaki etkileşimi de göstermektedir.

### 3.2.11 Etkileşim tanıtma diyagramları

Etkinlik diyagramlarının etkileşimi gösteren kısımlarını içermektedir. Hemen hemen etkinlik diyagramları ile aynı olmakla birlikte iki yeni özelliği etkileşim olayları ve etkileşim öğelerini bu diyagramlara eklemesidir.

### 3.2.12 Karma yapı

Bu diyagramlar bir sınıflandırıcının dâhili yapısını sistemin diğer parçaları ile olan etkileşim noktalarını belirterek göstermek için kullanılır. Sınıflandırıcıyı içeren

davranışı gerçekleştirmek için beraber olan parçalarının ilişki ve konfigürasyonunu gösterir.

### 3.2.13 Paket diyagramları

Paketler ve öğelerinin organizasyonunu yansıtmak amacıyla kullanılır, isim uzaylarının görselleştirilmesini sağlar, hem fiziksel hem de mantıksal ilişkileri göstermek için kullanılabilir.

UML Diyagramları bir sistemin farklı yönlerini ve özelliklerini gösteren diyagramlardır. Bir sistemin modellenmesinde sistem hangi yönden incelenmek isteniyorsa ona uygun diyagram seçilir ve çizilir.

# 4 SANAL KAMPÜSÜN MODELLENMESİ

Bir yazılım sistemi oluşturulurken sadece tek boyutta analiz ve modelleme yapılmaz. Bu amaçla; yazılım geliştirme sürecinin farklı aşamalarında farklı UML diyagramlarını kullanmak gerekmektedir. 4+1 bakış, bu diyagramları sınıflandırmak ve yazılım yaşam çevrimindeki kullanım yerlerini ortaya koymak için kullanılan bir kavramdır. Bu bakışlar;

1-) Kullanıcı Bakışı (User View): Müşteri gereksinimlerini ortaya koymak ve müşteriye, sistemi tanıtmak amacı ile kullanılan bakış açısıdır. Bazı kaynaklarda; kullanım senaryosu (use case) bakışı olarak da açıklanmaktadır. Bu amaçla; "kullanım durum" (use case) diyagramları kullanılmaktadır.

2-) Yapısal Bakış (Structural View): Sistemin nelerden meydana geldiğini gösteren bakış açısıdır. Bazı kaynaklarda bu bakış, tasarım (design) veya mantıksal (logical) bakış olarak da açıklanmaktadır. Bu amaçla; "sınıf" (class) diyagramları ve "nesne" (object) diyagramları kullanılmaktadır.

3-) Davranış Bakışı (Behavioral View): Sistemin dinamik yapısını ortaya koyan bakış açısıdır. Bazı kaynaklarda; süreç (process) bakışı olarak da açıklanmaktadır. Bu amaçla; "ardışık sıra" (sequence), "iletişim" (communication), "durum" (statechart), "etkinlik" (activity) diyagramları kullanılmaktadır.

4-) Gerçekleştirme Bakışı (Implementation View): Sistemin alt modüllerini ortaya koyan bakış açısıdır. Bazı kaynaklarda; bileşen (component) bakışı olarak da açıklanmaktadır. Bu amaçla; "bileşen" (component) diyagramları kullanılmaktadır.

5-) Ortam Bakışı (Environment View): Donanımın (üzerinde yazılımın çalışacağı), fiziksel mimarisinin ortaya konduğu bakış açısıdır. Bazı kaynaklarda; yaygınlaştırma (deployment) bakışı olarak da açıklanmaktadır. Bu amaçla "yaygınlaştırma" (deployment) diyagramları kullanılmaktadır.

## 4.1 1. Aşama: Kullanıcı Bakışı

Kullanıcı bakışında ilk olarak projenin amacını ortaya koymak gerekir. Sanal Kampus projesinin amacı, örgün eğitim veren bir kurumun İnternet tabanlı olarak öğrencilerine uzaktan eğitim verebilmesidir. Öğrenciler eşzamanlı veya eşzamansız olarak ve istediği yerde internet üzerinden eğitim sitesine bağlanarak eğitimini alabilmesi ve öğretmenlerin de öğrenciler ile aynı zamanda ve aynı mekânda olmak zorunda kalmadan sistem üzerinden oluşturduğu dokümanlar ile eğitimini verebilmesi amaçlanmaktadır.

### 4.1.1 Aktörler

Bu genel yaklaşımdan sonra öncelikle sistemin temel aktörlerini yani uzaktan eğitim sistemimizin temel katılımcılarını belirlemek ve aktörlerin rollerini ortaya çıkarmak gerekir. Aktörlerin belirlenmesi önemlidir çünkü iş akışlarında aslında hiçbir şey kendiliğinden olmaz. Her olayı başlatan bir aktör bulunur. Bazen bu aktör belli bir algoritmaya göre çalışan bir yazılım bileşeni de olabilir ancak genellikle aktör bir bireydir. Uzaktan eğitim sistemin de kullanıcı gurupları açıkça birbirinden ayrılmaktadır. Bu çalışmada üzerinde durulan kullanıcı gurupları yani aktörler;

- Öğrenci
- Öğretmen
- Yönetici
- Alt Sistem

Bu aktörlerin sistemle her etkileşimlerinde farklı bir amaçları olabilir. Ancak aynı anda sadece bir amacı olduğunu varsayarız. Aktörlerin birçok özellikleri olduğundan bir şablon kullanmak anlaşılabilirliği daha fazla kolaylaştıracaktır.

AKTÖR: ÖĞRENCİ

[SANAL KAMPUS ÖĞRENCİSİ]

| Adı | Sanal Kampus Öğrencisi |
|---|---|
| Aktör Tanımlayıcısı | ÖĞRENCİ |
| Rol (ler) | - Öğrenci, sanal kampüse eğitim almak için oturum açar. (Kullanım Durum No:1)<br>- Öğrenci, girdiği bölüme dönem kaydını yapar. (Kullanım Durum No:9)<br>- Öğrenci, yatırmış olduğu harç paralarını sistemden görür. (Kullanım Durum No:10)<br>- Öğrenci, dönem kaydı yaptırdığı derslerden birini dokümanlarına erişmek veya sınav olmak için seçer. (Kullanım Durum No:7)<br>- Öğrenci, seçmiş olduğu Dersin dokümanlarına erişir. (Kullanım Durum No:11)<br>- Öğrenci, ders notlarını çalışırken bölüm sonlarında ara sınavlar olur. (Kullanım Durum No:12)<br>- Öğrenci, ders sonunda genel sınav olur. (Kullanım Durum No:13)<br>- Öğrenci, derslerden almış olduğu sınav sonuçlarına bakar. (Kullanım Durum No:14)<br>- Öğrenci, öğretmenin ders için oluşturduğu değerlendirme formunu doldurur. (Kullanım Durum No:15)<br>- Öğrenci, yardım menüsünden sistemin işleyişiyle ilgili yardım alır. (Kullanım Durum No:8)<br>- Öğrenci, bilgilerini günceller. (Kullanım Durum No:6)<br>- Öğrenci, diğer öğrencilere, öğretmenlerine veya yöneticiye mesaj yazıp gönderir. (Kullanım Durum No:4)<br>- Öğrenci, kendisine gelen mesajları okur. (Kullanım Durum No:5)<br>- Öğrenci, site içi bilgi arama yapar. (Kullanım Durum No:3)<br>- Öğrenci, şifresini değiştirir. (Kullanım Durum No:2)<br>- Öğrenci, görüntülü konferansa katılır. (Kullanım Durum No:25) |
| İş Bilgisi | Düşük, İş Akışı Teknik terimler ve sistemin yapısı hakkında fazla bilgisi yoktur. |
| Teknik Bilgisi | Orta, Bilgisayar bilgisi vardır. Sanal Kampus üzerinde bulunan özellikleri kullanabilir fakat sistemin işleyişi hakkında yardım sağlanmalıdır. |
| Diğer Özellikler | Sanal Kampus öğrencisidir. Sistemde bilgileri mevcuttur. Sisteme giriş yapabilir. |
| Sayı | Hedeflenen Sanal Kampüse kayıtlı öğrenci sayısı, ilk yıl için 5 bin, ikinci yıl için 10 bin ve bu sayının 50 bine ulaşması öngörülmektedir. |

Çizelge 4.1 Öğrenci Aktörü Bilgileri

AKTÖR: ÖĞRETMEN

[SANAL KAMPUS ÖĞRETMENİ]

| Adı | Sanal Kampus Öğretmeni |
|---|---|
| Aktör Tanımlayıcısı | ÖĞRETMEN |
| Rol (ler) | - Öğretmen, Sanal kampüse eğitim vermek için oturum açar. (Kullanım Durum No:1)<br>- Öğretmen, Üzerine atanan derslerden birini dokümanlar girmek, sınav soruları hazırlamak ve ders değerlendirme formu oluşturmak için seçer. (Kullanım Durum No:7)<br>- Öğretmen, Seçmiş olduğu derse doküman ekler, siler, düzenler. (Kullanım Durum No:16)<br>- Öğretmen, Ders sınav soruları girer. (Kullanım Durum No:17)<br>- Öğretmen, Dönem sonu ders değerlendirme formu hazırlar. (Kullanım Durum No:18)<br>- Öğretmen, Öğrencilerin sınav sonuç listelerine bakar. (Kullanım Durum No:19)<br>- Öğretmen, Yardım menüsünden sistemin işleyişiyle ilgili yardım alır. (Kullanım Durum No:8)<br>- Öğretmen, Bilgilerini günceller. (Kullanım Durum No:6)<br>- Öğretmen, diğer Öğretmenlere, öğrencilerine veya yöneticiye mesaj yazıp gönderir. (Kullanım Durum No:4)<br>- Öğretmen, kendisine gelen mesajları okur. (Kullanım Durum No:5)<br>- Öğretmen, site içi bilgi arama yapar. (Kullanım Durum No:3)<br>- Öğretmen, şifresini değiştirir. (Kullanım Durum No:2)<br>- Öğretmen, öğrencileri ile görüntülü, sesli yazılı konferans yapar. (Kullanım Durum No:25) |
| İş Bilgisi | Düşük, İş Akışı Teknik terimler ve sistemin yapısı hakkında fazla bilgisi yoktur. |
| Teknik Bilgisi | Orta, Bilgisayar bilgisi vardır. Sanal Kampus üzerinde bulunan özellikleri kullanabilir fakat sistemin işleyişi hakkında yardım sağlanmalıdır. |
| Diğer Özellikler | Sanal Kampus Öğretmenidir. Sistemde bilgileri mevcuttur ve işleyişin vazgeçilmez bir parçasıdır. Sisteme giriş yapabilir. |
| Sayı | Hedeflenen Sanal Kampüse kayıtlı Öğretmen sayısı, ilk yıl için her dersten sorumlu 1 Öğretmen, ikinci yıl 1 den fazla ve bu sayının her dersi farklı yöntemlerle öğrenciye ulaştıracak en az 3 Öğretmene ulaşması öngörülmektedir. |

Çizelge 4.2 Öğretmen Aktörü Bilgileri

AKTÖR: YÖNETİCİ

[SANAL KAMPUS YÖNETİCİSİ]

| Adı | Sanal Kampus Yöneticisi |
|---|---|
| Aktör Tanımlayıcısı | YÖNETİCİ |
| Rol (ler) | - Yönetici, Sanal kampüse destek vermek için oturum açar. (Kullanım Durum No:1)<br>- Yönetici, Sanal kampüse öğrenci Kayıt ekler, siler ve düzenler. (Kullanım Durum No:21)<br>- Yönetici, Sisteme Öğretmen ekler, siler ve düzenler. (Kullanım Durum No:23)<br>- Yönetici, Sisteme Ders ekler, siler ve düzenler. (Kullanım Durum No:22)<br>- Yönetici, Sisteme Bölüm ekler, siler ve düzenler. (Kullanım Durum No:20)<br>- Yönetici, Sisteme Fakülte ekler, siler ve düzenler. (Kullanım Durum No:27)<br>- Yönetici, Sisteme Üniversite ekler, siler ve düzenler. (Kullanım Durum No:26)<br>- Yönetici, Öğretmenler üzerine sorumlu oldukları dersleri atar. (Kullanım Durum No:24)<br>- Yönetici, Bilgilerini günceller. (Kullanım Durum No:6)<br>- Yönetici, diğer Yöneticilere, öğrencilere veya öğretmenlere mesaj yazıp gönderir. (Kullanım Durum No:4)<br>- Yönetici, kendisine gelen mesajları okur. (Kullanım Durum No:5)<br>- Yönetici, site içi bilgi arama yapar. (Kullanım Durum No:3)<br>- Yönetici, şifresini değiştirir. (Kullanım Durum No:2)<br>- Yönetici, yönetim kurulu kararıyla öğrenci notu düzeltme yapar. (Kullanım Durum No:30)<br>- Yönetici, mezuniyet durumuna gelmiş öğrencileri mezun eder. (Kullanım Durum No:29) |
| İş Bilgisi | Yüksek, İş Akışı Teknik terimler ve sistemin yapısı hakkında yeterli bilgisi vardır. |
| Teknik Bilgisi | Yüksek, Bilgisayar bilgisi vardır. Sanal Kampus üzerinde bulunan tüm özellikleri kullanabilir ve sistemin işleyişi hakkında yardım sağlamaktadır. |
| Diğer Özellikler | Sanal Kampus Yöneticisidir. Sistemde bilgileri mevcuttur ve işleyişin vazgeçilmez bir parçasıdır. Sistemin düzenli işlemesinden sorumludur |
| Sayı | Hedeflenen Sanal kampüse kayıtlı Yönetici sayısı, ilk aşamada 15 Yönetici ve bu sayının 100 e ulaşması öngörülmektedir. |

Çizelge 4.3 Yönetici Aktörü Bilgileri

AKTÖR: ALT SİSTEM

[SANAL KAMPUS ALT SİSTEMİ]

| Adı | Sanal Kampus Alt Sistemi |
|---|---|
| Aktör Tanımlayıcısı | ALT SİSTEM |
| Rol (ler) | - Alt Sistem, Soru bankasından sınav oluşturur. (Kullanım Durum No:32)<br>- Alt Sistem, Sınav sonunda öğrencinin notunu hesaplar. (Kullanım Durum No:31)<br>- Alt Sistem, mezuniyet durumuna gelen öğrenci listesi oluşturur. (Kullanım Durum No:33) |
| İş Bilgisi | |
| Teknik Bilgisi | |
| Diğer Özellikler | Yordamlardan oluşan bir bileşendir. |
| Sayı | |

Çizelge 4.4 Alt Sistem Aktörü Bilgileri

### 4.1.2 Kullanım durumu diyagram ve senaryoları

Aktörler ve rollerini ortaya çıkardıktan sonra ilk olarak bu roller için Kullanım Durum Senaryolarını ve diyagramlarını ortaya koymak gerekir. Kullanım durum senaryoları sistemin işlevsel gereksinimlerine denk gelmektedirler. Bu gereksinimleri iyi belirlemek yazılım sisteminin talep edilen işlevselliğini doğru bir şekilde anlamayı sağlar. Öğrenci sistemde hizmet alan yani müşteri konumundaki tek kullanıcı gurubudur. Diğer kullanıcı gurupları öğrenciye iyi bir eğitim vermek için çalışan kullanıcı guruplarıdır. Bu yüzden sistemin temel kullanıcısı olarak öğrenci kullanıcısını düşünebiliriz.




Şekil 4.1. Sanal Kampus Kullanım Durumu Genel Diyagramı

Bu diyagram ile sanal kampüsümüzde bulunan aktörler ve onların rolleri genel olarak tek bir kullanım durum diyagramında gösterilmektedir. Aktörlerin kullandıkları ortak kullanım diyagramları bu gösterimde belirtilmektedir. Kullanım durumları arasındaki ilişkileri inceleyebilmek için her aktörü ve rollerini ayrı bir kullanım durum diyagramı ile göstermek gerekir.




Şekil 4.2. Öğrenci Aktörü Kullanım Durum Diyagramı

Öğrenci aktörü için kullanım durumları değerlendirilecek olursa, bazı senaryolar içerisinde diğer kullanım durum senaryolarını da barındırabilmektedir. Mesela, "Şifre değiştirme" kullanım durum senaryosu içinde "Oturum Açma" senaryosunu da barındırmaktadır. Yani öğrenci şifre değiştirebilmek için oturum açması gerekir. Aynı şekilde "Görüntülü Konferans" senaryosu içerisinde "Ders Seçme" ve "Oturum Açma" senaryolarını içinde barındırır.



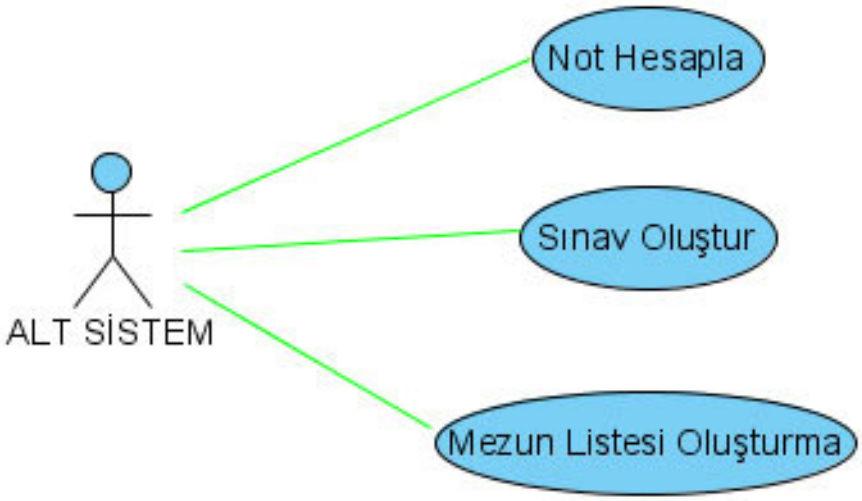

Şekil 4.3. Öğretmen Aktörü Kullanım Durum Diyagramı

Öğretmen aktörü içinde öğrenci aktörü için geçerli olan durumların birçoğu geçerlidir. Mesela bütün kullanım durumları içerisinde "Oturum Açma" senaryosunu barındırır. Oturum açma sistemin güvenliği için ve uzaktan eğitimin sahip olması gereken özelliklerden biri olduğu için önemlidir. Bazı Kullanım durumları sadece Öğretmen aktörüne ait bazıları ise diğer aktörler için de farklı zamanlar ve farklı durumlar için kullanılabilir fakat "Görüntülü Konferans" durum senaryosu hem öğretmen hem de öğrenci için aynı anda kullanılan bir senaryodur. Bu durumlar düşünülerek senaryo ortaya çıkarılmalıdır.




Şekil 4.4. Yönetici Aktörü Kullanım Durum Diyagramı

Yönetici aktörünün uzaktan eğitimde eğitim ve öğretim açısından bir etkinliği olmasa da sistemin sağlıklı yürüyebilmesi, sanal kampus sabitlerinin oluşturulması, düzenlenmesi ve diğer aktörlere destek vermesi açısından önemi büyüktür.




Şekil 4.5. Alt Sistem Aktörü Kullanım Durum Diyagramı

Alt Sistem veritabanında veya yazılım içinde oluşturulmuş bir fonksiyon veya bir yordam gurubunun tamamıdır fakat kullanım durum senaryolarını çıkarırken alt sistemi de bir aktör gibi düşünmekte fayda vardır. Bu yordamların çalışabilmesi için başka bir aktör mutlaka tetiklemelidir. Mesela "Sınav Oluştur" durumu öğrencinin dersi seçip sınavı başlat demesiyle tetiklenir ama sınavı oluşturmak bu alt sistemin bir yordam veya fonksiyonun işidir.

Aktörleri ve kullanım durum diyagramlarını ayrıntılı inceledikten sonra kullanım durum senaryolarını oluşturmak gerekir. Bu senaryoları hazırlarken bazı şeylere dikkat etmek gerekecektir. Öncelikle senaryoların belli bir şablonu olmalıdır. Kolay anlaşılabilir olmalıdır. Projelerin büyüklüğüne göre, her senaryo için ayrı bir uzmanın üzerinde çalışması gerekmektedir.

Gereksinimlerimizin sayısı arttıkça bunları sistematik bir biçimde ele almak da zorunlu bir hale geliyor. Bu yüzden bütün senaryolarımız da belirli kurallar belirleyip bu kurallar doğrultusunda bir şablon ortaya koymak gerekir.

<SN#> <Senaryo İsmi>

| | |
|---|---|
| Senaryo # | <SN#> |
| Senaryo İsmi | <Senaryo İsmi> |
| Kapsam | |
| Hedef Seviye | |
| Aktör(ler) | |
| Hedef | |
| Amaç | |
| Olgunluk | |
| Sorular | |
| Ön Durumlar | |
| Başarılı bitiş durumları | |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| |
| Alternatif İş Akışları |
| 1a. Koşul:<br>1a. Adım 1<br>1a. Adım 2 |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| | | | |

Çizelge 4.5 Kullanım Durum Senaryosu Şablonu

**Şablon açıklamaları:**

**Senaryo başlığı**

Senaryo başlığı proje ismi ve senaryonun kısaltılmış hali ile senaryonun numarasından ve isminden oluşur. Ek olarak en sona sürüm numarasıda eklenebilir fakat çokta gerekli değildir.Yani:

proje ismi + SN + senaryo numarası + Senaryo ismi + (sürüm no)

Örneğin "SANALKAMPUS.SN1 Sınav Ol" senaryo başlığıdır. Proje ismi ve SN1 nokta işareti ile birbirinden ayrılmıştır.

**Senaryo #**

Senaryonun numarasını tutar. Her senaryoya birden başlayarak bir numara vermek gerekir

**Senaryo İsmi**

Senaryonun ismi burada belirtilir. Açıklayıcı bir kelimeyi takiben bir fiilden oluşur. Birincil Aktörün yapmak istediği işi belirtir. Örneğin "Yeni Müşteri Oluşturma", "Sınav Olma" vs.

**Kapsam**

Senaryonun kapsamı belirtilir. Kapsamlar, üzerinde tartışılan sistemlerde olabilir (İş Süreci, Sistem, Alt-Sistem). Bu üç sistem için oluşturulan Senaryo'ları Şeffaf-Kutu (white-box), yada Kara-Kutu (black-box) olarak tanımlamamızda gerekir. Şeffaf-kutu olan senaryoların girdi ve çıktı'ları ile içinde geçen tüm işlemler ve veri yapısı tamamen Aktör'ler tarafından bilinir. Öbür taraftan Kara-Kutu senaryolarının sadece girdileri ve çıktıları bilinir.

**Hedef seviye**

Hedef Seviye, senaryonun 3 aşamalı senaryolandırma sınıflarından hangisine ait olduğunu söyler (Genel Senaryolar, Müşteri Hedefleri, Detay Fonksiyonlar). Senaryoları bu şekilde sınıflandırmanın amacı, analizin belli bir düzen içerisinde olmasını sağlamak içindir. Konu hakkında genel bilgi sahibi olmak isteyen yönetici ekibi sadece Genel Senaryo sınıfındaki senaryoları okuyarak işin ne olduğunu kavrayabilirler.

Genel Senaryolar, sistemi tanımlayıcı 3 ana görevi yerine getirir.

Büyük sistem içerisinde Müşteri Hedeflerinin nereye oturduğunu gösterir.

Müşteri Hedeflerinin hayat döngülerini belirtir.

Bir kitabın içindekiler bölümü gibi daha alt seviye senaryoların başlıklarını belirtir

Müşteri Hedefleri ise gerçekte birincil aktörlerin yapmak istedikleri işleri belirtir. Müşteri Hedefi olan bir senaryo "Aktör bu senaryoyu uyguladıktan sonra sistemden mutlu bir şekilde ayrılacak mı?" sorusuna "Evet" yanıtını vermeye çalışmaktadır.

Detay Fonksiyonlar, Müşteri Hedeflerinin yerine getirilmesi için atılacak adımları belirler. Detay fonksiyonları sadece gerçekten ihtiyaç varsa eklenmelidir. Örneğin "Sisteme Oturum Aç", "Öğrenci bul", "Ders seç" gibi işlemler birer detay fonksiyondur.

**Aktörler**

Aktör, senaryo ile ilişkili olan veya senaryoyu kullanan kişi veya sistemdir. Bir senaryoyu başka bir sistemde kullanabilir. Her senaryo için en azından bir adet birincil aktör olmalıdır. Eğer senaryoyu kullanan başka aktörler de varsa onlarıda sıralamak gerekir.

**Hedef**

Senaryonun hedefini belirtir. Sadece, senaryo ile yapılması amaçlanan iş anlatılır. Senaryo ne işe yarar sorusuna cevap verir. Bir paragraftan fazla olmamalı ve senaryoyu okuyan kişiye genel bir bilgi vermelidir. Detay bilgi barındırmaz.

**Amaç**

Senaryonun sistemde nereye oturduğunu söyler. Bir işi veya sistemi analiz ederken, o sistemi parçalara bölmek ve küçük parçalar halinde ele almak bize zaman kazandırır. Bu açıdan bakıldığında her senaryo büyük sistemin bir parçasıdır ve genelde diğer senaryolar ile ilişki içerisindedir. Ayrıca "Amaç" senaryonun sistem için neden önemli olduğunu da söyler.

**Olgunluk**

Olgunluk senaryonun hangi süreçte olduğunu gösterir. Renkler ile kodlama UML dilinde oturmuş bir gösterim biçimidir.

Kırmızı: İş süreci senaryosu tanımlanır fakat henüz detaylandırılmamıştır.

Turuncu: Senaryonun iş akışı şekillenmeye başlar ve sorular bölümünde sorular belirmiştir. Turuncu seviyesi biraz uzun sürdüğü için Turuncu1 Turuncu2... biçiminde çoğaltılabilir.

Sarı: Senaryo, kontrol mekanizmaları için yayınlanabilir bir hale gelmiştir. Senaryonun bitirilebilmesi için temel adımlar ortaya çıkarılmış ve alternatif adımlardan önemli olanlar tamamen genişletilmiştir. Ayrıca senaryonun büyük sistem içerisindeki durumuda güvenli bir hale gelmiştir (Gerçekten sistemin bu senaryoya olan ihtiyacı ortaya çıkartılmıştır).

Yeşil: Senaryo bir sonraki süreç için hazırdır, fakat henüz onaylanmamıştır. Şablondaki tüm sahalar doldurulmuş ve Soru kısmında hiç bir soru kalmamıştır.

Mavi: Senaryo onaylanmış ve bitmiştir. Son Gereksinim Modeli'nde yerini alır.

Gümüş: Senaryo başka bir projede yeniden kullanılmıştır. Küçük değişiklikler yapılmış olabilir.

Altın: Senaryo birden fazla başka projede değişiklik yapılmadan kullanılmıştır.

**Sorular**

Senaryonun tamamlanabilmesi için ortaya çıkması gereken konuların sorulduğu bölümdür. Yeşil konumuna gelmiş bir senaryonun Sorular kısmında hiç bir soru olmaması gerekir.

**Ön durumlar**

Senaryonun işlemeye başlayabilmesi için gerekli ön durumları belirtir. Ön durumlar gerçeklenmeden senaryo işleme başlayamaz. Numaralı bir liste şeklinde ve hiyerarşik bir yapıda olması gerekir.

**Başarılı bitiş durumları**

Senaryo bittikten sonra, sistemin alacağı durumdur. Ana iş akışı sonunda yada Alternatif iş akışları sonucunda oluşacak her türlü başarılı bitiş durumu burada numaralı liste biçiminde not edilir.

**Minimum bitiş durumu**

Minimum bitiş durumları, senaryonun bitiş durumu ne olursa olsun (başarılı yada başarısız) her zaman doğru olacak durumlardır. Örnek olarak bir senaryo başarılı veya

başarısız biçimde sonlanırsa, senaryonun çalıştırıldığına dair sonuç kütüklerine bir kayıt eklenir. Kütüklere kayıt ekleme işlemi her iki durumda da geçerlidir.

**Ana iş akışı**

Senaryonun hedefine ulaşabilmesi için gerekli adımların sıralanması ile oluşturulur. Bu akış aktör ile sistem arasında geçen konuşmadır. Senaryonun akışı ön koşullardan sonlanma koşullarına doğru olmalıdır. Her hangi bir hata durumunu barındırmaz. Her türlü ön koşulun ve tahmin edilemez hataların ortaya çıkmayacağı var sayılır.

Her adım için:

Bir hedefin başarıldığı gösterilir

Aktörün nasıl tepki verdiğini yakalamaya çalışmak lazım, kullanıcı arayüzünü değil.

Her adım bir aktör ile başlar (Aktör şu işlemi yapar, Sistem şu bilgiyi gönderir)

Aktörlerin isimleri ile kullanılmasına özen göstermek gerekir, çünkü aynı işi yapan birden fazla aktör olabilir.

**Alternatif iş akışları**

Ana iş akışında belirli durumlara göre sapmalar olabilir veya parametrik yapılarda işlemler parametrelere göre değişebilir veya senaryodaki problemin başka bir çözüm yolu olabilir. Buna göre alternatif çözüm yollarını ve hata durumlarındaki senaryonun vereceği yanıtı bu kısımda ele alıyoruz.

Mesela 3 adımdan oluşan bir ana iş akışımız var. Birinci adım için 2 adet alternatif çözümümüz olsun. İlk çözümü 1a olarak isimlendireceğiz. İkinciyi de 1b. Alternatif çözümlerin adımlarına 1a1, 1a2 de diyebiliriz fakat çok fazla karıştırmamak için baştaki adım numarasını düşürüyoruz.

1a. Ana akışın 1. adımına alternatif olabilecek 1. durum

a1. Adım 1

a2. Adım 2

1b. Ana akışın 1. adımına alternatif olabilecek 2. durum

b1. Adım 1

b2. Adım 2

3a. Ana akışın 3. adımına alternatif olabilecek 1. durum

a1. Adım 1

a2. Adım 2

Yukarıdan da anlaşıldığı gibi üçüncü adım içinde bir adet alternatif çözümümüz mevcut.

**Belge Tarihçesi**

Hazırladığımız senaryo üzerinde değişiklik yapmamız gereken durumlar olabilir. Olgunluk dönemi kırmızıdan maviye ilerlerken önemli bulduğumuz senaryo değişiklikleri veya yeni eklentileri belirtiriz.

Farklı projeler için farklı şablonlar kullanmamız gerekli olabilir fakat bir proje içinde şablon değiştirmemek bütünlüğü korumak ve anlaşılabilir olmak için gereklidir. Kullandığımız şablonun değiştirme kayıtlarını aşağıdaki tablodadır.

| Şablon değiştirme kayıtları | | | |
|---|---|---|---|
| Tarih | Sürüm No | Açıklama | Yazar |
| 02.09.2002 | 0.1 | İlk Sürüm | Gürkan YENİÇERİ |
| 30.04.2007 | 0.2 | Düzenleme | Sait Ali UYMAZ |

Çizelge 4.6 Kullanım Durum Senaryosu Şablon değişiklik Tablosu

Bu şablon projenin büyüklüğüne ve ihtiyaçlarına göre daha da genişletilebilir veya daha daraltılabilir bunun belirlenmesi yazılım ekibinin görevidir. Proje de kullanılacak şablon ortaya koyulduktan sonra kullanım durum senaryoları oluşturulmaya başlanacaktır.

**SANALKAMPUS.SN1 Oturum Açma**

| | |
|---|---|
| Senaryo # | 1 |
| Senaryo İsmi | Oturum Açma |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Detay Fonkiyonlar |
| Aktör(ler) | Öğrenci, Öğretmen, Yönetici |
| Hedef | Bu senaryoda kayıtlı kullanıcıların sisteme giriş yapmaları hedeflenmiştir. |
| Amaç | Sistemde farklı görevleri olan bir çok kullanıcı vardır. Bu kullanıcıların kendi işlemlerini yapabilecek sayfalarına ulaşmaları amaçlanmıştır. Sisteme giriş sadece yetki dahilinde olmalıdır. |
| Olgunluk | Sarı |
| Sorular | 1-Kullanıcı şifrelerinin belli bir uzunlukta ve formatta olması gerekli midir?<br>Güvenliği arttırmak için gerekli olabilir. |
| Öndurumlar | Bir Öğrencinin sisteme giriş yapabilmesi için harcını yatırmış olması gerekir.<br>Bir Öğretmenin sisteme giriş yapabilmesi için üzerine ders tanımlanmış olması gerekir. |
| Başarılı bitiş durumları | Sistemde oturum açma gerçekleşir. |
| Minimum bitiş durumları | Sisteme girme denemesi log dosyalarına kayıt edilir |
| Ana İş Akışı | |
| 1. Kullanıcı, kullanıcı adını ve şifresini girer.<br>2. Sistem kullanıcı adı ve şifreyi veritabanından kontrol eder ve doğrulanırsa aktörün yapabileceği işlemler menüsüne geçer.<br>3. Kullanıcıya özel oturum açar. | |
| Alternatif İş Akışı | |
| 2A. Koşul: Kullanıcı hatalı kullanıcı adı veya şifre girer.<br>A2. Adım 1 Sistem Kullanıcı adı ve/veya şifrenin hatalı olduğuna dair uyarır ve kullanıcı adı ve şifreyi yeniden ister.<br>A2. Adım 2 Kullanıcı bilgileri yeniden girer. | |

| | | | |
|---|---|---|---|
| A2. Adım 3 Bilgiler doğru ise ana akışa dönülür.<br>A2. Adım 4 Tekrar yanlış girilirse yeniden sorulur.<br>A2. Adım 5 Beş kez yanlış girilirse akış sona erdirilir. | | | |
| Belge Tarihçesi | | | |
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 01/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |
| 07/Mayıs/2007 | 0,1 | Alternatif iş akışına 5. madde eklendi | Sait Ali Uymaz |

Çizelge 4.7 Oturum Açma Kullanım Durum Senaryosu

**SANALKAMPUS.SN2 Şifre Değiştirme**

| | |
|---|---|
| Senaryo # | 2 |
| Senaryo İsmi | Şifre Değiştirme |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Detay Fonkiyonlar |
| Aktör(ler) | Öğrenci, Öğretmen, Yönetici |
| Hedef | Bu senaryoda kayıtlı kullanıcıların sisteme giriş şifrelerini değiştirmeleri hedeflenmiştir. |
| Amaç | Sistemde kullanıcıların güveliklerini sağlayabilmeleri için zaman zaman şifrelerini değiştirebilmeleri amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | 1. Kullanıcı şifreleri belli zaman aralıklarıyla değiştirilmeye zorlanmalımıdırlar?<br>Güvenliği arttırmak için gerekli olabilir.<br>2. Daha önce kullandığı şifreleri girmelerine izin verilmeli midir?<br>Daha önce kullanmış olduğu son 3 şifrenin kullanılmasına izin verilmeyebilir |
| Öndurumlar | Kullanıcı şifresini değiştirebilmesi için sistemde oturum açmış olması gereklidir. |
| Başarılı bitiş durumları | Şifre değiştirme başarı ile gerçekleşir. |
| Minimum bitiş durumları | Şifre değiştirme denemesi log dosyalarına kayıt edilir |

| Ana İş Akışı |
|---|
| 1. Kullanıcı, var olan şifresini girer.<br>2. Sistem şifreyi veritabanından kontrol eder ve doğrulanırsa yeni şifreyi girmesini ister.<br>3. Şifresini ikinci kez girmesini ister.<br>4. Girmiş olduğu iki şifre bir biri aynı ise ve şifre formatına uygun ise şifresini değiştirir.<br>5. Yeni şifre veritabanına işlenir. Kullanıcıya bilgi mesajı verilir. |
| Alternatif İş Akışı |

1A. Koşul: Kullanıcı var olan şifreyi hatalı girer

A1. Adım 1 Şifre hatalı uyarısını verir ve yeniden şifreyi ister

A1. Adım 2 Kullanıcı şifreyi tekrar girer

A1. Adım 3 Şifre Doğruysa ana akışa döner hatalı ise oturumu kapatır.

4A. Koşul: Kullanıcı birbiriyle eş olmayan iki yeni şifre girer.

A4. Adım 1 Sistem girilen şifrelerin aynı olmadığı uyarısını verir ve yeniden girmesini ister.

A4. Adım 2 Kullanıcı şifreleri yeniden girer.

A4. Adım 3 Şifreler aynı ise ana akışa dönülür.

A4. Adım 4 Tekrar yanlış girilirse şifreler yeniden sorulur.

4B. Koşul: Kullanıcı şifreyi belirlenen formatta girmez

B4. Adım 1 Sistem girilen şifrelerin istenen formatta olmadığı uyarısını verir ve yeniden girmesini ister.

B4. Adım 2 Kullanıcı şifreleri yeniden girer.

B4. Adım 3 Şifreler belirlenen formatta ise ana akışa dönülür.

B4. Adım 4 Tekrar yanlış girilirse şifreler yeniden sorulur.

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 01/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.8 Şifre Değiştirme Kullanım Durum Senaryosu

**SANALKAMPUS.SN3 Site içi Arama**

| | |
|---|---|
| Senaryo # | 3 |
| Senaryo İsmi | Site içi Arama |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Detay Fonkiyonlar |
| Aktör(ler) | Öğrenci, Öğretmen, Yönetici |
| Hedef | Bu senaryoda kayıtlı kullanıcıların sistemde ihtiyaç duydukları bir bilgiyi aramaları hedeflenmiştir. |
| Amaç | Sistem kullanıcılarının site içinde bulunan herhangi bir bilgiyi arayıp ona ulaşabilmeleri amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | 1. Arama sonuçları yetki dahilinde mi kullanıcıya gösterilmedir?<br>Kullanıcıya izni dahilinde olmayan bir sayfada ki arama sonuçları gösterilmemelidir. |
| Öndurumlar | |
| Başarılı bitiş durumları | Kullanıcı aradığı bilgiye sistemde mevcut ise ulaşması. |
| Minimum bitiş durumları | Sistemdeki bilgilerin içinde arama yapılabilmesi |

| Ana İş Akışı |
|---|
| 1. Kullanıcı arama yapacağı kelime veya kelimeleri yazıp arama butonuna basar.<br>2. Sistem kelimeleri veritabanında ve statik web sayfalarında arar.<br>3. Sonuçları kullanıcıya verir. |
| **Alternatif İş Akışı** |
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 02/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.9 Site İçi Arama Kullanım Durum Senaryosu

**SANALKAMPUS.SN4 Mesaj Yazma**

| | |
|---|---|
| Senaryo # | 4 |
| Senaryo İsmi | Mesaj Yazma |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Detay Fonkiyonlar |
| Aktör(ler) | Öğrenci, Öğretmen, Yönetici |
| Hedef | Bu senaryoda kayıtlı kullanıcıların diğer kullanıcılara mesaj göndermesi hedeflenmiştir. |
| Amaç | Kullanıcıların diğer kullanıcılar ile bilgi paylaşımı ve yardımlaşması amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | 1. Tüm kullanıcılar birbirlerine mesaj gönderebilmeli mi?<br>Belirli standartlar oluşturulmalı.<br>2. Sistemdeki tüm mesajlar yedeklenmeli mi?<br>İlerde doğabilecek sorunlar düşünülerek mesajlar uzun zaman saklanmalı. |
| Öndurumlar | Kullanıcı Oturum açmış olmalı. |
| Başarılı bitiş durumları | Mesaj gönderilmek istenen kişiye gönderilmiş. |
| Minimum bitiş durumları | Mesaj log dosyalarına kayıt edilir |

Ana İş Akışı

1. Kullanıcı, mesaj göndereceği kişiyi seçer.
2. Mesajını yazar ve gönder butonuna basar.
3. Sistem mesajın içeriğinin uygunluğunu kontrol eder. Uygun ise gönderir.
4. Sistem mesajı gönderilen kullanıcının veritabanındaki ilgili alanlarına kayıt eder.

Alternatif İş Akışı

3A. Koşul: Mesajın içeriği uygun değildir.

A3. Adım 1 Sistem Kullanıcıya mesajın içeriğinin kurallara uygun olmadığını iletir ve mesajı düzeltmesini ister.

A3. Adım 2 Kullanıcı mesajı yeniden yazar.

| | | | |
|---|---|---|---|
| A3. Adım 3 Mesaj kurallara uygun ise ana akışa dönülür.<br>A3. Adım 4 Tekrar mesaj kurallara uygun değilse yeniden düzeltilmesi istenir.<br>A3. Adım 5 Beş kez uygun olmayan içerik yazılırsa kullanıcının mesaj gönderme hakkı iptal edilir ve akış sona erdirilir. | | | |
| Belge Tarihçesi | | | |
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 02/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.10 Mesaj Yazma Kullanım Durum Senaryosu

**SANALKAMPUS.SN5 Mesaj Okuma**

| | |
|---|---|
| Senaryo # | 5 |
| Senaryo İsmi | Mesaj Okuma |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Detay Fonkiyonlar |
| Aktör(ler) | Öğrenci, Öğretmen, Yönetici |
| Hedef | Bu senaryoda kayıtlı kullanıcıların diğer kullanıcılardan gelen mesajları okuyabilmesi hedeflenmiştir. |
| Amaç | Kullanıcıların diğer kullanıcılar ile bilgi paylaşımı ve yardımlaşması amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Kullanıcı Oturum açmış olmalı. |
| Başarılı bitiş durumları | Gelen mesaj okunmuş. |
| Minimum bitiş durumları | Mesaj okundu bilgisi log dosyalarına kayıt edilir |

| Ana İş Akışı |
|---|
| 1. Kullanıcı, gelen mesajlarını açar.<br>2. Görmek istediği mesajı seçer ve oku butonuna tıklar.<br>3. Mesajı okur ve kapatır. |

| Alternatif İş Akışı |
|---|
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 02/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.11 Mesaj Okuma Kullanım Durum Senaryosu

**SANALKAMPUS.SN6 Bilgi Güncelleme**

| | |
|---|---|
| Senaryo # | 6 |
| Senaryo İsmi | Bilgi Güncelleme |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Detay Fonkiyonlar |
| Aktör(ler) | Öğrenci, Öğretmen, Yönetici |
| Hedef | Bu senaryoda sistemde kayıtlı kullanıcıların sistemde kayıtlı bilgilerinde değişiklik yapmalarını hedefler. |
| Amaç | Sistemde kullanıcıların bilgilerindeki yanlışlıkları ve değişiklikleri düzeltebilmeleri amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | 1. Kullanıcı tüm bilgilerini bu modülden yapabilmelimidir? Tablolarda tutulan birincil anahtar özelliği taşıyan alanların değiştirilmesine izin verilmemeli. |
| Öndurumlar | Kullanıcı bilgilerini güncelleyebilmesi için sistemde oturum açmış olması gereklidir. |
| Başarılı bitiş durumları | Bilgi güncelleme başarı ile gerçekleşir. |
| Minimum bitiş durumları | Kullanıcı bilgileri sisteme kaydedilir. |

| Ana İş Akışı |
|---|
| 1. Kullanıcı, bilgi güncelleme butonuna tıklar.<br>2. Sistem kullanıcının var olan bilgilerini ekrana getirir.<br>3. Kullanıcı bu bilgiler üzerinde her alanın belirtilen formata uygun değişikliklerini yapar.<br>4. Yeni bilgiler sisteme kaydedilir ve kullanıcıya bilgi mesajı verilir. |
| Alternatif İş Akışı |
| 3A. Koşul: Değiştirilen bilgiler formata uygun değildir.<br>A3. Adım 1 Sistem Kullanıcıya girmiş olduğu bilgi veya bilgilerin istenen formata uygun olmadığı mesajını verir ve yeniden girmesini ister.<br>A3. Adım 2 Kullanıcı bilgi veya bilgileri yeniden girer.<br>A3. Adım 3 Bilgiler istenen formata uygun ise ana akışa dönülür.<br>A3. Adım 4 Tekrar uygun değilse bilgi veya bilgiler yeniden istenir. |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 02/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.12 Bilgi Güncelleme Kullanım Durum Senaryosu

**SANALKAMPUS.SN7 Ders Seçimi**

| | |
|---|---|
| Senaryo # | 7 |
| Senaryo İsmi | Ders Seçimi |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Detay Fonkiyonlar |
| Aktör(ler) | Öğrenci, Öğretmen |
| Hedef | Bu senaryoda Öğretmen ve Öğrenci Aktörlerinin üzerinde işlem yapacakları dersleri seçmelerini hedefler. |
| Amaç | Öğretmenlerin sınav sorusu hazırlamak veya ders dökümanı girmek, Öğrencilerin sınav olmak veya ders dökümanlarına ulaşmak için bulundukları dönem içinde sahip oldukları derslerden birini seçmeleri amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Ders seçebilmek için sistemde oturum açmış olmak gerekir.<br>Dönem Kaydını yapmış olması gerekir. |
| Başarılı bitiş durumları | Ders seçimi başarı ile gerçekleşir. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Sistem kullanıcının o dönem için üzerinde işlem yapabileceği dersleri listeler.<br>2. Kullanıcı üzerinde işlem yapacağı derse tıklar.<br>3. Sistem seçilen derse ilişkin yapılabilecek işlemleri ekrana getirir. |

| Alternatif İş Akışı |
|---|
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 02/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.13 Ders Seçimi Kullanım Durum Senaryosu

**SANALKAMPUS.SN8 Yardım**

| | |
|---|---|
| Senaryo # | 8 |
| Senaryo İsmi | Yardım |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Detay Fonkiyonlar |
| Aktör(ler) | Öğrenci, Öğretmen |
| Hedef | Bu senaryoda aktörlerin iş akışı konusunda yardım almalarını hedeflemektedir. |
| Amaç | Sistemde tüm aktörler iş akışını tam olarak bilemeyebilir. Bu aktörlere gerektiği yerde sistemin işleyişi hakkında yardım sağlamak amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Aktör Yardım alabilmesi için sistemde oturum açmış olması gereklidir. |
| Başarılı bitiş durumları | Aktör sistemin işleyişi ile ilgili bilgiye ulaşmıştır. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Kullanıcı yardım butonuna tıklar.<br>2. Sistem Aktörün iş akışı ile ilgili yardım bilgilerini ekrana getirir.<br>3. Aktör aradığı bilgiyi bu bilgiler arasından bulur. |
| Alternatif İş Akışı |
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 02/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.14 Yardım Kullanım Durum Senaryosu

**SANALKAMPUS.SN25 Görüntülü Konferans**

| | |
|---|---|
| Senaryo # | 25 |
| Senaryo İsmi | Görüntülü Konferans |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğrenci, Öğretmen |
| Hedef | Bu senaryoda öğretmen ve öğrencilerin çevrim içi sesli,görüntülü ve yazılı olarak iletişim kurması hedeflemektedir. |
| Amaç | Dersi veren öğretmen ile dersi alan öğrenciler arasında çevrim içi olarak sesli, görüntülü ve yazılı olarak iletişim sağlayarak öğrencilerin öğretmenlerinden yardım alabilmeleri amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | 1. Öğretmenin Görüntülü konferansa istediği öğrenciyi alma veya istemediği öğrenciyi alamama hakkı olmalı mı? |
| Öndurumlar | Sistemde oturum açmış olması gereklidir. |
| Başarılı bitiş durumları | Görüntülü Konferans başarı ile yapılmıştır. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Öğretmen görüntülü konferans yapacağı dersi seçer ve oturumu açar.<br>2. Öğrenci katılmak istediği görüntülü konferansın dersini seçer. Oturuma giriş talebinde bulunur.<br>3. Sistem Öğretmene oturuma katılmak isteyen öğrencileri listeler.<br>4. Öğretmen oturuma kabul ettiği öğrencileri seçer ve oturumu başlatır.<br>5. Öğretmen ve öğrenciler isteğe bağlı olarak yazılı, sesli ve görüntülü iletişim kurar.<br>6. Öğretmen oturumu kapatır. Ana Akış sonlanır. |
| Alternatif İş Akışı |
| 5A. Koşul: Öğrenci oturumdan ayrılır.<br>A5. Adım 1 Öğretmen oturumu sonlandırmadan öğrenci ayrıl butonuna basar.<br>A5. Adım 2 Öğrenci için akış sonlanır. Ana Akışa geri dönülür. |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 02/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.15 Görüntülü Konferans Kullanım Durum Senaryosu

**SANALKAMPUS.SN9 Dönem Kayıt**

| | |
|---|---|
| Senaryo # | 9 |
| Senaryo İsmi | Dönem Kayıt |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğrenci |
| Hedef | Bu senaryoda Öğrencinin okul dönem kaydını yapması hedefler. |
| Amaç | Öğrencinin okuluna devam edebilmesi için bulunduğu dönemin kaydını yapabilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | 1. Öğrenci dönem kaydını herzaman yapabilmeli midir?<br>Öğrenci dönem kaydını yapabilmesi için belirli süreler verilmelidir.<br>2. Kredi limiti uygulanmalı mıdır?<br>3. Öğrenci başka bölümlerden ders alabililirmi? |
| Öndurumlar | Öğrenci dönem harcını yatırmış olmalı.<br>Öğrenci sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Dönem Kaydı başarı ile gerçekleşir. |
| Minimum bitiş durumları | Dönem Kaydı girişimi log dosyalarına kaydedilir. |

| Ana İş Akışı |
|---|
| 1. Öğrenci Dönem Kaydı butonuna tıklar.<br>2. Sistem öğrencinin harcını yatırıp yatırmadığını kontrol eder. Yatırmış ise o dönem alabileceği dersleri ve geçmiş dönemlerden kalan derslerini listeler.<br>3. Öğrenci dersleri seçer ve onay butonuna basar.<br>4. Sistem Öğrencinin alttan kalan derslerinin tamamını alıp almadığının kontrolünü yapar.<br>5. Öğrenci alttan derslerinin tamamını aldıysa tüm derslerinin kredi toplamlarını kontrol eder ve uygunsa kaydını alır.<br>6. Öğrencinin dönem dersleri veritabanına kayıt edilir. |
| Alternatif İş Akışı |

2A. Koşul: Öğrenci harcını yatırmamıştır.

A2. Adım 1 Sistem Öğrenciye harcını yatırmadan ders kaydını yapamayacağı uyarısını verir.

A2. Adım 2 Akış sonlanır.

4A. Koşul: Öğrenci alttan kalan derslerinin tamamını seçmez.

A4. Adım 1 Sistem Öğrenciye öncelikle alttan derslerini alması konusunda uyarır.

A4. Adım 2 Öğrenci tekrar dersleri seçer.

A4. Adım 3 Öğrenci Alttan kalan derslerin tamamını seçtiyse ana akışa dönülür.

A4. Adım 4 Tamamı seçilmediyse tekrar uyarı verilir.

5A. Koşul: Öğrenci alabileceğinden fazla ders almıştır.

A5. Adım 1 Sistem Öğrencinin kredi limitini aştığı uyarısını verir.

A5. Adım 2 Öğrenci derslerin bir kısmının seçimini kaldırır.

A5. Adım 3 Öğrenci Kredi limitini aşmıyorsa ana akışa dönülür.

A5. Adım 4 Kredi limiti aşılıyorsa tekrar uyarı verir.

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 02/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.16 Dönem Kayıt Kullanım Durum Senaryosu

**SANALKAMPUS.SN10 Harç Kontrol**

| | |
|---|---|
| Senaryo # | 10 |
| Senaryo İsmi | Harç Kontrol |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğrenci |
| Hedef | Bu senaryoda Öğrencinin harç durumlarını görmesi hedeflenmektedir. |
| Amaç | Öğrencinin harç durumlarını görmesini eğitimi boyunca ne kadar para harcadığını görmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Öğrenci sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Öğrenci harç durumlarını görür. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Öğrenci Harç Durumları butonuna tıklar.<br>2. Sistem Öğrenciye ait harç durumlarını listeler. |
| **Alternatif İş Akışı** |
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 02/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.17 Harç Kontrol Kullanım Durum Senaryosu

**SANALKAMPUS.SN11 Ders Notları Oku**

| | |
|---|---|
| Senaryo # | 11 |
| Senaryo İsmi | Ders Notları Oku |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğrenci |
| Hedef | Bu senaryoda Öğrencinin seçmiş olduğu dersin dokumanlarına erişmesini hedeflenmektedir. |
| Amaç | Öğrencinin dönem sonu sınavlarına hazırlanabilmesi için o ders ile ilgili dokumanlara erişebilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Öğrenci dönem harcını yatırmış olmalı.<br>Öğrenci sisteme oturum açmış olmalı.<br>Öğrenci Dökümanına erişmek istediği dersi seçmiş olmalı. |
| Başarılı bitiş durumları | Öğrenci ders dökümanlarına başarı ile ulaşmıştır. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Öğrenci çalışmak istediği dersin adına tıklar.<br>2. Sistem seçilen dersin yazılı, sesli, görüntülü ve indirilebilen dokümanlarını listeler.<br>3. Öğrenci görmek istediği yazılı sesli görüntülü dokumanı işaretler.<br>4. Sistem Öğrencinin seçtiği dokümanı ekrana getirir ve akışı sonlandırır. |
| Alternatif İş Akışı |
| 2A. Koşul: Seçilen dersin dokümanı sistemde yoktur.<br>A2. Adım 1 Sistem öğrenciye bu derse ait henüz bir doküman olmadığını söyler.<br>A2. Adım 2 Akış sonlanır.<br>3A. Koşul: Öğrenci indirilebilir dokümanlardan birini seçer.<br>A3. Adım 1 Sistem Öğrenciye Dosya yükleme penceresini açar.<br>A3. Adım 2 Öğrenci dosyayı bilgisayarında nereye yükleyeceğini seçer ve kaydet butonuna basar.<br>A3. Adım 3 Akışı sonlandırır. |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 02/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.18 Ders Notları Oku Kullanım Durum Senaryosu

**SANALKAMPUS.SN12 Ara Sınav Ol**

| | |
|---|---|
| Senaryo # | 12 |
| Senaryo İsmi | Ara Sınav Ol |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğrenci |
| Hedef | Bu senaryoda Öğrencinin dönem içinde ara sınav olması hedeflenir. |
| Amaç | Öğrencinin ders içerisinde belli bölümler sonun da ara sınavlar olabilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | 1. Ara sınav sonuçları dönem ortalamasına etki etmeli mi?<br>Bölüm sonların da öğrencinin seviyesini belirlemeye yönelik olmalı.<br>2. Soruların cevapları öğrenciye gösterilmeli mi?<br>Soruların cevapları sınav bitiminde öğrenciye gösterilmeli.<br>3. Öğrenci bu ara sınavı tekrar alabilmeli mi? |
| Öndurumlar | Öğrenci dönem harcını yatırmış olmalı.<br>Öğrenci sisteme oturum açmış olmalı.<br>Öğrenci Dersi seçmiş olmalı. |
| Başarılı bitiş durumları | Öğrenci ders ara sınavını başarı ile tamamlamıştır. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Öğrenci seçmiş olduğu dersin çalıştığı bölüm sonundaki ara sınav seçeneğini seçer.<br>2. Sistem öğrenciye seçmiş olduğu ara sınavın sorularından rasgele 10 tanesini seçer ekrana getirir.<br>3. Öğrenci soruları cevaplar ve sınavı bitir butonuna basar.<br>4. Sistem cevapların kontrolü yaptıktan sonra öğrenciye sınav sonuçlarını gösterir.<br>5. Öğrenci cevapları görmek istiyorum butonuna tıklar. |

| | | | |
|---|---|---|---|
| 6. Sistem soruların cevaplarını ekrana getirir.<br>7. Öğrenci doğru cevaplar ile kendi cevaplarının kontrolünü yapar ve ara sınavı sonlandırır | | | |
| Alternatif İş Akışı | | | |
| 5A. Koşul: Öğrenci soruların cevaplarını görmek istemez.<br>A5. Adım 1 Öğrenci Sınavı bitir butonuna basar.<br>A5. Adım 2 Sistem sınavı sonlandırır. | | | |
| Belge Tarihçesi | | | |
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 03/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |
| 30/Mayıs2007 | 0,2 | Alternatif Akış eklendi | Sait Ali Uymaz |

Çizelge 4.19 Ara Sınav Ol Kullanım Durum Senaryosu

**SANALKAMPUS.SN13 Sınav Ol**

| | |
|---|---|
| Senaryo # | 13 |
| Senaryo İsmi | Sınav Ol |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğrenci |
| Hedef | Bu senaryoda Öğrencinin dönem sonu sınav olması hedeflenir. |
| Amaç | Öğrencinin dönem sonun da her dersten seviyesini belirleyecek genel bir sınav olması amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | 1. Sınav süreli olmalı mı?<br>Sistem öğrenciye sınavı tamamlaması için süre vermeli.<br>2. Soruların cevapları öğrenciye gösterilmeli mi?<br>Soruların cevapları sınav bitiminde öğrenciye gösterilmeyebilir. |
| Öndurumlar | Öğrenci sisteme oturum açmış olmalı.<br>Öğrenci Derse ait arasınavların en az %70 ine katılmış olmalı. |
| Başarılı bitiş durumları | Öğrenci sınavını başarı ile tamamlamıştır. |
| Minimum bitiş durumları | Öğrencinin sınavı başlatma bilgileri log dosyalarına yazılmıştır. |

| Ana İş Akışı |
|---|
| 1. Öğrenci seçmiş olduğu dersin genel sınav seçeneğini seçer. Başlat Butonuna basar.<br>2. Sistem öğrenciye seçmiş olduğu sınavın sorularından rasgele 20 tanesini seçer ekrana getirir ve sınav süresini başlatır.<br>3. Öğrenci soruları cevaplar.<br>4. Öğrenci soruları bitirdikten sonra sınavı bitir butonuna basar.<br>5. Sistem cevapların kontrolünü yapar.<br>6. Öğrenciye doğru, yanlış cevap sayılarını ve notunu gösterip veritabanına kaydeder. |
| Alternatif İş Akışı |
| 4A. Koşul: Öğrenci soruları cevaplarken süre biter.<br>A4. Adım 1 Sistem sınavı sonlandırır.<br>A4. Adım 2 Ana akışa dönülür. |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 03/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.20 Sınav Ol Kullanım Durum Senaryosu

**SANALKAMPUS.SN14 Sınav Sonuçlarına Bak**

| | |
|---|---|
| Senaryo # | 14 |
| Senaryo İsmi | Sınav Sonuçlarına Bak |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğrenci |
| Hedef | Bu senaryoda Öğrencinin sınav sonuçlarına bakması hedeflenir. |
| Amaç | Öğrencinin okul hayatı boyunca girmiş olduğu sınav sonuçlarına bakabilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | 1. Dönem ortalamaları görüntülenmeli mi? |
| Öndurumlar | Öğrenci dönem harcını yatırmış olmalı.<br>Öğrenci sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Öğrenci sınav sonuçlarını, ders durumlarını görmüştür. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Öğrenci Sınav Notlarına Bak butonuna tıklar.<br>2. Sistem Öğrencinin 1. dönemden itibaren bulunduğu döneme kadar ki tüm notlarını ve dönem sonu not ortalamalarını hesaplar ve ekrana getirir. |

| Alternatif İş Akışı |
|---|
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 03/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.21 Sınav Sonuçlarına Bak Kullanım Durum Senaryosu

**SANALKAMPUS.SN15 Ders Değerlendirme Formu Doldur**

| | |
|---|---|
| Senaryo # | 15 |
| Senaryo İsmi | Ders Değerlendirme Formu Doldur |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğrenci |
| Hedef | Bu senaryoda Öğrencinin dönem sonu ders değerlendirme formunu doldurması hedeflenir. |
| Amaç | Öğrencinin dönem sonunda öğretmenin hazırlamış olduğu ders ile ilgili formu doldurması amaçlanmıştır. Öğrencinin Dönem sonunda dersin işlenişi, sınavı, konu anlatımları hakkında fikrini almak asıl amaç. |
| Olgunluk | Sarı |
| Sorular | 1. Bu formu doldurmak öğrenciye zorunlu kılınmalı mı?<br>2. Öğrenci istediği anda bu formu doldurabilmeli mi?<br>Öğrenci dersin genel sınavını tamamladıktan sonra form sunulmalı. Dönem başında veya ortasında rastgele dolduramamalı. |
| Öndurumlar | Öğrenci sisteme oturum açmış olmalı.<br>Formu doldurmak için dönem sonu olmalı. |
| Başarılı bitiş durumları | Öğrenci formu başarı ile doldurmuştur. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Sistem genel sınav bitiminde öğrenciye değerlendirme formunu sunar.<br>2. Öğrenci değerlendirme sorularını yanıtlar.<br>3. Değerlendirme Kayıt butonuna basar.<br>4. Sistem cevapları veritabanına işler ve akışı sonlandırır. |
| Alternatif İş Akışı |
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 03/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.22 Ders Değerlendirme Formu Doldur Kullanım Durum Senaryosu

**SANALKAMPUS.SN16 Ders Notu Ekle, Sil, Düzelt**

| | |
|---|---|
| Senaryo # | 16 |
| Senaryo İsmi | Ders Notu Ekle, Sil, Düzelt |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğretmen |
| Hedef | Bu senaryoda Öğretmenin ders için doküman eklemesi silmesi veya varolan dökümanı düzeltmesi hedeflenir. |
| Amaç | Öğretmenin öğrencilerin derslere çalışabilmesi için sınava hazırlanabilmesi için düzenli anlaşılır dokümanlara sahip olmalıdır. Öğretmenin bu ders dokümanlarını düzenleyebilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Öğretmen sisteme oturum açmış olmalı.<br>Öğretmen doküman düzenlemesi yapacağı dersi ve konu başlığını seçmiş olmalı. |
| Başarılı bitiş durumları | Öğretmen ders dökümanlarını başarı ile sisteme kaydetmiştir |
| Minimum bitiş durumları | |

Ana İş Akışı

1. Sistem Öğretmene Ekle Sil düzelt seçeneklerini sunar.
2. Öğretmen doküman ekle butonuna basar.
3. Sistem Öğretmene dokümanı yazı olarak mı yoksa dosya olarak mı ekleyeceğini sorar.
4. Öğretmen dokümanı yazılı olarak ekleyeceğim seçeneğini işaretler.
5. Sistem doküman eklemesi için yazı editörünü açar.
6. Öğretmen konunun içeriğini yazar. Kaydet butonuna basar.
7. Sistem derse konu başlığına içeriği ekler ve kaydeder. Akışı sonlandırır.

Alternatif İş Akışı

2A. Koşul: Öğretmen doküman sil butonuna basar.

A2. Adım 1 Sistem dersin dokümanlarını listeler.

A2. Adım 2 Öğretmen silmek istediği dokumanı işaretler.

A2. Adım 3 Sistem seçili dokümanın bağını koparır pasif hale çeker. Akışı sonlandırır.

2B. Koşul: Öğretmen doküman düzelt butonuna basar.

B2. Adım 1 Sistem dersin dokümanlarını listeler.

B2. Adım 2 Öğretmen düzenleme yapacağı dokümanı işaretler.

B2. Adım 3 Sistem seçilen dokümanın bilgilerini ekrana getirir.

B2. Adım 4 Öğretmen dokümandaki düzeltmeleri yapar kaydet butonuna basar.

B2. Adım 5 Sistem düzeltme işlemini veritabanına uygular. Akışı sonlandırır.

4A. Koşul: Öğretmen dokümanı dosya olarak ekleyeceğim seçeneğini işaretler.

A4. Adım 1 Sistem Öğretmene dosya yöneticisini açar.

A4. Adım 2 Öğretmen daha önceden hazırlamış olduğu sesli, görüntülü veya yazılı dokumanı seçer ve kaydet butonuna basar.

A4. Adım 3 Sistem bu dokümanı ismi ile dersin dokümanları arasına ekler.

A4. Adım 4 Akışı sonlandırır.

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 03/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.23 Ders Notu Ekle, Sil, Düzelt Kullanım Durum Senaryosu

**SANALKAMPUS.SN17 Sınav Sorusu Ekle, Sil, Düzelt**

| | |
|---|---|
| Senaryo # | 17 |
| Senaryo İsmi | Sınav Sorusu Ekle, Sil, Düzelt |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğretmen |
| Hedef | Bu senaryoda Öğretmenin sınav sorusu girmesi hedeflenir. |
| Amaç | Öğrencilerin bölüm sonlarında seviyelerini ölçebilmeleri ve dönem sonunda göğrencilerin başarılarını ölçebilmek için Öğretmenin sınav sorularını girebilmeleri amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | 1. Farklı soru tipleri olmalı mı?<br>Öğretmene birkaç seçenek sunulabilir. |
| Öndurumlar | Öğretmen sisteme oturum açmış olmalı.<br>Öğretmen soru gireceği dersi seçmiş olmalı.<br>Öğretmen soru gireceği dersin konusunu seçmiş olmalı. |
| Başarılı bitiş durumları | Öğretmen ders için başarı ile soru girmiştir. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Sistem Öğretmene Sınav tiplerini (ara sınav genel sınav) ekrana getirir.<br>2. Öğretmen genel sınav tipini seçer.<br>3. Sistem ekleme silme düzeltme seçeneklerini sunar.<br>4. Öğretmen soru eklemeyi seçer.<br>5. Sistem Öğretmene soru giriş formunu açar.<br>6. Öğretmen Soruyu, cevap şıklarını ve doğru cevabı sisteme girer.<br>7. Akışı sonlanır. |
| **Alternatif İş Akışı** |
| 2A. Koşul: Öğretmen Ara sınav seçeneğini seçer.<br>A2. Adım 1 Sistem Öğretmene konu başlıklarını ekrana getirir. |

| |
|---|
| A2. Adım 2 Öğretmen konuyu seçer. Ana akışa dönülür.<br>4A. Koşul: Öğretmen soru silmeyi seçer.<br>A4. Adım 1 Sistem Öğretmene soruları listeler.<br>A4. Adım 2 Öğretmen silmek istediği soruyu seçer ve sil butonuna basar.<br>A4. Adım 3 Sistem soruyu siler. Akışı sonlandırır.<br>4B. Koşul: Öğretmen soru düzeltmeyi seçer.<br>B4. Adım 1 Sistem Öğretmene soruları listeler.<br>B4. Adım 2 Öğretmen düzeltmek istediği soruyu seçer ve düzelt butonuna basar.<br>B4. Adım 3 Sistem soruya ait bilgileri ekrana getirir.<br>B4. Adım 4 Öğretmen düzeltmeyi yapar kaydet butonuna basar.<br>B4. Adım 5 Sistem soruyu kaydeder. Akışı sonlandırır. |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 03/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.24 Sınav Sorusu Ekle, Sil, Düzelt Kullanım Durum Senaryosu

**SANALKAMPUS.SN18 Ders Değerlendirme Formu Oluştur**

| | |
|---|---|
| Senaryo # | 18 |
| Senaryo İsmi | Ders Değerlendirme Formu Oluştur. |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğretmen |
| Hedef | Bu senaryoda Öğretmenin ders için değerlendirme formu oluşturması hedeflenir. |
| Amaç | Öğretmenin, öğrencilerin dönem sonlarında ders ile ilgili görüşlerini almak için bir değerlendirme formu oluşturulması amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Öğretmen sisteme oturum açmış olmalı.<br>Öğretmen değerlendirme formu oluşturacağı dersi seçmiş olmalı. |
| Başarılı bitiş durumları | Öğretmen değerlendirme formunu başarı ile oluşturmuştur. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Öğretmen Değerlendirme Formu oluştur seçeneğini oluştur butonuna basar.<br>2. Sistem Öğretmene değerlendirme formu giriş ekranını açar.<br>3. Öğretmen değerlendirme sorularını ve seçenekleri yazar. Kaydet butonuna basar.<br>4. Sistem formu veritabanına kaydeder.<br>5. Akışı sonlandırır. |
| Alternatif İş Akışı |
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 03/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.25 Ders Değerlendirme Formu Oluştur Kullanım Durum Senaryosu

**SANALKAMPUS.SN19 Sınav Sonuç Listelerine Bak**

| | |
|---|---|
| Senaryo # | 19 |
| Senaryo İsmi | Sınav Sonuç Listelerine Bak |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Öğretmen |
| Hedef | Bu senaryoda Öğretmenin öğrencilerin sınav sonuçlarını görmesi hedeflenir. |
| Amaç | Öğretmenin, öğrencilerin dönem içinde almış olduğu notları, girmiş olduğu ara sınavları ve başarı durumlarını görebilmeleri amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Öğretmen sisteme oturum açmış olmalı.<br>Öğretmen sınav sonuçlarını inceleyeceği dersi seçmiş olmalı. |
| Başarılı bitiş durumları | Öğretmen sınav sonuçlarını görmüştür. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Sistem Öğretmenin seçtiği dersin sınavlarını listeler.<br>2. Öğretmen görmek istediği sınavı seçer. Sınav Sonuçları butonuna basar.<br>3. Sistem seçilen sınavın sonuçlarını ekrana listeler. Akış sonlanır. |
| Alternatif İş Akışı |
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 03/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.26 Sınav Sonuç Listelerine Bak Kullanım Durum Senaryosu

**SANALKAMPUS.SN20 Bölüm Ekle, Kaldır, Düzelt**

| | |
|---|---|
| Senaryo # | 20 |
| Senaryo İsmi | Bölüm Ekle, Kaldır, Düzelt |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Yönetici |
| Hedef | Bu senaryoda Yöneticinin Okula bölüm eklemesi silmesi ve düzeltme yapması hedeflenir. |
| Amaç | Yöneticinin okulda hizmet verilen bölümler üzerinde işlem yapabilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Yönetici sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Yönetici bölümleri başarı ile düzenlemiştir. |
| Minimum bitiş durumları | |

Ana İş Akışı

1. Yönetici Bölüm düzenle butonuna basar.
2. Sistem Yöneticiye Ekle Sil düzelt seçeneklerini sunar.
3. Yönetici Bölüm Ekle butonuna basar.
4. Sistem Yöneticiye Bölüm bilgisi giriş formunu sunar.
5. Yönetici üniversite ve fakülteyi seçer bölüm tanımlamalarını yapar ve kaydet butonuna basar.
6. Sistem bölüm bilgilerini veritabanına kaydeder. Akışı sonlandırır.

Alternatif İş Akışı

3A. Koşul: Yönetici bölüm kaldır butonuna basar.

A3. Adım 1 Sistem bölümleri listeler.

A3. Adım 2 Yönetici uygulamadan kaldırmak istediği bölümü işaretler.

A3. Adım 3 Sistem seçilen bölümün altında tanımlı bir sınıf yoksa bölümün aktifliğini kaldırır.

A3. Adım 4 Bölümün altında tanımlı bir sınıf varsa sistem yöneticiyi uyarır. Akışı sonlandırır.

| |
|---|
| 3B. Koşul: Yönetici bölüm düzelt butonuna basar.<br>B3. Adım 1 Sistem bölümleri listeler.<br>B3. Adım 2 Öğretmen düzenleme yapacağı bölümü işaretler.<br>B3. Adım 3 Sistem seçilen bölümün bilgilerini ekrana getirir.<br>B3. Adım 4 Öğretmen bölümdeki düzeltmeleri yapar kaydet butonuna basar.<br>B3. Adım 5 Sistem düzeltme işlemini veritabanına uygular. Akışı sonlandırır.<br>5A. Koşul: Yönetici var olan bölümü tekrar girer.<br>A5. Adım 1 Sistem bu bilgilere sahip bölüm var uyarısı yapar.<br>A5. Adım 2 Yönetici bölüm bilgilerini tekrar girer.<br>A5. Adım 3 Bu bilgilere sahip bölüm yoksa ana akışa dönülür varsa tekrar uyarı verir. |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 04/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.27 Bölüm Ekle, Kaldır, Düzelt Kullanım Durum Senaryosu

**SANALKAMPUS.SN21 Öğrenci Ekle, Sil, Düzelt**

| | |
|---|---|
| Senaryo # | 21 |
| Senaryo İsmi | Öğrenci Ekle, Sil, Düzelt |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Yönetici |
| Hedef | Bu senaryoda Yöneticinin Okula Öğrenci eklemesi silmesi ve düzeltme yapması hedeflenir. |
| Amaç | Yöneticinin okula kayıt yaptıran öğrenciler üzerinde işlem yapabilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | Öğrencinin hangi bilgileri yönetici tarafından sisteme girilir? |
| Öndurumlar | Yönetici sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Yönetici öğrenci işlemlerini başarı ile sisteme kaydetmiştir |
| Minimum bitiş durumları | |

Ana İş Akışı

1. Sistem Yöneticiye öğrencisi üzerinde işlem yapacağı bölümleri listeler.
2. Yönetici bölümü ve sınıfını seçer.
3. Sistem Yöneticiye varsa öğrenci listesini ve ekleme silme düzeltme seçeneklerini sunar.
4. Yönetici Öğrenci ekleme butonuna basar.
5. Sistem öğrenci ekleme formunu ekrana getirir.
6. Yönetici öğrenci bilgilerini (sisteme giriş yapabilmesi için öğrenci numarası, şifre vb.) girer kaydet butonuna basar.
7. Sistem seçilen bölüm ve sınıfa öğrenciyi kaydeder. Akışı sonlandırır.

Alternatif İş Akışı

4A. Koşul: Yönetici öğrenciyi seçer ve sil butonuna basar.

A4. Adım 1 Sistem seçilen öğrenciyi pasif hale getirir. Akışı sonlandırır.

4B. Koşul: Yönetici öğrenciyi seçer ve düzelt butonuna basar.

B4. Adım 1 Sistem seçilen öğrencinin bilgilerin ekrana getirir.

B4. Adım 2 Yönetici öğrenci bilgilerini düzeltir ve kaydet butonuna basar.

| | | | |
|---|---|---|---|
| B4. Adım 3 Sistem öğrencinin bilgilerini veritabanına kaydeder. Akışı sonlandırır.<br>6A. Koşul: Yönetici var olan öğrenciyi tekrar girer.<br>A6. Adım 1 Sistem bu bilgilere sahip öğrenci var uyarısı yapar.<br>A6. Adım 2 Yönetici öğrenci bilgilerini tekrar girer.<br>A6. Adım 3 Bu bilgilere sahip öğrenci yoksa ana akışa dönülür varsa tekrar uyarı verir. | | | |
| Belge Tarihçesi | | | |
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 04/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.28 Öğrenci Ekle, Sil, Düzelt Kullanım Durum Senaryosu

**SANALKAMPUS.SN22 Ders Ekle, Sil, Düzelt**

| | |
|---|---|
| Senaryo # | 22 |
| Senaryo İsmi | Ders Ekle, Sil, Düzelt |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Yönetici |
| Hedef | Bu senaryoda Yöneticinin eğitim müfredatına Ders eklemesi silmesi ve düzeltme yapması hedeflenir. |
| Amaç | Yöneticinin okula yeni dersler açabilmesi bölüm müfredatlarını düzenleyebilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Yönetici sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Yönetici Ders işlemlerini başarı ile sisteme kaydetmiştir |
| Minimum bitiş durumları | |

Ana İş Akışı

1. Sistem Yöneticiye Ders müfredatı üzerinde işlem yapacağı bölümleri, dönemleri ve sınıflarını listeler.
2. Yönetici bölümü, dönemi ve sınıfını seçer.
3. Sistem Yöneticiye varsa Ders listesini ve ekleme silme düzeltme seçeneklerini sunar.
4. Yönetici Ders ekleme butonuna basar.
5. Sistem Ders ekleme formunu ekrana getirir.
6. Yönetici üniversite, fakülte, bölüm ve sınıfı seçer Ders bilgilerini girer kaydet butonuna basar.
7. Sistem seçilen bölüm ve sınıfa Dersi kaydeder. Akışı sonlandırır.

Alternatif İş Akışı

4A. Koşul: Yönetici Dersi seçer ve sil butonuna basar.

A4. Adım 1 Sistem seçilen Dersi ve altında tanımlı konuları, dokümanları ve soruları pasif hale getirir. Akışı sonlandırır.

4B. Koşul: Yönetici Dersi seçer ve düzelt butonuna basar.

| | | | |
|---|---|---|---|
| B4. Adım 1 Sistem seçilen Dersin bilgilerin ekrana getirir.<br>B4. Adım 2 Yönetici Ders bilgilerini düzeltir ve kaydet butonuna basar.<br>B4. Adım 3 Sistem Dersin bilgilerini veritabanına kaydeder. Akışı sonlandırır.<br>6A. Koşul: Yönetici var olan Dersi tekrar girer.<br>A6. Adım 1 Sistem bu bilgilere sahip Ders var uyarısı yapar.<br>A6. Adım 2 Yönetici Ders bilgilerini tekrar girer.<br>A6. Adım 3 Bu bilgilere sahip Ders yoksa ana akışa dönülür varsa tekrar uyarı verir. | | | |
| Belge Tarihçesi | | | |
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 04/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.29 Ders Ekle, Sil, Düzelt Kullanım Durum Senaryosu

**SANALKAMPUS.SN23 Öğretmen Ekle, Sil, Düzelt**

| | |
|---|---|
| Senaryo # | 23 |
| Senaryo İsmi | Öğretmen Ekle, Sil, Düzelt |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Yönetici |
| Hedef | Bu senaryoda Yöneticinin Okula Öğretmen eklemesi silmesi ve düzeltme yapması hedeflenir. |
| Amaç | Yöneticinin okula eğitim verecek Öğretmenler üzerinde işlem yapabilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | Öğretmennin hangi bilgileri yönetici tarafından sisteme girilir? |
| Öndurumlar | Yönetici sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Yönetici Öğretmen işlemlerini başarı ile sisteme kaydetmiştir |
| Minimum bitiş durumları | |

Ana İş Akışı

1. Sistem Yöneticiye varsa Öğretmen listesini ve ekleme silme düzeltme seçeneklerini sunar.
2. Yönetici Öğretmen ekleme butonuna basar.
3. Sistem Öğretmen ekleme formunu ekrana getirir.
4. Yönetici Öğretmen bilgilerini (sisteme giriş yapabilmesi için Sicil numarası, şifre vb.) girer kaydet butonuna basar.
5. Sistem Öğretmeni kaydeder. Akışı sonlandırır.

Alternatif İş Akışı

2A. Koşul: Yönetici Öğretmeni seçer ve sil butonuna basar.

A2. Adım 1 Sistem seçilen Öğretmene ait tanımlı ders yoksa pasif hale getirir.

A2. Adım 2 Tanımlı ders varsa yöneticiyi uyarır. Akışı sonlandırır.

2B. Koşul: Yönetici Öğretmeni seçer ve düzelt butonuna basar.

B2. Adım 1 Sistem seçilen Öğretmenin bilgilerin ekrana getirir.

B2. Adım 2 Yönetici Öğretmen bilgilerini düzeltir ve kaydet butonuna basar.

| | | | |
|---|---|---|---|
| B2. Adım 3 Sistem Öğretmenin bilgilerini veritabanına kaydeder. Akışı sonlandırır.<br>4A. Koşul: Yönetici var olan Öğretmeni tekrar girer.<br>A4. Adım 1 Sistem bu bilgilere sahip Öğretmen var uyarısı yapar.<br>A4. Adım 2 Yönetici Öğretmen bilgilerini tekrar girer.<br>A4. Adım 3 Bu bilgilere sahip Öğretmen yoksa ana akışa dönülür varsa tekrar uyarı verir. | | | |
| Belge Tarihçesi | | | |
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 04/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.30 Öğretmen Ekle, Sil, Düzelt Kullanım Durum Senaryosu

**SANALKAMPUS.SN24 Öğretmene Ders Atama**

| | |
|---|---|
| Senaryo # | 24 |
| Senaryo İsmi | Öğretmene Ders Atama |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Yönetici |
| Hedef | Bu senaryoda Yöneticinin Öğretmenler üzerine ders ataması hedeflenir. |
| Amaç | Yöneticinin okula eğitim verilen derslerin hangi öğretmenler tarafından verileceğinin belirlenmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | 1. Bir derse birden fazla öğretmen atanabilmeli mi?<br>2. Bir öğretmene birden fazla ders atanabilmeli mi?<br>Bir öğretmenin birden fazla uzmanlık alanı olabilir. |
| Öndurumlar | Yönetici sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Yönetici Öğretmene ders atama işlemini başarı ile sisteme kaydetmiştir |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Sistem Yöneticiye kayıtlı Öğretmenleri ekrana listeler.<br>2. Yönetici ders eklemek istediği Öğretmeni seçer.<br>3. Sistem Öğretmen üzerine atılabilecek dersleri listeler.<br>4. Yönetici Öğretmenin vereceği dersleri işaretler ve kaydet butonuna basar.<br>5. Sistem veritabanında bu derslerin sorumlusu kısmına seçilen öğretmeni kaydeder. Akışı sonlandırır. |

| Alternatif İş Akışı |
|---|
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 04/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.31 Öğretmene Ders Atama Kullanım Durum Senaryosu

**SANALKAMPUS.SN26 Üniversite Ekle, Kaldır, Düzelt**

| | |
|---|---|
| Senaryo # | 26 |
| Senaryo İsmi | Üniversite Ekle, Kaldır, Düzelt |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Yönetici |
| Hedef | Bu senaryoda Yöneticinin üniversite oluşturabilmesi ve düzenleyebilmesi hedeflenir. |
| Amaç | Yöneticinin uzaktan eğitimin çatısı olan üniversiteyi oluşturabilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Yönetici sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Yönetici üniversiteyi başarı ile düzenlemiştir. |
| Minimum bitiş durumları | |

Ana İş Akışı

1. Yönetici Üniversite düzenle butonuna basar.
2. Sistem Yöneticiye Ekle Sil düzelt seçeneklerini sunar.
3. Yönetici Üniversite Ekle butonuna basar.
4. Sistem Yöneticiye üniversite giriş formunu sunar.
5. Yönetici üniversite tanımlamalarını yapar ve kaydet butonuna basar.
6. Sistem üniversite bilgilerini veritabanına kaydeder. Akışı sonlandırır.

Alternatif İş Akışı

3A. Koşul: Yönetici üniversite kaldır butonuna basar.

A3. Adım 1 Sistem tanımlı üniversiteleri listeler.

A3. Adım 2 Yönetici uygulamadan kaldırmak istediği üniversiteyi işaretler.

A3. Adım 3 Sistem seçili üniversitenin altında tanımlı fakülte yoksa aktifliğini kaldırır.

A3 Adım 4 Tanımlı fakülteler varsa sistem yöneticiyi uyarır. Akışı sonlandırır.

3B. Koşul: Yönetici üniversite düzelt butonuna basar.

B3. Adım 1 Sistem üniversiteleri listeler.

| | | | |
|---|---|---|---|
| B3. Adım 2 Yönetici düzenleme yapacağı üniversiteyi işaretler.<br>B3. Adım 3 Sistem seçilen üniversitenin bilgilerini ekrana getirir.<br>B3. Adım 4 Yönetici üniversitedeki düzeltmeleri yapar kaydet butonuna basar.<br>B3. Adım 5 Sistem düzeltme işlemini veritabanına uygular. Akışı sonlandırır.<br>5A. Koşul: Yönetici var olan üniversiteyi tekrar girer.<br>A5. Adım 1 Sistem bu bilgilere sahip üniversite var uyarısı yapar.<br>A5. Adım 2 Yönetici üniversite bilgilerini tekrar girer.<br>A5. Adım 3 Bu bilgilere sahip üniversite yoksa ana akışa dönülür varsa tekrar uyarı verir. | | | |
| Belge Tarihçesi | | | |
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 25/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.32 Üniversite Ekle, Kaldır, Düzelt Kullanım Durum Senaryosu

**SANALKAMPUS.SN27 Fakülte Ekle, Kaldır, Düzelt**

| | |
|---|---|
| Senaryo # | 27 |
| Senaryo İsmi | Fakülte Ekle, Kaldır, Düzelt |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Yönetici |
| Hedef | Bu senaryoda Yöneticinin Fakülte oluşturabilmesi ve düzenleyebilmesi hedeflenir. |
| Amaç | Yöneticinin uzaktan eğitimde farklı alanlarda eğitim verebilmek için fakülteler oluşturabilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Yönetici sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Yönetici fakülteleri başarı ile düzenlemiştir. |
| Minimum bitiş durumları | |

Ana İş Akışı

1. Yönetici fakülte düzenle butonuna basar.
2. Sistem Yöneticiye Ekle Sil düzelt seçeneklerini sunar.
3. Yönetici üniversiteyi seçer ve fakülte Ekle butonuna basar.
4. Sistem Yöneticiye fakülte giriş formunu sunar.
5. Yönetici fakülte tanımlamalarını yapar ve kaydet butonuna basar.
6. Sistem fakülte bilgilerini veritabanına kaydeder. Akışı sonlandırır.

Alternatif İş Akışı

3A. Koşul: Yönetici fakülte kaldır butonuna basar.

A3. Adım 1 Sistem tanımlı fakülteleri listeler.

A3. Adım 2 Yönetici uygulamadan kaldırmak istediği fakülteyi işaretler.

A3. Adım 3 Sistem seçili fakültenin altında tanımlı bölümü yoksa aktifliğini kaldırır.

A3 Adım 4 Tanımlı bölümler varsa sistem yöneticiyi uyarır. Akışı sonlandırır.

3B. Koşul: Yönetici fakülte düzelt butonuna basar.

B3. Adım 1 Sistem fakülteleri listeler.

| |
|---|
| B3. Adım 2 Yönetici düzenleme yapacağı fakülteyi işaretler.<br>B3. Adım 3 Sistem seçilen fakültenin bilgilerini ekrana getirir.<br>B3. Adım 4 Yönetici fakültedeki düzeltmeleri yapar kaydet butonuna basar.<br>B3. Adım 5 Sistem düzeltme işlemini veritabanına uygular. Akışı sonlandırır.<br>5A. Koşul: Yönetici var olan üniversiteyi tekrar girer.<br>A5. Adım 1 Sistem bu bilgilere sahip fakülte var uyarısı yapar.<br>A5. Adım 2 Yönetici fakülte bilgilerini tekrar girer.<br>A5. Adım 3 Bu bilgilere sahip fakülte yoksa ana akışa dönülür varsa tekrar uyarı verir. |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 25/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.33 Fakülte Ekle, Kaldır, Düzelt Kullanım Durum Senaryosu

**SANALKAMPUS.SN28 Sınıf Ekle, Kaldır, Düzelt**

| | |
|---|---|
| Senaryo # | 28 |
| Senaryo İsmi | Sınıf Ekle, Kaldır, Düzelt |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Yönetici |
| Hedef | Bu senaryoda Yöneticinin sınıf oluşturabilmesi ve düzenleyebilmesi hedeflenir. |
| Amaç | Yöneticinin uzaktan eğitimde bölümlerin eğitimlerinin kaç yıl süreceğini belirlemeleri amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Yönetici sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Yönetici Sınıfları başarı ile düzenlemiştir. |
| Minimum bitiş durumları | |

Ana İş Akışı

1. Yönetici sınıf düzenle butonuna basar.
2. Sistem Yöneticiye Ekle Sil düzelt seçeneklerini sunar.
3. Yönetici üniversiteyi, fakülteyi ve bölümü seçer ve sınıf Ekle butonuna basar.
4. Sistem Yöneticiye sınıf giriş formunu sunar.
5. Yönetici sınıf tanımlamalarını yapar ve kaydet butonuna basar.
6. Sistem sınıf bilgilerini veritabanına kaydeder. Akışı sonlandırır.

Alternatif İş Akışı

3A. Koşul: Yönetici sınıf kaldır butonuna basar.

A3. Adım 1 Sistem tanımlı sınıfları listeler.

A3. Adım 2 Yönetici uygulamadan kaldırmak istediği sınıfı işaretler.

A3. Adım 3 Sistem seçili sınıfın altında tanımlı öğrenci ve ders yoksa aktifliğini kaldırır.

A3 Adım 4 Tanımlı öğrenci ve dersler varsa sistem yöneticiyi uyarır. Akışı sonlandırır.

3B. Koşul: Yönetici sınıf düzelt butonuna basar.

| |
|---|
| B3. Adım 1 Sistem sınıfları listeler.<br>B3. Adım 2 Yönetici düzenleme yapacağı sınıfı işaretler.<br>B3. Adım 3 Sistem seçilen sınıfın bilgilerini ekrana getirir.<br>B3. Adım 4 Yönetici sınıftaki düzeltmeleri yapar kaydet butonuna basar.<br>B3. Adım 5 Sistem düzeltme işlemini veritabanına uygular. Akışı sonlandırır.<br>5A. Koşul: Yönetici herhangi bölümde var olan bir sınıfı tekrar girer.<br>A5. Adım 1 Sistem bu bilgilere sahip sınıf var uyarısı yapar.<br>A5. Adım 2 Yönetici sınıf bilgilerini tekrar girer.<br>A5. Adım 3 Bu bilgilere sahip sınıf yoksa ana akışa dönülür varsa tekrar uyarı verir. |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 25/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.33 Sınıf Ekle, Kaldır, Düzelt Kullanım Durum Senaryosu

**SANALKAMPUS.SN29 Mezun Etme**

| | |
|---|---|
| Senaryo # | 29 |
| Senaryo İsmi | Mezun Etme |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Yönetici, Alt Sistem |
| Hedef | Bu senaryoda Yöneticinin öğrenciyi mezun etmesi hedeflenir. |
| Amaç | Yöneticinin mezuniyet durumuna gelen öğrencileri mezun edebilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Yönetici sisteme oturum açmış olmalı. |
| Başarılı bitiş durumları | Yönetici öğrencileri başarı ile mezun etmiştir. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Yönetici Mezun Etme butonuna basar.<br>2. Sistem Yöneticiye Alt Sistemde hazırlanan mezuniyet durumuna gelmiş öğrenci listelerini ekrana getirir.<br>3. Yönetici mezun et butonuna basar.<br>4. Sistem öğrencilere birer diploma numarası verir ve arşivli duruma çeker.<br>5. Akışı sonlandırır. |
| Alternatif İş Akışı |
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 25/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.34 Mezun Etme Kullanım Durum Senaryosu

**SANALKAMPUS.SN30 Not Düzeltme**

| | |
|---|---|
| Senaryo # | 30 |
| Senaryo İsmi | Not Düzeltme |
| Kapsam | İş Süreci (Şeffaf-Kutu) |
| Hedef Seviye | Genel Senaryolar |
| Aktör(ler) | Yönetici |
| Hedef | Bu senaryoda Yöneticinin herhangi bir öğrencinin notunu düzeltmesi hedeflenir. |
| Amaç | Yöneticinin uzaktan eğitimde kişisel hatalardan kaynaklanan öğrenci notlarında ki yanlışlıkların yönetim kurulu kararıyla düzeltilebilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Yönetici sisteme oturum açmış olmalı.<br>Yönetici not düzeltmesi yapacağı öğrenciyi seçmiş olmalı. |
| Başarılı bitiş durumları | Yönetici öğrencinin hatalı olan notunu başrı ile düzeltmiştir. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Yönetici not düzelt butonuna basar.<br>2. Sistem Yöneticiye öğrencinin dönem derslerini listeler.<br>3. Yönetici notu düzelteceği dersi seçer.<br>4. Sistem Yöneticiye seçilen dersten öğrencinin almış olduğu notları listeler.<br>5. Yönetici notu düzeltmeden önce Yönetim Kurulu kararı bilgilerini sisteme girer.<br>6. Yönetici Notu düzeltir. Kaydet butonuna basar.<br>7. Sistem Yönetim Kurulu kararı bilgilerini sisteme kaydeder. Öğrencinin notunu veritabanından düzeltir.<br>8. Akışı sonlandırır. |
| Alternatif İş Akışı |
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 25/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.35 Not Düzeltme Kullanım Durum Senaryosu

**SANALKAMPUS.SN31 Not Hesapla**

| | |
|---|---|
| Senaryo # | 31 |
| Senaryo İsmi | Not Hesapla |
| Kapsam | Sistem (Kara-Kutu) |
| Hedef Seviye | Detay Fonksiyonlar |
| Aktör(ler) | Alt Sistem |
| Hedef | Bu senaryoda Alt Sistemin sınav notunu hesaplaması hedeflenir. |
| Amaç | Sistemde bulunan alt yordamların öğrenci sınavını tamamladığı anda sınav notunu hesaplaması amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Öğrenci sınavını tamamlamış olmalı. |
| Başarılı bitiş durumları | Alt Sistem öğrencinin sınav notunu başarı ile hesaplamıştır. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Öğrenci sınavını tamamlar ve sınavı bitir butonuna basar.<br>2. Alt Sistem öğrencinin cevaplarını alır ve öğretmenin sisteme girmiş olduğu doğru cevaplarla karşılaştırır.<br>3. Doğru cevapları daha önce belirlenen formüle uygulayarak öğrencinin puanını hesaplar.<br>4. Hesaplanan puanı veritabanına kaydeder. Akışı sonlandırır. |
| Alternatif İş Akışı |
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 28/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.36 Not Hesapla Kullanım Durum Senaryosu

**SANALKAMPUS.SN32 Sınav Oluştur**

| | |
|---|---|
| Senaryo # | 32 |
| Senaryo İsmi | Sınav Oluştur |
| Kapsam | Sistem (Kara-Kutu) |
| Hedef Seviye | Detay Fonksiyonlar |
| Aktör(ler) | Alt Sistem |
| Hedef | Bu senaryoda öğrenciye sınav oluşturmak hedeflenir. |
| Amaç | Genel sınav ve ara sınav olacak her öğrenciye farklı sorulardan oluşan bir sınav oluşturmak amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | Öğrenci sınavı başlatmış olmalı. |
| Başarılı bitiş durumları | Alt Sistem öğrenciye veritabanındaki sorulardan bir sınav hazırlamıştır. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Öğrenci Sınavı başlat butonuna basar.<br>2. Alt Sistem öğrencinin seçmiş olduğu ders veya bölümün veritabanındaki ilgili sorularından daha önce belirlenen formüle göre soruları oluşturur.<br>3. Soruları web modülüne aktarır.<br>4. Akışı sonlandırır. |
| Alternatif İş Akışı |
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 28/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.37 Sınav Oluştur Kullanım Durum Senaryosu

**SANALKAMPUS.SN33 Mezun Listesi Oluşturma**

| | |
|---|---|
| Senaryo # | 33 |
| Senaryo İsmi | Mezun Listesi Oluşturma |
| Kapsam | Sistem (Kara-Kutu) |
| Hedef Seviye | Detay Fonksiyonlar |
| Aktör(ler) | Alt Sistem |
| Hedef | Bu senaryoda mezuniyet durumuna gelmiş öğrencilerin listesini oluşturmak hedeflenir. |
| Amaç | Yöneticinin yıl sonu ve ara dönem de mezuniyet durumuna gelen öğrencileri tespit edebilmesi amaçlanmıştır. |
| Olgunluk | Sarı |
| Sorular | |
| Öndurumlar | |
| Başarılı bitiş durumları | Alt Sistem yöneticiye mezuniyet durumuna gelen öğrencilerin listesini hazırlamıştır. |
| Minimum bitiş durumları | |

| Ana İş Akışı |
|---|
| 1. Yönetici mezuniyet listeleri butonuna basar.<br>2. Alt Sistem bölümlere göre maksimum yıl ve üzerinde olan öğrencileri bulur.<br>3. Bu öğrencilerin tüm derslerini geçenleri seçer ve listeler.<br>4. Akışı sonlandırır. |

| Alternatif İş Akışı |
|---|
| |

| Belge Tarihçesi | | | |
|---|---|---|---|
| Tarih | Sürüm | Açıklama | İsim/Soyisim |
| 28/Mayıs/2007 | 0,1 | İlk sürüm oluşturuldu | Sait Ali Uymaz |

Çizelge 4.38 Mezun Listesi Oluşturma Kullanım Durum Senaryosu

Kullanım durum senaryoları ile uzaktan eğitim sisteminin genel durumu ve müşteri gereksinimleri ortaya koyuldu. Sistemi kullanacak olan firma yetkilileri kullanıcı bakışı bölümünde ortaya çıkarılan genel senaryoları inceleyerek bu uzaktan eğitim sistemi hakkında genel bir bilgiye sahip olabilirler.

## 4.2 2. Aşama: Yapısal Bakış

Kullanım durum senaryoları oluşturduktan sonra bu senaryolar üzerinde ki sınıfları ortaya koymak gerekmektedir. Bu senaryolara dayanarak ve genelleme yaparak; Üniversite, Fakülte, Bölüm, Sınıf, Ders, Konu, Doküman, Soru, Alt Sistem, Yönetici, Öğretmen, Öğrenci, Adres, Oturum isimli 14 tane sınıf seçilebilir.

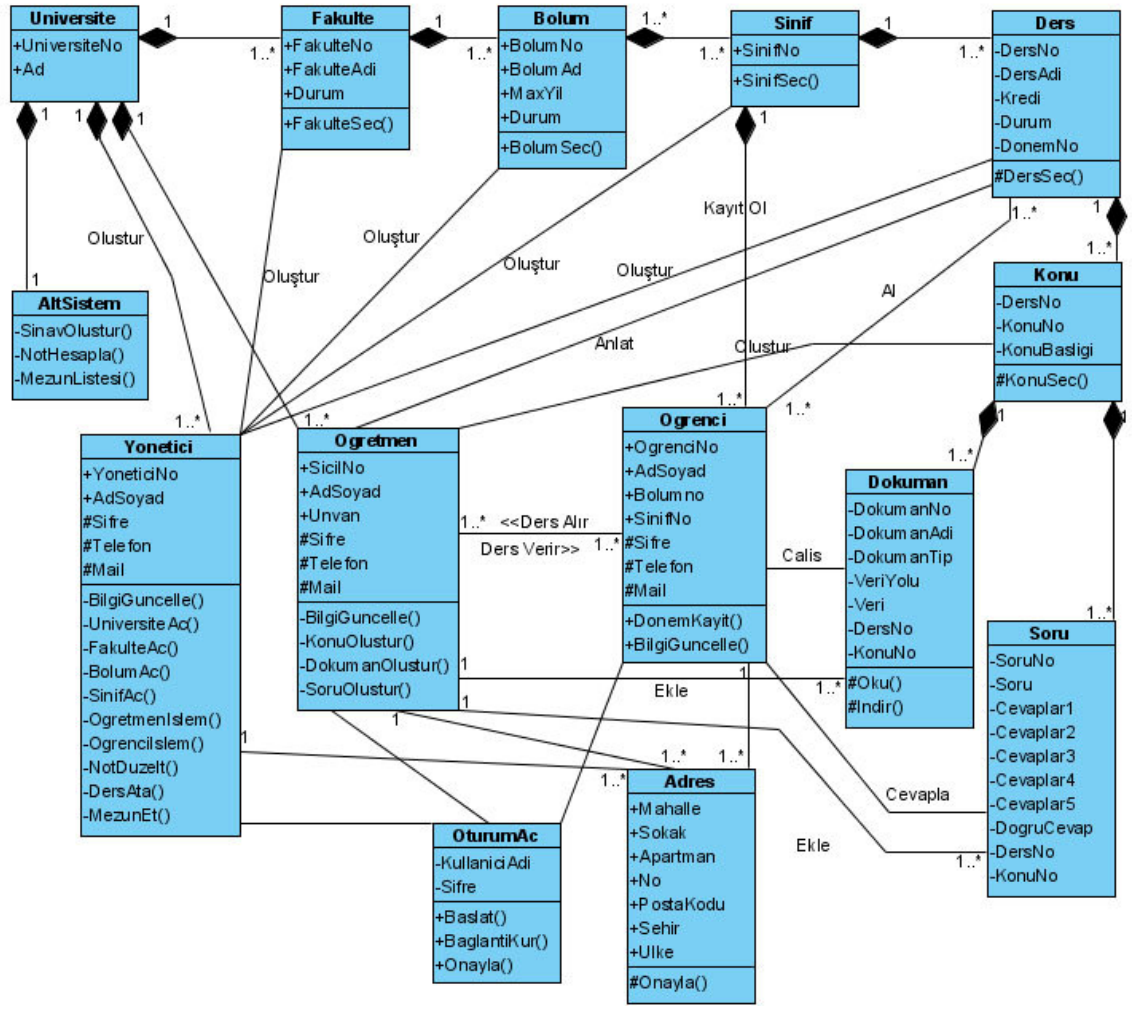
Sınıf diyagramımızı çizerken öncelikle sınıflarımızın özelliklerini, işlevlerini ve diğer sınıflar ile ilişkilerini belirlememiz gerekir. Örnek olarak Adres sınıfına bakmak gerekirse;

- Adres sınıfımızın özellikleri mahalle, sokak, apartman, no, posta kodu, şehir ve ülkedir.
- Adres sınıfımızın işlevi ise onayladır.
- Adres – Yönetici arasında bire bir veya daha fazla ilişki var.
- Adres – Öğretmen arasında bire bir veya daha fazla ilişki var.
- Adres – Öğrenci arasında bire bir veya daha fazla ilişki var.

Sınıflar arasındaki en temel ilişkiler aşağıdaki gibi listelenebilir:

1. Bire-bir
2. Bire-çok
3. Bire-bir veya daha fazla
4. Bire-sıfır veya bir
5. Bire-sınırlı aralık (mesela: bire-[0,20] Aralığı)
6. Bire-n (UML de birden çok ifadesini kullanmak için '*' simgesi kullanılır.)
7. Bire-Beş ya da Bire-sekiz.

Şekil 4.6. ile Sanal Kampus projesi için seçilen 14 sınıfın özellikleri, işlevleri ve aralarındaki ilişkiler Uml'nin sınıf diyagramı ile modellenmiştir.



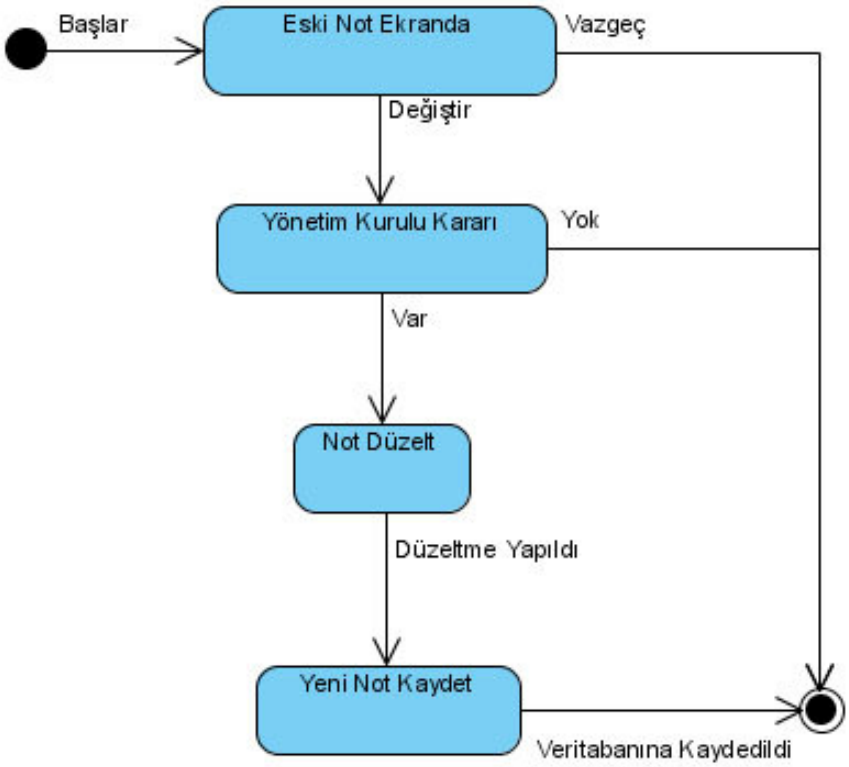

Şekil 4.6. Sanal Kampus Sınıf Diyagramı

## 4.3 3. Aşama: Davranış Bakışı

Şu ana kadar genel olarak senaryolar ve bu senaryolarda ki sınıflar, sınıflar arasındaki kavramsal ilişkiler gösterildi. Davranış bölümünde nesnelerin ve sistemin durumlarını etkileyen olaylar ve etkinlikler soyut bir düzeyde incelenecektir. Nesnelerin programın kendi akışı içerisinde gerçekte nasıl bir işbirliği yaptığı, nasıl bir etkileşimde bulunduğu modellenecektir.

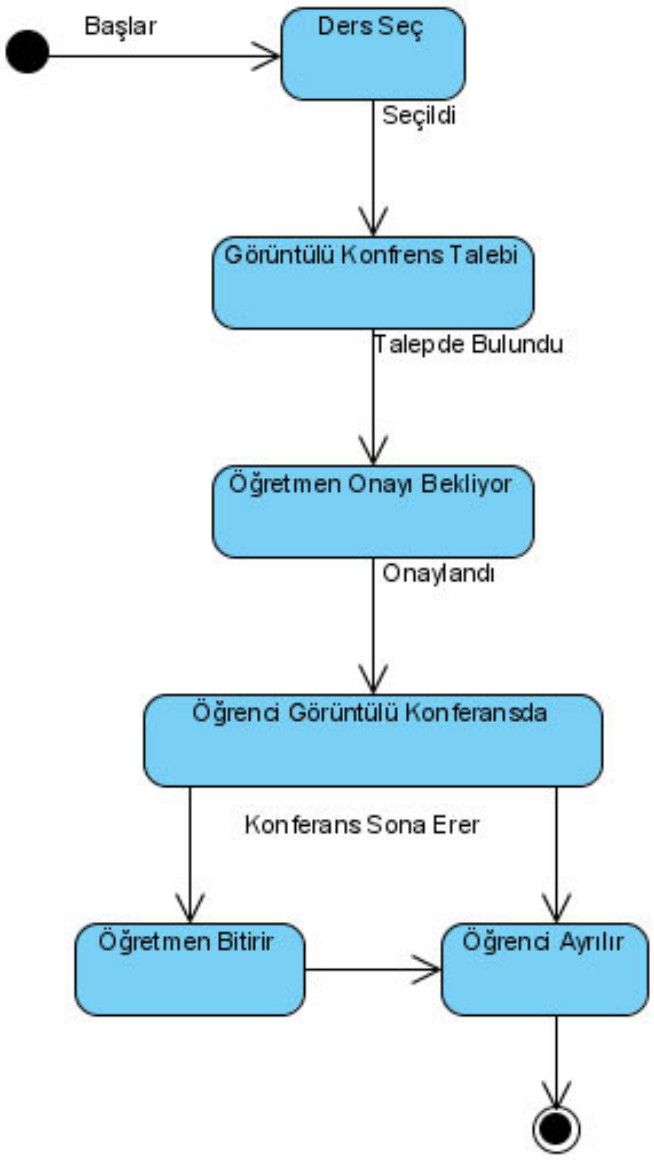
Bu modelleme de Uml'nin durum, etkinlik ve ardışık sıra diyagramlarından faydalanılacaktır.

Daha önce tanımlanan senaryolardan yola çıkarak şekil 4.7. ile yöneticinin bir öğrencinin sınav notunda yönetim kurulu kararına dayanarak düzeltme yapması Uml'nin durum şeması ile gösterilecektir.



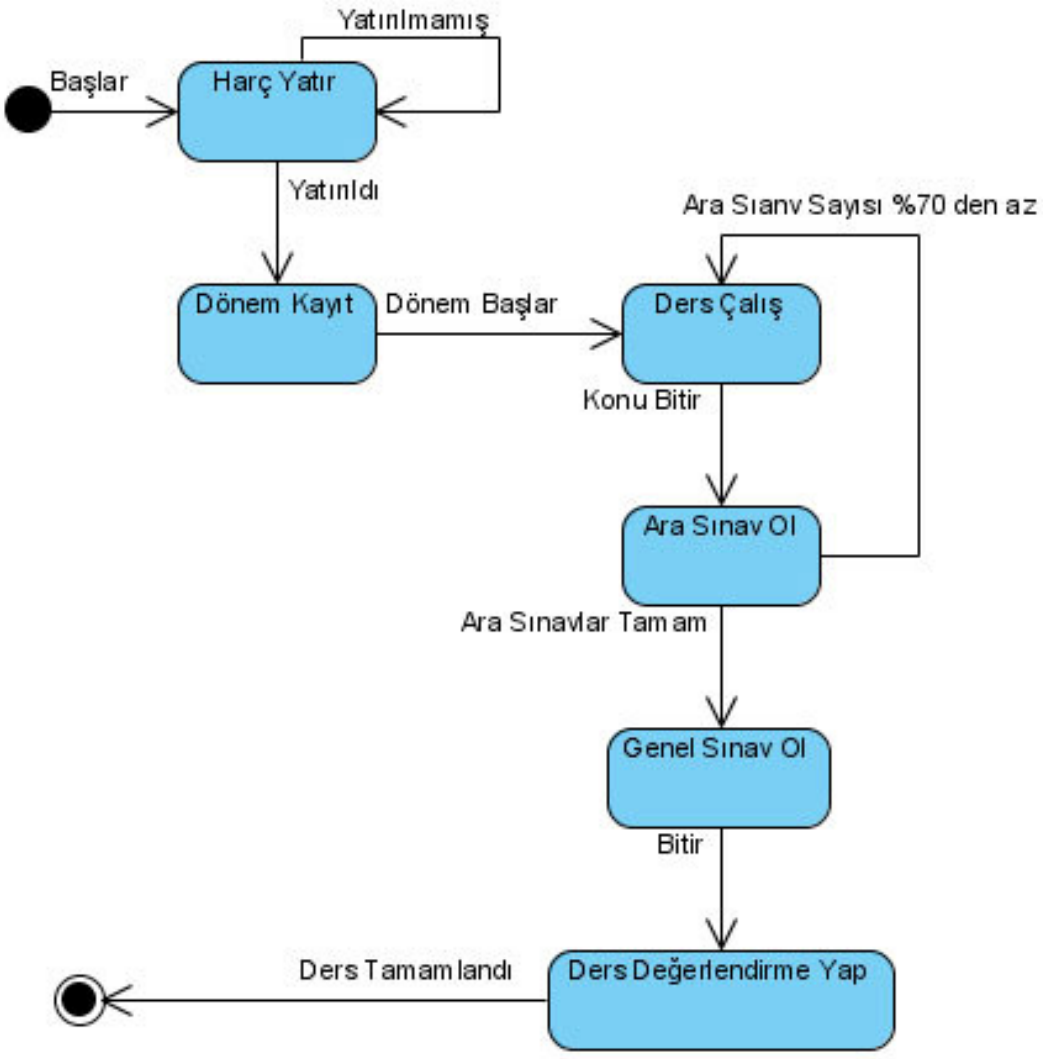

Şekil 4.7. Not Düzeltme Durum Diyagramı

Şekil 4.8. ile Öğrencinin Görüntülü konferansa katılmasını yani görüntülü konferansın başlaması ile bitişi arasındaki durumlarının gösterildiği durum senaryosu diyagramı gösterilmiştir.




Şekil 4.8. Görüntülü Konferans Durum Diyagramı

Şekil 4.9. ile öğrencinin bir dönem boyunca durumu gösterilecektir. Öğrencinin dönem harcını yatırmasıyla başlayan ve ders sonu değerlendirmelerini yapıp dönemi tamamlamasına kadar geçirdiği durumlar bu diyagramda gösterilecektir.

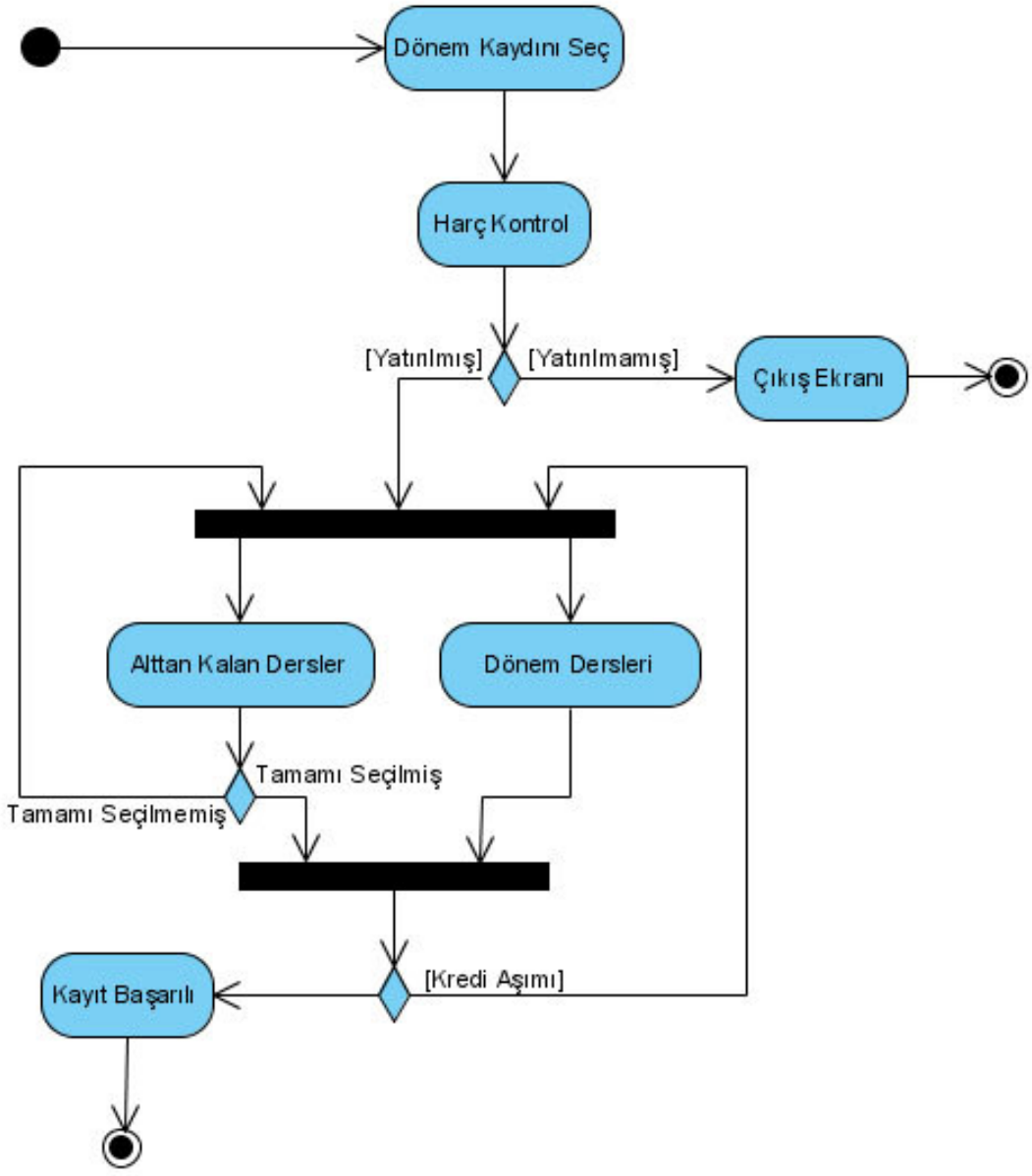



Şekil 4.9. Öğrenci Durum Diyagramı

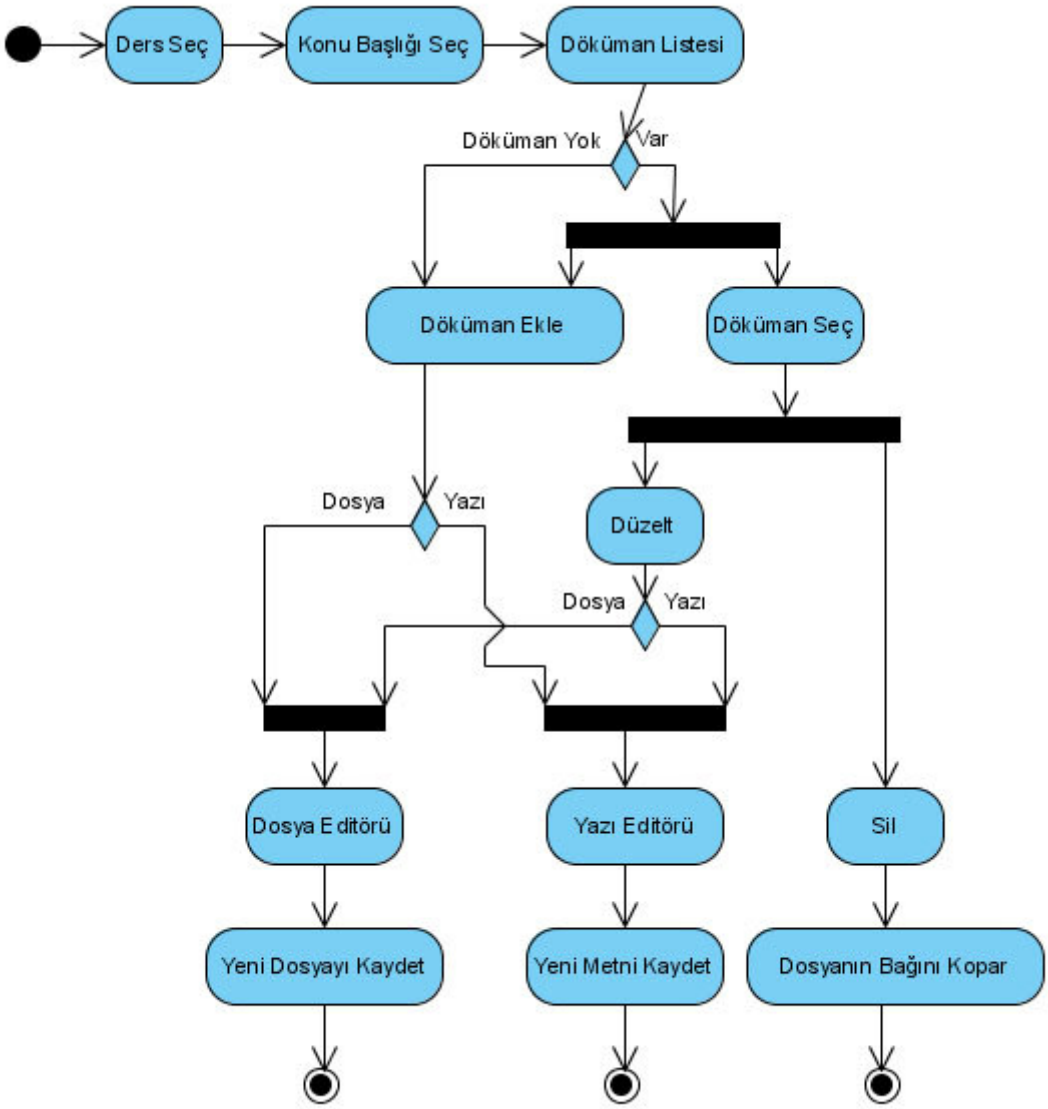
Durum diyagramları ile nesnelerin süreç tamamlanana kadar aldığı durumlar gösterilmektedir. Bu durumlar sırasında yapması gereken aktiviteleri ise etkinlik şemaları ile gösterelim. Etkinlik diyagramları bir davranışın giriş ve çıkışlarını göstermektedir. Etkinlik diyagramları akış diyagramlarına benzetilir fakat etkinlik

diyagramları çok daha gelişmiş gösterimlere sahiptir. Etkinlik diyagramlarının en belirgin avantajı paralel davranışları desteklemesi ve alternatif sonlara izin vermesidir. Ayrıca ayrılan akışların nerede birleşecekleri de gösterilebilmektedir.

Şekil 4.10 ile öğrencinin dönem kaydını yapması ve şekil 4.11 ile öğretmenin sorumlu olduğu bir derse doküman eklemesi etkinlik diyagramı ile gösterilecektir.



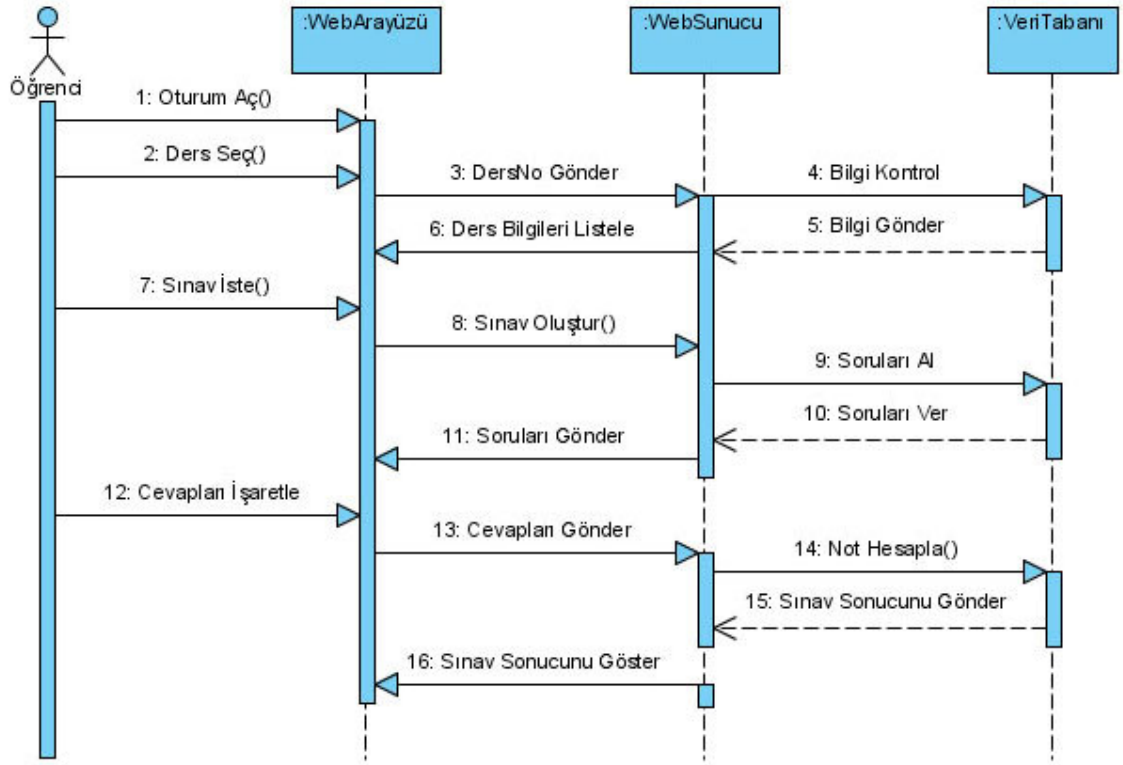

Şekil 4.10. Dönem Kayıt Etkinlik Diyagramı



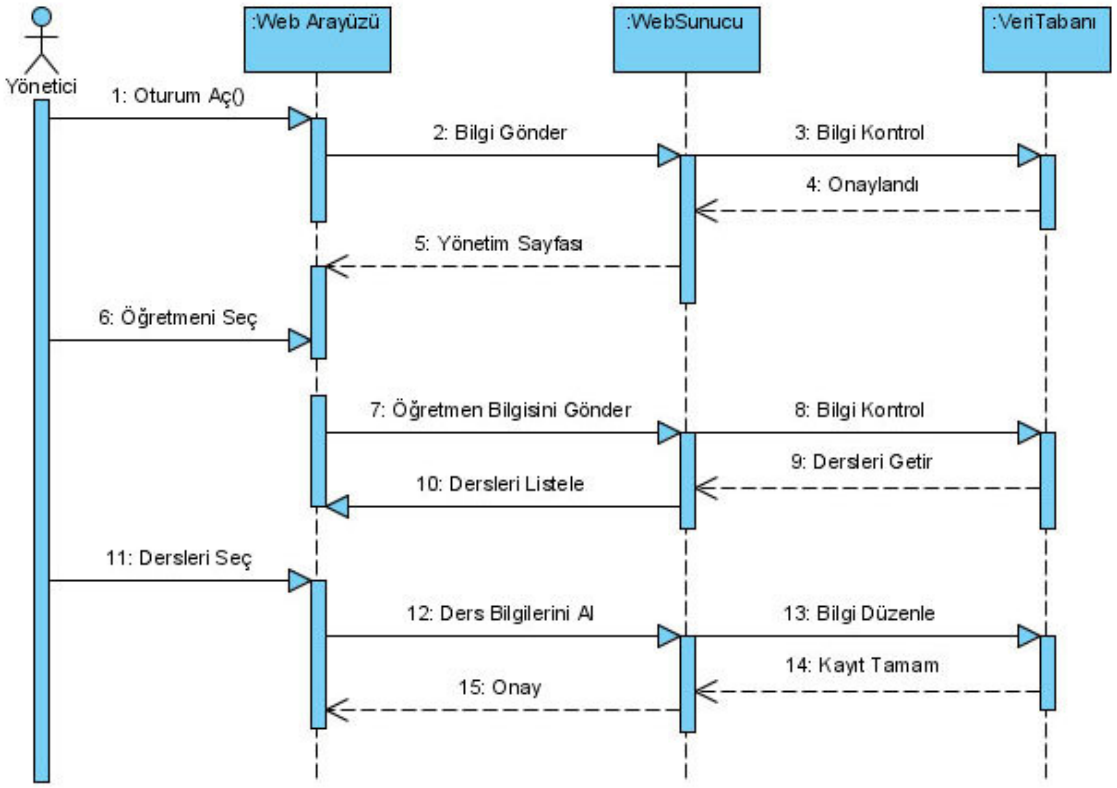

Şekil 4.11. Doküman Ekleme Etkinlik Diyagramı

Ardışık sıra diyagramları ile nesnelerin birbirleri ile haberleşmesi zamana bağlı olarak ele alınır. Bir diyagramın zamana bağlı olmasından kasıt, nesnelerin gerçekleştirdikleri aktivitelerin peşi sıra gerçekleşmesi ve bu peşi sıralığın belirlenen zaman dilimleri içerisinde meydana gelmesidir.

Ardışık sıra diyagramlarında yukarıdan aşağıya doğru inen hatlar bizim zaman çizgimizi gösteriyor. Soldan sağa doğru sistemde ilerleyen oklar mesaj ya da emirleri, sağdan sola doğru gelen oklar ise yanıtları belirtiyor.




Şekil 4.12. Sınav Ardışık Sıra Diyagramı



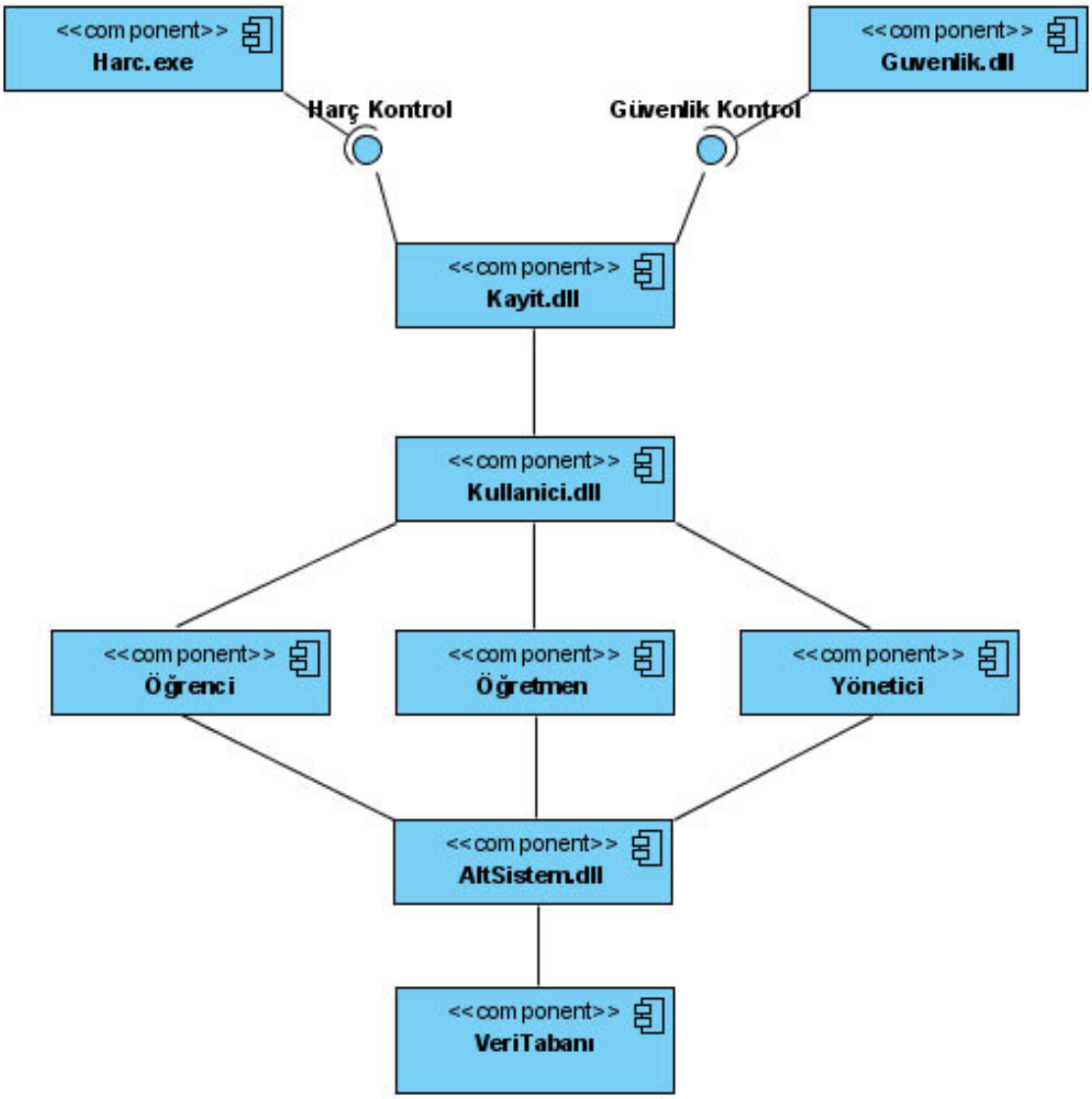

Şekil 4.13. Öğretmen Ders Atama Ardışık Sıra Diyagramı

## 4.4 4. Aşama: Gerçekleştirme Bakışı

Gerçekleme bakışı, geliştirilen yazılım sisteminin alt modüllerini ortaya koymayı hedefler. Bu amaçla kullanılan modelleme elemanları, bileşenler ve paketlerdir. Paketler; yazılımın üst seviyedeki fiziksel parçaları olup katmanlar halinde oluşturulur. Her katmanın bir ara yüzü bulunmaktadır. Geliştirilecek sistemin gereksinimlerine bağlı olarak, bu paketlerin sayısını arttırabilir veya azaltabilirsiniz.

Genel olarak; tek başına çalışabilen alt modülleri bileşen olarak düşünebilirsiniz. Bu bileşenler; yer değiştirilebilir olmalı ve birini çıkarıp diğerini yerleştirmek, sistem

üzerinde fazla değişikliğe yol açmamalıdır. Bu amaçla; bileşenlerin ara yüzlerinin gerçekleniyor olması yeterlidir. Bileşenler sayesinde, sistem daha esnek bir yapıya kavuşurken bakım yapılabilirliği de artmaktadır. Aynı ara yüzü sağlayan başka bir bileşen sisteme entegre edildiği durumda sistem, mevcut yapıda herhangi bir değişiklik olmadan çalışmaya devam edecektir.

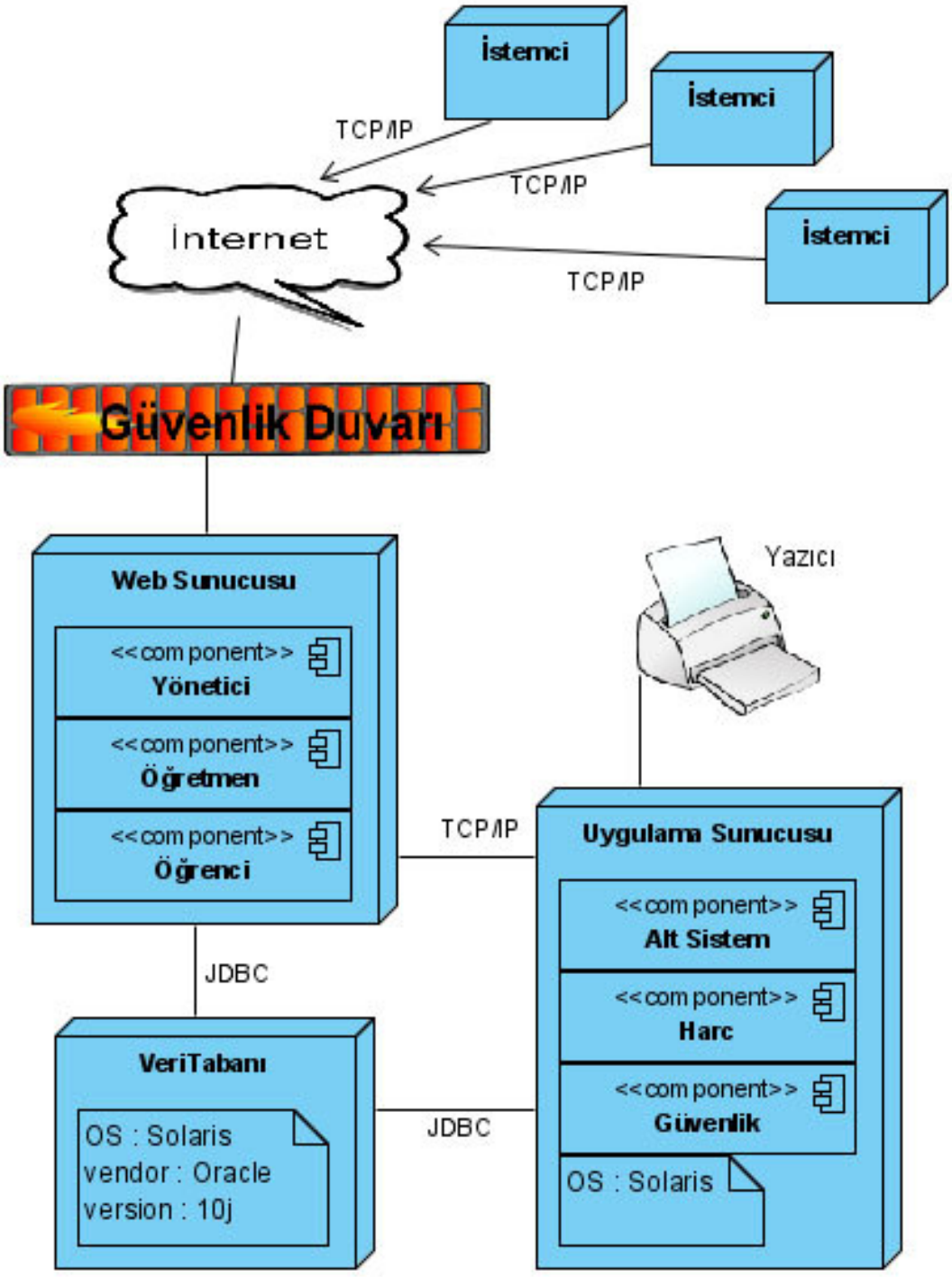



Şekil 4.14. Sanal Kampus Bileşen Diyagramı

Bu modelimizde yazılımı oluşturan fiziksel bileşenlerin modellendiği, bileşen diyagramı oluşturuldu. Bu diyagramda asıl amaç; bileşenler arasındaki bağımlılıkları belirlemek ve bileşenler arasındaki organizasyonu net olarak ortaya koyabilmektir. Projelerdeki detaylı tasarımının sonunda, bu diyagramları oluşturarak hangi kaynak kodların yazılması gerektiği saptanabilir ve bu kodlar programcılar arasında paylaştırılabilir. Son olarak; bu diyagramlar, sistemin tüm parçalarını içermesi ve birçok ekip üyesi tarafından incelenebilir olması nedeni ile daha dikkatli hazırlanmalı ve kolay anlaşılır bir yapıda oluşturulmalıdır.

## 4.5 5. Aşama: Ortam Bakışı

Ortam Bakışı, Donanımın (üzerinde yazılımın çalışacağı) fiziksel mimarisinin ortaya konduğu bakış açısıdır. Bu sistemin nasıl çalışacağını gösterirken kullanılan en üst düzey bakışlardan birisi de sistemi kurarken fiziksel olarak kullanılacak olan bilgisayarların, diğer cihazların, barındırdıkları bileşenlerin ve bu cihazların nasıl iletişim kuracaklarının gösterildiği ortam bakışıdır. Yaygınlaştırma diyagramları ile bu görünümleri sunabiliriz. Bileşenleri gösterebilmek için Bileşen diyagramlarını Yaygınlaştırma diyagramlarının içinde etkin olarak kullanabiliriz.




Şekil 4.15. Sanal Kampus Yaygınlaştırma Diyagramı

# 5 SONUÇ

Yükseköğretimdeki eğilimler üniversiteleri elektronik ortamları kullanmaya itmekte ve öğrencilerine çevrimiçi olanaklar sunmalarını gerektirmektedir. Üniversitelerin iletişim ve bilgi teknolojilerine önem verip çevrimiçi eğitim altyapılarına yatırım yapmalarına sebep olan gerekçeleri de şöyle sıralayabiliriz:

- Yükseköğretime olan ihtiyacın ve talebin artması,
- Yükseköğretim alanların profillerinin değişmesi,
- Öğrencilerin durumlarına uygun ders veya program arayışına girmesi,
- Geleneksel yerleşke anlayışının değişmesi ve kâr öncelikli eğitim kurumlarının sayısının artması,
- Öğretimin daha öğrenci merkezli, doğrusal olmayan ve öğrenci yönetimli hale gelmesi,
- Akademik vurgunun ders bitirmekten çok yeterliliğe önem vermesi,
- Yükseköğretime akan kaynakların azalması ve kurumların ortak arayışına girmesi,
- Teknolojinin çok yönlü ve yaygın hale gelmesi
- İnternet kullanımındaki hızlı artış,
- Yaşam boyu öğrenimin rekabet için gerekli hale gelmesi,
- Daha çok dersin, derece programlarını ve üniversitenin çevrimiçi uzaktan eğitime geçmesi,
- Uzaktan eğitimde İnternetin baskın ortam haline gelmesi,
- Uzaktan ve yerel eğitim arasındaki farkın azalmaya başlaması.

Özetle, yükseköğretimdeki birçok eğilim çevrimiçi eğitimi etkilemektedir. Öğrenci sayıları artmakta, geleneksel altyapı imkânları yetersiz kalmakta, öğrenci profilleri değişmekte, öğrenciler ihtiyaçlarına göre –marketlerde alışveriş yapar gibi – ders ve program arayışına girmektedir. Kurumların ve yükseköğretimin yapısı akademik

uygunluğu, rekabeti, içerik standardizasyonunu ve öğrenci-müşteri ihtiyaçlarına adapte olabilmeyi ön plana çıkarmaya başlamıştır. İnternet ve diğer bilgi teknolojileri daha da yaygınlaşmakta iş gücünün bu teknolojileri kullanabilmesi beklenmektedir. Yaşam boyu öğrenmenin bir gereklilik olması ile beraber çevrimiçi yükseköğretim daha da yaygınlaşmakta ve konumdan bağımsız hale gelmektedir. Bu eğilimler ile beraber yükseköğretim kurumları öğrenci ihtiyaçlarını karşılamak için geleneksel eğitim ile yapamadıklarını çevrimiçi eğitim ile yapmaya yönelmektedirler. Bu dönüşümde yaşanabilecek en büyük sorun vizyon eksikliği ve teknolojinin kullanımında yapılacak stratejik hatalar olabilir. Burada dikkat edilecek husus çevrimiçi eğitimin kurumun geleceği için neler vaat ettiği ve kurumun görevlerindeki yeridir.

Görüldüğü üzere iyi planladığı takdirde inanılmaz getirileri olan İnternet tabanlı uzaktan eğitim konusunda, vizyon sahibi işletmelerin kayıtsız kalmaları beklenemez. Ancak her işletme için en iyi çözümün İnternet tabanlı uzaktan eğitim olduğunu söylemek zordur. Bu açıdan İnternet tabanlı uzaktan eğitimin işletmenin ihtiyaçlarına uygun bir çözüm olup olmayacağı konusunda ciddi bir çalışma ve hazırlık gerekir. İnternet tabanlı uzaktan eğitim platformunun uygulama için hazır hale gelmesi oldukça maliyetli bir işlem olduğundan, işletmeler iyice düşünerek karar vermelidir.

Çok iyi getirisi olan bir internet tabanlı uzaktan eğitim sistemi de çok iyi düşünülmüş önceden tasarlanmış iyi analiz edilmiş bir sistem olmalı ve öğrenciye internet üzerinden eğitimini tamamlayabilmesi için bütün internet tabanlı teknolojik kaynakları sunmalıdır.

# 6 KAYNAKLAR


1. Atıcı ve Gürol (2001), Nesnelci Öğretim Yaklaşımlarından Oluşturmacı Öğrenme Yaklaşımlarına Doğru İnternet Tabanlı Uzaktan Eğitime Yönelik Gelişimsel Bir Model Önerisi. Bilişim Teknolojileri Işığında Eğitim (BTIE), 3–5 Mayıs 2001, ODTÜ Kültür ve Kongre Merkezi: Ankara, s. 177–183.
2. Bay ve Tüzün (2002),Yüksek Öğretim Kurumlarında Ders İçeriğinin Web Tabanlı Olarak Aktarılması, Journal of Polytechnic Vol: 5 No: 1 pp. 13-22.
3. Silveira ve ark. (2003), From Socrates to PIAGET: Patterns for Distance Learning.
4. Lucca ve ark. (2003), Abstracting Business Level UML Diagrams from Web Applications.
5. Kovacic ve Skocir (2003), Development of Distance Learning System Based on Dialogue, EmOCON 2003 Ljubljana, Slovenia.
6. Beheshti ve Dado (2003), Simplified UML Techniques for System Development in an Educational Setting.
7. Ammar ve ark. (2001), Uml Archıtecture Of A Web-Based Interactıve Course Tool, Dept. of Computer Science and Electrical Engineering, West Virginia University Morgantown.
8. Markovic ve ark. (2005), Development UML model of Distance Learning System, Serbia and Montenegro, Nis, September 28 - 30, 2005
9. Doban ve Pataricza (2003), Uml Based Software Process Management, Periodica Polytechnica Ser. El. Eng. Vol. 47, No. 3–4, PP. 213–228,2003
10. Kadavová ve ark. (2006), Process Based Modeling of Virtual University, 35th International IGIP Symposium in cooperation with IEEE / ASEE / SEFI, 2006–09–18, Tallinn, Estonia
11. Kartyas ve Gati (2005), Model Based Distance Learning on the Internet Budapest Tech Polytechnical Institution Doberdo ut 6, H-1034 Budapest Hungary.

12. Learner outcomes in an asynchronous distance education environment, Carswell, A.D. ve Venkatesh, V. (2002).
13. Alkan ve ark. (1995), Eğitim Teknolojisine Giriş,
14. Aslantürk (2002), Bir Web tabanlı uzaktan eğitim sisteminin tasarlanması ve gerçekleştirilmesi, Hacettepe Üniversitesi,
15. Docent (2002), E–learning strategies for executive education and corporate training.
16. Uml ile Nesne Tabanlı Çözümleme ve Tasarım, Bora Güngören, Seçkin Yayıncılık, 2005
17. http://www.csharpnedir.com/makale.asp?cat=uml
18.http://vezir.analystdeveloper.com/vezir/default.aspx/MyWiki/UMLVeCBDIleYazilimGelistirme.html
19. http://www.cde.psu.edu/de/ide/guiding_principles/ssi/section_1.html
20. http://www.uidaho.edu/eo/dist1.html
21. http://www.cdlponline.org/index.cfm?fuseaction=whatis