

İNÖNÜ ÜNİVERSİTESİ



YAZILIM GEREKSİNİMLERİ VE ANALİZ DERSİ

Öğrenci Bilgi Sistemi(OBS)

İçindekiler

1.Giriş

1.1 Amaç

1.2 Belge Kuralları

1.3 Proje Kapsamı

1.4 Referanslar

2.Genel Açıklama

2.1 Ürün Perspektifi

2.2 Kullanıcı Sınıfları Ve Özellikleri

2.3 Çalışma Ortamı

2.4 Tasarım Ve Uygulama Kısıtlamaları

2.5 Varsayımlar Ve Bağımlılıklar

3. Sistem Özellikleri

3.1 Yönetici Özellikleri

3.2 Öğrenci Özellikleri

3.3 Öğretmen Özellikleri

3.4 Veli Özellikleri

4. Veri Gereklilikleri

4.1 Mantıksal Veri Modeli

4.2 Veri Sözlüğü

4.3 Raporlar

4.4 Veri Toplama, Bütünlük, Saklama Ve İmha Etme

5. Harici Arabirim Gereksinimleri

5.1 Kullanıcı Arayüzleri

5.2 Yazılım Arayüzleri

5.3 Donanım Arayüzleri

5.4 İletişim Arayüzleri

6.Kalite Özellikleri

6.1 Kullanılabilirlik

6.2 Performans

6.3 Güvenlik(Security)

6.4 Güvenlik(Safety)

7. Uluslararasılaştırma ve yerelleştirme gereklilikleri

8. Diğer Gereksinimler

9.Sonuç

9.1 Sözlük

9.2 Referanslar

1.Giriş

Bu SRS dokümanı, OBS'nin başarılı bir şekilde geliştirilmesi, uygulanması ve sürdürülmesi için bir temel oluşturmayı amaçlamaktadır. Bu belgenin, tüm proje sürecinde doğru ve eksiksiz bir anlayış sağlamak için dikkatlice takip edilmesi ve kullanılması önemlidir.

Günümüzde bilginin kaynağı ve bu kaynakla ilgili kişiler arasındaki iletişim büyük önem kazanmıştır. Hızla gelişen teknolojiler ve bu gelişmelere her alanlarda ayak uydurmaya çalışan insanoğlu eğitim alanında da gelişmelerden de maksimum düzeyde faydalanma için çalışmalarını sürekli geliştirmektedir. Bu noktada akla bu konu ile ilgili çeşitli sorular gelmektedir. Bilişim teknolojileri yardımıyla lisansüstü eğitimde bilgi akışını ve iletişimi düzenleyebilir miyiz? Bu tür bir sistem ne gibi özelliklere sahip olmalıdır? Sorularına verilecek cevaplar aşağıda ifade edilmiştir. Bu tür sistemler; Kolay; kullanımı kolay olmalı, kullanıcılar her türlü bilgiye rahatça ve her yerden ulaşabilmeli, her formatta raporlama yapılabilmesi, sistemdeki her şey yeniden tanımlanabilecek kadar esnek olmalıdır. Güvenli; kurumsal veri güvenliğine önem verilmeli, veriler otomatik olarak yedeklenmeli, geçmiş yıllara ve dönemlere ait verilerin güvenliği sağlanabilmelidir. Güçlü; sistem, iletişimin güçlü olması için web tabanlı çalışmalı, sınırsız kullanıcı desteği olmalıdır. Ekonomik; toplam sahip olma maliyeti düşük olmalı, çok kullanıcı olmasına rağmen ek maliyet gerektirmemeli, sadece merkezi bir bilgisayarda kurulması yeterli olmalı, bu sayede değişiklikler ve güncellemeler bakımından da ekonomik davranmalı, Enstitünün yönetmelik değişiklikleri sisteme kolay ve hızlı bir şekilde adapte edilebilmelidir.

İdeal bir öğrenci bilgi sisteminde yukarıdakilerin hepsi olmalı, komple bir çözüm sunabilmeli. Dolayısıyla kullanımı kolay, güvenli, güçlü ve ekonomik olmalıdır (Yıldırım, 2007).

Gelişen bilgi teknolojileri, eğitim kurumlarının kendi içinde kullandıkları bilgi sistemlerini de yeniden düzenleme ihtiyacını gündeme getirmektedir. Bazı üniversitelerde daha önce kurulmuş, eski teknolojilere dayanan ve birbiriyle bütünleşik olmayan, sınırlı sayıda kullanıcının erişebildiği bilgi sistemleri artık günümüz ihtiyaçlarını karşılayamamaktadır. Gelişen PC, istemci/sunucu ve ağ teknolojileri, yaygın, ucuz ve kullanımı kolay bilgi sistemlerinin geliştirilmesine olanak tanımaktadır. Bu teknolojileri kullanarak, üniversite içinde var olan sistemleri, artan ihtiyaçları karşılayabilecek tümleşik bir bilgi sistemi haline dönüştürmek kaçınılmaz hale gelmiştir (Balki, 2004).

Bu bölüm, OBS (Öğrenci Bilgi Sistemi & Eğitim Kurumu Bilgi Sistemi) için hazırlanan SRS

dokümanının genel bir girişini sağlar. Aşağıda, bu bölümde bulunan alt bölümler ve onların amacı detaylandırılmıştır:

1.1 Amaç

Bu SRS dokümanının amacı, OBS (Öğrenci Bilgi Sistemi & Eğitim Kurumu Bilgi Sistemi) için gereksinimleri belirlemektir. Bu belge, OBS'nin tasarımı ve geliştirilmesi sürecinde yer alan tüm paydaşların ihtiyaçlarını ve beklentilerini karşılamayı hedeflemektedir.

Bu araştırmada, pek çok üniversitenin öğrenci bilgi sistemi içinden Ankara ve Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Bahçeşehir Üniversitesi Sosyal Bilimler Enstitüsü ile Cambridge Üniversitesi öğrenci bilgi sistemlerinin verimliliği araştırılmıştır. Yapılan çalışmada, örneklem bilgilerinden bir evrene genelleme yapma amacı güdülmendiğinden bu araştırma, ilişkisel tarama modellerinden karşılaştırma türünde, betimsel nitelikli bir çalışmadır. Betimleme yöntemi, olayların, objelerin, varlıkların, kurumların, grupların ve çeşitli alanların “ne” olduğunu betimlemeye veya ikiden daha çok özelliğinin ilişkisel düzeyde “nasıl” olduğunu açıklamaya çalışır ve belli bir zaman kesiti içinde çok sayıda denek ya da objeden elde edilen verilerin analizi ile araştırma problemine yanıt aranır (Kaptan, 1998; Arseven, 2001).

OBS'nin amacı, bir eğitim kurumunun öğrenci bilgilerini etkin bir şekilde yönetmek, akademik süreçleri izlemek ve eğitim kaynaklarını düzenlemek için bir bütün olarak bir çözüm sunmaktır. Bu sistem, öğrenci kayıtlarının takibi, not yönetimi, ders programı oluşturma, sınav takvimi oluşturma, öğrenci devamsızlık kontrolü, öğretmenlerin öğrenci performansını değerlendirmesi gibi işlevleri içermektedir.

Bu SRS dokümanı, geliştiricilere, proje yöneticilerine, pazarlama personeline, kullanıcılara, testçilere ve belge yazarlarına yol göstermek için tasarlanmıştır. Bu paydaşların, sistemin gereksinimlerini anlamalarını, doğru bir şekilde uygulamalarını ve sistemi başarılı bir şekilde kullanmalarını sağlamak için bu belge detaylı bir açıklama sunmaktadır.

SRS dokümanının amaçları şunlardır:

- OBS'nin genel gereksinimlerini belirlemek ve tanımlamak.
- OBS'nin işlevsel ve işlemsel gereksinimlerini açıklamak.

- OBS'nin performans, güvenlik, kullanılabilirlik ve diğer sistem gereksinimlerini belirlemek.
- OBS'nin kullanıcı arayüzü, veritabanı yapısı, veri alışverişi ve diğer teknik gereksinimlerini belirlemek.
- OBS'nin sınırlamalarını, varsayımlarını ve bağımlılıklarını tanımlamak.
- OBS'nin tasarım ve geliştirme sürecinde rehberlik etmek ve paydaşlar arasında bir referans noktası sağlamak.

1.2 Belge Kuralları

Bu alt bölümde, OBS için hazırlanan SRS dokümanının düzenlenmesinde ve kullanılmasında izlenen belge kuralları açıklanır. Aşağıda, belge kurallarının ayrıntıları verilmiştir:

Metin Stilleri : Belgede kullanılan metin stilleri, farklı bölümleri, başlıkları, alt başlıkları ve paragrafları ayırt etmek için kullanılır. Başlıklar genellikle kalın veya büyük puntıyla belirtilirken, alt başlıklar ve normal metin genellikle düz veya orta puntıyla yazılır. Özel vurgular veya önemli bilgiler için italik veya koyu metin kullanılabilir.

Vurgulamalar : Belgede belirli gereksinimlerin veya önemli bilgilerin vurgulanması gerekebilir. Bu durumda, vurgulanan metin genellikle kalın veya italik olarak belirtilir.

Gösterimler : Belgede, örnekler, kod parçacıkları veya ekran görüntüleri gibi gösterimlerin kullanılması gerekebilir. Gösterimlerin anlaşılır ve okunabilir olması için uygun biçimlendirme ve etiketleme kullanılır.

Standartlar : Belgenin hazırlanmasında kullanılan standartlar, şablonlar ve formatlar belirtilir. Bu, sayfa düzeni, başlık düzeni, numaralandırma, dipnotlar ve referanslar gibi unsurları içerir. Belgenin bütünlüğünü ve tutarlılığını sağlamak için standartlara uyulması önemlidir.

Tipografik Kurallar : Belgede kullanılan tipografik kurallar, noktalama işaretleri, başlık ve alt başlık formatları, paragraf düzenlemeleri gibi unsurları içerir. Bu kuralların amacı, belgenin okunabilirliğini ve anlaşılabilirliğini artırmaktır.

Etiketleme : SRS dokümanında yer alan gereksinimlerin veya bölümlerin etiketlenmesi, diğer belgelere veya ilgili bölümlere referans verilmesi için kullanılır. Bu, belge içinde bağlantılar oluşturulmasını ve gereksinimlerin takip edilmesini kolaylaştırır.

Güncelleme Süreci : SRS dokümanının güncellenmesi gerektiğinde, belge kurallarına uygun bir şekilde güncelleme yapılmalıdır. Güncellemelerin tarih ve sürüm bilgileriyle birlikte kaydedilmesi ve belgeye yansıtılması önemlidir.

1.3 Proje Kapsamı

Bu alt bölümde, OBS'nin proje kapsamı ve amaçları daha ayrıntılı bir şekilde açıklanır. Aşağıda, OBS'nin proje kapsamı hakkında detaylı bilgiler verilmiştir:

OBS (Öğrenci Bilgi Sistemi & Eğitim Kurumu Bilgi Sistemi), bir eğitim kurumunun öğrenci bilgilerini etkin bir şekilde yönetmek, akademik süreçleri izlemek ve eğitim kaynaklarını düzenlemek için bir çözüm sunmayı hedefler. Bu sistem, öğrencilerin kayıtlarını tutmak, ders programları oluşturmak, not yönetimini gerçekleştirmek, sınav takvimini planlamak, öğrenci devamsızlık kontrolü yapmak, öğretmenlerin öğrenci performansını değerlendirmesine yardımcı olmak gibi bir dizi işlevi içerir.

Projenin ana amacı, eğitim kurumlarının süreçlerini daha verimli hale getirmek, öğrenci bilgilerinin güvenli ve erişilebilir bir şekilde saklanmasını sağlamak ve tüm paydaşların ihtiyaçlarını karşılamaktır. OBS'nin hedefleri aşağıda belirtilmiştir:

Öğrenci Bilgilerinin Yönetimi: OBS, öğrencilerin kayıtlarını takip etmeyi, demografik bilgilerini saklamayı ve öğrenci profilini oluşturmayı amaçlar. Ayrıca, öğrenci performansı, devamsızlık durumu ve akademik geçmiş gibi bilgilerin güncel tutulmasını sağlar.

Ders Programı Oluşturma: OBS, öğrencilere ders seçme imkânı sunar ve otomatik olarak ders programlarını oluşturur. Öğrencilerin tercihlerine ve mevcut kaynaklara göre ders programının en iyi şekilde düzenlenmesini sağlar.

Not Yönetimi: OBS, öğrencilerin sınav notlarını kaydetmeyi ve not ortalamalarını hesaplamayı amaçlar. Öğretmenlere not girişi yapma, düzeltme ve not analizi yapma imkânı sağlar.

Sınav Takvimi ve Planlama: OBS, eğitim kurumunun sınav takvimini oluşturmayı ve güncellemeyi sağlar. Öğrencilerin sınav programlarını takip etmelerine ve uygun şekilde hazırlanmalarına olanak tanır.

Öğrenci Devamsızlık Kontrolü: OBS, öğrencilerin devamsızlık durumunu kaydetmeyi ve takip etmeyi amaçlar. Eğitim kurumunun devamsızlık politikalarını uygulamayı ve devamsızlık raporlarını oluşturmayı sağlar.

Proje, eğitim kurumlarının süreçlerini daha verimli hale getirerek, öğrenci bilgilerini güvenli bir şekilde yöneterek ve tüm paydaşların ihtiyaçlarını karşılayarak eğitim kalitesini artırmayı hedefler.

1.4 Referanslar

Çeken, C. (2001). Öğrenci bilgi sistemi. Yayınlanmamış Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli.

Yıldırım, T. (2007). Ajax ve uml destekli web tabanlı öğrenci bilgi sistemi portalının hazırlanması. Yayınlanmamış Yüksek Lisans Tezi, Havacılık ve Uzay Teknolojileri Enstitüsü, İstanbul.

Kaptan, S. (1998). Bilimsel araştırma ve istatistik teknikleri, Ankara: Bilim Kitap.

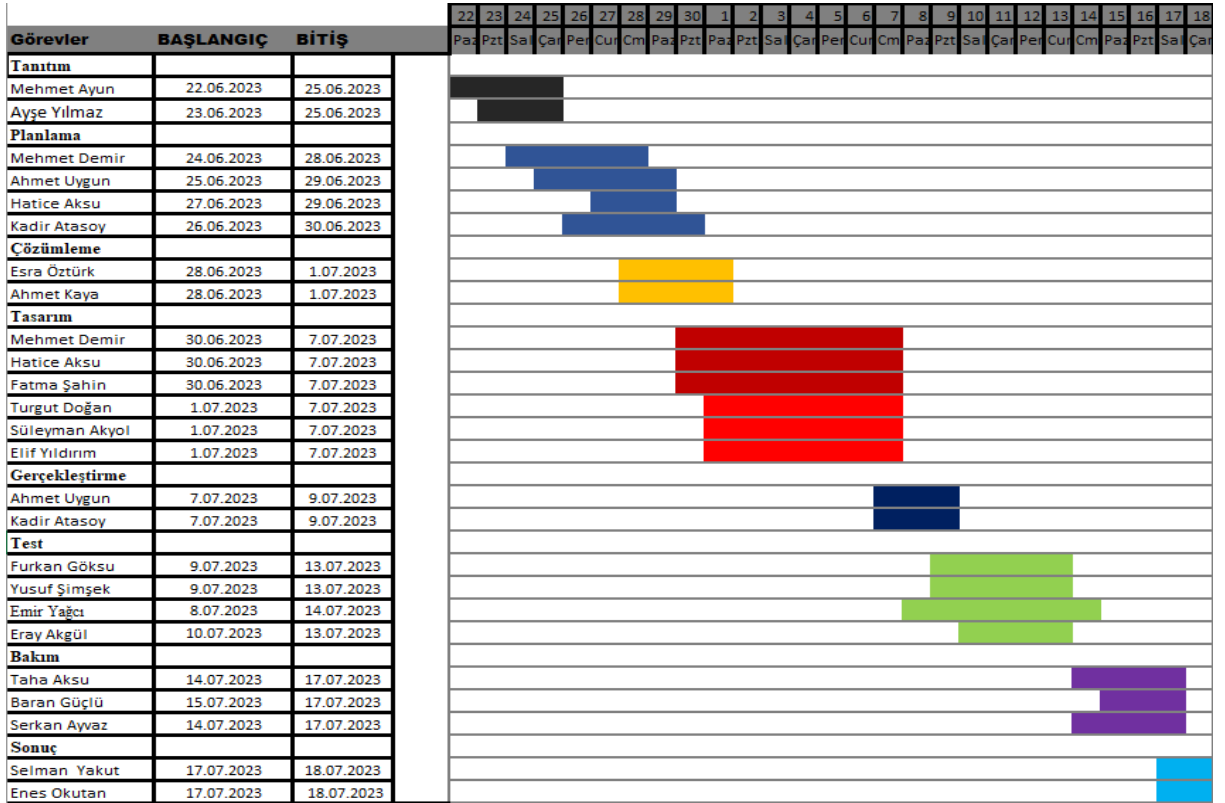
Arseven, A.D. (2001). Alan Araştırma Yöntemi, İlkeler, Teknikler, Örnekler, Ankara: Gündüz Eğitim Ve Yayıncılık.

Balki, R.S. (2004). Student and parent appraisal of learning management system in a private schoolGraduate Program İn Secondary School Science And Mathematics Education Boğaziçi University, İstanbul.

2. Genel Açıklama

Ürün Genel Bakışı:

OBS (Öğrenci Bilgi Sistemi & Eğitim Kurumu Bilgi Sistemi), modern eğitim kurumlarının ihtiyaçlarına yönelik kapsamlı bir yazılım çözümüdür. Bu sistem, eğitim kurumlarında öğrenci yönetimi, akademik kayıt, not takibi, sınıf ve ders programlama, iletişim ve diğer ilgili süreçleri desteklemek amacıyla tasarlanmıştır. OBS, kullanıcı dostu arayüzü ve esnek yapısıyla eğitim kurumlarının verimliliğini artırmayı, öğrenci verilerini etkin bir şekilde yönetmeyi ve kurum içi iletişimi geliştirmeyi hedefler.



Yukarıda görülen Gantt şeması örnek bir OBS oluşturma için gerekli ortam ve zamanı tahmini bir süre zarfında oluşturmuştur.6 adımda kişinin ne kadar çalıştığını bu şemada görebilir ve bilgi alabiliriz.

Kullanılacağı Ortam:

OBS, çeşitli eğitim kurumlarında kullanılabilen çok platformlu bir yazılımdır. Sistem, hem çevrimiçi hem de yerel sunucularda barındırılabilir ve farklı işletim sistemleri ve cihazlarla uyumlu olmalıdır. OBS, masaüstü bilgisayarlar, dizüstü bilgisayarlar, tabletler ve akıllı telefonlar gibi farklı cihazlardan erişilebilir olmalıdır. Ayrıca, sistem çevrimiçi olarak da erişilebilir olmalı ve kullanıcıların web tarayıcıları aracılığıyla kolayca erişebilmesi sağlanmalıdır.

Beklenen Kullanıcılar:

OBS, farklı eğitim kurumlarındaki kullanıcıların ihtiyaçlarını karşılamak üzere tasarlanmıştır. Aşağıda, OBS'nin beklenen kullanıcıları ve rolleri belirtilmiştir:

Öğrenciler:

OBS'yi kullanacak olan temel kullanıcı grubu öğrencilerdir. Öğrenciler, sisteme giriş yaparak kendi kişisel bilgilerini görüntüleyebilir, ders programlarını takip edebilir, notlarını kontrol edebilir, sınav takvimini inceleyebilir, ödevleri gönderebilir ve iletişim araçlarından faydalanabilir.

Öğretmenler:

Öğretmenler, OBS'yi derslerin yönetimi ve öğrenci değerlendirme süreçleri için kullanacaklardır. Sistem üzerinden ders programlarını oluşturabilir, öğrenci katılımını takip edebilir, not girişi yapabilir, ödevleri değerlendirebilir, sınavlar planlayabilir ve öğrencilerle iletişim kurabilir.

Yöneticiler ve Okul Personeli:

Okul yöneticileri ve diğer okul personeli, OBS'yi eğitim kurumunun genel yönetimi ve operasyonları için kullanacaklardır. Sistem üzerinden öğrenci kayıtlarını yönetebilir, personel bilgilerini takip edebilir, sınıf ve ders programlarını düzenleyebilir, finansal bilgileri yönetebilir, raporlar oluşturabilir ve diğer okul süreçlerini yönetebilir.

Veliler:

Veliler, OBS üzerinden çocuklarının akademik performansını takip edebilir, ders programlarını görüntüleyebilir, notlarını inceleyebilir, öğretmenlerle iletişim kurabilir ve öğrencilerin okuldaki etkinliklerini takip edebilir.

Bilinen Kısıtlamalar, Varsayımlar ve Bağımlılıklar:

OBS'nin geliştirilmesi ve kullanımı sürecinde aşağıdaki kısıtlamalar, varsayımlar ve bağımlılıklar göz önünde bulundurulmalıdır:

-OBS'nin kullanılacağı bilgisayar donanımı ve işletim sistemi gereksinimleri belirlenmelidir.

-Sistem, güvenlik ve gizlilik konularında uygun önlemler almalıdır.

-OBS, mevcut eğitim kurumlarına entegre olabilecek şekilde tasarlanmalıdır.

-Sistem, öğrenci verilerinin doğru ve güncel tutulması için düzenli yedeklemeler yapmalıdır.

-OBS'nin kullanıcı dostu bir arayüze sahip olması ve kolay kullanılabilir olması önemlidir.

-Sistem, hızlı ve güvenilir bir şekilde çalışmalı ve kullanıcıların verilere anında erişimini sağlamalıdır.

2.1 Ürün Perspektifi

OBS, bir eğitim kurumunda kullanılmak üzere tasarlanan bir yazılımdır. Bu yazılım, öğrenci bilgi sistemi ve eğitim kurumu bilgi sistemi işlevlerini yerine getirir. Ürün, eğitim kurumunun öğrencileri ve personeli arasında bilgi paylaşımını, öğrenci kayıtlarının yönetimini, ders programlarının oluşturulmasını, sınav notlarının takibini ve diğer ilgili işlemleri kolaylaştırmayı amaçlar.

OBS'nin bağlamı, mevcut bir uygulamanın yerine geçen ve genişletilen bir ürün olarak tanımlanabilir. Eğitim kurumları genellikle çeşitli veritabanları, elektronik tablolar ve manuel iş süreçleri kullanarak öğrenci bilgilerini yönetirler. OBS, bu süreçleri otomatikleştirmek ve verimliliği artırmak için geliştirilmiştir.

Eğer OBS, daha büyük bir sistem veya yazılım ekosisteminin bir bileşeni olarak tasarlanmışsa, bu durumda genel sisteme olan ilişkisini belirtmek önemlidir. OBS'nin diğer sistemlerle veri entegrasyonu sağlaması, kullanıcıların tek bir arayüz üzerinden birden fazla sisteme erişimini kolaylaştırması veya belirli sistemlerle entegre çalışması gibi senaryolar söz konusu olabilir. Bu durumda,

OBS'nin genel sistemle nasıl etkileşimde olduğunu ve arayüzlerin nasıl tasarlandığını belirtmek gerekmektedir.

Ürün perspektifi, OBS'nin mevcut durumunu, geliştirilme aşamasını ve amacını net bir şekilde açıklamalıdır. OBS, eğitim kurumlarının bilgi yönetimi süreçlerini iyileştirmek için tasarlanmıştır. Ürün, mevcut sistemlerin yerine geçebilecek, genişletilebilir ve özelleştirilebilir bir yapıya sahip olmalıdır. OBS'nin kullanıcıların ihtiyaçlarını karşılaması ve eğitim kurumlarının iş süreçlerini etkin bir şekilde desteklemesi hedeflenmelidir.

2.2 Kullanıcı Sınıfları Ve Özellikleri

2.2.1 Öğrenciler:

Öğrenci kimlik bilgilerini yönetmek: Öğrenciler, kişisel bilgilerini (ad, soyadı, iletişim bilgileri, T.C. kimlik numarası vb.) sisteme kaydetmek ve güncellemek ister.

Ders programlarını görüntülemek ve değiştirmek: Öğrenciler, ders programlarını incelemek, dersler arasında seçim yapmak ve ders programlarını düzenlemek isteyebilir.

Akademik bilgileri takip etmek: Öğrenciler, sınav notlarını, ders başarılarını, devamsızlık durumlarını ve diğer akademik bilgilerini sisteme girilmiş olarak görmek ve takip etmek isteyebilir.

Ders materyallerine erişmek: Öğrenciler, ders materyallerini (ders notları, sunumlar, ödevler vb.) görüntülemek, indirmek ve ödevlerini sisteme yüklemek isteyebilir.

2.2.2 Öğretmenler:

Ders programlarını oluşturmak ve güncellemek: Öğretmenler, derslerin planlanması, zamanlaması ve ders programının oluşturulması için sistemi kullanır.

Öğrenci kayıtlarını yönetmek: Öğretmenler, öğrencilerin kayıtlarını oluşturur, günceller ve sınıf listelerini görüntüler.

Değerlendirme materyallerini yönetmek: Öğretmenler, sınavlar, ödevler ve projeler gibi değerlendirme materyallerini sisteme yükler, düzenler ve değerlendirme sonuçlarını girer.

Öğrenci performansını değerlendirmek: Öğretmenler, öğrencilerin akademik performansını takip eder, notları girer ve raporlar oluşturur.

2.2.3 Yöneticiler:

Okulun genel yönetimini yapmak: Yöneticiler, okulun genel işleyişini ve yönetimini sistemi kullanarak takip eder, öğrenci ve öğretmen kayıtlarını yönetir.

Okul kaynaklarını planlamak: Yöneticiler, sınıflar, laboratuvarlar, kütüphane gibi okul kaynaklarının planlamasını yapar ve bu kaynakları sisteme kaydeder.

İstatistiksel verileri analiz etmek: Yöneticiler, sistemin sağladığı istatistiksel verileri kullanarak öğrenci başarılarını, devamsızlık oranlarını ve diğer performans metriklerini analiz eder ve raporlar oluşturur.

2.2.4 Veli veya Veliler:

Çocuklarının akademik ilerlemesini takip etmek: Veliler, çocuklarının sınav notlarını, ders başarılarını, devamsızlık durumlarını ve diğer akademik bilgilerini sisteme girilmiş olarak görüntüler ve takip eder.

Öğrenci devamsızlıklarını izlemek: Veliler, çocuklarının devamsızlık durumunu sisteme kaydedilen veriler üzerinden takip eder ve gerekli durumlarda müdahalede bulunur.

İletişim kurmak: Veliler, öğretmenler ve okul yöneticileri ile iletişim kurmak, randevu talepleri göndermek veya iletişim bilgilerini güncellemek için sistemi kullanır.

2.3 Çalışma Ortamı

2.3.1. Donanım Platformu:

Sunucular: OBS, yüksek performanslı sunucularda çalışır. Sunucuların donanım özellikleri, işlemci hızı, bellek kapasitesi, depolama alanı ve ağ bağlantısı gibi faktörlere bağlı olarak belirlenir. Önerilen minimum sistem gereksinimleri belirlenmeli ve bu gereksinimler doğrultusunda uygun sunucu altyapısı oluşturulmalıdır.

İstemci Cihazları: OBS'ye erişim sağlayacak olan kullanıcıların kullandığı cihazlar da dikkate alınmalıdır. OBS, çeşitli platformlarda çalışabilecek şekilde tasarlanmalıdır. Bu platformlar arasında Windows, macOS, Linux, iOS ve Android gibi işletim sistemleri bulunabilir. Kullanıcıların geniş bir yelpazedeki cihazlardan erişim sağlaması göz önünde bulundurulmalıdır.

2.3.2. İşletim Sistemleri:

Sunucu İşletim Sistemi: OBS, yaygın olarak kullanılan işletim sistemlerinden birinde çalışabilir. Örneğin, Linux tabanlı dağıtımlar (Ubuntu, CentOS), Windows Server gibi işletim sistemleri tercih edilebilir. İşletim sistemi seçimi, performans, güvenlik ve uyumluluk faktörleri dikkate alınarak yapılmalıdır.

İstemci İşletim Sistemi: Kullanıcıların kullandığı cihazlarda farklı işletim sistemleri bulunabilir. OBS'nin, yaygın olarak kullanılan işletim sistemlerinde (Windows, macOS, Linux) sorunsuz çalışabilmesi hedeflenmelidir. Kullanıcıların çeşitli işletim sistemlerinde tutarlı bir deneyim yaşaması sağlanmalıdır.

2.3.3. Coğrafi Konumlar:

OBS, farklı eğitim kurumlarında ve coğrafi konumlarda kullanılabilir. Sistem, birden fazla eğitim kurumunda kullanılmak üzere ölçeklenebilir olmalıdır. Her coğrafi konumun bağlantı hızı ve altyapısı dikkate alınmalı, düşük gecikme süreleri ve yüksek veri aktarım hızları sağlanmalıdır.

2.3.4. Veritabanları ve Sunucular:

OBS, bir veritabanı yönetim sistemi üzerinde çalışır. Veritabanı, öğrenci bilgilerini, ders programlarını, sınav notlarını ve diğer ilgili verileri depolamak için kullanılır. Veritabanı yönetim sistemi olarak yaygın olarak kullanılan seçenekler arasında MySQL, PostgreSQL, Oracle vb. bulunur. Veritabanının performansı, yüksek erişilebilirlik ve veri bütünlüğü sağlayacak şekilde yapılandırılmalıdır.

Sunucular, OBS'nin çalışacağı ortamları barındıran sistemlerdir. Bu sunucular, veritabanı sunucusu, web sunucusu, uygulama sunucusu ve diğer gereksinim duyulan bileşenleri içerebilir. Sunucuların yüksek performanslı, güvenli ve ölçeklenebilir olması önemlidir.

2.3.5. Diğer Yazılım Bileşenleri:

OBS'nin çalışması için gerekli diğer yazılım bileşenleri ve uygulamalar belirlenmelidir. Örneğin, web sunucusu yazılımları (Apache, Nginx), programlama dilleri ve çerçeveler (Java, Python, PHP), veri entegrasyon araçları, eposta sunucuları ve güvenlik yazılımları gibi çeşitli bileşenler kullanılabilir. Bu bileşenlerin doğru yapılandırılması ve uyumlu çalışması sağlanmalıdır.

2.4 Tasarım Ve Uygulama Kısıtlamaları

Programlama Dili: OBS'nin geliştirilmesi için belirli bir programlama dilinin kullanılması gerekebilir. Bu kısıtlama, geliştirme ekibinin belirli bir programlama dili üzerinde uzmanlık sahibi olmasını veya mevcut altyapının bu dili desteklemesini gerektirebilir. Genel olarak tercih edilen diller arasında Python, Java, C#, Javascript, PHP vardır.

Halihazırda Yatırım Yapılmış Kod Kitaplığı: OBS'nin geliştirilmesi sırasında, halihazırda yatırım yapılmış ve kullanılması gereken belirli bir kod kitaplığı veya yazılım bileşeni olabilir. Bu kısıtlama, belirli bir işlevin sağlanması için harici bir kaynağın kullanılmasını gerektirebilir. Bu kaynak, belirli bir standartı desteklemek, performansı artırmak veya belirli bir fonksiyonelliği sağlamak amacıyla tercih edilmiş olabilir.

Donanım Kısıtlamaları: OBS'nin çalışması için belirli donanım gereksinimlerine ihtiyaç duyulabilir. Bu kısıtlama, işletim sistemi, bellek, işlemci hızı, depolama alanı gibi donanım bileşenlerini içerebilir. OBS'nin belirli bir donanım platformunda çalışması veya belirli bir minimum donanım gereksinimini karşılaması gerekebilir.

Entegrasyon Kısıtlamaları: OBS, diğer sistemlerle entegre çalışması gereken bir yazılım olabilir. Bu durumda, belirli bir entegrasyon arayüzünün veya standardının kullanılması gerekebilir. OBS'nin belirli bir veritabanı, sunucu veya diğer yazılım bileşenleriyle uyumlu olması sağlanmalıdır.

Güvenlik ve Veri Gizliliği Kısıtlamaları: OBS, öğrenci bilgileri gibi hassas verileri işleyeceği için güvenlik ve veri gizliliği kısıtlamalarına tabi olabilir. Veri şifreleme, erişim kontrolleri, kullanıcı kimlik doğrulama gibi güvenlik önlemlerinin uygulanması gerekebilir. Ayrıca, ilgili yasal düzenlemelere (ör. Kişisel Verilerin Korunması Kanunu) ve veri gizliliği politikalarına uyum sağlanmalıdır.

Performans Kısıtlamaları: OBS'nin belirli bir performans standardını karşılaması veya belirli bir kullanıcı yükü altında verimli çalışması

gerekebilir. Bu kısıtlama, hızlı yanıt süreleri, büyük veri işleme kapasitesi veya aynı anda çok sayıda kullanıcıyı destekleme gibi performans hedeflerini içerebilir.

Tasarım ve uygulama kısıtlamaları, geliştirme sürecinde dikkate alınması gereken faktörlerdir. Bu kısıtlamalar, belirli teknolojik seçimleri, gereksinimleri veya dış faktörleri etkileyebilir. Ancak, tasarım kısıtlamalarının gereksinimlerden kaynaklanan ve gerçekçi çözüm gerekliliklerine dayandığından emin olunmalıdır. Gereksiz tasarım kısıtlamalarından kaçınılmalı ve geliştirme sürecinin esnekliği ve inovasyonu desteklemesi sağlanmalıdır.

2.5 Varsayımlar Ve Bağımlılıklar

Varsayımlar:

Kullanıcı Arayüzü: OBS'nin öğrenciler, öğretmenler ve yöneticiler gibi farklı kullanıcılar arasında paylaşılan bir kullanıcı arayüzüne sahip olacağı varsayılmaktadır. Kullanıcı arayüzü, kolay kullanılabilirlik, erişilebilirlik ve farklı kullanıcı profillerine uygun özellikler içerecektir.

Yazılım Geliştirme Dili ve Platformu: OBS'nin belirli bir yazılım geliştirme dili ve platformu kullanılarak geliştirileceği varsayılmaktadır. Bu varsayım, geliştirme ekibinin belirli bir dil ve platformda deneyimli olduğunu gösterir ve geliştirme sürecinde bu dil ve platformun kullanılacağını öngörür.

Veri Entegrasyonu: OBS'nin mevcut okul sistemleri ve veritabanlarıyla entegre olacağı varsayılmaktadır. Bu entegrasyonlar, öğrenci bilgileri, ders programları, sınav sonuçları ve devamsızlık gibi verilerin diğer sistemlerden alınmasını veya diğer sistemlere aktarılmasını içerebilir.

Veri Güvenliği: OBS'nin kullanıcı verilerinin güvenliğini sağlamak için gerekli önlemleri alacağı varsayılmaktadır. Bu önlemler, veri şifreleme, kullanıcı kimlik doğrulama, erişim kontrolü ve veri güvenliği politikalarını içerebilir.

Ölçeklenebilirlik: OBS'nin büyüme ve artan kullanıcı taleplerini karşılamak için ölçeklenebilir bir yapıya sahip olacağı varsayılmaktadır. Bu, sistem altyapısının kullanıcı trafiğine ve veri hacmine uyum sağlayabilecek şekilde tasarlanacağı anlamına gelir.

Bağımlılıklar:

Harici Sistemler: OBS'nin belirli harici sistemlere bağımlılığı olabilir. Örneğin, öğrenci kayıt sistemi, mali işlemler sistemi veya kütüphane yönetim sistemi gibi entegre edilecek harici sistemlere bağımlılık olabilir.

Donanım ve Altyapı: OBS'nin belirli donanım ve altyapı bileşenlerine bağımlılığı olabilir. Bu bileşenler arasında sunucular, ağ altyapısı, veritabanı sunucusu ve yedekleme sistemleri gibi unsurlar bulunabilir. Bu bağımlılıklar, sistem performansı ve kullanılabilirliği üzerinde etkili olabilir.

Teknolojik Standartlar: OBS'nin belirli teknolojik standartlara uyumlu olması gerekebilir. Örneğin, veri iletişimi için belirli bir protokol veya veri tabanı için belirli bir standardın takip edilmesi gibi bağımlılıklar olabilir.

Kaynaklar ve Yetenekler: OBS'nin başarılı bir şekilde geliştirilmesi için belirli kaynaklara (insan kaynakları, maddi kaynaklar, teknik uzmanlık) ve yeteneklere bağımlılığı olabilir. Bu kaynaklar ve yetenekler, proje planlaması, yazılım geliştirme, test etme ve dağıtma süreçlerini etkileyebilir.

Bu varsayımlar ve bağımlılıklar, OBS'nin planlama, tasarım ve uygulama süreçlerinde dikkate alınmalıdır. Her bir varsayımın ve bağımlılığın doğruluğu ve geçerliliği proje ilerledikçe doğrulanmalı ve güncellenmelidir.

3.Sistem Özelliği

Açıklama:

OBS'nin yönetici, öğrenci, öğretmen ve veli kullanıcıları için belirli işlevleri içermelidir. Aşağıda, her bir kullanıcı sınıfı için özelliklerin açıklamaları verilmiştir.

3.1. Yönetici İşlevleri

3.1.1 Açıklama:

Bu özellik, yöneticinin OBS üzerindeki yönetimsel görevlerini gerçekleştirebilmesini sağlar. Yönetici kullanıcıları, genel sistem yapılandırması, veri yönetimi, rapor oluşturma ve diğer yönetimsel işlemlere erişebilir.

3.1.2 İşlevsel gereksinimler:

Kullanıcı yönetimi: Yöneticinin yeni kullanıcı hesapları oluşturabilmesi, kullanıcı izinlerini ayarlayabilmesi ve hesapları devre dışı bırakabilmesi gerekmektedir.

Veri yönetimi: Yöneticinin öğrenci, öğretmen ve diğer kayıtları düzenleyebilmesi, veri güncellemelerini yapabilmesi ve veritabanını yönetebilmesi gerekmektedir.

Raporlama: Yöneticinin öğrenci başarı raporları, devamsızlık raporları, finansal raporlar gibi farklı raporları oluşturabilmesi gerekmektedir.

Sistem yapılandırması: Yöneticinin OBS'nin genel yapılandırmasını yapabilmesi, sistem ayarlarını ve parametrelerini yönetebilmesi gerekmektedir.

Yönetici Senaryosu:

Yönetici, kullanıcı adı ve şifresiyle OBS'ye giriş yapar.

Yönetici, öğretmenlerin ve öğrencilerin hesaplarını yönetmek için kullanıcı yönetimi bölümüne erişir.

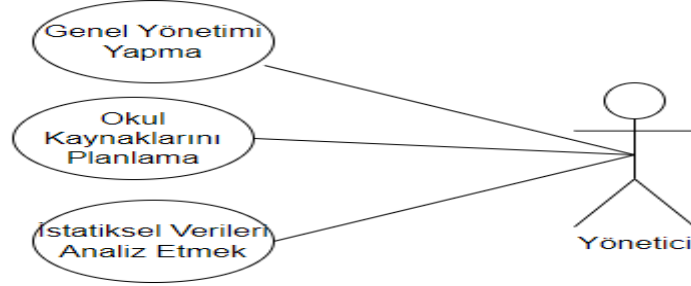
Yeni bir öğretmen hesabı oluşturur ve hesap ayarlarını yapılandırır.

Yönetici, öğrenci kayıtlarını düzenlemek için veri yönetimi bölümüne erişir.

Bir öğrencinin kişisel bilgilerini günceller ve iletişim bilgilerini ekler.

Raporlama bölümüne girerek, öğrenci başarı raporları ve finansal raporlar gibi çeşitli raporlar oluşturur.

Sistem yapılandırması bölümünden, OBS'nin genel yapılandırmasını ve sistem ayarlarını düzenler.



3.2. Öğrenci İşlevleri

3.2.1 Açıklama:

Bu özellik, öğrencilerin OBS'ye erişebilmesini ve kendi öğrenci bilgilerini görüntüleyebilmesini sağlar.

3.2.2 İşlevsel gereksinimler:

Oturum açma: Öğrencilerin OBS'ye giriş yapabilmesi için kullanıcı adı ve şifreyle oturum açmaları gerekmektedir.

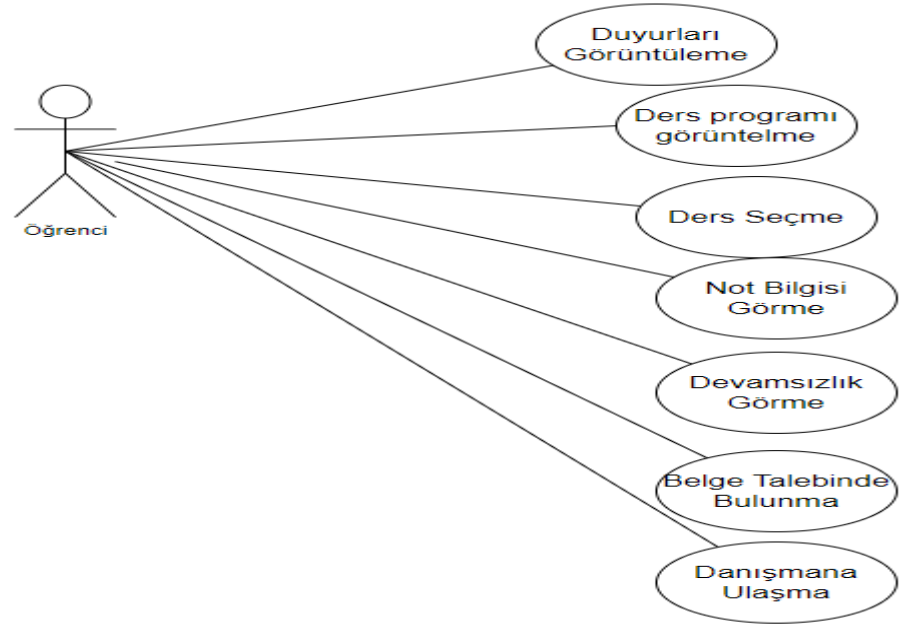
Kişisel bilgiler: Öğrencilerin kişisel bilgilerini güncelleyebilmesi, iletişim bilgilerini görüntüleyebilmesi ve diğer öğrenci bilgilerine erişebilmesi gerekmektedir.

Ders programı: Öğrencilerin ders programını görüntüleyebilmesi ve güncellemeler yapabilmesi gerekmektedir.

Sınav ve notlar: Öğrencilerin sınav tarihlerini ve notlarını görüntüleyebilmesi gerekmektedir.

Öğrenci Senaryosu:

- Öğrenci, kullanıcı adı ve şifresiyle OBS'ye giriş yapar.
- Kişisel bilgilerini kontrol eder ve güncellemeler yapar.
- Ders programını görüntüler ve güncellemeler yapar.
- Sınav tarihlerini kontrol eder ve sınav notlarını görüntüler.
- Ödevleri takip eder ve gönderir.
- Gerekirse öğretmenlerle iletişim kurar ve randevu talepleri gönderir.



3.3. Öğretmen İşlevleri

3.3.1 Açıklama:

Bu özellik, öğretmenlerin OBS üzerinde akademik ve sınıf yönetimi işlevlerini gerçekleştirebilmesini sağlar.

3.3.2 İşlevsel gereksinimler:

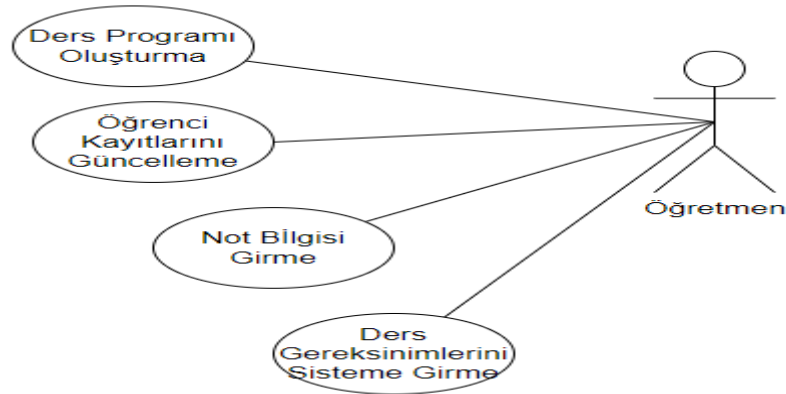
Ders yönetimi: Öğretmenlerin ders programlarını oluşturabilmesi, ders materyallerini yükleyebilmesi ve ödevleri yönetebilmesi gerekmektedir.

Sınıf katılımı: Öğretmenlerin öğrenci katılımını takip edebilmesi, devamsızlık kayıtlarını tutabilmesi gerekmektedir.

Sınav değerlendirmesi: Öğretmenlerin sınavlarını değerlendirebilmesi, not girişi yapabilmesi ve öğrenci performansını takip edebilmesi gerekmektedir.

Öğretmen Senaryosu:

- Öğretmen, kullanıcı adı ve şifresiyle OBS'ye giriş yapar.
- Ders programını görüntüler ve ders materyallerini yükler.
- Öğrenci katılımını takip eder ve devamsızlık kayıtlarını tutar.
- Sınavlarını değerlendirir, not girişi yapar ve öğrenci performansını takip eder.
- Öğrencilerle iletişim kurar ve gerekirse velilerle iletişimde olur.
- Öğrencilerin ödevlerini kontrol eder ve geribildirim sağlar.



3.4. Veli İşlevleri

3.4.1 Açıklama:

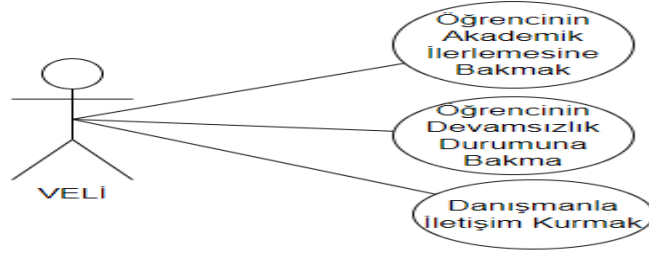
Bu özellik, velilerin OBS üzerinde çocuklarının akademik ilerlemesini takip edebilmesini ve iletişim kurabilmesini sağlar.

3.4.2 İşlevsel gereksinimler:

Öğrenci takibi: Velilerin çocuklarının devamsızlık durumunu, notlarını ve performansını takip edebilmesi gerekmektedir.

İletişim: Velilerin öğretmenlerle iletişim kurabilmesi, randevu talepleri gönderebilmesi gerekmektedir.

Duyurular: Velilerin okul yönetiminin veya öğretmenlerin duyurularını görüntüleyebilmesi gerekmektedir.



Veli Senaryosu:

- Veli, kullanıcı adı ve şifresiyle OBS'ye giriş yapar.
- Çocuğunun devamsızlık durumunu kontrol eder ve gerekirse mazeret bildirir.
- Öğrencinin notlarını görüntüler ve performansını takip eder.
- Öğretmenlerle iletişim kurar, randevu talepleri gönderir veya mesaj gönderir.
- Okul yönetiminin duyurularını görüntüler ve okul etkinliklerine katılır.
- Öğrencinin ders programını kontrol eder ve güncellemeleri takip eder.

Veri gerekliliklerini daha da detaylandıralım:

4. Veri Gereklilikleri

Kullanıcı Verileri:

Öğrenci bilgileri: Öğrenci numarası, doğum yeri, TC kimlik numarası, adres bilgileri, ebeveyn bilgileri.

Öğretmen bilgileri: Öğretmen numarası, unvan, mezun olduğu okul ve bölüm bilgileri, çalışma süresi.

Yönetici bilgileri: Yönetici numarası, pozisyon, yetki seviyesi, çalışma süresi.

Akademik Veriler:

Ders bilgileri: Ders kodu, ders tanımı, dersin verildiği bölüm, dersin kredisi.

Sınıf bilgileri: Sınıf kodu, sınıfın bulunduğu bina ve oda numarası, sınıfın kapasitesi.

Ders programı: Dersin günü, saati ve süresi, dersi veren öğretmen, sınıf bilgisi.

Not ve Performans Verileri:

Sınav notları: Ders kodu, sınav tarihi, öğrencinin aldığı not.

Ödev notları: Ders kodu, ödevin verildiği tarih, öğrencinin aldığı not.

Devamsızlık kayıtları: Ders kodu, öğrencinin devamsızlık tarihleri.

Finansal Veriler:

Öğrenci harç bilgileri: Öğrenci numarası, harç tutarı, ödeme tarihi, ödeme yöntemi.

Burs bilgileri: Öğrenci numarası, burs miktarı, bursun verildiği dönem.

Finansal raporlar: Gelirgider tablosu, mali durum raporu, bütçe tahminleri.

İletişim Verileri:

Mesajlaşma geçmişi: Gönderici ve alıcı kullanıcı numaraları, mesaj içeriği, mesaj tarihi.

Eposta bildirimleri: Alıcı eposta adresi, konu, içerik, gönderim tarihi.

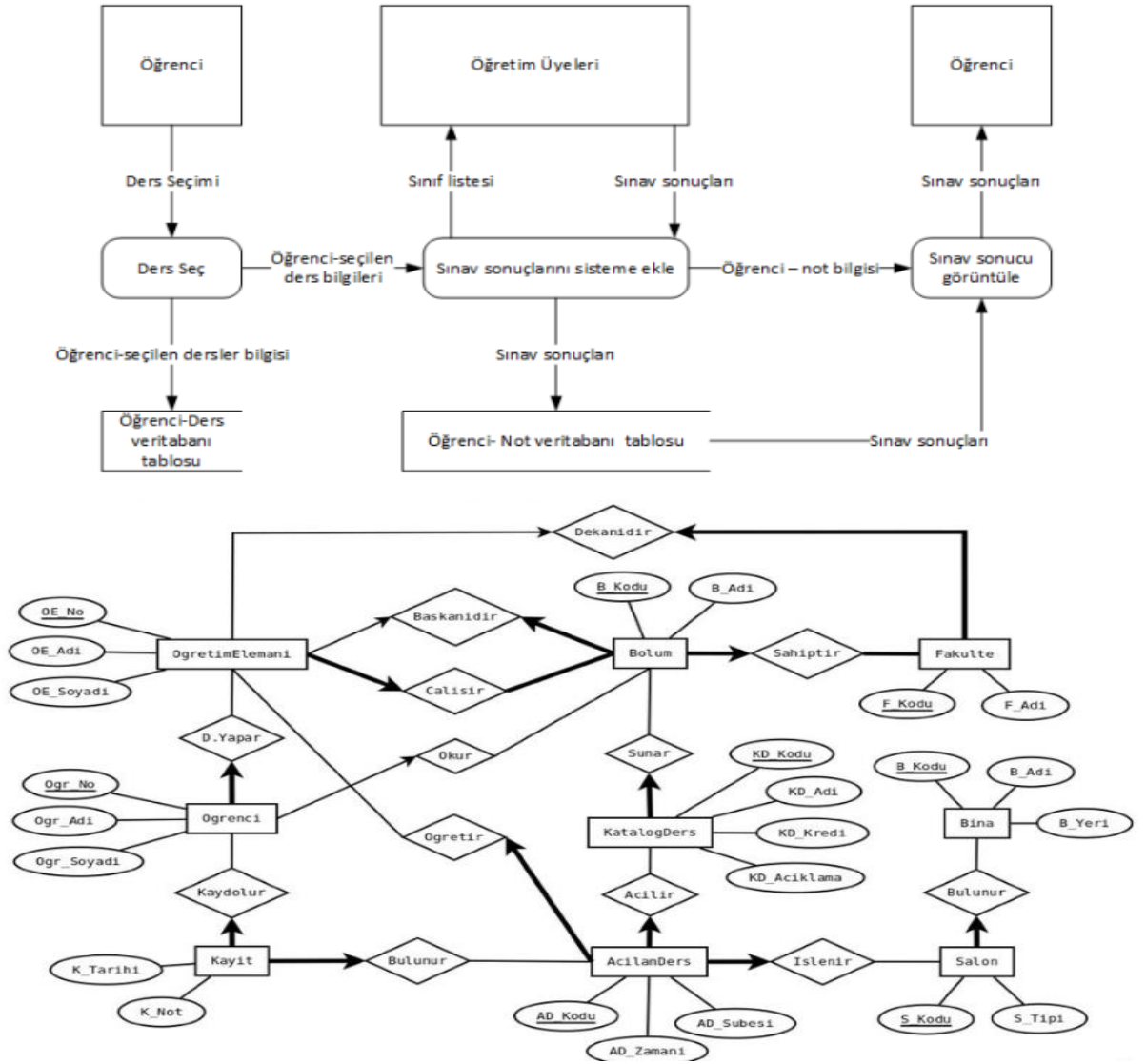
Kaynak Veriler:

Kütüphane kaynakları: Kitap adı, yazar, yayınevi, yayın yılı, ISBN.

Dış kaynaklar: Harita verileri, çevrimiçi veritabanlarından alınan bilgiler.

4.1 Mantıksal Veri Modeli

OBS için bir mantıksal veri modeli oluşturalım. Bu, veri nesneleri ve aralarındaki ilişkilerin görsel bir temsilini sağlayacaktır. Aşağıda, OBS'nin mantıksal veri modeline örnek olarak bazı veri nesneleri ve ilişkileri verilmiştir:



Bu örnek veri modeli, OBS'nin temel veri nesnelerini ve aralarındaki ilişkileri temsil etmektedir. Öğrenci, öğretmen, sınıf, ders, sınav ve devamsızlık gibi veri nesneleri tanımlanmıştır. Bu veri modeli, OBS'nin mantıksal yapısını anlamak için bir temel sağlar.

4.2 Veri Sözlüğü

Kullanıcı:

KullanıcıID: Kullanıcının benzersiz kimlik numarası (sayısal)

Adı: Kullanıcının adı (metin, maksimum 50 karakter)

Soyadı: Kullanıcının soyadı (metin, maksimum 50 karakter)

KullanıcıAdı: Kullanıcının kullanıcı adı (metin, maksimum 20 karakter)

Parola: Kullanıcının parolası (metin, maksimum 20 karakter)

Öğrenci:

ÖğrenciID: Öğrencinin benzersiz kimlik numarası (sayısal)

Adı: Öğrencinin adı (metin, maksimum 50 karakter)

Soyadı: Öğrencinin soyadı (metin, maksimum 50 karakter)

DoğumTarihi: Öğrencinin doğum tarihi (tarih)

Cinsiyet: Öğrencinin cinsiyeti (metin, 'Erkek' veya 'Kadın')

SınıfID: Öğrencinin kayıtlı olduğu sınıfın kimlik numarası (sayısal)

Öğretmen:

ÖğretmenID: Öğretmenin benzersiz kimlik numarası (sayısal)

Adı: Öğretmenin adı (metin, maksimum 50 karakter)

Soyadı: Öğretmenin soyadı (metin, maksimum 50 karakter)

Unvan: Öğretmenin unvanı (metin, maksimum 50 karakter)

BranşID: Öğretmenin uzman olduğu branşın kimlik numarası (sayısal)

Sınıf:

SınıfID: Sınıfın benzersiz kimlik numarası (sayısal)

SınıfAdı: Sınıfın adı (metin, maksimum 50 karakter)

BölümID: Sınıfın bağlı olduğu bölümün kimlik numarası (sayısal)

Bölüm:

BölümID: Bölümün benzersiz kimlik numarası (sayısal)

BölümAdı: Bölümün adı (metin, maksimum 50 karakter)

Branş:

BranşID: Branşın benzersiz kimlik numarası (sayısal)

BranşAdı: Branşın adı (metin, maksimum 50 karakter)

Ders:

DersID: Dersin benzersiz kimlik numarası (sayısal)

DersAdı: Dersin adı (metin, maksimum 50 karakter)

BranşID: Dersin bağlı olduğu branşın kimlik numarası (sayısal)

Sınav:

SınavID: Sınavın benzersiz kimlik numarası (sayısal)

DersID: Sınavın ilişkili olduğu dersin kimlik numarası (sayısal)

SınavTarihi: Sınavın tarihi (tarih)

Puan: Sınav sonucu (sayısal)

Devamsızlık

DevamsızlıkID: Devamsızlığın benzersiz kimlik numarası (sayısal)

ÖğrenciID: Devamsızlığın ilişkili olduğu öğrencinin kimlik numarası (sayısal)

DersID: Devamsızlığın ilişkili olduğu dersin kimlik numarası (sayısal)

Tarih: Devamsızlık tarihi (tarih)

4.3 Raporlar

Öğrenci Not Raporu:

Açıklama: Bu rapor, belirli bir dönem veya sınav için öğrencilerin notlarını gösterir.

Özellikler: Öğrenci adı, öğrenci numarası, sınıf, ders adı, sınav adı, alınan not.

Öğrenci Devamsızlık Raporu:

Açıklama: Bu rapor, belirli bir dönem için öğrencilerin devamsızlık durumunu gösterir.

Özellikler: Öğrenci adı, öğrenci numarası, sınıf, ders adı, toplam devamsızlık günü.

Öğretmen Ders Programı Raporu:

Açıklama: Bu rapor, öğretmenlerin belirli bir dönem için ders programını gösterir.

Özellikler: Öğretmen adı, öğretmen numarası, ders adı, gün, saat.

Sınıf Listesi Raporu:

Açıklama: Bu rapor, belirli bir sınıfın öğrenci listesini gösterir.

Özellikler: Sınıf adı, sınıf numarası, öğrenci adı, öğrenci numarası.

Branş Bazlı Sınav Sonuçları Raporu:

Açıklama: Bu rapor, belirli bir dönem için branşlara göre sınav sonuçlarını gösterir.

Özellikler: Branş adı, ders adı, sınav adı, öğrenci adı, öğrenci numarası, alınan not.

Öğrenci Başarı Raporu:

Açıklama: Bu rapor, belirli bir dönemde öğrencilerin akademik başarılarını gösterir.

Özellikler: Öğrenci adı, öğrenci numarası, sınıf, ders adı, alınan not, başarı durumu.

Öğretmen Performans Raporu:

Açıklama: Bu rapor, belirli bir dönemde öğretmenlerin performansını gösterir.

Özellikler: Öğretmen adı, öğretmen numarası, ders adı, sınıf, sınav ortalaması.

Velilere Gönderilen Duyuru Raporu:

Açıklama: Bu rapor, belirli bir dönemde velilere gönderilen duyuruları gösterir.

Özellikler: Duyuru başlığı, duyuru metni, alıcı veli adı, alıcı veli numarası.

4.4. Veri Toplama, Bütünlük, Saklama ve İmha Etme

Veri Toplama:

- Veri toplama sürecinde, doğru ve eksiksiz verilerin öğrenci, öğretmen, yönetici ve velilerden toplanması önemlidir.
- Veri toplama yöntemleri, web tabanlı formlar, otomatik entegrasyonlar ve kullanıcı girişi gibi çeşitli teknikleri içerebilir.
- Veri toplama sırasında, veri doğrulama kontrolleri kullanılmalı ve hatalı veya tutarsız verilerin tespit edilmesi için gerekli mekanizmalar sağlanmalıdır.

Veri Bütünlüğü:

- Veri bütünlüğü, verilerin doğruluk, eksiksizlik, tutarlılık ve güncellik açısından uygunluğunu sağlar.
- Veri güncelleme veya değişiklik işlemleri, yetkilendirme ve izin kontrolleri ile sınırlanmalıdır.

- Veri bütünlüğü hataları tespit edildiğinde, düzeltme mekanizmaları ve geri alma işlemleri kullanılmalıdır.

Veri Saklama:

- Verilerin güvenli bir şekilde saklanması, veri gizliliği ve güvenliğinin sağlanması açısından önemlidir.
- Verilerin saklanması için güvenli bir veritabanı veya depolama ortamı kullanılmalıdır.
- Verilerin yedeklenmesi, düzenli olarak gerçekleştirilmeli ve yedekleme politikalarına uygun olarak depolanmalıdır.

Veri İmha:

- Gereksiz hale gelen verilerin güvenli bir şekilde imha edilmesi önemlidir.
- Veri imha süreci, verilerin tamamen silinmesini ve kurtarılamaz hale getirilmesini gerektirir.
- Veri imha süreci, güvenli imha yöntemlerinin kullanılmasını ve veri güvenliğini sağlamak için uygun önlemlerin alınmasını gerektirir.

Bu süreçlerin etkili bir şekilde yönetilmesi, verilerin güvenliğini, bütünlüğünü ve uyumluluğunu sağlar. Ayrıca, veri koruma yasalarına ve ilgili düzenlemelere uyum sağlamak önemlidir ve bu süreçlere ilişkin politikaların ve prosedürlerin düzenli olarak gözden geçirilmesi gerekmektedir.

5. Harici Arabirim Gereksinimleri

Donanım Arabirimleri:

- Sistem, harici donanımlarla etkileşimde bulunabilir. Bu donanımlar arasında yazıcılar, tarayıcılar, kart okuyucular, biyometrik cihazlar vb. bulunabilir.

- Donanım arabirimleri için uyumlu protokoller, bağlantı noktaları, veri formatları ve iletişim hızları belirtilmelidir.
- Donanım aygıtlarının kullanılabilirlik durumu, hata kodları ve sorun giderme yönergeleri gibi bilgilerin belgelenmesi gerekmektedir.

Yazılım Arabirimleri:

- Sistem, diğer yazılımlarla etkileşimde bulunabilir. Bu yazılımlar öğrenci yönetim sistemleri, maliye sistemleri, mesajlaşma platformları, ödeme sistemleri vb. olabilir.
- Yazılım arabirimleri için uyumlu veri formatları, API'lar, veri alışverişi protokolleri ve entegrasyon gereksinimleri belirtilmelidir.
- Veri alışverişi için standartlar ve protokoller kullanılmalı ve uyumluluk gereksinimleri belgelenmelidir.

Kullanıcı Arabirimleri:

- Sistem, kullanıcılarla etkileşimde bulunur ve kullanıcı arabirimleri aracılığıyla bilgileri gösterir ve alır.
- Kullanıcı arabirimleri, kullanıcı dostu ve kullanıcıların sistemle etkileşimi için kolay anlaşılır olmalıdır.
- Kullanıcı arabirimindeki öğeler, menüler, formlar, düğmeler ve alanlar gibi bileşenlerin tasarımı, işlevselliği ve erişilebilirliği belgelenmelidir.

Ağ Arabirimleri:

- Sistem, ağ üzerinden veri alışverişi yapabilir ve ağ arayüzleriyle uyumlu olmalıdır.
- Ağ arabirimleri için kullanılacak protokoller (HTTP, TCP/IP, FTP vb.), bağlantı noktaları ve veri alışverişi yöntemleri belirtilmelidir.
- Güvenlik önlemleri, şifreleme gereksinimleri ve kimlik doğrulama yöntemleri gibi ağ güvenliği ile ilgili gereksinimler belgelenmelidir.

Bu gereksinimler, sistem içindeki farklı bileşenlerin ve harici kaynakların birbirleriyle etkileşimini sağlamak için belirli arabirim gereksinimlerini tanımlar. Bu arabirim belgeleri, sistem mimarisini, protokolleri ve uyumlu bileşenlerin kullanımını açıklayarak diğer belgelere referans oluşturur. Bu sayede, sistemin entegrasyon ve uyumluluk gereksinimlerinin karşılanması sağlanır ve doğru bir şekilde dış kaynaklara bağlanabilir.

OBS için hangi yazılım dilinin ve araçlarının, hangi teknolojilerin neden seçildiğini ve nelere bağlı olduğunu bilmek önemlidir. Bunlar:

Web Tabanlı Uygulama: OBS, genellikle web tabanlı bir uygulama olarak kullanılır. Bu nedenle, web geliştirme için yaygın olarak kullanılan programlama dilleri ve teknolojiler tercih edilir. Örneğin, Java, C#, PHP veya Python gibi diller ve bunlara uygun web çerçeveleri (frameworks) kullanılabilir.

Veritabanı Entegrasyonu: OBS, öğrenci verilerinin yönetimi için bir veritabanı kullanır. Bu nedenle, veritabanı işlemleri ve sorguları için etkili bir şekilde kullanılabilen diller ve araçlar tercih edilir. Örneğin, SQL tabanlı veritabanı yönetim sistemleriyle uyumlu diller (Java, C#, vb.) ve ORM (ObjectRelational Mapping) araçları kullanılabilir.

Kullanılabilirlik ve UI Tasarımı: OBS'nin kullanıcı arayüzü, kullanıcı dostu ve kolay anlaşılabilir olmalıdır. Bu nedenle, kullanıcı arayüzü tasarımı için uygun araçlar ve teknolojiler tercih edilir. Örneğin, HTML, CSS ve JavaScript gibi web teknolojileriyle birlikte UI/UX tasarım çerçeveleri kullanılabilir.

Güvenlik ve Veri Koruma: OBS, hassas öğrenci verilerini içerdiği için güvenlik önemlidir. Güvenlik açıklarının önlenmesi ve veri koruması için uygun araçlar ve teknikler kullanılmalıdır. Örneğin, şifreleme algoritmaları, oturum yönetimi ve erişim kontrolü gibi güvenlik konularına odaklanan kütüphaneler ve araçlar tercih edilebilir.

Ölçeklenebilirlik ve Performans: OBS, bir eğitim kurumunda genellikle büyük miktarda veriyi yönetmek zorundadır. Bu nedenle, sistem ölçeklenebilirliği ve performansı önemlidir. Veritabanı optimizasyonu, önbellekleme mekanizmaları, paralel işleme ve yük dengeleme gibi teknolojilerin kullanılması, performansı artırabilir.

Gerçekleştirme aşamasında meydana gelebilecek olağan dışı durumlar, beklenmeyen hatalar, aksaklıklar veya sorunlarla karşılaşılması durumlarını içerebilir. İşte bazı olası olağan dışı durumlar:

Hata Durumları:

Programlamada hataların oluşması: Yanlış syntax kullanımı, hatalı değişken tanımlamaları, hatalı hesaplama işlemleri vb.

Harici faktörlerden kaynaklı hatalar: Veri kaynaklarının erişilemez olması, ağ bağlantı sorunları, veri bütünlüğü kaybı vb.

Performans Sorunları:

Yavaş çalışma: Uygulamanın beklenenden daha yavaş çalışması, yüksek bellek veya işlemci kullanımı, optimize edilmemiş algoritma vb.

Verimlilik sorunları: Büyük veri kümelerinin işlenmesi, yoğun veri trafiği, aynı anda çok sayıda kullanıcının erişimi vb.

Uyumluluk Sorunları:

Farklı platformlar arasında uyumsuzluk: Uygulamanın farklı işletim sistemleri, tarayıcılar veya donanımlar üzerinde sorunsuz çalışmaması.

Veritabanı uyumluluğu: Uygulamanın kullanılan veritabanı yönetim sistemiyle uyumlu olmaması veya veritabanı sürümüyle uyumsuzluk yaşanması.

Güvenlik Sorunları:

Zafiyetler ve saldırılar: İstismar edilebilecek güvenlik açıkları, kötü niyetli kullanıcıların girişiyle veri sızıntısı veya sistem ihlali riski.

Kaynak Sorunları:

Sınırlı kaynaklar: Bellek, işlemci gücü, disk alanı veya ağ bant genişliği gibi kaynakların yetersiz olması veya etkili kullanılamaması.

Bu olağan dışı durumlar, gerçekleştirme aşamasında beklenmeyen sorunlarla karşılaşabileceğimizi göstermektedir. Bu durumlarla başa çıkmak için, hatayı tanımlama ve ayıklama yetenekleri, hata yönetimi stratejileri, hata kodlaması, kaynak izleme ve performans optimizasyonu gibi yöntemler kullanılabilir. Ayrıca, sürekli test ve deneme yaparak bu sorunların mümkün olduğunca erken tespit edilmesi önemlidir.

Kod gözden geçirme işlemi, geliştirilen yazılım kodunun kalitesini artırmak ve hataları tespit etmek amacıyla yapılan bir inceleme sürecidir. İşte genel olarak kullanılan adımlar:

Gözden Geçirme Eki Oluşturma:

- Gözden geçirme sürecini yönetecek bir ekip oluşturun.
- Ekip üyeleri, kodun alanında uzman ve deneyimli kişilerden oluşmalıdır.

Gözden Geçirme Kapsamının Belirlenmesi:

- Gözden geçirme işlemi için hangi kod parçalarının inceleneceğini belirleyin.
- Kodun belli bir bölümü, belirli bir modül veya tamamı gözden geçirilebilir.

Gözden Geçirme İlkelerinin Belirlenmesi:

-Gözden geçirme sürecinde takip edilecek ilkeleri belirleyin. Bu ilkeler, kodlama standartları, hata işleme yöntemleri, verimlilik optimizasyonları gibi konuları içerebilir.

Gözden Geçirme Oturumunun Planlanması:

- Gözden geçirme oturumunun zamanını ve yerini planlayın.
- Tüm ekip üyelerinin katılım sağlayabileceği bir zaman dilimi seçmeye özen gösterin.

Kod Gözden Geçirme Oturumu:

- Gözden geçirme oturumunda kod parçaları ekip tarafından dikkatlice incelenir.
- Kodun okunabilirliği, doğruluğu, verimliliği, güvenliği ve yapısal uyumu gibi konular değerlendirilir.
- Potansiyel hatalar, iyileştirmeler ve en iyi uygulamalar üzerinde tartışmalar yapılır.

Geribildirim ve İyileştirme:

- Gözden geçirme sonucunda elde edilen geribildirimler kaydedilir ve geliştirme sürecine dahil edilir.
- Bulunan hatalar düzeltilir, önerilen iyileştirmeler yapılır ve kodun kalitesi artırılır.

Kod gözden geçirme işlemi, disiplinli ve objektif bir yaklaşım gerektirir. Ayrıca, açık iletişim ve işbirliği de önemlidir. Gözden geçirme süreci, yazılımın kalitesini artırırken, takım üyeleri arasında bilgi paylaşımı ve tecrübe aktarımını da sağlar.

5.1 Kullanıcı Arayüzleri

Kullanıcı arayüzleri, kullanıcıların sistemi etkileşimde bulunmak ve işlemlerini gerçekleştirmek için kullandıkları bileşenlerdir. Kullanıcı arayüzlerinin tasarımı, kullanıcı deneyimini iyileştirmek ve kullanıcıların sistemi verimli bir şekilde kullanmalarını sağlamak için büyük önem taşır. İşte kullanıcı arayüzleriyle ilgili daha detaylı bilgiler:

Kullanıcı Arayüzü Standartları: Kullanıcı arayüzü tasarımında, belirli bir standarta veya ürün hattı stili kılavuzlarına uymak önemlidir. Bu standartlar, yazı tipleri, simgeler, düğme etiketleri, resimler, renk şemaları, alan sekme dizileri, yaygın olarak kullanılan kontroller, marka grafikleri, telif hakkı ve gizlilik bildirimleri gibi unsurları içerebilir. Standartlara uyum, tutarlı bir görünüm ve kullanıcı deneyimi sağlamak için önemlidir.

Ekran Sınırlamaları: Kullanıcı arayüzünün hedeflendiği cihazların ekran boyutu, düzen veya çözünürlük gibi sınırlamaları belirlenmelidir. Farklı cihazlarda ve ekran boyutlarında kullanılabilirlik sağlamak için, kullanıcı arayüzü bu sınırlamalara uygun şekilde tasarlanmalıdır. Örneğin, mobil cihazlarda daha küçük ekranlara uygun bir kullanıcı arayüzü tasarımı düşünülebilir.

Standart Düğmeler ve Gezinme Bağlantıları: Kullanıcı arayüzünde her ekranda görünecek standart düğmeler, işlevler veya gezinme bağlantıları tanımlanmalıdır. Bu, kullanıcıların kolaylıkla erişebileceği ve kullanabileceği önemli işlevleri vurgulamak için kullanılır. Örneğin, geri dönüş düğmesi veya ana sayfa bağlantısı gibi standart gezinme kontrolleri kullanılabilir.

Kısayol Tuşları: Kullanıcıların sistemi daha hızlı kullanabilmeleri için kısayol tuşları tanımlanabilir. Özellikle klavye kullanımını tercih eden kullanıcılar için kısayol tuşları, işlemleri hızlandırır ve kullanıcı deneyimini iyileştirir. Örneğin, belirli bir işlemi gerçekleştirmek için "Ctrl+S" tuş kombinasyonu gibi kısayol tuşları kullanılabilir.

Veri Doğrulama Yönergeleri: Kullanıcı girişlerinin doğrulanması ve giriş değerlerinin sınırlamaları gibi veri doğrulama yönergeleri belirlenmelidir. Bu, kullanıcıların geçerli ve uygun verileri girebilmelerini sağlar ve hatalı girişlerin önlenmesine yardımcı olur. Örneğin, bir sayı girişi yapılması gereken bir alanda sadece rakam kabul edilmesi veya bir eposta adresi girişi yapılması gereken bir alanda geçerli bir eposta formatının kontrol edilmesi gibi doğrulama yönergeleri uygulanabilir.

Kullanıcı Konaklaması: Kullanıcı arayüzünün farklı kullanıcı ihtiyaçlarına uyum sağlaması önemlidir. Görme engelli, renk körü veya başka sınırlamaları olan kullanıcılar için konaklama önlemleri alınmalıdır. Bu, metin boyutu ayarları, yüksek kontrast seçenekleri, ekran okuyucu desteği gibi özellikleri içerebilir. Kullanıcıların arayüzü kolaylıkla kullanabilmeleri ve içerikleri anlayabilmeleri için erişilebilirlik önemlidir.

5.2 Yazılım Arayüzleri

Yazılım arayüzleri, bu ürünle diğer yazılım bileşenleri arasındaki bağlantıları tanımlar. Bu arayüzler, diğer uygulamalar, veritabanları, işletim sistemleri, araçlar, kitaplıklar, web siteleri ve entegre ticari bileşenler gibi yazılım bileşenleriyle etkileşimi sağlar. İşte yazılım arayüzleriyle ilgili daha detaylı bilgiler:

Arayüz Tanımları: Diğer yazılım bileşenleriyle yapılan bağlantılar, bileşenlerin adı ve sürümüyle tanımlanır. Örneğin, belirli bir veritabanı yönetim sistemiyle veya bir web servis sağlayıcısıyla iletişim sağlayan arayüzlerin adları ve sürümleri belirtilir.

Mesajlaşma ve Veri Alışverişi: Yazılım bileşenleri arasında değiş tokuş edilen mesajlar, veriler ve kontrol değerleri açıklanır. Bu, mesaj biçimlerini, amaçlarını ve içeriklerini kapsar. Örneğin, bir API aracılığıyla gönderilen JSON formatındaki bir isteğin yapısı ve içeriği gibi detaylar belirtilir.

Veri Eşlemeleri ve Çevirileri: Sistemler arasında giriş ve çıkış verilerinin eşlemeleri ve bir sistemden diğerine veri aktarımı için gereken çeviriler tanımlanır. Veri formatları, veri

dönüşümleri ve veri entegrasyonu gibi konulara odaklanılır. Örneğin, bir XML dosyasından alınan verilerin bir veritabanına kaydedilmesi için nasıl dönüşüm yapılması gerektiği gibi bilgiler belirtilir.

Hizmetler ve İletişim: Harici yazılım bileşenlerinin ihtiyaç duyduğu hizmetler ve bileşenler arası iletişim doğası tanımlanır. Bu, belirli bir bileşenin diğer bileşenlere sağladığı hizmetlerin ve işlevlerin açıklanması anlamına gelir. Örneğin, bir veritabanı bağlantısı sağlama veya bir web servisine istek gönderme gibi hizmetlerin belirtilmesi söz konusu olabilir.

Veri Paylaşımı: Yazılım bileşenleri arasında değiş tokuş edilecek veya paylaşılacak veriler tanımlanır. Bu, belirli veri alanlarının ve yapılarının belirlenmesi anlamına gelir. Örneğin, bir kullanıcının adı, eposta adresi ve profil resmi gibi verilerin paylaşılması gerekiyorsa, bu verilerin tanımlanması söz konusu olabilir.

İşlevsel Olmayan Gereksinimler: Yanıt süreleri, sıklıkları, hizmet seviyeleri, güvenlik kontrolleri ve kısıtlamalar gibi işlevsel olmayan gereksinimler belirtilir. Bu gereksinimler, arayüzün performansını, güvenliğini ve diğer önemli faktörleri etkiler. Örneğin, bir API'nin maksimum yanıt süresi veya günlük istek sayısı gibi sınırlamaların belirtilmesi söz konusu olabilir.

5.3 Donanım Arayüzleri

Veri Toplama Cihazları:

Öğrenci Bilgisayarları: OBS, öğrencilerin bilgisayarlarından verileri alabilir ve bu cihazlarla etkileşim kurabilir. Bu, web tabanlı bir arayüz veya özel bir istemci uygulaması kullanılarak gerçekleştirilebilir.

Akıllı Kart Okuyucuları: Öğrencilerin kimlik kartlarından veya öğrenci kartlarından verileri okuyan akıllı kart okuyucuları kullanılabilir. Bu, öğrenci kimlik doğrulaması veya katılım takibi gibi işlevler için kullanılabilir.

Veri Depolama Cihazları:

Sunucu Sabit Diskleri: OBS'nin veri depolaması için sunucu sabit diskleri kullanılabilir. Bu diskler, öğrenci bilgilerini güvenli bir şekilde depolamak ve erişmek için kullanılır.

Veritabanı Sunucuları: OBS'nin verilerini ilişkisel bir veritabanında depolayabilir. Bu, verilerin yapılandırılmış bir şekilde tutulmasını ve hızlı erişimi sağlar.

Giriş/Çıkış Aygıtları:

Klavye ve Fare: Kullanıcıların OBS'ye veri girişi yapması veya etkileşimde bulunması için standart klavye ve fare kullanılabilir.

Yazıcılar ve Tarayıcılar: OBS'den raporların yazdırılması veya belgelerin taranması gerekiyorsa, uyumlu yazıcılar ve tarayıcılar kullanılabilir.

Ağ Bağlantısı:

Ethernet Kartları: OBS'nin ağ üzerinden diğer sistemlerle iletişim kurması için uygun Ethernet kartları kullanılabilir.

Ağ Yönlendiricileri ve Switch'ler: OBS'nin ağ bağlantısını sağlamak için yönlendiriciler ve switch'ler kullanılabilir.

Veri Yedekleme ve Kurtarma Cihazları:

Harici Sabit Diskler: OBS'nin verilerinin yedeklenmesi ve kurtarılması için harici sabit diskler kullanılabilir. Bu diskler, veri kaybı durumunda yedek kopyaların korunmasını sağlar.

Bulut Depolama Hizmetleri: OBS'nin verileri bulut depolama hizmetlerine yedekleyebilir, böylece verilerin güvenliği ve erişilebilirliği sağlanır.

5.4. İletişim arayüzleri

Eposta İletişimi:

Sistem, kullanıcıların eposta gönderip alabilmesi için uygun eposta protokolleri ve standartlarını desteklemelidir. Örneğin, POP3, IMAP veya SMTP gibi protokoller kullanılabilir.

Eposta iletişimi için gerekli olan sunucu ayarlarını belirtmelisiniz, örneğin sunucu adresi, bağlantı portu, kimlik doğrulama yöntemleri vb.

Web Tarayıcısı:

Sistem, kullanıcıların web tarayıcıları aracılığıyla uygulamaya erişebilmesini sağlamalıdır. Bu, desteklenen tarayıcı türlerini ve sürümlerini belirtmeyi içerir.

Kullanıcı arabirimi, farklı tarayıcılarda tutarlı bir şekilde görüntülenecek şekilde tasarlanmalıdır.

Ağ Protokolleri:

Sistem, ağ üzerinden diğer sistemlerle iletişim kurarken kullanılan protokolleri belirtmelidir. Örneğin, HTTP, HTTPS, FTP, TCP/IP gibi protokoller kullanılabilir.

İletişim güvenliği için HTTPS protokolünün kullanılması önerilir. Bu, veri aktarımının şifrelenmesini sağlar.

Elektronik Formlar:

Sistem, kullanıcıların elektronik formlar doldurabilmesini ve gönderebilmesini sağlamalıdır. Bu, kullanıcı arabirimindeki form alanlarını ve veri doğrulama gereksinimlerini belirtmeyi içerir.

İletişim Güvenliği:

Sistem, iletişim sırasında veri güvenliğini sağlamak için uygun güvenlik önlemlerini almalıdır. Bu, veri şifreleme, kimlik doğrulama, yetkilendirme ve güvenli bağlantı protokollerini içerebilir.

Veri Aktarım Oranları:

Sistem, iletişim arayüzleri üzerinden veri aktarım hızını belirtmelidir. Bu, sunucu ve istemci arasındaki veri transferindeki maksimum hızı ve performans gereksinimlerini içerir.

El Sıkışma ve Senkronizasyon Mekanizmaları:

İletişim arayüzleri arasındaki el sıkışma ve senkronizasyon süreçlerini belirtmelisiniz. Örneğin, istemci ve sunucu arasındaki bağlantı kurma ve sonlandırma işlemleri, senkronizasyon protokolleri vb.

Eposta Ekleri:

İletişim arayüzlerinin kabul edebileceği eposta ekleriyle ilgili kısıtlamaları belirtmelisiniz. Örneğin, kabul edilebilir dosya türleri, maksimum dosya boyutu, virüs taraması gereksinimleri vb.

6. Kalite Özellikleri

Kalite özellikleri, bir sistemin işlevsel gereksinimlerin ötesindeki niteliklerini tanımlar. İşlevsiz gereksinimler, sistemin performansını, güvenilirliğini, kullanılabilirliğini, güvenliğini, taşınabilirliğini ve diğer benzer özelliklerini kapsar. Kalite Özellikleri şunlardır:

Performans: Sistemin hızı, yanıt süresi, verimlilik ve kaynak kullanımı gibi performans metrikleri.

Güvenilirlik: Sistemin istikrarı, hata toleransı, kesinti süresi ve veri bütünlüğü gibi güvenilirlik özellikleri.

Kullanılabilirlik: Kullanıcı dostu arayüz, kolay kullanılabilirlik, öğrenme eğrisi, erişilebilirlik ve kullanıcı deneyimi gibi kullanılabilirlik faktörleri.

Güvenlik: Veri gizliliği, yetkilendirme, kimlik doğrulama, saldırı önleme ve güvenlik açıklarının azaltılması gibi güvenlik önlemleri.

Taşınabilirlik: Farklı platformlar arasında uyumluluk, bağımlılık azaltma, taşınabilirlik ve sistem entegrasyonu yetenekleri.

Bakım ve Yönetilebilirlik: Sistemin bakım kolaylığı, değişiklik yönetimi, hata ayıklama, güncelleme ve ölçeklenebilirlik gibi yönetim özellikleri.

Uyumluluk: Uluslararası standartlara uygunluk, sektörel uyum, mevzuat ve regülasyonlara uyum gibi uyumluluk gereksinimleri.

Doğruluk: Sistem çıktılarının doğruluğu, veri doğrulama mekanizmaları ve hatalı işlem riskini azaltma.

Esneklik: Sistemin değişen gereksinimlere ve iş kurallarına uyum sağlama yeteneği.

Bu kalite özellikleri, sistemin sağlaması gereken performans, güvenilirlik, kullanılabilirlik, güvenlik, taşınabilirlik ve diğer önemli nitelikleri belirtir. Her bir kalite özelliği, proje gereksinimlerine ve kullanıcı beklentilerine uygun olarak önceliklendirilmeli ve uygulanmalıdır.

Test (doğrulama) planı, bir sistemin gereksinimlere uygun olarak çalıştığını ve beklentileri karşıladığını doğrulamak için kullanılan bir belgedir. Test planı, hangi test yöntemlerinin kullanılacağını ve neden tercih edildiğini detaylı olarak açıklamalıdır. İşte bazı yaygın test yöntemleri ve neden kullanıldıklarıyla ilgili açıklamalar:

Birim Testleri: Yazılım bileşenlerinin (fonksiyonlar, modüller) doğruluğunu kontrol etmek için kullanılır. Birim testleri, kodun parçalarının ayrı ayrı test edilmesini ve hata ayıklama sürecinin kolaylaştırılmasını sağlar.

Bütünleşik Testler: Farklı bileşenlerin bir araya getirilerek sistemin parçalarının birlikte çalışmasının doğrulandığı testlerdir. Bütünleşik testler, bileşenler arasındaki etkileşimleri ve veri akışını kontrol etmeyi sağlar.

Sistem Testleri: Tamamlanmış sistemin gereksinimlere uygun olarak çalıştığını doğrulamak için kullanılır. Sistem testleri, sistemin tüm işlevlerini, performansını, güvenilirliğini ve kullanılabilirliğini değerlendirir.

Kabul Testleri: Kullanıcıların sistemi kullanmaya hazır olduğunu onaylamak için yapılan testlerdir. Kabul testleri, kullanıcıların beklentilerini karşıladığını doğrulamak amacıyla gerçek dünya senaryolarını simüle eder.

Performans Testleri: Sistemin performansını değerlendirmek için yapılan testlerdir. Performans testleri, yük testleri, stres testleri, dayanıklılık testleri gibi farklı alt kategorilere ayrılabilir. Bu testler, sistemin belirli bir yük altında nasıl performans gösterdiğini ve sınırlarının nerede olduğunu belirlemek için kullanılır.

Güvenlik Testleri: Sistemin güvenlik açıklarını tespit etmek ve önlemek için yapılan testlerdir. Güvenlik testleri, yetkilendirme, kimlik doğrulama, veri şifreleme, saldırı önleme gibi güvenlik önlemlerinin etkinliğini değerlendirir.

Kullanılabilirlik Testleri: Sistemin kullanıcı dostu olduğunu ve kolay kullanılabildiğini doğrulamak için yapılan testlerdir. Kullanılabilirlik testleri, kullanıcı deneyimini, erişilebilirliği ve kullanıcı arayüzünün etkinliğini değerlendirir.

TEST ADIMLARI	BAŞLANGIÇ	BİTİŞ
Furkan Göksu		
Birim Testler	9.07.2023	9.07.2023
Bütünleşik Testler	9.07.2023	10.07.2023
Yusuf Şimşek		
Sistem Testleri	10.07.2023	11.07.2023
Kabul Testleri	12.07.2023	12.07.2023
Emir Yağcı		
Performans Testleri	12.07.2023	13.07.2023
Güvenlik Testleri	13.07.2023	13.07.2023
Eray Akgül		
Güvenlik Testleri	13.07.2023	14.07.2023
Kullanılabilirlik	14.07.2023	14.07.2023

6.1 Kullanılabilirlik

OBS (Öğrenci Bilgi Sistemi & Eğitim Kurumu Bilgi Sistemi) için kullanılabilirlik gereksinimleri, kullanıcıların sistemi kolayca öğrenmelerini, kullanmalarını, hataları önlemelerini ve hatalardan kurtulmalarını, etkileşimleri verimli bir şekilde gerçekleştirmelerini ve sistem erişilebilirliğini sağlamalarını hedefler. Aşağıda, OBS için kullanılabilirlik gereksinimlerinin bazı örneklerini bulabilirsiniz:

Öğrenme Kolaylığı: Kullanıcılar, OBS'yi hızlı bir şekilde öğrenebilmeli ve sistemi etkin bir şekilde kullanabilmelidir. Kullanıcı arayüzü, sezgisel bir tasarıma sahip olmalı, kullanıcıların işlemleri kolayca gerçekleştirmelerini sağlamalı ve gereksiz karmaşıklığı önlemelidir.

Kullanım Kolaylığı: Kullanıcılar, OBS'yi kolaylıkla kullanabilmeli ve istedikleri işlemleri hızlı bir şekilde gerçekleştirebilmelidir. Menüler, düğmeler, formlar vb. kullanıcılar tarafından rahatlıkla anlaşılabilir ve erişilebilir olmalıdır.

Hata Önleme ve Kurtarma: OBS, kullanıcıların hatalar yapmalarını önlemeli ve hata durumlarında kullanıcıları doğru şekilde yönlendirmelidir. Kullanıcılar, hataları fark ettiklerinde kolayca geri dönebilmeli, düzeltmedikleri hatalar için açıklayıcı hata mesajları alabilmeli ve hatalarını düzeltmeleri için yardım alabilecekleri kaynaklara erişebilmelidir.

Etkileşim Verimliliği: Kullanıcıların işlemleri hızlı bir şekilde gerçekleştirebilmeleri için OBS, etkili etkileşimler sağlamalıdır. Ara yüz, kullanıcıların sıkça kullandıkları işlevlere kolay erişim sağlamalı, işlemlerin adımları minimize edilmeli ve kullanıcıların verimli bir şekilde gezinmelerini sağlayacak doğru düzenlemeler yapılmalıdır.

Erişilebilirlik: OBS, farklı kullanıcı gruplarının (engelli kullanıcılar, görme veya işitme engeli olan kullanıcılar, yaşlı kullanıcılar vb.) sistem ile etkileşimde bulunabilmesini sağlamalıdır. Erişilebilirlik gereksinimleri, ekran okuyucu desteği, kontrastlı ve büyük yazı tipleri, klavye kısayolları gibi özellikler içerebilir.

6.2 Performans

OBS (Öğrenci Bilgi Sistemi & Eğitim Kurumu Bilgi Sistemi) için performans gereksinimleri, çeşitli sistem işlemleri için belirlenir. Farklı işlevsel gereksinimler veya özelliklerin farklı performans gereksinimleri olabilir. Bu gereksinimler, ilgili fonksiyonel gereksinimlerle ilişkilendirilerek belirlenebilir.

Veritabanı İşlemleri: OBS, öğrenci bilgilerini, ders programlarını, sınav sonuçlarını vb. içeren bir veritabanı kullanır. Veritabanı işlemleri (veri ekleme, güncelleme, sorgulama) için hedeflenen performans, belirli bir sürede ne kadar veri işlenebileceğini veya sorguların ne kadar hızlı yanıtlanacağını belirtebilir.

Kullanıcı İşlemleri: Kullanıcılar OBS'yi kullanarak ders kaydı yapabilir, notları görüntüleyebilir, sınav sonuçlarını kontrol edebilir vb. Kullanıcı işlemleri için performans gereksinimleri, kullanıcıların işlemlerini ne kadar hızlı gerçekleştirebileceklerini ve yanıt sürelerini belirleyebilir.

Raporlama İşlemleri: OBS, farklı raporlar oluşturabilir, öğrenci performansını analiz edebilir ve istatistiksel veriler sunabilir. Raporlama işlemleri için performans gereksinimleri, raporların ne kadar hızlı oluşturulabileceğini ve büyük veri kümelerini nasıl yöneteceğini belirleyebilir.

Eşzamanlı Kullanıcılar: OBS, aynı anda birçok kullanıcının sistemle etkileşimde bulunabileceği bir ortamda çalışır. Performans gereksinimleri, sistemdeki eşzamanlı kullanıcı sayısına bağlı olarak sistem yanıt sürelerini ve hizmet seviyelerini belirleyebilir.

6.3 Güvenlik(Security)

Fiziksel Güvenlik: OBS'nin barındırıldığı sunucu veya donanım bileşenleri için fiziksel erişim kontrolü sağlanmalıdır. Bu, sunucu odalarının güvenliği, güvenlik kameraları, biyometrik kimlik doğrulama gibi önlemlerle gerçekleştirilebilir.

Veri Güvenliđi: OBS, öđrenci bilgileri, sınav sonuçları, mali veriler gibi hassas verileri içerir. Bu verilerin güvenliđi için şifreleme, veri erişim kontrolü, güvenli veri transferi ve depolama önlemleri alınmalıdır. Veri güvenliđi gereksinimleri, örneđin verilerin şifreli olarak depolanması veya belirli kullanıcıların yalnızca yetkili erişime sahip olması gibi şekillerde belirlenebilir.

Yazılım Güvenliđi: OBS'nin yazılım bileşenleri, güvenlik açıklarına karşı korunmalıdır. Bu, güvenlik açıklarının düzgün bir şekilde ele alındıđı ve güncel yazılım güvenlik standartlarını takip eden bir geliştirme süreci gerektirir. Gerekli güvenlik düzeltmeleri ve yamaları düzenli olarak uygulanmalıdır.

Güvenlik Politikaları ve Düzenlemeler: OBS, eğitim kurumunun güvenlik politikaları veya düzenlemeleriyle uyumlu olmalıdır. Bu politikalar, öđrenci verilerinin korunması, şifreleme gereksinimleri, kullanıcı hesapları ve erişim hakları yönetimi gibi konuları kapsayabilir. OBS'nin bu politikalara ve düzenlemelere uygun olarak tasarlanması ve işletilmesi önemlidir.

Güvenlik gereksinimleri, iş kuralları, güvenlik politikaları ve düzenlemelerden kaynaklanır. Bu gereksinimlerin belirlenmesi ve belgelenmesi, güvenlik açıklarının tespit edilmesini, güvenlik önlemlerinin alınmasını ve uygun güvenlik testlerinin yapılmasını sağlar. Güvenlik gereksinimlerine uygun bir şekilde tasarlanan ve uygulanan OBS, öđrenci verilerinin gizliliđini ve sistem güvenliđini sağlar.

6.4 Güvenlik(Safety)

OBS (Öđrenci Bilgi Sistemi & Eğitim Kurumu Bilgi Sistemi) için güvenlik gereksinimleri arasında "safety" yani güvenlik kavramı, kullanım sırasında ortaya çıkabilecek potansiyel kayıp, hasar veya zararları ele almaktadır. Bu gereksinimler, kullanıcıların ve sistemde yer alan verilerin güvenliđini sağlamayı hedefler.

Veri Kaybını Önleme: OBS'nin güvenlik kopyaları alınmalı ve düzenli olarak veri yedeklemesi yapılmalıdır. Böylece, olası bir sistem hatası, veri kaybı veya felaket durumunda verilerin geri yüklenmesi mümkün olacaktır.

Veri Bütünlüğü: OBS'de yer alan verilerin bütünlüğü korunmalıdır. Veri deđişiklikleri ve güncellemeleri, yetkilendirilmemiş erişimlere karşı korunmalı ve dođrulama mekanizmaları kullanılarak izlenmelidir.

Kimlik Doğrulama ve Yetkilendirme: Kullanıcıların kimlikleri doğrulanmalı ve yetkilendirme kontrolleri uygulanmalıdır. Kullanıcıların sadece yetkileri dahilinde işlemler yapabilmeleri sağlanmalıdır.

Güvenlik Sertifikaları ve Standartlar: OBS'nin, güvenlik sertifikaları ve standartlarına uygun olması gerekmektedir. Bu sertifikalar, ürünün güvenli olduğunu ve belirli güvenlik gereksinimlerini karşıladığını gösterir.

Acil Durum Yönetimi: OBS'nin, acil durum yönetimi prosedürleri belirlenmeli ve kullanıcıların güvenli bir şekilde sistemden çıkış yapabilmeleri veya acil durumlarla başa çıkabilmeleri için gerekli önlemler alınmalıdır.

7. Uluslararasılaştırma ve yerelleştirme gereklilikleri

Dil Desteği: OBS, farklı dilleri desteklemeli ve kullanıcıların tercih ettikleri dili seçebilmelerine olanak sağlamalıdır.

Bölgesel Ayarlar: OBS, kullanıcıların bulundukları bölgeye özgü ayarları desteklemeli, tarih formatları, saat dilimleri ve para birimi sembolleri gibi faktörleri kullanıcıların tercihlerine göre ayarlamalıdır.

Yerelleştirilmiş İçerik: OBS'nin içerikleri, kullanıldığı ülkelerin dil, kültür ve yazım kurallarına uygun olmalıdır. Metinler, etiketler ve iletişimler yerel dilde ve anlaşılır şekilde sunulmalıdır.

Sayı ve Para Birimi Formatları: OBS, farklı ülkelerin sayı formatlarına ve para birimi sembollerine uygun olarak çalışmalıdır. Kullanıcılar, kendi ülkelerine özgü sayı formatlarını ve para birimlerini görebilmelidir.

Coğrafi Konum Desteği: OBS, kullanıcıların coğrafi konum bilgilerini ve coğrafi verileri kullanabilmelerine olanak tanımalıdır. Harita entegrasyonu, konum bazlı bilgiler ve yerel hizmetler gibi özellikler sunulmalıdır.

Kültürel Hassasiyet: OBS, farklı kültürlerin adetlerine ve hassasiyetlerine saygı göstermelidir. Örneğin, belirli dini veya milli bayramlar, resmi tatiller ve diğer kültürel etkinlikler OBS'de göz önünde bulundurulmalıdır.

Tarih ve Zaman Dilimleri: OBS, farklı ülkelerin tarih ve zaman dilimlerini desteklemelidir. Kullanıcılar, kendi yerel tarih ve saat dilimlerine göre etkinlikleri planlayabilmelidir.

Uluslararası Düzenlemelere Uyum: OBS, farklı ülkelerin veri koruma yasalarına ve güvenlik düzenlemelerine uyum sağlamalıdır. Kullanıcı verilerinin gizliliği ve güvenliği ön planda tutulmalıdır.

Çoklu Dil Desteği: OBS, birden fazla dilin aynı anda kullanılmasına izin vermelidir. Kullanıcılar, arayüzde istedikleri dili seçebilmeli ve dil tercihlerini yönetebilmelidir.

Yerel İşlevsellik: OBS, kullanıldığı ülkelerin eğitim sistemlerine ve yerel gereksinimlerine uygun işlevsellik sunmalıdır.

8.Diğer Gereksinimler

Yasal ve Düzenleyici Uyum: OBS, geçerli yasal ve düzenleyici gereksinimlere uyum sağlamalıdır. Bu, gizlilik yasaları, veri koruma yönetmelikleri ve eğitim sektöründe geçerli olan diğer yasal düzenlemeleri içerebilir.

Finansal Uyum: OBS, mali işlemler ve raporlama açısından finansal uyum gereksinimlerini karşılamalıdır. Örneğin, gelir ve gider takibi, bütçeleme, muhasebe kuralları ve denetim gereksinimleri gibi finansal süreçleri desteklemelidir.

Kurulum ve Yapılandırma: OBS'nin kurulumu ve yapılandırması kolay olmalı ve kullanıcıların sistemi hızlı bir şekilde başlatmasını sağlamalıdır. Kurulum süreci, sistem gereksinimleri, donanım ve yazılım uyumluluğu, veritabanı yapılandırması ve diğer kurulum adımlarını içermelidir.

Başlangıç ve Kapatma Gereksinimleri: OBS, başlatma ve kapatma süreçlerini yönetebilmeli ve kullanıcıların sistemdeki oturumlarını düzgün bir şekilde sonlandırmasına olanak tanımalıdır. Örneğin, otomatik oturum kapatma, veri kaydetme ve sistem kapanış prosedürleri gibi gereksinimler yerine getirilmelidir.

Günlük Tutma, İzleme ve Denetim: OBS, kullanıcıların etkinliklerini günlüğe kaydetmeli, izlemeli ve denetlemelidir. Bu, kullanıcıların işlem geçmişlerini takip edebilmesini, hataları tespit edebilmesini ve denetim gereksinimlerini karşılayabilmesini sağlar.

9.Sonuç

9.1 Sözlük

SRS: Software Requirements Specification - Yazılım Gereksinimleri Belgesi. Bir yazılım projesinin gereksinimlerini, işlevlerini ve performansını tanımlayan resmi bir belge.

SR: System Requirement - Sistem Gereksinimi. Bir sistemin veya yazılımın karşılaması gereken işlevsel veya performansla ilgili bir gereklilik.

GUI: Graphical User Interface - Grafiksel Kullanıcı Arayüzü. Kullanıcının bir yazılım veya sistemi etkileşimde bulunmak için kullandığı grafik tabanlı arayüz.

API: Application Programming Interface - Uygulama Programlama Arayüzü. Yazılım bileşenlerinin veya sistemlerin birbirleriyle iletişim kurmasını sağlayan bir arayüz.

DBMS: Database Management System - Veritabanı Yönetim Sistemi. Veritabanı oluşturma, yönetme ve sorgulama işlemlerini gerçekleştiren bir yazılım sistemi.

UI: User Interface - Kullanıcı Arayüzü. Kullanıcının bir yazılım veya sistemi etkileşimde bulunmak için kullandığı arayüz.

UX: User Experience - Kullanıcı Deneyimi. Kullanıcının bir ürün veya sistemi kullanırken yaşadığı deneyimi ve memnuniyeti ifade eder.

QA: Quality Assurance - Kalite Güvencesi. Bir ürün veya sistemde belirlenen kalite standartlarının sağlanmasını ve hataların önlenmesini sağlayan süreçler ve faaliyetler.

R&D: Research and Development - Araştırma ve Geliştirme. Yeni ürünlerin veya teknolojilerin araştırılması ve geliştirilmesi süreci.

CMS: Content Management System - İçerik Yönetim Sistemi. İnternet sitelerinde içerik oluşturma, düzenleme ve yönetme işlevlerini sağlayan bir yazılım sistemi.

API: Application Program Interface - Uygulama Program Arayüzü. Yazılımlar arasında iletişim sağlamak için kullanılan tanımlanan yöntemler, işlevler ve protokoller kümesi.

CRM: Customer Relationship Management - Müşteri İlişkileri Yönetimi. Bir işletmenin müşterileriyle ilişkilerini yönetme, müşteri verilerini takip etme ve satış faaliyetlerini destekleme süreci.

ERP: Enterprise Resource Planning - Kurumsal Kaynak Planlaması. Bir işletmenin farklı departmanları arasındaki kaynakların planlanması, yönetimi ve entegrasyonunu sağlayan bir yazılım sistemi.

BPM: Business Process Management - İş Süreçleri Yönetimi. Bir işletmedeki süreçlerin analiz edilmesi, tasarlanması, uygulanması ve sürekli iyileştirilmesi süreci.

SDLC: Software Development Life Cycle - Yazılım Geliştirme Yaşam Döngüsü. Bir yazılım projesinin başlangıcından sonuna kadar olan süreçlerin aşamalarını tanımlayan bir yönetim metodolojisi.

UI/UX Design: User Interface/User Experience Design - Kullanıcı Arayüzü/Kullanıcı Deneyimi Tasarımı. Bir yazılımın kullanıcı dostu bir arayüze ve olumlu bir kullanıcı deneyimine sahip olmasını sağlayan tasarım süreci.

CI/CD: Continuous Integration/Continuous Deployment - Sürekli Entegrasyon/Sürekli Dağıtım. Yazılım geliştirme sürecinde, kodun sürekli olarak test edilmesi, entegre edilmesi ve dağıtılması süreci.

API Testing: Application Programming Interface Testing - Uygulama Programlama Arayüzü Testi. Bir API'nin doğruluğunu, güvenilirliğini ve performansını test etme süreci.

DevOps: Development and Operations - Geliştirme ve Operasyonlar. Yazılım geliştirme ve işletim ekiplerinin işbirliği içinde çalışarak hızlı ve sürekli yazılım dağıtımını sağlayan bir kültür ve metodoloji.

SLA: Service Level Agreement - Hizmet Düzey Anlaşması. Bir hizmet sağlayıcının müşterisiyle belirli hizmet düzeyleri ve performans hedefleri üzerinde anlaşma sağladığı bir sözleşme.

UAT: User Acceptance Testing - Kullanıcı Kabul Testi. Bir yazılımın veya sistemin, kullanıcının gereksinimlerini ve beklentilerini karşıladığını doğrulamak için kullanıcının gerçekleştirdiği test süreci.

POC: Proof of Concept - Kavramın Kanıtı. Bir fikrin veya yeni bir teknolojinin işleyişini ve potansiyelini göstermek için yapılan küçük ölçekli bir deneme veya prototip çalışması.

ROI: Return on Investment - Yatırım Getirisi. Bir yatırımın elde edilen karın yatırılan sermayeye oranı.

MVP: Minimum Viable Product - Minimum İşlevsel Ürün. Bir ürünün veya hizmetin gerekli temel işlevselliğe sahip en basit versiyonu.

KPI: Key Performance Indicator- Anahtar Performans Göstergesi. Bir organizasyonun veya projenin başarı seviyesini ölçmek ve değerlendirmek için kullanılan ölçülebilir hedefler ve metrikler.

BOM: Bill of Materials - Malzeme Listesi. Bir ürünün üretiminde kullanılan tüm malzemelerin ve bileşenlerin listesi.

ROI: Return on Investment - Yatırım Getirisi. Bir yatırımın elde edilen karın yatırılan sermayeye oranı.

SLA: Service Level Agreement - Hizmet Düzey Anlaşması. Bir hizmet sağlayıcının müşterisiyle belirli hizmet düzeyleri ve performans hedefleri üzerinde anlaşma sağladığı bir sözleşme.

DNS: Domain Name System - Alan Adı Sistemi. İnternet üzerindeki alan adlarını IP adresleriyle eşleştiren ve ağ trafiğini yönlendiren bir sistem.

VPN: Virtual Private Network - Sanal Özel Ağ. İnternet üzerinden güvenli bir şekilde bağlantı kurmayı sağlayan bir ağ teknolojisi.

CPU: Central Processing Unit - Merkezi İşlem Birimi. Bilgisayarlarda verilerin işlendiği ve komutların yürütüldüğü ana bileşen.

RAM: Random Access Memory - Rastgele Erişim Belleği. Bilgisayarda geçici olarak verilerin depolandığı ve hızlı erişim sağlayan bellek.

API: Application Programming Interface - Uygulama Programlama Arayüzü. Yazılım bileşenlerinin veya sistemlerin birbirleriyle iletişim kurmasını sağlayan bir arayüz.

HTML: Hypertext Markup Language - Hipertext İşaretleme Dili. Web sayfalarının oluşturulması için kullanılan standart işaretleme dil.

CSS: Cascading Style Sheets - Basamaklı Stil Şablonları. Web sayfalarının görünümünü ve düzenini belirlemek için kullanılan stil şablonları dil.

9.2 Kaynakça

Akseki, B. (2011). Yüksek lisans eğitiminde kullanılan öğrenci bilgi sistemi, Ankara Üniversitesi Eğitim Bilimleri Enstitüsü öğrenci bilgi sistemine yönelik bir değerlendirme. Yayınlanmamış Tezsiz Yüksek Lisans Projesi, Ankara Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.

Altınışık, U. (2006). Öğrenci bilgi sisteminde veri madenciliğinin uygulanması. Yayınlanmamış Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli.

Arseven, A.D. (2001). Alan Araştırma Yöntemi, İlkeler, Teknikler, Örnekler, Ankara: Gündüz Eğitim Ve Yayıncılık.

Balki, R.S. (2004). Student and parent appraisal of learning management system in a private schoolGraduate Program İn Secondary School Science And Mathematics Education Boğaziçi University, İstanbul.

Başar, M.S. (1999). Lisans öğrencilerinin takibi için bir öğrenci bilgi sisteminin geliştirilmesi. Yayınlanmamış Yüksek Lisans Tezi, Atatürk Üniversitesi Sosyal Bilimler Enstitüsü, Erzurum.

Çeken, C. (2001). Öğrenci bilgi sistemi. Yayınlanmamış Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli.

Ergin, İ. (2008). Lisansüstü eğitimde kullanılan öğrenci bilgi sistemi. Yayınlanmamış Tezsiz Yüksek Lisans Projesi, Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.

Kaptan, S. (1998). Bilimsel araştırma ve istatistik teknikleri, Ankara: Bilim Kitap.

Yıldırım, T. (2007). Ajax ve uml destekli web tabanlı öğrenci bilgi sistemi portalının hazırlanması. Yayınlanmamış Yüksek Lisans Tezi, Havacılık ve Uzun Teknolojileri Enstitüsü, İstanbul.

Eğitim teknolojisinin tarihi gelişimi. (2010, 10, 12). 12, 10, 2010 tarihinde <http://www.veribaz.com: http://www.veribaz.com/viewdoc.html?egitim-teknolojisinin-tarihi-gelisimi-343365.html> adresinden alındı.

E-learning: Yeni Yüzyılın Eğitim Sistemi. (2010, 11, 19). 19, 11, 2010 tarihinde www.bilim.org adresinden alındı.

İnternet Nedir? (2010, 11, 14). 14, 11, 2010 tarihinde http://ogrenci.hacettepe.edu.tr : http://ogrenci.hacettepe.edu.tr/~b0343623/baglantilar/internet_nedir.html adresinden alındı.

İnternetin Tarihsel Gelişimi. (2010, 11, 14). 14, 11, 2010 tarihinde http://tr.wikipedia.org: http://tr.wikipedia.org/wiki/İnternetin_tarihsel_gelisimi adresinden alındı.

Avrupa'da Bilgi Toplumunun Oluşturulması İçin Ortak Girişim Eylem Planı . (2010, 11, 23). 23, 11, 2010 tarihinde http://www.bilgitoplumu.gov.tr/Documents/3/Diger/ 010600_eAvrupa+ EylemPlani. pdf adresinden alındı.

İnternet Tabanlı Uzaktan Eğitim Uygulamaları. (2010, 11, 23). 23, 11, 2010 tarihinde inettr.org.tr/inetconf7/bildiriler/37.doc adresinden alındı.

Türkiye'de Üniversitelerde İnternet Kullanımı. (2010, 11, 23). 23, 11, 2010 tarihinde inettr.org.tr/inetconf6/tammetin/tuna-tam.doc adresinden alındı.