Öğrenci Bilgi Sistemi Sistem Tasarımı

**1. Sistem Mimarisi ve Modüller**

**1.1. Sistem Mimarisi**

Sistem mimarisi, ÖBS'nin genel yapısını ve bileşenlerinin nasıl etkileşeceğini tanımlar. Önerilen mimari:

**1.2. Modüller**

* **Öğrenci Yönetimi Modülü:** Öğrenci kayıtları, bilgilerinin güncellenmesi.
* **Ders Yönetimi Modülü:** Derslerin oluşturulması, güncellenmesi ve silinmesi.
* **Not Yönetimi Modülü:** Notların girilmesi ve görüntülenmesi.
* **İletişim Modülü:** Duyurular ve mesajlaşma işlevleri.

**2. Detaylı Tasarım**

**2.1. Modül Tasarımı**

Her modül için işlevler ve etkileşimler belirlenir.

* **Öğrenci Yönetimi:**
  + **Fonksiyonlar:** Kayıt ol, bilgileri güncelle, bilgileri görüntüle.
* **Ders Yönetimi:**
  + **Fonksiyonlar:** Ders ekle, ders güncelle, ders sil.
* **Not Yönetimi:**
  + **Fonksiyonlar:** Not ekle, not görüntüle.
* **İletişim Modülü:**
  + **Fonksiyonlar:** Duyuru ekle, mesaj gönder.

**2.2. Sınıf Diyagramları**

Sınıf diyagramları, sistemdeki nesneleri ve ilişkilerini gösterir.

**2.2.1. Sınıf Tanımları**

* **Öğrenci Sınıfı:**
  + Özellikler: öğrenciID, isim, soyisim, doğumTarihi, email.
  + Metotlar: kayıtOl(), bilgileriGüncelle().
* **Ders Sınıfı:**
  + Özellikler: dersID, dersAdı, krediSayısı, öğretimÜyesi.
  + Metotlar: dersEkle(), dersSil().
* **Not Sınıfı:**
  + Özellikler: notID, öğrenciID, dersID, notDeğeri.
  + Metotlar: notEkle(), notGörüntüle().

**2.3. Veri Tabanı Tasarımı**

Veri tabanı, sistemin verilerini nasıl depolayacağını belirler.

**2.3.1. Tablolar**

* **Öğrenci Tablosu:**
  + Alanlar: öğrenciID (PK), isim, soyisim, doğumTarihi, email.
* **Ders Tablosu:**
  + Alanlar: dersID (PK), dersAdı, krediSayısı, öğretimÜyesi.
* **Not Tablosu:**
  + Alanlar: notID (PK), öğrenciID (FK), dersID (FK), notDeğeri.

**2.3.2. İlişkiler**

* Öğrenci ile Not arasında 1

ilişkisi (bir öğrenci birden fazla not alabilir).

* Ders ile Not arasında 1

ilişkisi (bir ders için birden fazla not kaydı olabilir).

**3. API Tasarımı**

**3.1. API Uç Noktaları**

API tasarımı, farklı sistem bileşenlerinin etkileşimini sağlar.

* **Öğrenci Kayıt API:**
  + **Uç Nokta:** /api/ogrenci/kayit
  + **Metot:** POST
  + **Açıklama:** Yeni öğrenci kaydı oluşturur.
* **Ders Ekleme API:**
  + **Uç Nokta:** /api/ders/ekle
  + **Metot:** POST
  + **Açıklama:** Yeni ders ekler.
* **Not Görüntüleme API:**
  + **Uç Nokta:** /api/not/goruntule/{ogrenciID}
  + **Metot:** GET
  + **Açıklama:** Öğrencinin notlarını görüntüler.

**3.2. Veri Formatları**

* **JSON:** Tüm API uç noktaları JSON formatında veri alır ve gönderir.

**4. Algoritma ve Akış Diyagramları**

**4.1. Algoritmalar**

Algoritmalar, belirli işlevlerin nasıl gerçekleştirileceğini tanımlar.

**4.1.1. Öğrenci Kayıt Algoritması**

1. Kullanıcı kayıt formunu doldurur.
2. Veriler kontrol edilir (geçerlilik).
3. Veritabanına kaydedilir.
4. Kullanıcıya onay mesajı gönderilir.

**4.2. Akış Diyagramı**

Akış diyagramı, süreç akışını görselleştirir.

* Başla
* Kullanıcıdan verileri al
* Verileri kontrol et
* Veritabanına kaydet
* Onay mesajı gönder
* Bitir

**5. Hata Yönetimi**

**5.1. Hata Türleri**

Hata yönetimi, sistemde karşılaşılabilecek hataları tanımlar.

* **Doğrulama Hataları:** Kullanıcıdan alınan verilerin geçerliliğini kontrol etme.
* **Veritabanı Hataları:** Veritabanı bağlantısı sorunları veya veri kaydı hataları.

**5.2. Hata Yönetim Stratejileri**

* **Loglama:** Hataların kaydedilmesi ve analiz edilmesi.
* **Kullanıcı Bildirimleri:** Hata durumunda kullanıcıya bilgilendirici mesajlar gönderme.

**6. Performans Optimizasyonu**

**6.1. Veri Tabanı Optimizasyonu**

* **Index Kullanımı:** Sorgu performansını artırmak için sık kullanılan alanlarda indeks oluşturma.
* **Sorgu Optimizasyonu:** Karmaşık sorguların basitleştirilmesi ve performanslarının iyileştirilmesi.

**6.2. Cache Yönetimi**

* **Cache Kullanımı:** Sık erişilen verilerin bellekte tutulması. Örneğin, kullanıcı oturum bilgileri ve ders bilgileri.

**6.3. Yük Dengeleme**

* **Sunucu Yük Dengelemesi:** Sistem yükü arttığında, yük dengeleyiciler kullanarak sunucu performansını artırma.