

**T.C.  
YILDIZ TEKNİK ÜNİVERSİTESİ  
ELEKTRONİK ELEKTRİK MÜHENDİSLİĞİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**YAZILIM MÜHENDİSLİĞİ**  
**Yazılım Gereksinim Belgesi**

---

**Stardust Language Course**

**Proje Grubu**

16011005 M. Serdar Ormancı

16011017 Mert Öz

16011084 Mustafa Katipoğlu

16011087 Caner Kaya

17011044 Uğur Keskin

# İçerik

<b>İçerik</b>	<b>2</b>
<b>1- Giriş</b>	<b>4</b>
1.1 Amaç	5
1.2 Doküman Standartları	5
1.3 Hedef Kitle	5
<b>2- Genel Tanımlama</b>	<b>6</b>
2.1 Ortam, Teknoloji ve Donanımı	7
2.2 Tasarım ve Uygulama Kısıtları	7
2.3 Kullanıcı Dökümantasyonun Taşınması Gereken Özellikler	7
2.4 Kabuller ve Etkileşimler	7
<b>3-Arayüz Gereksinimleri</b>	<b>8</b>
3.1 Kullanıcı Arayüzleri	8
3.2 Donanım Arayüzleri	9
3.3 Yazılım Arayüzleri	9
3.4 İletişim Arayüzleri	9
<b>4-Sistem Özellikleri</b>	<b>10</b>
4.1 Öğrenciyi Sisteme Ekleme	10
4.1.1 Tanım	10
4.1.2 Uyarılar	10
4.1.3 Fonksiyonel Gereksinimler	10
4.2 Öğrenciyi Derse Kaydetme	11
4.2.1 Tanım	11
4.2.2 Uyarılar	11
4.3 Sisteme Yeni Kurs Ekleme	12
4.3.1 Tanım	12
4.3.2 Uyarılar	12
4.3.3 Fonksiyonel Gereksinimler	12
4.4 Ders Programını Görüntüleme	12
4.4.1 Tanım	12
4.4.2 Uyarılar	12
4.4.3 Fonksiyonel Gereksinimler	13
4.5 Şubeleri Görüntüleme	14

4.5.1 Tanım	14
4.5.2 Uyarılar	14
4.5.3 Fonksiyonel Gereksinimler	14
4.6 Profil Bilgilerini Görüntüleme	15
4.6.1 Tanım	15
4.6.2 Uyarılar	15
4.6.3 Fonksiyonel Gereksinimler	16
4.7 Profil Bilgilerini Düzenleme	16
4.7.1 Tanım	16
4.7.2 Uyarılar	16
4.7.3 Fonksiyonel Gereksinimler	16
4.8 Yeni Şube Ekleme	17
4.8.1 Tanım	17
4.8.2 Uyarılar	17
4.8.3 Fonksiyonel Gereksinimler	18
4.9 Şube Silme	19
4.9.1 Tanım	19
4.9.2 Uyarılar	19
4.9.3 Fonksiyonel Gereksinimler	19
4.10 Yeni Ders Ekleme	19
4.10.1 Tanım	19
4.10.2 Uyarılar	20
4.10.3 Fonksiyonel Gereksinimler	20
4.11 Ders Silme	20
4.11.1 Tanım	20
4.11.2 Uyarılar	21
4.12 Yeni Kur Ekleme	22
4.12.1 Tanım	22
4.12.2 Uyarılar	22
4.12.3 Fonksiyonel Gereksinimler	22
4.13 Kur Silme	23
<b>5- Fonksiyonel Olmayan Diğer Gereksinimler</b>	<b>25</b>
5.1 Performans Gereksinimleri	25
5.2 Mantıksal Veritabanı Gereksinimleri	25
5.3 Kalite Özellikleri	26
5.3.1 Güvenilirlik	26
5.3.2 Kullanılabilirlik	26
5.3.3 Güvenlik	26

5.3.4 Taşınabilirlik	26
5.3.5 Kullanılabilirlik	26

# 1- Giriş

## 1.1 Amaç

Stardust Language Course şirketi yeni şubeler açarak gelişmeyi planlamaktadır. Bu gelişim sürecinde ve sonrasında bilgi işlemlerinin en az karmaşıklıkta ilerleyebilmesi için otomasyon sistemine ihtiyaç duymaktadır.

Müşterinin bu eksiğini gidermek amacıyla kurs otomasyon sistemi geliştirilecektir. Kurs otomasyon sisteminin temel amacı öğretmen, öğrenci, kayıt eleman ve sistem yöneticisi dahil tüm sistem kullanıcılarına kullanımı kolay, hızlı, kolay erişilebilir ve arayüz sağlamaktır.

## 1.2 Doküman Standartları

Doküman IEEE 830 standartlarına uygun olarak geliştirilmiştir.

## 1.3 Hedef Kitle

Stardust Language Course şirketinde bulunan:

Öğrenciler

Öğretmenler

Kayıt Elemanları

Şube Yöneticileri

Uygulama geliştirilirken yukarıda belirtilen hedef kitlesinin ihtiyaç ve istekleri göz önünde bulundurulmalıdır.

## 1.4 Ürün Kapsamı

Ürün dil kursu işlemlerini dijital dönüşüm kapsamında daha kullanışlı hale getirmektedir.

- Teknolojinin nimetlerinden faydalanılarak kursa kayıt işlemlerinin hızlandırılması, takip edilmesinin kolaylaştırılması sağlanacaktır.
- Ödeme işlemlerinin açık bir şekilde takip edilmesi ve sorgulanmasını sağlayacaktır.
- Farklı şubelerden yapılan işlemler kolayca tek bir çatı altında toplanabilecektir.
- Verilerin tutarlılığı ve güvenliği kontrol altına alınacaktır. Sistemimiz diğer dil kurslarıyla rekabette güçlü bir koz olarak yer alacaktır.
- Dil Kursu Bilgi Sistemiyle birlikte kurslara ait kampanyaların yönetilmesi de duyurulması internet ortamında çok daha pratik hale gelecektir.

## 2- Genel Tanımlama

Stardust Language Course uygulaması kurumsal şirketler için tasarlanmış bir sistemdir, kişiye özel değildir. Uygulamayı kullanmak isteyen kurumsal firmaların, öğretmenlerin ve öğrencilerin istenildiği bilgileri vermek zorundadır.

### 2.1 Ortam, Teknoloji ve Donanımı

Veritabanı yönetim sistemi olarak PostgreSQL, mobil uygulama tasarımı Android işletim sistemi ve tasarım için Material design yaklaşımları ve Adobe XD kullanıldı.

### 2.2 Tasarım ve Uygulama Kısıtları

Sistemde, admin kullanıcısı her kullanıcının bütün fonksiyonlarını yerine getirebilir. Sistemde sadece kayıt elemanı öğrenci kayıt yapabilmektedir. Öğrenciler öğretmenlerin, öğretmenler öğrencilerin bilgilerini görememektedir. Burada bahsedilen kısıtlamalar kurumsal firmanın istediği kısıtlamalardır.

### 2.3 Kullanıcı Dökümantasyonun Taşınması Gereken Özellikler

Dökümanda, uygulamada bulunan, buton, checkbox gibi kullanıcıdan input alındığı her birinin ne tür input aldığı neler yaptığı bulunmalıdır. Dökümanda kullanıcıları bilgilendirici senaryolar olmalıdır. Uygulamayı kullanan kurumsal firmalar, herhangi bir hatada nasıl online bilgi alabilecekleri anlatılmalıdır.

### 2.4 Kabuller ve Etkileşimler

Tasarladığımız sistemde hedeflenen kitle bireysel kullanıcılardan çok kurumsal şirketlerdir. Sistemde yöneticiler belli bir dil için yeni bir ders açabilir, bu işlem için sistem o şubedeki hem dil bilgisi ve boş zamanları uygun olan öğretmenlerin, hem de boş sınıfların listesini sunarak sistem yöneticisinin kolayca seçim yapabilmesine olanak sağlar.

Şubeler için çalışan kayıt elemanları uygulamayı kullanarak, gelen öğrencileri herhangi bir şubede açılan derslere kayıt edebilir.

Uygulamada;

- Şubelerin iletişim bilgileri, toplu taşıma veya özel otomobil ile ulaşım talimatları, sosyal olanakları
- Öğretmenler için isim, öğretebileceği diller, ev ve cep telefonları, işe başladığı tarih, çalışabileceği günler ve saatler, ders verebileceği şubeler.
- Öğrenciler için isim, yazıldığı kurs ve kur, ev ve cep telefonları, ödeme bilgileri.

gibi bilgiler uzak veritabanında saklanmaktadır.

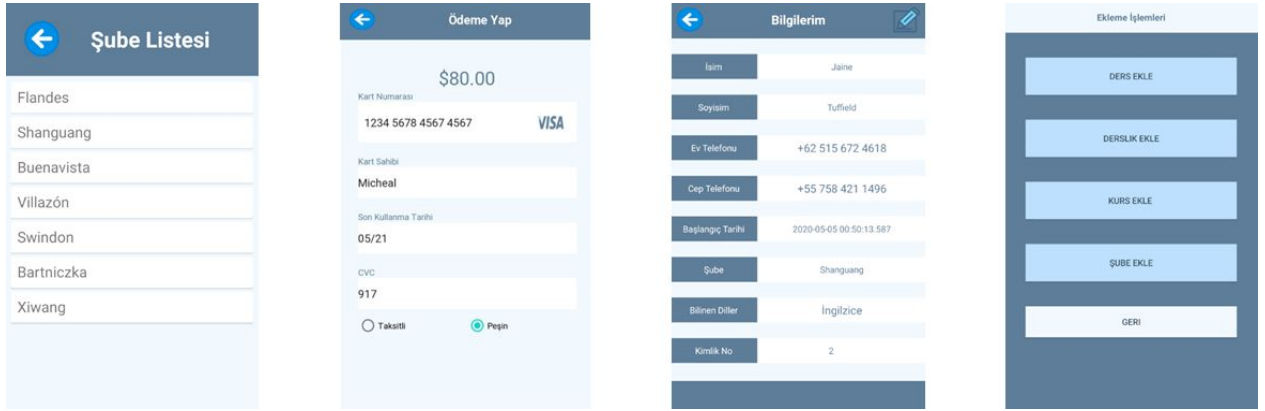
Uygulamayı kullanan öğrenciler kurs ücretini peşin veya taksitli olarak ödeyebilir. Bir öğrencinin kaydı açıldığında ödemediği taksitler varsa onlar da gösterilmeli, kayıt memuru uygulamayı kullanarak bir taksiti tahsil ettiğinde bunu ödendi olarak sisteme işaretler.



## 3-Arayüz Gereksinimleri

### 3.1 Kullanıcı Arayüzleri

Kullanıcılar yetkileri olan ekranlara yönlendirilmelidir. Öğrenci, öğretmen, kayıt elemanı ve sistem yöneticisi farklı kullanıcı arayüzü kullanır. Her ekranda kullanıcıların aksiyonlarına bağlı butonlar olacaktır. Kullanıcı hatalı işlem yaptığında bilgilendirilmelidir. Kullanıcı login ekranında hatalı bilgi girerse uyarı almalıdır. Uygulamanın kullanımı kolay olup, her tür kullanıcıya hitap eden bir tasarıma sahip olmalıdır.



### 3.2 Donanım Arayüzleri

Bu uygulama android OS destekleyen mobil cihazlarda ve tabletlerde çalışır. Başka bir donanıma gereksinim yoktur.

### 3.3 Yazılım Arayüzleri

Uygulamayı kullanacak cihaz Android v8 ve üzeri versiyonlarına sahip olmalıdır.

### 3.4 İletişim Arayüzleri

Sistem Amazon Web Services tarafından sunulan RDS hizmetinde online olarak sunulur. Kullanıcı aksiyonları ile veritabanında gerçek zamanlı güncelleme yapabilir. Bu nedenle uygulamayı kullanacak cihaz internet erişimi yapabiliyor olmalıdır.

## 4-Sistem Özellikleri

### 4.1 Öğrenciyi Sisteme Ekleme

#### 4.1.1 Tanım

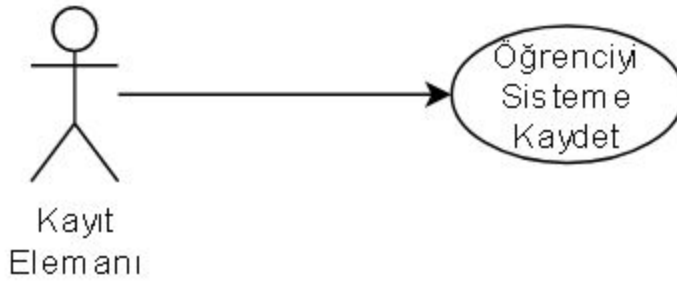
Sistemin temel fonksiyonlarından biridir. Sisteme öğrenci kaydının yapılmasını sağlar. Bu modülü kayıt elemanları gerçekleştirebilir.

#### 4.1.2 Uyarılar

Öğrencinin e-posta hesabı kendine özeldir. Daha önceden eklenip sistemde mevcut olan e-posta hesap ile yeni bir öğrenci kaydı yapılamaz. Aynı durum öğrencinin kimlik numarası bilgisi için de geçerlidir.

#### 4.1.3 Fonksiyonel Gereksinimler

Sistemde bulunan e-posta adreslerinin ve öğrenci kimlik numaralarının kontrol edilmesi gerekir.



### 4.2 Öğrenciyi Derse Kaydetme

#### 4.2.1 Tanım

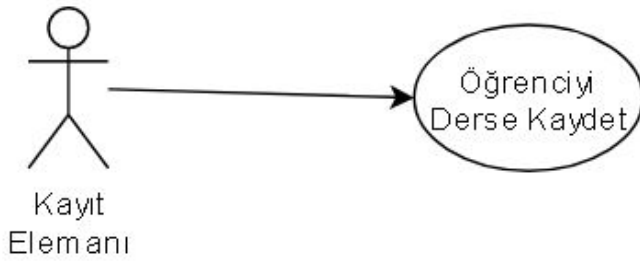
Sistemin temel fonksiyonlarından biridir. Sistemdeki öğrencinin seçilen şubenin seçilen kursundaki derse kaydının yapılmasını sağlar. Bu modülü yalnızca Kayıt Yöneticisi gerçekleştirebilir.

#### 4.2.2 Uyarılar

Eklenecek öğrencinin sistemde kayıtlı olması gereklidir.

#### 4.2.3 Fonksiyonel Gereksinimler

Sistemde mevcut olan öğrencilerin getirilmesi gereklidir. Sistemde mevcut olan şubelerin listelenmesi gereklidir. Seçilen şubenin kurslarının listelenmesi gereklidir. Seçilen şube ve kurs bilgileri ile eşleşen kurların listelenmesi gerekmektedir.



### 4.3 Sisteme Yeni Kurs Ekleme

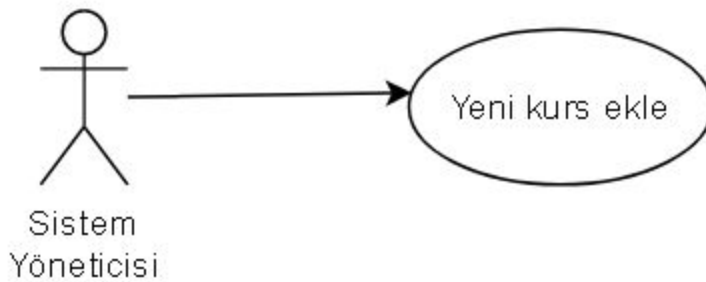
#### 4.3.1 Tanım

Belirtilen şubeye yeni kurs eklemesinin yapılmasını sağlar. Bu modülü yalnızca sistem yöneticisi gerçekleştirebilir.

#### 4.3.2 Uyarılar

Eklenecek olan kurs, ekleneceği şubede daha önceden mevcut olmamalıdır.

#### 4.3.3 Fonksiyonel Gereksinimler



Sistemden, seçilmiş olan şubede bulunan kursların getirilmesi gerekmektedir.

## 4.4 Ders Programını Görüntüleme

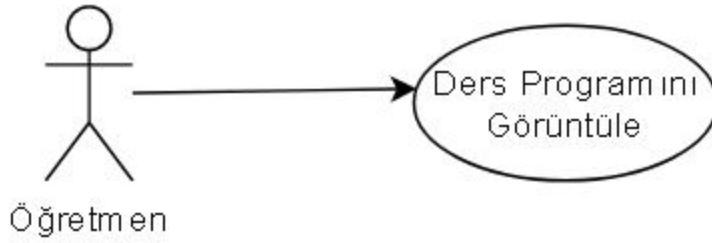
### 4.4.1 Tanım

Sisteme kayıtlı olan öğretmenlerin mevcut ders programını görüntüleyebilmesini sağlayan modüldür. Sistemde mevcut olarak bulunan öğretmenlerin haftalık ders programı tablosunu görmesini sağlar. Bu modülü yalnızca Öğretmenler kullanabilir.

### 4.4.2 Uyarılar

Öğretmen, sistemde kayıtlı olarak bulunmalıdır.

### 4.4.3 Fonksiyonel Gereksinimler



Sistemden, öğretmenler getirilmelidir.

Sistemden, mevcut öğretmenin çalışma saatleri getirilmelidir.

Getirilen çalışma saatleri uygun formata getirilecek şekilde işlenmelidir.

<b>ID:</b>	#7
<b>Title:</b>	Ders Programını Görüntüle
<b>Description:</b>	Öğretmen haftalık ders programını görüntüler
<b>Primary Actor:</b>	Öğretmen
<b>Preconditions:</b>	Öğretmene ders
<b>Postconditions:</b>	-
<b>Main Success Scenario:</b>	Öğretmen kendine ait ders programını tablo şeklinde görüntüler.

<b>Extensions:</b>	-
<b>Frequency of Use:</b>	Arada Sırada
<b>Status:</b>	Finished
<b>Owner:</b>	Serdar ORMANCI
<b>Priority:</b>	Yüksek

## 4.5 Şubeleri Görüntüleme

### 4.5.1 Tanım

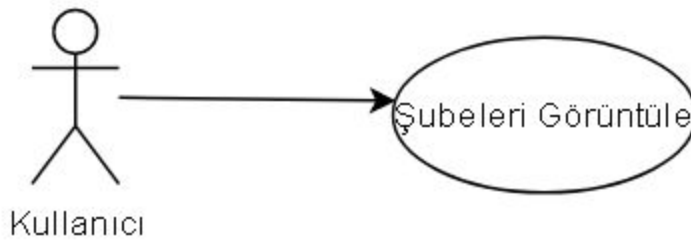
Sistemin, şubeler ile ilgili bilgiler görüntülemesini sağlayan modüldür. Uygulama ile ilgili kişilerin şubelerdeki sosyal olanaklar, ulaşım ve iletişim bilgileri konularında bilgilendirilmesini sağlar. Bu modül sistemde kayıtlı olan herkes kullanabilir.

### 4.5.2 Uyarılar

Herhangi bir uyarı olmasını gerektirecek bir durum yoktur.

### 4.5.3 Fonksiyonel Gereksinimler

### 4.5.3 Fonksiyonel Gereksinimler



Sistemde mevcut olan şubelerin getirilmesi gerekmektedir.  
Getirilen şubelerin liste formatında ekranda görüntülenmesi gerekmektedir

<b>ID:</b>	#5
<b>Title:</b>	Şubeleri görüntüle
<b>Description:</b>	Şube listesi görüntülenir.
<b>Primary Actor:</b>	Kayıt Elemanı, Sistem Yöneticisi, Öğrenci, Öğretmen
<b>Preconditions:</b>	Sistemde şube olması gerekiyor.
<b>Postconditions:</b>	-
<b>Main Success Scenario:</b>	Kurumun bütün şubeleri bilgileriyle listelenir.
<b>Extensions:</b>	---
<b>Frequency of Use:</b>	Arada Sırada
<b>Status:</b>	Finished
<b>Owner:</b>	Serdar ORMANCI
<b>Priority:</b>	Yüksek

## 4.6 Profil Bilgilerini Görüntüleme

Sistemde bulunan her bir aktörün sistemde tutulan bilgilerinin gösterildiği kısım.

### 4.6.1 Tanım

Kullanıcının, kendisi ile ilgili bilgileri görüntülemesini sağlayan modüldür. Bu modülü sisteme kayıtlı olan herkes kullanabilir.

### 4.6.2 Uyarılar

Herhangi bir uyarı olmasını gerektirecek bir durum yoktur.

#### 4.6.3 Fonksiyonel Gereksinimler



Sistemde seçili kişi bilgileri getirilmelidir.  
Bilgiler belirlenen alanlarda ekranda uygun alanlarda gösterilmelidir.

### 4.7 Profil Bilgilerini Düzenleme

#### 4.7.1 Tanım

Kullanıcının, kendisi ile ilgili iletişim bilgilerini düzenlemesini sağlayan modüldür. Bu modülü sisteme kayıtlı olan herkes kullanabilir.

#### 4.7.2 Uyarılar

Kullanıcı, telefon bilgilerini uygun bir formatta girilmelidir.

#### 4.7.3 Fonksiyonel Gereksinimler



Seçili kişinin bilgileri bilgilere ait alanlarda düzenlenmeli ve onaylanmalıdır



<b>ID:</b>	#3
<b>Title:</b>	Profili Görüntüle
<b>Description:</b>	Kullanıcı profil resmi ve diğer kişisel bilgilerini günceller
<b>Primary Actor:</b>	Öğrenci, Öğretmen
<b>Preconditions:</b>	Kayıt Ol
<b>Postconditions:</b>	---
<b>Main Success Scenario:</b>	Kullanıcı profil bilgilerini görüntüler. Ayrıca öğrenci ödeme bilgilerini, öğretmen ise ders programını görüntüler.
<b>Extensions:</b>	Profil bilgilerini güncelle, Ödeme seçeneği belirle, Ödenmemiş taksitleri görüntüle
<b>Frequency of Use:</b>	Sıklıkla
<b>Status:</b>	Finished
<b>Owner:</b>	Serdar ORMANCI
<b>Priority:</b>	Normal

## 4.8 Yeni Şube Ekleme

Bu kısımda sistem yöneticisine özel şube ekranı

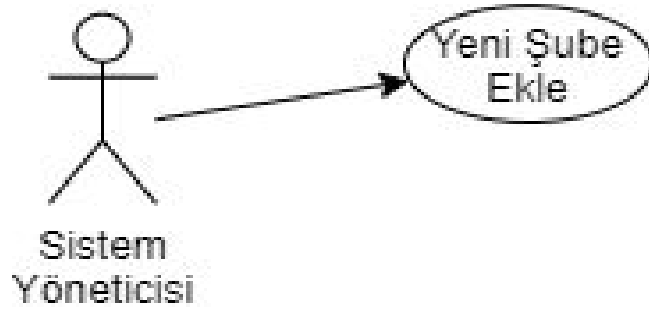
### 4.8.1 Tanım

Sistemin, yeni bir şube eklemesini sağlayan modüldür. Bu modülü yalnızca Sistem Yöneticisi kullanabilir.

### 4.8.2 Uyarılar

Aynı isimde 2 tane şube aynı anda mevcut olamaz. Şube adının kendine özgü bir ad seçilmesi gereklidir.

### 4.8.3 Fonksiyonel Gereksinimler



Sisteme yeni şube eklerken gerekli kontrol yapıldıktan sonra sistem yöneticisi yeni şube ekliyor.

<b>ID:</b>	#1
<b>Title:</b>	Yeni Şube Ekle
<b>Description:</b>	Sistem yöneticisi eğitim kurumuna yeni bir şube ekler
<b>Primary Actor:</b>	Sistem Yöneticisi
<b>Preconditions:</b>	Şube bilgilerini ayarla
<b>Postconditions:</b>	Şubeyi onayla
<b>Main Success Scenario:</b>	Eğer başarıyla şube bilgileri girilirse yeni bir şube eklenir.
<b>Extensions:</b>	-
<b>Frequency of Use:</b>	Çok nadir
<b>Status:</b>	Finished
<b>Owner:</b>	Serdar ORMANCI
<b>Priority:</b>	Yüksek

## 4.9 Şube Silme

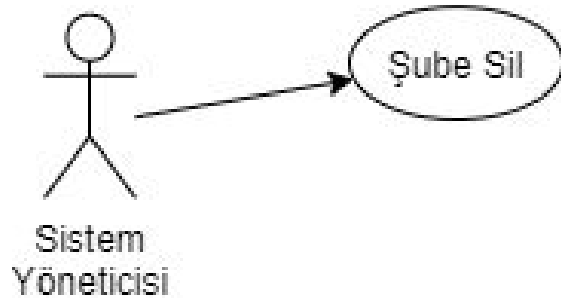
### 4.9.1 Tanım

Sistemin, mevcut bir şubeyi sistemden silmesini sağlayan modüldür. Bu modülü yalnızca Sistem Yöneticisi kullanabilir.

### 4.9.2 Uyarılar

İsmi girilen şubenin sistemde mevcut olması gereklidir.

### 4.9.3 Fonksiyonel Gereksinimler



Sistem yöneticisi seçilen şubeyi görmeli ve daha sonrasında silme butonuna basarak silmelidir. Silme işlemi başarıyla tamamlandıysa doğrulama mesajı, tamamlanmadıysa hata mesajı almalıdır.

## 4.10 Yeni Ders Ekleme

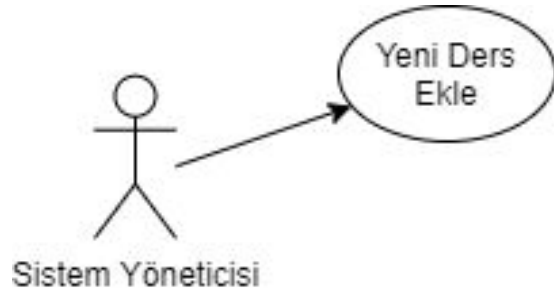
### 4.10.1 Tanım

Sisteme yeni bir ders eklenmesini sağlayan modüldür. Bu modülü yalnızca Sistem Yöneticisi kullanabilir.

## 4.10.2 Uyarılar

Aynı şubedeki aynı kursun aynı kurunun aynı anda birden fazla aynı isimdeki dersi mevcut olamaz. Dersin, daha önceden ekleme yapılacak aynı şubedeki aynı kurstaki ve aynı kurdaki derslerden farklı bir ders olması gerekmektedir.

## 4.10.3 Fonksiyonel Gereksinimler



Sistem yöneticisi geçerli ders bilgileri girdiği kontrol edildikten sonra ekle butonuna basarak ders eklemelidir.

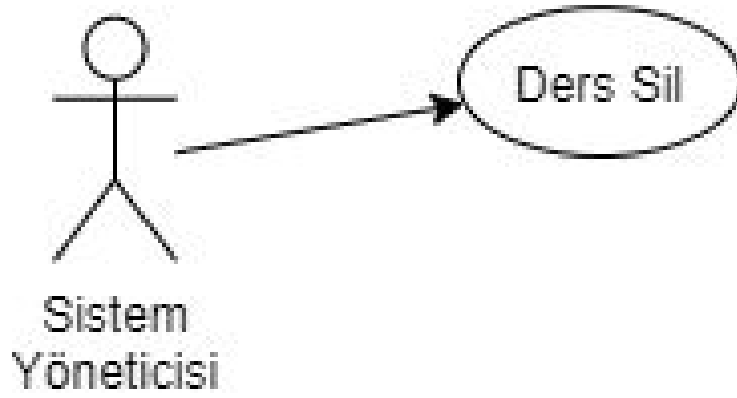
## 4.11 Ders Silme

### 4.11.1 Tanım

Sistemin, mevcut bir dersi sistemden silmesini sağlayan modüldür. Bu modülü yalnızca Sistem Yöneticisi kullanabilir.

## 4.11.2 Uyarılar

Şube, kurs ve kur bilgisi girilmiş olan dersin sistemde mevcut olması gereklidir.4.11.3  
Fonksiyonel Gereksinimler



Sistem yöneticisi seçtiği dersi butona bastığında silmeli, işlem sonucunda doğrulama mesajı ya da başarısızsa hata mesajı almalıdır.

<b>ID:</b>	#2
<b>Title:</b>	Ders Ekle
<b>Description:</b>	Yeni bir ders oluştur
<b>Primary Actor:</b>	Sistem Yöneticisi
<b>Preconditions:</b>	Ders eklenecek şubeyi seç. Ders dilini ve bilgilerini seç
<b>Postconditions:</b>	---
<b>Main Success Scenario:</b>	Belirlenen şubede yeni bir ders oluşturulur.
<b>Extensions:</b>	---
<b>Frequency of Use:</b>	Yılda birkaç kez
<b>Status:</b>	Finished

<b>Owner:</b>	Serdar ORMANCI
<b>Priority:</b>	Yüksek

## 4.12 Yeni Kur Ekleme

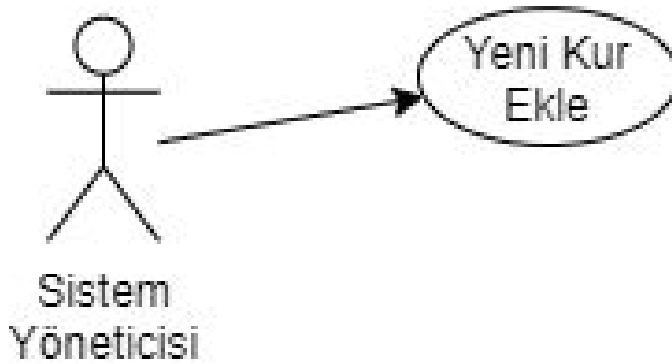
### 4.12.1 Tanım

Sisteme yeni bir kur eklenmesini sağlayan modüldür. Bu modülü yalnızca Sistem Yöneticisi kullanabilir.

### 4.12.2 Uyarılar

Aynı şubedeki aynı kursun aynı anda birden fazla aynı kuru mevcut olamaz. Kurun, daha önceden ekleme yapılacak aynı şubedeki aynı kurstaki kurlardan farklı bir kur olması gerekmektedir.

### 4.12.3 Fonksiyonel Gereksinimler



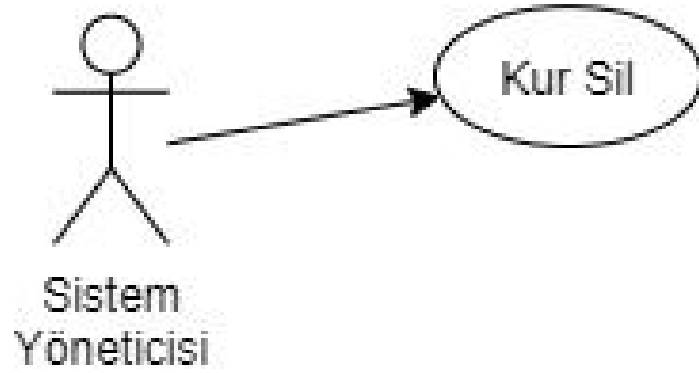
Sisteme, şube bilgileri getirilmelidir

Sisteme, seçilen şubenin seçilen kursunun bilgileri getirilmelidir.

## 4.13 Kur Silme

### 4.13.1 Tanım

Sistemin, mevcut bir kuru sistemden silmesini sağlayan modüldür. Bu modülü yalnızca Sistem Yöneticisi kullanabilir.



Use Case Adı:	Öğrenci Kaydı
Aktörler:	Kayıt Elemanı(Birincil), Müşteri(İkincil)
Tanım:	Dil Kursu Otomasyonu Sistemi üzerinden Kayıt Elemanının bir öğrenciyi derse kaydetmesinden bahsedilmiştir.
Ön Koşullar:	Kayıt Elemanı, doğru kullanıcı adı ve şifre bilgilerini girerek sisteme giriş yapmalıdır.
Son Koşullar:	Öğrenci başarıyla hem sisteme hem de dil kursunun kendi istediği bir şubesine kayıt edilmiş olacaktır.
Ana Başarı Senaryosu:	<ol style="list-style-type: none"> <li>1. Kayıt Elemanı "İşlemler" menüsünden "Öğrenci Kaydet" butonuna tıklar</li> <li>2. Kayıt Elemanı kayıt edeceği öğrencinin ismini girer</li> <li>3. Kayıt Elemanı kayıt edeceği öğrencisinin soyadını girer</li> <li>4. Kayıt Elemanı kayıt edeceği öğrencisinin e-posta adresini girer</li> <li>5. Kayıt Elemanı daha sonradan öğrencinin sisteme giriş yapabilmesi için öğrencinin belirleyeceği bir şifreyi girer</li> <li>6. Kayıt Elemanı öğrencinin ev telefonunu girer</li> <li>7. Kayıt Elemanı öğrencinin cep telefonunu girer</li> <li>8. Kayıt Elemanı öğrencinin Kimlik No'sunu girer</li> <li>9. Kayıt Elemanı "Kaydet" butonuna basıp öğrenciyi sisteme kaydeder</li> <li>10. Kayıt Elemanı "İşlemler" Menüsünden "Kurs Kaydı" Butonuna tıklar</li> <li>11. Kayıt Elemanı "Şube Seçiniz" alanından öğrencinin kaydedileceği şubeyi seçer</li> <li>12. Kayıt Elemanı "Kurs Seçiniz" alanından öğrencinin kaydedileceği kursu seçer</li> <li>13. Kayıt Elemanı "Kurs Seçiniz" alanından öğrencinin kaydedileceği kuru seçer</li> <li>14. Kayıt Elemanı "İlerle" butonuna basar</li> <li>15. Ödemeyi yapacak olan kişi kredi kartı bilgilerini girer.</li> <li>16. Ödemeyi yapacak olan kişi "Peşin" ya da "Taksitli" ödeme seçeneklerinden birini seçer</li> <li>17. Ödemeyi yapacak olan kişi "Ödemeyi Yap" butonuna tıklar</li> </ol>
Alternatif Senaryolar:	<ol style="list-style-type: none"> <li>2a. Kayıt Elemanı öğrencinin ismini boş bırakır <ol style="list-style-type: none"> <li>1. Sistem "Gerekli alanları boş bırakmayınız" uyarı mesajı verir.</li> </ol> </li> <li>3a. Kayıt Elemanı öğrencinin soyadını boş bırakır <ol style="list-style-type: none"> <li>1. Sistem "Gerekli alanları boş bırakmayınız" uyarı mesajı verir.</li> </ol> </li> <li>4a. Kayıt Elemanı öğrencinin e-posta alanını yanlış formatta girer <ol style="list-style-type: none"> <li>1. Sistem hata mesajı verir</li> </ol> </li> <li>4b. Girilen eposta sistemdeki başka bir eposta ile eşleşir <ol style="list-style-type: none"> <li>1. Sistem "bu e posta zaten kayıtlı" mesajı verir.</li> </ol> </li> <li>5a. Kayıt elemanı öğrencinin şifre alanını boş bırakır. <ol style="list-style-type: none"> <li>1. Sistem "Gerekli alanları boş bırakmayınız" uyarı mesajı verir.</li> </ol> </li> <li>15a. Ödemeyi yapacak olan kişi yanlış kredi kartı bilgileri girer. <ol style="list-style-type: none"> <li>1. Sistem ödeme işlemini gerçekleştirmez</li> <li>2. Sistem "Yanlış kart bilgileri" mesajı verir</li> </ol> </li> </ol>



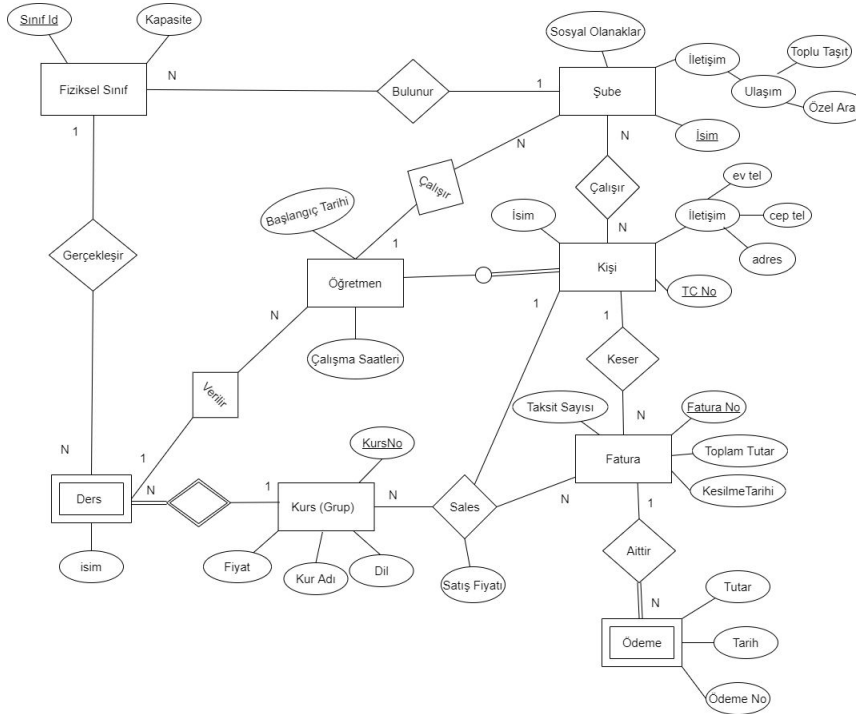
## 5- Fonksiyonel Olmayan Diğer Gereksinimler

### 5.1 Performans Gereksinimleri

Uygulama üzerinde yaklaşık 1000 kullanıcı eş zamanlı olarak işlemlerini yapabilmektedir. Sistem yöneticileri eş zamanlı olarak sistemde düzenlemeler, eklemeler ve silmeler yapabilmektedir. Sistem üzerinde yapılan değişiklikler, sistemde bulunan kullanıcılara anlık olarak yansıtılır. Bunun için sunucuların saniyede ortalama 100 işlem kapasitesine ve tepki hızı 500 ms' yi aşmayacak özellikte olmalıdır. Ana hafıza olarak minimum 2 GB kapasite öngörülen işlem hacmini kaldırmak için yeterli olacağı belirlenmiştir.

### 5.2 Mantıksal Veritabanı Gereksinimleri

Veritabanında tutulması gereken tablolar ve bu tabloları arasındaki ilişkiler aşağıdaki ER diyagramında gösterilmiştir. Sistem tasarlanırken karşılaşılan tüm kısıtlar ifade edilmiştir. Örneğin 1 öğretmen 1 den fazla şubede çalışabilir. 1 sınıfta 1'den fazla ders işlenebilir. Aynı ismi taşıyan farklı kurslara ait dersler olabilir gibi.



## 5.3 Kalite Özellikleri

Sistemin kalitesini belirlerken güvenilirlik, kullanılabilirlik, güvenlik, erişilebilirlik ve kullanılabilirlik gibi metrikler üzerinden değerlendirilmiştir. Aşağıda bu metriklerin sistemde nasıl uygulandığı anlatılmıştır.

### 5.3.1 Güvenilirlik

Sistemin sağladığı bilgilere erişim istenilen zamanda mümkün olmalıdır. Kayıt ve ödeme sırasında sistemde hata ve çökme meydana gelmemesi gereklidir. Gerçek zamanlı işlemlerde yaşanacak aksamanın yarattığı ticari risk düşük olduğu için çökme sırasında ayağa kaldırmak üzere yedek bir sunucu atanmasına ihtiyaç yoktur. Sunucuların çökmesi durumunda sunucunun yeniden başlatılma süresinin olabildiğince kısa olması faydalıdır. Ödeme modülü güvenli ödeme şirketlerinin altyapısı kullanılarak sağlanmalı ve bu durum kullanıcıya iletilmelidir.

### 5.3.2 Kullanılabilirlik

Sistem tasarlandıktan sonra yazılım tüm kullanıcı rolleri tarafından kullanıma hazır olmalı ve tüm yazılım testleri tamamlanmış olmalıdır.

### 5.3.3 Güvenlik

Sistemin güvenliği oldukça önemlidir. Depolanan veriler kullanıcılara ait özel bilgileri içerdiği için sunucular güvenliği ispatlanmış şirketlere ait olmalıdır. Sisteme karşı yapılacak saldırılara bağışıklık olmalı ve bu tür ataklar sırasında veri sızdırılması, veri kaybı asla olmamalıdır. Ödeme sırasında 3D secure gibi altyapılar kullanılmalı ve veri trafiği https protokolüne uygun biçimde şifreli olarak aktarılmalıdır.

### 5.3.4 Taşınabilirlik

Yazılım ihtiyaç duyulduğunda farklı platformlara uyarlanabilmelidir. Başlangıçta mobil platform hedef alınarak en çok kullanıcı kitlesine sahip olan Android cihazlara uyumlu yazılım geliştirilebilir.

### 5.3.5 Kullanılabilirlik

Sistemdeki modüller basit ve kolayca anlaşılır yapıda olmalıdır. Modüller gerekli alt ekranlara ayrılarak kullanım kolaylığı sağlanmalıdır.