1.GİRİŞ

1.1 Amaç Ve Kapsam

Projemizin amacı: ÇOMÜ öğrencilerinin kullanabileceği bir sosyal medya platformu oluşturmak. Bu platformda öğrenciler internet üzerinde sosyalleşebilecek, arkadaş olabilecek, fotoğraf ve konum paylaşabilecek, mekanlara yorum yapabilecek, etkinlik ekleyebilecek ve etkinliklere katılabilecek.

Projemizin Kapsamı: Çanakkale'de okuyan üniversite öğrencileri

1.2 Hedefler Ve Başarı Kriterleri

Hedefler:

* ÇOMÜ öğrencilerinin internet üzerinden sosyalleşebilmesini sağlamak
* ÇOMÜ öğrencilerinin kampüsteki etkinliklerden haberdar olabilmesini sağlamak
* ÇOMÜ öğrencilerinin kampüsü ve birbirlerini daha yakından tanıyabilmesini sağlamak

Başarı Kriterleri:

* Öğrenciler sorunsuz bir şekilde sisteme üye olmalıdırlar ve profil sayfalarını başarılı bir şekilde oluşturabilmelidirler.
* Durum, konum veya fotoğraf paylaşmak isteyen öğrenciler başarılı bir şekilde işlemlerini gerçekleştirebilmelidirler.
* Etkinlik eklemek isteyen kullanıcılar başarılı bir şekilde etkinlik ekleyebilmelidirler.
* Mekanlara yorum yapmak isteyen kullanıcılar başarılı bir şekilde yorum yapabilmelidirler.
* Arkadaş olmak isteyen kullanıcılar birbirleri ile arkadaş olabilmelidirler.

1.3 Genel Bakış

Uygulamamız ÇOMÜ öğrencilerinin internet üzerinden sosyalleşebilmelerini sağlamak için düşünülmüştür. Uygulamamızda öğrenciler kampüsteki etkinliklerden haberdar olabilecek, mekanlarda bulunan insanları görebilecek, mekanlara yorum yapabilecek, birbirleri ile arkadaş olabilecektir.

Sistem Gereksinimleri

* Sistem meydana gelecek aksaklıkları gidermelidir.
* Kurum ihtiyaçlarını analiz etmelidir.
* Sistem kurallarına uymayan üyeleri sistemden çıkarmalıdır.
* Sunucu tarafında yapılacak işleri gerçekleştirir.

Üye Gereksinimleri

* Üyeler sisteme giriş yapmalıdır.
* Üyeler profillerini oluşturabilmelidir.
* Üyeler birbirleri ile arkadaş olabilmelidir.
* Üyeler birbirleri ile mesajlaşabilmelidir.
* Üyeler fotoğraf, durum ve konum paylaşabilmelidir.
* Üyeler mekanları görebilmeli ve onlara yorum yapabilmelidirler.
* Üyeler başka üyelerin paylaştıkları içerikleri beğenebilmeli ve yorum yapabilmelidir.
* Üyeler etkinlik ekleyebilmelidirler.
* Üyeler etkinlikleri görüntüleyebilmeli ve katılım durumlarını belirtebilmelidir.

2.Fonksiyonel Gereksinimler

* Bu sistem adı, soyadı, doğum tarihi, mail adresi, okunulan bölüm bilgisi ve okul numarası ve belirlenen parola ile sisteme kayıt yapmaya imkan sağlamalıdır.
* Sisteme üye olan birinin kayıt olurken belirlediği parola ve kullanıcı adı ile sisteme giriş yapmasına imkan sağlamalıdır.
* Üyenin profil sayfası oluşturmasına ve gerektiğinde bilgilerini güncellemesine imkan sağlamalıdır.
* Üyenin siteye içerik eklemesine imkan sağlamalıdır.
* Üyenin siteye yüklediği içeriğin detaylandırmasına imkan sağlamalıdır.
* Üyenin diğer üyelerin paylaşımlarını görüntülemesine ve yorum yapabilmesine imkan sağlamalıdır.
* Üyenin başka üyelerle arkadaş olabilmesine imkan sağlamalıdır.
* Üyenin mekanlara ve etkinliklere yorum yapabilmesi sağlanmalıdır.
* Üyenin mekanlara puan vermesine imkan sağlamalıdır.
* Üyenin mekanlara ve etkinliklere yaptığı yorumların herkes tarafından görülmesine olanak sağlamalıdır.
* Üyenin diğer üyelere mesaj gönderebilmesini sağlamalıdır.
* Üyenin diğer profil sayfalarını görmesine imkan sağlamalıdır.
* Üyelerin sistem veya diğer üyelerle ilgili şikayet ve isteklerini bildirmesine imkan sağlamalıdır.
* Üyenin güvenli bir şekilde çıkış yapmasına olanak sağlamalıdır.

3.Fonksiyonel Olmayan Gereksinimler

3.1.Security ( Güvenlik ) :

● Sisteme sadece sistem yöneticisi tarafından erişilebilmelidir.

● Şifre politikası uygulanmalıdır. (Minimum bir büyük, küçük harf ve bir sayı.)

● Üye bilgileri kimse ile paylaşılmamalıdır.

● Sistem bakımını sadece sistem yöneticisi yapmalıdır.

● Yetkili kişiler sistemi güvenli bir şekilde kullanmalıdır.

3.2.Usability ( Kullanılabilirlik ) :

● Sistem her kullanıcı türü için anlaşılır, sade ve kullanılabilir olmalıdır.

● Sistemde yapılmak istenen işlemler en hızlı şekilde yapılmalıdır.

3.3.Performance ( Performans ) :

● Sistem, kapatılana kadar sorunsuzca çalışmalıdır.

● Üyelerin sistemde yapacakları işlemler kısa süre içerisinde olmalıdır.

● Sistemin aynı anda birden fazla kullanıcıyı kaldırma kapasitesi olmalıdır.

3.4.Supportability ( Bakım ) :

● Sitemin bakımı üyelerin en az işlem yaptığı vakitlerde yapılmalıdır.

● Sistem bakımı yapılırken oluşabilecek sorunlara karşı, sistemde giriş yapmış üyeler uyarılarak çıkışları yapılmalıdır.

● Bakım yapılacağına yakın mağduriyet olmaması için daha önceden üyelere haber verilmelidir.

● Sisteme yeni donanım eklendiğinde uyumluluk sağlanmalıdır.

3.5.Constraints ( Sistem Kısıtlamaları ) :

● Sistemde bulunması gereken belgeler ve sertifikalar eksiksiz bir şekilde sistem yöneticisinde bulunmalıdır.

● Belli zamanlarda sistem istatistiği tutulmalıdır.

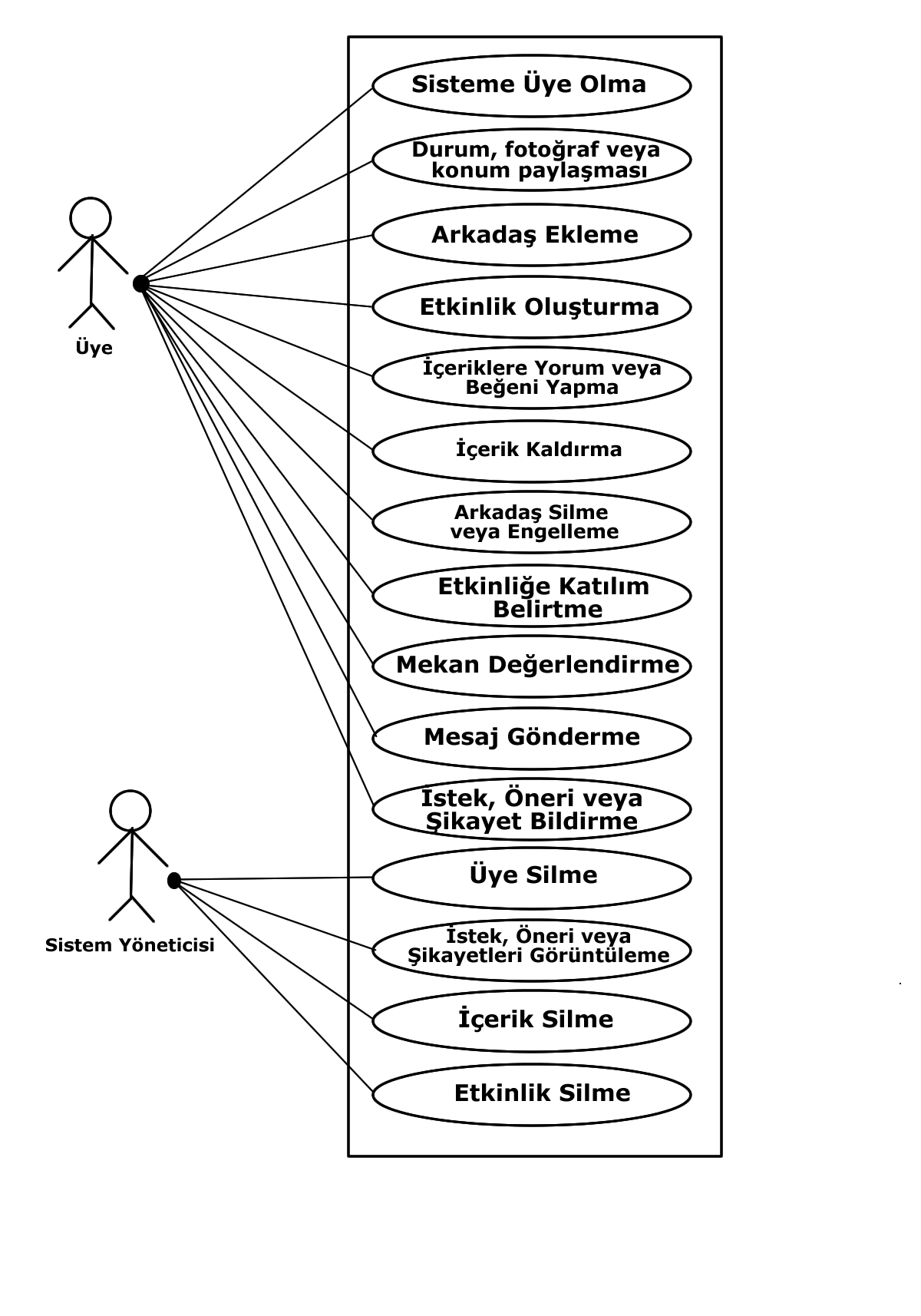
● Sistemi kullanmak için mutlaka üyelerin interneti olmalıdır.

● Üyeler sistem kurallarına uymalıdır.

● Sistem kurallarına uymayan üyeler yöneticiler tarafından sistemden uzaklaştırılmalıdır.

4. Aktor Goal Model

| **Actor** | **Goal** |
| --- | --- |
| Üye | Sisteme Üye Olma  Durum, fotoğraf veya konum paylaşması  Arkadaş Ekleme  Etkinlik Oluşturma  İçeriklere Yorum veya Beğeni Yapma  İçerik Kaldırma  Arkadaş Silme veya Engelleme  Etkinliğe Katılım Belirtme  Mekan Değerlendirme  Mesaj Gönderme  İstek, Öneri veya Şikayet Bildirme |
| Sistem Yöneticisi | Üye Silme  İstek, Öneri veya Şikayetleri Görüntüleme  İçerik Silme  Etkinlik Silme |



5. En Kritik 2 Use Case

USE CASE 1: Üyenin durum, fotoğraf veya konum paylaşması

Scope: Sosyal Kampüs

Level: User Goal

Primary Actor: Üye

Stekholders and Interest: Üye sisteme hızlı ve başarılı bir şekilde içerik eklemek ister.

Precondition:Üye sisteme kayıtlı ve giriş yapmış olmalıdır.

Postcondition:Üye sisteme yeni bir durum eklemiş olmalıdır.

Ana Başarı Senaryosu

1.İçerik ekleme işlemi başlatılır.

2.Sistem üyenin fotoğraf, durum veya konum paylaşımlarından birini seçmesini ister

3.Üye sistemin içerik ile ilgili istediği bilgileri eksiksiz bir şekilde sisteme girer ve kaydeder.

4.Sistem girilen bilgileri üyeye gösterir.

5.Üye içerik bilgilerini onaylar.

6.Sistem içerik bilgilerini kaydeder ve yayınlar.

Alternatif Senaryolar

\*a.Sistem herhangi bir anda çökerse,

1.Sistem yeniden başlatılır.

2.Üye işleme 2. adımdan devam eder.

3.a.Üye ürün bilgilerini eksik girerse,

1.Sistem tarafından üyeye hata mesajı gönderilir.

2.Üye eksik bilgileri tamamlar ve ana başarı senaryosu 4. Adımdan devam eder.

5.a Üye içerik bilgilerini onaylamazsa,

1. Üye işleme 3. adımdan devam eder.

USE CASE 2:Üyenin arkadaş eklemesi

Scope: Sosyal Kampüs

Level:User Goal

Primary Actor:Üye

Stekholders and Interest: Üye başka üyelere arkadaş olma isteği gönderebilmelidir.

Precondition:Üye sisteme kayıtlı ve giriş yapmış olmalıdır.

Postcondition:Üye başarılı bir şekilde istek gönderebilmiş olmalıdır.

Ana Başarı Senaryosu

1.Arkadaş ekleme işlemi başlatılır.

2.Sistem arkadaş ekleme sayfasını ve üyenin tanıyor olabileceği üyeleri görüntüler.

3.Üye arkadaş olarak eklemek istediği üyenin bilgilerini girerek arama işlemini gerçekleştir.

4.Sistem bilgilere uygun kullanıcıları listeler.

5.Üye eklemek istediği üyeyi seçer.

6.Sistem seçilen üyenin bilgilerini gösterir.

7.Üye arkadaşlık isteğinin gönderilmesini onaylar.

8.Sistem arkadaşlık isteğini diğer üyeye göndererek sisteme kaydeder.

Alternatif Senaryolar

\*a.Sistem herhangi bir anda çökerse,

1.Sistem yeniden başlatılır.

2.Üye işleme 2. adımdan devam eder.

7.a.Üye arkadaşlık isteğinin gönderilmesini onaylamazsa,

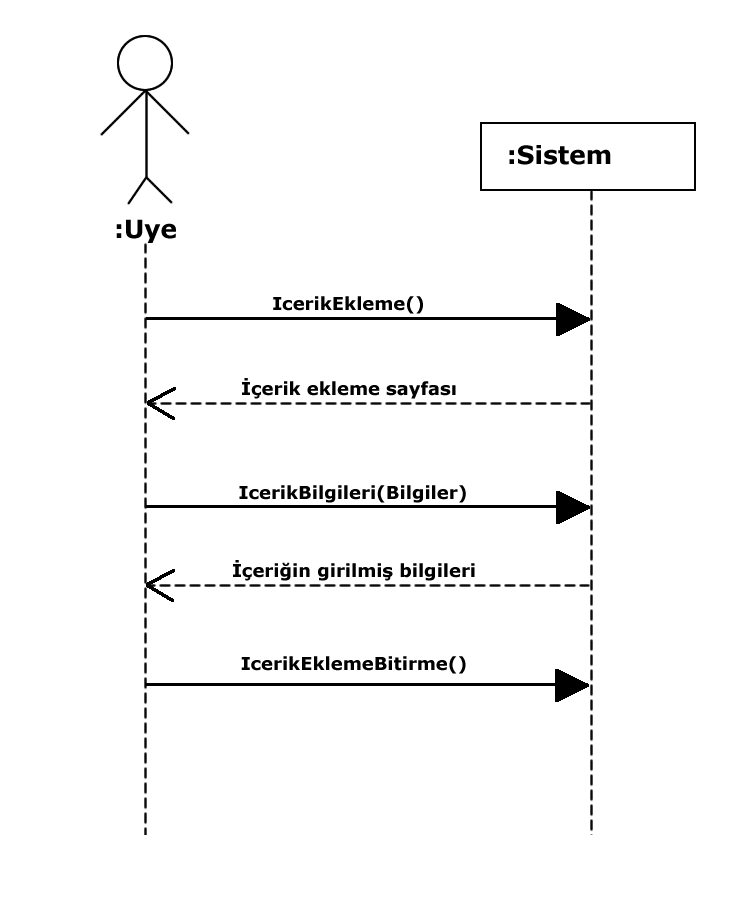
1.Üye diğer ürünleri görmek ister.

2.Ana başarı senaryosunun 2.adımından devam eder.

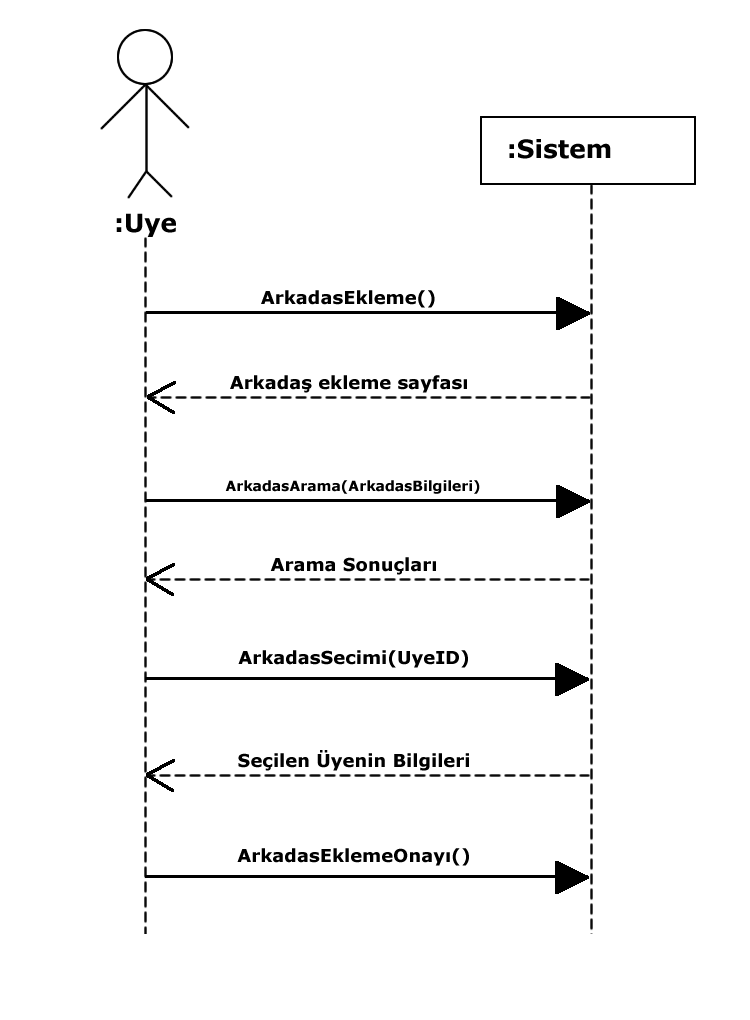
2.Üye işleme 2. adımdan devam eder.

6. SSD Diagram

6.1 SSD Diagram Use Case 1

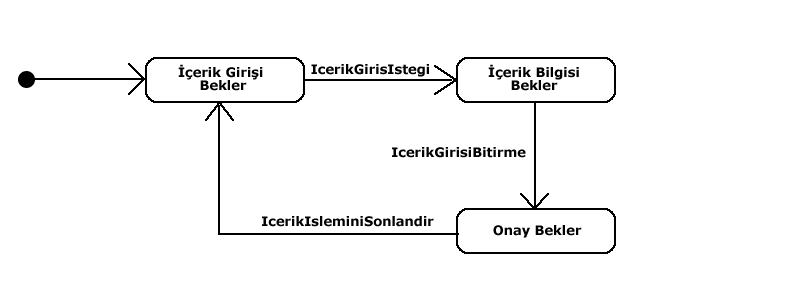


6.2 SSD Diagram Use Case 2

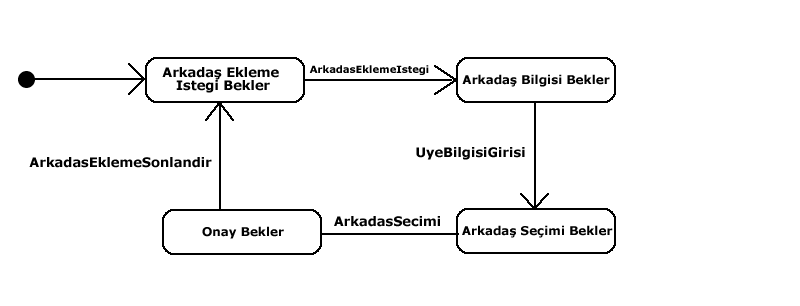


7. State Machine Diagram

7.1 State Machine Diagram Use Case 1

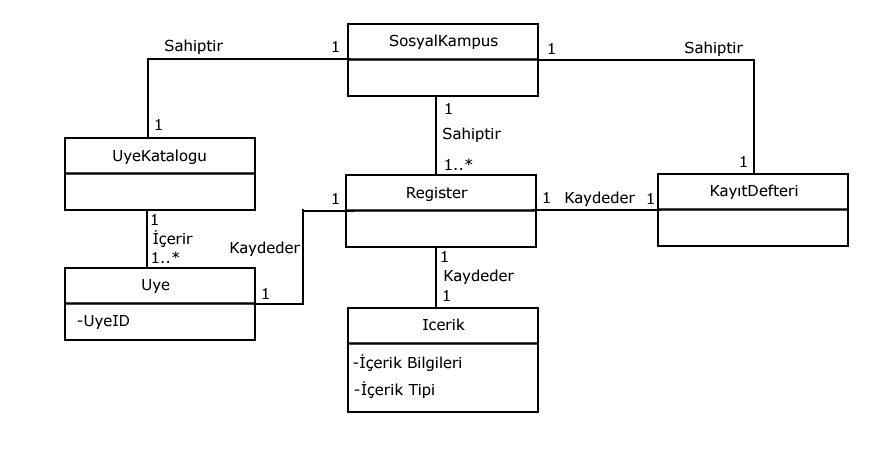


7.2 State Machine Diagram Use Case 2

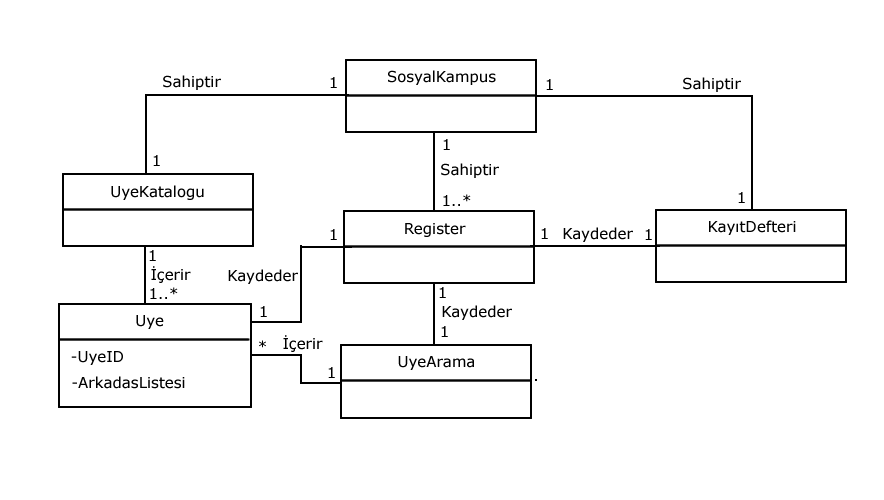


8. Domain Model

8.1 Domain Model Use Case 1



8.2 Domain Model Use Case 2



9. Domain Model

9.1 Operation Contracts Use Case 1

**Contract OC1**

**Name:** IcerikEkleme()

**Cross References:** Use Cases: Üyenin durum, fotoğraf veya konum paylaşması

**Precondition:** none

**Postcondition:**

* Icerik sınıfının YeniIcerik adında bir nesnesi oluşturulmuş olmalıdır.
* YeniIcerik nesnesi Register nesnesi ile ilişkilendirilmiş olmalıdır.

**Contract OC2**

**Name:** IcerikBilgileri(Bilgiler)

**Cross References:** Use Cases: Üyenin durum, fotoğraf veya konum paylaşması

**Precondition:** Devam eden bir Icerik nesnesi var olmalıdır.

**Postcondition:**

* Devam etmekte olan YeniIcerik nesnesinin Icerik Bilgileri özelliği parametre olarak gelen Bilgiler ile set edilmiş olmalıdır.

**Contract OC3**

**Name:** IcerikEklemeBitirme()

**Cross References:** Use Cases: Üyenin durum, fotoğraf veya konum paylaşması

**Precondition:**

* Devam eden bir YeniIcerik nesnesi var olmalıdır.

**Postcondition:**

* Devam etmekte olan YeniIcerik nesnesi Kayıt Defteri nesnesi ile ilişkilendirilmiş olmalıdır.

9.2 Operation Contracts Use Case 2

**Contract OC1**

**Name:** ArkadasEkleme()

**Cross References:** Use Cases: Üyenin arkadaş eklemesi

**Precondition:** none

**Postcondition:**

* UyeArama sınıfının YeniUyeArama adında bir nesnesi oluşturulmuş olmalıdır.
* YeniUyeArama nesnesi Register nesnesi ile ilişkilendirilmiş olmalıdır.

**Contract OC2**

**Name:** ArkadasArama(ArkadasBilgileri)

**Cross References:** Use Cases: Üyenin arkadaş eklemesi

**Precondition:**

* Devam eden bir YeniUyeArama nesnesi var olmalıdır.
* Devam eden bir Uye nesnesi var olmalıdır.

**Postcondition:**

* Devam etmekte olan YeniUyeArama nesnesinin UyeID özelliği parametre olarak gelen Bilgiler'in belirttiği Uye nesneleri ile ilişkilendirilmiş olmalıdır.

**Contract OC3**

**Name:** ArkadasSecimi(UyeID)

**Cross References:** Use Cases: Üyenin arkadaş eklemesi

**Precondition:**

* Devam eden bir Uye nesnesi var olmalıdır.

**Postcondition:**

* Devam etmekte olan Uye nesnesinin ArkadasListesi özelliği parametre olarak gelen UyeID ile set edilmiş olmalıdır.

**Contract OC4**

**Name:** ArkadasEklemeOnayi()

**Cross References:** Use Cases: Üyenin arkadaş eklemesi

**Precondition:**

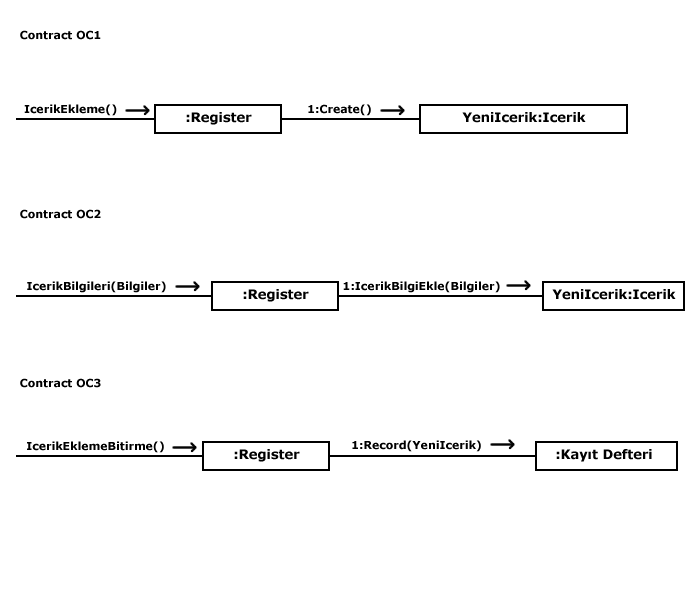
* Devam eden bir Uye nesnesi var olmalıdır.

**Postcondition:**

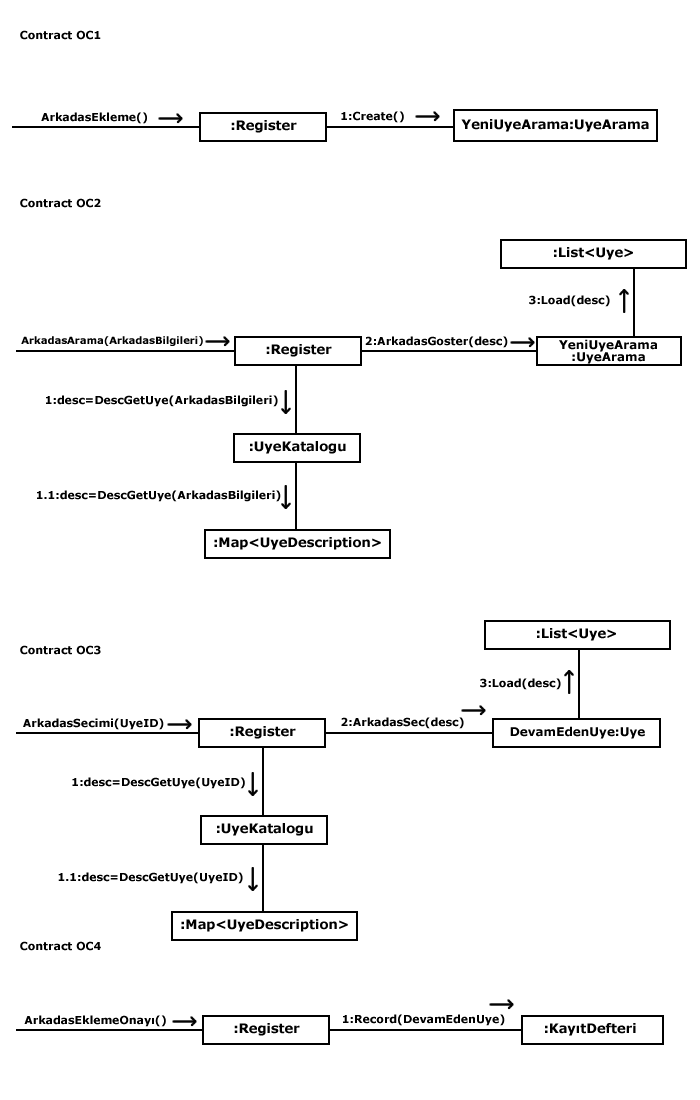
* Devam etmekte olan Uye nesnesi Kayıt Defteri nesnesi ile ilişkilendirilmiş olmalıdır.

10. Interaction Diagram

10.1 Interaction Diagram Use Case 1

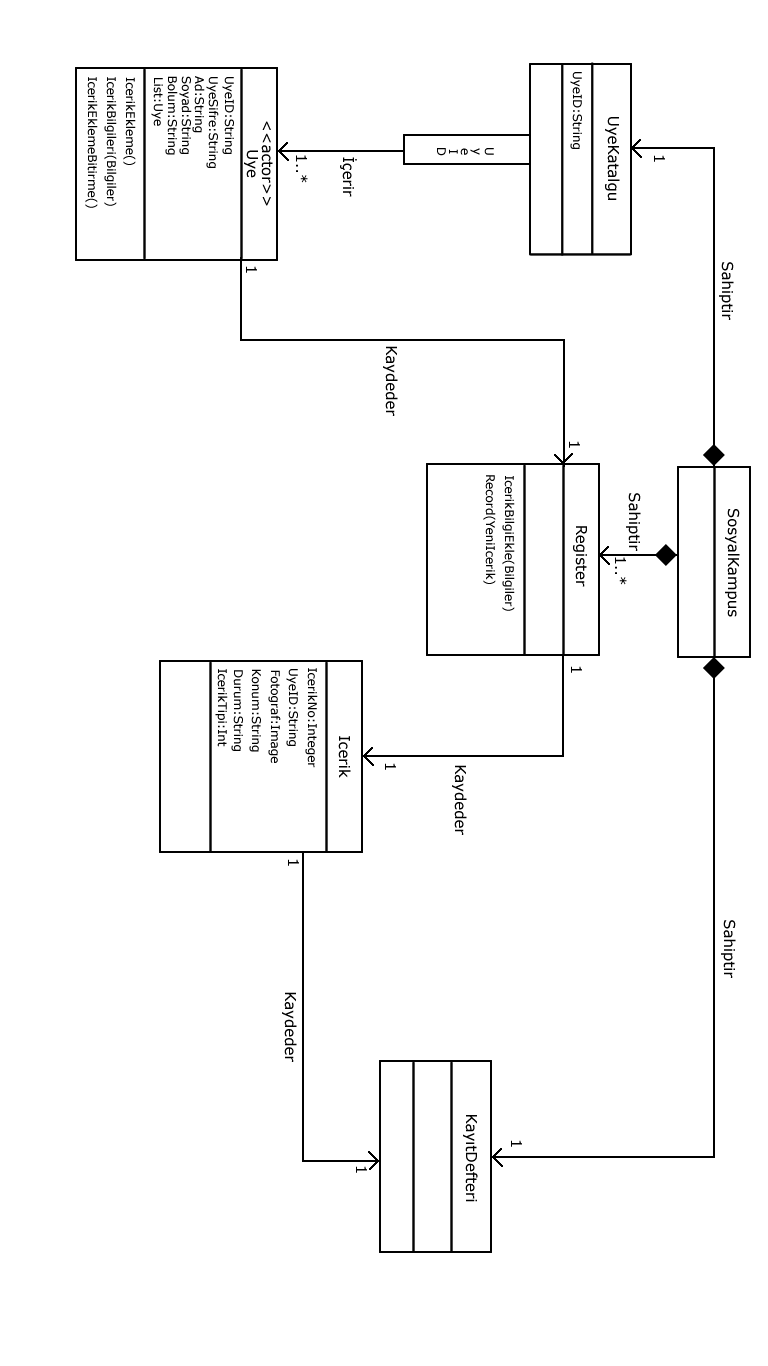


10.2 Interaction Diagram Use Case 2



11. Class Diagram

11.1 Class Diagram Use Case 1



11.2 Class Diagram Use Case 2

