

1 GİRİŞ

Günümüzde gerek yazılım, gerekse donanım projeleri geliştirilme aşamasından kullanıcıya gidene kadar birçok aşamadan geçmekte ve birçok kişi tarafından test edilip kullanılmaktadır. Gerek maliyetlerin artması gerekse kullanıcı ve geliştirici sayısının artması yüzünden projelerde bulunan eksik veya hatalı kısımlar hem para hem de zaman kaybına sebep olarak projelerden sorumlu kişilerin çok fazla zaman kaybetmesine, hem de firmalarının çok para kaybetmesine sebep olmaktadır. Her iki tarafın da başının ağrıması ve hata iletmek ile gidermek arasındaki sürenin minimuma indirilmesi için birçok çalışmalar yapılmaktadır. Bunlardan biri de eksik veya yanlış çalışan kısımların bildiriminin kolaylaştırılması için gerçekleştirilen internet tabanlı uygulamalardır. İnternet gittikçe gelişen ve neredeyse hayatın her yerinden erişilebilen bir teknoloji olduğu için en kısa zamanda en kolayca bilgi aktarımı yapılabildiği için iletim yöntemlerinden en ucuzu ve en etkini olarak seçilmektedir. Örneğin gelişen teknolojiler sayesinde kişiler el bilgisayarları veya kişisel cep telefonları ile internete ulaşabilmekte böylece neredeyse zaman ve yer bağımsız olarak internet uygulamalarına erişebilir duruma gelmişlerdir. Aynı şekilde artık kişisel bilgisayarların her ofiste 7/24 bağlı oldukları internet sağlayıcı teknolojilerin (xDSL, ISDN, Frame Relay vb.) ucuzlayarak hayatta bir gereksinim haline dönüşmeleri interneti neredeyse kusursuz bir bilgi iletim merkezi haline getirmiştir.

Bu proje; günümüzde büyük bir sorun olan hata takibinin hem hata gidermekle yükümlü kişiler, hem de hatayı iletecek kişiler için mümkün olduğunca kolay ve hızlı şekilde yapılabilmesini sağlamak amacıyla geliştirilmek istenmiştir. Hata takibi konusunda en büyük sorun, hatayı bulup ileten kişi ile hatayı düzeltmek ile sorumlu kişinin arasındaki iletişim kopukluğudur. İletici kendince hatayı tanımladığını ve tam olarak ifade ettiğini düşünse de genelde yetersiz bilgi verilmesi, yeterince açık ifade edilmemiş ayrıntılar yüzünden zaman kaybı yaşanmakta ve anlaşmazlıklar oluşmaktadır. Bu sorunun ortadan kaldırılarak, bir nevi iki tarafın da “aynı” dili konuşmasının sağlanması ve bilgilerin eksiksiz sunulması zaman kaybını en aza indirmek için bir zorunluluktur. İşte bu ortak dilin konuşulabilmesi için yapılan bu proje ile hem eksiksiz bilgi aktarımı hem de ileticiden en fazla ayrıntıyı sorumlunun anlayabileceği şekilde alma işlemi kolay ve hızlı bir şekilde sağlanmak istemiştir.

Hata takip ve raporlama alanında daha çok sadece yazılım projeleri için web tabanlı uygulamalar geliştirilmiştir. Genelde bu uygulamalar akıllı olmaktan ziyade sadece veritabanı uygulaması veya forum tarzı uygulamalar olarak tasarlanmışlardır. Örneğin “FogBugz: Akıllı insanlar için proje yönetimi” projesi bu konuda en ünlü yazılımlardan biridir.[1] Bu projenin en can alıcı noktası hata oluşturma adımlarının sorumluya bildirilmesi zorunluluğudur. Çünkü iletilen kişi her ne kadar hatayı anlatsa da sorumlu bu hatayı oluşturmamış süreçte herhangi bir düzeltme işlemi yapamamaktadır.

Açık kaynak kodlu bir başka proje ise “Mantis arıza takibi sistemi” dir.[2] Bu proje de PHP-MySQL ile geliştirilmiştir GPL lisansı ile dağıtılmakta ve web sitesinden indirilebilmektedir.

Bir diğer proje ise Bugzilla’dır. Mozilla’nın eksik kısımlarının takibi ile ünlenmiş bu yazılımı kullanan firma ve organizasyonların listesi için bkz: [3]

Bu projedeki çalışma ile mümkün olduğunca her yerden hata gönderme imkânı sağlanmış, hataların takibi ve yeni bir hatanın gelmesi durumunda bilgilendirme imkânı en üst seviyede tutulmuştur. Düz bir forum veya mesaj tahtası şeklindeki bir sistemden ziyade sadece hata takibi ve raporlama alanında özelleşmiş yazılım parçaları her tür proje için modüler bir şekilde gerçeklenmeye çalışılmıştır.

Bu raporda; projenin başından itibaren tanımlanması ve planlanması aşaması bölüm 2’de, projenin gerçeklenmesi ile ilgili teknolojilerin kısa bir açıklaması bölüm 3’te, sorunun çözümü için oluşturulan analiz ve modelleme adımları bölüm 4’te, projenin gerçeklenmesi için tasarım, gerçekleştirme ve test adımları bölüm 5’te, proje üzerindeki deneysel sonuçlar bölüm 6’ta ve proje hakkında sonuçlar, öneriler bölüm 7’de anlatılmıştır.

2 PROJE TANIMI VE PLANI

Projede ana amaç, sistemi kullanacak herkese mümkün olan maksimum kolaylığı ve yardımı sağlayabilmektir. Bunun sağlanabilmesi için hata giderici ile hata iletici arasında ortak bir iletişim kanalı tasarlanmanın mecburi olduğu görülmüştür. Her zaman bir kişinin anlatmak isteyip de anlatabildiği aynı olmamakla birlikte, anlatılabilenin de anlaşılma seviyesi arasında uçurumlar olması sebebiyle; iki tarafın da standart bir iletişim kanalından anlaşması sağlanmak istenmiştir. Hataların takip edilebilmesi için doğal olarak sisteme eklenmeleri gereklidir, bundan dolayı hem hata ileticiler hem de hata gidericiler için sisteme bilgi yüklenmesi sırasında kolaylık sağlanmasına da çalışılmış, zaman kazandırıcı bir arayüz ve alternatifler ile kullanıcıların en kısa sürede en doğru şekilde bilgi aktarımı yapması sağlanmak istenmiştir.

Tarafların gerektiğinde mobil bir araçtan (PDA, cep telefonu) sisteme erişebilmeleri de hata iletme için platforma bağımlılık azaltılarak, ciddi ve acil durumlar için erişilebilirlik sorunu mümkün olduğunca ortadan kaldırılmak istenmiştir. Örneğin herhangi bir bilgisayarın veya internet bağlantısının olmadığı elverişsiz şartlarda çalışan bir makinenin test edilmesinde ve test sonuçlarının merkeze bildirilmesi için bu akıllı cihazlardan birinden faydalanılabilir. Bir başka örnek vermek gerekirse tatilde olan bir sorumlu, cep telefonu vasıtasıyla kritik bir proje ile ilgili çözülmemiş hata sayısını kontrol edebilir ve böylece işlerin yürütüldüğünden emin olabilir.

Herhangi bir proje hakkında herhangi bir hata veya arızasının iletilebilmesi ve bu yapılırken sorumlu kişiler tarafından belirlenmiş belli başlı soruların cevaplanarak hata hakkında daha ayrıntılı bilgi alınabilmesi sağlanarak, kişinin anlatımından bağımsız olarak hatanın nerede ve nereyle ilgili olduğu, soruları belirlemiş kişi tarafından tespit edilebilmesi kolaylaştırılmak istenmiştir. Sadece yazılım projeleri için değil, istenilen herhangi bir proje için hata takibi yapılabilmesi için gerekli esneklik mevcuttur.

Proje, üç ana uygulamadan oluşmaktadır: Web tabanlı ana ASP.NET web uygulaması, Windows Mobile 5.0 akıllı cihaz (PDA) uygulaması ve Java 2 Micro Edition cep telefonu (GSM) uygulaması. Ana uygulama tüm bilgileri SQL Server 2005 Express Edition veritabanında tutmakta ve diğer iki uygulamanın da web servisleri ile sisteme erişmesine izin vermektedir.

Projedeki modüller şunlardır:

- Veritabanı tabloları ve saklı yordamlar
- Veritabanı tercüman sınıfı
- İş katmanı sınıfları
- Hata giderici için web-tabanlı uygulamalar
- Hata iletici için web-tabanlı uygulamalar
- Akıllı cihazların erişimi için gerekli web servisleri
- PDA uygulaması
- Cep telefonu uygulaması

Tablo 2.1’de projenin zaman çizelgesi için belirlenmiş işlerin listesi verilmiştir. Şekil 2.1’de ise proje ile ilgili zaman çizelgesi (Gantt Chart) verilmiştir.

Tablo 2.1: Zaman Çizelgesi iş tablosu

İş 1	Yazılımın modellenmesi ve tasarımı
İş 2	.NET ve J2ME eğitimi
İş 3	Hata giderici web-tabanlı uygulama kodlaması
İş 4	Hata iletici web-tabanlı uygulama kodlaması
İş 5	Cep telefonu ve PDA eklentileri
İş 6	Yazılım testleri
İş 7	Dokümantasyon

	6 Şubat	13 Şubat	20 Şubat	27 Şubat	6 Mart	13 Mart	20 Mart	27 Mart	3 Nisan	10 Nisan	17 Nisan	24 Nisan	1 Mayıs	8 Mayıs	15 Mayıs	22 Mayıs
İş 1																
İş 2																
İş 3																
İş 4																
İş 5																
İş 6																
İş 7																

Şekil 2.1: Zaman çizelgesi (Gantt Chart)

Proje tek bir bitirme öğrencisi tarafından yapıldığından iş bölümü mevcut değildir, tüm modüllerden ve iş parçalarından tek bir öğrenci sorumludur.

3 KURAMSAL BİLGİLER

Projenin geliştirilme şartları ve proje kapsamının iyice anlaşılabilmesi için hayati teknolojiler, geliştirme ortamı ve kolaylaştırıcı yazılım eklentileri aşağıda açıklanmıştır:

3.1 SQL Server 2005 (Express Edition)

Verilerin tutulma ortamı olarak SQL Server 2005 Express Edition kullanılmıştır. Visual Studio 2005 ile beraber gelen SQL Server 2005'in kısıtlı bir sürümü olan bu veritabanı projede kullanılan ilişkisel veritabanı tabloları ve saklı yordamları destekleyerek ücretsiz ve yeterli bir veritabanı seçimidir. Ayrıca .Net'in System.Data.SqlClient ad uzayındaki sınıflar vasıtası ile .veritabanı üzerinde tam bütünlük ve uyumlu bir geliştirme ortamı sağlamaktadır.

3.2 .NET

.NET Microsoft'un 2001 yılında ortaya çıkardığı ve devamlı geliştirerek en son istikrarlı sürümünü 2.0 ile dilden mümkün olduğunca bağımsız (desteklenen dillerin bazıları: C#, Visual Basic.NET, J#) .NET çatısı üzerinde yazılım geliştirmeye olanak sağlayan; yazılım geliştirme dünyasına kolaylıklar, yenilikler ve yeni standartlar getirmiş bir mimaridir.

3.2.1 Katmanlı Mimari

Eski yazılımlarda genelde iki katmanlı mimari kullanılmış ve kalın bir istemci tarafı, ince bir sunucu tarafı mevcut tutulmuştur.[4] Bu da dengesiz bir dağılım oluşturmuş, kontrolü ve uygulama üzerinde değişiklik yapmayı zorlaştırmıştır.

Üç katmanlı mimari fakat temelde n katmanlı bir istemci-sunucu yazılım mimarisi modelidir. Üç katmanlı mimaride uygulamanın arayüz, fonksiyonel iş mantığı, veri depolaması ve veri erişim kısımları birbirlerinden ayrı birer parça olarak gerçekleşir.[5] Bu katmanlar kendi içlerinde de parçalara ayrılabilirler ve bu şekilde oluşan mimari n veya çok katmanlı olarak adlandırılır.

Katmanlı mimaride amaç yazılım parçalarının mümkün olduğunca birbirinden bağımsız olarak geliştirilip, daha sonra bir değişim yapılması gerektiğinde mümkün olduğunca az ve temel yerlerde değişiklik yapılmasını sağlayarak yönetimi ve kontrolü arttırmaktır. Ayrıca bu şekilde yeniden kullanılabilirliği yüksek yazılım parçaları elde edilebilir veya benzer işler için şablon bir yazılım yapılarak ufak değişikliklerle birbirinden çok farklı uygulamalar geliştirilebilir.

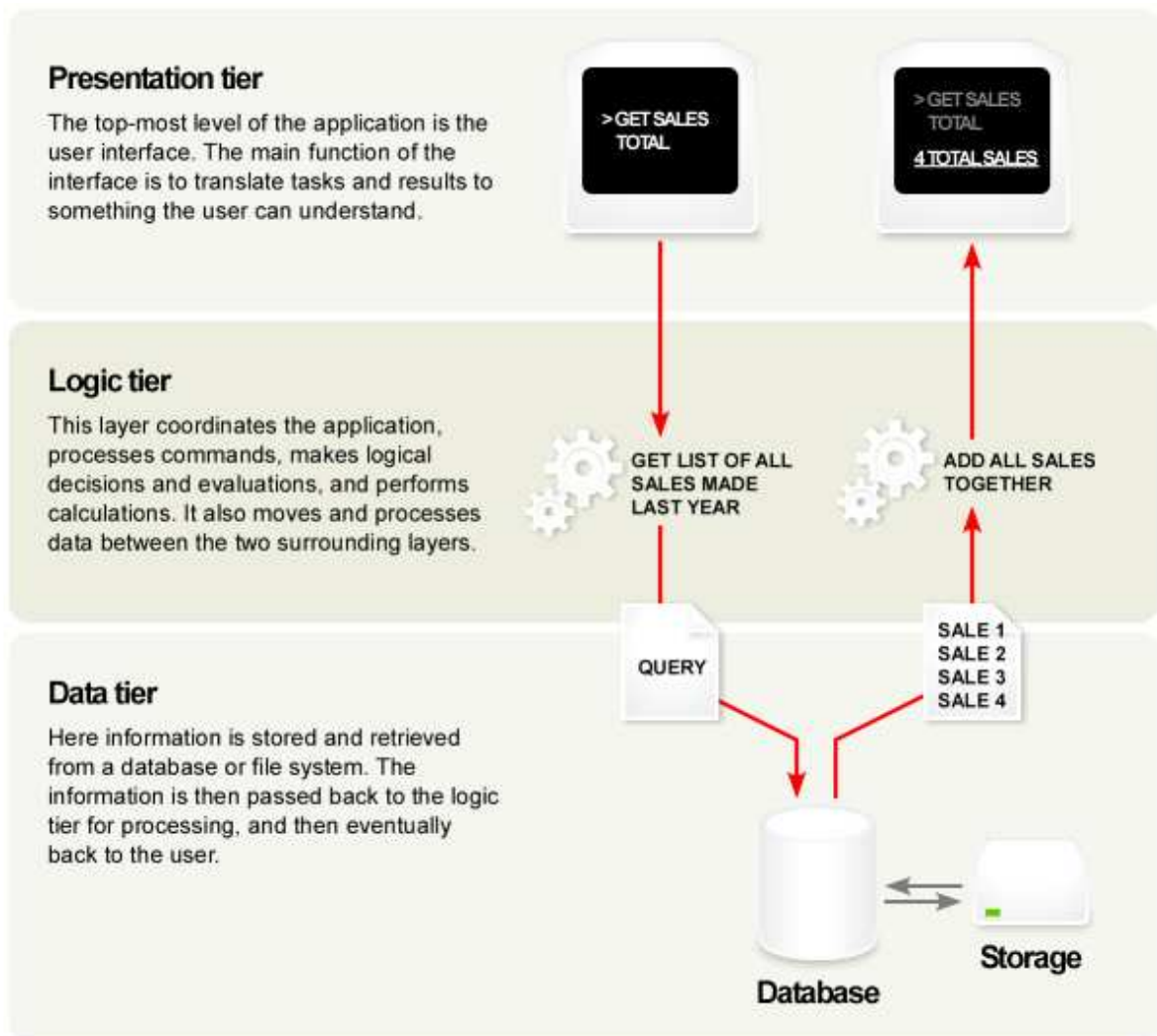
Üç katmanlı mimarideki katmanlar ve işlevleri aşağıda açıklanmıştır:

Veri katmanı: Verileri tutan, veri almak ve yazmak için gelen isteklere cevap veren katmandır. SQL Server, MS Access, MySQL, Oracle vb. veritabanları, veritabanı tabloları ve saklı yordamlar bu katmandadır. Bu katmanda ayrıca bir veri erişim katmanı bulunabilir, veri erişim katmanı ile veri katmanına özel işlemler yaptırılabilir.

İş katmanı: Veri katmanından aldığı bilgileri sunum katmanındaki uygulamaların anlayabileceği şekilde işleyerek ileten katmandır. Tam tersi yönde de sunum katmanından gelen bilgileri işleyerek veri katmanına iletir. Genelde tüm önemli ve karmaşık işler bu katmanda yapılır. Örneğin hesaplamalar, verilerin işlenmesi vb. Genelde bu katmanda işlemleri gerçekleyen sınıflar olur.

Sunum katmanı: İş katmanından aldığı düzenlenmiş ve anlamlı verileri kullanıcının anlayacağı şekilde sunar ve kullanıcıdan bilgi olarak iş katmanına iletir. Bu katmanda çok çeşitte cihaz (PDA, GSM, web tarayıcısı vb.) ve uygulamalar olabilir.

Örnek bir üç katmanlı mimari yapısı resmi Şekil 3.1’de verilmiştir.

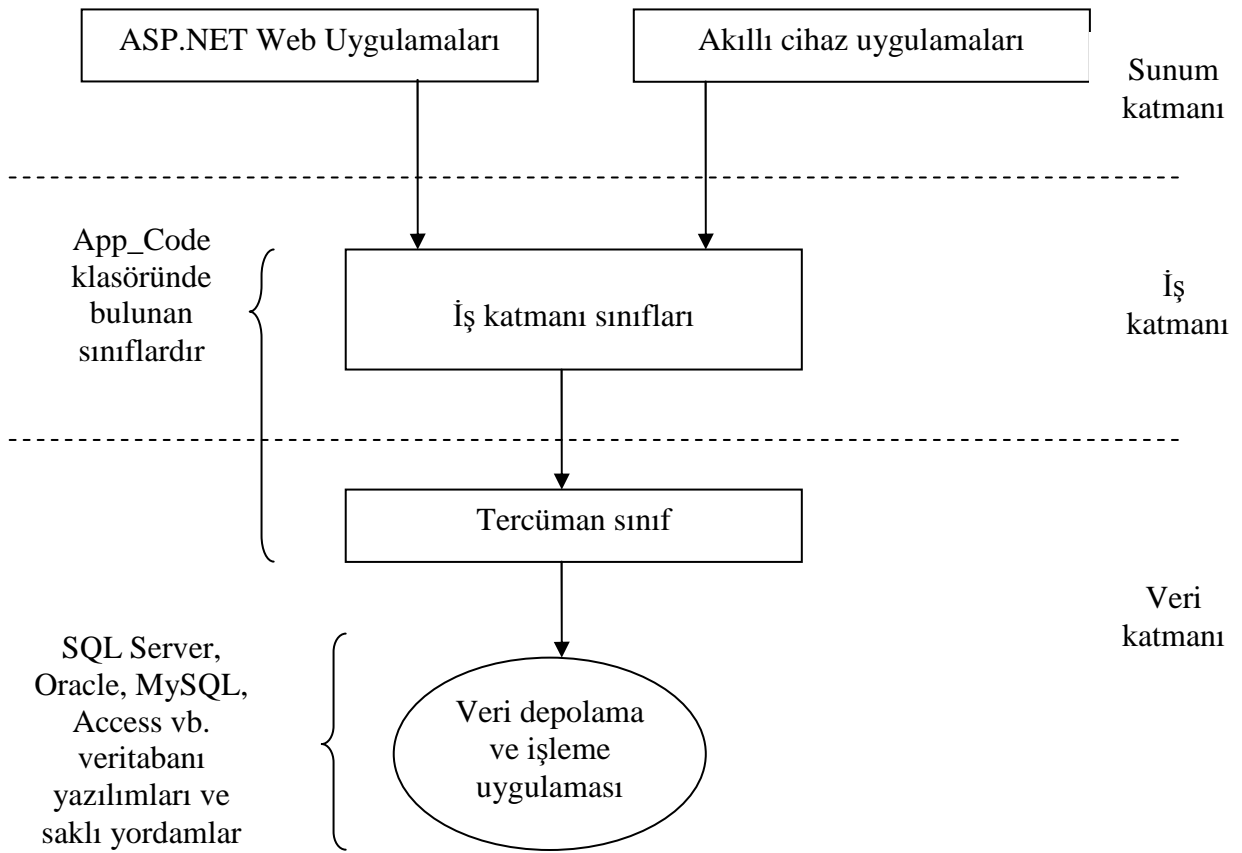


Şekil 3.1: Üç katmanlı mimariye genel bir bakış. [6]’dan alınmıştır. Telifsizdir.

3.2.2 ASP.NET 2.0

ASP.NET, .NET ile bütünleştirilmiş “Active Server Pages” teknolojisinin devamıdır. En son istikrarlı sürüm olan 2.0 ile birçok yeni hazır nesne ve kullanılabilir yeni özellikler edinmiştir. Üç katmanlı mimariye ve SQL Server’a uyumluluğundan dolayı bu şekilde uygulama geliştirme yapmak isteyenlere büyük kolaylıklar sağlamaktadır. Örneğin ObjectDataSource ile bir iş katmanı nesnesinin methodu sunum katmanındaki bir nesneye kaynak olarak gösterilebilmektedir.

ASP.NET için katmanlı mimari ve katmanları içeren dosyalar Şekil 3.2’de gösterilmiştir.



Şekil 3.2: ASP.NET’te üç katmanlı mimari ([7]’den esinlenilmiştir)

3.2.3 Windows Mobile 5.0

Windows Mobile 5.0 akıllı cihazlar için özel tasarlanmış Microsoft Win32 API temelli kısıtlı bir Windows işletim sistemidir. Windows Mobile 5.0 cep bilgisayarları, akıllı telefonlar ve taşınabilir medya merkezlerinde kullanılmaktadır. Windows CE 5.0’a dayanan ve .NET Compact Framework 1.0 SP2’yi bulunduran WM5.0 2005 yılında piyasaya sürülmüştür.[8]

Visual Studio 2005 ile bu işletim sistemi üzerinde uygulama geliştirebilmek için “Windows Mobile 5.0 Pocket PC SDK” kurulmalıdır. Varsayılan olarak WM5 emülatörü ve yazılım geliştirme araçları mevcut değildir.

3.2.4 Web Servisleri

Web servisleri ASP.NET web uygulamaları içerisinde geliştirilebilen belli başlı işleri yaparak sonuçları belli standartlarda (WSDL) XML tabanlı veriler olarak döndüren özel uygulamalardır. Akıllı cihazların web uygulamalarına bağlanarak, daha önceden tanımlanmış işleri yapabilmelerine olanak tanıyarak, sistemler arasında birlikte çalışırılık ve uyumluluk sağlarlar.

3.3 J2ME

J2ME Java 2 Micro Edition’ın kısaltmasıdır. J2ME, cep telefonlarının kısıtlı ortamında uygulama geliştirebilmek için tasarlanmış Java dilinin kısıtlı, ufak ve cep telefonlarına özel bir tipidir.

3.3.1 netBeans 5.0

netBeans Java temelli bir yazılım geliştirme ortamıdır.

3.3.2 netBeans Mobility Pack

netBeans Mobility Pack, netBeans geliştirme ortamına J2ME’ye özel yeni araçların, şablonların ve nesnelerin eklenmesini sağlayan bir eklentidir. Bu eklenti ile birçok yeni özellik ile J2ME uygulaması geliştirmek kolaylaştırılmıştır. Örneğin J2ME Web Service Client ile hedefi verilen bir web servisi için tüm mevcut methodlar kullanılabilir şekilde nesneler olarak yaratılmaktadır.

4 ANALİZ VE MODELLEME

4.1 Analiz

Projede 5 adet ana kavram olacaktır. İlk olarak hataların ait olduğu bir *proje* olmalıdır. Bir projeye ait *sorular* olmalıdır. Projeye ait gönderilen *hatalar* olmalıdır. Hataları gönderebilen *kullanıcılar* ve hataları düzeltmekle sorumlu özel kullanıcı olan *sorumlu* kişiler olmalıdır. Bu kavramlar veritabanında tablolar olarak temsil edilerek, gerekirse ufak nesneler olarak da uygulama içerisinde veri tipi olarak kullanılabilir.

Web uygulaması için veri katmanından verileri işleyip getirecek iş katmanı sınıfları gereklidir. İş katmanı sınıflarının getirdiği verileri alarak kullanıcıya/kullanıcıdan giriş/çıkış işlemleri yapacak sunum katmanı için de asp.net web sayfaları hazırlanmalıdır. Mümkün olduğunca sunum katmanını daha düzenli yapabilmek adına sadece arayüzü içeren aspx sayfaları ile bu arayüze bilgileri mantıklı bir şekilde ileten arka kod sayfaları (aspx.cs) gerçekleştirilmelidir.

Akıllı cihaz erişimi için web servisleri gerçekleştirilmelidir. Akıllı cihazlardan yapılması beklenen veya istenen işlere göre tasarlanacak web servisleri mümkün olduğunca basit işler yapılmalıdır, mümkün olduğunca az veri alıp göndermelidir ki zaten kısıtlı imkânlarla çalışan akıllı cihazların bellek ve işlemci durumu yüzünden işlemler verimsiz bir şekilde gerçekleşmesin.

PDA uygulaması için compact framework ve windows mobile işletim sisteminin kısıtlamaları göz önüne alınarak, mümkün olduğunca sade bir arayüz ile sadece gerektiğinde web servislerine bağlanarak işlem yapılmasının sağlanması, web servislerine bağlanılmak ve verileri getirmek için gereken sürenin uzunluğundan dolayı göz önüne alınması gereken bir kısıtlamadır.

Cep telefonu uygulaması için ise PDA' dan daha kısıtlı bir ortam olduğu için bellek gereksinimleri ve cep telefonlarında yeterli derecede olmayan standartlaşma sebebiyle daha dikkatli olunması gereklidir. Aynı şekilde GPRS, 3G vb. cep telefonu internet erişim teknolojilerinin hız kısıtlamaları nedeniyle az veri alıp göndermek ve sadece gerektiğinde web servislerine bağlanarak işlem yapılması sağlanmalıdır.

4.2 Modelleme

Modelleme ve tasarım aşaması birlikte yürütüldüğünden dolayı modelleme kısmı da tasarım aşaması ile birlikte ayrıntılı olarak incelenmiş ve bir sonraki bölümde açıklanmıştır.

5 TASARIM, GERÇEKLEME VE TEST

5.1 Veri Katmanı

5.1.1 Veritabanı

Projeler için veritabanında *Proje* adlı tabloda tutulacak alan adları ve veritabanı tipleri şunlardır:

```
[ID] [int] IDENTITY(1,1) NOT NULL,
[ISIM] [nvarchar](50) COLLATE Turkish_CI_AS NOT NULL,
[TUR] [nvarchar](20) COLLATE Turkish_CI_AS NOT NULL
```

Sorular için veritabanında *Soru* adlı tabloda tutulacak alan adları ve veritabanı tipleri şunlardır:

```
[ID] [int] IDENTITY(1,1) NOT NULL,
[PROJEID] [int] NOT NULL,
[SORU] [nvarchar](100) COLLATE Turkish_CI_AS NOT NULL CONSTRAINT
[DF_Soru_SORU] DEFAULT (N'bu soru eksik'),
[GELISTIRICINOT] [nvarchar](100) COLLATE Turkish_CI_AS NULL,
[KULLANICINOT] [nvarchar](100) COLLATE Turkish_CI_AS NULL,
[EVETID] [int] NULL CONSTRAINT [DF_Soru_EVETID] DEFAULT ((0)),
[HAYIRID] [int] NULL CONSTRAINT [DF_Soru_HAYIRID] DEFAULT ((0)),
[ONCEKIID] [int] NULL CONSTRAINT [DF_Soru_ONCEKIID] DEFAULT ((0))
```

Buradaki PROJEID *Proje* tablosunun birincil anahtarı olan ID' ye yabancı anahtardır.

Hatalar için veritabanında *Hata* adlı tabloda tutulacak alan adları ve veritabanı tipleri şunlardır:

```
[ID] [int] IDENTITY(1,1) NOT NULL,
[PROJEID] [int] NOT NULL,
[SENSORUID] [int] NOT NULL,
[ACIKLAMA] [nvarchar](max) COLLATE Turkish_CI_AS NOT NULL,
[YOLLAYANID] [int] NOT NULL,
[COZUM] [text] COLLATE Turkish_CI_AS NULL,
[COZUMONAY] [bit] NOT NULL CONSTRAINT [DF_Hata_COZUMONAY]
DEFAULT ((0)),
[CEVAP] [bit] NOT NULL CONSTRAINT [DF_Hata_CEVAP] DEFAULT ((0)),
[ZAMAN] [datetime] NULL
```

Buradaki PROJEID *Proje* tablosunun birincil anahtarı olan ID' ye yabancı anahtardır.
Buradaki SENSORUID *Soru* tablosunun birincil anahtarı olan ID' ye yabancı anahtardır.
Buradaki YOLLAYANID *Kullanici* tablosunun birincil anahtarı olan ID' ye yabancı anahtardır.

Kullanıcılar için veritabanında *Kullanici* adlı tabloda tutulacak alan adları ve veritabanı tipleri şunlardır:

```
[ID] [int] IDENTITY(1,1) NOT NULL,
[TIPID] [int] NOT NULL,
[BILGOZELLIK] [nvarchar](max) COLLATE Turkish_CI_AS NULL,
[ISIM] [nvarchar](20) COLLATE Turkish_CI_AS NOT NULL CONSTRAINT
[DF_Kullanici_ISIM] DEFAULT (N'varsayılan'),
[SIFRE] [nvarchar](10) COLLATE Turkish_CI_AS NOT NULL CONSTRAINT
[DF_Kullanici_SIFRE] DEFAULT ((123)),
[MAIL] [nvarchar](40) COLLATE Turkish_CI_AS NOT NULL CONSTRAINT
[DF_Kullanici_MAIL] DEFAULT (N'yok@hatatakip.com')
```

Buradaki TIPID *KTip* tablosunun birincil anahtarı olan ID' ye yabancı anahtardır. Kullanıcı tipleri için veritabanında *KTip* adlı tabloda tutulacak alan adları ve veritabanı tipleri şunlardır:

```
[ID] [int] IDENTITY(1,1) NOT NULL,
[TIP] [nvarchar](50) COLLATE Turkish_CI_AS NOT NULL
```

Sorumlulukları belirtmek için kullanıcı tipi 'sorumlu' olan kullanıcıların ID alanları ile sorumlu olacakları hata veya projelerin ID alanları arasında ilişkilerin tutulacağı iki adet tablo daha vardır. *ProjeSorumlusu* ve *HataSorumlusu*. Adlarından da anlaşılacağı kişinin sorumlu olduğu proje ID ve kendisinin kullanıcı ID'si *ProjeSorumlusu* tablosunda, aynı şekilde kişinin sorumlu olduğu hata ID ve kendisinin kullanıcı ID'si *HataSorumlusu* tablosunda belirtilecektir. Bu tablolar:

ProjeSorumlusu

```
[PROJEID] [int] NOT NULL,
[SORUMLUID] [int] NOT NULL
```

Buradaki PROJEID *Proje* tablosunun birincil anahtarı olan ID' ye yabancı anahtardır. Buradaki SORUMLUID *Kullanici* tablosunun birincil anahtarı olan ID' ye yabancı anahtardır.

HataSorumlusu

```
[SORUMLUID] [int] NOT NULL,
[HATAID] [int] NOT NULL
```

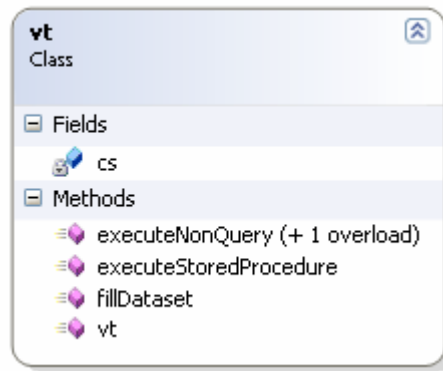
Buradaki HATAID *Hata* tablosunun birincil anahtarı olan ID' ye yabancı anahtardır. Buradaki SORUMLUID *Kullanici* tablosunun birincil anahtarı olan ID' ye yabancı anahtardır.

5.1.2 Saklı Yordamlar

Eklenilen bir kaydın otomatik olarak atanan identity değerini kaydı ekledikten hemen sonra geri dönüş değeri olarak alabilmek için veritabanına Stored Procedure olarak bilinen iki saklı yordam yazılmıştır. Bunlar **hataEkle** ve **soruIcinYerAyir** adları ile çağırılabilir ve derlenmiş olarak veritabanında tutulurlar, geri dönüş değeri olarak eklenilen kaydın identity değerini döndürürler.

5.1.3 Tercüman (Adapter) Sınıfı

İş katmanından gelen veritabanı işlem isteklerini veritabanının anlayacağı şekilde veritabanına özel nesneleri ve methodları kullanarak bir nevi arada tercümanlık yaparak adaptör görevini yerine getiren sınıf ile veritabanı değiştirilmek istendiğinde sadece bu sınıfın değiştirilmesi yeterli olacaktır. Böylece büyük bir ölçeklenebilirlik avantajı elde ettiren bu sınıfın sınıf diyagramı Şekil 5.1’de UML diyagramı ise Şekil 5.2’de verilmiştir.



Şekil 5.1: Veritabanı tercüman sınıfı, sınıf diyagramı *

vt
{namespace=_vt}
- cs : string
+ vt()
+ executeNonQuery(sql : string) : bool
+ executeNonQuery(sql : string, parameterNameList : ArrayList, parameterList : ArrayList) : bool
+ executeStoredProcedure(storedProcedureName : string, parameterNameList : ArrayList, parameterList : ArrayList) : uint
+ fillDataset(sql : string) : DataSet

Şekil 5.2: Veritabanı tercüman sınıfı, UML diyagramı **

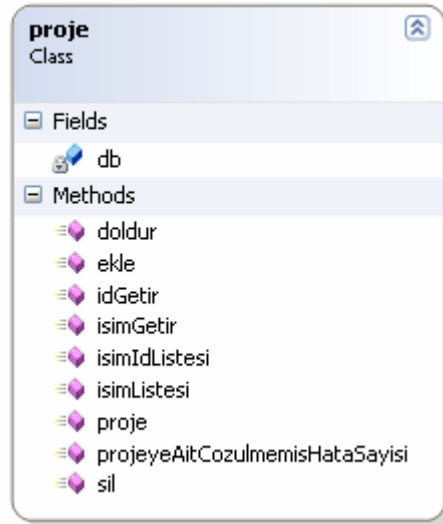
*: Tam olarak UML sınıf diyagramı olmayan bu özet sınıf diyagramları Visual Studio 2005 ile gerçekleştirilmiştir.

** : Tüm UML diyagramları Metamill 4.2 programı deneme sürümü ile gerçekleştirilmiştir.

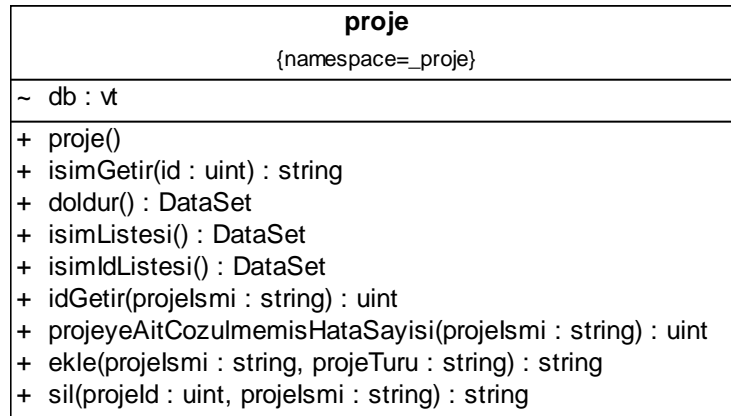
5.2 İş Katmanı

5.2.1 proje.cs

Projelerle ilgili işlemlerden sorumlu, gelen isteklere cevap veren sınıftır. Proje ile ilgili tüm işlemler bu sınıfın methodları vasıtası ile yapılır. App_Code klasöründe bulunup kullanılacağı zaman bir örnek nesne olarak yaratılması gerekir. Sınıf diyagramı Şekil 5.3'te, UML diyagramı ise Şekil 5.4'te verilmiştir.



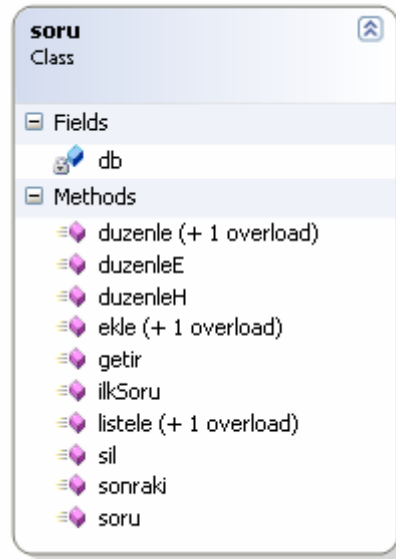
Şekil 5.3: Proje işlemlerinden sorumlu sınıf, sınıf diyagramı



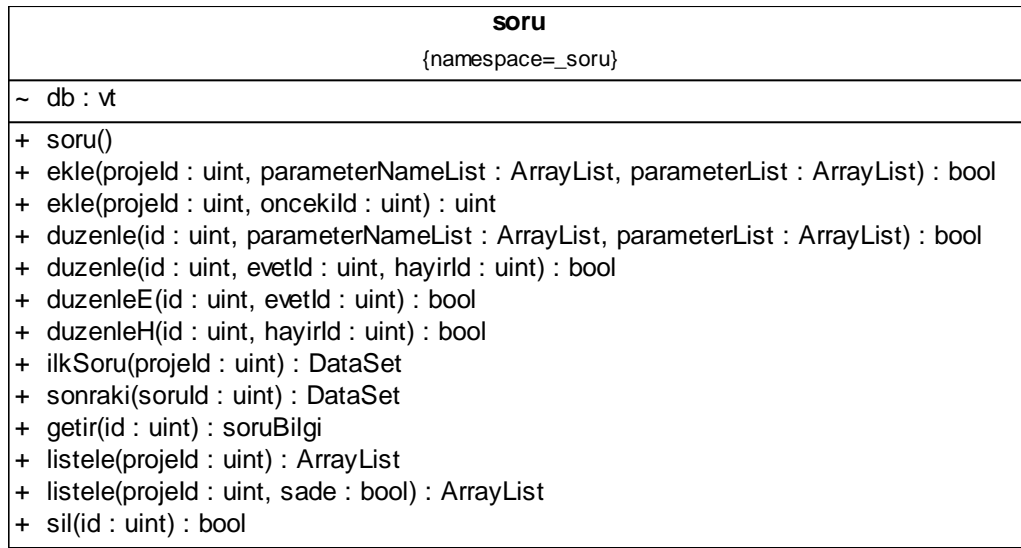
Şekil 5.4: Proje işlemlerinden sorumlu sınıf, UML diyagramı

5.2.2 soru.cs

Sorularla ilgili işlemlerden sorumlu, gelen isteklere cevap veren sınıftır. Soru ile ilgili tüm işlemler bu sınıfın methodları vasıtası ile yapılır. App_Code klasöründe bulunup kullanılacağı zaman bir örnek nesne olarak yaratılması gerekir. Sınıf diyagramı Şekil 5.5'te, UML diyagramı ise Şekil 5.6'da verilmiştir.



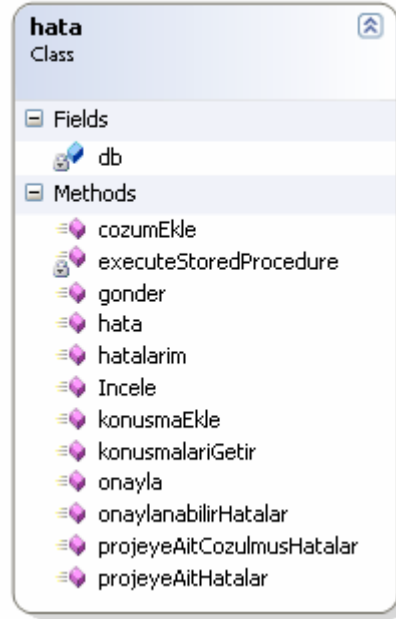
Şekil 5.5: Soru işlemlerinden sorumlu sınıf, sınıf diyagramı



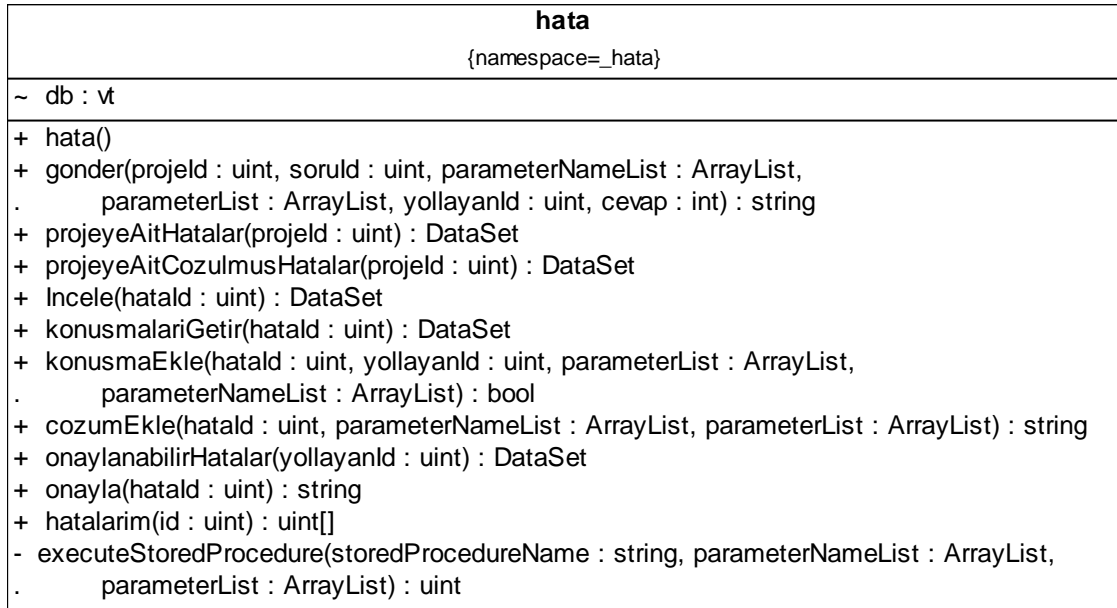
Şekil 5.6: Soru işlemlerinden sorumlu sınıf, UML diyagramı

5.2.3 hata.cs

Hatalarla ilgili işlemlerden sorumlu, gelen isteklere cevap veren sınıftır. Hata ile ilgili tüm işlemler bu sınıfın methodları vasıtası ile yapılır. App_Code klasöründe bulunup kullanılacağı zaman bir örnek nesne olarak yaratılması gerekir. Sınıf diyagramı Şekil 5.7’de, UML diyagramı ise Şekil 5.8’de verilmiştir.



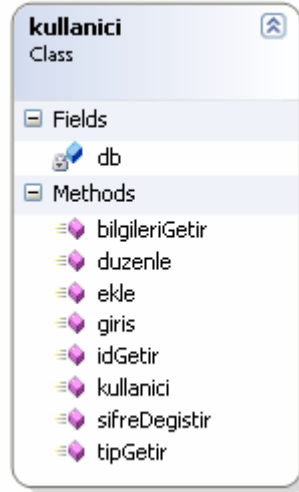
Şekil 5.7: Hata işlemlerinden sorumlu sınıf, sınıf diyagramı



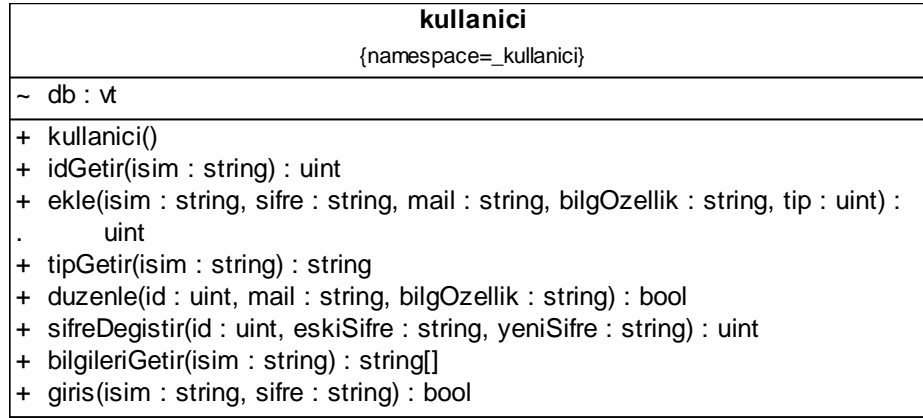
Şekil 5.8: Hata işlemlerinden sorumlu sınıf, UML diyagramı

5.2.4 kullanıcı.cs

Kullanıcılarla ilgili işlemlerden sorumlu, gelen isteklere cevap veren sınıftır. Kullanıcı ile ilgili tüm işlemler bu sınıfın methodları vasıtası ile yapılır. App_Code klasöründe bulunup kullanılacağı zaman bir örnek nesne olarak yaratılması gerekir. Sınıf diyagramı Şekil 5.9’da, UML diyagramı ise Şekil 5.10’da verilmiştir.



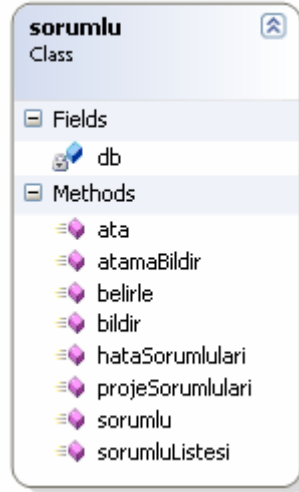
Şekil 5.9: Kullanıcı işlemlerinden sorumlu sınıf, sınıf diyagramı



Şekil 5.10: Kullanıcı işlemlerinden sorumlu sınıf, UML diyagramı

5.2.5 sorumlu.cs

Özel kullanıcılar olan sorumlularla ilgili işlemlerden sorumlu, gelen isteklere cevap veren sınıftır. Sorumlular ile ilgili özelleşmiş tüm işlemler bu sınıfın methodları vasıtası ile yapılır. App_Code klasöründe bulunup kullanılacağı zaman bir örnek nesne olarak yaratılması gerekir. Sınıf diyagramı Şekil 5.11’de, UML diyagramı ise Şekil 5.12’de verilmiştir.



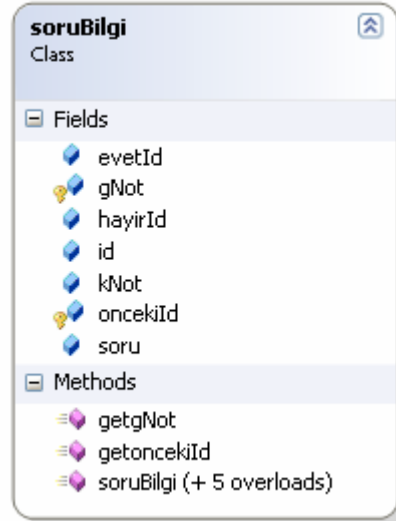
Şekil 5.11: Sorumlu kişi işlemlerinden sorumlu sınıf, sınıf diyagramı

sorumlu {namespace=_sorumlu}
~ db : vt
+ sorumlu() + sorumluListesi() : DataSet + belirle(projeld : uint, sorumluld : uint, projelsim : string, sorumlulsim : string) : string + ata(hataId : uint, sorumluld : uint, sorumlulsim : string) : string + bildir(hataId : uint, projeld : uint) : uint + atamaBildir(hataId : uint, sorumluld : uint) : uint + projeSorumlulari() : DataSet + hataSorumlulari() : DataSet

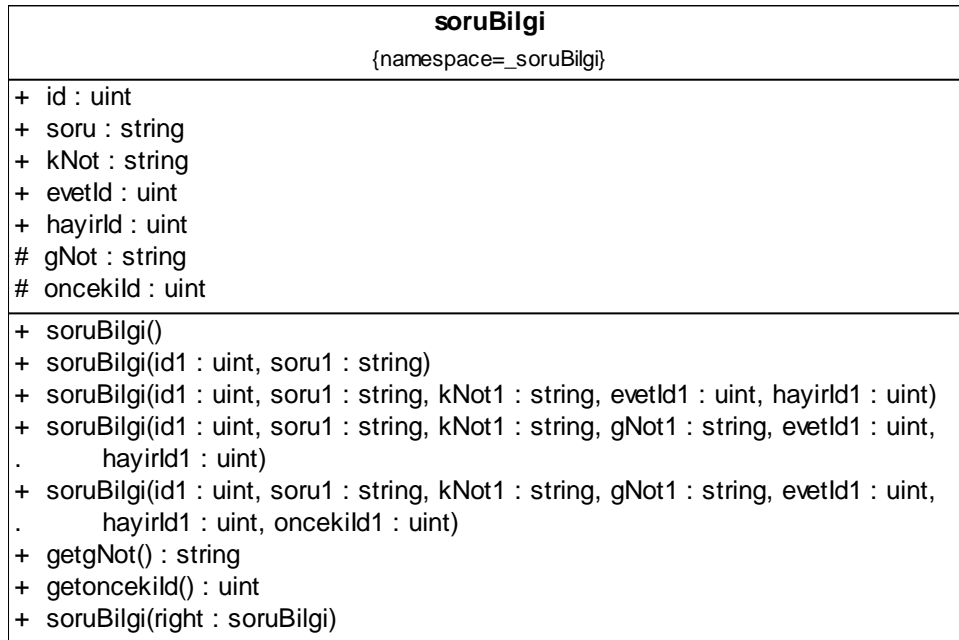
Şekil 5.12: Sorumlu kişi işlemlerinden sorumlu sınıf, UML diyagramı

5.2.6 soruBilgi.cs

Web servislerinde “soru” kavramını temsil etmek için gerçekleştirilmiş ufak bir sınıftır. Diğer web uygulamalarında da soruyu temsil etmek için kullanılabilir. App_Code klasöründe bulunup kullanılacağı zaman bir örnek nesne olarak yaratılması gerekir. Sınıf diyagramı Şekil 5.13’te, UML diyagramı ise Şekil 5.14’te verilmiştir.



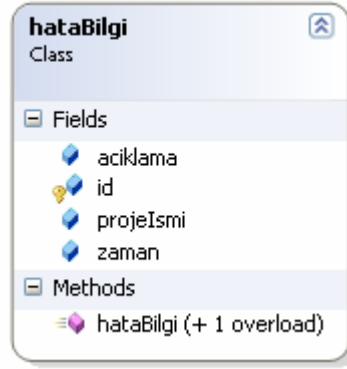
Şekil 5.13: Soruları temsil eden sınıf, sınıf diyagramı



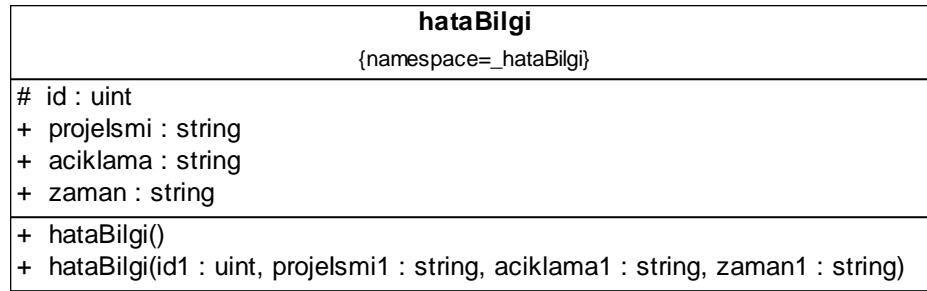
Şekil 5.14: Soruları temsil eden sınıf, UML diyagramı

5.2.7 hataBilgi.cs

Web servislerinde “hata” kavramını temsil etmek için gerçekleştirilmiş ufak bir sınıftır. Diğer web uygulamalarında da hatayı temsil etmek için kullanılabilir. App_Code klasöründe bulunup kullanılacağı zaman bir örnek nesne olarak yaratılması gerekir. Sınıf diyagramı Şekil 5.15’te, UML diyagramı ise Şekil 5.16’da verilmiştir.



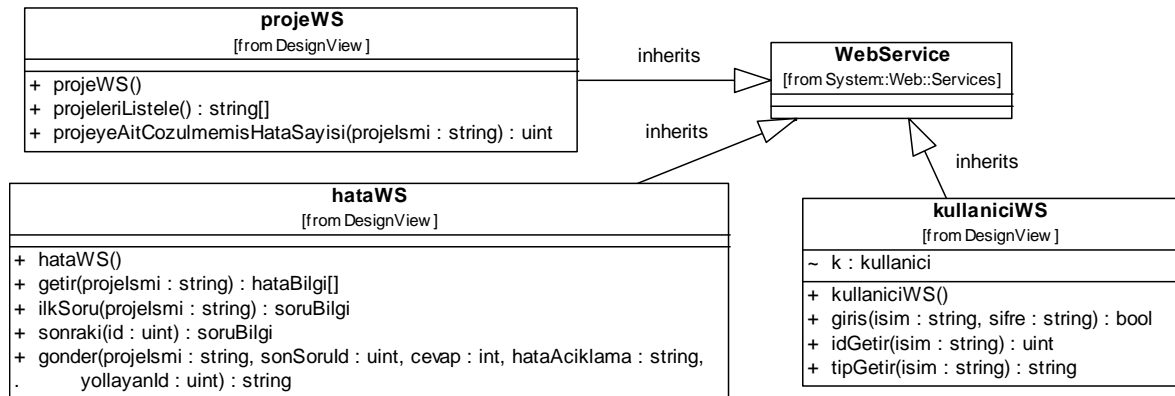
Şekil 5.15: Hataları temsil eden sınıf, sınıf diyagramı



Şekil 5.16: Hataları temsil eden sınıf, UML diyagramı

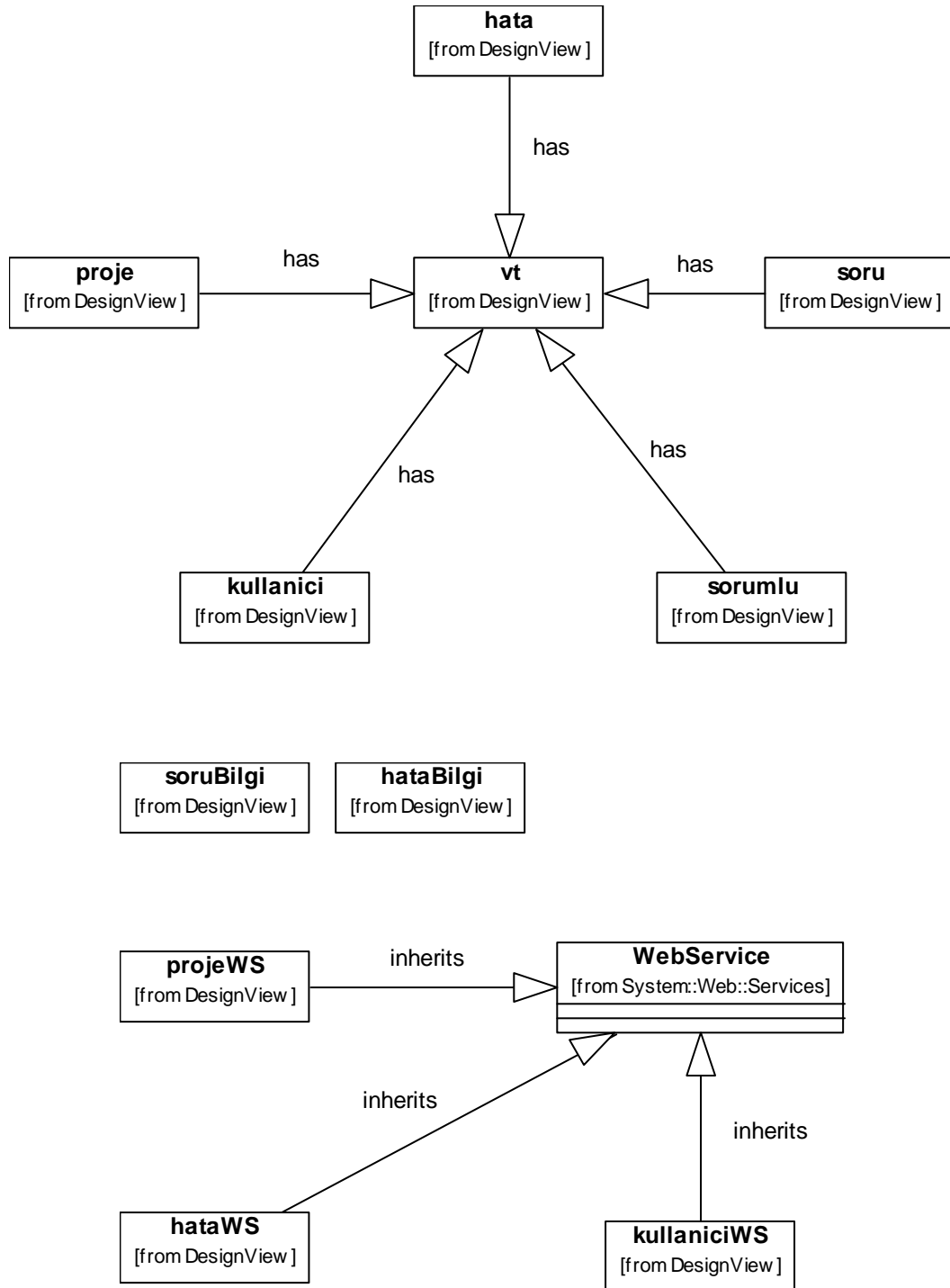
5.2.8 Web servisleri sınıfları

Akıllı cihazların erişimi ve işlem yapabilmesi için üç adet web servisi gerçekleştirilmiştir. Kullanıcı işlemlerine cevap veren **kullaniciWS.cs**, projelerle ilgili işlemler yapan **projeWS.cs** ve hatalarla ilgili işlemler yapmaya olanak tanıyan **hataWS.cs**. Her biri System.Web.Services’den türeyerek public web methodları vasıtası ile işlem yapılmasını sağlarlar. UML diyagramları Şekil 5.17’de verilmiştir.



Şekil 5.17: Web servisleri sınıfları, UML diyagramı

App_Code klasörü içindeki tüm sınıfları içeren UML diyagramı Şekil 5.18’de verilmiştir.



Şekil 5.18:App_Code tüm sınıflar özet, UML diyagramı

5.3 Sunum Katmanı

Sunum katmanında 4 adet uygulama vardır. ASP.NET ile gerçekleştirilmiş hata giderici ve hata iletici web uygulamaları ana uygulama olup, ASP.NET sayfaları olarak; PDA uygulaması ise compact framework ve Windows Mobile 5.0 ile gerçekleştirilmiştir. Başka bir akıllı cihaz uygulaması olan cep telefonu uygulaması ise Java destekleyen telefonlarda çalışabilen J2ME vasıtasıyla gerçekleştirilmiştir. PDA ve cep telefonu uygulamalarının da hem hata giderici hem de hata ileticiler için iki farklı çalışma şekli vardır. Eğer bir iletici sisteme giriş yapmışsa sadece göndericiler tarafından kullanabilen işlemleri görüntüleyemez.

Her uygulama ve her çeşit kullanıcı için bir sisteme giriş ekranı mevcuttur. Ana uygulama için, ASP.NET 2.0 ile gelen bir yenilik olan master sayfası özelliği MasterPage.master adıyla projeye eklenmiştir. Bu master sayfası tüm sayfaların şablonu olup, her sayfa için düzeni ve menüleri barındırmaktadır. Her sayfada sadece orta kısımdaki alan değişebilmektedir, diğer kısımlar sabittir. Yalnız kişinin kullanıcı tipine göre yapabileceği işlemler değişebildiğinden dolayı eğer kullanıcı tipi gönderici ise sorumlu menüsü görünmemektedir.

5.3.1 Hata giderici web uygulaması

Ana uygulama ASP.NET tabanlı olduğundan dolayı tüm önemli ve kilit işlemler aspx sayfaları olarak gerçekleştirilmiştir. Hata gidericilerin erişebileceği sayfalar ve hangi işlemleri yapabilecekleri aşağıda açıklanmıştır.

gir.aspx : Sisteme giriş yapmayı sağlayan sayfa.

cik.aspx : Sistemden çıkış yapılmasını sağlayan sayfa.

projeEkle.aspx : Hata eklenebilecek yeni bir proje ekleme veya mevcut bir projeyi silme işlemi yapılabilen sayfa.

soruBelirle.aspx : İstenen projeye soru ekleme, mevcut soruları düzenleme, ağaç yaprağı durumundaki soruları silme işlemlerine erişimi sağlayan sayfa.

SoruEkle.aspx : Projeye ait yeni bir soru ekleme sayfası. soruBelirleme.aspx'ten veya SoruDuzenle.aspx'ten erişilebilir.

SoruDuzenle.aspx : Mevcut bir soruyu değiştirme, mevcut soru sonrasında evet veya hayır dendiğinde yeni bir soru sorulmasının istendiğinde eklenebilmesini sağlayan sayfadır. soruBelirle.aspx'ten erişilebilir.

hataIncele.aspx : Seçilen projeye ait gelmiş hataların özet bilgilerini görüntülemeyi ve hatanın ayrıntılarına, hatadan sorumlu kişi belirlemeye, çözüm eklemeye ulaşmayı sağlayan sayfadır.

Incele.aspx : Hata hakkında ayrıntılı bilgilerin sunulduğu, hata hakkında konuşma şeklinde fikir alışverişi yapılabilirdiği, hatadan sorumlu kişi atama ve hataya çözüm ekleme işlemlerine erişim yapılabilirdiği sayfa.

sorumluBelirle.aspx : Projelerden sorumlu kişiler belirleme işlemi yapılabilen sayfa.

HataCozumEkle.aspx : Hataya çözüm ekleme sayfasıdır. Çözüm her zaman değiştirilebilir. hataIncele.aspx ve Incele.aspx'ten erişilebilir. Çözüm eklendiğinde hatayı yollamış kişiye hataya çözüm eklendiği hakkında bilgi içeren, çözümü onaylama sayfasına ve hata ayrıntılı inceleme sayfasına link içeren bir e-posta yollanır.

E-posta şablonu:

Gönderen: Hata Takip Yazılımı (haberci@hatatakip.com)
 Alıcı: hatayı gönderen kişi e-posta adresi

Başlık: Hataya çözüm eklendi
 Metin:
 Göndermiş olduğunuz bir hataya çözüm eklendi. Lütfen kontrol ediniz.

Buradan hatanıza erişebilirsiniz.
Buradan da hata çözümüne onay verebilirsiniz.

Hata Takip Yazılımı (2006)

SorumluAta.aspx : Hatalara sorumlu kişiler atama işlemi yapılabilen sayfa. hataIncele.aspx ve Incele.aspx'ten erişilebilir. Hatadan sorumlu olarak atanan kişiye bu durumu bildiren ve ilgili hataya erişim için bir link içeren bir bilgilendirme e-postası yollanır.

E-posta şablonu:

Gönderen: Hata Takip Yazılımı (haberci@hatatakip.com)
 Alıcı: sorumlu kişi e-posta adresi

Başlık: Bir hatadan sorumlu olarak atandınız
 Metin:
 Bir hatadan sorumlu olarak belirlendiniz.

Buradan erişebilirsiniz.

Hata Takip Yazılımı (2006)

sorumluDurumlari.aspx : Tüm sorumluların hangi projelerden ve hangi hatalardan sorumlu olduklarını listeleyen sayfa. Hatalara linkler içerir.

kullaniciEkle.aspx : İstenilen kullanıcı tipinde yeni bir kullanıcı ekleme yapılabilen sayfa.

Hata gidericilerin bu sayfalardan başka diğer tüm hata iletici sayfalarına da erişimleri mevcuttur. Yani bir hata giderici, hata ileticilerin yapabildikleri tüm işlemleri yapabilir.

5.3.2 Hata iletici web uygulaması

Hata ileticiler tarafından erişilebilir sayfalar ve yapabilecek işlemler aşağıda açıklanmıştır.

gir.aspx : Sisteme giriş yapmayı sağlayan sayfa.

cik.aspx : Sistemden çıkış yapılmasını sağlayan sayfa.

kullaniciEkle.aspx : Eğer kişi bir kullanıcı değilse sadece ve sadece hata iletici olarak kaydolabilir. Başka bir kullanıcı tipi ile kaydolması sadece hata gidericiler tarafından gerçekleştirilebilir.

hataGonder.aspx : Seçilen bir projeye ait soruları evet veya hayır şeklinde cevaplayarak hata gönderme işlemi yapılabilen sayfa. Eğer istenirse soru seçerek hata gönderimi işlemi sayfasına erişim mevcuttur. Eğer hata yollanan projeden sorumlu kişiler varsa her yeni hata eklendiğinde o sorumlu kişi veya kişilere hatanın ayrıntılı incelenme sayfasına bir link içeren bir bilgilendirme e-postası yollanır.

E-posta şablonu:

Gönderen: Hata Takip Yazılımı (haberci@hatatakup.com)

Alıcı: sorumlu kişi e-posta adresi veya adresleri

Başlık: Yeni bir hata var (*proje ismi*)

Metin:

proje ismi projesine yeni bir hata eklendi.

Buradan erişebilirsiniz.

Hata Takip Yazılımı (2006)

hataGonderU.aspx : Uzmanca hata gönderimi sayfası. İstenen projenin istenilen sorusundan başlayarak hata gönderimi yapılabilir. Aynı şekilde e-posta ile sorumlu kişilere bilgilendirme yapılır.

hataIarim.aspx : Kişi tarafından gönderilmiş tüm hataların listesini ve erişim bilgilerini içeren sayfa.

hataOnay.aspx : Kişinin çözüm eklenmiş hatalarına onay verebileceği sayfa. Ayrıca hata üzerinde konuşma yapabileceği ve hatayı inceleyebileceği sayfaya erişim linki mevcuttur.

cozulmusHatalar.aspx : Seçilen projeye ait, çözümleri onaylanmış hataların özet bilgileri ve ayrıntılı bilgilerine erişim içeren sayfa.

Incele.aspx : Hata hakkında ayrıntılı bilgileri listeleyen ve hata hakkında konuşma yapılabilen sayfadır. Hata göndericilerin aksine sorumlu atama ve çözüm ekleme erişimleri mevcut değildir.

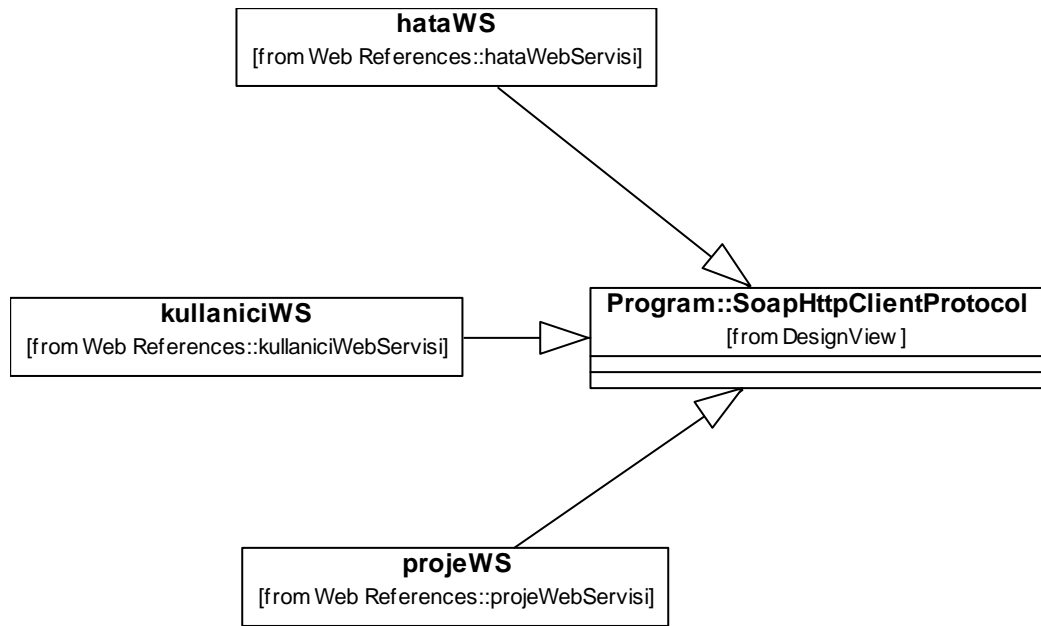
kullaniciBilgiler.aspx : Kullanıcının kişisel bilgilerini düzenleyebileceği sayfa. Şifre, mail, bilgisayar özellikleri bilgileri değişimi yapılabilir.

5.3.3 PDA uygulaması

PDA uygulaması iki adet form ve bir ana sınıftan oluşmaktadır. Program başladığında ana sınıf çalışarak giriş formunu çağırılmaktadır. Giriş formu kullanıcıWS.asmx web servisine bağlantı yaparak sisteme giriş yapmayı sağlamakta, eğer giriş başarılı olursa belli başlı kullanıcı değişkenlerinin değerlerini belirleyip ikinci forma parametre olarak vererek ikinci formu yaratıp açmaktadır. Eğer kullanıcı sorumlu ise mevcut tüm işlemleri yapabilir olduğu için form üzerinde herhangi bir değişiklik yapmamakta ama giriş yapan kullanıcı sorumlu değilse ikinci formda bazı işlemleri görünmez ve erişilemez yaparak sadece hata gönderme işlemini yapabilir kılar. Yenile butonuna basılarak projeWS.asmx web servisine bağlanan uygulama, mevcut projeleri alarak seçilebilir bir dropdownlist'e yerleştirir. Hata göndermek isteyen kişi hata gönderme kısmında yolla butonuna bastığında uygulama hataWS.asmx web servisine bağlanarak projeye ait ilk soruyu getirir. İlk soru evet veya hayır butonları ile cevaplandıktan sonra eğer yeni bir soru tanımlıysa aynı şekilde hataWS.asmx web servisinden bir sonraki soru alınarak kişiye sorulur. Bu rutin iş bir sorulacak soru kalmayınca dek sürer ve en sonunda kişiden hata hakkında ayrıntılı bilgi alınarak hata gönderme işlemi tamamlanır. Eğer kişi sorumlu ise, seçtiği projeye ait çözülmemiş hata sayısını da görüntüleyebilir. Çok kritik projeler için kontrol amacı taşıyan bu işlem iletilerinde kullanılamaz ve görüntülenemez. İşlemlerin yapıldığı ikinci formun tasarımı Şekil 5.19'da verilmiştir. Ayrıca uygulamanın UML diyagramı Şekil 5.20'de verilmiştir.

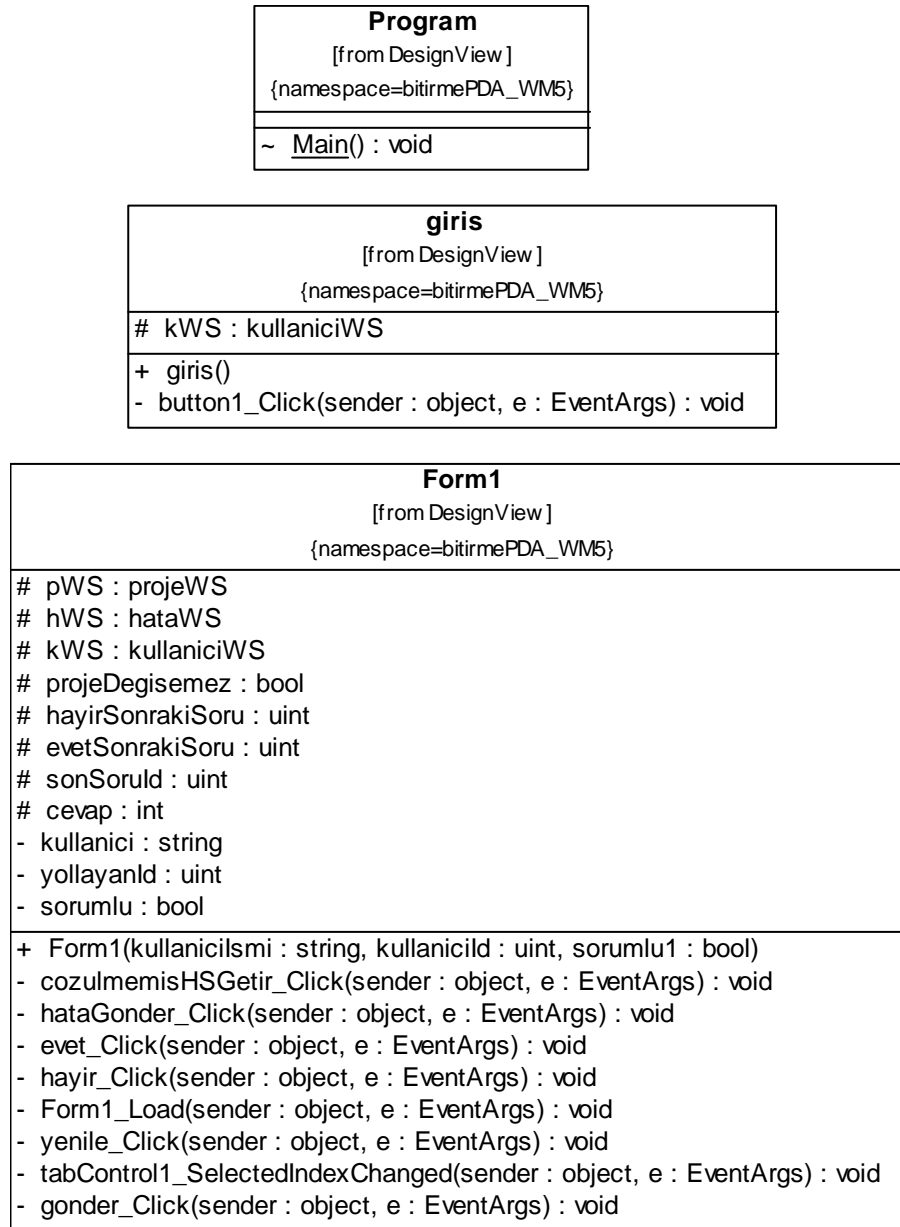


Şekil 5.19: PDA Yazılımı işlem formu



hataBilgi
[from Web References::hataWebServisi] {namespace=bitirmePDA_WM5::hataWebServisi}
<ul style="list-style-type: none"> - projelsmiField : string - aciklamaField : string - zamanField : string
<ul style="list-style-type: none"> + projelsmi() : string {accessor} + aciklama() : string {accessor} + zaman() : string {accessor}

soruBilgi
[from Web References::hataWebServisi] {namespace=bitirmePDA_WM5::hataWebServisi}
<ul style="list-style-type: none"> - idField : uint - soruField : string - kNotField : string - evetIdField : uint - hayirdField : uint
<ul style="list-style-type: none"> + id() : uint {accessor} + soru() : string {accessor} + kNot() : string {accessor} + evetId() : uint {accessor} + hayird() : uint {accessor}



Şekil 5.20: PDA Yazılımı UML sınıf diyagramı

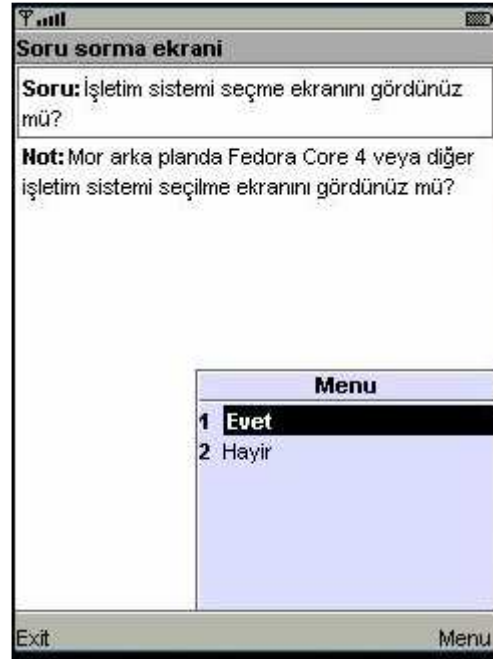
5.3.4 Cep telefonu uygulaması

Cep telefonu uygulaması üç adet paketten oluşmaktadır: Tüm işlemlerin yapıldığı main paketi içerisinde HataTakip.java ve HataTakip.mvd dosyaları vardır. HataTakip.java MIDlet'ten türemiş bir sınıftır ve ana uygulamanın koşturulduğu nesnedir, arayüzsel bileşenleri ve iş akışı algoritmalarını içerir. HataTakip.mvd ise akış diyagramını içeren netBeans'in otomatik olarak oluşturduğu formlar ve akışlar şemasıdır.

Üç adet web servisi için üç ayrı paket mevcuttur, her paket web servisinin ismini taşır. Her web servisi paketi "J2ME Web Service Client" ile oluşturulmuştur. Bu web service client bileşeni web servisi methodlarını kullanabilmek için gerekli sınıfları ve gerekli method şablonlarını oluşturur. Aynı şekilde gerekli tanımlamaları ve gerekli ayar dosyalarını

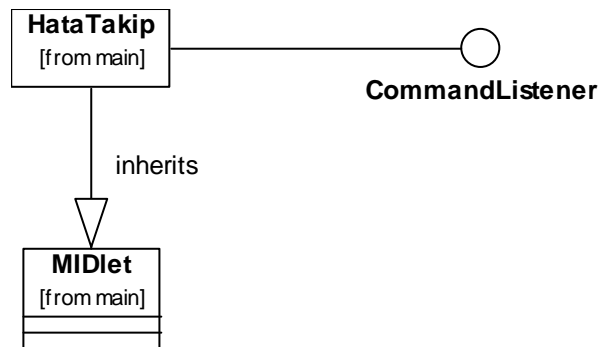
otomatik olarak oluşturur. Bu paketler import ile ana uygulama sınıfına eklendiğinde Soap şablon nesneleri tanımlanarak web servis methodları kullanılabilir hale gelmektedir.

Uygulamaya sorumlu kullanıcı tipi ile giriş yapılırsa mevcut tüm işlemlere erişilebilir fakat başka bir tip kullanıcı ile giriş yapıldığında sorumlu kişilere özel işlemler görüntülenemez. Mevcut tüm işlemler listelenir ve kişi yapmak istediği işlemi SELECT tuşu ile seçer. Yapılabilir işlemler: Projeleri listelemek, Hata göndermek ve çözülmemiş hata sayısına bakmaktır. Son işlem sadece sorumlu kişilere özeldir. Hata göndermeden önceki soru sorma ekranı örneği Şekil 5.21’de verilmiştir.

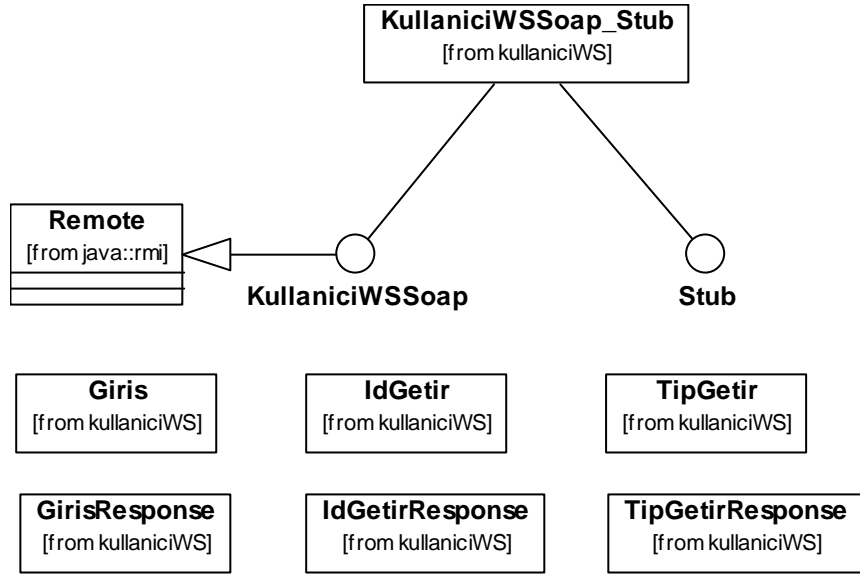


Şekil 5.21: Cep telefonu uygulaması soru sorma ekranı örnek çıktı

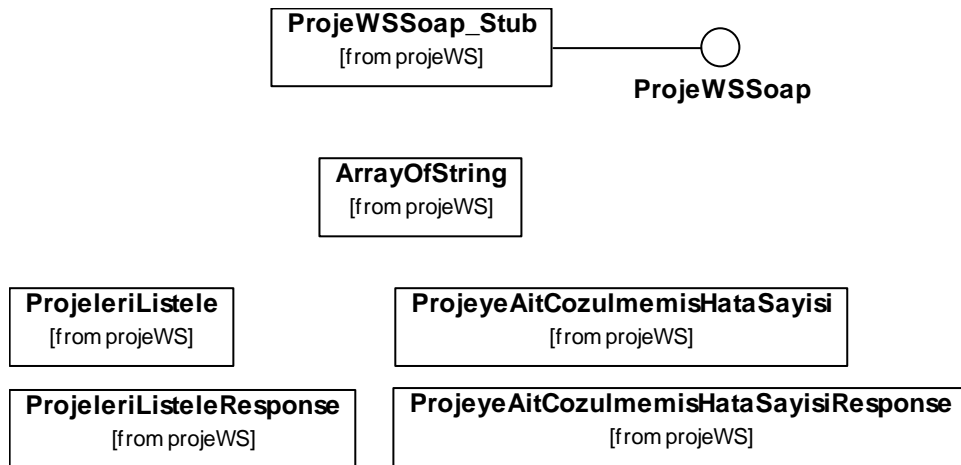
Uygulamanın paket paket UML sınıf diyagramları Şekil 5.22’de verilmiştir.



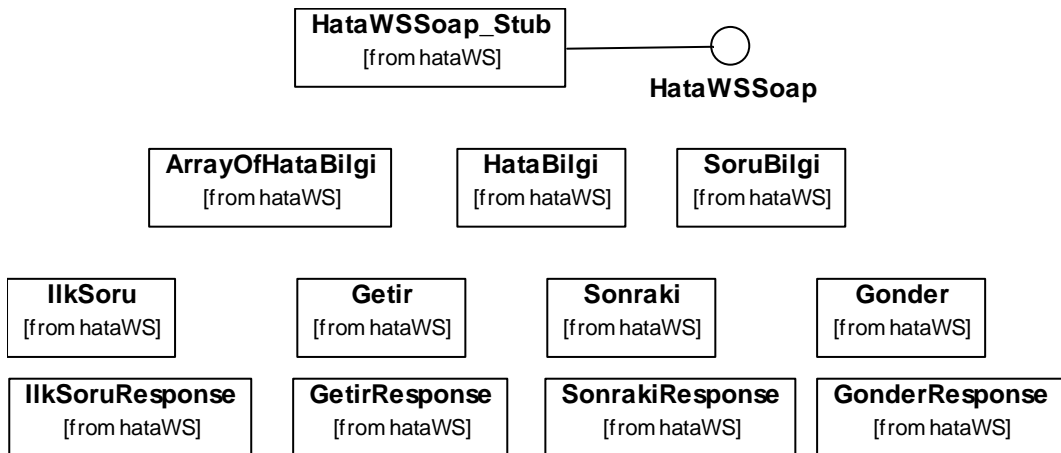
Şekil 5.22.a: main paketi özet UML sınıf diyagramı



Şekil 5.22.b: kullanıcıWS paketi özet UML sınıf diyagramı



Şekil 5.22.c: projeWS paketi özet UML sınıf diyagramı



Şekil 5.22.d: hataWS paketi özet UML sınıf diyagramı

5.4 Test

Proje tüm bileşenleri ile birlikte işlemcisi Pentium 4 1.4 GHz, belleği 384 MB olan bir bilgisayarda geliştirilmiştir. İşlemcisi Pentium 4 1.7 GHz, belleği 512MB olan başka bir bilgisayarda daha test edilmiştir. ASP.NET ve SQL Server veri işleme, erişim performansı bakımından her iki bilgisayarda da son derece tatmin edici sonuçlar alınmıştır.

PDA uygulaması Windows Mobile 5.0 Pocket PC Emulator ile test edilirken gerekli net bağlantısını sağlayabilmesi için Device Emulator Manager ile sanal olarak bilgisayara bağlı gösterilmiştir. Bu sıradaki senkronizasyon için ActiveSync 4.1 kullanılmıştır. Gerek bilgisayarın işlemci gücünün azlığı, gerekse Windows Mobile 5.0 Pocket PC Emülatörünün tüm sistem kaynaklarını kullanması sebebiyle işlemlerin kısmen yavaş yapıldığı gözlenmiştir. Ayrıca web servislerine bağlanıp verileri getirme süresinin 10 saniye mertebesine kadar çıktığı gözlenmiştir. İkinci bilgisayarda ise daha kısa sürede tepki veren emülatörün daha hızlı tepki verdiği ve daha hızlı işlem yaptığı gözlenmiştir.

Cep telefonu uygulaması Wireless Toolkit 2.2'nin emülatörü ile denenmiştir. Her iki bilgisayarda da PDA uygulamasına nazaran çok daha hızlı işlem yapıldığı ve çok daha hızlı web servislerinden bilgi alınıp işlendiği görülmüştür. İki bilgisayar arasında cep telefonu uygulaması performansı farkı gözlemlenmemiştir.

6 DENEYSEL SONUÇLAR

Hata takibi ve raporlaması alanında birçok ürün mevcuttur. Örneğin FogBugz, Bugzilla ve Mantis bu alanda en ünlü yazılımlar olarak sayılabilir. Bir başka yazılım ise GNATS (The GNU Bug Tracking System) dir.[9]

FogBugz'ın bazı güçlü özellikleri; kendine ait bir ekran görüntüsü alma programı, hata tip ayırımı, işlem düzeltme son mühlet süresi verebilme imkânı olarak sayılabilir.

Bugzilla'nın bazı güçlü özellikleri; kamuya açık bir proje olduğu için devamlı ve her an geliştirilmekte, hataları en kısa sürede bulunmaktadır, ücretsiz olarak ticari amaçsız olarak indirilip kullanılabilir, [3]'te görüldüğü gibi çok yaygın olarak kullanılmakta ve birçok büyük projenin (örneğin Mozilla, RedHat vb.) hata takibi için kullanılmaktadır.

Mantis'in bazı güçlü özellikleri; Bugzilla gibi ticari amaçlar haricinde ücretsiz olarak edinilebilmesi, PHP ve MySQL ile mümkün olduğunca hızlı çalışabilmesi olarak sayılabilir.

Bu projenin diğerlerinden en büyük farkı hata iletici ile giderici arasında ortak bir dil standardizasyonuna gitmeye çalışmasıdır. Hata giderici tarafından belirlenmiş sorulara cevap veren kullanıcı sayesinde hatayı gidermekle yükümlü kişi hatanın nerde oluşmuş olabileceği hakkında mümkün olduğu kadar çok bilgi edinebilmektedir.

Bir başka can alıcı fark ve kolaylık ise hem bir cep bilgisayarından, hem de bir cep telefonundan sisteme erişerek hata gönderebilme imkânının sağlanmasıdır. Böylece platform bağımlılığı mümkün mertebe azaltılmış ve her yerden hata gönderme ve kritik bir projeye ait hata sayısı kontrol edilebilmektedir.

7 SONUÇ ve ÖNERİLER

Uzun yıllardan beri çok fazla kişiyle, profesyonel olarak geliştirilen diğer projelerle karşılaştırıldığında, projenin bir takım eksiklikleri görülebilir. Tasarımından gerçekleşmesine, tek bir kişi ve kısıtlı bir okul dönemi içerisinde zaman darlığı içerisinde sebeplerinden şu anda çok ciddi bir alternatif olarak gözükmebilir. Ancak akıllı cihazlardan hata gönderebilme seçeneği bu projenin çok ciddi bir artısı olarak sayılabilir ve “her yerden erişim” imkânı günümüz dünyası için çok müthiş bir olanak olarak görünebilir.

Projenin iyileştirilmesi için aşağıdaki öneriler göz önüne alınabilir.

- Web uygulaması kısımlarına site içi mesajlaşma, sorumlulara mesaj ile bildirim gerçekleştirilebilir. Bu eklentiden sonra gerekli web servisi eklentileri ile akıllı cihazlar için mesajlaşma, mesaj okuma desteği eklenebilir.
- Hata tip ayrımı özelliği oluşturulabilir, hata önem sırası özelliği eklenebilir. Örneğin “çok ciddi bir hata” veya “yazım hatası” şeklinde hata tipleri ve “hemen çözülmeli”, “acelesi yok”, “çözülmek için 3 saati var” tarzı hata çözüm zamanı kısıtlama özellikleri eklenebilir.
- Akıllı cihazlara hata inceleme, hatalardan sorumlu atama, hataya çözüm ekleme gibi işlemlerin yapılma olanağı eklenebilir.
- E-posta ile bildirimin yanında ilgili kişinin cep telefonuna kısa mesaj ile bildirim yapılabilir. Böylece çok ciddi projeler için zaman kazanımı sağlanmış olur.
- Sadece Java destekleyen telefonlardan başka, SmartPhone tarzı yeni nesil cep telefonları için PDA yazılımından uyarılma bir yazılım yapılabilir. Hatta Symbian destekli telefonlar için de bir yazılım geliştirilebilir.
- Web uygulamasını bir masaüstü programı olarak kullanabilmek amacıyla bir masaüstü yazılımı gerçekleştirilebilir. Hem Linux hem de Windows işletim sistemleri türevlerinde çalışabilmesi amacıyla bu yazılım Java ile gerçekleştirilebilir.
- Sorulara sadece Evet veya Hayır diye cevap vermekten başka türlü cevap verme imkânları sağlanabilir. Örneğin “bu aşamada çıkan yazıyı giriniz” tarzı sorular eklenebilir.
- Projelerle anlaşarak yazılımlara eklenti olarak ufak yazılımlar kurulabilir. Böylece bir hata olduğunda o yazılımın durumu, değişken değerleri, işletim sistemi bilgileri gibi bilgileri elde etme imkânı sağlanabilir.
- Firma isteklerine ve gereksinimlerine göre veritabanı değiştirilebilir. Örneğin Oracle, MySQL gibi veritabanları ile uyumlu çalışması için gereken tercüman nesneler gerçekleştirilebilir.
- Zaman kısıtlamalarından dolayı aradan soru çıkartma veya araya soru ekleme tarzı üst düzey işlem destekleri sağlanamamıştır, bu destek eklenebilir.
- Kullanıcı tipleri fazlalaştırılabilir, Takım Lideri, Yazılım Sorumlusu, Donanım Sorumlusu, Yazılım Mimarı, Testçi, Kalite Müdürü vb. tarzda kullanıcı tipleri için özel işlemler tanımlanabilir.
- Her kullanıcı tipi ayrı bir klasörde ayrı uygulama sayfaları gerçekleştirilerek bir sayfa üzerinde yapılan işlemler azaltılarak basitleştirme ve ileride güncelleme kolaylığı sağlanabilir.

İleride bu projeyi referans alacaklara tavsiyeler şunlar olacaktır:

- Her zaman için daha önce yapılmış ürünlere ve kaynaklara göz gezdirmek, fikir edinmek açısından yararlı olacaktır. Daha önce yapılmamış bir şeyler yapmak, hem insanlığa daha faydalı olacak hem de yığinsal olarak büyüyen bilime katkı yapmanızı sağlayacaktır. Yeni bir şeyler üretmek, mühendislik eğitiminizi pekiştirecek ve devamlı yeni şeyler öğrendiğiniz meslek hayatınıza katkısı büyük olacaktır.
- Bir projeyi modelleyip tasarlamak yazılım projelerinde en önemli adımdır. Projeyi modelleyip tasarımını yapmadan kod yazmaya girişmek çoğunlukla çok zorlanmaya sebep olur ve çok büyük zaman kayıpları ile sonuçlanır. Tasarımda yapılan bir hatayı düzeltmek ile gerçekleştirme anında yapılan bir hatayı düzeltmek arasında uçurum mevcuttur. Tasarım yanlış veya eksik olduğunda o hatayı düzeltmek gerçekleştirme anında çok zordur ama gerçekleştirme anı hatası kolayca düzeltilebilir. Sorunun iyi analiz edilmesi, çözümün kişinin mesleki mühendislik becerileri ile ortaya konulmasında çok yardımcı olur.
- Eğer ki yüksek düzeyli uygulamalar gerçekleştiriyorsa her zaman hazır bileşenler, nesneler veya örnekler mevcuttur. Bunların iyi araştırılması ile kişi tekerleği yeniden icat etme gereği duymaz ve zaman kazanımı ile daha gelişmiş, kaliteli ürünler gerçekleyebilir.
- Gelişen teknoloji iyi takip edilmeli, her zaman son durağan sürümlü yazılımlar kullanılmalıdır. En yüksek durağan sürümlü yazılımlar yapılan işi kolaylaştırmaya ve ek imkânlar sunmaya elverişlidirler. Bazen “keşke şu da olsa” denilen bir şeyin zaten en son sürümde mevcut olduğu görülebilir. Gelişen teknoloji her zaman daha kolay, daha hızlı ve daha kaliteli yapılabilen ürünler için imkânlar sunmaktadır.
- Ayrıca her yazılım devamlı gelişmektedir, hiçbir zaman bir yazılım kusursuz ve tam değildir, her zaman eklenebilecek veya çıkarılabilecek bir şeyler olabilir. Yazılım geliştirme süreci uzundur ve gerçekten kaliteli yazılımların kısa sürede tamamlanmaları çok rastlanan bir olay değildir.

8 KAYNAKLAR

- [1] Fog Creek Software, FogBugz, <http://www.fogcreek.com/FogBugz/> , son ziyaret tarihi: 2006-05-15
- [2] Mantis Bug Tracker, <http://www.mantisbt.org/> , son ziyaret tarihi: 2006-05-15
- [3] Bugzilla, <http://www.bugzilla.org/installation-list/> , son ziyaret tarihi: 2006-05-15
- [4] Three-Tier Mimari Nedir?,
http://www.godoro.com/Divisions/Ehil/Mahzen/Programming/ThreeTierArchitectural/txt/html/document_Definition.html , son ziyaret tarihi: 2006-05-17
- [5] Three-tier (computing), Wikipedia: http://en.wikipedia.org/wiki/Three_tier , son ziyaret tarihi: 2006-05-17
- [6] Üç katmanlı mimariye genel bakış,
http://upload.wikimedia.org/wikipedia/en/6/66/Overview_of_a_three-tier_application.png , son ziyaret tarihi: 2006-05-17
- [7] N-Tier Web Applications using ASP.NET 2.0 and SQL Server 2005 - Part 1, <http://www.15seconds.com/issue/050721.htm> , son ziyaret tarihi: 2006-05-18
- [8] Windows Mobile, Wikipedia, http://en.wikipedia.org/wiki/Windows_Mobile_5.0 , son ziyaret tarihi: 2006-05-18
- [9] GNATS, The GNU Bug Tracking System, <http://www.gnu.org/software/gnats/> , son ziyaret tarihi: 2006-05-19