

BAŐKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

JAVA TEKNOLOJİLERİ İLE BİR MİLLETVEKİLİ
DANIŐMANLIK VE ZİYARETÇİ TAKİP SİSTEMİ
GELİŐTİRİLMESİ

ERMAN NECİP BAŐ

YÜKSEK LİSANS TEZİ

2008

**JAVA TEKNOLOJİLERİ İLE BİR MİLLETVEKİLİ
DANIŞMANLIK VE ZİYARETÇİ TAKİP SİSTEMİ
GELİŞTİRİLMESİ**

**DEVELOPING A DEPUTY CONSULTANCY
AND VISITOR TRAILING SYSTEM
WITH JAVA TECHNOLOGIES**

ERMAN NECİP BAŞ

Başkent Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin
İSTATİSTİK ve BİLGİSAYAR Bilimleri Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.
2008

KABUL VE ONAY SAYFASI

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma, jürimiz tarafından **İSTATİSTİK ve BİLGİSAYAR BİLİMLERİ ANABİLİM DALI** 'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan :

Prof.Dr. Timur KARAÇAY

Üye (Danışman) :

Yrd.Doç.Dr. Güvenç ASLAN

Üye (Var ise Eş Danışman) :

Yrd.Doç.Dr. Harun ARTUNER

ONAY

Bu tez 05/06/2008 tarihinde, yukarıdaki jüri üyeleri tarafından kabul edilmiştir.

05/06/2008

Prof.Dr. Emin AKATA

FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRÜ

TEŐEKKÜR

Çalıřmalarım boyunca göstermiř olduđu sabır, anlayıř, yardımlarından ötürü ve ayrıca bilgi iletiřim alanında sahip olduđu tecrübeleri benimle paylařtıđı için, danıřmanım olan Yrd. Doç. Dr. Güvenç Arslan'a teőekkürü bir borç bilirim.

Hayatım boyunca, içinde bulunduđum çalıřmalarda göstermiř olduđu destek ve ilgiden ötürü, annem Nilüfer CINCİK' a teőekkür ederim.

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Erman N. BAŞ

Signature: _____

ÖZ

JAVA TEKNOLOJİLERİ İLE BİR MİLLETVEKİLİ DANIŞMANLIK VE ZİYARETÇİ TAKİP SİSTEMİ GELİŞTİRİLMESİ

Erman Necip BAŞ

Başkent Üniversitesi Fen Bilimleri Enstitüsü

İstatistik ve Bilgisayar Bilimleri Anabilim Dalı

Günümüz teknoloji çağında bilgisayar kullanımı artık bir ihtiyaç haline gelmiştir ve bilgisayar her alanda özellikle de iş hayatımızda bizlere büyük kolaylık sağlamaktadır. Ülkemizde, özel sektörün yanı sıra kamu kuruluşlarında da işlemler (dokümantasyon, yazışma, bilgi paylaşımı vs.) bilgisayarlar aracılığı ile yürütülmektedir. Bilgisayar kullanımının bu kadar yaygınlaştığı ülkemizde son yıllarda bu gelişime ayak uyduran en önemli kurumlardan birisi de TBMM (Türkiye Büyük Millet Meclisi) dir. TBMM'nin üyesi olarak mecliste görev yapan milletvekillerine yakın bir geçmişte diz üstü bilgisayar dağıtımı ile, hayata geçirilen e-meclis projesi kapsamında milletvekillerinin ve aynı zamanda danışman ve sekreterlerinin bilgisayar ve interneti daha etkin kullanmaları amaçlanmış ve kurslar düzenlenmiştir. Aynı proje kapsamında her milletvekilinin kendi adına siteleri oluşturulmuştur.

Bu çalışmada, ziyaretçi takip sistemleri incelenmiş; TBMM milletvekillerinin ve danışman ve sekreter gibi çalışanların ziyaretçilere yönelik her türlü kaydı daha sağlıklı bir şekilde tutmaları amaçlanmış ve de pratik, hızlı, hizmet standartları yüksek, yeni teknolojilerden faydalanılmış bir uygulama yazılımı geliştirmek hedeflenmiştir.

ANAHTAR SÖZCÜKLER: TBMM, Milletvekili, Ziyaretçi, Talep, Arama, Java, JSF (Java Server Faces), Webservice, Bilişim, Teknoloji

Danışman: Yrd.Doç.Dr. Güvenç ARSLAN, Başkent Üniversitesi, İstatistik ve Bilgisayar Bilimleri Bölümü

ABSTRACT

DEVELOPING A DEPUTY CONSULTANCY AND VISITOR TRAILING SYSTEM WITH JAVA TECHNOLOGIES

Erman Necip BAS

Başkent University Institute of Science

The Department of Statistics and Computer Science

In the time of the present day, the usage of computer becomes necessity anyway and computer provides us great ease in every field especially in the business life. In our country, operations (documentation, correspondence, info sharing etc.) are implemented through computers also in the public organization in addition to the private sector. In our country in which the usage of computer becomes so widespread TGNA (Turkish Grand National Assembly) is one of the most important organizations which keeps up with this development in recent years. Of recent date with the allocation of laptops to the deputies who officiate in assembly as members of TGNA it was aimed that the deputies and also their consultants and secretaries to use computers and internet more effective and courses were ordered within the context of e-assembly project which is implemented. Within the context of same project for each deputy his/her own website was formed.

In this work the existing visitor trailing program is analysed; by the deputies of TGNA and the employee such as consultant and secretary about visitors to keep all sorts of records more healthy is aimed and also improving an implementation software which is functional, speedy, with high-level service standards and taken the advantage of new technology is planned.

KEYWORDS: TGNA, Deputy, Visitor, Request, Searching, Java, JSF, Webservice, Informatics, Technology

Consultant: Yrd.Doç.Dr. Güvenç ARSLAN, Başkent University, The Department of Statistics and Computer Science

İÇİNDEKİLER LİSTESİ

| | <u>Sayfa</u> |
|------------------------------------------------------------------------------------------|--------------|
| ÖZ | i |
| ABSTRACT | ii |
| İÇİNDEKİLER LİSTESİ | iii |
| ŞEKİLLER LİSTESİ | v |
| SİMGELER VE KISALTMALAR LİSTESİ | vi |
| 1 GİRİŞ | 1 |
| 1.1 Genel Bakış..... | 2 |
| 2 ZİYARETÇİ TAKİP SİSTEMLERİ | 3 |
| 2.1 Ziyaretçi Takip Sistemlerinin Genel Yapısı | 4 |
| 2.2 TBMM’de Ziyaretçi Takip Sistemi..... | 7 |
| 2.3 Milletvekili Danışmanlık ve Ziyaretçi Takip Sistemi Süreç Analizi Diyagramı | 9 |
| 3 JAVA TEKNOLOJİLERİ | 10 |
| 3.1 JSF Teknolojisi ve Avantajları | 10 |
| 4 MİLLETVEKİLİ DANIŞMANLIK VE ZİYARETÇİ TAKİP SİSTEMİ YAPISI | 23 |
| 4.1 Proje Genel Yapısı | 23 |
| 4.2 Proje Dosyaları ve Sayfaları | 25 |
| 4.3 Ziyaretçi Takip Yetki Sistemi | 29 |
| 4.3.1 Admin | 31 |
| 4.3.2 Milletvekili | 31 |
| 4.3.3 Danışman ve sekreter | 31 |
| 4.4 Milletvekili Danışmanlık ve Ziyaretçi Takip Sistemi Ekranları ve Sınıfları | 32 |
| 4.4.1 Kullanıcı giriş ekranı | 32 |
| 4.4.2 Ziyaretçi ekranı | 33 |
| 4.4.3 Ziyaretçi kayıt ekranı | 33 |
| 4.4.4 Ziyaretçi arama ekranı..... | 34 |
| 4.4.5 Bugün? ekranı | 35 |
| 4.4.6 Not ekleme ve görüntüleme ekranı | 36 |
| 4.4.7 Kullanıcı tanımlama ekranı..... | 37 |
| 4.4.8 Sol menü | 38 |
| 4.4.9 SessionControl sınıfı | 40 |
| 4.4.10 SessionBean1 sınıfı..... | 41 |

| | | |
|----------|------------------------------------------------------|-----------|
| 4.4.11 | Functions.java sınıfı..... | 43 |
| 4.5 | Veri Tabanı Tabloları ve Alanları..... | 44 |
| 4.5.1 | Kişi tablosu..... | 44 |
| 4.5.2 | Liste tablosu..... | 46 |
| 4.5.3 | Meslek tablosu..... | 47 |
| 4.5.4 | Adres tablosu..... | 47 |
| 4.5.5 | Adres listesi tablosu..... | 48 |
| 4.5.6 | Davet tablosu..... | 49 |
| 4.5.7 | Görüşen kişi tablosu..... | 50 |
| 4.5.8 | İlçe tablosu..... | 51 |
| 4.5.9 | İl tablosu..... | 51 |
| 4.5.10 | Kullanıcı tablosu..... | 52 |
| 4.5.11 | Hareket tablosu..... | 53 |
| 4.5.12 | Parti tablosu..... | 54 |
| 4.5.13 | İşlem tablosu..... | 55 |
| 4.5.14 | Programa katılım tablosu..... | 56 |
| 4.5.15 | Randevu tablosu..... | 57 |
| 4.5.16 | Tebrik listesi tablosu..... | 58 |
| 4.5.17 | Vekil tablosu..... | 59 |
| 4.6 | WebService Tarafından Kullanılan Örnek Sorgular..... | 59 |
| 5 | SONUÇ, TARTIŞMA VE ÖNERİLER..... | 63 |
| 5.1 | Sonuç..... | 63 |
| 5.2 | Tartışma ve Öneriler..... | 64 |
| | KAYNAKLAR LİSTESİ..... | 66 |

ŞEKİLLER LİSTESİ

| | <u>Sayfa</u> |
|-------------------------------------------------|---------------------|
| Şekil 3.1 Client-Server ilişkisi..... | 12 |
| Şekil 3.2 MVC SUN Mic..... | 15 |
| Şekil 3.3 MVC..... | 17 |
| Şekil 3.4 JSF – MVC..... | 18 |
| Şekil 3.5 Faces-config.xml yapısı..... | 20 |
| Şekil 3.6 Managed Bean yapısı..... | 21 |
| Şekil 4.1 Sorgular..... | 26 |
| Şekil 4.2 Sorguların kullanıldığı proje..... | 27 |
| Şekil 4.3 Projede kullanılan sayfalar..... | 28 |
| Şekil 4.4 Tüm paketlerin oluşturduğu bütün..... | 29 |
| Şekil 4.5 Kullanıcı giriş ekranı..... | 30 |
| Şekil 4.6 Hata ekranı..... | 31 |
| Şekil 4.7 Kullanıcı giriş ekranı..... | 32 |
| Şekil 4.8 Ziyaretçi ekranı..... | 33 |
| Şekil 4.9 Ziyaretçi kayıt ekranı..... | 34 |
| Şekil 4.10 Ziyaretçi arama ekranı..... | 35 |
| Şekil 4.11 İşlem bugün? ekranı..... | 36 |
| Şekil 4.12 Notlar.jsp ekranı..... | 37 |
| Şekil 4.13 Kullanıcı tanımlama ekranı..... | 38 |
| Şekil 4.14 Sol menü kod yapısı..... | 39 |
| Şekil 4.15 Sol menü ekran görünümü..... | 40 |
| Şekil 4.16 Sessioncontrol Class..... | 41 |
| Şekil 4.17 Kişi tablosu..... | 46 |
| Şekil 4.18 Liste tablosu..... | 47 |
| Şekil 4.19 Meslek tablosu..... | 47 |
| Şekil 4.20 Adres tablosu..... | 48 |
| Şekil 4.21 Adres listesi tablosu..... | 49 |
| Şekil 4.22 Davet tablosu..... | 50 |
| Şekil 4.23 Görüşen kişi tablosu..... | 51 |
| Şekil 4.24 İlçe tablosu..... | 51 |
| Şekil 4.25 İl tablosu..... | 52 |
| Şekil 4.26 Kullanıcı tablosu..... | 53 |
| Şekil 4.27 Hareket tablosu..... | 54 |
| Şekil 4.28 Parti tablosu..... | 55 |
| Şekil 4.29 İşlem tablosu..... | 56 |
| Şekil 4.30 Programa katılım tablosu..... | 57 |
| Şekil 4.31 Randevu tablosu..... | 58 |
| Şekil 4.32 Tebrik listesi tablosu..... | 58 |
| Şekil 4.33 Vekil tablosu..... | 59 |
| Şekil 4.34 Web Metodlar..... | 60 |

SİMGELER VE KISALTMALAR LİSTESİ

| | |
|--------|-------------------------------------|
| API | Application User Interface |
| CRM | Customer Relationship Management |
| EJB | Enterprise Java Beans |
| ERP | Enterprise Resource Planning |
| GUI | Graphical User Interface |
| HTML | Hypertext Markup Language |
| IOC | Inversion Of Control |
| J2EE | Java 2 Enterprise Edition |
| J2SDK | Java 2 Standard Development Kit |
| J2SE | Java Standard Edition |
| JAF | JavaBeans Activation Framework |
| JavaEE | Java Enterprise Edition |
| JAX | Java API XML |
| JAXR | Java API for XML Registries |
| JB | JavaBean |
| JDBC | Java Database Connectivity |
| JMS | Java Message Service |
| JNDI | Java Naming and Directory Interface |
| JS | Java Servlets |
| JSF | Java Server Faces |
| JSP | Java Server Page |
| JTA | Java Transaction API |
| MVC | Model View Controller |
| RPC | Remote Procedure Call |
| SA | System Admin |
| SOA | Service Oriented Architecture |
| TBMM | Türkiye Büyük Millet Meclisi |
| XHTML | XML Hypertext Markup Language |
| XML | Extensible Markup Language |
| XUL | XML User Interface Language |
| URL | Uniform Resource Locator |

1 GİRİŞ

TBMM’de, meclis yasama ve denetim faaliyetlerinin önemli bir kısmının internet üzerinden yapılarak; zaman, mekan, insan ve maliyetlerden tasarruf sağlanması amacıyla ile "Meclis’te Avrupa Birliği Reformu" Projeleri kapsamında yer alan e-meclis projesi hayata geçirilmiştir. Söz konusu proje ile milletvekilinin seçmeni ile daha iyi iletişim kurması, sivil toplum kuruluşları ve vatandaşların yasama faaliyetlerine daha fazla katılım sağlamaları amaçlanmıştır.

Kendisini teknolojik gelişmelere adapte etmekte olan TBMM’de milletvekillerinin ziyaretçileri ile daha etkin ve yakından ilgilenebilmesi de önem arz etmektedir. Bu amaçla milletvekillerinin kendilerinin, danışman ve sekretelerinin kullanacakları bir programa gereksinim duyulmaktadır.

Bu çalışmada, milletvekiline ait geniş kapsamlı bir ekran ile danışmanı ve sekreterinin farklı kullanıcı adı ve şifre ile ulaşabilecekleri sınırlı bir ekran olmak üzere iki ayrı kullanıcı ara yüzüne sahip; milletvekillerinin, danışmanlarının ve sekreterlerinin kullanabileceği bir uygulama programı geliştirilmiştir. Programda, Meclise gelen konuk, misafir veya ziyaretçilerin özet kimlik bilgilerinin ilk olarak sekreter ya da danışman tarafından girileceği ardından da ziyaretçinin kısaca talebi, talep konusu, gerekli açıklamalar, yapılan işlem ve sonucu, ziyaretçi ile ilgilenen kişi, (Milletvekili, Sekreter, Danışman), taleple ilgili yapılacak işlem gibi temel alanların girilebileceği ekranlar tasarlanmıştır. Çalışma, JAVA-Web tabanlı bir uygulamadır ve Meclis Bilgi İşlem Dairesi’nde bulunan mevcut Oracle veri tabanı ile entegre olacak şekilde tasarlanmıştır. Bu çalışmada, günümüzde yazılım sektöründe yeni çıkan JSF 1.2 visual framework kullanılmıştır. Kullanıcı adı ve şifresi olmayanların sisteme erişimi engellenmiştir.

Hâlihazırda TBMM’de milletvekillerinin ziyaretçiler ile ilgili yaptıkları kayıtların tutulduğu programın işleyişi incelendiğinde birçok sorunla karşılaşılmıştır. Ziyaretçi arama sayfaları, var olan kayda hızlı erişim imkânı, milletvekiline ait not ekranı vb. içeriklerin olmaması; programın özellikle sekreterler ve danışmanlar tarafından kullanılmadığı gözlemlenmiştir.

Bu sorunların ortadan kaldırılması, ancak kayıtların sisteme taşınması ile gerçekleşebileceği için, öncelikle mevcut sistemlerin analizi yapıldıktan sonra bu yapıya uygun bir yazılım geliştirilmesiyle mümkün olacaktır.

1.1 Genel Bakış

Bu bölüm teze bir genel bakış sunmaktadır.

Bölüm 2, piyasada bulunan mevcut ziyaretçi takip paket programları ve TBMM’de kullanılan programın incelenmesi, sorunları gibi konulara değinilmiştir.

Bölüm 3 ise, Java teknolojileri hakkında detaylı bilgi vermektedir. Bu bölümde Java teknolojilerinin yapısı, kodların çalışma şekli, Java veritabanı bağı ve JSF, WebService* hakkında detaylı bilgi verilmiştir.

Bölüm 4, bu çalışma sonucunda ortaya çıkarılmış olan “Milletvekili Danışmanlık ve Ziyaretçi Takip Sistemi”ni tanıtmaktadır. Milletvekili Danışmanlık ve Ziyaretçi Takip Sistemi’nin milletvekili, danışmanlar ve sekreterler tarafından nasıl kullanıldığı, sistemdeki yetkilendirmelerin nasıl yapıldığı, kaydın nasıl tutulduğu, sistemin yapısının tanımı ve aşamaları bu bölümde tartışılmıştır.

Son olarak bölüm 5’te ise, bu çalışma sonunda ortaya çıkan sistemin kazandırdığı yenilikler, karşılaşılan zorluklar ve çözümler, ve son olarak da ileriye yönelik yapılması gereken çalışmalara değinilmiştir.

* Bir ağ üzerinde makine-makine birlikteliğini destekleyebilen yazılım sistemi

2 ZİYARETÇİ TAKİP SİSTEMLERİ

Türkiye’de son yıllarda müşteri memnuniyeti, hizmet kalitesi gibi deyimler hem hanehalkının hem de üreticinin en az ürün kadar değer verdiği kavramlar haline gelmiştir. Bir ürünü üretip halkın beğenisine sunmaya hazırlanan üretici aynı zamanda ürünün kullanımı ile başlayacak olan hizmetin kalitesi ile de yakından ilgilenmektedir.

Kalite; bir ürünün ifade edilen veya beklenen ihtiyaçları karşılama kabiliyetini oluşturan özelliklerin toplamıdır (TS 9005 – ISO 8402), kullanıma uygunluktur (Dr. J.M. Juran), ihtiyaçlara uygunluktur (P.B. Crosby). Ölçemediğiniz şeyi yönetemezsiniz, geliştiremezsiniz. Maliyet, sonuç, çalışan memnuniyeti, müşteri memnuniyeti vs. ölçülebilen parametrelerdir. Kalite aynı zamanda müşteri memnuniyetidir. ¹

İster özel sektör, ister kamu kurumu olsun bir organizasyonun kendi çalışanını, müşterisini takip etmek istemesi, misyonu* gereği ortaya koyduğu stratejik planının bir parçasıdır. Gelişen teknoloji sayesinde insan kaynağını da müşteriye de etkin bir şekilde takip etmek kolaylaşmıştır.

Hanelerin %18.94’ünün internete erişim imkanının olduğu ülkemizde hanehalkı bireylerinin bilgisayar ve internet kullanım oranları sırasıyla %29.46 ve %26.67’dir.²

Bilgisayar ve internet kullanımının gün geçtikçe arttığı ülkemizde, özel sektörün ağırlıklı olarak benimsediği, son yıllarda kamu kurum ve kuruluşlarının da

¹ Tokatlıoğlu, Mücella, temel kalite kavramları ve toplam kalite yönetimi, IQ uluslararası kalite danışmanlık A.Ş.

<http://www.tkgm.gov.tr/turkce/dosyalar/diger%5Cicerikdetaydh323.ppt#1>, 09 Mayıs 2008

* **Misyon**, (mission) bir organizasyonun varlık nedeninin; müşterileri, karşılanan ihtiyaçları ve kullanılan teknolojiyi de belirtecek şekilde bir cümle ile ifade edilmesidir. (insan kaynakları yönetimi terimleri sözlüğü, <http://www.rcbdoor.com/makalevekitaplar/insankaynaklari.htm>), 13 Nisan 2008

²TÜİK (20 Kasım 2007) Haber Bülteni 2007 Yılı Hanehalkı Bilişim Teknolojileri Kullanımı Araştırması Sonuçları, Sayı: 186, (<http://www.tuik.gov.tr/PreHaberBultenleri.do?id=605>)

önemmediği kaliteli hizmet anlayışı çerçevesinde hizmetleri belli bir standartta sağlamak amacıyla birçok veri dijital ortama taşınmaktadır.

Özel firmaların ve şirketlerin yanısıra kamu kurum ve kuruluşlarında da insan kaynağı yönetimi özellikle e-devlet çalışmaları ve bu alanda geliştirilen projelerle daha da önem kazanmaktadır³.

E-Devlet, kamu yönetiminde her türlü faaliyetin vatandaşlar, işletmeler, kamu çalışanları ve devletin diğer fonksiyonlarıyla ilgili etkileşimlerinin elektronik ortam üzerine taşınması, birebir ve aracısız olarak kullanılabilir hale gelmesidir⁴. Artık günümüzde birçok kamu kurum ve kuruluşu e-devlet yapısını benimsemiştir.

2.1 Ziyaretçi Takip Sistemlerinin Genel Yapısı

Eski yıllardan günümüze hemen her yerde (okul, hastane, kamu binası, şirket, vs.) bina girişinde bulunan, binaya girenlerin basitçe kaydının tutulduğu bir ziyaretçi defteri görmek mümkündür. Ziyaretçi defterleri ucuz ve kullanımı kolay defterlerdir. Ancak toplanan veri eksik, okunması zor ve verinin analizi imkansızdır. Ve daha da önemlisi organizasyonunuzu kimin ziyaret ettiği gibi mahrem bir bilginin herkes tarafından ulaşılabilir olmasıdır. Yıllar önce kabul edilebilir bu tür sakıncalar günümüzde en başta güvenlik açısından tehlike oluşturmaktadır⁵.

Bu kaygı ile Güney Afrika'da bir firma tarafından geliştirilen 'secured management systems' sayesinde ziyaretçilerin ID'leri (işyeri kartı, ehliyet, vs.) elektronik olarak taranmakta ve onlar hakkındaki ilgili bilgileri (fotoğraflarını, isteğe bağlı olarak imzalarını, ziyaret edilen kişinin adı ve soyadını, ziyaret sebebini, giriş ve çıkış saatlerini) amniyetli veri tabanı dosyasına aktarmaktadır. 20 saniye veya daha az

³ KOCABAL Ahmet (<http://www.caginpolicisi.com.tr/26/16-17-18-19.htm>, POLNET ve EMNİYET BİLGİ SİSTEMİ,)

⁴ Erdal, M. (2008) Elektronik Belediye Kavramı ve İstanbul Büyükşehir Belediyesi Uygulaması, <http://66.102.1.104/scholar?hl=tr&lr=&q=cache:e-LO9TTm2fUJ:www.edevlet.net/eTurkiye/ebelediye.pdf+ziyaret%C3%A7i+takip+program%C4%B1>, 11 Nisan 2008

⁵ <http://www.procurex.co.za/>, 07 Mayıs 2008,

sürede ziyaretçinin profesyonel olarak kontrolü yapılmakta ve yüksek kaliteli ziyaretçi kimlik kartı otomatik olarak basılmaktadır.

Firma; organizasyonun güvenliğinin daha sağlıklı bir biçimde sağladığı, ziyaretçiye ait detaylı bilgileri daha doğru ve otomatik olarak tuttuğu, ziyaretçiye ve organizasyona ait bilgilerin mahremiyetinin korunduğu, çalışanlara ve ziyaretçilere profesyonel bir kart verildiği ve de mevcut kontrol sistemlerine kolay entegrasyon sağladığı gerekçeleri ile birçok küçük-büyük, özel-kamu organizasyonların güncelliğini kaybetmiş olan ziyaretçi kayıt defterlerini 'secured managemeny systems' ile yer değiştirdiğini belirtmektedir. Programı kullanan organizasyonlardan bazıları; Adidas Amerika, Fox News, Mitsubishi Motors, Panasonic, VISA International, Canon, Tivoli Systems/IBM, Phizer, Roche, UNICEF, Coca Cola, Pepsico, Shell Oil ve birçok üniversite, kamu kurum ve kuruluşları.

Ülkemizde de benzer yapıda birçok ziyaretçi takip programları mevcuttur. Kullandıkları teknolojiler bakımından farklılıklar gösteren mevcut programlar genel olarak binanın güvenliği kaygısı ile hazırlanmıştır. Bina içerisinde bulunan ziyaretçinin giriş ve çıkış kontrolünün yapıldığı, aynı zamanda da ziyaretçiye ait bilgilerin tutulduğu programlara göz atacak olursak:

- Ziyaretçi Bilgileri, Ziyaretçi Kartı Takibi, Giriş - Çıkış Raporları, Günlük Rapor Mailleri, Yasaklı Ziyaretçi Takibi, Kargo, Kurye Takibi özelliklerine sahip program sayesinde hem bina içerisinde bulunanlar, binadan ayrılanlar gibi giriş ve çıkışlara ait bilgilere ulaşmak; hem de detaylı olarak ziyaretçiye ait bilgiler, yasaklı ziyaretçiler gibi bilgilere ulaşmak mümkündür.

Program, iki tarih arasında sorgulama yaparak ziyaretçi listesine ulaşma imkanı sağlamaktadır. Aynı program sayesinde ziyaretçilere ait kart tanımlanması ve bina içerisinde ziyaretçinin takibi yapılabilmektedir. Görevli personel, kullanıcı, yasaklı ziyaretçilerin tanımlanabildiği alanlar da

mevcuttur. Kargo gönderimi takibi yapmak da mümkün olup raporları Excel'e aktarım kolaylığı sağlamaktadır⁶.

- Hangi ziyaretçinin belirli tarih ve saatte kimi ziyaret ettiğinin ayrıntılı olarak raporlandığını görebilmek mümkündür. Sistemde tanımlı kartlar ile entegre edilerek ziyaretçilerin hangi kapılardan geçebileceklerini (geçiş kontrol sistemleri) kontrol altına alma ve raporlama imkanı sağlamaktadır⁷.
- Ziyaretçi modülünde ziyaretçinin adı soyadı, hangi firmadan geldiği, şehri, hangi kimlik kartı verildiği, ziyaret için kime geldiği ve bölümünün tanımlanabildiği, yetkili olduğu saat veya tarih aralığının girebildiği bir programdır. Aynı program sayesinde alınan raporlarla kime ne kadar ziyaretçi geldiğinin istatistiğine de kolayca ulaşma imkanı bulunmaktadır⁸.
- En çok ziyaret edilen kişi ve en çok ziyaret eden ziyaretçi tipi, günlük ziyaret sayısı, bölümlerin ziyaret edilmiş oranları; ziyaretçilerin ne zaman, kimi, ne sebeple ziyarete geldiğini, departmanlar bazında istatistiki olarak ziyaret eğilimlerini, ziyarette bulunan kişilerin bilgilerini kullanım kolaylığı ile çok kısa zamanda ziyaretçinin dahi hissedemeyeceği hızla kayıt yapılabildiğini görmek mümkündür⁹.
- Web tabanlı bir başka programı incelediğimizde, kuruluşa ait ziyaretçilerin geçmişe dönük, şimdiki zamana göre ve geleceğe dönük tüm hareketleri izlenebildiğini görebiliriz. Personel internet ve intranet ortamı üzerinden randevu kaydı yapabilir, ziyaretçi kuruluşa geldiği zaman görevlinin personele ulaşma zorunluluğunu ortadan kaldırır. Personel, web üzerinde randevu kaydını çok basit işlemlerle yapabilir. Personel, kendi ziyaretçilerini

⁶ http://www.ritmateknoloji.com/ziyaretci_takip_programi.asp, 10 Nisan 2008,

⁷ http://www.personeltakip.com/ziyaretci_gecis_sistemi.asp, 10 Nisan 2008

⁸ http://www.erben.com.tr/prd_detail.asp?id=48, 10 Nisan 2008

⁹ www.kozbilisim.com/index.php?option=com_content&task=view&id=29&Itemid=1, 10 Nisan 2008,

online olarak takip edebileceği bir modül kullanır. Bu modül sayesinde kuruluşa randevusuz gelen ziyaretçileri takip edebilir. Personel web üzerinden ziyaretçi kaydını yapınca görevli kişi ekranında randevu kaydını görebilir. Personel, sadece görevlinin okuması için mesaj ekleyebilir ¹⁰.

Programlar incelendiğinde görüldüğü üzere hemen hemen hepsinde yapı birbirine benzerlik göstermektedir. Verilerin tutulduğu alanlar, girişlerin yapıldığı ekranlar farklılık göstermekle beraber özet olarak hemen hepsinde ziyaretçiye ait detaylı bilgi (kimlik bilgileri, iletişim bilgileri, vs.), kimi ziyaret ettiği, hangi tarihte ve hangi saat aralıklarında bina içerisinde bulunduğu bilgilerine ulaşmak mümkündür.

2.2 TBMM'de Ziyaretçi Takip Sistemi

TBMM, 'Mecliste AB Reformu' Projeleri kapsamında yer alan e-meclis projesinin hayata geçmesi ile birlikte elektronik ortamda bilginin paylaşımı ve takibi anlamında büyük yol almıştır. Yasama ve denetim faaliyetlerinin önemli bir kısmının internet üzerinden yapıldığı, milletvekilinin halkla internet üzerinden iletişim kurduğu, her türlü güvenliğin elektronik ortamda diğer programlarla entegre edilerek sağlandığı meclisimizde ziyaretçi takibinin yapıldığı bir programın da kullanılmakta olduğu gözlemlenmiştir.

Piyasadaki örnekler ve TBMM'de kullanılan program incelenmiştir. Mecliste milletvekilleri, sekreterleri ve danışmanlarının kullanımına sunulan ve de son bir yılda programın etkin kullanımıyla ortaya çıkan ihtiyaçların karşılanması; ayrıca gelişen teknolojiye ayak uydurarak veri denetimi ve güvenliğinin sağlanması, kaliteli ve hızlı hizmet anlayışının benimsenmesi, meclisteki diğer programlara kolay entegre edilebilmesi ve ihtiyaçlara yönelik olarak geliştirilebilmesi açısından Java Web uygulamalı bir program geliştirilmesi ihtiyacı doğmuştur.

Meclis tarafından kullanılan program ile milletvekilini ziyarete gelen ziyaretçinin kaydı yapılabilmekte, meclis binasına güvenli olarak giriş yapabilmesi açısından

¹⁰ http://www.icsltd.com.tr/pk_ziyaretci.aspx, 11 Nisan 2008,

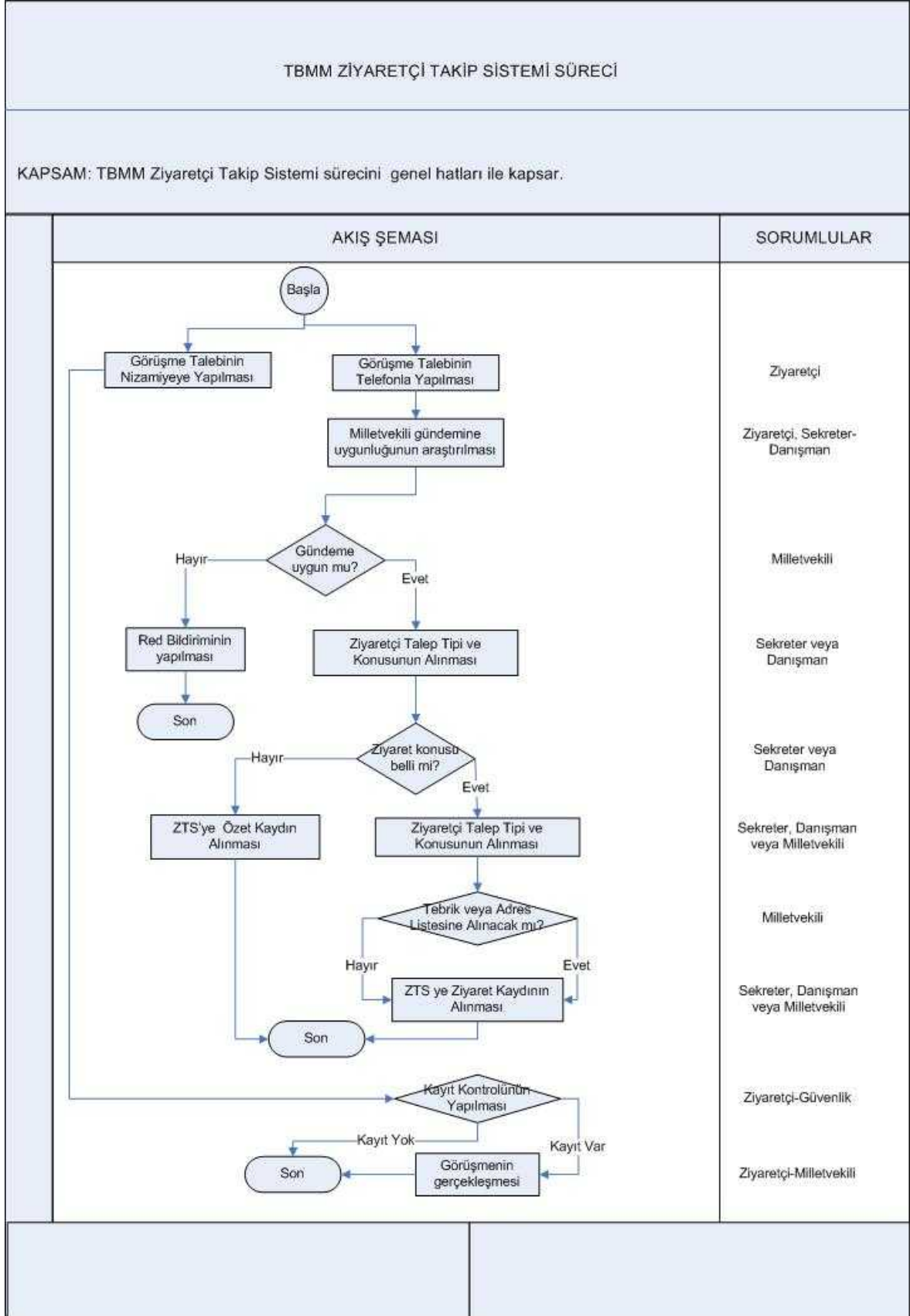
bilgileri (eğer araç ile geleceksen plaka numarası) TBMM giriş kapısına aktarılabilir. Program ile kayıtlar günlük olarak tutulmakta ve de ileriye veya geçmişe yönelik herhangi bir sorgulama yapılamamaktadır. Giriş kapısındaki kullanıcıya iletilecek olan bilgiler için farklı kullanıcı adı ve şifresi ile yeniden oturum açılmakta ve aynı bilgiler tekrar kaydedilmektedir. Programın, kullanıcı bilgisayarların herbirine ayrı kurulumunun yapılması gerektirir. Ayrıca milletvekili tarafından da programın kullanmadığı gözlemlenmiştir. Özellikle bu eksikliklerden yola çıkarak yeni bir uygulama geliştirilmiştir.

İleride geçiş kontrol sistemine entegre edilebilecek bir yapıya sahip olan sistem, Meclisin ihtiyaçları da göz önünde bulundurulduğunda ziyaretçiye ait basit bir kaydın tutulduğu bir sistem olmaktan da öte; ziyaretçilerin kişisel, iletişim bilgilerinin ve istek/talep/randevu kayıtlarının yapılabildiği, kayıtlı ziyaretçilerin görüntülenebildiği, hızlı ve özet ziyaretçilerin eklenebildiği ve kayıtlı ziyaretçi bilgilerinin güncellenebildiği ve de randevu, davet, tebrik listesi, adres listesi, katılacak programlar kategorilerinde zaman aralığına, adı-soyadı, davet yeri, randevu ismi vb. gibi alanlarda detaylı aramanın yapılabildiği, günlük randevu ve davet, tebrik, telefon vb. işlerin listesine ulaşabilme imkanının bulunduğu bir yapıya sahiptir.

Ayrıca uygulamanın web tabanlı olması özellikle kullanılan teknoloji bakımından birçok avantaj sağlamaktadır. Aynı zamanda sistemin web tabanlı olmasından dolayı sistem, her milletvekillinin bilgisayarına ayrı kurulum yapmayı gerektirmemektedir.

TBMM’de önceden kullanılan programla karşılaştırdığımız zaman birçok avantajını gördüğümüz “Milletvekili Danışmanlık ve Ziyaretçi Takip Sistemi”, piyasadaki mevcut ziyaretçi programları ile karşılaştırıldığında ise bir paket program olmaktan öte TBMM ihtiyaçlarına yönelik özel olarak hazırlanmış olması açısından büyük bir avantaj sağlamaktadır.

2.3 Milletvekili Danışmanlık ve Ziyaretçi Takip Sistemi Süreç Analizi Diyagramı



3 JAVA TEKNOLOJİLERİ

3.1 JSF Teknolojisi ve Avantajları

Bu bölümde J2EE (Java 2 Enterprize Edition) nin ne olduğu ve nerelerde kullanıldığına değinildikten sonra JSF detaylı bir şekilde anlatılmaktadır.

J2EE, Java'nın çok katmanlı kurumsal uygulama geliştirme standartıdır. J2EE, standart ve modüler bileşenlerden oluşan (ERP-Enterprice Resource Planning-, CRM -Customer Relationship Management- vb.) bir yapının ve çok sayıda makine, sunucu, veritabanı ve uygulamadan oluşan bir ortamın uyumlu çalışması için bir alt yapı oluşturmaktadır.

J2EE'nin içerdiği teknolojiler¹¹:

- XML (Extensible Markup Language) tabanlı RPC (Remote Procedure Call) (JAX -Java Api XML- / RPC) için Java API (Application User Interface)
Farklı makinelerdeki nesnelerin birbirleriyle XML konuşarak haberleşmesini sağlamaktadır.
- JSP (Java Server Page) ve JS (Java Servlet)
Web programlama teknolojilerini içermektedir.
- EJB (Enterprice Java Beans) Bileşenleri
Nesnelerin veritabanında saklanmasını ve networkteki herhangi bir makineden erişilmesini sağlamaktadır.
- J2EE Connector Mimarisi
Başka tekonolojilerle geliştirilmiş sistemlerin J2EE ile birlikte çalışmasını sağlamaktadır.
- J2EE Management Modeli
Network yönetim sistemleri ve protokolleriyle entegrasyon sağlamaktadır.

¹¹http://www.godoro.com/divisions/ehil/mahzen/Java/Faqs/txt/html/faq_J2EE.html

- J2EE Deployment API
J2EE bileşenlerinin geliştirilmesi ve sisteme yüklenmesini sağlamaktadır.

- XML Registries (JAXR - Java API for XML Registries-) için Java API

WebServislerinin oluşturulmasını, kurulmasını ve erişilmesini sağlamaktadır.

- JMS (Java Message Service)
Uygulamaların ve nesnelerin eşzamansız (asen kron) iletişimini sağlamaktadır.

- JNDI (Java Naming and Directory Interface) Nesnelerin belli bir isimle veya belli bir dizinde saklanmasını sağlamaktadır.

- JTA (Java Transaction API)
Transactional veri erişimini sağlamaktadır.

- CORBA
Java dışında C++ gibi diğer dillerin de nesneleriyle iletişim kurabilmesini sağlamaktadır.

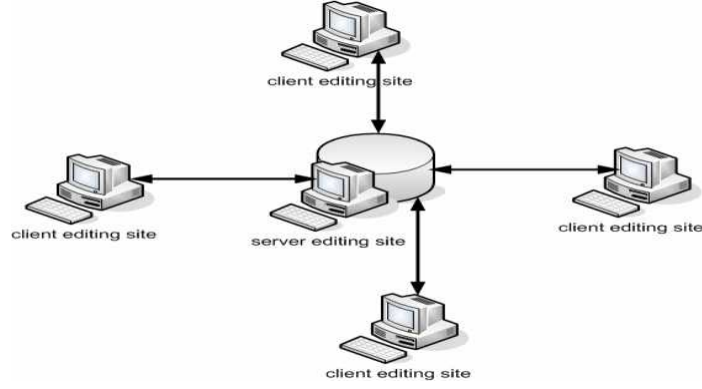
- JDBC (Java Data Base Connectivity) data access API
SQL Veritabanlarına erişimi sağlamaktadır.

- Java Mail
Mail sunucularına erişip mail alma/vermeyi sağlamaktadır.

- JAF (JavaBeans Activation Framework)
Belli bir veri tipinde çeşitli komutların çalıştırılmasını sağlamaktadır.

- J2SE (Java Standart Edition)
Standart Java'daki her şeyin enterpriseda kullanılmasını sağlamaktadır.

İnternet kullanımının çok hızlı bir şekilde geliştiği günümüz teknoloji çağında web uygulamaları da geliştiriciler tarafından yaygın bir şekilde kullanılmaktadır. Söz konusu web uygulamalarından SUN firmasının çıkarmış olduğu JSF teknolojisi, diğer bir tanım olarak frameworkü bu uygulamalardan bir tanesidir. Web'in çalışma mantığı Şekil 3.1'de gösterildiği gibi istek tabanlıdır (request-based).



Şekil 3.1 Client-Server ilişkisi*

JSF teknolojisi, web uygulamaları geliştirirken, alışıldan (request-based) farklı bir bakış açısıyla görsel kullanım (component-based) ortaya çıkınca bilişim dünyasında kolaylık sağlamıştır. HTML (Hypertext Markup Language) etiketleri, artık kendi sınıfları olan birer bileşen haline gelmişlerdir. Uygulamamızın içerisinde bileşenlerin özelliklerini değiştirebilmekte, en önemlisi bileşenlerimize bağlayacağımız olaylar ile uygulamamızın akışını oluşturabilmekteyiz. Aslında bu masaüstü uygulamalarımızdan tanıdığımız mantıktır. JSF, bu mantığı web uygulamamızda da kullanmamızı sağlayacak yapıyı bizlere sunmaktadır. Aşağıda farklı kaynaklar incelenerek elde edilen veriler sunulmaktadır.

JSF, JavaEE (Java Enterprise Edition) uygulamalarında kullanıcı ara yüzlerinin gelişimini kolaylaştırmak için tasarlanmış bir Java tabanlı web uygulama frameworküdür. Diğer request-driven MVC web frameworklerinden farklı olarak JSF, parça bazlı yaklaşımı kullanmaktadır. Alıcı yeni sayfa talep ettiği zaman kullanıcı arayüzü parçaları saklanır ve talep yanıtlandığında geri yüklenir. Bunun dışında JSF, teknoloji görüntüsü için JSP kullanmakta ve aynı zamanda diğer teknolojileri (XUL - XML User Interface Language- gibi) birbirine uyarlarmaktadır.

JSF:

- Kullanıcı arayüzleri bileşenlerini ifade eden ve onları yöneten, olayları ve girdilerin geçerliliklerini işleyen, değerleri çeviren, sayfa

* http://wiki.eclipse.org/images/b/b7/Cola_client_server.png

dolaşımını tanımlayan ve uluslararası kontrol ve erişilebilirliği destekleyen bir API setini,

- JSP sayfasının içindeki JSF ara yüzünü belirtmek için iki tane JSP özel tag frameworkünü içermektedir¹².

Aynı zamanda:

- Hazır kullanıcı ara yüzü bileşenleri topluluğunu,
- Sunucu destekli olay modelini,
- Durum yönetimini,
- Hem JSP 2.0 hem de JSF 1.2 için Managed Beans Unified Expression Language içermektedir¹³.

JSF, Java geliştiricilerinin web tabanlı kullanıcı ara yüzleri inşa etmeleri için gereksinim duydukları tag frameworkleri ve standart API'ler sağlayan teknolojidir. Apache Struts frameworkünün yazarı Craig McClanahan, Sun'da JSF projesini önermektedir. Bu, popüler Apache projesinden standart JSF'ye basit bir göçü sağlamaktadır. JSF; Struts framework gibi, JavaBean'in özelliklerine bağlanabilen HTML yapılı unsurları meydana getiren bir JSP tag setini tanımlamaktadır. Geliştiricilerin açısından uygulamada iki framework de benzer yapıdadır ancak JSF bir Java standardı olduğu için muhtemelen araç geliştiricilerinden daha fazla destek alacaktır. Gelecekte bütün J2EE uygulama sunucuları JSF'yi desteklemeye gereksinim duyacaklardır¹⁴.

Java API ise Java yazılımlarında kullanılan yazılım frameworklerine genel olarak verilen isimdir. Java API ile disk, grafik, ağ, veri tabanı, güvenlik gibi yüzlerce

¹² SUN Microsystem, Java Server Faces Technology Overview, <http://java.sun.com/javaee/javaserverfaces/overview.html>, 05 Mayıs 2008

¹³ Cioroianu, Using JSF, http://www.onjava.com/pub/a/onjava/2003/09/03/using_jsf.html, 06 Mayıs 2008

¹⁴ Java Server Faces, http://en.wikipedia.org/wiki/JavaServer_Faces, 11 Mayıs 2008

konuda kullanıcılara erişim imkânı sunulmaktadır. Java API, J2SDK (Java 2 Standart Development Kit) nın bir parçasıdır¹⁵.

Kullanıcılar tarafından JSF sayfasından request (sunucuya yapılan istek) işlemleri sırasında kullanılan önemli API'ler aşağıda gösterilmektedir:

- FacesContext :Kullanıcının o andaki requesti ile ilgili tüm durum bilgisini içermektedir.
 - Application :Kullanıcının o andaki yaptığı Action, NavigationHandler vb. requestlerini içermektedir.
 - ExternalContext :Hızlı şekilde bileşenlere ve servlet bilgilerine ulaşabilmeye imkân sağlamaktadır.

Örnek;

```
ExternalContext externalContext = getFacesContext().getExternalContext();
HttpServletRequest request = (HttpServletRequest) externalContext.getRequest();
String errorPageURL = externalContext.getRequestContextPath();(sunucudaki yolu alıyor)
```

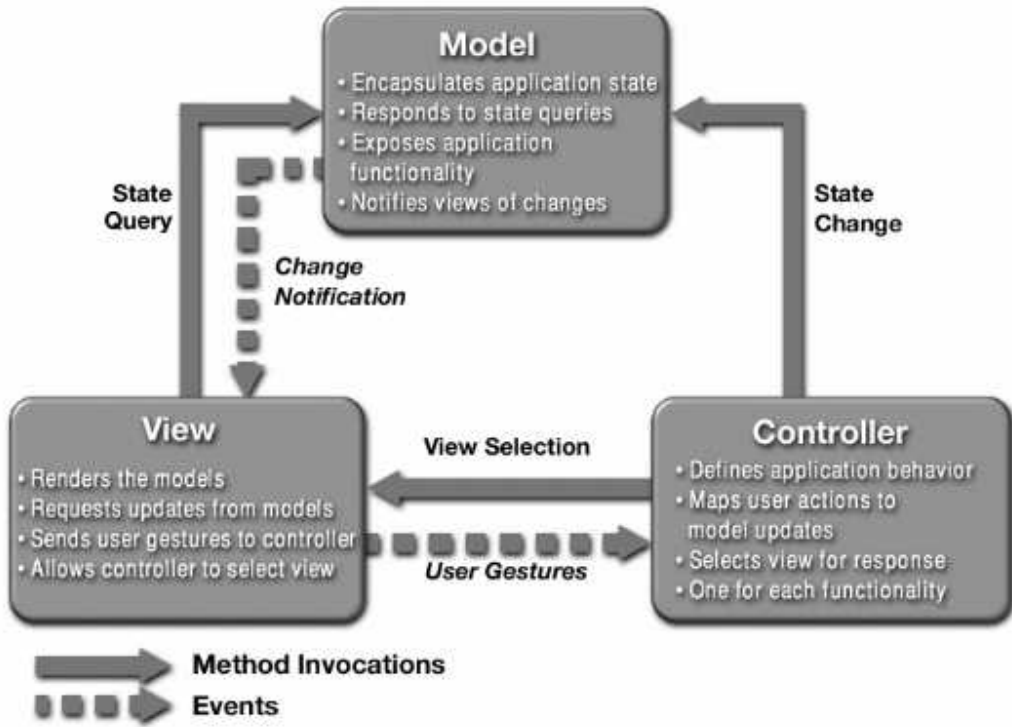
- ViewRoot :Bileşenlerin soyağacı yapısındaki bilgilerini almaktadır.
- MessageQueue
- RenderKit
- ResponseStream ve ResponseWriter
- getCurrentInstance :Kullanıcının o andaki isteğini almaktadır. Thread.currentThread() gibi.

- FacesMessage
- ResponseStream
- ResponseWriter
- FacesContextFactory :JSF tabanlı web uygulamaların hepsinde olan bu sınıf FacesContext instanceları içerir.

¹⁵ Akın, Java nedir, nasıl çalışır,

http://www.godoro.com/divisions/ehil/mecmua/Magazines/Articles/txt/html/article_WhatisJavaAndHowRun.html

JSF, MVC (Model View Controller) tasarım kalıbının gerçekleştirildiği bir yapıdır. Model(model), view(görünüm) ve controller(denetleyici) bileşenlerin baş harflerinin birleştirilmesi ile oluşturulmuştur. MVC yapısını anlamak JSF'nin alt yapısını anlamak ile eş değerdedir. MVC'nin çıkış mantığının temelinde büyük yazılımların kolay yönetilebilmesi yer almaktadır. Programların kontrolü tag framework yapılarına göre daha kolay sağlanmaktadır. Herbir bileşen, katman görevini oluşturmaktadır. Model katmanı, verilere ve servislere ara yüz görevini üstlenir ve uygulama verilerini değiştirme gibi olaylar ile ilgilenmez. Bu katmanın iş mantığında bussiness logic kısmı olan JBları (JavaBean) veya WebServiceleri vardır. Görünüm katmanı, JSP'ler veya HTML sayfaları gibi, kullanıcıların gördüğü katmandır. Kontrol katmanı ise uygulama içindeki istekleri görünüm katmanından alır ve yapılacak işlemlerden sonra model katmanına iletir.



Şekil 3.2 MVC SUN Mic.*

* http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/images/app-archa2.gif

MVC'yi oluşturan ve aynı zamanda katman görevi gören bileşenlerin fksiyonlarına göz atacak olursak:

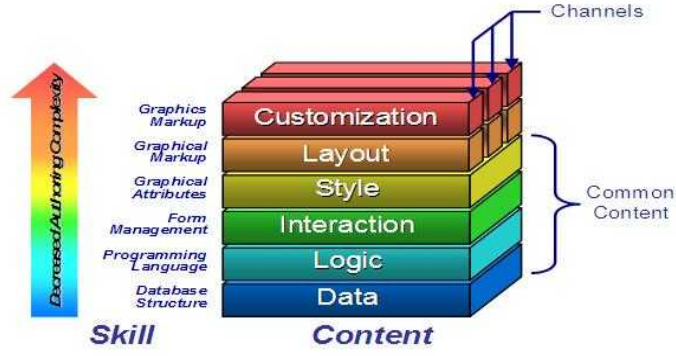
Model – model; güncelleme ve erişimini yönettiği kurum verisini ve iş kurallarını ifade etmektedir. Model genellikle dünyadaki uygulamalar için bir yazılım yaklaşımı gibi görev yapmaktadır, böylece modeli tanımlarken basit dünya modelleme teknikleri uygulanmaktadır.

View – görüntüleme; bir modelin içeriklerini göstermektedir. Modelin vasıtası ile kurumun verisine erişir ve verinin nasıl sunulması gerektiğini belirtir. Model değişirken gösterilişindeki tutarlılığı sürdürmek görüntülemenin sorumluluğundadır. Uyarıları değiştiren model ile görüntülemenin kendisini kaydettiği bir push model kullanarak veya görüntülemenin en geçerli veriye ulaşması gerektiği zaman modeli çağırmakla sorumlu olduğu bir pull model kullanarak bu başarılabilir.

Controller – denetleyici; etkileşimleri, görüntüleyici ile, model tarafından gerçekleştirilmesi gereken etkilere dönüştürür. Bir stand-alone GUI (Graphical User Interface) alıcıda Web uygulamalarında gibi görünen kullanıcı etkileşimleri düğme klik veya menü seçmeli olmalıdır¹⁶.

MVC mimarisi Şekil 3.3'te gösterildiği gibi birbiri ile ilişkili çeşitli katmanları içermektedir.

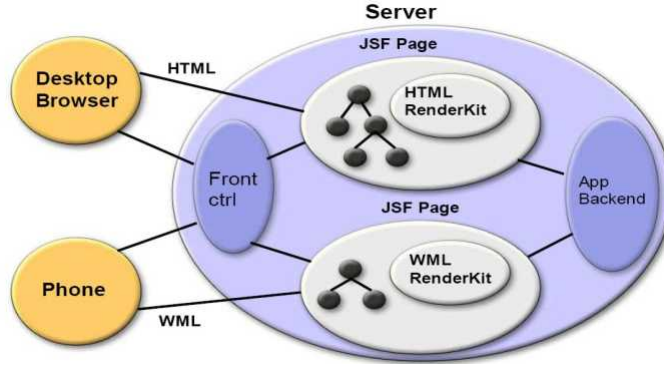
¹⁶ SUN Microsystem, Java BluePrints Model-View-Controller,
<http://java.sun.com/blueprints/patterns/MVC-detailed.html>, 05 Mayıs 2008



Şekil 3.3 MVC

- Customization (Kişiselleştirme) Katmanı : Görüntüleme katmanının parçasıdır. Bu kısım kullanıcının gördüğü ve hissedebildiği şekil, renk vb. kısımlardan oluşan yapıdır.
- Layout (Yerleşim) Katmanı : Görüntüleme katmanının parçasıdır. Sayfa organizasyonunun içeriği kısmından oluşan yapıdır.
- Style (Stil) Katmanı : Görüntüleme katmanının parçasıdır. Sayfa oluşumunu içeren stiller topluluğudur.
- Interaction (Etkileşim) Katmanı : Denetleyici katmanına ait bir yapıdır. Kullanıcı tarafından veri girişlerinin yapıldığı ve sayfalararası geçişler olan submit işlemlerinin (Form Bilgilerinin gönderilmesi) yapıldığı yapıdır.
- Logic (Mantık) Katmanı : Model katmanına ait yapıdır. Application veya business logic yapılarını içeren katmandır.
- Data (Veri) Katmanı : Model katmanına ait yapıdır. Sisteme ait veritabanı durumlarını içeren katmandır¹⁷.

¹⁷ www.formfaces.com/Web%20Forms%20Technology%20Comparison.doc



Şekil 3.4 JSF – MVC

MVC mimarisinden sonra JSF'nin yapısını inceleyek olursak JSF frameworkünü kullanabilmeye olanak sağlayan birçok paket bulunduğunu görebiliriz. JSF frameworkünü kullanabilmek için sayfa başında eklenmesi gereken isim uzayları mevcuttur. Bunlardan önemli olan başlıcaları aşağıda anlatılmaktadır.

JSF frameworkünün paketleri;

- **javax.faces :**

Bu paket JSF API'nin üst düzey sınıflarını içermektedir. Bu paketteki en önemli sınıf, kullanıcıların kendilerinin uygulamanın birçok anahtar parçalarına önem verdiği mekanizmadır. Her JSF uygulaması kendine ait bir FacesServlet'e sahiptir ve bu yapıları içeren bir frameworktür.

- **javax.faces.application:**

Bu paket, JSF tabanlı uygulamayı icra etmeyi yöneten, uygun düzenli çalışan mekanizmalara ek olarak uygulamanın business logic objelerini JSF'ye bağlamak için kullanılan API'leri içermektedir. Bu paketteki ana sınıf uygulamadır.

- **javax.faces.component :**

Bu paket kullanıcı ara yüzü bileşenlerinin önemli API'leri içermektedir.

- **javax.faces.component.html:**

Bu paket her geçerli bileşen birleşimleri için somut temel sınıfları içermektedir.

- **javax.faces.context:**

Bu paket her talep bilgisini tanımlayan sınıfları ve ara yüzlerini içermektedir. Bu paketteki ana sınıf, diğer çeşitli yardımcı sınıfların giriş yerinin yanı sıra, her talep bilgisinin tamamının erişim noktası olan FacesContext'tir.

- **javax.faces.convert:**

Bu paket, dönüştürücüleri tanımlayan sınıfları ve ara yüzlerini içermektedir. Bu paketteki ana sınıf 'Dönüştürücüdür'.

- **javax.faces.el:**

Bu paket, referans ifadeleri işleyen ve değerlendiren sınıfları ve ara yüzlerini içermektedir.

- **javax.faces.lifecycle:**

JSF uygulamasının yaşam döngüsü yönetimini tanımlayan sınıfları ve ara yüzlerini içermektedir. Bu paketteki ana sınıf 'Yaşam döngüsüdür'.

- **javax.faces.event:**

Bu paket, olayları ve EventListener'ları tanımlayan ara yüzleri ve somut olay uygulama sınıflarını içermektedir. Bütün bileşen seviyeli olayları FacesEvent'tan ve bütün bileşen seviyeli listenerları FacesListener'dan sunmaktadır.

- **javax.faces.render:**

Bu paket sayfaya yükleme modelini tanımlayan sınıfları ve arayüzlerini içermektedir. Bu paketteki anasınıf spesifik alıcı düzen tipinin sayfaya yükleme kabiliyetini sağlayan Renderer örneklerini gösteren RenderKit.RenderKit tir.

- **javax.faces.validator:**

Arayüz, somut validator uygulama sınıflarını ve validator modelini tanımlamaktadır.

- **javax.faces.webapp:**

JSF'nin web uygulamalarına entegrasyonu için Standart bir servlet, JSP özel bileşen taglarının temel sınıflarını ve temel taglar için somut tag uygulamalarını içeren gerekli sınıflardır.

JSF'nin yapısını incelemeye devam edecek olursak JSF frameworkünü kullanabilmeye olanak sağlayan birçok paketin yanısıra diğer framework'ler gibi JSF kullanıcı arayüzü, veri ve application katmanlarından "ManagedBean" yapısı ile karşılaşırız. JSF tag'ları ile HTML kullanıcı arayüzü oluştuğu zaman, JSF JB Model'den bilgileri alır ve HTML formunu oluşturan kullanıcı ara yüzündeki bileşenlerin durumlarını oluşturur. Kullanıcı, bu oluşan formda onaylama yaptığı zaman eğer validaton kontrollerinden geçtiyse JSF, kullanıcının giriş yaptığı bilgileri JB'ye kaydeder.

JB sınıfının bir modeli olan PBean sınıfı, Java.io.Serializable isim uzayından türemektedir. JSF her bir kullanıcı için PBean oluşturmaktadır. Bu PBean sınıfında; metin, boyut, font, renk, dizin, kalın, italikve altı çizili metodları tutulmaktadır. Her kullanıcının uygulamaları için JSF sunucuda oturum açmaktadır. JSF bu uygulama özelliklerini PBean ID'si ile bir Faces-config.xml dosyası içinde tutmaktadır¹⁸.

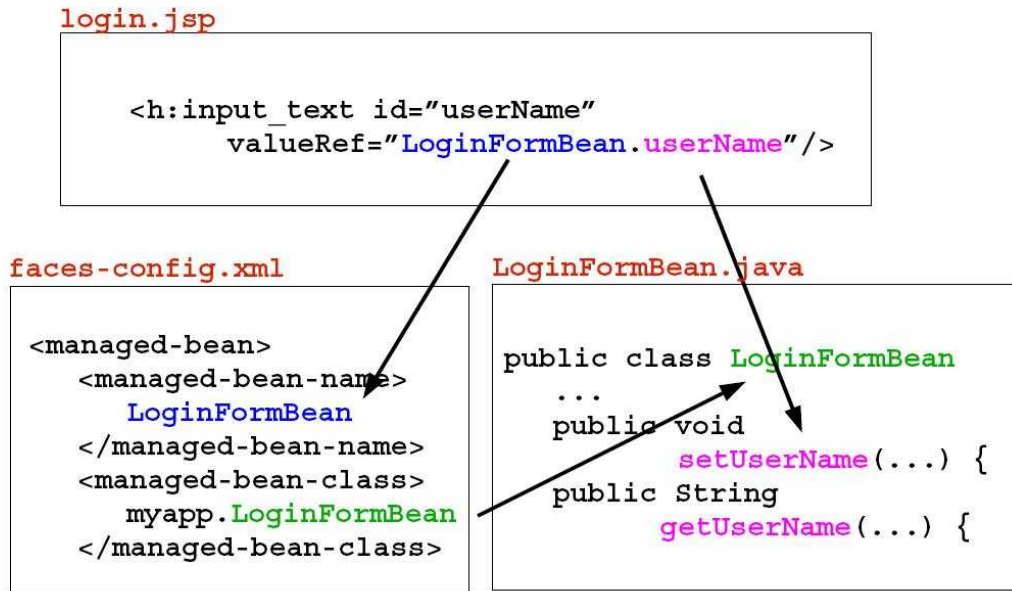
```
<faces-config>
.....
<managed-bean>
  <managed-bean-name>pbean</managed-bean-name>
  <managed-bean-class>
    com.devsphere.articles.usingjsf.PBean
  </managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
  <managed-property>
    <property-name>text</property-name>
    <null-value/>
  </managed-property>
  <managed-property>
    <property-name>size</property-name>
    <value>3</value>
  </managed-property>
  <managed-property>
    <property-name>font</property-name>
    <values>
      <value>Arial</value>
      <value>Courier New</value>
    </values>
  </managed-property>
  <managed-property>
    <property-name>color</property-name>
    <value>green</value>
  </managed-property>
  <managed-property>
    <property-name>align</property-name>
    <value>left</value>
  </managed-property>
  <managed-property>
    <property-name>bold</property-name>
    <value>>false</value>
  </managed-property>
  <managed-property>
    <property-name>italic</property-name>
    <value>>true</value>
  </managed-property>
  <managed-property>
    <property-name>underline</property-name>
    <value>>false</value>
  </managed-property>
</managed-bean>
</faces-config>
```

Şekil 3.5 Faces-config.xml yapısı

¹⁸ <http://www.keesbroenink.nl/doc/Java%20frameworks%20-%20middle%20layer.doc>, 0 5 Mayıs 2008

"ManagedBean", JSF ile doğan bir kavramdır. Çeşitli kaynaklarda "BackingBean" olarak geçmesine rağmen temelde aynı kavramı ifade etmektedir. Adından da anlaşılacağı üzere ortada bir Java bean'i vardır ve JSF kendi içindeki IOC (Inversion Of Control) konteyniri ile bu bean'in yaşam döngüsünü yönetmektedir. ManagedBean genelde pratik olarak, her sayfanın iş mantığını ve model nesnelerini içermektedir. Buna benzer bir dizaynı Asp.Net ve code behind sayfalarında da görebiliriz. Asp.Net'den farkı; bir sayfada birçok ManagedBean'in kullanılabilmesi ve ManagedBeanlerin Page gibi bir sınıfı eklemesini gerektirmemesidir.

Binding UI to Managed Bean



Şekil 3.6 Managed Bean yapısı

Diğer yandan, JSF gibi JSP üzerinde çalışan bir diğer framework ise 'struts' dır. Struts, Asp.Net gibi JSF bileşen tabanlıdır. Yani bir düğme yerleştirip bunun kliklenmesi ile sunucuda istenilen kod parçası çalıştırılabilmektedir. Dolayısıyla Request-Response döngüsü sırasında sayfa elemanlarının durumlarını korumak için de ekstra bir çaba harcanmasına gerek kalmamaktadır.

JSF, SUN mic. sayfasından ücretsiz olarak temin edebilecek netbeans, eclipse gibi platformlar üzerinde kullanılabilir. JSF; Java tabanlı web uygulamalarını kolaylaştırmak için MVC yapısına uygun olarak geliştirilmiş bir framework olması, iyi tanımlanmış yaşam döngüsü ile tek bir biçimde kod yapısının olması (sadelik), XHTML (XML Hypertext Markup Language) tabanlı olması, kod geliştirmede hızlı, sağlam ve güvenli olması, her gün gelişen bilişim sektöründe internetin hızla insanların hayatına yerleşmesi ve böylelikle web uygulamalarının gelişmesinin kaçınılmaz bir yer alması gibi nedenlerden dolayı yazılımcıların tercihi haline gelmiştir. Bunun yanında bu özelliklere yakın veya kapsayan platform olarak Microsoft'un sunduğu. NET teknolojisi bulunmaktadır. Fakat JSF'nin açık kaynak olması ve her platformda çalışabilmesi, bilişim firmalarının ve yazılımcıların tercihi sebebi olmuştur.

4 MİLLETVEKİLİ DANIŞMANLIK VE ZİYARETÇİ TAKİP SİSTEMİ YAPISI

TBMM’de kullanılan ziyaretçi programının incelenmesi sonucunda hem meclis kullanıcılarının ihtiyaçlarını karşılayabilecek hem de gelişen teknolojiye ayak uydurabilecek bir sistem geliştirilmiştir. Sistem, ileride isteklere karşı geliştirilebilecek yapıya sahiptir.

Günümüz teknolojisinin ilerlemesi ile birlikte yeni framewokler ve yeni işleyişler ortaya çıkmaktadır. Mevcut programların yeni teknolojilere ayak uyduramayacak durumda olması, diğer uygulamalar ile entegrasyonda sorun oluşturması, sürekli olarak kurulum gerektirmesi ve esnek olmaması gibi sorunlar sonucunda; yeni teknolojiler ile geliştirilmiş, tarayıcılarımız ile zahmet gerektirmeden ulaşabileceğimiz, diğer mevcut programlar ile kolaylıkla uyum sağlayabilecek yeni bir web tabanlı uygulama yapılmasını zorunlu hale getirmiştir.

4.1 Proje Genel Yapısı

Proje kapsamında kullanıcılar, TBMM vekilleri ve bu vekillere bağlı danışman ve sekreterler olarak tanımlanmıştır.

Sistem, kullanıcının ziyaretçiye ait kısa özet girişi yapmasına imkân sağladığı gibi istenildiği takdirde sistemde detaylı giriş de yapılabilmektedir. Ziyaretçinin talebi doğrultusunda randevu, davet, iş vb. gibi bilgiler tutulup takip edebilmektedir. Sistem ayrıca programda milletvekilinin kolaylıkla ulaşabileceği, günlük olarak not ekranı ve günlük iş listesi vb. uygulamaları içermektedir. TBMM Bilgi İşlem Dairesi tarafından gerekli alt yapı hazırlandığı takdirde, meclis içi ziyaretçi kontrolü ve ziyaretçilerin geçiş kontrol sistemlerine entegre edilebilecek alt yapıya sahiptir. Söz konusu entegrasyonun gerçekleşmesi durumunda; ilgili milletvekili sekreterini arayarak talebini bildiren ziyaretçinin gerekli kişisel bilgileri sisteme kaydedilecek, ziyaretçi kaydı yapılınc TBMM giriş kapılarında bulunan görevli kişi ekranında randevu kaydını detaylı olarak (hangi milletvekili ile hangi tarihte görüşeceği vb.) görebilecek, böylece randevusuz gelen ziyaretçiler takip edilebilecek, ziyaretçilere

sistem tarafından tanımlanarak verilecek olan ziyaretçi kartları sayesinde bina içerisinde bulunma süreleri, giriş ve çıkış saatleri kontrol edilebilecektir.

Genel olarak sistemin alt yapısı, sistemde hangi teknolojiler ve donanımlar kullanıldığı, servisler, veri tabanı bağlantıları vb. yapılar aşağıda detaylı olarak anlatılmaktadır.

Proje TBMM içerisinde iç-ağda çalışacaktır. İstenildiği takdirde dışardan erişilebilecektir.

Milletvekili Danışmanlık ve Ziyaretçi Takip Sistemi proje yapısı J2EE mimarisinde kurulmuş olup, JSP ve JSF kullanılmıştır. Application sunucu olarak Sun firmasının çıkarmış olduğu Sun Application Server 9.0 update 1 versiyonu kullanılmıştır.

Veri tabanı sorguları olarak, ortak bir yapıda kullanılabilmeye imkân tanıyan Webservice olan SOA (Service Oriented Architecture) teknolojisi kullanılmıştır. Bu sayede yazılan veri tabanı sorguları bir sunucuya yüklendiği takdirde istenilen yerde veya programda kullanılabilmeye imkân sağlamaktadır.

Uygulamanın geliştirilmesi için Netbeans IDE 6.0 kullanılmıştır. Sun firmasının çıkarmış olduğu platformun önemli özelliklerini; programcıya modelleme ve görsel düzenlemede kolaylık sağlaması, uygulama geliştirmede zaman kazandırması, çoğu plug-in leri kurulumu ile beraberinde getirmesi, dokümanları ve tutoriaları ile kolaylık sağlaması, ortaya çıkan yeni güncellemeler ve plug-in leri kolayca güncelleme ve eklemede program geliştiricilere kolaylık sağlaması olarak tanımlayabiliriz.

Veri tabanı olarak TBMM'de kullanılan mevcut veri tabanı ile uyum sağlaması açısından Oracle 10g seçilmiştir. Sorguların denenmesi ve çalıştırılması için yardımcı program olarak Sql developer programı kullanılmıştır. Sql developer, tablolara erişimde ve tabloların kolayca birleştirilmesinde, yazılan sorguların denenmesinde kolaylık sağlamaktadır. Tabloların ve kolon isimlerinin

oluřturulmasında diđer programlarda kullanılan isimler dikkate alınarak genel deđerler atanmıřtır.

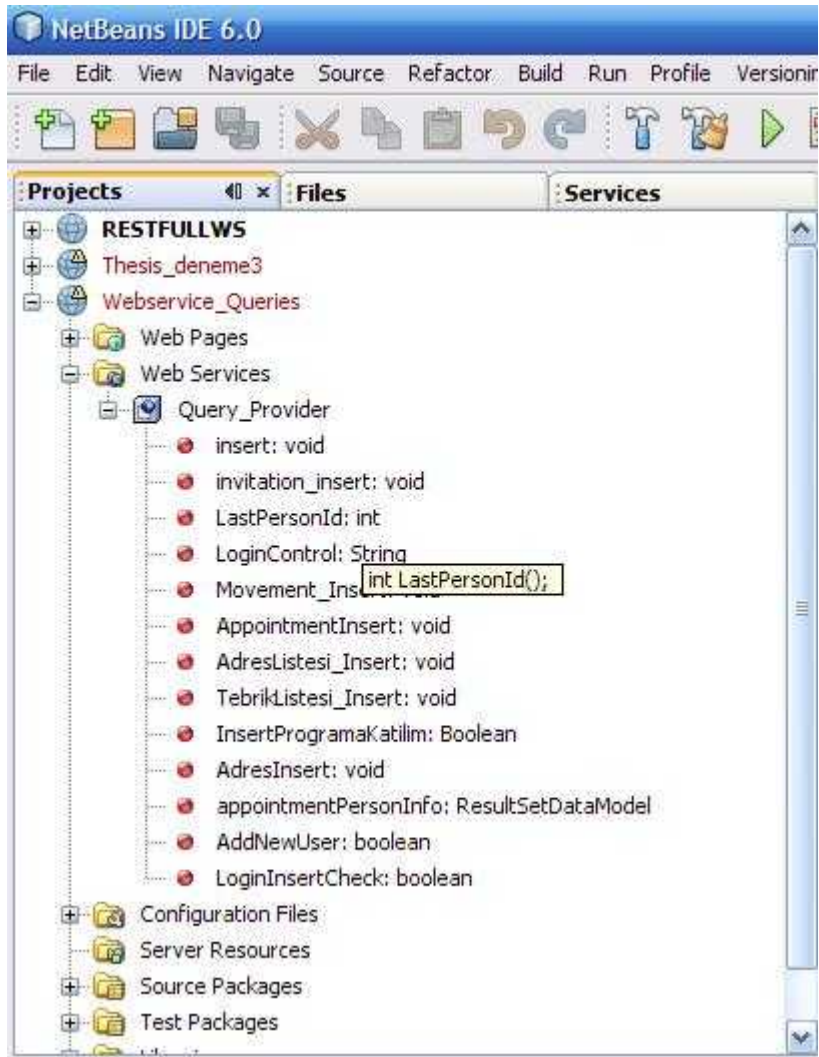
Veri tabanında kullanılacak sorguların bazı kısımlarında Webservice teknolojisinden faydalanılmıřtır. Bylece istenildiđi takdirde istenilen sorgular dıřardan bařka programlarda kullanılabilme imkânı sađlamaktadır. Diđer yandan silme ve gncelleme gibi kısımlarda kullanıcının yaptıđı iřlemin takibini sunmaktadır.

Veri tabanının genelinde kısaca; milletvekiline gelen ziyaretilere ait bilgiler, yapılacak iřlem eřidi, yapılacak iřlem eřidine gre aıklama ve diđer bilgilerin tutulması, ziyaretiye ait kimlik ve iletiřim bilgileri tutulmaktadır.

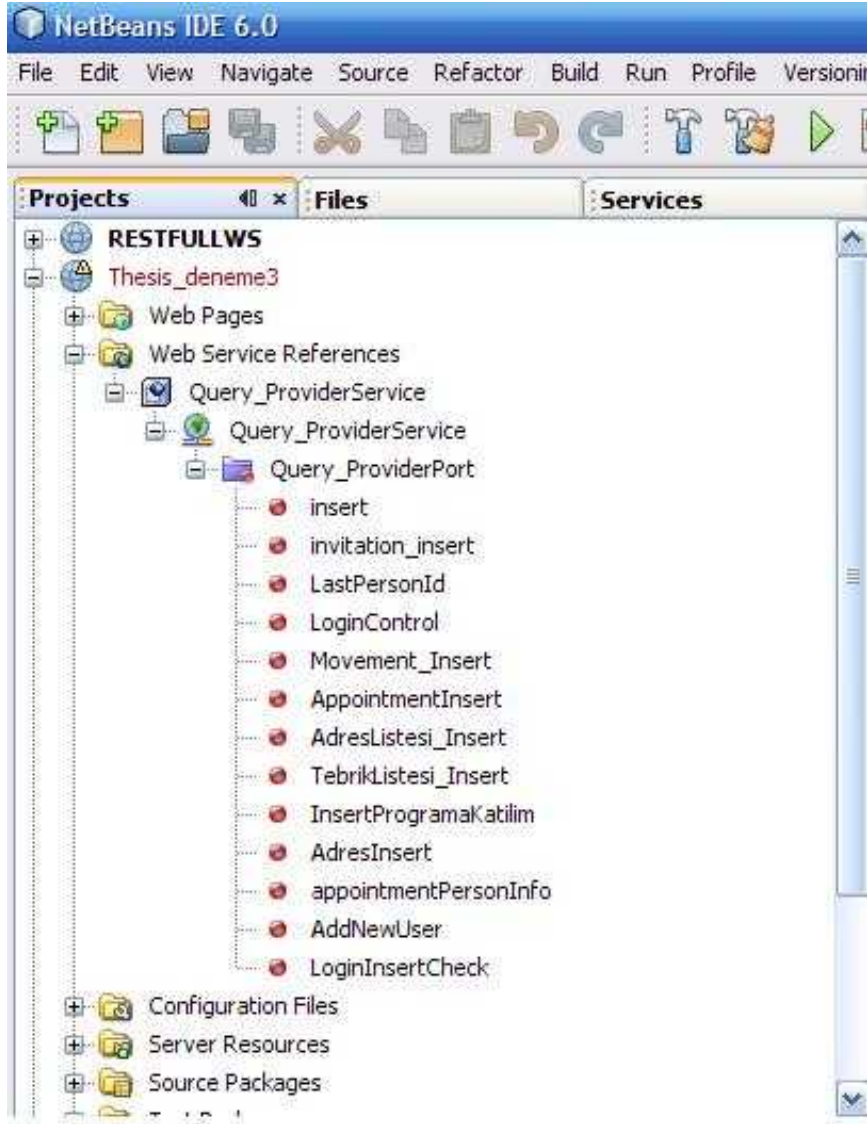
Program ierisinde kullanılan JSF'nin sunduđu temelde, master page mantıđı ile kullanıcı haklarını belirleme alanlarının oluřturulması iin sol men class'ı oluřturulmuřtur.

4.2 Proje Dosyaları ve Sayfaları

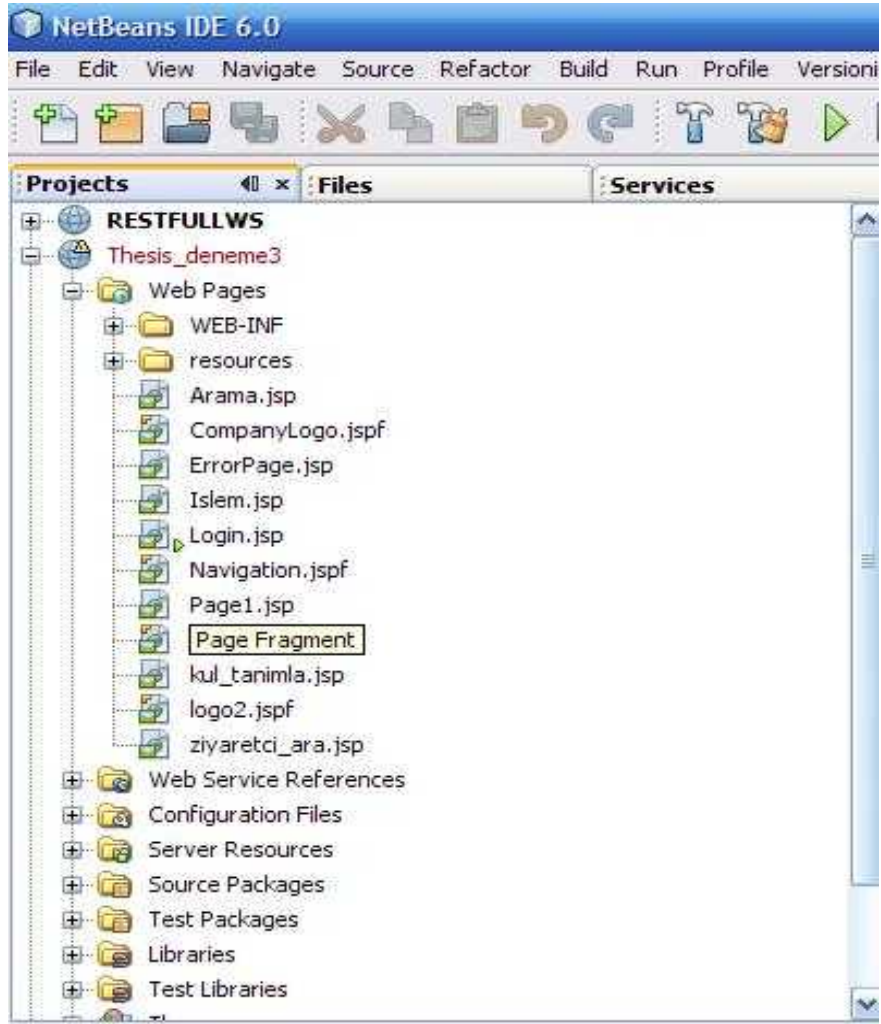
Sistemde 2 adet netbeans projesi kullanılmıřtır. Bu projelerden birincisi Webservice'deki sorguların bulunduđu Webservice-Queries, diđerisi ise bu servisin kullanıldıđı olan thesis_deneme projesidir. Thesis_deneme projesinin ierisinde ise Webservice references adlı paket bulunmaktadır. Bunu Webservice client olarak da adlandırabiliriz. Őekil 4.1, Őekil 4.2, Őekil 4.3 ve Őekil 4.4'te sorguları, sorguların kullanıldıđı projeyi, projede kullanılan sayfaları ve tm paketlerin oluřturduđu btn grebilirsiniz.



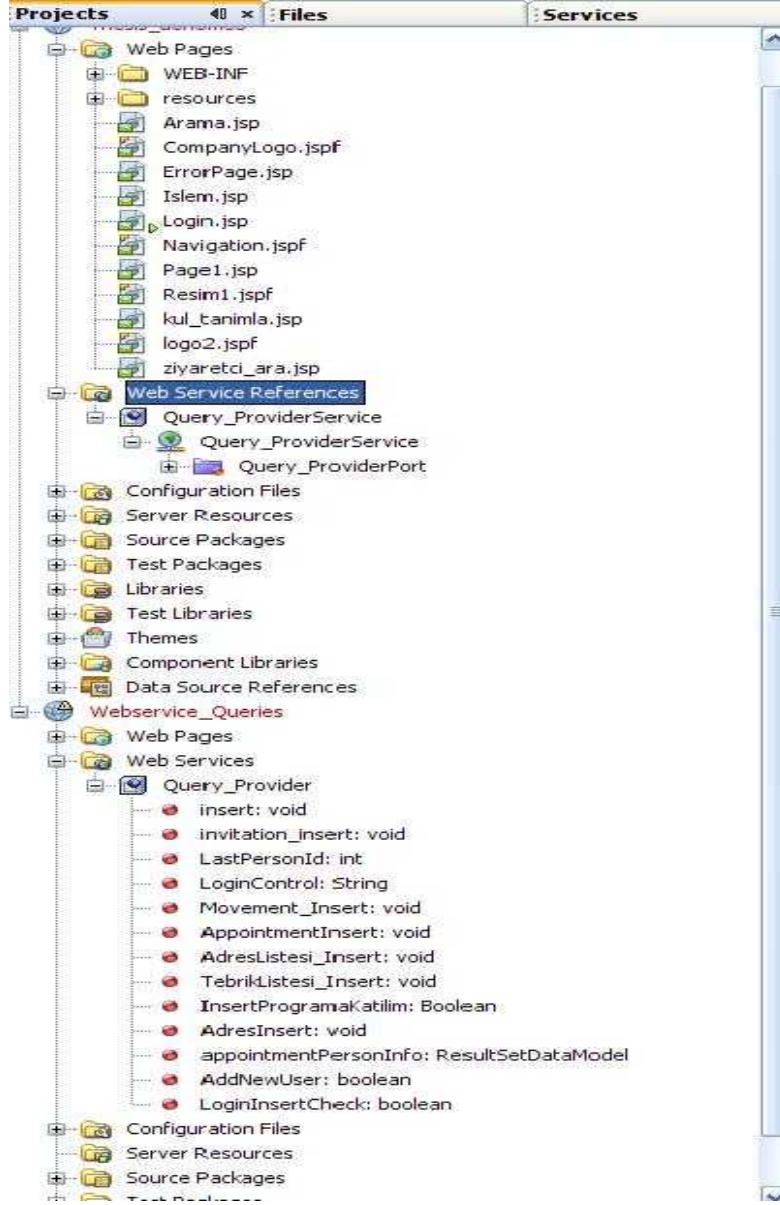
Şekil 4.1 Sorgular



Şekil 4.2 Sorguların kullanıldığı proje



Şekil 4.3 Projede kullanılan sayfalar

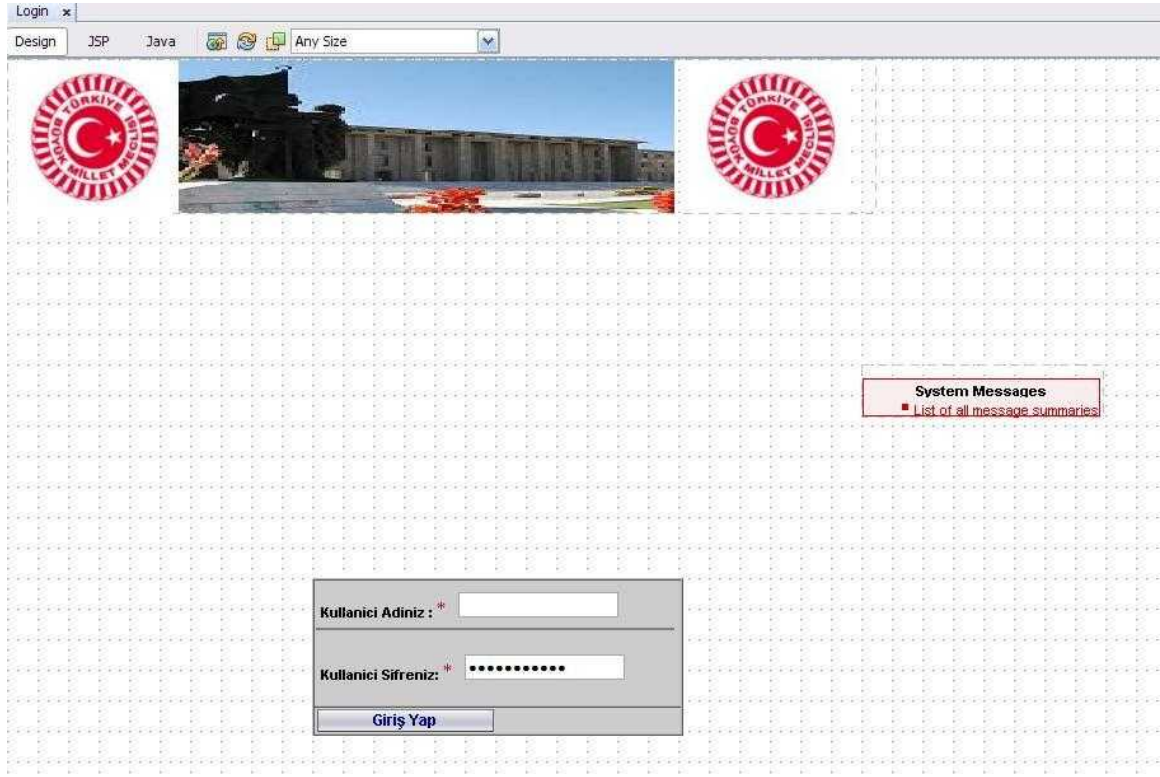


Şekil 4.4 Tüm paketlerin oluşturduğu bütün

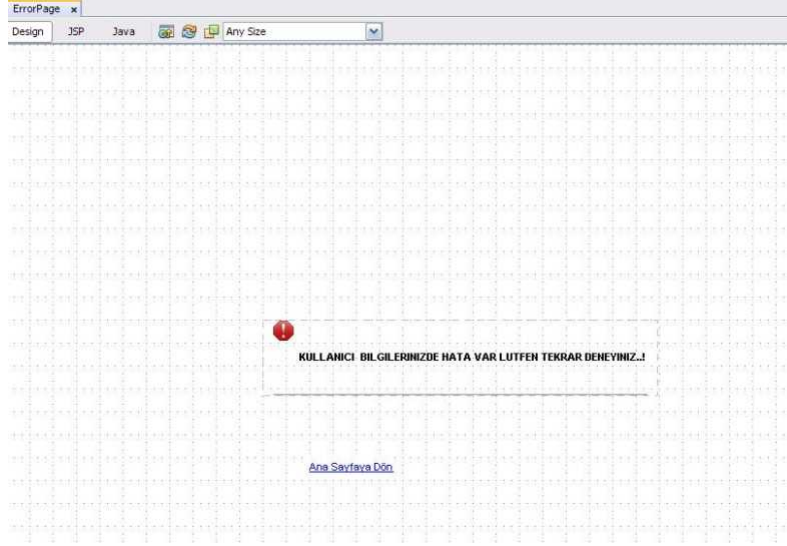
4.3 Ziyaretçi Takip Yetki Sistemi

Sisteme kaydı yapılan ziyaretçiler ile ilgili bazı işlemlerin veri güvenliğinin sağlanması için, alan ve ekran tabanlı olmak üzere kullanıcıya göre yetkilendirme yapılmıştır. Projede 4 değişik kullanıcı tipi bulunmaktadır. Böylece herkes üzerine düşen görevi yerine getirmektedir. Kullanıcılar; admin, milletvekili, danışman ve sekreterden oluşmaktadır. Şekil 4.5'deki ekranı kullanarak kullanıcılar Login.jsp'den kullanıcı adı ve şifrelerini yazarak girişlerini yapmaktadırlar. Kullanıcı

adı veya şifre bilgilerinden herhangi birinin yanlış girilmesi durumunda kullanıcı ErrorPage.jsp'ye yönlendirilmektedir.



Şekil 4.5 Kullanıcı giriş ekranı



Şekil 4.6 Hata ekranı

4.3.1 Admin

Admin tipindeki kullanıcı olarak, sadece TBMM Bilgi İşlem Dairesi kullanıcıları tanımlanmıştır. Sistemde, admin kullanıcı; tanımlama ekranına ulaşabilmekte, veriler üzerinde değişiklik yapabilmekte ve de silme yapabilmektedir.

4.3.2 Milletvekili

Milletvekili tipindeki kullanıcı, tanımlama ekranı dışındaki tüm ekranlara ulaşabilmektedir. Verileri değiştirebilme ve silebilme yetkilerine sahip kullanıcı tipidir.

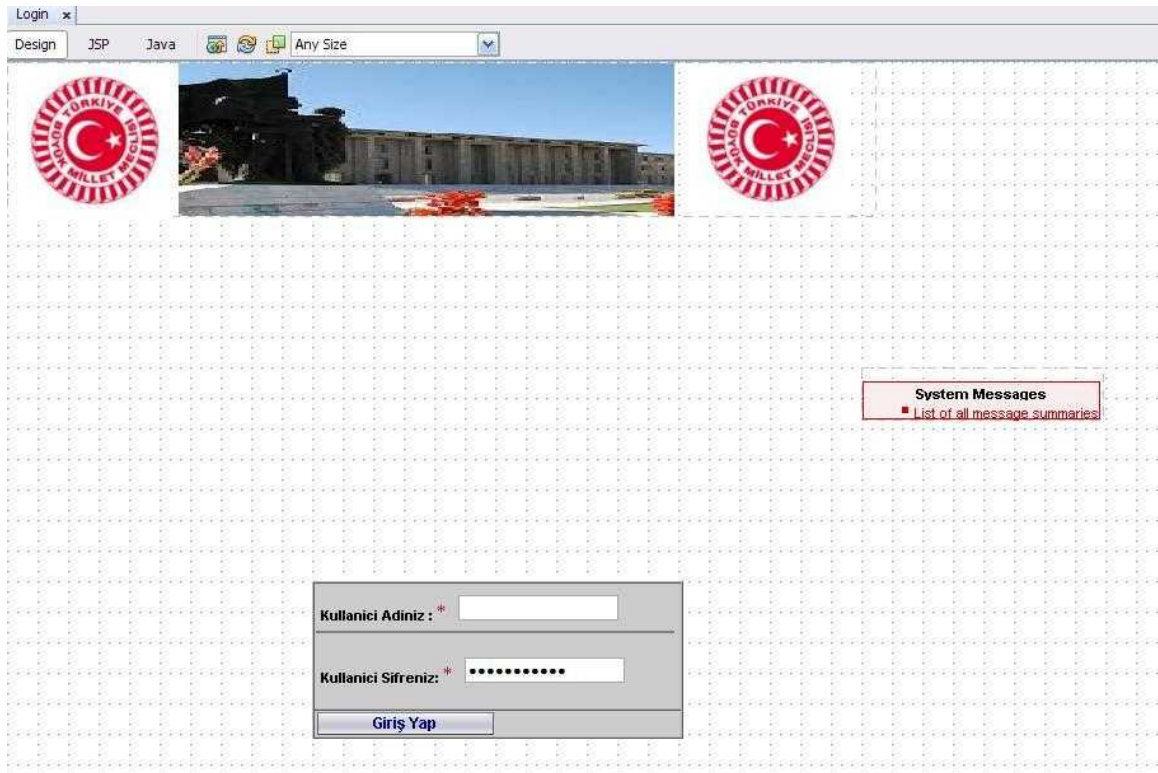
4.3.3 Danışman ve sekreter

Danışman – Sekreter tipindeki kullanıcı, tanımlama ekranı dışında tüm ekranlara ulaşabilmektedir. Veri girişi yapabilmekte fakat silme yapamamaktadır.

4.4 Milletvekili Danışmanlık ve Ziyaretçi Takip Sistemi Ekranları ve Sınıfları

4.4.1 Kullanıcı giriş ekranı

Bu projede kullanıcılara login.jsp den giriş yapabilmeleri için kullanıcı adı ve kullanıcı şifresi verilmiştir. Kullanıcı, kullanıcı adı ve kullanıcı şifresi ile ekranda giriş yaparak veri tabanında kayıtlı olması durumunda sisteme ulaşabilecektir. Veri tabanında kayıtlı olmaması durumunda hata sayfası olan error.jsp'ye yönlendirilecektir. Diğer yandan giriş tanımları doğru olduğu takdirde kullanıcıya ait bilgiler WebService'den alınarak sessionclass'a set edilecektir. Bu yapı, diğer işlemlerde veri tabanına sorgu yollarak ağı meşgul etmemeyi ve serverdan hızlı cevap alabilmeyi sağlamaktadır.



Şekil 4.7 Kullanıcı giriş ekranı

4.4.2 Ziyaretçi ekranı

Şekil 4.7’teki kullanıcı giriş ekranından giriş yapıldıktan sonra, Şekil 4.8’teki ziyaretçi ekranına ulaşılmaktadır. Bu sayfada ziyaretçileri görüntüleme, hızlı ve özet ziyaretçi ekleyebilme ve kayıtlı olan ziyaretçilerin değiştirilebilecek bilgilerinin güncellenmesi işlemleri yapılabilmektedir. Sistem, kullanıcı giriş ekranından giriş yapan kullanıcının yetkisine göre, table’daki ‘sil’ butonunun açılmasına imkân vermektedir.



Şekil 4.8 Ziyaretçi ekranı

4.4.3 Ziyaretçi kayıt ekranı

TBMM’ye gelen ziyaretçilerin kişisel ve iletişim bilgileri, istek nedenleri, talebi, varsa başka randevu talepleri gibi kayıtların yapıldığı ve yapılacak işlemlerin veri tabanına kayıtlarının gerçekleştirildiği ekrandır.


The screenshot shows a web application interface for visitor registration. The form is set against a grid background. On the left, there is a sidebar with navigation links: 'Go Login Page', 'Ziyaretçi Kayıt', 'Arama', 'Detay Bilgiler', 'Bugün?', 'Kullanıcı Tanımlama', and 'Güvenli Çıkış'. The main form area contains the following fields and options:

- Adı:** * (Text input)
- Soyadı:** * (Text input)
- Unvanı:** * (Dropdown menu with 'abc' selected)
- Referans:** (Text input)
- Telefon:** * (Text input)
- Adres:** (Text input)
- İl:** (Dropdown menu with 'abc' selected)
- İlçe:** (Dropdown menu with 'abc' selected)
- Talebi:** (Dropdown menu with 'abc' selected)
- Talep Konusu:** (Dropdown menu with 'abc' selected)
- T.C. Kimlik No:** * (Text input)
- Örnek:** 505587XXYY
- Görüşen:** * (Radio buttons for 'Milletvekili', 'Danışman', 'Sekreter')
- Yapılacak İşlem:** (Checkboxes for 'Hatırlat', 'Talep Konusu', 'Randevu', 'Programa Katılım', 'Telefon', 'Takip Et')
- Açıklama:** (Text input)
- Tebrik Listesine Ekle** (checkbox)
- Adres Listesine Ekle** (checkbox)
- Listede Yok** (checkbox)
- Dokuman Yükle** (checkbox)
- Verileri Kaydet** (button)
- System Messages** (div with a red border and a list of global message summaries)

Şekil 4.9 Ziyaretçi kayıt ekranı

4.4.4 Ziyaretçi arama ekranı

Kullanıcının her türlü sorgulama kriterine imkan tanıyan detaylı arama ekranıdır. Randevular, Davetler, Tebrik Listesi, Adres Listesi, Katılınacak Programlar kategorilerinde zaman aralığı, adı-soyadı, davet yeri, randevu ismi vb gibi alanlarda aramaların yapılabileceği ekrandır.

[Go Login Page](#)
[Ziyaretçi Kayıt](#)
[Arama](#)
[Detay Bilgiler](#)
[Bugün ?](#)
[Kullanıcı Tanımlama](#)
[Gözetim Çıkış](#)

[Static Text](#)

Lütfen Aramak İstedığınız Kriteri Seçiniz..!

Tüm Ziyaretçiler Randevular Davetler Tebrik Listesi Adres Listesi Katılacak Programlar

Randevu İsmi

Davet Yeri :

Ad :

Soyad :

Başlangıç Tarih Bitiş Tarih

[Liste Getir](#)

Ziyaretçiler

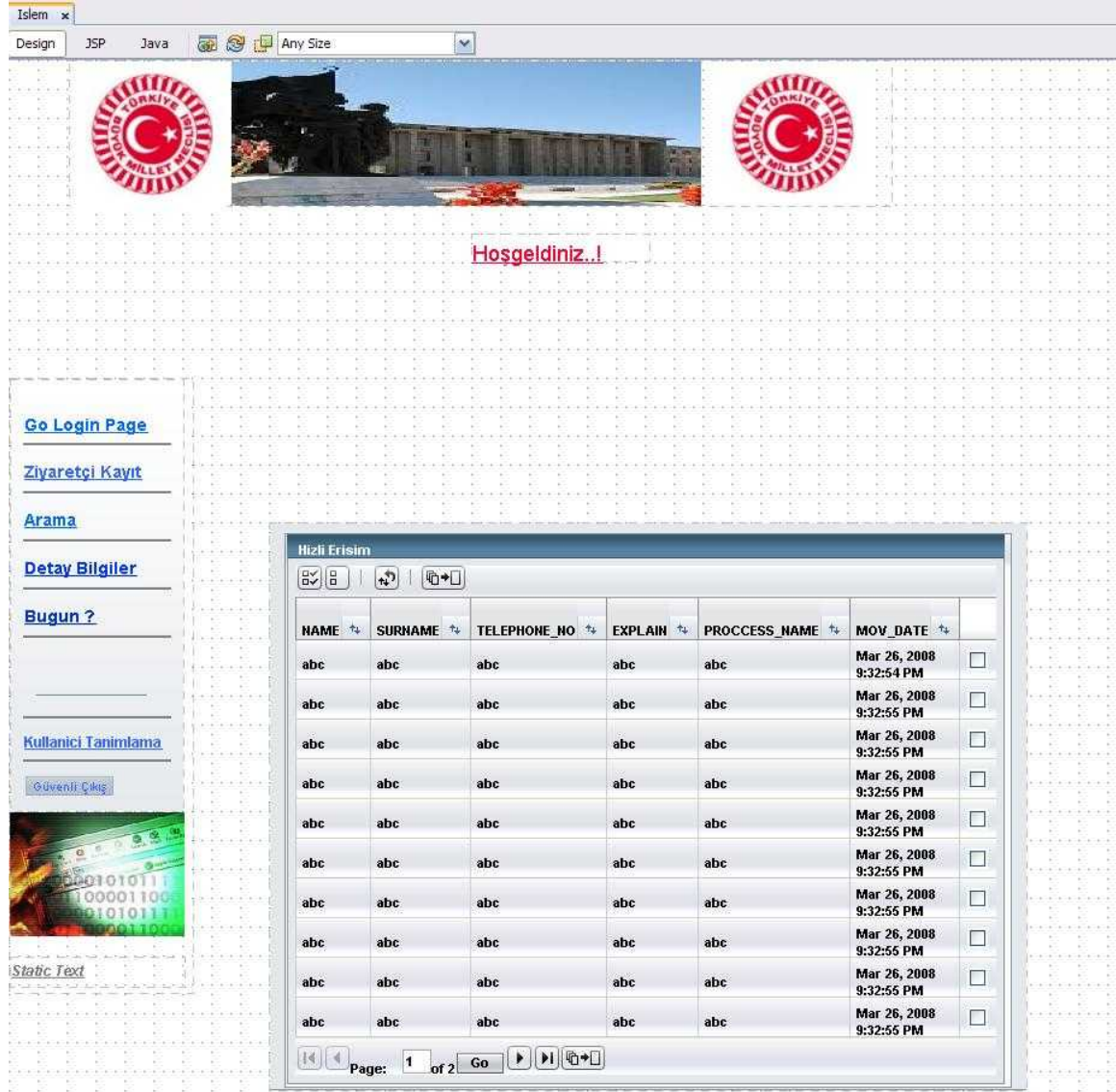
| NAME | SURNAME | IDNUM | EXPLAIN | REFERENCE | ENTRY_DATE |
|------|---------|-------|---------|-----------|------------|
| abc | abc | 1 | abc | abc | abc |
| abc | abc | 1 | abc | abc | abc |
| abc | abc | 1 | abc | abc | abc |
| abc | abc | 1 | abc | abc | abc |
| abc | abc | 1 | abc | abc | abc |
| abc | abc | 1 | abc | abc | abc |
| abc | abc | 1 | abc | abc | abc |
| abc | abc | 1 | abc | abc | abc |
| abc | abc | 1 | abc | abc | abc |
| abc | abc | 1 | abc | abc | abc |
| abc | abc | 1 | abc | abc | abc |

Page: 1 of 2 Go

Şekil 4.10 Ziyaretçi arama ekranı

4.4.5 Bugün? ekranı

Bu ekran kullanıcıya, arama sayfasına gidip tarihi bugün seçip arama yapması yerine kısaca o gün içinde randevu, davet, tebrik telefon vb. işlerin listesine ulaşabilme imkanı sağlamaktadır.



Şekil 4.11 İşlem bugün? ekranı

4.4.6 Not ekleme ve görüntüleme ekranı

Not ekleme ve görüntüleme ekranında danışman veya sekreter tipindeki kullanıcı ve milletvekili tipindeki kullanıcıya ait olmak üzere iki senaryo bulunmaktadır. Sistem, danışman veya sekreter tipindeki kullanıcı tarafından kullanılmaktayken, sayfa yüklendiğinde ait olduğu milletvekiline iletilecek not ekleme alanı görüntülenmektedir. Böylece danışman veya sekreter, ait olduğu vekilin yanına gitmeden sistem aracılığı ile not(ları) iletmiş olacaktır. Eğer sistem milletvekili tipindeki kullanıcı tarafından kullanılmakta ise sayfa yüklendiği zaman not ekleme

kısmı yerine sadece eklenen not(lar) görüntülenmektedir. Şekil 4.12'de ekranın tamamı gösterilmektedir.

The screenshot displays a web application interface for managing notes. It is divided into two main sections. The top section, titled "Vekile İletilecek Notlar:", contains a text input field labeled "Not Giriniz:" and a "Kaydet>>>" button with a green checkmark icon. The bottom section, titled "Notlar:", features a table with the following data:

| NO | NOT | TARİH | SAAT |
|----|-----|--------------------------|------|
| 1 | abc | May 16, 2008 10:18:21 PM | abc |
| 1 | abc | May 16, 2008 10:18:21 PM | abc |
| 1 | abc | May 16, 2008 10:18:21 PM | abc |

Below the table is a section titled "Günlük Notlar" with a large empty text area and a vertical scrollbar on the right side.

Şekil 4.12 Notlar.jsp ekranı

4.4.7 Kullanıcı tanımlama ekranı

Admin tipindeki kullanıcının erişebileceği ekran olup, tanımlanmış 1 kullanıcı bulunmaktadır. Bu da Bilgi İşlem Dairesi personeline aittir. Sisteme yeni bir kullanıcı ekleneceğinde kullanıcı adı, kullanıcı şifresi vb. alanların veri tabanına eklenmesini ve sisteme giriş hakkı verilmesini sağlayan ekrandır.

The screenshot shows a web application window with the title 'kul_tanimla'. The window has a menu bar with 'Design', 'JSP', 'Java', and 'Any Size'. The main content area is titled 'YENI KULLANICI EKLEME' and contains a form with the following elements:

- Two dropdown menus: 'Vekil Ad' and 'Parti Ismi'.
- A section titled 'Kullanici Cesidi' with three radio buttons: 'Vekil', 'Sekreter', and 'Danisman'.
- Text input fields for 'Kullanici Adi', 'Isim', and 'Soyad'.
- A text input field for 'Ekleme Tarihi' with a calendar icon.
- Text input fields for 'Sifre' and 'Sifre Tekrar'.
- A button labeled 'Bilgeleri Kaydet'.

Şekil 4.13 Kullanıcı tanımlama ekranı

4.4.8 Sol menü

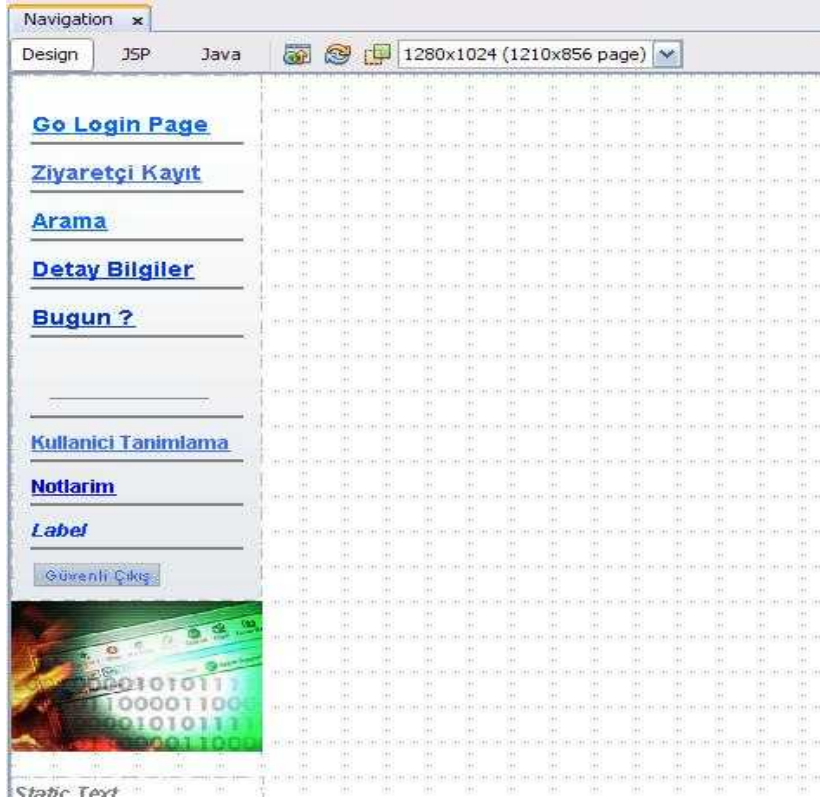
Sistemdeki ekranlara ulaşmak için hazırlanmış menüdür. Yetkiye dayalı olarak görünebilen alanlar değişmektedir. Her sayfada yer almaktadır. Arka tarafta init () fonksiyonu içinde kullanıcı adı ve şifre kontrolü yapmaktadır. Kayıt ekranı hariç her ekranda kontrol bulunmaktadır. Düzen ve performans açısından her sayfada kullanıcı adı ve kullanıcı şifresi kontrolü yapmak yerine sol menüde 1 kere yapılmaktadır. Şekil 4.14 ve Şekil 4.15’de sırası ile sol menü kod yapısı ve sol menü ekran görünümü gösterilmektedir.

```
public void init()
{
    super.init();

    Visual-Web-managed Component Initialization

    String kul_ad=getSessionBean1().getUser_name();
    String pass=getSessionBean1().getPassword();
    if(kul_ad ==null || pass==null)
    {
        javax.faces.context.ExternalContext externalContext = getFacesContext().getExternalContext();
        HttpServletRequest request = (HttpServletRequest) externalContext.getRequest();
        HttpSession session = request.getSession();
        try
        {
            String errorPageURL = externalContext.getRequestContextPath() +
                "/faces/ErrorPage.jsp";
            externalContext.redirect(errorPageURL);
        }
        catch(IOException ioe)
        {
            System.out.println("=====");
            ioe.printStackTrace(System.out);
            System.out.println(ioe.toString());
            System.out.println("=====");
        }
    }
}
```

Şekil 4.14 Sol menü kod yapısı



Şekil 4.15 Sol menü ekran görünümü

4.4.9 SessionControl sınıfı

Bu sınıf servlet olarak yazılmış olup, server-client ilişkisinde istek ve cevap nesnelerinin filtre edilmesini yapmaktadır. Bu filtrede server tarafında sessionın yeni olup olmadığının kontrolü yapılarak session timeout olduğu takdirde 'ErrorPage.jsp' ye yönlendirilmektedir. Şekil 4.16'da detaylı resmi gösterilmektedir.

```
1 package thesis_deneme3;
2 import java.io.*;
3 import java.net.*;
4 import java.util.*;
5 import java.text.*;
6 import javax.servlet.*;
7 import javax.servlet.http.*;
8 import javax.servlet.Filter;
9 import javax.servlet.FilterChain;
10 import javax.servlet.FilterConfig;
11 import javax.servlet.ServletException;
12 import javax.servlet.ServletRequest;
13 import javax.servlet.ServletResponse;
14
15
16 public class LoginAndSessionControl implements Filter {
17
18     Functions func=new Functions();
19     private static int firstRequest = 0;
20     public void doFilter(ServletRequest request, ServletResponse response,
21         FilterChain chain) throws IOException, ServletException {
22         HttpServletRequest hreq = (HttpServletRequest)request;
23         HttpServletResponse hres = (HttpServletResponse)response;
24         HttpSession session = hreq.getSession();
25         SessionBean1 ses=new SessionBean1();
26
27         if (session.isNew()) {
28             if(firstRequest == 0){
29                 firstRequest++;
30             } else {
31                 hres.sendRedirect("Faces/ErrorPage.jsp");
32                 return;
33             }
34         }
35         chain.doFilter(request, response);
36
37
38     public void init(FilterConfig filterConfig) throws ServletException {}
```

Şekil 4.16 Sessioncontrol Class

4.4.10 SessionBean1 sınıfı

SessionBean 1 sınıfı, server tarafından programa giren her kullanıcıya tek oturum açmaktadır. Bu sınıfta veri tabanı sorgusu Login sayfasında bir kez kullanıcı adı ve şifresine göre sorgulandıktan ve dönen değerler bu sınıfa set edildikten sonra

oturum boyunca diğer sayfalarda bu sınıftan get metotları ile verilere ulaşılabilmektedir.

Kullanıcı adı ve şifresinin ve kullanıcının Person tablosundaki bilgilerinin set edilmesi ile ilgili bir örnek olarak aşağıda gösterilmektedir.

Örnek SessionBean1 kodları;

```
private String user_name;
public String getUser_name() {
    return this.user_name;
}

public void setUser_name(String user_name) {
    this.user_name = user_name;
}

private String password;
public String getPassword() {
    return this.password;
}

public void setPassword(String password) {
    this.password = password;
}

java.util.HashMap hm=new java.util.HashMap();
public void LogInfos(String total){
    String[] a=total.split("-");
    hm.put("LOGINID",    a[0]);
    hm.put("PARTY_ID",   a[1]);
    hm.put("MET_ID",     a[2]);
    hm.put("PERSONID",  a[3]);
    hm.put("PERSON_NAME", a[4]);
    hm.put("PERSON_SURNAME", a[5]);
    hm.put("DEPUTY_ID",  a[6]);
    this.setLogInfo(hm); }
}
```

4.4.11 Functions.java sınıfı

Functions.java Sınıfı; sistemde kullanılan giriş kontrolü ve ziyaretçi kayıt ekranında kullanılan seçenekli işlemlerin value-text ve text-value değerlerini döndüren, gridview (tablo) bileşenlerine kolon ekleme, tabloda kullanılan kolonların kullanım kontrolünü yapma (if exist), ve ID'ye sayfaya yükleme veya yüklememe, vb. işlemlerin yapıldığı sınıftır. Bu nedenle aynı kod parçacıklarını birden fazla yazmak yerine, nesne tabanlı yazılıma uygun bir biçimde kodların düzenli tutulmasını sağlamak ve anlaşılabilirliğini kolaylaştırmaktadır. Örnek olarak aşağıda birkaç metod gösterilmektedir.

Function.java örnek kodları; aşağıdaki fonksiyon table içerisinde ID'ye göre var ya da yok kontrolü yapmaktadır. Varsa true, mevcut değilse false değeri döndürmektedir.

```
public boolean OneColumnCheckIfExist(Table t, TableRowGroup trg, String id)
{
    FacesContext fc = FacesContext.getCurrentInstance();
    UIComponent comp = null;
    UIViewRoot vr = fc.getViewRoot();
    TableColumn tc = null;
    boolean flag = true;
    List<UIComponent> children = t.getChildren();
    int count = 0;
    for (UIComponent com : children)
    {
        trg = (TableRowGroup) com;
        List<UIComponent> children1 = trg.getChildren();//kolonlari aliyor
        for (UIComponent com1 : children1)
        {
            tc = (TableColumn) com1;
            String kolon = tc.getId();
            if (tc.getId().equalsIgnoreCase(id) == true)
```



```
    {  
        count++;  
    }  
}  
if (count == 1) {  
    flag = false;} }  
return flag; }
```

4.5 Veri Tabanı Tabloları ve Alanları

Programa ilişkin olarak kullanılan Oracle veri tabanı tabloları, alanları ve açıklamaları şekilleri ile birlikte aşağıda anlatılmıştır.

4.5.1 Kişi tablosu

Kişi tablosu, sistemde ziyaretçi kayıt ekranındaki ziyaretçiye ait alanlar doldurulup, kaydet butonuna basıldıktan sonra arka tarafta veri tabanına eklendiği tablodur.

Kişi tablosundaki alanlar:

- personId :Her eklenen kayıt için sistem tarafından yazılan sequence ve triggerla birlikte otomatik atan unique bir alandır.
- name-surname-idnum :Ziyaretçinin adı-soyadı ve TC Kimlik Numarası bilgilerinin tutulduğu alandır.
- Cityid-countyid :Ziyaretçiye ait il-ilçe bilgilerinin tutulduğu alandır.
- Explain :Ziyaretçi kayıt sayfasında bulunan açıklama kutucuğuna ait bir alandır. Kayıt işlemlerine ve programın esnekliğinin arttırmasına olanak sağlayan eklenecek açıklamaların saklandığı alandır.
- Reference :Programda kayıt sayfasında bulunan referans kısmına denk gelen alandır. Ziyaretçinin varsa referans olarak göstereceği kişinin adı-soyadı bilgisinin tutulduğu alandır.

- Metid :Ziyaretçinin kimle görüştüğü bilgisinin tutulduğu alandır.
- Process :Ziyaretçi hakkında yapılacak işlem, randevu, davet, telefon açma, vb. bilgilerin tutulduğu alandır.
- Jobtitleid :Ziyaretçinin meslek bilgisinin tutulduğu alandır.
- JobtitleName :Ziyaretçinin meslek bilgisinin, dropdownlist'te bulunmaması durumunda diğer kutucuğu işaretlenerek açılan textbox bileşenine manuel olarak yazıldığında tutulduğu alandır.
- Deputyid :Ziyaretçinin görüşmeye geldiği milletvekilinin ID'sinin kaydının sistem tarafından otomatik olarak tutulduğu alandır.
- Telephone_no :Ziyaretçinin telefon numarasının tutulduğu alandır.
- Entry_Date :Sistem tarafından veri tabanına otomatik olarak eklenen kaydın yapıldığı tarihin tutulduğu alandır.
- List_id :Milletvekiline gelen ziyaretçi hakkında yapılacak işlemde özel misafirler için oluşturulmuş adres listesi ve tebrik listesi tabloları bulunmaktadır. Bu tabloda ID olarak tutulup, ismi List tablosunda join ile çekilmektedir.

| Column Name | Data Type | Nullable | Data De... | CO... | Primary Key | COM... |
|--------------|---------------------|----------|------------|-------|---------------|--------|
| PERSONID | NUMBER | No | (null) | 1 | 1 (null) | |
| NAME | VARCHAR2(4000 BYTE) | Yes | NULL | 2 | (null) (null) | |
| SURNAME | VARCHAR2(4000 BYTE) | Yes | NULL | 3 | (null) (null) | |
| IDNUM | NUMBER | Yes | (null) | 4 | (null) (null) | |
| COUNTYID | NUMBER | Yes | (null) | 5 | (null) (null) | |
| CITYID | NUMBER | Yes | (null) | 6 | (null) (null) | |
| EXPLAIN | VARCHAR2(4000 BYTE) | Yes | NULL | 7 | (null) (null) | |
| REFERENCE | VARCHAR2(4000 BYTE) | Yes | NULL | 8 | (null) (null) | |
| MET_ID | NUMBER | Yes | NULL | 9 | (null) (null) | |
| PROCESS | VARCHAR2(4000 BYTE) | Yes | NULL | 10 | (null) (null) | |
| JOBTITLEID | NUMBER | Yes | (null) | 11 | (null) (null) | |
| JOBTITLENAME | VARCHAR2(4000 BYTE) | Yes | NULL | 12 | (null) (null) | |
| DEPUTY_ID | NUMBER | Yes | (null) | 13 | (null) (null) | |
| TELEPHONE_NO | VARCHAR2(4000 BYTE) | Yes | (null) | 14 | (null) (null) | |
| ENTRY_DATE | VARCHAR2(4000 BYTE) | Yes | (null) | 15 | (null) (null) | |
| LIST_ID | NUMBER | Yes | (null) | 16 | (null) (null) | |

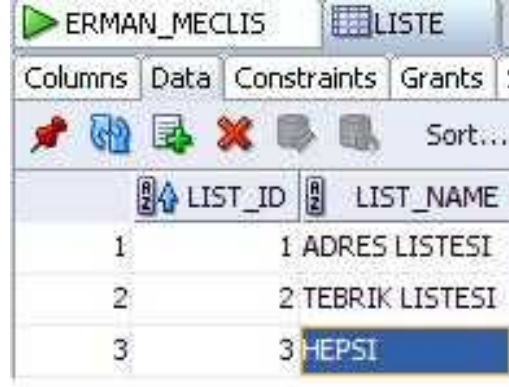
Şekil 4.17 Kişi tablosu

4.5.2 Liste tablosu

Liste tablosu, gelen ziyaretçilerden özel misafirlerin eklendiği tablodur.

Liste tablosundaki alanlar:

- Listid :PrimaryKey alanı olup kişi tablosundaki listID ile join işleminde kullanılmaktadır.
- ListName :Id lere karşılık gelen alanların isimleri tutulmaktadır.



| LIST_ID | LIST_NAME |
|---------|------------------|
| 1 | 1 ADRES LISTESI |
| 2 | 2 TEBRIK LISTESI |
| 3 | 3 HEPSI |

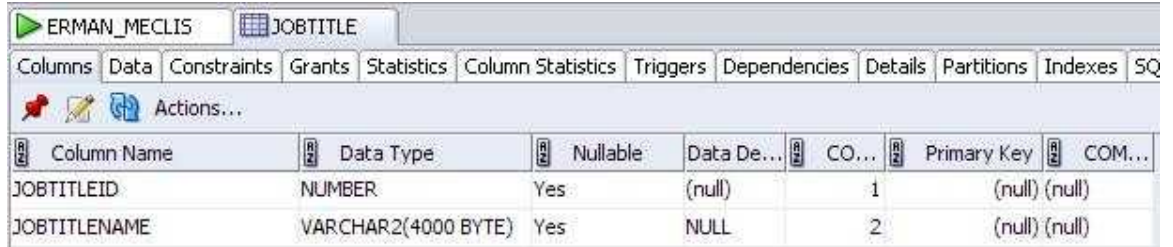
Şekil 4.18 Liste tablosu

4.5.3 Meslek tablosu

Mesleklerinin tutulduğu tablodur.

Meslek tablosundaki alanlar:

- Jobtitleid :Mesleklerin unique olarak ID'lerinin tutulduğu alandır.
- JobtitleName :Mesleklerin isimlerinin tutulduğu alandır.



| Column Name | Data Type | Nullable | Data De... | CO... | Primary Key | COM... |
|--------------|---------------------|----------|------------|-------|-------------|--------|
| JOBTITLEID | NUMBER | Yes | (null) | 1 | (null) | (null) |
| JOBTITLENAME | VARCHAR2(4000 BYTE) | Yes | NULL | 2 | (null) | (null) |

Şekil 4.19 Meslek tablosu

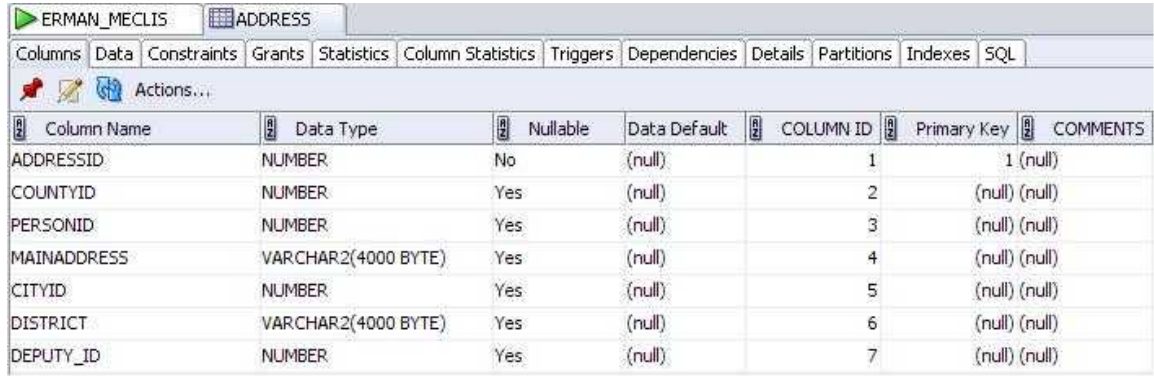
4.5.4 Adres tablosu

Ziyaretçilerin adres bilgilerinin tutulduğu tablodur.

Adres tablosundaki alanlar:

- Address id :Primary Key olan bir alandır ve dinamik olarak eklendikçe artmaktadır.
- Personid :Ziyaretçinin unique olan ID'sinin tutulduğu alandır.

- Countyid :Adres bilgilerinde gelen ziyaretçinin ilçe bilgisinin tutulduğu alandır.
- Cityid :Adres bilgilerinde gelen ziyaretçinin il bilgisinin tutulduğu alandır.
- District :Adres bilgilerinde gelen ziyaretçinin sokak bilgisinin tutulduğu alandır.
- Deputyid :Ziyaretçinin hangi milletvekiline geldiğinin kolaylıkla sorgu tarafında çekilmesi için eklenmiş olan alandır. Bu alanda Vekil ID'si tutulmaktadır.



| Column Name | Data Type | Nullable | Data Default | COLUMN ID | Primary Key | COMMENTS |
|-------------|---------------------|----------|--------------|-----------|---------------|----------|
| ADDRESSID | NUMBER | No | (null) | 1 | 1 (null) | |
| COUNTYID | NUMBER | Yes | (null) | 2 | (null) (null) | |
| PERSONID | NUMBER | Yes | (null) | 3 | (null) (null) | |
| MAINADDRESS | VARCHAR2(4000 BYTE) | Yes | (null) | 4 | (null) (null) | |
| CITYID | NUMBER | Yes | (null) | 5 | (null) (null) | |
| DISTRICT | VARCHAR2(4000 BYTE) | Yes | (null) | 6 | (null) (null) | |
| DEPUTY_ID | NUMBER | Yes | (null) | 7 | (null) (null) | |

Şekil 4.20 Adres tablosu

4.5.5 Adres listesi tablosu

Özel misafir veya ziyaretçiler için hazırlanmış tablodur. Milletvekilinin sonradan ulaşmak istediği ziyaretçilerinin bilgilerinin tutulduğu tablodur.

Adres listesi tablosundaki alanlar:

- AdresListeld :Primary Key alandır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.
- PersonId :Person tablosundan adı-soyadı vb. alanların çekilmesi için eklenmiş bir alandır.
- Deputyid :Ziyaretçinin hangi milletvekiline ait olduğu bilgisini tutan anlatan alandır.
- AdresEklemeTarihi :Sistem tarafından otomatik olarak eklenen kayıt tarihini tutan alandır.

| Column Name | Data Type | Nullable | Data Default | COLUMN ID | Primary Key | COMMENTS |
|---------------------|-----------|----------|--------------|-----------|---------------|----------|
| ADRESLIST_ID | NUMBER | No | (null) | 1 | 1 (null) | |
| PERSONID | NUMBER | Yes | (null) | 2 | (null) (null) | |
| DEPUTY_ID | NUMBER | Yes | (null) | 3 | (null) (null) | |
| ADRES_EKLEME_TARIHI | DATE | Yes | (null) | 4 | (null) (null) | |

Şekil 4.21 Adres listesi tablosu

4.5.6 Davet tablosu

Ziyaretçi kaydı yapılırken, talebi davet ise davet bilgilerinin tutulduğu tablodur.

Davet tablosundaki alanlar:

- invitationId :Primary Key alanıdır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.
- InvitationName :Davet isim bilgisinin tutulduğu alanıdır.
- InvitationDate :Davet tarih bilgisinin tutulduğu alanıdır.
- InvitationPlace :Davet yer bilgisinin tutulduğu alanıdır.
- InvitationTime :Davet saat bilgisinin tutulduğu alanıdır.
- InvitationExplain :Varsa davet açıklama bilgisinin tutulduğu alanıdır.
- PersonId :Davet talebini getiren kişinin ID bilgisinin tutulduğu alanıdır.
- DeputyId :Davet bilgisinin hangi milletvekiline ait olduğu bilgisinin tutulduğu alanıdır.

| Column Name | Data Type | Nullable | Data De... | CO... | Primary Key | COM... |
|--------------------|---------------------|----------|------------|-------|---------------|--------|
| INVITATION_ID | NUMBER | No | (null) | 1 | 1 (null) | |
| INVITATION_NAME | VARCHAR2(4000 BYTE) | Yes | (null) | 2 | (null) (null) | |
| INVITATION_DATE | DATE | Yes | NULL | 3 | (null) (null) | |
| INVITATION_PLACE | VARCHAR2(4000 BYTE) | Yes | (null) | 4 | (null) (null) | |
| INVITATION_TIME | VARCHAR2(4000 BYTE) | Yes | (null) | 5 | (null) (null) | |
| INVITATION_EXPLAIN | VARCHAR2(4000 BYTE) | Yes | (null) | 6 | (null) (null) | |
| PERSONID | NUMBER | Yes | (null) | 7 | (null) (null) | |
| DEPUTY_ID | NUMBER | Yes | (null) | 8 | (null) (null) | |

Şekil 4.22 Davet tablosu

4.5.7 Görüşen kişi tablosu

Kişi tablosunda tutulan metid alanının ismine ait bilginin sorgulandığı tablodur.

Görüşen kişi tablosundaki alanlar:

- Metid :Primary Key alandır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.

- MetName :Unique id lere karşılık gelen açıklama isimlerinin tutulduğu alandır.

Şuan için sabit bir tablodur.

- Milletvekili
- Danışman
- Sekreter
- SA (System Admin) gibi bilgiler tutulmaktadır.

| MET_ID | MET_NAME |
|--------|----------------|
| 1 | 1 MİLLETVEKİLİ |
| 2 | 2 DANIŞMAN |
| 3 | 3 SEKRETER |
| 4 | 4 SA |

Şekil 4.23 Görüşen kişi tablosu

4.5.8 İlçe tablosu

Adres tablosunda tutulan countyld alanındaki ilçe ID'sine ait ismin sorgulanması için hazırlanmış tablodur. Türkiye'deki bütün ilçeler yüklenmiştir.

İlçe tablosundaki alanlar:

- Countyld :Primary Key alandır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.
- CountyName :İlçelerin isimlerini tutulduğu alandır.
- Cityld :İlçelerin ait olduğu il ID'lerinin tutulduğu alandır.

| Column Name | Data Type | Nullable | Data De... | CO... | Primary Key | COM... |
|-------------|---------------------|----------|------------|-------|-------------|--------|
| COUNTYID | NUMBER | No | (null) | 1 | 1 | (null) |
| COUNTYNAME | VARCHAR2(4000 BYTE) | Yes | (null) | 2 | (null) | (null) |
| CITYID | NUMBER | Yes | (null) | 3 | (null) | (null) |

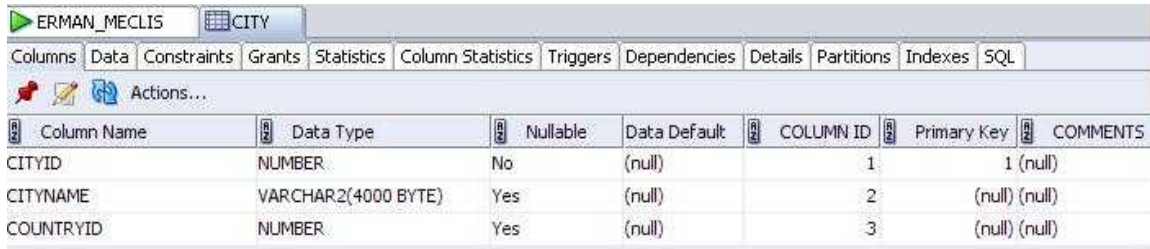
Şekil 4.24 İlçe tablosu

4.5.9 İl tablosu

Adres tablosunda tutulan cityld alanındaki il ID'sine ait ismin sorgulanması için hazırlanmış tablodur. Türkiye'deki bütün iller yüklenmiştir.

İl tablosundaki alanlar:

- CityId :Primary Key alandır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.
- CityName :İllerin isimlerinin tutulduğu alandır.
- CountryId :Bu alan şuan için kullanılmamaktadır. İleride yurt dışından gelen ziyaretçilerin kaydının yapılması durumunda tüm ülkeler eklenerek ülke ID'lerine ait şehirlerin tutulması için hazırlanmış alandır.



The screenshot shows a database table structure for 'ERMAN_MECLIS.CITY'. The table has three columns: CITYID, CITYNAME, and COUNTRYID. CITYID is the primary key. The data types are NUMBER for CITYID and COUNTRYID, and VARCHAR2(4000 BYTE) for CITYNAME. The table is shown in a software interface with various tabs like Columns, Data, Constraints, etc.

| Column Name | Data Type | Nullable | Data Default | COLUMN ID | Primary Key | COMMENTS |
|-------------|---------------------|----------|--------------|-----------|---------------|----------|
| CITYID | NUMBER | No | (null) | 1 | 1 (null) | |
| CITYNAME | VARCHAR2(4000 BYTE) | Yes | (null) | 2 | (null) (null) | |
| COUNTRYID | NUMBER | Yes | (null) | 3 | (null) (null) | |

Şekil 4.25 İl tablosu

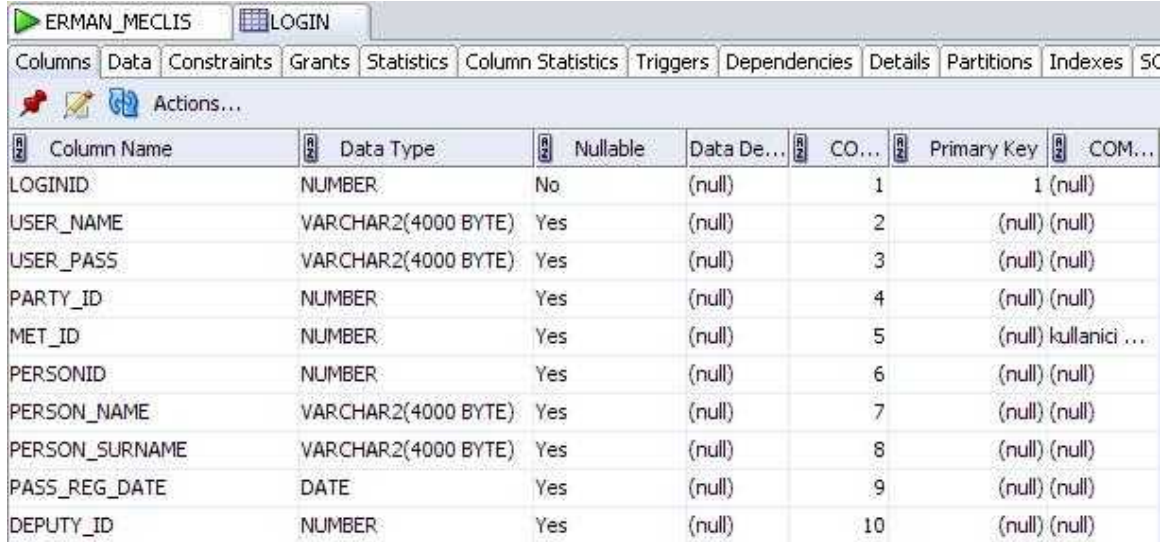
4.5.10 Kullanıcı tablosu

Program kullanıcılarının tutulduğu tablodur.

Kullanıcı tablosundaki alanlar:

- LoginId :Primary Key alandır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.
- UserName :Programa giriş yapılması için hazırlanmış kullanıcı adı bilgisinin tutulduğu alandır.
- UserPass :Program kullanıcıların şifre bilgilerinin tutulduğu alandır.
- PartyId :Program kullanıcılarının hangi pariyeye ait olduğu bilgisinin tutulduğu alandır.
- MetId :Kullanıcının kim olduğu bilgisinin tutulduğu alandır.
 - Milletvekili
 - Danışman
 - Sekreter

- System Admin
- PersonId :İstenildiği takdirde kullanıcılara ait bilgilerin tutulması için hazırlanmış alandır.
- PersonName-surname :Kullanıcıların adı-soyadı bilgilerinin tutulduğu alandır.
- PassRegDate :Kullanıcının veri tabanına eklendiği tarih bilgisi tutulduğu alandır. Sistem tarafından otomatik olarak eklenmektedir. Ayrıca bu alan güvenlik açısından 3 aylık süre dolduğu takdirde kullanıcıya şifre değişikliğinin yaptırabilmek için eklenmiştir.
- DeputyId :Kullanıcıların hangi milletvekiline ait olduğu bilgisinin tutulduğu alandır.



| Column Name | Data Type | Nullable | Data De... | CO... | Primary Key | COM... |
|----------------|---------------------|----------|------------|-------|----------------------|--------|
| LOGINID | NUMBER | No | (null) | 1 | 1 (null) | |
| USER_NAME | VARCHAR2(4000 BYTE) | Yes | (null) | 2 | (null) (null) | |
| USER_PASS | VARCHAR2(4000 BYTE) | Yes | (null) | 3 | (null) (null) | |
| PARTY_ID | NUMBER | Yes | (null) | 4 | (null) (null) | |
| MET_ID | NUMBER | Yes | (null) | 5 | (null) kullanıcı ... | |
| PERSONID | NUMBER | Yes | (null) | 6 | (null) (null) | |
| PERSON_NAME | VARCHAR2(4000 BYTE) | Yes | (null) | 7 | (null) (null) | |
| PERSON_SURNAME | VARCHAR2(4000 BYTE) | Yes | (null) | 8 | (null) (null) | |
| PASS_REG_DATE | DATE | Yes | (null) | 9 | (null) (null) | |
| DEPUTY_ID | NUMBER | Yes | (null) | 10 | (null) (null) | |

Şekil 4.26 Kullanıcı tablosu

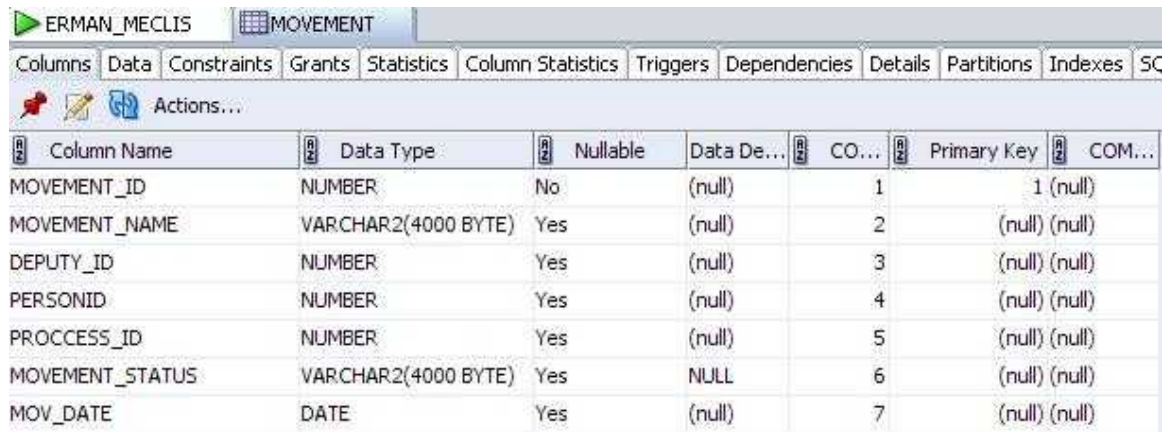
4.5.11 Hareket tablosu

Ziyaretçi kayıt sayfasında talep doğrultusunda yapılan işlemlerin tutulduğu ana bir tablodur. Bütün hareketler eklenmektedir.

Hareket tablosundaki alanlar:

- MovementId :Primary Key alandır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.

- MovementName :Yapılan harekete ilişkin ismin tutulduğu alandır.
- DeputyId :Ziyaretçiye ait hareketin hangi milletvekiline ait olduğu bilgisinin tutulduğu alandır.
- PersonId :Ziyaretçiye ait detaylı bilgilerinin kişi tablosundan sorgulanabilmesi için eklenmiş alandır.
- ProcessId :Hareket ID'sine ait bilginin tutulduğu alandır. Process Tablosu ile sorgulamada hareketin adı alınmaktadır.
- MovementStatus :Hareketin gerçekleşip gerçekleşmeme bilgisinin tutulduğu alandır.
- MovDate :Hareketin yapıldığı tarih bilgisinin tutulduğu alandır. Sistem tarafından otomatik eklenmektedir.



| Column Name | Data Type | Nullable | Data De... | CO... | Primary Key | COM... |
|-----------------|---------------------|----------|------------|-------|---------------|--------|
| MOVEMENT_ID | NUMBER | No | (null) | 1 | 1 (null) | |
| MOVEMENT_NAME | VARCHAR2(4000 BYTE) | Yes | (null) | 2 | (null) (null) | |
| DEPUTY_ID | NUMBER | Yes | (null) | 3 | (null) (null) | |
| PERSONID | NUMBER | Yes | (null) | 4 | (null) (null) | |
| PROCESS_ID | NUMBER | Yes | (null) | 5 | (null) (null) | |
| MOVEMENT_STATUS | VARCHAR2(4000 BYTE) | Yes | NULL | 6 | (null) (null) | |
| MOV_DATE | DATE | Yes | (null) | 7 | (null) (null) | |

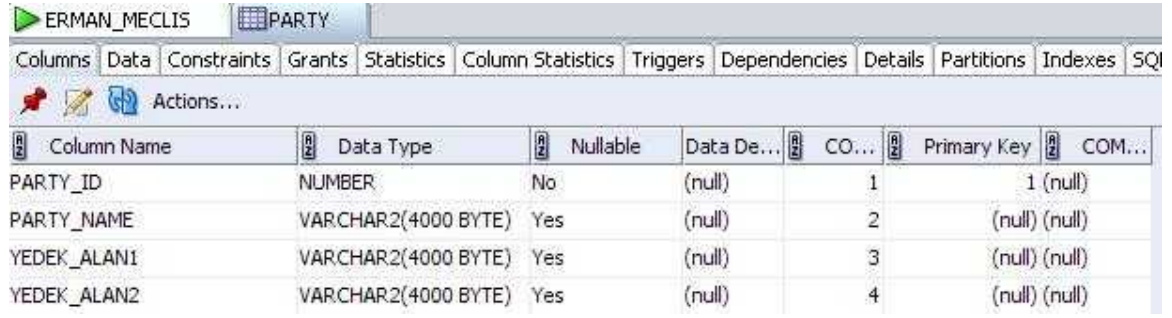
Şekil 4.27 Hareket tablosu

4.5.12 Parti tablosu

TBMM'de bulunan partilerin isim ve numaralarının tutulduğu tablodur.

Parti tablosundaki alanlar:

- PartyId :Primary Key alandır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.
- PartyName :Partilerin isim bilgilerinin tutulduğu alandır.



| Column Name | Data Type | Nullable | Data De... | CO... | Primary Key | COM... |
|-------------|---------------------|----------|------------|-------|---------------|--------|
| PARTY_ID | NUMBER | No | (null) | 1 | 1 (null) | |
| PARTY_NAME | VARCHAR2(4000 BYTE) | Yes | (null) | 2 | (null) (null) | |
| YEDEK_ALAN1 | VARCHAR2(4000 BYTE) | Yes | (null) | 3 | (null) (null) | |
| YEDEK_ALAN2 | VARCHAR2(4000 BYTE) | Yes | (null) | 4 | (null) (null) | |

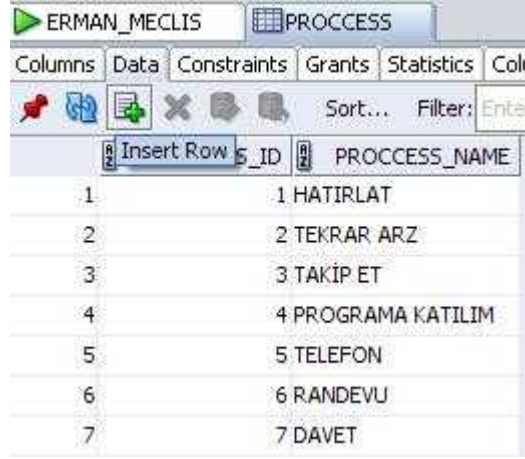
Şekil 4.28 Parti tablosu

4.5.13 İşlem tablosu

Gelen ziyaretçiler hakkında yapılan işlem bilgilerinin isimlerinin tutulduğu tablodur. Sabit bilgileri içeren tablodur.

İşlem tablosundaki alanlar:

- ProcessId : Primary Key alanıdır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.
- ProcessName : İşlem tiplerinin tutulduğu alanıdır.
 - Hatırlatma
 - Tekrar Arz
 - Takip Et
 - Programa Katılım
 - Telefon
 - Randevu
 - Davet



| S_ID | PROCESS_NAME |
|------|------------------|
| 1 | HATIRLAT |
| 2 | TEKRAR ARZ |
| 3 | TAKIP ET |
| 4 | PROGRAMA KATILIM |
| 5 | TELEFON |
| 6 | RANDEVU |
| 7 | DAVET |

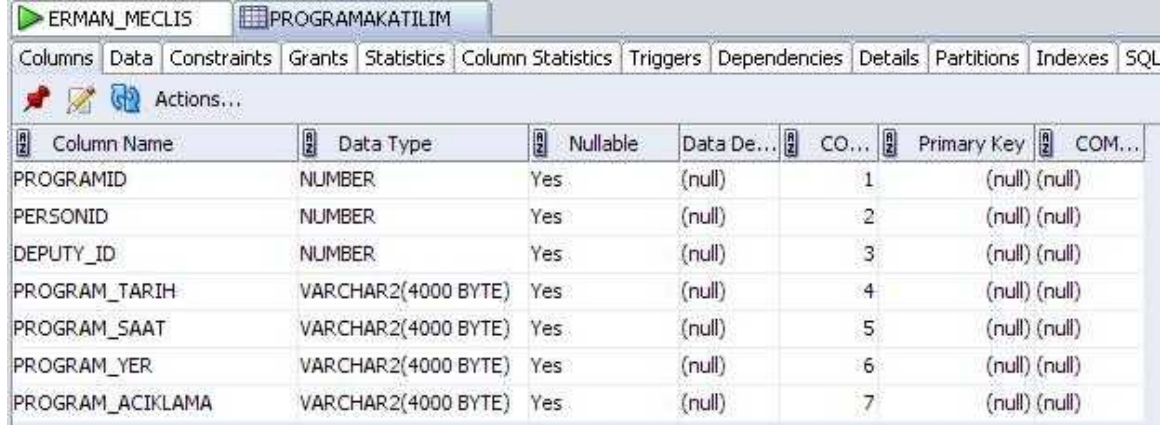
Şekil 4.29 İşlem tablosu

4.5.14 Programa katılım tablosu

Gelen ziyaretçilerin programa davet önerisi ile geldiği takdirde bilgilerinin kaydedildiği tablodur.

Programa katılım tablosundaki alanlar:

- ProgramId :Primary Key alanıdır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.
- PersonId :Programa katılım önerisini getiren kişinin ID'sinin tutulduğu alanıdır.
- DeputyId :Hangi milletvekiline ait olduğu bilgisinin tutulduğu alanıdır.
- ProgramTarih :Katılınacak olan programın tarih bilgisinin tutulduğu alanıdır.
- ProgramSaat :Program saat bilgisinin tutulduğu alanıdır.
- ProgramYer :Programın yer bilgisinin tutulduğu alanıdır.
- ProgramAciklama :Varsa programa ilişkin açıklama bilgisinin tutulduğu alanıdır.



| Column Name | Data Type | Nullable | Data De... | CO... | Primary Key | COM... |
|------------------|---------------------|----------|------------|-------|---------------|--------|
| PROGRAMID | NUMBER | Yes | (null) | 1 | (null) (null) | |
| PERSONID | NUMBER | Yes | (null) | 2 | (null) (null) | |
| DEPUTY_ID | NUMBER | Yes | (null) | 3 | (null) (null) | |
| PROGRAM_TARİH | VARCHAR2(4000 BYTE) | Yes | (null) | 4 | (null) (null) | |
| PROGRAM_SAAT | VARCHAR2(4000 BYTE) | Yes | (null) | 5 | (null) (null) | |
| PROGRAM_YER | VARCHAR2(4000 BYTE) | Yes | (null) | 6 | (null) (null) | |
| PROGRAM_ACIKLAMA | VARCHAR2(4000 BYTE) | Yes | (null) | 7 | (null) (null) | |

Şekil 4.30 Programa katılım tablosu

4.5.15 Randevu tablosu

Ziyaretçiden gelen talep doğrultusunda randevunun başka bir tarihe ertelenmesi durumunda kullanılan tablodur.

Randevu tablosundaki alanlar:

- AppointmentId :Primary Key alandır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.
- AppointmentName :Randevu isim bilgisinin tutulduğu alandır.
- AppointmentTime :Randevu tarih bilgisinin tutulduğu alandır.
- AppointmentDate :Randevuya ilişkin tarih bilgisinin tutulduğu alandır.
- DeputyId :Hangi milletvekiline ait olduğu bilgisinin tutulduğu alandır.
- PersonId :Randevusu alınan kişiye ait ID'nin tutulduğu alandır.

| Column Name | Data Type | Nullable | Data Default | COLUMN ID | Primary Key | COMMENTS |
|------------------|--------------------|----------|--------------|-----------|---------------|----------|
| APPOINTMENT_ID | NUMBER | No | (null) | 1 | 1 (null) | |
| APPOINTMENT_NAME | VARCHAR2(100 BYTE) | Yes | NULL | 2 | (null) (null) | |
| APPOINTMENT_DATE | DATE | Yes | NULL | 3 | (null) (null) | |
| APPOINTMENT_TIME | VARCHAR2(10 BYTE) | Yes | NULL | 4 | (null) (null) | |
| DEPUTY_ID | NUMBER | Yes | (null) | 5 | (null) (null) | |
| PERSONID | NUMBER | Yes | (null) | 6 | (null) (null) | |
| PROCESS_ID | NUMBER | Yes | (null) | 7 | (null) (null) | |

Şekil 4.31 Randevu tablosu

4.5.16 Tebrik listesi tablosu

Ziyarete gelen özel misafirler için milletvekilinin isteği doğrultusunda kullanılan tablodur.

Tebrik listesi tablosundaki alanlar:

- TebrikId :Primary Key alandır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.
- PersonId :Gerektiğinde detaylı bilgisine ulaşılabilmesi için kişiye ait ID'nin tutulduğu alandır.
- DeputyId :Ziyaretçinin hangi milletvekiline ait olduğu bilgisinin tutulduğu alandır.
- TebrikEklemeTarih :Sistem tarafından otomatik olarak veri tabanına eklenen tarih bilgisinin tutuluğu alandır.

| Column Name | Data Type | Nullable | Data De... | CO... | Primary Key | COM... |
|----------------------|-----------|----------|------------|-------|---------------|--------|
| TEBRIK_ID | NUMBER | No | (null) | 1 | 1 (null) | |
| PERSONID | NUMBER | Yes | (null) | 2 | (null) (null) | |
| DEPUTY_ID | NUMBER | Yes | (null) | 3 | (null) (null) | |
| TEBRIK_EKLEME_TARIHI | DATE | Yes | (null) | 4 | (null) (null) | |

Şekil 4.32 Tebrik listesi tablosu

4.5.17 Vekil tablosu

TBMM’de bulunan milletvekillerinin tutulduğu tablodur.

Vekil tablosundaki alanlar:

- DeputyId :Primary Key alandır. Dinamik olarak artan ve unique bir numara verilmesi için hazırlanmıştır.
- DeputyName :Milletvekillerine ait ad bilgisinin tutulduğu alandır.
- DeputyCityId :Milletvekillerine ait hangi ilden seçildiği bilgisini tutan alandır.
- DeputyCountyId :Milletvekillerine ait hangi ilçeden seçildiği bilgisini tutan alandır.
- PartyId :Milletvekillerinin hangi parti mensubu olduğu bilgisinin tutulduğu alandır.
- PartyName :Yedek olarak hazırlanmış alandır. Gerekğinde değiştirilerek kullanılmak üzere hazırlanmıştır.

| Column Name | Data Type | Nullable | Data De... | CO... | Primary Key | COM... |
|-----------------|---------------------|----------|------------|-------|---------------|--------|
| DEPUTY_ID | NUMBER | No | (null) | 1 | 1 (null) | |
| DEPUTY_NAME | VARCHAR2(4000 BYTE) | Yes | (null) | 2 | (null) (null) | |
| DEPUTY_CITYID | NUMBER | Yes | (null) | 3 | (null) (null) | |
| DEPUTY_COUNTYID | NUMBER | Yes | (null) | 4 | (null) (null) | |
| PARTY_ID | NUMBER | Yes | (null) | 5 | (null) (null) | |
| PARTY_NAME | VARCHAR2(4000 BYTE) | Yes | (null) | 6 | (null) (null) | |

Şekil 4.33 Vekil tablosu

4.6 WebService Tarafından Kullanılan Örnek Sorgular

WebService tarafından veri tabanı işlemleri için kullanılan sorgular bulunmaktadır. Şekil 4.34’te Web metodlar ve Web metodlara ilişkin örnek kodlar sırasıyla aşağıda gösterilmektedir.



Şekil 4.34 Web Metodlar

Web Metodlara ilişkin örnek kodlar;

Bu web metotta, kullanıcı tanımlama ekranında veri tabanına kayıt yapılma esnasında kullanılan sorgu ve Webservice örneği gösterilmektedir.

```

@WebMethod(operationName = "AddNewUser")
public boolean AddNewUser(@WebParam(name = "u_uname")
String u_uname, @WebParam(name = "u_upass")
String u_upass, @WebParam(name="u_masterid")
int u_masterid, @WebParam(name = "u_metid")
int u_metid, @WebParam(name = "u_name")
String u_name, @WebParam(name = "u_surname")
String u_surname, @WebParam(name = "u_date")
String u_date, @WebParam(name = "u_deptid")
int u_deptid)
{
    PreparedStatement pr=null;
    boolean flag=true;
    try
    {
        Connection con=getConnectionORACLE();
        con.setAutoCommit(true);

        String sql="INSERT INTO
LOGIN(USER_NAME,USER_PASS,PARTY_ID,MET_ID,PERSON_NAME,PERS
ON_SURNAME,PASS_REG_DATE,DEPUTY_ID) values (? , ?, ?, ?,
?,?,TO_DATE(""+u_date+"','MON.DD.YYYY'),?)";
        pr=con.prepareStatement(sql);
        pr.setString(1,u_uname );
        pr.setString(2,u_upass );
        pr.setInt (3,u_masterid);
        pr.setInt (4,u_metid );
        pr.setString(5,u_name );
        pr.setString(6,u_surname);
        pr.setInt (7,u_deptid);

        pr.executeQuery();
    }
}

```

```
    }  
    catch (SQLException ex)  
    {  
        flag=false;  
        ex.printStackTrace();  
    }  
    finally  
    {  
        try {  
            pr.close();  
        } catch (SQLException ex) {  
            Logger.getLogger(Query_Provider.class.getName()).log(Level.SEVERE,  
null, ex);  
        }  
    }  
    return flag;  
}
```

5 SONUÇ, TARTIŞMA VE ÖNERİLER

5.1 Sonuç

Bu çalışmada TBMM'de kullanılan ziyaretçi programı incelenmiş ve milletvekillerinin, danışmanların ve sekreterlerin beklentilerini sağlayacak; hızlı ve kolay erişilebilir, detaylı bilgi tutabilme imkanı sunan, milletvekiline ait notların tutulabileceği, detaylı sorgulamanın yapılabileceği, yeni Java framework teknolojilerini kullanan bir ziyaretçi takip sistemi geliştirilmesi hedeflenmiştir.

Bu kapsamda, önceki programda sadece sekreter veya danışmanların kayıt yaparak sistemi kullandıkları gözlemlenmiş olup aynı zamanda milletvekillerinin de programı etkin bir şekilde kullanması ve milletvekillerin ulaşabilecekleri bilgilerin sunulduğu bir sistem oluşturulması amaçlanmıştır.

Günümüzde gelişen bilişim-web teknolojileri doğrultusunda, kurumlarda kullanılmakta olan ve yeni ihtiyaçlar doğrultusunda ortaya çıkacak olan yeni sistemlerin ortak bir yapı halinde çalışabilmesini karşılayabilmek açısından WebService teknolojisi kullanılmıştır.

TBMM bünyesinde kullanılmakta olan sunucunun Linux tabanlı olması ve de programlama dili olarak Bolum 3'te detaylı olarak bahsedilmiş olunan Web tabanlı olması nedeniyle JSP seçilmiş ve JSP üzerinde uygulama geliştirmeye, yeni teknolojileri ile, kolaylık sağlayan MVC özelliği taşıyan JSF 1.2 frameworkü kullanılmıştır. Bu durum, service wrapper sayesinde URL (Uniform Resource Locator) de numaraların şifreli bir şekilde tutulmasına olanak sağlamıştır.

Bu tez çalışmasında ortaya çıkan “Ziyaretçi Kayıt Sistemi” üzerinde durum çalışması yapılmış olup; bu kapsamda milletvekili ve sekreter ile görüşülmüş, mevcut durum değerlendirmeleri yapılmış, kullanılan programın “server exception”

hataları ile gözlemlendiği, böyle bir durumda programın kapatılıp tekrar açılması halinde geçtiği gözlenmiştir.

İnternet tarayıcısının 10 dakika işlem yapılmadan beklenmesi halinde, sistem tarafından otomatik olarak ilk yapılacak işlemde kullanıcı giriş ekranına gönderilmesi sağlanmıştır.

Yukarıdaki sonuçla birlikte ziyaretçiye ait geçmiş bilgileri, belirlenen zaman aralıklarında hangi zamanlarda, hangi talepler ile geldiği bilgilerine karşılık olarak yeni kullanılacak olan programda giderilerek “Milletvekili Danışmanlık ve Ziyaretçi Takip Sistemi” haline getirilmiştir.

“Milletvekili Danışmanlık ve Ziyaretçi Takip Sistemi”nin kazandırdığı bir diğer özellik ise Bilgi İşlem Dairesi personeli ile yapılan görüşme sonucunda kayıtların TBMM standartlarında yapılmakta olduğudur.

5.2 Tartışma ve Öneriler

Bu çalışmaya başlamadan önce yapılan araştırmalar sonucunda; JSF’ nin MVC’ yi bünyesine yerleştirmiş olması, dünyaca kullanılan program geliştiriciler için büyük kolaylıklar gösteren vs.net teknolojisini yakalamış olması ve de JSP üzerinde kullanılan en büyük frameworklerden biri olan struts bilişimcileri tarafından beğenilmesi ve tavsiyesi üzerine JSF teknolojisinden faydalanılmıştır.

Yukarıda vs.net ile verilmiş olan ilişkiyi genişletmek gerekirse; vs.net platformu, bilişim dünyasında geliştiriciler için birçok özellik sunmaktadır. Java’ya göre çok daha kolay kod geliştirebilme, projelerde daha hızlı ilerleme ve geliştiricilerin kod yazmada yapacağı çoğu olayları kendi yapması, vb. özellikleri bulunmasına karşın, Java’nın da platform bağımsızlığı, veritabanı bağımsızlığı, performans, güvenlik ve de en önemlisi bir şirkete değil de bir konsorsiyoma bağlı olması (SUN, IBM, Macromedia, BEA, Oracle, Siemens, vs) tercih nedeni olmuştur.

Modüler yazılımlarla ilgili olarak bu çalışmada yapılan araştırma ve incelemeler sonucunda projeyi ele alacak olursak; Türkiye’de değişik amaç ve içeriklerde kullanılan programlar temel bakımından ortak bir hale getirilmeli, kuruma veya kullanılacak yere göre artan istek doğrultusunda belirlenmiş olan modülün üzerine eklenerek şekillenmesi sağlanmalıdır. Organizasyona ait herhangi bir veri sisteme bir kere girilmeli, organizasyonun hangi departmanı, hangi amaçla söz konusu veriye ihtiyaç duyarsa eşzamanlı olarak veri kullanılabilirdir.

Bu sisteme ihtiyaç duyan ve bünyesinde kullanılan programların ayrı ayrı yapılması yerine, bütünlüğü sağlayan ERP sistemi çerçevesinde toplanarak bütünlük oluşturmalıdır. Belki her kamu kurum ve kuruluşunun veya özel firmaların aynı teknolojilerle donatılmış programları tercih etmesi mümkün değildir ancak en azından aynı organizasyon içerisinde uygulamalarının modüler yapıda olduğu ve fonksiyonlarının tamamının entegre yapıda çalıştığı ERP sistemi tercih edilmelidir. Böylelikle farklı sunucular, farklı programlama dilleri, farklı teknolojileri bir araya getirmek vb. nedenlerden kaynaklanan performans, güvenlik, zaman, maddi kayıplar önlenmiş olacaktır.

TBMM’de ileride ERP uygulamasına geçilmesi ihtimali göz önünde bulundurularak, web tabanlı “Milletvekili Danışmanlık ve Ziyaretçi Takip Sistemi”, kullanıcılar veya ziyaretçiler tarafından ileride oluşabilecek istekleri karşılayabilmek için uygun teknolojiler kullanılarak geliştirilmiş olup, aynı zamanda TBMM bünyesinde diğer programlara (özellikle geçiş kontrol sistemine) kolay entegre edilebilecek yapıya sahiptir.

KAYNAKLAR LİSTESİ

[1] Tokatlıođlu, Mücella, temel kalite kavramları ve toplam kalite yönetimi, IQ uluslararası kalite danışmanlık A.Ş.,

<http://www.tkgm.gov.tr/turkce/dosyalar/diger%5Cicerikdetaydh323.ppt#1>, 09

Mayıs 2008

[2] TÜİK (20 Kasım 2007) Haber Bülteni 2007 Yılı Hanehalkı Bilişim Teknolojileri Kullanımı Araştırması Sonuçları, Sayı: 186,

(<http://www.tuik.gov.tr/PreHaberBultenleri.do?id=605>

[3] KOCABAL Ahmet (<http://www.caginpulisi.com.tr/26/16-17-18-19.htm>, POLNET ve EMNİYET BİLGİ SİSTEMİ,)

[4] Erdal, M. (2008) Elektronik Belediye Kavramı ve İstanbul Büyükşehir Belediyesi Uygulaması, <http://66.102.1.104/scholar?hl=tr&lr=&q=cache:e-LO9TTm2fUJ:www.edevlet.net/eTurkiye/ebelediye.pdf+ziyaret%C3%A7i+takep+program%C4%B1>, 11 Nisan 2008

[5] <http://www.procurex.co.za/>, 07 Mayıs 2008,

[6] http://www.ritmateknoloji.com/ziyaretci_takip_programi.asp, 10 Nisan 2008

[7] http://www.personeltakip.com/ziyaretci_gecis_sistemi.asp, 10 Nisan 2008

[8] http://www.erben.com.tr/prd_detail.asp?id=48, 10 Nisan 2008

[9]

www.kozbilisim.com/index.php?option=com_content&task=view&id=29&Itemid=1, 10 Nisan 2008

- [10] http://www.icsltd.com.tr/pk_ziyaretc.aspx, 11 Nisan 2008
- [11] http://www.godoro.com/divisions/ehil/mahzen/Java/Faqs/txt/html/faq_J2EE.html
- [12] SUN Microsystem, *Java Server Faces Technology Overview*, <http://java.sun.com/javaee/javaserverfaces/overview.html>, 05 Mayıs 2008
- [13] Cioroianu, *Using JSF*, http://www.onjava.com/pub/a/onjava/2003/09/03/using_jsf.html, 06 Mayıs 2008
- [14] *Java Server Faces*, http://en.wikipedia.org/wiki/JavaServer_Faces, 11 Mayıs 2008,
- [15] Akın, *Java nedir, nasıl çalışır*, http://www.godoro.com/divisions/ehil/mecmua/Magazines/Articles/txt/html/article_WhatisJavaAndHowRun.html, 11 Mayıs 2008
- [16] SUN Microsystem, *Java BluePrints Model-View-Controller*, <http://java.sun.com/blueprints/patterns/MVC-detailed.html>, 05 Mayıs 2008
- [17] www.formfaces.com/Web%20Forms%20Technology%20Comparison.doc
- [18] <http://www.keesbroenink.nl/doc/Java%20frameworks%20-%20middle%20layer.doc>, 05 Mayıs 2008