



T.C.  
İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ



**YÜKSEK LİSANS TEZİ**

**J2EE KULLANILARAK WEB TABANLI ERASMUS  
HAREKETLİLİK BİLGİ SİSTEMİNİN GELİŞTİRİLMESİ**

**Bello MUHAMMAD**

**Enformatik Anabilim Dalı**

**Enformatik Programı**

**Danışman**

**Prof.Dr.Sevinç GÜLSEÇEN**


**Haziran, 2014**

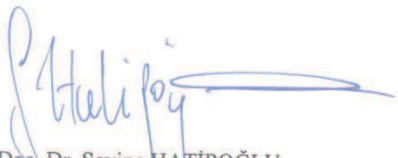
**İSTANBUL**

Bu çalışma 09/07/2014 tarihinde aşağıdaki jüri tarafından Enformatik Anabilim Dalı Enformatik programında Yüksek Lisans Tezi olarak kabul edilmiştir.

**Tez Jürisi:**

  
Prof. Dr. Sevinç GÜLSEÇEN (Danışman)  
İstanbul Üniversitesi  
Enformatik Bölümü

  
Prof. Dr. Rıza GÜVEN  
İstanbul Üniversitesi  
Mühendislik Fakültesi

  
Doç. Dr. Sevinç HATİPOĞLU  
İstanbul Üniversitesi  
Hasan Ali Yücel Eğitim Fakültesi

  
Doç. Dr. Melih KIRLIDOĞ  
Marmara Üniversitesi  
Mühendislik Fakültesi

  
Doç. Dr. L. Aylin AKTÜKÜN  
İstanbul Üniversitesi  
İktisat Fakültesi

## **ÖNSÖZ**

Yüksek lisans öğrenimim sırasında ve tez çalışmalarım boyunca gösterdiği her türlü destek ve yardımdan dolayı çok değerli danışmanım Prof.Dr.Sevinç GÜLSEÇEN'e en içten dileklerle teşekkür ederim.

Bu çalışma boyunca yardımlarını esirgemeyen çalışma arkadaşlarıma ve çalışmamın uygulama kısmını destekleyen Şebnem ÖZDEMİR'e teşekkürü borç bilirim.

Bu günlere gelmemde büyük pay sahibi olan aileme ve dostlarıma teşekkürlerimi sunarım.

Son olarak da her konuda beni destekleyen, gurbette olmadığımı hissettiren ve aynı zamanda bana sabırla Türkçe öğreten çok değerli, sevgili arkadaşım Emre AKADAL'a teşekkür ederim.

**Haziran, 2014**

**Bello MUHAMMAD**

# İÇİNDEKİLER

Sayfa No

ÖNSÖZ.....	i
İÇİNDEKİLER.....	ii
ŞEKİL LİSTESİ .....	v
TABLO LİSTESİ.....	vii
SİMGE VE KISALTMA LİSTESİ .....	viii
ÖZET .....	ix
SUMMARY.....	xi
<b>1 GİRİŞ.....</b>	<b>1</b>
1.1 PROBLEM.....	2
1.2 KONUNUN AMACI VE ÖNEMİ.....	3
1.3 ÇALIŞMANIN SINIRLANDIRILMASI.....	4
<b>2 GENEL KISIMLAR .....</b>	<b>5</b>
2.1 ÖĞRENCİ DEĞİŞİM PROGRAMI .....	5
2.2 ERASMUS PROGRAMI .....	6
2.3 İSTANBUL ÜNİVERSİTESİ ERASMUS PROGRAMI VE WEB SİTESİ .....	7
2.4 BİLGİ YÖNETİM SİSTEMİ.....	8
2.5 WEB TABANLI BİLGİ YÖNETİM SİSTEMİ.....	9
2.5.1 Veritabanı Entegrasyonu.....	11
2.5.2 Veritabanından Arama.....	11
2.6 SİSTEM ENTEGRASYON.....	12
<b>3 YÖNTEM VE ARAÇLAR.....</b>	<b>13</b>
3.1 YÖNTEM .....	13
3.1.1 Tezin Çalışma Planı.....	13
3.1.2 Literatür Tarama.....	14
3.2 ARAÇLAR .....	14
3.2.1 Java.....	14
3.2.2 UML (UNIFIED MODELING LANGUAGE).....	15
3.2.3 Eclipse IDE .....	15

3.2.4	Spring Framework .....	16
3.2.4.1	<i>Spring Application Context</i> .....	19
3.2.4.2	<i>Spring Security</i> .....	20
3.2.4.3	<i>Spring Access Control List (ACL)</i> .....	21
3.2.4.4	<i>Spring ACL ER Diyagramı</i> .....	22
3.2.4.5	<i>Spring Web XML</i> .....	23
3.2.5	Hibernate (ORM).....	25
3.2.6	Java Persisten API (JPA) .....	27
3.2.7	MySQL .....	28
3.2.8	Java Server Pages (JSP).....	28
3.2.9	Apache Tomcat Server.....	29
3.2.10	Maven .....	30
<b>4</b>	<b>ERASMUS DENEYİM PAYLAŞIM SİSTEMİ (EDEPAS) .....</b>	<b>32</b>
4.1	ERASMUS DENEYİM PAYLAŞIM SİSTEMİ (EDEPAS) .....	32
4.1.1	Sistem Tasarımı .....	32
4.1.2	Kullanıcı Ara Yüzü Sayfaları.....	32
4.1.2.1	<i>Giriş Yapma Ekranı</i> .....	32
4.1.2.2	<i>Kayıt Olma Ekranı</i> .....	33
4.1.2.3	<i>Ana Sayfa</i> .....	34
4.1.2.4	<i>Ülke Listesi</i> .....	34
4.1.2.5	<i>Deneyim Listesi</i> .....	35
4.1.2.6	<i>Deneyim Okuma Ekranı</i> .....	35
4.1.2.7	<i>Sistem Menüleri</i> .....	36
4.1.2.8	<i>Profil Güncelleme Ekranı</i> .....	36
4.1.2.9	<i>Duyuru Ekranı</i> .....	37
4.1.3	Yönetici Ara Yüzü Sayfaları .....	37
4.1.3.1	<i>Yönetici Ara Yüzü</i> .....	37
4.1.3.2	<i>Ülke Yönetme Ara Yüzü</i> .....	38
4.1.3.3	<i>Deneyim Yönetme Ara Yüzü</i> .....	38
4.1.3.4	<i>Kullanıcı Yönetme Ara Yüzü</i> .....	39
4.1.3.5	<i>Duyuru Posta Ekranı</i> .....	39
<b>5</b>	<b>BULGULAR.....</b>	<b>41</b>
5.1	SİSTEM GELİŞTİRME .....	41
5.2	USECASE DİYAGRAMI .....	41

5.2.1	EDEPAS ile İlişki.....	42
5.2.1.1	<i>Usecase Diyagramı.....</i>	42
5.2.2	Kayıt ve Giriş Aktivite Diyagramı .....	43
5.2.3	Deneyim Aktivite Diyagramı .....	46
5.2.4	E/R Diyagramı.....	47
5.2.5	Veritabanının Oluşturulması .....	48
5.2.5.1	<i>Hibernate ve JSR Anotasyon Tablosu .....</i>	49
5.2.5.2	<i>Hesap Tablosu ve Ülke Tablosu.....</i>	49
5.2.5.3	<i>Duyuru Tablosu.....</i>	51
5.2.5.4	<i>Deneyim Tablosu.....</i>	52
5.2.5.5	<i>İzin Tablosu.....</i>	53
5.2.5.6	<i>Kullanıcı Rolü Tablosu .....</i>	54
5.2.6	E-posta Sistemi.....	55
5.2.7	Kayıt Onaylama E-posta ve URL Ayarlama.....	56
5.2.8	“Bizeulasın” E-posta Ayarlama.....	57
5.2.9	Abone E-posta ve URL Ayarlama.....	57
5.2.10	Sistem Gereksinimler.....	58
5.3	SİSTEMİN TESTİ VE SONUÇLARI.....	58
<b>6</b>	<b>TARTIŞMA VE SONUÇ .....</b>	<b>60</b>
	<b>KAYNAKLAR.....</b>	<b>63</b>
	<b>ÖZGEÇMİŞ.....</b>	<b>66</b>

## ŞEKİL LİSTESİ

### Sayfa No

Şekil 2.1: İstanbul Üniversitesi Erasmus Programı Otomasyon Sistem Giriş Sayfası. ....	7
Şekil 3.1: Eclipse IDE Programının Genel Görünümü. ....	16
Şekil 3.2: Spring Framework'un yapısı. ....	17
Şekil 3.3: Spring ACL Beans Grafiği. ....	21
Şekil 3.4: Spring ACL Kod Bileşenleri ER Diyagramı. ....	23
Şekil 3.5: Apache Tomcat sunucu. ....	29
Şekil 3.6: Tomcat uygulama yükleme ara yüzü. ....	30
Şekil 3.7: Maven proje dosyaları. ....	31
Şekil 4.1: EDEPAS Giriş Yapma Ekranı. ....	33
Şekil 4.2: Kayıt Olma Ekranı. ....	33
Şekil 4.3: Sistem Ana Sayfa Ekranı. ....	34
Şekil 4.4: Ülke Liste Ekranı. ....	34
Şekil 4.5: Deneyim Liste Ekranı. ....	35
Şekil 4.6: Deneyim Okuma Ekranı. ....	35
Şekil 4.7: Sistem menüleri. ....	36
Şekil 4.8: Profil Güncelleme Ekranı. ....	36
Şekil 4.9: Duyuru Ekranı. ....	37
Şekil 4.10: Yönetici Ara Yüzü. ....	38
Şekil 4.11: Ülke Yönetme Ekranı. ....	38
Şekil 4.12: Deneyim Yönetme Ara Yüzü. ....	39
Şekil 4.13: Kullanıcı Yönetme Ekranı. ....	39
Şekil 4.14: Duyuru Posta Ekranı. ....	40
Şekil 5.1: Usecase Diyagramı. ....	42

<b>Şekil 5.2:</b> Usecase Diyagramı.....	43
<b>Şekil 5.3:</b> Giriş Aktivite Diyagramı.....	44
<b>Şekil 5.4:</b> Kayıt ve Giriş Aktivite Diyagramı. ....	45
<b>Şekil 5.5:</b> Deneyim Aktivite Diyagramı. ....	46
<b>Şekil 5.6:</b> Kod Bileşenleri E/R Diyagramı. ....	47



## TABLO LİSTESİ

	<b>Sayfa No</b>
<b>Tablo 3.1:</b> Spring Application Context. ....	19
<b>Tablo 3.2:</b> Spring Güvenlik XML Dosyası. ....	20
<b>Tablo 3.3:</b> Spring ACL Tabloların Açıklamaları. ....	22
<b>Tablo 3.4:</b> Spring Web XML. ....	24
<b>Tablo 3.5:</b> Spring, Hibernate ve Mysql Entegrasyon Dosyası. ....	26
<b>Tablo 5.1:</b> Sistem Tabloları. ....	48
<b>Tablo 5.2:</b> Hibernate ve JSR Anotasyon Tablosu. ....	49
<b>Tablo 5.3:</b> Hesap Tablosu. ....	50
<b>Tablo 5.4:</b> Ülke Tablosu. ....	51
<b>Tablo 5.5:</b> Duyuru Tablosu. ....	52
<b>Tablo 5.6:</b> Deneyim Tablosu. ....	53
<b>Tablo 5.7:</b> İzin Tablosu. ....	54
<b>Tablo 5.8:</b> Rolu Tablosu. ....	55

## SİMGE VE KISALTMA LİSTESİ

### Kısaltmalar Açıklama

<b>AB</b>	: Avrupa Birliđi
<b>VTYS</b>	: Veritabanı Yönetim Sistemi
<b>CBS</b>	: Cođrafi Bilgi Sistemleri
<b>SQL</b>	: Structured Query Language (Yapılandırılmış Sorgu Dili)
<b>ORM</b>	: Object Relational Mapping
<b>EDEPAS</b>	: Erasmus Deneyim Paylaşım Sistemi
<b>J2EE</b>	: Java 2 Enterprise Edition
<b>JDBC</b>	: Java Database Connectivity
<b>JVM</b>	: Java Virtual Machine
<b>UML</b>	: Unified Modeling Language (Birleşik Modelleme Dili)
<b>IDE</b>	: Integrated Development Environment (Tümleşik Geliştirme Ortamı)
<b>POJO</b>	: Plain Old Java Object
<b>EJB</b>	: Enterprise JavaBean
<b>JDK</b>	: Java Development Kit
<b>MVC</b>	: Model View Controller
<b>JDO</b>	: Java Data Objects
<b>AOP</b>	: Aspect Oriented Programming
<b>OOP</b>	: Object Oriented Programming (Nesne Yönelimli Programlama)
<b>DI</b>	: Dependency Injection
<b>XML</b>	: Extensible Markup Language (Genişletilebilir İşaretleme Dili)
<b>ACL</b>	: Access Control List
<b>ER</b>	: Entity Relation
<b>DD</b>	: Deployment Descriptor
<b>DAO</b>	: Data Access Object
<b>URL</b>	: Uniform Resource Locator (Birörnek Kaynak Konumlayıcı)
<b>JPA</b>	: Java Persistence API
<b>JSP</b>	: Java Server Pages
<b>JSR</b>	: Java Specification Request

## **ÖZET**

### **YÜKSEK LİSANS TEZİ**

## **J2EE KULLANILARAK WEB TABANLI ERASMUS HAREKETLİLİK BİLGİ SİSTEMİNİN GELİŞTİRİLMESİ**

**Bello MUHAMMAD**

**İstanbul Üniversitesi**

**Fen Bilimleri Enstitüsü**

**Enformatik Anabilim Dalı**

**Danışman : Prof.Dr.Sevinç GÜLSEÇEN**

İletişim teknolojilerindeki gelişmeler, sınır tanımayan rekabet ve küreselleşme sonucu, bilginin elde edilme hızı sadece işletmeler için değil eğitim kurumları için de oldukça önemli bir hale gelmiştir. Bilginin ifade ettiği bu değerle beraber 21. yüzyıl bilgi çağı haline dönüşmüştür. Bilginin bu şekilde koordinasyonu ve hızlıca kullanılabilir hale getirilmesinde bilişim sistemlerinin de katkısı bulunmaktadır.

Bilgi; değeri sürekli artan, rekabette üstünlük sağlayan ve kritik öneme sahip kurumsal bir kaynak olarak görülmektedir. Bununla birlikte stratejik bir kaynak olarak bakıldığında, deneyim ve bilginin, bilgi kaynaklarının bilinçli bir şekilde yönetilmesi bireyin başarısını artıran en önemli faktör olarak kabul edilmektedir. Benzer şekilde kurumsal bilgi kaynaklarının derlenmesi, organize edilmesi ve en önemlisi de bu işlemlerin bir sistem dahilinde yapılması bilginin yönetimi açısından büyük fayda sağlayacaktır. Bu açıdan değerlendirildiğinde Erasmus değişim programı ile giden ve gelen öğrencilerin üniversiteler açısından önemli bir bilgi kaynağını barındırdığı açıktır. Bu yolla kazanılan bilgi kaynaklarının doğru yönetilmesi gelecekteki öğrenci hareketliliğine önemli bir ivme ve kalite kazandıracaktır. Bu çalışma ile gerçekleştirilen sistem; deneyim yönetiminin başarıya ulaşması için en önemli bir unsur olan Erasmus hareketlilik deneyiminin derlenmesi ve etkili bir biçimde yönetilmesini amaçlamaktadır. Özellikle günümüzde İnternet tabanlı uygulamalarda yeni teknolojilerin kullanımı arttığı için bu tez kapsamında Erasmus deneyim paylaşım sistemi uygulamasının web tabanlı olarak geliştirilmesi tercih edilmiştir. Sistem sayesinde; tutulan veriler ile sadece deneyim yönetimi değil; öğrencilerin diğer öğrencilerin deneyimlerini inceleyerek, Erasmus programı için karar verme süreçlerine de destek sağlanacaktır.

Haziran 2014, 78 Sayfa.

**Anahtar kelimeler:** Bilgi Sistemi, Erasmus Hareketliliđi, J2EE Web Uygulama, Deneyim Paylaşım Sistemi.

## **SUMMARY**

**M.Sc. THESIS**

### **DEVELOPMENT OF WEB BASED ERASMUS MOBILITY INFORMATION SYSTEM BY USING J2EE**

**Bello MUHAMMAD**

**Istanbul University**

**Graduate School of Science and Engineering**

**Department of Informatics**

**Supervisor : Prof.Dr.Sevinç GÜLSEÇEN**

The advancement of telecommunication technologies nowadays, brought technological development, and competition recognizing no limit towards globalization; what mostly matters for organizations are knowledge and the rate of knowledge-acquisition. This is what made the early 21st century an era of information and skills set.

Knowledge has been increasingly seen as a critical institutional source which provides superiority in rivalry. Moreover, the management of knowledge, experience and knowledge sources as a strategic source is the most important factor that increases human success. In this context, it is necessary to collect, organize, and most importantly manage student knowledge and experiences for future reference. This study focuses on the collection of student and teachers educational travel experiences and management of that knowledge as experience for other students to benefit from such experience, which is the most important factor to achieve successful knowledge management.

The Erasmus mobility experience system was developed as a web-based information system because, nowadays information and communication technologies have gotten a new momentum in the area of knowledge management.

In this thesis, the plausible use of “Management Information Systems” as Erasmus student experience management system will be analyzed and used to capture, manage and process the knowledge past by the students through the Erasmus program. A complete system is build for that purpose named Erasmus Experience Management System.

June 2014, 78 Pages.

**Keywords:** Information system, Erasmus Mobility, J2EE Web Application, Erasmus Experience Sharing System.

## 1 GİRİŞ

Bilgi; günümüz dünyasında stratejik anlamda önemli bir değer ve rekabet avantajıdır. Bilginin en ham hali olan veri, özel kurumlardan eğitim kurumlarına kadar veritabanları aracılığıyla; toplanmakta ve saklanmaktadır. Veritabanları; farklı bilgi yapılarını ve büyük miktarda verileri düzenlemek, işlemek ve geri almak için kullanılan profesyonel yapılarıdır [1]. Günümüzde bireysel ya da kurumsal pek çok iş web üzerinde gerçekleştirilmektedir. Son dönemlerde özellikle bulut tabanlı yapıların da entegrasyonu ile internet üzerinden yapılan işlem talepleri ve işlemler daha etkili ve verimli şekilde yerine getirilmektedir.

Web tabanlı programlama; özellikleri, arayüzleri, istemci ve veritabanı arasındaki etkileşim için kullanılacak programlama dillerinden birini veya birden fazlasını kullanarak geliştirilmiş İnternet tabanlı bir yazılım geliştirme uygulamasıdır [2]. İnternet üzerinde web uygulama veya web tabanlı bilişim sistemi geliştirebilmek için PHP, Java, Python vb. gibi çeşitli programlama dilleri bulunmaktadır. Bu tez kapsamında geliştirilen sistem de bir web tabanlı yazılım uygulamasıdır. Bu sayede kullanıcılara zaman ve mekandan bağımsızlık imkanı sağlanmaktadır. Erişimdeki bu kolaylığın yanı sıra web tabanlı bilişim sistemleri ile, hızı yüksek güvenli erişim imkanı ve kullanıcıların bilgilerini bir bütünlük içinde görüntüleyebilmek de mümkündür. [3]. Bu avantajlardan bazıları veritabanı yönetim sistemi (VTYS) tarafından sağlanmaktadır, ancak uzman olmayan kullanıcıların VTYS arayüzü ile doğrudan çalışması pek de mümkün değildir. Ayrıca herhangi bir kullanıcının bu seviyede veri erişimine yetkisinin olması güvenlik açısından da bazı sakıncalar barındırmaktadır.

Türkiye’de Erasmus programından öğrenme programı, coğrafya veya kültürler bakımından referans olabilmektedir. Global öğrenmenin desteklenmesi, büyük ölçüde çevredeki değişim, bu değişime başarılı olarak uyum sağlama ve bilgi paylaşma yeteneğine bağlıdır. [4]. Türkiye’nin 2023 için belirlediği, Dünyanın ilk on ekonomisi arasında yer almak hedefini gerçekleştirebilmesi için teknoloji üretmesi şarttır [5].

Ancak bu teknoloji kapsamında üretilmesi gereken teknolojilerden biride öğrenme sürecini destekleyecek bilgi teknolojilerinin gerçekleştirilmesidir. Bu çalışmada Erasmus değişim programını kullanan/kullanacak öğrencilerin gerçekleştirilen bilgi sistemi ile; yeni bir deneyime hazırlanma sürecinde uygun tepkide bulunma, çevreye nasıl uyum sağlayacağını ve yaşamı sürdürebilmek temel gereksinimlerini ve diğer öğrencilerin deneyimlerini paylaşmaları yoluyla sunulmuştur. Bu sayede öğrencilerin yaşam boyu öğrenme felsefelerine gerçekten entegre olmaları sağlanacaktır. Hiç şüphesiz içinde bulunduğumuz bilgi teknolojisi çağında bireylerin var olabilmesi ve yaşam koşullarını iyileştirebilmesi için sahip olmaları gereken en önemli becerilerden biri de yaşam boyu öğrenmedir [6].

## 1.1 PROBLEM

Yüksek öğretim sürecinde global öğrenmenin desteklenmesi, genç neslin farklı kültürler ve insanlarla tanışarak bakış açılarını genişletmeleri açısından Erasmus gibi değişim programları oldukça önemlidir. Ancak bir değişim programında kazanılan deneyimler ve yeni öğrenmeler; örgütlü bilgiler halinde düzenlenerek paylaşılmadıkları müddetçe bu deneyimi yaşayan bireyle sınırlı kalabilmektedir. Bu önemli bilgi yığınının bilgi teknolojisi aracılığıyla aynı deneyimi yaşamak isteyen potansiyel kitleye aktarılması, hem öğrenmenin sürekliliğini hem de bilginin paylaşılmasına katkı sağlayacaktır.

Bu açıdan değerlendirildiğinde; değişim programlarında kazanılan bilginin organize edilmesi, depolanması ve paylaşılması amacıyla kullanılacak bir bilgi sisteminin olmayışı, önemli bir boşluğa işaret etmektedir. Bu tezin amacı; oluşan bu boşluğu doldurmak amacıyla; İstanbul Üniversitesi Erasmus Değişim Programı kapsamında, belli yetki seviyeleri olan, giden ve gelen öğrencilerin bireysel deneyimlerini paylaşabilecekleri web tabanlı bir bilişim sistemi geliştirmektir. Bu tez kapsamında; sistemin geliştirilmesi açısından önemli olan öncelikle bilgi ve bilgi teknolojileri konuları ortaya konulmuş, ardından sistemin gerekliliği ve nasıl geliştirildiği ve son olarak da kullanımı ile ilgili bilgilere yer verilmiştir.



## 1.2 KONUNUN AMACI VE ÖNEMİ

Bir bilişim sistemi yönetiminin temeli insandır. İnsanlar bilgi, yeni fikirler ve yeni ürünleri yaratır ve tamamen iş süreçlerini oluşturan bağlantıları kurarlar [7]. Bilişim sistemi, belirli amaçları başarmak için entegre olmuş biçimde girdiler, süreçler, çıktılar ve geri bildirim içeren kısımların toplamı olan büyük bir bütündür, bir diğer tanıma göre ise bilişim sistemi; belirli görevlerin gerçekleştirilmesi için organize edilmiş insanların, makinelerin ve yöntemlerin toplamıdır [8]. Bilişim sistemleri e-ticaret, eğitim, coğrafya v.b. gibi çok çeşitli uygulama alanlarına sahiptir. Bu sistemler çok büyük veri yönetmek ve özelleştirilmiş hizmetler sunmak için tasarlanmıştır. Ayrıca bilgiyi çoklu ortam (multimedia) biçiminde sunarken genellikle gömülü navigasyon araçları (hiper bağlar ve/veya dinamik web sayfaları aracılığıyla) kullanılmaktadır. Tüm bu özellikler düşünüldüğünde web tabanlı bilişim sistemlerinin oldukça çekici bir yapıya sahip olduğu söylenebilir.

Bu tez çalışmasında; öğrencilerin daha global bir anlayışa sahip olması, içinde buldukları dünyayı daha iyi algılamaları amacı ile oluşturulan Erasmus ve Mevlana gibi çeşitli öğrenci değişim programlarında kullanılacak Web tabanlı bir bilişim sistemi geliştirilecektir. Geliştirilen sistem ile öğrenciler farklı ülkelerde kazandıkları deneyimleri paylaşabilecek, değişim programına katılmak isteyen öğrenciler de, benzer ilgilere ve önkoşullara sahip olan diğer öğrencilerin deneyimlerini, izlenimlerini öğrenebileceklerdir.

Sistem ayrıca aşağıda verilmekte olan avantajları sağlayacaktır:

- Öğrenci, öğretim üyesi ve bilim insanlarının zaman yönetimine yardımcı olmak;
- Değişim programına yönelik işlemlerde daha az sayıda program koordinatörü ile sürecin yönetilmesini sağlamak ve bu sayede mevcut personel eksikliğinin giderilmesine katkı sağlamak;
- Değişim programına katılmış ve katılacak olan tüm öğrencilerin ortak bir yapıda buluşmasını sağlamak;

- Değişim programı sürecinde, öğrencinin ihtiyaç duyduğu bilgilere erişmesinde üniversite personeline ve öğretim üyelerine daha az bağımlı olmasını sağlamak.

### 1.3 ÇALIŞMANIN SINIRLANDIRILMASI

Bilgi ve iletişim teknolojileri yakın dönemin en popüler ve gerekli kaynakları haline gelmiştir. Bu teknolojilerin yaşamın her alanına yayılmış olması ve çeşitliliği nedeniyle belli bir çatı altında toplama zorunluluğu bulunmaktadır. Nitekim örgütsel düzlemde açılan her kapının ardında farklı bir iletişim teknolojileri aracına rastlanılmakta ve her birey direkt ya da dolaylı olarak bu araçlardan istifade etmektedir. Bilgi ve bilgi teknolojilerinin değişim hızının oldukça yüksek olması, dünün bugünden farklı olmasına ve yeniliklerin takibinin zorlaşmasına neden olmaktadır.

Bilgi teknolojilerindeki bu süratli değişim ve ürün yelpazesindeki çeşitlilik kullanıcıların algılaması ve araştırmanın somut bir şekilde ortaya konulmasını zorlaştırmaktadır. Bilgi teknolojilerinin çeşitli ürünlerinin olmasına rağmen bazı sistemlerin okullarda pek tanınmıyor ya da az kullanılıyor olması çalışmanın sınırlılıklarından birini oluşturmaktadır. Teze ilişkin bir diğer sınırlılık ise sistemdeki öğrencilerin deneyimlerinin sadece değişim programı kapsamında gittikleri ülkelerle sınırlı tutulmasıdır.

Tez kapsamında gerçekleştirilen sistem; global öğrenmenin yanısıra eğitim kurumunun örgütsel performansını da destekleyecektir. Ancak İnternette var olan sistemlerin büyük çoğunluğunun bireysel performans üzerine olması, örgütsel performansı içeren ve eğitim alanında özellikle bu tür sistemleri destekleyen araştırmaların azlığı tezin güçlü bir yönünü oluştururken, bilgi anlamında da bir diğer sınırlılığını göstermektedir.

Araştırmayı destekleyen çalışmalarda farklı boyutların bulunması ana çerçevenin oluşturulmasında önemli sınırlılıklar oluşturmuş olsa da bu çalışmalardan sağlanan veriler temel yapıya zemin hazırlamıştır.

## 2 GENEL KISIMLAR

Genel kısımlar bölümünde tezin konusu ile ilgili yapılan arařtırmalar, çalışmanın unsurları ve geliştirilen sistemde kullanılan teknolojileri açıklanmıştır.

### 2.1 ÖĞRENCİ DEĞİŐİM PROGRAMI

Öğrenci deęişim programı olarak adlandırılan öğrenci ve öğretim üyesi deęişim programı, üniversite ve yüksek teknoloji enstitüleri bünyesinde ön lisans, lisans, yüksek lisans ve doktora düzeyinde eğitim-öğretim yapan yükseköğretim kurumları arasında öğrenci ve öğretim üyesi deęişim programıdır [9]. Türkiyede örgün eğitim veren yükseköğretim kurumlarında okuyan ön lisans, lisans, yüksek lisans ve doktora öğrencileri ile idari kadroda yer almayan tüm öğretim üyeleri Farabi, Erasmus ve Mevlana gibi deęişim programlarından faydalanabilir [10].

Deęişim programları öğrencilerin eğitim öğrenim hayatında önemli bir deneyimi ifade etmektedir. Farklı kültürlerle 6 ay ila 1 yıl arası deęişen etkileşim; dil, bakış açısı, iletişim becerileri gibi pek çok unsurda gelişim ve deęişim sağlamaktadır.

Avrupa’da Erasmus öğrencilerinin deneyimlerini baz alan bir araştırma yapılmıştır. Araştırma kapsamında sorulan; “Erasmus programının kazandırdığı deneyimleri geleceğiniz için ne kadar önemlidir” sorusuna katılımcıların %58.1 “çok önemli”, %33.1 sadece “önemli” cevabını vermiştir. Katılımcıların ancak %8.9’u önemli olmadığını ifade etmiştir [11]. Yine benzer bir başka arařtırmada, öğrencilerin deęişim programları sayesinde farklı eğitim deneyimleri ve çeşitli bilgiler kazandıkları ortaya konulmaktadır [12]. Böylece deęişim programı ile elde edilen deneyimin oldukça değerli bilgiler barındırdığı, öğrencinin kişisel ve gelecekteki mesleğine yönelik gelişimine katkıda bulunduğu söylenebilir.

## 2.2 ERASMUS PROGRAMI

Erasmus programı, Avrupa'daki yükseköğretim kurumlarının birbirleri ile çok yönlü işbirliği yapmalarını teşvik etmeye yönelik bir Avrupa Birliği (AB) programıdır [13]. Başka deyişle; Erasmus Avrupa'da yükseköğretim kalitesini artırmak amacı taşıyan ve üniversiteler arasında ülkelerarası işbirliğini teşvik ederek "üniversite öğrencisinin uluslararası hareketliliğini" öngören bir yüksek öğretim programıdır [14]. Yükseköğretim kurumlarının birbirleri ile ortak projeler üretip hayata geçirmeleri, kısa süreli öğrenci ve akademik personel değişimi yapabilmeleri için hibe niteliğinde karşılıksız mali destek sağlamaktadır [15]. Programın amacı; AB ülkeleriyle aday ülkelerin yükseköğretim kurumları arasındaki işbirliğini teşvik edip geliştirerek yükseköğretimde Avrupa boyutunu ön plana çıkarmaktır. Bu amaçla Erasmus programı kapsamında her yıl binlerce öğrenciye ve öğretim görevlisine eğitim ve öğretim faaliyetlerinin bir kısmını yurtdışında geçirme imkanı tanınmaktadır [16].

Erasmus programı, Hayatboyu Öğrenme Programına dahil ülkeler olan Avrupa Birliği üyesi 27 ülke, AB üyesi olmayıp Avrupa Ekonomik Alanı üyesi İzlanda, Lihtenştayn, Norveç, İsviçre ve AB üyesi olmaya aday Türkiye ve Hırvatistan yüksek öğretim kurumlarının istifadesine açıktır [17]. Erasmus eylemi oluşturulurken hedefler arasında Avrupa'da yükseköğrenimin kalitesini arttırmak, farklı kültürler ve yaşam biçimlerine karşı toleranslı yaklaşabilme becerisini geliştirmek ve dünyanın en rekabetçi ekonomileri arasında yer alabilmek için küreselleşme sürecinde istihdam piyasasının niteliklerine uygun eleman yetiştirebilmek amaçlanmıştır.

### Erasmus Kapsamı

- Erasmus 1: Avrupa İşbirliği Projeleri
- Erasmus 2: Ülke Merkezli Faaliyetler
- Erasmus 3: Akademik Uzmanlık Ağları (Thematic Networks)

### Erasmus kapsamında desteklenen etkinlikler

- Öğrenci ve öğretim görevlisi değişimleri,
- Eğitim programlarının geliştirilmesine yönelik işbirlikleri,
- Dil kursları ve yoğun programlar

### 2.3 İSTANBUL ÜNİVERSİTESİ ERASMUS PROGRAMI VE WEB SİTESİ

İstanbul Üniversitesi 2003/2004 öğretim yılında pilot proje ile katılmış olduğu Erasmus programını 2004/2005 öğretim yılından itibaren lisans ve lisansüstü düzeyde öğrenci ve öğretim elemanı değişimini Erasmus beyannamesiyle sürdürmektedir [18].

İÜ Uluslararası Akademik İlişkiler Kurulun'a bağlı olarak faaliyet gösteren Erasmus ofisi öğrenci, öğretim üyesi ve ilgili diğer kişilere yönelik işlemlerini elektronik ortama taşımış ve <http://erasmusprogram.istanbul.edu.tr/Students/Login/Login.aspx> web sitesi Şekil 2.1'de aracılığı ile hizmet vermektedir.

**Şekil 2.1:** İstanbul Üniversitesi Erasmus Programı Otomasyon Sistem Giriş Sayfası.

İlgili web sitesi İstanbul Üniversitesi Enformatik Bölümü tarafından geliştirilmiştir. Erasmus hareketliliği ile ilgili bilgi almak isteyenler siteye girdiklerinde, ilgili linkler aracılığı ile bunu yapabilmektedirler. Oldukça yoğun bilgi içeren Erasmus web sitesinde, hareketlilikten yararlananların önerilerini, deneyimlerini ve görüşlerini sunabileceği bir bölümün eksikliği göze çarpmaktadır. Günümüzde çok önemli olan bilgi ve bilginin yönetimi için güncel BİT'den yararlanılabilir ve mevcut sistem ve ortamlara bu fonksiyonu icra eden eklentiler yapılarak, bunların etkinliği artırılabilir.

## 2.4 BİLGİ YÖNETİM SİSTEMİ

Bilgi teknolojilerindeki hızlı ilerleme bilgi kavramını ön plana çıkarmıştır. Bilgi, büyük ölçüde bilmeye, kavramaya ilişkin ileri düzeyde kişisel bir etkinliktir [19]. Gerçekten de bilginin politikada, sanatta, iş ve insan ilişkilerinde kısaca yaşamın her alanında önemi hissedilmektedir. Bu açıdan değerlendirildiğinde; bilginin yönetilmesinin oldukça önemli bir olgu olduğu ortaya çıkmaktadır.

Bilgi yönetimi, doğrudan işletme içinden veya işletme dışı kaynaklardan edinilen verilerin tasnif edilmesi, stoklanması, yorumlanmak üzere ilgili yerlere gerekli zamanlarda dağıtılması, güncellenmek üzere gözden geçirilmesi sürecidir [20]. Bilgi yönetimi, ve eğitim, öğrenim ve deneyimlerin kurumsal faaliyetlere yansması sonrasında oluşan bireysel ve kurumsal, kayıtlı ya da kayıtlı olmayan her türlü bilgi kaynağının belirlenmesi, tanımlanması, yönetilmesi ve paylaşılması işlemlerini organizasyonun yapısına göre uyarlayan ve uygulayan yönetim biliminin bir alanıdır [21].

Bilgi yönetim sistemi, bir örgütteki veri ve bilgi akışını güçlendirir ve çeşitli görevleri yürütmekle sorumlu bireylere ve gruplara sunar. Firmanın sadece pazarlama, üretim vs. Konularında gelişme göstermesi, değişime ayak uydurması başarılı olması için yeterli değildir. Pazar koşullarıyla birlikte yönetim anlayışı da gelişmekte bilgi yönetimi gibi yeni bileşenlerini doğurmaktadır. Bugünün koşullarında örgütlerin sahip olduğu ve etkin olarak kullandığı bilgi, onların entelektüel malvarlığını oluşturmaktadır. Bilgi çeşitli mekanizmalar ve teknolojiler yardım ile yakalanabilir ve daha sonra diğer insanlar tarafından kullanılabilir ve paylaşılabilir. Bilgi paylaşma sistemleri, örgüt içinde bilginin doğru ve etkili bir şekilde paylaşılmasını sağlar [22].

Bilginin çok önemli bir konumda bulunmasından ötürü, günümüz yönetim ve örgüt teorisinde bilgi, yönetim uzmanları tarafından örgütler ve/veya işletmeler için klasik üretim unsurlarına ek bir unsur olarak anılmaya ve değerlendirilmeye başlanmıştır [23]. Bilgi yada Enformasyon toplumunda bilginin kullanımı arttıkça, üretimin yapısı da değişmekte, bilgi yönetimi işgücü ve semayeden daha önemli bir etken olarak üretime girmektedir. Dolayısıyla bilgi yönetimi yalnızca global işletmeler açısından değil, bütün işletmeler açısından önemlidir. Enformasyon genellikle, bireyler veya kurumlar

tarafından bir sorunun çözümü, herhangi bir çalışmanın başlatılması ya da bitirilmesi gibi faaliyetler sonucunda ortaya çıkarılan verilerin bütünü ifade etmektedir. Bilgi ise herhangi bir konuda insanların kendi alanlarında edindikleri enformasyonu yorumlayarak elde ettiği kararlar ve yeteneklerdir [24]. Gelecekte, birçok bilgi yönetimi etkinliği ve süreci, işletme bölümlerinin veya iş görenlerin olağan bir işi durumuna gelecektir. Bununla beraber, proaktif düşünen işletmeler gelecekte rekabet edebilmek için, sistematik olarak bilgiyi daha iyi yönetme çabası içine gireceklerdir.

Bilgi yönetiminin en önemli yapı taşı enformasyondur. Bilgi yönetimi ile çalışanların var olan enformasyona erişimini mümkün kılınarak enformasyon kaynaklarından yeni bilgilerin üretilmesini sağlamaktır. Bilgi yönetimi, örgütsel amaçları başarmak için bilginin nasıl yaratılabileceği, elde edilebileceği, kullanılabilirliği ve yönetilebileceğine ilişkin sistematik bir süreçtir. Bilgi yönetiminde, iki temel bilgi türü vardır, bunlar; açık bilgi ve örtülü veya içsel bilgidir. Açık ve örtülü bilgiler birbirleri arasında ve kendi içlerinde dönüşüm işlemleri geçirdiklerinde, bilginin yaratılması ve paylaşılması gerçekleşir. Bu nedenle, bilgi elde edilmesinde yöneticilerin dikkat etmesi gereken iki önemli nokta vardır. Bunlardan birincisi bilginin güvenilir olması ikincisi ise, kurumların ihtiyaçlarına cevap verebilmesi ve rekabet avantajı kazandırabilmesidir [25].

## **2.5 WEB TABANLI BİLGİ YÖNETİM SİSTEMİ**

Bilgi sistemleri, bilgi ve veriyi işlem sürecinden geçirerek, anlamlı çıktılara dönüştürürler [26]. Web ortamından faydalanmak kaydıyla kurgulanmış bir bilgi sisteminde, verilere internet bağlantısı kurulabilen her noktadan her an ve her yerden ulaşılabilir. Veritabanında yapılan değişikliklerden anında haberdar olunması güncel veriden faydalanma imkânını beraberinde getirmektedir. Web tabanlı bilgi yönetim sistemi teknik olarak, organizasyonlardaki karar verme desteğine kadar bilgiyi düzenlemek, saklamak, işlemek, toplamak olan birbirleriyle ilgili parçaların kümesi olarak tanımlanabilir.

Web tabanlı öğretim sistemlerinin bilgi ve karar verme anlamında sağladıkları bu avantajlar dışında öğrenme ve öğretme sürecine de katkıları bulunmaktadır. Bu sistemler sayesinde daha fazla öğrenciye, zaman ve mekan sınırlılıkları olmadan ulaşma imkanı bulunmaktadır. Benzer şekilde web ortamının sağladığı iletişimle öğrencilerin

düşünceleri zenginleşir, gerçek problem durumlarına ilişkin çözümler ve düşünceler geliştirirler [27].

Bilişim sistemleri, organizasyon içinde ve çevresinde önemli insanlar, yerler vb hakkında da bilgi içerirler. Zaman zaman bazı verilerin farklı veri kaynaklarından temin edilmesi gerekebilir. Dolayısıyla bu verilere ulaşmak hem maliyetlidir ve hem de ekstra uğraş gerektirir. Ancak web tabanlı bilgi yönetim sistemleri ile dünyanın herhangi bir yerindeki veri kaynağına erişmek ve o verileri kullanmak mümkündür.

İşletmeler açısından son yirmi yıla kadar bilgi kendi başına önemli bir varlık olarak gözükmemektedir. İşletmeler; gerekli olan bilgileri yüz yüze görüşmelerle, çalışanların becerisiyle ya da ikinci elden elde ederek bununla yetinmektedir. Ancak, küreselleşmesi sonucu bilginin ve buna bağlı bilgi sistemlerinin yayılması ve gelişmesiyle birlikte işletmelerin her bölümünde bilgiye olan gereksinim artmaktadır. Bilgi sistemlerini kullanmayan işletmeler, ne gelişme gösterebilirler ne de değişime ve gelişime ayak uydurabilirler. Örnek olarak, web tabanlı Coğrafi Bilgi Sistemleri (CBS) verilebilir. İnternet üzerindeki CBS uygulamalarının sahip olduğu olanaklar ve kolaylıklar çok sayıda kullanıcı tarafından fark edilmiş ve buna bağlı olarak birçok CBS uygulaması internet tabanlı olarak değişik amaçları yerine getirmek üzere kullanıcıların hizmetine sunulmuştur [28].

Bir bilgi sisteminde üç aktivite bulunmaktadır;

Bu aktiviteler:

- Girdi (input): organizasyonun içinden veya dış çevresinden, ham bilgileri (data) ele geçirmek veya toplamaktır.
- İşlem (processing): bu ham veriyi daha anlamlı ve işlenmiş bilgiye (information) çevirmektir.
- Çıktı (output): işlenmiş bilgiyi (information), insanlara veya kullanılacak olan aktivitelere aktarmaktır.

Bilişim sistemleri bilgisayarla sınırlı değildir. Bilişim sistemlerini etkili kullanmak, sistemleri şekillendiren organizasyonu, yönetim ve bilişim teknolojilerini anlamayı gerektirir.



### 2.5.1 Veritabanı Entegrasyonu

Web sistemlerinde bilgi saklayabilmek ve bu bilgiler üzerinde sorgulama yapabilmek için sistemin bir “veri tabanı” desteğine ihtiyacı bulunmaktadır. Windows sunucu üzerindeki uygulamalar için MS SQL veri tabanı, hem LINUX hem de Windows sunucu üzerindeki uygulamalar için MySQL ya da Oracle veritabanı yönetim sistemlerinden destek alınabilir.

Farklı organizasyonların tuttuğu verilerin kullanıcıların ve yöneticilerin erişimi kolaylaştırmak için bir araya getirmesine veri entegrasyonu denmektedir. Kurumsal Uygulama Tümüleşimi, basit bir şekilde, bir organizasyonun bilgi işlem ortamını oluşturan farklı sistemler arasında standart köprüler oluşturulması süreci olarak tanımlanabilir [29]. Veritabanı entegrasyonu yalnızca birden fazla veritabanının bir araya gelmesi değildir. Günümüzde kurumsal web uygulamaları için kullanılacak faydalı paketler bulunmaktadır. Java Enterprise Edition kullanırken Spring Framework ve Hibernate gibi bazı frameworklerden faydalanmak gerekmektedir. Bunları veritabanı entegrasyonu ile birlikte kullanmak güçlü bir altyapının kazanılmasını sağlayacaktır. Altyapı entegrasyonu Spring framework tarafından “Dependency Injection” şeklinde ifade edilmektedir, Hibernate tarafından ise “Object Relational Mapping (ORM)” ifadesi kullanılmaktadır. Hibernate sınıfları (entity) direct veritabanına tablo olarak tanımlanıyor.

### 2.5.2 Veritabanından Arama

Sistem içerisinde, onlarca veritabanı içerisinde arama yapılabilmesini kolaylaştıracak herhangi biri tercih edilebilir. Çalışma içerisinde, geliştirilen sistem için tercih edilmiş olan MySQL veritabanında arama desteğinin ne şekilde gerçekleştiği ile ilgili ayrıntılı bilgi sunulmaktadır. Sitemizin kullandığımız terimler, alışkın olmayanlar için çoğu kez yabancı gelebiliyor. Veritabanında arama işlemini gerçekleştirebilmek için MySQL veritabanına özgü bazı komutların kullanılması gerekmektedir. Kullanılan veritabanının resmi dokümantasyonunda, kullanılması gereken komutlar ve uygulama adımları verilmektedir. Geliştirilen sistem içerisinde arama yapabilmek için bulunan bir metin kutusu bulunmaktadır. Kullanıcılar, sistemde arama yapmak için bu kutuyu kullanarak aramaya yapabilmektedirler. Girilen metin veritabanında bir satır ile eşleşirse sonuçlar kullanıcıya gösterilir. Sonuç bulunamadığı takdirde kullanıcı bir mesaj ile uyarılır.

## 2.6 SİSTEM ENTEGRASYON

Sistem Entegrasyonu yazılım göre bugün için kullandığımız sistemlerin ve ya yazılım araçların bütünleşikliğinin sağlanması olarak algılansa da bu aslında bunun çok daha ötesinde bir hizmetler bütünüdür. Uzmanlıkların, yazılım ihtiyaçları doğrultusunda baştan sona süreci kesintisiz hale getirecek bütünleşik çözümler için kullanılması sonucunda varılan sürdürülebilir çözümlerdir. Sistem entegrasyonu bir takım avantajlar sağlamaktadır:

- Güçlü bir çözüm yazılım mimarisinin hayata geçirilmesi
- Varolan sistemlerin bütünleşik bir yapıda çalışmasının sağlanması
- Teknolojik gelişmeler ışığında, işinizi her zamankinden daha avantajlı olarak yapacak yeni platformlar oluşturulması
- Kurulan entegrasyonun kopmaması için gerekli tüm önleyici tedbirlerin sistem içinde tasarlanması

Bütünleşik yapıyı hayata geçirdikten sonra, nasıl yönetileceğinin bilgisini sorumlu kişilere aktarılması ve zaman içinde oluşan sorulara doğru cevaplar verilmesi sistem entegrasyonunun başarı ile sürdürülmesi için gerekli koşuldur.

### 3 YÖNTEM VE ARAÇLAR

Bu tez çalışmasında İstanbul Üniversitesi Erasmus Değişim programı ile giden öğrencilerin hareketliliği ile oluşan deneyimleri yönetme ihtiyacına bağlı olarak bir web tabane bilgi yönetim sistemi geliştirilmiştir. Sistem sayesinde değişim programı ile giden ve gitmek isteyen öğrencilerin elektronik ortamında bir çatı altında toplanması mümkün olmaktadır. Böylece gidilecek ülkelerin kararının verilmesinde, yaşanan temel zorlukların neler olduğu, kültürel alışkanlıklardan doğan sorunlar gibi pek çok bilgi Erasmus Deneyim Paylaşım Sistemi (EDEPAS) olarak adlandırılan sistem ile doğru bir şekilde yönetilebilecektir.

#### 3.1 YÖNTEM

EDEPAS tasarımı için; ihtiyaç analizi ve literatür taraması yapılmıştır. İhtiyaç analizinin ardından fizibilite çalışması yapılarak sistemin geliştirilmesinde kullanılacak yazılımlar belirlenmiştir. Son olarak sistem çevik (agile) yazılım metodolojisi kullanılarak geliştirilmiş ve testi yapılmıştır.

##### 3.1.1 Tezin Çalışma Planı

Tez çalışmasının amacına ulaşabilmesi için öncelikle Java Tabanlı Açık Kaynaklı Taslaklar, ORM, JPA, RDBMS ve Web Sunucusu entegrasyonu derinlemesine incelenmiştir. Bu teknolojilerin tek bir sistem içinde birlikte çalıştırılması, günümüzün kurumsal web tabanlı siteleri arasında en iyi olmasını sağlayacaktır.

Bahar dönemi boyunca, JPA destekli Hibernate (ORM), JSP uygulaması, Maven, MySQL veritabanı sunucusu ve Apache Tomcat web sunucusu kullanılarak, sistemin kurulacağı altyapı oluşturulmuştur. Ayrıca bir sistem mimarının, web tabanlı bir bilisim sistemi ile yapılandırabileceği farklı türdeki olasılıklar üzerinde çalışılmıştır. Bu süreç içinde hangi desenlerin web tabanlı sistemin farklı tipleri için en uygun olduğunu anlayabilmek için tasarım desenleri üzerine de araştırmalar yapılmıştır. Bu çalışmaların

ardından açık kaynak kodlu yazılımlar kullanılarak, Erasmus Deneyim Paylasim (EDEPAS) adi verilen sistem oluşturulmuştur.

### 3.1.2 Literatür Tarama

Java Enterprise Edition hakkında temel ve ileri olmak üzere pek çok kaynak bulunmaktadır. Bu kaynaklardan en bilindik olanı Expert One-on-One J2EE Design and Development isimli kitaptır. Bunun dışında kurumsal uygulama geliştirmek için bazı hatalara yer veren kaynaklardan biri de efektif Java'dır. Bu tez çalışmasında Java frameworkları ve veritabanı nesne ilişkisel haritalama frameworkları ile ilgili çalışmalar incelenmiştir.

## 3.2 ARAÇLAR

Kurumsal bilgi sistemi geliştirilmesi aşamasında programlama dili seçiminin dikkatli bir şekilde yapılması gerekmektedir. Yazılım firmalarında gerek dokümantasyon gerekse sağladığı güçlü yazılım geliştirme ortamı sebebiyle genellikle Java dili tercih edilmektedir. Üniversiteyi bir firma gibi düşündüğümüzde arz ve talebi doğru karşılamak adına açık kaynak kodlu Java programlama dili tercih edilmiştir.

Sistem geliştirilmesinde veri tabanı yönetimi için MySQL veritabanı tercih edilmiştir. MySQL ve Java arası birlikte işlerliği kolaylaştırmak için Java'yı ilk geliştiren firma olan Sun tarafından "Java Database Connectivity (JDBC)" geliştirilmiştir [30]. Bu çalışmada hibernate üzerinde Java'dan JDBC kullanarak bağlanmıştır.

### 3.2.1 Java

Java; ilk adı Oak olan daha sonra adı Java olarak değiştirilen, Sun Microsystems mühendislerinden James Gosling tarafından geliştirilmeye başlanan, açık kodlu, nesneye yönelik, zeminden bağımsız, yüksek verimli, çok işlevli, yüksek seviye bir programlama dilidir [31]. Java programının bazı avantajları aşağıdaki gibidir:

- Açık kaynak kodu
- Güçlü dokümantasyon
- Yüksek güvenlik
- Yüksek performans

- Çoklu veritabanı desteđi
- Tařınabilirlik
- Bir kez yazılma her yerde çalıştırabilme
- Çoklu donanım desteđi
- Nesne yöneliml olması
- Java programının bazı dezavantajları ařađıdaki gibidir:Yüksek maliyet
- Öğrenim ve kullanım zorluđu
- Küçük bir yazılım için karmařık kodu gerektirebilmesi
- Sadace Java Virtual Machine (JVM) ile çalıştırılabilmesi.

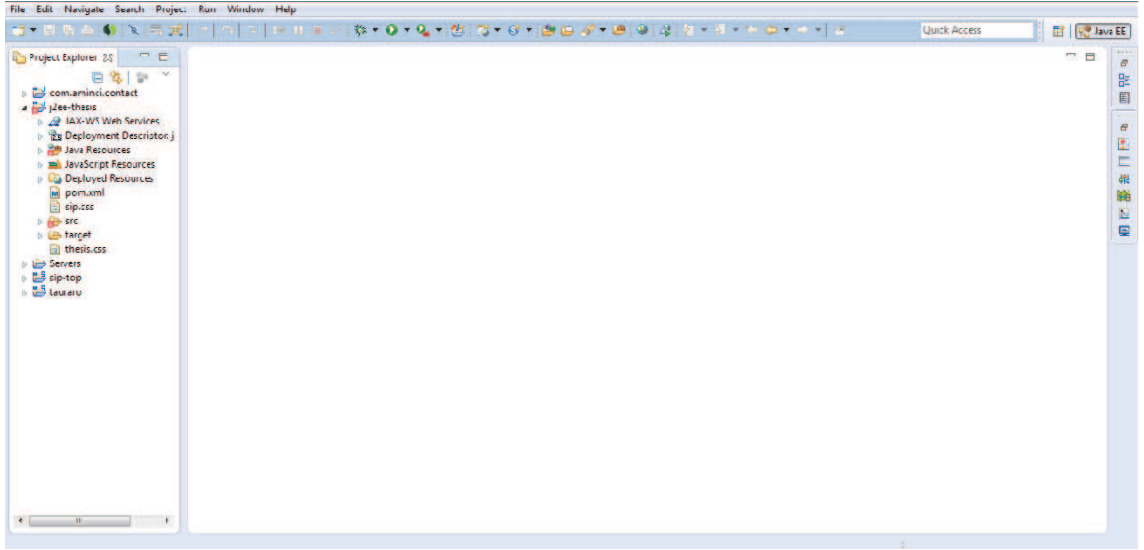
Javanın diđer çok önemli özelliklerinden birisi nesne kökenli (object oriented) bir dil olmasıdır. Nesne kökenli diller, nesneleri gerçek dünyadaki benzer bir yapıda tanımlayarak anlaşılmasını kolaylařtırmaktadır.

### **3.2.2 UML (UNIFIED MODELING LANGUAGE)**

UML (Açık Kaynak Kodlu: Birleşik Modelleme Dili), bir yazılım sisteminin olgularını anlatmak ve tanımak, sunmak ve oluşturmak ya da sistemin dokümanete etmek için kullanılabilir. Farklı programlama dilleri kendi içinde geliřtirdiđi bir takım komut argümanın programcı tarafında iyi seviyede ezberlenmesi gerektiđi için olası yeni teknolojik ihtiyaçları karşılama sürecinde programcının yeniden kendini geliştirme adıyla bir eğitim sürecine dahil olması kaçınılmazdır [32]. Bu dil UML gibi, yazılımların gerçekleştirilme süreci içinde önemli bir dildir.

### **3.2.3 Eclipse IDE**

Eclipse, güçlü ve kullanıcıların kolayca kullanabileceđi bir programlama editörüdür. Java uygulama, c++ uygulama, PHP uygulama ve Android uygulama gibi pek çok programlama dili için de kullanılabilir. Şekil 3.1'de Eclipse ara yüzü verilmektedir.



**Şekil 3.1:** Eclipse IDE Programının Genel Görünümü.

Java yazılımlarında Eclipse kullanılması nedeniyle, bütün Java frameworkleri tarafından desteklenmektedir. Otomatik olarak kodu tamamlama, anahtar kelimelerin farklı renklendirilmesi, otomatik olarak kod hatasının işaretlemesi yazılımcılara önemli bir avantaj sağlamaktadır. Bu avantajlarından dolayı tez çalışmasında sistemin geliştirilmesi için Eclipse IDE yazılımı kullanılmıştır.

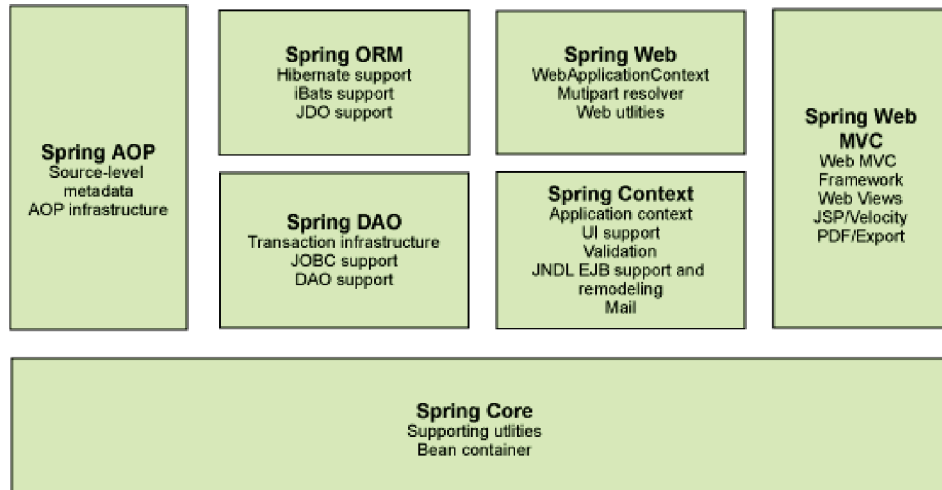
### 3.2.4 Spring Framework

Spring kütüphanesi J2EE geliştirmeyi daha kolay hale getirmek amacıyla açık kaynak kodlu olarak geliştirilen bir java platformudur. J2EE, Java 2 Platform Enterprise Edition ifadesinin kısaltılmış halidir. Sun tarafından geliştirilen J2EE, web tabanlı çevrimiçi uygulamalar geliştirmek amaçlı platform bağımsız, java merkezli geliştirme ortamıdır. Spring basit, taşınabilir, hızlı ve esnek JVM tabanlı sistemler ve uygulamalar geliştirilmesine olanak sağlamaktadır [33].

Spring, kurumsal uygulamalar geliştirmek için en iyi java geliştirme ortamıdır. Dünya genelinde milyonlarca geliştirici Spring Framework üzerinde kolay test edilebilir, yüksek performanslı uygulamalar geliştirmektedir. Rod Johnson tarafından geliştirilmiş olup ilk olarak 2003 yılında Haziran ayında Apache 2.0 lisansı altında yayınlanmıştır. Spring Framework'ün temel özellikler herhangi bir Java uygulaması tarafından kullanılabilmesidir. Eklentileri ile birlikte Java Enterprise platform üzerinde web uygulamaları geliştirmek için de kullanılabilir. Spring Framework'ün temel özellikler herhangi bir Java uygulaması tarafından kullanılabilmesidir. Eklentileri ile birlikte Java Enterprise platform üzerinde web uygulamaları geliştirmek için de kullanılabilir.

- Spring, geliştiricilere POJO kullanarak kurumsal sınıf uygulamaları geliştirme imkanı sağlar. POJO ile herhangi bir sınıf bağımsız olarak tanımlanabilir.
- Spring modüler olarak geliştirilmiştir. Paket ve sınıf sayıları önemli olsa da, ihtiyacınız olan sınıfları ve paketleri kullanarak diğerlerini görmezden gelmenizi sağlamaktadır.
- Spring, tam anlamıyla bir ORM yapısı sağlayarak JEE, Kuvars ve JDK sayaçlarını bünyesinde bulundurarak diğer teknolojileri de kullanmak için alt yapı sağlamaktadır.
- Spring ile yazılmış bir uygulamanın test edilmesi kolaydır.
- Spring Web Framework, Struts ve diğer kütüphaneler gibi iyi tasarlanmış bir Web MVC alt yapısı sağlamaktadır.
- Spring tutarlı veya kontrolsüz(teknolojiye özel istisna durumlar) teknolojilerin kullanımı için uygun bir API sağlamaktadır.(Hibernate, JDBC, JDO)
- Spring küçük bir uygulamadan küresel işlem yapan bir uygulamaya kadar gerekli alt yapıyı ve desteği sağlar.

Şekil 3.2 Spring'in tümü altıyapı amacı programcının hayatını kolaylaştırmak ve ona birçok kullanım özgürlüğü tanımadır. Bu sebepten dolayı Spring frameworkünü oluşturan parçalar birbirinden bağımsız şekilde kullanılabilir.



Şekil 3.2: Spring Framework'un yapısı.

## **Spring Modülleri**

Spring, birbirinden bağımsız kullanılabilen modüllerden oluşmaktadır. Spring modül yapısı Şekil 3.2’de yer almaktadır.

### **Spring Core**

Spring core paketi, Spring’in temelini oluşturur. Dependency injection gibi temel fonksiyonlar Spring Core içine entegre edilmiştir.

### **Spring AOP**

Aspect Oriented Programming (AOP) bu modüle entegre edilmiştir. Spring’in AOP özellikleri direk kullanılmasa da deklaratif transaksyon ve güvenlik uygulamalarında Spring AOP Spring Core tarafından dolaylı olarak kullanılmaktadır. Spring’in önemli ikinci yapısı olan AOP uygulama içerisindeki servislerin çalışma zamanlarının kontrolü ve doğru zamanda yapacakları işi gerçekleştirmelerini sağlamaktadır. AOP yapısı izlediği bu yol ile Nesne Tabanlı Programlama (OOP) yapısını tamamlar.

### **Spring ORM**

Object Relational Mapping (ORM: Nesne İlişki Haritalama) modülü Hibernate, JDO, TopLink ve Ibatis gibi popüler ORM frameworkler ile entegrasyonu sağlamaktır.

### **Spring AOP**

Bilgi bankaları üzerinde işlem yapmak için kullanılan JDBC teknolojisini kullanımda daha basit hale getiren sınıflar bu modül içinde yer alır.

### **Spring Web MVC**

Spring MVC (Model-View-Controller) Spring baz alınarak web tabanlı programlar oluşturulabilmektedir. Spring MVC bir web frameworküdür.

### **Spring Context**

ApplicationContext ve WebApplicationContext gibi sınıflar bu modüle entegre edilmiştir. XML dosyalarından bulunan Spring Bean tanımlamalarını okumak ve Spring Bean nesnelere oluşturmak için kullanılmaktadırlar.



## Spring Dependency Injection (DI)

Dependency Injection bir yazılım geliştirme yöntemidir ve Spring'in iki temel önemli olan modülleri DI ve AOP yapısından biridir. Bu yapı yazılımın daha anlaşılır, geliştirici açısından kolay ve başarılı şekilde gerçekleştirilmesini sağlamaktadır. Spring kütüphanesi POJO sınıflarını kullanarak sınıfların bağımsız olarak çalışmasına ve kolayca değişiklik yapılmasına imkan tanımaktadır.

### 3.2.4.1 Spring Application Context

Spring Framework altında bulunan modüllerden Spring Context başlığı, ApplicationContext ve WebApplicationContext gibi sınıflar bu modüle entegre edilmiştir. Spring'in ana birimi, bağımsız çekirdekleri ve tanımları bir araya getirerek uygulamayı çalışır hale getiren XML çekirdek fabrikasıdır [34]. Tablo 3.1 ise Spring'in XML ApplicationContext dosyasıdır.

**Tablo 3.1:** Spring Application Context.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="URL"
  xmlns:context="URL" xsi:schemaLocation="URL">
  <mvc:resources mapping="/resources/**" location="/resources/mytheme/" />
  <context:component-scan base-package="tr.com.iu.informatics.thesis.controller" />
    </mvc:interceptors>
  <bean id="localeResolver" class="package ve sınıf adı" p:defaultLocale="en" />
    <mvc:annotation-driven />
    <mvc:default-servlet-handler/>
    <mvc:view-controller path="/secure/home.html" />
    <mvc:view-controller path="/users/registrationConfirmed" view-
name="/secure/home.html" />
  <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"
    p:viewClass="org.springframework.web.servlet.view.JstlView"
    p:prefix="/WEB-INF/jsp/"
    p:suffix=".jsp" />
</beans>
```

Spring Application ile bütün gereken spring sınıfları bu <beans > tagı kullanarak entegre edilebilmektedir.

Format: <beans xmlns="sınıfların linki" xsi:schemaLocation="sınıfların yeri linki">, <mvc:annotation-driven /> otomatik olarak spring anotatin uygulanır, <mvc:default-servlet-handler/> gereken serverleri uygulama için kullanılır ve <mvc:view-controller /> ile arayüzün kontrolleri uygulanır.

### 3.2.4.2 Spring Security

Bu tez çalışması kapsamında sistem güvenliği için spring framework güvenlik paketi kullanılmıştır. Spring güvenliği sayesinde eklenen her sayfaya hangi kullanıcıların erişebileceği belirlenebilmektedir. Spring Güvenlik Paketi J2EE tabanlı ticari yazılım uygulamaları için kapsamlı ve esnek bir güvenlik servisi sunmaktadır. Spring çatısı altında geliştirilen projelerin desteklenmesi bakımından bu kısım oldukça önemlidir. Sistem kapsamında bütün sayfaların güvenliğinin nasıl sağlandığı Tablo 3.2'deki örnekte sunulmaktadır.

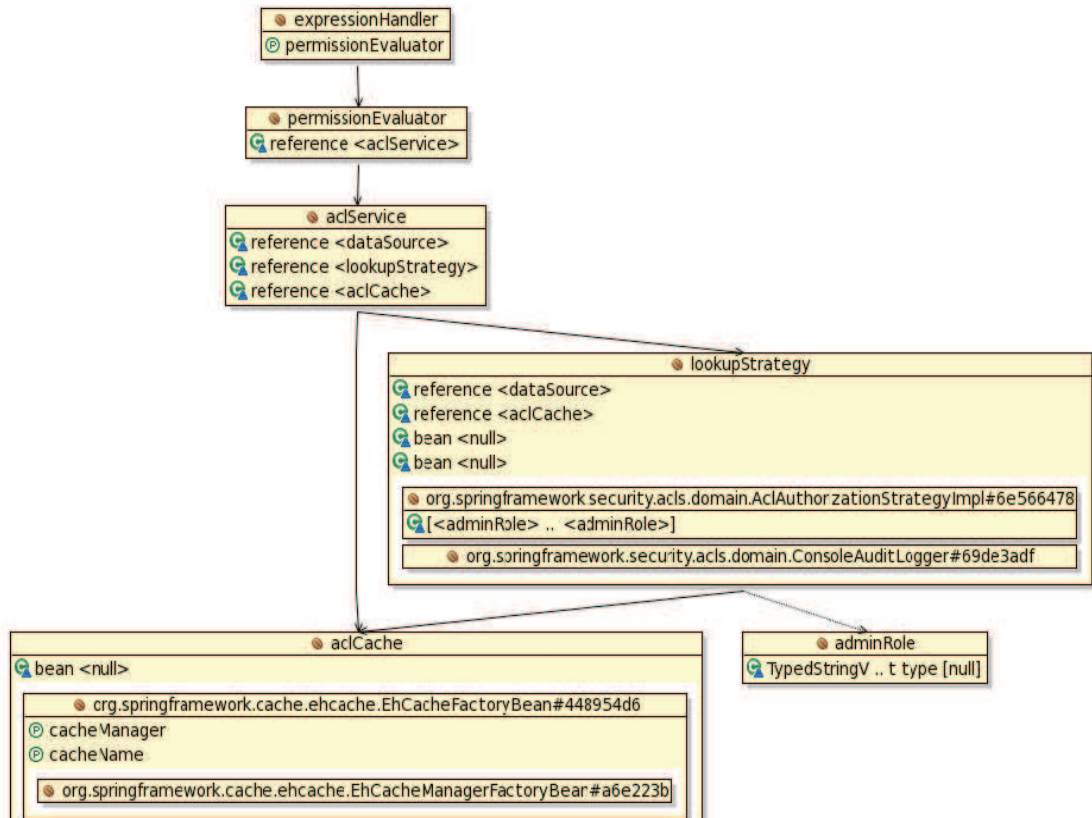
**Tablo 3.2:** Spring Güvenlik XML Dosyası.

```
<http auto-config="true" use-expressions="true">
  <form-login login-page="/sayfa adi" default-target-url="/sayfa adi" />
    <logout logout-success-url="/sayfa adi" />
    <intercept-url pattern="/ sayfa adi" access="permitAll"/>
    <intercept-url pattern="/**" access="isAuthenticated"/>
    <intercept-url pattern="/ klasör /**" filters="none" />
    <intercept-url pattern="/login.html" method="GET"
access="isAnonymous()"/>
  <intercept-url pattern="/users" method="POST" access="isAnonymous()"/>
  <intercept-url pattern="/admin.html" method="GET"
access="hasRole('ROLE_ADMIN')"/>
  <intercept-url pattern="/**" access="denyAll" /> </http>
  <beans:bean id="passwordEncoder" class="package.ShaPasswordEncoder">
    <beans:constructor-arg value="256" /> </beans:bean>
  <beans:bean id="saltSource" class="package.ReflectionSaltSource"
    p:userPropertyToUse="id" />
</beans:beans>
```

Tablo 3.2’de ilk tag olan `<http />`, bütün güvenlik dosyalarındaki sayfaları http kullanarak açılması gerektiği anlamındadır.

### 3.2.4.3 Spring Access Control List (ACL)

Sistemde Spring güvenliği kullanmak için, Spring destek erişim kontrolü listesi kullanılmıştır (ACL). Spring ACL ile bütün sistemin kullanıcıların izinleri ve sayfaların izinleri ACL olarak veritabanında tutulmaktadır. Bu sayede, kullanıcıların erişimleri kontrol altında tutulabilmektedir. ACL için veritabanında Spring ACL tabloları oluşturmak gerekmektedir. Şekil 3.3 ACL beanslerin’in oluşturulmuş beans diagramdır. Spring Güvenlik ACL bean grafik; bütün beans-security-acl.xml deki beanslerin grafikleridir Beansler arasındaki ilişkiler sunmaktadır. Bu ilişkiler; birden çoğa, çoktan bire şeklinde olabilmektedir. Şekil 3.3, spring güvenliği oluşturmak için gereken beansleri göstermektedir.



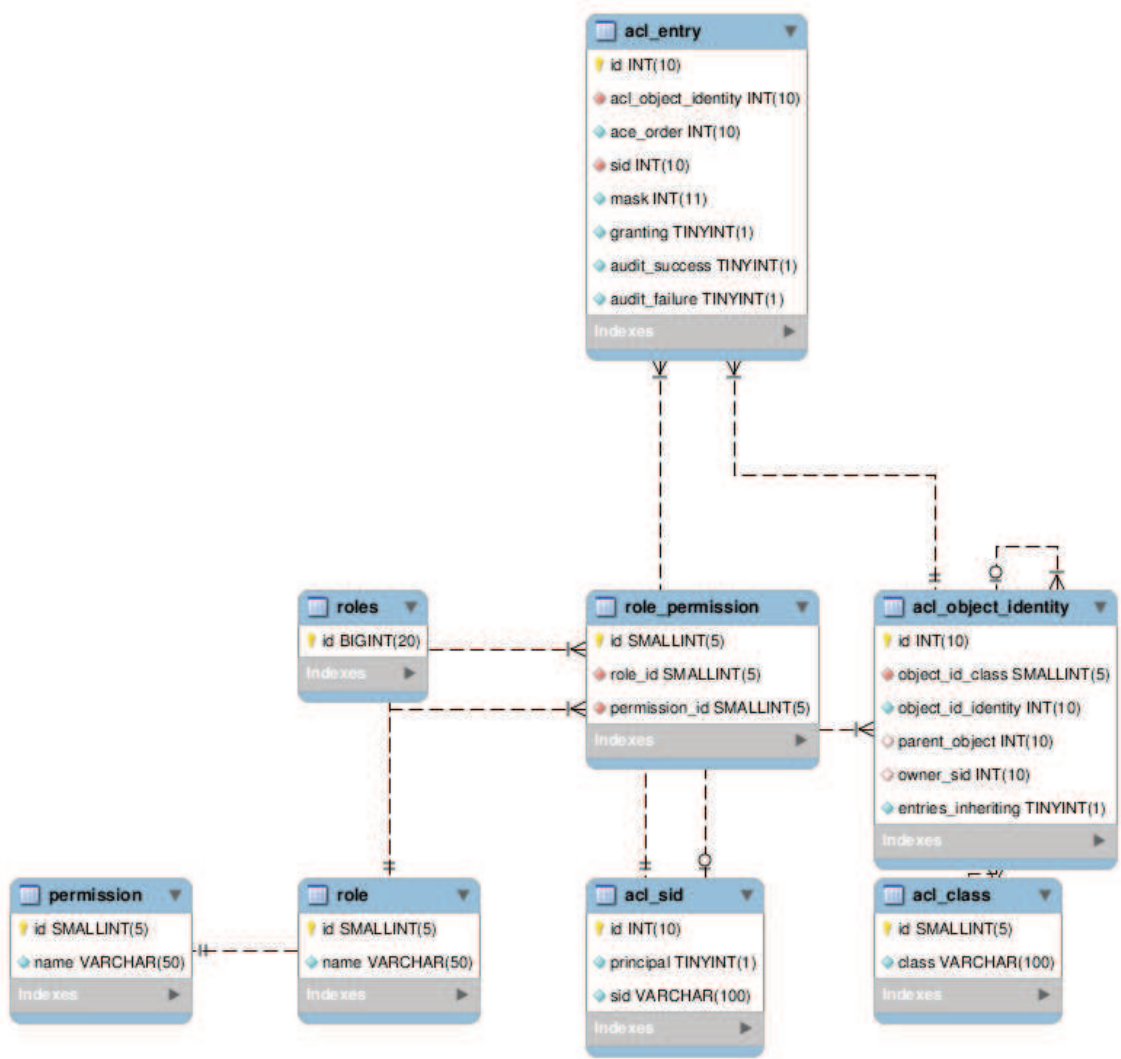
Şekil 3.3: Spring ACL Beans Grafiği.

### 3.2.4.4 Spring ACL ER Diyagramı

Şekil 3.4’de verilen yapının oluşturulması, spring güvenlik için oldukça önemlidir. her içeren diyagramda adı veritabanında tablosu olması gerekmektedir.

**Tablo 3.3:** Spring ACL Tabloların Açıklamaları.

<b>Tablo Adı</b>	<b>Tablo Açıklaması</b>
acl_entry	Erişim kontrolü listesi girişi tablosu
Roles	Kullanıcılar ve nesnelerin rolleri tablosu
roles_permission	Rolün izini ilişki tablosu
acl_object_identity	Erişim kontrolü listesinin nesne id tablosu
Permission	İzin tablosu
acl_sid	İzin id tablosu
acl_class	Erişim kontrolü listesinin sınıfları tablosu



Şekil 3.4: Spring ACL Kod Bileşenleri ER Diyagramı.

Spring ACL veritabanı ER diyagramı yukarıda ACL'in beans grefiği şeklinde sunulmuştur. Beanslerin isimlerine göre veritabanında aynı tablolar olmasına rağmen, Tablo 3.4'de gösterildiği gibi, ya kullanıcı oluşturmalı ya da hibernate otomatik olarak meydana getirmelidir. Şekil 3.4'dün tabloları veritabanında ACL'in ER diyagramıdır.

### 3.2.4.5 Spring Web XML

Spring mvc çalıştırmak için Deployment Descriptor (DD) – web.xml içine tüm erişimleri karşılayacak servleti belirtmek gerekmektedir. Gerekli dispatcherServlet, listeners ve filter-mapping aşağıdaki Tablo 3.4'de içerik web.xml'in karşılığıdır. Bu dosyada temel olarak Spring çalıştırmak gerekli bütün beansleri tutmaktadır. Tablo 3.4'te web.xml için gerekli beansler verilmektedir.

**Tablo 3.4:** Spring Web XML.

```

<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>classpath:/spring/beans-dataSource.xml</param-value>
</context-param>
<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener
  </listener-class>
</listener>
<filter>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<servlet>
  <servlet-name>spring</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
  <init-param><param-name>contextConfigLocation</param-name>
  <param-value>classpath:/spring/applicationContext.xml</param-value>
  </init-param>
</servlet>
  <servlet-mapping>
    <servlet-name>spring</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>

```

Web xml özelliği kullanılan teknolojileri otomatik olarak sistem bulabilmektedir.

**<context-param>**: Bütün sistemde xml dosyaları contex'te parametre olarak bulunmasıdır. Parametre isimi vermek için **<param-name>** tagı kullanılmaktadır. Parametre değeri için ise **<param-value>** tagı kullanılabilir.

**<listener>**: Spring Framework'un bazı sınıfları mesela servlet dosyalarının nerede oldukları listener tagı **<listener-class>** içinde belirtilerek sistem otomatik bulunması sağlanır.

**<servlet>**: Bu tag, daha önce Tablo 3.1 application context detay olarak bahsedilmişti. Burada o dosyayı sistemin bulabilmesi için servlet ismi, sınıfı ve dosyanın nerede olduğunu sisteme detaylı bir şekilde vermek gerekmektedir.

### 3.2.5 Hibernate (ORM)

ORM, veritabanında oluşturulan her bir nesneye (tabloya) karşılık uygulama tarafında bir nesne oluşturma işidir. Bu işlem bazı Frameworklerde ara yazılımlar sayesinde (ORM Tools), bazı frameworklerde ise elle gerçekleştirilmektedir. İlişkisel veritabanları ile Object-Oriented (Nesneye dayalı) uygulamaların arasındaki veri uyumunu sağlayan bir tekniktir. Birbirine uyuşmayan veri tipleri arasındaki veri dönüşümünü sağlar. Bu sayede biz uygulamalarımızda nesnelere kullanırken bunların veritabanında nasıl tutulduğunu dert etmeyiz. Eğer veritabanımızı iyi şekilde tasarlamışsak, nesnelimizi de ona uygun şekilde oluşturmuşsak işimiz çok kolaylaşır.

Veritabanı seviyesine inmeden uygulama verileriyle işlem yapabiliriz. ORM ile SQL cümlelerini yazmak zorunda kalmazsınız. Oluşturulan bir class ile tabloda saklanacak değerler; oluşturulan classtaki değişkenlerle ilişkilendirilmiş, classınızın bir değişkeni olmuştur artık. Bununla da kalmayıp, ekleme, silme, güncelleme ve veri çekme işlemlerini icra eden metodları da oluşturmuş ve kullanımınıza sunmuştur. Hibernate, Java platformunda yazılmış bir ORM (Object/Relational Mapping) aracıdır [35]. Spring, Hibernate ve MySQL bir birine ihtiyaç olmasına rağmen,

*“İki temel yöntem sunmaktadır. Bunlardan birisi veri erişim katmanını DAO tasarım örüntüsü ile iş mantığı katmanından soyutlamak ve bunun için sağlanan “DaoSupport” sınıflarıdır. Diğeri ise kullanılan veri erişim teknolojisinin “plumbing” olarak tabir edilen kısımlarının uygulama geliştiricilerin omuzlarından alınarak, sadece veri manipülasyon ve sorgulama işlemleri ile dönen sonuçların üzerinde işlem yapan kısımların kodlanmasını sağlayan “Template” yapılarıdır. Hibernate entegrasyonu çerçevesinde de bu sınıfların karşılıkları HibernateDaoSupport ve HibernateTemplate olarak tanımlanmıştır. Hibernate tarafındaki gelişmeler sonucunda artık bu yapılar “legacy” olarak tabir edilmektedir.”*

Tablo 3.5 spring, hibernate ve herhangi bir veritabanına entegre edebilmek için “datasource” xml dosya oluşturarak ve veri tabanının parametreler vererek entegre edebilir.

**Tablo 3.5:** Spring, Hibernate ve Mysql Entegrasyon Dosyası.

```
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">
<property name="driverClassName" value="com.mysql.jdbc.Driver" />
<property name="url"
value="jdbc:mysql://localhost:3306/ac1DB?autoReconnect=true" />
    <property name="username" value="kullanıcı adı" />
    <property name="password" value="şifre" />
</bean>
<bean class="org.springframework.jdbc.core.JdbcTemplate" p:dataSource-
ref="dataSource" />
    <util:properties id="hibernateProperties">
        <prop
key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>
        <prop key="hibernate.show_sql">>false</prop>
    </util:properties>
```

Yukarıdaki tabloda dataSource id olan beansi spring’den hibernate doğru MySQL’a bağlanmak için BasicDataSource sınıfı kullanılıyor, bu sınıfta olması gereken parametreler aşağıdaki gibidir:

#### **driverClassName**

MySQL bağlantı sınıfı yani connectJ bu paket’ten com.mysql.jdbc.Driver spring bulucaktır.



## URL

Hangi tarafa doğru bağlanacağını gösterir. Bu tez çalışmasında localhost kullanmıştır. Bilgisayarda, tutulan tablolar, veritabanının adı, bağlantıda sıkıntı yaşanmaması için MySQL'in her zaman otomatik olarak bağlanmasına izin verilmiştir.

**name**="kullanıcı-adi" kullanıcının MySQL kullanıcı adı

**name**="şifre": kullanıcı şifresi

## JdbcTemplate

Hibernate için MySQL'a bağlantı bütün dataSource'teki parametreler alıp bağlantı kurur.

## Hibernate.dialect

Birçok veritabanı olduğu için birbirleri arasında bazı farklılıkları vardır, bu farklılıklar hibernate iyice çalıştırmak için kullanılacak veritabanının önceden bildirilmesi gerekmektedir.

## hibernate.show\_sql

Seçilen IDE göre, kodu çalıştırırken hibernate, komutları ekrana göndermekte, bu parametre doğru (true) ise ekrana komutların kabul edildiğini, yanlış (false) ise kabul edilmediğini göstermektedir. Burada sistemin güvenliğini tehlikeye atmamak için yanlış (false) seçilmiştir.

### 3.2.6 Java Persisten API (JPA)

Java Persistence API (JPA), Object Relational Mapping (ORM) teknoloji apilerinden bir tanesi olup database tarafında tüm tabloların program tarafındaki bir (entity) sınıfa denk gelmesi ve tüm işlemlerinin bu (entity) class üzerinde yapılmasını sağlayan bir teknolojidir [36]. Amaç tabi ki sadece bu değil. JPA teknolojisi ile veritabanı ile bilgi alışverişi esnasında veri güvenliği ve performans da üst düzeye çıkarılmış olur. Entity JPA'yı kullanarak veri tabanında kalıcı kılınacak olan nesneye entity denir. Bir nesnenin entity olması için gerek ve yeter şart o nesnenin kendisinden üretileceği sınıfın ya @Entity notuyla notlandırılması ya da XML ayarlarında entity olarak ifade edilmesidir. (Biz bu yazıda genel olarak notlu ayarı temel alacağımızdan okuyucu, aksi söylenmedikçe bütün ayarların notlarla ifade edildiğini varsaymalıdır.).

JPA ile bir Hibernate veya Eclipse-Link gibi bir kütüphane değil, bunların kullanılmasını sağlayan bir spesifikasyondur. Yani biz JPA sayesinde Hibernate kütüphanesini veya diğer kütüphaneleri bilmek zorunda değiliz. JPA bu kütüphaneleri kullanmamızda bize ara bir katman oluyor. Yalnızca belirtilen araç kullanarak belirtilen veritabanına belirtilen veriyi kaydetme komutu vererek işlem gerçekleştirilebilmektedir. Bu şekilde de JPA işlerimizi oldukça kolaylaştırmaktadır.

### 3.2.7 MySQL

MySQL güçlü ve çok hızlı bir veritabanı sunucusudur. Birçok programlama dilini destekler ve birden fazla kullanıcı ile birlikte kullanılabilir. UNIX, OS/2 ve Windows platformları için ücretsiz dağıtılmakla birlikte ticari lisans kullanmak isteyenler için de ücretli bir lisans seçeneği de mevcuttur. Linux altında daha hızlı bir performans sergilemektedir. Bu çalışma kapsamında MySQL bağlantısı aşağıdaki biçimde gerçekleştirilmiştir.

```
<dependency>
  <groupId>com.mysql.jdbc</groupId>
  <artifactId>com.springsource.com.mysql.jdbc</artifactId>
</dependency>
```

Dependency (bağımlılık) kullanılarak MySQL sürücüsü otomatik olarak maven depodan yüklenmektedir.

### 3.2.8 Java Server Pages (JSP)

JSP bir java sunucu üzerinde çalışan bütün istemcilerinin scriptlerinin sonucunu ekrana bastırmak için üretilen, java tasarım teknolojisi olarak tanımlanmıştır. Java Server Pages (JSP) java web uygulamalarında kullanılan bir yapıdır. Java web programlama yaparken servlet ve Jsp kullanarak kod tasarlarlarken benzer bir kullanım söz konusudur ancak JSP kod tasarımlarında altyapı çok daha farklıdır. JavaServer Pages güçlü ve dinamik web uygulamaları geliştirmeyi kolaylaştıran web tabanlı ara yüzler hazırlamak için tasarlanmıştır. Javanın güvenli programlamaya verdiği önemi göz önünde bulundurursak bunun çok faydalı bir şey olduğunu anlamış oluruz. JSP ve Servlet ile programlamada objelerin yaşam süreçleri vardır.



Servlet web programlamada objeler vardır. Birincisi session düzeyinde diğeri ise

database düzeyindedir. Bu objelere database object'te denilebilmektedir. Bildiğimiz gibi Java programlama temeli objelerin kontrolü altında tutulup performanslı kullanılmasına dayanmaktadır. Bunun içinde bu objelerin kontrollerinin sağlanması için teknolojiler bulunmaktadır. (örn: JPA, EJB, Spring vs.).

JSP, sunucularda belli bir application server içerisinde çalışan programlardır. Tıpkı Client'ta appletlerin çalışması gibi. Servlet'lerde HTTP üzerinden erişim sağlanır. Servlet'ler çıktı olarak da yine HTML üretirler. JSP ise; bir java kodunun HTML kodu içerisinde kullanılması sağlayan yapıdır. JSP kodları PHP, ASP ve Python'da olduğu gibi HTML içerisinde belli yerlerde kullanılabilir. JSP sayfaları aslında içerisinde java kodları ile yazılmış olan JSP kodlarının bulunduğu JSP uzantılı HTML sayfalarıdır. Bu durumun getirdiği bir çok avantaj söz konusudur ki örneği; bu model sayesinde görsellik oldukça gelişmiş ve bu karmaşıktır, statik bir HTML sayfasına bir kaç satır JSP kodu ekleyerek sayfaları daha dinamik içerik hale getirebilir.

### 3.2.9 Apache Tomcat Server

Enterprise java uygulamaları geliştirirken uygulamalarımızı local'de ve ya internette çalıştırabilmek için bir application server'a ihtiyaç vardır. Tomcat java uygulamalarımızı üzerinde çalıştırabileceğimiz bir application server hizmeti sağlamaktadır. Apache'nin bir alt projesi olan ve Jakarta olarak bilinen Tomcat sunucusu kullanılır. Tomcat Java tabanlı, web uygulama container'ıdır. Tomcat sunucuda uygulamaları yönetilmesi, yönetici ara yüzünden manager-gui adı ile yönetilir. Şekil 3.5 Tomcat sunucunun yönetici arayüzüdür.

#### Tomcat Web Application Manager

Message: OK

**Manager**

[List Applications](#)      [HTML Manager Help](#)      [Manager Help](#)      [Server Status](#)

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat:	true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle > 30 minutes

Şekil 3.5: Apache Tomcat sunucu.

**Deploy**

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

WAR file to deploy

Select WAR file to upload  No file chosen

**Diagnostics**

Check to see if a web application has caused a memory leak on stop, reload or undeploy

This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

**Server Information**

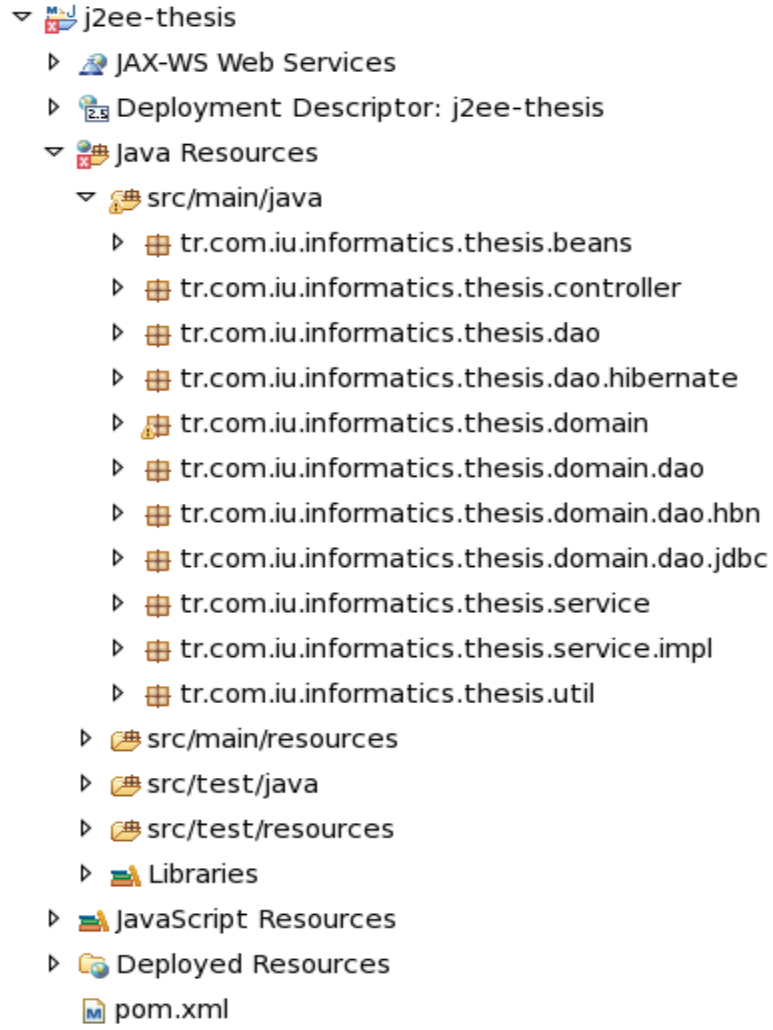
Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/7.0.42	1.7.0_40-b43	Oracle Corporation	Windows 7	6.1	amd64	husnamuhsinPC	192.168.1.4

**Şekil 3.6:** Tomcat uygulama yükleme ara yüzü.

Tomcat sunucuna java uygulama yüklemek için geliştirilen sistemde eclipse tarafından `<Context docBase="uygulama dosya ve adı" path="/tauraro" reloadable="true"/>` şeklinde otomatik olarak yüklenebilmekte, istendiği takdirde Şekil 3.6'de gösterilmiş arayüzünden war dosyası yüklenerek aynı işlem gerçekleştirilebilmektedir.

### 3.2.10 Maven

Apache Maven, Repository kavramında; depo gibi anlamlara gelmektedir ve Maven da yaptığı görev itibarıyla de bu kelimeyi tam olarak karşılamaktadır. Maven da 3 tane Repository bulunmaktadır. Maven konfigürasyon dosyaları ile kullanmak istenen kütüphaneler belirtilir[37], Maven bu kütüphaneleri indirir ve projelere yükler.



**Şekil 3.7:** Maven proje dosyaları.

Maven'in proje geliştirmeyi kolaylaştırmak için bir çok özelliği vardır. Geliştirilen java projelerinin geliştirme süreçlerini basitleştirmek, standartlaştırmak, dökümantasyon hazırlamak, kütüphane bağımlılıklarından kullanıcıyı kurtarmak için kullanılan bir araçtır. Şekil 3.7 Maven Proje iskeleti standart olduğu için java projenizi ister eclipse ile ister netbeans ile isterseniz başka ide'lerle geliştirebilirsiniz. Maven ile hemen hemen bütün java projelerinin iskeleti oluşturulabilir. Bu iskelet standartlaştığı için, oluşturulmuş projeyi geliştirme ortamından bağımsız olarak geliştirir.

## **4 ERASMUS DENEYİM PAYLAŞIM SİSTEMİ (EDEPAS)**

Tezin bu bölümünde EDEPAS sistemine erişim ve sistemin kullanımı ile ilgili bilgilere yer verilmiştir.

### **4.1 ERASMUS DENEYİM PAYLAŞIM SİSTEMİ (EDEPAS)**

Bu amaçlar sistemin kullanıcı ara yüzleri ile yönetim ara yüzlerine ilişkin açıklamalar verilecektir.

#### **4.1.1 Sistem Tasarımı**

Sistemin geliştirilmesine başlanmadan önce çeşitli firmalarla görüşülerek, framework seçenekleri arasından hangisinin daha düzgün ve özenli olduğu belirlenmiştir. Örneğin, kurumsal uygulama geliştirme sürecinde hangi teknolojilerin sistemin yoğunluğunu karşılayamayacağı, hangilerinde bellek kapasitesinin fazlaca kullanıldığı, yazılım geliştirme sürecinde kullanılacak kodun niteliği gibi bilgiler araştırılmıştır. Tüm bu hususlar detaylı olarak incelenmiş ve uygulamada hangilerine yer verileceği belirlenmiştir. Daha sonra J2EE açık kaynak kodu ile veritabanı tasarımı gerçekleştirilmiştir.

#### **4.1.2 Kullanıcı Ara Yüzü Sayfaları**

Sistem uygulaması olarak geliştirilecek olan kullanıcı arayüzü, kullanıcının Erasmus için yazdığı deneyim ile ilgili bilgilere ulaşmasını sağlayacaktır. Bu sayede kullanıcı bütün sistemdeki aktif deneyimleri okuyabilecek, kendi deneyimlerini yazabilecektir. Benzer şekilde yönetici rolüne sahip olan kişilerde girilen deneyimleri inceleyebilecek ve uygun olanları aktif hale getirecektir. Sistemden aynı zamanda Erasmus duyurularının takip edilebilmesi mümkündür. Geliştirilen sistemin kullanıcı ara yüzü adım adım alt başlık halinde aşağıda verilmiştir.

##### **4.1.2.1 Giriş Yapma Ekranı**

EDEPAS'a giriş yapmak için kullanıcı adı ve şifre istenmektedir. Giriş yapan öğrencinin bilgilerine ulaşma ve öğrenci olup olmadığının kontrolü ileride

üniversitemizin ilgili veritabanlarıyla yapılacak entegrasyonla sağlanacaktır. Ancak en basit haliyle kullanıcı bilgileri sistemde veritabanlarındaki bilgiler ile karşılaştırılarak başarılı giriş sağlanabilmektedir. Şekil 4.1’de EDEPAS giriş ekranı verilmektedir.

Şekil 4.1: EDEPAS Giriş Yapma Ekranı.

#### 4.1.2.2 Kayıt Olma Ekranı

Sisteme giriş yapmak için sistemde kullanıcı kayıdı olması gerekmektedir. Giriş yapma ekranından kayıt ol bağlantısına tıklayınca Şekil 4.2’de yer alan kayıt olma formu ile karşılaşmaktadır.

Şekil 4.2: Kayıt Olma Ekranı.

### 4.1.2.3 Ana Sayfa

Giriş ekranından girilen bilgilerin doğruluğu sistem tarafından kontrol edilmektedir. Eğer bilgiler doğru ise kullanıcı ilgili sayfaya yönlendirilmektedir. Şekil 4.3'te bilgilerini doğru giren kullanıcının kendi sayfası verilmektedir.



Şekil 4.3: Sistem Ana Sayfa Ekranı.

### 4.1.2.4 Ülke Listesi

Bu çalışma kapsamında, öğrencilerin deneyimleri ülkelere göre yönetilmektedir. Böylece sistemdeki bütün deneyimler diğer kullanıcılar tarafından kolayca bulunabilir.

The screenshot shows the 'Ülkeler' (Countries) list page. It features a table with the following data:

Ülke	# Deneyimler	Son Yazılmış mesaj	Moderatörü
<a href="#">Iran</a>	3	01.Eki.2012 14:05	<a href="#">Julia Cazares</a>
<a href="#">Iraq</a>	5	01.Eki.2012 21:58	<a href="#">Julia Cazares</a>
<a href="#">Kanada</a>	3	01.Eki.2012 17:53	<a href="#">Julia Cazares</a>
<a href="#">USA</a>	1	09.Haz.2014 12:32	<a href="#">Elvira Cazares</a>
<a href="#">UK</a>	5	01.Eki.2012 11:02	<a href="#">Elvira Cazares</a>
<a href="#">Pakistan</a>	4	30.Eyl.2012 19:43	<a href="#">Elvira Cazares</a>
<a href="#">Niqeria</a>	0		<a href="#">Bello Muhammad</a>
<a href="#">Turkey</a>	0		<a href="#">Bello Muhammad</a>
<a href="#">Ghana</a>	0		<a href="#">Bello Muhammad</a>

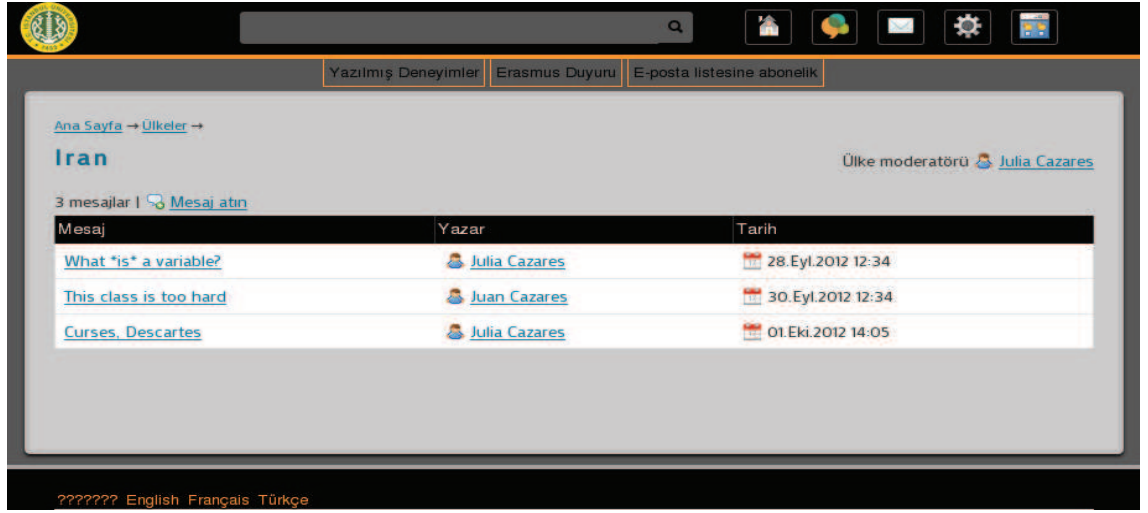
At the bottom, there is a language selection bar with 'English', 'Français', and 'Türkçe' options.

Şekil 4.4: Ülke Liste Ekranı.



#### 4.1.2.5 Deneyim Listesi

Yukarıda görülen ülkelere göre öğrenci deneyimleri tutulmaktadır. Şekil 4.4’de her ülkedeki öğrencilerin deneyimlerinin linkleri verilmektedir. Sistemdeki bütün aktif deneyim linkleri öğrenciler tarafından erişebilir durumdadır. Bu linklere tıklayarak, deneyime ilişkin metine erişilebilmesi mümkündür. Şekil 4.5’te deneyim bağlantılarından biri tıklandığında görülecek metin verilmektedir.



Şekil 4.5: Deneyim Liste Ekranı.

#### 4.1.2.6 Deneyim Okuma Ekranı

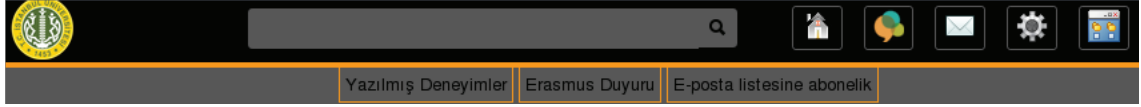
Şekil 4.6’de bir deneyim metini görülmektedir.



Şekil 4.6: Deneyim Okuma Ekranı.

#### 4.1.2.7 Sistem Menüleri

Sistem kontrolünün bir yönetici rolüyle sağlanması oldukça önemlidir. Bu nedenle standart kullanıcı ve yönetici düzeyindeki kullanıcının farklı ekran, yetki ve menü görüntüleme seçeneklerine sahip olması gerekmektedir. Şekil 4.7’de sistem menüleri verilmektedir.



Şekil 4.7: Sistem menüleri.

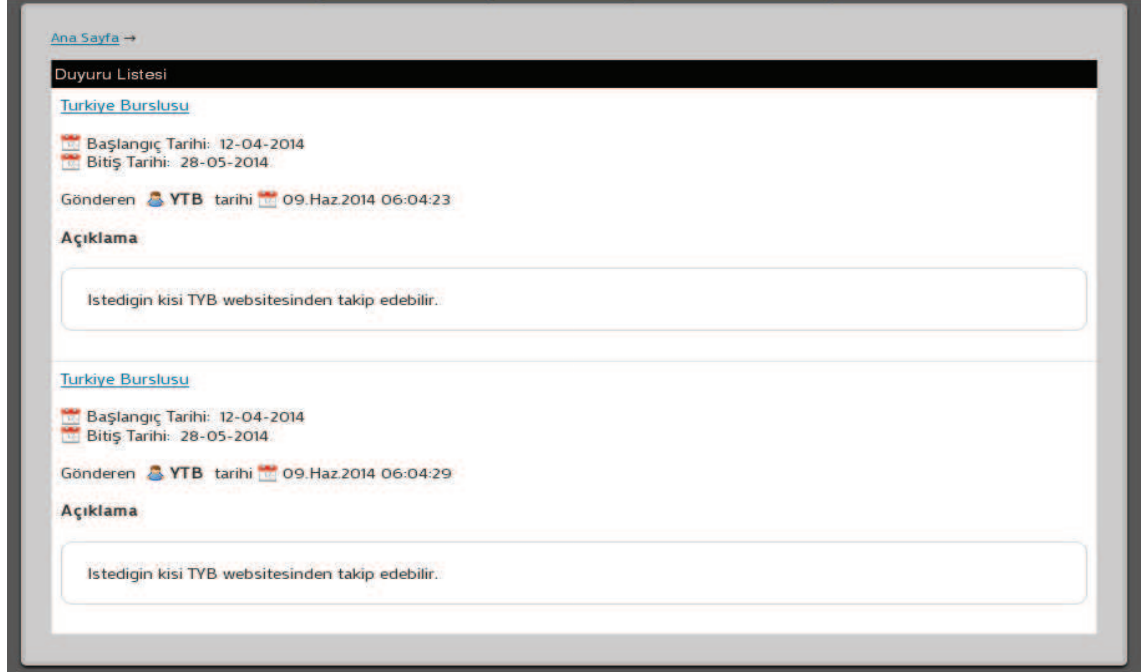
#### 4.1.2.8 Profil Güncelleme Ekranı

Profil güncellemeleri, yöneticilerin ve kullanıcıların sistem kullanırken kişiselleştirme açısından önemli bir ihtiyacı oluşturmaktadır. Böylece öğrenci-yönetici-sorumlu arasında daha sağlıklı iletişim kurulması mümkün olabilmektedir. Şekil 4.8’de sistemde bütün kullanıcıların bilgilerini güncelleyebileceği ekran verilmektedir.

Şekil 4.8: Profil Güncelleme Ekranı.

#### 4.1.2.9 Duyuru Ekranı

Erasmus giden veya gitmek isteyen öğrencilere ilişkin duyuruları web üzerinden veya okullara/bölgelere/fakülterlere resmi yazılar yoluyla duyuru yaparak bilgi ulaştırmaktadır. Şekil 4.9'da Erasmus duyurularının yapılabileceği arayüz verilmektedir.



Şekil 4.9: Duyuru Ekranı.

#### 4.1.3 Yönetici Ara Yüzü Sayfaları

Sistem uygulamasında işlerliğin yanısıra kontrol ve güvenlik iki önemli unsurdur. Özellikle kullanıcı hareketlerinin kontrol edilmesi bir yönetim arayüzünün oluşturulması ile mümkündür. Burada kullanıcı kaydından, veri girişlerine, kullanıcıların yaptıkları deneyim paylaşımlarının uygunluğuna kadar geniş bir kontrol yelpazesinden bahsedilmektedir.

##### 4.1.3.1 Yönetici Ara Yüzü

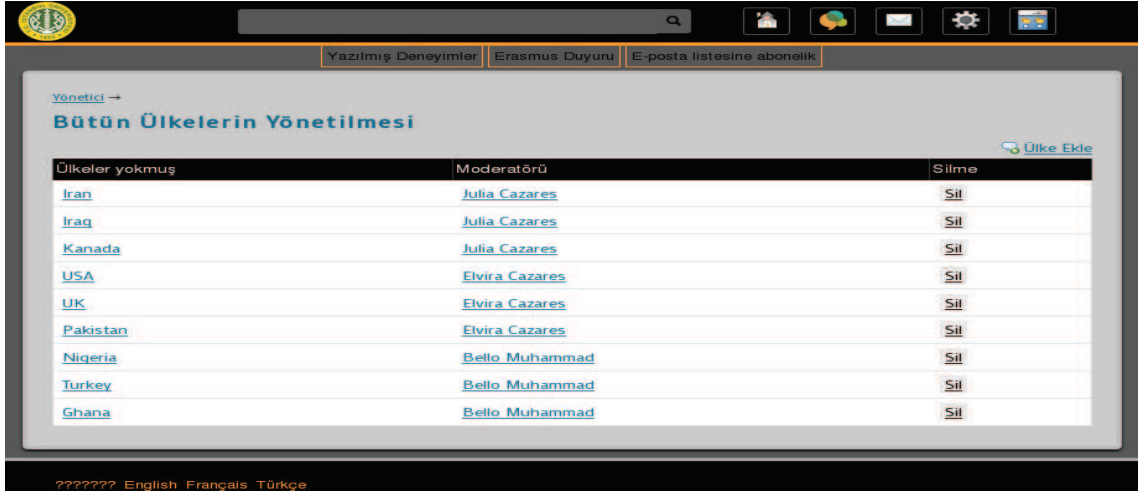
Sistem yöneticisi sistemin genel sorumlusu durumundadır. Sistem yöneticisi; aktif kullanıcı hesap kaldırma, silme ve engelleme, aktif deneyim kaldırma, silme, güncelleme ve engelleme, diğer ülke yöneticilerinin yönetimini kontrol etmek gibi pek çok sorumluluğa sahiptir. Şekil 4.10 sistem yöneticisi ara yüzünün menüleri gösterilmektedir



Şekil 4.10: Yönetici Ara Yüzü.

#### 4.1.3.2 Ülke Yönetme Ara Yüzü

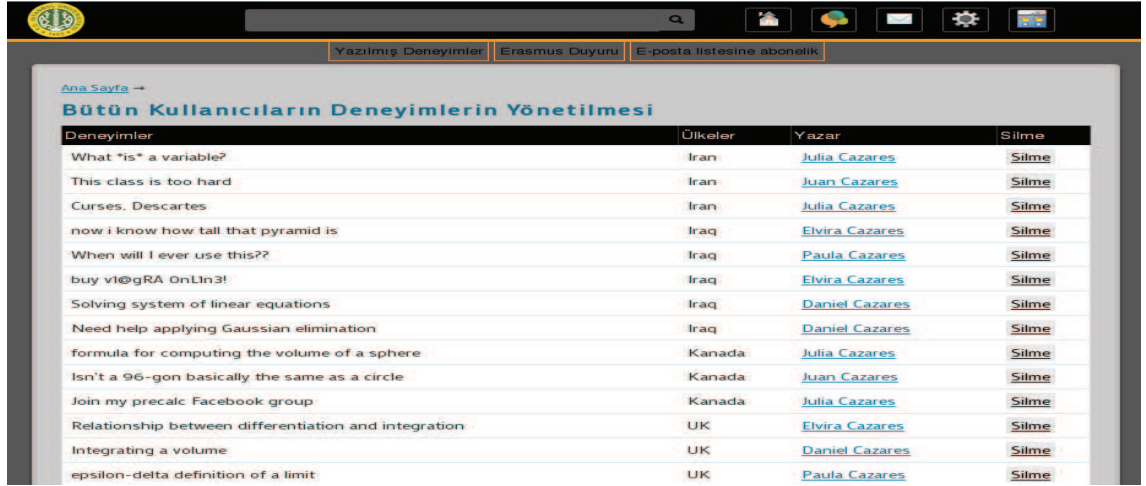
Yönetici işlerinden birisi ülke ekleme, silme ve güncellemesidir. Ülke moderatörlerinin ülkelerini silme ya da ekleme yetkisine sahip olması sistem tasarımı açısından pek de uygun bir karar olmadığından bu yetki sistem yöneticisine atanmıştır. Şekil 4.11’de ülke yönetim ekranı gösterilmektedir.



Şekil 4.11: Ülke Yönetme Ekranı.

#### 4.1.3.3 Deneyim Yönetme Ara Yüzü

Kullanıcıların paylaşımının yönetilmesi, içerik olarak doğru, kurumlara ve kişilere zarar vermeyecek ifadeler içermesi, hukuksal sorunlara neden olmaması açısından oldukça önemlidir. Bu nedenle ülke moderatörlerinin kullanıcıların yaptığı paylaşımları kontrol edebilmesi ve gerekli onaylama süreçlerinde yetkin olması gerekmektedir. Şekil 4.12’de deneyim yönetim ekranı verilmektedir.

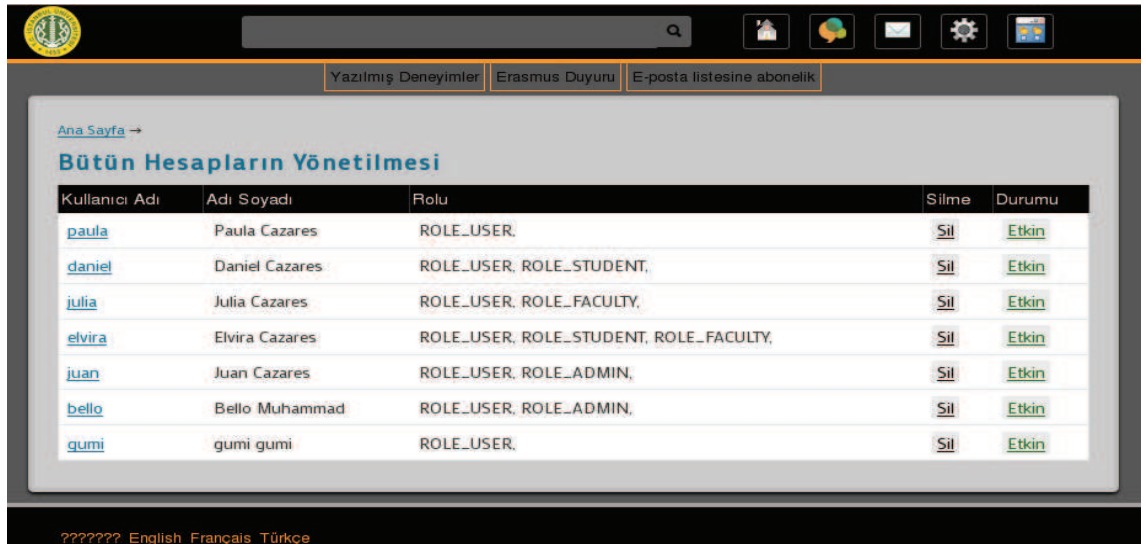


Deneyimler	Ülkeler	Yazar	Silme
What "is" a variable?	Iran	<a href="#">Julia Cazares</a>	<a href="#">Silme</a>
This class is too hard	Iran	<a href="#">Juan Cazares</a>	<a href="#">Silme</a>
Curses, Descartes	Iran	<a href="#">Julia Cazares</a>	<a href="#">Silme</a>
now i know how tall that pyramid is	Iraq	<a href="#">Elvira Cazares</a>	<a href="#">Silme</a>
When will I ever use this??	Iraq	<a href="#">Paula Cazares</a>	<a href="#">Silme</a>
buy vi@gRA OnLn3!	Iraq	<a href="#">Elvira Cazares</a>	<a href="#">Silme</a>
Solving system of linear equations	Iraq	<a href="#">Daniel Cazares</a>	<a href="#">Silme</a>
Need help applying Gaussian elimination	Iraq	<a href="#">Daniel Cazares</a>	<a href="#">Silme</a>
formula for computing the volume of a sphere	Kanada	<a href="#">Julia Cazares</a>	<a href="#">Silme</a>
Isn't a 96-gon basically the same as a circle	Kanada	<a href="#">Juan Cazares</a>	<a href="#">Silme</a>
Join my precalc Facebook group	Kanada	<a href="#">Julia Cazares</a>	<a href="#">Silme</a>
Relationship between differentiation and integration	UK	<a href="#">Elvira Cazares</a>	<a href="#">Silme</a>
Integrating a volume	UK	<a href="#">Daniel Cazares</a>	<a href="#">Silme</a>
epsilon-delta definition of a limit	UK	<a href="#">Paula Cazares</a>	<a href="#">Silme</a>

Şekil 4.12: Deneyim Yönetme Ara Yüzü.

#### 4.1.3.4 Kullanıcı Yönetme Ara Yüzü

Öğrencilerin mezun olması ya da herhangi bir sebeple okuldan ayrılması durumunda aktif olarak kullanılmayan hesapların silinmesi sistem yöneticisi tarafından yapılabilmektedir. Şekil 4.13'te kullanıcı yönetme ara yüzü verilmektedir.



Kullanıcı Adı	Adı Soyadı	Rolü	Silme	Durumu
<a href="#">paula</a>	Paula Cazares	ROLE_USER,	<a href="#">Sil</a>	<a href="#">Etkin</a>
<a href="#">daniel</a>	Daniel Cazares	ROLE_USER, ROLE_STUDENT,	<a href="#">Sil</a>	<a href="#">Etkin</a>
<a href="#">julia</a>	Julia Cazares	ROLE_USER, ROLE_FACULTY,	<a href="#">Sil</a>	<a href="#">Etkin</a>
<a href="#">elvira</a>	Elvira Cazares	ROLE_USER, ROLE_STUDENT, ROLE_FACULTY,	<a href="#">Sil</a>	<a href="#">Etkin</a>
<a href="#">juan</a>	Juan Cazares	ROLE_USER, ROLE_ADMIN,	<a href="#">Sil</a>	<a href="#">Etkin</a>
<a href="#">bello</a>	Bello Muhammad	ROLE_USER, ROLE_ADMIN,	<a href="#">Sil</a>	<a href="#">Etkin</a>
<a href="#">gumi</a>	gumi gumi	ROLE_USER,	<a href="#">Sil</a>	<a href="#">Etkin</a>

Şekil 4.13: Kullanıcı Yönetme Ekranı.

#### 4.1.3.5 Duyuru Posta Ekranı

Erasmus duyurularının hızlı bir şekilde yapılabilmesi, hatta o anda sistemde aktif olmayan öğrenciler tarafından okunabilmesi için, duyuru ara yüzünün yanısıra e-posta ile duyuru gönderme fonksiyonu da eklenmiştir. Şekil 4.14'te yönetici duyuru ekranı verilmektedir.

Konu

Gönderen

Linkler

Başlangıç Tarihi

Bitiş Tarihi

Açıklama

Şekil 4.14: Duyuru Posta Ekranı.

## **5 BULGULAR**

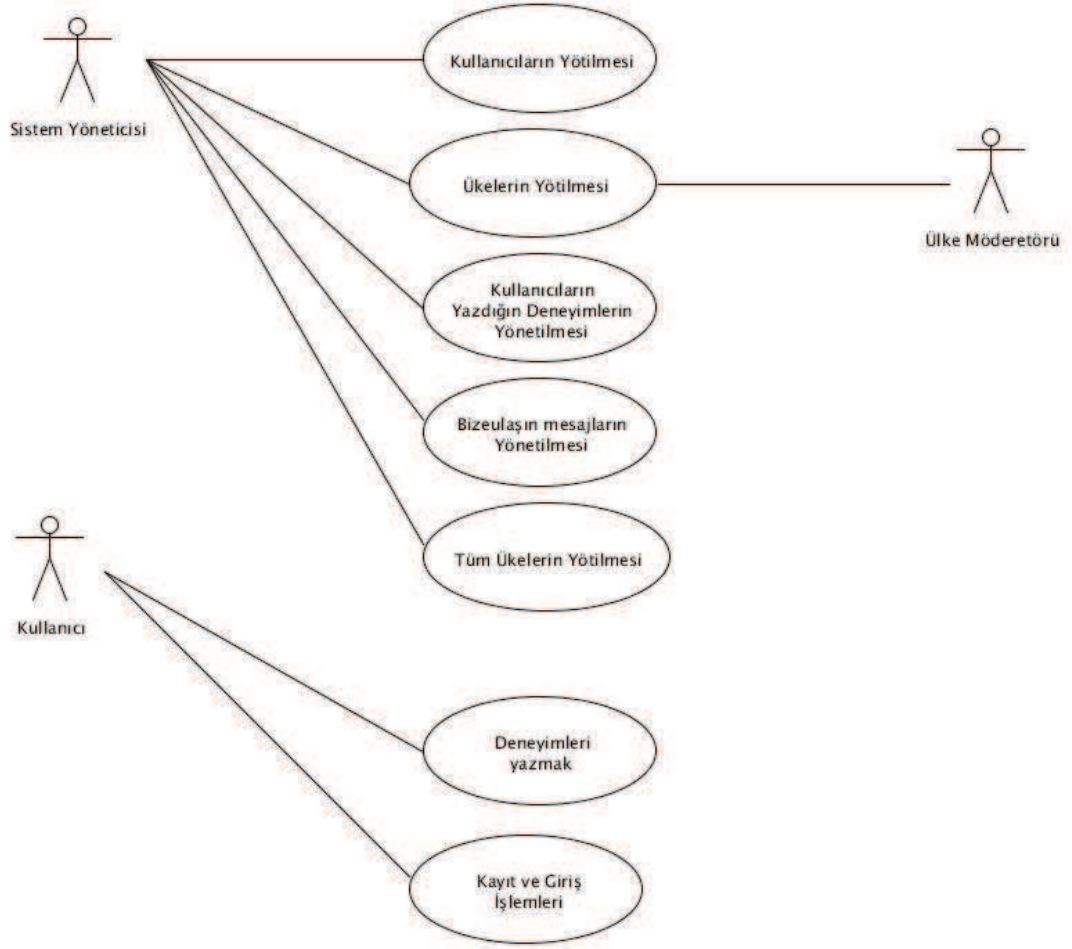
Bu bölümde İstanbul Üniversitesi Erasmus öğrencilerin deneyimleri yönetme ve paylaşma sistemi EDEPAS'ın bütün kodlama işlemleri ve gerçekleştirilmesi sürecine yer verilmektedir. Sistemin analizi ve tasarımı aşamalarının gerçekleştirilmesi UML diyagramı ile sunulmuştur (Şekil 5.1, Şekil 5.2, Şekil 5.3, Şekil 5.4, Şekil 5.5).

### **5.1 SİSTEM GELİŞTİRME**

Sistemin geliştirilmesine başlanmadan önce çeşitli firmalarla görüşülerek, framework seçenekleri arasından hangisinin daha düzgün ve özenli olduğu belirlenmiştir. Örneğin kurumsal uygulama geliştirme sürecinde hangi teknolojilerin sistemin yoğunluğunu karşılayamayacağı, hangilerinde bellek kapasitesinin fazlaca kullanıldığı, yazılım geliştirme sürecinde kullanılacak kodun niteliği gibi. Tüm bu noktalar detaylı olarak incelenmiş, ve uygulamada hangilerine yer verileceği belirlenmiştir. Daha sonra J2EE açık kaynak kodu ile veritabanı tasarımı gerçekleştirilmiştir.

### **5.2 USECASE DİYAGRAMI**

EDEPAS ile yöneticileri ve kullanıcıları; deneyim, kayıt ve giriş modülü uygulamaları sağlayıcıların sistemi ile etkileşimi Şekil 5.1'de usecase diyagramı ile gösterilmiştir. Usecase diyagramları EDEPAS mevcut işlevselliğini daha detaylı göstermek amacıyla kullanılmıştır.



Şekil 5.1: Usecase Diyagramı.

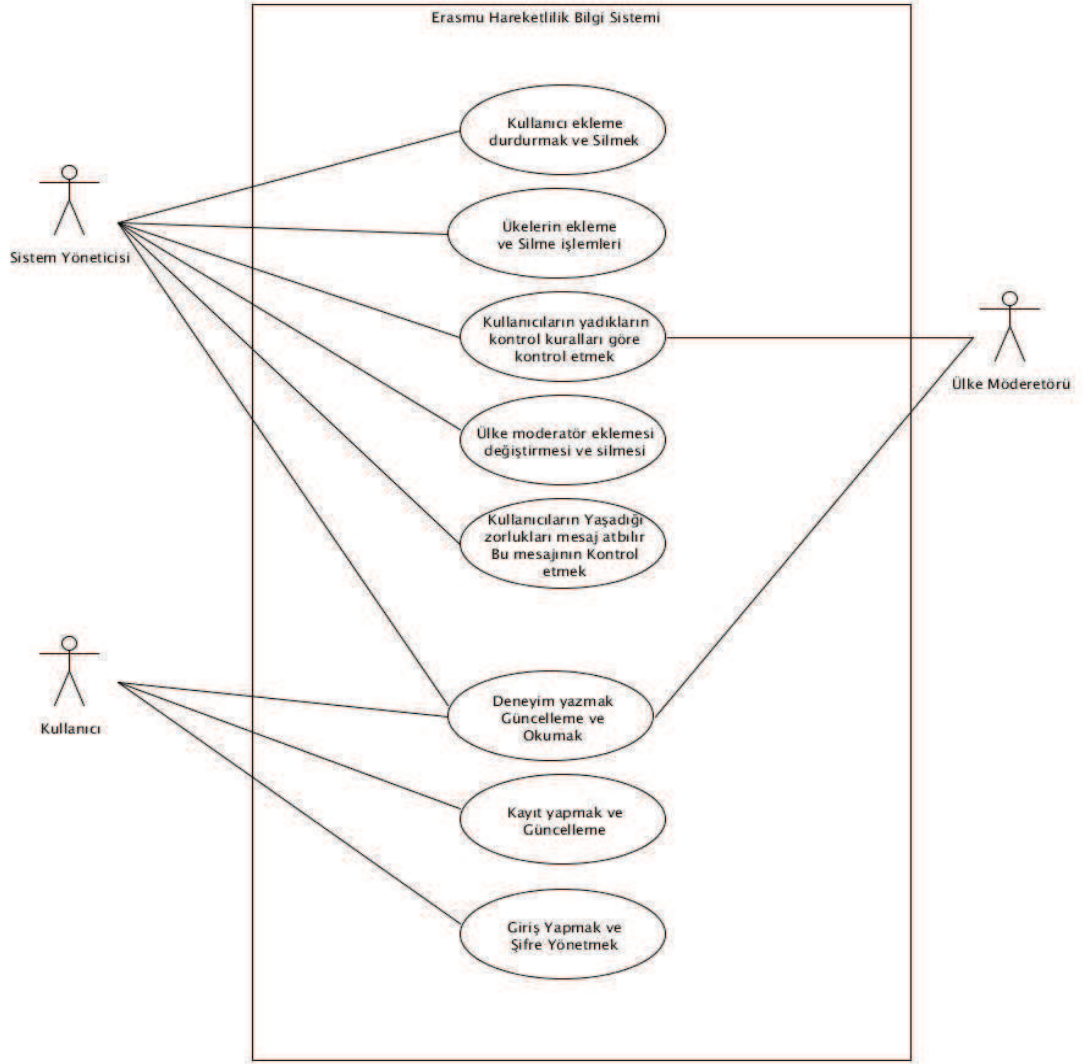
### 5.2.1 EDEPAS ile İlişki

EDEPAS sistemi, sisteme girilen öğrenci deneyim verilerini, EDEPAS veritabanı ile bağlantı kurarak gerekli bilgileri kullanmayı ve sunmayı hedeflemektedir. Bu sistem ile öğrenci deneyimlerinin paylaşımı ve güncellenmesi; yazar ve sistem yöneticinin kontrolü altındadır.

#### 5.2.1.1 Usecase Diyagramı

Sistemdeki kullanıcılar, yönetici ve moderatörü ilişkisini gösteren Usecase diyagramı Şekil 5.2’de verilmiştir.

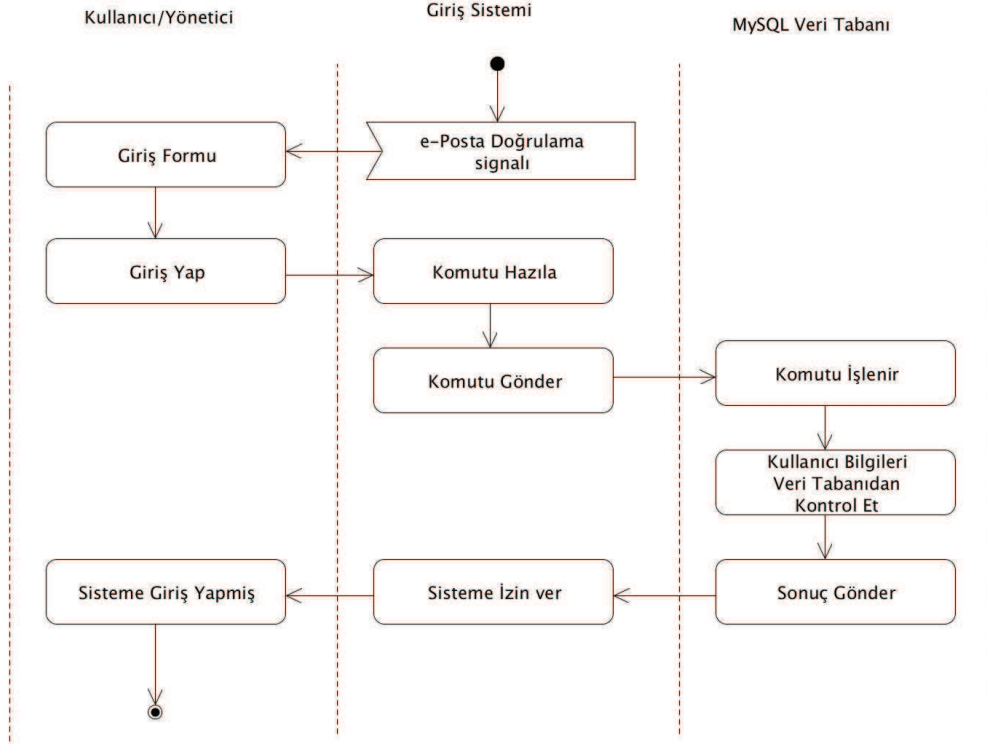




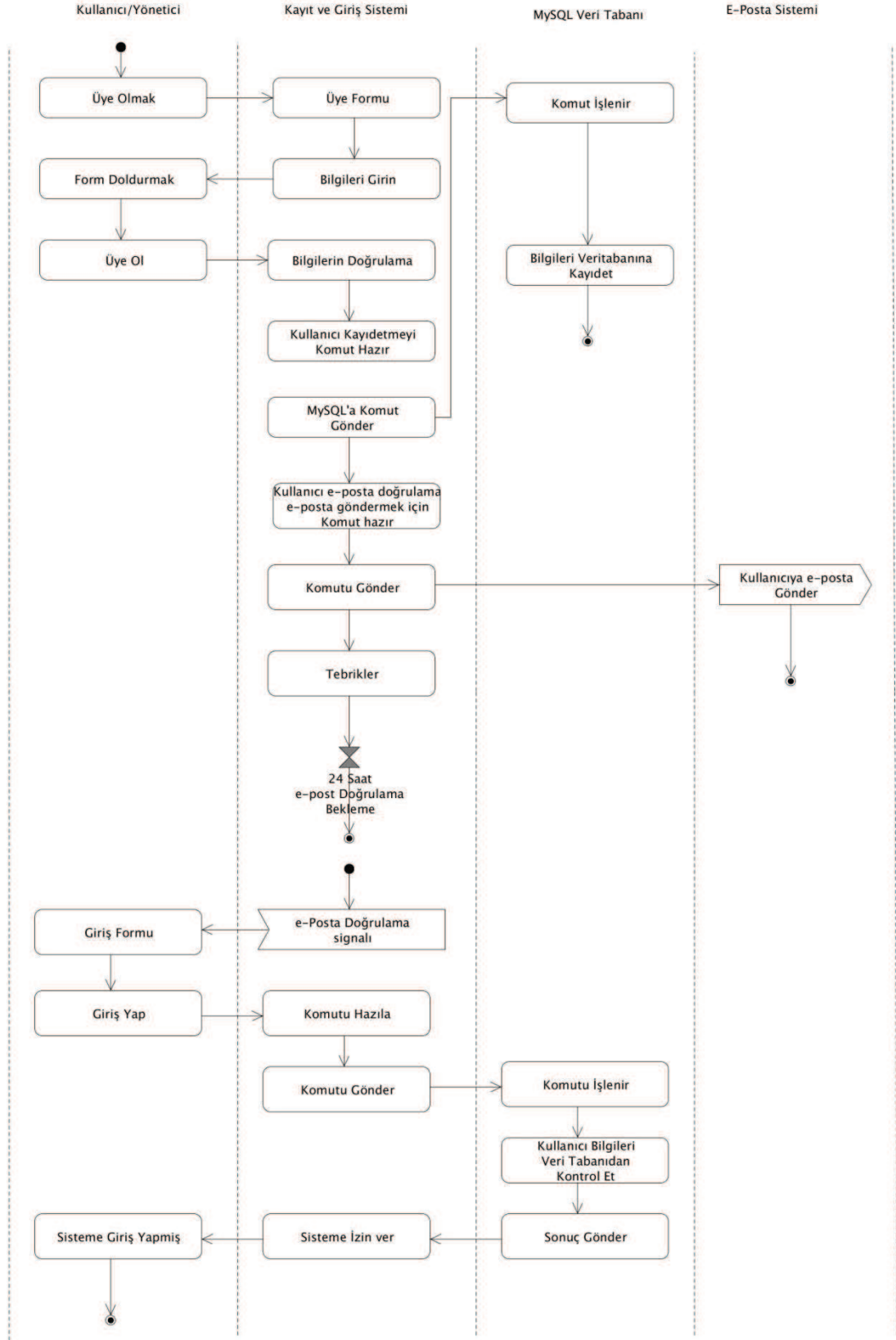
Şekil 5.2: Usecase Diyagramı.

### 5.2.2 Kayıt ve Giriş Aktivite Diyagramı

Erasmus Hareketlilik sisteminin kullanıcıları ve yöneticileri, veri girmek, güncelleme yapmak, görüntülemek ya da kontrol sağlamak üzere sisteme girebilmektedir. Sistem ilk girişte üye kaydı istemektedir. İlgili form doldurulduktan sonra e-posta ile aktivasyon yapılmaktadır. Şekil 5.3 ve Şekil 5.4’de aktivite diyagramı gösterilmiştir. Aktivite diyagramları ile mevcut sistemin giriş ve kayıt işlevselliği aktiviteler bazında daha detaylı ortaya koyulmasına yardımcı olmaktadır.



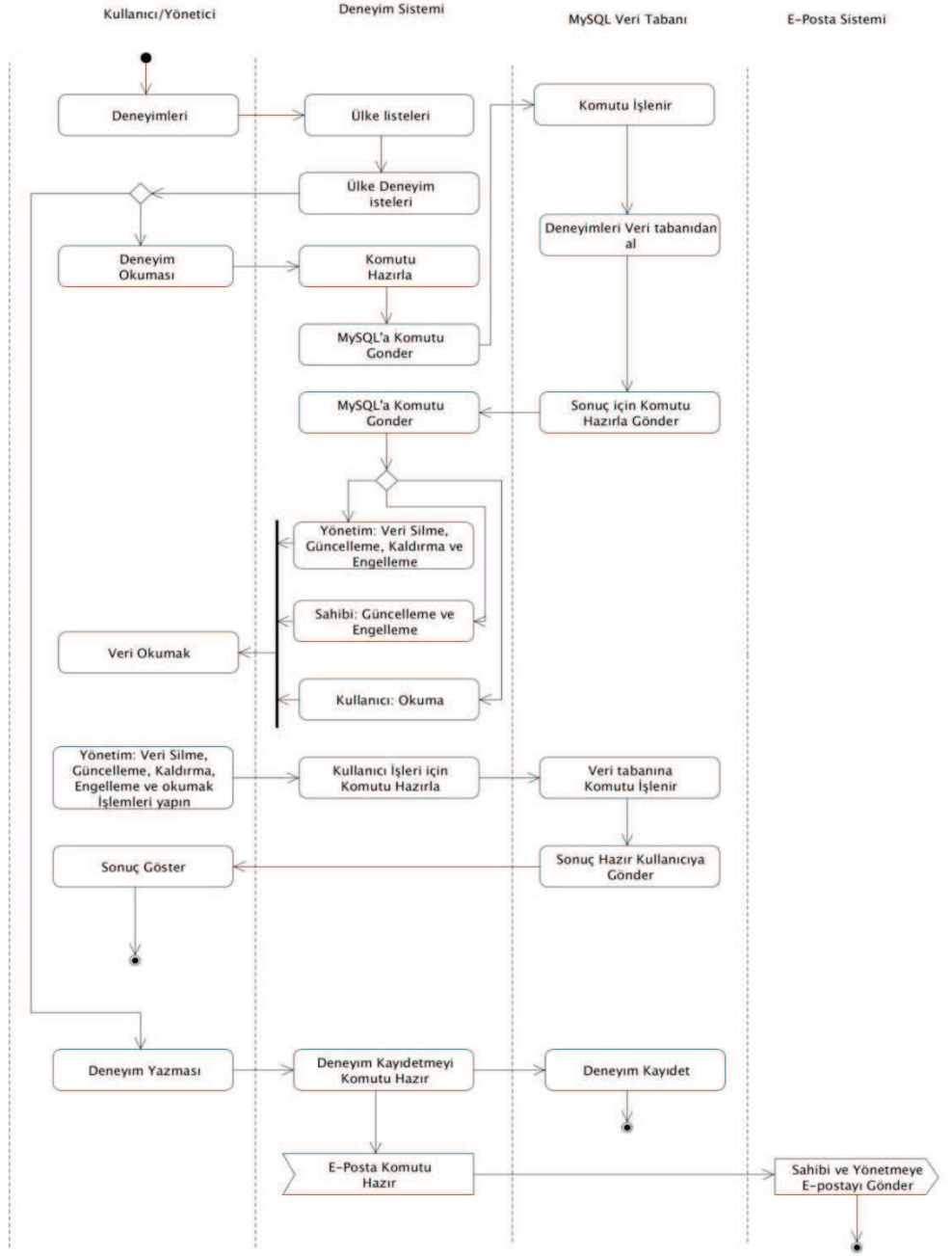
Şekil 5.3: Giriş Aktivite Diyagramı.



Şekil 5.4: Kayıt ve Giriş Aktivite Diyagramı.

### 5.2.3 Deneyim Aktivite Diyagramı

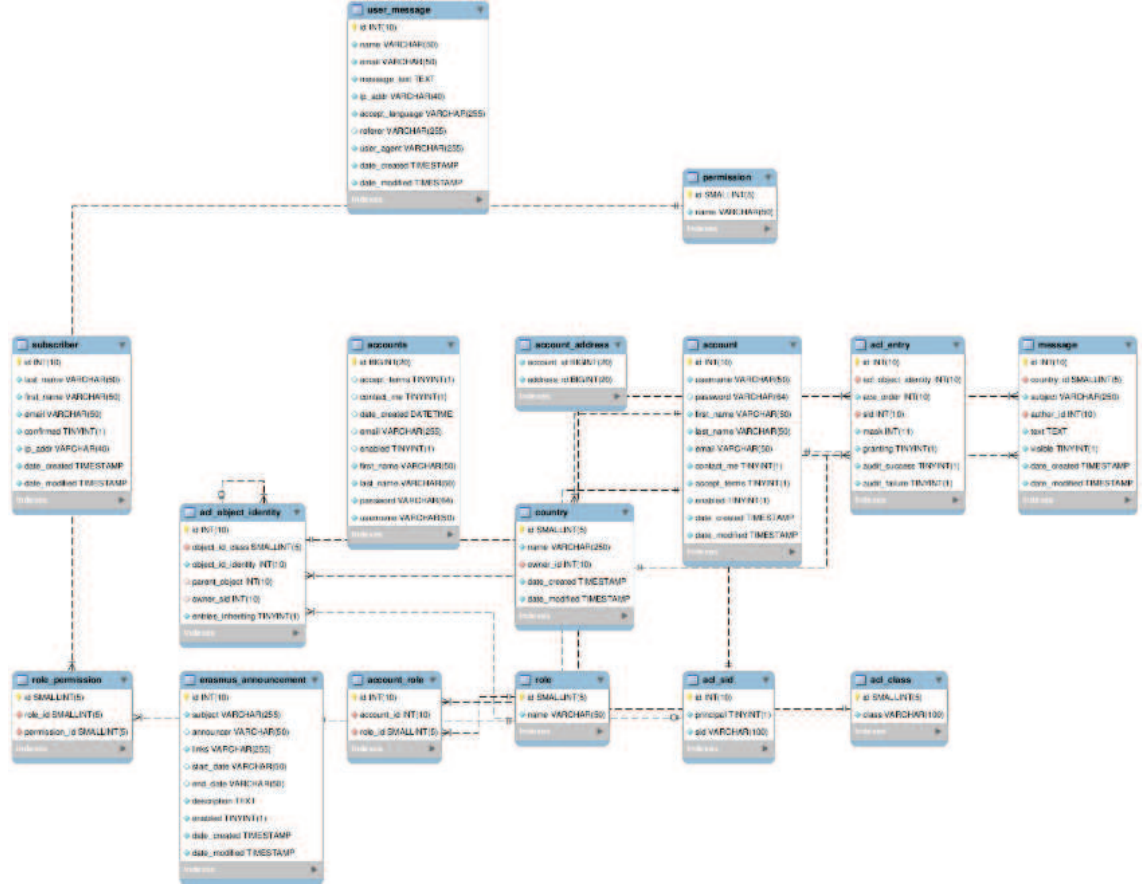
Sistemin hedefi Erasmus deneyim paylaşımıdır. Şekil 5.5’da görülen Aktivite diyagramda, öğrenci, moderatör ve yöneticilerin deneyimlerinin nasıl gerçekleştirildiği gösterilmektedir.



Şekil 5.5: Deneyim Aktivite Diyagramı.

### 5.2.4 E/R Diyagramı

Bölüm 3’de yer alan şekil 3.4’de spring güvenliğinin kullanıldığı ACL tablolarının E/R diyagramı sunulmuştur, bunlarla ve diğer veritabanında sistemin tablolarının E/R diyagramı Şekil 5.6’de gösterilmektedir.



Şekil 5.6: Kod Bileşenleri E/R Diyagramı.

### 5.2.5 Veritabanının Oluşturulması

Tasarlanan Erasmus Hareketlilik sisteminin veritabanı bir önceki bölümde detaylı olarak ER Diyagramı biçiminde sunulmuştur. Erasmus Hareketlilik veritabanı tasarlandıktan sonra Java Enterprise Eddition (JEE) uygulamasının kullanılması için gerekli koşullar araştırılmıştır. Java Enterprise Eddition ile daha güvenli ve uyumlu olması için Spring Framework ve Hibernate (ORM) yazılımı seçilmiştir. Tasarlanan veritabanında bulunan her bir tablo için MySQL veritabanı üzerinde bir sınıf oluşturulmuştur. Oluşturulan sınıflar Tablo 5.1’de verilmektedir. Bu tabloların özelliklerinden biri Hibernate ve JSR anotasyondur, bunlarla veritabanının kontrolü eklenebilmektedir.

**Tablo 5.1:** Sistem Tabloları.

Sınıf Adı	Tablo Açıklaması
Account.java	Kullanıcıların hesap tablosu
Country.java	Deneyim ülke tablosu
ErasmusAnnouncement.java	Sistemin duyuru tablosu
Message.java	Deneyim tablosu
Permission.java	Kullanıcıların güvenlik sınırlanması tablosu
Role.java	Kullanıcıların Rolu tablosu
Subscriber.java	e-posta iletişim tablosu
UserMessage.java	Bizeulaşın tablosu

### 5.2.5.1 Hibernate ve JSR Anotasyon Tablosu

Sistemin kodu içerisinde, hazırlanan veritabanı tabloları kullanılan Hibernate ve JSR anotasyonlar Tablo 5.2’de sunulmuştur. Aşağıdaki tablo anotasyonları Java kodu ile uygulanan açıklamalarına yer verilmektedir.

**Tablo 5.2:** Hibernate ve JSR Anotasyon Tablosu.

Anotasyon İsmi	Anotasyon Açıklama
@Entity	Bir sınıf ile bir tabloyu eşleştirmek
@Table	
@Id	Birincil anahtar yapmak bir özelliği
@GeneratedValue	Otomatik birincil anahtar yapmak bir özelliği
@Column	Sütun yapmak bir özelliği
@NotNull	Sütun boş olmaz özelliği
@Size	Sütun büyüklüğün özelliği
@Email	E-posta yapmak bir özelliği
@NamedQuery	Veritabanı sorgu oluşturmak özelliği
@Transient	Değişken kaydedilmez bir özelliği
@OneToMany	Birden çoğa sütun ilişkisi
@ManyToOne	Çuğuldan bire sütun ilişkisi
@JoinTable	İlişki tabloların birleştirme
@JoinColumn	İlişkili sütunları birleştirme

### 5.2.5.2 Hesap Tablosu ve Ülke Tablosu

Sistemin veritabanında Hesap tablosu ve Ülke tablosu için oluşturulan veritabanı tabloların kodları Tablo 5.3 ve 5.4’de verilmiştir. Koddan da görüleceği üzere tablodaki her bir sütun farklı farklı veri tipi olabilmektedir. Örneğin integer, long, String ve Object oluşturulmuştur. Tabloya gönderilen verilerin kaydedilmesi, güncellemesi ve geri çağırma için gerekli olan constructor ve metotlar da tabloda yer alamadığı için sistemde bırakılmıştır. Aşağıdaki tablolarda sadece sütunların adı, Hibernate ve JSR anotasyonlar oluşturulmuştur.

**Tablo 5.3:** Hesap Tablosu.

```

@Entity
@Table(name = "account")
public class Account implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private Long id;
    @NotNull
    @Size(min = 1, max = 50)
    @Column(name = "username")
    private String username;

    @NotNull
    @Size(min = 1, max = 50)
    @Column(name = "first_name")
    private String firstName;

    @NotNull
    @Size(min = 1, max = 50)
    @Column(name = "last_name")
    private String lastName;

    @NotNull
    @Size(min = 6, max = 50)
    @Email
    @Column(name = "email")
    private String email;
    @Column(name = "contact_me")
    private boolean contactMe = true;
    @AssertTrue(message = "{account.acceptTerms.assertTrue.message}")
    @Column(name = "accept_terms")
    private boolean acceptTerms = false;
    @Column(name = "enabled")
    private boolean enabled = false;
    @Column(name = "date_created")
    private Date dateCreated;
}

```



**Tablo 5.4:** Ülke Tablosu.

```
@Entity
@Table(name = "country")
public class Country {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private Long id;
    @Column(name = "name")
    private String name;
    @ManyToOne
    @JoinColumn(name = "owner_id", nullable = false)
    private Account owner;
    @Column(name = "date_created")
    private Date dateCreated;
    private List<Message> messages = new ArrayList<Message>();
    private boolean calculateMessageStats = true;
    private int numVisibleMessages;
    private Date lastVisibleMessageDate;
}
```

### 5.2.5.3 Duyuru Tablosu

Sistem veritabanında duyuru tablosu için oluşturulan veritabanı tablosu kodları Tablo 5.5'te verilmiştir. Bu tablo yöneticiden öğrencilere duyuru yapılabilmesi için kullanılmaktadır.

**Tablo 5.5:** Duyuru Tablosu.

```

@Entity
@Table(name = "erasmus_announcement")
public class ErasmusAnnouncement implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private Long id;
    @NotNull
    @Size(min = 3, max = 255)
    @Column(name = "subject")
    private String subject;
    @Size(min = 3, max = 50)
    @Column(name = "announcer")
    private String announcer;
    @NotNull
    @Column(name = "links")
    private String links;
    @Column(name = "start_date")
    private String startDate;
    @Column(name = "end_date")
    private String endDate;
    @NotNull
    @Column(name = "description")
    private String description;
    @Column(name = "enabled")
    private boolean enabled = true;
    @Column(name = "date_created")
    private Date dateCreated;
}

```

#### 5.2.5.4 Deneyim Tablosu

Sistem veritabanında Deneyim tablosu için oluşturulan veritabanı tablosu kodları Tablo 5.6'de verilmiştir. Bu tabloda öğrencilerin yazdıkların deneyimler tutulmaktadır.

**Tablo 5.6:** Deneyim Tablosu.

```

@Entity
@Table(name = "message")
public class Message {
    @Id @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private Long id;
    @ManyToOne @JoinColumn(name = "country_id", nullable = false)
    private Country country;
    @ManyToOne
    @JoinColumn(name = "author_id", nullable = false)
    private Account author;
    @NotNull
    @NotEmpty
    @Column(name = "subject")
    private String subject;
    @NotNull
    @NotEmpty
    @Column(name = "text")
    private String text;
    @Column(name = "visible")
    private boolean visible = true;
    @Column(name = "date_created")
    private Date dateCreated;
}

```

#### 5.2.5.5 İzin Tablosu

Tablo 5.7’de sistemdeki her kullanıcının sisteme, deneyimlerine ve diğer kullanıcılara erişimdeki izinlerini göstermektedir. Bu tablo Spring Framework’un ACL tablolarıyla çalışmaktadır.

**Tablo 5.7:** İzin Tablosu.

```
@Entity
@Table(name = "permission")
public class Permission implements GrantedAuthority {
    @Id @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private Long id;
    @Column(name = "name")
    private String name;
    @Override
    @Transient
    public String getAuthority() {
        return name;
    }
}
```

#### 5.2.5.6 Kullanıcı Rolü Tablosu

Sistemdeki kullanıcıların kendilerine atanan yetki düzeylerine göre belli rolleri bulunmaktadır. Bu roller; normal kullanıcı, moderatör ve yönetici şeklindedir. Tablo 5.8’de sistem rolleri için gerekli tablo ve kodlamalara yer verilmektedir.

**Tablo 5.8:** Rolu Tablosu.

```

@Entity
@Table(name = "role")
public class Role implements GrantedAuthority {
    @Id @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private Long id;
    @Column(name = "name")
    private String name;
    @ManyToMany(fetch = FetchType.EAGER)
    private Set<Permission> permissions = new HashSet<Permission>();
}

```

### 5.2.6 E-posta Sistemi

Bugünkü kurumsal sistemlerde, kullanıcıların sisteme kayıt edilmelerine ve kaydın başarılı olup olmadığını kontrol etme amaçlı çeşitli yöntemler kullanılmaktadır. Telefon onayı, e-posta onayı, captcha ile kontrol bu yöntemlerden en sıklıkla karşılaşılanlarıdır. EDEPAS sisteminde bu kontrol onayı e-posta yolu ile yapılmaktadır. Çalışma kapsamında Gmail E-posta sunucu kullanılmıştır. Gmail e-posta servisi kullanmak için, başka firmadan motoru yüklemek gerekmektedir. Bu motor yardımı ile Gmail e-posta sunucusunun kullanımı kolaylaşmaktadır. Motorun adı velocityEngine'dir, sistemde src/main/resources'de dosyalardan beans-service.xml ve bu dosya içine aşağıdaki kodu velocityEngine ve Gmail e-posta sunucu ayarlamıştır.

```

<bean id="mailSender"
class="org.springframework.mail.javamail.JavaMailSenderImpl">
    <property name="host" value="smtp.gmail.com" />
    <property name="port" value="587" />
    <property name="username" value="kullanacak Gmail e-posta" />
    <property name="password" value="e-posta şifresi" />
    <property name="javaMailProperties">
        <props>
            <prop key="mail.transport.protocol">smtp</prop>
            <prop key="mail.smtp.auth">>true</prop>
            <prop key="mail.smtp.starttls.enable">>true</prop>
        </props>
    </property>
</bean>

```

```

<!-- p:resourceLoaderPath aşağıdaki dosıların olduğu yeredir -->
<bean id="velocityEngine"
class="org.springframework.ui.velocity.VelocityEngineFactoryBean"
p:resourceLoaderPath="classpath:/velocity/" />

```

```

<util:properties id="contactServiceProps"
location="classpath:/spring/contactService.properties" />

```

```

<util:properties id="mailingListServiceProps"
location="classpath:/spring/maillingListService.properties" />

```

```

<util:properties id="accountServiceProps"
location="classpath:/spring/accountService.properties" />

```

### 5.2.7 Kayıt Onaylama E-posta ve URL Ayarlama

Öğrenci sistemde kayıt ekranından formu doldurmakta, sonlandır butonuna basınca, eğer doğru bilgileri girilmişse, otomatik bir e-posta atılmaktadır. Bu e-posta içerisinde onay linki bağlantısı bulunmaktadır. Bu bağlantı linkinin tıklanması ile sisteme giriş izni verilmiş olmaktadır.

**noReplyEmailAddress**=[noreply@example.com](mailto:noreply@example.com)

**confirmAccountRegistrationUrl**=<http://localhost:8080/dosya-yeri/dosya-adi.html>

**confirmAccountDeletionUrl**= <http://localhost:8080/dosya-yeri/dosya-adi.html>

**confirmationKey**=15cl@nbt\$1s\$h@IBharLR557+B

**noReplyEmailAddress**: Mesaj gönderen e-posta

**confirmAccountRegistrationUrl**: Onay link bağlantı

**confirmAccountDeletionUrl**: Hesap silme onay link bağlantı

### 5.2.8 “Bizeulasın” E-posta Ayarlama

Sistem kullanımlarında kullanıcı hatası, sistemin tam olarak anlaşılabilmesi, sunucu ya da sistem kaynaklı pek çok hatadan dolayı, kullanıcılar sisteme girmekte, veri girmekte, sistemde dolaşmakta sıkıntılar yaşayabilmektedir. Bize ulaşın formu ile sıkıntı ve şikayetlerin alınması, takip edilmesi ve çözüme ulaştırılması mümkündür. Gönderilen şikayetlerin ilgili kişilere ulaştırılabilmesi için `src/main/resources/spring/contactService.properties` kullanılmıştır.

**sendConfirmation=true**

**notifyAdmin=true**

**noReplyEmailAddress=[noreply@example.com](mailto:noreply@example.com)**

**adminEmailAddress=[örneke-posta@örnek.com](mailto:örneke-posta@örnek.com)**

Bundan sonra 4.2.7’de gösterilmiş dosyada velocityEngine’ne yukarıd linkleri tutan dosya adı koymak gerekiyor. Örneğin:

```
<util:properties id="contactServiceProps"
location="classpath:/spring/contactService.properties" />
```

### 5.2.9 Abone E-posta ve URL Ayarlama

Kişilerin sisteme kayıt olma sürecinin ve aktivasyon linkinin gönderilebilmesi amacıyla kayıt dosyasından ayrı bir dosya oluşturulmuştur. Kullanılan bu dosyanın yer ve adı `src/main/resources/spring/maillingListService.properties` şeklindedir.

**noReplyEmailAddress=[noreply@example.com](mailto:noreply@example.com)**

**confirmSubscriptionUrl=<http://localhost:8080/dosya-yeri/dosya-adi.html>**

**confirmUnsubscriptionUrl= <http://localhost:8080/dosya-yeri/dosya-adi.html>**

**confirmationKey=l5cl@nbt\$1s\$h@IBharLR557+B**

**Velocity dosyada ayarlama:**

```
<util:properties id="maillingListServiceProps"
location="classpath:/spring/maillingListService.properties" />
```

### 5.2.10 Sistem Gereksinimler

Sistem web tabanlı olmasından dolayı sağlıklı bir biçimde erişimin sağlanması için internet bağlantısının olması gerekmektedir. Benzer şekilde sistemin diğer gereksinimleri aşağıdaki gibidir:

- MySQL veritabanı versiyon 5.0 ve üstü
- Sistem veritabanlarının MySQL'a yüklemesi
- MySQL komutlarla acIDB database hesap (account) tablosu
- Tomcat sunucu 7.0 ve üstü veya JEE sunucu
- Java 7.0 ve üstü
- MySQL connector/J 5.0 ve üstü
- Kodu derlemek için Eclipse 4.3 ve üstü.

### 5.3 SİSTEMİN TESTİ VE SONUÇLARI

Uygulama geliştirme süreci boyunca birçok hata ile karşı karşıya kalınması oldukça doğal bir durumdur. Test Odaklı Geliştirme (Test Driven Development) en basit haliyle; uygulama geliştirirken test kodlarını yazarak başlamak ve diğer gereken sınıfların ve metodların bu test koduna göre yazılmasıdır. Bu test yazılan fonksiyonların çalışıp çalışmadığının testlerini yapmak için kullanılır. Ekstrem programlama ve günümüzdeki birçok çevik (agile) modern yazılım geliştirme süreçlerinin kodlama bakımından bel kemiğini oluşturmaktadır [38].

Günümüzde en çok kullanılan test altyapısı Birim Testidir (Unit Test). Bir java uygulaması geliştirme sürecinde kullanılan test aracı JUnit'dir. Bu araç, Birim Testleri yazılmasını ve kontrol edilmesini kolaylaştıracak birçok Framework içermektedir. Bir projede bu araçlardan uygun olanını kullanacağı neredeyse her dil için bir framework bulunmaktadır. Bu çalışmada da Java ile geliştirilen EDEPAS için Birim Testi kullanmıştır.

Örnek olarak, öğrenci kayıt için formu seçer ve gerekli bilgileri girdikten sonra kayıt butonuna tıklar. Bu aşama Birim Test ile kontrol edilmezse sistem "Runtime" hatası



verip kullanıcıyı durdurabilir. Bu testler, alınabilecek hataların daha yazılım geliştirme sürecinde fark edilmesini sağlayarak yazılım geliştirme sürecinin sonlanmasına katkı sağlar. Birim Test, sistemin güvenliğini otomatik olarak ele alabilir ve yazılımcıya güvenlik sağlama anlamında çok daha az iş bırakabilir. Bu çalışmada kullanılan Birim Test ile ilgili elde edilen sonuçlar aşağıda verilmektedir.

#### **Sistemin Birim Testinin Sonuçları:**

1. Kodun anlaşılmasını kolaylaştırmıştır.
2. Daha hızlı yazılım geliştirmeyi sağlamıştır.
3. Kodların hataları azalmıştır.
4. Kodların kalitesi artmıştır.
5. Hataların çabuk tespit edilip düzenlenmesi sağlanmıştır.

## 6 TARTIŞMA VE SONUÇ

Erasmus deęişim programı ile giden ve gelen öğrenciler üniversiteler açısından önemli bir bilgi kaynağını barındırmaktadır. Bu yolla kazanılan bilgi kaynaklarının doğru yönetilmesi gelecekteki öğrenci hareketliliğine önemli bir ivme ve kalite kazandıracaktır.

Bir öğrenci için başka bir ülkeye gitmek, orada eğitim almak bazen oldukça zorlu bir süreç olarak karşısına çıkabilmektedir: ilk adım olan yasal işlemler, sonrasında ülkeler arası görüşme ve anlaşmalar, yabancı ülkeye varış ve orada yaşam. Tüm bu adımların kendi içinde büyük zorluklar barındırmasının yanı sıra, öğrencinin ilk kez bu deneyimle karşılaşması bu zorlukları arttırıcı bir etken oluşturmaktadır.

Bu çalışma ile gerçekleştirilen sistem; deneyim yönetiminin başarıya ulaşması için en önemli unsurlardan birisi olan Erasmus hareketlilik deneyiminin derlenmesi ve etkili bir biçimde yönetilmesini amaçlamaktadır. Özellikle günümüzde İnternet tabanlı uygulamalarda yeni teknolojilerin kullanımı arttığı için bu tez kapsamında Erasmus deneyim paylaşım sistemi uygulamasının web tabanlı olarak geliştirilmesi tercih edilmiştir.

Bu amaçla Java Enterprise Edition (J2EE) kullanılarak Erasmus deneyim paylaşım sistemi tasarlanmış, prototipi, geliştirilmiş ve test edilmiştir. Geliştirilen sistem için kullanılan teknoloji, günümüzde kurumsal uygulamalarda da kullanılmakta ve örneğin bankacılıkta da sıkça tercih eden teknolojilerden biri Java teknolojisidir. JEE/J2EE teknolojileri kurumsal yazılım geliştirme verimliliği arttırmaktadır.

Bir sistemin web altyapısı üzerinde çalışıyor olması, platform bağımsızlığına büyük katkı sağlar. Diğer web sistemleri gibi EDEPAS da internet bağlantısı ve internet tarayıcısı mevcut herhangi bir işletim sistemine sahip, herhangi bir cihazda çalıştırılabilmektedir. Böylece kullanıcıların belirli platform ya da cihazlara bağımlılıkları kalmamakla birlikte sistemin erişilebilirliğinin de oldukça yüksek olması beklenmektedir.

Geliştirilen sistemin sağlıklı olarak çalışıp çalışmadığı yine tez çalışması kapsamında test edilmiştir. Sistemin başlangıçta kurgulanan amaçlara hizmet edip etmediği, yazılımın çalışır hale gelmesinin ardından kontrol edilmiş, ayrıca çalışır haldeki sistemin kullanımında herhangi bir aksaklık olup olmadığı yazılımcı tarafından kullanım testleri ile kontrol edilmiş ve yazılım testlerden başarı ile geçmiştir.

Sistemin veritabanında bulunan bilgiler MySQL'in kendi güvenlik önlemleri ile korunmakta, ayrıca sistem içerisinde yapılan her işlem birden fazla katmanlı güvenlik kontrollerinden geçerek işlenmektedir. Bu sayede, sisteme bilgilerini yüklemiş kişilerin bilgilerine, sistem yöneticisinin izni olmadan erişilebilmesi mümkün olamayacaktır.

Bu tez çalışması kapsamında geliştirilen sistemin Erasmus ile ilgili tüm süreçte yardımcı hizmet olabileceği düşünülmektedir. Sistemin sağlayabileceği faydalardan bazıları aşağıdaki gibidir:

1. Arkadaş edinmek,
2. Farklı kültürler hakkında kazanımlar elde etmek,
3. Aynı deneyimi daha önce yaşamış bireylerden tecrübe aktarımı sağlayabilmek,
4. Erasmus ofisi ile sıkı bir ilişki ve iletişim içerisinde kalabilmek,
5. Erasmus sürecinde karşılaşılabilecek her türlü durum ile ilgili bilgi sahibi olmak.

EDEPAS'ın, Erasmus ile ilgili olarak; öncesini, süreci ve sonrasını kapsayan neredeyse tüm durumlar için bir yol gösterici olması beklenmektedir. Sistem özellikle kurum ve bireyler arasındaki ilişkinin çok daha kuvvetli ve etkin olmasında ön plana çıkmaktadır.

Erasmus veya ülke kültürü sorunlarına çözüm bulmak amacıyla, öğrencilerin birbiriyle yüz yüze ilişkiler kurmak ve birbirlerine bilgi sunmak her zaman mümkün değildir. Bu durum Erasmus gibi değişim programlarında görevli yöneticilerle öğrenciler arasında kopukluklar yaşanabilmesine neden olabilmektedir. EDEPAS ile hem yönetici açısından, hem de öğrenciler açısından verimli bir Erasmus deneyim süreci oluşturulabilmesinin mümkün olacağı düşünülmektedir. Bunun da ötesinde EDEPAS'ın bulut (cloud) ile entegre edilmesi sonucunda mekandan bağımsız bir şekilde sistemin kullanılması ve daha yüksek kapasitede analizler yapılarak kapsamlı sonuçlar üretmek mümkün olabilir. Böylece tüm Erasmus öğrencileri ve yöneticileri için hem zaman hem de emek tasarrufu sağlanmış olacaktır.

Geliştirilen sistem, ilk aşamada yalnızca İstanbul Üniversitesi e-posta hesabına sahip kişiler tarafından kullanılabilirdiğinden, farklı üniversite ya da yurtdışından gelen öğrencilerin kullanımına kapalıdır. Bu sayede sistem, Facebook ve benzeri sosyal ağlardan farklı olarak, yalnızca belirli grubun iletişimde kalmasına olanak sağlayacak, ayrıca sistem içerisindeki kişilerin yönetiminin, sistemden sorumlu üniversite birimi tarafından yönetilmesini sağlayabilecektir. Sistemin daha evrensel hale getirilmesi, sonraki çalışma adımları arasında yer almaktadır.

İstanbul Üniversitesi gibi her yıl yüzlerce kişinin Erasmus'dan faydalandığı bir kurum için, tecrübe aktarımının, iletişimin ve yasal işlemlerin elektronik bir ortamda gerçekleştirilmesi hayati önem taşımaktadır. Böylesi büyük bir kurumda, çok fazla sayıda kişinin her durumda Erasmus ofisine başvurmak yerine anında cevap alabileceği çevrimiçi bir kaynağın var olması bir üniversitenin kalitesini ve öğrenci memnuniyetini çok daha üst seviyelere çıkarabilecektir.

## KAYNAKLAR

- [1]. Elmasri, R., and Navathe, S.B., 2011, *Fundamental of Database Systems*, Sixth Edition. Addison-Wesley ISBN-13: 978-0-136-08620-8 pp. 75 – 79.
- [2]. Elmasri, R., and Navathe, S.B., 2011, *Fundamental of Database Systems*, Sixth Edition. Addison-Wesley ISBN-13: 978-0-136-08620-8 pp. 481 – 491.
- [3]. Peter, H., ve Lasse, V., 2011, *Design of Web-Based Information Systems New Challenges for Systems Development? Global Co-Operation in the New Millennium The 9th European Conference on Information Systems Bled, Slovenia, June 27-29, 2001.*
- [4]. Toprak, M., ve Erdoğan, A., 2012, *Yaşamboyu Öğrenme: Kavram, Politika, Araçlar ve Uygulama*, Yükseköğretim ve Bilim Dergisi, DOI: 10.5961.
- [5]. Günay, D., 2011, *Türk Yükseköğretiminin Yeniden Yapılandırılması Bağlamında Sorunlar, Eğilimler, İlkeler ve Öneriler – I*, Yükseköğretim ve Bilim Dergisi, DOI: 10.5961.
- [6]. Ungan, S., 2006, *Avrupa Birliğinin Dil Öğretimine Karşı Tutumu ve Türkçe'nin Yabancı Dil Olarak Öğretilmesi*, Türkçe'nin Yabancı Dil Olarak.
- [7]. Mehmet, I., ve Ercan O., 2006, *Bilginin Bir Stratejik Güç Olarak Önemi Ve Örgütlerde Bilgi Yönetimi*, Selçuk Üniversitesi Karaman İ.İ.B.F Dergisi.
- [8]. Dirlik, S., 2008, *Yönetim Bilişim Sistemlerinin Örgüt Yapısına Ve Karar Vermeye Etkisi: Çoklu Örnek Olay Araştırması*, Muğla Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Yüksek Lisans Tezi, Muğla.
- [9]. Yüksek Öğretim Kurulu Web Sayfası, <http://www.yok.gov.tr/web/farabi/farabi-degisim-programi>, [Ziyaret tarihi: 10 Nisan 2014].
- [10]. Akdeniz Üniversitesi Web Sayfası, <http://farabi.akdeniz.edu.tr/sikca-sorulan-sorular-9bc.i3.farabi-programi-nedir->, [Ziyaret tarihi: 11 Mayıs 2014].
- [11]. Hubert, B.V. H., ve Marja J.V., 2008, *Wine Is for Drinking, Water Is for Washing: Student Opinions About International Exchange Programs.*
- [12]. Kelly, R., ve Robert O., 2004, *Outcome Assessment of a Short-Term U.S. Thailand Student Exchange Program.*
- [13]. Muğla Sıtkı Koçman Üniversitesi Web Sayfası, <http://erasmus.mu.edu.tr/tr/erasmus-programi-1450> [Ziyaret tarihi: 16 Ekim 2013].
- [14]. Yüksek Lisans Web Sayfası, <http://yukseklisans.com.tr/erasmus.php> [Ziyaret tarihi: 16 Ekim 2013].

- [15]. EU Center İstanbul Teknikal Üniversitesi Web Sayfası, <http://www.eucentre.itu.edu.tr/Icerik.aspx?sid=240> [Ziyaret tarihi: 16 Ekim 2013].
- [16]. Can., 2008, <http://www.erasmusrehberi.com/erasmus-programi/> [Ziyaret tarihi: 16 Ekim 2013].
- [17]. EK-6., 2012, *Hayatboyu Öğrenme Programı (LLP) Yükseköğretim Kurumları İçin Erasmus Uygulama El Kitabı*, Akademik Yıl 2012/2013.
- [18]. İstanbul Üniversitesi Web Sayfası, <http://www.istanbul.edu.tr/uaik/ABegitim/> [Ziyaret tarihi: 16 Ekim 2013].
- [19]. Özdemirci, F., ve Aydın, C., 2008, *Türk Kütüphaneciliği* 22, 1, 59-81.
- [20]. Gümüştekin, G.E., 2005, [http://www.ekodialog.com/makaleler/bilgi\\_yonetimi\\_genel.html](http://www.ekodialog.com/makaleler/bilgi_yonetimi_genel.html) [Ziyaret tarihi: 01 Mayıs 2014].
- [21]. Hüseyin, O., 2006, *Bilgi Yönetimi*, Bilgi... içinde (99-108). Ed.: Hüseyin Odabaş ve Hakan Anameriç. Ankara: Referans Yayıncılık.
- [22]. Ok, K., Yalçınkaya, İ., Ayhan, K., Özçakır, F.C., Baktır, H.Ö., Huysal, K., Erkoç, M.F., ve Çelik B., 2013, *Bilgi ve Bilginin Yönetimi*, Papatya yayıncılık eğitim, ISBN: 978-605-4220-38-0.
- [23]. Yılmaz, M., 2009, *Enformasyon ve bilgi kavramları bağlamında enformasyon yönetimi ve bilgi yönetimi*.
- [24]. Hüseyin, O., 2005, *Bilgi Yönetimi Sistemi*, Bilgi Çağı Bilgi Yönetimi ve Bilgi Sistemleri. Ed.: Coşkun Can Aktan ve İstiklal Y. Vural. Konya: Çizgi Kitabevi.
- [25]. Uzun, H., ve Durna, U., 2008, *Niğde Üniversitesi İİBF Dergisi*, Haziran 2008, Cilt: 1, Sayı: 1, 33-40.
- [26]. Akan, C.C., ve Vural, İ.Y., 2005, *Bilgi Çağı Bilgi Yönetimi ve Bilgi Sistemleri*.
- [27]. Mustafa, K., ve Orhan, E., 2002, *Web Tabanlı Öğretme-Öğrenme ve Değerlendirme Süreçlerinde Teknoloji Eğitimi Bilgi Koluna İlişkin Bir Uygulama Örneği*.
- [28]. Alkan, R.M., Kalkan, Y., İpbüker, C. ve Yanalak, M., 2003, *İnternet Ortamında Coğrafi Bilgi Sistemi Uygulaması*, Web-tabanlı Haliç Bilgi Sistemi (WHBS) Bilimsel Toplantısı Coğrafi Bilgi Sistemleri ve Jeodezik Ağlar Çalıştayı Konya.
- [29]. Hale, S., Erdem, E., ve Önal, A., 2004, *xml ve veritabanı entegrasyonu ile esnek uygulama geliştirme*.
- [30]. Mark, M., Jim, C., ve Joseph, D.G., 2003, *MySQL and Java Developer's Guide* pp. 30-31.
- [31]. Oracle Web Sayfası, [https://java.com/tr/download/faq/whatis\\_java.xml](https://java.com/tr/download/faq/whatis_java.xml) [Ziyaret tarihi: 16 Mayıs 2013].

- [32]. Sefer, A., 2002, <http://www.csharpedir.com/articles/read/?id=6> [Ziyaret tarihi: 16 Mayıs 2013].
- [33]. Craig, W., 2011, *Spring in action*, Third Edition., Manning Publications Co, ISBN: 9781935182351.
- [34]. Orçun, D., 2004, *Spring Java/J2EE uygulama Çatısı & Spring AOP*.
- [35]. Hibernate Web Sayfası, <http://docs.jboss.org/hibernate/orm/4.2/quickstart/en-US/html/pr01.html> [Ziyaret tarihi: 30 Nisan 2014].
- [36]. Bauer, C., ve King, G., 2007, *Java Persistence with hibernate*, Revised Edition of Hibernate in Action, Manning Publications Co, ISBN 1-932394-88-5.
- [37]. Apache Software Foundation Web Sayfası, <http://maven.apache.org/what-is-maven.html> [Ziyaret tarihi: 30 Nisan 2014].
- [38]. Acar, Ö., 2008, *Unit Testing Konsetpleri*, Extreme Programming.

## ÖZGEÇMİŞ



### Kişisel Bilgiler

Adı Soyadı	Bello Muhammad
Uyruğu	Nijerya
Doğum tarihi, Yeri	01 / 08 / 1983
Telefon	05378133998
E-mail	muhdgumi@gmail.com
Web adres	

### Eğitim

Derece	Kurum/Anabilim Dalı/Programı	Yılı
Doktora	İ.Ü. Fen Bilimleri Enstitüsü/ /	
Yüksek Lisans	İ.Ü. Fen Bilimleri Enstitüsü/Enformatik Anabilim Dalı /	2014
Lisans	Hamdard Üniversitesi Yeni Delhi India	2007
Lise	Government Unity Secondary School Gummi, Zamfara State	2003

### Makaleler / Bildiriler

Bello, M., Emre, A., Mohammed, A., Mohanad, A., Muhammad, I., Serges, S. and Sevinç, G., 2012, An Overview on ICT Development in Turkey, Pakistan, Middle East and Africa IDIA2012 Conference: International Development Informatics Association 6 – 8 September, 2012, Istanbul

Bello, M., Şebnem, Ö., and Sevinç, G., 2014, Cultural e-learning through Erasmus experience management 5th International Future-Learning Conference on Innovations in Learning for the Future 2014: e-Learning, İstanbul, TURKEY