

Veritabanı Yönetim Sistemleri Projesi

Akademik Sosyal-Ağ

COLLABORATORS

	<i>TITLE :</i> Veritabanı Yönetim Sistemleri Projesi		<i>REFERENCE :</i>
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Grup 06 and Yiğit Bekir Kaya, 040090501	29.12.2011	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Giriş	1
1.1 Proje Tanımı	1
1.2 Geliştirme ve Çalıştırma Ortamı	1
1.3 İş Bölümü	1
2 Kurulum	3
3 Kullanım Kılavuzu	4
3.1 Profile Sayfası	4
3.2 Profil Menüleri	5
3.2.1 Edit Profile	5
3.2.1.1 Add/Remove Website	5
3.2.1.2 Add/Remove Email	5
3.2.1.3 Change Birthdate	6
3.2.1.4 Change Profile Image	6
3.2.2 Manage Connections	6
3.2.3 Manage Rewards	7
3.2.4 Manage Experiences	7
4 Teknik Kılavuz	8
4.1 Veritabanı Tasarımı	8
4.2 Yazılım Tasarımı	9
4.2.1 Nesneye Dayalı Tasarım	9
4.2.2 Wicket Kullanımı Tasarımı	10
4.2.3 Oturum (Session) Tasarımı	10
4.2.4 Veritabanı Yönetici Tasarımı	10
4.2.5 Model Sınıfları Tasarımı	11
4.2.5.1 Profile	11
4.2.5.2 Message	11
4.2.5.3 Experience	11
4.2.5.4 Reward	12

4.2.6	Kontrolör Sınıfların Tasarımı	12
4.2.6.1	Profile Manager	12
4.2.6.2	Message Manager	12
4.2.6.3	Experience Manager	12
4.2.6.4	RewardManager	12
4.2.7	Görüntü (View) Sınıfları Tasarımı	12

List of Figures

3.1	Profil Sayası Genel Bakış	4
3.2	Profil Menüleri	5
3.3	Web Sayfası Ekle/Çıkar	5
3.4	E-posta Ekle/Çıkar	6
3.5	Doğum Günü Değiştir	6
3.6	Profil Resmi Değiştir	6
3.7	Bağlantı Yönetimi	7
3.8	Ödül Yönetimi	7
4.1	Ödül Yönetimi	8
4.2	Sınıf Diyagramı	9
4.3	Wicket Panel Sınıfları	10

List of Tables

4.1	ProfileManager Metotları	13
4.2	Message Manager Metotları	14
4.3	ExperienceManager Metotları	14
4.4	RewardManager Metotları	14
4.5	Görüntü Sınıfları	15

Chapter 1

Giriş

1.1 Proje Tanımı

Akademik sosyal ağ projesi, sosyal medyanın gücünü akademik birikim ile birleştirmeyi amaçlayan bir sosyal ağ projesidir. Projemizin esin kaynağı, Ninova öğrenci otomasyon sistemi ve günümüzde revaçta olan çeşitli profesyonel ve kişisel sosyal ağlar (Facebook, Twitter, LinkedIn gibi) olmuştur.

Projemizin amaçları arasında, Öğrenciler, akademisyenler ve Ar-Ge şirketlerini ortak bir platformda birleştirmek, çok amaçlı bir paylaşım sistemi kurmak, güvenli eğitim kurumları ile uzaktan eğitim platformu kurmak ve iş bulma ve proje geliştirme ortamı sunmak da vardır.

Akademik sosyal ağ projesi beş ana alt projeden oluşmaktadır. Bunlar; üniversite yapısı yönetimi, giriş işlemleri ve profil yönetimi, iletişim platformu (Forum ve Profil mesajlaşması) yönetimi, dış bağlantıların (olaylar, yarışmalar, projeler ve işler) yönetimi ve dosya yönetimidir.

1.2 Geliştirme ve Çalıştırma Ortamı

Proje Mysql veritabanı ve Java programlama dili kullanılarak hazırlanmıştır. Projenin hazırlanması sırasında proje Apache Tomcat 7.0.23 sunucusu üzerinde çalıştırılmış, testler bu sunucu üzerinde yapılmıştır. Java ve html arasındaki bağlantıyı kurmak için Wicket 1.4.10 arayüzü kullanılmıştır. Bu arayüze ek olarak projeye mysql veritabanı komutlarını desteklemesi için MYSQL JDBC Driver eklenmiştir. Ayrıca FTP sunucusuyla iletişimi sağlamak amacıyla commons-net-3.0.1.jar kütüphanesi de projeye eklenmiştir.

Proje belli bir işletim sistemine bağlı olmayıp, Netbeans 7.0.1 IDE'sinde geliştirilmiştir.

Bunlara ek olarak, görselliği sağlamak amacıyla Wicket framework'üne ek olarak CSS ve JavaScript eklenmiştir. Bu eklentiler ise projenin tarayıcıya bağımlılığını getirmiştir. Mozilla Firefox ve Google Chrome'da bazı görsellikler gösterilebilirken bazıları ise CSS bozuklukları(görsellikte kaymalar) olmadan Internet Explorer'da gösterilebilmiştir.

1.3 İş Bölümü

040090501 - Yiğit Bekir KAYA

Bu projenin genel tasarımı ve yönetimi ile ilgilendim. Tüm projede ortak kullanılan sınıfların olduğu main paketini geliştirdim. Projeyi 5 ana (standalone) alt projeye böldüm ve kendim de projenin tümüyle bağlantılı olan profil paketini aldım. Paketimi MVC yapısına uyacak şekilde üç ayrı alt pakete böldüm. Bu paketin Model kısmında yer alan Profile, Reward, Experience ve Message sınıfları benim sorumlu olduğum dört nesnedir.

040090533 - Abdullah Aydeğer

Projede;

- itucs.blg361.main paketindeki FTPManager.java dosyası
- Geliştirme ve Çalıştırma Ortamı Bölümü
- Kurulum Bölümü

040080901 - Anwarshah HASANİ

Dış Bağlantı Yönetimi

- Yarışma
- Etkenlik
- Proje
- İş

itucs.blg361.external paketindeki tüm sınıflar

040080905 - Esam GHALEB

Profile girişi, siteye kayıt işlemleri ve ADMIN giriş yapmayı yapılmıştır. Öğrenciler, akademisyenler ve Ar-Ge şirketlerini ortak bir platformda birleştirmek için ADMIN girişi ile üniversiteleri ve şirketleri ekleme, çıkarma ve güncellemeyi gerçekleştirdi. Bunların hepsi BASE paketinde yer almıştır. Bu paket içinde, üniversiteler ve onların fakülteleri ve bölümleri, ayrıca şirketler için Java sınıfları oluşturulmuştur.

040080252 - Güney GÜNER

Bu projede forum paketinin içerisindeki tüm sınıfların geliştirilmesi bana aitti. Bu paketin içindeki Field, Subject ve Post sınıflarını ve görsel şemaları tasarladım. Sayfaların görünümünün güzelleştirilmesi için CSS şablonu araştırdım. Bulduğum şablonları yetkili arkadaşlara bildirdim.

Chapter 2

Kurulum

Projenin 2 türlü kurulumu mümkündür:

1. Akademik-Ağ projesi Java ile yazıldığından dolayı war dosyası olarak dağıtılabilir. War dosyası tomcat server üzerinde çalıştırabilir.
2. Akademik-Ağ projesinin kodları Mercurial sürüm deposundaki <http://mercurial.intuxication.org/hg/itucs-blg361-2011-g06/> adresinden indirilip, veritabanının internette bulunması sebebiyle veritabanı oluşturmaya gerek duymadan kendi yerinizde çalıştırabilirsiniz. Bunun için Apache Tomcat'in projenin çalıştırılacağı bilgisayarda yüklü olması gerekmektedir. Ayrıca projeye commons-net-3.0.1 kütüphanesi de eklendiği için kodları sürüm deposundan çekecek kişinin bu kütüphaneyi de projeye eklemesi gerekmektedir.

Ayrıca projenin veritabanı Mysql Workbench üzerinde ERD olarak tasarlanmıştır. Bu tasarımdan sonra kod otomatik olarak Workbench tarafından oluşturulmuştur.

Chapter 3

Kullanım Kılavuzu

3.1 Profile Sayfası

Kişi başarılı giriş yaptığı zaman karşısına aşağıdaki gibi profil sayfası çıkacaktır. Bu sayfada diğer sayfalarda (BasePage'den türeyen) olduğu gibi gezinme çubuğu yer alır. Burada ortada menü, sağda profil resmi, gelen mesajlar ve profilden çıkma bağlantısı yer alır.

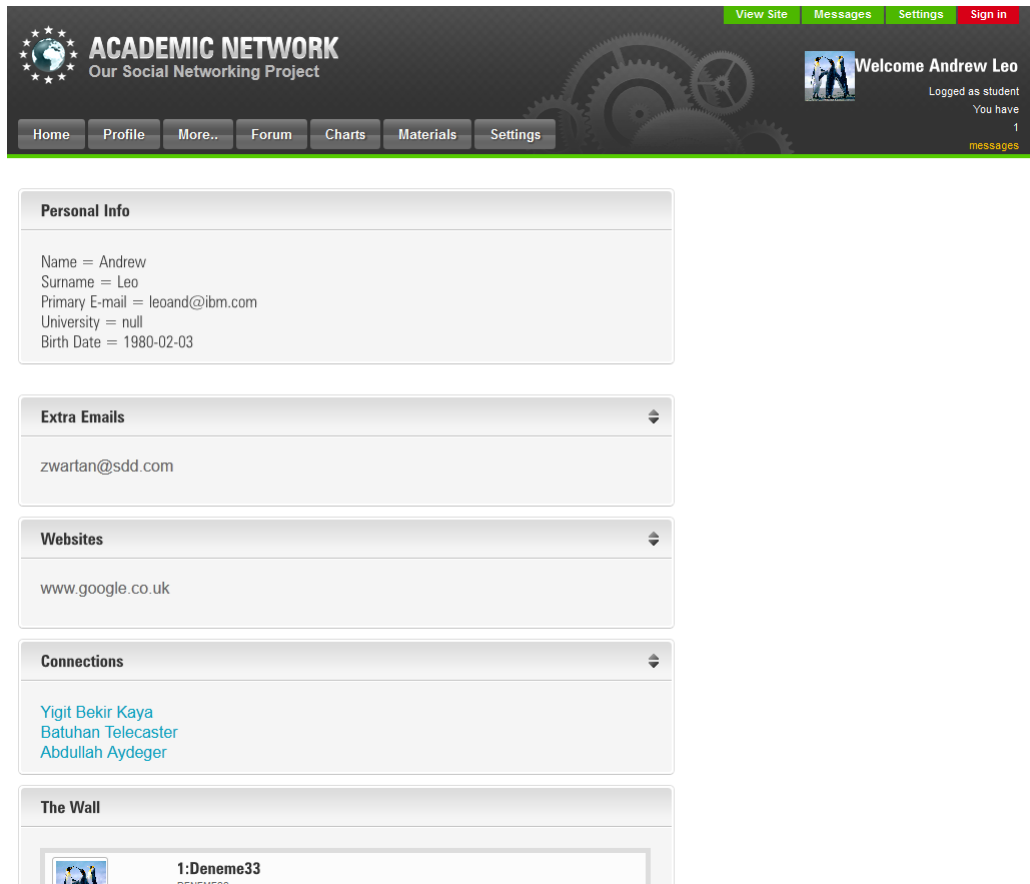


Figure 3.1: Profil Sayfası Genel Bakış

3.2 Profil Menüleri

Bu kısımda profilin bilgileri değiştirilebilir.

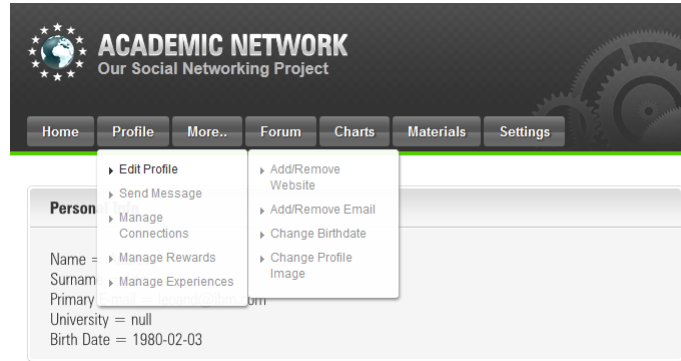


Figure 3.2: Profil Menüleri

3.2.1 Edit Profile

Bu kısımda profilin bilgileri değiştirilebilir.

3.2.1.1 Add/Remove Website

Bu kısımda add ve remove düğmeleriyle web sayfası eklenip, çıkarılabilir.

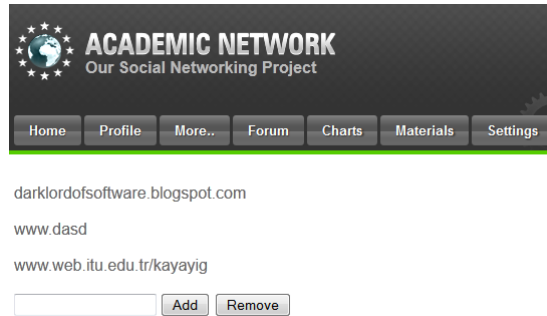


Figure 3.3: Web Sayfası Ekle/Çıkar

3.2.1.2 Add/Remove Email

E-posta ekleme çıkarma da, web sayfası ekleme çıkarma gibidir.



Figure 3.4: E-posta Ekle/Çıkar

3.2.1.3 Change Birthdate

Bu sayfada kişinin doğum tarihi değiştirilebilir.



Figure 3.5: Doğum Günü Değiştir

3.2.1.4 Change Profile Image

Bu sayfada kişinin resmi upload panelde dosya sisteminde yüklenen resim ile değiştirilir.

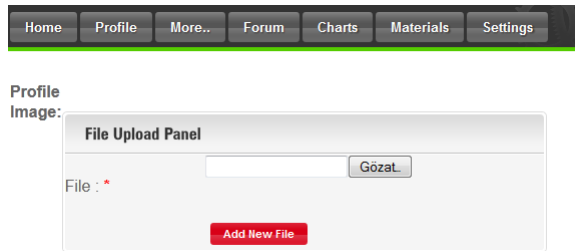


Figure 3.6: Profil Resmi Değiştir

3.2.2 Manage Connections

Bu sayfada bağlantı web sayfası yada ekstra e-post gibi eklenip çıkarılabilir.

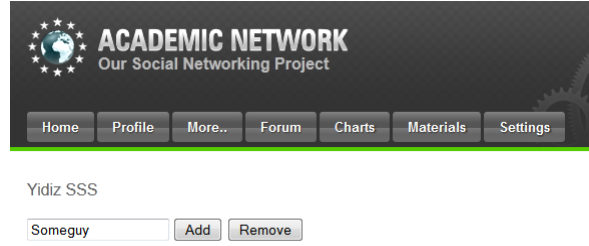


Figure 3.7: Bağlantı Yönetimi

3.2.3 Manage Rewards

Bu sayfada Competiton çubuğuna yarışmanın veritabanındaki competionId'si ve description çubuğuna o yarışmada alınan başarıyla ilgili açıklama yazılıp add düğmesine basarak ödül eklenebilir yada Reward çubuğuna var olan ödülün rewardId'si yazılıp remove düğmesine basarak ödül silinebilir.

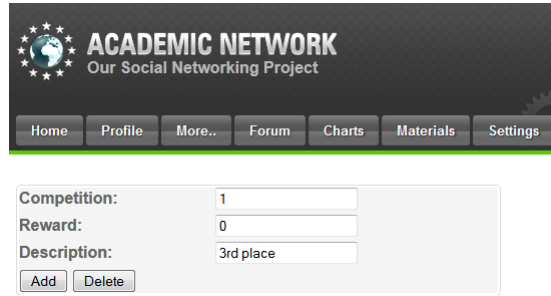


Figure 3.8: Ödül Yönetimi

3.2.4 Manage Experiences

Bu sayfa tamamlanamamıştır ancak bitseydi ilk önce deneyimin türü (akademik yada iş) seçilecek sonra kurum adı yazılacak, başlangıç ve bitiş tarihleri girilecek ve en son olarak açıklama girilip add ile eklenecekti. Aynı şekilde var olan deneyim, deneyimler listesinden seçilip remove ile silinebilecekti.

Chapter 4

Teknik Kılavuz

4.1 Veritabanı Tasarımı

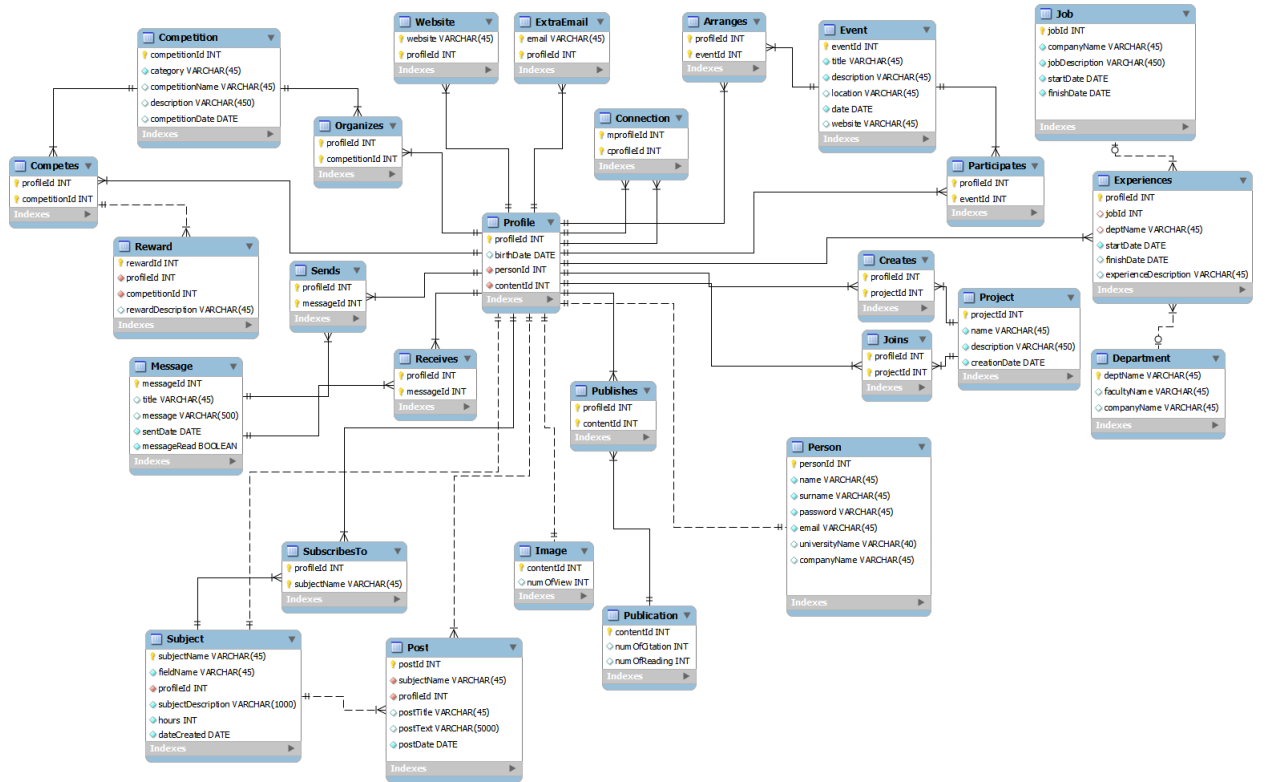


Figure 4.1: Ödül Yönetimi

Yukarıda Profile tablosu ile 1. dereceden bağlantılı tablolar gösterilmiştir. Bir profilin tasarlanan yetenekleri şunlardır:

- Giriş yapıldıktan sonra üretilen Person tablosu üzerinden Profile üretilmesi
- Connection tablosu ile bağlantılı profillerin haritalanması
- Sends/Receives ve Message tablosu ile mesajlaşma sistemi
- Website ve ExtraEmail tabloları ile profile web sayfaları ve ekstra e-posta adresi ekleme/çıkarma işlemleri

- Post tablosu ile Forum girdilerinin duvarda listelenmesi
- SubsribesTo ve Subject tabloları ile (Observer tasarım kalıbı yardımıyla) daha önce girdi yazılmış konularda yeni bir girdi geldiğinde hemen bu girdilerin duvarda listelenmesi
- Department/Job ve Experiences tabloları ile akademik yada iş hayatı deneyimlerinin listelenmesi
- Organizes/Competes ve Competition tabloları ile bir profilin yarışma düzenleyebilmesi veya düzenlenen bir yarışmaya katılabilmesi
- Reward ve Competes tabloları ile profilin belirli yarışmadan elde ettiği ödüllerin listelenmesi
- Arranges/Participates ve Event tabloları ile bir olayı organize etme yada katılabilmesi
- Creates/Joins ve Project tabloları ile bir proje oluşturma veya katılabilmesi
- Publishes ve Publication tabloları ile kişinin yayınlarının listelenmesi

4.2 Yazılım Tasarımı

4.2.1 Nesneye Dayalı Tasarım

Profil paketi Wicket çatısının (framework) da sunduğu yetenekle rahatça MVC tasarım kalıbına uygun olarak gerçekleştirilmiştir. org.blg361.profile paketinin içinde bu kalıbı özellikle netleştirmek için model, view ve control adında üç alt paket oluşturulmuş ve ilgili sınıflar ilgili paketlere konulmuştur. Model paketinde sorumlu olduğum Profile, Message, Experience ve Reward model sınıfları bulunmaktadır. Bu sınıfların amacı veritabanından yapılan sorgulamaları işlemesi en kolay şekilde tutmaktır. Bu sınıflar veri türleri olabilirler, çünkü esas veriyi işleme kontrol ve görüntüleme katmanı tarafından yapılmaktadır. Hibernate benzeri bir SQL soyutlama çatısında reverse engineering yapıldığında elde edilen sonuç ile büyük benzerlik gösterdiği tespit edilmiştir. Kontrol katmanı görüntü katmanından gelen isteklere göre veritabanını ve model sınıflarını işler. Bu sınıflar yönetici (Manager) olarak adlandırılmışlardır. Her modele bir yönetici sınıf düşmektedir.

Wicket görüntü katmanında devreye girmektedir. Projede görüntü katmanı olarak başka framework de kullanılabilir çünkü MVC yapısındaki tüm MC ile V arasında bağımlılık yoktur (loose coupled yada no coupling).

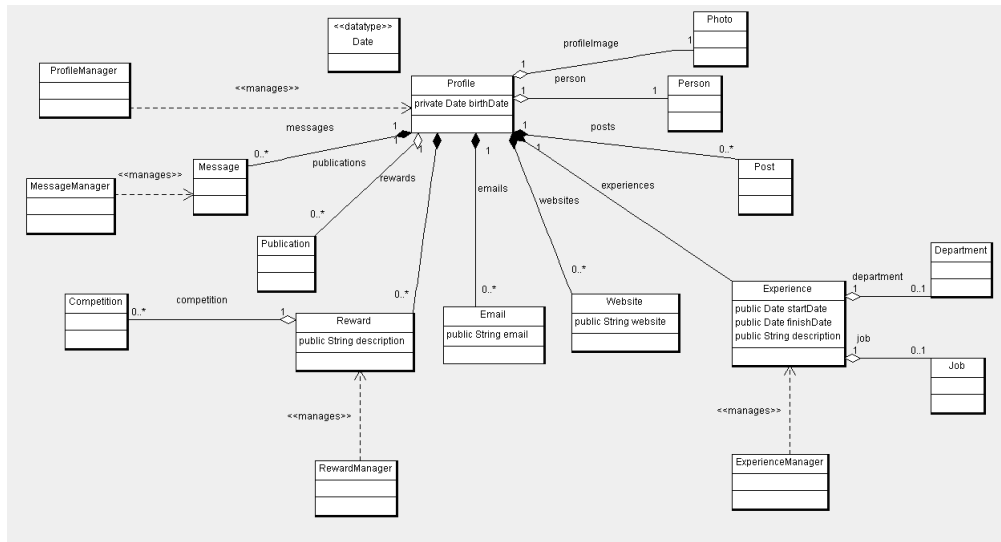


Figure 4.2: Sınıf Diyagramı

4.2.2 Wicket Kullanımı Tasarımı

Kullanıcı profilinin gösterilmesinde Wicket çatısının yetenekli Panel sınıfları özelliklerini kullandım. Sayfada gösterilen her türlü bileşen (component) aslında özelleşmiş bir Wicket Panel sınıfıdır. Bu yüksek derecede modülerliği de sağlamıştır. Bu Panel sınıfları genel ve özel amaçlı olmak üzere iki şekilde de tasarlanmıştır. İki tür Panel sınıfı da oldukça taşınabilir. Yani sadece bir div Wicket takısı ile her türlü sayfaya entegre edilebilir. Bu sayede ufak bir bileşen kütüphanesi de gerçekleşmiştir.

Her panel sınıfının html kısmında CSS ve Javascript yeteneklerinden yararlanılmıştır. Java tarafı ise tamamen Wicket çatısının bileşenleri kullanılmıştır.

HeaderPanel ve FooterPanel de bu tasarıma dahildir. En önemli kısım bu panellerin esas CSS ve JavaScript taşıyıcıları olmalıdır. BasePage'de yer alırlar ve tüm sayfa sınıfları BasePage'den türeyerek bu CSS ve JavaScript'leri miras alırlar.

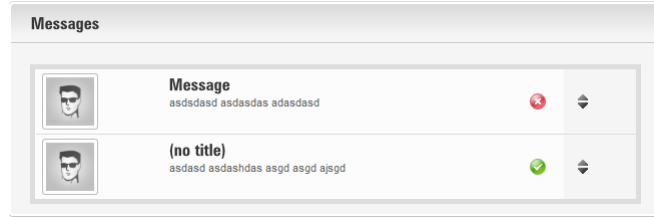


Figure 4.3: Wicket Panel Sınıfları

4.2.3 Oturum (Session) Tasarımı

Klasik yaklaşımda session genellikle çatının elindedir ve erişim bir çeşit Session fabrikası (Session Factory) tarafından sağlanmaktadır. Wicket Java dilinin yeteneklerini kullanma imkanı sunduğu için ben de session kontrolünü ProfileManager sınıfındaki statik, giriş yapmış profil olarak CurrentProfile ve o anda sayfası görülen profil olarak DisplayedProfil olmak üzere iki Profil nesnesinde sağladım. Böylece her sayfada kullanıcı ProfileManager.CurrentProfile yada ProfileManager.DisplayedProfile çağrılarıyla bu nesneleri session olarak kullanabildiler.

4.2.4 Veritabanı Yönetici Tasarımı

Veritabanı bizim projemizde Singleton tasarım kalıbına uymaktadır. Veritabanı bağlantısı bir kere kurulur ve tüm projede bu tek örnek (instance) görünür olmalıdır.

```
public class DatabaseManager {

    public static Connection Connection = null;

    private DatabaseManager() {
    }

    public static void start() {
        String userName = "blg361_g06";
        String password = "019910";
        String url = "jdbc:mysql://www.freesql.org/blg361_acanetw"; //AcademicNetwork

        start(userName, password, url);
    }

    public static void start(String userName, String password, String url) {
        if (Connection != null) {
            return;
        }

        try {
            Class.forName("com.mysql.jdbc.Driver");
```



```
        } catch (ClassNotFoundException ex) {  
            throw new UnsupportedOperationException(ex.getMessage());  
        }  
  
        try {  
  
            Connection = DriverManager.getConnection(url, userName, password);  
  
        } catch (SQLException ex) {  
            Logger.getLogger(DatabaseManager.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    }  
  
    public static void stop() {  
        try {  
            Connection.close();  
        } catch (SQLException ex) {  
            Logger.getLogger(DatabaseManager.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    }  
}
```

Burada bağlantı nesnesi Singleton olarak davranır. DatabaseManager.start() yada DatabaseManager.stop() bir kere çağrılırken DatabaseManager.Connection her veritabanı bağlantısı gerektiren yerde kullanılır.

4.2.5 Model Sınıfları Tasarımı

Bu kısımda bana ait model sınıflarını tanıtacağım.

4.2.5.1 Profile

Projedeki en temel sınıflardan biridir. Veri tabanında küçük bir tablodur fakat ilişkilerinin yoğunluğundan dolayı yönetim sistemi tarafında oldukça büyük bir sınıfa denk gelir.

Profil sınıfının temel bilgileri kayıt olurken elde edilen bilgileri temsil eden Person sınıfına ait bir referansta tutulur. Ayrıca kişinin profil resmi (yada avatarı) resim nesnelerini temsil eden Photo sınıfına ait bir referans ile temsil edilir. Ayrıca kişinin doğum günü tarih biçiminde tutulur

Profil sınıfının büyük kısmı profil sayfasında gösterilecek koleksiyonlardan meydana gelir. Bunlar:

Ekstra e-posta adresleri, web sayfaları, bağlantılar (yada arkadaşlar), mesajlar, girdiler (posts), deneyimler, ödüller ve yayınlar.

4.2.5.2 Message

Mesaj sınıfı iki bağlantılı profil arasındaki mesajlaşmayı temsil eder. Sınıfta kime yollandığına dair profil türünden bir referans vardır. Kimden geldiği bilgisi zaten Session tarafından oluşturulur. Ayrıca mesajın başlığı ve içeriği da yazı olarak tutulur. Mesaj yollandığı zaman yollanma anındaki zaman elde edilir ve gönderme zamanına yazılır. Ayrıca mesajın okunup okunmadığına dair mantıksal bilgi de tutulmaktadır.

4.2.5.3 Experience

Kullanıcının sahip olduğu deneyimleri (şirket veya akademik) temsil eden model sınıfıdır. İş türünden ve departman türünden iki nesne tutar. Bu nesneler aslında union şeklinde bir ilişki içindedir, yani aynı anda sadece ikisinden birinde değer bulunabilir ve diğeri null olur. Hangisinde o anda değer olduğu isJobExperience mantıksal değişkeni ile kontrol edilir.

Bir deneyimin başlangıç ve bitiş tarihleri de tarih biçiminde saklanır. Eğer bir iş daha bitmediyse bitiş tarihi null olur. Bunun kontrolü inProgress mantıksal değişkeni üzerinden yapılabilir.

Son olarak bir deneyime ait açıklamada yazı olarak veritabanında tutulur.

4.2.5.4 Reward

Bu sınıf belirli bir yarışmada kazanılan ödülü temsil eder. Bu sınıfın içinde ödülün alındığı yarışmaya ait bir referans vardır. Ayrıca bu ödülün tanımı hakkında bir yazı türünden nesne de içerir. Kimin ödülü kazandığı bilgisi veritabanında mevcuttur. Bununla Profil nesnesi ilgilenmektedir.

4.2.6 Kontrolör Sınıfların Tasarımı

Bu kısımda bana ait kontrolör sınıflarını tanıtacağım.

4.2.6.1 Profile Manager

Bu sınıf Profil nesnesiyle ilgili tüm veritabanı işlemleri yapan sınıftır. Projedeki en yoğun veritabanı işlemleri burada gerçekleşmiştir çünkü Profile tablosunun veritabanındaki 40 tablodan 20 tanesi ile bağlantısı vardır.

```
public static Profile CurrentProfile = null;  
public static Profile DisplayedProfile = null;
```

Bu değişkenler session nesneleri olarak kullanılırlar.

4.2.6.2 Message Manager

Bu sınıf Message nesnesini yönetmektedir. Bir mesaj hakkında yapılabilecek temel iki işlem vardır. Bunlar mesajın gösterilmesi ve mesajın bir yere gönderilmesidir. Bu sınıf da bu özellikleri gerçekleştirmiştir.

4.2.6.3 Experience Manager

Bu sınıf sadece var olan deneyimleri veritabanından çekmektedir.

4.2.6.4 RewardManager

Bu sınıf kullanıcının belirli bir yarışmada kazanılan ödülleri listeler, ekler ve çıkarır.

4.2.7 Görüntü (View) Sınıfları Tasarımı

Bu bölümde tüm Wicket Java/HTML sayfaları ve Wicket Panelleri tanıtılacaktır.

Metot	Açıklama	SQL Sorgusu
updateCurrentProfile()	Bu metot o anki profili veritabanı ile senkronize eder.	SQL komutu yok
int getProfileId(int)	Bu metot personId'si verilen kişiye ait profilId'sini döner. Bu birebir ilişkiyi kurmak için gereklidir.	select profileId from Profile where personId = %
Profile getBasicProfile(int)	Bu metot profileId'si verilen profilin temel kişisel bilgilerini döner. Temel bilgi olma kriteri diğer sınıflar tarafından kullanım sıklığı göz önünde bulundurularak belirlenmiştir.	select birthDate, personId, contentId from Profile where profileId = %
Profile getBasicProfileByName(String)	Bu metot ismi verilen yazıya benzeyen ilk profili veritabanından çeker.	select profileId, birthDate, personId, contentId from Profile natural join Person where name like '%s'
Profile getProfile(int)	Bu metot verilen profileId'ye sahip profili arkasına gizlediği private erişimli bir dizi SQL tabanlı metod çağırarak suretiyle oluşturur.	select birthDate, personId, contentId from Profile "where profileId = %
createProfile(int)	Bu metot verilen personId'den temel bilgilere sahip profil nesnesi oluşturur ve veritabanına ekler.	insert into Profile (personId, contentId) VALUES (?, ?)
updateBirthDate(int, Date)	Bu metot profileId ile belirtilen profilin doğum tarihini verilen tarih olarak günceller.	update Profile set birthDate = ? where (profileId = ?)
updateProfilePicture(int, int)	Bu metot profileId ile belirtilen profilin resmini contentId ile belirtilen fotoğraf türünden içerik olarak günceller.	update Profile set contentId = ? where (profileId = ?)
addWebsite(int, String)	Bu metot profileId ile belirtilen profile yeni web adresi ekler.	insert into Website (website, profileId) VALUES (?, ?)
removeWebsite(int, String)	Bu metot profileId ile belirtilen profilden website ile belirtilen web adresini siler.	delete from Website where website = ? and profileId = ?
addExtraEmail(int, String)	Bu metot profileId ile belirtilen profile yeni email adresi ekler.	insert into ExtraEmail (email, profileId) VALUES (?, ?)
removeExtraEmail(int, String)	Bu metot profileId ile belirtilen profilden extraEmail ile belirtilen email adresini siler.	delete from ExtraEmail where email = ? and profileId = ?
addConnection(int, Profile)	Bu metot profileId ile belirtilen profile başka bir profili bağlar	insert into Connection (mprofileId, cprofileId) VALUES (?, ?)
removeConnection(int, Profile)	Bu metot profileId ile belirtilen profilin bağlantı listesinden profile ile belirtilen profili siler.	delete from Connection where mprofileId = ? and cprofileId = ?
ArrayList<String> getProfileWebsites(int)	Bu private metot profileId ile belirtilen profilin web sayfalarını String listesi olarak döner.	select website from Website where profileId = %
ArrayList<String> getExtraEmails(int)	Bu private metot profileId ile belirtilen profilin email adreslerini String listesi olarak döner.	select email from ExtraEmail where profileId = %
ArrayList<Profile> getConnections(int)	Bu private metot profileId ile belirtilen profilin bağlantılarını temel bilgiler içeren profil listesi olarak döner.	select cprofileId from 'Connection' where mprofileId = %

Table 4.1: ProfileManager Metotları

Metot	Açıklama	SQL Sorgusu
ArrayList<Message> getMessages(int)	Bu metot profileId ile belirtilen profile gönderilmiş mesajların listesini mesaj listesi olarak döner.	select messageId, title, message, sentDate, messageRead from Receives natural join Message natural join Profile natural join Person where profileId = % select profileId from Sends natural join Profile natural join Person where messageId = %
sendMessage(int, int, String, String)	Bu metot senderProfileId ile belirtilen profilden, receiverProfileId ile belirtile profile içeriği ve başlığı belirtilmiş (belirtilmemişse varsayılan (no title) olarak atanır) mesaj yollar.	insert into Message (title, message, sentDate, messageRead) values (?, ?, ?, ?) insert into Sends (profileId, messageId) values (?, ?) insert into Receives (profileId, messageId) values(?, ?)

Table 4.2: Message Manager Metotları

Metot	Açıklama	SQL Sorgusu
ArrayList<Experience> getExperiences(int)	Bu metot profileId ile belirtilen profilin deneyimlerini deneyim listesi olarak döndürür.	select jobId, deptName, startDate, finishDate, experienceDescription from Experiences where profileId = %

Table 4.3: ExperienceManager Metotları

Metot	Açıklama	SQL Sorgusu
ArrayList<Reward> getRewards(int)	Bu metot profileId ile belirtilen profilin deneyimlerini deneyim listesi olarak döndürür.	select rewardId, competitionId, rewardDescription from Reward natural join Competes where profileId = %
addReward(int, int, String)	Bu metot profileId ile belirtilen profile competitionId ile belirtilmiş yarışmada alınan ödülü desc ile belirtilmiş açıklama ile ekler.	insert into Reward (profileId, competitionId, rewardDescription) values (?, ?, ?)
removeReward(int)	Bu metot rewardId ile belirtilen yarışmayı veri tabanından siler.	delete from Reward where rewardId = ?

Table 4.4: RewardManager Metotları

Sınıf	Açıklama	Kullanım (HTML)	Kullanım (Java)
AddRemoveConnection	Bağlantı ekleme çıkarma formu içeren sayfa.	Link	Link
AddRemoveExperience	Deneyim ekleme çıkarma formu içeren sayfa.	Link	Link
AddRemoveExtraEmail	Ekstra Email ekleme çıkarma formu içeren sayfa.	Link	Link
AddRemoveReward	Ödül ekleme çıkarma formu içeren sayfa.	Link	Link
AddRemoveWebsite	Web sayfası ekleme çıkarma formu içeren sayfa.	Link	Link
ChangeBirthdate	Doğum tarihi güncelleme formu içeren sayfa.	Link	Link
ChangeProfilePicture	Profil resmi (avatar) değiştirme formu içeren sayfa.	Link	Link
SendMessagePage	Geçerli profil üzerinden bağlantılara mesaj atma formu içeren sayfa.	Link	SendMessagePage(message)
ConnectionsPanel	Bağlantıların gösterildiği panel sınıfı.	Div	ConnectionsPanel(wicketId, connections)
InformationPanel	Belirtilen bilgi mesajını mavi bildiri penceresinde gösteren panel sınıfı.	Div	InformationPanel(wicketId, title, message)
MessagePanel	Kişiye gelen mesajların gösterildiği panel.	Div	MessagePanel(wicketId, messages)
MultiLineStringPanel	Genel amaçlı çok satırlı yazı göstermek için kullanılan panel.	Div	MultiLineStringPanel(wicketId, header, strings)
ProfileInfoPanel	Profil temel bilgilerini statik olarak gösterildiği panel	Div	ProfileInfoPanel(wicketId, profile)
ProfilePostPanel	Profilde yer alan girdilerin gösterildiği panel.	Div	ProfilePostPanel(wicketId, posts)
RewardPanel	Ödüllerin gösterildiği panel.	Div	RewardPanel(wicketId, rewards)
DisplayProfile	Tüm panellerin gösterildiği ana profil sayfası.	Link	DisplayProfile(profile)

Table 4.5: Görüntü Sınıfları