

Google Web Toolkit ile Öğretim Elemanı Otomasyon Sistemi

Mehmet Karakoç¹, Orçun Dayıbaş², Doç. Dr. Melih Günay¹

mehmetkarakoc@akdeniz.edu.tr, odayibas@onami.com.tr, mgunay@akdeniz.edu.tr

¹ Akdeniz Üniversitesi – Bilgisayar Bilimleri Araştırma ve Uygulama Merkezi (BAUM), Antalya

² Onami Ltd., Antalya

Özet: Bu bildiri, GWT ve GWT-Bootstrap ile geliştirilen *Öğretim Elemanı Bilgi Yönetim Sistemi* ele alınmaktadır. Bu sistem geliştirilirken, bu araçlar ve bileşenleri kullanılarak kullanıcı arayüzleri tasarlanmış ve JavaScript bilgisi gerekmeksizin tüm kodlama Java programlama dili ile yapılmıştır. Bu bildiri ile amaçlanan, önemli bir özgür web teknolojisi olan GWT ile web tabanlı ve mobil uyumlu, esnek ve verimli bir bilgi yönetim sisteminin gerçekleştirilmesidir.

Anahtar Kelimeler: Özgür Web Teknolojileri, Google Web Toolkit (GWT), GWT-Bootstrap, Java, JavaScript, Bilgi Yönetim Sistemi.

Abstract: In this paper, *Information Management System for the Faculties* developed with GWT and GWT-Bootstrap is addressed. While developing this system, user interfaces were designed using these tools and their components; and all of coding was done with Java programming language without JavaScript knowledge. Aim of this paper is to develop a web based and responsive, flexible and efficient information management system with an important web technology GWT.

Keywords: Open Web Technologies, Google Web Toolkit (GWT), GWT-Bootstrap, Java, JavaScript, Information Management System.

1. Giriş

Özgür web teknolojilerinden Google Web Toolkit (GWT), tarayıcı tabanlı uygulamalar geliştirmek ve bu uygulamaları eniyilemek için kullanılan açık kaynaklı ve ücretsiz bir geliştirme ortamıdır. GWT, web uygulamaları geliştirmede genel bir çatı olmakla beraber; GWT ile geliştirilen uygulamalara aynı zamanda mobil platformlardan da erişilebilir. Dünya çapında binlerce geliştirici tarafından kullanılan GWT, geliştiricilere uygulamalarının istemci taraflarını Java ile geliştirme olanağı sağlar [1]. GWT ile yazılan uygulamalar tarayıcılar arası uyumludur ve her tarayıcı için uygun ve optimize edilmiş JavaScript (JS) kodu üretilir. Böylece, tarayıcıya özel kodlama gerekmemekte ve modülerlik doğrudan Java ile sağlanabilmektedir. GWT derleyicisi Java'da yazılmış tüm uygulama kodunu JS'ye derler. GWT, büyük ölçekli ve yüksek performanslı sürdürülebilir uygulamalar geliştirmede ve bakımlarını kolay bir şekilde yapabilmeye son derece iyi bir seçenektir.

GWT'nin kullanıcı arayüzü kütüphanesi, pek çok arayüz bileşenini içermekte olup; GWT uygulamaları, tüm belli başlı tarayıcılarda, Android/iOS tabanlı telefon ve tabletlerde çalışabilir [1]. GWT'nin gücü Java ile yazıp (Java geliştiricileri için düşük ölçüde öğrenme çabası) JS ile çalıştırma imkânından gelir.

GWT, geliştiricinin tarayıcı detayları ve AJAX gibi konularda uzman olmasını gerektirmeksizin; verimli web uygulamaları geliştirilmesine olanak tanımayı amaçlar [2]. Açık kaynaklı bir yazılım geliştirme çatısı olarak GWT, dinamik web uygulamaları ve bu alanda benzer çatıların geliştirilmesi için sağlam bir temel sağlar.

Java ile Gmail, Google Maps ve Google Calendar tarzında ve hızlı AJAX uygulamaları oluşturulabilir. Geliştiriciler, JS, HTML ve CSS kullanımı seçimlik olacak şekilde; masaüstü benzeri uygulamalar geliştirip zengin kullanıcı arayüzleri (dinamik web sayfaları) tasarlayabilirler. GWT ile *Zengin İnternet Uygulamaları* geliştirilebilir.

Sürüklenebilir, yeniden boyutlandırılabilir esnek pencere ve paneller tasarlanabileceği gibi; GWT bileşenleri (*Widget*) diğer çatılarla geliştirilmiş eski uygulamalarda da kullanılabilir. Bu bileşenler (basit ya da karma) piksel yönelimli grafikler yerine dinamik olarak yaratılan HTML kullanılarak oluşturulurlar [3]. Böylece, arayüzü oluşturan tablolar sorgulardan dönen listelerle hızlı bir biçimde doldurulabilir.

Çalışma kapsamında, ilk olarak pek çok web tabanlı uygulama geliştirme çatısı incelenmiş ve daha sonra öğretim elemanlarına yönelik bilgi yönetim sistemini geliştirmek için GWT kullanılmıştır. Bu çalışmanın en önemli katkısı, modern bir web teknolojisi olan GWT'nin GWT-Bootstrap ile birlikte kullanılarak, web tabanlı ve mobil uyumlu bir projenin geliştirilmesidir. GWT bileşenlerinin Bootstrap karşılıkları bulunduğu takdirde arayüzlerin bu şekilde tasarlandığı, diğer durumlarda ise GWT bileşenlerinin kullanıldığı bir yaklaşım izlenmiştir.

GWT ile yazılan uygulamada, Java kaynak koduna doğrudan JS dâhil etme, Java kodunu JS koduna çağırma (ya da tersi); geliştirme modunda Java *debugger* ile Java kaynağı ve script debugger ile JS için hata ayıklama mümkündür [4].

Bu çalışmada, liste görünümlü ekranlar için *CellTable* bileşeni ağırlıklı olarak kullanılmış, tablodaki kolon ve hücreleri düzenlemek için ise tablo hücre tipleri için tanımlı sınıflardan özel sınıflar türetilerek ve gerekli durumlarda HTML de kodlayarak görsellik zenginleştirilmiştir. Kullanıcı arayüzlerini daha esnek ve görsel bir hâle getirebilmek için ayrıca GWT bileşenleri yerine GWT-Bootstrap muadilleri de kullanılmış ve arayüz tasarımında CSS kullanımı tercih edilmiştir. Çalışmanın geri kalan kısmı şu şekilde düzenlenmiştir: 2. bölümde öğretim elemanlarına yönelik olarak geliştirilen bilgi yönetim sistemi anlatılmakta, 3. bölümde deneysel çalışma

kapsamındaki detaylar verilmekte, 4. bölümde ise sonuç kısmı yer almaktadır.

2. Öğretim Elemanı Bilgi Yönetim Sistemi

Bilgi yönetim sisteminin gerçekleşmesi aşamasında, altyapı bileşeni olarak, ilerleyen kısımlarda ayrıntıları verilen çeşitli metodoloji ve yaklaşımlardan yararlanılmıştır.

UiBinder

Kullanıcı arayüzü ve iş mantığı (*Business Logic*) kodlarını birbirinden ayırık tutarak modülerliği destekleyen bu çözümde, her görünüm (*view*) iki parçadan oluşur:

- UserDetailsView.ui.xml: Tanım tabanlı kullanıcı arayüzü (Şekil 1).
- UserDetailsView.java: İlgili kullanıcı arayüzüne ait mantıksal işlevler (Şekil 2).

```
<!DOCTYPE ui:UiBinder SYSTEM "http://dl.google.com/gwt/DTD/xhtml.ent">
<ui:UiBinder xmlns:ui="urn:ui:com.google.gwt.ui.binder"
  xmlns:g="urn:import:com.google.gwt.user.client.ui">
  <ui:style>
    .important {
      font-weight: bold;
    }
  </ui:style>
  <g:HTMLPanel>
    Hello,
    <g:Button styleName="{style.important}" ui:field="button" />
  </g:HTMLPanel>
</ui:UiBinder>
```

Şekil 1: Örnek Kullanıcı Arayüzü.

```
public class UserDetailsView extends Composite {
  private static UserDetailsViewUiBinder uiBinder = GWT
    .create(UserDetailsViewUiBinder.class);

  interface UserDetailsViewUiBinder extends UiBinder<Widget, UserDetailsView> {
  }

  public UserDetailsView() {
    initWith(uiBinder.createAndBindUi(this));
  }

  @UiField
  Button button;

  public UserDetailsView(String firstName) {
    initWith(uiBinder.createAndBindUi(this));
    button.setText(firstName);
  }

  @UiHandler("button")
  void onClick(ClickEvent e) {
    Window.alert("Hello!");
  }
}
```

Şekil 2: Arayüz için Tanımlı Mantıksal İşlevler.

Bu çalışmada UiBinder, arayüz ve Java metodlarını birleştirmek amacıyla kullanılmıştır. Ayrıntılı bilgi için [5] adresinden yararlanılabilir. Java dosyasına karşılık gelen XML dosyasında stilleri saklamak, düzenlemek ve kolayca bulabilmek mümkündür. Bu durum, arayüzün kolayca yönetilmesine ve diğer noktalarda otomatik olarak kullanımına imkân tanır. Böylelikle CSS stil isimlerinde yaşanan sorunlar oldukça azalır.

UiBinder bileşenler için bildirimsel (*declarative*) bir arayüz oluşturma imkânı sunar [6]. GWT'nin temel yararlarından biri, arayüz için hazırlanan kısımların alt parçalara ve farklı bileşenlere ayrılabilmesi ve *UiBinder* ile basit bir şekilde daha sonra bir araya toplanabilmesidir. Çalışma kapsamında, ders ve öğrenci detaylarını görme ve öğrenci arama gibi işlemler için ayrı arayüzler tasarlanmış ve bu arayüzler tasarlanan ekranların (genel liste ve detay görünümü) birer parçası olarak bir araya getirilmişlerdir.

Kullanılmayan CSS stillerinin çıkarılması, uygun yerlerde birleştirilmesi, sınıf isimlerinin, CSS dosyalarında yer alan koşul ve sabitlerin basitçe yönetilebilmesi mümkün olup; UiBinder kullanılarak, yüzlerce satırlık koddaki gezinim son derece kolaylaşmakta ve görsel stiller kolaylıkla yönetilebilmektedir. Ana uygulamanın mantığı farklı bileşenler ve birleşimleri ile sağlanabilir.

UiBinder, yazılım geliştiriciler arasındaki işbirliğini kolaylaştıran bir yaklaşımdır. Çalışma kapsamında, kullanıcı arayüzleri XML dosyası içinde ve iş mantığı kısmından ayrı olarak tasarlanmıştır. Sayfa yerleşimleri/düzeni (*layout*) programlanmadığı için (XML, HTML ve CSS kullanımı) işlevselliği etkilemeden değiştirmek kolaydır. Zengin kullanıcı arayüzü kütüphanesi, sürükle bırak özellikli ayırıcılardan diyalog menülerine; buton, metin kutusu, etiket ve uyarı kutucuklarından panel ve içeriksel menülere kadar pek çok bileşen içerir.

GWT-Bootstrap

Bu araç, Twitter Bootstrap bileşenlerini içeren GWT arayüz kütüphanesidir. Basit ve esnek bileşenler, stiller ve eklentiler içerir ve daha hızlı ve görsel GWT uygulamaları geliştirmek için kullanılır. Geliştirilen web tabanlı GWT uygulamasının mobil cihazlar aracılığıyla da esnek bir biçimde kullanılabilmesi için bu araçtan yararlanılmıştır, çünkü GWT bileşenlerinin mobil

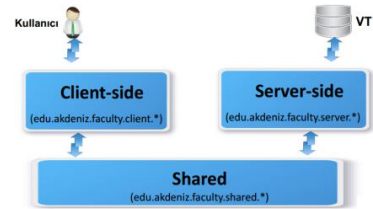
platformlarda performansı yeterince iyi değildir. Ayrıntılı bilgi için [7]'deki bağlantı kullanılabilir.

Google Guava EventBus

Bileşenler arası veri iletişimini kolaylaştırmak için bu teknolojiye yararlanılmıştır. Doğrudan modülerliği destekleyen ve kod satır sayısını azaltan bu teknoloji, katmanlı yapıdaki projenin alt metodlarının olaylar üzerinden izlenebilmesini ve test işlemlerinin kolay bir biçimde yapılabilmesini sağlar. Böylece, istemci taraf ile kullanıcı arayüzü arasında doğrudan bağlantı kurulmadan, tüm işlemler bu ara yol üzerinden gerçekleştirilmiştir. GWT'de *EventBus* kullanım örneğine [8] adresinden erişilebilir.

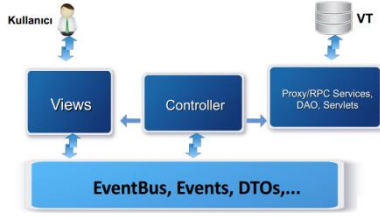
MyBatis

Bu teknoloji, veri tabanı katmanını ve yapısını uygulamanın diğer katmanlarından ayırarak tutabilmek için bir çözüm olup *iBatis* tabanlıdır (geliştiricileri ortaktır). Diğer ORM (*Object-Relational Mapping*: nesne ilişkisel eşleme/haritalama) çerçevelerinden farklı olarak, Java nesnelere tablolarla değil yöntemler sorgulara eşlenir. Çalışmada Java nesnelere, ara objeler için tanımlanan ve veri tabanındaki ilgili tablo alanlarını içeren sınıflardır. Sorgulamalar ise her sorguya ilgili nesne üzerinden verilen sorgu belirleyicisi ile (örneğin *Course* nesnesi için *searchCourses*) gerçekleştirilir. Ayrıntılı bilgi için [9] bağlantısına başvurulabilir. Geliştirilen sistem, istemci ve sunucu taraf ve ortak kısım olmak üzere Şekil 3'te verilmiştir.



Şekil 3: Projenin Genel Yapısı.

Genel akış mantığı ise Şekil 4'te görülmektedir.



Şekil 4: Katmanlar Arasındaki Akış.

Çalışmaya ait uygulamayı oluşturan katmanlar ve genel içerikleri Tablo 1’de verilmiştir.

Tablo 1: GWT Uygulama Yapısı.

İstemci (client/)	Grafiksel kullanıcı arayüzü kodu, servis ve proxy tanımlamaları
Sunucu (server/)	Servislerin gerçekleştirimi (istemci-sunucu çağrılarını için kodlamalar)
Paylaşılan (shared/)	Ortak kod

Tanımlanan kontrolör ile kullanıcı arayüzü ve veri tabanı arasında doğrudan bağlantı kurulmasının önüne geçilmiştir. *EventBus* ve tanımlanan olaylar aracılığıyla, genel yapı *MainController* ile ve kullanıcı girişi *LoginView* ile yönetilir; veri tabanı işlemleri ise *FacultyService* ile yapılır. *EventBus* doğrudan veri tabanı ile konuşulmamaktadır.

Çalışmadaki önemli noktalar şunlardır:

- HTML ve CSS destekli kodlama ile özelleştirilebilir ekran tasarımı
- Bootstrap destekli bileşenler sayesinde mobil uyumluluk
- MyBatis ile Java nesnelere ve sorgular arasındaki ilişkinin otomatik olarak kurularak farklı katmanlardan yönetilebilmesi
- MyBatis ve EventBus kullanımı ile istemci, servis ya da sunucu tarafında çalışan geliştiricilerin sorgu yordamlarına rahatça erişebilmeleri

GWT’nin temel yararları şunlardır:

- Statik içerik için HTML, görsel tasarım için CSS (dinamik olarak biçimlendirme ile stil ekleme/çıkarma) ve dinamik içerik için JS [10]

- Hataları bulmada ve kodu geliştirmede kolaylık, kodun yeniden kullanımı ve yönetimi (*refactoring*)
- Düşük bakım maliyeti, HTML5 desteği
- Web projelerinde nesne yönelimli programlama
- Pek çok güçlü çatı, oldukça büyük topluluk, JS kütüphanelerini kullanabilme
- Web sayfalarını oluşturan içeriği (metin, görüntü vb.) kolayca yönetebilme
- Herhangi bir modern web tarayıcısının parçası olan JS dışında eklenti gerektirmeme
- Kodun hataları kolayca ayıklanabilir (Java hata ayıklayıcıları JS muadillerine göre genel olarak daha gelişmiştir).
- Büyük projeler/ekipler için uygun olup derleme zamanı kontrolleri yapılabilir.
- Nihai JS kodunda ana akış içerisinde çağırılmayan kısımlar çıkarılır.
- Tarayıcı geçmiş yönetimi (*Browser history management*) [11]
- Statik tip kontrolü (*Static type checking*) hataları azaltırken verimliliği artırır [11].
- Olay sistemi, kaynak yönetimi
- GWT, Java standart sınıf kütüphanesindeki en yaygın kullanılan sınıfların JS gerçekleştirmelerini ve tüm genel HTML/JS bileşenlerini içerir.
- Kullanıcı arayüzü güncellemeleri için sunucudan yeni HTML isteği gerekmez [3].
- Geliştirme modunda GWT kodu derlemesiz test edilebilir (Kullanıcı arayüzü değişikliği için sayfanın yenilenmesi yeterlidir).
- Lokal desteği ile tarih/saat/sayı türündeki veriler kolayca formatlanabilir [12].

- Herhangi bir yerde kullanılabildiği kadar herhangi bir kaynağın yüklenmesi beklenebilir.
- Web servisleri ile bütünleştirme
- Ayrıca GWT uygulamaları harici bir sunucuya konuşlandırılabilir (*deployment*).

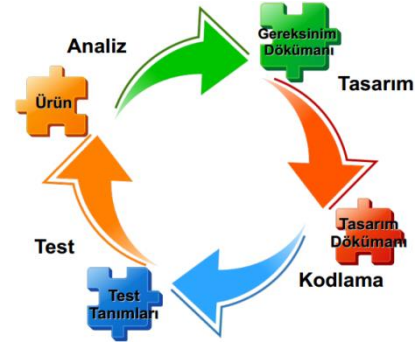
Web sayfaları arasındaki etkileşimler için HTML5 standartları kullanılabilir; fakat masaüstü yazılımın pek çok karakteristiğine sahip büyük web uygulamaları geliştirmede GWT son derece iyi bir seçenektir [6]. Bu çalışma ile bir öğretim elemanı aktif dönem için tüm derslerini yönetebilmekte, ders ve öğrenci detaylarını görebilmekte ve sınav işlemlerini yapabilmektedir. Not girişi için Excel'den yükleme seçeneği de vardır. Çalışma kapsamında geliştirilen uygulama için gerçekleştirim adımları şunlardır:

- Geliştirme süresince Ubuntu üzerinde *Mozilla Firefox* web tarayıcısı ile çalışılması
- *Eclipse* geliştirme ortamında Java dili ile uygulamanın yazılması ve hata giderimi
 - Kullanıcı arayüzü tasarımı için HTML ve CSS tabanlı bir yapı (görsellik) kullanılır.
 - XML ile sorgular hazırlanır (Veri tabanı yönetim sistemi MS SQL Server ve yapı olarak değişime açık).
 - ORM olarak Spring çatısı ile bütünleştirme
 - Tüm sorgulama metotları başarılı ve başarısız durumlarda yanıt verebilecek şekilde yapılandırılır.
 - Derleme gerektirmeyen yapılandırmalar için özellik dosyaları kullanılır.
 - Eclipse için Google eklentisi ile test süreci kolayca yönetilir.
- GWT'nin JS derleyicisinin uygulamayı JS ve HTML dosyalarına derlemesi

- Java kodu optimize edilir, kullanılmayan kod blokları çıkarılır.
- JS'ye dönüştürülen kod tekrar optimize edilir.

- Uygulamanın herhangi bir web sunucusu ile servis edilmesi
- Uygulamanın çeşitli tarayıcılarda çalışıp çalışmadığının sınanması
 - Birim testler (*unit test*) ile kullanıcı arayüzlerinin tasarlanan gibi çalışıp çalışmadığı doğrulanabilir.

Çalışma kapsamındaki yazılım geliştirme süreçleri genel olarak Şekil 5'te verilmiştir.



Şekil 5: Yazılım Geliştirme Yaşam Döngüsü.

Geliştirme ortamı kapsamında, proje derleme, raporlama ve belgeleme için yazılım proje yönetim aracı olarak *Apache Maven* kullanılmıştır. Kaynak kod, web sayfaları ve belgeleme ile ilgili dosyalar için mevcut ve geçmiş sürümlerin tutulması ve yönetilebilmesi (ayrıca bakım işlemi) amacıyla da sürüm/revizyon kontrol sistemi olarak *SVN (Apache Subversion)* kullanılmaktadır.

Proje yaşam döngüsünü oluşturan araçlar ve teknolojiler Tablo 2'de verilmiştir.

Tablo 2: Proje Yaşam Döngüsü.

Proje Yönetim Aracı	Trac
Sürekli Entegrasyon Sunucusu	Jenkins
Geliştirici Referans Kaynakları	JavaDoc
Test Uygulama Sunucusu	Tomcat

Modül (AUFaculty) 'M' olarak gösterilmek üzere;

1. {M}.gwt.xml: Modül XML tanım dosyası olup; proje bağımlılıkları, giriş noktası,

RPC (*Remote Procedure Call*) sunucu uygulama girişini içerir. GWT uygulaması için modüller tanımlanabilir.

2. {M}.java (*client*): Giriş noktası (EntryPoint arayüzü) olup *Main* metodu (onModuleLoad()) gibidir. Bileşenler kök panele eklenir.
3. {M}.html (*public*): Web uygulamasının baş sayfası olarak istemciye fiilen gönderilir.

Proje, her gece 23:00'te derlenecek, Java dokümantasyonu oluşturulup yayınlanacak ve derlemenin başarılı olduğu durumda otomatik olarak uygulama sunucusuna konuşlandırılacak şekilde yapılandırılmıştır.

3. Deneysel Çalışma

Sorgu metotlarının asenkron çağruları ve kullanıcı arayüzlerinin tasarımları istemci tarafında (tarayıcı içerisinde çalışan web uygulaması) yapılırken, iş mantığı kısmı (sorgulamalar) sunucu tarafında gerçekleşir. Uygulamayı hızlandırmak amacıyla, kodlama bölünerek gerçekleştirilmiş, çok fazla sunucu çağrısı yapılmamış ve uygun durumlarda sorgulamalar küçük alt sorgulara bölünerek gerçekleştirilmiştir. Tasarlanan esnek kullanıcı arayüzü bileşenleri, farklı ekranlar için Java tabanlı nesne yönelimli bir yapı ile yeniden kullanılmıştır.

Ağ üzerinden sunucu ile etkileşimli çağrı mekanizmasında (istemci-sunucu arası veri takası), yeni HTML sayfaları getirmek yerine sunucudan veri getirmek söz konusudur (sayfa yerine sunucudaki verinin yenilenmesi yeterli). Doğru bir şekilde kullanıldığında, uzak sunucu çağruları (RPC), tüm kullanıcı arayüzü mantığını istemciye taşıma fırsatı verir ve bu durum büyük ölçüde iyileştirilmiş performans, daha düşük bant genişliği, indirgenmiş web sunucusu yüklemesi ve daha akıcı kullanıcı deneyimi sonuçları sağlar [3].

GWT uygulamaları iki mod içerir:

1. Geliştirme esnasında (*Hosted mode*), uygulama Java Virtual Machine içinde Java *bytecode* olarak çalışır [13]. Bu noktada, Java hata ayıklama olanaklarından yararlanılabilir.
2. *Web modunda* ise Java'dan JS'ye derleme (GWT derleyicisi) ile uygulama salt JS (son kullanıcılar için istem) ve HTML olarak çalışır (konuşlandırma).

Çalışma kapsamında geliştirilen yazılım, hem geliştirme modunda hem de ürün modunda ayrı ayrı sınanmıştır. Bu iki mod bire-bir aynı olmamakla beraber; bazı hatalar sadece geliştirme modunda, bazı hatalar ise sadece ürün modunda yakalanabilmiştir. Tespit edilen hataların giderilip yazılımın iyileştirilmesi ile her iki mod için de sorunsuz çalışan kararlı bir yazılım ortaya çıkmıştır. Bu süreci ifade eden döngü şu şekildedir:

- Her iki mod için de kararlı olana kadar yinele:
 - Geliştirme modunda bilinen hata kalmayana kadar devam et
 - Yazılımı bu modda test et
 - Bilinen hatalarını gider
 - Ürün modunda yazılımı gerçek verilerle test et
 - Bu moddaki hataları tespit et
 - Geliştirme modunda dikkate alınacak hata yoksa çık

Maven projesi olarak yapılan derlemeler yaklaşık 2 dakika sürmektedir. Her ne kadar geliştirme modunda derleme işlemleri uzun sürse de ürün modu geliştirme moduna göre belirgin ölçüde hızlıdır. Çalışma ile ilgili örnek ekran görüntüleri Şekil 6'da ve Şekil 7'de verilmiştir. Ayrıca, araştırılan bazı uygulama çatıları (*frameworks*) ve GWT ile ilgili çeşitli karşılaştırmalar için [14, 15, 16] bağlantılarına başvurulabilir.

4. Sonuç

Bu çalışma ile GWT ve GWT-Bootstrap kullanılarak öğretim elemanlarına yönelik etkili bir bilgi yönetim sistemi geliştirilmiştir. Web

katmanındaki programlamanın Java'ya taşınması sayesinde, GWT ile iş mantığı yazılırken web sayfaları da yazılmış; bu yapı geliştiriciler için iş mantığı ile bu kısmın kolayca örtüştürülmesini, kullanıcılar için ise sistemin son derece verimli bir biçimde kullanılabilmesini sağlamıştır. Tüm veri tabanı sorguları servis arkasında ayrı metotlara karşılık gelmekte, temel Java pratiklerinin ve tasarım örüntülerinin kullanımı ile veri tabanı sorgulamaları esnek bir şekilde yönetilebilmektedir. Böylece verimli bir web yazılımı ortaya çıkmıştır. Java nesneleri kullanılarak istemci-sunucu arasında iletişim kurulabilmekte ve bu durum istemciye bir Java uygulamasını (*applet*) yapabileceğinden çok daha basit olmaktadır [3]. Bu işlevselliğin sunduğu yararlar şu şekilde listelenebilir:

- Sunucudan istemciye istisnai durumların kolayca yakalanabilmesi
- Kullanıcı arayüzleri tasarımı ve etkileşimi üzerinde odaklanabilme imkânı
- Daha iyi kullanıcı deneyimi sonuçları [17]

Yazım hatası ve tip uyumsuzluğu (eşleşmeme) gibi yaygın JS hataları, derleme zamanında kolayca yakalanır [13]. Fakat JS genellikle HTML'den daha yavaş olduğundan ve Java kodu GWT ile otomatik olarak JS'ye derlendiğinden; bu çalışma kapsamında özel olarak JS kodlanmamış, CSS stillerinden büyük ölçüde yararlanılmıştır. Öte yandan, programlama kısmının HTML'den fazla JS'ye, UiBinder'in ise JS'den fazla HTML'ye derlenmesi [18], arayüzlerin verimliliğini artıran ve daha hızlı olmalarını sağlayan diğer bir noktadır.

GWT, temel web teknolojileri bilgisinden çok daha öte, pek çok disiplini içeren ve kapsamlı uygulamalar geliştirmeye olanak sağlayan önemli bir teknolojidir. Bu çalışmada, GWT'nin tüm bu avantajlarının Bootstrap'in sağladığı esneklik ve görselleştirme ile hem kullanıcı etkileşimli hem de oldukça verimli bir yönetim sistemi geliştirilmiştir.

Kaynaklar

1. "Google Web Toolkit Tutorial", http://www.tutorialspoint.com/gwt/gwt_tutorial.pdf [11.12.2014].
2. "GWT Name Use Policy", <http://www.gwtproject.org/GWTPolicy.html> [14.11.2014].
3. Sri Lakshmi Ramya Sreedharan, "Google Web Toolkit", [jmvidal.cse.sc.edu/...](http://jmvidal.cse.sc.edu/) [02.01.2015].
4. "JavaScript: JSNI", <http://www.gwtproject.org/doc/latest/DevGuideCodingBasicsJSNI.html> [14.11.2014].
5. "UiBinder", <http://www.gwtproject.org/doc/latest/DevGuideUiBinder.html> [27.11.2014].
6. "Why Google Web Toolkit (GWT)?", <http://bear-z.com/gwt/why-google-web-toolkit-gwt/> [15.01.2015].
7. "GWT-Bootstrap", <http://gwtbootstrap.github.io/> [14.11.2014].
8. "Event Bus: GWT Tutorial (Google Web Toolkit)", <http://www.youtube.com/watch?v=2939gdrNyng> [14.11.2014].
9. "SQL Mapping Framework for Java", <https://code.google.com/p/mybatis/> [14.11.2014].
10. "Java Web Application Frameworks", <http://www.syger.it/Topics/JavaWebApplicationFrameworks.html> [28.01.2015].
11. Gufran Mohammed, "Google Web Toolkit", cs.calstatela.edu/wiki/images/4/41/Gwt.ppt [20.11.2014].
12. San Antonio JUG - March 2010, "Google Web Toolkit", https://java.net/downloads/satjug/Introduction_to_GWT.pdf [10.10.2014].
13. "Google Web Toolkit", http://en.wikipedia.org/wiki/Google_Web_Toolkit [12.12.2014].

14. "A Web framework comparison",
<http://blog.websitesframeworks.com/2013/03/web-framework-comparison-matt-raible-opinion-138/> [27.01.2015].
15. "Web frameworks in Java",
<https://speakerdeck.com/michaelisvy/web-frameworks-in-java-10-years-of-history>
 [28.01.2015].
16. "Top 4 Java Web Frameworks Revealed",
<http://zeroturnaround.com/rebellabs/top-4-java->

web-frameworks-revealed-real-life-usage-data-of-spring-mvc-vaadin-gwt-and-jsf/
 [28.01.2015].

17. Mark Volkmann, "Google Web Toolkit (GWT)",
<http://java.ociweb.com/mark/other-presentations/GWT.pdf> [02.12.2014].
18. "Cute and Cuddly UIs with GWT",
<http://Java.ociweb.com/javasig/knowledgebase/2010-09/GWT.ppt> [03.11.2014].

BYS BETA Bilgi Yönetim Sistemi

Aktif Dönem: 2014-2015 Güz Yarıyılı Program: İnşaat Mühendisliği (Örgün Öğretim) Ders Kodu: 101 Ders Adı: Temel Bilgisayar ve Algoritma Şube No: 1

Haftalık Programım MelihGunay Çıkış

GENEL TERCİHLER

Akademik Yılı - Dönem - Ders - Şube -

DERLERSİM (Şube Listesi)

Ders ID	Ders Kodu	Ders Adı	Ders-Şube ID	Şube Adı	Kur.-Uyg.	Değerlendirme	Değerlendirildi	Kontenjan	Öğrenci Sayısı
17502	İNM101	Temel Bilgisayar ve Algoritma	260567	İnşaat Mühendisliği 1 Nolu Şube	2 - 2	Mutlak	Mutlak	55	54
17502	İNM101	Temel Bilgisayar ve Algoritma	260577	İnşaat Mühendisliği 2 Nolu Şube	2 - 2	Mutlak	Mutlak	55	51

Ders Kataloğu Aktif Ders-Şube

DERLERSİM İZLEMLERİ

Ders Programı Şube Öğrencileri

SINAV İZLEMLERİ

Sınavlanım Sınav Sonuçları Not Girişi Excel'den Not Yükleme

İSTATİSTİKLER

Şube İstatistikleri

LINKLER

AJ Ana Sayfa İletişim EBYS E-Posta

Şube Tipi: Ders

Öğrenci Kayıt Tipi: Program Öğrencileri

Ders Durumu: Açık

Başlangıç Tarihi: 15-09-2014

Bitiş Tarihi: 02-01-2015

Program Dışı Öğrenci: 0

Değerlendirilen Öğrenci: 54

Standart Sapma: 17,25

Aritmetik Ortalama: 58,7

Standart Hata: 2,54

Seçilen Değerlendirme Tipi: Mutlak

Hesaplanan Değerlendirme Şekli: Mutlak

Şekil 6: Öğretim Elemanı Girişi Sonrası Yüklenen İlk Arayüz.

BYS BETA Bilgi Yönetim Sistemi

Aktif Dönem: 2014-2015 Güz Yarıyılı Program: İnşaat Mühendisliği (Örgün Öğretim) Ders Kodu: 101 Ders Adı: Temel Bilgisayar ve Algoritma Şube No: 2

Haftalık Programım MelihGunay Çıkış

GENEL TERCİHLER

Akademik Yılı - Dönem - Ders - Şube -

DERLERSİM İZLEMLERİ

Ders Programı Şube Öğrencileri

SINAV İZLEMLERİ

Sınavlanım Sınav Sonuçları Not Girişi Excel'den Not Yükleme

İSTATİSTİKLER

Şube İstatistikleri

LINKLER

AJ Ana Sayfa İletişim EBYS E-Posta

Sınav Türü

Sınav Türü	Sınav Süresi (dk)	Sınav Tarihi	Sınav Saati	Sınav Yeri	Sınav İlan Tarihi	Etki Yüzdesi (%)	Bilgi	İşlem
Ara Sınav(lar)	50	13-11-2014	16:30 - 17:20		Tarih Girilmemiş!	25	Ort: 52,14 Std: 21,528	Tutlamak Aİ
Ders Devam	50	31-12-2014			16-01-2015	5	Ort: 93,14 Std: 24,293	Tutlamak Aİ
Yıl / Yarıyıl Sonu Sınavı	50	09-01-2015			16-01-2015	40	Ort: 48,48 Std: 17,34	Tutlamak Aİ
Uygulama (Lab. Atölye, Arazi, PDC)	50	31-12-2014			16-01-2015	10	Ort: 64,59 Std: 40,921	Tutlamak Aİ
Ödev(ler) / Seminer(ler)	50	31-12-2014			16-01-2015	5	Ort: 70,82 Std: 31,204	Tutlamak Aİ
Ödev(ler) / Seminer(ler)	50	31-12-2014			16-01-2015	5	Ort: 0 Std: 0	Tutlamak Aİ
Ödev(ler) / Seminer(ler)	50	31-12-2014			16-01-2015	5	Ort: 0 Std: 0	Tutlamak Aİ
Uygulama (Lab. Atölye, Arazi, PDC)	50	31-12-2014			16-01-2015	5	Ort: 77,29 Std: 38,454	Tutlamak Aİ

Değerlendirme Ölçütü Dağılımı

Değerlendirme Sistemi: Bağlı Değerlendirme, Mutlak Değerlendirme

Kaydet

• "Sınav Tarihi" Girilmemiş Sınavlarınız için "Not Girişi" Yapamazsınız!

• Henüz "Not Girişi" Yapılmamış bir Sınavınız! Silebilmek için "Etki Yüzdesi" Değerini "0" Olarak Belirlemelisiniz.

Şekil 7: Öğretim Elemanının Aktif Ders-Şubesi için Sınavlarım Ekranı.