



**T.C.**

**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**

**MÜHENDİSKLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**SOCİALUNİ BİLECİK**

**Efecan Altay**

**Proje II Raporu**

**Proje Danışmanı : Yusuf Muştı**

**Bilecik**

**01.01.2017**

## ÖZET

Haberleşmenin mektuplardan Digital ortamlara geçmesiyle birlikte sosyal platformlu uygulamaların artmasına yol açmıştır. Üniversitelerin haberleşme sistemleri üniversitedeki öğrencilerin haberleşmesine yardımcı olmak,birbirlerini tanımamış fakat tanırsa birlikte cevherler yaratacak arkadaşlıkların bir arada birbirini tanıma imkanını yaratmak amaçlanmıştır.

Uygulamanın Haber kanalı sistemi sayesinde Yazı,Etkinlik ve Fotoğraf paylaşımı sağlanmıştır. Mesajlaşma sistemi Arkadaş ekleme gibi özelliklere sahip bir uygulama platformu oluşturulmuştur.

## ABSTRACT

When the correspondent switches from letters to digital media Has led to increased social platform applications. The communication systems of the universities are aimed at helping the students in the university to communicate, to create the opportunity to know each other in a way that friendships that they did not know each other but know each other together.

Thanks to the application's News channel system Writing, Activity and Photo sharing are provided. The messaging system has features like adding friends An application platform has been established.

Sosyal,Arkadaşlık,Etkinik,Paylaşım

## **Teşekkürler**

Bu projenin başından sonuna kadar hazırlanmasında emeği bulunan ve beni bu konuya yönlendiren saygıdeğer hocam ve danışmanım Sayın Öğr.Gör.Yusuf Muştı'ya tüm katkılarından ve hiç eksiltmediği desteğinden dolayı teşekkür ederim.

## İçindekiler

<b>ÖZET</b>	<b>2</b>
<b>ABSTRACT</b>	<b>2</b>
<b>Giriş</b>	<b>6</b>
<b>SocialUni Nedir?</b>	<b>7</b>
<b>Proje Hedefi</b>	<b>7</b>
<b>PROJEDE KULLANILAN MODÜL VE KAVRAMLAR</b>	<b>7</b>
<b>1.Apache Cordova</b>	<b>7</b>
1.1.Semantic UI	8
1.2.AngularJS	8
<b>2.NodeJS Sunucusu</b>	<b>8</b>
2.1.API Nedir ?	9
2.2.Npm : Node Package Manager	9
2.3.Express Server	9
2.4.Mongodb Modülü	10
<b>3.MongoDB Veritabanı</b>	<b>10</b>
3.1MongoDB Nasıl Kurulur ?	10
3.3 MongoDB kullanımı	11
3.4.Veritabanını Yönetmek	11
3.5.NodeJS ile MongoDB Veritabanına Bağlanmak	12
<b>UYGULAMA KATMANLARI</b>	<b>13</b>
<b>FRONTEND : SOCIALUNİ ARAYÜZLERİ</b>	<b>14</b>
<b>BACKEND : NODEJS EXPRESS SERVER</b>	<b>15</b>
<b>SOCİAL UNİ SUNUCUSU API DÖKÜMANI</b>	<b>16</b>
<b>1.Kayıt İşlemleri</b>	<b>16</b>
1.1.Kayıtlı kullanıcı kontrol işlemi	16
1.2.Kayıt İşlemi	16
<b>2.Giriş ve Çıkış İşlemi</b>	<b>18</b>
2.1.Giriş ve Logid(token) Kaydı	18
2.2.Çıkış İşlemi	19
<b>3.Profil Bilgileri Alma İşlemi</b>	<b>19</b>
<b>4.Arkadaş Bilgileri Alma İşlemi</b>	<b>19</b>
<b>6.Haberleri Okuma İşlemi</b>	<b>20</b>
<b>7.Haber Oluşturma İşlemi</b>	<b>20</b>
<b>8.Yorum Okuma İşlemi</b>	<b>21</b>

9.Haber Beğenme/ Vazgeçme İşlemi	21
<b>GENYMOTION TESTLERİ</b>	<b>22</b>
<b>ÇIKARILAN SONUÇLAR</b>	<b>23</b>
<b>Kodlar</b>	<b>23</b>
Arayüz Cordova Kodları	23
<a href="https://github.com/EfecanAltay/SocialUni-Bilecik">https://github.com/EfecanAltay/SocialUni-Bilecik</a>	23
Sunucu Kodları	23
<a href="https://bitbucket.org/EfecanAltay/cordovamobiltest/src/81dac9702ee7d9798e297ec31664a7332c9b6790">https://bitbucket.org/EfecanAltay/cordovamobiltest/src/81dac9702ee7d9798e297ec31664a7332c9b6790</a>	23
<b>Kaynaklar</b>	<b>24</b>



## Giriş

Social Uni Bilecik Projesi,

**Hedef Olarak,**

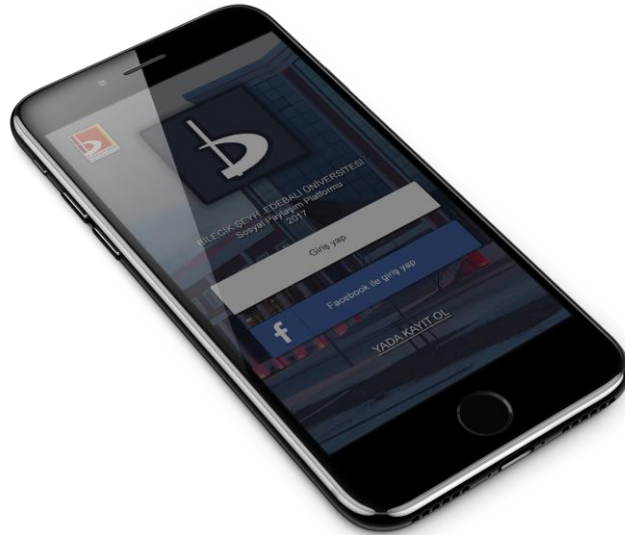
Aynı üniversitede farklı dallarda eğitim gören öğrenciler arasındaki iletişimi sağlayıp, birçok projede ortak çalışma imkanı sağlamaktır.

**Teknik Olarak,**

Cordova projesi geliştirme merakıyla başlayarak, frontend ve backend programlama merakıyla devam etmiştir.

Haberleşme verisi olarak json data kullanılmıştır.

Server tarafında Sqlden farklılık olması amacıyla açık kaynaklı nosql(MongoDB) veritabanı üzerine geliştirilmiş bir sistemdir.



## **SocialUni Nedir?**

Social Uni, Üniversite öğrencileri için geliştirilmiş sosyal platformdur. Bu platformda sosyal arkadaşlıklar, haberleşme ve paylaşımlar yer almaktadır.

Bilindiği üzere üniversitelerin öğrencilerinin birbirlerinden haberdar olabilme imkanı, birbirini tanıma imkanı sadece sınıf ortamları, katıldığı kulüplerle sağlanırken SocialUni platformuna kayıtlı tüm öğrenciler, bulundukları bölüm dalları farketmeksizin birbirlerini tanıyıp hangi alanda kendilerini geliştirdiğini tanımada olanak sağlayacaktır.

## **Proje Hedefi**

Temel olarak hedef aynı üniversitede farklı dallarda eğitim gören öğrenciler arasındaki iletişimi sağlayıp, birçok projede ortak çalışma imkanı sağlamaktır.

## **PROJEDE KULLANILAN MODÜL VE KAVRAMLAR**

### **1. Apache Cordova**

Apache Cordova Uygulama geliştiricileri için düşünülmüş ve bir kez kod yazarak bütün cihazlar için uygulama geliştirme düşüncesiyle geliştirdiği Cordova çekirdeğini açık kaynak olarak sunan sistemdir.

Cordova ile yazılan uygulama bir cihazlarda web arayüzünde çalışır. Cordova çekirdeği hemen hemen cihazın sahip olduğu sensörlere ve bilgilere erişmemize imkan sağlar. Bu sayede cihazın sensörlerini kullanarak uygulama yazabiliriz.

Cordova çekirdeği yazılan uygulamayı derleyip çalıştırır ve istediğimiz cihaz için app çıkışı yapar. Bu appi cihaza kurduğumuzda web arayüzde uygulamamız çalışır.

Cordova'yı yüklemek için,

```
$ npm install -g cordova
```

Proje Oluşturmak için,

```
$ cordova create projeAdı
```

Projeye Android projesi eklemek için,

```
$ cordova platform add android
```

Derleyip çalıştırmak için,

```
$ cordova run android
```

Cordovada android'te internete Çıkabilmek için ,

```
$ cordova plugin add cordova-plugin-whitelist
```

Modulunu krumamız gerekiyor çünkü internet üzerinden çektiğimiz scriptler yüklenmezler vede sunucumuza bağlanamayız.

## **1.1.Semantic UI**

Web sitelerinde kullanılan elemanlar daha güzel,şık tasarımlarda göstermek için açık kaynak grafik kütüphanesi sunmaktadır.Kullanımı çok basit olup birden çok tasarım yer almaktadır.

## **1.2.AngularJS**

Web site tasarımlarına fonksiyonel işlemler sağlamak için geliştirilmiş kütüphanedir.Javascript dili üzerinde geliştirilmiştir.Kullanımı kolay ve sadedir.

## **2.NodeJS Sunucusu**

NodeJS javascript dili ile geliştirilen sağlam bir sunucu oluşturmamıza olanak sağlayan kütüphanedir.Nodejs ile birlikte npm paket yönetim sistemide birlikte gelmektedir.npm sayesinde birçok yazılmış sistemi tekrar yazmaya gerek duymadan uygulamamıza kurup kullanabiliriz.



## 2.1.API Nedir ?



Application Programming Interfaces,

Anlamı Uygulama Programlama Arayüzleridir. Sunucu ile haberleşme yolundan biridir. Api yolu sayesinde cihazların veritabanına veya sunuculara güvenli erişim sağlanmaktadır.

## 2.2.Npm : Node Package Manager

Npm nodejs modülünün paket yönetim sistemidir. Birçok paketler bu sistem sayesinde kurulur. Nodejs kurulum ile beraber kurulur.

## 2.3.Express Server

Express Server, nodejs'nin modülü olup npm üzerinden yüklenen bir pakettir. Express server Sunucu üzerinde yönlendirme (routerlama) sağladığımız sistemdir. Sunucuya gelen istekleri kolayca geri döndürmeyi sağlayan modüldür. Express modülü sayesinde NodeJs'de yönlendirme (router) yapmaya gerek duymayıp hızlıca gelen istekleri geri döndürücek yönlendirme (router) modelleri oluşturabiliriz.

Kurulumu:

```
$ npm install express --save
```

şeklindedir.

### 2.3.1.Express Generator

Express Generator , hazır basit express sunucu projesi oluşturmayı sağlayan modüldür.Bu modülün çalışması için express modülünün kurulu olması gerekir.Express generator ile kurulan projede gelen verinin adreslemesi “/router” klasörünün içinde modül şeklinde tutulur ve işlemler bu modüllerde gerçekleşir.

Kurulumu:

```
$ npm install express-generator -g
```

Proje Oluşturmak için :

```
$ express -c styl “projeAdı”
```

Şeklinde projemizi oluşturabiliriz.

Eğer uygulamamızda bir web server tarzında geri dönüşü tasarım dökümanı(html) gibi olmasını istiyorsak.Jade tasarım dökümanını bilip kullanmamız gerekir.Eğer jade’den anlamıyorsak, html doküman geri dönüş için

<https://github.com/EfecanAltay/DefaultNodeExpressServer> adresinden proje kopyalanıp tasarım olarak html görünümde geri dönüş veren server kurulabilir.

Social Uni Projesinde Express sunucusu api olarak çalışacağı için POST’la alınan verilerin geri dönüşü sağlanacağından tasarımsal bir dönüş kullanılmamıştır.

### 2.4.Mongodb Modülü

MongoDb Modülü NodeJS Sunucusu için MongoDb veritabanına bağlanmayı sağlayan kütüphanedir.Npm üzerinden kurulabilir.

Kurulumu :

```
$ npm install mongodb --save
```

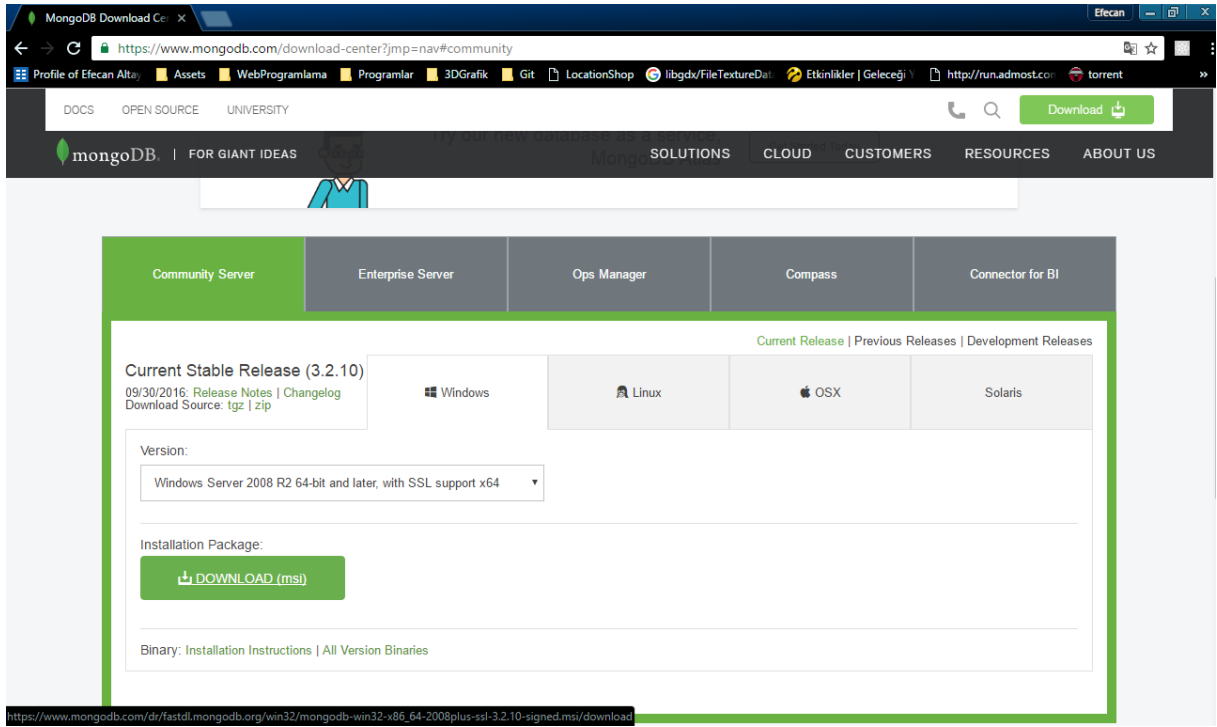
## 3.MongoDB Veritabanı

MonoDB veritabanı NOSQL ve kolay kurulup ,kolay kullanılan veritabanıdır.Big data verilerini en iyi saklandığı sistemdir.Verileri json datalar ile tutan ve haberleşmesinde json data kullanan veritabanıdır.

### 3.1MongoDB Nasıl Kurulur ?

Windows için;

MongoDB nin resmi sitesinden indirme işlemi gerçekleştirilebilir ve sistemimize kurulum yapılabilir.



Kurulum işlemi tamamlandıktan sonra mongoDB nin yüklendiği dizin bizim için önemlidir. Çünkü o dizinden cmd ile serverımızı başlatıp oluşturacağız.

### 3.3 MongoDB kullanımı

MongoDB kullanımı için yüklediğimiz konum önemli demiştik. (Not : Default ayarlarda kurulumumuzu gerçekleştirmişsek window kurulumunu C://Program Files/MongoDB olarak konumlandırmış olabilirsiniz.)Şimdi cmd ile serverimizi başlatalım;

➔ C://Programfiles/MongoDB/Server/3.2/Bin>mongod

Yukarıdaki komut veritabanı serverımızı başlatır.

**Önemli : Eğer ilk defa çalıştırıyorsak hata verecektir. bizden C:// konumunda C://data/db olarak klasörler isteyecektir.Bunun sebebi veritabanı bilgilerini kendi saklayacağı tiplerle dosyalar da tutar.ve bizden bunları saklamak için konum ister.bu klasörlerin C://data/db olarak oluşturması hatayı çözücek ve veritabanınız bir sonraki çalıştırmada çalışacaktır.**

### 3.4.Veritabanını Yönetmek

Veritabanını Yönetmek için hazırda olan shelli kullanabiliriz.

➔ C://Programfiles/MongoDB/Server/3.2/Bin>mongo

Koduyla shell e girip,

➔ show dbs

Koduyla veritabanlarını listeleyebiliriz .vb.

➔ Help

Koduyla komut listesine ulaşabilirsiniz.

### **3.5.NodeJS ile MongoDB Veritabanına Bağlanmak**

NodeJS ile veritabanına bağlanmak için projemize mongodb modülünü eklememiz gerekiyor.

Bunu modülü yüklemek için npm üzerinden

Projemize npm init etmediyse ;

➔ npm init

ardından

➔ npm install mongodb --save

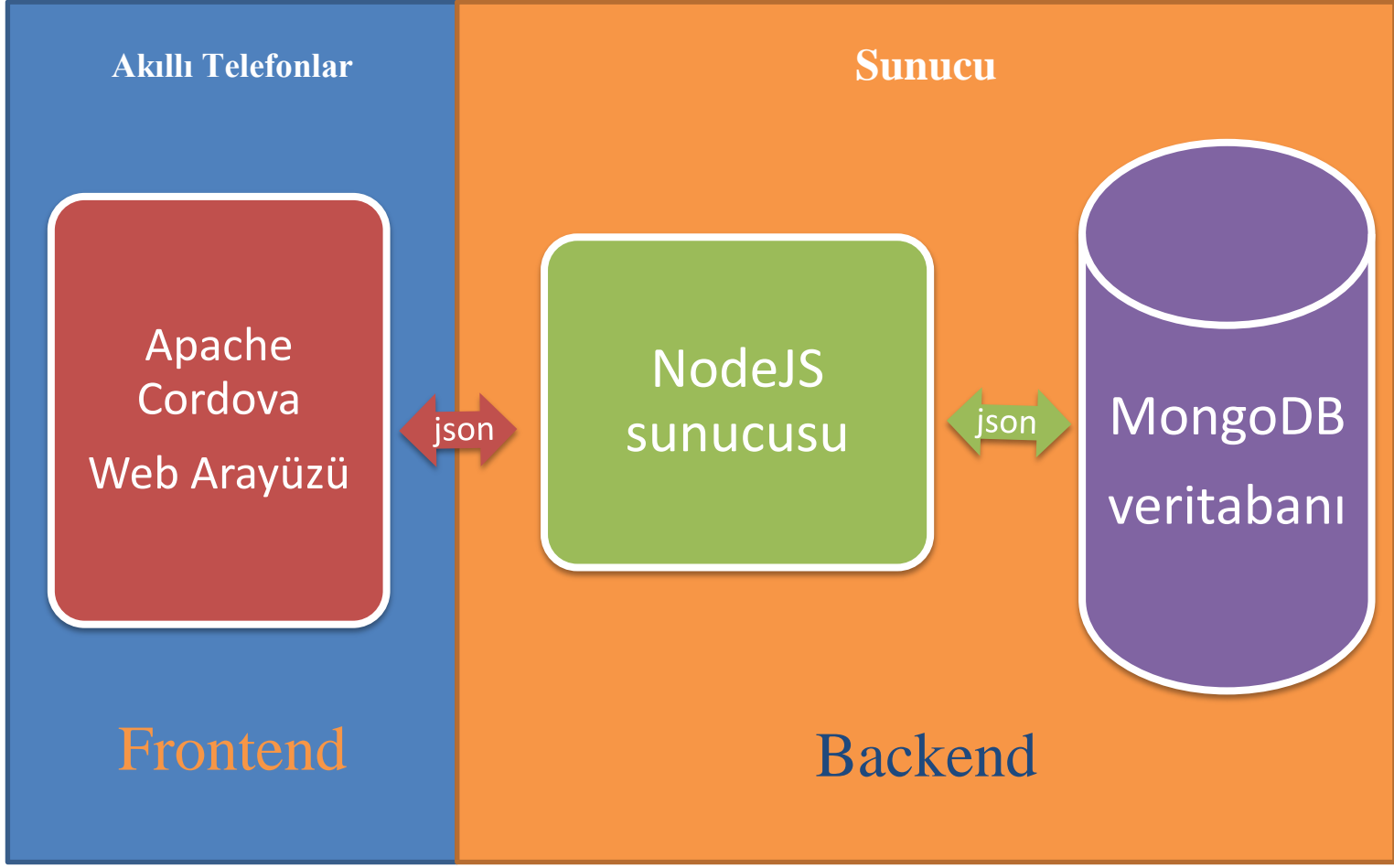
komutunu çalıştırabiliriz. Bu sayede yazacağımız projeye

**import** mongodb

olarak ekleyip kullanabiliriz.

Kodlara Ek-1 den Ulaşabilirsiniz.

## UYGULAMA KATMANLARI

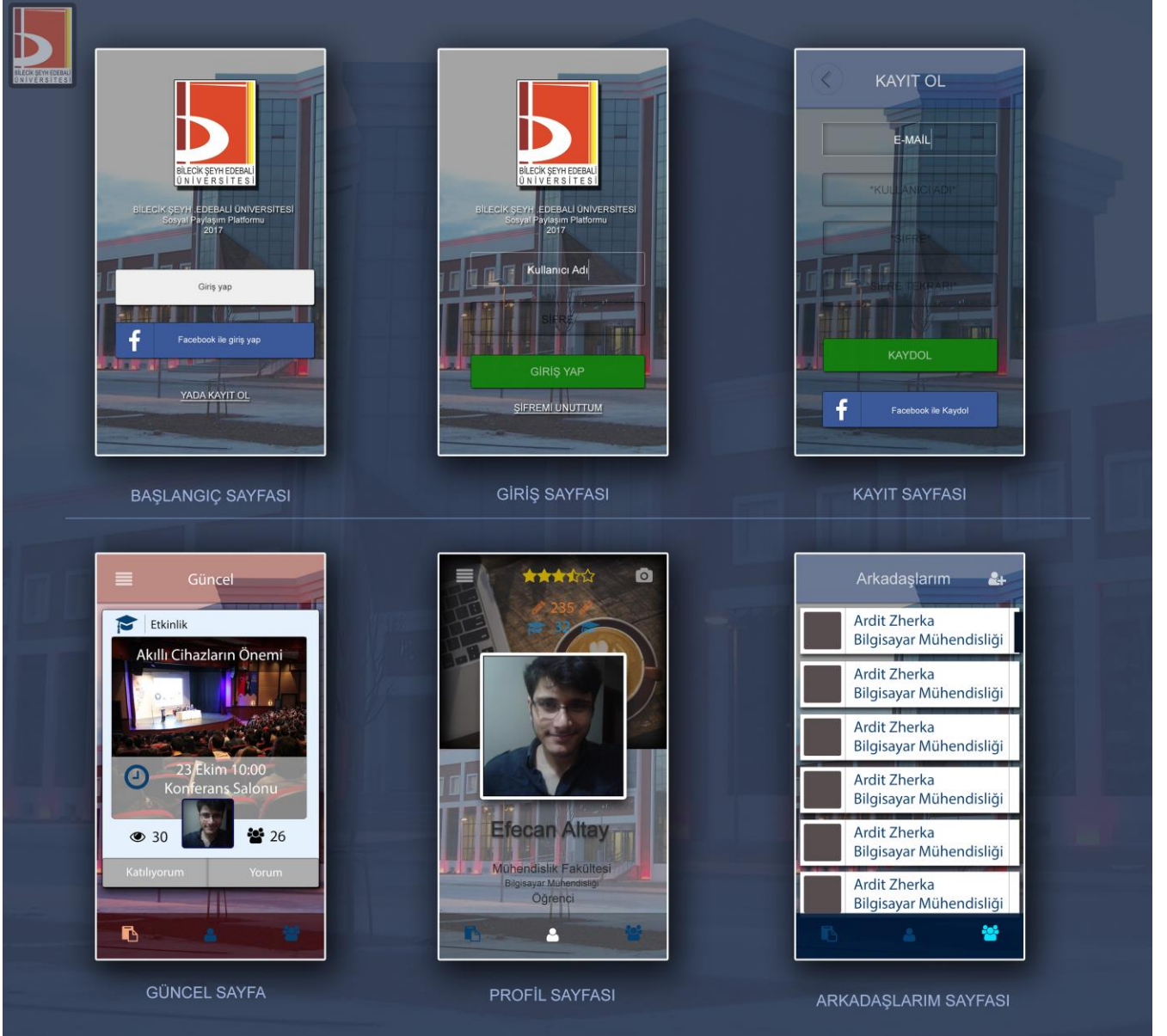


Şekil 1: SocialUni Çalışma Katmanları

Projenin çalışma planı olarak tasarladığım sistem Şekil 1 de görülmektedir. Uygulama arayüzü (Cordova Web Arayüzü) json data ile haberleşmesini sağlayıp, NodeJS Sunucusu ile iletişim kurmaktadır. NodeJS Sunucusu aynı zamanda bir Express sunucudur. (Express modülünü kullandım.) Express sunucusu API görevi görür.

## FRONTEND : SOCIALUNİ ARAYÜZLERİ

Proje planlama aşamasında arayüzler photoshop programı ile tasarlandı ve proje tasarım dökümanı taslakları olarak saklandı.

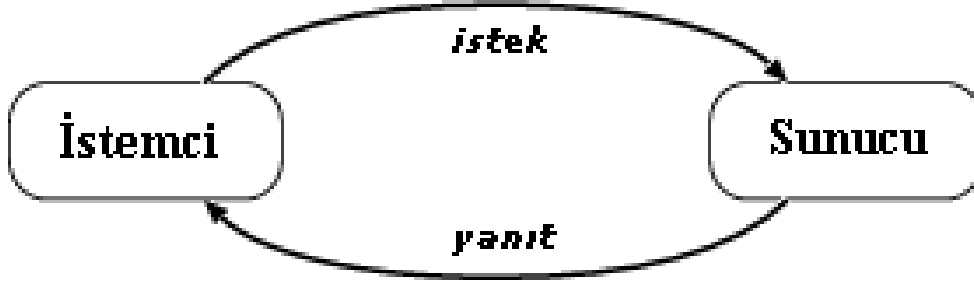


Proje Kodlama aşamasına kadar üzerinde değişiklikler yapıp yeni tasarımlar ile birlikte arayüz kodlama aşamasına geçildi. Her sayfa ayrı ayrı olarak html,css kodları oluşturuldu. AngularJS ile tasarımlara kontroller sağlandı. Tasarım kodları ekler kısmında yer almaktadır.

## BACKEND : NODEJS EXPRESS SERVER

İstemci Sunucu İlişkisinde,

İstemçiden gelen veri(istek) sunucuya bir adres yoluyla ulaşır.Sunucunun bu adresten aldığı veri sunucun işlem yapıp geri gelen adrese geri veri(yanıt) göndermesiyle son bulur.Sunucu ve istemci iletişimi bu sayede gerçekleşmiş olur.



NodeJS Javascript dili basit sunucu oluşturmaya sağlayan bir modüldür.Javascript bilindiği üzere hemen hemen her cihazda çalıştığı gibi her cihaz üzerinde bir sunucu kurmak mümkün olmaktadır.Sunucularda gelen veriye göre bir veri hesapanıp veya işlem yapıp geri veri döndürülmesi sağlanır.

NodeJS de sunucuya gelen adresleri ayıklamak bir hayli uzun bir iştir.bunun kolaylaştırılmış hali Express modülünü kullanmaktır.

Express modülünde adresleme hazır olarak yazılmıştır.bizim yapmamız gereken sadece aynı kalıplardan farklı adresleyiciler oluşturmaktır.

# SOCİAL UNİ SUNUCUSU API DÖKÜMANI

Veri tipleri JSON Formatındadır.Sunucu adresine gönderilen isteklere cevap yanıtları aşağıda listelenmiştir.

## 1.Kayıt İşlemleri

### 1.1.Kayıtlı kullanıcı kontrol işlemi

>Post /isRegistered

Giriş Parametresi:

Adı	Tipi	Açıklaması
kadi	String	Kullanıcı adı
mail	String	Mail adresi

Çıkış Parametresi:

Adı	Tipi	Açıklaması
statusMail	String	Mail Kullanılıyor mu? Sorusunun geri dönüşü
statusKad	String	Kullanıcı adıKullanılıyor mu? Sorusunun geri dönüşü

### 1.2.Kayıt İşlemi

>Post /registered

Giriş Parametresi:

Adı	Tipi	Açıklaması
kadi	String	Kullanıcı Adı Giriş için kullanılan
pass	String	Kullanıcı şifresi
mail	String	Kullanıcı Mail Adresi
name	String	Kullanıcı görünen Adı
surname	String	Kullanıcının Soyadı



Fakulte	int	Fakulte Kodu [ 1 ]
bolum	int	Bolum Kodu [ 2 ]

Çıkış Parametresi:

Adı	Tipi	Açıklaması
status	boolean	Kayıt başarılı olduğunda geri dönüş değeri
message	string	Hata veya Başarı mesajı

**Fakülte Kodları:**

[ 1 ] Fakülte No	Fakülte Adı	[ 2 ] Bölüm No	Bölüm adı
0	Mühendislik Fakültesi	0	Bilgisayar Mühendisliği
		1	Elektrik Elektronik Mühendisliği
		2	İnşaat Mühendisliği
		3	Makine ve İmalat Mühendisliği
		4	Kimya ve Süreç Mühendisliği
1	İktisadi ve İdari Bilimler Fakültesi	0	İktisat
		1	İşletme
		2	Maliye
		3	Siyaset Birimi ve Kamu Yönetimi
		4	Yönetim ve Bilişim Sistemleri
		5	Uluslararası ilişkiler
2	Fen Edebiyat Fakültesi	0	Arkeoloji
		1	Coğrafya
		2	İstatistik
		3	Fizik
		4	Kimya
		5	Matematik
		6	Moleküler Biyoloji ve Genetik
		7	Tarih
		8	Türk Dili Ve Edebiyatı
3	Güzel Sanatlar ve	0	Güzel Sanatlar

	Tasarım Fakültesi		
4	İslami İlimler Fakültesi	0	Felsefe ve Din Bilimleri
		1	İslam Tarihi ve Sanatları
		2	Temel İslam Bilimleri
5	Ziraat ve Doğa Fakültesi	0	Bahçe Bitkileri
		1	Biyosistem Mühendisliği
		2	Tarla Bitkileri

## 2.Giriş ve Çıkış İşlemi

Giriş işlemi sırasında istemciden sunucuya gelen veriler ,veritabanında kontrol işleminden geçirilerek yanıt olarak işlem yapılarak döner.Girişi onaylanan kullanıcıya bir token atanır ve veritabanında token tutulur.Token güvenlik için önemlidir.Giriş yapmış bir kullanıcının herhangi bir kişinin dışarıdan kullanıcı idsini bilerek işlem yapmasını engeller.

### 2.1.Giriş ve Logid(token) Kaydı

>Post /user/login

Giriş Parametreleri:

Adı	Tipi	Açıklaması
kadi	String	Giriş için kullanılan Kullanıcı adı
pass	String	Giriş İçin Kullanılan şifre

Çıkış Parametreleri:

Adı	Tipi	Açıklaması
status	Boolean	İşlem başarılıysa true ,değilse false yanıtı döner.
logid	String	Eğer giriş yapıldıysa logid döner ve karakter dizisidir.logid ile birlikte diğer işlemler yapılmaktadır.
message	String	İşlemlerin başarılı olup olmama durumlarında mesaj döner.

## 2.2.Çıkış İşlemi

>Post /user/logout

Giriş Parametreleri:

Adı	Tipi	Açıklaması
logid	String	Çıkış işlemi yapmak için logid gönderimi sonucu token kaydımız sistemden silinir.

Çıkış Parametreleri:

Adı	Tipi	Açıklaması
status	boolean	Çıkış işlemi başarılıysa true ,değilse false Döner.

## 3.Profil Bilgileri Alma İşlemi

>Post /user/getProfile

Giriş Parametreleri:

Adı	Tipi	Açıklaması
logid	String	İşlem yapabilmek için logid gönderilmesi gerekir.

Çıkış Parametreleri:

Adı	Tipi	Açıklaması
status	Boolean	İşlem başarılıysa true değilse false döner.
profileData	String	ProfilBilgileri içeren veri kalıbı döner.

## 4.Arkadaş Bilgileri Alma İşlemi

>Post /user/getFriends

Giriş Parametreleri:

Adı	Tipi	Açıklaması
logid	String	İşlem yapabilmek için logid gönderilmesi gerekir.

Çıkış Parametreleri:

Adı	Tipi	Açıklaması
status	Boolean	İşlem başarılıysa true değilse false döner.
friendData	String	Arkadaşları içeren veri listesi döner.

## 6.Haberleri Okuma İşlemi

>Post /news

Giriş Parametreleri:

Adı	Tipi	Açıklaması
logid	String	İşlem yapabilmek için logid gönderilmesi gerekir.

Çıkış Parametreleri:

Adı	Tipi	Açıklaması
status	Boolean	İşlem başarılıysa true değilse false döner.
news	Haber Dizisi	İşlem başarılı ise haber dizisi döner.

## 7.Haber Oluşturma İşlemi

>Post /news/CreateShare

Giriş Parametreleri:

Adı	Tipi	Açıklaması
logid	String	İşlem yapabilmek için logid gönderilmesi gerekir.
type	İnt	Yazı:0 Etkinlik:1 Fotograf:2
* date	String	Etkinliğin Tarihi
* yer	String	Etkinliğin Yeri
* time	String	Zaman Etkinliğin Zamanı

\* olan değişkenler sadece Etkinlik tipinde(type = 1) gönderilen değişkenlerdir.

Çıkış Parametreleri:

Adı	Tipi	Açıklaması
status	Boolean	İşlem başarılıysa true değilse false döner.

## 8.Yorum Okuma İşlemi

>Post /news/getYorums

Giriş Parametreleri:

Adı	Tipi	Açıklaması
logid	String	İşlem yapabilmek için logid gönderilmesi gerekir.
newsId	MongoDb ObjectId	Haberin id'si

Çıkış Parametreleri:

Adı	Tipi	Açıklaması
status	Boolean	İşlem başarılıysa true değilse false döner.
yorumData	Yorum Dizisi	İşlem başarılı ise yorum dizisi döner.
message	String	Eğer newsIdye ait haber yoksa “yorum yapılacak haber nesnesi bulunamadı” mesajı döner.

## 9.Haber Beğenme/ Vazgeçme İşlemi

>Post /news/likeShare

Eğer kişi tarafından haber beğenilmemişse, beğenilir fakat beğenilmişse,beğenmekten vazgeçilir.

Giriş Parametreleri:

Adı	Tipi	Açıklaması
logid	String	İşlem yapabilmek için logid gönderilmesi gerekir.
newsId	MongoDb ObjectId	Haberin id'si

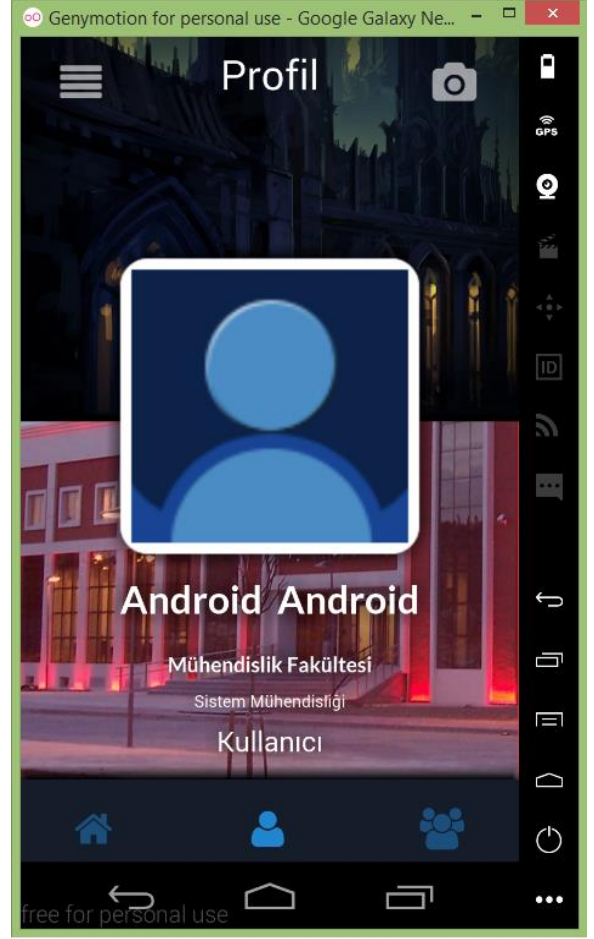
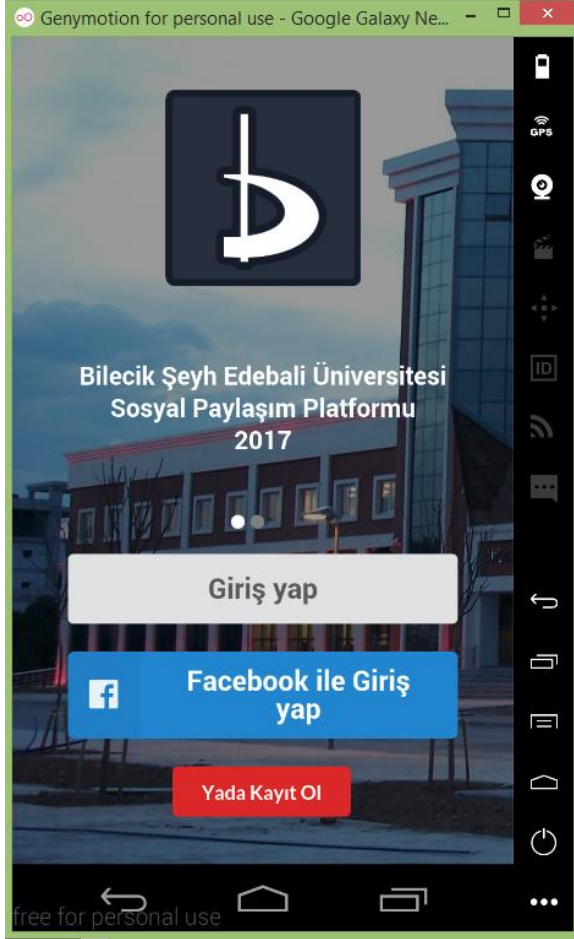
Çıkış Parametreleri:

Adı	Tipi	Açıklaması
status	Boolean	İşlem başarılıysa true değilse false döner.

## GENYMOTION TESTLERİ

SocialUni projesini android olarak derleyip çalıştırırken derleme işlemini android sdk yapmaktadır.Cordova'yı bilgisayarımıza kurduğumuzda localde android sdk'nın yerini bilmektedir.android sdk localde çalışan cihazlara bakmaktadır Eğer bilgisayarda sanal bir android cihaz çalıştırıyorsak onu görecektir ve projeyi ona yükleyip çalıştıracaktır.

Bilgisayarımızda bir çalışan sanal android cihaz yoksa android sdk ile gelen android emulator programını başlatacak ve onda çalıştıracaktır fakat bu program yavaş olduğundan biz genymotion programından yararlanmaktayız.



## ÇIKARILAN SONUÇLAR

Sosyaluni projemiz iyi bir şekilde çalışmaktadır. NodeJS ile Api yazmak yaklaşık olarak projenin %20 sini alırken, istemci için tasarımları oluşturmak geri kalan %80 lik zamanı alındığını söyleyebilirim

Semantic UI Mobil cihazlar için düşünülmediğinden bazı tasarım sorunları yarattı ama responsif olduğundan tasarımların çoğu uyum sağladı.

Jquary animasyonlarda mobil cihazlarda yavaş çalışma görüldü. (çözümü kendi elimizle css animasyonlar yazmak olduğu yorumlarına rastladım fakat proje süresini uzatan bir işlem).

AngularJs sorunsuz bir şekilde çalıştı fakat kodların yüklenme sırasına göre yazılan bir cookie javascript dosyası angularjs'nin çalışma zamanına yetişemeyip angularjs modülü görememekte ve okunamadı hatası vermektedir.

Sunucu sorunsuz bir şekilde çalıştı fakat kullanıcı sayısına göre hız karşılaştırma testleri yapılmadı. Sunucudan veri gönderme ve yanıt hızları anlık yavaş bir hıza asla rastlanmadı.

Sunucun veritabanı ile iletişim bağlantısında sorunsuz bir şekilde çalıştı. Veritabanından veri alma, yazma işlemleri asla yanlış bir sonuç yaratmadı.

Yazılan Sunucu api adreslemelerde sorunsuz bir şekilde çalışmaktadır.

## Kodlar

### Arayüz Cordova Kodları

<https://github.com/EfecanAltay/SocialUni-Bilecik>

### Sunucu Kodları

<https://bitbucket.org/EfecanAltay/cordovamobiltest/src/81dac9702ee7d9798e297ec31664a7332c9b6790>

## Kaynaklar

NodeJS Resmi Sitesi

<https://nodejs.org/en/>

Genymotion Resmi Sitesi,

<https://www.genymotion.com/>

Apache Cordova Resmi Sitesi,

<https://cordova.apache.org/>

Npm Resmi sitesi,

<https://docs.npmjs.com/>



## EKLER

### EK-1.dbConnect Modülü

---

Aşağıdaki kodlara

<https://github.com/EfecanAltay/DefaultNodeExpressServer/blob/master/routes/dbConnect.js> adresinden ulaşabilirsiniz.

//Database Modülü

**var** databaseLocate = "mongodb://localhost:27017/deneme"; //mongoDB nin local adresi

// mongodb bağlantısı gerçekleştirmek için kullandığımız kütüphaneler.

**var** mongodb = **require**('mongodb'); //MongoDB modülü çağrılıyor.

**var** MongoClient = mongodb.MongoClient; //mongoDB MongoClient Modülü çağrılıyor.

// database bağlantısı için gerekli fonksiyonlarımızı oluşturuyoruz.

**var** ConnectDB = {

  connect:**function**(callback){ //connect fonksiyonuyla veritabanımıza bağlantı gerçekleştiriyoruz.

    mongoClient.connect(databaseLocate, **function**(err, db) {

**if**(!err) {

**if**(**typeof** callback === "function")

        {

          callback(db); // callback fonksiyon olarak db parametrelili kullanılır(örn.

**function**(db){...yapılacak işlemler[örn:var c=db.collection]...})

        }

      }

      db.close();

    });

  },

  setLocate:**function**(locateUrlString){ //setlocal fonksiyonunla veritabanı bağlantı adresini değiştirebiliyoruz

    databaseLocate = locateUrlString;

  }

};

---