

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI

TASARIM BAŞLIĞI BURAYA YAZILACAK
GEREKİRSE İKİNCİ SATIR
GEREKİRSE ÜÇÜNCÜ SATIR

B181210389 – Abdulkadir ATAR

Bölüm : **BİLGİSAYAR MÜHENDİSLİĞİ**
Danışman : **Öğr.Gör.Dr. Yüksel YURTAY**

2020-2021 Güz Dönemi

ÖNSÖZ

Büyük firmalar yada kamu kurumları başta olmak üzere bina yapısı büyük, son kullanıcısı fazla olan iş yerlerinde donanımların envanterleri, bakım-onarım ve işletilmesi, son kullanıcı tarafından destek talepleri, içeride kullanılan dahili hatların elektronik rehber yönetimi, birimlerde çalışan personellerin özlük takipleri gibi çok sayıda kalem iş yükü bulunmaktadır. Teknoloji çağı olan günümüzde bu işlemler hatta daha fazlası doğrudan yada dolaylı olarak iş yerlerinin Bilgi İşlem birimlerinin omuzlarında kalmaktadır. Kullanıcı yada birim sayıları arttıkça bu işlerin takibinin basit şekilde hesap tablolarında tutulması yada fiziki dokümantasyon yoluyla saklanması kayıplara yol açmaktadır. Bunun yerine günümüzde tüm bunlara özel yazılımlarla çözüm aramak en uygun sonucu vermektedir.

Dünyanın en büyük Adalet Sarayı projesi olan İstanbul Anadolu Adalet Sarayı'nda çalışıyor olmam sebebiyle yaşanan bu sorunlardan yola çıkarak yapımını hedeflediğim projemde hem eğitimimin tamamlayıcı unsuru hem de çalıştığım kurumum aracılığı ile bir kamu kurumu tarafından kullanılacak olmasının verdiği ayrı bir heyecanla çalışmalarımın ilk aşamasının tamamlanmaya bu kadar yaklaşmasının mutluluğu içerisindeyim.

Çalışmalarım süresince başta bu projeyi üretmeme destek olan Proje Danışmanım Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü Öğr.Gör.Dr. Yüksel YURTAY, İstanbul Anadolu Adliyesi Bilgi İşlem Müdürü Mehmet AYDOĞMUŞ olmak üzere bu süreçte manevi desteğini esirgemeyen eşim Ebru ATAR'a saygılarımı ve teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖNSÖZ.....	i
İÇİNDEKİLER.....	ii
ŞEKİLLER LİSTESİ.....	iv
TABLolar LİSTESİ.....	v
ÖZET.....	vi

BÖLÜM 1.

GİRİŞ.....	1
1.1. Programlama Dili ve Kullanılacak Argümanların Seçimi.....	1
1.2. Geliştirilecek Yazılım İhtiyacı.....	7

BÖLÜM 2.

PROJE İŞ SÜREÇLERİNİN BELİRLENMESİ.....	11
2.1. İş Paketleri.....	11
2.2. Olası Riskler.....	12
2.3. Başarı Kriterleri.....	12

BÖLÜM 3.

VERİTABANI.....	13
3.1. Veritabanı Diyagramları.....	13

BÖLÜM 4.

PROJE EKran GÖRÜNTÜLERİ.....	15
------------------------------	----

BÖLÜM 5.

SONUÇLAR VE ÖNERİLER.....	20
---------------------------	----

ÖZGEÇMİŞ.....	21
BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI.....	22

ŞEKİLLER LİSTESİ

Şekil 1.1.	Laravel MVC Modeli	3
Şekil 3.1.	Mevcut kodlama aşamasında oluşan Veritabanı diyagramı.	12
Şekil 3.2.	Mevcut kodlama aşamasında oluşan Veritabanı diyagramı ve Tablolar.....	13
Şekil 4.1.	Giriş Sayfası.....	14
Şekil 4.2.	Administrator Ana Ekranı	15
Şekil 4.3.	Bina Yönetim Sistemi Portal Ekranı	16
Şekil 4.4.	Telefon Yönetim Sistemi Portal Ekranı	17
Şekil 4.5.	Kullanıcı ve İzin Yönetimi Sistemi Portal Ekranı	18
Şekil 4.6.	Envanter Takip Sistemi Portal Ekranı	19

TABLÖLAR LİSTESİ

Tablo 2.1.	İş Zaman Çizelgesi	10
Tablo 2.2.	Risk Yönetimi Tablosu	11
Tablo 2.3.	Başarı Ölçütleri Tablosu.....	11
Tablo 3.1.	Tek sayfada bulunan tablo örneği.....	26

ÖZET

Anahtar kelimeler: Yönetim Sistemleri, Web Uygulama

Büyük kuruluşları yönetmek epeyce zor bir iş. Günümüzde bu zorluğu aşmak adına geliştirilen özel yazılımlarla daha rahat bir şekilde işlerimizi halledebiliyoruz. Bu bağlamda tasarladığım yazılım ile kuruluşların tüm ihtiyaçlarını ortak bir sistem üzerinden giderebilmeyi planlamaktayım. Sistem içerisinde temel kullanıcıdan yöneticiye kadar yükselen bir hiyerarşi ile çeşitli yetkilendirmeler yaparak işleri küçük parçalar haline bölmeyi, bu küçük parçaların kusursuz entegrasyonu ile de bütünde sorunsuz ilerleyen bir yazılım sistemi oluşturmayı hedeflemekteyim.

Bu tasarım çalışması ile Web yazılımları ile dökümantasyon ve yönetim tarafında yorucu ve birimler arası büyük sorunlar haline dönüşen iş yüklerini hafifletmeyi, bir düzen oluşturulması amaçlanmıştır. Projenin tamamlanması ile ilk canlı testi İstanbul Anadolu Adliyesinde kapalı devre olarak yayına alınacaktır. Bu sebeple projenin ilerlemesi safhasına bazı kısımlar kuruma özelleştirilmiş olsa da proje tamamlandığında bu kısımların da kuruluşlara göre ayarlanabilmesi sağlanacaktır.

Proje PHP(Laravel) yazılım dilinde kodlanacak olup, Mysql veritabanı yönetim sistemi kullanılacaktır. Projenin tarafımca tek başıma yürütülecek olmasından dolayı arayüz(UI) tasarımını en baştan yapmak yerine ücretsiz dağıtım verilen çeşitli panellerin HTML derlemesi ve düzenlemesi yapılarak oluşturulacaktır.

BÖLÜM 1. GİRİŞ

Yazılım geliştirme süreçleri, yazılım bilgisi ve yeteneğinin yanında hazırlanacak yazılımın kullanılacağı alanlarda ki çeşitli bilgi birikimine ve iş süreçlerinin nasıl ilerlemesi gerektiğine dair tecrübelerle de dayanmaktadır. Geliştirilecek yazılımın başarılı bir süreç yönetimine hizmet etmesi için söz konusu kuruluştaki işlerin nasıl yürüdüğü konusunda sizi bilgilendirecek bir danışmana sahip olmanız yada bu süreçlere bizzat hakim olmanız gerektiği gibi aynı zamanda da durumu iyi analiz edebilmeniz gerekmektedir. Bunun yanında yine geliştirme yapılacak platform, yazılım dili, kullanılacak veritabanı yönetim sistemlerinin de uygunluğu çok önemli olacaktır.

1.1. Programlama Dili ve Kullanılacak Argümanların Seçimi

Yazılım geliştirme süreçlerinde önemli kısımlardan biri kodlama dili seçimidir. Bu süreci eğer bir ekiple yapmayı planlıyorsanız bu durum daha da önemli bir hal alacağı gibi ileride projenin genişlemesi yada desteğe ihtiyaç duyulması halinde gerekli ekip arkadaşının hızlıca dahil edilmesi için güncel bir yazılım dilinin seçilmesi her zaman avantaj sağlayacaktır. Kullanılacak yazılım dili bunun dışında yayına alınması sırasında sunucu vb. gereksinimleri de etkileyeceği düşünülerek seçilmelidir. Mümkünse güncel ve sık kullanılan platformlarda yayın yapılabilecek bir dil ile geliştirme yapılmalıdır.

Programlama dili bir makineye belli bir algoritma dahilinde nasıl çalışması gerektiğini, hangi verileri saklayıp hangilerini işlemesini istediğimizi anlatmamıza yarayan kendine göre kuralları olan üretilmiş bir yapıdır. Burda makineden kasıt bilgisayar olabileceği gibi evinizdeki buzdolabıda olabilir. Anlatılmak istenen kodları çalıştıracak mikroişlemcisi/mikrodenetleyicisi olan bir donanımdır. İnsan dili gibi doğal değildir. Üretilmiştir ve ifadeler sınırlıdır. Algoritma bir problemi çözmek için tasarlanan, belli girdilere ve şartlara göre ilerleyen bir başlangıcı ve sonu olan yoldur. Bilgisayar tarihi boyunca binlerce programlama dili geliştirilmiştir. Bunlardan büyük bir çoğunluğu tutunamayıp yok olmuş bazıları ise benimsenip onlarca seneden beri hala daha kullanılmaktadır. Bunlardan bazıları devletin, askerin sistemlerinde özel alanlarda çalışırken bazıları ise insanlığa mal olmuş çok geniş bir uygulama alanı bulmuştur.

Programlama dillerini makine seviyesi, alt seviye, orta seviye, üst seviye şeklinde sınıflandırabiliriz. Makine seviyesi dilleri kolay anlaşılması pek mümkün değildir. Üst seviye dillerden belli bir kurala göre oluşturulmuş sayılardan oluşmaktadırlar. Bir programlayıcı vasıtasıyla hedef mikroişlemciye/mikrodenetleyiciye yüklenip direk olarak çalışmaya başlayabilirler. Bunlardan en popülerleri Intel Hex'tir. Intel firmasının zamanında geliştirdiği hexadecimal yani 16 lık sayı sisteminde hazırlanmış bir dosyadır. İsmi burdan gelmektedir. Hedef platforma göre oluşturulmaktadır. Hedef platformdan kastım şuan bu yazıyı okuduğunuz bilgisayarınızın işlemcisinden, cep telefonunuzun ARM mikrodenetleyicisine, LED yakıp söndürdüğünüz PIC mikrodnetleyicinize kadar geniş bir alan olabilir. Aşağıda paylaştığım Intel HEX formatındaki bir dosyayı gördüğünüz gibi anlamaya çalışmanız pek mümkün değil ve anlamsızdır.

Projemizi bir web uygulaması olarak geliştirmeyi planladığımızdan dolayı gelişmiş yazılım dillerinden olan PHP dilinde kodlama yapılması tercih edilmiştir. PHP programlama dili, internet için üretilmiş, sunucu taraflı, çok geniş kullanımlı, genel amaçlı, HTML içerisine gömülebilen betik ve programlama dilidir. İlk kez 1995 yılında Rasmus Lerdorf tarafından yaratılan PHP'nin geliştirilmesi bugün PHP topluluğu tarafından sürdürülmektedir.

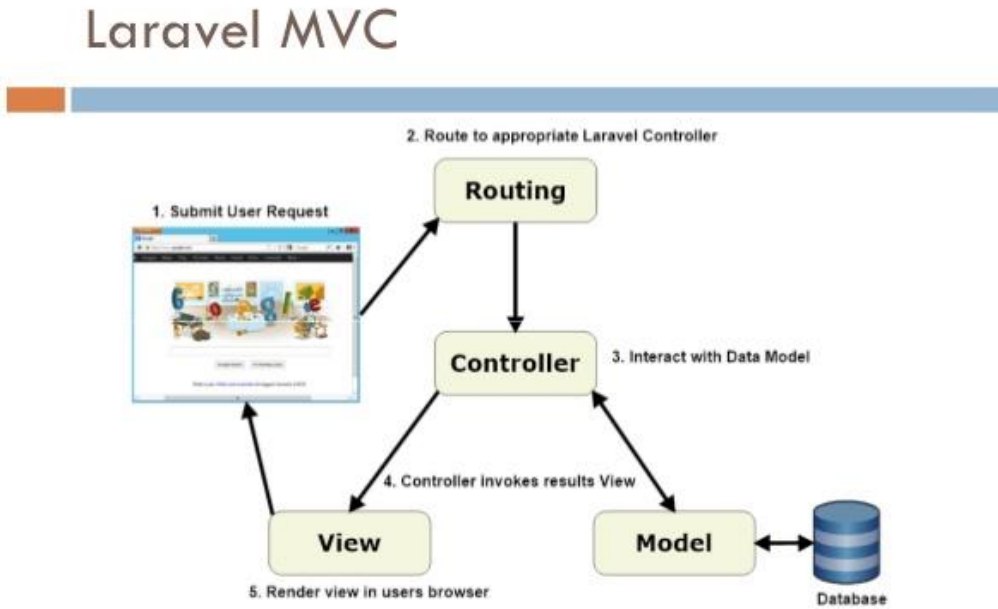
PHP kodları PHP işleme modülü bulunan bir web sunucusu tarafından yorumlanır ve çıktı olarak web sayfası üretilir. Bu kodlar veriyi işlemek üzere harici bir dosyaya kaydedilerek çağırılabilirdiği gibi doğrudan HTML kodunun içine de gömülebilir. PHP zaman içinde bir komut satırı arayüzü sunacak şekilde evrilmiştir, PHP-GTK yardımıyla grafiksel masaüstü uygulaması geliştirmek de mümkündür.

PHP özgür bir yazılım olup PHP Lisansı ile dağıtılmaktadır. Bu lisans kullanım şartları kısmında GNU Genel Kamu Lisansı ile örtüşmese de, PHP tüm web sunuculara ve hemen hemen tüm işletim sistemi ve platforma ücretsiz olarak yüklenebilir.

PHP dilinin ham hali ile de yazılabilecek olmasına karşın, günümüzde framework kullanımı oldukça yaygın hale gelmiş ve süreçleri de kolaylaştırdığı açıkça görülmektedir. Framework, yazılım geliştiricilerin kullandığı önceden hazırlanmış kütüphanelerin bulunduğu ve bunlara yenilerini ekleyebileceği yapıların adıdır. Gelişmiş frameworklerde form kontrolü, veri tabanı bağlantısı, kullanıcı giriş çıkış, mail atma, tema motoru gibi kütüphaneler mevcuttur. Frameworkler ayrıca MVC

(Model, View, Controller) gibi bölümlerden oluşarak projenizin daha okunabilir düzenlenebilir olmasını sağlamaktadır.

Yapılan projelerin çoğunda aradan belirli bir zaman geçtikten sonra düzenlemek zorlaşır. Hem kodların artması hem de yazılan kodun unutulması revize sürecini uzatmakta. Bu yapılar Modelde veritabanı işlemlerini, Controllerda genel mekaniği, Viewda ise görünüm kodu içerdiği için müdahale edilmesi gereken yer tam olarak bulunabilmektedir. Aşağıda Laravel için örnek MVC yapısını görebilirsiniz.



Şekil 1.1. Laravel MVC Modeli

Frameworkler temel yapıları en çok kullanılan kütüphaneleri ve modülleri barındırır. Hiç bir framework ana yapısında gelişmiş yapıları içermez. Bunun yerine genişletmek yoluyla projenize uydurulur. Örneğin giriş yaparken genel olarak eposta değişkeni ile giriş yapılır. Siz eğer kullanıcı adı ile giriş yapmak istiyorsanız sadece o fonksiyonu değiştirerek projenizi yazmaya devam edebilirsiniz. Framework kullanmak size bu bağlamda hız katar. Veri tabanı bağlantıları, güvenlik sorunları veya yetkilendirme gibi en temel bölümleri yazmanıza gerek yoktur. Frameworkün ya içinde vardır yada çok kolay bir şekilde dahil edilebilir. Bu sayede tam olarak

bitirilmesi 2 ayı bulacak projeler bazen 2-3 hafta da bitebilmektedir. Diğer bir kazancı ise belli standartları olduğu için bu standartlarda geliştirilen projeye yeni takım arkadaşları kolaylıkla entegre olabilir. Bu gibi faydalara rağmen kısıtlanmış ve bazı kriterlerinizde sorun yaşıyorsanız framework kullanmayabilirsiniz.

Bu projemizde kullanacağımız PHP yazılım dilinin frameworkleri araştırıldığında Laravel, CodeIgniter, Symfony, CakePHP gibi başlıca frameworkler karşımıza çıktığı görülmüştür. Laravel'in geliştirici sayısının fazla olması, piyasada güncel olarak aranan geliştirme yöntemlerinden olması ve hala framework geliştiricileri tarafından yenilikler ekleniyor olması sebebiyle projede kullanılması tercih sebebi olmuştur.

Laravel ihtiyaç duyulan, gelişmiş bir çok özellik ve yapıyı üzerinde barındıran, PHP ve OOP tüm nimetlerinden yararlanan, web uygulamaları geliştirmeyi sağlayan açık kaynak PHP framework' tür. Yapısı gereği gerçekten çok sade ve temiz kod yazarak istediğiniz uygulamaları geliştirme imkanı verir. Söz dizimi çok basit ve anlamlıdır. Alışmak için zorlanmazsınız, çabuk öğrenilebilir. Bir kaç işlem barındıran küçük uygulamalardan, büyük kurumsal projelere kadar her türlü web uygulamasını tasarlama esnekliğine sahiptir. OOP ve PHP nin tüm nimetlerinden yararlanır böylece güncel php özelliklerinde oop uygun şekilde çalışırız. Laravel'in bize sunduğu kolaylıklara baktığımızda; ORM, Blade Template, Route, Migrations (Sürüm Kontrolü, Göçler), Unit Test (Birim Test), Automatic Pagination (Otomatik Sayfalandırma), Modüler Paket Yönetimi ve Composer, Performans, Topluluk gibi avantajları görmekteyiz.

ORM : Database ile uygulamamızda (Object-Oriented) nesnelerimiz sayesinde bağlantı kurup yönetmemizi sağlayan bir yapıdır. Klasik SQL cümleleri yazmadan nesnelerimiz üzerinden veri tabanına erişim sağlayıp kontrol edebiliyor sorgular çalıştırabiliyoruz. ORM database den bağımsız çalışır. Yani Mysql, SQLite, postgresql, MSSql, Oracle gibi bir çok database için aynı kodları kullanırsınız.

Blade Template: Blade adı verilen template engine sahiptir. Uygulamanıza yine çok kolay ve sade şekilde arayüzle bütünleştirebiliriz. Öğrenilmesi yine kolaydır,temiz ve dinamik arayüzler hazırlayabilirsiniz html ve php kodları içinde savaşı vermezsiniz.

Route: Müthiş bir route (yönlendirme) mekanizması vardır. Yorulmadan temiz URL elde eder api ler için uygun erişim yönlendirmeleri yapabilirsiniz. Php yapısında

yabancı olmadığınız şekilde. Burada ister yönlendirmelerinizi yapabilir ister filtrelerden kontrollerden geçirebilir ister Controller Class'larınıza yönlendirebilirsiniz. Hatta burada bu fonksiyonda uygulamanızın gerçekleştireceği işlemleri dahil controller class lara gerek kalmadan gerçekleştirebilirsiniz ancak bu pek önerilmez.

Migrations (Sürüm Kontrolü, Göçler): Veri tabanı sürüm kontrol sistemidir. Artisan Komut Satırı ile uygulamanızın veritabanına şemalar ekleyebilir düzenleyebilirsiniz. Veri tabanı yönetim sistemine gitmeden sisteminizdeki veritabanınızı oluşturmaya yada güncellemeye yarayan yapı. Örneğin uygulamanızı başka bir ortama taşıdınız, veri tabanını oluşturmaya çalışmaktansa uygulamanızda ki hali hazırdaki yapıyı çalıştırarak sistemi hazır hale getirmiş oluyoruz.

Unit Test (Birim Test): Uygulamanızı test etmek için birim testler oluşturup çalıştırmamızı sağlar. Artisan komut satırıyla hazırladığımız testleri çalıştırabiliriz.

Automatic Pagination (Otomatik Sayfalandırma): Laravel bizim yerimize sayfalama sistemini düşünmüş ve bizi zahmete sokmadan el atmış. Kolay şekilde kullanabilir istersek değiştirebiliriz.

Modüler Paket Yönetimi ve Composer: Composer, uygulamanızın üçüncü parti paketlerini kontrol edip hızlı şekilde ekleyip yönetmeyi sağlar. Composer ile bağımlılıklarınızı paketlerinizi dert etmez tek tek uğraşmak yerine çok kolay bir şekilde dahil edebilirsiniz.

Performans: Cache mekanizmaları sunar. Redis ile bütünleşmiş bir yapısı var adeta evlat edinmişcesine sahip çıkar, bünyesine katmıştır. Projenize redis dahil etmek için ekstra uğraşmanıza gerek yoktur. Çok hızlı ve basit şekilde yapılandırabilirsiniz. Dahili olarak auth, filter gibi bir çok yapıyı içinde barındırıyor. Bunu yine kendi yapısına uygun ve basit şekilde sunar. Hemen hemen her uygulamada auth mekanizması olur ve bunu yapmak için ekstra çabaya gireriz. İşte laravel bunu da pas geçmemiş çok güzel bunun gibi yapılarda sunmuş. Bir çok Symfony bileşeni üzerine kurulmuştur. Güvenli, hızlı, sitenizi ayağa kaldırmak çok kolaydır. Tek komutla sitenizi aktif hale veya yapım aşamasına alma gibi seveceğiniz bir çok özelliği daha mevcuttur.

Topluluk: Bir yapıda en büyük özelliklerden birisi de bana göre topluluktur. Başınız sıkıştığında yardım alabileceğiniz, danışacağınız, müthiş örnekler ve çözümler bulabilmeniz gibi bir çok yardımı dokunur. Çok hızlı büyüyen topluluğa sahiptir. Henüz genç bir freamwork sayılır. Bu sebeple örneğin bir codeigniter kadar topluluğa ve aşinalığa sahip değildir. Ancak çok hızlı büyümektedir. Popülerliği her geçen gün artmaktadır. Türkiye’de de kabul görmüş hızla atan bir kitlesi var, ayrıca iş ilanlarında da laraveli görmek mümkün. Çok iyi bir Türkçe doküman hazırlanmış. Ayrıca resmi sitesinde, yerli ve yabancı forumlarına yine ulaşabilirsiniz.

Projede yazılım dilinin seçimi kadar veritabanının da seçimi önemli bir husustur. Sayıları gittikçe çoğalan veritabanı yönetim sistemlerinin, maliyeti ve kullanımı için gerekli uzmanlık düzeyi çok farklılık gösterdiği için, uygulama yazılımı ile birlikte hangi veritabanı yönetim sisteminin kullanılacağını belirlemek, uygulamanın yapılacağı ortamdan çok başlangıç ve bakım maliyeti üzerine yoğunlaşmayı zorunlu kılmıştır.

Günümüzde veritabanları ile ilgili yapılan araştırmaların bir çoğu veri tabanlarının hızı, güvenliği, erişilebilirliği ve yapılacak işlemlerin hatasız olabilmesi üzerine yoğunlaşmıştır. Ancak satış politikaları nedeniyle, başlangıç ve bakım maliyetleri hep ikinci plana itilmiştir. Bazı büyük veritabanı yazılım üretici firmaları tarafından yapılan araştırma geliştirme (ARGE) çalışmaları, çoğunlukla maliyeti ikinci planda bırakıp bütün yükü son kullanıcıya yükleyerek, uygulama yazılımlarının bu duruma çözüm getirmelerini beklemişlerdir. Özellikle, KOBİ’lerde maliyet öneminin ön planda olduğu durumlarda, geliştirilecek uygulamaların bakım ve işletme maliyetlerinin düşük seviyede olmasını gerektirir.

Konu bir seçim yapmaksa eğer, öncelikle alternatiflerimizin ne olduğunu bilmemiz gerekir. Bu alternatifler hakkında ne kadar derinlemesine bilgi sahibi olursak alacağımız kararlarda o kadar doğru olacaktır. İlişkisel veri tabanları ve yönetim sistemleri(RDBMS) ile ilişkisel olmayan (NoSQL-Not Only SQL) veri tabanlarından hangisi ile çalışmamız gerektiğini tespit etmekle başlamak en doğrusu olacaktır. Eğer ki doküman tabanlı NoSQL veri tabanı kullandığınız bir servisiniz de, transaction kullanımına ihtiyaç duyuyorsanız, aynı anda birden fazla dokümanı güncelleme gibi bir ihtiyacınız oluyorsa, NoSQL’in kapsama alanından çıkmış, ilişkisel Veritabanı Dünyasına girmişsiniz demektir.

Bugün piyasaya baktığımızda en popüler ilişkisel veri tabanları olarak, MySQL (ücretsiz), PostgreSQL (ücretsiz), MSSQL, Oracle, IBM DB2 gibi veri tabanlarını sıralayabiliriz. Bu veri tabanlarından hepsi ilişkisel olmasına rağmen, bunlar arasında da yine seçim yaparken bilinçli bir seçim yapmak bizim faydamıza olacaktır.

MS Access, MsSQL, MySQL, Oracle, PostgreSQL gibi Veritabanı sistemleri içersinden PHP yazılım dili ile çalışmasının uyumu, projemize uygunluğu ve yazılımcının hakimiyeti dikkate alındığında MySQL kullanılmasında tercih sebebi olmuştur.

Proje PHP yazılım dili ile hazırlanacağı için Linux tabanlı bir işletim sistemi üzerinde yayına alınması en sağlıklı yöntem olacağı gibi Windows Server üzerinde de son dönemde desteklenen cross platform yayınları yada üçüncü parti yazılımlar sayesinde(WAMP, XAMP vb.) yayına alınabileceği için bu husus projenin yanına alınacağı kuruluşun alt yapısı doğrultusunda işletilebilecektir.

1.2. Geliştirilecek Yazılım İhtiyacı

Özel amaçlı uygulama yazılımları belirli bazı işleri yapabilmek için geliştirilen, dolayısıyla sınırlı amaçları olan yazılımlardır. Planlama ve programlama, analiz etme, sahip olunan varlıkların kaydını tutabilme, anlık iş akışlarını verimli şekilde yönetmek ve takibini tutmak gibi bir çok farklı birim ve kişilerce yönetilen ancak bütünde birbirine domino taşları gibi bağlı tüm süreci yönetmeyi kolaylaştırmayı amaçlarız. Bu sayede de iş yoğunlukları azaltılırken fiziki dokümantasyon da minimize edilmiş olur.

Bu projede orta ölçek üzerindeki kuruluşların ihtiyaçlarının başında gelen son kullanıcı desteği, idari yönetim, envanter takibi, telefon rehberi gibi kısımları standart şekilde bir omurga oluşturarak bunun yanında kuruluşlara özel farklı modüller eklenebilir şekilde tasarlandı.

Orta ölçek ve üzeri kurumların Bilgi İşlem Müdürlüğü başta olmak üzere tüm teknik kısımlarının son kullanıcı desteği büyük sorun teşkil etmektedir. Hem destek taleplerini kısa sürede sonuçlandırmak hem de destek taleplerini karşılayan ekiplerin eşit ve uzmanlıklarına göre iş dağılımı yapabilmek, verilen desteği en iyi şekilde sunulduğunun geri dönüşünü alabilmeyi yönetmek epey meşakkatli ve zor bir

süreçtir. Bu süreci kolaylaştırmak adına Portal’da Destek Takip Sistemi adı ile oluşturulacak bir modül bulunacak.

Destek Takip Sistemi modülünde son kullanıcılar destek istedikleri konuyu konu başlıklarından seçerek özelleştirecek bu sayede uzmanlığına göre bir personel destek verecek. Destek verecek personel arasındaki iş yoğunluğunu dengelemek için hem personele düşen iş hem de tamamladığı iş için puantaj tutulacak. Destek talebi tamamlandığında talep eden kişiden bir memnuniyet puantajı alınacak, bunun yanında sorunun kısa süre içerisinde tekrar etmesi halinde ise destek talebini aynı teknik personele yeniden iş olarak gönderilebilecek. Destek taleplerinin anlık akışı dengelemesinin yanında birim amirlerince sistemden saatlik, günlük, aylık, yıllık olmak üzere genel ve personel bazında raporlama yapması sağlanarak performans değerlendirmesi yapılabilecek.

Orta ölçek ve üzeri kurumların önemli sorunlarından biri de idari konular olan Oda yerleşim planları ve bu kısımlara ait iç hat numara karmaşasıdır. Oda planlaması yapmak, sirkülasyonunu düzenli şekilde sağlamak, bunun listesini tutmak özellikle model aldığım İstanbul Anadolu Adliyesi gibi büyük kurumlarda çok önemli sorun teşkil etmektedir. Bu sebeple “Bina Yönetim Sistemi” ve “Telefon Yönetim Sistemi” isimli modüller tasarlanmıştır.

Bina Yönetim Sistemi sayesinde Birim Bazlı ve Blok, Kat kıstaslarına göre tanımlama yapılarak bina yapısı sanal ortamda yapılandırılabilir. Birimler ana birim altında alt birimler olarak tasarlandığından genelden özele doğru bir çalışma yapılabilecektir. Model aldığım İstanbul Anadolu Adliyesi’nden yola çıkarsak Cumhuriyet Başsavcılığı > Bilgi İşlem Müdürlüğü yada Adalet Komisyonu Başkanlığı > Aile Mahkemesi > 20. Aile Mahkemesi şeklinde ilerleyerek birime ait oda ve oda tipi seçilebilir. Bu işlem aslında bu projenin temel işlemidir. Buna bağlı olarak oda da bulunan Mahkeme ve buna bağlı iç hat numaraları, envanterler ve ihtiyaç dahilinde farklı verilere rahatlıkla yetkisi olan kişilerce erişilebilecektir.

Telefon Yönetim Sistemi sayesinde, iç hat numarası adeti fazla olan kuruluşlarda oluşan büyük karmaşaların önüne geçilmesi hedeflenmektedir. Projeyi tamamladığımda, hayata geçirmeyi teklif ettiğim İstanbul Anadolu Adliyesinde 9000 adet aktif iç hat numarası bulunmaktadır. Bunların doğru bir şekilde kullanıcılarının ve birimlerinin tutulmasının yanında güncel tutulabilmesi zor bir süreçtir. Bu modül sayesinde güncel ve doğru veri tutulması kolaylaşmış olacak hem de rehber sistemi

doğrudan aynı veritabanından çalışacağı için ayrıca ikinci bir güncelleme işlemine gerek kalmayacaktır.

Kurumların personelleri ile ilgili özlük bilgileri ilgili birimlerce tutulmaktadır. Bu veriler günümüzde artık personel dosyalarından ziyade dijital ortamlara aktarılacak hem ihtiyaç duyulduğunda ulaşılması hem de arşivlenmesinin kolaylaştırılması sağlanmaktadır. “Büro Personeli Yönetim Sistemi” modülü sayesinde istenilen personelin özlük bilgilerine kolaylıkla ulaşılacaktır.

Bilgi İşlem Birimlerinin en büyük problemlerinden biri de işletilmesinden ve bakım-onarımından sorumlu olduğu envanterin sahaya dağıtımından sonra zimmet işlemleri ve gerektiğinde bu envanterin nerede olduğunun tespiti ve dokümanlarının tutulmasıdır. Özellikle binlerce son kullanıcının olduğu durumlarda bu iş içinden çıkılmaz bir hal alabilmektedir. Bu sebeple “Envanter Takip Sistemi” modülü ile tüm envanterdeki cihaz listesini görebileceğimiz gibi yine bu cihazların bina içerisinde hangi birimde, lokasyon olarak da hangi oda numarasında bulunduğu ve kime zimmetlendiğinin tespiti kolaylaşacaktır.

Segbis Takip Sistemi modülü yine Envanter Takip Sistemi modülü ile benzer bir yapıya sahip ancak birkaç ekstra veri tutmamızı sağlayacak kuruma özel bir modül tasarımıdır. 2013 yılından bugüne kadar Adalet Bakanlığı bünyesinde kullanılmakta olan SEGBİS (Ses ve Görüntü Bilişim Sistemi) cihazları İstanbul Anadolu Adliyesi’nde 134 adet bulunmaktadır. Temelde Video Konferans sistemi olarak adlandırabileceğimiz sistemi, CISCO marka C60 model Kodek, Ses Mikseri, 2 adet TV, en az 5 mikrofon, 2 adet Kamera’dan oluşan bir yapı oluşturmaktadır. Bu yapının işletilmesi esnasında arızalara müdahale edilmekte, çeşitli bakım işlemleri yapılmaktadır. Segbis Takip Sistemi modülü ile cihaz kimlik bilgilerinin yanında işletme süreçlerindeki bakım geçmişi, arıza olması halinde bu arızaların başlangıcı ve onarım süreçleri, yapılan onarım içerikleri, kullanılan malzeme detayları vb. adımlar kayıt altında tutularak hem bir sonraki arızada yapılacak işlemler hem de sık nükseden arızaların tespiti kolaylaşacaktır. Bunun yanında yıl içerisinde bakım ve onarım işlemleri için maliyet çıkarılabileceği gibi bir sonraki yıl için alınacak yedek parça ve sarf malzemesi ihtiyacı için bir kanı uyandıracaktır.

Proje modüler sistemde tasarlandığı için modül içinde ihtiyaçlara göre düzenlemeler yapılması daha kolay olacağı gibi ihtiyaç duyulabilecek yada başka bir kuruluşa özelleştirmek için yeni modüller eklenmesi kolay olacaktır.

BÖLÜM 2. PROJE İŞ SÜREÇLERİNİN BELİRLENMESİ

Projeyi tek kişi yürütmek zorunda olmanın zorlukları olduğu açık bir gerçek. Modül sayılarının fazlalığı ve temelde olması gereken modüller düşünüldüğünde yorucu gibi görünüyor olsa da bir taraftan da tüm projeyi tek başına kurgulamak ve yönetmek bir bakıma işleri kolaylaştırmakta. Projeyi toplamda 12 ay gibi bir süre de bitirmeyi planladım. Bu süreçte öncelikle yapacağım yazılımın hangi işlerde nasıl çalışacağını detaylıca inceledim. Bunun için uygun olan platforma karar verdim. Sonrasında arayüz için birden fazla hazır admin paneller buldum, bunlar içerisinde en uygun olanını alarak editledim ve ana şablonumu ortaya çıkardım. Sonrasında ise kodlamaya başladım. İşin burdan sonraki kısmı uzun bir kodlama ve test süreci olacak.

2.1. İş Paketleri

İŞ-ZAMAN ÇİZELGESİ (*)

İP No	İş Paketlerinin Adı ve Tanımı	Kim Tarafından Gerçekleştirileceği	AYLAR											
			1	2	3	4	5	6	7	8	9	10	11	12
1	Proje Yönetimi	<u>Abdulkadir ATAR</u>	X	X	X	X	X	X	X	X	X	X	X	X
2	Proje UI Tasarımı ve Düzenlemeler	<u>Abdulkadir ATAR</u>	X	X										
3	Veritabanı Tasarımı	<u>Abdulkadir ATAR</u>			X									
4	Projenin Kod Yazımı	<u>Abdulkadir ATAR</u>			X	X	X	X	X	X	X	X		
5	Modüllerin Testlerinin Yapılması	<u>Abdulkadir ATAR</u>					X	X	X	X	X	X	X	X
6	Projenin Yayınlanması ve Canlı Testlerinin Yapılması	<u>Abdulkadir ATAR</u>												X

Tablo 2.1. İş Zaman Çizelgesi

2.2. Olası Riskler

RİSK YÖNETİMİ TABLOSU

İP No	En Önemli Riskler	Risk Yönetimi (B Planı)
1	Projenin kodlanması sırasında programcı bilgisinin yetersiz kalması.	Projeyi tek kişi ile yürütülecek olmasından kaynaklı olarak yeni bilgiler öğrenmek.
2	Projenin kodlanması sırasında gerçek iş akışlarına aykırı durum tespit edilmesi	İş akışlarının yeniden analiz edilerek gerekli algoritmaların düzenlenmesi <u>yada</u> en başından yeni algoritmalar çıkarılması
3	Modül testleri aşamasında tespit edilen büyük ölçekli hataların olması	Modüller sistem olmasından dolayı modül içerisinde düzenlemeler yaparak sorunu çözüme kavuşturmak.
4	Tüm modüllerin tamamlanmasından sonra birbirleri ile bağıntılı olan temel modüllerde hataların tespiti	Bu durumda temel modüllerin son aşamaya gelinmesini beklemeden sürekli testlerinin yapılması. Özellikle çalışma zamanı hatalarının itina ile test edilmesi.

Tablo 3.2. Risk Yönetimi Tablosu

2.3. Başarı Kriterleri

BAŞARI ÖLÇÜTLERİ TABLOSU

İP No	İş Paketi Hedefi	Başarı Ölçütü	Projenin Başarısındaki Önemi (%)
1	Proje Yönetimi	Projenin işleyişe ve kurallara uygun tasarlanması ve yürütülmesi	%40
2	Projenin Arayüz Tasarımı	Projenin arayüz tasarımının kullanıcıların kolay kullanımına uygun ve sade olması	%15
3	Projenin Kodlanması	Projenin hızlı, anlaşılabilir ve tekrara düşmeden kodlanması	%30
4	Proje Testlerinin Yapılması	Proje testlerinin zamanında, aksatılmadan ve doğru sonuçlar verecek şekilde yapılması	%15

Tablo 3.3. Başarı Ölçütleri Tablosu

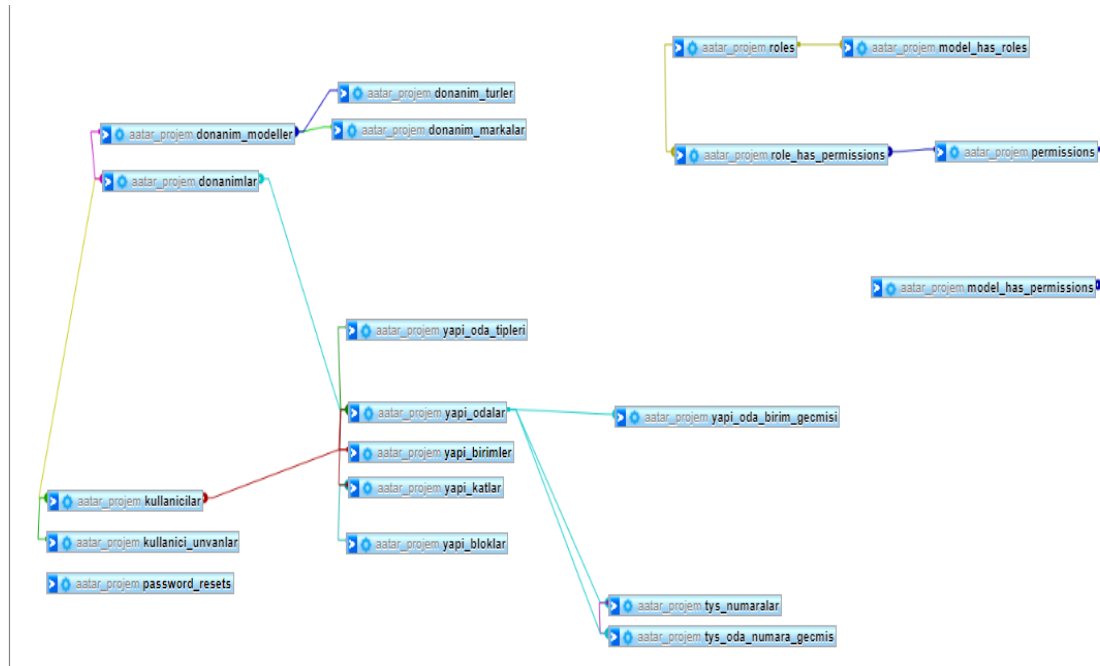
BÖLÜM 3. VERİTABANI

Projede yazılım dilinin seçimiyle uyumlu çalıştırabileceğimiz, bakım maliyetleri düşük veritabanının seçimi önemli bir husustur. Günümüzde her geçen gün sayıları artan veritabanı yönetim sistemlerinin, maliyeti ve kullanımı için gerekli uzmanlık düzeyi çok farklılık gösterdiği için, uygulama yazılımı ile birlikte hangi veritabanı yönetim sisteminin kullanılacağını belirlemek, uygulamanın yapılacağı ortamdaki çok başlangıç ve bakım maliyeti üzerine yoğunlaşmayı zorunlu kılmıştır.

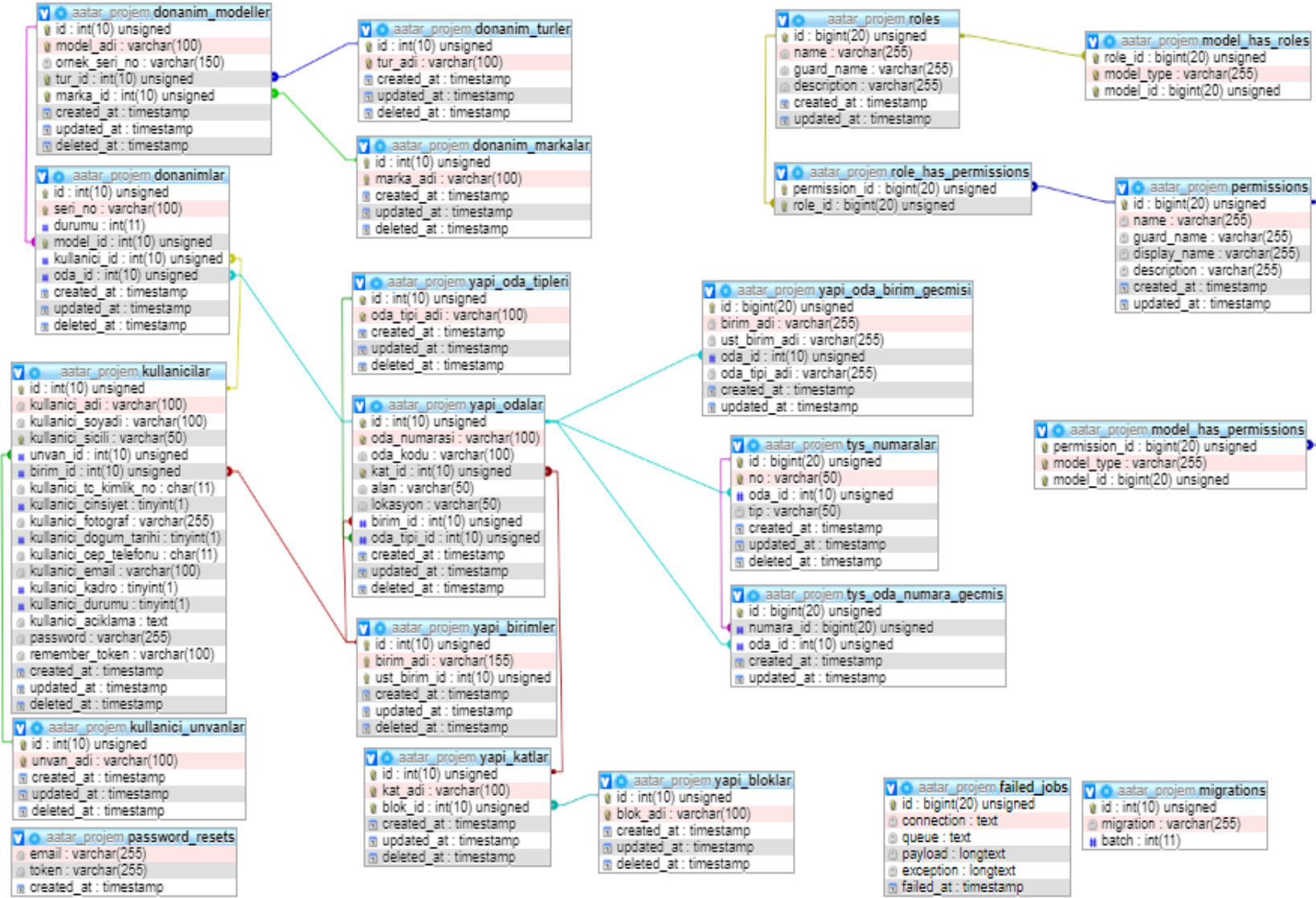
Projemizde hem ücretsiz olması hem de kaynak ve bakımının çok zor olmaması ayrıca projenin PHP yazılım dili ile hazırlanacak bir web uygulaması olması sebebiyle MySQL veritabanı yönetim sistemi kullanılmıştır.

3.1. Veritabanı Diyagramları

Veritabanımızın genel bakışta kullanıcı tablosu tüm modüllerde ortak kullanılacak olup, bunun dışında kalan tüm modüller için ayrı ayrı tablolar planlanmaktadır.




Şekil 3.1. Mevcut kodlama aşamasında oluşan Veritabanı diyagramı



Şekil 3.2. Mevcut kodlama aşamasında oluşan Veritabanı diyagramı ve Tablolar

BÖLÜM 4. PROJE EKRAN GÖRÜNTÜLERİ



Sicil

Parola

Giriş Yap

İstanbul Anadolu Bilgi İşlem Müdürlüğü Proje ve Yazılım Geliştirme Şubesi
© 2020 | Abdulkadir ATAR



Şekil 4.1. Giriş Sayfası



Merhaba, ABDULKADİR

Destek Takip Sistemi

UYGULAMAYA GİT

Envanter Takip Sistemi

UYGULAMAYA GİT

Kullanıcı ve İzin Yönetimi

UYGULAMAYA GİT

SEGBİS Takip Sistemi

UYGULAMAYA GİT

Büro Personeli Yönetim Sistemi

UYGULAMAYA GİT

Bina Yönetim Sistemi

UYGULAMAYA GİT

Telefon Yönetim Sistemi

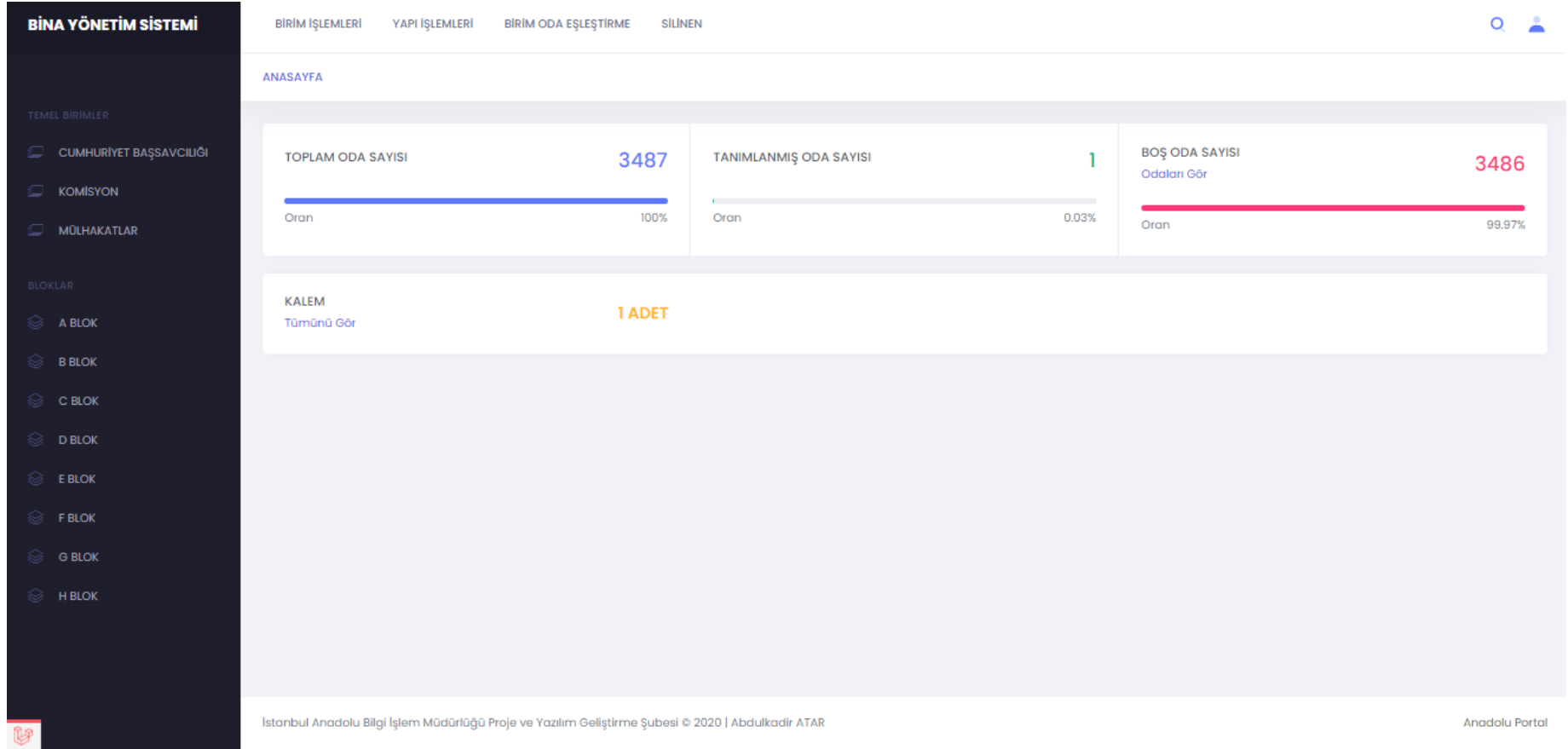
UYGULAMAYA GİT



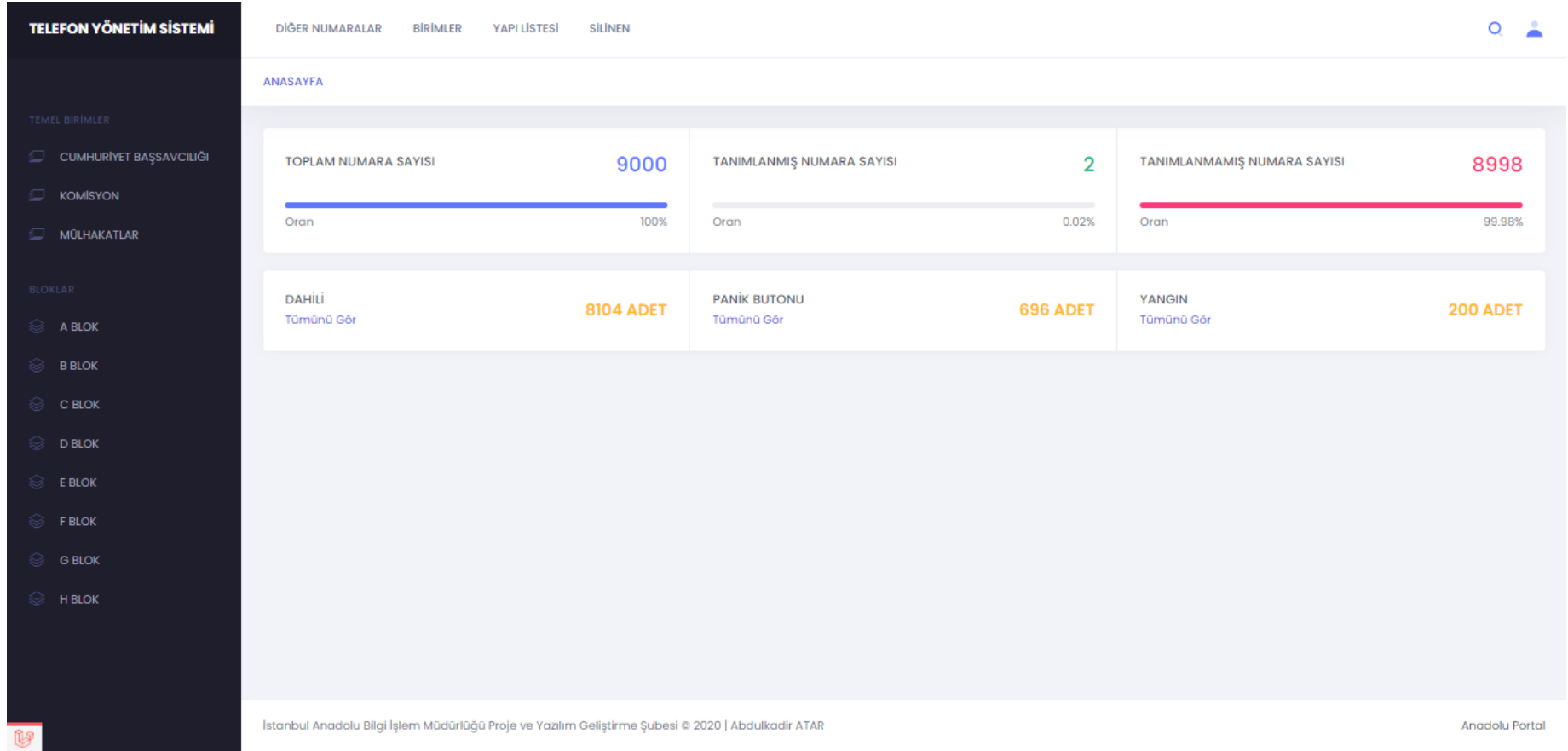
İstanbul Anadolu Bilgi İşlem Müdürlüğü Proje ve Yazılım Geliştirme Şubesi © 2020 | Abdulkadir ATAR

Anadolu Portal

Şekil 4.2. Administrator Ana Ekranı



Şekil 4.3. Bina Yönetim Sistemi Portal Ekranı



Şekil 4.4. Telefon Yönetim Sistemi Portal Ekranı



ABDULKADİR ATAR

AB152595

TEKNİKER

BİRİME TANIMLI DEĞİL

DETAY

UFUK TUĞRAL

AB135629

ZABIT KATİBİ

BİRİME TANIMLI DEĞİL

DETAY

ÜMİT YAŞAR DOĞAN

ab132417

ZABIT KATİBİ

BİRİME TANIMLI DEĞİL

DETAY

YASİN DEMİR

ab188795

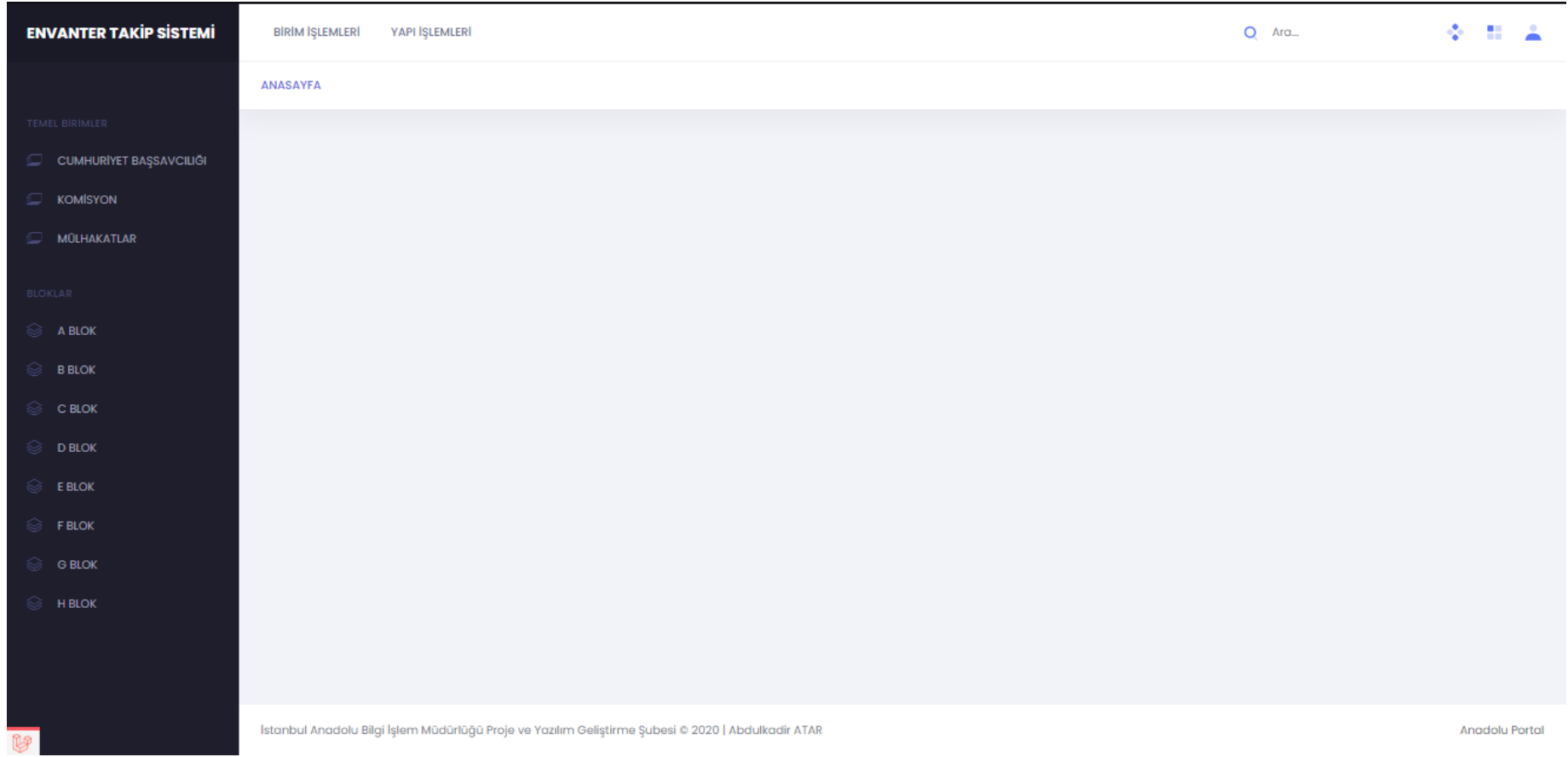
ZABIT KATİBİ

BİRİME TANIMLI DEĞİL

DETAY



Şekil 4.5. Kullanıcı ve İzin Yönetimi Sistemi Portal Ekranı



Şekil 4.6. Envanter Takip Sistemi Portal Ekranı

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

Projemize başlamamızın ardından geçen yaklaşık 4 aylık süreç içerisinde iş süreç takvimi ile orantılı şekilde ilerleme kaydederek Kullanıcı Yönetimi, Bina Yönetimi, Telefon Yönetimi modüllerini büyük ölçüde tamamlamış duruma geldiği, Envanter Takip Sistemi Modülüne de başlandığı görülmektedir. İlerleyen süreçte de planlı bir şekilde devam ederek geriye kalan yaklaşık 7-8 aylık süreçte projeyi nihayete erdirmek üzere çalışmalarımıza hız kesmeden devam edilecek.

Bu süreçte test işlemleri ve iş analizi için ekip içerisinde 1-2 kişilik bir destek alınması projenin daha verimli ilerlemesi için yararlı olabilecektir.

Projenin bu güne kadar geliştirildiği hali <http://tasarimproje.abdulkadiratar.com.tr/public/> linkinden incelenmek üzere yayına alınmış, kaynak kodları “[akadiratar](https://github.com/akadiratar)” kullanıcı adıyla Github üzerine yüklenmiş, <https://github.com/akadiratar/portal/> adresinden public olarak erişilebilir haldedir.

Projenin <http://tasarimproje.abdulkadiratar.com.tr/public/> adresinden incelenmesi sırasında “**test**” kullanıcı için adalet şifresi ile giriş yapılabilir. Ayrıntılı bilgi https://github.com/akadiratar/portal/beni_oku.txt içerisinde yer almaktadır.

ÖZGEÇMİŞ

Abdulkadir ATAR, 18.02.1991' Kadıköy/İSTANBUL'da doğdu. İlk, orta ve eğitimini Pendik/İSTANBUL'da tamamladıktan sonra Kütahya Dumlupınar Üniversitesinde Seramik ve Malzeme Bilimi Mühendisliği okuduğu sırada T.C. Adalet Bakanlığı'nda Zabıt Katibi olarak iş hayatında başladı. Lisans eğitimini çalışma hayatı ile birlikte tamamladıktan sonra çocukluk hayali olan ve amatör olarak gönül verdiği bilişim dünyasındaki eğitimine Sakarya Üniversitesi Bilgisayar Programcılığı ile başlayarak ardından da Sakarya Üniversitesi Bilgisayar Mühendisliği bölümünde devam etti. Eğitimi sırasında sırası ile T.C. Adalet Bakanlığı Donanım Tamir Bakım şefliği, T.C. Adalet Bakanlığı Adli Destek Şube Şefliği(SEGBİS), T.C. Adalet Bakanlığı Yazılım Geliştirme(UYAP) şefliğinde stajlarını yapmıştır.

T.C. Adalet Bakanlığı İstanbul Anadolu Adliyesi'nde Bilgisayar Teknikeri unvanı ile tedviren Ağ-Sistem, SEGBİS ve Proje Yönetimi birim şefliği görevi ile çalışma hayatına devam etmekteyken çalıştığı kurumda bu güne kadar Kartlı Geçiş Sistemi, Büro Personeli Özlük Takip Sistemi, Envanter Takip Sistemi vb. küçük çaplı Web tabanlı yazılımlar geliştirmiştir.

Lisans eğitiminin ardından akademik kariyeri için çalışmalar yapmaya devam etmesinin yanında, çalıştığı kurumda Mühendis unvanı sınavlarına hazırlık yapmaktadır.

Evli ve 1 çocuk babasıdır.

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : Kuruluşlara Özel Amaçlı Yazılım Üretilmesi

ÖĞRENCİLER : B181210389 Abdulkadir ATAR

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN : Öğr.Gör.Dr. Yüksel YURTAY

DANIŞMAN İMZASI: