

**T.C.
KASTAMONU ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ORGANİZASYON YÖNETİMİ İÇİN WEB TABANLI ÖRNEK
BİR ÇALIŞMA**

Ahmed M. M. BUKER

**Danışman
Jüri Üyesi
Jüri Üyesi**

**Dr. Öğr. Üyesi Yasemin GÜLTEPE
Dr. Öğr. Üyesi Can Doğan VURDU
Dr. Öğr. Üyesi Javad RAHEBI**

**YÜKSEK LİSANS TEZİ
MALZEME BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI**

KASTAMONU – 2019

TEZ ONAYI

Ahmed Mohammed Mokhtar BUKER tarafından hazırlanan "**Organizasyon Yönetimi İçin Web Tabanlı Örnek Bir Çalışma**" adlı tez çalışması aşağıdaki jüri üyeleri önünde savunulmuş ve oy birliği ile Kastamonu Üniversitesi Fen Bilimleri Enstitüsü **Malzeme Bilimi ve Mühendisliği Ana Bilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman

Dr. Öğr. Üyesi Yasemin GÜLTEPE
Kastamonu Üniversitesi

.....


Jüri Üyesi

Dr. Öğr. Üyesi Can Doğan VURDU
Kastamonu Üniversitesi

.....


Jüri Üyesi

Dr. Öğr. Üyesi Javad RAHEBI
Türk Hava Kurumu Üniversitesi

.....


12/04/2019

Enstitü Müdürü

Prof. Dr. Hasbi YAPRAK

.....


TAAHHÜTNAME

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildirir ve taahhüt ederim.

Ahmed M. M. BUKER



ÖZET

Yüksek Lisans Tezi

ORGANİZASYON YÖNETİMİ İÇİN WEB TABANLI ÖRNEK BİR ÇALIŞMA

Ahmed M. M. BUKER

Kastamonu Üniversitesi

Fen Bilimleri Enstitüsü

Malzeme Bilimi ve Mühendisliği Ana Bilim Dalı

Danışman: Dr. Öğr. Üyesi Yasemin GÜLTEPE

Sürekli olarak değişen ve gelişen dünyamızda teknoloji, hayatımızın vazgeçilmez bir unsuru haline gelmiştir. Gelişen teknolojiye en büyük örneklerden biri ise gerek iş ortamımızda gerekse özel yaşamımızda önemli ölçüde yer tutan internettir. İnternet sektörünün hızla gelişmesiyle beraber web teknolojisi de oldukça önem kazanmıştır. Web uygulamaları ile yenilenen dünya, insanlara hız, teknoloji ve yenilikler kazandırmaktadır. Web tabanlı sistemler günümüzde birçok alanda uygulanmaktadır. Libya’da aile en önemli kültür unsurlarından biri sayılmaktadır. Düğün merasimleri ise bir evladın kendi ailesini oluşturmada ilk adımdır. Bu nedenle sosyolojik açıdan son derece öneme sahip olan düğün, aynı zamanda eğlence kültürünün bir parçası olarak, insanların güzel zaman geçirdikleri şenlik alanlarına dönüşmektedir. Düğün planlama en zor ve karmaşık organizasyonlardan biridir. Düğün planlama çok fazla alt başlıklardan oluşmaktadır. En temel problem, düğün salonunu ayarlamaktır. Bunun için insanlar en iyi seçeneği bulmak için uzun mesafeler kat ederek bir çok mekanı kontrol etmesi gerekmektedir. Çünkü bu salonlar/mağazalar için uzaktan iletişim kuracakları herhangi bir çevrimiçi sistemleri bulunmamaktadır. Bu yüzden düğün hazırlıkları keyifli olduğu kadar detaylı, yorucu ve stresli bir süreçtir. Ayrıca maliyet ve zaman gerektiren bir işlemdir.

Bu tez çalışmasında, Libya’da organizasyon sistem eksikliklerine yönelik genel çözüm olarak web tabanlı yazılım geliştirilmiştir. Yapılan bu çalışmanın amacı; web tabanlı bir akıllı düğün organizasyon sistemini gerçekleştirmektir. Bu sistemi en değerli kılan şey, farklı ve dağınık konumlardan kullanılmalarıdır. Sistem, kullanıcıların internet olan her yerden erişim sağlayabildikleri bir sistemdir. Masaüstü yazılımlara nazaran güvenlik açıklarının takibi ve kontrolü daha rahat bir şekilde gerçekleştirilebilir. Geliştirilen web tabanlı akıllı düğün organizasyon sistemi, A’dan Z’ye tüm detayların (düğün salonu, kıyafetler, ikramlar, güzellik merkezleri, aksesuarlar, ses sistemi, gelin arabası, fotoğrafçı, vb.) gerçekleştirilmesi hedeflenmektedir. Geliştirilen web yazılım için PHP programlama dili, MySQL ilişkisel veritabanı ve tasarım için HTML, CSS dilleri kullanılmıştır.

Anahtar Kelimeler: Web tabanlı sistemler, organizasyon, MYSQL, PHP

2019, 57 sayfa

Bilim Kodu: 101

ABSTRACT

MSc. Thesis

A WEB-BASED SAMPLE STUDY FOR ORGANIZATION MANAGEMENT

Ahmed M. M. BUKER

Kastamonu University

Graduate School of Natural and Applied Sciences

Department of Material Science and Engineering

Supervisor: Assist. Prof. Dr. Yasemin GÜLTEPE

In our ever-changing and developing world, technology has become an indispensable element of our lives. One of the biggest examples of developing technology is the internet which has a significant place both in our business environment and in our private life. With the rapid development of the internet sector, web technology has gained importance. The renewed world with web applications gives people speed, technology and innovations. Web-based systems are currently being implemented in many areas. In Libya, the family is considered one of the most important cultural elements. Wedding ceremonies are the first step for a child to create his own family. Therefore, the wedding, which is of great importance in terms of sociology, is also transformed into a festive area where people have a good time as part of the entertainment culture. Wedding planning is one of the most difficult and complex organizations. Wedding planning consists of many sub-headings. The most basic problem is to set the wedding hall. To do this, people need to control a lot of places by taking long distances to find the best option. Because they do not have any online systems to communicate remotely for these halls/stores. That is why wedding preparations are as enjoyable as detailed, tiring and stressful. It is also a process that requires cost and time.

In this thesis, web based software has been developed as a general solution for organizational system deficiencies in Libya. The aim of this study is; is to realize a web based smart wedding organization system. What makes this system the most valuable is the use of different and diffuse positions. The system is a system where users can access from anywhere on the internet. The security vulnerabilities can be monitored and controlled more easily than the desktop software. The web-based intelligent wedding organization system, A-Z, is aimed at realizing all details (wedding hall, clothes, refreshments, beauty centers, accessories, sound system, bridal car, photographer, etc.). PHP programming language is used for web based system development. MySQL is used for relational database management. HTML and CSS languages were used for the design of the web-based system.

Key Words: Web-based systems, organization, MYSQL, PHP

2019, 57 pages

Science Code: 101

TEŞEKKÜR

Öncelikle bu tez konusu üzerinde bana çalışma imkanı sunan tez danışmanım Dr. Öğr. Üyesi Yasemin GÜLTEPE'e tez çalışmam süresince deneyimleri, bilgileri ve önerileriyle araştırma ve geliştirmeyi yönlendirmesi ve sağladığı kaynaklarla destek olmasından dolayı teşekkürü bir borç bilirim.

Tez izleme toplantılarında jüri olarak görev alan Dr. Öğr. Üyesi Can Doğan VURDU'a ve Dr. Öğr. Üyesi Javad RAHEBI'a değerli önerileri ve çalışmaya olan katkıları için teşekkürlerimi sunarım.

Ayrıca tez çalışmaları sırasında fikirlerine başvurduğum Kastamonu Üniversitesi Bilgisayar Mühendisliği'nin değerli öğretim üyelerine ve elemanlarına teşekkürlerimi sunarım.

Her zaman yanımda olan sevgili aileme ve yakınlarıma minnettar olduğumu belirtmek isterim.

Kastamonu Üniversitesinde aldığım eğitim için yaptıkları destekten dolayı Libya Hükümetine ve Türkiye'deki Libya Büyükelçiliğine teşekkür ederim.

Ahmed Mohamed Mokhtar BUKER
Kastamonu, Nisan, 2019

İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR DİZİNİ	ix
ŞEKİLLER DİZİNİ.....	x
TABLOLAR DİZİNİ	xi
1. GİRİŞ	1
2. ALT YAPI VE İLGİLİ ÇALIŞMALAR.....	5
2.1. Alt Yapı	5
2.1.1. Web Tabanlı Uygulamalar.....	5
2.1.2. Web Uygulama Hataları	7
2.1.3. Web Tabanlı Uygulama Geliştirme Çatıları	7
2.1.3.1. <i>PHP</i>	7
2.1.3.2. <i>Java Servlets</i>	8
2.1.3.3. <i>Java Server Pages</i>	8
2.1.3.4. <i>JSF</i>	9
2.1.3.4. <i>CSS</i>	9
2.1.3.5. <i>HTTP</i>	10
2.1.2. MySQL	10
2.1.3. NodeMCU ESP8266.....	11
2.1.4. UML	11
2.2. İlgili Çalışmalar	12
3. WEB TABANLI ORGANİZASYON SİSTEMİNİN MODELLENMESİ ...	14
3.1. Sistemin Yapısı.....	14
3.2. Modelleme	16
3.2.1. Kullanıcı Bakışı	17
3.2.2. Kullanım Durumu Diyagramı ve Senaryoları.....	19
3.2.3. Yapısal Bakış	22
3.2.4. Davranış Bakışı.....	24
3.2.5. Ortam Bakışı.....	27
3.3. Veritabanı Tasarımı	28
3.3.1. Veritabanı Tabloları.....	28
4. UYGULAMA	41
4.1. Uygulama Dosyaları.....	41
4.2. Kayıt Yapma.....	43
4.3. Giriş Yapma	45
4.4. Gezinti ve Filtreleme Sonuçları	46

4.5. Rezervasyon	47
4.6. Rezervasyon Kontrolü	49
4.7. Profil Yönetimi	50
4.8. Ürün Yönetimi	51
4.9. Rezervasyon Yönetimi	51
4.10. Web Sunucu Olarak NodeMCU Kullanımı	52
4.11. SMS Yönetimi	53
5. SONUÇLAR	54
KAYNAKLAR	55
ÖZGEÇMİŞ	57



SİMGELER Ve KISALTMALAR DİZİNİ

API	Uygulama Programlama Arayüzü
BT	Bilişim Teknolojileri
CMS	İçerik Yönetim Sistemi
CSS	Basamaklı Stil Sayfaları
ERD	Varlık İlişki Diyagramı
HTML	Hiper Metin Transfer Protokolü
HTTP	Hiper Metin Aktarım Protokolü
IP	İnternet Protokolü
JS	JavaScript
PDA	Kişisel Dijital Yardımcı
PHP	Hiper Metin Ön İşleme
SMS	Kısa Mesaj Servisi
UI	Kullanıcı Arayüzü
UML	Birleşik Modelleme Dili
URI	Standart Kaynak Tanımlayıcısı
URL	Standart Kaynak Bulucu
VB	Visual Basic
WWW	Dünya Çapında Web

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. Bir Web Uygulamasının Çalışma Prensibi	6
Şekil 2.2. Web'in Gelişimi.....	6
Şekil 3.1. Sistem Uygulaması	15
Şekil 3.2. Genel Kullanıcı İçin Bir Senaryo.....	19
Şekil 3.3. Editör İçin Bir Senaryo	20
Şekil 3.4. Yönetici İçin Bir Senaryo	21
Şekil 3.5. Süper Yönetici İçin Bir Senaryo	22
Şekil 3.6. Web Tabanlı Organizasyon Sistemi Sınıf Diyagramı.....	23
Şekil 3.7. Yöneticinin Sisteme Giriş Durum Diyagramı.....	24
Şekil 3.8. Editörün Sisteme Giriş Durum Diyagramı	25
Şekil 3.9. Süper Yöneticinin Sisteme Giriş Durum Diyagramı	26
Şekil 3.10. Kullanıcının Sisteme Giriş Durum Diyagramı	27
Şekil 3.11. Varlık İlişki Diyagramı	40
Şekil 4.1. Uygulama Dosyaları	41
Şekil 4.2. Sisteme Kayıt Ekranı	44
Şekil 4.3. Kayıt Yapma İşlevi İçin Örnek Kod	44
Şekil 4.4. Giriş Yapma İşlevi İçin Örnek Kod.....	45
Şekil 4.5. Genel Bir Görünüm.....	46
Şekil 4.6. İşlem Sürecine Ait Bir İş Akışı.....	47
Şekil 4.7. Rezervasyon İşlemi.....	48
Şekil 4.8. Rezervasyon Bilgileri.....	48
Şekil 4.9. Rezervasyon İşlem Kod Örneği	49
Şekil 4.10. Rezervasyon Durumu.....	49
Şekil 4.11. Rezervasyon İptali	50
Şekil 4.12. Servislerin Yönetimi	50
Şekil 4.13. Ücret Yönetimi	51
Şekil 4.14. Ürün Yönetimi	51
Şekil 4.15. NodeMCU Kullanıcı Arayüzü	52

TABLÖLAR DİZİNİ

	Sayfa
Tablo 3.1. Kullanıcı Hesap Tablosu.....	29
Tablo 3.2. Düğün Salonları Tablosu	30
Tablo 3.3. Düğün Salonları ve Şubeleri Tablosu	31
Tablo 3.4. Servisler Tablosu	32
Tablo 3.5. Mağazalar Tablosu.....	33
Tablo 3.6. Kategoriler Tablosu	34
Tablo 3.7. Ürünler Tablosu	34
Tablo 3.8. Adres Tablosu	35
Tablo 3.9. Mağaza/Salon Tam Adresleri Tablosu	35
Tablo 3.10. Fotoğraf Tablosu.....	36
Tablo 3.11. Kullanıcı Sepeti Tablosu.....	37
Tablo 3.12. Rezervasyon Tablosu.....	38

1. GİRİŞ

Günümüz işletmelerinin iş süreçlerini sorunsuz ve rekabet şartlarına uygun şekilde yürütmeleri Bilişim Teknolojileri (BT) bölümlerinin başarısı ile doğru orantılı olduğu görülmektedir. Servis yönetimi mantığından yoksun işletmelerde sunulan hizmetlerin kalitesi ve devamlılığı tartışmalıdır. Belirli bir sürece göre hareket edilmeden verilen hizmetlerde kalite ve süreklilik sağlamak mümkün değildir (Demirel ve Karaağaç, 2014). Bilgisayar teknolojisindeki gelişmeler, beraberinde BT'ni, iş dünyası için en önemli destek unsuru haline getirmekte ve getirmeye de devam etmektedir (Esener, 2005). Günümüzde BT servislerinin yönetimi, organizasyonların stratejik hedeflerine ulaşmaları için bir gerekliliktir (Yılmaz, 2014). Günümüzde artık insanların günlük yaşamında veya iş hayatında ihtiyaç duydukları bilgilere hızlı ve kolay erişebilmeleri ve bu bilgilerden verimli sonuçlar üretebilmeleri çok önemlidir.

İşte bu noktada BT'nin sunduğu imkanlar bu gibi ihtiyaçlara çözüm sunabilmektedir. Müşteriler ve işletmeler arasındaki, işletmeler ve diğer işletmeler arasındaki, çalışanlar ve işletmeler arasındaki, işletmeler ve devlet arasındaki dijital etkileşim her geçen gün artmakta ve artarak devam etmektedir. Bu dijital etkileşim işletmelerin BT bütçelerini ve BT yatırımlarını artırmaktadır.

Bu kadar büyük verilerin depolanması, işlenmesi ve bu verilenlerden anlamlı sonuçlar elde edilmesi bilinen yöntemlerin ötesinde bir teknik ve donanım çözümü gerektirmektedir (Gökşen ve Aşan, 2015). Bu tespitten yola çıkarak, organizasyon çalışanlarının BT bölümlerinden talep ettikleri iş isteklerini yönetilebilmesi, bu isteklerin bir veritabanında toplanabilmesini gerektirir. Toplanan bu verilerin bilgiye dönüştürülmesi ve bilginin analiz edilerek yorumlanabilmesi içinse bilgi sistemlerine ihtiyaç duyulmaktadır (Druzdzet ve Flynn, 2010; Hersh, 1999).

Web tabanlı bir uygulama, bir web sitesini arayüz veya ön uç olarak kullanan herhangi bir uygulamadır. Kullanıcılar, standart bir tarayıcı kullanarak internete bağlı herhangi bir bilgisayardan uygulamaya kolayca erişebilirler. Web tabanlı uygulamalarının avantajlarını aşağıdaki alt başlıklarda tanımlayabiliriz (Afroz, Rani & Priyadarshini, 2011; Bruno, Tam & Thom, 2005; Yao, 2008):

Az maliyet: Web tabanlı uygulamalarla, kullanıcılar sisteme tek bir ortam üzerinden (web tarayıcısı) erişebilirler. Uygulama ile kullanıcı etkileşiminin farklı web tarayıcılarında iyice test edilmesi gerekirken, uygulamanın kendisinin yalnızca tek bir işletim sistemi için geliştirilmesi gerekir.

Kolay erişim: Geleneksel uygulamaların aksine, web sistemlerine her zaman, her yerden ve internet bağlantısı olan herhangi bir bilgisayar üzerinden erişilebilir. Bu, kullanıcıyı uygulamaya nereden ve ne zaman eriştiğinden sorumlu tutar.

Kolay özelleştirilebilir: Web tabanlı uygulamaların kullanıcı arayüzü, masaüstü uygulamalarda olduğu gibi özelleştirilmesi daha kolaydır. Bu, uygulamanın görünüm ve izlenimini güncellemeyi veya bilgi sunumunu farklı kullanıcı gruplarına göre özelleştirmeyi kolaylaştırır.

Birçok cihaz için erişilebilir: Kullanıcı grupları için özelleştirilebilir olmasının yanı sıra, içerik, internete bağlı herhangi bir cihazda kullanılmak üzere de özelleştirilebilir. Bu, Kişisel Dijital Yardımcı (Personel Dijital Assitant, PDA)'ların, cep telefonlarının ve tabletlerin beğenilerini içerir. Bu, kullanıcının kendilerine uygun bir şekilde bilgi alma ve bilgi edinme yeteneğini de genişletir. Bu şekilde, güncel bilgiler her zaman ihtiyacı olan kişilerin parmaklarının ucundadır.

Birlikte çalışabilirlik: Web uygulamaları arasında izole edilmiş masaüstü sistemlerine göre çok daha fazla bir birlikte çalışabilirlik düzeyi elde etmek mümkündür. Örneğin, web tabanlı bir alışveriş sepeti sistemini web tabanlı bir muhasebe paketi ile entegre etmek, iki özel sistemin birbiriyle konuşmasını sağlamaktan daha kolaydır. Bu nedenle, web tabanlı mimari, kurumsal sistemleri hızla entegre etmeyi, iş akışını ve diğer iş süreçlerini iyileştirmeyi mümkün kılar. İnternet teknolojilerinden faydalanarak değişen pazar taleplerine göre değiştirilebilecek esnek ve uyarlanabilir bir iş modeli elde edersiniz.

Kolay kurulum ve bakım: Web tabanlı yaklaşımla kurulum ve bakım da daha az karmaşık hale gelir. Ana sunucuya yeni bir sürüm veya yükseltme yüklendikten sonra, tüm kullanıcılar doğrudan erişebilir ve her bir potansiyel kullanıcının bilgisayarını yükseltmesine gerek yoktur.

Artan iş yüküne uyarlanabilir: İşlemci kapasitesini artırmak, web tabanlı uygulamalarla çok daha kolay bir işlem haline gelir. Bir uygulamanın görevleri gerçekleştirmek için daha fazla güç gerektirmesi durumunda, yalnızca sunucu donanımının yükseltilmesi gerekir. Web tabanlı yazılımın kapasitesi "kümeleme" veya yazılımı aynı anda birkaç sunucuda çalıştırarak artırılabilir. İş yükü arttıkça, sisteme kolayca yeni sunucular eklenebilir.

Güvenlik: Web tabanlı uygulamalar genellikle deneyimli sunucu yöneticileri tarafından izlenen ve bakımı yapılan özel sunucularda dağıtılır. Bu, masaüstü uygulamalarında olduğu gibi yüzlerce hatta binlerce istemci bilgisayarını izlemekten çok daha etkilidir.

Web tabanlı organizasyon sistemi günümüzde birçok alanda uygulanmaktadır. Organizasyonlar insanların hayatında önemli yer tutmaktadır. Düğün ve nişan gibi mutluluk verici organizasyonlar toplumda birlik ve beraberliğin oluşması adına bir güven ve dayanışma ortamı oluşturur. Bu nedenle düğün salonu, etkinlik ve alışverişleri yönetmek önemlidir. Bu ihtiyaçları kolayca karşılayabilecek web tabanlı akıllı düğün salonu organizasyon sistemleri gerekmektedir.

Bu tez çalışmasında örnek uygulama olarak Libya’da bir düğün organizasyon sistemi ile ilgili A’dan Z’ye tüm detayları (düğün salonu, kıyafetler, ikramlar, güzellik merkezleri, aksesuarlar, ses sistemi, fotoğrafçı, vb.) tam olarak bütünleşik bir hale getirilmesi hedeflenmektedir.

Tez çalışmasında hızlı uygulama geliştirme araçlarından yararlanarak web tabanlı akıllı bir uygulama geliştirilmiştir. Geliştirilen Web tabanlı Akıllı Düğün Salonu Düzenleme Uygulaması (ADSDU), müşterilerin ve çalışanların işlerini kolaylaştıran, standart ofis uygulamalarının ötesinde hizmet sunan, bilgiyi etkili yöneten ve bilgilerden anlamlı sonuçlar üretebilen bütünleşik bir yazılımdır. Ayrıca, internet erişimi olan herhangi bir yerden herhangi bir mobil cihaz veya bilgisayar tarafından kolayca rezervasyon yapılabilir. Müşteriler, çeşitli düğün salonları, gıda fabrikaları ve giyim mağazalarından birçok ürün, hizmet ve fiyat teklifini görebilirler. Geliştirilen sistem kullanıcıya, tüm ürün ve hizmetlerin sağlayıcılar tarafından düzenli takibi ve denetimine bağlı olarak, rezervasyon yapabilme, rezervasyon iptali

ve rezervasyon takibi ve ödeme için hızlı ve kolay işlem saęlayan arayüzler sunmaktadır.

Birinci bölümde tez çalışmasının fikir temelleri ve yapılan çalışmanın kısaca ifadesini içeren giriş bölümü bulunmaktadır.

Tezin ikinci bölümünde çalışma için gerekli teknolojiler ve bu alanda daha önce gerçekleştirilmiş çalışmalar hakkında bilgi verilmektedir. Tezin kapsamına uygun olarak teknolojiler kısmında web tabanlı uygulamalar, web uygulama hataları, web tabanlı uygulama geliştirme çatıları, MYSQL, NodMCU ESP8266 ve UML anlatılmıştır. Literatürde bu alanda gerçekleştirilen çalışmalar aktarılmıştır. Son olarak literatürde yer alan bu çalışmalara ait genel bir değerlendirme gerçekleştirilmiş ve bu tez kapsamında ortaya konan çalışmanın bu çalışmalara göre farkları anlatılmıştır.

Sistemin işlevsel gereksinimleri tanımlanmış, kullanıcıları ve veritabanı ile uygulama arayüzlerinin tasarımı tezin üçüncü bölümünde anlatılmıştır.

Tezin 4. bölümünde web tabanlı akıllı sistemin oluşturma aşamalarının sahip olduęu üstünlükler ve sınırlılıklar tartışılarak Libya’da bir Düğün Salonu Düzenleme sistemine ait bir uygulama geliştirilmiştir.

Tezin 5. ve son bölümünde ise sonuç ve çalışmayı ileriye götürebilecek öneriler yer almaktadır.

2. ALT YAPI VE İLGİLİ ÇALIŞMALAR

Bu bölümde tez çalışması için gereken alt yapı bilgileri ve daha önce bu konuda gerçekleştirilmiş olan çalışmalar yer almaktadır. Alt yapı başlıklı alt bölümünde web tabanlı uygulamalar, web uygulama hataları, PHP, HTTP, Java Servlets, JSP, JSF, CSS, MYSQL, NodeMCU ESP8266, UML hakkında bilgiler yer almaktadır. İkinci alt bölümde ise web tabanlı akıllı düğün/etkinlik gerçekleştirme sistemleri ile ilgili diğer çalışmalar hakkında bilgi verilmiştir.

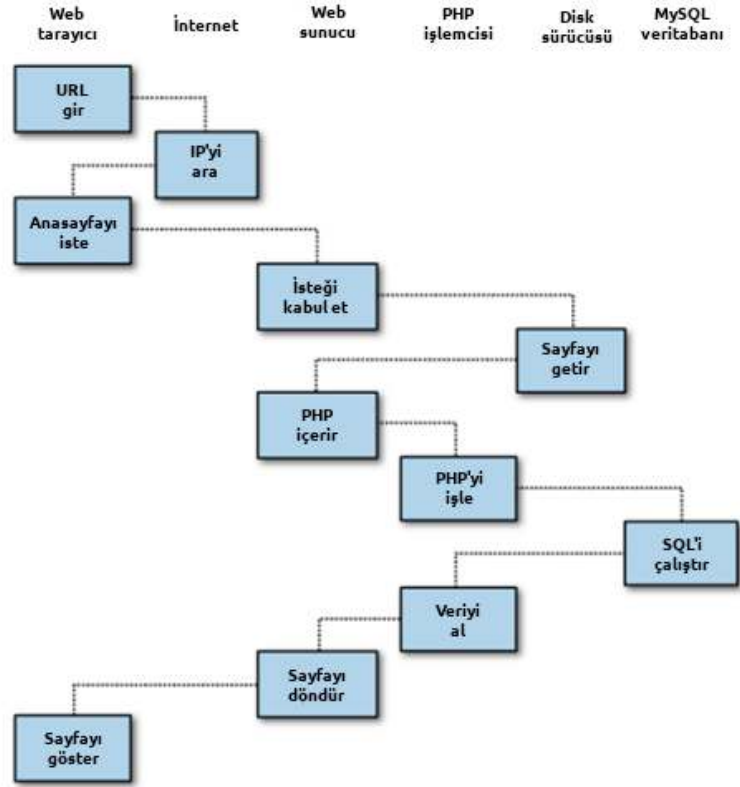
2.1. Alt Yapı

2.1.1. Web Tabanlı Uygulamalar

Web uygulamalarından önce server-client yazılımlarının kullanıldığı sistemler bulunmaktaydı. Bir uygulamanın kullanılması, sunucu üzerinde çalışan bir programa, istemci bilgisayara yüklenen özel bir program sayesinde ulaşılması şeklinde gerçekleşmektedir. Ancak web uygulamaları ile sadece sunucu taraflı bir uygulama ile internet ya da intranet üzerinden browser aracılığıyla buna ulaşan bir istemci yeterli olmaktadır. İstemci bilgisayar üzerinden yapılan her istek, sunucu tarafından yorumlanır, sonuç HTML kodları içeren dinamik sayfa yardımıyla tarayıcıya gönderilir (Schwarz, 2013).

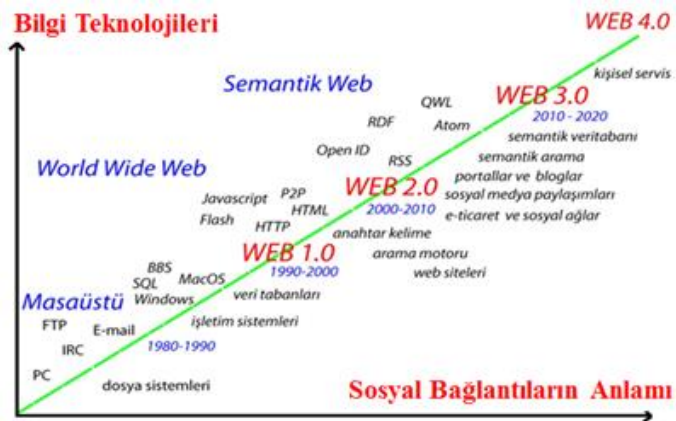
Şekil 2.1.'de statik bir web sayfası isteğinde bulunan web tarayıcıya, web sunucu hizmeti tarafından ne şekilde yanıt verildiği gösterilmiştir. Web tarayıcısı tarafından istenen statik sayfa, sunucu tarafından bulunur ve gönderilir.

Web tabanlı uygulamalar, yalnızca programlama dilleri yardımıyla oluşturulurlar. Bir web uygulaması geliştirmek için kullanabileceğiniz birçok farklı programlama dili vardır. Her dilin işlevi, yapısı ve kullanım özellikleri ayrı olmasına karşın bir yazılım birçok programlama dili kullanılarak aynı işlevlere sahip olarak meydana getirilebilir.



Şekil 2.1. Bir web uygulamasının çalışma prensibi

Web'in gelişimi göz önüne alındığında kişisel bilgisayarlarla, FTP, e-posta, dosya sistemleri, HTML gibi teknolojilerle başlayan iletişim sürecinin, RSS, atom, flash, java, semantik gibi yeni teknolojilerle zenginleştiği, web ortamında uygulamaların birbiriyle iletişim kurduğu dolayısıyla web'in çok daha akıllı ve karmaşık bir yapıya kavuştuğu söylenebilir. Şekil 2.2.'de web'in gelişim süreci gösterilmiştir (Radar Networks ve Nova Spivack, 2007).



Şekil 2.2. Web'in gelişimi

2.1.2. Web Uygulama Hataları

Web uygulama hatalarından iki hata grubu doğrudan müşteri etkileşimleriyle ilgilidir. Bu nedenle web uygulama geliştiricileri için özel bir öneme sahiptir. Bunlar: Çıktı doğruluğu ve güvenlik açıklarıdır. Geçerli web uygulaması çıktısı, programcıya standartlara uygun tüm web tarayıcılarının belgeleri tutarlı bir şekilde vereceğini garanti eder. Güvenlik açıkları, kötü niyetli müşterilerin sisteme saldırmasına ve istenmeyen ayrıcalıklar veya bilgi edinmelerine izin verebilir (Chaudhari ve Vaidya, 2014).

Web uygulaması hatalarına yönelik çözümler üç kategoriye ayrılır:

- 1) Programın program çalıştırılmadan önce bir araç tarafından analiz edildiği statik çözümler,
- 2) Çalışma zamanında hatanın tespit edilip önlendiği dinamik çözümler,
- 3) Dilin bulunduğu çerçeve çözümleri veya çerçeve, haksız yere savunmasız bir program yazmak imkansız olacak şekilde inşa edilir. Yani, çerçeve varsayılan olarak güvenlidir.

2.1.3. Web Tabanlı Uygulama Geliştirme Çatıları

Web uygulama iskeleti ya da web uygulama çatısı, dinamik web siteleri, web uygulamaları ve web servislerinin geliştirilmesini desteklemek amacıyla tasarlanmış yazılım iskeletleridir.

2.1.3.1. *PHP*

PHP (Hypertext Preprocessor, Hiper Metin Ön İşleme), yaygın olarak kullanılan web tabanlı ve açık kaynak kodlu bir programlama dilidir. Sunucu tarafında çalışan ve HTML'e gömülebilen PHP, özellikle web uygulamaları geliştirme ve dinamik web sayfaları hazırlamak için kullanılır.

PHP yazılım dilinin birçok avantajı bulunmaktadır. Avantajlarını aşağıdaki gibi sıralayabiliriz:

- Ücretsizdir.
- Açık kodlu olması sebebiyle sürekli geliştirilir.
- Çok geniş bir PHP kütüphanesi bulunmaktadır.
- Tüm işletim sistemleri içerisinde çalışabilir.
- MySQL, MSSQL, Oracle gibi veritabanları yanı sıra birçok veirtabanı ile çalışabilir.
- Çalışma mantığı karmaşık değildir. PHP yapısının ve kullanımının karmaşık olmaması da onun kolay şekilde öğrenilmesini sağlar.
- Kodları yazmak için basit bir kelime işlem programı yeterlidir.

PHP programlama dilinin kullanım alanları çok yaygındır. PHP, web sitelerinin arka planlarında yani sunucu tarafında çalışarak bazı işlemleri yerine getirir. PHP ile yazılan işlemler tarayıcılar (browserler) tarafından yorumlanır ve bazı sonuçlar üretilir. Bu sonuçlarla da HTML, CSS ya da JavaScript tarzı betik dillerin daha görsel ve kullanışlı olması sağlanır. Bu şekilde kodlanan web sitelerine de dinamik web siteleri denilir. Web tabanlı veritabanı uygulamaları, e-ticaret siteleri, e-posta uygulamaları, kullanıcı etkileşimli web siteleri, vb. kullanım alanları bulunmaktadır.

2.1.3.2. *Java servlets*

Java Sunucuları bir J2EE özelliğidir (Mordani, 2009). Servlet çerçevesi, web uygulamalarını Java programlama dilinde uygulamak için temel bir API sağlar. Java Servlets, GET ve POST istek yöntemlerinin iki farklı Java yöntemine (sırasıyla doGet ve doPost) ayrılmasını sağlar. HTTP spesifikasyonuna göre GET talepleri önbelleklenebilir ve yan etkileri olamaz (Fielding, 2009). Servlet çerçevesi bunu zorlamaz ancak geçerli sunucu durumunun görünümünü doGet yönteminden döndüren ve durumu doPost yöntemiyle güncelleyebilecek istekler alan bir Servlet yazmayı mümkün kılar.

2.1.3.3. *Java server pages*

Java Server Pages (JSP), dinamik içerik içeren web sayfaları geliştirmek için bir teknolojidir. Her zaman aynı kalan statik içeriği içeren düz bir HTML sayfasının

aksine, bir JSP sayfası, kullanıcının kimliği, kullanıcının tarayıcı türü, kullanıcı tarafından sağlanan bilgiler ve kullanıcı tarafından yapılan seçimler dahil olmak üzere herhangi bir sayıda değişken öğeye dayalı olarak içeriğini değiştirebilir (Bergsten, 2002).

Bir JSP sayfası, normal bir web sayfası gibi, HTML etiketleri gibi standart biçimlendirme dili öğelerini içerir. Ancak, bir JSP sayfası ayrıca sunucunun sayfaya dinamik içerik eklemesini sağlayan özel JSP öğeleri de içerir. JSP öğeleri, bir veritabanından bilgi almak veya kullanıcı tercihlerini kaydetmek gibi çeşitli amaçlar için kullanılabilir. Bir kullanıcı bir JSP sayfası istediğinde, sunucu JSP öğelerini yürütür, sonuçları sayfanın statik bölümleriyle birleştirir ve dinamik olarak oluşturulmuş sayfayı tarayıcıya geri gönderir (Pekgöz, 2013).

2.1.3.4. JSF

Java Server Faces (JSF), bileşen tabanlı, olaya yönelik web arayüzleri oluşturmak için Java standart teknolojisidir. JSP gibi JSF’de sunucu tarafındaki verilere ve mantığa erişim sağlar. Temel olarak sunucu tarafı özelliklere sahip bir HTML sayfası olan JSP’den farklı olarak JSF, mantıksal bir ağaçtaki resmi bileşenleri temsil eden bir XML belgesidir. JSF bileşenleri, HTML’den bağımsız olan ve uzak API’lere ve veritabanlarına erişim de dahil olmak üzere tüm Java yeteneklerine sahip Java nesneleri tarafından desteklenir (Sathyaseelan ve Cordova, 2016).

2.1.3.5. CSS

Basamaklı Stil Şablonu (Cascading Style Sheets, CSS), HTML etiketleri kullanılarak oluşturulan web site görüntülerini şekillendirir. Tüm dökümanın stilini ayarlayabileceğimiz gibi bir linkin, bir resmin veya bir yazınında stilleri örnek olarak rengi, yazı türü css aracılığıyla ayarlanabilir. CSS eklemenin üç farklı yolu bulunmaktadır. Bunlar; Dış stil sayfası, dahili stil sayfası ve satır içi stildir.

CSS kodları HTML kodlarının içine yazılabilir. Fakat bunun haricinde dışta bir css dosyası oluşturabilir ve bu css dosyası html sayfanıza entegre edilebilir.

2.1.3.4. HTTP

HTTP, World Wide Web (Dünya Çapında Web, WWW) tarafından iletişim için kullanılan ana protokoldür. HTTP, mesajların internet üzerinden nasıl biçimlendirildiğini ve iletildiğini tanımlar. HTTP protokolü, istemci sunucu modeline dayanır. Bir tarayıcı istemci olarak kabul edilir, çünkü sunucuya istek göndermek için kullanılır. Sunucu daha sonra yanıtı istemciye geri gönderir. Sunucunun isteği dinlemesi için varsayılan bağlantı noktası 80'dir (Singh ve Kumar, 2016). HTTP tarafından kullanılan ana yöntemler GET, HEAD ve POST'tur.

GET: HTTP tarafından kullanılan en yaygın yöntemdir. İstenilen bilgileri almak için kullanılır. İstenen dosya bir HTML dosyası ise içeriği tarayıcı tarafında gösterilecektir. İstenen dosya dinamik bir dosyaysa, sunucu önce bu dosyayı işler, komutlarını çalıştırır ve son olarak bu komutun çıktısı talep eden tarayıcıya gönderilir.

HEAD: Bu yöntem neredeyse GET yöntemine benzer, ancak istenen verileri döndürmez. Başlık bölümünü, durum satırını ve sunucu yanıt kodunu aktarmak için kullanılır.

POST: Bu yöntem sunucuya veri göndermek ve daha sonra üzerinde hareket etmek için kullanılır. POST yöntemleri, sunucu tarafı komut dosyası kullanıldığında kullanılır.

2.1.2. MySQL

MySQL, çoklu iş parçacıklı, çok kullanıcı, hızlı ve sağlam bir veritabanı yönetim sistemidir. MySQL, UNIX, Linux, OS/2 ve Windows platformlarında kullanılabilir. Güçlü bir veritabanı yönetim sistemi olan MySQL veritabanı gerektiren hemen hemen her ortamda rahatlıkla kullanılabilir. Ama özellikle web sunucularında en çok kullanılan veritabanıdır. ASP, PHP gibi birçok web programlama dili ile kullanılabilir.

Açık kaynaklı teknolojiler çok popüler ve güvenilirlerdir. Bu tez çalışmasında, PHP ve MySQL kullanımı tercih edilmiştir. Günümüzde çok yaygın kullanılan araçlardır. Bununla birlikte, açık kaynak olmaları birçok avantajı birlikte getirmektedir. Böylelikle programlama ekipleri tarafından, istedikleri ve ihtiyaç duydukları özellikleri yazabilen, herkesin görmesi ve değiştirebilmesi için orijinal kodla birlikte geliştirildiği anlamına geldiği söylenebilir. En büyük avantajlarından birisi de ücretsiz olmasıdır (Nixon, 2012).

2.1.3. NodeMCU ESP8266

Açık kaynak kodlu bir kontrol ünitesidir. Geniş bir kullanım alanı olan ünite; “nesnelerin interneti” dediğimiz alanda kullanılarak birçok sorunun çözülmesini sağlar. ESP8266’nın uygulama yönünden en büyük avantajı çevrimiçi veri çekebilme yani istemci özelliğidir.

Bu tez çalışmasında NODEMCU ESP8266 kartıyla wi-fi mesaj gönderme uygulaması gerçekleştirilmiştir. Web tarayıcı üzerinden gönderilen yazılar anlık olarak OLED ekran üzerinde gösterilmektedir. OLED’in temel prensibi; ekran üzerindeki her pikselin kendi ışık kaynağına sahip olmasıdır.

2.1.4. UML

Birleşik Modelleme Dili (Unified Modelling Language, UML), yazılımın bileşenlerinin ve bunlar arasındaki ilişkilerin görsel olarak ifade edilmesini sağlayan bir standarttır. UML, bir yazılım geliştirme sürecinin tamamını kapsar.

UML yazılım ürünlerinde dizayn uygulamalarında artarak kullanılmaktadır. UML’nin bir dizayn dili olarak kullanmanın ana faydalarından biri başlangıç aşamalarından uygulamaya kadar baştan sona kullanılabilmesidir.

UML’nin özelliklerini kullanarak bir sistemin dinamik davranışları modellenabilir iş akışları değerlendirilebilir bu yolla geliştirme süreci kendiliğinden tanımlanmış olmaktadır. UML 4+1 görünüm modeli, sistemin yazılım yaşam döngüsündeki

görünüşünü anlatmaktadır. Her görünüm sistemin nasıl modelleneceğini gösterir (Kruchten, 1995).

2.2. İlgili Çalışmalar

İnternet teknolojilerinin ve sanal piyasanın büyük yükseliş gösterdiği açık bir şekilde karşımıza çıkmaktadır. Bu yükselişte web tabanlı yazılımların önemi çok büyük olduğu gibi, web tabanlı yazılım uygulamalarının sunduğu fırsatlar da çok fazladır. Web tabanlı sistem ihtiyaçlarının arttığı bu dönemde farklı alanlarda bir çok proje geliştirilmiştir. Örnek olarak web tabanlı rezervasyon sistemi, uçuş rezervasyonu, otel rezervasyonu, yemek siparişi ve çevrimiçi e-ticaret verilebilir.

Suudi Arabistan'da bir üniversitede konferans salonları, tiyatro, yüzme havuzları ve spor salonları gibi üniversitenin tesislerinin dengesiz kullanımını ve üniversitenin toplum hizmetlerinde rolünün bir parçası olarak çevrimiçi sosyal tesis rezervasyon sistemi geliştirilmiştir. Bu sistemin temel işlevleri, rezervasyon yapısına dayanmaktadır. Sistem, UML (Birleşik Modelleme Dili), MySQL ve Microsoft firması tarafından programlama dili üzerine geliştirilmiş nesne tabanlı ve görsel bir programlama dili olan Visual Basic (VB) programlama dili kullanılarak gerçekleştirilmiştir (Alkhaldi vd., 2018).

Başka bir akademik çalışmada ise futbol maçları için halı saha organizasyon sisteminin oluşturulmuştur. Bu sistemde, Java Çatısı (Framework) Spring Çatısı kullanılmıştır. Bu sistemin temel amacı, futbol oyun organizatörünün yeni maçlar oluşturmasını, maç arama, maç yapılacak sahanın kiralanmasını kapsayacak zaman, tarih, yer ve maliyet hakkında futbol takımında yer alan oyunculara bildirimler gönderilmesini sağlayan futbol maçlarını düzenlemektir. Bu temel amaca ulaşmak futbol oynamak isteyen kişiler için web tabanlı sistem sağlanmaktadır (Kosariiev, 2018).

Baig, Muzamil ve Dalvi (2016) çalışmasında bir web tabanlı bir ev otomasyonu geliştirilmiştir. Kullanıcıların web arayüzlerini kullanarak yerel olarak veya uzaktan cihazları yönetmelerine ve kontrollerine olanak sağlamaktadır. Ev ortamındaki

cihazların kontrolü, wi-fi cihaz aracılığıyla yapılmaktadır (Kishore, Veeramanikandasamy, Sambath & Veerakumar, 2017).

Yukarıda incelenen sistemler gözönünde bulundurularak Libya’da ihtiyaçlar üzerine bir inceleme yapıldığında web tabanlı düğün salonu düzenleme sisteminin olmadığı tespit edilmiştir. Bu tez çalışmasında bu ihtiyaçların giderilmesine yönelik olarak; düğün organizasyon/etkinlikleri yapılacak bir sistem geliştirilmiştir. Bu sistem modüler yapısı sayesinde farklı organizasyon sistemleri için kullanılacak modüller veya özellikler seçilerek farklı sistemlere de uyarlanabilir.



3. WEB TABANLI ORGANİZASYON SİSTEMİNİN MODELLENMESİ

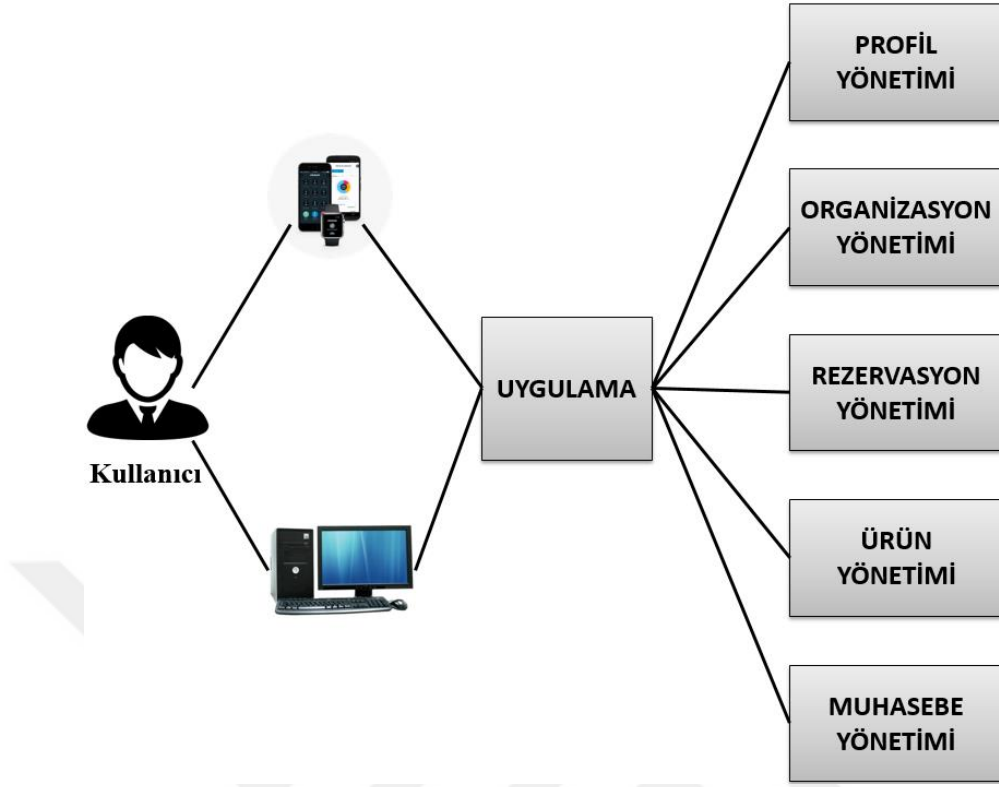
Organizasyonlar, birçok etkinliklerin bir arada bulunduğu yerlerdir. Birden çok şube ile yürütülmesi söz konusu olan organizasyon işlerinde karşılaşılan zorluklardan biri farklı faaliyetlerin yürütüldüğü birimler (mağazalar) arasındaki koordinasyon ve denetimdir. Söz konusu bu sistem, ifade edilen ihtiyaçlara cevap verecek içeriğe sahip olacaktır. Organizasyon sahibi, birbirinden bağımsız ve farklı kişilerle yaptığı anlaşmaları aynı anda hakim olabilecek, takip edebilecektir.

Organizasyon faaliyetleri ile ilgili günlük tutulabilecek ve programa girilen tüm bilgiler saklanabilecektir. Bütün bu işlemlere erişim, kişilere verilecek yetki düzeyinde olacaktır. Hazırlanan bu sistemin web tabanlı olması, bilgisayarlara ayrı ayrı program kurulumunu gerektirmeyecek, internet olanağı olan her yerden erişimi mümkün kılacak, bu sayede kullanıcı hataları veya virüs nedeniyle oluşacak veri kayıpları önlenebilecektir. Sistemin yapısı aşağıdaki alt bölümlerde detaylı olarak anlatılmaktadır.

3.1. Sistemin Yapısı

Bu tez çalışmasında gerçekleştirilen sistemden 5 alt bölümde oluşmaktadır. Bunlar sırasıyla: Profil yönetimi, organizasyon yönetimi, rezervasyon yönetimi, ürün yönetimi ve muhasebe yönetimi işlemleri yapan bir uygulama tasarlanmıştır. İnternet erişimi olan herhangi bir yerden herhangi bir mobil cihaz veya bilgisayar tarafından kolayca sistem kullanılabilir.

Şekil 3.1.'de tanımlanan modüllerde kendi arasında farklı alt işlemleri kapsamaktadır. Bu alt işlemler birbirleri ile etkileşimli elemanlarından oluşmaktadır. Bu elemanlar arasında ilişkiler vardır. Web tabanlı bir organizasyon sistemi, birden fazla faaliyetleri gerçekleştirmek amacıyla bütünleştirilmiş bir plan oluşturmak üzere birbirleri ile ilişkili Kayıt İşlemleri, Yetkilendirme, Profil Yönetimi, Rezervasyon İşlemleri, Ürün ve Hizmetler Yönetimi, SMS ve E-posta Bildirim Sistemi süreçlerini kapsamaktadır.



Şekil 3.1. Sistem uygulaması

Şekil 3.1.'de gösterilen sistem uygulamasındaki alt bölümlerde gerçekleşen işlemler hakkında aşağıda bilgiler verilmiştir:

Kayıt İşlemleri: Libya'da bir organizasyona dahil olmak isteyen kişilere ait veritabanı oluşturmak için öncelikli olarak kayıt işlemi gerçekleştirilmelidir. Bu sistem ile Libya'da yaşayan insanlar, veritabanına tek bir noktadan kayıt olarak bilgilerini girebilecek ve güncelleyebileceklerdir. Sisteme kayıt yaptıran ve bilgilerini girerek onay alan kişiler, sistemden yararlanabileceklerdir.

Yetkilendirme: İstemcinin hangi kaynaklara erişim hakkı olduğunu denetleyen ve belirlenen kısıtlamalara göre erişim kontrolü sağlayan bir yetkilendirme sistemidir. Yetkilendirme süreci kendi içinde ayrı bir kimlik doğrulama mekanizması barındırmaktadır.

Profil Yönetimi: Salon müdürüne, orkestra şefine, fotoğrafçıya vs. yetki vererek sistemde bulunan rezervasyonlardan haberdar olabileceklerdir. Yetkili kullanıcılar kişisel bilgilerini ve verilen hizmetleri güncelleyebilirler.

Salon Yönetimi: Sisteme sınırsız sayıda salon ekleyerek her bir salon ile ilgili organizasyonları ayrı ayrı yönetebilirsiniz. Salonunuza hizmet listesi tanımlayabilir, organizasyonlarınızı kaydederken hangi hizmetlerin satın alındığını seçebilirsiniz.

Rezervasyon İşlemleri: Rezervasyon takvimi ile bütün etkinlikler aynı sayfada görünüp yönetilebilir. Ön kayıt oluşturulabilir, oluşturulan ön rezervasyonlar kesin rezervasyona dönüştürülebilir. Oluşturulan etkinlikler isim etkinlik adı ve saat olarak seçilen tarihte takvim üzerinde görüntülenebilir.

Ürünler ve Hizmetler Yönetimi: Bir yetkili yöneticiler böyle kendi sunulan hizmetleri güncelleyebilir düğün salonları, fotoğrafçılar ve müzik gruplarının olarak servis sağlayıcıları yanında, yeni ürünler ve bunların mağazalara ekleyebilir.

SMS ve E-posta Bildirim Sistemi: SMS ile sözleşme teyitleri ve ödeme hatırlatmaları yapılabilmektedir. SMS ve Mail modülü sayesinde yaklaşan rezervasyonlar önceden müşterilere bilgilendirmektedir. Ayrıca rezervasyonu yapılan müşteriye işlemleri hakkında salon, tarih, saat, vb. gibi konu ile ilgili bilgilendirme SMS gönderilebilir.

Muhasebe Yönetimi: Tahsilat ve yapılan ödemeleri gün gün nakit akışı grafiğinden görüntülenebilir. Düğün salonunuzun gelir ve giderlerin kayıtlarını girerek salonunuzun yıllık bazında ya da aylık bazında gelir gider pusulasını görebilir ve detaylı raporlar alabilirsiniz. Rezervasyon sayfası içerisinde salonu kiraladığınızda kaporo ve diğer gelir işlemleri halledilebilir.

3.2. Modelleme

Yazılım geliştirme sürecinin farklı aşamalarını farklı UML diyagramları ile modellemek sistemin yaşam sürecini ortaya koymak için gereklidir. Bu amaçla; yazılım geliştirme sürecinin farklı aşamalarında farklı UML diyagramlarını kullanmak gerekmektedir. UML 4+1 görünüm modeli, bu diyagramları

sınıflandırmak ve yazılım yaşam çevrimindeki kullanım yerlerini ortaya koymak için kullanılan bir kavramdır. Bu bakışlar;

- a. Kullanım Senaryosu Bakışı (Use Case View): Müşteri gereksinimlerini ortaya koymak ve müşteriye, sistemi tanıtmak amacı ile kullanılan bakış açısıdır.
- b. Yapısal Bakış (Structural View): Sistemin nelerden meydana geldiğini gösteren bakış açısıdır.
- c. Davranış Bakışı (Behavioral View): Sistemin dinamik yapısını ortaya koyan bakış açısıdır.
- d. Gerçekleştirme Bakışı (Implementation View): Sistemin alt modüllerini ortaya koyan bakış açısıdır.
- e. Ortam Bakışı (Environment View): Donanımın fiziksel mimarisinin ortaya konduğu bakış açısıdır.

3.2.1. Kullanıcı Bakışı

Kullanıcı bakışında ilk olarak çalışmanın amacını ortaya koymak gerekir. Web tabanlı organizasyon sisteminin amacı, bir organizasyon kurumunun web tabanlı olarak Libya’da evlilik hazırlığı yapan kişilere uzaktan hizmet verebilmesidir. Evlilik hazırlığı yapan kişiler eşzamanlı veya eşzamansız olarak ve istediği yerde internet üzerinden organizasyon sitesine bağlanarak hizmet alabilmesi ve hizmet verenlerin hizmet alanla ile aynı zamanda ve aynı mekanda olmak zorunda kalmadan sistem üzerinden işlem yapılabilmesini amaçlamaktadır.

Aktörler: Sistemin en önemli özelliklerinden biri de kullanıcılar arasındaki iş ve işlem paylaşımıdır. Bir organizasyonda her kademedeki çalışanın her bilgiye erişmesi sakıncalı olabilir. Bu durum dikkate alınarak sistemde, hiyerarşik bir düzende bilgilere erişim olanağı sağlanmaktadır. Yöneticiler her tür bilgiye erişebiliyorken, birim çalışanları sadece kendilerine verilen yetki doğrultusunda bazen sadece kendi birimi, bazen de ilişkili olduğu faaliyetlere erişerek sistemi kullanabilmektedir.

Bu tez çalışmasında gerçekleştirilen sistemde tanımlanan aktörler aşağıdaki şekilde listelenebilir:

- Okuyucu/Misafir/Ziyaretçi
- Genel Kullanıcı
- Editör
- Yönetici
- Süper Yönetici

Kullanıcılara atanan roller vasıtasıyla sistem yönetimi daha düzenli bir şekilde gerçekleşir. Her bir rol kendi yetkileri dahilinde içerik üzerinde işlem yaparlar. Yönetim kısmında yeni kullanıcı ve roller sisteme eklenebilir.

Okuyucu/Misafir/Ziyaretçi: Anonim bir web sitesi ziyaretçisidir, web sitesinin bölümlerini görüntüleyebilir ve genel kullanıcı hesabı için kayıt olabilir. Genelde içeriğin sadece yayımlanmış ya da yayımlanmak üzere olan halini görme hakkına sahip, içerikten faydalanan son kullanıcıdır.

Genel Kullanıcı: Sisteme kayıt yapan bir kullanıcıdır. Oluşturulan hesabı aracılığıyla rezervasyon yapma, rezervasyon kontrolü veya rezervasyon iptal gibi haklara sahip kullanıcılarıdır.

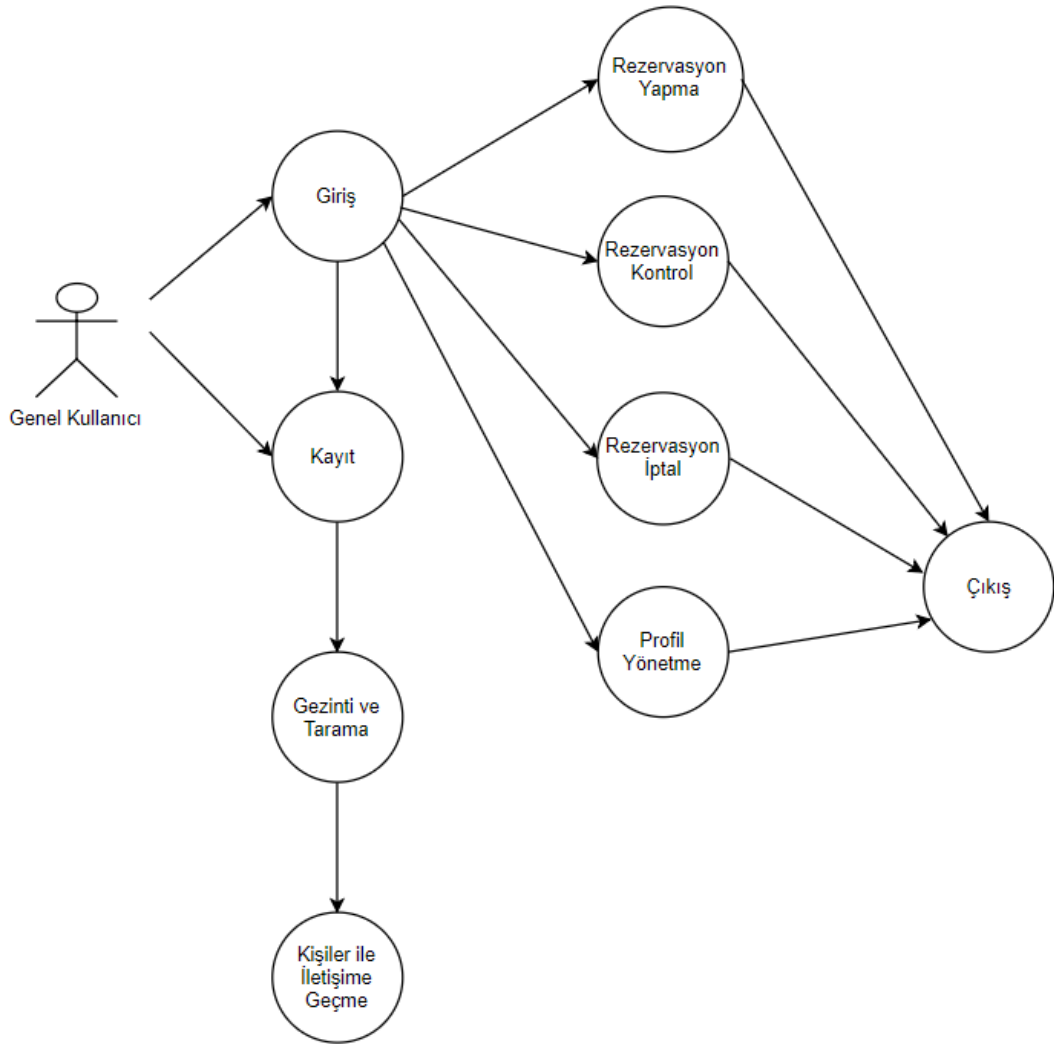
Editör: Bu web uygulamasındaki yönetici, yönetici çalışanları ”çalışanları” için yeni hesap oluşturma yetkisine sahip bir mağaza ya da salonun sahibi ve ürün ve hizmetlerini kontrol etme gibi haklara sahip kullanıcılarıdır. Bu hesap sadece sistem sahibi olan süper yönetici tarafından yaratılabilir ve verilebilir.

Yönetici: Kullanıcıların rollerini belirler ve içerikle ilgili teknik sorumluluğu üstlenir. Yönetici kullanıcılar mağazaların/salonların çalışanlarıdır. Ürünler/hizmetler ile ilgili genel düzenlemeler, yönetim, rezervasyon gibi organizasyon faaliyetleriyle ilgili haklara sahip kullanıcılarıdır. Bu hesap sadece sistem sahibi olan süper yönetici tarafından oluşturabilir.

Süper yönetici: Süper yöneticiler sistemin genel düzenlemeleri, servis sağlayıcılar (mağazalar, salonlar, müzik grupları, fotoğrafçılar, vb.) için hesapların oluşturulması ve denetimi, kullanıcı hesaplarının yönetimi haklarına sahiptir.

3.2.2. Kullanım Durumu Diyagram ve Senaryoları

Aktörler ve rollerini ortaya çıkardıktan sonra ilk olarak bu roller için Kullanım Durum Senaryolarını ve diyagramlarını ortaya koymak gerekir. Kullanım durum senaryoları sistemin işlevsel gereksinimlerine denk gelmektedir. Bu gereksinimleri iyi belirlemek yazılım sisteminin talep edilen işlevselliğini doğru bir şekilde anlamayı sağlar.



Şekil 3.2. Genel Kullanıcı için bir senaryo

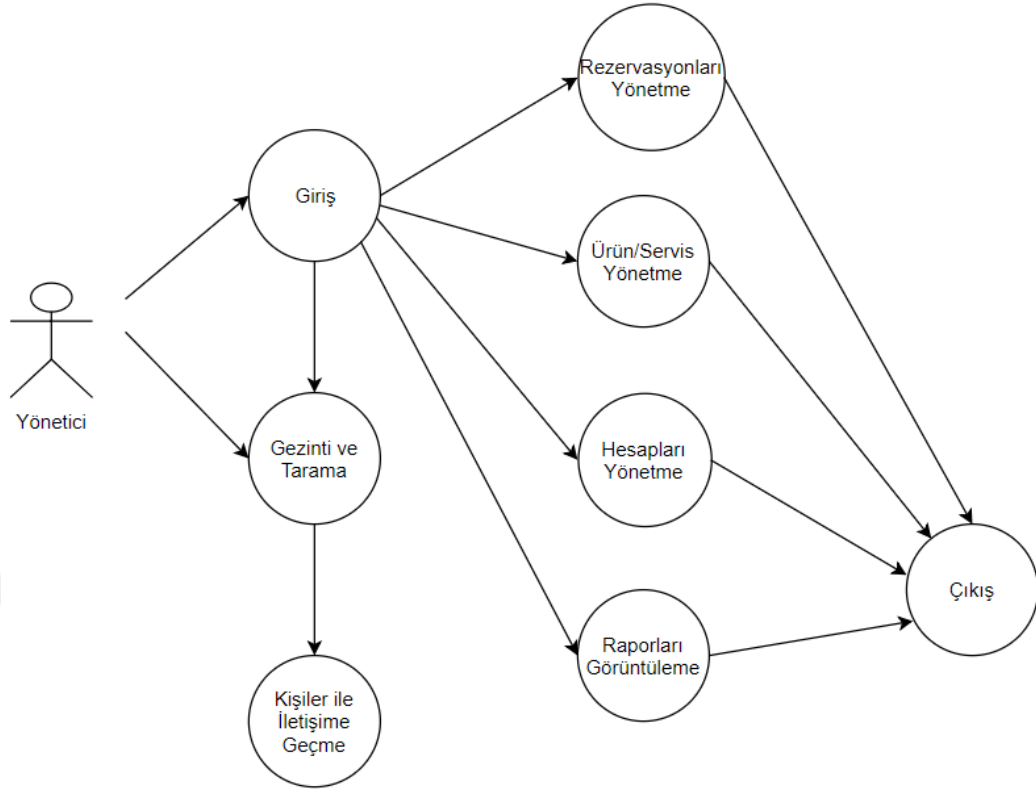
Şekil 3.2.'de, rezervasyonların ve profilin yönetildiği bir sisteme ilişkin bir kullanım senaryosu diyagramı gösterilmiştir. Bu kullanım senaryosu diyagramında Genel Kullanıcı, sisteme giriş yaparak yetkisi dahilinde işlemleri gerçekleştirebilir veya kişiler ile iletişime geçme gibi sistemde gezinti yapabilmektedir.



Şekil 3.3. Editör için bir senaryo

Şekil 3.3.'de rezervasyonların, ürünler/servislerin ve hesapların yönetildiği, yapılan işlemlere ait raporların düzenlendiği bir sisteme ilişkin bir kullanım senaryosu diyagramı gösterilmiştir. Bu kullanım senaryosu diyagramında Editör kullanıcısı, sisteme giriş yaparak yetkisi dahilinde işlemleri gerçekleştirebilir veya genel kullanıcı gibi sistemde gezinti yapabilmektedir.

Şekil 3.4.'de, rezervasyonların, ürünler/servislerin ve hesapların yönetildiği, yapılan işlemlere ait raporların düzenlendiği bir sisteme ilişkin bir kullanım senaryosu diyagramı gösterilmiştir. Bu kullanım senaryosu diyagramında Yönetici kullanıcısı, sisteme giriş yaparak yetkisi dahilinde işlemleri gerçekleştirebilir veya genel kullanıcı gibi sistemde gezinti yapabilmektedir.

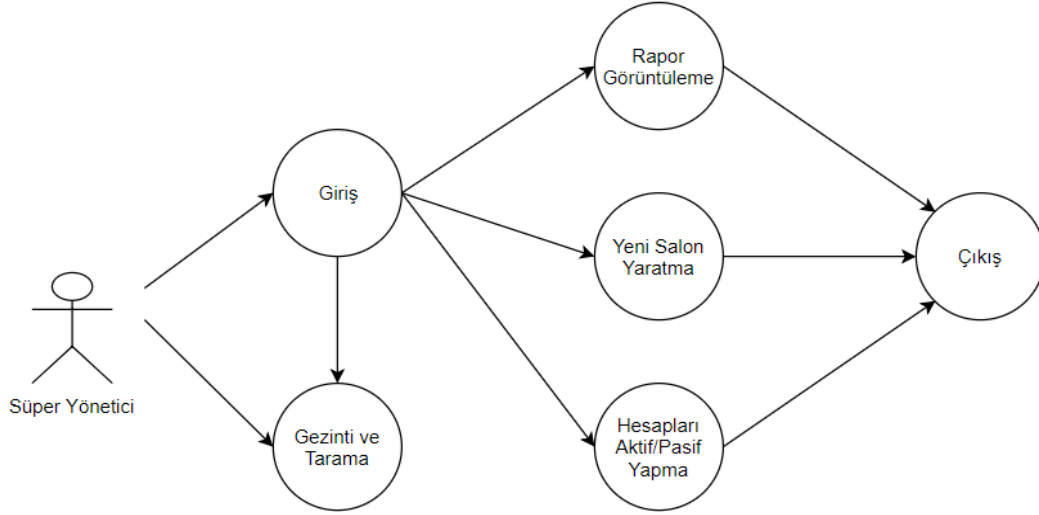


Şekil 3.4. Yönetici için bir senaryo

Şekil 3.5.'de organizasyonların/etkinliklerin yapılabacağı sistemdeki Süper Yöneticiye ilişkin bir kullanım senaryosu diyagramı gösterilmiştir.

Bu diyagramda gösterilen aktörler, sistemi kullanarak rapor görüntüleme, yeni salon yaratma ve hesap kontrolü veya sistemde gezinti/tarama yapacak olan Süper Yöneticiler'dir.

Elipsler kullanılarak resmedilen kullanım senaryoları ise bir yada birkaç aktör tarafından, belli bir amaca yönelik olarak gerçekleştirilen işlemleri ifade etmektedir.

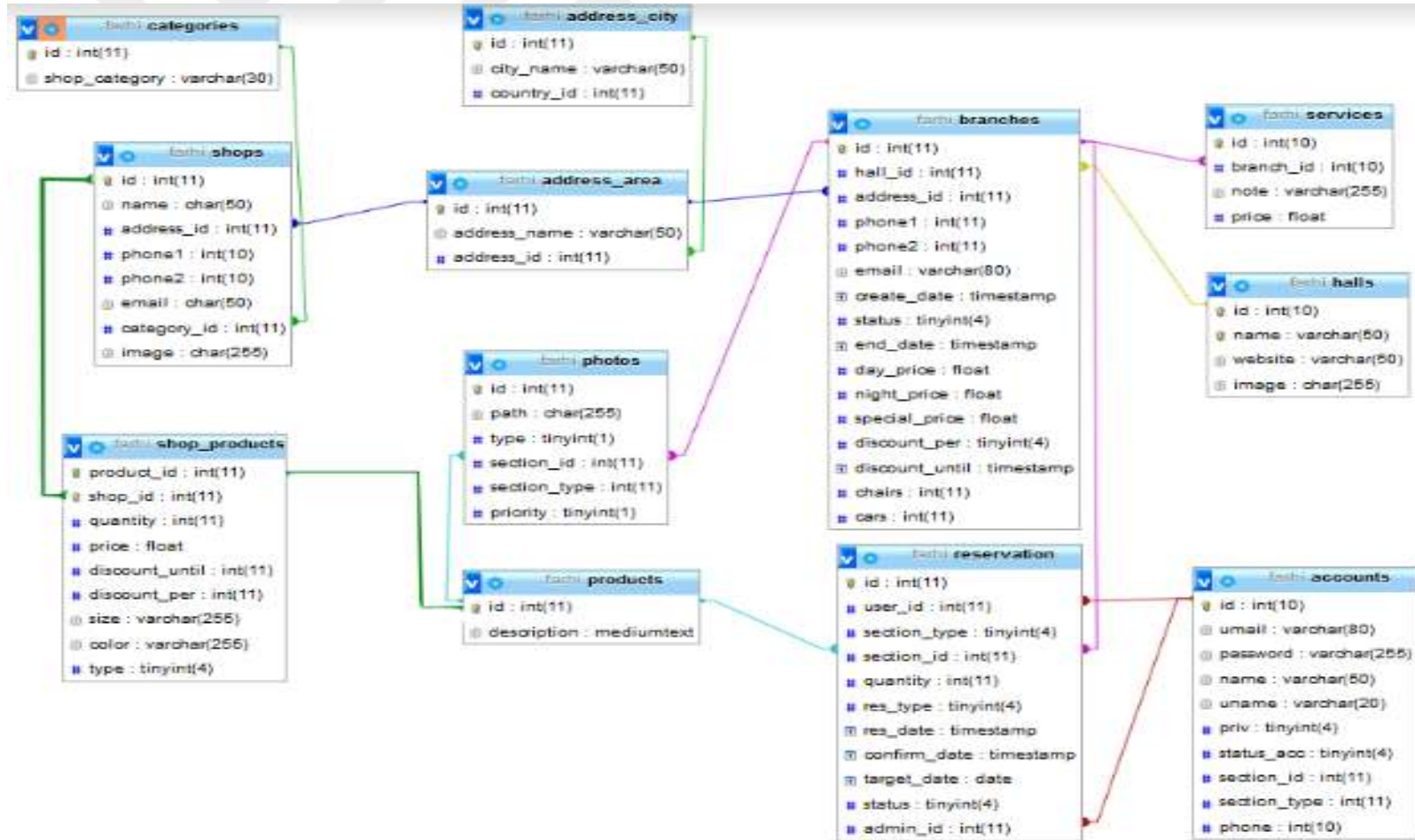


Şekil 3.5. Süper Yönetici için bir senaryo

3.2.3. Yapısal Bakış

Kullanım durum senaryoları oluşturduktan sonra bu senaryolar üzerinde ki sınıfları ortaya koymak gerekmektedir. Bu senaryolara dayanarak ve genelleme yaparak; Kategoriler (Categories), mağazalar (shops), mağazalar (shops_products), ürünler (products), adres_şehir (address_city), adres_alanı (address_area), photos (fotoğraflar), şubeler (branches), hizmetler (services), salonlar (halls), rezervasyon (reservation) ve hesaplar (accounts) isimli 12 tane sınıf seçilebilir.

Şekil 3.6. ile Web Tabanlı Organizasyon Yönetimi sistemi için seçilen 12 sınıfın özellikleri, işlevleri ve aralarındaki ilişkiler UML'nin sınıf diyagramı ile modellenmiştir.



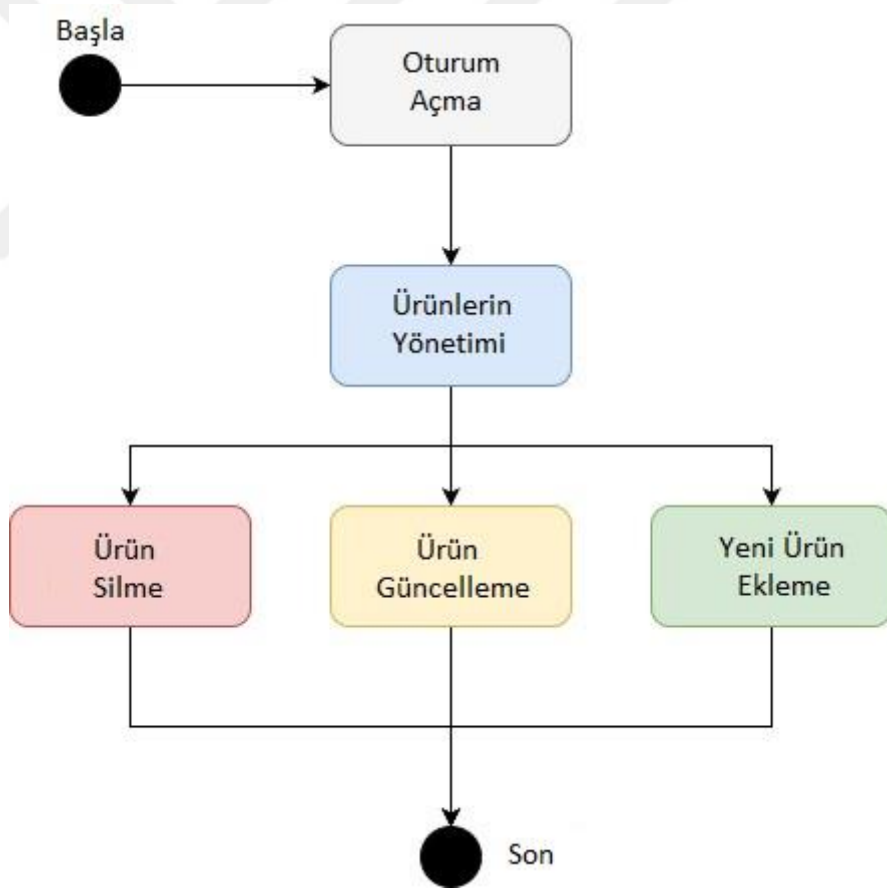
Şekil 3.6. Web tabanlı organizasyon sistemi sınıf diyagramı

3.2.4. Davranış Bakışı

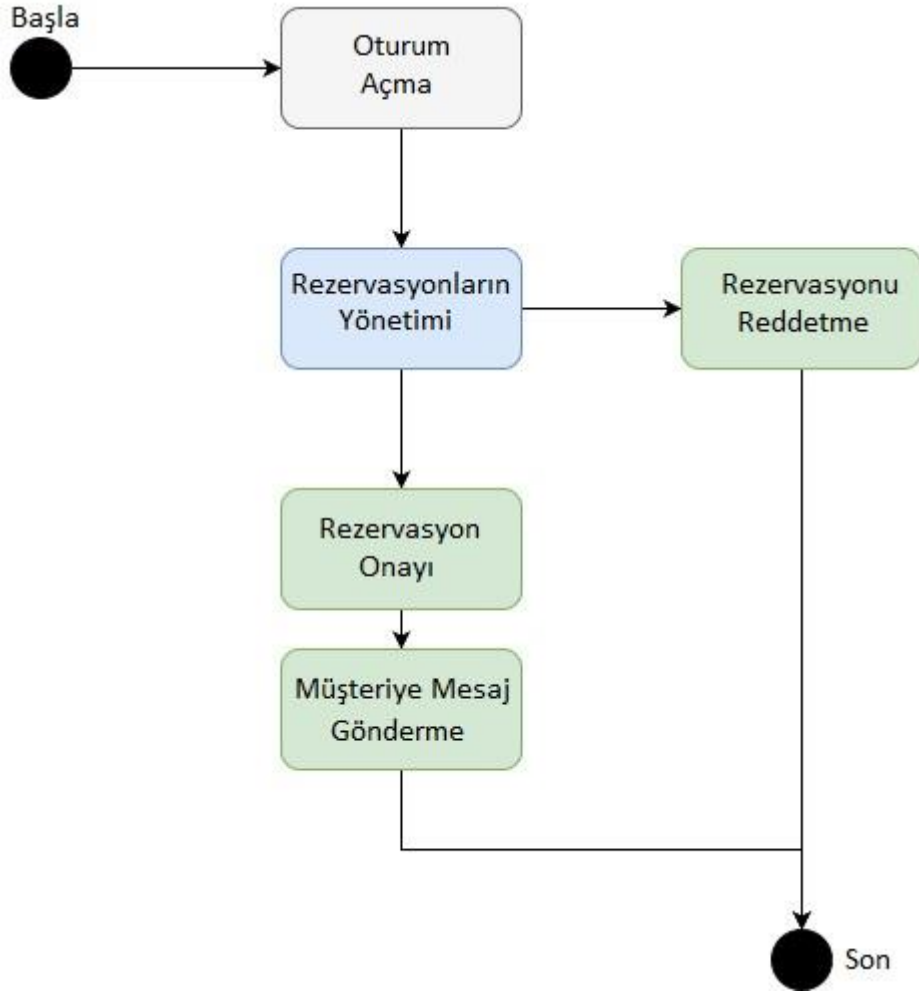
Bu bölümde nesnelerin ve sistemin durumlarını etkileyen olaylar ve etkinlikler soyut bir düzeyde incelenecektir. Nesnelerin, programın kendi akışı içerisinde gerçekte nasıl bir işbirliği yaptığı, nasıl bir etkileşimde bulunduğu modellenecektir.

Bu modelleme de UML'nin durum, etkinlik ve ardışık sıra diyagramlarından faydalanılacaktır.

Daha önce tanımlanan kullanım senaryosu diyagramları göz önünde bulundurularak Şekil 3.7. ile Yöneticinin sisteme giriş yapması ile Yönetici Kullanıcısının izin verdiği işlemleri görme ve yapma izni ile ilgili UML durum şeması gösterilmiştir.

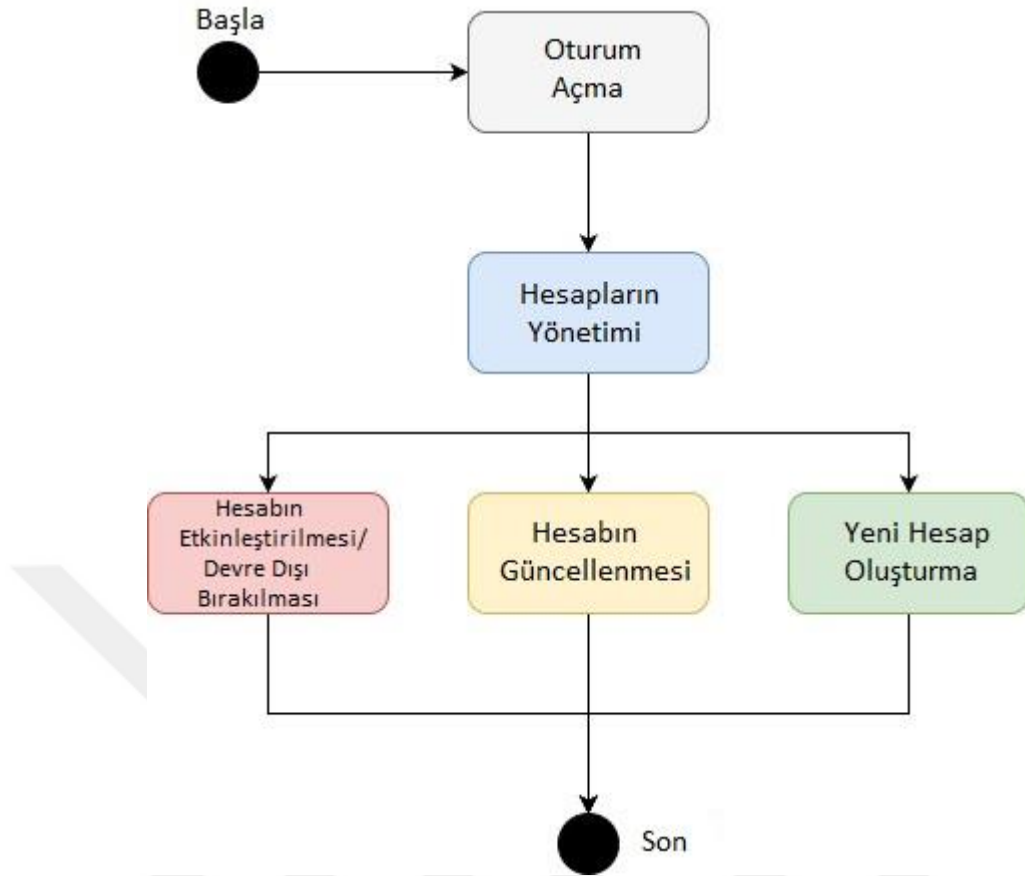


Şekil 3.7. Yönetici kullanıcıasına ait sisteme giriş durum diyagramı



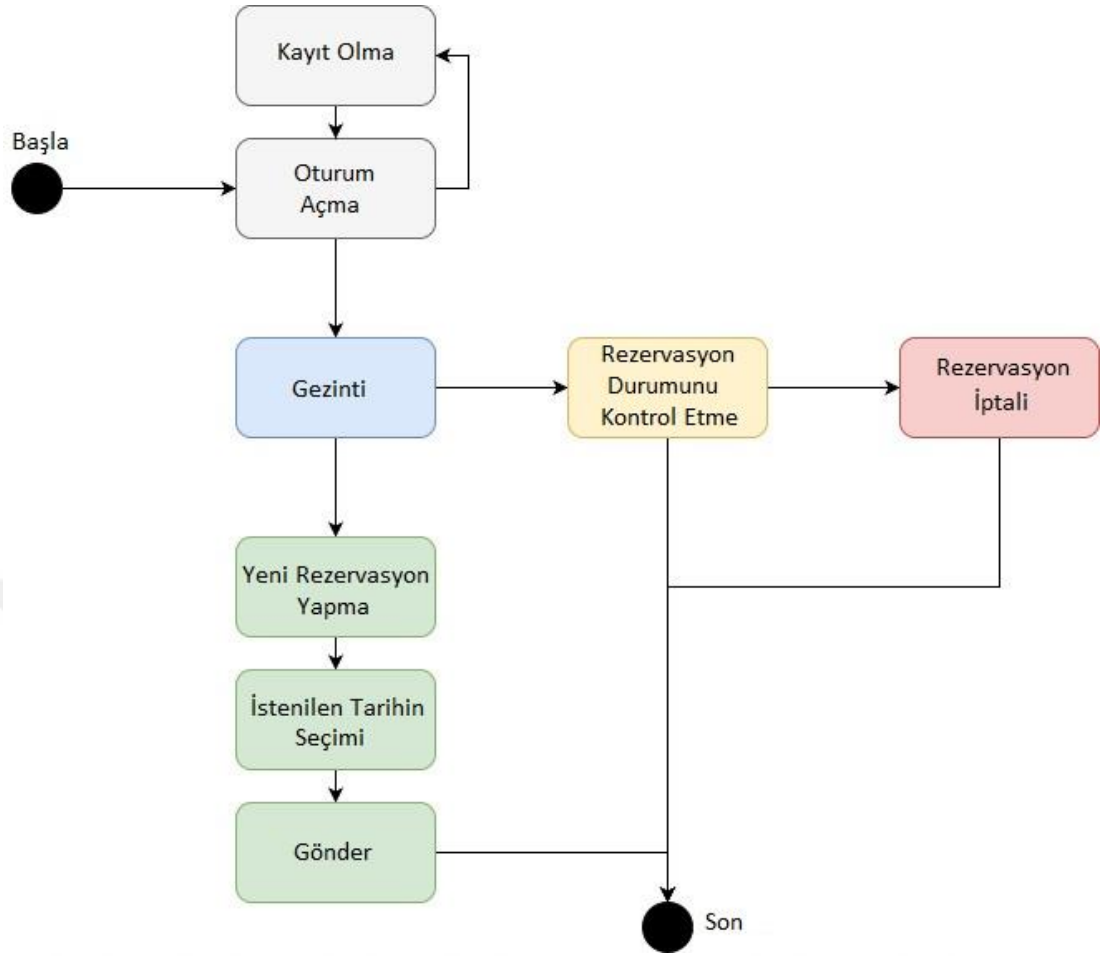
Şekil 3.8. Editör kullanıcısının sisteme giriş durum diyagramı

Şekil 3.8.'de Editörün sisteme girişi ile yapabileceği işlemler UML'nin durum şeması ile gösterilmiştir.



Şekil 3.9. Süper Yönetici kullanıcısının sisteme giriş durum diyagramı

Şekil 3.9.’de Süper Yöneticinin sisteme girişi ile yapabileceği işlemler UML’nin durum şeması ile gösterilmiştir.



Şekil 3.10. Genel bir kullanıcının sisteme giriş durum diyagramı

Şekil 3.10.'de Kullanıcının sisteme girişi ile yapabileceği işlemler UML'nin durum şeması ile gösterilmiştir.

3.2.5. Ortam Bakışı

Ortam Bakışı, donanımın fiziksel mimarisinin ortaya konduğu bakış açısıdır. Bu sistemin nasıl çalışacağını gösterirken kullanılan en üst düzey bakışlardan birisi de sistemi kurarken fiziksel olarak kullanılacak olan bilgisayarların, diğer cihazların, barındırdıkları bileşenlerin ve bu cihazların nasıl iletişim kuracaklarının gösterildiği ortam bakışıdır.

3.3. Veritabanı Tasarımı

İlişkisel veritabanı yönetim sistemi, verilerin tablolarda satır ve sütunlar halinde tutulduğu ve yüksek bir veri tutarlılığına sahip veri depolama sistemidir. İlişkisel veritabanını çeşitli tablolar arasında organize edilmiş verilerden oluşan veritabanıdır. Her satır aynı sütunlara yani alanlara sahiptir. Her bir sütun o tabloda bulunması gereken ortak özellikleri yansıtır.

3.3.1. Veritabanı Tabloları

Veritabanında bilgiler tablolar halinde düzenlenmektedir. Bu tablolar satır ve sütunlardan oluşan listelerdir. Tez çalışmasında tanımlanan veritabanı, aşağıda kısaca açıklanan tablolardan oluşmaktadır.

Kullanıcı Hesapları Tablosu: Sistemde yer alan kullanıcılara ait hesapların bilgilerini içerir. Yeni bir kullanıcı açıldığında yeni bir kayıt eklenir. Sisteme giriş hesabı olmayan kişiler eğer varsa genel kullanıcı giriş hesabını kullanırlar. Yönetici hesabına sahip kişiler bu tablonun kayıtlarına erişebilir. Kullanıcı hesabı genel kullanıcı, yayımcı, yönetici veya süper yönetici yetkisine sahip olabilir. Kullanıcılara ait rollerde tanımlanmaktadır. Tüm web sitesi kullanıcılarına genel kullanıcı yetkisi verilir. Yönetici hesabına Salon/mağaza sahipleri, salon/mağazada çalışanlarını, ürünlerini ve hizmetlerini kontrol etmeleri için yönetici yetkisi verilir. Süper yönetici hesabı sistem etki alanına sahiptir. Sistemde yer alan tüm işlemlerden sorumludur.

Table 3.1. *Kullanıcı hesap tablosu*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(10)			No	None		AUTO_INCREMENT
2	umail	varchar(80)	utf8_general_ci		No	None		
3	password	varchar(255)	utf8_general_ci		No	None		
4	name	varchar(50)	utf8_general_ci		No	None		
5	uname	varchar(20)	utf8_general_ci		No	None		
6	priv	tinyint(4)			No	None		
7	status_acc	tinyint(4)			No	1		
8	section_id 	int(11)			Yes	NULL		
9	section_type 	int(11)			Yes	NULL		
10	phone	int(10)			No	None		

Tablo 3.1.'de sistemde yer alacak olan kullanıcılara ait hesapların bilgileri tutulmaktadır. Bu tabloda her bir satırda alanlar yer almaktadır. Alanlara ait özellikler ise sütunlardadır. Özellikler farklı veri tiplerinde olabilir. Aşağıda, Tablo 3.1.'de yer alan alanlar hakkında bilgilere yer verilmiştir.

- **Id (primary key):** Tablonun birincil anahtarıdır. Kullanıcı kimlik numarasını içerir.
- **Umail:** Kullanıcıya ait e-mail adresini içerir.
- **Password:** Kullanıcı parolasını içermektedir.
- **Name:** Kullanıcı adını temsil eder.
- **Uname:** Sistem tarafından otomatik olarak oluşturulan kullanıcı adıdır (kullanıcı kontrol edemez).
- **Priv:** Kullanıcı ayrıcalığı 1, 2 veya 3 olabilir. Bunlar sırasıyla; genel kullanıcı, yönetici kullanıcı veya süper yöneticidir.
- **Status_acc:** Hesap durumu aktif için varsayılan olarak 1 veya etkin olmayan hesaplar için 0 olabilir.
- **Section_id (foreign key):** Bu, dallar tablosundaki branch_id veya bölüm tipi alanına bağlı olarak mağazalardaki shop_id anlamına gelir; hesap alanı 1 (genel kullanıcı) ise bu alan NULL olabilir.
- **Section_type:** Bu alan, hesabın şubeye atıfta bulunup bulunmadığını belirlemek için bir bayrak olarak görev yapar. Bayrak değeri 1 ise salonu

temsil eder. Eğer bayrak değeri 2 ise mağazayı temsil eder. Süper Kullanıcı hesabında bu değer alanı NULL olabilir.

- Phone: Kullanıcı telefon numarasını belirtir.

Salonlar Tablosu: Sisteme katılan düğün salonları hakkında genel bilgi içermektedir.

Sadece süper yönetici bu tabloya erişebilir. Tablo 3.2.'de düğün salonları tablosu gösterilmiştir.

Table 3.2. *Düğün salonları tablosu*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(10)			No	None		AUTO_INCREMENT
2	name 	varchar(50)	utf8_general_ci		No	None		
3	website	varchar(50)	utf8_general_ci		Yes	NULL		
4	image	char(255)	latin1_swedish_ci		No	None		

Aşağıda, Tablo 3.2.'de yer alan alanlar hakkında bilgilere yer verilmiştir.

- Id (primary key): Düğün salonun kimlik numarası.
- Name: Düğün salon adıdır.
- Image: Düğün salonu için logoyu temsil eder.

Şubeler Tablosu: Düğün organizasyon sektörünün gelişimini desteklemek, şube sayısını artırmak tüm Libya'ya yayılmak her firmanın vizyonudur. Bu nedenle geliştirilen sistem, Libya'nın her bir şehrine yayma amacıyla firmaların şubelendirilmesi önemli bir faktördür. Tablo 3.3.'de, düğün salonları ve düğün salonu şubelerine ait bilgileri tutan veritabanı tablosu gösterilmiştir.

Tablo 3.3. *Düğün salonları ve şubeleri tablosu*

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id 🔑	int(11)			No	None		AUTO_INCREMENT
hall_id 🔑	int(11)			No	None		
address_id 🔑	int(11)			No	None		
phone1	int(11)			No	None		
phone2	int(11)			Yes	NULL		
email	varchar(80)	utf8_general_ci		No	None		
create_date	timestamp			No	CURRENT_TIMESTAMP		
status	tinyint(4)			No	1		
end_date	timestamp			Yes	NULL		
day_price	float			Yes	NULL		
night_price	float			Yes	NULL		
special_price	float			Yes	NULL		
discount_per	tinyint(4)			Yes	NULL		
discount_until	timestamp			Yes	NULL		
chairs	int(11)			No	None		
cars	int(11)			No	None		

Aşağıda, Tablo 3.3.'de yer alan alanlar hakkında bilgilere yer verilmiştir.

- **id** (primary key): Şube kimlik numarasıdır. Bu tablonun birincil anahtarıdır.
- **hall_id** (foreign key): Salonlar tablosundaki kimlik alanını ifade eder.
- **address_id** (foreign key): Address_city tablosunda yer alan kimlik alanını ifade eder.
- **phone1**: İletişim amacıyla birinci tercih telefon numarasıdır.
- **phone2**: İkinci tercih telefon numarasıdır.
- **create_date**: Şube için sistem abonelik gün değeridir.
- **status**: Şube durumu 1 ise aktif olarak 0 “varsayılan” değeri veya 0 ise aktif değildir.
- **end_date**: Şubenin sisteme abonelik günü.
- **day_price**: Gündüz kiralama fiyatıdır.
- **night_price**: Gece kiralama fiyatıdır.
- **special_price**: Hafta sonları ve özel günler için kiralama fiyatıdır.

- discount_per: Kiralama işleminde belirlenen fiyat üzerinden uygulanabilecek iskontosu yüzdesidir.
- discount_until: Kiralama fiyatına uygulanacak iskontonun bitiş süresi (indirim için son gün tarihidir).
- Chairs: Salondaki sandalye sayısı (maksimum davet numarası).
- Cars: Otoparkın araç kapasitesidir (maksimum araç sayısı).

Servisler Tablosu: Yönetici kullanıcısı bu tabloya hizmet ve fiyatları ekleme etkisine sahiptir.

Tablo 3.4.'de salonların ek hizmetleri ile ilgili bilgiler gösterilmiştir. Sistem ile ilgili gerekli değişiklikleri bu tablo üzerinden yapmaktadır.

Table 3.4. *Servisler tablosu*

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id 	int(10)			No	None		AUTO_INCREMENT
branch_id 	int(10)			No	None		
note	varchar(255)	latin1_swedish_ci		No	None		
price	float			No	None		

Aşağıda, Tablo 3.4.'de yer alan alanlar hakkında bilgilere yer verilmiştir.

- id (birincil anahtar): Servis kimlik numarası.
- branch_id (yabancı anahtar): “Şubeler” tablosundaki “id” alanına karşılık gelmektedir.
- Note: Servis hakkında bilgi vermektedir.
- Price: Servis (hizmet) ücretidir. Bazı servisler ücretsiz bir hizmet de olabilir.

Mağazalar Tablosu: Yönetici kullanıcısı, bu tabloya yeni mağaza ekleyebilir veya mevcutta varolan mağazalar ile ilgili gerekli güncelleme, silme, vb. işlemleri gerçekleştirebilir.

Tablo 3.5.'de sistemdeki mağazalar hakkında bilgiler gösterilmiştir.

Table 3.5. *Mağazalar tablosu*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	name	char(50)	utf8_general_ci		No	None		
3	address_id	int(11)			No	None		
4	phone1	int(10)			No	None		
5	phone2	int(10)			Yes	NULL		
6	email	char(50)	utf8_general_ci		No	None		
7	category_id	int(11)			No	None		
8	image	char(255)	latin1_swedish_ci		No	None		

Aşağıda, Tablo 3.5.'de yer alan alanlar hakkında bilgiler listelenmiştir.

- Id (birincil anahtar): Mağazanın kimlik numarası.
- Name: Mağazanın ismidir.
- Address_id (yabancı anahtar): address_city tablosundaki kimlik alanını karşılık gelmektedir.
- Phone1: İlk telefon numarası.
- Phone2: İkinci telefon numarasıdır. Bu alan tercihe kalmıştır. NULL değerini de alabilir.
- E-mail: Mağazanın e-posta adresidir.
- Category_id (yabancı anahtar): Kategori tablosundaki kimliği ifade eder.
- Image: Mağazaya ait logodur.

Kategoriler Tablosu: Bir organizasyon/etkinlik planlaması farklı işlemler ile birlikte yapılmaktadır. Bu tez çalışmasında durum çalışması olarak ele alınan düğün organizasyonunda farklı işlemler için farklı şirketler veya mağazalar ile etkileşim halinde olunmalıdır. Bu nedenle düğün ile ilgili işlemlerin yapılacağı mağazalar aşağıdaki şekilde dört farklı alt kategoriye ayrılabilir:

- Giyim mağazaları
- Gıda mağazaları.

- Güzellik merkezi.
- Makyaj ve parfüm mağazaları.

Tablo 3.6.'da mağaza kategorileri hakkında bilgiler gösterilmiştir.

Table 3.6. *Kategoriler tablosu*

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id 🔑	int(11)			No	None		AUTO_INCREMENT
shop_category	varchar(30)	utf8_general_ci		No	None		

Aşağıda, Tablo 3.6'de yer alan alanların özellikleri hakkında bilgiler listelenmiştir.

- Id (birincil anahtar): Kategori kimlik numarası.
- Shop_category: Kategori adı.

Ürünler Tablosu: Süper yönetici bu tabloya yeni ürün ekleyebilir veya bu ürünler üzerinde gerekli güncellemeleri yapabilir.

Tablo 3.7.'de mağazalarda satılan veya mağazanın kapsadığı ürünler listelenmiştir. Bu tabloya erişim yetkisi sadece Süper Yöneticidir. Ürünler aşağıdaki gibi üç kategoriye ayrılabilir:

- Giyim Hizmetleri
- Kuaför Hizmetleri
- Gıda Hizmetleri

Table 3.7. *Ürünler tablosu*

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id 🔑	int(11)			No	None		AUTO_INCREMENT
description	mediumtext	utf8_general_ci		No	None		
category_id 🔑	int(11)			No	None		

Aşağıda, Tablo 3.7.'de yer alan alanların özellikleri hakkında bilgiler listelenmiştir.

- Id (birincil anahtar): Ürün kimlik numarası.
- Description: Ürün adı.
- Category_id (yabancı anahtar): Kategori tablosundaki kimlik alanına karşılık gelmektedir.

Şehir Adresi Tablosu: Tablo 3.8.'da sistemde yer alan şehirler hakkında bilgiler yer almaktadır.

Tablo 3.8. *Adres tablosu*

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id 	int(11)			No	None		AUTO_INCREMENT
city_name	varchar(50)	utf8_general_ci		No	None		

Aşağıda, Tablo 3.8.'de yer alan alanların özellikleri hakkında bilgiler listelenmiştir.

- Id (birincil anahtar): Şehir kimlik numarasıdır.
- City_name: Şehrin adıdır.

Adres alanı

Tablo 3.9.'da sistemde yer alan mağaza/salon tam adresleri hakkında bilgiler yer almaktadır.

Table 3.9. *Mağaza/salon tam adresleri tablosu*

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id 	int(11)			No	None		AUTO_INCREMENT
address_name	varchar(50)	utf8_general_ci		No	None		
address_id 	int(11)			No	None		

Aşağıda, Tablo 3.9.'da yer alan alanların özellikleri hakkında bilgiler listelenmiştir.

- Id (birincil anahtar): Adres kimlik numarası.
- Address_name: Salonun/mağazanın tam adresi.

- Address_id (yabancı anahtar): Kimlik alanı adresi şehir tablosu anlamına gelir.

Fotoğraflar Tablosu: Hizmet alınacak tüm ürünler ile ilgili ve mağazalardaki salonlara ait fotoğraf bilgileri yer almaktadır. Tablo 3.10’de fotoğraf tablosu gösterilmiştir.

Tablo 3.10. *Fotoğraf tablosu*

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id 	int(11)			No	None		AUTO_INCREMENT
path	char(255)	utf8_general_ci		No	None		
type	tinyint(1)			No	None		
section_id	int(11)			No	None		
section_type	int(11)			No	None		
priority	tinyint(1)			No	0		

Aşağıda, Tablo 3.10.’de yer alan alanların özellikleri hakkında bilgiler listelenmiştir.

- Id (birincil anahtar): Fotoğraflı kimlik numarası.
- Path: Yerel alandaki fotoğrafın bağlantısını temsil eder.
- Type: Verinin tipini belirler. Fotoğraf veya video seçenekleri olabilir.
- Section_id: Salon kimliğine veya ürün kimliğini temsil eder. Bir sonraki özniteliğe bağlı olarak, section_type, bir sonraki bayrağa bağlı olarak dallar tablosundaki veya ürünler tablosundaki id alanını belirtir.
- Section_type: Bu fotoğraf bir salona veya bir ürüne aitse uzmanlaşmış bir bayrak olarak çalışır.
- Priority: Fotoğrafın kapak resmi mi yoksa ürünün normal resmi mi olduğunu tanımlar.

Sepet Tablosu: Kullanıcının alışveriş sepeti, kullanıcının sepete eklediği ve henüz siparişi verilmemiş tüm seçili rezervasyonları içerir. Tablo 3.11.’de kullanıcı sepeti tablosu gösterilmiştir.

Table 3.31. *Kullanıcı sepeti tablosu*

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id 	int(11)			No	None		AUTO_INCREMENT
user_id 	int(11)			No	None		
section_id 	int(11)			No	None		
section_type	int(11)			No	None		
quantity	int(11)			Yes	NULL		

Aşağıda, Tablo 3.11.'de yer alan alanların özellikleri hakkında bilgiler listelenmiştir.

- Id (birincil anahtar): Alışveriş sepeti kimlik numarası. Bu alan birincil anahtardır.
- User_id (yabancı anahtar): Kullanıcı kimliği hesap tablosundaki kimliği alanıyla ilgilidir.
- Section_id (yabancı anahtar): Bir sonraki öznitelik olan section_type alanına bağlı kimlik bu bir salon veya ürün kimliği olabilir.
- Section_type: Rezervasyon şeklini temsil eder. Bu rezervasyon bir salon veya bir ürüne atıfta bulunursa uzmanlaşmış bir bayrak olarak çalışır.
- Quantity: Rezerve edilen ürünlerin miktarı, eğer rezervasyon tipi bir salon rezervasyonu ise NULL olabilir.

Rezervasyonlar Tablosu: Düğün planlayanlar için erken rezervasyon oldukça önemlidir. Özellikler düğün salonları ve düğün tarihleri için rezervasyonlar yaptırılmaktadır. Rezervasyon işleminin dört farklı durumu vardır. Bunlar sırasıyla: Rezervasyon beklemede, rezervasyon onaylandı, rezervasyon tamamlandı ve rezervasyon iptaldir.

Yönetici kullanıcısı ürünleri, salonları ve hizmetleri ile ilgili rezervasyon durumlarını kontrol edebilir. Tablo 3.12.'de rezervasyon yapılan öğeleri ile ilgili özellikleri içeren tablo gösterilmiştir.

Table 3.12. *Rezervasyon tablosu*

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id 🔑	int(11)			No	None		AUTO_INCREMENT
user_id 🔑	int(11)			No	None		
section_type	tinyint(4)			No	None		
section_id	int(11)			No	None		
quantity	int(11)			Yes	NULL		
res_type	tinyint(4)			No	1		
res_date	timestamp			No	CURRENT_TIMESTAMP		
confirm_date	timestamp			Yes	NULL		
target_date	date			No	None		
status	tinyint(4)			No	0		
admin_id 🔑	int(11)			Yes	NULL		

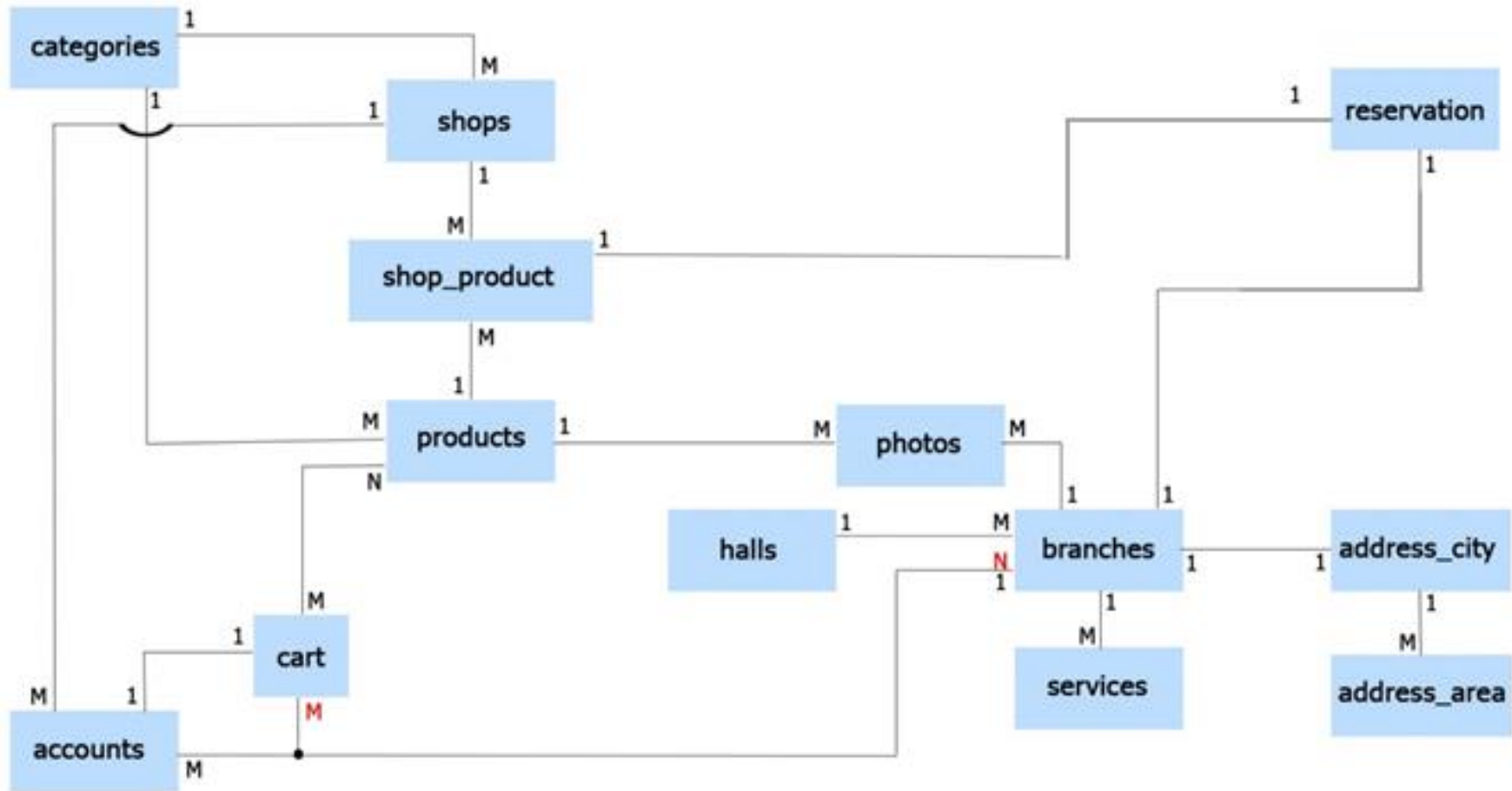
Aşağıda, Tablo 3.12.'de yer alan alanların özellikleri hakkında bilgiler listelenmiştir.

- Status: Rezervasyon durumu (rezerve edilmiş "varsayılan", önceden ödenmiş, ödenmiş veya iptal edilmiş).
- Id (birincil anahtar): Rezervasyon kimlik numarası.
- User_Id (yabancı anahtar): Kullanıcı kimlik numarası, hesap tablosundaki kimliği alanına atıfta bulunur.
- Section_id (yabancı anahtar): bölüm_tipi alanına bağlı olarak dallar tablosundaki branch_id ya da shop_product tablosundaki product_id anlamına gelir.
- Section_type: Rezervasyonun şubeye ve değerinin 1 olacağını mı yoksa bir mağazayı mı ifade ettiğini ve değeri 2 olacağını belirlemek için bayrak olarak hareket eder.
- Res_type: Rezervasyon tipi bir kira olabilir veya varsayılan değer olan kiralık 1'dir.
- Res_date: Kullanıcının rezervasyon formunu gönderdiği gün.
- Target_date: Gerekli rezervasyon günü.
- Status: Rezervasyon durumu (yukarıda belirtildiği gibi dört statüye sahiptir).

- Admin_id (yabancı anahtar): Hesaplar tablosundaki kimlik alanını belirtir, mağaza / şube için birden fazla yönetici varsa, bu alan hangi yöneticinin herhangi bir rezervasyonu onayladığını fark etmelerini sağlar.

Varlık-ilişki modeli, veritabanı yönetim sistemlerinden bağımsız veri çözümlemede ve anlamsal veri modellemede en çok kullanılan modeldir. Bu model kullanılarak kavramsal veri modellenmesi yapılır. Veriler ve veriler arası ilişkilerin anlamları ve özellikleri incelenerek Varlık İlişki (Entity Relationship, ER) çizelgeleri oluşturulur. Kullanılacak veritabanı yönetim sistemi belirlendikten sonra ise ER çizelgeleri bu sistemin veri modeline, yani mantıksal modele dönüştürülerek veri tabanı şemaları oluşturulur. Veri türlerinin ve boyutların da verilmesi ile fiziksel model ortaya çıkar.

Şekil 3.11.'de sisteme ait veritabanı uygulaması için ER çizelgesi gösterilmiştir.



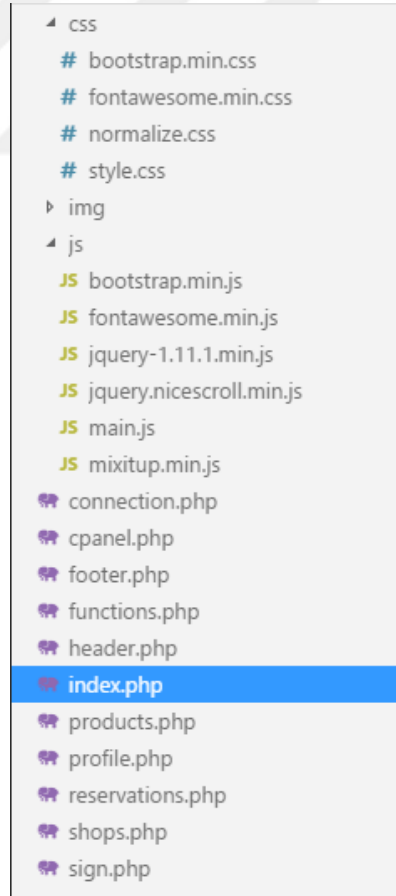
Şekil 3.11. Varlık ilişki diyagramı

4. UYGULAMA

Web tabanlı mimariye sahip uygulama yazılımları, browser içerisinde çalıştırılır ve herhangi bir ek kurulum veya sistem ayarlaması yapılmasını gerektirmez. Web tabanlı mimariye sahip tüm uygulamalar bu tarayıcı programlarda çalışabilir. Tez çalışmasında geliştirilen uygulamaya ait bilgiler aşağıdaki alt bölümlerde detaylı olarak anlatılmaktadır.

4.1. Uygulama Dosyaları

Gerçekleştirilen sistem yazılımı birden fazla web dosyasından oluşmaktadır. Şekil 4.1.'de sistemde oluşturulan dosyaların listesi gösterilmiştir. Aşağıda sırasıyla bu dosyalar hakkında kısa bilgiler verilmiştir



Şekil 4.1. Uygulama dosyaları

/CSS: Bu dizinde stil dosyaları, stil çerçeveleri ve simgeler bulunur.

- bootstrap.min.css: Bu dosya, önyükeme çerçevesinin küçültülmüş kodudur, web sayfalarının düzenini kolaylaştıran ve bunları uyarlayan hazır stiller ve komut dosyaları içerir.
- fontaowsome.min.css: fontaowsome framework, ikonları resimler yerine bir metin olarak basmak için kullanılır, böylece sayfa yükleme çok daha hızlı olabilir.
- normalize.css: Normalize.css, tarayıcıların tüm öğeleri daha tutarlı ve modern standartlara uygun hale getirmesini sağlar. Sadece normalleştirilmesi gereken stilleri kesin olarak hedefler.
- style.css: Bu stil sayfası, elemanların şekillendirilmesinden, renklendirilmesinden, organize edilmesinden ve sipariş edilmesinden sorumlu ana dekorasyon dosyasıdır.

/js: Bu dizin tüm java scriptlerini ve JavaScript kütüphanelerini içerir.

- bootstrap.min.js: Bootstrap bir CSS ve JavaScript çerçevesidir. Web sayfalarının düzenini kolaylaştıran ve uyumlu hale getiren hazır stiller ve komut dosyaları içerir.
- mixitup.min.js: Bu JavaScript tabanlı bir eklentidir, sayfayı yenilemeden istemci tarafındaki verileri filtrelemek için kullanılır.
- jquery.nicescroll.js: Bu, tarayıcının kaydırma çubuğunu değiştirmek ve özelleştirmek için kullanılan JavaScript tabanlı bir eklentidir.
- jquery-1.11.1.min.js: jQuery version 1.11.1, sistemin ön ucunu değiştirmek ve geliştirmek için kullanılan bir JavaScript kütüphanesidir.
- main.js: Bu dosya, uygulamanın istemci tarafına uygulanan tüm olayları ve eylemleri içerir.

/php: PHP kaynak kodu olarak bilinir ve bu PHP dosyaları interaktif fonksiyonlarını dinamik web sayfaları olarak uygulanmaktadır.

- connection.php: Bu dosya MySQL sunucusunda tanınabilir, yetkili bir kullanıcı hesabı kullanarak uygulama ve veritabanı arasında bir bağlantı oluşturur.
- cpanel.php: Sistemin ana kontrol panelidir. Kullanıcılar hesaplarını kullanarak bu panele erişebilirler. Ayrıcalıklarına bağlı olarak bu panelden yararlanabilirler.
- footer.php: Bu dosya, yakın html etiketlerini ve js linklerini içerir, bu dosyayı oluşturmanın amacı web sayfalarında kod tekrarını engellemektir, bu nedenle tüm dosyalara dahil edilmelidir.
- functions.php: İşlevler dosyası, veritabanındaki ve kullanıcılardan gelen verilerdeki heyecan verici talimatlardan sorumlu olan sistemin işlemcisidir.
- header.php: Açılış html etiketlerini, menülerini, stil bağlantılarını ve alt dosyalarını içerir, tıpkı footer.php gibi bu dosyanın amacı kod tekrarını azaltmaktır.
- index.php: Bu dosya sistemin ana sayfasıdır.
- products.php: Bu dosya, kullanıcının isteğine bağlı olarak kategorize edilmiş ürünleri görüntüler, kullanıcıların kendi kategorilerine göre filtreleyebilmeleri için kullanıcı dostu bir filtre kullanır.
- profile.php: Bu dosya ürün / mağaza veya salon hakkındaki tüm bilgileri gösterir ve kullanıcıya bu görüntülenen içeriği ayırma imkanı verir.
- reservations.php: Bu dosya, kullanıcıların rezervasyonlarını ve durumlarını, rezervasyon iptali ile geri alır.
- shops.php: Bu sayfa, kullanıcının isteğine bağlı olarak salonların, güzellik merkezlerinin ve fotoğrafçıların bilgilerini görüntüler ve kullanıcıya rezervasyon olanağı sunar.
- sign.php: Bu dosya, kullanıcının sisteme erişmesi için kimlik doğrulamasından ve kullanıcılar için yeni bir hesap oluşturmasından sorumludur.

4.2. Kayıt Yapma

Genel kullanıcı “müşteriler” yeni hesaba kaydolabilir, böylece sistemden faydalanabilirler. Kullanıcı kayıt üzerine tıkladıktan sonra, sistem kullanıcıyı

rezervasyon formunu görüntülemek için http GET isteğini kullanarak sign.php sayfasına yönlendirir.

Şekil 4.2. Sisteme kayıt ekranı

Formu gerekli bilgilerle doldurduktan ve kaydolmayı tıkladıktan sonra, tarayıcı genel kullanıcı ayrıcalıklarına sahip yeni bir hesap oluşturmak için kaydolma işlevine http POST isteği kullanarak verileri gönderir. Şekil 4.3.'de örnek kod verilmiştir.

```
function
signup($name,$email,$phone_no,$password,$confirm_password,&$error,$conn){
    try {
        if ($password==$confirm_password){

            $insert="INSERT INTO accounts
(name,umail,uname,password,priv,phone) VALUES
('$name','$email','$name','$password',4,$phone_no)";
            $conn->exec($insert);
            $id = $conn->lastInsertID();
            $_SESSION['id']=$id;
            $_SESSION['name']=$name;
            $_SESSION['phone_no']=$phone_no;
            $_SESSION['email']=$email;
            $_SESSION['priv']=4;

            ?>
            <script>alert("registered successfully");</script>
            <?php
            echo "<script>window.top.location='index.php'</script>";
        }
        else {
            $error="password not match";
        }
    }
    catch (PDOException $e){
        echo $e->getMessage();
    }
}
```

Şekil 4.3. Kayıt yapma işlevi için örnek kod

Ardından, uygulama yeni bir kullanıcıyı otomatik olarak yetkilendirir ve onları ana sayfaya yönlendirir.

4.3. Giriş Yapma

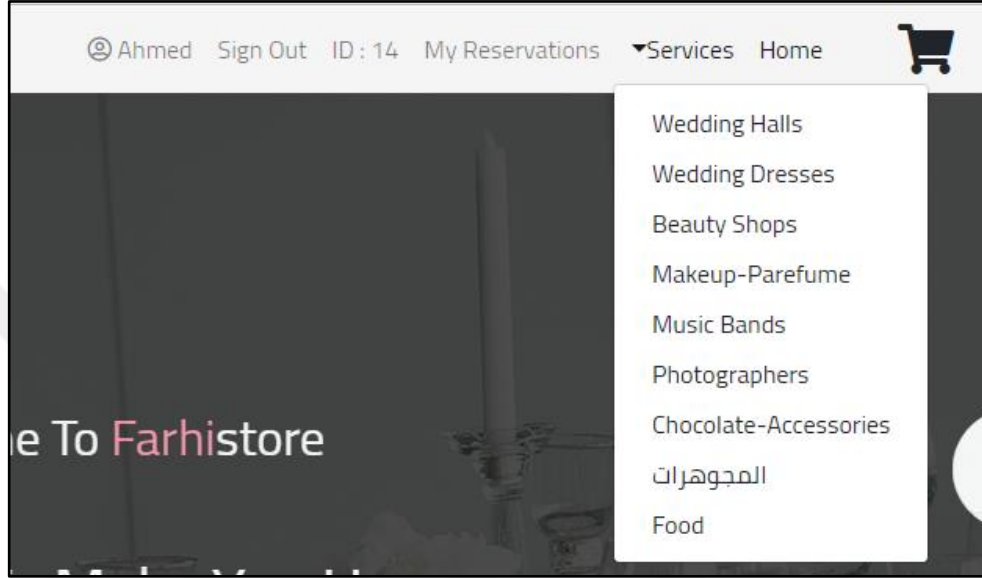
Hesap oluşturma işlemlerinden sonra kullanıcılar sign.php sayfasına giderek sisteme giriş yapabilirler, sistem ayrıcalıklarına bağlı olarak giriş sayfasına veya kontrol paneli sayfasına yönlendirme yetkisine sahip olup olmadığını kontrol eder. Herhangi bir yetkilendirilme yok ise bir hata mesajı görünecektir. Şekil 4.4.'de örnek kod verilmiştir.

```
function signin($id,$password,&$error,$field,$conn){
    try {
        $stmt = "SELECT * FROM accounts WHERE $field='$id' ";
        $result=$conn->query($stmt);
        $rows=$result->fetchAll();
        foreach ($rows AS $row){
            if ($row['password']==$password){
                $_SESSION['id']=$row['id'];
                $_SESSION['priv']=$row['priv'];
                $_SESSION['email']=$row['umail'];
                $_SESSION['name']=$row['name'];
                if ($row['priv']==1){
                    echo
"<script>window.top.location='cpanel.php'</script>";
                }
                elseif ($row['priv']==2){
                    echo
"<script>window.top.location='cpanel.php'</script>";
                }
                elseif ($row['priv']==3){
                    echo
"<script>window.top.location='cpanel.php'</script>";
                }
                elseif ($row['priv']==4){
                    echo
"<script>window.top.location='index.php'</script>";
                }
            }
            elseif ($row['password']!=$password){
                $error="password incorrect";
            }
        }
        if (count($rows)==0) {
            $error = "username does not exist";
        }
    }
    catch (PDOException $e){
        echo $e->getMessage();
    }
}
```

Şekil 4.4. Giriş yapma işlevi için örnek kod

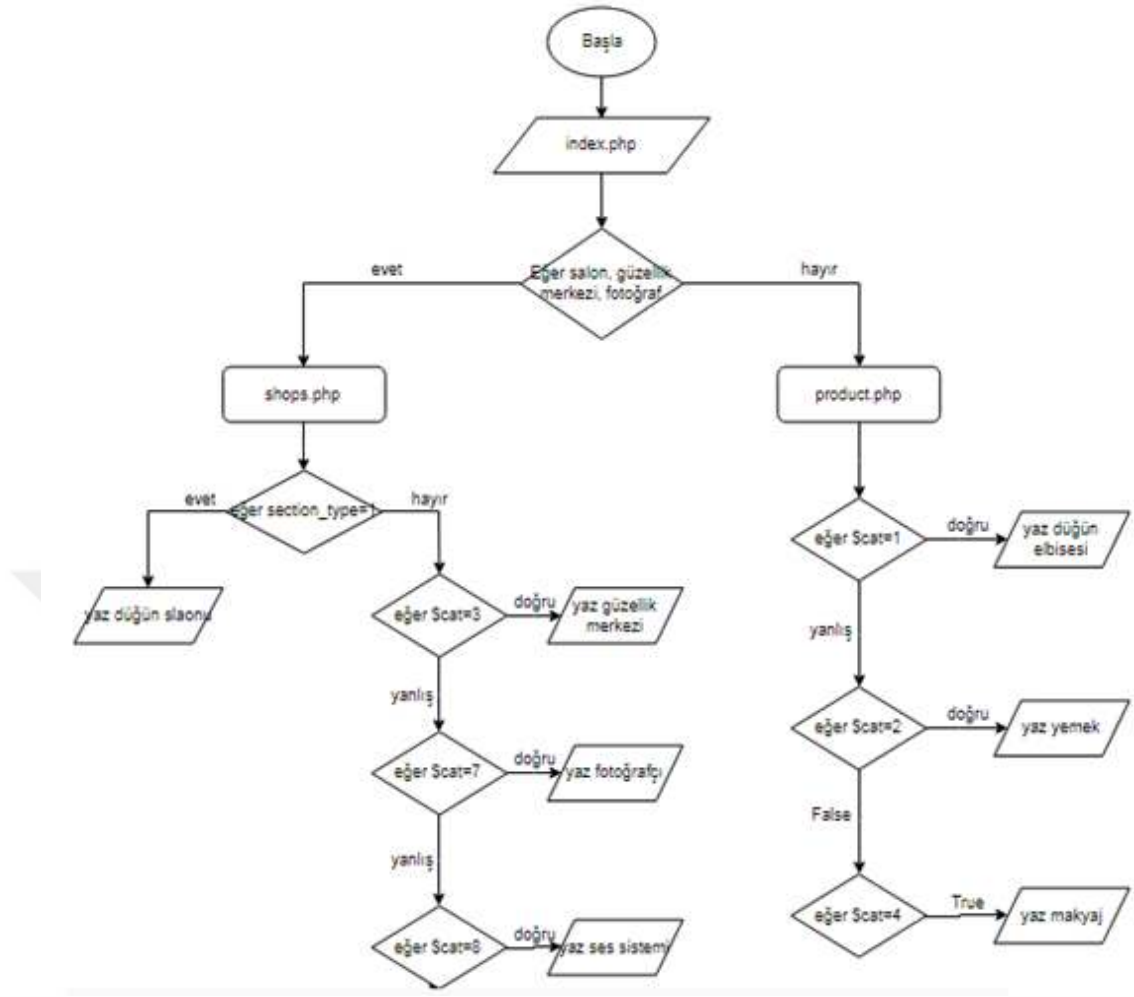
4.4. Gezinti ve Filtreleme Sonuçları

Şekil 4.5.'te gösterilen ekranda kullanıcılar ekranın en üstüne yerleştirilmiş olan dostça bir gezinme menüsünü kullanarak kategoriler arasında dolaşabilir ve ihtiyaçlarını kolayca bulabilir.



Şekil 4.5. Genel bir görünüm

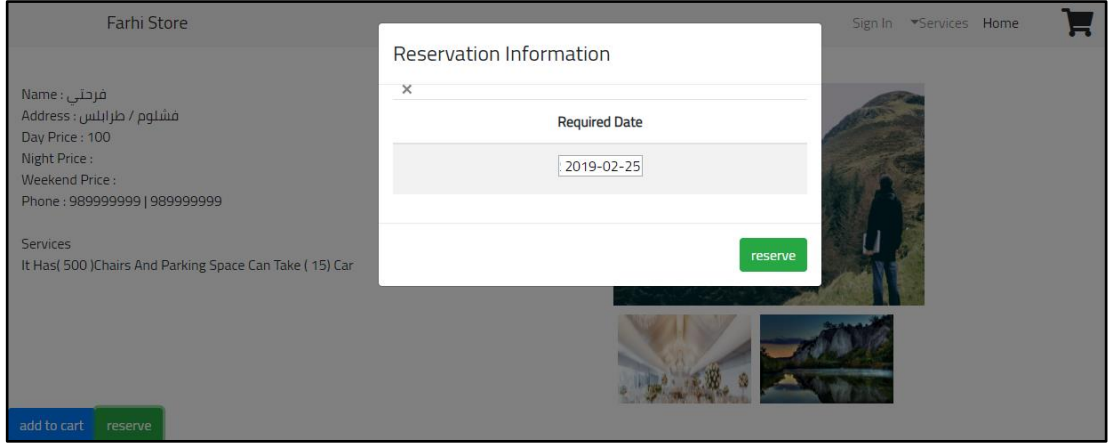
Kullanıcı belirli bir “kategori” servisine tıkladığında, sistem onları shops.php veya products.php sayfasına yönlendirir ve bu içeriğin hangi içeriğin gösterileceğini belirlemek ve belirlemek için http GET isteğini kullanarak (\$ section_type ve \$ cat) değişkenlerini gönderir. sayfaları. Shops.php içeriği iki parametreye bağlıdır; ilk parametre, içeriğin bir salon veya mağaza olup olmadığını belirlemek için \$ section_type değişkeni, ikinci parametre, bir mağaza olup olmadığını kategoriye belirlemek için \$ cat değişkenidir. Products.php içeriği, \$ cat olan bir parametreye bağlıdır. Şekil 4.5.'de sistemin nasıl dinamik içerik oluşturduğunu göstermektedir.



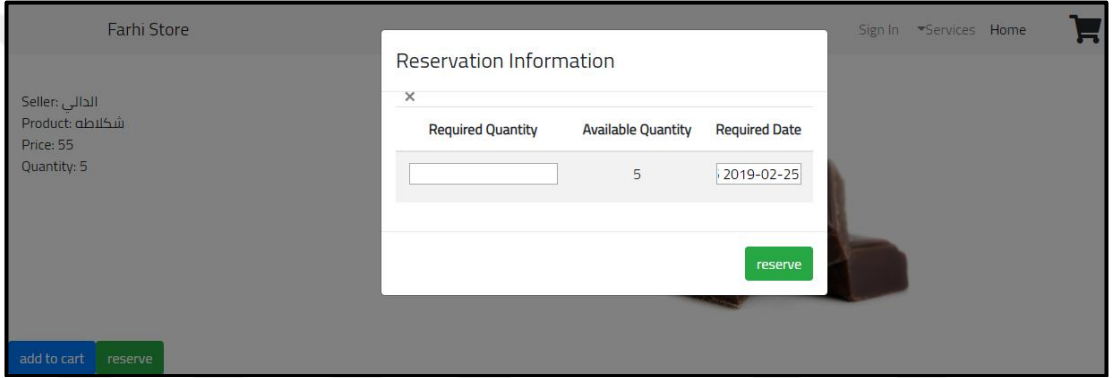
Şekil 4.6. İşlem sürecine ait bir iş akışı

4.5. Rezervasyon

Kullanıcılar dostça filtreler kullanarak herhangi bir ürüne/servise kolayca gidebilirler, bundan sonra rezerv düğmesine tıklayarak belirli bir tarihte (ve bir ürün rezervasyonunda gerekli olan miktar) rezervasyonu yapabilirler. Şekil 4.7. ve Şekil 4.8.'de rezervasyon modülleri gösterilmektedir.



Şekil 4.7. Rezervasyon işlemi



Şekil 4.8. Rezervasyon bilgileri

İstenilen tarih/adet doldurulduktan ve rezerv düğmesine tıklandıktan sonra, tarayıcı rezervasyonu tamamlamak için http POST isteği verilerini rezervasyon fonksiyonuna gönderir, rezervasyon kabul edene veya reddedene kadar bekleme durumunda olur. Şekil 4.9.'da bu işleme ait kod örneği gösterilmiştir.

```

function reservation($u_id,$section,$s_id,$qty,$date,$t_date,$conn) {
//add new reservation
try {
    if ($section==1) {$qty=NULL;}
    $stmt = "INSERT INTO reservation
(user_id,section_type,section_id,quantity, res_date,target_date)
VALUES($u_id,$section,$s_id,$qty,'$date','$t_date')";
    $conn->exec($stmt);

    ?>
<script>alert("successfully reserved ")</script><?php
}

catch (PDOException $e) {
    echo $e;
}
}

```

Şekil 4.9. Rezervasyon işlem kod örneği

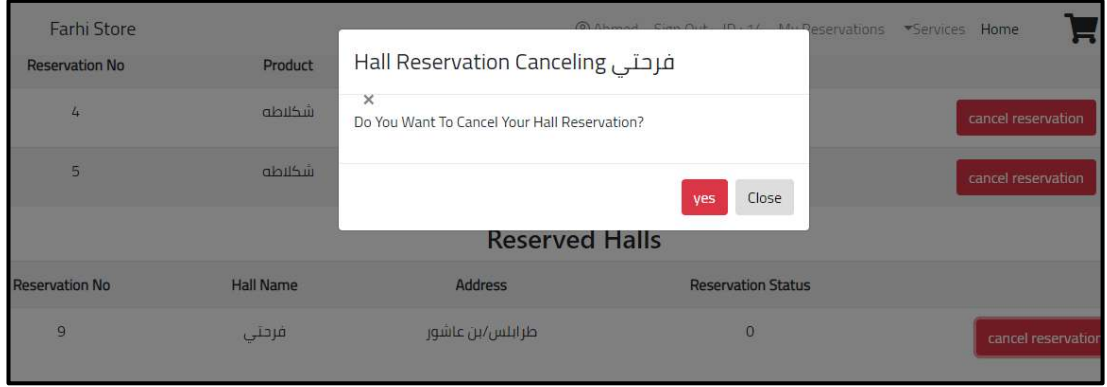
4.6. Rezervasyon Kontrolü

Kullanıcılar gezinme menüsü üzerinden booking.php sayfasına giderek, ayrılmış ürünlerini / salonlarını gösterebilir, böylece kabul edilip edilmediğini, iptal edildiğini veya hala beklediğini bilmek için rezervasyon durumunu kontrol edebilirler. Şekil 4.10.'da gösterilmiştir. Şekil 4.11.'de rezervasyon iptal işlemi gösterilmiştir.

Farhi Store

</

Şekil 4.10. Rezervasyon durumu



Şekil 4.11. Rezervasyon iptali

4.7. Profil Yönetimi

Genel Kullanıcılar, ad, e-posta, telefon ve şifre gibi kişisel bilgilerini yönetebilir. Servis sağlayıcılar ise yöneticiler sağlanan servisleri yönetebilirler. Şekil 4.12.'de profil yönetimi gösterilmiştir.

Şekil 4.12. Servislerin yönetimi

Sistem ile gerçekleştirilen servislerin belirli ücretleri bulunmaktadır. Bu ücretlendirme için Şekil 4.13.'de gösterilmiş olan ücret yönetimi modülü kullanılmaktadır.

Prices Services Photos Reservations

: Day Price 1000 : Night Price 4000

Discount : Discount Per

Discount Until : 2019-01-20 1/20/2019 % 11

UPDATE

Şekil 4.13. Ücret yönetimi

4.8. Ürün Yönetimi

Yöneticiler, ürünlerini Şekil 4.14.'de gösterildiği şekilde yönetmektedirler.

New Product Products Reservations

New Product

: Name : Category

enter product name

Specifications : Color

Size : Color

Price : Quantity

\$ price quantity

: Selling Type

to rent

Discount : Discount Per

Discount Until : dd/mm/yyyy %

ADD ADD PHOTOS

Şekil 4.14. Ürün yönetimi

4.9. Rezervasyon Yönetimi

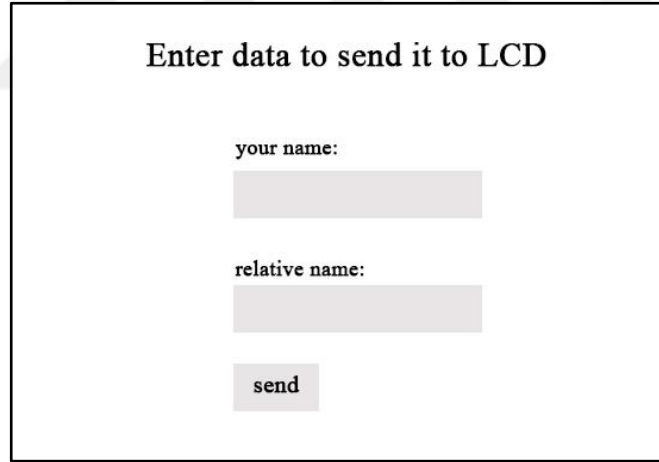
Yöneticiler müşterilerinin rezervasyon durumunu kontrol edebilir, işlem yapmadan önce bekleme listesinde olması gereken kullanıcıların rezervasyonlarını onaylayabilir

veya reddedebilirler. Eğer yönetici birinin rezervasyonunu onaylarsa, otomatik olarak kendisine bir SMS gönderilir.

4.10. Web Sunucusu Olarak NodeMCU Kullanımı

NodeMCU programlanabilir bir Wi-Fi modülüdür, Wi-Fi ağına bağlanabilir ve HTTP protokolü üzerinden web barındırma hizmeti sağlayabilir. Bu hizmet özellikle kadınlar içeri girerken salona giremeyen “erkekler” kullanıcılarına uygulanmakta olup, bu hizmet kullanıcıların mikrokontrolörün IP adresini girerek HTTP GET isteğini kullanarak salondaki akrabalarıyla iletişim kurmasını sağlayacaktır web tarayıcısı, kullanıcılar genel bir IP varsa, mikrodnetleyiciye aynı kablosuz ağ üzerinden (WLAN) veya internet üzerinden erişebilirler.

Mikrodnetleyicinin kullanıcı arayüzüne eriştikten sonra, kullanıcılar adlarını ve akrabalarının adlarını Şekil 4.15.’de gösterilmiştir.



The image shows a web interface for a NodeMCU microcontroller. It has a title "Enter data to send it to LCD". Below the title, there are two input fields. The first is labeled "your name:" and the second is labeled "relative name:". Below these fields is a button labeled "send".

Şekil 4.15. NodeMCU kullanıcı arayüzü

Ardından web tarayıcısı HTTP GET isteğini kullanarak verileri mikrodnetleyiciye gönderir. Ondan sonra, mikrodnetleyici verileri işlemde geçirir ve isimleri görüntülemek için 16X2LCD'ye bağlı seri adaptör I2C'ye gönderir. LCD'yi NodeMCU'ya bağlamak için 16 kablolu bağlantı gerekir, I2C seri adaptör bağlantısı sayesinde yalnızca bir tane topraklama pimi için biri voltaj kaynağı pimi için diğeri veri için iki kablo alır (“NodeMCU - 5V çıkışlı Vin pimi”).

NodeMCU kartı Arduino IDE yazılımı kullanılarak programlanabilir, Arduino IDE'yi başlattıktan sonra, kartı programlamak için ESP8266 kütüphanesi kurulmalıdır. Ardından pano tipi NodeMCU 1.0 (ESP-12module) olarak değiştirilmelidir. Başka kütüphanelerin, LCD'ye veri gönderen seri adaptörü tanımlayan HTTP ve LiquidCrystal_I2C.h olan panodaki portu (80) açmak için ESP8266WebServer gibi dahil edilmesi gerekir.

4.11. SMS Yönetimi

Kısa Mesajlar, çevrimdışı iletişim için günlük hayatımızın bir parçası oldu. Artık çoğu servis sağlayıcı internet üzerinden SMS API kullanarak metin mesajları gönderiyor, herhangi bir web tabanlı sistemden istemcilere SMS göndermek ve göndermek için, sisteminizi kendi API'larına entegre etme olanağı sağlayan bir SMS ağ geçidi sağlayıcısı olmalı (Tandon, 2012).

Bu sağlayıcılar üçüncü taraf olarak kabul edilir, pek çok ücretsiz sağlayıcı vardır. Ancak yerel iletişim şirketleri medya ortamlarını kullanmalarını engellemeye devam ettiklerinden, ön ödemeli bir SMS ağ geçidi sağlayıcısı araştırılmıştır. Geliştirilen sisteme SMS ağ geçidinin API'ları kullanılarak bütünleştirilmiştir. Böylece sistemde rezervasyonlar onayladıktan sonra rezervasyon makbuzları gerekli müşterilere otomatik olarak bilgi amaçlı SMS gönderilmektedir. Yönetici tarafından yapılan rezervasyon onayından sonra, mesajı son kullanıcının telefonuna yönlendirecek olan SMS ağ geçidi sağlayıcısının API'sine bir mesaj gönderilecektir.

5. SONUÇLAR

Web tabanlı otomasyon sistemi günümüzde birçok alanda uygulanmaktadır. Sistemin yaygınlaşması sunmuş olduğu avantajlardan kaynaklanmaktadır. Avantajların en önemlisi sistem kullanıcılarının donanım bağımlılığı olmamasıdır. Kullanıcılar dünyanın her yerinde internet bağlantısı ile bilgisayar ya da akıllı telefonda web tabanlı otomasyon sistemini kullanabilmektedirler.

Bu çalışmanın amacı, düğün düzenlenmesi sorununun mevcut durumunu değerlendirmek, bu sorunla ilgili mevcut uygulamaları analiz etmek, son kullanıcının tüm ihtiyaçlarını karşılayan bir web tabanlı uygulama tasarlamak ve ardından prototipin uygulanmasını sağlamaktır.

Akıllı düğün düzenleme uygulaması SWAA, sistemi kurmak için tamamen açık kaynaklı PHP ve MySQL Server kullanılarak geliştirilen önceki bölümlerde başarılı bir şekilde geliştirilmiş ve gerekli işlevleri sağlamıştır. Libya'daki imkanlar gözönünde bulundurularak kaynak yetersizliği ve sınırlı zaman dilimine rağmen, sistem başarılı bir şekilde uygulanmıştır ve işlevselliği web tabanlı olarak test edilmiştir.

Bu projenin çalışması sırasında ortaya çıkabilecek yetersizlikleri doldurmak için gelecekteki geliştirmeler yapılabilir. Projede yer alan her adımı takip ederek, bu konuyla ilgilenen diğer öğrenciler gelecekteki bazı çalışmalarını takip edebilirler. Sistem, tüm dünyada uluslararası alanda kullanılabilmesi için daha fazla çalışma ve geliştirme ekleyecek kadar esnektir.

KAYNAKLAR

- Afroz, Dr. S. M., Rani, E. N., ve Priyadarshini, I. (2011). Web application-A study on comparing software testing tools, *International Journal of Computer Science and Telecommunications*, 2(3), 1-6.
- Alkhaldi, D., Aldossary, H., Alsmadi, M.K., Al-Marashdeh, I., Badawai, U., et al. (2018). Developing and implementing web-based online university facilities reservation system, *International Journal of Applied Engineering Research*, 13(9), 6700-6708.
- Bergsten, H. (2002). *JavaServer Pages*, Publisher: O'Reilly, 608.
- Bruno, V., Tam, A., ve Thom, J. (2005). Characteristics of web applications that affect usability: A review, 2005 Australasian Computer-Human Interaction Conference.
- Chaudhari, G. R., ve Vaidya, M.V. (2014). A survey on security and vulnerabilities of web application, *International Journal of Computer Science and Information Technologies*, 5(2): 1856-1860.
- Demirel, M. Y., ve Karaağaç, İ. (2014). Bilgisayar destekli üretim süreçlerine genel bir bakış. *Engineer & the Machinery Magazine*, 1(652), 51-61.
- Druzdzel, M. J., ve Flynn, R. R. (1999). *Decision support systems*. Encyclopedia of library and information science. A. Kent. Marcel Dekker, Inc. Last Login, 1999. 10(03), 2010p.
- Esener, H. (2005). Hizmet yönetimi sistemi, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, İstanbul.
- Fielding, R. (2009). Hypertext Transfer Protocol-http/1.1. Network Working Group.
- Gökşen, Y., ve Aşan, H. (2015). Veri büyüklüklerinin veritabanı yönetim sistemlerinde meydana getirdiği değişim: NOSQL. *Bilişim Teknolojileri Dergisi*, 8(3), 126.
- Hersh, M. (1999). Sustainable decision making: the role of decision support systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 29(3), 395-408.
- Kishore, P., Veeramanikandasamy, T., Sambath, K., ve Veerakumar, S. (2017). Internet of Things based low-cost real-time home automation and smart security system, *International Journal of Advanced Research in Computer and Communication Engineering*, 6(4), 505-509.

- Kosariev, N. (2018). Reservation System for Football Matches, Assignment of Bachelor's Thesis, *Faculty of Information Technology CTU in Prague*, 77.
- Kruchten, P. (1995). The 4+1 View Model of Architecture, *IEEE Software*, 12(6), 45-50.
- Mordani, R. (2009). Java Servlet Specification, Sun Microsystems Inc., Version 3.0.
- Networks, R., ve Spivack, N. (2007). How the WebOS evolves? 29/09/2018 tarihinde http://novaspivack.typepad.com/nova_spivacks_weblog/2007/02/steps_towards_a_adresinden_alinmistir.
- Pekgöz, N. (2013). *Java ile dinamik web sayfaları oluşturun*, Pusula Yayıncılık, 400.
- Sathyaseelan, B., ve Cordova, R.S. (2016). A comparative study of top web design models that are using Java technologies, *International Journal of Innovations & Advancement in computer Science*, 5(5), 41-44.
- Schwarz, M. (2013). Design and analysis of web application frameworks, PhD Dissertation Aarhus University, Denmark.
- Tandon, S. (2012). Application of incoming SMS to a website, control the web site to send bulk SMS, *International Journal of Computer Science and Information Technologies*, 3, 4287-4289.
- Yao, T. (2008). An introduction to web-based support systems, *Journal of Intelligent Systems*, 17(1-3), 267-281.
- Yılmaz, H. (2014). Bilgi liderliğinin işletme performansı üzerine etkilerinin değerlendirilmesi. *Journal of Economics & Management Sciences/Ekonomi ve Yönetim Bilimleri Dergisi*, 1(1), 51.

ÖZGEÇMİŞ

Adı Soyadı : Ahmed M.M. BUKER
Doğum Yeri ve Yılı : Tripoli, LİBYA 1995
Medeni Hali : Bekar
Yabancı Dili : Arapça, İngilizce
E-posta : ahmadm199@gmail.com



Eğitim Durumu

Lise : Zawit-Aldahmani Lisesi, Libya, 2012
Lisans : Tripoli Elektronik Teknoloji Koleji, Libya, 2017