

**DÜZCE ÜNİVERSİTESİ  
TEKNOLOJİ FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ**

**PYTHON FLASK FRAMEWORK İLE SQLALCHEMY  
KULLANARAK E-TİCARET SİTESİ**

**172114009 Betül Saliha KUZU  
172114013 Cihan ALBARAK  
172114205 Mücahit YILMAZ**

**BMT327 BETİK DİLLER DERS PROJESİ**

**Ocak 2020**

## İÇİNDEKİLER

GİRİŞ .....	4
PROJENİN TANIMI VE PLANI .....	5
2.1 Projenin Tanımı .....	5
2.2. Projenin Planı ve Zamanlama.....	5
KULLANILAN TEKNOLOJİLER .....	6
3.1 Flask .....	6
3.1.1 Temel Proje Dizini.....	6
3.1.2 Flask Uygulama Nesnesi .....	6
3.1.3 URL ve Görünüm İşlevleri .....	6
3.1.4 Flask Arayüz Şablonları .....	7
3.1.5 redirect ve url_for Tanımlı Fonksiyonları .....	7
3.1.6 Request Nesnesi.....	8
3.1.7 Sözlük Nesnesi.....	8
3.1.8 Flask ile SQLAlchemy Modülü .....	8
3.1.9 Flash Nesnesi (Mesajların Gösterilmesi).....	9
3.1.10 Flask User Modülü .....	10
3.1.11 Flask Login Modülü .....	10
3.1.12 Flask Babalex Modülü .....	11
3.1.13 secure_filename .....	12
3.2 Css (Cascading Style Sheets – Basamaklı Stil Sayfası) .....	12
3.3 Javascript (JS).....	12
3.4 jQuery .....	13
3.5 HTML ve HTML5.....	13
3.6 SASS (Syntactically Awesome Style Sheets - Sözdizimsel Müthiş Stil Sayfaları) .....	13
3.7 Bootstrap .....	13
4.1 Kullanım Senaryosu .....	14
TASARIM VE KODLAMA .....	16
5.1 Veritabanı Tasarımı .....	16
5.2 Veri Modeli .....	17
SONUÇ .....	18
KAYNAKÇA.....	19

## ŞEKİLLER

Şekil 1 Projenin çubuk diyagram ile gösterilmesi.....	5
Şekil 2 route() dahili işlevi.....	7
Şekil 3 URL alanında değişken kullanabilme.....	7
Şekil 4 redirect ve url_for fonksiyonu kullanımı.....	8
Şekil 5 request ile route() dahili işlev kullanımı.....	8
Şekil 6 Sözlük nesnesi kullanımı.....	8
Şekil 7 Veritabanı nesnesinin oluşturulması.....	9
Şekil 8 Veritabanı db.Model sınıfı yazımı.....	9
Şekil 9 Flask User nesnesinin oluşturulması.....	10
Şekil 10 Flask Login nesnesinin oluşturulması.....	11
Şekil 11 Flask Babalex dahil edilmesi ve nesne oluşturulması.....	11
Şekil 12 secure_filename işlevinin projeye dahil edilmesi.....	12
Şekil 13 Flask E-Ticaret web uygulaması modellenmesi.....	14
Şekil 14 Kullanım Senaryosu.....	15
Şekil 15 Veritabanı Tasarımı.....	16
Şekil 16 Varlık-İlişki Diyagramı.....	17

## GİRİŞ

Günümüzde internet kullanımı önemli bir yere sahiptir. İnternet bilgi edinme, araştırma, sosyalleşme ve alışveriş gibi çeşitli konularda çokça kullanılan bir yardımcıdır. İnternet üzerinden yemek, elektronik, kıyafet ve market gibi birçok farklı ürün siparişi veriyor, her gün bu alışverişlerde kredi kartı kullanılıyor.

İnternette alışverişlerimizin birçoğunu elektronik, kitap ve kıyafet gibi e-ticaret alışverişleri oluşturmaktadır. E-Ticaret, bir ürün veya hizmet için bir internet platformu üzerinden alışveriş yapmaya sağlayan bir yöntemdir.

Bu çalışmada alışveriş de önemli yere sahip olan internet platformu kodlanacaktı

Bu çalışmada kodlama için, nesne tabanlı ve fonksiyonel programlamayı destekleyen bir programlama dili olan Python kullanılmıştır.

Python nesne tabanlı, yorumlanabilir, modüler ve kullanışlılığı, yüksek seviyeli bir programlama dildir. Programlama dilinin makine çalışmasına ve mantığına daha yakın olması, daha hızlı çalışması düşünülerek kullanılmıştır.

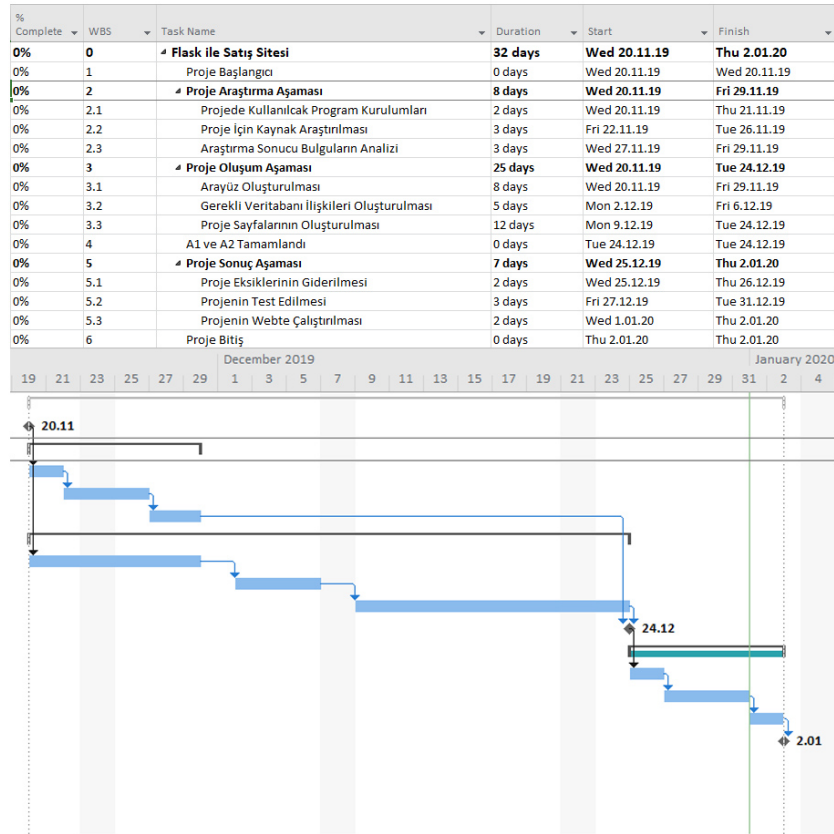
Bu çalışmada internet platformunda hızlı sonuç elden edebilme, performans açısından daha iyi sonuçlar alabilmek için esnek yapısıyla Python Flask Frameworkünden yararlanılmıştır.

## PROJENİN TANIMI VE PLANI

### 2.1 Projenin Tanımı

Bu projede Python Flask Framework ile SQLAlchemy kullanılarak, çeşitli ürünlerin sergilenebileceği ve potansiyel müşterilerin alışveriş yapabilecekleri, E-Ticaret internet platformu kodlanacaktır.

### 2.2. Projenin Planı ve Zamanlama



Şekil 1 Projenin çubuk diyagram ile gösterilmesi

## KULLANILAN TEKNOLOJİLER

### 3.1 Flask

Flask, Werkzeug ve Jinja2 tabanlı bir mikro web çatısıdır. Bünyesinde geliştirme sunucusu, hata ayıklama ve birim test desteği gibi özellikleri bulundurur. Web çatıları, web uygulamalarını çalıştırmak için gerekli olan genel işlemleri doğrudan kendileri veya eklentileri aracılığıyla sağlarlar.

#### 3.1.1 Temel Proje Dizini

Python Flask web uygulamalarında genel olarak kabul edilen bilgiye göre .html uzantıya sahip dosyalar “templates”, .css, .js uzantılı dosyalar veya resim dosyaları “static” adlarıyla klasörler halinde tutulur. Genellikler bilinen ve kullanılan temel proje dizini şu şekildedir:

```
/proje
run.py
/static
    style.css
/templates
    base.html
    index.html
```

#### 3.1.2 Flask Uygulama Nesnesi

Python Flask nesnesinin bir örneği olarak kullanılır. Web uygulamalarının temel nesnesidir. Parametre olarak web uygulama paketinin ismini alır. Tek bir paket kullanılıyorsa parametre olarak “\_\_name\_\_” verilmelidir.

#### 3.1.3 URL ve Görünüm İşlevleri

route() işlevi ile belirtilen bir URL adresini belirtilen fonksiyon ile bağlamak için kullanılır.

```
@app.route('/iletisim')
def iletisim():
    return render_template('contact.html')
```

Şekil 2 route() dahili işlevi

Şekil 2'deki örnek uygulanıp çalıştırıldığında domain/iletisim URL adresine gidildiğinde iletisim fonksiyonun çalıştığı görülecektir.

Route dahili işlevi ile URL alanında değişken kullanabilmek için Şekil 2'deki gibi <degisken\_adi> kullanılabilir.

```
@app.route('/<urun>')
def urun_goster(urun):
    return 'Ürün %s' % urun
```

Şekil 3 URL alanında değişken kullanabilme

Görünüm fonksiyonları ise route() işlevinin bir URL adresi ile eşleştiği fonksiyonlardır.

### 3.1.4 Flask Arayüz Şablonları

Şablonlar Flask web uygulamasının arayüzünü oluşturan .html uzantılı dosyalardan oluşmaktadır. Flask web uygulamalarında genellikle .html uzantılı dosyalar “templates” klasörü altında bulunurlar.

### 3.1.5 redirect ve url\_for Tanımlı Fonksiyonları

redirect fonksiyonu ziyaretçiyi farklı bir URL adresine yönlendirmek için kullanılır. url\_for fonksiyonu ise belirli bir fonksiyonu çağırmak için URL adresi üretme amacıyla kullanılır. Bu fonksiyon ilk parametre olarak fonksiyon adını almaktadır.

```

from flask import redirect, url_for

@app.route('/index')
def index():
    return redirect(url_for('login'))

```

Şekil 4 redirect ve url\_for fonksiyonu kullanımı

### 3.1.6 Request Nesnesi

Bir URL adresi için istek oluşturulduğunda bu adres ile ilgili tüm bilgileri global olarak “request” nesnesinde barındırır. “methods” özelliği kullanılarak istek tipi belirlenebilir. İstek tipi route() dahili işlevi parametre olarak gönderilmelidir.

```

@app.route('/login', methods=['GET', 'POST'])

```

Şekil 5 request ile route() dahili işlev kullanımı

### 3.1.7 Sözlük Nesnesi

Flask sözlük nesnesi, içerisinde anahtar ve değeri olmak üzere veri barındırmak için kullanılır. Sözlük nesnesi ile barındırılan veriye tanımlanırken kullanılan anahtar ile yeniden erişim yapılabilir.

```

Cart.products[urun] = {
    'id': urun,
    'adi': request.form['adi'],
    'fiyat': float(request.form['fiyat']),
    'adet': int(request.form['adet']),
    'toplam': float(request.form['fiyat']) * int(request.form['adet'])
}

```

Şekil 6 Sözlük nesnesi kullanımı

### 3.1.8 Flask ile SQLAlchemy Modülü

SQLAlchemy modülü Python için yazılmış, geliştiricilerin kullanımına sunulmuş, açık kaynak kodlu SQL ve ORM (Object Relational Mapping – Nesne İlişkisel Eşleme) yardımcı aracıdır. Bu projede Flask E-Ticaret internet platformu kodlamasında SQLAlchemy kullanabilmek için flask\_sqlalchemy modülünden yararlanılır.



Veritabanı oluşturmadan önce dizinde bulunan başlangıç dosyası olan app.py içinde veritabanı işlemlerinde kullanılacak nesne Şekil 7’deki gibi oluşturulur.

```
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config.from_object(__name__+'.ConfigClass')

db = SQLAlchemy(app)
```

Şekil 7 Veritabanı nesnesinin oluşturulması

Oluşturulan db nesnesi ile Veritabanı ile ilgili tüm işlemler içeri aktararak kullanılmalıdır. Veritabanı tabloları Python sınıf olarak oluşturulur, db.Model sınıfından dinamik olarak türetilir. Şekil 8’de örnek verilmiştir.

```
class User(db.Model, UserMixin):
    __tablename__ = 'users'
    id = db.Column(db.Integer(), primary_key=True)
    active = db.Column('is_active', db.Boolean(), nullable=False, server_default='1')
    email = db.Column(db.String(255, collation='NOCASE'), nullable=False, unique=True)
    email_confirmed_at = db.Column(db.DateTime())
    password = db.Column(db.String(255), nullable=False, server_default='')
    name = db.Column(db.String(100, collation='NOCASE'), nullable=False, server_default='')
    phone = db.Column(db.String(30), server_default='')
    roles = db.relationship('Role', secondary='user_roles')
```

Şekil 8 Veritabanı db.Model sınıfı yazımı

Flask ile SQLAlchemy kullanırken iki tablo arasında ilişki kurulma yapılırken db.ForeignKey() ve relationship() işlevleri kullanılmalıdır.

### 3.1.9 Flash Nesnesi (Mesajların Gösterilmesi)

Ekranda mesajları göstermek için Python Flask içinde dahili olarak bulunan flash nesnesi kullanılır. Bu dahili fonksiyon parametre olarak ekran gösterilmesi istenen mesajı alır. Bu projede web uygulama arayüzünde kullanıcının oluşturduğu işlevlerin bilgilerini göstermek için kullanılır.

### 3.1.10 Flask User Modülü

Flask User modülü Python için kodlanmış, açık kaynak kodlu bir kullanıcı aracıdır. Bu projede kullanılabilmesi için Flask-User modülü dahil edilir.

Flask User modülü web uygulamalarında kullanıcı ve rol tabanlı yetkilendirme işlemlerini sunmaktadır. Eklentiyle birlikte projeye eklenecek ek işlevler şunlardır:

- Kullanıcı adı ve / veya e-posta ile kullanıcı kaydı
- E-posta onayı
- Kullanıcı adı değiştirme
- Kullanıcı girişi ve kaydı
- Kullanıcı şifre yenileme / değiştirme
- Kullanıcı şifre unuttum

Flask User modülü dahil edildikten sonra dizinde bulunan başlangıç dosyası olan app.py içinde Flask web uygulamasında kullanıcı işlemlerinde kullanılacak nesne Şekil 9'deki gibi oluşturulur.

```
# Setup Flask-User
user_manager = UserManager(app, db, User)
```

Şekil 9 Flask User nesnesinin oluşturulması

### 3.1.11 Flask Login Modülü

Flask Login modülü Python için kodlanmış, geliştiricilerin kullanımına sunulmuş, açık kaynak kodlu kullanıcı oturum yönetim eklentisidir. Bu projede kullanılabilmesi için flask-login modülü dahil edilmelidir.

Oturum açma, oturum kapatma ve kullanıcı oturumlarının uzun süre boyunca hatırlama gibi genel görevleri yerine getirir. Flask Login ile projenize eklenecek işlevler şunlardır:

- Etkin kullanıcının kimliği oturumda güvenli olarak saklar

- Projede bulunan görünüm fonksiyonlarında oturumu açmış (veya çıkış yapmış) kullanıcılarla yetkilendirme yapar
- “Beni Hatırla” işlevselliğini projenize dahil eder

Flask Login modülü projeye dahil edildikten sonra dizinde bulunan başlangıç dosyası ile app.py içinde kullanıcı oturum işlemlerinde kullanılacak nesne Şekil 10’daki gibi oluşturulur.

```
app = Flask(__name__)

login_manager = LoginManager()

login_manager.init_app(app)
```

Şekil 10 Flask Login nesnesinin oluşturulması

### 3.1.12 Flask Babalex Modülü

Flask Babalex modülü, Python için geliştirilmiş, açık kaynak kodlu, web uygulamalarına i18n ve i10n desteği sunan bir modüldür. Saat dilimi işlemleri ile tarih biçimlendirme işlemleri için dahili desteğin yanı sıra metin çevirilerinde kullanmak için çok basit ve kolay bir arayüze sahiptir. Bu projede Flask web uygulama arayüzünde oluşan uyarı ekranlarında Türkçe dil desteği sunması amacıyla kullanılmaktadır.

Projede kullanılabilmesi için Flask-BabelEx modülü dahil edilmesi ve dizinde bulunan başlangıç dosyasında nesne oluşturma işlemi tamamlanmalıdır.

```
from flask import Flask
from flask_babel import Babel

app = Flask(__name__)
babel = Babel(app)

@babel.localeselector
def get_locale():
    translations = [str(translation) for translation in babel.list_translations()]
```

Şekil 11 Flask Babalex dahil edilmesi ve nesne oluşturulması

### 3.1.13 secure\_filename

Bu modül, kullanıcı tarafından form aracılığıyla seçilen herhangi bir dosyayı proje dizininde seçilen yere yüklemeye ve ilgili dosya adını güvenli olarak belirleme için ek işlevler desteği sağlar.

Bu projede Flask web uygulamasında yönetici rolüne sahip kullanıcının seçimini yaptığı .jpg, .png, ve .jpeg uzantılı resim dosyalarını yüklemek için kullanılır.

```
from werkzeug.utils import secure_filename
```

Şekil 12 secure\_filename işlevinin projeye dahil edilmesi

### 3.2 Css (Cascading Style Sheets – Basamaklı Stil Sayfası)

CSS, HTML’e ek olarak metin ve format biçimlendirme alanlarında fazladan yetenekler sunan web teknolojisidir. HTML ile oluşturulan öğelere stil vermek için kullanılır.

Bu projede kullanıcı arayüz şablonlarında tasarımsal düzenlemeler ve değişiklikler yapmak için kullanılmıştır. Yapılan değişiklikler “style.css” dosyası adı altında, tek bir dosyada toplanmıştır. Bu sayede güncellenmesi kolaylaşmış ve hafızadaki boyutu küçük tutulmuştur.

### 3.3 Javascript (JS)

Javascript prototip tabanlı, birinci sınıf fonksiyonlara sahip, genel olarak web tarayıcılarında kullanılan dinamik betik programlama dilidir. Javascript ile kodlanan istemci tarafı betikler sayesinde kullanıcıyla etkileşim halinde olması, eşzamansız bir şekilde sunucuyla iletişime geçmesi ve web sayfasının içeriğini değiştirilmesi işlevleri sunar. Fonksiyonel ve nesne yönelimli programlama kurallarına sahiptir.

### **3.4 jQuery**

John Resig ile 2006 yılında geliştirilmeye başlanılan ve şu an jQuery ekibi tarafından geliştirilmesi devam eden açık kaynak kodlu javascript kütüphanesidir.

Bu projede kullanıcı web arayüzünde listeleme ve bilgilendirme metinlerde eşzamansız olarak sunucu ile değiştirilmesi amacıyla kullanılmıştır.

### **3.5 HTML ve HTML5**

HTML (Hyper Text Markup Language – Hiper Metin İşaretleme Dili), web sayfaları oluşturmak için kullanılan standart bir dildir.

HTML5, internetin teknolojilerinden olan HTML işaretleme standardının beşinci sürümüdür. Daha önceki sürümlerde yetersiz kalan ihtiyaçları karşılamak adına geliştirilmiştir.

### **3.6 SASS (Syntactically Awesome Style Sheets - Sözdizimsel Müthiş Stil Sayfaları)**

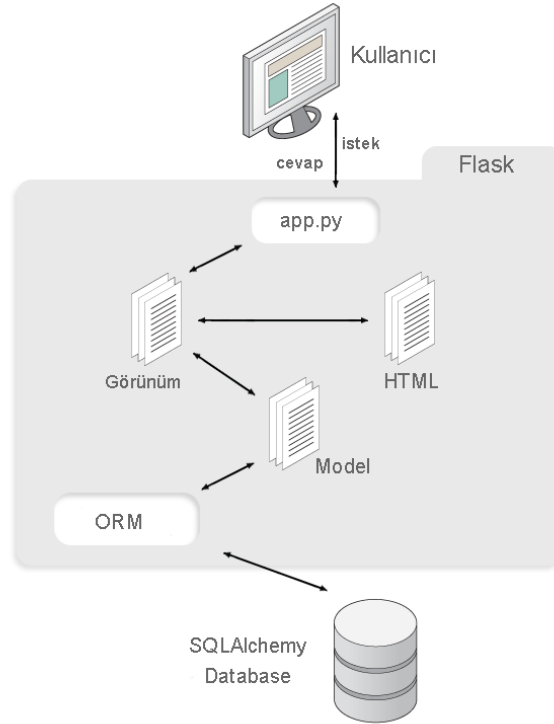
SASS, kolayca CSS formatına derlenebilen dinamik bir biçim şablon dilidir. İlk sürümleri yayımlandıktan sonra içinde bulunan SassScript geliştirilerek SASS'ın yetenekleri artırılmıştır. Bu projede CSS ile tasarlanan web arayüzünde tasarımsal yetenekleri artırmak için kullanılmıştır.

### **3.7 Bootstrap**

Bootstrap (Twitter Bootstrap), geliştiricilerin kullanımına sunulmuş, açık kaynak kodlu, web sayfaları ve uygulamaları geliştirmek için kullanılabilecek ön yüz ve araçlar bütünü kapsayan bir web çatısıdır. Bootstrap, web sayfalarında ve uygulamalarında kullanılabilecek HTML ve CSS tabanlı şablonlar içerir. Bu projede kullanıcı arayüzünde gösterilen HTML sayfaları, Bootstrap'ın sunduğu elemanların kullanımıyla tasarlanmıştır.

## MODELLEME

MVC (Model – View - Controller), kodlanan uygulamanın iş mantığı ile kullanıcı arayüzünü birbirinden ayıran yazılım mimarisidir. Model, Flask uygulamasının web arayüzü ile veritabanı arasında aracı görevi gören parçasıdır. Teknik açıdan, web uygulamasının veri alanı için mantığı uygulayan nesnedir. Bu projenin modellemesi Şekil 13’de örneklendirilmiştir.



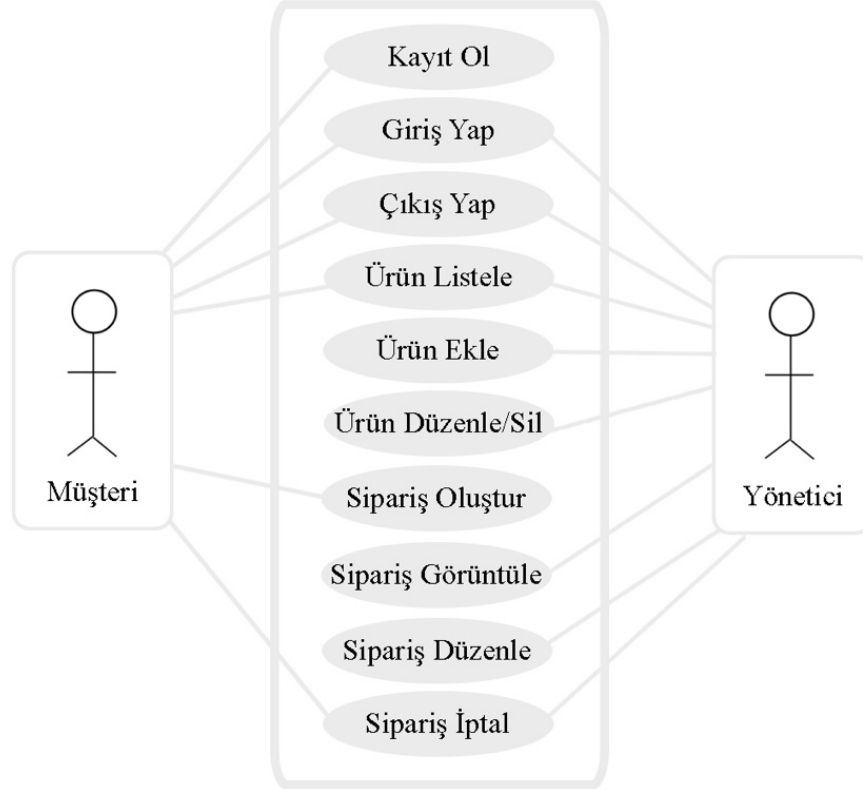
Şekil 13 Flask E-Ticaret web uygulaması modellemesi

### 4.1 Kullanım Senaryosu

Flask E-Ticaret web uygulamasında müşteri ve yönetici olmak üzere iki tip kullanıcı bulunmaktadır. Kullanım senaryo diyagramı Şekil 14’de gösterilmiştir.

Birinci tip kullanıcı olan müşteri, uygulamaya kayıt olabilir, giriş veya çıkış yapabilir, uygulamada bulunan ürünleri listeleyebilir, seçtiği herhangi bir ürün ile sipariş oluşturabilir. Oluşturduktan sonra dilerse kendi siparişini iptal edebilir.

İkinci tip olan yönetici ise giriş veya çıkış yapabilir, ürünleri listeleyerek sipariş oluşturabilir. Uygulamada yer alan ürünleri düzenleyebilir. Oluşturulan siparişleri listeleyebilir, seçilen siparişi düzenleyebilir.

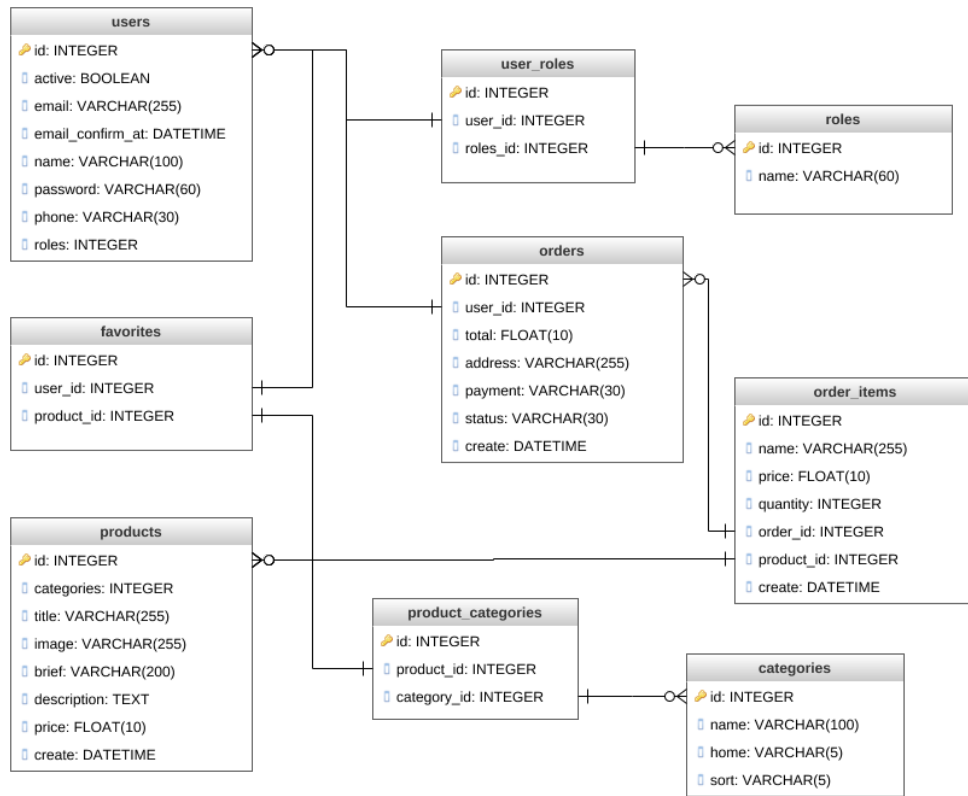


Şekil 14 Kullanım Senaryosu

## TASARIM VE KODLAMA

### 5.1 Veritabanı Tasarımı

Şekil 15’de gösterilen users tablosu web uygulamasında kullanıcı bilgilerinin tutulduğu tablodur. roles alanı “User” ise müşteri, “Admin” ise yönetici olduğunu gösterir. products tablosu yöneticinin oluşturduğu ve web uygulamasında ziyaretçiye sergilenen ürün bilgilerinin tutulduğu tablodur. orders tablosu müşterinin oluşturduğu siparişlere ait genel bilgiyi, order\_items tablosu ise siparişlere ait detaylı bilgileri tutmaktadır. favorites tablosu müşterilerin seçtiği ürünü favoriye aldığı bilgileri kullanıcı bilgileriyle birlikte tutmaktadır.

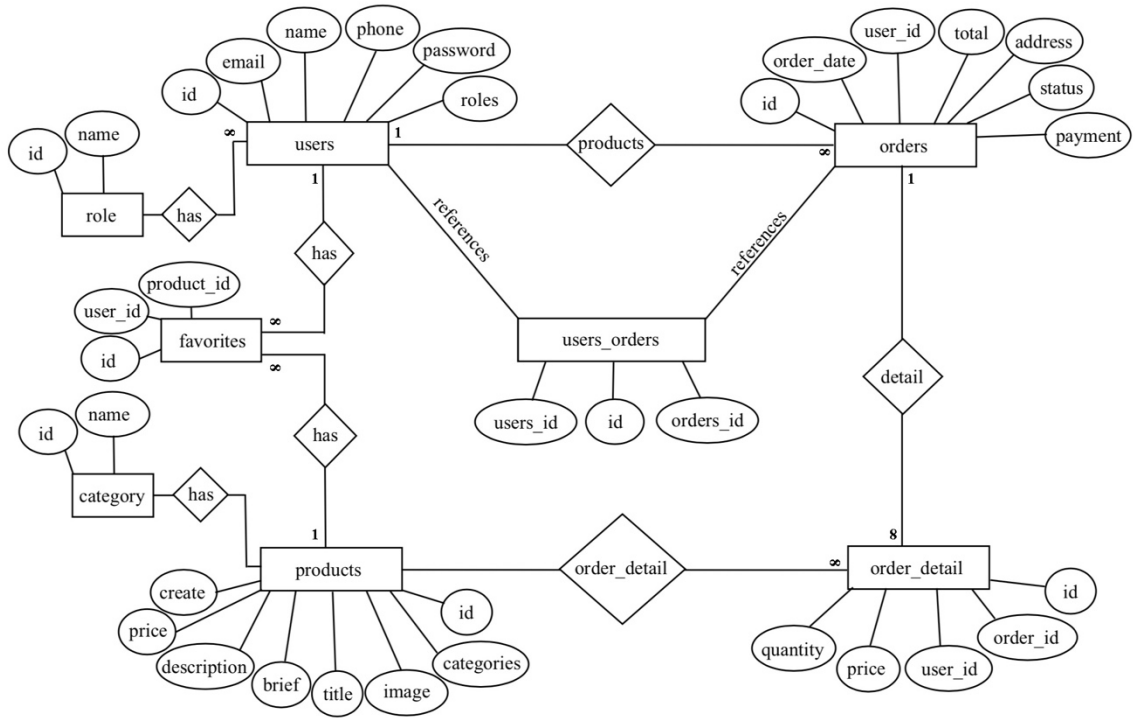


Şekil 15 Veritabanı Tasarımı



## 5.2 Veri Modeli

Veri modelinde kullanıcılar, ürünler, siparişler, sipariş detayı ve favorileri içeren varlıklar mevcuttur. Her kullanıcı kendine özel id'si ve şifresi vardır. Kullanıcıların sistemde yapacağı işlemleri belirleyen roller varlığı vardır. Her eklenen yeni ürününde özel id'si vardır. Oluşturulan siparişini kendine özel id'si ve kullanıcıya ait bilgisi vardır. Kullanıcının id'si ile oluşturacağı ve içinde ilgili ürününde id'sini içeren favoriler varlığı mevcuttur.



Şekil 16 Varlık-İlişki Diyagramı

## SONUÇ

Bu projede Flask ile SQLAlchemy kullanarak E-Ticaret internet uygulaması kodlanması amaçlanmış ve kodlama işlemi tamamlanmış, yerel bilgisayar ortamında çalıştırılarak hedefe ulaşılmıştır. Bu projede müşteri sergilenen ürünlerin her birini sipariş verebiliyor durumdadır. Yönetici ise ürün veya ürünlerini ekleyerek sergilemede bulunuyor. Bu sayede müşteri ile satıcı bir araya geliyor. Flask E-Ticaret projesinde uygulanması düşünülen hedefe ulaşılmıştır.

Bu projede kodlanamayan durum ise siparişin sonunda oluşan ödeme işleminde kredi kartı ile online ödeme entegrasyonudur.

Daha sonra proje üzerinde çalışma yapmak isteyenler şu anda yapılamayan sipariş ödeme işleminde kredi kartı entegrasyon işlemine çalışmalıdırlar. Bu sayede ödeme işleminde kredi kartı entegrasyonu ile online ödeme seçeneği de eklenebilir.

## KAYNAKÇA

Virtualenv [Online] : <https://virtualenv.pypa.io/en/latest/>

Flask [Online] : <https://flask.palletsprojects.com/en/1.1.x/>

Flask Tutorial [Online] : <https://flask.palletsprojects.com/en/master/tutorial/>

Flask User [Online] : <https://flask-user.readthedocs.io/en/latest/installation.html>

Flask Login [Online] : <https://flask-login.readthedocs.io/en/latest/>

Flask SQLAlchemy [Online] : <https://flask-sqlalchemy.palletsprojects.com/en/2.x/>

Flask Babalex [Online] : <https://pythonhosted.org/Flask-BabelEx/>

Flask secure\_filename [Online] : <https://flask.palletsprojects.com/en/1.1.x/patterns/fileuploads/>

Bootstrap Documentation [Online] : <https://getbootstrap.com/docs/4.4/getting-started/introduction/>

Jinja2 [Online] : <https://jinja.palletsprojects.com/en/2.10.x/>

Python Framework [Online] : <https://wiki.python.org/moin/WebFrameworks>