

İSTANBUL TEKNİK ÜNİVERSİTESİ
Bilgisayar ve Bilişim Fakültesi

**1C Enterprise Uygulama Tasarımı ve
Android-1C Entegrasyon Uygulaması**

STAJ
Talha ÇOMAK
150160726

Yaz / 2019

İstanbul Teknik Üniversitesi
Bilgisayar ve Bilişim Fakültesi
STAJ RAPORU

Akademik Yıl: 2018-2019

Staj yapılan dönem: ☒ Yaz ☐ Bahar ☐ Güz

Öğrenci ile ilgili bilgiler

Adı ve Soyadı: Talha Çomak

Öğrenci Numarası: 150160726

Bölüm: Bilgisayar Mühendisliği

Program: %30 İngilizce

E-posta Adresi: comak16@itu.edu.tr

(Cep) Tel No: 0 507 309 6271

ÇAP öğrencisi misiniz? ☐ Evet (ÇAP yaptığınız Fakülte/Bölüm: _____)

☒ Hayır

Mezuniyet
durumunda mısınız? ☐ Evet

☒ Hayır

Yaz okulunda ders
alıyor musunuz? ☒ Evet (Ders sayısı: 3)

☐ Hayır

Öğrencinin çalıştığı kurum ile ilgili bilgiler

İsmi: Kobi Yazılımları A.Ş.

Birimi: Yazılım

Web Adresi: <http://kobiyazilimlari.com>

Kısa Adresi: Küçükbakkalköy Mah. Kıyıdağı cad. No:1 Kat:5 Alianz Tower
Ataşehir/İstanbul

Yetkili kiři ile ilgili bilgiler

Bölümü: Yazılım
Unvanı: Departman sorumlusu
Adı ve Soyadı: Çağatay Akpolat
(Kurumsal) E-posta: info@kobi yazilimleri.com
(Kurumsal) Tel. No.: 0216 470 7310
0216 305 0444

Yapılan iş ile ilgili bilgiler

Staj yeri ☒ Türkiye
☐ Yurtdışı
Staj başlangıç tarihi 16.08.2019
Staj bitiş tarihi 13.09.2019
Stajda çalışılan net **gün** sayısı 20
Staj süresince sigortanız var mıydı? ☒ Evet, İTÜ tarafından sigortalandım.
☐ Evet, kurum tarafından sigortalandım.
☐ Hayır, yurtdışı stajı yaptım.
☐ Hayır.

İÇİNDEKİLER

1. KURULUŞ HAKKINDA BİLGİLER.....	1
2. GİRİŞ.....	1
3. STAJ PROJESİNİN TANIMI VE ANALİZİ.....	2
3.1. Proje Detayları	
3.1.1. 1C Enterprise Uygulama Tasarımı	
3.1.2. Android-1C İletişim Uygulaması Tasarımı	
4. SONUÇ.....	15
5. REFERANSLAR.....	15

1. KURULUŞ HAKKINDA BİLGİLER

Staj yaptığım kuruluş KOBİ Yazılımları A.Ş. kobilere özel yazılım hizmeti veren bir firma olup bünyesinde 1C Yazılım, UDS Türkiye, Desen ERP Yazılımı, SAP Business One, Online Veri Yedekleme, Phexum Yapay Zeka, BayiPro B2B Yazılımı, Sanal Santral, Deos – Gayrimenkul Yönetimi gruplarını barındırır.

Yüksek teknolojiye sahip ürünlerin ulusal ve uluslararası pazarlarda geliştirilmesi ve pazarlanmasında yetkin bir bilişim firması olup bilgi teknolojileri alanında yaşanan gelişmeleri yakından takip ederek iş ortakları ile profesyonel çözümleri işletmelere sunmayı hedefleyen bir firmadır [1]. Firmaların ihtiyaçlarına yönelik özel farklı platformlara uygun yazılımlar geliştirmek, desteklerini vermek, projeler yapmak, eğitimlerini gerçekleştirmek, firmalara özel yazılımsal ve donanımsal entegrasyonlar yapmak firmanın verdiği hizmetler arasındadır.

Staj projemin konusu 1C Yazılım uygulaması tasarımı ve Android-1C mobil entegrasyon uygulaması tasarımıdır. Şirketin müşterilerine özel olarak tasarladığı çözüm projeleriyle ilişkilidir.

İnternet üzerinden <http://kobi yazilimlari.com/> adresinden firma hakkındaki bilgilere erişebilir ve gerekli iletişimi sağlayabilirsiniz.

e-posta: info@kobi yazilimlari.com

adres: Küçükbakkalköy Mah. Kınışdağı cad. No:1 Kat:5 Alianz Tower
Ataşehir/İstanbul
tel: 0216 470 7310
0216 305 0444

2- GİRİŞ

Staj, Kobi Yazılım Hizmetleri A.Ş.'de 1C yazılım uygulaması ve bağlantılı android uygulaması geliştirme üzerine yapıldı. Proje aşamasında şirket çalışanı İsmail Gürkan eğitim ve uygulama geliştirme aşamalarında danışmalık yaptı.

1C yazılım uygulaması 1C:Enterprise platformunda geliştirilen Rus merkezli açık kaynak kodlu, isteğe özel tasarlanılan bir muhasebe uygulamasıdır. Ayrıca bir şirketin finansal ve daha geniş operasyonel faaliyetlerini otomatikleştirmek için kullanılan evrensel bir bulut ve şirket içi program sistemidir. 1C geliştirme ortamı, günümüz işletmelerinin farklı ihtiyaçlarını karşılama kapasitesine sahiptir. Geliştirme ortamının sağladığı şeylerden biri de şirketlerin özel ihtiyaçlarına ve iş süreçlerine göre özelleştirilebilmesidir.

1C:Enterprise, sabit işletme kurallarını otomatikleştiren bir çözümden daha fazlasıdır. Aksine, geliştiriciler ve kullanıcılar tarafından kullanılan bir yazılım araçları paketidir. Sistem mantıksal olarak birbiriyle ilişkili iki ana bileşene ayrılabilir: tasarımcı ortamı ve uygulamanın üzerinde çalıştığı platform.

1C: Enterprise platformu aşağıdaki avantajları sunar:

- Kurumsal yazılım sistemlerinin teknolojik karmaşıklığını ve performans sorunlarını büyük ölçüde azaltır.
- Uzman kullanıcıların belirli iş süreçlerini uygulamalarına izin verir.
- Uygulama geliştirme, özelleştirme ve desteğini hızlandırır ve standartlaştırır.
- İş süreçlerinin daha iyi anlaşılmasını ve değiştirilmesini sağlayan arayüz ve çözüm kodunun tamamen açık olmasını sağlar.
- Mevcut 1C uygulamaları ve üçüncü taraf sistemlerle entegrasyonu sağlanabilir.
- Web servislerini, ODBC, COM vb. destekler.

- Tercih ettiğiniz mimariyi destekler: Windows / Linux, MS SQL, PostgreSQL, IBM DB2 ve Oracle DB [2].

Projenin ilk kısmı basit bir muhasebe uygulamasını 1C:Enterprise’ın tasarımcı ortamında geliştirmektir. İkinci kısım ise mevcut, şirkete özel geliştirilmiş bir uygulamanın kullanımını kolaylaştırmak amacıyla android ortamında geliştirilen bir bağlantı uygulamasıydı. Android projesini geliştirmek için Android Studio uygulama geliştirme ortamı tercih edildi. Android Studio, Google tarafından önerilen resmi programlama aracı olup çok fonksiyonel özelliklere sahiptir. Bu özellikler temel olarak şunlardır:

- *Gradle* tabanlı, elastik yapıli proje oluřturma sistemi.
- Hızlı ve geniş özelliklere sahip bir emülatör.
- Farklı ekran boyutları ve işletim sistemi sürümlerine adapte olabilen yapı.
- GitHub entegrasyonu ve adapte edilebilen kod şablonları.
- C ++ ve NDK desteęi.
- Sürükle-bırak özellięine sahip fonksiyonel editörü sayesinde kolay bir tasarlanabilen bir yapı.
- Uygulamanın sürdürülebilirlięi, performansı ve farklı sürümlere adaptasyonunun kontrol edilebileceęi test araçları ve frameworkler.
- Kolay ve güvenli Android Application Package imzalanması.
- Google hizmetlerini uygulamaya kolayca ekleyebilme [3].

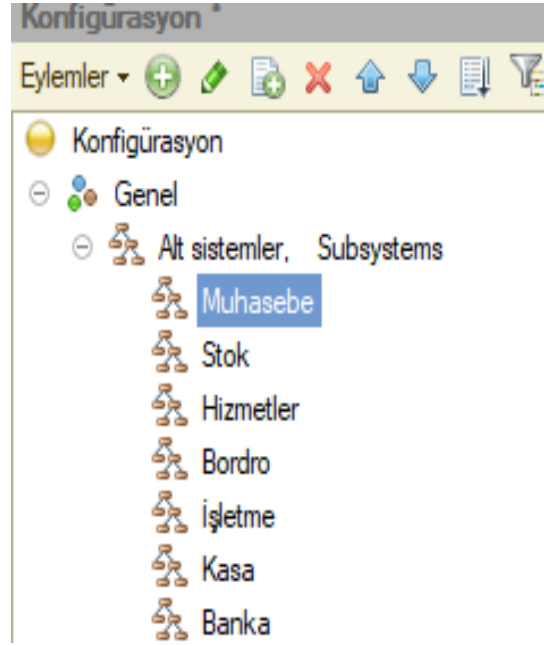
3- STAJ PROJESİNİN TANIMI VE ANALİZİ

Proje ilk olarak basit bir muhasebe uygulamasını 1C:Enterprise’ın tasarımcı ortamında geliştirmek sonrasında şirkete özel geliştirilmiş bir muhasebe uygulamasının kullanımını kolaylaştırmak amacıyla android ortamında geliştirilen bir bağlantı uygulamasıdır.

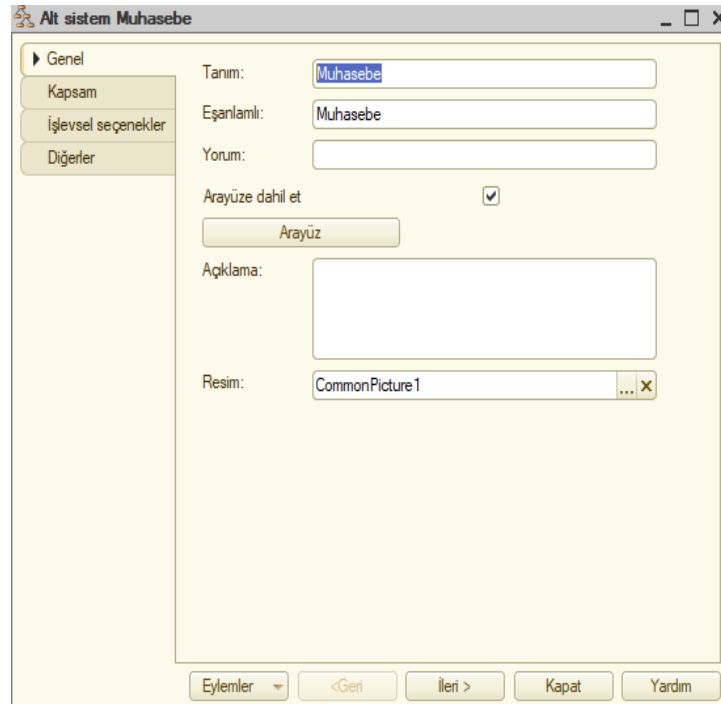
3.1- Proje Detayları

3.1.1. 1C Enterprise Uygulama Tasarımı

Bu kısımda 1C:Enterprise ile tasarlanan basit bir muhasebe uygulaması geliřtirdim. Uygulama geliřtirme aşaması tasarımcı ortamında yapılır. Bu ortamda “Konfigürasyon” adlı kullanılacak genel nesneleri kolayca oluřturup tasarım yapmak amaçlı bir nesnemiz var. Öncelikle menü kısmının tasarımı için “Konfigürasyon” nesnesinden ”Genel” sekmesini oradan da “Alt sistemler” sekmesini açıyoruz. Alt sistemler kısmında ekle butonunu kullanarak řekil-1’de görüldüęü üzere “Muhasebe”, “Stok Hizmetler”, “Bordro”, “İřletme”, “Kasa” ve “Banka” menülerini oluřturdum. Menü oluřturma kısmında açılan pencereden (řekil-2) ilgili menü için “Resim:” kısmındaki “...” butonuna tıklayarak ilgili resimleri ekledim.

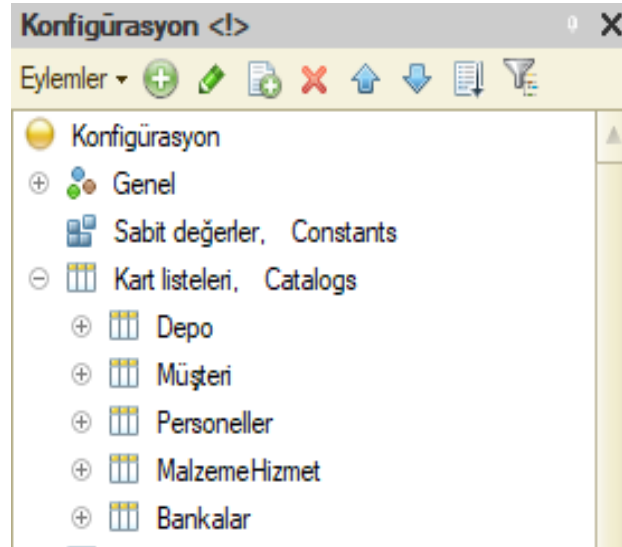


Şekil-1: “Alt sistemler”



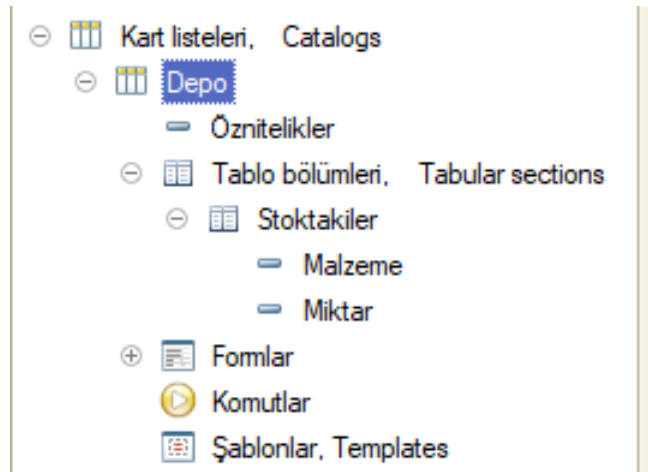
Şekil-2: Menü açılan pencere

Bu aşamadan sonra “Konfigürasyon” kısmından projede kullanılacak olan kart listelerini oluşturdum. “Kart listesi” içine tablo bölümleri ve öznelikler eklenilebilen temel nesnelerdir. Daha sonra kullanılmak üzere 5 adet kart listesini Şekil-3’de görüldüğü üzere oluşturdum.



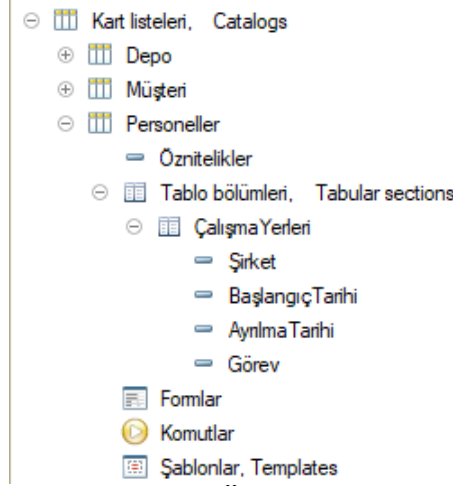
Şekil-3: “Kart listeleri”

Daha sonra bu tablolara tablo bölümleri ekledim. “Depo” kart listesinde “Stoktakiler” isimli tablo bölümü oluşturup içine “Malzeme” ve “Miktar” isimli öznelikler ekledim. “Depo” kart listesinin konfigürasyondaki görünümü Şekil-4’deki gibidir.



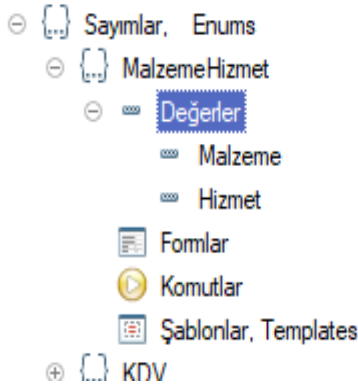
Şekil-4: “Depo” kart listesi

“Müşteri” kart listesi sadece müşteri bilgilerini tutmaya yarar. “Personeller” kart listesi kayıtlı personellerin bilgilerini saklar. Personellerle ilgili detaylı bilgiler için yine tablo bölümü ekleyip “Şirket”, “Başlangıç Tarihi”, “Ayrılma Tarihi” ve “Görev” isimli öznelikleri Şekil-5’de görüldüğü üzere ekledim.

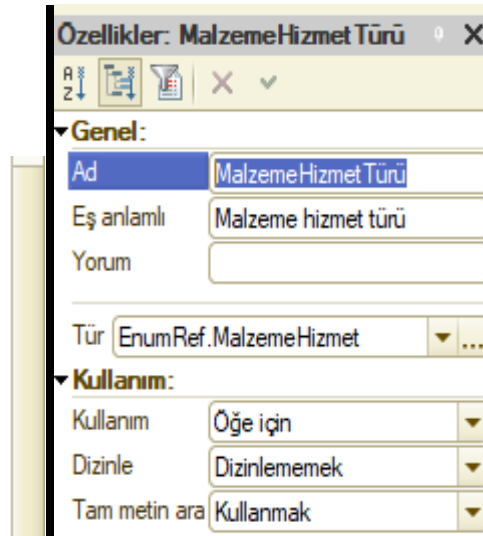


Şekil-5: Öznelikler

“MalzemeHizmet” kart listesi için gerekli özenitelik olan “MalzemeHizmetTürü” isimli özniteliği ise doğrudan bu kart listesine ekledim. Bu kart listesi eklenen nesnenin malzeme mi yoksa hizmet mi olduğu belirtmek içindir. Bu ayrımı gerçekleştirmek için “Sayımlar” (Enums) nesnesi kullanılır. Şekil-6’da görüldüğü gibi “Sayımlar” sekmesinde ekle butonuna basarak “MalzemeHizmet” isimli sayım değerini ekledim. Daha sonra nesnemize sağ tıklayarak ekle kısmından “Değerler” isimli sayım değerini ekledim. Öznelik olarak da yine ekle butonunu kullanarak “Malzeme” ve “Hizmet” isimli öznelikler ekledim. Bu özelliği aktifleştirmek için kart listelerindeki “MalzemeHizmetTürü” isimli özniteliğe tıklayarak özellikler penceresindeki tür değerini “Enum.Ref.MalzemeHizmet” olarak seçtim. Özellikler penceresi Şekil-7’de görülmektedir.



Şekil-6: “Sayımlar”



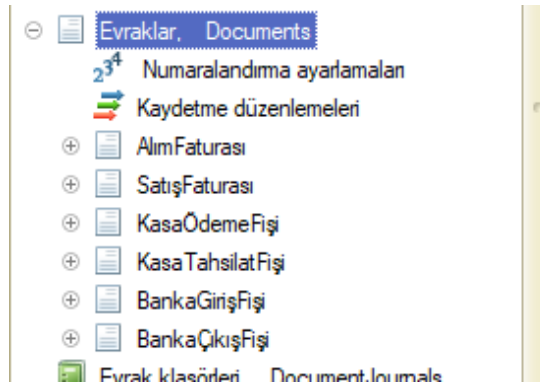
Şekil-7: Özellikler

Bu aşamadan sonra her kart listesini içinde bulunması gereken menü ile ilişkilendirmek için ilgili kart listesine çift tıklayarak açılan pencereden “Altsistemleri” sekmesini açmamız gerekiyor. Şekil-8’de görüldüğü üzere bu sekmede önceden oluşturduğumuz altsistemler yer almaktadır. Yer almasını istediğimiz alt sistemin solundaki kutucuğa tik atarak her kart listesinin ilgili menülerde yer almasını sağlıyoruz.

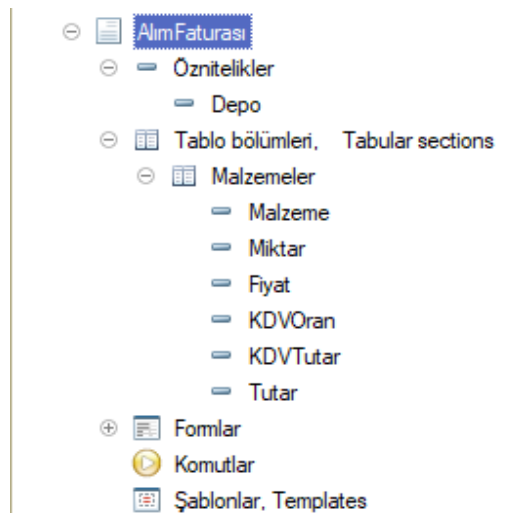


Şekil-8: “Altsistemleri” sekmesi

Bir sonraki aşamada “Evraklar” nesnesine geçiyoruz. Bu nesne muhasebe işlemleri sonucu oluşan evrakların saklanması ve bu evraklardan raporlar oluşturulması içindir. İlk olarak Konfigürasyonda bulunan “Evraklar” nesnesine tıklayıp ekle butonundan “AlımFaturası”, “SatışFaturası”, “KasaÖdemeFişi”, “KasaTahsilatFişi”, “BankaGirişFişi” ve “BankaÇıkışFişi” isimli evrak nesnelerimizi oluşturuyoruz. Oluşturulan nesneler Şekil-9’da görülmektedir.



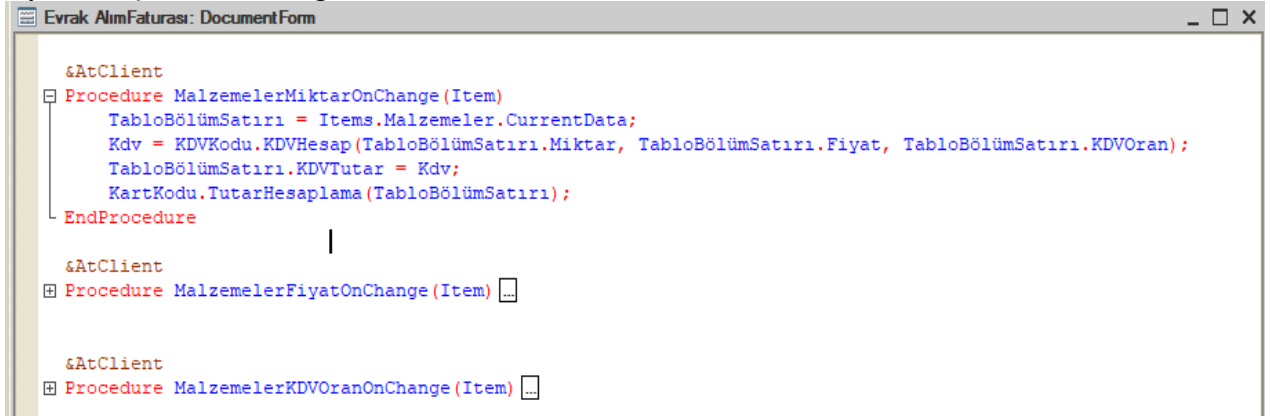
Şekil-9: “Evraklar”



Şekil-10: “Alım Faturası”

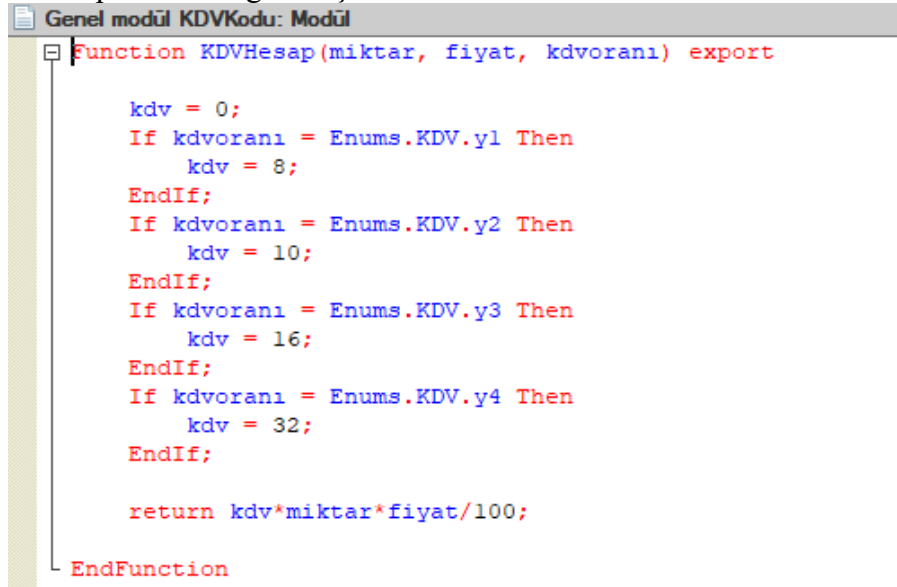
Oluşturduğum bu evrakların gerekli tablo bölümü ve özniteliklerini “Kart Listeleri” kısmındakiyle aynı yöntemle oluşturduktan sonra yine aynı şekilde gerekli alt sistemlere eklemeler yaptım. Bu evraklardan alım faturasının özellikleri Şekil-10’daki gibidir. “Alım

Faturası”nda görüntülenecek olan “Fiyat”, “KDV Tutar”, “Tutar” gibi özelliklerin otomatik hesaplanması için ise formlar sekmesinden bir adet “DocumentForm” oluşturmamız gerekiyor. Oluşturduğum bu forma çift tıklayıp “Modül” sekmesini açtığımda ilgili özellikleri hesaplamak için gerekli kodların yazılabileceği bir sayfa karşımıza çıkar. Bu sayfada *&AtClient* metoduyla yazılan kodun ilgili nesneyle ilişkilendirilmesi sağlanır. Şekil-11’deki *MalzemelerMiktarOnChange* prosedürü uygulama çalıştığı esnada Item nesnesindeki her Miktar değişikliğinde *Kdv*’nin ve *Tutar*’ın yeniden hesaplanmasını sağlar. Buradaki *KDVKodu.TutarHesaplama* fonksiyonu ise Konfigürasyonumuzdaki “Genel” sekmesindeki “Genel modüller” kısmında oluşturduğum modüldeki “Kdv”yi hesaplayan fonksiyondur. Bu fonksiyon da Şekil-12’deki gibidir.



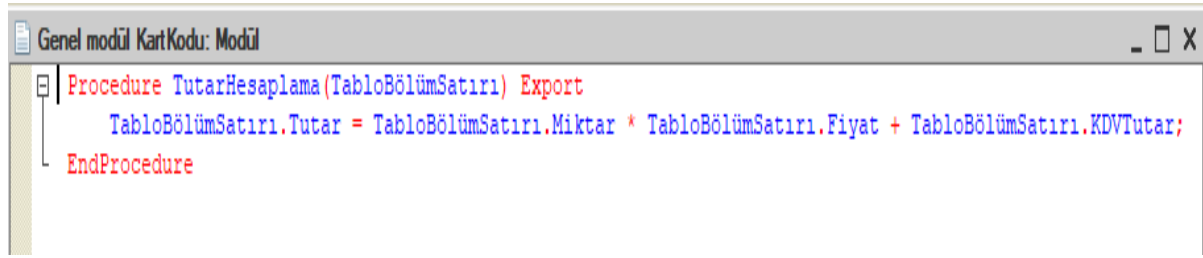
Şekil-11: MalzemelerOnChange

Şekil-12’deki *Enums*, sayımlar’ın İngilizce karşılığıdır ve kod kısımlarında Konfigürasyondaki nesnelerin İngilizceleri kullanarak ilgili özelliklere doğrudan erişim sağlayabiliriz. Bu fonksiyon sayesinde 5 farklı Kdv değerinden seçilmiş olan değere göre toplam Kdv’nin hesaplanması sağlanmıştır.



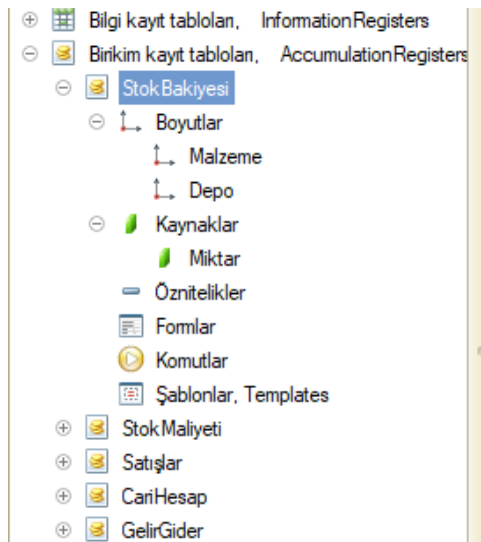
Şekil-12: KDVHesap

Evrağımızla ilgili olarak yukarıda sadece miktar değişikliği için gerekli fonksiyon verilmiştir. Aynı zamanda fiyat ve Kdv oranındaki değişiklikler için de benzer fonksiyonlar ve prosedürler yazarak uygulamanın kullanıcı açısından kolaylaşmasını sağlamış olduk. Şekil-11’deki prosedüre geri dönecek olursak, *KartKodu.TutarHesaplama* fonksiyonu yine “Genel modüller” kısmındaki “KartKodu” isimli modülde bulunan *TutarHesaplama* fonksiyonudur. Bu fonksiyon Şekil-13’deki gibidir.

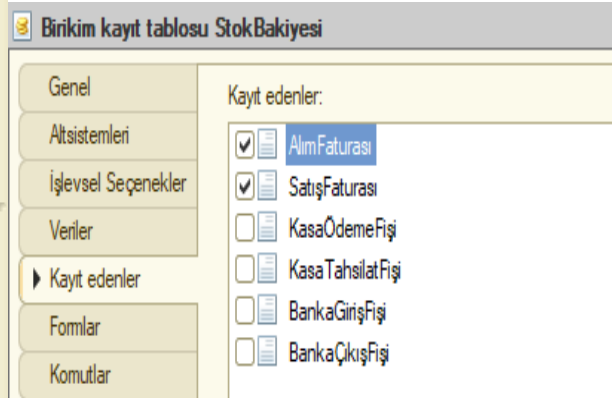


Şekil-13: “Genel Modül”, *TutarHesaplama*

Bu aşamadan sonra diğer evraklarımızla ilgili de aynı benzer işlemleri takip ederek evrak oluşturma aşamamızı da tamamlamış olduk. Sırada “Birikim Kayıt Tabloları” (AccumulationRegisters) var. Bu tablolar sayesinde stok hareketlerini birçok alanda kayıt ederek çeşitli raporlar oluşturabiliriz. Tablolarımız hem yapılan işlemleri görüntülemek hem de rapor oluşturmak açısından önemlidir. Konfigürasyonumuzda bulunan “Birikim Kayıt Tabloları”na tıklayıp ekle butonuna basılarak oluşturulan “StokBakiyesi”, “Stok Maliyeti”, “Satışlar”, “CariHesap”, “GelirGider” isimli tablolarımız Şekil-14’teki gibidir. İlk olarak “StokBakiyesi” isimli tablomuzda boyutlar sekmesine tıklayıp ekle butonuna basarak “Malzeme” ve “Depo” isimli tablo boyutlarımızı oluşturuyoruz. Kaynaklar kısmına da “Miktar” özniteliğini ekleyerek hangi depoya hangi malzemeden ne miktarda giriş çıkış yaptığının kaydını tutacak olan bir adet “Birikim kayıt tablosu” oluşmuş oldu. Gerekli bağlantıları sağlamak amacıyla bu tablomuzda çift tıklayarak “Altsistemleri” sekmesinden daha önceki kısımlarda yaptığımız gibi ilgili altsistemimize kaydediyoruz. Sonra da “Kayıt edenler” sekmesine tıklayıp Şekil-15’te görüldüğü gibi “AlımFaturası” ve “SatışFaturası” yazan kısımlara tik atıp bu faturalarla ilişkilendiriyoruz ve sonucunda istediğimiz tablo uygulama ortamında kaydedilmeye hazır hale gelmiş oluyor.

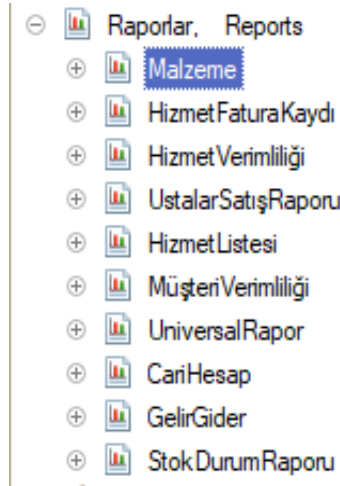


Şekil-14: “Birikim kayıt tabloları”



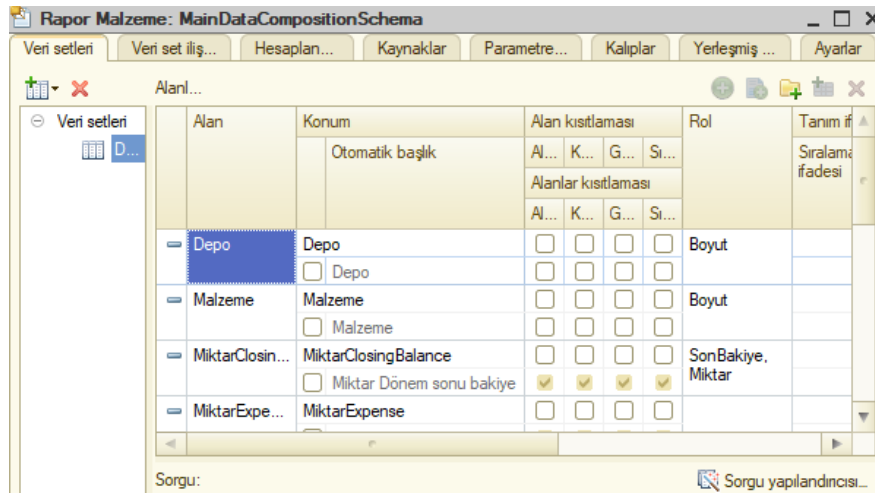
Şekil-15: “Kayıt edenler”

Diğer Birikim Kayıt Tablolarını da benzer işlemlerle oluşturup uygulama ortamı için hazır hale getirdim. Sırada raporlarımız var. Raporları oluşturmak için yapmış olduğumuz “Birikim kayıt tabloları”nı kullanacağız. Öncelikle “Raporlar” (Reports) sekmesine tıklayıp ekle butonu ile “Malzeme”, “HizmetFaturaKaydı” gibi Şekil-16’daki raporları oluşturdum.

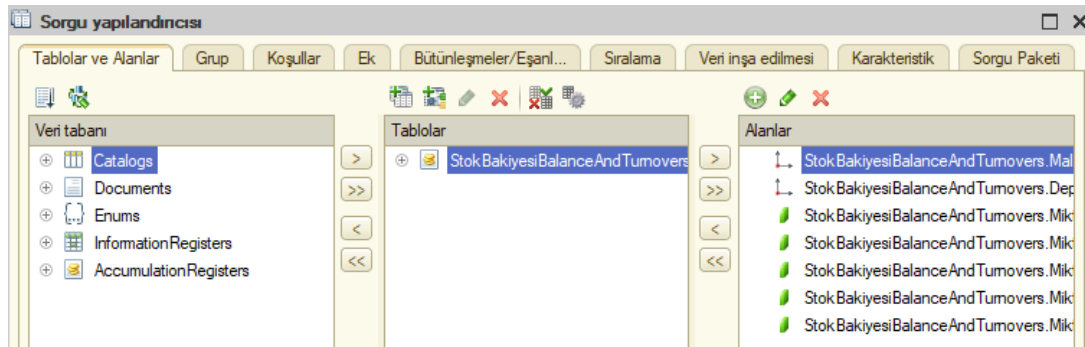


Şekil-16: “Raporlar”

Rapor içeriğini oluşturmak için seçilen rapora çift tıklayıp, “Genel” sekmesindeki “Veri İnşa Edilme Şeması”nı Aç” butonuna tıklayarak Şekil-17’de görülen “Sorgu Yapılandırıcısı” butonuna tıklanır. Açılan pencerede (Şekil-18) raporda kullanılabilecek veriler “Veri tabanı” sekmesinin altında bulunur. Ok işaretleri ile kullanılmak istenilenler “Tablolar ve Alanlar” sekmelerinin altına eklenir. “Tamam” butonuna basıldığında raporun temel verileri ortaya çıkar. Daha sonra Şekil-17’deki pencereden ayarlar sekmesine tıklanarak rapor bu kısımdan müşterinin ihtiyacına göre detaylı ve efektif hale getirilebilir.



Şekil-17: “Veri İnşa Edilme Şeması”



Şekil-18: “Sorgu Yapılandırıcısı”

Uygulamanın tasarımcı ortamı bu şekildedir. “Konfigürasyon” üzerinden ve formlar yardımıyla kod ekleyerek müşterinin isteğine uygun şekilde farklı özellikler türetmek

mümkündür. Sonuç olarak uygulamam temel muhasebe özelliklerini yansıtacak bir düzeyde oluşmuş oldu.

Uygulamanın kullanıcı ortamına gelecek olursak; geliştirme ortamında tasarlanmış olduğum “Muhasebe”, “Bordro”, “Hizmetler”, “Stok”, “İşletme” ve “Kasa” bölümlerini içeren menü uygulama ortamında Şekil-19’deki gibidir.



Şekil-19: Menü

“Muhasebe” sayfasında yer alacak şekilde; “Alım Faturaları”, “Depolar”, “Malzeme ve Hizmetler”, “Malzeme/Hizmet Fiyat”, “Müşteriler”, “Personeller”, “Satış Faturaları”, “Tahsilat Fişleri” ve “Ödeme Fişleri” için basit sekmeler tasarladım. Yine bu sayfada “Ayrıca bakınız” başlığı altında “Stok Bakiyesi”, “Stok Maliyet Tablosu”, “Satışlar Kayıt Tablosu”, “Gelir gider” ve “Cari hesap” sekmeleri oluşturdum.

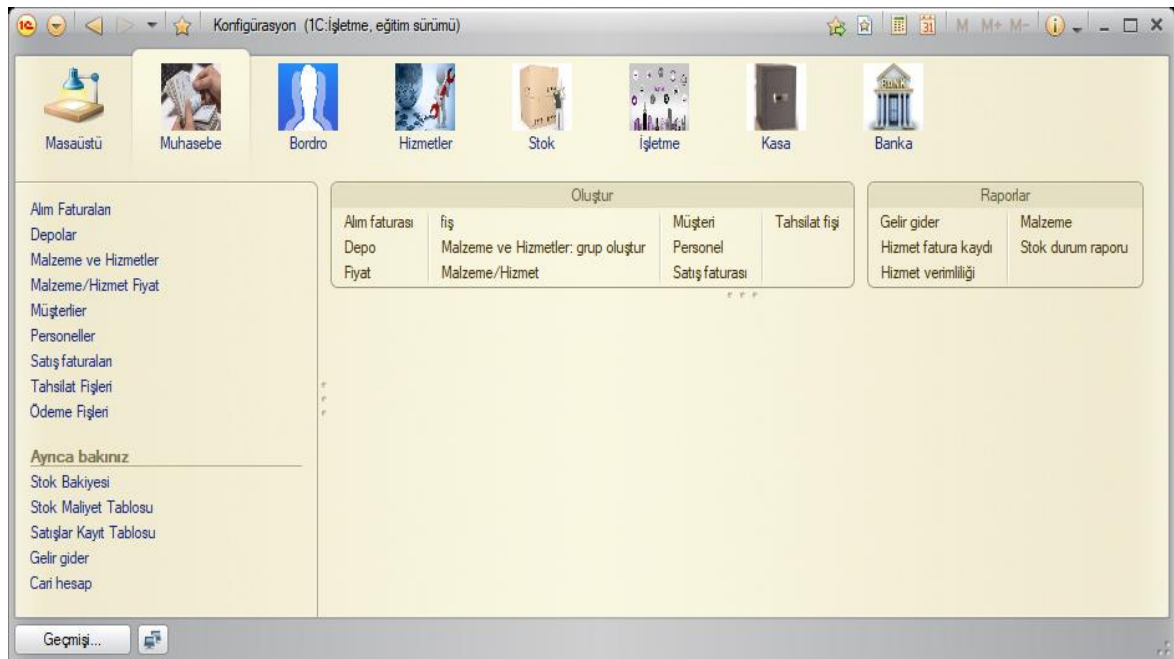


Şekil-20: “Muhasebe”

“Hizmetler” sayfasında da benzer özellikler yer almakta. Ayrıca muhasebe işlemleri sonucunda görüntülenmek üzere bütün raporların yer aldığı bir tablo bulunmaktadır.



Şekil-21: “Hizmetler”



Şekil-22: Uygulamanın Görünümü

Uygulamanın genel şablonu bu şekildedir.

3.1.2. Android-1C İletişim Uygulaması Tasarımı

Daha önce tasarlanmış olduğumuz 1C uygulamasının kullanımını kolaylaştırmak amacıyla bir mobil uygulama ile entegrasyonunu sağlayabiliriz. Bu amaçla projenin devamında Android Studio üzerinden bir uygulama oluşturdum. Uygulamanın fonksiyonu; 1C üzerinden oluşturulan bir QR kodu tarayıp sonucu üzerinden 1C uygulamasına giriş ve çıkış fonksiyonlarını yerine getirmek. Bu fonksiyon 1C kullanan mağazalarda kasiyerlerin kolay bir ve hızlı şekilde giriş, çıkış işlemlerini yapmasını ve mağaza tarafından rahatlıkla denetlenmesini sağlamaktadır. Android Studio, android işletim sistemleri için mobil uygulama geliştirme platformu olup, işletim sistemine özel olduğu için kullanımı kolay ve fonksiyoneldir. Birkaç farklı yazılım dili desteği mevcuttur. Kendi uygulamam için java dilini tercih ettim. Geliştirme ortamında uygulama sayfaları activity olarak isimlendirilip her activity için bir adet xml uzantılı dosya oluşturulur. Bu dosyalar da kendilerine ait birer sınıf (*class*) içeren Java uzantılı java dosyalarıyla birlikte çalışırlar.

Uygulamamda ilk olarak “activity_main” olarak isimlendirdiğim uygulamanın çalışacağı sayfayı tasarladım. Sayfa tasarımında iki adet buton (*Button*), iki adet yazı (*TextView*) ve bunların sayfada yer almasını sağlayan yapılar (*LinearLayout*) bulunmaktadır. Her sayfa elemanının kendine has özellikleri bulunmaktadır. Projemde kullandığım özellikler şunlardır:

- *android:orientation* = layout’un elemanlarının hangi ekseninde yerleşeceğini belirler.
- *android:background* = arkaplan rengini belirler.
- *android:layout_width* = layout’un genişliğini belirler.
- *android:layout_height* = layout’un yüksekliğini belirler.
- *android:layout_marginTop* = layout’un yukarisından ne kadar boşluk bırakacağını belirler.
- *android:id* = elemana kimlik verir.
- *android:gravity* = elemanın layout’taki konumunu belirler.
- *android:textSize* = yazı boyutu
- *style* = özel oluşturulmuş tasarım özelliği
- *android:layout_gravity* = layoutun sayfadaki konumunu belirler.
- *android:text* = elemana yazı ekler.

Daha sonra “activity_main” ile ilişkili Java sınıfımı oluşturdum. Oluşturduğum bu sınıfa projemde kullanmak üzere birkaç Java kütüphane sınıfları ekledim. Bu kütüphane sınıfları ve kullanılan özellikleri aşağıda yer almaktadır.

- *androidx.appcompat.app.AlertDialog;*
- *androidx.appcompat.app.AppCompatActivity;*
- *android.annotation.SuppressLint;*
- *android.content.Context;*
- *android.content.DialogInterface;*
- *android.content.Intent;*
- *android.os.AsyncTask;*
- *android.os.Bundle;*
- *android.util.Log;*
- *android.view.View;*
- *android.widget.Button;*
- *android.widget.TextView;*
- *android.widget.Toast;*
- *org.json.JSONObject;*
- *java.io.BufferedReader;*
- *java.io.BufferedWriter;*
- *java.io.InputStream;*
- *java.io.InputStreamReader;*
- *java.io.OutputStream;*
- *java.io.OutputStreamWriter;*
- *java.net.Authenticator;*
- *java.net.HttpURLConnection;*
- *java.net.PasswordAuthentication;*
- *java.net.URL;*
- *java.util.UUID.randomUUID;*

Bu kütüphane sınıflarından *android* ve *androidx* ile başlayanlar, androidin kendi sınıfları olup uygulamanın temel özelliklerini gerçekleştirmek içindir. *java.io* kütüphane sınıfları Java dilinde girdi ve çıktı işlemleri için, *java.net* kütüphane sınıfları sunucu

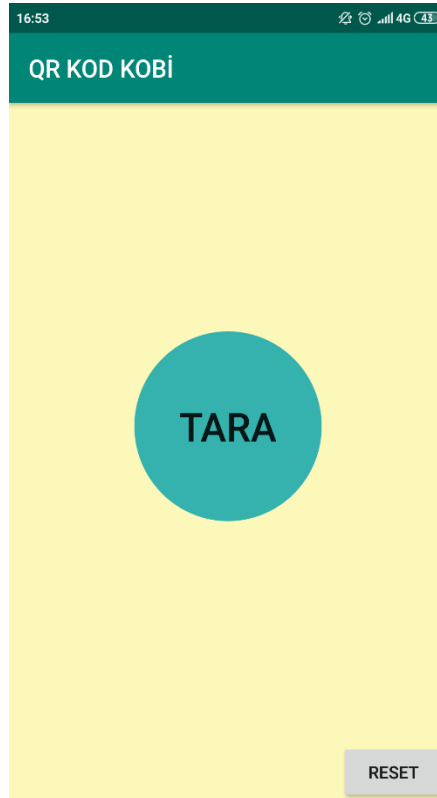
bağlantısı, *get_request*, *post_request* işlemleri için ve *java.util* sınıfı ise rastgele bir GUID kodu üretmek içindir. Ayrıca JSON objesi oluşturmak için *org.json.JSONObject* sınıfını kullandım.

“activity_main”de kullanılan metotlar:

- *AsyncTask<..., ..., ...>* metodu: istenilen işlemin arka planda gerçekleşmesini sağlar.
- *doInBackground()* metodu: *AsyncTask*’da hesaplamaları arka planda yapmak için bu metod çağrılır. Hesaplamaların sonucu bu adımda döndürülür.
- *onPostExecute()* metodu: *AsyncTask*’da arka plan işlemlerinin sonucu bu adıma geçerek sonlanır.
- *onCreate()* metodu: Bu metot Java aktivitelerinin uygulama tarafından başlatılmasını sağlayan metoddur. Sınıf yok olana ya da durana kadar çalışmaya devam eder.
- *item.setOnClickListener()* metodu: “item” nesnesine tıklandığında yapılacak işlemleri belirlemeye yarar.

“activity_main”de yapılan işlemler ise şunlardır:

- “Giriş” sayfasından bilgi alma, bilgi alındıysa tek seferlik bir GUID kodu üretip dosyaya yazdırma. (Bu kod her kullanıcı için giriş kimliği görevi görecektir.) Daha sonra ekrana bir “hoş geldiniz” mesajı yazdırma.
- Bir önceki aşamada GUID kod üretildiyse bunu dosyadan okuma.
- Giriş yapılmamışsa “Giriş” sayfasına yönlendirme.
- Tarama butonuna basıldığında telefonun kamerasına erişim QR kod okuma.
- Okunan kodu belirli bir ağa *post_request* metoduyla gönderip ağdan yanıt alma işlemi. Bu işlem için öncelikle okunan kod bir JSON objesine dönüştürülür. Bu işlemden sonra oluşturulan obje *post_request* ile ağa gönderilir ve *get_request* metoduyla ağdan yanıt alınır.



Şekil-23: “activity_main” sayfası

Uygulamamın ikinci sayfası ise giriş sayfasından oluşmaktadır. Bu sayfa uygulamayı yükleyen kullanıcıların bir kereliğine üretilen GUID kod ile sisteme kayıt olup

tekrar giriş işlemi yapmalarını önlemek amaçlıdır. Bu xml sayfasında iki adet yazı (*TextView*) bir adet buton (*Button*) bir adet yazı girişi (*EditText*), bir adet kayan seçim (*Spinner*) ve bunları barındıran yapılar (*LinearLayout*) elemanları bulunmaktadır. Bu elemanlar “activity_main” sayfamız için verilen özelliklere ek olarak aşağıdaki özellikleri barındırmaktadır.

- *android:layout_marginLeft="20dp"* : *layout*’un soldan 20dp’lik boşluk bırakmasını sağlar.
- *android:entries="@array/language"* : *spinner*’daki listeye *language* adlı dizinin elemanlarını atar.
- *android:spinnerMode="dropdown"* : *spinner*’ın kayan modda çalışmasını sağlar.

Bu sayfada kullanılan metotlar ise “activity_main”deki gibi *onCreate()*, *setOnClickListener()* metotları ve bir de *onItemSelected()* metodudur. Bu metot liste özelliği olan nesnelerden bir eleman seçildiğinde seçilen elemanla ilgili yapılacak işlemlerin belirlenmesini sağlar.

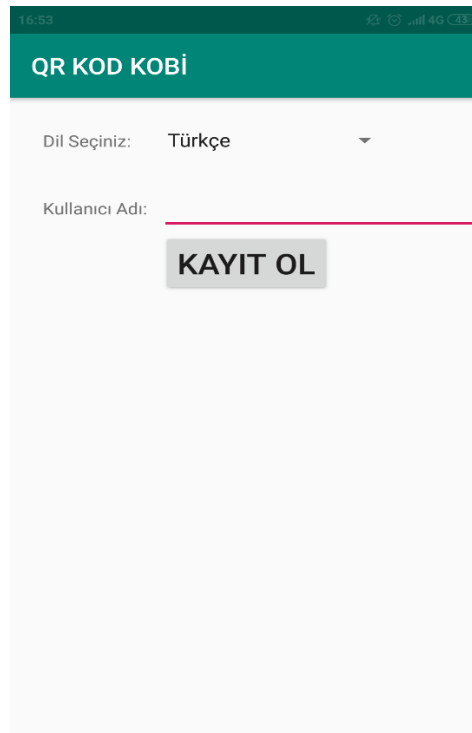
Daha sonra bu xml sayfası ile bağlantılı Java sınıfını içeren Java uzantılı dosyayı oluşturdum. Bu sayfada bir önceki kısımdaki kütüphane sınıflarına ek olarak aşağıdaki sınıflar da kullanılmıştır.

android.widget.Spinner;

androidx.annotation.Nullable;

Bu sınıflardan ilki *spinner* adlı android elemanını uygulamaya eklemeyi, ikincisi ise bir fonksiyonu *nullable*, yani boş bırakılabilir yapabilmeyi sağlamaktadır. “Giriş” sayfasında yapılan işlemler şunlardır:

- “activity_main” tarafından oluşturulduğunda faaliyete geçme.
- Kullanıcıya *spinner* üzerinden üç farklı dil seçeneği sunma.
- Kullanıcıdan kullanıcı adı bilgisini alma. Bu bilgi boş ise kayıt yapılmasını önleyerek kullanıcı adı bilgisi isteme.



Şekil-24: “Giriş” sayfası

Uygulamada kullandığım ek özellikler ise aşağıda yer almaktadır.

- Renk oluşturma özelliği. Örnek: `<color name="colorAccent">#D81B60</color>`

- Metinler için string değişkenleri oluşturup seçilen dile göre farklı değer verme özelliği.
Örnek : `<string name="scan">Tara</string>`
- `android.content.SharedPreferences` sınıfındaki `SharedPreferences` fonksiyonu ile kullanıcının cihazında diğer uygulamaların erişiminden bağımsız bir dosya oluşturup bu dosyaya istenilen bilgileri kaydetme özelliği.
- QR kod okuma işlemi için kullandığım kütüphane sınıfı olan `me.dm7.barcodescanner.zxing.ZXingScannerView`, bu kütüphanenin fonksiyonu olan `ZXingScannerView.ResultHandler()` fonksiyonu ve `ZXingScannerView scannerView` itemi.

Kullanılan ek metodlar:

- `handleResult()`: Sonuçla ilgili işlemler tanımlamayı sağlar.
- `onPause()`: Aktivite durduğu zaman yapılacak işlemleri tanımlamayı sağlar.
- `onResume()`: Aktivite devam ettiği zaman yapılacak işlemleri tanımlamayı sağlar.

4. SONUÇ

Staj süresince 1C:Enterprise üzerinden bir işletmeye özel bir muhasebe uygulamasının nasıl tasarlanabileceğini ve bu süreçte nelerle karşılaşabileceğimi öğrendim. Projemde belirttiğim özelliklerin dışında bu platformların çeşitli özellikleriyle ilgili staj eğitmenlerim tarafından eğitim aldım. Ayrıca Android Studio platformu üzerinden mobil uygulama geliştirme konusunda da kapsamlı bir eğitim gördüm. Bu platformda geliştirdiğim uygulama ile Java dilinde çeşitli metodlar ve fonksiyonlar öğrendim. Benim için en önemlisi sunucuya bilgi gönderme ve sunucudan yanıt alma kısmıydı.

Stajdaki çalışma ortamı bir ofis içindeki masalarda karşılıklı iletişimin kolay olduğu bir alandı. Çalışma saatleri sabit olmasına karşın serbest bir düzen vardı ve herkes kendini en iyi hissedebileceği şekilde çalışıyordu. Güzel keyifli bir ortamda herkes işini zaman zaman birbirleriyle yardımlaşarak, toplantılar yaparak, fikir alışverişleriyle yürütüyorlardı.

Sonuç olarak bir 1C uygulaması ve 1C uygulamalarının kullanımını kolaylaştırmayı amaçlayan mobil android uygulaması geliştirmiş oldum. İlk kez bir iş deneyimim oldu. İş yerindeki çalışanlar bildikleri konularda yardımcı oldular ve çok güzel bir iş ortamı vardı. Benim için güzel bir staj deneyimi oldu.

5. REFERANSLAR

- [1] <http://kobi-yazilimlari.com/kobi-yazilim-hizmetleri-kurumsal/>
 [2] https://1c-dn.com/1c_enterprise/what_is_1c_enterprise/
 [3] <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-201/android-studioyu-taniyalim>