

CSE222 GRUP 15 FİNAL PROJE TASARIM RAPORU

A. Problemin Tanımlanması ve İfade Edilmesi

Projenin kapsamı belirli bir grubun erişebildiği, ürün satın alabildiği Android ortamda geliştirilmiş bir elektronik alışveriş sistemidir. Burada belirli bir grup olarak ifade edilmek istenen bir üniversite çatısı altında bulunan insanlardır. Bu projeyi **n11, Aliexpress, Letgo** gibi tüm insanlığa hitap eden bir uygulamadan ziyade kullanıcı profilini kısıtlamadaki nedenler şu şekilde sıralanabilir:

1. Elektronik alışveriş sistemleri genele hitap eden ve daha önceden birçok örneği yapılmış uygulamalardır. (**n11, Aliexpress, Letgo vs....**) Biz de bu sistemi genele yönelik değil de daha önceden ele alınmamış bir problem üzerinden yaklaşarak üniversite içi e-ticaret sistemi olarak ele aldık.
2. İnsanlar alışveriş sitelerinden alışveriş yaparken satın aldıkları ürünü tedarik eden kişi ya da firmaya karşı güven sorunu yaşayabiliyorlar. Bu soruna çözüm olarak projemizde üniversite içi ticaretin öğrenci ve personeller için daha güvenilir olduğu düşünüldü.
3. E-ticaret üzerinden yapılan alışverişlerde satın alınan ürünün müşteriye ulaştırılması için kargo sistemine ihtiyaç duyulur. Bu sistem zaman ve ulaşım maliyet bakımından ekstra bir yük oluşturur. Tasarladığımız sistemde üniversite içi bir ticaret sistemi olduğu için ürün teslimat işlemi satıcı-müşteri arasında direkt olarak gerçekleşir. Böylece ulaşım maliyeti ve zamandan tasarruf edilmiş olur.
4. Proje belirli bir gruba hitap ettiği için ve bu grup belirli ortak özelliklere sahip olduğu için kullanıcılar tarafından ihtiyaç duyulan ürün profili herkes aynı olacaktır. Bu sayede bir ürün daha kısa zamanda alıcı bulacaktır.
5. Örnek verecek olursak ikinci el kitap, ev eşyası, deney malzemeleri, elektronik eşyalar, ders notları vb. öğrenci ve okul personeline yönelik ürün alım satımı gerçekleştirilebilir.

B. Kullanıcı Senaryoları

1. Senaryo-1(Sisteme kayıtlı olmadığı durum):

Kullanıcı uygulamayı açar, açılan ekranda sisteme **@gtu.edu.tr** uzantılı e-maili, adı, soyadı ve belirlediği şifresi ile sisteme kaydolur.

2. Senaryo-2(Sisteme kayıtlı olduğu durum-alıcı):

Kullanıcı uygulamayı açar. E-mail uzantılı kullanıcı adı ve şifresi ile sisteme giriş yapar. İsteddiği ürünü arayabilir. Beğendiği takdirde ürünü sepete ekleyip satın alabilir.

3. Senaryo-3(Sisteme kayıtlı olduğu durum-alıcı):

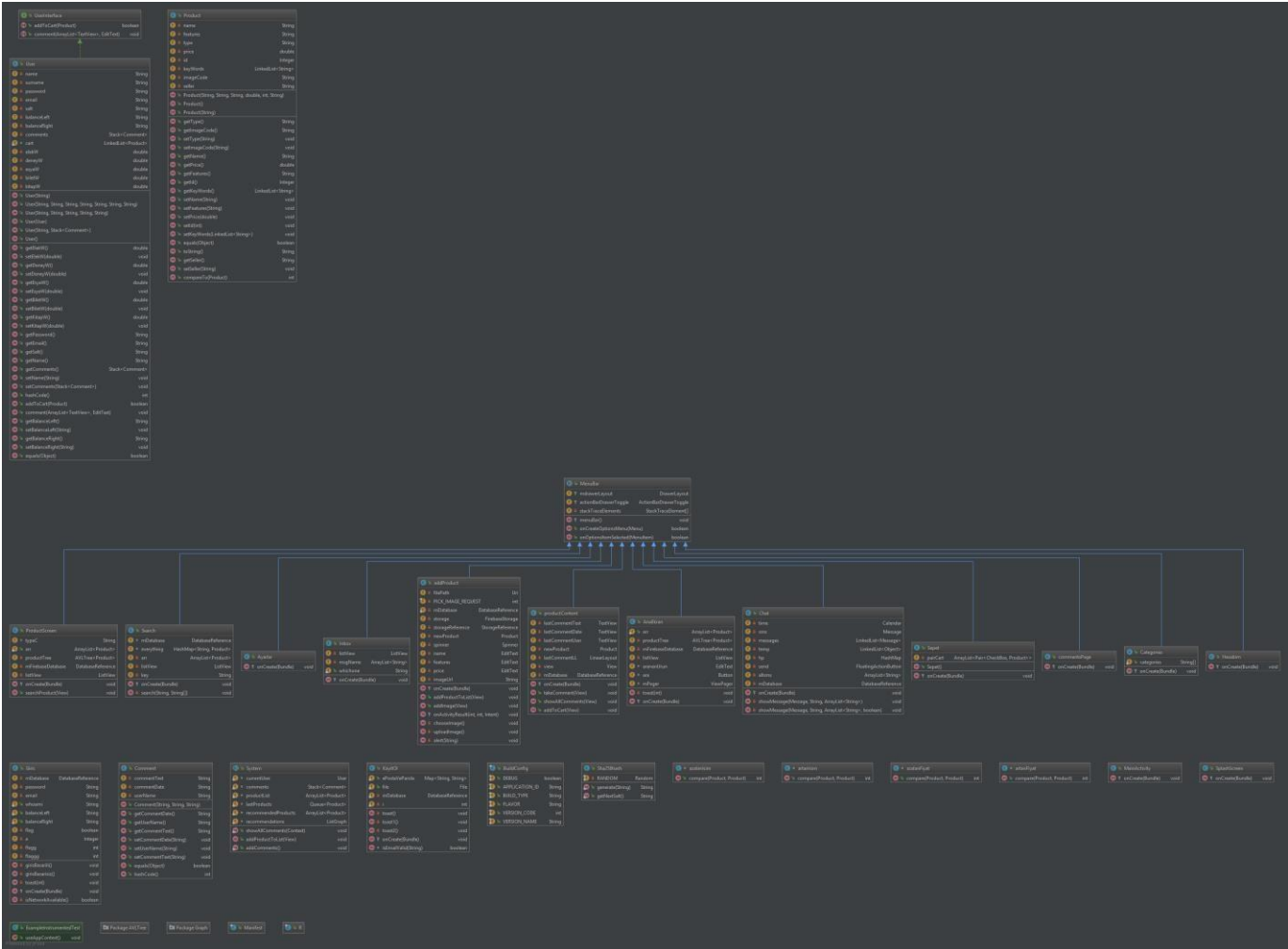
Kullanıcı uygulamayı açar. E-mail uzantılı kullanıcı adı ve şifresi ile sisteme giriş yapar. Ana sayfada bulunan popüler ürünlerden bir ya da birkaçını seçip bunları sepetine ekler. Ve yine isteğe bağlı olarak ürünü satın almak için satıcıya mesaj gönderir ya da ürünleri siler.

4. Senaryo-4(Sisteme kayıtlı olduğu durum-alıcı):

Kullanıcı uygulamayı açar. E-mail uzantılı kullanıcı adı ve şifresi ile sisteme giriş yapar. İster ana sayfada bulunan kategoriler butonu üzerinden isterse de soldan kayan menüde bulunan kategoriler butonu ile kategorilere girer. İsteddiği kategoriye seçer. Beğendiği ürünleri üzerine tıklayarak detaylı inceleyebilir. Beğendiği ürünleri sepetine ekler.

Diyaframdan da görüldüğü gibi kullanıcı hem alıcı olarak hem de satıcı işlemler gerçekleştirebilmektedir. Bir kullanıcı sisteme giriş yapmadan uygulamaya hiçbir erişimi yoktur. Kullanıcıyı alıcı olarak ele alırsak sisteme kayıtlı değilse kayıt olabilir, ürün arayabilir, popüler ürünleri görebilir, ürün satın alabilir (kullanıcıya mesaj gönderir), ürünleri sıralayabilir. Kullanıcıyı satıcı olarak ele alırsak alıcının yaptığı her şeyi yapabilir ve buna ek olarak sisteme bilgileri ve fotoğrafı ile ürün ekleyebilir.

Sınıf Diyagramı



SearchTree		
m	add(E)	boolean
m	contains(E)	boolean
m	find(E)	E
m	delete(E)	E
m	remove(E)	boolean

BinaryTree		
f	root	Node<E>
m	BinaryTree()	
m	BinaryTree(Node<E>)	
m	BinaryTree(E, BinaryTree<E>, BinaryTree<E>)	
m	getLeftSubtree()	BinaryTree<E>
m	getRightSubtree()	BinaryTree<E>
m	getData()	E
m	isLeaf()	boolean
m	toString()	String
m	preOrderTraverse(Node<E>, int, StringBuilder)	void

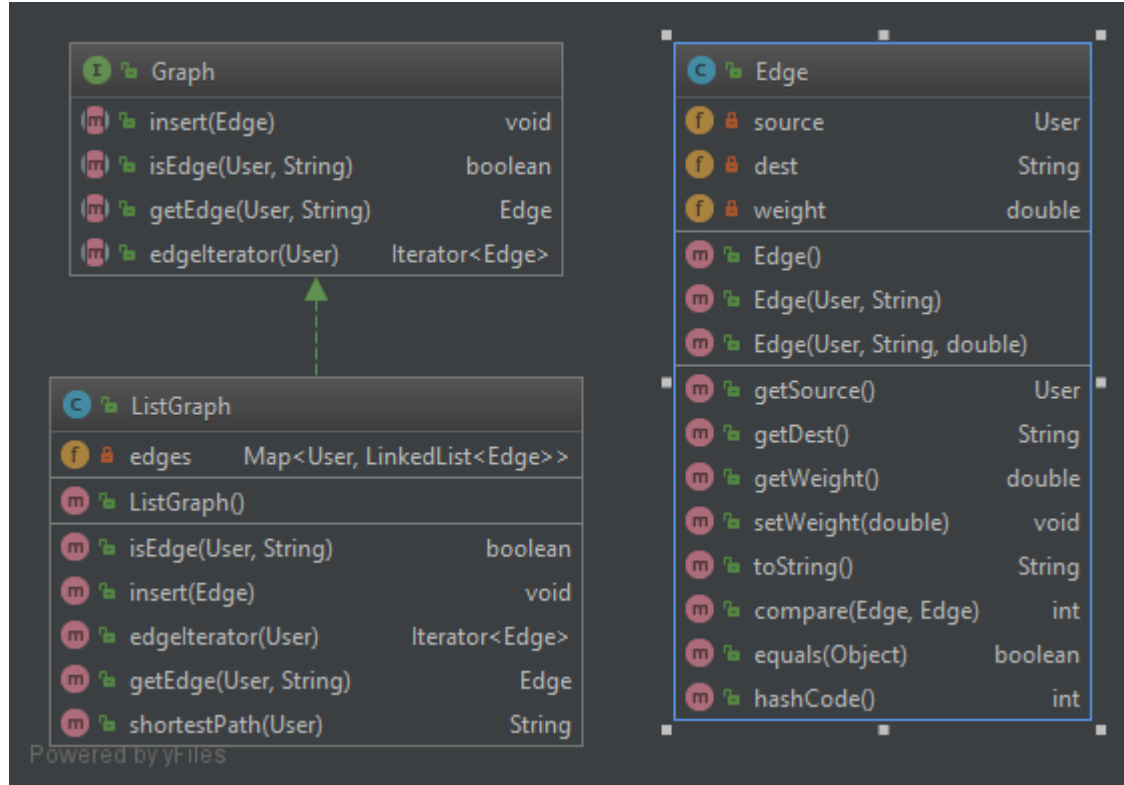
BinarySearchTree		
f	addReturn	boolean
f	deleteReturn	E
m	find(E)	E
m	find(Node<E>, E)	E
m	add(E)	boolean
m	add(Node<E>, E)	Node<E>
m	delete(E)	E
m	delete(Node<E>, E)	Node<E>
m	remove(E)	boolean
m	contains(E)	boolean
m	findLargestChild(Node<E>)	E

Node		
f	data	E
f	left	Node<E>
f	right	Node<E>
m	Node(E)	
m	toString()	String

BinarySearchTreeWithRotate		
m	rotateRight(Node<E>)	Node<E>
m	rotateLeft(Node<E>)	Node<E>

AVLTree		
f	increase	boolean
f	decrease	boolean
m	add(E)	boolean
m	add(AVLNode<E>, E)	AVLNode<E>
m	delete(E)	E
m	delete(AVLNode<E>, E)	AVLNode<E>
m	findReplacementNode(AVLNode<E>)	AVLNode<E>
m	findLargestChild(AVLNode<E>)	E
m	incrementBalance(AVLNode<E>)	void
m	rebalanceRight(AVLNode<E>)	AVLNode<E>
m	rebalanceLeftR(AVLNode<E>)	AVLNode<E>
m	rebalanceRightL(AVLNode<E>)	AVLNode<E>
m	rebalanceLeft(AVLNode<E>)	AVLNode<E>
m	decrementBalance(AVLNode<E>)	void

AVLNode		
f	LEFT_HEAVY	int
f	BALANCED	int
f	RIGHT_HEAVY	int
f	balance	int
m	AVLNode(E)	
m	toString()	String



10 tanesi arayüz için kullanılmak üzere toplam 21 tane sınıf oluşturulmuştur. Uygulamanın tüm sayfalarında kayan menü bulunmaktadır. MenuBar sınıfı (AppCompatActivity sınıfı miras alınmıştır.) kayan menüyü oluşturur ve tüm arayüz sınıfları tarafından miras alınmıştır.

Giriş sınıfında Map yapısı kullanılmıştır. <e-mail, şifre> şeklinde belirlenmiş bu şekilde arama gerçekleştirilmektedir.

Product sınıfı ürün oluşturmak için kullanılmakta ve diğer sınıflar tarafından kullanılmaktadır. Ürün oluşturmak, ürün eklemek, ürün bilgilerini kullanıcıya göstermek, sepete ürün eklemek için Sepet, ProductScreen, ProductContent gibi sınıflar tarafından kullanılmaktadır.

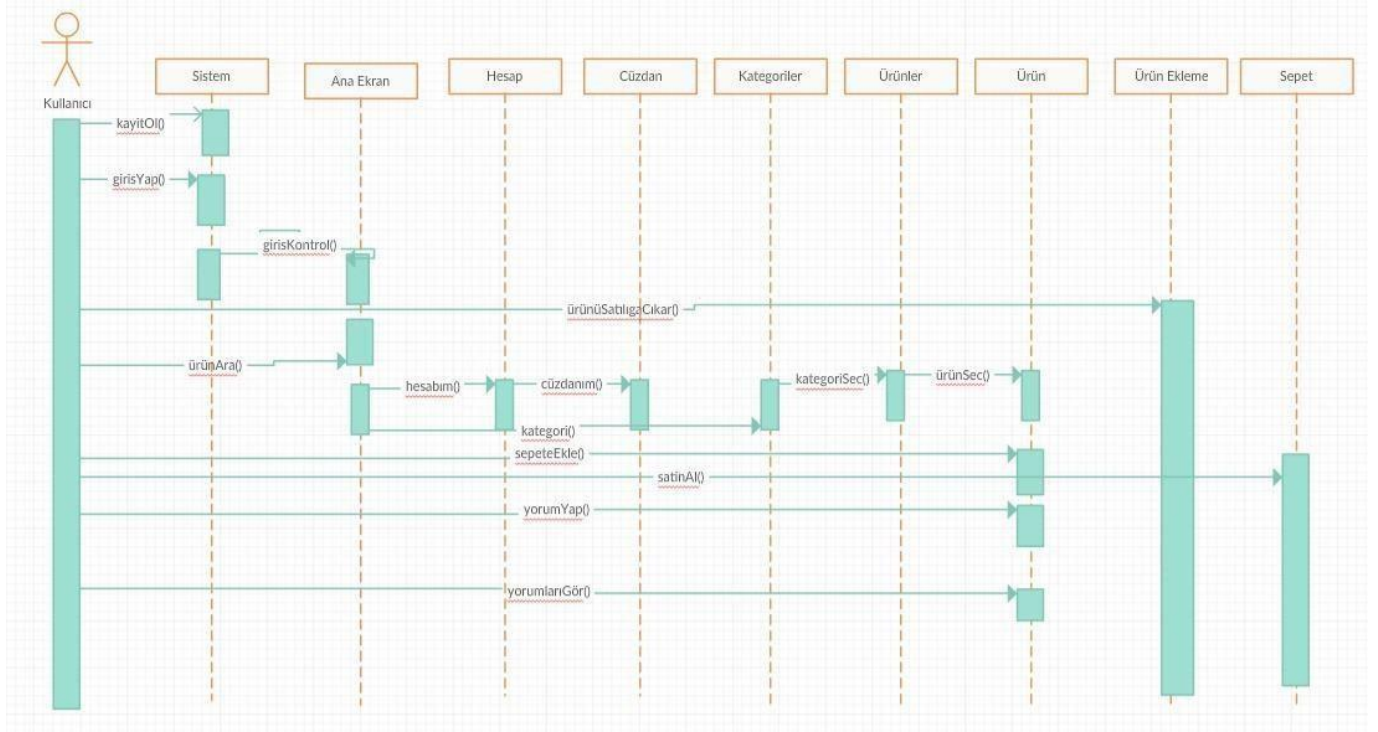
UserInterface soyut sınıfı oluşturulmuştur. Bu sınıfta ürün parametresi alan ve sepete ürün eklemek için addToCart(Product) metodu ve yorum eklemek için comment () metodu tamamlanmak üzere bırakılmıştır. User sınıfında UserInterface in metotları tamamlanmıştır.

User sınıfında comment metodu tamamlanırken Comment sınıfı kullanılmaktadır. Yığın Comment tipinde oluşturulmaktadır. Çünkü en son yapılan yorum en üstte olmalıdır. addToCart metottu tamamlanmak için Product sınıfı kullanılmıştır. Sepete ürün eklemek için kullanılmaktadır.

Sepet sınıfında satın alınmak istenen ürünler bulunmaktadır. Bunun için bu metot yazılmıştır. Ürünleri tutmak için ürün sayısı değişeceği için dinamik bir yapıya ihtiyaç duyulmuş ve bu ihtiyaç ArrayList ile giderilmiştir.

Comment sınıfında yorumlar için oluşturulmuştur. String olarak yorum, tarih ve kullanıcı adı değişkenleri bulunur.

Sıra Diyagramı



Kullanıcı önce sisteme kayıt olur. Ardından sisteme giriş yapar. **Sistem** bu girişi e posta ve şifre ikilisine bakarak kontrol eder. Giriş başarılı olduğunda kullanıcı **Ana Ekran**a yönlendirilir. **Ana ekranda** kullanıcı ürün arayabilir. Ya da sol üstteki üç çizgi butonu ile **hesap**, **kategoriler** ve **ürün ekleme**e gidebilir. **Kategori** bölümünden kategori seçilebilir ve seçilen kategoride bulunan ürünler arasından ürün seçebilir. Daha sonra seçilen bu ürünü kendi **sepetine** ekleyebilir, yorum yapabilir ya da yorumları görüntüleyebilir. Eğer **sepete** eklenen ürün satın alınmak isteniyorsa sağ üstteki sepet butonuna tıklanarak **Sepet** bölümüne ulaşılır ve istenilen ürün seçilerek Satın alma işlemi için satıcıya mesaj gönderilir.

Eğer kullanıcı ürününü satışa çıkarmak istiyorsa **Ürün ekleme** bölümüne giderek istediği ürünü satılığa çıkartabilir.

D. Şimdiye Kadar Kullanılan Veri Yapıları

- ✚ Her müşterinin satın aldığı ürünlerin eklendiği bir ürün sepeti vardır. Bu sepetteki ürünlerin bilgilerinin tutulması için **bağlı liste** veri yapısı kullanıldı. Çünkü bağlı liste yapısı, ürün sepetine eklenmiş olan bir ürün sepetten çıkartıldığında geriye kalan ürünlerin hızlı ve etkili bir şekilde tekrar düzenleme olanağını bize sunar.
- ✚ Ürünleri kategorize olarak listelerken **ArrayList** yapısı kullanıldı.
- ✚ Ürüne gelen yorumlar **yığın** veri yapısı kullanıldı. Bu sayede en son gelen yorumlar **hızlı** şekilde en üstte gösterilebilecek. Çünkü yığın yapısında son gelen ilk çıkar mantığı işlemektedir.
- ✚ Kullanıcı adı ve şifrelerinin tutulması için **Map** yapısı kullanıldı. Her kullanıcının benzersiz bir kullanıcı adı vardır. Map yapısında, anahtar olarak okul

uzantılı e-mail değeri olarak şifre tutulacak. Bu sayede anahtar olarak okul uzantılı email verildiğinde o kullanıcının şifresine kolaylıkla erişilmiş olacak

- ✚ Kullanıcının daha önceden aldığı ürünlerden yola çıkarak ürün tipine göre kullanıcıya öneri sunulması için **çizge** kullanılacaktır. Kullanıcı ve ürünler arasında kenar(edge) ilişkisi oluşturulur, ürünler ve kullanıcılar tepedir(vertex). Başlangıçta tüm kenarlara maksimum bir ağırlık verilir. Sonrasında kullanıcının gezdiği kategorinin kenarının ağırlığı azaltılır. Yani en az ağırlığa sahip kenarın kategorisi en çok ziyaret edilmiş olur. Sonuç olarak bu graph üzerinde en kısa yol algoritması uygulanıldığında kullanıcının en çok ziyaret ettiği kategori bulunmuş olur.

E. Yeni Eklenen Veri Yapıları

- ✚ Kullanıcı menüden kategorilere tıkladığında istediği bir kategoriye seçer ve ardından kategorideki ürünler listelenir. Burada istediği ürünü arayabilir. Bu aramayı yapmak için ürünler **AVL ağacında** tutuldu ve arama işlemi **AVL ağacında** yapıldı.
- ✚ Kullanıcının, arama sonucunda yada kategorilerde karşısına çıkan ürünleri sıralaması için Javanın kendi **MergeSort** u kullanıldı. Bu sort sayesinde ürünler fiyatına göre yada alfabetik olarak sıralandı.

F. Uygulamadan Görüntüler



Kayıt Ekranı(Doldurulmuş)

GTÜ E-Alışveriş

[Geri Dön](#) [Kayıt Ol](#)

Kayıt Olundu

GTÜ E-Alışveriş

[Kayıt Ol](#) [Giriş Yap](#)

Kayıt Başarılı. Lütfen Giriş Yapınız.

Kayıtsız E-Posta İle Giriş

GTÜ E-Alışveriş

[Kayıt Ol](#) [Giriş Yap](#)

Bu E-posta Adresi Sisteme Kayıtlı Değil.

Ana Ekran

GTÜ E-Alışveriş

İlgi Alanınıza Göre

asus vv121

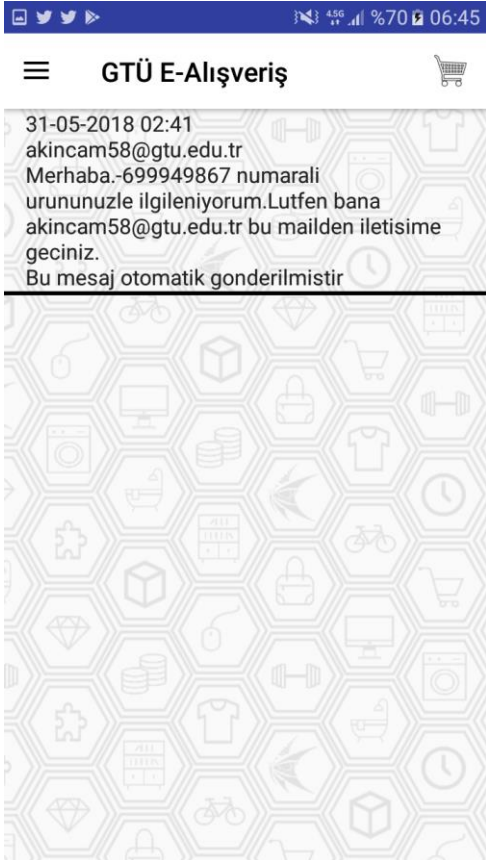
intel core i7 işlemci, 8 gb ram, sıvı soğutmalı, acil satılık pazarlık yapılır.

1995.0 TL

Kayan Menü



Mesajlar



Hesabım Ekranı



Kategoriler



Elektronik Kategorisi - Ürünler





Ürün Arama Sonucu



Sepetim



Yorumlar



Sepet Uyarı

