

Gebze Teknik Üniversitesi

Bilgisayar Mühendisliği

CSE 222 - 2018 Bahar
Rapor 3

14 Mayıs, 2018

Hazırlayanlar

Mehmet Sami ERTEKİN	05104400
Sinan ELVEREN	111044074
Mesut BUNALDI	111044077
Ahmet ÖZYILMAZ	111044014

Asistanlar:

Fatma Nur ESİRCİ
Tuğbagül ALTAN AKIN
Mehmet Burak Koca

1. Giriş

Veri Yapıları ve Algoritmalar Dersi kapsamında öğrencilerin derste öğrendiklerini uygulayarak pekiştirebilmeleri için proje ekibi oluşturarak belirlemiş oldukları projeyi hayata geçirmeleri ve raporlamaları istenmektedir. Bu doğrultuda devam etmekte olduğumuz projemize ait hazırlamış olduğumuz rapor içerisinde; kullandığımız veri yapılarını, oluşturduğumuz sınıfları ve bu sınıflar arasındaki ilişkileri belirten sınıf diyagramlarını , kullanıcı arayüz tasarımını ve projenin işleyişini genel olarak belirten sıralı diyagramı görebilirsiniz.

2. Kullanılması Planlanan Veri Yapıları

2.1. Liste Yapısı

Her kullanıcı sınırlı bir envantere sahip olacak bu envanterleri tutmak için “ArrayList” kullanılacaktır.

2.2. İkili Arama Ağacı

Ürünlerin sıralanması gerektiğinde ürünler map veri yapısından ikili arama ağacında tutulacaktır.Kullanıcı ürünleri mağazaya ekleniş zamanına göre sıralamak istediğinde ağaçtaki eşyaları “INORDER” traverse ederek ekleniş tarihine göre ilk önce eklenenden son eklenene göre listelenecektir.

2.3. Öncelikli Kuyruk

Eşya alış ve satışları sırasında oluşturulan kayıtları “Zaman” ölçekli öncelikli kuyruk veri yapısında tutmayı tasarladık.Bu sayede bu veri yapısından ilk çekilecek verinin zaman olarak en önce kuyruğa eklenen veri olacaktır.

2.4. Set/Map Yapısı

Map(Harita) i neden kullanıyoruz ? Mağazadaki eşyaların birbirine benzemeyen ID’leri olacaktır. Ve bu ID’lerle ürünlere erişim için ürünleri “Map” veri yapısında tutmaya karar verdik böylelikle erişimleri algoritma karmaşıklığı olarak (sabit zamanda)“Constant Time “ da gerçekleşecektir.

2.5. Çizge(Graf) Veri Yapısı

Proje içerisinde kullanıcıların birbirlerini takip edebilmesi için bir yapı oluşturulacaktır. Bu yapı çift yönlü çizge veri yapısı ile gerçekleştirilecektir. Birbirini takip eden kullanıcılar çizge yapısına eklenerek, kullanıcıların birbiri ile bağlantıları üzerinden, ilişkisi olduğu kullanıcıların mağazadaki ürünlerini görebilecek(filtreleme) Bu yapılan işlemler çizge üzerinden yapıldığında daha verimli şekilde gerçekleştirilmiş olacaktır. Sonraki aşamalarda verilerimizin daha kolay ve hızlı analiz edilebilmesi için,

çizge üzerinden farklı ilişkiler ekleyebiliriz. Verinin daha kolay ve hızlı analiz edilebilmesi için, farklı ilişkiler eklenebilir, çizge veri yapısı üzerinde daha az gezinerek sonuçlara daha hızlı erişim sağlanabilir.

2.6. Ağaç Veri Yapısı

2.6.1. Kırmızı-Siyah Ağaç Yapısı

Projede çok fazla veri girişi olacağı ve arama/sıralama işlemleri çok fazla olacağı için, bu verilerin hızlıca erişilip yönetilmesi gerektiğinden kırmızı siyah ağaç yapısı kullanmayı tercih ettik. Veriler bu yapıya eklendiği için çok sayıda veri eklenip, silinmesi ya da aranması durumunda bile, dengeli ağaç yapısını koruyabilecek bir yapı olduğu için en kötü durumda bile çalışma zamanı $O(\log n)$ olacaktır. 2.2 de açıklanan durum bu veri yapısı ile tasarlanacaktır.

3. Algoritmalar

3.1. Yinelemeli Yöntemler

3.1.1. Kökün Ortada Olduğu Dolaşım Yöntemi (Inorder) ve Kökte Sonlanan Dolaşım Yöntemi (Postorder)

Kırmızı-Siyah Ağaç Yapısı üzerinde saklanan verileri sıralı bir şekilde listelemek için bu yöntemi kullanacağız. Bu sıralamalar; zaman sıralı, fiyat sıralamalı, satıcı ismi alfabetik sıralı, ürün ismi alfabetik sıralı şeklinde olacak.

Arama algoritmalarında ağaç üzerinde olan veriler için yenilemeli yöntemler kullanılacak, ayrıca bazı sıralama algoritmaları hali hazırda yenilemeli olarak tasarlanmış olacak.

Ağaç üzerinden yapılan bir çok işlem(ekleme, silme, sorgulama vb) rekürsif şekilde gerçekleştirilecek.

3.2. Arama / Sıralama Yöntemleri

Projede özgün tasarım olarak; yapıdaki verinin büyüklüğü değiştiği durumda çalışma zamanı daha iyi olacak algoritmaları kullanmayı tasarlanıyor. Bu sayede az sayıda kullanıcı veya veri olduğu durumlarda maliyeti düşük olan karmaşıklığı yüksek algoritmaları kullanıp, verinin büyüklüğü arttıkça karmaşıklığı düşük olan algoritmaya geçiş yapılacaktır.

Siyah-Kırmızı ağaç üzerinde ikili ağaca uygun olmayan veriler “Birleştirme Sıralaması” algoritması ile sıralanıp, özelleşmiş aramalar yapılabilecek. Veri miktarı çok değişken olduğundan dolayı ve “Birleştirme Sıralaması” en kötü durumda bile $O(\log n)$ zamanda çalıştığı için seçilmiştir.

Ürünlerin sıralanması gerektiğinde Siyah Kırmızı Ağaç Yapısı ya da İkili Arama Ağacı kullanıyor olacağız ve ağaçtan sıralı şekilde elemeyi alabildiğimiz için için arama yöntemi olarak ta ikili arama algoritmasını kullanmayı tasarladık. Ortalama çalışma süresi $O(\log n)$ olan bu algoritma en iyi durumda $O(1)$ zamanda çalışmaktadır.

4. Arayüz ve Özellikleri

4.1. Giriş Ekranı Arayüzü

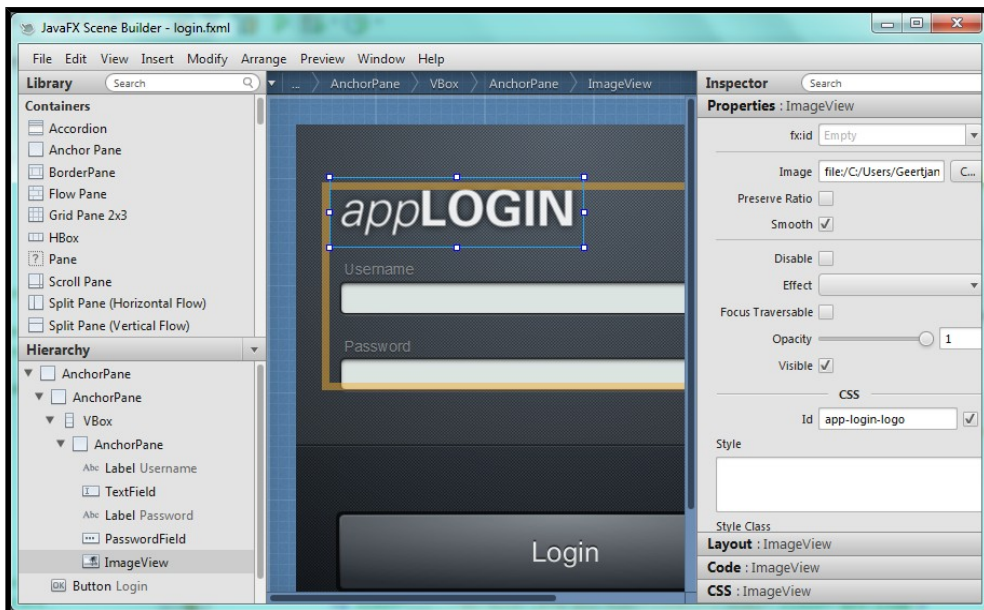
İsim ve şifre ile sisteme girilecek ve kullanıcı adına göre kullanıcının yönetici veya oyuncu olduğu belli olacak. Yeni kayıt için ise aşağıdaki bölüm kullanılabilir. Genel olarak durum böyle olmakla birlikte tasarımda değişiklikler olabilir.

4.2. Kullanıcı Ekranı Arayüzü

Kullanıcıların giriş yaptıktan sonra eşya alıp, eşyayı satışa koyabilecekleri bölüm. Tasarımın değişme olasılığı vardır.

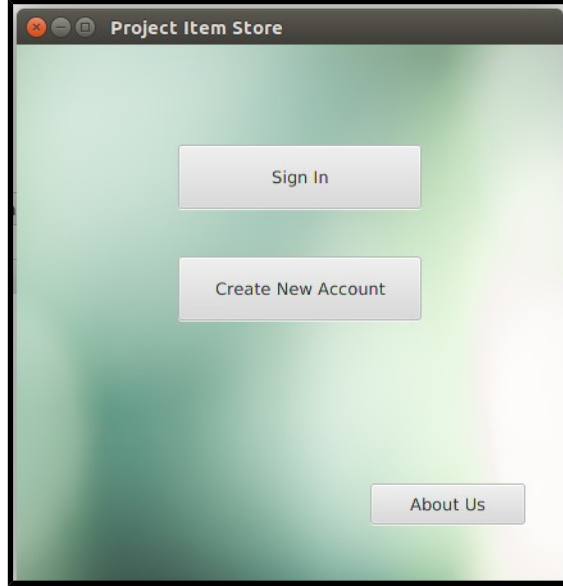
Projenin arayüz tasarımı için yaptığımız çalışmaların zaman maliyetini en aza indirmek ve yeni teknolojiler öğrenip projemiz üzerinde kullanabilmek için yapmış olduğumuz araştırmalar sonucu, kullanılacak araç olarak Oracle'ın JavaFX Scene Builder aracında karar kıldık.

JavaFX kısaca; Swing in yerine yeni bir teknoloji olarak gelmiş Swing' in eksik yönlerini güçlendirmiştir. JavaFX, görsel olarak daha esnek ve gaha güçlü, 3D, Canvas, Audio, Video, Web Service' leri çağırabilme CSS ile şekillendirebilme özellikleriyle arayüz tasarımının daha kolay yapılabilmesini sağlayan bir araçtır. Scene Builder, bu aracın içerisinde bize sürükleyip bırak yöntemi ile tasarımlarımızı kolayca oluşturup xml tabanlı dosya yapısına benzer şekilde projemize kullanıcı arayüzünü dahil edebilme imkanını bize sunmaktadır. Böylece zaman maliyeti en aza indirilerek kullanıcı dostu arayüzler tasarlanabilmektedir. Scene Builder aracının kullanıcı arayüzü aşağıdaki görselde yer aldığı gibidir.

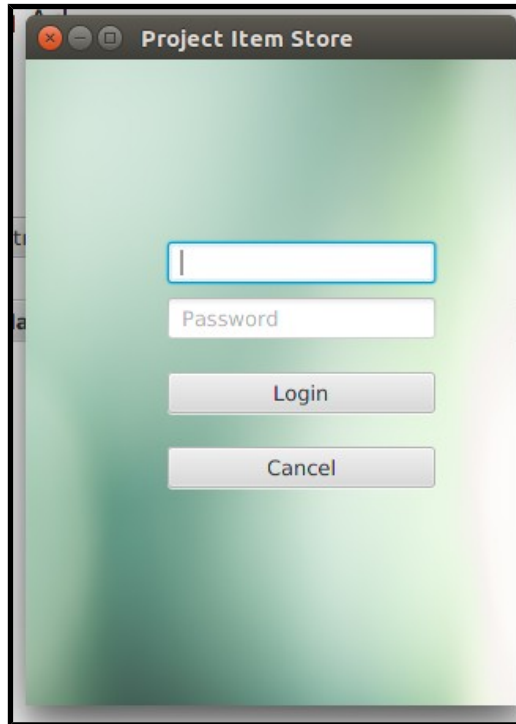


Proje arayüz tasarım görselleri :

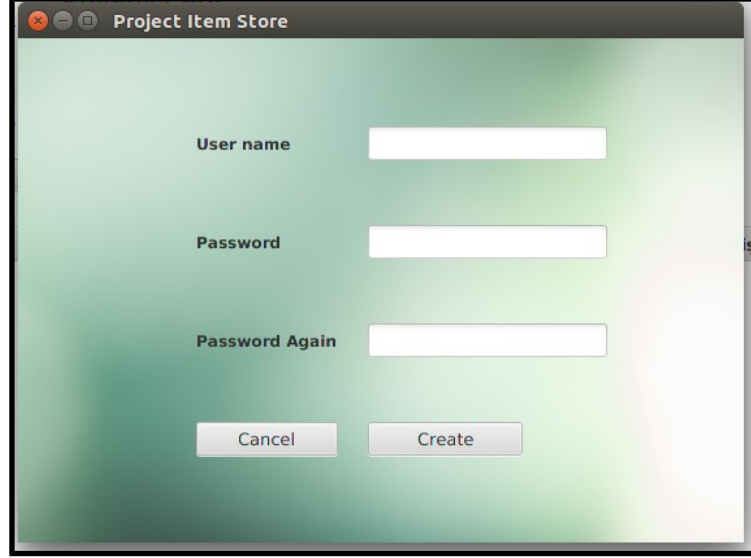
Uygulamanın açılış ekranında bizi yapmak istediğimiz işlemi seçmek üzere bir pencere karşılamaktadır. Buradan yeni hesap oluşturabilir, hesabımız zaten varsa giriş yapabilir ve uygulamayı kullanmaya başlayabiliriz. About kısmında da program hakkında bilgiler yer almaktadır.



Kullanıcı giriş ekranı , kullanıcı adı ve şifresi daha önceden kayıtlanmış bir hesaba aitse giriş yaparak ana ekrana gidilmesini sağlayan pencere.



Yeni hesap oluřturma penceresi, uygulamayı kullanmak isteyen kullanıcının yeni bir hesap oluřturması için kullanabileceđi pencere.

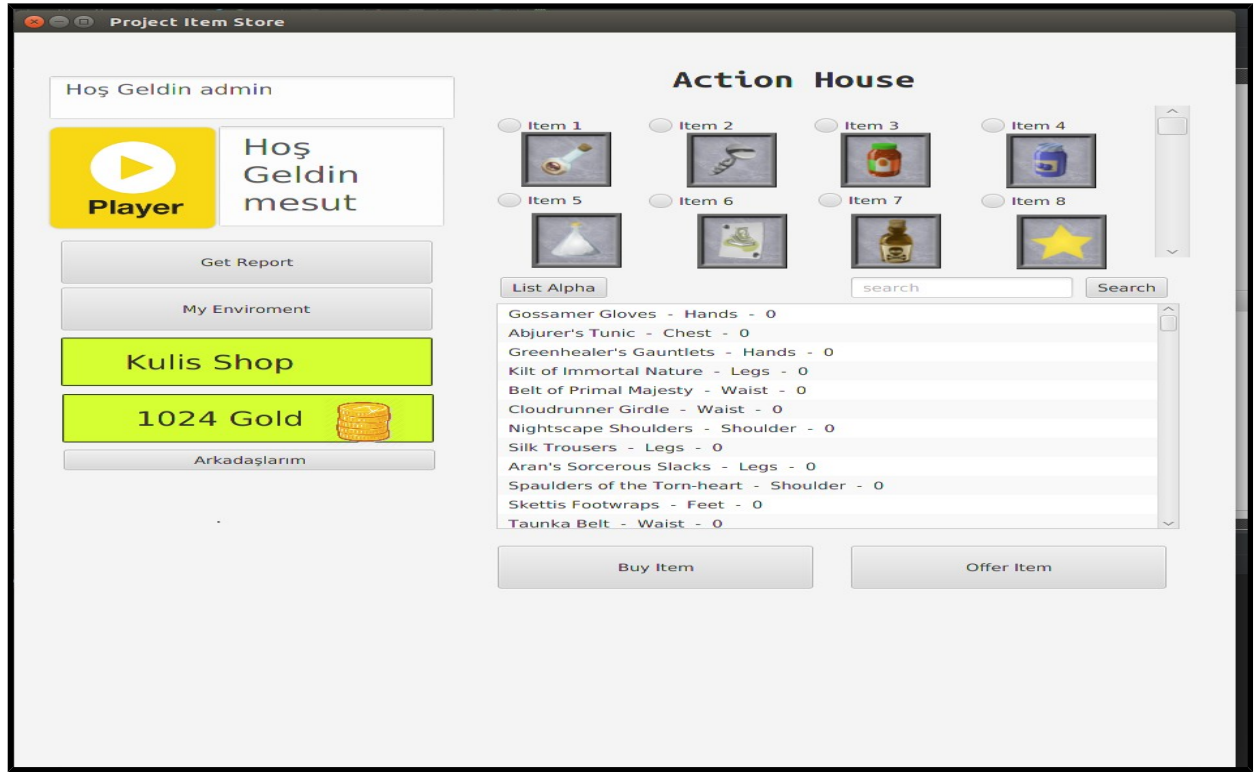
A screenshot of a registration window titled "Project Item Store". The window has a light green background with a subtle gradient. It contains three input fields: "User name", "Password", and "Password Again". Below these fields are two buttons: "Cancel" and "Create". The window has standard OS window controls (minimize, maximize, close) in the top left corner.

Herhangi bir eřya üzerine gelindiđinde satın alma iřlemi yapabilmek için gereken bedelin gzktđ ve satın alma butonun yer aldđđı aılır-kapanır pencere.

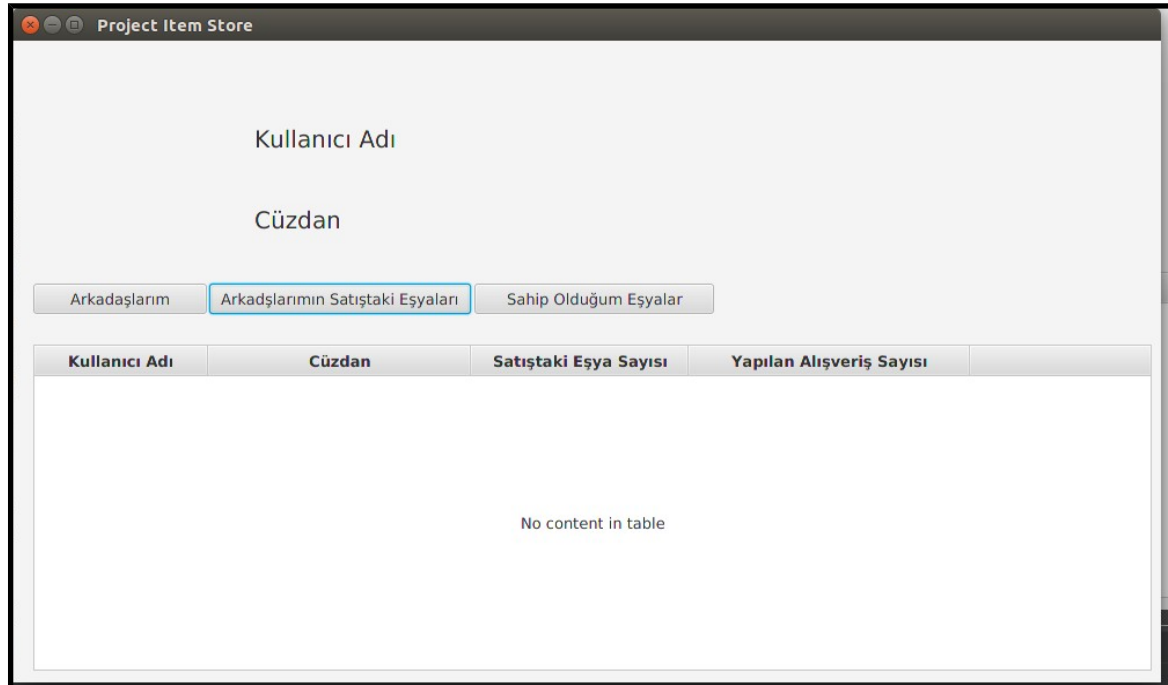
A screenshot of a purchase popup window. The popup has a yellow background and is set against a dark gray background. It displays an item card for "Mythical Axe of Fre...". The card includes an icon of an axe, the item name, a description "Axe is powerful item for war...", and a table showing the price and value. A "BUY" button is located to the right of the table.

Price	130
Value	145

Ana pencere, kullanıcı giriři yaptıktan sonra karřımıza ıkan bu pencerede eřyaları listelebilir , listelenen bu eřyaların zelliklerini grebilir , eřyalara teklif verebilir ya da direk satın alma iřlemi gerekleřtirebiliriz. Diđer yandan kullanıcı bilgilerinin grntlenmesi ve isteniyorsa deđiřtirilmesi iřlemleri de bu pencereden gerekleřtirilebilecektir.



Kullanıcı profil sayfası , kullanıcının kendi bilgileri ve takip ettiği kullanıcılara ait bilgileri görebileceği sayfadır. Bu sayfada veri yapısı olarak çizge kullanılması amaçlanmaktadır.

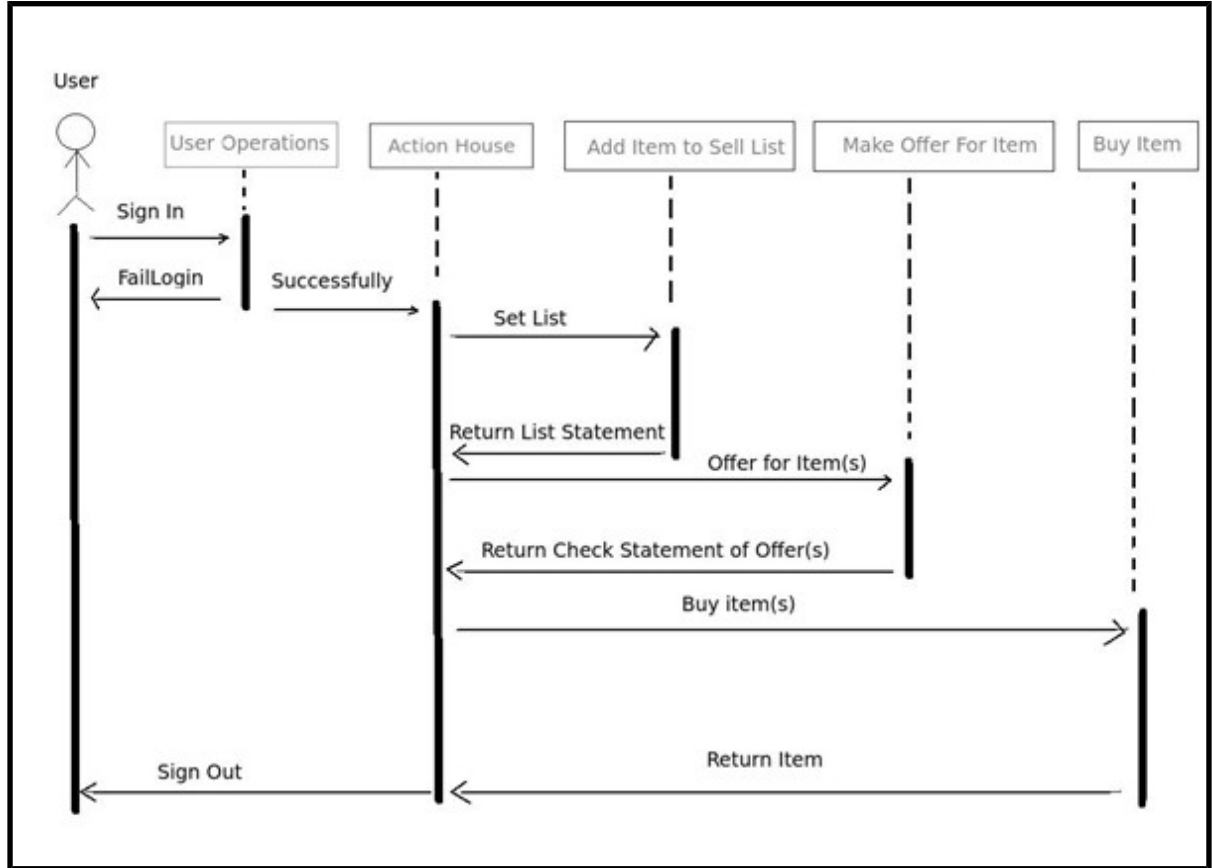


Arayüz tasarımı geliştirilmeye devam edilmekte olup, bu tasarımlar şuan projenin gelinen safhadaki görüntüleridir.

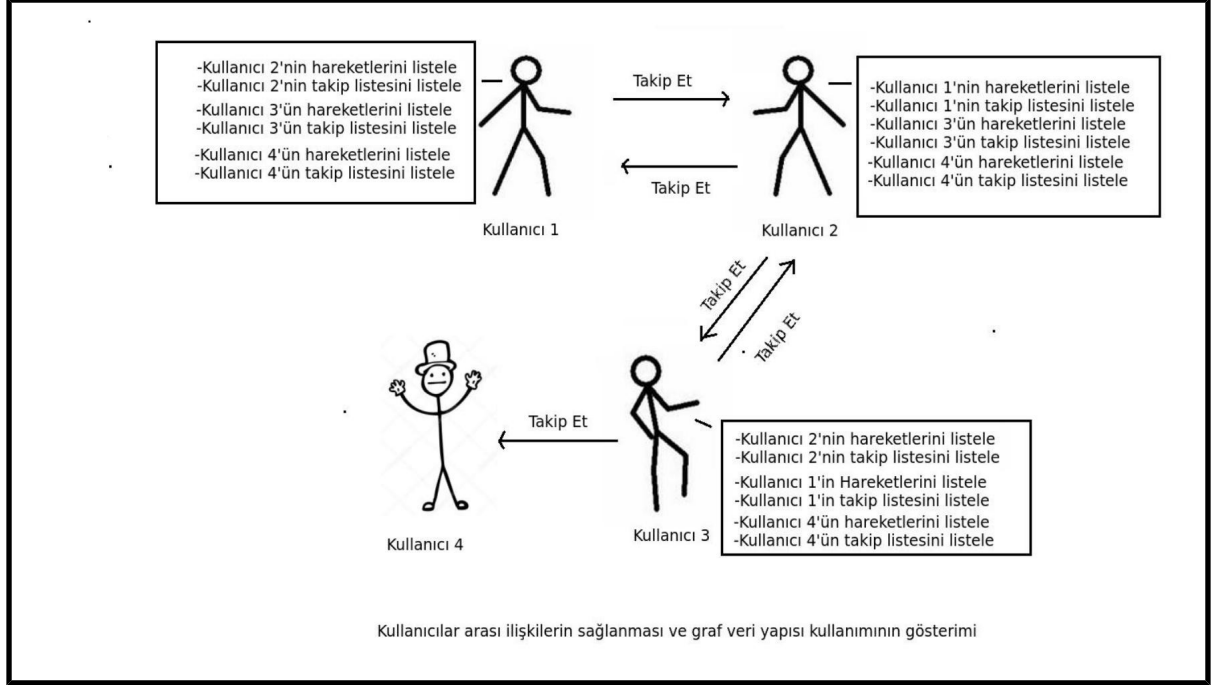
5. Diyagramlar

5.1. Sıralı (Sequence) Diyagram

Sıralı (Sequence) diyagram üzerinde gösterildiği üzere , kullanıcı sisteme giriş yapar ve giriş yaptığı anda programın kendine sunduğu aksiyonları gerçekleştirebileceği ekrana yönlendirilir. Bu ekran üzerinde satışa çıkarmak istediği eşyayı satış listesine koyabilir, satışa çıkarılmış eşyalara teklif verebilir ya da direk eşyayı satın alabilir. Teklif verilen eşyalarda durum kontrol edilir ve şartlar sağlanıyorsa teklif verilen eşya otomatik olarak kullanıcının envanterine eklenir.



5.2. Kullanım Diyagramı (Use Case)



5.3. UML Diyagram

Tasarlamış olduğumuz Sınıf Diyagramları şekildeki gibidir.

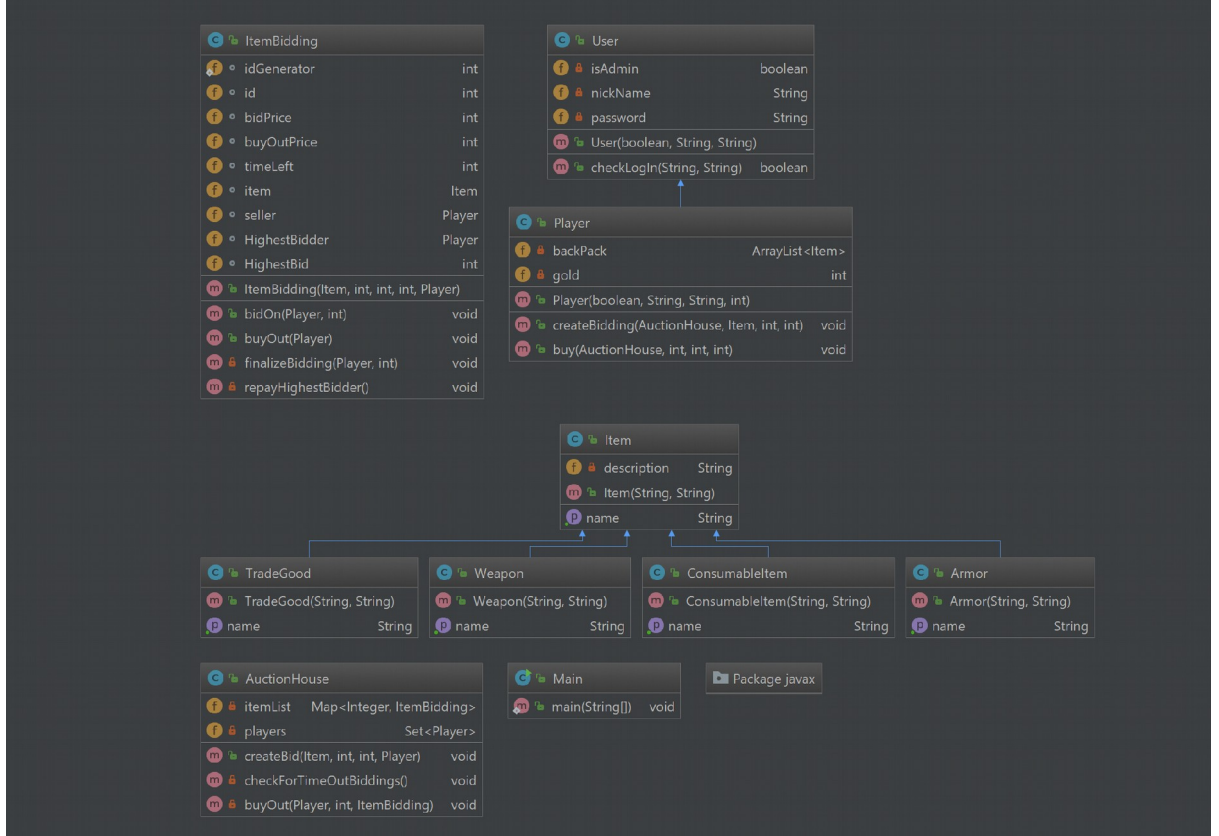
Diyagramdan da anlaşılacağı gibi soyut olarak tanımlanmış olan sınıfımız Item sınıfıdır. Kullanıcı (User) sınıfı dışında kalan Tüketilebilir(Consumable), Zırh(Armor) ve Silah(Weapon) sınıfları Item soyut sınıfından türetilmiştir.

Türetilmiş olan 3 ana sınıfı ayrı ayrı incelemek gerekirse ;

Consumable Sınıfı : Genel tüketim eşyalarının ya da ürünlerinin türetilmiş olduğu ana sınıfımızdır. Diyagramda da görüldüğü gibi tüketilebilir sınıflarımız olan Foods,Elixir,Drinks,Bandages sınıfları bu sınıftan türetilmiştir.

Armor Sınıfı : Giyilebilir eşyaların türetildiği ana sınıfımızdır. Diyagramda da görüldüğü gibi hangi tarzda giyilebilir eşyalar var ise ; Ayak(Feet), Baş(Head), Göğüs(Chest) , Kol(Hands) ve Bacaklar(Legs) sınıfları olarak bu ana sınıfımızdan türetilmiştir.

Weapon Sınıfı : Adından da anlaşılacağı üzere tahrip edici eşyaları temsil eden Silah sınıfıdır. Bu ana sınıftan da türetilmiş 5 alt sınıf bulunmaktadır. Bunlar da 5 farklı kullanılabilir eşyayı temsil eden sınıflardır.



6. Projenin Fonksiyonları

- Kullanıcı Kayıt
- Kullanıcı Giriş
- Kullanıcı çıkış
- Eşya listeleme
- Satışta olan eşyalar
- Sayfayı yenileme
- Eşya arama
- Eşya Fiyat Bıçme
- Eşyayı Satışa Koyma
- Kullanıcıların birbirini takip etmesi
- Birbirini takip eden kullanıcıların birbirlerinin hareketlerini listeleyebilmesi

7. Projenin Çalışma ortamı

Windows 8

Windows 10

Mac OS X

Linux

JAVA 8 ,9 u destekleyen bütün platformlarda proje çalışacaktır.