



# به نام خدا

## درس برنامه سازی پیشرفته

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

ترم پاییز ۱۴۰۵-۱۴۰۴

استاد :

علی نجیمی

---

عنوان پروژه :

## Amazon Clone

Product Owner

آین کریمیان

Product Manager

محمد حسین سورانی

# فهرست مطالب

نکته قابل توجه

مشخصات پروژه

اهداف کلی

اهداف پروژه

شرح خلاصه‌ی پروژه

پروژه

۴	تحلیل نیازمندی‌ها
۶	طراحی و ساختار فنی
۸	تقسیم‌بندی نمره
۹	چک لیست تحویل نهایی



## نکته قابل توجه

- این داک صرفا توضیحات مربوط به محتوای پژوهه **Amazon Clone** می‌باشد. توضیحات مربوط به فازهای پژوهه، قوانین انجام پژوهه و نحوه نمره‌دهی در **داک کلی راهنمای پژوهه‌ها** نوشته شده. لذا قبل از مطالعه هرگدام از داک‌ها، داک کلی راهنمای پژوهه‌ها را مطالعه نمایید.



## مشخصات پروژه

- عنوان : پلتفرم خرید و فروش شبیه سازی شده از آمازون (Amazon Clone)
- تعداد اعضا : ۳
- هدف : طراحی و توسعه یک پلتفرم تجارت الکترونیک مبتنی بر معماری چندلایه، با پشتیبانی از نقش های خریدار و فروشنده، مدیریت داده های پایدار، سیستم اطلاع رسانی هوشمند، و قابلیت اجرای همزمان و تحت شبکه.



# اهداف کلی

## اهداف پروژه

- احراز هویت و مدیریت کاربران با نقش‌های خریدار (Seller) و فروشنده (Buyer)
- مدیریت محصولات توسط فروشنده‌گان (افزودن، ویرایش، حذف)
- جستجو (جستجو بر اساس دسته‌بندی و جستجوی پیشرفته) و مشاهده محصولات
- افزودن محصول به سبد خرید و ثبت سفارش
- اعلان رسانی به کاربران در مورد تغییر وضعیت سفارش
- ذخیره و بازیابی داده‌ها از طریق فایل (Concurrency)
- پشتیبانی از چند کاربر همزمان (Concurrency)

## شرح و خلاصه پروژه

پروژه «پلتفرم خرید و فروش شبیه‌سازی شده از آمازون» (Amazon Clone) یک پلتفرم تجارت الکترونیک شبیه‌سازی شده از سایت آمازون است. این سامانه امکان مدیریت نقش‌های خریدار و فروشنده، جستجوی پیشرفته محصول، فرآیند کامل خرید از سبد تا سفارش، و سیستم اطلاع‌رسانی خودکار را فراهم می‌کند. در معماری آن از الگوهای طراحی مانند Singleton و Observer استفاده شده و داده‌ها به صورت پایدار در فایل ذخیره می‌شوند. در نهایت، سیستم به صورت یک برنامه کلاینت-سرور چندنخی توسعه می‌یابد که از اتصال همزمان چندین کاربر پشتیبانی می‌کند.



# پروژه

## تحلیل نیازمندی ها

### جدول ۱. نیازمندی های کاربردی (Functional Requirements)

کد	نیازمندی	تحلیل پیاده سازی
FR-01	ثبت نام کاربر جدید	سیستم باید بتواند یک کاربر جدید را با نقش "خریدار" یا "فروشنده" بر اساس اطلاعات اولیه (مانند نام کاربری، رمز عبور، ایمیل) ثبت نام کند. یک User پایه ایجاد شده و بسته به نقش، یک شیء Buyer یا Seller از آن ارث بری می‌کند.
FR-02	ورود به سیستم	کاربر باید بتواند با استفاده از نام کاربری و رمز عبور معتبر وارد سیستم شود. سیستم باید نقش کاربر را تشخیص داده و دسترسی مناسب را فراهم کند.
FR-03	مدیریت پروفایل کاربر	کاربران (هر دو نقش) باید بتوانند اطلاعات پروفایل خود (مانند آدرس، شماره تلفن) را مشاهده و ویرایش کنند.
FR-04	افزودن محصول جدید	فروشنده باید بتواند یک محصول جدید با مشخصات (عنوان، توضیحات، قیمت، تعداد موجودی، دسته بندی) به سیستم اضافه کند.
FR-05	ویرایش و حذف محصولات	فروشنده باید بتواند مشخصات محصولات خود از جمله قیمت و موجودی را ویرایش کند یا از لیست محصولات حذف کند.
FR-06	سفارش های دریافت شده	فروشنده باید بتواند لیست سفارش های دریافت شده را ببیند و محصول را ارسال کند.
FR-07	مشاهده کاتالوگ محصولات خریدار	مشاهده کاتالوگ محصولات خریدار باید بتواند لیست تمام محصولات موجود را به همراه اطلاعات پایه (نام، قیمت) مشاهده کند.
FR-08	جستجو و فیلتر محصولات	خریدار باید بتواند محصولات را بر اساس معیارهای مختلف (مانند عنوان، نام فروشنده، محدوده قیمت، دسته بندی) جستجو و فیلتر کند.
FR-09	افزودن محصول به سبد خرید	خریدار باید بتواند یک محصول را به سبد خرید خود اضافه کند. سیستم باید بررسی کند که محصول در انبار موجود است.
FR-10	مشاهده و مدیریت سبد خرید	خریدار باید بتواند محتويات سبد خرید خود (محصولات، تعداد، قیمت کل) را مشاهده، تعداد آیتم ها را تغییر یا آیتمی را حذف کند.
FR-11	ثبت سفارش	خریدار باید بتواند پس از بررسی نهایی سبد خرید، سفارش خود را نهایی کند. با ثبت سفارش، موجودی محصولات مربوطه در انبار باید کاهش یابد و یک شیء Order جدید ایجاد شود.



کد	نیازمندی	تحلیل پیاده سازی
FR-12	مشاهده تاریخچه سفارش‌ها خریدار باید بتواند تاریخچه و وضعیت تمام سفارش‌های قبلی خود را مشاهده کند.	
FR-13	اتصال کلاینت به سرور	برنامه کلاینت باید بتواند از طریق شبکه به برنامه سرور متصل شود.
FR-14	مدیریت درخواست‌های همزمان	سرور باید بتواند با استفاده از Thread‌ها، درخواست‌های چندین کلاینت را به صورت همزمان پردازش کند.
FR-15	مدیریت موجودی همروند	سیستم باید با استفاده از مکانیزم‌های هماهنگی مانند Lock synchronized یا "رقابت" (Race Condition) هنگام خرید همزمان یک محصول با موجودی محدود توسط چندین خریدار جلوگیری کند.
FR-16	ذخیره داده‌ها	سیستم باید بتواند وضعیت فعلی تمام داده‌ها (کاربران، محصولات، سفارش‌ها) را در قالب فایل (با استفاده از Serialization یا فرمات متنی مانند JSON/CSV) روی دیسک ذخیره کند.
FR-17	بازیابی داده‌ها	هنگام راه‌اندازی مجدد برنامه، سیستم باید بتواند داده‌های ذخیره‌شده را از فایل بخواند و state برنامه را بازیابی کند. از الگوی Singleton برای مدیریت متمرکز این عملیات استفاده می‌شود.

## جدول ۲. نیازمندی‌های غیرکاربردی (Non-Functional Requirements)

کد	نیازمندی	تحلیل پیاده سازی
NFR-01	کارایی (Performance)	اطمینان از سرعت مناسب سیستم در پردازش عملیات‌های مختلف با اقداماتی نظیر کاهش زمان پاسخگویی به کمتر از ۲ ثانیه برای کلیه عملیات‌ها، بهینه‌سازی دیتابیس و...
NFR-02	قابل اطمینان (Realability)	جلوگیری از از دست رفتن داده‌ها و عملکرد پایدار سیستم با اقداماتی نظیر سیستم Backup خودکار، مدیریت خط با try-catch و...
NFR-03	امنیت (Security)	کنترل دسترسی کاربران و محافظت از اطلاعات حساس با اقداماتی نظیر احراز هویت کاربران، سطوح دسترسی بر اساس نقش (مدیر، تکنسین، بازرس) و...
NFR-04	قابلیت نگهداری (Maintainability)	طراحی سیستم به گونه‌ای که امکان توسعه و تغییرآسان وجود داشته باشد با اقداماتی نظیر معماری لایه‌ای، مستندسازی کد، استفاده از Design Patterns و...
NFR-05	قابلیت گسترش (Scalability)	امکان افزودن قابلیت‌های جدید بدون تغییر در ساختار اصلی با اقداماتی نظیر طراحی ماژولار، امکان افزودن انواع جدید خودرو و قطعات و...
NFR-06	قابلیت استفاده (Usability)	طراحی رابط کاربری ساده و قابل درک برای کاربران نهایی با اقداماتی نظیر تدوین رابط کاربری ساده، راهنمای استفاده، مدیریت خطاها کاربرپسند و...



## طراحی و ساختار فنی

### جدول ۳. استفاده از مفاهیم OOP

توضیح	مفهوموم
طراحی کلاس‌های مجزا برای موجودیت‌های اصلی: User, Buyer, Seller, Product, Category, ShoppingCart, Order, OrderItem	کلاس‌ها و شی‌گرایی
تعریف کلاس پایه User و ارث بری Buyer و Seller از آن. امکان افزودن کلاس پایه AbstractEntity برای فیلدهای مشترک مانند id و createdAt	Inheritance
متدهای login() و displayDashboard() با رفتار متفاوت برای Seller و Buyer. امکانOverride متد toString() در کلاس‌های مختلف.	Polymorphism
تعیین سطوح دسترسی private برای فیلدها و استفاده از getter/setter برای کنترل دسترسی و اعتبارسنجی.	Encapsulation
استفاده از ShoppingCart نیز از OrderItem است. OrderItem شامل لیستی از Order و تعداد است Product خود شامل یک OrderItem می‌کند.	Composition
تعریف اینترفیس‌های مانند Notifiable با متد update() برای پیاده‌سازی الگوی Observer برای موجودیت‌های قابل ذخیره در فایل Storable یا	Interfaces

توجه: الزامی بودن نمره «کلاس‌ها و شی‌گرایی» به این معنی است که دریافت نمره نهایی ملزم به قبول شدن در این بخش است. به عنوان مثال، اگر نمره این بخش ۵٪ شود، باید بخش «کلاس‌ها و شی‌گرایی» را انجام داده باشد.

### جدول ۴. نمودارهای مورد نیاز

نمودار	شرح	نمره
Use Case Diagram	نمایش کامل کلاس‌های سیستم، فیلدها، متدها و مهم‌تر از همه، روابط بین آن‌ها (Association, Aggregation و Composition).	%۳
Class Diagram	نمایش ساختار کلاس‌ها و روابط ارث بری و ترکیب.	%۷
Sequence Diagram	نمایش ترتیب و تبادل پیام‌ها در سناریوهای اصلی، مانند ثبت سفارش یا افزودن محصول به سبد خرید.	%۳
State Diagram	نمایش وضعیت‌های سفارش (اضافه شده به سبد خرید، پرداخت موفق، در انتظار ارسال، ارسال شده)	%۲



## تقسیم‌بندی نمره

درصد	شرح	بخش
%۷۴	هر نیازمندی کاربردی ۴٪ و هر نیازمندی غیرکاربردی ۱٪ (کسب ۴٪ برای نمره کامل این بخش کافیست)	پیاده‌سازی نیازمندی‌ها
%۱۰	مطابق جدول ۳	اصول OOP و ساختار فنی
%۱۵	مطابق جدول ۴	طراحی نمودارها
%۱۵	نوشتن سناریو تست، تست واحد، اجرای بدون خطای	تست و اعتبارسنجی
%۲۰	ارائه در جلسه‌ی نهایی کلاس، توضیح عملکرد، نوشتن مستند Word یا PDF مناسب	مستندسازی و ارائه

توجه: طراحی گرافیکی و UI جزو مباحث اصلی درس نیست. با این حال، برای افرادی که قصد پیاده‌سازی پروژه گرافیکی دارند نمره امتیازی در نظر گفته می‌شود. به این صورت که ۱۰٪ از نمره پروژه (۰.۸٪ از ۲۰٪) بطور اضافی و جدا از نمره‌های امتیازی دیگر محاسبه می‌شود.



## چک لیست تحويل نهایی

● نمودارها

● کد برنامه با توضیحات

● فایل اجرایی

● گزارش تست و خروجی

● ارائه نهایی